



Desenvolvimento de um Sistema de Monitorização de Vibração de Baixo Custo

Mestrado em Engenharia Civil - Construções Cívicas

Nuno Pedro Clemente Albuquerque

Leiria, setembro de 2019



Desenvolvimento de um Sistema de Monitorização de Vibração de Baixo Custo

Mestrado em Engenharia Civil - Construções Civas

Nuno Pedro Clemente Albuquerque

Dissertação realizada sob a orientação do Professor Hugo Filipe Pinheiro Rodrigues

Leiria, setembro de 2019

Originalidade e Direitos de Autor

A presente dissertação de projeto é original, elaborado unicamente para este fim, tendo sido devidamente citados todos os autores cujos estudos e publicações contribuíram para a/o elaborar.

Reproduções parciais deste documento serão autorizadas na condição de que seja mencionado o Autor e feita referência ao ciclo de estudos no âmbito do qual o mesmo foi realizado, a saber, Curso de Mestrado em Engenharia Civil - Construções Civas, no ano letivo 2018/2019, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, Portugal, e, bem assim, à data das provas públicas que visaram a avaliação destes trabalhos.

Agradecimentos

Quero agradecer sucintamente aos meus pais pelo todo o apoio prestado e paciência que tiveram durante o desenvolvimento deste trabalho.

Agradeço também ao professor Hugo Rodrigues pela prestimosa ajuda na orientação deste projeto.

O meu muitíssimo obrigado.

Resumo

A conjuntura económica atualmente vivida afeta de igual forma vários setores da vida económica tanto a nível nacional como global.

Os setores de atividade associados á gestão, segurança, manutenção e reparação de estruturas, empregam muitos recursos humanos, materiais e equipamentos para manter essas mesmas em bom estado funcionamento. Contudo têm-se vindo a registar a nível mundial alguns problemas com estruturas e infraestruturas devido à carência de recursos, o que tem vindo a motivar o interesse e desenvolvimento de novas tecnologias capazes de lidar com estes e outros problemas.

Uma dessas tecnologias é a *Structural Health Monitoring* (SHM) ou Monitorização da Integridade Estrutural. Fundamental é um sistema que emprega múltiplas disciplinas com intuito de melhorar performance de confiabilidade das estruturas.

Na base do sistema SHM estão os sensores instalados sobre as estruturas, necessários para medir e monitorizar as grandezas físicas, assim obtendo dados que são processados, analisados e arquivados para análises posteriores.

Neste trabalho desenvolveu-se um dispositivo de monitorização e aquisição de vibrações com base em sensores produzidos com tecnologia *Micro Electro Mechanical Systems* (MEMS) ou sistemas micro eletromecânicos, de baixo custo, sensores usados têm a capacidade de medir acelerações.

O dispositivo foi desenvolvido no conceito do faça-você-mesmo e em código-aberto, oferecendo a possibilidade de modificar, adicionar, alterar componentes e programação, por parte de terceiros interessados em desenvolver outras ou mais capacidades deste dispositivo e sistema.

Palavras-chave: Monitorização de vibrações, *Structural Health Monitoring* (SHM), *Micro Electric Mechanical Systems* (MEMS), *Arduino*, *Do-It-Yourself* (DIY), *Open-Source*.

Abstract

The current economic environment equally affects various sectors of economic life, nationally and globally.

The areas associated with the management, safety, maintenance and repair of structures employ many human resources, materials and equipment to keep them in good working order. However, problems with structures and infrastructures have been occurring worldwide due to lack of resources, which has motivated the interest and development of new technologies capable of dealing with these and other problems.

One of such technology's is the Structural Health Monitoring (SHM). In its essence is a system that employs multiple disciplines to improve the reliability performance of structures. At the base of the SHM system are the sensors installed on the structures, necessary to measure and monitor the physical quantities, thus obtaining data that is processed, analyzed and archived for further analysis.

In this work was developed a device for vibration monitoring and data acquisition based on sensors produced with low cost Micro Electro Mechanical Systems (MEMS) technology, the sensors use have the ability to measure accelerations.

The device was developed in the concept of do-it-yourself and open source, offering the possibility of modifying, adding, altering components and programming by third parties interested in developing other or more capabilities of this device and system.

Keywords: Vibration Monitoring, *Structural Health Monitoring (SHM)*, *Micro Electric Mechanical Systems (MEMS)*, *Arduino*, *Do-It-Yourself (DIY)*, *Open-Source*.

Índice

| | |
|--|-------------|
| Originalidade e Direitos de Autor | iii |
| Agradecimentos..... | iv |
| Resumo | v |
| Abstract..... | vi |
| Lista de Figuras | xii |
| Lista de tabelas..... | xiv |
| Lista de gráficos | xvi |
| Lista de siglas e acrónimos | xvii |
| 1. Introdução | 1 |
| 1.1. Enquadramento geral | 1 |
| 1.2. Objetivos da dissertação | 2 |
| 1.3. Organização da dissertação..... | 3 |
| 2. Sctrutural Health Monitoring SHM | 5 |
| 2.1. Introdução | 5 |
| 2.2. Structural Health Monitoring SHM | 6 |
| 2.2.1. Metodologia..... | 6 |
| 2.2.2. Componentes do sistema SHM | 7 |
| 2.2.3. Categorias do sistema SHM | 8 |
| 2.2.4. Classificação do sistema SHM | 9 |
| 2.3. Sensores de deteção de vibração..... | 9 |
| 2.3.1. Classificação de sensores | 10 |
| 2.3.2. Tipos de sensores de vibração | 11 |
| 2.3.3. Sensores de vibração | 12 |
| 2.3.3.1. Sensores piezoelétrico | 13 |
| 2.3.3.2. Sensores piezoresistivos | 14 |
| 2.3.3.3. Sensores capacitivos..... | 14 |
| 2.3.3.4. Servo Force balance | 15 |

| | | |
|-------------|---|-----------|
| 2.3.4. | Principais desenvolvimentos de sensores..... | 16 |
| 2.3.4.1. | Fiber Bragg Grating..... | 16 |
| 2.3.4.2. | Micro Electric Mechanical System (MEMS)..... | 19 |
| 3. | Seleção de sensor vibrações..... | 21 |
| 3.1. | Introdução..... | 21 |
| 3.2. | Processo de seleção do sensor inercial MEMS..... | 23 |
| 3.2.1. | Pesquisa e recolha de informação sobre MEMS..... | 24 |
| 3.2.2. | Definição de critérios de seleção para comparação de sensores MEMS..... | 25 |
| 3.2.2.1. | Saída de sinal analógico/digital..... | 25 |
| 3.2.2.2. | Número de eixos..... | 28 |
| 3.2.2.3. | Escala dinâmica (<i>Dynamically selectable full scale</i>)..... | 28 |
| 3.2.2.4. | Resolução..... | 29 |
| 3.2.2.5. | Resolução da conversão de sinal digital (ADC)..... | 30 |
| 3.2.2.6. | Ruído..... | 33 |
| 3.2.2.7. | Taxas de amostragem e largura de banda..... | 34 |
| 3.2.2.8. | Interface de comunicação..... | 37 |
| 3.2.2.9. | Disponibilização em formato <i>breakout board</i> | 41 |
| 3.2.2.10. | Custo..... | 42 |
| 3.2.2.11. | Disponibilidade de bibliotecas em função da API adotada..... | 42 |
| 3.2.3. | Avaliação comparativa de sensores inerciais MEMS..... | 43 |
| 3.2.4. | Aquisição de sensores..... | 47 |
| 4. | Protótipo de aquisição de vibrações..... | 50 |
| 4.1. | Introdução..... | 50 |
| 4.1.1. | Arduino..... | 50 |
| 4.1.2. | Especificações do sistema..... | 53 |
| 4.2. | Componentes e montagem do protótipo 1..... | 55 |
| 4.2.1. | Placa de desenvolvimento - Microcontrolador..... | 55 |
| 4.2.2. | Fonte de energia..... | 57 |
| 4.2.3. | <i>Bi-Directional Logic Level Converter</i> | 58 |
| 4.2.4. | Leitor de cartões micro SD e cartão micro SD..... | 59 |
| 4.2.5. | Suporte físico de montagem de componentes..... | 59 |
| 4.2.6. | Jumpers..... | 60 |
| 4.2.7. | Diagrama de ligações e montagem do Dispositivo 1..... | 61 |

| | |
|---|-----------|
| 4.3. IDE Arduino | 63 |
| 4.3.1. Instalação da IDE | 64 |
| 4.3.2. Instalação da placa de desenvolvimento | 65 |
| 4.3.3. Bibliotecas | 65 |
| 4.3.4. Contributos | 66 |
| 4.3.5. Atualizações do IDE | 67 |
| 4.4. Desenvolvimento de Sketch ou Programa | 68 |
| 4.4.1. Definições..... | 70 |
| 4.4.2. Teste de avaliação dos Sketchs | 71 |
| 4.5. Solução Joy Worrier 24F14 - Sensor BMA 180 | 74 |
| 4.5.1. Avaliação do dispositivo Joy Warrior 24F14 | 75 |
| 5. Ensaio de avaliação de sensores..... | 77 |
| 5.1. Introdução | 77 |
| 5.2. Ensaio Estático..... | 78 |
| 5.2.1. Objetivos | 79 |
| 5.2.2. Procedimento de ensaio estático..... | 79 |
| 5.2.3. Constituição do dispositivo e sistema | 80 |
| 5.2.4. Local de ensaio | 80 |
| 5.2.5. Pós-processamento..... | 80 |
| 5.2.6. Apresentação de resultados | 81 |
| 5.2.7. Conclusões sobre o ensaio estático | 81 |
| 5.3. Ensaio Dinâmico | 83 |
| 5.3.1. Introdução | 83 |
| 5.3.2. Objetivos | 84 |
| 5.3.3. Sistema pendular gravítico | 84 |
| 5.3.4. Composição do sistema | 85 |
| 5.3.5. Desenvolvimento do ensaio..... | 87 |
| 5.3.6. Verificação de conformidade dos resultados do ensaio | 88 |
| 5.3.7. Pós-processamento de dados | 89 |
| 5.3.8. Regularização dos dados numa nova taxa amostragem..... | 89 |
| 5.3.9. Normalização dos vetores X e Y..... | 91 |
| 5.3.10. Transformação de dados pelo método <i>Fast Fourier Transformation</i> (FFT) ... | 93 |
| 5.3.11. Execução de gráfico com histórico da amplitude de Fourier | 94 |

| | | |
|-------------|---|------------|
| 5.3.12. | Apresentação de resultados..... | 95 |
| 5.3.13. | Conclusões sobre os resultados do ensaio dinâmico | 100 |
| 5.4. | Conclusão das avaliações dos sensores MEMS | 103 |
| 6. | Protótipo de aquisição de vibrações e monitorização de vibrações | 104 |
| 6.1. | Introdução | 104 |
| 6.2. | Especificações do sistema | 105 |
| 6.3. | Aquisição peças, procedimento de montagem do protótipo 2..... | 106 |
| 6.3.1. | Placa de desenvolvimento – Microcontrolador | 106 |
| 6.3.2. | Leitor de cartões micro SD e Relógio digital com calendário | 107 |
| 6.3.3. | Fonte de alimentação..... | 108 |
| 6.3.4. | Caixa de Proteção..... | 110 |
| 6.3.5. | Caixa de Proteção..... | 110 |
| 6.3.6. | Sensor de temperatura | 112 |
| 6.3.7. | Digrama de ligações e montagem do Dispositivo 2 | 113 |
| 6.4. | Desenvolvimento de Sketch (Programa)..... | 115 |
| 6.4.1. | Introdução..... | 115 |
| 6.4.2. | Algoritmo de <i>Trigger action</i> | 115 |
| 6.4.3. | Relógio e calendário - <i>Real Time Clock</i> RTC | 121 |
| 6.4.4. | Termómetro | 121 |
| 6.4.5. | Bibliotecas..... | 121 |
| 6.4.6. | <i>Sketch</i> | 123 |
| 6.4.7. | Teste de avaliação do <i>Sketch</i> | 128 |
| 7. | Ensaio de validação do dispositivo 2..... | 129 |
| 7.1. | Introdução | 129 |
| 7.2. | Comparação de especificações dos dispositivos | 129 |
| 7.3. | Estruturas de ensaio | 131 |
| 7.3.1. | Miradouro..... | 131 |
| 7.3.2. | Ponte Pedonal | 132 |
| 7.4. | Caracterização dos ensaios..... | 133 |
| 7.4.1. | Miradouro..... | 133 |

| | | |
|-------------|--|------------|
| 7.4.2. | Ponte Pedonal..... | 134 |
| 7.5. | Apresentação de resultados | 135 |
| 7.5.1. | Resultados do Miradouro..... | 136 |
| 7.5.2. | Resultados da Ponte Pedonal | 140 |
| 7.6. | Comparação de resultados | 144 |
| 8. | Avaliação de performance..... | 146 |
| 8.1. | Introdução | 146 |
| 8.2. | Objetivos..... | 146 |
| 8.3. | Definições do <i>Trigger action</i> | 146 |
| 8.4. | Local de aplicação | 147 |
| 8.5. | Eventos sísmicos..... | 148 |
| 8.6. | Resultados..... | 149 |
| 8.7. | Conclusões do ensaio de performance..... | 150 |
| 9. | Caso prático – Mosteiro da Batalha..... | 151 |
| 9.1. | Introdução | 151 |
| 9.2. | Desenvolvimento dos trabalhos | 152 |
| 9.3. | Resultados..... | 153 |
| 10. | Conclusões ou Conclusão | 157 |
| | Referências Bibliográficas | 158 |
| | Anexos..... | 162 |
| | ANEXO A1 – Instruções de montagem dispositivo 1 | 163 |
| | ANEXO A2 - Instruções de montagem dispositivo 2..... | 170 |
| | ANEXO B1 – <i>Sketchs</i> para o dispositivo 1..... | 189 |
| | ANEXO B2 – <i>Sketch</i> para o dispositivo 2..... | 209 |

Lista de Figuras

| | |
|---|----|
| Figura 2.1 - Esquema visual de sistema típico SHM [1] | 7 |
| Figura 2.2 - Componentes de sistema SHM [1] | 8 |
| Figura 2.3 - Categorias do sistema SHM [1] | 9 |
| Figura 2.4 - Conceito de sensor FBG [3]..... | 16 |
| Figura 2.5 - Conceito de um sensor FBG [3]..... | 18 |
| Figura 2.6 - Diagrama esquemático de FBG, para deteção de micro vibrações [10]..... | 19 |
| Figura 2.7 - Estrutura de um sensor acelerómetro capacitivo MEMS [3] | 20 |
| Figura 3.1 - Top 30 de produtores de MEMS, por mercados. [13] | 22 |
| Figura 3.2 - Relação entre largura de banda pela faixa de trabalho..... | 29 |
| Figura 3.3 - Representação visual do fenómeno <i>aliasing</i> | 35 |
| Figura 3.4 - Reconstrução de sinal para diferentes taxas amostragem | 36 |
| Figura 3.5 - Esquemas de configuração por protocolo de comunicação | 40 |
| Figura 3.6 - Sensor MPU6050, InvenSense..... | 41 |
| Figura 3.7 - Breakout Board ITG/MPU | 41 |
| Figura 3.10 - Sensor MPU6050..... | 49 |
| Figura 3.8 Sensor LIS331HH | 49 |
| Figura 3.9 - Sensor MMA8451Q | 49 |
| Figura 3.1 - Diagrama de bloco do Dispositivo 1 | 55 |
| Figura 4.2 - Arduino UNO R3 | 57 |
| Figura 4.3 - Fonte de alimentação interna | 58 |
| Figura 4.5 - Pormenor de Logic Level Converter | 58 |
| Figura 4.4 - Logic Level Converter | 58 |
| Figura 4.7 - Cartão micro SD..... | 59 |
| Figura 4.6 - Módulo de cartão micro SD | 59 |
| Figura 4.8 - Breadboard de 400 terminais..... | 60 |
| Figura 4.9 - <i>Jumpers</i> 10cm e 20cm..... | 60 |
| Figura 4.10 - Digrama de ligações e montagem do Dispositivo 1 | 62 |

| | |
|---|-----|
| Figura 4.11 - Dispositivo 1 com o sensor LIS331HH | 63 |
| Figura 4.12 - IDE do Arduino..... | 64 |
| Figura 4.13 - Mensagem de aviso sobre atualizações disponíveis para IDE Arduino..... | 68 |
| Figura 4.14 - Fluxograma de operação do <i>sketch</i> do Dispositivo 1..... | 70 |
| Figura 4.15 - Joy Warrier – 24F14, com o sensor BMA180..... | 74 |
| Figura 4.16 - Solução COTS da Joy Warrier, dispositivo de aquisição de vibrações..... | 74 |
| Figura 5.1 - Sistema pendular gravítico | 85 |
| Figura 5.2 - Embalagem com o Dispositivo 1 e diafragma com sensor..... | 86 |
| Figura 5.3 - Movimentos verificados um pêndulo livre não controlado..... | 92 |
| Figura 5.4 - Programa Seismosignal..... | 94 |
| Figura 5.4 - Representação do movimento pendular em tempo/aceleração do eixo Z..... | 101 |
| Figura 6.1 - Placa de desenvolvimento Adafruit Feather M0 Basic Proto [30] | 106 |
| Figura 6.2 - Adafruit Adalogger FeatherWing [30] | 108 |
| Figura 6.3 - Bateria de lítio 3,7V 500mAh e botão <i>click-clak</i> instalado..... | 110 |
| Figura 6.4 - Caixa ABS..... | 111 |
| Figura 6.5 - Placa em acrílico para fixação de equipamento..... | 112 |
| Figura 6.6 - Bucins desmontado | 112 |
| Figura 6.7 - Sensor DS18B20 | 113 |
| Figura 6.8 - Digrama de ligações e montagem do Dispositivo 2 | 114 |
| Figura 6.7 - Fluxograma de operação do <i>sketch</i> do Dispositivo 2..... | 127 |
| Figura 7.1 - Dispositivo 2 e dispositivo padrão | 131 |
| Figura 7.2 - Miradouro..... | 132 |
| Figura 7.3 - Ponte Pedonal | 133 |
| Figura 7.4 - Dispositivo 2 (esquerda), Dispositivo Padrão (centro) sobre o miradouro..... | 134 |
| Figura 7.5 - Dispositivo 2 (esq.), Dispositivo Padrão (centro) sobre a ponte pedonal..... | 135 |
| Figura 8.1 - Localização do Dispositivo 2, cave no laboratório de estruturas EC da ESTG | 148 |
| Figura 8.1 - Planta do Mosteiro da Batalha [36]do Mosteiro da Batalha [36] | 151 |

Lista de tabelas

| | |
|--|-----|
| Tabela 2.1 - Especificações técnicas de sensores | 11 |
| Tabela 2.2 - Tipos de sensores por tipologia | 12 |
| Tabela 3.1 – Cálculo de sensibilidade ADC | 32 |
| Tabela 3.2 - Seleção provisória de sensores inerciais MEMS capacitivos | 44 |
| Tabela 3.3 - Distribuição de linguagens de programação pelos sensores inerciais MEMS capacitivos | 45 |
| Tabela 3.4 - Informação de escalas dinâmicas, ODR dos sensores inerciais MEMS | 46 |
| Tabela 3.5 - Seleção final de sensores inerciais MEMS | 48 |
| Tabela 4.1 - Especificações da placa de desenvolvimento Arduino UNO R3 | 56 |
| Tabela 4.2 - Bibliotecas para componentes eletrônicos em linguagem Arduino | 66 |
| Tabela 5.1 - Resultados dos ensaios estáticos | 81 |
| Tabela 5.2 - Normalização de resultados dos ensaios estáticos | 82 |
| Tabela 5.4 - Base de dados regularizada | 91 |
| Tabela 5.3 - Base de dados do Dispositivo 1 | 91 |
| Tabela 5.5 - Tabela de variação de amplitude Fourier ao longo do ensaio - Ensaio 1 | 95 |
| Tabela 5.6 - Tabela de variação de amplitude Fourier ao longo do ensaio - Ensaio 2 | 98 |
| Tabela 6.1 - Especificações da placa Adafruit, Feather M0 Basic, Proto Cortex M0 | 107 |
| Tabela 6.2 - Biblioteca para sensor de temperatura NPX em linguagem Arduino | 121 |
| Tabela 7.1 - Valores de aceleração de pico, do ensaio feito no Miradouro | 138 |
| Tabela 7.2 - Relação entre valores de pico dos dispositivos 2 e padrão, ensaio do Miradouro.... | 139 |
| Tabela 7.3 - Valores de frequências e amplitude, obtidos do eixo X, no Miradouro | 139 |
| Tabela 7.4 - Valores de frequências e amplitude, obtidos do eixo Y, no Miradouro | 140 |
| Tabela 7.5 - Valores de aceleração de pico, do ensaio feito na Ponte Pedonal | 142 |
| Tabela 7.6 - Relação entre valores de pico dos dispos. 2 e padrão, ensaio na Ponte Pedonal ... | 143 |
| Tabela 7.7 - Valores de frequências e amplitude, obtidos do eixo Y, na Ponte Pedonal | 143 |
| Tabela 7.8 - Valores de frequências e amplitude, obtidos do eixo Z, na Ponte Pedonal | 144 |
| Tabela 8.1 - Quadro resumo da atividade sísmica | 149 |
| Tabela 8.2 - Resultados obtidos do ensaio performance..... | 149 |

| | |
|---|-----|
| Tabela 9.1 - Resultados obtidos da monitorização do Mosteiro da Batalha..... | 153 |
| Tabela 9.2 - Resultados obtidos da monitorização do Mosteiro da Batalha, por dia..... | 156 |

Lista de gráficos

| | |
|--|-----|
| Gráfico 5.1 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaio 1 – Eixo X..... | 96 |
| Gráfico 5.2 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaio 1 – Eixo Y..... | 96 |
| Gráfico 5.3 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaio 1 – Eixo Z..... | 97 |
| Gráfico 5.4 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaio 1 – Eixo XY | 97 |
| Gráfico 5.5 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaio 2 – Eixo X..... | 98 |
| Gráfico 5.6 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaio 2 – Eixo Y..... | 99 |
| Gráfico 5.7 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaio 2 – Eixo Z..... | 99 |
| Gráfico 5.8 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaio 2 – Eixo XY ... | 100 |
| Gráfico 7.1 - Tempo/aceleração ensaio sobre Miradouro, eixo X..... | 137 |
| Gráfico 7.2 - Tempo/aceleração ensaio sobre Miradouro, eixo Y..... | 138 |
| Gráfico 7.3 - Esp. freq. eixo X disp. 2, ensaio Miradouro..... | 139 |
| Gráfico 7.4 - Esp. freq. eixo X disp. padrão, ensaio Miradouro | 139 |
| Gráfico 7.5 - Esp. freq. eixo Y disp. 2, ensaio Miradouro..... | 140 |
| Gráfico 7.6 - Esp. freq. eixo Y disp. padrão, ensaio Miradouro | 140 |
| Gráfico 7.7 - Tempo/aceleração ensaio sobre Ponte Pedonal, eixo Y. | 141 |
| Gráfico 7.8 - Tempo/aceleração ensaio sobre Ponte Pedonal, eixo Z. | 142 |
| Gráfico 7.9 - Esp. freq. eixo Y disp. 2, ensaio Ponte Pedonal | 143 |
| Gráfico 7.10 - Esp. freq. eixo Y disp. padrão, ensaio Ponte Pedonal..... | 143 |
| Gráfico 7.11 - Esp. freq. eixo Z disp. 2, ensaio Ponte Pedonal | 144 |
| Gráfico 7.12 - Esp. freq. eixo Z disp. padrão, ensaio Ponte Pedonal | 144 |
| Gráfico 9.1 - Tempo/aceleração de evento a 17/09/2019 08:51, Mosteiro da Batalha eixo X | 154 |
| Gráfico 9.2 - Tempo/aceleração de evento a 17/09/2019 08:51, Mosteiro da Batalha eixo Y | 155 |
| Gráfico 9.3 - Espectro de freq. eixo X, Mosteiro da Batalha evento a 17/08/2017 08:51 | 155 |
| Gráfico 9.3 - Espectro de freq. eixo Y, Mosteiro da Batalha evento a 17/08/2017 08:51 | 156 |

Lista de siglas e acrónimos

| | |
|------|---|
| B2C | <i>Business to Customer</i> |
| C2C | <i>Customer to Customer</i> |
| Cots | <i>Commercial off-the-shelf</i> |
| DIY | <i>Do-It-Yourself</i> |
| FBG | <i>Fiber Bragg Grating</i> |
| FFT | <i>Fast Fourier Transformation</i> |
| IDE | <i>Integrated Development Environment</i> |
| IoE | <i>Internet Of Everythings</i> |
| IoT | <i>Internet Of Things</i> |
| MEMS | <i>Micro Electric Mechanical System</i> |
| NDE | <i>Non-Destructive Evaluation</i> |
| NDT | <i>Non-Destructive Testing</i> |
| SHM | <i>Scrutural Health Monitoring</i> |

| | |
|------------------|-----------------------------|
| A | Corrente elétrica |
| AC | Corrente alternada |
| °C | Temperatura (graus Celsius) |
| DC | Corrente contínua |
| g | Aceleração da gravidade |
| Hz | Frequência |
| Kg | Massa |
| m | Metro |
| m/s ² | Aceleração |
| N | Peso (Newton) |
| s | Segundo |
| T | Período |
| V | Volt |

1. Introdução

1.1. Enquadramento geral

Reduções dos recursos económicos, materiais e humanos dificultam as operações de manutenção, inspeção, controlo e reparação do património em geral quer no setor privado como público. As estruturas inerentes precisam de uma manutenção regular, continuada para que o seu desempenho seja adequado á função pretendida. A carência de manutenção impele a aceleração da degradação, podendo levar a incapacitação da estrutura e em situações extremas ao colapso, em risco estão vidas humanas, riscos ambientais, impactos económicos e sociais severos, etc.

Neste paradigma novas soluções estão a ser investigadas e desenvolvidas, produzindo novas ferramentas que oferecem aos decisores com informação de melhor qualidade, com maior rapidez, oferecendo a escolha entre uma visão local ou global sobre um problema ou situação, garantias sobre o estado funcionamento e desempenho, alertas de potenciais perigos e situações, redução de custos de monitorização/inspeção e manutenção, entre outras mais soluções.

Das várias soluções emergentes o *Structural Health Monitoring* (SHM) ou Monitorização da Integridade Estrutural procura dar resposta a estes e outros problemas. No que concerne ao setor da engenharia civil e muitos dos seus ramos da SHM encontra-se no campo académico, com exceção da sua aplicação em alguns casos concretos como é o caso de pontes com grandes vãos, já se verificando resultados satisfatórios.

A SHM é um sistema que se implementa sobre uma estrutura com o objetivo de avaliar a sua segurança, integridade e performance e identificar de danos. Esta avaliação é feita com base no tratamento e análise de dados recolhido de um sistema sensorial aplicado sobre a estrutura, de uma forma periódica ou continua.

1.2. Objetivos da dissertação

Esta dissertação tem como primeiro objetivo o estudo e a conceção de um sistema de medição de vibrações utilizando materiais e equipamentos de baixo custo e validação dos resultados obtidos para aplicação em estruturas de engenharia civil.

O segundo objetivo é adequar este dispositivo para ser implementado como parte dum sistema de monitorização estrutural a num nível mais elementar, segmentado para a monitorização contínua de vibrações ambientais, com vista a avaliar alterações das propriedades dinâmicas (frequências naturais, modos de vibração).

O conceito subjacente á execução deste projeto é utilização do método autodidata *Do It Yourself* (DIY) ou faça-você-mesmo. A base deste conceito está a procura individual de conhecimentos e materiais necessários à execução de um projeto ou problema.

A validação deste dispositivo foi realizada através de vários ensaios de controlo com recurso a métodos simples, de comparação de resultados.

Este sistema tem o intuito de integrar, num programa em curso dedicado ao estudo do estado de conservação e evolução de patologias de uma estrutura emblemática da cultura Portuguesa. A estrutura é o Mosteiro de Santa Maria da Vitória também conhecido por Mosteiro da Batalha, património da cultura Portuguesa e da Humanidade.

Sobre esta estrutura já se realizaram vários estudos, ensaios, modelos, projeto. Numa progressão natural dos trabalhos já executados sobre esta estrutura, pretende iniciar um programa de monitorização contínua, para controlar a evolução de danos detetados e identificar novos danos.

Outro objetivo deste trabalho é dar um primeiro passo para futuros trabalhos, dentro do conceito de monitorização contínua. A versatilidade das tecnologias *low cost* e conceitos empregues por estas, possibilitam sua expansão e evolução dos projectos. Estas permitem, a integração de várias tecnologias por módulos, a contante evolução sobre um a mesma base de trabalho. Também estão constantemente a surgir novos sensores, mais eficientes no mercado.

1.3. Organização da dissertação

A presente encontra-se organizada em nove capítulos, separando a várias etapas de desenvolvimento do sistema de monitorização de vibrações de baixo custo.

O presente capítulo e referente a apresentação do tema estudado, com uma pequena introdução, objetivos da dissertação.

O segundo capítulo é apresentado a área da engenharia que se dedica ao desenvolvimento onde este tema se contextualiza.

O terceiro capítulo é apresentado resultados de pesquisa de informação sobre sensores de vibrações, parâmetros de comparação e apresenta-se uma avaliação comparativa entre sensores encontrados durante a pesquisa com base nos parâmetros verificados.

O quarto capítulo descreve todo o processo de desenvolvimento de um protótipo de aquisição de vibrações.

No quinto capítulo são relatados ensaios as sensores adquiridos e ao dispositivo de aquisição de vibrações. São apresentados resultados dos ensaios com o objetivo de selecionar um sensor específico a utilizar no sistema de monitorização de vibrações.

No sexto capítulo é descrito o desenvolvido de um protótipo final, com capacidade de monitorização em tempo real, onde é aplicado o sensor com as melhores características.

O sétimo capítulo são descritos ensaios de campo em estrutura de construção, de avaliação comparativa entre o dispositivo final desenvolvido e um outro com características de alta performance, usado em experiências laboratoriais.

O oitavo capítulo é descrito um ensaios de performance do dispositivo de monitorização desenvolvido em ambiente controlado assim estabelecendo uma *baseline*, são apresentados resultados e conclusões obtidos desses ensaios.

O novo capítulo relata a aplicação do sistema num caso prático. O local escolhido foi o Mosteiro da Batalha, e durante um período de tempo este sistema monitorizou a estrutura, com o propósito de obter dados e analisa-los.

O decimo capítulo relata as principais conclusões obtidas durante o desenvolvimento do sistema.

2. Sctrutural Health Monitoring SHM

2.1. Introdução

As redes infraestruturais são sinónimo do nível desenvolvimento de uma civilização, como por exemplo, da extinta civilização romana ainda existem se conservam parcialmente intactas, pontes, aquedutos, entre outras obras de arte, algumas ainda em funcionamento. As infraestruturas antigas e modernas precisam de manutenção continuada para manter um ótimo desempenho, uma boa prestação de qualidade de serviço, longevidade, etc, caso contrário estas entram em colapso e ficam foram de serviço.

A necessidade de avaliar pontualmente ou globalmente uma estrutura, sobre o seu estado, condições de serviço, integridade estrutural, ou outro tipo de informação relevante para a sua gestão de operação, originou o aparecimento de várias tecnologias e disciplinas, apoiadas na utilização de vários tipos de equipamentos e instrumentação, técnicas e sistemas, com intuito de providenciar informação fidedigna que contribua para avaliação das condições de serviço dessa estrutura.

As tecnologias existentes subdividem-se em várias áreas de intervenção, mas há um fator importante que as divide em dois grupos, esse é o período de avaliação, que pode ser curta duração e aí inserem-se as avaliações pontuais e as de longa duração e aí inserem-se as avaliações contínuas. As avaliações de curta duração são comparativamente de mais execução e gastam menos recursos dos que as longa duração.

No caso das avaliações pontuais encontramos as tecnologias designadas por *Non-Destructive Testing* (NDT) ou Teste Não-Destrutivos ou *Non-Destructive Evaluation* (NDE) Avaliações Não-Destrutivas [1], enquanto a SHM foca-se na monitorização contínua da estrutura.

A SHM é assim o conceito mais amplamente discutido e desenvolvido no que concerne a avaliação, monitorização da segurança de estruturas, um conceito aplicável em vários sectores da engenharia com é o caso da engenharia aeroespacial, automóvel, civil, entre outros.

2.2. Structural Health Monitoring SHM

O *Structural Health Monitoring* (SHM) ou Monitorização da Integridade Estrutural, é um processo de implementação de uma estratégia de identificação de danos, aplicável a vários tipos de estruturas de engenharia. [2]

Que utiliza métodos de avaliação não-destrutivo *in situ*, por observação contínua de uma estrutura ou sistema mecânico, coletando dados ou informação com recurso a vários tipos de sensores fixados ou incorporados na estrutura. [1] Os dados ou informação são recolhidos de forma em tempo real *livestream*, ou por amostragem em períodos regulares ao longo do tempo.

Após recolha de dados são realizadas análises, com recursos a vários métodos e técnicas, como é caso análise estatística, assim procurando obter padrões de comportamento, caracteriza-los.

A cada comportamento distinto é caracterizado e analisado, procurando determinar se está ou não associado a dano. Sabendo que, um dano estrutural causa uma grande variação no comportamento normal de uma estrutural, mas também, ter a atenção que uma grande variação no comportamento nem sempre é causado por um dano, como é caso do aumento de massa por sobrecargas. Assim toda a informação é armazenada e referenciada para que seja possível fazer futuros diagnósticos e interpretações.

O propósito desta avaliação é determinar a segurança, estado de integridade do sistema, a capacidade ou performance da estrutura bem como identificar danos, em tempo real ou quase real.

A estratégia a longo prazo deste processo obter um histórico da evolução da capacidade estrutural, tendo em conta a função para a qual foi destinada, e a inevitável acumulação de danos resultantes do ambiente operacional. [2]

2.2.1. Metodologia

Um sistema ideal SHM deve providenciar informação, sobre pedido sobre a saúde estrutural e também deve ter a capacidade de detetar danos e emitir alertas. Para

desenvolver um sistema SHM são necessárias várias disciplinas que contribuem com conhecimentos nas áreas dos materiais, estruturas, deteção de danos, sensores, gestão e dados e processamento inteligente de dados, computadores e comunicação. Os componentes do sistema SHM são descritos no próximo capítulo.

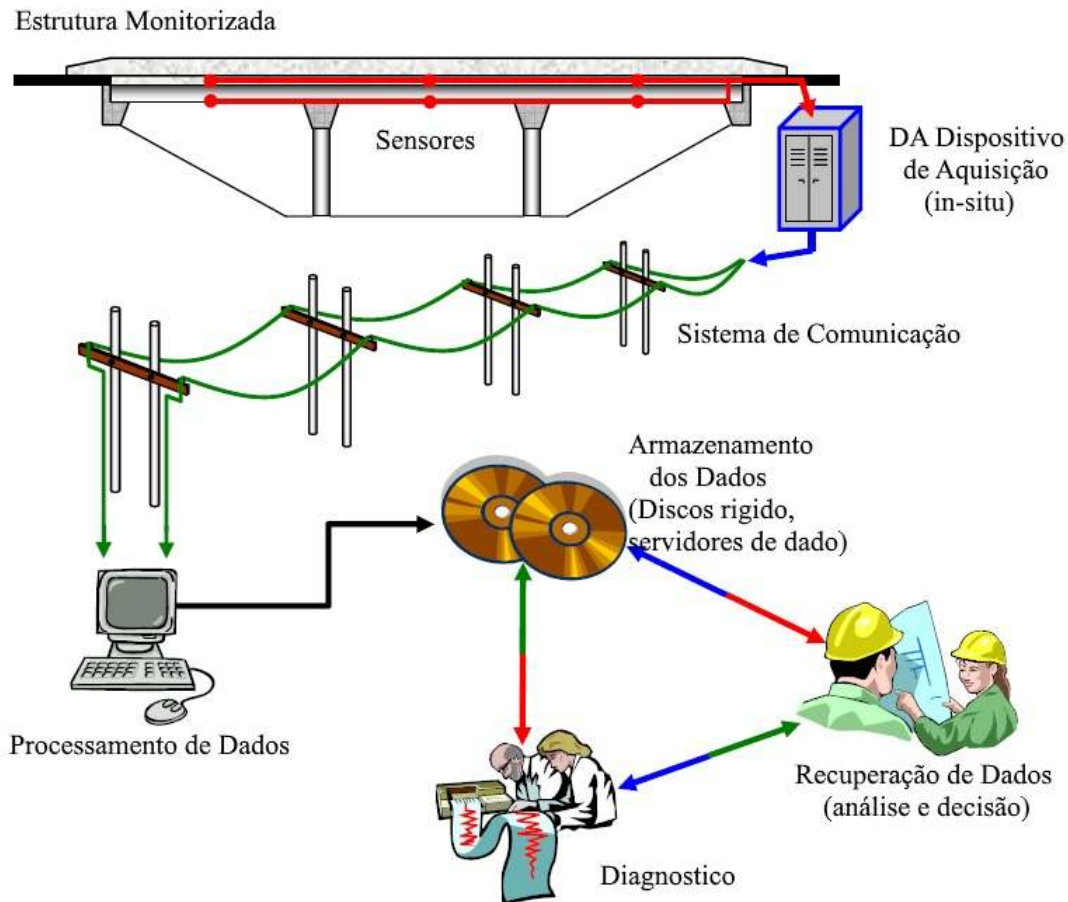


Figura 2.1 - Esquema visual de sistema típico SHM [1]

2.2.2. Componentes do sistema SHM

Um sistema típico SHM tem seis componentes, interligados por uma linha orientadora, onde a informação flui entre cada componente, a informação pode fluir numa direção apenas ou ser bidirecional, os componentes são:

- Aquisição de dados (sistema sensorial);
- Comunicação de dados;
- Processamento inteligente;
- Armazenamento dos dados processados;
- Diagnóstico;

- Recuperação de dados.

O padrão apresentado na figura 2.2 é um modelo possível de um sistema SHM, poderão existir outros válidos.

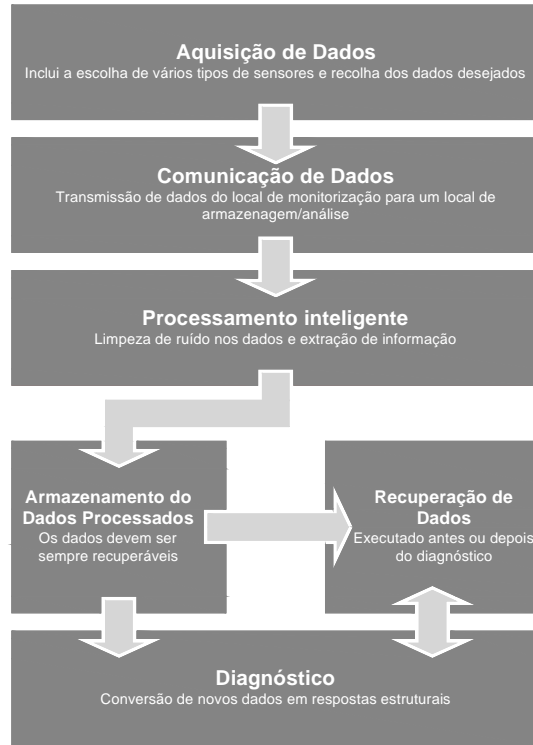


Figura 2.2 - Componentes de sistema SHM [1]

2.2.3. Categorias do sistema SHM

O sistema SHM está classificado em quatro categorias. As distinções entre cada categoria são feitas por critérios. O primeiro critério é relativo a obtenção dos dados, se são obtidos em contínuo ou por amostragem pontual.

O segundo critério é o tipo de teste a executar, e em função define-se da forma como os dados são coletados. A razão para haver esta distinção deve-se a natureza das ações que se pretendem medir, por exemplo, um ensaio onde pretende avaliar características dinâmicas de uma estrutura em que o impulso de ativação é uma explosão só podem ser realizada esporadicamente, devidos aos riscos que a realização do ensaio acarreta.

Para ilustrar as várias categorias e subcategorias apresenta-se abaixo a figura 2.3.



Figura 2.3 - Categorias do sistema SHM [1]

2.2.4. Classificação do sistema SHM

A classificação de um sistema é feita de acordo com o seu nível de sofisticação, tipo de informação processada e fornecida, assim temos quatro níveis:

- **Nível I** – No nível básico o sistema é capaz de detetar dano na estrutura, mas não é capaz providenciar a informação a cerca da natureza, localização, severidade, etc, do dano, também não é capaz de avaliar a segurança da estrutura.
- **Nível II** – Neste nível o sistema é capaz de detetar a presença de dano e também é capaz de identificar a sua localização aproximada.
- **Nível III** – A este nível o sistema já é capaz de detetar e fornecer a localização pormenorizada do dano e estimar a severidade.
- **Nível IV** – Este nível o sistema atinge um nível máximo de sofisticação, capaz de fornecer informação detalha a cerca da presença, localização e severidade do dano, sendo também capaz de avaliar o estado de segurança da estrutura com base na informação recolhida.

2.3. Sensores de deteção de vibração

Não existem limitações na escolha do tipo de sensor em SHM de estruturas de engenharia civil. As especificações técnicas do sensor escolhido deve cumprir os requisitos definidos na fase de concepção de projeto de um sistema de monitorização. Os critérios gerais de seleção de um sensor são precisão, confiabilidade, limitações de instalação, consumo energético e fonte de receção, transmissão do sinal de informação, durabilidade ou tempo de vida e custo do sensor [1]. Para além do custo associado a aquisição do sensor, á que

ter em conta o custo de desenvolvimento do restante sistema em sua função, neste caso a rede transmissão de informação e energia, sistema de aquisição e processamento da informação, entre outros elementos.

2.3.1. Classificação de sensores

Existem várias formas de classificar sensores, algumas simples outras complexas, que dependem de critérios de classificação. Uma possível forma prática de distinção de sensores é analisado os seguintes aspetos, a fonte de energia, o tipo de referência usada e as propriedades [3].

A fonte de alimentação de um sensor define se este é passivo ou ativo. Os sensores passivos não necessitam de uma fonte alimentação elétrica externa, pois gera a sua própria corrente elétrica em resposta ao estímulo externo, essa corrente é então convertida num sinal de saída. Um exemplo deste tipo de sensor é piezoelétrico.

Os sensores ativos necessitam de um fonte de alimenta elétrica externa para poder operar, que que designa por excitação de sinal. O estímulo externo altera as propriedades intrínsecas do sensor e conseqüentemente modifica por modulação o sinal de saída, essa modelação transporta a informação que é interpretada num valor. Um exemplo deste tipo de sensor é extensómetro.

A referência é usada para distinguir os sensores entre absolutos e relativos. Os sensores com escalas absolutas detetam o estímulo em referência a um escala física absoluta independentemente das condições de medição. Quanto aos sensores com escalas relativas produz um sinal relativo a um caso especial. [3]

Por fim, as propriedades e características dos sensores podem ser usadas para distingui-los, ordena-los e classifica-los de várias formas. Abaixo é apresentado um quadro com diferentes de especificações que se podem verificar em sensores.

Tabela 2.1 - Especificações técnicas de sensores

| | |
|------------------------------------|--------------------------------|
| Sensibilidade | Gama de leitura de um estímulo |
| Estabilidade a curto e longo tempo | Resolução |
| Precisão | Seletividade |
| Velocidade de resposta | Condições ambientais |
| Sobrecarga de características | Linearidade |
| Histerese | Banda morta <i>Dead band</i> |
| Tempo de vida em operação | Formato de saída do sinal |
| Custo, dimensões e peso | Outros |

2.3.2. Tipos de sensores de vibração

SHM define vários tipos de informação e dados que podem ser recolhidas com a monitorização de uma estrutura de engenharia civil com sensores, as principais são; carregamentos; deformações; tensões; temperatura; acelerações; pressão; velocidade do vento; emissões acústicas; imagem de vídeo. Abaixo são apresentadas vários quadros, com vários tipos de sensores divididos por tipologia, género, materiais de construção, etc.

Tabela 2.2 - Tipos de sensores por tipologia

| Presença, deslocamento e nível | Velocidade e aceleração |
|---|---------------------------------|
| Sensores potenciométricos | Sensores de velocidade fixo |
| Sensores piezoresistivos | Velocidade linear |
| Sensores capacitivos | Sensores de velocidade angular |
| Sensores indutivos e magnéticos | Sensores inercial de rotação |
| LVDT e RVDT | Giroscópio de rotor |
| Sensores indutivos transverso | Giroscópio vibratório |
| Sondas de corrente Eddy | Giroscópio ótico laser |
| <i>Pavements loops</i> | Sensores inercial lineares |
| Detetores de metal | Inclinómetros |
| Sensores de efeito-Hall | Sensores sísmicos |
| Sensores magneto resistivos | Acelerómetros capacitivos |
| Detetores magnetorestitivo | Acelerómetros piezoresistivos |
| Sensores Óticos | Acelerómetros piezoelétrico |
| Ponte ótica | Acelerómetros térmicos |
| Detetor de proximidade com polarização de luz | Closed-Loop accelerometer |
| Sensores prismáticos e refletivos | |
| Sensores Fabry-Perot | |
| Sensores de fibra ótica | |
| <i>Fiber bragg grating</i> | |
| <i>Grating photomodulator</i> | |
| Sensores de espessura e nível | Força e tensão |
| <i>Ablation sensors</i> | Extensómetro |
| Sensores de filme | Filmes sensíveis a pressão |
| <i>Sensores de nível líquido criogénico</i> | Sensores de força piezoelétrico |
| | Cabos piezoelétricos |
| | Sensores de força óticos |

2.3.3. Sensores de vibração

A vibração é um fenómeno mecânico, físico, podendo ser descrito por, um sistema que se movimenta em torno de ponto central ou ponto de equilíbrio. Assim um sistema que vibra, tem deslocamento relativo que pode ser medido ao longo de um intervalo de tempo, sofre

deformações, tensões e forças, todas estas podem ser lidas e medidas. Sendo um oscilatório ou repetitivo, tem um padrão associado que pode ser interpretado e analisado. Um sistema em equilíbrio não sofre de vibrações, estas só podem ocorrer graças a estímulos externos, neste caso forças externas que excitam esse sistema.

Os sensores possíveis de serem utilizados para medir vibrações, agrupam em três grupos, os primeiros são utilizados para medir, presença, deslocamentos e nivelamento, os segundos grupo velocidade e aceleração, finalmente o terceiro grupo de sensores são usados para medir forças e tensões. A tabela abaixo apresenta uma lista de tipos sensores agrupado por conjunto.

Os sensores mais usados para medir vibrações, em estruturas de construção civil são: Acelerómetros Piezoelétricos; Acelerómetros Piezoresistivos; Acelerómetros Capacitivos; *Servo force balance accelerometer*, fibra ótica tipo *Fiber Bragg Grating (FBG)*.

2.3.3.1. Sensores piezoelétrico

Os sensores piezoelétricos são construídos com um elemento, que tem a propriedade de produzir uma corrente elétrica, quando sujeito a uma ação externa, como por exemplo a compressão axial. Sendo carga elétrica produzida pelo elemento piezoelétrico, proporcional a ação exercida sobre este. O elemento piezoelétrico é um cristal, sobre o qual é aplicado uma massa, quando combinados, o conjunto forma um aparato que funciona similarmente a um sistema massa-mola [4]. O sensor é constituído pelo dito elemento piezoelétrico, uma massa sísmica, uma base e um invólucro ou caixa que protege todo o sistema. Existem três *design* estruturais elementares para este tipo de sensor, esse *design* varia em função do tipo de comportamento a estudar, como é o caso da flexão, compressão, esforço transversal, etc.

Estes são de pequenas dimensões, dispensam alimentação externa pois geram a sua própria energia elétrica a partir da energia mecânica, possuem boa estabilidade e durabilidade em monitorizações de longa duração, são praticamente imunes às variações de temperatura e têm um comportamento linear com uma larga gama de frequências [5]. Oferecem uma boa precisão de resultados e por esta razão são muito procurados, a grande desvantagem é que, degradam a longo prazo obrigando a calibrações periódicas.

2.3.3.2. Sensores piezoresistivos

Um pouco à semelhança dos materiais piezoelétricos, os materiais piezoresistivos alteram a sua resistividade elétrica quando sujeitos a uma ação. Neste caso, quando se sujeita este material uma corrente elétrica, obtendo-se o valor da resistência à passagem da corrente. O valor dessa resistência elétrica altera proporcionalmente em função da ação externa aplicada.

A estrutura deste tipo sensor é semelhante aos piezoelétricos, mas com uma disposição diferente. Também possui uma massa sísmica suspensa numa consola, em vez de um elemento sensível é composto por dois, são aplicado diametralmente oposto a consola que antecede a massa sísmica. Este tipo de sensor opera com energia externa, corrente elétrica passa pelos dois elementos sensitivos criando uma ponte, designada por ponte *Weathstone*, a massa ao vibrar desequilibra o sistema, criando um diferencial no circuito elétrico proporcional à aceleração estrutural monitorizada.

Uma grande vantagem deste tipo de sensor apresentam boa sensibilidade e uma boa resposta a solicitações de baixa frequência (chegando até ao nível DC ou 0 Hz), ao contrário do que acontece com os piezoelétricos. Estes são adequados a estruturas flexíveis de Engenharia Civil [6].

2.3.3.3. Sensores capacitivos

Os sensores capacitivos medem a variação da capacitância ou capacidade elétrica, gerada pela movimentação de uma massa sísmica entre dois elétrodos. A estrutura é idêntica ao sensor piezoresistivos, difere-se apenas na localização dos elétrodos, que se encontram junto à massa sísmica. Este tipo de sensor necessita de corrente elétrica para operar, pela passagem da corrente pelos elétrodos, é criada uma meia ponte capacitiva, está é registada pelo sensor continuamente. Aquando uma vibração, a massa sísmica destabiliza o campo magnético criando um diferencial que é proporcionalmente ao valor da ação externa.

Existem vários formatos de sensores capacitivos, a semelhança dos sensores piezoresistivos, apresentam uma boa sensibilidade, boa resposta em baixas frequências e

grande gama de acelerações. São bastante precisos, com boas resoluções, a gama de acelerações pode variar entre os mg aos 100g, em altas taxas de frequência.

2.3.3.4. Servo Force balance

Este tipo de sensor difere dos anteriores por ter uma estrutura interna diferente. Esta permite minimizar erros de não-linearidade causados pela flexão da estrutura de suporte da massa. A determinação dos valores de aceleração é feita em proporcionalmente com o deslocamento da massa sísmica, mas devido ao tipo de estrutura pendular, a massa ao deslocar-se deflete, gerando imprecisões ou erros de não-linearidade. Este tipo de estrutura foi desenvolvido de modo a que a flexão seja mínima. [7]

Outra diferença construtiva deste sensor é de ser de circuito-fechado, em que é utilizado um servo para limitar o movimento da massa, assim controlado a flexão e os erros de não-linearidade.

São utilizadas força magnéticas para manterem a massa na posição nula. Quando a massa se desloca entre campos magnéticos, gera uma corrente elétrica, que o sensor de capacitância lê e então a capacidade elétrica é interpretada numa leitura. A corrente elétrica gerada, por sua vez cria um campo elétrico, que contraria o movimento, repondo a massa a sua posição nula.

A grande vantagem deste tipo de sensores é a sua grande precisão, robustez, uma gama de leitura de acelerações, possui ainda boa resposta a ações dinâmicas até DC (0 Hz), sendo dos mais usados nas aplicações de engenharia civil, nomeadamente em grandes estruturas flexíveis e em estruturas em que os tempos de respostas são grande [5]. A grande desvantagem deste sensor tem um custo até 10 vezes superiores aos referidos [7] e gama limita de frequências disponíveis.

2.3.4. Principais desenvolvimentos de sensores

2.3.4.1. Fiber Bragg Grating

No que concerne especificamente a sensores de vibração, uma tecnologia em franco desenvolvimento, fomentada por investigadores ligados à SHM, são os sensores de fibra ótica *Fiber Bragg Grating* (FBG), capazes de medir deformações, temperatura, inclinações e acelerações, podendo ser empregues em variados tipos de estruturas, como por exemplo pontes, oleodutos e gasodutos, tuneis, fundações, estradas, barragens, etc. Estes têm capacidade de detetar tanto vibrações fortes, bem como vibrações fracas, até mesmo micro vibrações.

O sensor de medição de tensões é desenvolvido a partir de uma fibra ótica de sílica (SiO_2) que é modificada de forma a criar obter estruturas FBG no seu núcleo. Estes sensores usam princípios distintos dos sensores extensómetros tradicionais. As FBG são microestruturas construídos no interior de uma fibra de vidro, estas idênticas as usada em redes de telecomunicações, essas fibra têm apenas alguns milímetros de comprimento. A fibra ótica é modificada com recurso a uma luz laser de *germanium-doped*, que modifica o núcleo em intervalos periódicos, criando microestruturas com diferentes índice de refração. [8]

O mecanismo de funcionamento de uma fibra ótica FBG, é simples, faz-se passar pelo seu interior um feixe de luz com uma gama de espectro definida. As microestruturas *Bragg Grating* com índices de refração diferentes da restante fibra de vidro, refletem uma parte do espectro, no sentido oposto, deixando passar o remanescente espectro. São medidos os dois espetros de resposta, e analisados procurando alterações, assim determinando variações do comportamento, que se relacionem com fenómeno de estudo.

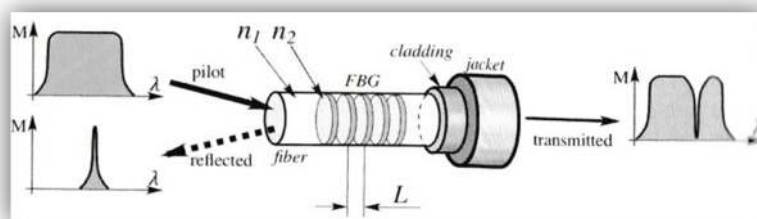


Figura 2.4 - Conceito de sensor FBG [3]

De acordo com o fabricante alemão HBM de sensores baseados em FBG, aponta grandes vantagens na utilização deste tipo de sensores, mas também têm as suas desvantagens [8], que se enumeram de seguida:

- Possuem propriedade semelhantes aos novos materiais, como são o caso dos reforços com fibra de vidro e carbono, podendo se aplicados sobre estes;
- Os FBG podem medir tensões muito elevadas ($>10000 \mu\text{m/m}$), sendo indicados para medir as tensões elevadas suportadas pelas estruturas compósitas;
- Estes são pequenos e leves;
- São imunes as interferências eletromagnéticas e de luz;
- Não necessitam de energia elétrica direta, assim podendo ser aplicados em atmosferas explosivas;
- O sinal podem viajar grandes distâncias ao longo da fibra sem grandes perdas, até 50km;
- Uma fibra ótica pode ligar vários sensores em série, até mais de 20 unidades, com um aparelho especial interrogator podem ser ligados mais de 100 unidades, sem necessidade de fibra de retorno de sinal;
- Possuem uma boa estabilidade ao longo do tempo;
- Boa resistência a corrosão;
- Versões especiais de FBG podem suportar temperaturas superiores a 700°C ;
- Possuem uma condutividade térmica bastante baixa;
- Possuem baixa interação com campos magnéticos;
- São de fácil instalação, baixos tempo de aplicação e de teste.

As desvantagens destes sensores são as seguintes:

- Os resultados mostram grande dependência com a variação da temperatura, que não pode ser ainda compensada, tendo que ser compensada por meio de leitura de dois sensores, instalados de uma forma própria;
- A compensação automática, causa pela expansão térmica, ainda não é possível, é necessário executar medidas de temperatura e fazer correções a posteriori;
- As fibras FBG possuem grande sensibilidade às forças laterais ou pressão causando distorção fazendo-a refratar;
- Limitados valores de calibração;
- Devido a reduzida espessura em termos de diâmetro, é difícil a reparação;

- A elevada rigidez estrutural das fibras óticas, pode ser transferida para o alvo da amostragem;
- O elemento sensor da fibra ótica FBG, posicionado no núcleo, pode devolver leituras com uma certa extremidade, devido ao afastamento do ponto de amostragem, para corrigir esta situação as fibras devem ser posicionados de forma curva.
- O raio da fibra ótica pode ser superior a 10mm, assim as rosetas FBG podem tomar dimensões consideráveis;
- O custo de um dispositivo de compensação especial interrogator possui um custo elevado, que apenas pode ser compensado com grande utilização de sensores, assim obtendo rentabilidade por unidade de sensor instalado.

O sensor de vibração mais comum é o acelerómetro, a construção deste tipo de sensor e outros similares obedece a uma estrutura conceptual clássica. Esta possui uma massa de prova, suspensa por molas, e amortecida. O movimento é medido através de sensores que medem uma propriedade específica, que define o tipo de sensor, todo o sistema está encapsulado dentro de uma estrutura protetora que se fixa ao alvo de medição. Abaixo apresenta-se uma imagem ilustrativa do mecanismo de funcionamento de um acelerómetro.

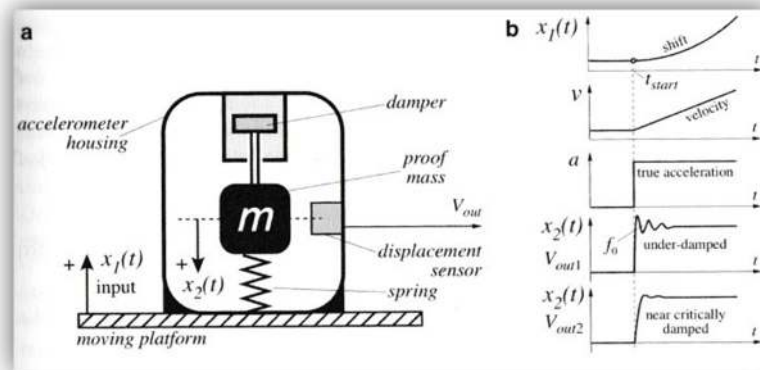


Figura 2.5 - Conceito de um sensor FBG [3]

No trabalho desenvolvido pelos Tianliang Li, Yuegang Tan e Zude Zhou, de um sensor para deteção de micro-vibrações [10], utilizando fibras óticas FBG. Estes criaram um sensor que utiliza a estrutura conceptual de sensor de aceleração com a tecnologia FBG, a figura 2.6 ilustra o esquema que estes usaram para construir o seu sensor.

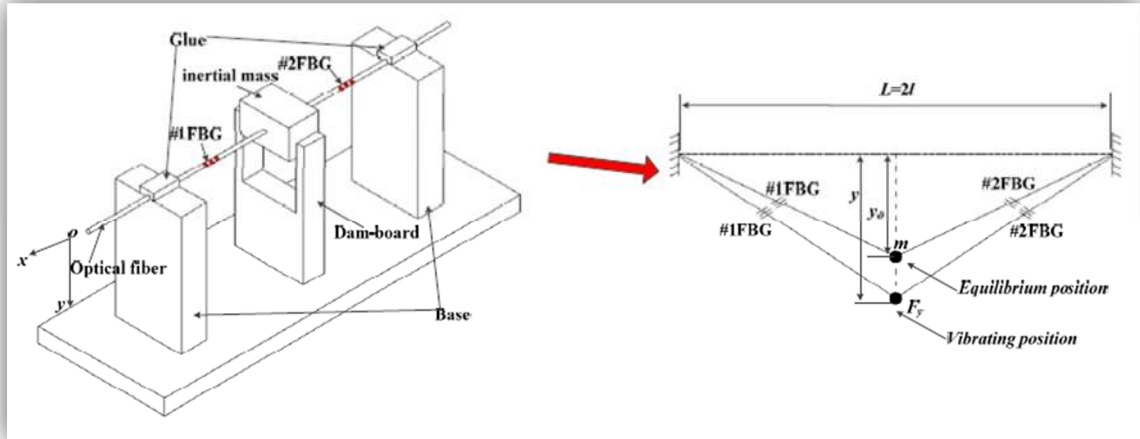


Figura 2.6 - Diagrama esquemático de FBG, para deteção de micro vibrações [10]

2.3.4.2. Micro Electric Mechanical System (MEMS)

MEMS ou sistema micro electro mecânico, são dispositivos que concentram na mesma unidade, de pequena dimensão (em alguns milímetros), vários sistemas combinados de natureza mecânica com sistema de medição elétrico, conversão de sinal analógico para digital, processamento de dados, entre outras possibilidades.

O sensor MEMS de um acelerómetro linear é baseado numa estrutura em silicó, com interdigitações e em forma de pente composta por dedos fixos e móveis. A aceleração é obtida da medição dos deslocamentos de elementos móveis que estão associados aos eixos. O movimento medido pelo sensor é então convertido num sinal analógico ou digital. [11]

O princípio de funcionamento de um sensor de vibração MEMS, é o de um sistema de massa, mola, amortecido. A figura abaixo ilustra a estrutura física, mecânica do sensor.

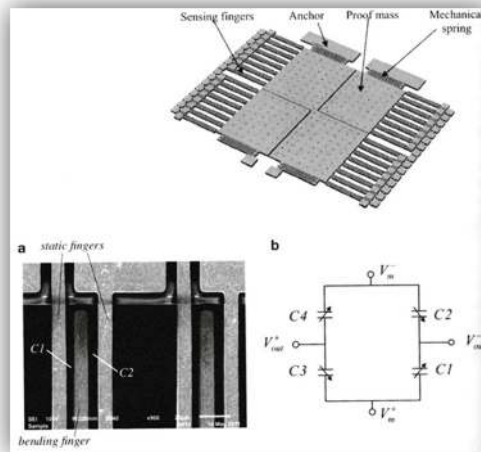


Figura 2.7 - Estrutura de um sensor acelerómetro capacitivo MEMS [3]

As principais vantagens de utilização da utilização de MEMS, são a possibilidade de fazer leituras em três direções distintas, o baixo custo por unidade, disponibilidade, baixo peso e pequenas dimensões, tornando possível aplicação de vários sensores aplicado diretamente na estrutura. São apontados também desvantagens como o fraco desempenho a baixas frequências, um índice de ruído superior ao verificado nos sensores convencionais, limitações ao nível da alimentação. Devido ainda ao grande potencial de desenvolvimento tecnológico, visto que esta está em grande desenvolvimento, as desvantagens apontadas poderão vir a ser ultrapassar num futuro próximo.

3. Seleção de sensor vibrações

3.1. Introdução

No mercado especializado dos *Micro Electric Mechanical System* MEMS pode ser encontrada uma grande variedade de sensores inerciais MEMS, desenvolvidos para vários usos, aplicações, com performances e capacidades distintas. Estes sensores diferenciam-se em dois componentes chaves, o *hardware* e *software*. A primeira componente é a que mais contribui para a performance do sensor, definido pelo tipo de estrutura mecânica usada por cada sensor, componentes usados, como os transdutores, microprocessadores, etc.

Cada sensor MEMS é desenvolvido para uma determinada utilização ou função, nesse âmbito as especificações e funcionalidades são ajustadas para cumprir essa função com uma determinada performance.

Neste momento os fabricantes de sensores MEMS têm a sua produção focada para aplicações ou utilizações que não requerem resultados com grandes exatidão, precisão e performance, estando mais motivados em produzir MEMS com baixo custo, dimensões e consumos elétricos. [12]

A grande parte dos fabricantes de MEMS com fins industriais, projeta e desenvolve os seus produtos com grande versatilidade para que alcancem o maior número de aplicações. O mercado com maior crescimento em termos de volume de vendas, são os dispositivos móveis, como é caso dos telemóveis, *tablets*, *gaming*, *wearables* etc.

Para ter uma ideia do universo de fabricantes de componentes MEMS, é apresentada abaixo a figura 3.1 com os 30 maiores fabricantes, focados na produção de MEMS para o mercado de tablets, telemóveis, em relação com outras áreas de mercado. A informação foi disponibilizada pela i-Micronews em junho de 2014 [13].

A área de mercado ligado a produção de tabletes e telemóveis é que tem vindo a dinamizar o desenvolvimento tecnológico de MEMS e consequentemente baixando os preços de produção.

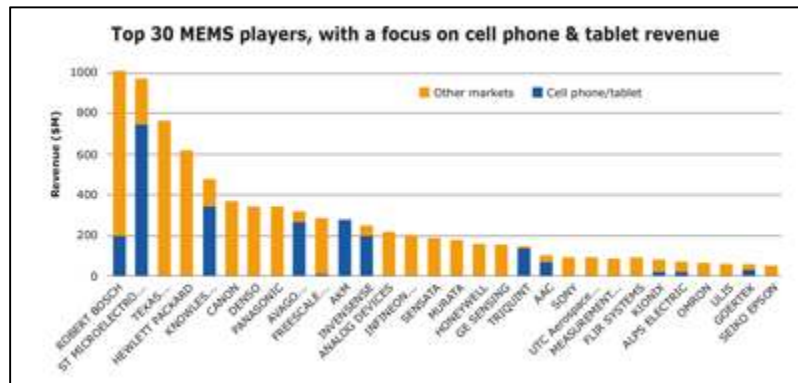


Figura 3.1 - Top 30 de produtores de MEMS, por mercados. [13]

Especificações necessárias para sensores ligados à área da SHM de estruturas de engenharia civil, são bastantes exigentes, devido a ordem de grandeza da acelerações e amplitudes medidas serem muito baixas, quando produzidas pelo ruído ambiental. [14]

Os fabricantes de sensores inerciais MEMS de elevada precisão e performance, aplicáveis à área da SHM representam um pequeno nicho de mercado. Os sensores MEMS produzidos por estes são de elevado custo, situação deve-se ao custo de investimento necessário ao desenvolvimento da elevada performance e precisão de cada MEMS, bem como a sua caracterização metrológica e certificação. O custo unitário é bastante elevado ultrapassando o custo dos acelerómetros tradicionalmente usados na área da SHM de estruturas de engenharia civil.

Os fabricantes de MEMS de baixo custo à data presente, estão a desenvolver as suas próprias soluções, neste momento já lançaram para o mercado os primeiros sensores inerciais vocacionados para monitorização de vibrações, SHM, e representam a novidade, mas ainda estão numa fase embrionária no seu desenvolvimento, não existem ensaios que validem ou comprovem as suas especificações ou o comportamento em aplicações.

De acordo com um estudo metrológico a sensores MEMS de baixo custo, cujas utilizações inicialmente previstas não incluem a área da SHM de estruturas de engenharia civil, este estudo caracterizou-os metrológicamente tendo em vista as especificações necessárias da área da SHM. Conclui que estes podem ser aí aplicados, possuem ótima relação qualidade/custo, visto que estes custam 10 vezes menos que os acelerómetros tradicionais e em termos de performance são comparáveis desde que alguns aspetos fundamentais sejam tidos em conta. [14]

Para obter um bom resultado em termos de performance e precisão de resultados, há que ajustar as características e especificações do sensor MEMS, às especificações do sistema a monitorizar.

É essencial saber *a priori* os parâmetros a medir das grandezas físicas, sabendo essas informações comparam-se com as especificações do sensor, assim adequado um sensor aos parâmetros que se pretendem medir.

Como exemplos das informações necessárias à escolha de um sensores inercial temos, o tipo de ações está o sensor sujeito, se são estáticas ou dinâmicas e quais os valores máximos e mínimos espectados, em que direções se movimenta o objeto de monitorização, quais os valores esperados em termos de intervalo de frequências são do sistema a monitorizar, etc.

Segundo vários investigadores ligados a área SHM, as frequências naturais registadas em estruturas de engenharia civil são relativamente baixas, situando-se abaixo dos 100Hz. [14]

A maioria das vibrações ambientais nas estruturas, caracterizam-se por serem de baixa amplitude e baixa aceleração, os valores encontram-se na ordem dos milig, assim a escala deve-se situar entre $\pm 1g$ a $\pm 2g$.

Estes necessitam de ser bastante sensíveis ao movimento, devem gerar muito pouco ruído para que assim se obtenha uma alta resolução. Numa situação ideal o ruído por um sensor deve-se situar na ordem $1 \mu g/\sqrt{\text{Hz}}$.

3.2. Processo de seleção do sensor inercial MEMS

Estabeleceu-se uma metodologia racional que produzisse informação necessária à escolha de um ou mais sensores inerciais MEMS, assim definiram-se quatro fases para este processo. Consideraram-se algumas especificações já referidas para o sistema de medição e monitorização, de seguida descrevem as fases do processo nomeadamente:

1ª Fase – Pesquisa e recolha de informação sobre sensores MEMS de baixo custo disponíveis no mercado;

2ª Fase – Definição de parâmetros de avaliação e comparação de sensores MEMS;

3ª Fase – Avaliação de sensores inerciais de acordo com os parâmetros estabelecidos;

4ª Fase – Aquisição de sensores a fornecedores selecionados;

3.2.1. Pesquisa e recolha de informação sobre MEMS

O processo de recolha de informação sobre componentes eletrónicos, sensores inerciais MEMS e sistemas operacionais, foi com recurso à World Wide Web (WWW). Aqui encontra-se informação sobre fornecedores de componentes eletrónicos de baixo custo, motores de busca de componentes eletrónicos e os respetivos produtores, plataformas de desenvolvimento de sistemas, etc.

Todos fornecedores encontrados disponibilizam alguma informação sumarizada dos componentes que expõem, alguns destes para além dessa informação básica, disponibilizam também informação detalhada sobre esses componentes, como por exemplo fichas técnicas, tutorias de desenvolvimento de aplicações e equipamentos.

Os motores de pesquisa de componentes eletrónicos utilizados foram Octopart [15] e Digikey [16] reúnem vastas listas de produtos, fornecedores, fichas técnicas e um sumário da informação técnica sobre componentes eletrónicos, foi excelente ponto de partida. Através das suas opções de listagem e filtragem, em função de características comuns aos componentes eletrónicos, como por exemplo, preço, resolução, fabricantes, modelos, consegue-se identificar os produtos mais adequados para as funções desejadas de cada projeto.

Durante esta fase localizou-se inúmeros sensores com características, especificações técnicas e para aplicações técnicas distintas.

Como nota informativa, durante a fase de procura de sensores MEMS de baixo custo, não foi se encontrado nenhum sensor destinado ao uso em SHM, mas durante o desenvolvimento deste trabalho foi colocado no mercado um sensor desenvolvido para aplicações em SHM, *seismic imaging*, entre outras.

3.2.2. Definição de critérios de seleção para comparação de sensores MEMS

No processo de escolha, pretendeu-se distinguir quais dos sensores encontrados, aqueles que apresentam um potencial para o fim que se pretende. Assim fez-se um levantamento dos vários parâmetros comuns indicados nas especificações técnicas, e elegeram-se, aqueles estão relacionados com capacidade de deteção de vibrações. Comparou-se os valores dos parâmetros de cada sensor, assim tentando identificar aqueles com maior potencial, para avaliar comportamento em estruturas de construção civil.

As características e fatores apresentados abaixo são apenas alguns dos possíveis ser analisados, para manter este processo simples foram apenas observados os seguintes:

- Saída de sinal analógico/digital;
- Número de eixos;
- Fundo escala dinâmica (*Dynamically selectable full scale*);
- Resolução;
 - Resolução da conversão de sinal (*analogic to digital converter ADC*);
 - Ruído;
- Taxas de amostragem (*output data rate ODR*);
- Interface de comunicação;
- Disponibilidade em formato placa (*printed circuit board PCB - breakout board*);
- Custo;
- Disponibilidade de bibliotecas em função da Interface de Programação de Aplicações (API) adotado.

3.2.2.1. Saída de sinal analógico/digital

Há que entender quais as vantagens e desvantagem entre utilização de um sinal digital ou analógicos, antes de optar por um dos dois, avaliou-se alguns aspetos importantes entre os dois.

Quanto avaliamos a oferta de sensores nos motores de busca verificamos que aproximadamente 40% dos sensores inercias MEMS apresentados são analógicos, há

partida pode ser uma desvantagem, visto haver menos disponibilidade e variedade de soluções desenvolvidas em cada sensor.

No *site* da Digikey aplicou-se o filtro do ruído na pesquisa de sensores de movimento, e ordenou-se por ordem crescente, verificou-se que o sensor que aparece em primeiro é um sensor analógico, neste caso é o modelo LIS344ALH da ST Microelectronics.

Avaliando os protocolos encontrados, verificou-se que o protocolo mais popular é o SPI, seguido pelo I²C, Análogo e Serial.

A maior vantagem do sinal analógico é possibilidade de enviar o sinal a grandes distâncias, estando limitado apenas pela resistência elétricas dos cabos de distribuição. O protocolo SPI permite a transmissão de dados, teoricamente e com condições particulares, até 100m mas com uma baixa taxa de amostragem. O protocolo I²C apenas permite enviar sinais a 1 a 2 metros de distância.

Em termos custos sensores analógicos de uma maneira geral são mais caros que os sensores digitais, muito devido ao fato de haver menos procurar e serem desenvolvidos para soluções particulares.

Outra situação que se dever ter em conta é o formato de saída dos dados, estes devem ter uma representação para área engenharia civil. As medições devem ser apresentadas ou convertidas para unidades de medida do S.I., neste caso a grandeza avaliada pelos sensores é a aceleração em m/s².

Por norma os sensores analógicos emitem sinais sobre forma de nível de tensão ou Volt, que tem que ser convertido para um formato digital, através de *analog digital converter* ADC assim obtendo informação que pode ser usada compreendida e manipulada.

O sistema de aquisição de dados designado por ADC ou conversor analógico-digital, deve ter a maior sensibilidade possível para que assim se obtenha uma boa resolução. A escolha de um sinal analógico obriga à aquisição de componente eletrónico para lidar com a conversão do sinal, o que encarece o desenvolvimento de qualquer sistema.

No caso dos sensores digitais MEMS estes já veem com ADC incorporados, obtendo registos no formato de números binários, que são por sua vez convertidos em número inteiros por programação específica de cada sensor. Os valores que o ADC fornece, estão

limitados a uma série de níveis, neste caso a sensibilidade do ADC, que estabelece o número de níveis que este pode subdividir, assim quanto maior o nível, maior será a resolução.

Independentemente do sistema de aquisição que se utilize, é necessário verificar a existência de informação e programação sobre um determinado sensor e em função do sistema de aquisição optado. De um ponto de vista informação disponibilizada na rede global verifica-se que existe mais informação sobre a programação de sensores digitais, devido ao facto de estes representarem a maioria.

Ponderando os prós e contras, optou-se neste trabalho pelo uso de sensores com saída digital, devido ao custo mais baixo, a existência de maior informação de programação deste tipo de sensores, não ser necessário adquirir um ADC que obrigaria a necessidade de programar este componente em combinação com o sensor analógico, considerado um acréscimo de dificuldade.

A questão da distância entre o sensor e sistema de aquisição de dados é ultrapassada a partida visto devido a natureza do sistema ser um dispositivo *standalone*, ou autossuficiente, com todos os seus sistemas integrados na mesma unidade.

Para resolver o problema do isolamento do dispositivo ou sistema, surgiu alguns anos um novo conceito a *Internet Of Things* IOT, que está impulsionar o aparecimento de componentes eletrónicos para a transmissão e gestão comunicações *wireless* ou redes sem fio, de baixo custo, fácil programação, permitindo a ligação de dispositivos eletrónicos as redes de bases de dados *online* de uma forma fácil e intuitiva.

Esta inovação vem alterar o paradigma dos sistemas de monitorização. Os sistemas de monitorização tradicionais centralizam todo sistema de aquisição, conversão ADC, armazenamento de dados e gestão energética, num central. Esta solução obriga a ligação de todos os sensores por meio de cabos elétricos ao sistema de aquisição de dados, que trás complicações da aplicação e gestão física da rede, bem como o custo acrescido na sua aquisição e montagem.

Neste momento começam a surgir estruturas, monitorizadas com módulos de instrumentação completamente autónomos, que não utilizam qualquer ligação física a um serviço de aquisição tratamento de dados, energia, etc. Estes módulos são de fácil

construção e aplicação, têm os seus próprios sistemas de comunicação sem fios, e no presente estão ser testados e validados.

3.2.2.2. Número de eixos

Quando se pretende analisar as características dinâmicas de uma estrutura de construção civil, na maioria dos casos é apenas monitorizado um plano. Na maior parte das estruturas o plano é o horizontal, pois é neste que se registam os maiores movimentos, enquanto os movimentos verticais são ignorados.

O número mínimo de eixos que um sensor inercial MEMS deve ser dois, porém existem muitos sensores MEMS com três eixos, sendo essa a norma.

3.2.2.3. Escala dinâmica (*Dynamically selectable full scale*)

Como referido vibrações de origem ambiental com pouca energia produzem em estruturas de construção pequenas acelerações e amplitudes de movimento. Os valores são na ordem 1/1000 da aceleração gravítica (g) ou mili g, assim um sensor inercial deve ter a menor escala possível, para poder detetar as pequenas variações.

Os fabricantes estabelecem as escalas dos sensores em função da utilização prevista, podemos verificar pela figura 3.2, abaixo a relação entre a largura de banda pela faixa de trabalho, exigido para cada uso.

Foi assinalado uma área com um círculo vermelho, que representa os valores das escalas ideais para sensores inerciais, usados em aplicações em SHM. No presente ainda não se encontram disponíveis nos sensores comerciais específicos na área da SHM, assim os sensores mais indicados para este trabalho, são os produzidos para aplicações de navegação inercial.

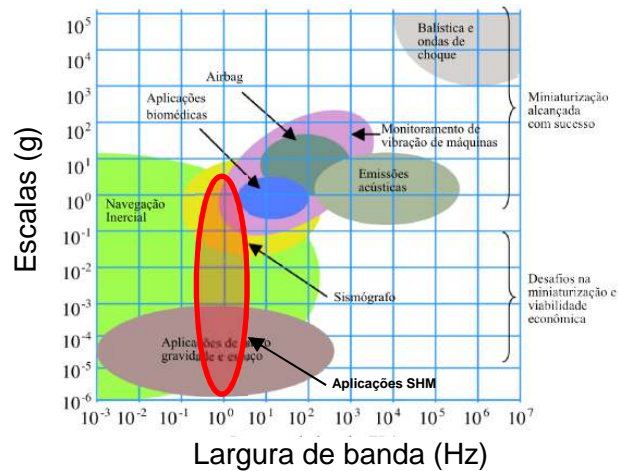


Figura 3.2 - Relação entre largura de banda pela faixa de trabalho

Para obter uma boa precisão nas medições é necessário que o sensor tenha uma boa sensibilidade e que as amplitudes dos movimentos e acelerações se enquadrem dentro da mesma escala de trabalho que o sensor é capaz de medir.

Se os movimentos foram superiores ou inferiores ao que sensor for capaz de registar, essa informação não é registada. A situação oposta também se verifica, mas pela perda de sensibilidade e precisão. Se a escala dinâmica for muito extensa, a resolução é muito baixa e há perda de sensibilidade nos valores registados.

Assim a escala de dinâmica de sensor deve estar bem ajustada aos movimentos esperados a registar. Assim no caso sensores inerciais para aplicação em SHM a escala máxima indicada deve ser $\pm 1g$, mas nunca ultrapassando os $\pm 2g$.

3.2.2.4. Resolução

A resolução é definida como a mais pequena mudança detetável [18]. O aspeto que contribui para a resolução das medições é a sensibilidade do ADC, medida em bits, quanto maior valor de bits maior é a resolução. Em contrapartida há um aspeto que contribui para diminuição da resolução que é o ruído registado pelo sensor.

O ruído de base ou *noise floor* é soma ou acumulação de várias fontes de ruído, tanto de origem internas como é o caso do ruído termomecânico e elétrico, ou de fontes externas como o caso de vibrações de origem sísmica, atividade humana.

A largura de banda optada pode refletir no resultado da resolução obtida. Quanto mais elevada a largura de banda a avaliar, maior é a quantidade de ruído, assim menor a resolução.

É então importante identificar a precisão do sensor, determinada pela sensibilidade do ADC, e caracterizar o sensor quanto à performance de ruído.

3.2.2.5. Resolução da conversão de sinal digital (ADC)

Como referido anteriormente o conversor de sinal analógico para sinal digital ADC, converte proporcionalmente um sinal analógico na forma de nível de tensão (V) ou intensidade de corrente elétrica (A) para um valor digital em magnitude idêntica ao sinal analógico, através de um processo de quantização.

O resultado da conversão de um sinal analógico para digital é um número binário sem expressão aparente da grandeza física que está a ser medida. A resolução do ADC é o número de bits que este consegue converter de um sinal analógico por leitura.

A quantidade de valores discretos que o ADC consegue fornecer é definida por resolução do ADC ou também conhecido por *output* do ADC.

A resolução é definida pelo fabricante do ADC e expressa em bits. Para determinar o número de níveis disponíveis, eleva-se o valor dado da resolução a 2. Por exemplo se um ADC tem 8 bits de resolução o número total de valores discretos ou níveis é $2^8=256$.

Os valores obtidos de ADC de sensores inerciais variam entre valores positivos e negativos, conforme se a aceleração é positiva ou negativa. A quantidade de níveis disponibilizada por um ADC é sempre par. Como existe um valor neutro que representa a ausência de aceleração, os valores da resolução máxima e mínima são obrigatoriamente diferentes. Usado o exemplo anterior de ADC de 8 bits, os valores extremos podem ser apresentados de duas formas distintas (-128; 127) ou (-127; 128).

Para indicar a localização do nível extra num ADC, é mencionada uma das seguintes expressões conforme a situação *Most Significant Bit* MSB ou *Least Significant Bit* LSB. No caso do nível extra estiver mais à esquerda do lado negativo então é mencionado no ADC

MSB. A situação mais encontrada em ADC e serem MSB neste caso para um ADC de 8 bit o valores variam entre -128 e 127.

É importante identificar onde se encontra o menor número significativo, se no lado negativo ou positivo, pois a conversão de bits para acelerações é diferente e os valores máximos também o são.

Para exemplificar essa diferença vamos atribuir uma escala dinâmica de $\pm 2g$ ao exemplo anterior, do ADC de 8 Bits. Lembrando que este tem 256 níveis que se dividem por 2 sentidos, um negativo outro positivo e um neutro, assim obtemos 128 registos negativos e 127 registos positivos e um registo neutro que é o zero. Numa situação em que é fornecido o valor -128 este valor corresponde a -2g, a situação oposta quando é fornecido o valor 127 este corresponde a 1,9873g.

O valor da aceleração máxima positiva foi determinado da seguinte forma, dividiu-se o valor da escala dinâmica, que foi no caso anterior de foi 2g pelo valor máximo absoluto que é 128, então obteve-se o seguinte valor 0,015625g, que define a sensibilidade mínima do ADC para a escala dinâmica seleccionada. Como os valores positivos de aceleração tem menos um nível, então subtraiu-se 0,015625 a 2g, igual a 1,9844g.

A metodologia de cálculo apresentada é aplicada na maioria dos sensores inerciais, contudo pode haver situações divergentes, portanto deve-se consultar sempre as notas informativas de cada fabricante.

A sensibilidade do ADC é definida pelo número de Bits que o ADC possui e pela escala gama seleccionada, visto que um sensor pode possuir várias escalas de gamas.

A sensibilidade de um sensor é alterada pela escala de gama de acelerações escolhida, na maioria dos sensores encontrados no mercado verificou-se que, quanto mais alta for a escala optada menor é sensibilidade dos resultados, e conseqüentemente menor é a resolução.

A sensibilidade e a resolução é definida pela escala dinâmica optada num sensor.

A maior sensibilidade possível num sensor, por norma está atribuída, á menor valor de escalas, esta informação está presente nas fichas de especificações técnicas e deve ser confirmada para cada sensor.

A sensibilidade é expressa em *counts* por g, ou contagem de níveis por g, também pode ser expressa em mili-g por LSB.

Por exemplo, para determinar da sensibilidade de ADC de 14 bits de output, que possui $2^{14}=16384$ níveis e atribuindo uma escala de $\pm 2g$. Primeiro, divide-se o número total de níveis pelo número por dois, obtendo 8192 *counts*. De seguida divide-se pelo número da escala selecionada, nesta situação foi 2g, obtendo 4096 *counts/g*. Se dividirmos 4096 *counts/g* por 1g que é o valor unitário e multiplicarmos por mil obtemos a matematicamente melhor sensibilidade do ADC que é 0,24 mili-g/LSB.

No caso anterior fosse atribuída a resolução máxima à escala dinâmica igual a $\pm 1g$, então a sensibilidade teria o dobro da que na primeira situação.

A abaixo apresenta-se uma tabela com os valores calculados de sensibilidade máximos em função da resolução do sensor em bits e da escala dinâmica. Estão definidos os números máximo de níveis para cada resolução, subdivididos por valores positivos e negativos e por cada escala dinâmica entre 1g a 16g, atribuindo a cada uma dela o valor máximo de sensibilidade para cada ADC.

Tabela 3.1 – Cálculo de sensibilidade ADC

| Resolução do ADC | | | | Sensibilidade | | | | | | |
|------------------|-------------------|---------|---------|---------------------------|----------|----------|----------|----------|----------|----------|
| Resolução do ADC | n.º máx de níveis | MSB | LSB | escala dinâmica do sensor | | | | | | |
| | | - range | + range | 1g | 2g | 4g | 6g | 8g | 12g | 16g |
| [Bit] | [Bit] | [Bit] | [Bit] | [mg/LSB] | [mg/LSB] | [mg/LSB] | [mg/LSB] | [mg/LSB] | [mg/LSB] | [mg/LSB] |
| 8 | 256 | -128 | 127 | 7,87 | 15,75 | 31,50 | 47,24 | 62,99 | 94,49 | 125,98 |
| 10 | 1024 | -512 | 511 | 1,96 | 3,91 | 7,83 | 11,74 | 15,66 | 23,48 | 31,31 |
| 12 | 4096 | -2048 | 2047 | 0,49 | 0,98 | 1,95 | 2,93 | 3,91 | 5,86 | 7,82 |
| 14 | 16384 | -8192 | 8191 | 0,12 | 0,24 | 0,49 | 0,73 | 0,98 | 1,47 | 1,95 |
| 16 | 65536 | -32768 | 32767 | 0,03 | 0,06 | 0,12 | 0,18 | 0,24 | 0,37 | 0,49 |

Erros de quantização

Um valor medido num sensor analógico e convertido digitalmente só pode tomar um número finito hipóteses, no processo de conversão ocorrem erros de quantização. [19] O bit que representa a menor alteração detetada é designado por LSB, e a cada quantização existe um erro que varia entre $\pm 0,5$ do LSB devido a diferença entre o valor do sinal analógico e o valor do sinal digital. Deve-se determinar o erro de quantização, comparando-

o com os restantes erros, para verificar se é ou não desprezável. Quanto maior a resolução e sensibilidade do sensor, este erro torna-se cada vez menor até chegar a um determinado ponto que se considera desprezável.

3.2.2.6. Ruído

O ruído interfere com a resolução e precisão das medições, introduzindo erros e indefinições de natureza aleatória as medições.

Existem duas fontes de ruído, uma de origem elétrica proveniente de flutuações elétricas do fornecimento de energia e outra de origem mecânica de proveniente de fontes de vibrações internas ou externas sobre o sensor. Quando se refere ao ruído elétrico, refere-se ao somatório de várias fontes de ruído, não é especificado ou discriminado todas as situações possíveis. Quanto ao ruído mecânico é proveniente de fontes termomecânicas e vibrações ambientais. [20]

O ruído base de um sensor afeta a resolução, introduzindo uma quantidade de incerteza nas medições. O valor do ruído é condicionado pela largura de banda optada, que aumenta com o valor da largura de banda.

A determinação do valor de ruído interno de sensor é apresentado nas especificações técnicas. Este diz respeito a densidade espectral ou *Power Spectral Density* (PSD) que não inclui o ruído relativo à quantização produzida durante a conversão do sinal pelo ADC. O valor apresentado do ruído (N) é expresso em mili-g pela raiz quadrada da largura de banda (BW), como apresentado na Equação 1.

$$N = \frac{\mu g}{\sqrt{BW}} \quad (1)$$

Especificações do ruído (R) apresentadas nas fichas técnicas referem-se sempre á melhor performance do ruído pelo sensor, que depende de configurações específicas, como é caso da situação referida largura de banda, que por norma é definida com valores baixos taxa de amostragem, também são definidos filtros de passagem com limites baixos, para que assim seja bloqueado o ruído proveniente das frequências altas, obtendo um valor baixo de ruído.

É possível estimar o nível de ruído de um sensor sabendo o valor de referência do ruído (R) e a largura de banda utilizada, expressa na Equação 2

$$R_{RMS} = N \times \sqrt{BW} \quad (2)$$

São apresentadas várias fórmulas de cálculo do ruído na nota de aplicação apresentada pelo fabricante NXP. [19]

Se por exemplo um sensor possuir um ruído de $150 \mu\text{g}/\sqrt{BW}$ ou $0,150 \text{ mg}/\sqrt{BW}$, para uma largura de banda e taxa de amostragem a 1200Hz. Para determinar o ruído com uma taxa de amostragem de 250Hz então $0,150 \times \sqrt{250} = 2,37 \text{ mg}$. Se a este valor multiplicarmos por valor padrão entre picos ou pk-pk [14] de referência igual a 4, nesta situação obtemos um ruído de 9,48mg.

Na escolha de sensor inercial, interessa verificar o ruído (R) de cada um e optar pelos que tenham o menor valor possível. Não esquecendo de apontar as condições em que foi determinado, neste caso a taxa de amostragem, a largura de banda, a filtros de passagem escolhidos, entre outro fatores definidos pelo fabricante.

3.2.2.7. Taxas de amostragem e largura de banda

A taxa de amostragem ou *output data rate* ODR é a quantidade de medições feita a cada segundo, por um sensor expressa em Hertz (Hz). A largura de banda define limites à taxa de amostragem, entre várias razões, estas limitações asseguram que as frequências detetadas são as corretas e que não sejam o resultado de fenómenos de distorção conhecido por *Aliasing*. [21]

Cada sensor está projetado para operar dentro de uma gama de frequências, assim têm características mecânicas únicas que os otimizam nesse sentido, por forma a minimizar interferências com as frequências dos elementos a medir, assim garantindo uma boa resolução e precisão nas medições.

Cada sensor está projetado para operar dentro de uma gama de frequências, assim têm características mecânicas únicas que os otimizam nesse sentido, por forma a minimizar interferências com as frequências dos elementos a medir, assim garantindo uma boa resolução e precisão nas medições.

Como referido informação produzida pelo movimento é de natureza analógica, infinita e contínua no tempo, o tratamento e processamento deste tipo de informação é muito pouco provável devido à finita capacidade de processamento dos sistemas existentes. Por esta razão a informação processada é obtida num número finito de registos.

Para executar análises estruturais com base nos dados recolhidos por sensores, executam-se pós-processamentos de dados recolhidos em ensaios, em que os sinais originais que são reconstruindo *software* e convertidos nos domínios do tempo e frequência. A reconstrução de um sinal com base em registos discretos insuficientes, origina em vários problemas, nomeadamente, erros de distorções, desvios, etc.

Um dos erros mais significativo é o de distorção conhecido por *Aliasing*, resulta do insuficiente número de registos para reconstruir um sinal analógico original.

As ilustrações apresentadas na figura 3.3, demonstram o fenómeno referido, através de diferentes reconstruções de um sinal analógico, quando usando diferentes taxas de amostragem. Os gráficos apresentados encontram-se tanto no domínio do tempo/amplitude e frequência/amplitude.

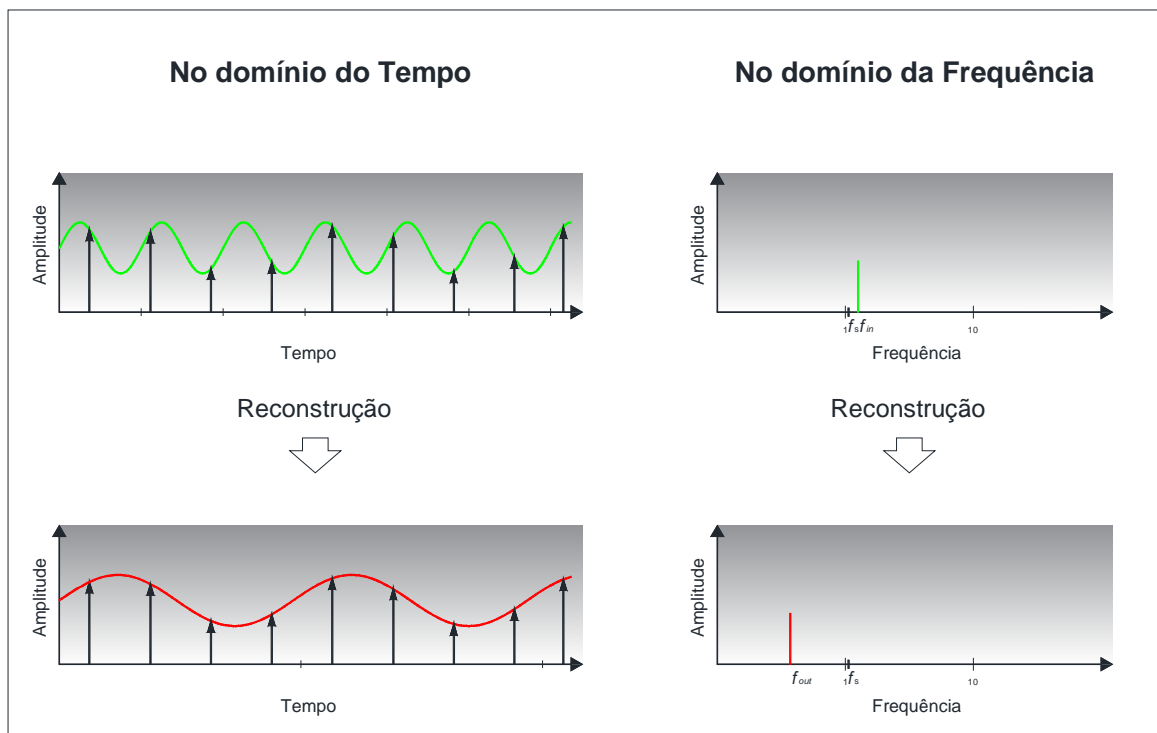


Figura 3.3 - Representação visual do fenómeno *aliasing*

A origem deste problema é a frequência de amostragem ou de registo, que é inferior a frequência verificada no sinal analógico. Para evitar este problema identificado por Nyquist, este sugeriu um critério de frequência que previne este problema. Este indica que a frequência de amostragem deve ser duas vezes superior à frequência do sinal analógico que se pretende medir.

$$f_s \geq 2 \times f_{in} \quad (3)$$

No entanto este critério pode ser o suficiente para evitar totalmente o problema de *aliasing* como se pode verificar pela próxima sequência de imagens ilustrada na figura 3.4. Em que a linha verde representa o sinal analógico e a linha vermelha representa a reconstituição do sinal com os registos discretos.

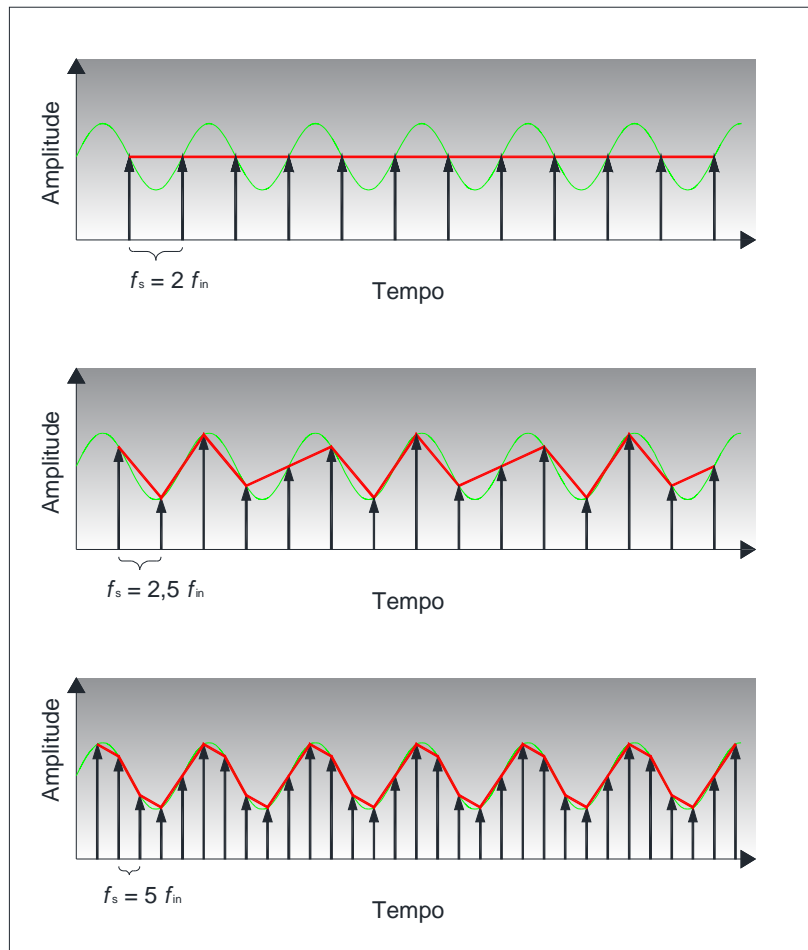


Figura 3.4 - Reconstrução de sinal para diferentes taxas amostragem

A primeira imagem da figura apresenta uma taxa de amostragem duas vezes igual a frequência do sinal analógico, conforme o critério de Nyquist. Caso o ponto de amostragem coincidir com o ponto médio do sinal analógico, a reconstrução deste sinal é um valor de

frequência igual a zero. Para evitar esta situação a frequência de amostragens deve ser um pouco acima do dobro da frequência do sinal analógico.

Verificamos que com o aumento da taxa ou frequência de amostragem melhora a definição e qualidade na reconstrução do sinal analógico. Uma desvantagem deste aumento da taxa é o aumento de dados conseqüentemente o aumento do tempo de pós-processamento e tratamento da informação recolhida devido ao acréscimo do volume de dados. Deve se então encontrar-se um equilíbrio entre a taxa de amostragem e conseqüente quantidade de informação adquirida em função da frequência dos sinais analógicos que se pretendem avaliar nas estruturas e tempo de processamento e tratamento dos mesmos.

Como referido anteriormente a gama de frequência a avaliar nas estruturas de construção civil encontram-se abaixo dos 100Hz. Assim optou-se neste trabalho que os sensores inerciais têm que ser capazes de ter ODR igual ou superior a 250Hz, visto que a aplicação do dispositivo poder ter vários fins.

3.2.2.8. Interface de comunicação

Existem protocolos gerir a transferência de informação entre processadores e circuitos integrados, em canais de uma maneira estruturada e organizada. Estes protocolos dividem-se em dois grupos, o primeiro em comunicação em paralelo e o segundo em série, por sua vez subdividem-se entre comunicação sincronizada e assíncrona.

Sensores são considerados dispositivos periféricos de um sistema de aquisição de dados e os protocolos de comunicação mais populares que se podem encontrar são: *Serial Peripheral Interface (SPI)*; *Inter-integrated Circuit (I²C)*; *Serial communication*, mas existem outros.

Existem inúmeras diferenças entre os protocolos, ambos com as suas vantagens e desvantagens. Só vão ser abordadas algumas dessas diferenças, para se tenha uma percepção das possibilidades, vantagens e limitações e entre protocolos.

As características observadas em cada protocolo foram de natureza prática, relacionadas com velocidade de transmissão, quantidade de conexões possíveis por cada protocolo,

quantidade de periféricos possíveis ligar-se a num único sistema e distância máxima de transmissão de dados entre os periféricos e sistema de aquisição de dados.

Os protocolos mais usados são o SPI e I²C, são protocolos que utilizam informação sincronizada. Isto quer dizer que o relógio interno da unidade de processamento é que regula o ciclo do sistema, e ritmo da comunicação, evitando que comunicação seja feita fora de tempo, e que assim não se perca.

Durante o processo de transmissão de dados é necessário definir que dispositivo transmite e que dispositivo recebe um determinado pacote de dados dentro de uma frequência de tempo definido pelo relógio da unidade processamento.

Caso isso não aconteça a comunicação pode ser transmitida pelo periférico, e não ser rececionada corretamente pelo processador, existindo assim perda de informação. Os protocolos SPI e I²C têm mecanismos que previnem essas falhas.

Em termos de velocidade de transmissão com o protocolo I²C é possível verificar em determinados periféricos e em determinadas condições uma velocidade de transmissão de superior a 3,4Mbps. Com o protocolo SPI pode ter uma velocidade de transmissão de vários MBps/s, dependendo em grande medida da velocidade do relógio da unidade de processamento, por exemplo se o relógio da unidade de processamento for 16MHz a velocidade de transmissão pode atingir os 2Mbps.

No caso do protocolo de comunicação *Serial*, este distingue-se dos anteriores por ser assíncrono. Este é mais complexo em termos de programação, oferece possibilidades diferentes dos protocolos anteriores, no entanto, a velocidade transmissão está condicionada a frequência do relógio da unidade processamento. Este protocolo é capaz de enviar grandes quantidades de informação por segundo, mas o sistema que recebe a informação tem que estar também preparado para receber à mesma velocidade a que esta esta a ser enviada. Assim a frequência do relógio da unidade de processamento limita a velocidade de transmissão de dados.

A distância entre periféricos e unidade de aquisição de dados é condiciona ao tipo de comunicação sincronizada ou assíncrona. A comunicação sincronizada limita a distância a poucos metros, enquanto o segundo tipo de comunicação usada, pode em determinadas circunstâncias obter comunicação a dezenas de metros com conexões físicas.

Um dos motivos da grande diferença é que as ligações sincronizadas foram desenvolvidas para grande velocidade, a curta distância, com poucas ligações e protocolos simplificados. Outro motivo que limita a distância é a resistência do fio elétrico a passagem de corrente elétrica, quanto maior for a distância, maior é o tempo necessário a transmissão de dados.

Ainda de referir a situação da quantidade de conexões usadas em cada protocolo e a sua possibilidade de uni ou bidirecionalidade da comunicação nas conexões. Na figura 3.5 ilustram-se três diagramas que ilustram as conexões necessárias para cada um dos três principais protocolos.

No caso dos protocolos SPI e I²C que a comunicação é sincronizada, existindo um dispositivo que controla todo o processo e transmite o sinal do relógio, que se designa por *Master* ou mestre. Este controla os vários periféricos que se designam por *Slave* ou escravos. Para simplificar o conceito só se apresenta um mestre e um escravo.

Os vários acrónimos inscritos nas figuras, são os que se encontram inscritos nos periféricos e a traduzem-se da seguinte forma:

- Rx – *Receive* ou receber, este canal destina-se a receber informação;
- Tx – *Transmite* ou transmitir;
- SCL e SCK – *Clock* ou relógio, este canal destina-se ao sinal de relógio;
- SDA – *Data Signal* sinal de dados, este canal serve para transmitir e receber informação alternadamente, sendo assim bidirecional;
- MISO – *Master In Slave Out*, este canal destina-se a transmissão de informação do dispositivo mestre para o dispositivo escravo;
- MOSI – *Master Out Slave In*, este canal destina-se a recepção de informação do dispositivo escravo para o dispositivo mestre;
- SS – *Slave Select*, seleção do escravo, este canal é utilizado pelo dispositivo mestre para definir com que dispositivo escravo pretende comunicar.

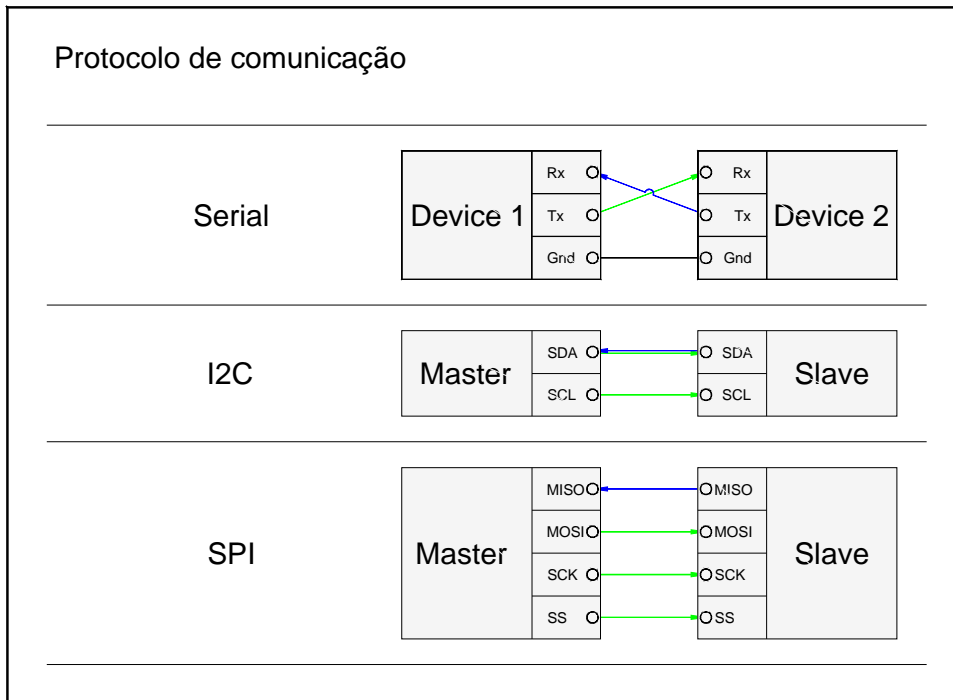


Figura 3.5 - Esquemas de configuração por protocolo de comunicação

O protocolo *Serial* permite a receção de informação de vários dispositivos periféricos à vez. Caso a programação de envio de informação para o processador não estiver coordenada é possível que vários dispositivos enviem informação em simultâneo, inviabilizando toda a informação. Esta situação deve à falta de uma conexão para gerir o envio da informação, ao contrário dos protocolos SPI e I²C, tornando-a mais complicada de programar entre os vários sistemas.

Os protocolos SPI e I²C permitem várias conexões a vários dispositivos periféricos. O protocolo I²C permite mais de 1000 dispositivos ligarem-se a um dispositivo mestre, mas por só ter uma conexão de comunicação bidirecional, necessita que cada periférico tenha uma morada atribuída, distinta dos restantes periféricos do sistema. Caso isso não aconteça, quando se solicita informação para uma morada específica, vários periféricos com a mesma morada enviam informação em simultâneo ocorrendo multiplicação de envio de informação e inutilizado a informação enviada.

Com o protocolo SPI também é possível a ligação de vários dispositivos ao mesmo dispositivo mestre, no entanto este controla os dispositivos escravos através de uma conexão designada por SS, o dispositivo mestre necessita de uma para cada um desses dispositivos, assim limitando o número de dispositivos ligados a ele.

Todos os protocolos anteriormente referidos são válidos neste projeto, tendo apenas que ter atenção as especificidades de cada um deles. Há que ter atenção à escolha do dispositivo mestre, neste caso um microcontrolador, este tem que executar os mesmos protocolos de comunicação que os periféricos que se interligam a ele.

3.2.2.9. Disponibilização em formato *breakout board*

Os sensores inerciais MEMS são de pequenas dimensões na ordem dos milímetros e de difícil manipulação. Para contornar este problema existem fabricantes que desenvolvem placas de circuito impresso ou *printed circuit board* PC, também conhecidos por *breakout boards*, tornando assim possível a sua aplicação em projetos de protótipos. Encontram-se em lojas especializadas em componentes eletrónicos

Estas *breakout board*, também elas são de pequenas dimensões, com apenas alguns centímetros, nelas são instalados vários componentes eletrónicos, circuitos elétricos, de acordo com uma arquitetura adequada de modo que a que o sensor ou outro instrumento nela instalada funcione corretamente. Apenas é necessário executar as conexões entre estes e dispositivos de controlos (microcontrolador).

Abaixo apresenta-se a figura 3.6 com o sensor inercial MPU6050 da InvenSense com dimensões 4x4x0,9 mm (lxcxh), ao lado desta, outra figura 3.7 que apresenta uma *Breakout Board* com o sensor instalado, e com as dimensões 21x16 mm (lxc).



Figura 3.6 - Sensor MPU6050, InvenSense



Figura 3.7 - Breakout Board ITG/MPU

Neste sentido o sensor escolhido para este projeto teve que ser encontrado no formato de *Breakout Board*.

3.2.2.10. Custo

Não houve uma base de comparação para estimar o custo total deste projeto com outros semelhantes. Tendo por base o custo médio de alguns componentes estimou-se um valor global apenas para aquisição e transporte, de todos os componentes necessários para a execução deste projeto que rondou na ordem dos 100€. Assim estimou-se que o valor máximo para aquisição do sensor seria na ordem dos 20% do valor global.

3.2.2.11. Disponibilidade de bibliotecas em função da API adotada

Um protótipo é composto por vários componentes físicos, sendo o componente principal aquele que gere todo o sistema a placa de desenvolvimento, no seu núcleo está o microcontrolador. Todos os módulos físicos interligados à placa de desenvolvimento, designam-se por *hardware*, que só conseguem funcionar entre si graças a um programa informático designado por *software*, instalado no microcontrolador. Para instalar um programa num microprocessador é necessário fazer uma ligação entre a placa de desenvolvimento e uma aplicação de desenvolvimento de *software* instalada num computador pessoal. A aplicação de desenvolvimento é designada por *integrated development environment* ou IDE.

A IDE é responsável por várias funções, nomeadamente para escrever o código ou programa, compilação do código no microprocessador da placa, é também usada para encontrar bibliotecas, instalar bibliotecas dos diferentes componentes físicos, entre outras funções.

Existem vários *softwares* em *open-source* disponibilizados por várias entidades, e conseqüentemente existem também vários microcontroladores que operam diferentes *softwares*, assim existem várias IDE possíveis. Algumas situações específicas não necessitam de IDE, sendo todo programa desenvolvido dentro do ambiente da própria placa.

Em *sites* dedicados ao desenvolvimento e divulgação de *software* em *open-source*, encontram-se para um mesmo sensor, vários códigos de controlo distintos, desenvolvido em função de uma placa de desenvolvimento própria.

A escolha do fabricante da placa de desenvolvimento da IDE e conseqüente *software* depende do código encontrado para gerir cada um dos módulos (*hardware*) usado no protótipo. O critério de escolha do fabricante foi aquele apresentou o maior número de soluções de código para cada um dos módulos necessários ao protótipo.

Assim para cada sensor avaliado determinou-se se o seu fabricante ou fornecedor disponibilizou uma ou várias bibliotecas para diferentes placas de desenvolvimento. Assim condicionou-se a aquisição do microcontrolador em função da disponibilização de uma biblioteca.

Caso o fornecedor não forneça uma biblioteca, existem plataformas *online* de desenvolvimento de *software open-source*, que disponibilizam bibliotecas, códigos e informação para muitos géneros de periféricos, microcontroladores, uma dessa plataformas, a mais popular para este tipo de projeto é o Github. [22]

3.2.3. Avaliação comparativa de sensores inerciais MEMS

Para encontrar sensores inerciais MEMS capacitivos, pesquisou-se em várias plataformas de fornecedores de componentes eletrónicos *online*, fez-se uma lista limitada com várias soluções possíveis. No processo de escolha teve em conta uma série de critérios de seleção que aplicou aos produtos disponibilizados pelos vários fabricantes de sensores MEMS, com objetivos de encontrar o mais adequado de cada fabricante. Assim elaborou-se uma lista preliminar de sensores selecionados que se apresenta na tabela 3.2.

Os vários parâmetros verificados durante a pesquisa foram os seguintes: tipo de sinal, número de eixos, resolução, ruído, ODR, e preço. A distribuição de sensores apresentada na tabela 3.2 foi feita em função do nível de ruído ND, de forma acedente, para uma escala dinâmica idêntica a todos os sensores. Foi usado uma segunda ordenação dentro da realizada anteriormente, para redistribuir valores idênticos de ruído, neste caso foi usado ODR, e distribuído de forma descendente.

Tabela 3.2 - Seleção provisória de sensores inerciais MEMS capacitivos

| Posição | Fabricante | Modelo | Saída de sinal | Triaxial | Resolução | Ruído (melhor performance) | | | Preço [€] |
|---------|---------------------|------------|----------------|-------------------------------------|-----------|------------------------------------|-----------------|--------|--------------|
| | | | | | | ND | Escala dinâmica | ODR | |
| 1 | Analog Devices | ADLX355 | Digital | <input checked="" type="checkbox"/> | 20bit | 25 $\mu\text{g}/\sqrt{\text{Hz}}$ | 2g | 500 Hz | ----- |
| 2 | NPX | MMA8451Q-1 | Digital | <input checked="" type="checkbox"/> | 14bit | 126 $\mu\text{g}/\sqrt{\text{Hz}}$ | 2g | 400Hz | 2 |
| 3 | ST Microelectronics | LIS3DSH | Digital | <input checked="" type="checkbox"/> | 16bit | 150 $\mu\text{g}/\sqrt{\text{Hz}}$ | 2g | 100Hz | 8 |
| 4 | BOSCH | BMA 180 | Digital | <input checked="" type="checkbox"/> | 14bit | 150 $\mu\text{g}/\sqrt{\text{Hz}}$ | 2g | 10 Hz | 6 |
| 5 | TDK InvenSense | MPU6050 | Digital | <input checked="" type="checkbox"/> | 16bit | 400 $\mu\text{g}/\sqrt{\text{Hz}}$ | 2g | 1000Hz | 1 |
| 6 | Analog Devices | ADLX345 | Digital | <input checked="" type="checkbox"/> | 10~13bit | 290 $\mu\text{g}/\sqrt{\text{Hz}}$ | 2g | 100Hz | 13 |
| 7 | ST Microelectronics | LIS331HH | Digital | <input checked="" type="checkbox"/> | 16bit | 650 $\mu\text{g}/\sqrt{\text{Hz}}$ | 6g | | 9 |
| | Analog Devices | ADLX354 | Analogico | <input checked="" type="checkbox"/> | 20bit | 20 $\mu\text{g}/\sqrt{\text{Hz}}$ | 2g | 500 Hz | ----- |
| | ST Microelectronics | LIS344ALH | Analogico | <input checked="" type="checkbox"/> | ----- | 50 $\mu\text{g}/\sqrt{\text{Hz}}$ | 2g | | 14 |

Como já referido optou-se por utilizar um sensor de *output* digital, contudo apresenta-se alguns sensores analógicos para ilustrar o seguinte aspecto, os sensores analógicos apresentam uma menor densidade de ruído, assim teoricamente apresentam uma melhor precisão comparativamente com os sensores digitais. O sensor ADLX354 e são ADLX355 são praticamente idênticos em todos os aspetos exceto que o ADLX355 tem ADC e ADLX354 não o tem, se analisarmos a densidade de ruído verificamos que a uma redução de $5 \mu\text{g}/\sqrt{\text{Hz}}$.

O sensor apresentado na primeira posição foi o que apresentou as melhores características em quase todos os critérios exceto dois. Este sensor foi lançado em setembro de 2016, é relativamente recente, ainda não são conhecidos trabalhos ou aplicações feitos com este sensor, portanto o seu potencial ainda não foi validado, é o único na lista mais indicado para aplicações SHM. Os critérios que não cumpriu foram o preço e a disponibilidade em *breakout board*, de acordo com o fabricante o preço deste sensor ronda os 23€, mas presente data não existem *breakout boards*, apenas existem placas de avaliação. O valor apresentado deve acrescentar o preço de aplicação numa PCB.

Em segundo lugar na lista apresenta-se o sensor digital com o segundo menor valor de ruído, com a vantagem de obtê-la com uma taxa de amostragem relativamente alta (400Hz). Um ponto menos favorável é resolução do ADC de apenas 14bit, enquanto a maioria dos seus congéneres é de 16bit. Apresenta uma segunda vantagem que é a baixa densidade de ruído, na ordem dos $100 \mu\text{g}/\sqrt{\text{Hz}}$, mas com perda de resolução de ADC, pois esta só está disponível para uma escala dinâmica para $\pm 4\text{g}$.

Os sensores na terceira e quarta posição estão sensores com a mesma densidade de ruído, mas obtidos com ODR e resoluções de ADC distintos. O sensor na posição 4

apresenta ser um sensor bem construído, com um tempo de vida bastante longo, muito explorado em várias aplicações e áreas distintas e com bons resultados, também são encontrados bastantes estudos com mencionado este sensor. Todos estes pontos contribuíram para inclusão deste sensor na lista.

O sensor na posição 5 tem uma densidade de ruído relativamente alta, mas esta é obtida com um ODR muito alto, teoricamente para taxa de amostragem mais reduzida, este valor de ruído deve ser mais baixo. Outra situação que gerou interesse neste sensor foi que este possui 6 graus de liberdade, para além de ser um acelerómetro também é um giroscópio com três eixos.

O sensor na sexta posição tem muitas semelhanças com o sensor na posição 4. É um sensor com grande longevidade, usado em muitas aplicações, mas o ADC tem a menor resolução da lista apenas com 10 bit. Este tem uma opção para aumentar a resolução até 13 bit, que passa por aumentar também o parâmetro da escala dinâmica.

O sensor na sétima posição não cumpre o critério da escala dinâmica, neste caso a escala mínima é $\pm 6g$ enquanto deveria ser $\pm 2g$. As razões que levaram a inclusão deste sensor nesta lista são justificadas num ponto mais a frente.

De seguida apresenta-se a tabela 3.3 com uma lista que para cada sensor listado na tabela 3.2 indica informação acerca dos protocolos de comunicação usados por cada, quantidade de bibliotecas e códigos de programação por cada uma das linguagens de programação encontradas. esta informação foi disponibilizada pela plataforma Github e fornecedores de componentes eletrónicos.

Tabela 3.3 - Distribuição de linguagens de programação pelos sensores inerciais MEMS capacitivos

| Posição | Fabricante | Modelo | Protocolos de comunicação | Software | | | | |
|---------|---------------------|------------|---------------------------|----------|-----|--------|---------|-------------|
| | | | | C++ | C | Python | Arduino | Java Script |
| 1 | Analog Devices | ADLX355 | I ² C; SPI | | | | | |
| 2 | NPX | MMA8451Q-1 | I ² C | 6 | 4 | 3 | 1 | 1 |
| 3 | ST Microelectronics | LIS3DSH | I ² C; SPI | 1 | 11 | | 1 | |
| 4 | BOSCH | BMA 180 | I ² C; SPI | 3 | 1 | 1 | 2 | |
| 5 | TDK InvenSense | MPU6050 | I ² C | 150 | 145 | 38 | 111 | 11 |
| 6 | Analog Devices | ADLX345 | I ² C; SPI | | | | 1 | |
| 7 | ST Microelectronics | LIS331HH | I ² C; SPI | 1 | | | 3 | |

Encontrou-se para quase todos os sensores listados na tabela 3.2 uma biblioteca ou informação de implementação de um código de programação, com a exceção do sensor apresentado na primeira posição.

A linguagem de programação mais popular é C++, no entanto não foi encontrado uma biblioteca para sensor na posição 6. A linguagem com a maior variedade, encontrando-se uma biblioteca para quase todos os sensores listado é a linguagem Arduino.

Nota-se também que, o sensor que excedeu em quantidade de informação é encontrado na quinta posição. As razões motivam os programadores a desenvolver código para este sensor é facto de este ter 6 graus de liberdade, ser muito económico, possuindo muitas funções extras, tornam-no muito desejável pelos fabricantes de protótipos.

De seguida apresenta abaixo a tabela 3.4 que para cada sensor indicado na tabela 3.2, as escalas dinâmicas e ODR disponíveis.

Tabela 3.4 - Informação de escalas dinâmicas, ODR dos sensores inerciais MEMS

| Posição | Fabricante | Modelo | Escalas dinâmicas | ODR |
|---------|---------------------|------------|----------------------------|--|
| | | | | [Hz] |
| 1 | Analog Devices | ADLX355 | ±2g ±4g ±8g | ±3,9 ±7,8 ±5,625 31,25 62,5 125 250 500 1000 2000 4000 |
| 2 | NPX | MMA8451Q-1 | ±2g ±4g ±8g | 1,56 6,25 12,5 50 100 200 400 800 |
| 3 | ST Microelectronics | LIS3DSH | ±2g ±4g ±8g ±16g | 3,125 6,25 12,5 25 50 100 200 400 800 1600 |
| 4 | BOSCH | BMA 180 | ±1g ±1,5g ±2g ±4g ±8g ±16g | 10 20 40 750 150 300 600 1200 |
| 5 | TDK InvenSense | MPU6050 | ±2g ±4g ±8g | 256 passos variando entre 31,25 a 8000 |
| 6 | Analog Devices | ADLX345 | ±2g ±4g ±8g ±16g | 6,25 12,5 25 50 100 200 400 800 1600 3200 |
| 7 | ST Microelectronics | LIS331HH | ±6g ±12g ±24g | 50 100 400 1000 |

Verifica-se que todos eles têm taxas de amostragem favoravelmente acima do estabelecido pelo critério de seleção, no entanto são poucos os que têm exatamente a taxa de amostragem requerida.

Os ODR estão estabelecidos por passos, não sendo possível definir a taxa de amostragem concreta que se pretende, a solução para ultrapassar este problema é selecionar uma taxa de amostragem acima da requerida.

O sensor na segunda posição apresenta as melhores características dentro do conjunto de critérios apresentados, no entanto sem uma caracterização metrológica, testes ou ensaios

comparativos em condições semelhantes às reais, não é possível afirmar que este é a melhor solução.

No início deste projeto o professor Hugo Rodrigues já tinha em sua posse um dispositivo com o qual intencionava realizar ensaios. Este dispositivo foi desenvolvido em torno do sensor que se encontra na quarta posição da tabela 3.2. É um dispositivo de aquisição de vibrações completo em termo de *software* e hardware. Este tipo de produto designa-se por *Commercial off-the-shelf*. Este produto encontra-se associado a uma rede monitorização sísmica, construída por voluntários que instalam este dispositivo nos seus computadores pessoais e com ajuda de *software* de gestão, recolhem informação relacionada com movimento fortes produzidos por sismos e enviam para um centro de processamento de dados.

Assim selecionou-se três dos sensores com a melhores característica, para que se realizem a cada um deles uma série de simples ensaios em condições idênticas, obtendo informação comparativa entre e assim avaliando qual o melhor sensor. Os sensores selecionado para aquisição foram são os encontrados nas posições 2,3 e 5 da tabela 3.2.

3.2.4. Aquisição de sensores

Em Portugal existem várias empresas *online* ligadas a venda de componentes de eletrónicos, mas no que concerne disponibilização de aos sensores inerciais MEMS a oferta apresentada por estas foi limitada quanto à variedade.

Para contornar este problema e obter maior diversidade de produtos, preços, serviços, etc, recorreu-se ao mercado internacional de importação, através de plataformas *online* dedicada a venda de produtos *Customer to Customer C2C*.

Como já referido este trabalho consistiu na produção de um dispositivo de monitorização de vibrações, por outras palavras consiste na construção de um protótipo formado com vários componentes modulares interligados com ligações semipermanentes, assim podendo ser alterado, modificado ou aumentado em termos de recursos.

Devido ao grande variedade e quantidade de materiais usados neste projeto e visto que nem todas as lojas *online* disponibilizavam os componentes necessários, teve que se

recorrer a múltiplas lojas localizadas em vários países e zonas, estes foram alguns desses, Portugal, Luxemburgo, Reino Unido, Hong Kong & China, Malásia.

Ao recorrer a vários fornecedores, conseguiu-se obter preços bastante competitivos e assim baixar o custo total do projeto. Neste trabalho foram adquiridas pequenas quantidades e muitos casos apenas unidades singulares. Nestas condições foi apenas possível recorrer aos pequenos negócios que operam em plataformas dentro deste tipo modelo, *Customer to Customer C2C* que são distinguem dos modelos de negócio mais comuns *Business to Customer B2C*.

Na solução C2C tem que ter em conta que o tempo de entrega é maior, para encomendas fora da comunidade europeia, em média levaram 25 dias e em situações especiais até 40 dias.

No decorrer do processo aquisição dos sensores, ocorreu um problema a com a encomenda do sensor que se encontra na terceira posição, este sensor foi encomendo a um fornecedor que não cumpriu o acordo estabelecido. Para solucionar o problema procurou-se outro fornecedor, mas não se localizou nenhum.

Para contornar esta situação optou-se por um sensor semelhante, do mesmo fabricante, mas um modelo diferente, com características distintas, que é o sensor indicado na sétima posição da tabela 3.2.

Assim a lista final dos sensores inerciais MEMS é a apresentada na tabela 3.5:

Tabela 3.5 - Seleção final de sensores inerciais MEMS

| Posição | Fabricante | Modelo | Observações |
|---------|---------------------|------------|-------------|
| 2 | NPX | MMA8451Q-1 | adquirido |
| 4 | BOSCH | BMA 180 | emprestado |
| 5 | TDK InvenSense | MPU6050 | adquirido |
| 7 | ST Microelectronics | LIS331HH | adquirido |

Os fabricantes que produzem *breakout board* para os sensores indicados na tabela 3.5, e que foram adquiridos, são os seguintes para cada um dos sensores: o sensor LIS331HH é

usado pela Sparkfun [1]; o sensor MPU6050 usado pela DFRobot [2]; o sensor MMA8451Q é usado pela GY.

Todas estas estas placas apenas estão preparadas para o protocolo de comunicação I²C. Nas especificações técnicas dos sensores MMA8451Q, LIS331HH e MPU6050, referem que a velocidade máxima permitida por estes quando usando o protocolo I²C é 400kHz.

A tensão elétrica aceite pelas placas da GY e DFRobot é entre 3,3V e 5,0V, enquanto a placa da Sparkfun só aceita 3,3V.

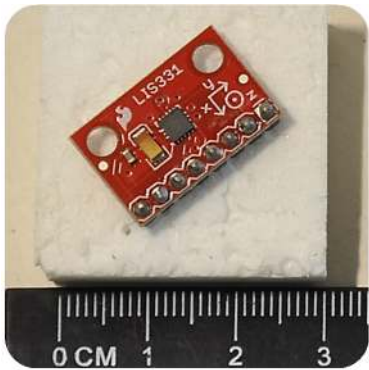


Figura 3.9 Sensor LIS331HH

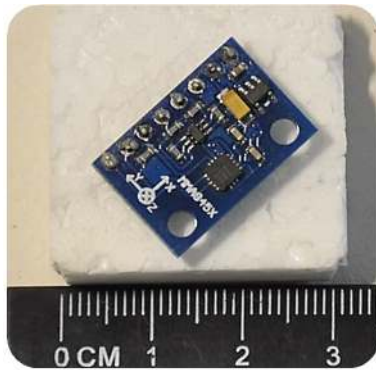


Figura 3.10 - Sensor MMA8451Q

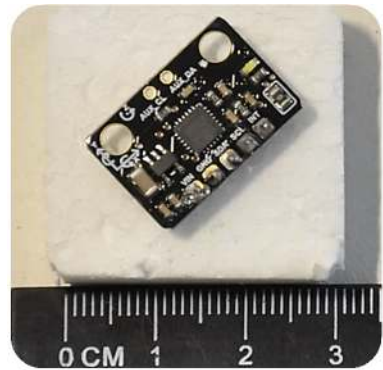


Figura 3.8 - Sensor MPU6050

4. Protótipo de aquisição de vibrações

4.1. Introdução

Nesta fase de concepção do protótipo focou apenas no desenvolvimento do sistema de aquisição e registos dos sinais digitais, para ser usado por todos os sensores selecionados. Assim criando um sistema compatível com os vários módulos de sensores, intermutáveis, abrindo a possibilidade de uma constante evolução.

O dispositivo foi desenvolvido com base filosofia *Do-It-Yourself* DIY ou faça-você-mesmo, e também com recurso a plataformas *open-source* ou código aberto, para que assim se obtenha um baixo custo de produção deste dispositivo.

Este dispositivo seria usado para testar todo conceito, bem como testar os próprios sensores. Por forma a simplificar o processo desenvolveu-se apenas a capacidade para adquirir e registar sinais dos sensores. Portanto o dispositivo não necessitou de ter uma grande capacidade de processamento, assim o microcontrolador escolhido foi o mais económico possível.

Durante a avaliação de sensores, verificou-se que o maior número de bibliotecas existentes, para todos os sensores selecionados, encontravam-se desenvolvidos para a linguagem de programação e IDE concebida pela Arduino.

4.1.1. Arduino

A Arduino é uma plataforma desenvolvimento e ensino de protótipos eletrónicos. Esta fabrica e comercializa *hardware*, disponibiliza *software* em *open-source*. Esta plataforma foi apresentada em 2005 com objetivo de produzir *hardware* e *software* de fácil utilização e disponível para todos. [23] Está vocacionada para aqueles que têm pouco conhecimento de eletrónica e querem apreender a desenvolver os seus próprios protótipos.

Esta plataforma desenvolveu-se rapidamente graças à sua adoção por parte da comunidade de praticantes e contribuidores que disponibilizam informação através de fóruns, como é caso do fórum existente na página na WWW da Arduino. [24] No site

podem-se encontrar muitos projetos de vários participantes, manuais de referência que ajudam a compreender a linguagem de programação. A Arduino que tem base na linguagem C, os programas na nomenclatura da Arduino se designam por *sketch*, e servem para controlar o *hardware* da Arduino. No seu fórum pode-se colocar dúvidas sobre problemas encontrados durante o desenvolvimento de projetos, programação, entre muitos outros assuntos, que por norma são rapidamente respondidos por outros membros da comunidade.

A Arduino desenvolveu o seu próprio ambiente de desenvolvimento integrado ou *Integrated Development Environment (IDE)*, que permite escrever programas e instalá-los em placas que aceitem a linguagem de programação Arduino. O IDE é instalado num computador pessoal, é neste aplicativo que se desenvolvem, compilam e importam os programas ou *sketch*, para dentro das placas usando uma ligação por cabo Universal Serial Bus (USB).

Uma IDE é um *software* de um computador que permite ao utilizador, criar, compilar, e atualizar código para um microprocessador. [24]

Este IDE da Arduino é compatível com três sistemas operativos correntes no mercado de PC pessoais, Windows, Mac OS X e Linux. Oficiosamente existem aplicações para os sistemas operativos móveis, Android e Windows, estes oferecem uma grande versatilidade e mobilidade, permitindo alterar os *sketches* de dispositivos fixos em locais remotos de difícil acesso.

O IDE comporta-se como um editor de texto, usa uma linguagem própria para criar um *sketch*. Os componentes da linguagem agrupam-se em três partes ou grupos distintos, cada um destes com características ou funções próprias necessárias para que um *sketch* opere corretamente, estas categorias da linguagem do IDE são: Estruturas; Variáveis e Funções.

Existem duas funções cruciais sem as quais nenhum *sketch* não funciona corretamente, estas são *setup()* e *loop()*, pertencem à categoria da Estruturas. A função *setup()* é a primeira a ser executada e apenas uma vez, dentro desta função são colocadas tarefas de iniciação, como iniciar bibliotecas ou ligar pins. A função *loop()* é uma função recursiva, todo o código, funções, etc, aplicado dentro desta função é repetido ciclicamente.

A linguagem Arduino é bastante simples comparativamente com as outras linguagens comerciais. É facilmente apreendida através de execução de exercícios simples, para isso

existem *kits* de iniciação que disponibilizam múltiplas experiências isoladas, permitindo explorar cada conceito, função, estrutura da linguagem Arduino separadamente. A medida que cada utilizador vai adquirindo os conceitos diferentes, vai tomando consciência das possibilidades desta linguagem, criando os seus próprios programas através da integração de códigos apreendidos nas experiências isoladas. A Arduino disponibiliza no seu *site* tutoriais sobre a linguagem e instruções para realizar cada experiência.

A linguagem Arduino possui vários recursos avançados que expandem as funcionalidades dos *sketches*, um destes são as Bibliotecas que permitem a reutilização de programa ou códigos previamente escritos, bastando invoca-lo através de comandos dentro de um novo *sketch*, assim evitando a repetição de código e poupando espaço na memória do microcontrolador.

Ao instalar o IDE da Arduino num computador pessoal, este já inclui várias bibliotecas para operar múltiplas placas, periféricos, exemplos de código, etc, este também inclui gestores que instalam, atualizam as várias novas bibliotecas.

Existem vários contribuidores que escrevem bibliotecas na linguagem Arduino, necessários para operar diversos dispositivos, componentes, etc, assim facilitando a utilização destes, sem estes contributos desta comunidade *open-source*, não seria possível a utilização de muita tecnologia existente no mercado e a existência de muitos produtos que se iniciaram em protótipos experimentais.

A IDE da Arduino permite três formas de instalação destas bibliotecas, a primeira através de um recurso existente no IDE que permite a instalação automaticamente de uma biblioteca, mas este só aceita ficheiros ZIP. O segundo processo é instalar a biblioteca manualmente, diretamente na pasta das bibliotecas dentro da pasta do programa. A terceira na opinião deste autor é a mais aconselhada, consiste em instalar as bibliotecas manualmente dentro duma pasta existente na pasta Documentos com nome Arduino, este é criada automaticamente durante o processo de instalação do IDE, a vantagem é, caso se atualize ou desinstale o IDE estas ficam salvaguardas de serem eliminadas.

As placas de desenvolvimento da Arduino interligam-se aos dispositivos e componentes elétricos ou eletrónicos através de *pins* e *headers*, que não são mais do que pequenas fichas machos e fêmeas. Também existem outras ligações ou portas com as USB tipo A, B, mini e conectores do género *barrel jack*. Estas ligações são usadas para receber ou fornecer, energia elétrica, sinais elétricos em formato analógicos ou digitais.

Por forma a aumentar as capacidades das suas placas, a Arduino desenvolve *shields* ou placas de expansão para cada modelo, múltiplas placas com funções diferentes podem ser empilhadas sobre uma placas base, através das ligações pins e *headers*. Cada placa de expansão ou *shield* adiciona novas funções a placa onde se insere, estas funções podem ser por exemplo, comunicação sem fios, tipo Bluetooth ou WIFI, *dataloggers* que entre outras funções, permite o registo de informação num componente físico, como é caso dos cartões SD.

4.1.2. Especificações do sistema

As especificações para o dispositivo 1 concernem as necessidades ou capacidades operacionais do *Hardware* e *Software*. Algumas destas foram abordadas durante o processo de seleção dos sensores inerciais MEMS capacitivos, assim os restantes componentes do sistema ajustam-se para cumprir essas especificações.

Abaixo resumem-se as especificações identificadas em duas partes, a primeira é relativa ao output ou resultados que dependem do programa e *software*, a segunda parte as características físicas ou *hardware*.

Especificações do *Software* / programa

- O dispositivo de aquisição e registo deve ser capaz de produzir dados a uma taxa de amostragem aproximadamente 250Hz, dos três eixos com uma escala dinâmica de $\pm 2g$.
- Os dados provenientes do sensor devem ser registados ou gravados na unidade de armazenamento, num formato compatível com a maioria dos *softwares* de análise de dados. A norma de formatação comum é a seguinte, quatro colunas que representam quatro campos, cada campo é separado por uma tabulação, isto no plano horizontal. No plano vertical, cada linha representa o registo de um evento num determinado tempo que é apresentado na primeira coluna. O formato dos registos é o obtido do sensor, evitando assim tempo de processamento, numa fase posterior em pós-tratamento são então convertidos em aceleração com as unidades de medidas S.I. (m/s^2).

Especificações do *Hardware*

- Em termo de fonte de alimentação ao sistema
 - Deve ter uma fonte alimentação independente, interna com uma autonomia ou duração necessária para realizar cada ensaio completo sem falhas de corrente.
 - Esta deve ser estável assim prevenindo erro associados a flutuações elétricas.
 - A alimentação externa é opcional.
- Este não necessita de suporte físico (caixa externa), mas deve ser compacto.
- O dispositivo de aquisição e registo deve ter uma unidade de armazenamento de dados, com uma taxa de receção e transmissão de dados adequada, de modo que se consiga registar dados à taxa de amostragem exigida, referida num ponto anterior, deve ainda ser de fácil acesso e remoção, do género cartão SD.
- O microcontrolador deve ter a capacidade processamento mínima necessária gerir todas as funções necessárias ao bom comportamento do sistema, em especial atenção à taxa de amostragem anteriormente referida.
- Os componentes do dispositivo devem ser padronizados, intermutáveis, de fácil aquisição, construção ou montagem, programação e utilização, com o propósito de ser reprodutível.

Durante fase de pesquisa de informação sobre microcontroladores e placas de desenvolvimento para protótipos verificou-se que a tensão corrente na maioria das placas é 5,0V por vezes existindo algumas a operar a 3,3V. No caso das placas da Arduino 54,2% destas funcionam a 5V, 37,5% funcionam a 3,3V e 8,3% funcionam com uma tensão de corrente entre 3,3 a 5,0V.

Quanto menor for a tensão de corrente menor são os consumos energéticos. As placas de desenvolvimento Arduino com tensão de corrente a 3,3V são por norma mais caras que as 5,0V, mas também com maior capacidade de processamento, isto deve a utilização de microcontrolador de gerações mais avançadas, com melhores prestações. Numa segunda análise verifica-se que há placas de 3,3V a preços abaixo dos modelos base a 5,0V, mas a contrapartida e que têm metade da capacidade de processamento.

As necessidades tensão de corrente ou voltagem dos sensores seleccionados variam entre os 1,6V a 3,6V. Mas a maior parte *breakout boards* sobre as quais estes sensores estão instalados aceitam tensões de corrente entre 3,3V a 5,0V. Um dos sensores seleccionados tem a particularidade de apenas aceitar 2,16V a 3,6V.

No caso da utilização de placas de desenvolvimento com tensão de corrente igual a 5,0V, deve-se tornar o sistema compatível com todas *breakout boards*, a solução é aplicar um componente eletrónico que converte a tensão elétrica proveniente da placa de desenvolvimento de 5V para uma tensão de corrente a 3,3V, esse componente tem como nome e função, *Bi-Directional Logic Level Converter*.

Abaixo são propostos dois diagramas de bloco para o mesmo dispositivo 1, mas a operar com uma tensão de elétrica a 5,0V ou de 3,3V.

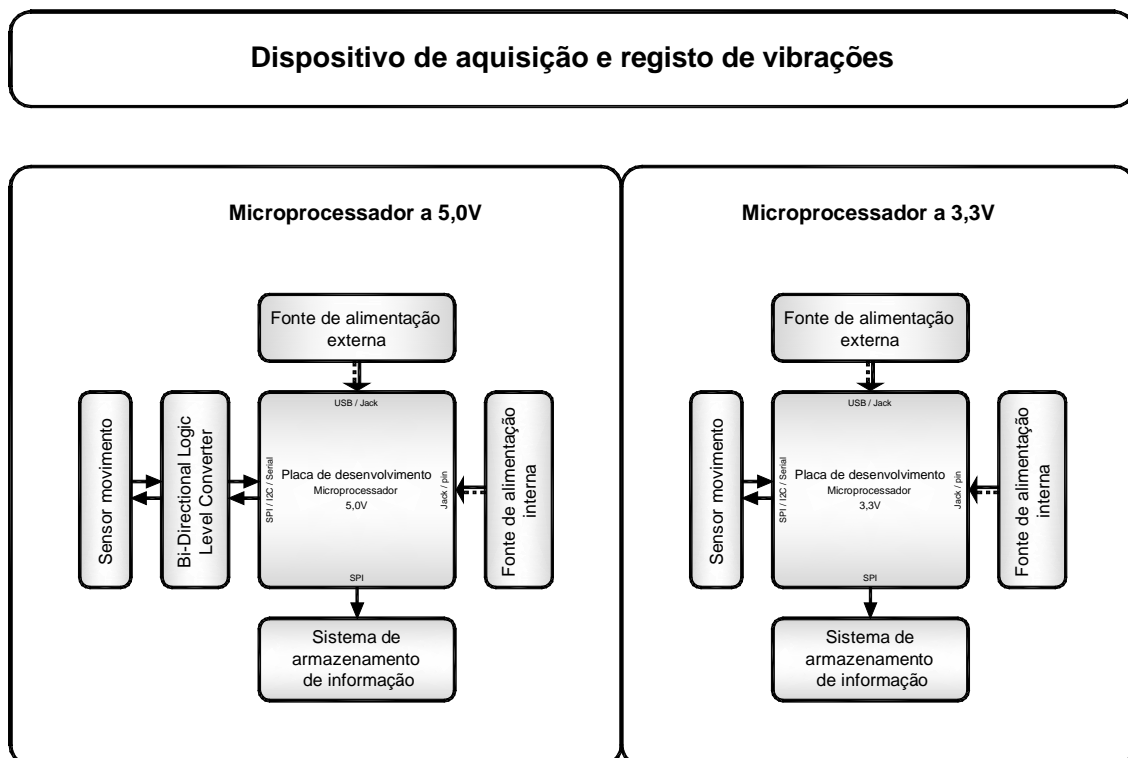


Figura 4.1 - Diagrama de bloco do Dispositivo 1

4.2. Componentes e montagem do protótipo 1

4.2.1. Placa de desenvolvimento - Microcontrolador

Uma placa de desenvolvimento de um protótipo eletrónico, é composta por muitos elementos e componentes, sendo o principal o microcontrolador, numa analogia com corpo humano este é o cérebro.

Basicamente é um computador numa única peça com um tamanho diminuto. Possui um núcleo de processamento, vários tipos de memória e periféricos programáveis de entrada e saída, analógica e/ou digital.

Estes são aplicáveis em vários tipos de cenários e aplicações, e são utilizados para executarem tarefas repetitivas, desde que sejam programados corretamente. Têm também a particularidade de consumirem pouca energia elétrica, assim aumentando o tempo de operação em situações em que a única fonte de energia é limitada.

O custo destas placas está relacionado com a capacidade de processamento, arquitetura de construção do processador, bem como a geração do microcontrolador, entre outros fatores. Quanto maior a capacidade de processamento ou estado de evolução mais caros são. Desta forma os microprocessadores mais económicos são os processadores com arquitetura de 8bits, existindo um grande volume de produção e procura, segue-se a estes os processadores de 16bits e 32bits.

No processo de concepção do Dispositivo 1 optou-se por uma placa de desenvolvimento da marca Arduino, modelo UNO R3. Esta é um dos primeiros modelos fabricados pela Arduino, já foi revista e atualizada 3 vezes, é recomendada para quem está iniciar na produção de protótipos existindo muita documentação e informação sobre esta *online*

As especificações da placa UNO R3 estão apresentadas na tabela 4.1:

Tabela 4.1 - Especificações da placa de desenvolvimento Arduino UNO R3

| Fabricante Modelo | processador | | | Voltagem | | Velocidade | Ligações - Conceções | | | | Memória | SDRAM | Flash | Portas | |
|----------------------|-------------|------------|----------------------|-----------------|----------------|--------------|----------------------|-------|----------|-----|----------------|-------|-------|------------|------|
| | Fabricante | Modelo | arquitetura [bit] | Operação [V] | Entrada [V] | CPU [MHz] | Análogicas | | Digitais | | EEPROM [kB] | [kB] | [kB] | USB | UART |
| | | | | | | | Entrada | Saída | IO | PWR | | | | | |
| Arduino - UNO | ATMEL | ATmega328P | 8 | 5 | 7-12 | 16 | 6 | ----- | 14 | 6 | 1 | 2 | 32 | Tipo B (F) | 1 |

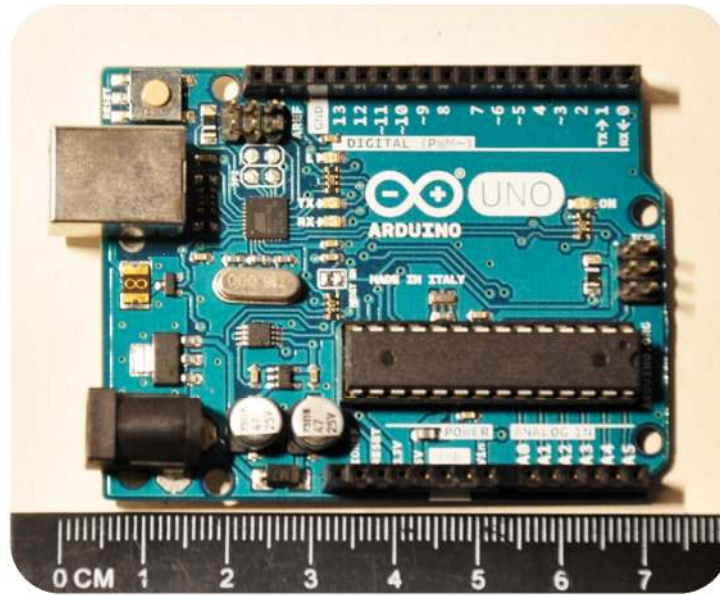


Figura 4.2 - Arduino UNO R3

Em termo de protocolos de comunicação esta placa suporta, SPI a uma velocidade de 1MHz, I²C a 400kHz, bem como UART. A velocidade máxima do protocolo de comunicação I²C, aceite por todos os sensores é compatível com esta placa.

Esta placa possui um *Real Time Clock* (RTC) incorporado ou seja um relógio em tempo real, mas tem uma capacidade de contagem limitada, apenas consegue contar até 2^{32} microssegundos o que equivale sensivelmente a 1:11:34,96 (h:mm:ss), ao fim deste tempo dá-se um fenómeno de *rollover* em que o tempo volta a zero, recomeçando a contagem do princípio.

4.2.2. Fonte de energia

A placa Arduino UNO R3 possui uma porta USB tipo B fêmea, pela qual pode-se fornecer energia elétrica externa, proveniente ou de um computador portátil ou transformador de corrente alternada em corrente contínua (AC-DC), com uma tensão elétrica a 5,0V e intensidade de corrente 0,5A. O consumo elétrico máximo desta placa é 0,5A.

Existem duas opções de fornecimento de energia elétrica interna, a primeira é por uma porta tipo *barrel jack* de 2,1mm de diâmetro e a segunda é por um pin designado Vin ou *voltage in* existente na placa.

A opção escolhida para fornecer energia a este dispositivo foi a através de uma pilha alcalina de 9V, em combinação com um adaptador que interliga a pilha à porta *barrel jack*, existente na placa. Abaixo ilustra-se a solução proposta.



Figura 4.3 - Fonte de alimentação interna

4.2.3. Bi-Directional Logic Level Converter

O *bi-directional logic level converter* ou conversor de nível logico bidirecional, são utilizados para transferir informação bidireccionalmente entre sistemas que operam em voltagens diferentes. Foram usados para transferir informação de sensores que operam a 3,3V e a placa Arduino R3 Uno que opera a 5V. Sem este módulo qualquer sistema que opere com voltagem mais baixa ficaria danificado.

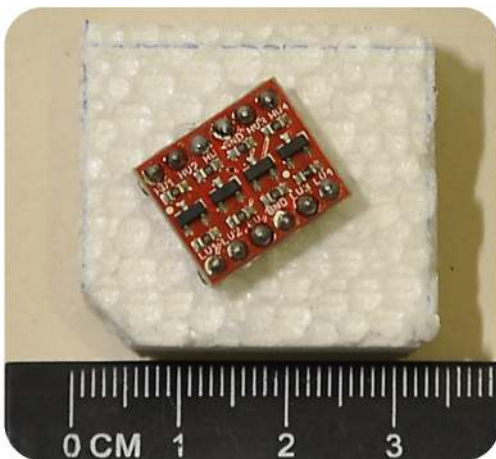


Figura 4.5 - Logic Level Converter

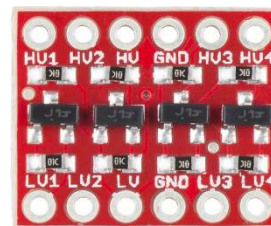


Figura 4.4 - Pormenor de Logic Level Converter

4.2.4. Leitor de cartões micro SD e cartão micro SD

Foi adquirido um módulo de cartão micro SD que funciona com um porta de ligação entre o microcontrolador e os cartões micro SD, assim permitindo ler e escrever informação num cartão. Este componente foi adquirido ao fabricante DFRobot,[26] este opera com tensão elétrica de 5,0V utiliza o protocolo de comunicação SPI. As velocidades de transmissão são dependentes da velocidade operação de transmissão de dados da placa microcontroladora e da classe cartão.

Segundo informações a placa Arduino UNO a 16MHz, com programa que controla o protocolo SPI, não permite transmitir informação com velocidades superiores a 1MHz, assim consultado uma tabela de classes de velocidade de cartão SD informa-nos que para este caso um cartão de classe 1 seria suficiente, contudo a classe mínima encontrada no mercado foi classe 4.

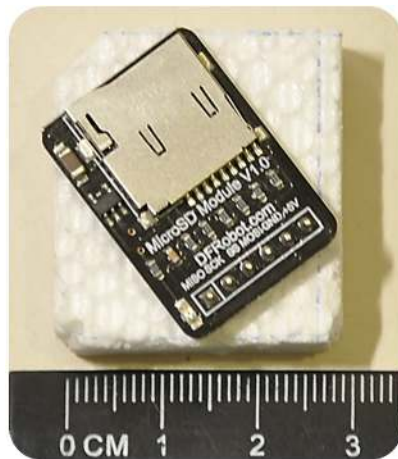


Figura 4.7 - Módulo de cartão micro SD

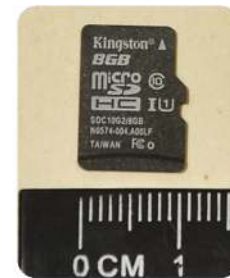


Figura 4.6 - Cartão micro SD

4.2.5. Suporte físico de montagem de componentes

Foi necessário uma estrutura para fixar os vários componentes e estruturar um circuito eletrónico entre estes. Existem várias soluções possíveis, a maioria são de carácter permanente, no entanto existe uma solução de carácter temporário, económica permitindo múltiplas utilizações, chamada de *breadboard*.

A *breadboard* é uma placa que possui múltiplos terminais espaçados regularmente entre si, nestes terminais podem-se aplicar componentes. Na sub-base da plataforma existem

pistas metálicas que agrupam os terminais em séries, assim permitindo a construção de circuitos pela interligação dos terminais com recurso a fios ou *jumpers*.

Por norma no lado inferior da *breadboard* possui uma fita adesiva para fixação permanente a local. Esta possui quatro pistas laterais e transversais aos grupos de terminais, destinadas à distribuição de energia elétrica. Por norma estas classificam-se pelo número de terminais que possuem.

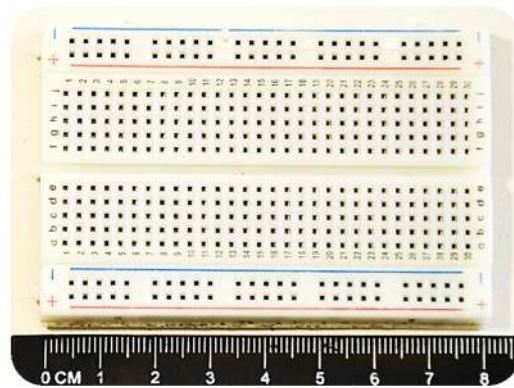


Figura 4.8 - Breadboard de 400 terminais

4.2.6. Jumpers

Um *jumper* não mais que uma ligação que permite a transmissão de energia elétrica entre dois terminais. A forma simples de criar um *jumper* é obter um fio elétrico com as dimensões necessária para ligar os terminais, revisto com material isolante e descarnado nas terminações. No mercado existem múltiplas soluções para vários fins, variam em termos de dimensões, conectores, quantidade conexões, etc.

Na concepção deste dispositivo utilizou os *jumpers* ilustrados na imagem abaixo, com 10cm e 20cm de comprimento.

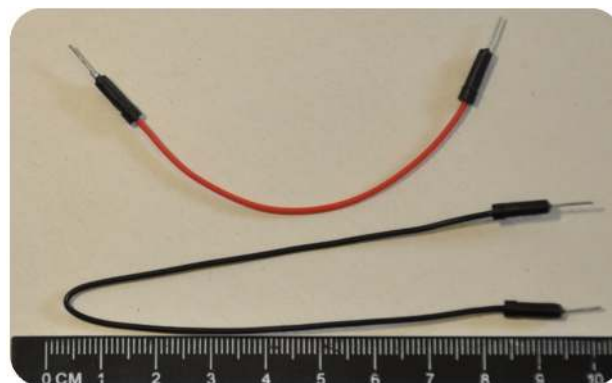


Figura 4.9 - Jumpers 10cm e 20cm

4.2.7. Diagrama de ligações e montagem do Dispositivo 1

Para ilustrar as ligações entre os vários componentes do Dispositivo 1, a figura 4.10 mostra o diagrama de ligações e montagem, assim facilitando a compreensão de montagem deste. Todos elementos apresentados estão a mesma escala. No Anexo A1 é apresentado um procedimento de montagem do dispositivo 1.

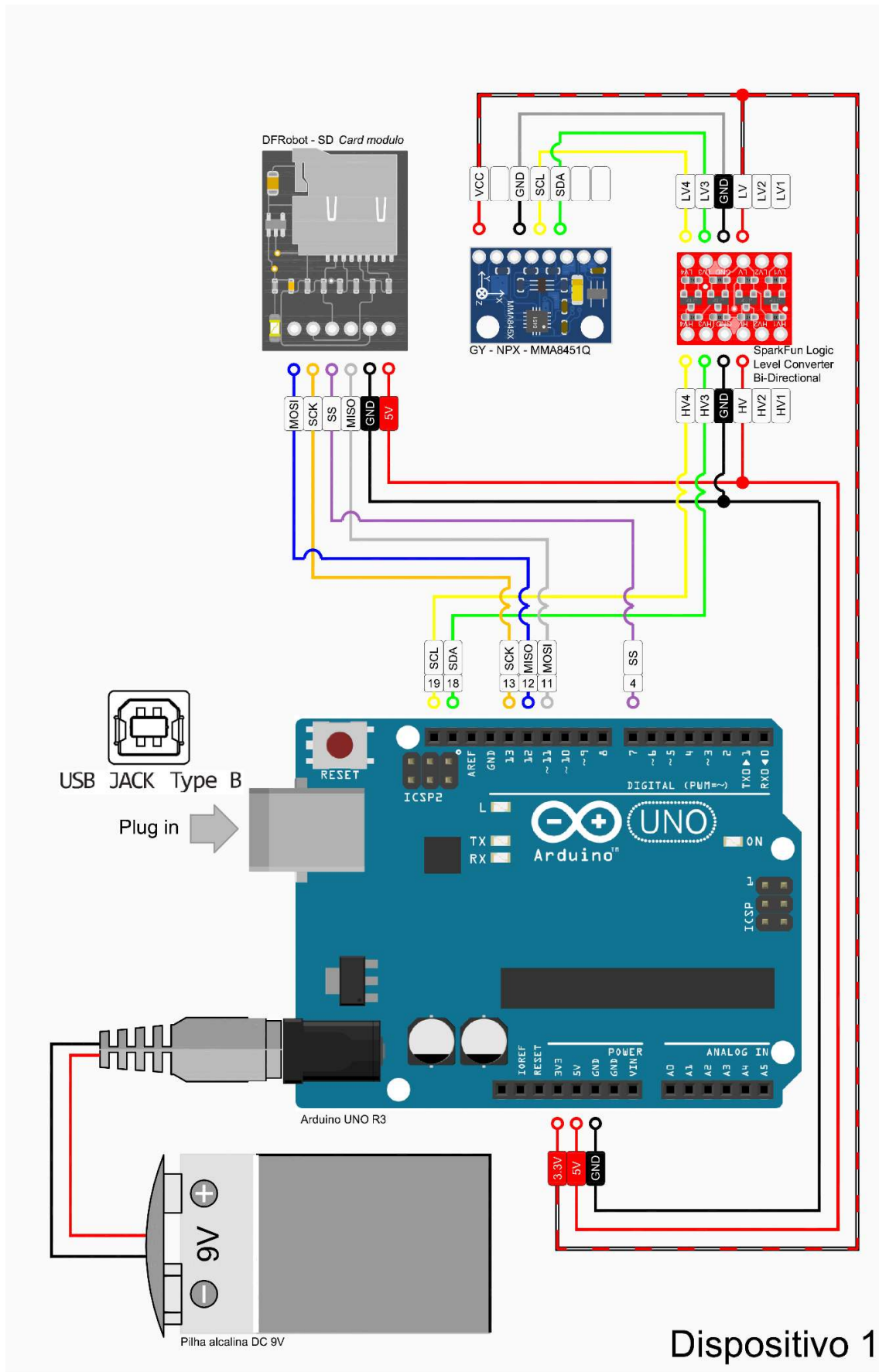


Figura 4.10 - Diagrama de ligações e montagem do Dispositivo 1

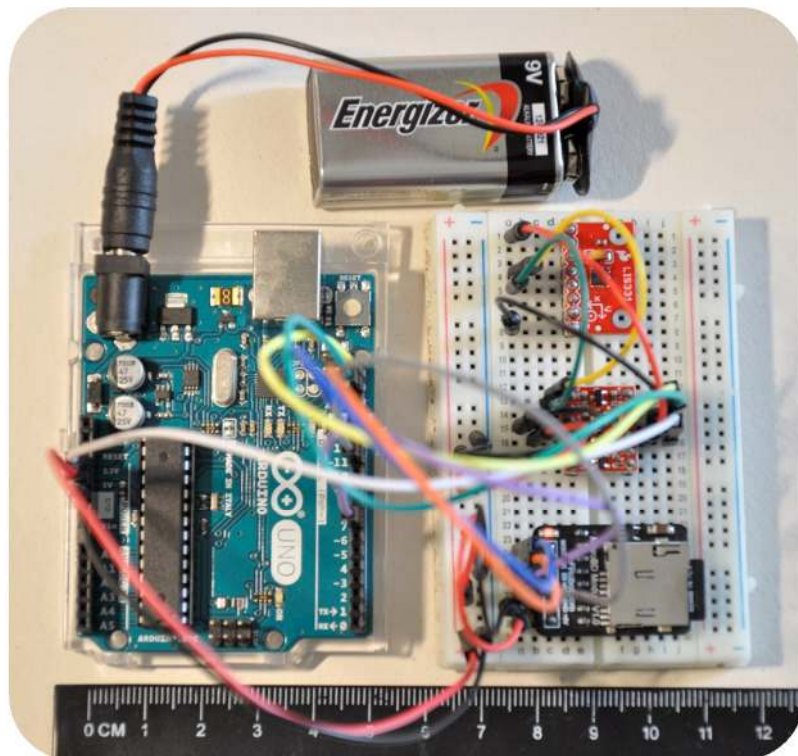


Figura 4.11 - Dispositivo 1 com o sensor LIS331HH

4.3. IDE Arduino

Este capítulo aborda algumas questões sobre a instalação do IDE da Arduino e a criação do código necessário implementar no microcontrolador da placa de desenvolvimento para que esta opere todos os componentes de acordo com os objetivos referidos anteriormente.

O desenvolvimento do *Sketch* é processo iterativo, executado em pequenos passos por forma a detetar erros até que atinja num código que opere sem problema. Por vezes é necessário otimizar o código, este processo só termina até que se obtenha uma solução de compromisso, dentro dos critérios pré estabelecidos.

O ponto seguinte, resume algumas das dificuldades encontradas durante o desenvolvimento do sketch e soluções encontradas para as ultrapassar.

4.3.1. Instalação da IDE

O software do IDE da Arduino é obtido diretamente do site da Arduino na separador que diz *Software*. Nesse site podemos encontrar para além da opção de *download* do IDE, a opção de usar um editor de código *online* e muitas aplicações associadas.

O *software* está disponível para os sistemas operativos Windows, Mac OS X e Linux. A versão utilizada durante a realização deste projeto foi a Arduino 1.8.3

A figura 4.12 mostra o aspeto do IDE da Arduino, com a estrutura com que inicia um novo *sketch*. Verifica na figura que existem dois comandos inscritos, estes são *setup()* e *loop()*, como referido anteriormente, sem este dois comandos sem o qual o *sketch* não opera corretamente dentro do microcontrolador da placa de desenvolvimento.

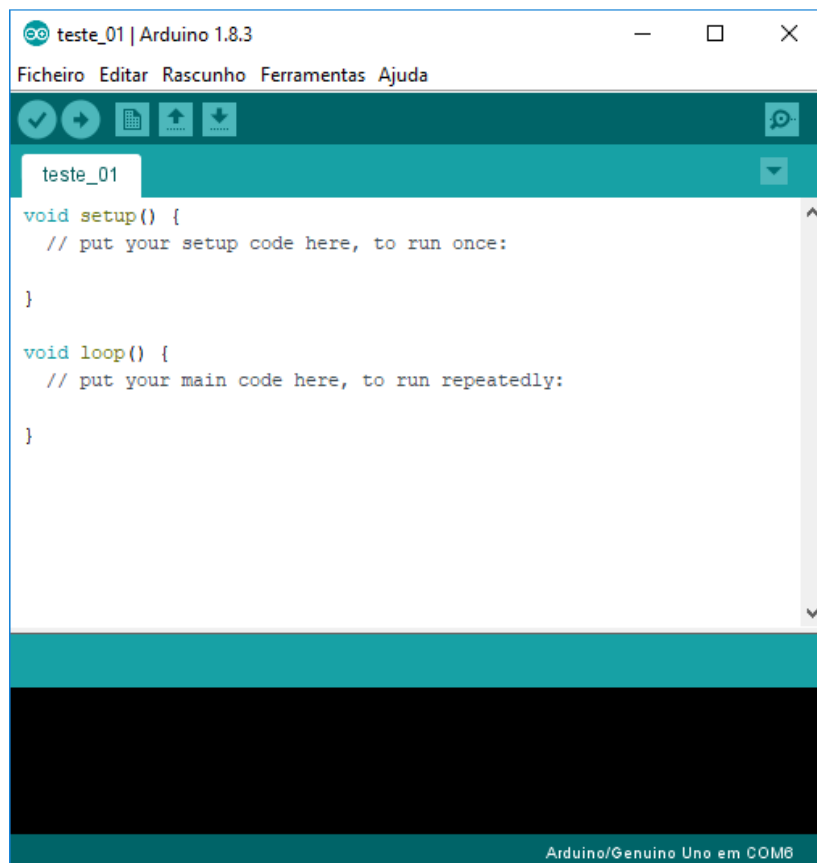


Figura 4.12 - IDE do Arduino

4.3.2. Instalação da placa de desenvolvimento

Uma vez iniciado o IDE da Arduino é necessário configurar o IDE Arduino para a placa que está a usar nesse momento. Ao abrir o separador “Ferramentas”, encontramos a opção “Placa:”, no momento que a selecionamos surge uma lista com inúmeras placas, deve-se então selecionar a placa pretendida.

A lista apresentada apenas mostra as placas Arduino, caso se pretende usar uma placa de outro fabricante, é necessário estabelecer uma ligação este aplicativo de gestão de placas e os servidores do fabricante em causa, desta forma são incorporadas automaticamente para a lista todas as placas que esse fabricante produza.

O processo de incorporação de novas placas no gestor de placas é o seguinte, primeiro é necessário obter a morada do servidor onde está alojado o catálogo das placas desse fabricante. De seguida deve-se transcreve-la para o campo “URL Adicionais do Gestor de Placas:” que se encontra sessão Ficheiro, Preferências, após esta configuração irão surgir novas placas em “Placas:”. É possível escrever múltiplas moradas neste campo desde que se separe cada morada com uma vírgula “,”.

O passo seguinte é ligar a placa ao computador através de cabo USB, esta também serve para estabelecer ligação a uma fonte de energia externa. Após feita a ligação USB deve-se confirmar o estado, observando se há emissão de luz do LED ON.

Se a placa estiver operar em condições normais, a opção “Porta:” fica disponível na sessão Ferramentas, devendo aparecer uma lista de “Portas série”, devendo escolher a opção que condiz com a marca e modelo da placa ligada ao computador. Após aplicação desta opção deve aparecer a nome da placa e a qual porta COM está ligada, no canto inferior do IDE Arduino, como se pode verificar no canto inferior direito da figura 4.12.

4.3.3. Bibliotecas

As bibliotecas são uma coleção de códigos que proporcionam a interação entre sensores, módulos ou outros componentes eletrónicos.[24] O IDE Arduino possui uma vasta lista de bibliotecas que se podem consultar e incluir no *Sketch*, estas encontram-se na sessão Rascunho, opção “Incluir Biblioteca”.

Para além das bibliotecas incluídas no IDE da Arduino, existem muitas outras disponíveis na internet, em plataformas de *open-source*, etc. Existe a possibilidade de cada utilizador criar a sua própria bibliotecas e disponibiliza-la *online*, assim acrescentado conhecimento.

Estas bibliotecas têm que ser incorporadas no IDE da Arduino, e existe duas formas de o fazer, a primeira implica que a biblioteca esteja compactada sobre a forma de um ficheiro ZIP e de acordo com uma estrutura própria definida pelo IDE. Dentro de o IDE Arduino na sessão Rascunho, na opção “Incluir Bibliotecas” aparece uma lista bibliotecas agrupadas por cada programador, nessa lista existe uma opção a “Adicionar Bibliotecas .ZIP”, ao clicar sobre esta abre um caixa de diálogo para procurar a localização do ficheiro desejado. A segunda opção é incorporar os ficheiros ou dentro da pasta Bibliotecas localizado em `c:\programs files\Arduino`, ou dentro da pasta `c:\users\user\documents\Arduino`.

4.3.4. Contributos

Os *sketchs* necessários para operar o Dispositivo 1, com os diferentes sensores, precisaram de várias bibliotecas, que foram encontradas e guardadas durante o processo de seleção dos sensores, para que numa fase posterior fossem incorporadas no IDE da Arduino. A seguir apresenta-se a tabela 4.2 que reúne bibliotecas usadas para escrever o programa utilizado no Dispositivo 1. Nesta encontram-se informações por componente usado, indicando o contribuidor que a escreveu a biblioteca e a sua localização na internet.

Tabela 4.2 - Bibliotecas para componentes eletrónicos em linguagem Arduino

| | |
|---|---------------------|
| Componente: | Sensor de movimento |
| Biblioteca: | LIS331 |
| Contribuidor: | Sparkfun |
| Localização | |
| https://github.com/sparkfun/Triple_Axis_Accelerometer_Breakout-LIS331/tree/V_1.2 | |

| | |
|---|---------------------|
| Componente: | Sensor de movimento |
| Biblioteca: | MMA8451 |
| Contribuidor: | Adafruit |
| Localização | |
| https://github.com/adafruit/Adafruit_MMA8451_Library | |

| | |
|---|---------------------|
| Componente: | Sensor de movimento |
| Biblioteca: | MPU6050 |
| Contribuidor: | Jeff Rowberg |
| Localização | |
| https://github.com/jrowberg/i2cdevlib | |

| | |
|---|---|
| Componente: | Leitor de cartão - necessária para ler e escrever em ficheiros FAT16/FAT32 usado pelos cartões de memória SD/SDHC |
| Biblioteca: | SDFat |
| Contribuidor: | Bill Greiman |
| Localização | |
| https://github.com/greiman/SdFat | |

| | |
|---------------|---|
| Componente: | Sistema – Código necessário para operar o protocolo de comunicação I ² C e TWI |
| Biblioteca: | Wire |
| Contribuidor: | Arduino |
| Localização | |
| IDE Arduino | |

| | |
|---------------|--|
| Componente: | Sistema – Código necessário para operar o protocolo de comunicação SPI |
| Biblioteca: | SPI |
| Contribuidor: | Arduino |
| Localização | |
| IDE Arduino | |

4.3.5. Atualizações do IDE

O IDE da Arduino possui dois gestores, um de placas e outro de bibliotecas, estas encarregam-se de informar automaticamente se existe uma atualização disponível para cada um desses componentes, o aviso é feito através de uma mensagem na base da janela do IDE.

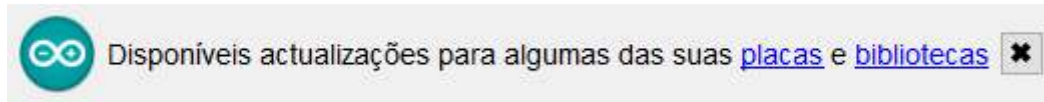


Figura 4.13 - Mensagem de aviso sobre atualizações disponíveis para IDE Arduino

É importante manter as bibliotecas atualizadas, pois poderão existir *bugs* que dão origem a erros de execução do programa dentro do microcontrolador da placa de desenvolvimento.

Este serviço não é extensível às bibliotecas importadas, por isso cabe ao utilizador verificar regularmente na plataforma que disponibilizou a biblioteca, a existência de atualizações.

4.4. Desenvolvimento de Sketch ou Programa

O Dispositivo 1 está projetado para operar todos os sensores selecionados, bastando proceder a troca física do sensor de movimento, respeitado a ordem das ligações indicadas no diagrama de ligação e montagem do Dispositivo 1.

No entanto o microcontrolador necessita de um programa ou *sketch* diferente para cada sensor de movimento, carregado para dentro do microcontrolador através do IDE Arduino. As alterações físicas ou *hardware*, são simples e fáceis, bastando trocar o sensor de movimento. Mas em termos de programação ou *sketch* são ligeiramente diferentes entre cada sensor de movimento, pois estes operam com bibliotecas diferentes, mas todos os *sketchs* obedecem a uma estrutura idêntica.

A estrutura de funcionamento dum *sketch*, subdivide-se três partes distintas, numa estrutura vertical. A primeira parte, antecede as funções *setup()* e *loop()*, é a área destinada às declarações de bibliotecas, variáveis e constantes, sistemas de ficheiro, etc. A segunda parte diz respeito à função *setup()*, todo o código aqui inserido é executado apenas uma vez, como é caso da iniciação das bibliotecas, sensores, e definições. A terceira parte diz respeito à função *loop()*, todo o código aqui inserido é executado ciclicamente.

A figura 4.14 apresentada abaixo representa o fluxograma usado para operar o Dispositivo 1

Declarações

Declaração de bibliotecas

- Arduino - Wire (*communication protocol*)
- Arduino - SPI (*communication protocol*)
- Adafruit - MMA8451
- Adafruit - Sensors
- Bill Greiman - SDFat
 - Definição do sistema de ficheiro usado

Declaração de constantes

- Definição da morada I2C do sensor inercial
- Pin usado pelo protocolo SPI para controlar o módulo leitor de cartões SD

Declaração de variáveis

- Tempo
- Número de contagem de registo de ciclo (**n**)
- Número de ciclo de espera entre cada salvaguarda do ficheiro .txt no cartão SD (**q**)
- Variáveis associadas a valores a extrair de um determinado sensor

Void setup()



Void loop()

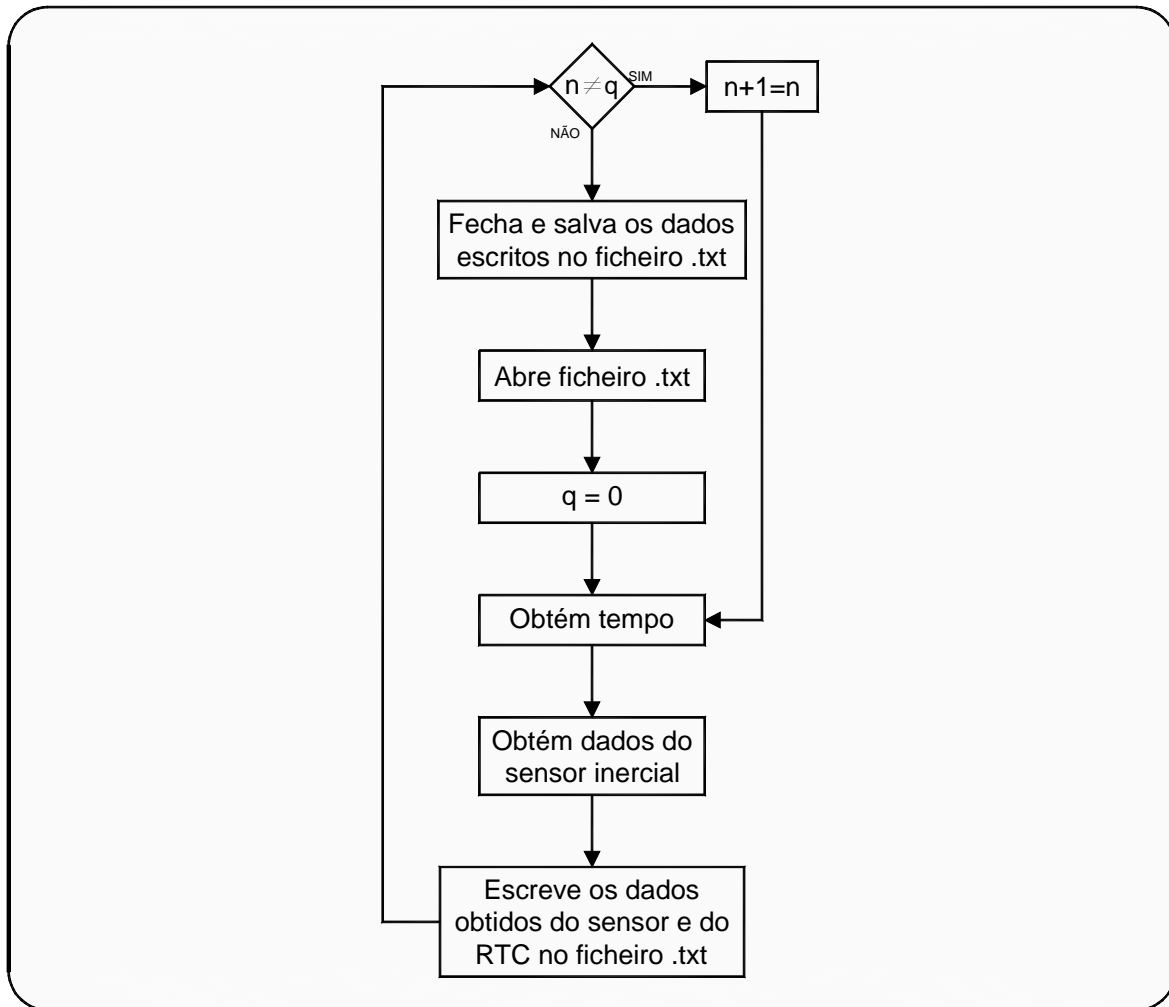


Figura 4.14 - Fluxograma de operação do *sketch* do Dispositivo 1

Os três *sketchs* utilizados durante a fase de seleção e ensaios dos sensores de movimentos, foram desenvolvidos de acordo com o diagrama de operação, apresentado na figura 4.14 e colocados no Anexo B1.

4.4.1. Definições

Ao longo de cada *sketch* são colocadas notas ou informações, identificadas pela cor cinza, estas antecedem o código e contribuem para entender o seu funcionamento ou opções tomadas. A informação é antecedida pelos seguintes símbolos “\” ou encapsulada entre “/* */”.

O IDE Arduino tem uma biblioteca própria para operar módulos de cartões SD, no entanto durante a fase de pesquisas de bibliotecas, encontrou-se uma biblioteca mais avançada do que a IDE Arduino, por este motivo que se optou por esta é que existem funções únicas. A biblioteca SDFat tem dois sistemas ficheiros, de declarar na primeira parte do *sketch* qual o sistema utilizado, sendo possível utilizar os dois.

A linguagem Arduino possui um comando para obter o tempo do RTC das placas designado por *time*. O resultado obtido é na unidade de tempo microssegundos.

O protocolo I²C da placa Arduino Uno tem uma velocidade de transmissão de dados standard igual a 100kHz, mas é possível aumentá-la até 400kHz ou diminuir até 12,5kHz. A alteração da velocidade de transmissão de dados é obtida com introdução do comando TWBR e um valor tabelado após iniciação da biblioteca, este procedimento é demonstrado no artigo “*I²C - Two-Wire Peripheral Interface - for Arduino*”. [27]

Para iniciar o protocolo SPI é necessário chamar a função dentro da parte destinada ao *setup()*, mas como se usou a biblioteca SDFat, não é necessário introduzir esta função, porque quando se inicia o módulo do cartão SD com biblioteca SDFat esta é automaticamente incluída no *sketch*.

Ao iniciar módulo SD com biblioteca SDFat, usando comando *sd.begin()*, este possibilita iniciar automaticamente o protocolo SPI, ou a metade da velocidade máxima ou na velocidade máxima. Deve-se também dentro deste comando definir o pin que seleciona o módulo em função escravo, nestes *sketch* definiu-se o pin 7.

O sensor MPU6050 é sensores inercial com 6 graus de liberdade, isto quer dizer que possui para além da capacidade para registar movimento, em termo de acelerações é capaz de registar também velocidades angulares. Este sensor também está equipado com um termómetro, necessário para compensar os desvios associados à variação do tempo. O *sketch* elaborado para este sensor está preparado para registar.

4.4.2. Teste de avaliação dos Sketchs

Foram feitos testes para avaliar o comportamento e performance do *sketch* ou programa que corre dentro do Dispositivo 1, assim verificando a conformidade dos resultados

registados com os critérios estabelecidos. Durante este processo foram feitas algumas alterações e correções a este programa por forma a atingir esses critérios, que se passam a descrever.

Velocidade de transmissão dos protocolos, registos no cartão SD e taxas de amostragens

A primeira avaliação da taxa de amostragem regista pelo Dispositivo 1, produziu valores muito baixo. A taxa de amostragem rondava os entre os 30Hz a 80Hz, dependendo do sensor usado. Após algumas alterações ao *sketch*, verificou-se que ao desligar o protocolo Serial foi possível um aumento das taxas de amostragens.

A porta *Serial* é usada para transmitir bidireccionalmente informação entre a placa e o computador pessoal, durante a operação do programa no interior da placa. Este é usado por exemplo para visualizar a informação produzida pela placa de desenvolvimento, através de monitor de série existente no IDE da Arduino, dar instruções ao programa, etc.

Uma das limitações da placa Arduino UNO, concretamente do microcontrolador usado, é que esta só tem canal por onde informação ou os dados viajam, assim ao desligar o protocolo Serial disponibiliza este canal para outras funções.

Nos *sketchs* encontram-se notas (em cor cinza) relativas a função *Serial.begin()* e *Serial.print()* usadas para iniciar e transmitir informação para o monitor série pelo protocolo Serial. Caso se pretenda ativa-las basta remover os símbolos `\\` ou `/* */`.

A resolução desta situação não foi suficiente para aumentar as taxas de amostragens para o valor desejado de 250Hz. Numa análise aos dados guardados no cartão, foi possível verificar que o processo de abertura do ficheiro, escrita, salvaguarda da informação dentro do cartão SD leva 650ms a 850ms. Este processo é utilizado por muitos programadores, de cada vez que se quer pretender guardar um registo dentro do cartão SD, mas necessita de muito tempo e assim pouco eficiente. A solução passou por não fechar e salvaguardar os dados de cada vez que se escreve, que é o processo normal, assim deixa o ficheiro aberto e só se fecha e salvaguarda a um período de tempo definido. O período mínimo determinado para que se obtenha uma taxa de amostragem aceitável foi 90 em 90 segundos.

Distância entre o sensor e placa de desenvolvimento

Durante a fase de pesquisa verificou-se em várias fontes que declaravam não ser possível transmitir a grandes distâncias quando usado o protocolo I²C. Uma fonte em concreto indica que a distância máxima é 2,25m. [28] Executou-se um ensaio para verificar esta situação, foi possível transmitir e receber informação usando um cabo com 2,1m de comprimento. Voltou-se a realizar a experiência com cabo de 10m e com este não possível obter qualquer resultado.

Possibilidade múltiplos sensores operarem em simultaneamente

É possível ligar múltiplos sensores a uma placa de desenvolvimento contudo esta opção diminui a taxa de amostragem, os recursos são divididos por entre os sensores.

Espaço ocupado pelo programa na memória do microcontrolador

Os diferentes *sketchs* usados utilizam sensivelmente 40% do espaço de armazenamento do programa e 60% da memória dinâmica. Isto significa que a memória remanescente é suficiente para o microcontrolador operar corretamente.

Quantidade de informação produzida por hora

Um sensor MPU6050 a operar com uma taxa de amostragem a 248Hz, produziu numa hora sensivelmente 24Mb. Este sensor é de 16bit, sensores com menos resolução produzem relativamente menos dados. Com um cartão SD *standard* de 8Gb é possível gravar dados durante 14 dias consecutivos.

Consumo de energia elétrica e duração da bateria 9V

Verificou-se por experiência que um pilha alcalina de 9V, usada como fonte de energia interna, pode fornecer energia durante 8h de utilização consecutiva.

4.5. Solução Joy Worrier 24F14 - Sensor BMA 180

O dispositivo desenvolvido pela empresa Joy Worrier, modelo 24F14, emprega o sensor BOSCH BMA180. Este dispositivo foi desenvolvido para ser aplicado em *joysticks*, é comercializado como uma *breakout board*, e apenas é disponibilizado o acesso ao sensor pelo protocolo comunicação USB.

A solução desenvolvida pela Joy Worrier é um produto *Comercial off the Shelf* COTS, numa tradução livre quer dizer, solução pronta disponível numa prateleira de loja. É vendida com uma caixa cúbica externa em plástico, na qual se instala a *breakout board* e uma ligação física por cabo USB.



Figura 4.15 - Joy Warrior – 24F14, com o sensor BMA180



Figura 4.16 - Solução COTS da Joy Warrior, dispositivo de aquisição de vibrações

A Joy Worrier disponibiliza para este e outros produtos similares, vários *softwares*, cada um com uma função específica. Por exemplo disponibilizam um *software* para calibrar o dispositivo, outro para fazer aquisição de dados, uma aplicação tipo sismógrafo, etc.

Este dispositivo recebeu a atenção de uma organização de interessada por sismografia amadora, e com este modelo desenvolveu uma rede de sismógrafos, cada um ligado a um computador pessoal, a fazer monitorização sísmica. Que após um evento sísmico os dados recolhidos são enviados automaticamente, para a sede da organização a fim de serem processados e analisado.

4.5.1. Avaliação do dispositivo Joy Warrior 24F14

A avaliação deste dispositivo começou pela aquisição de dados do sensor, com uma escala dinâmica igual aos restantes sensores, neste caso igual $\pm 2g$. Quanto à taxa de amostragem verificou-se que não é possível obter uma superior a 125Hz, esta limitação é imposta pelo *design* elaborado pelo fabricante, para esta placa, com este protocolo de comunicação.

Ao analisar a primeira colheita de dados, verificou-se que os dados não eram os esperados, verificando um valor de *counts* do eixo Z, sensivelmente duas vezes ao esperado. Então procedeu-se para a calibração, ao executar o problema subsistiu. Experimentou-se alterar a escala dinâmica para valores mais reduzidos, que é possível neste sensor, e uma vez mais o problema continuou, e não se encontrou uma solução para esta situação.

Os valores dos restantes eixos X e Y, apresentaram um comportamento expectável.

Mas sem um perfeito conhecimento do problema que está afetar o eixo Z deste dispositivo, não é possível garantir que os restantes eixos padeçam, em parte da anomalia do eixo Z.

Devido a esta anomalia encontrada neste dispositivo, não se executou nenhuma avaliação neste sensor.

Pelo fato não se poder fazer uma avaliação a este dispositivo, devido a anomalia encontrada, também ao fato este não permitir uma taxa de amostragem superior a 125Hz.

Equacionou-se encomendar uma nova *breakout board* com o sensor BMA180, mas o processo de procura, encomenda, transporte, montagem, programação e teste implica um atraso neste trabalho com um tempo que pode chegar aos 40 dias.

Por estas razões excluiu-se este sensor do processo de seleção de um sensor inercial MEMS.

5. Ensaios de avaliação de sensores

5.1. Introdução

Concluído o desenvolvimento do Dispositivo 1 em termos de *hardware* e *software*, de acordo com os critérios apontados, voltou-se novamente a atenção para os sensores inerciais. Foi necessário identificar entre qual dos sensores selecionados possuía as melhores características em função das necessidades espectadas em campo.

Para identificar o melhor dos sensores selecionados, neste caso aquele que possuía um conjunto de características mais adequadas ao tipo de registo de vibrações ou movimentos identificados em estruturas de construção civil, desenvolveram-se dois tipos de ensaios similares situações esperadas, com intuito observar o comportamento de cada um.

Um primeiro ensaio estático consistiu em recolher e analisar informação de um sensor, num local com reduzidas ou isento de vibrações ambientais externa, para que assim fosse registado o ruído branco produzido internamente. Sendo ruído a acumulação do ruído proveniente de várias fontes. A importância da análise do ruído é que este contribui para diminuição da resolução das medições, bem como a capacidade de medir micro vibrações, assim deve ser verificado qual o nível mínimo de ruído registado por cada um dos sensores, determinado aquele com o menor valor.

Um segundo ensaio dinâmico pretendeu-se caracterizar os sensores quanto à sensibilidade ao movimento, através de um sistema simples. Para executar a avaliação desenvolveu-se um sistema com um movimento harmónico simples, também conhecido por um pêndulo. Assim equipou-se o Dispositivo 1 a vez com cada um dos sensores inerciais, realizando uma série de ensaios, com o intuito de registar valores durante o movimento pendular. Após os ensaios extraíram-se os registos e executaram-se análises comparativas entre dados dos vários sensores para determinar aquele com maior sensibilidade ao longo do tempo.

5.2. Ensaios Estáticos

O nível de ruído influencia a mínima resolução de um sensor, portanto, é importante caracterizar a performance de cada sensor selecionado, assim contribuindo para identificar qual destes tem a melhor resolução.

A produção do ruído provém de fontes termomecânicas internas e externas e fontes elétricas internas e externas. Durante uma avaliação do ruído, é possível minimizar a produção de ruído de fontes externas, através do isolamento, assim melhorando a qualidade dos resultados.

A avaliação do ruído passa pela recolha e registo de dados, num local com fontes de ruído externas controladas e posterior análise estatística dos dados recolhidos. Controlando as fontes de ruído, oferece uma garantia que os resultados obtidos dos sensores não são de movimento ou acelerações e sim de fontes de ruído, não controlado. Os resultados dessas fontes são valores aleatórios, irregulares, causando distorção dos resultados reais.

A ferramenta estatística por excelência para avaliar dados é o Desvio Padrão, esta permite determinar a quantidade da variação ou dispersão de conjunto de dados. A fórmula do desvio padrão não permite o seu cálculo em tempo real, mas é possível determiná-lo em tempo real, através da relação de outras duas ferramentas estatísticas, neste caso a média e a *Root Mean Square* (RMS) que pode-se traduzir em média quadrática.

Devido a possibilidade do RMS e média serem calculados em tempo real, são ferramentas muito utilizadas na avaliação de sinais em tempo real. São adequadas para microcontroladores de baixa capacidade de processamento, devido a pouca quantidade de cálculos necessários para se obter os resultados pretendidos.

As fórmulas utilizadas no tratamento estatístico são as seguintes:

Desvio padrão da amostra

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (4)$$

Média

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (5)$$

RMS

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad (6)$$

Relação entre RMS, média e desvio padrão

$$x_{RMS}^2 = \bar{x}^2 + \sigma_x^2 = \overline{x^2} \quad (7)$$

5.2.1. Objetivos

Objetivo é executar um ensaio estático para cada um dos sensores selecionado, proceder ao registo, aquisição de dados usando o Dispositivo 1, e por fim proceder a análises estatísticas aos dados recolhidos numa folha de cálculo. E por fim através da comparação de resultados classificar estes sensores por ordem de performance de ruído.

5.2.2. Procedimento de ensaio estático

Os passos executados durante cada ensaio foram os seguintes:

- Primeiro instalou-se no Dispositivo 1 o sensor a avaliar.
- Conectou-se o Dispositivo 1 ao computador pessoal por uma ligação USB;
- Abriu-se plataforma IDE Arduino e o *sketch* correto para operar cada sensor;
- Alterou-se o nome do ficheiro .txt, dentro *sketch*, assim possibilitando criação de um novo registo com uma nova referência;
- Fez-se a instalação do *sketch* por *upload* para o Dispositivo 1, após o *upload* desconectou-se o Dispositivo 1 do computador;
- Instalou-se um cartão SD no computador, limpou-se toda a informação, desconecta-se do computador pessoa e instalasse-o no módulo de cartão SD.
- Aplicou-se o Dispositivo sobre o local de ensaio.
- Conectou-se o Dispositivo 1 à fonte de alimentação.

- Aguardou-se o tempo definido do ensaio.
- No fim desconectou-se o Dispositivo 1 da fonte de alimentação concluindo o ensaio.

5.2.3. Constituição do dispositivo e sistema

A constituição do dispositivo 1 foi demonstrada num capítulo anterior, apenas foi alterado o sensor em cada ensaio, e instalado o *sketch* adequado para o operar cada um dos sensores.

5.2.4. Local de ensaio

Por forma a reduzir as fontes de ruído termomecânicas, instalou-se o dispositivo 1 na fundação de uma cave, longe de atividade humana, assim e afastando-o de fontes de vibração, como por exemplo máquinas ou trânsito de veículos. As fontes de ruído elétrico foram minimizadas pelo uso de uma bateria química de 9V.

O dispositivo foi assente no pavimento e de seguida ligou-se a bateria de 9V, no *barrel jack* iniciando o programa e registo. O tempo de registo para todos os sensores é de 15m.

5.2.5. Pós-processamento

Após concluído cada ensaio removeu-se o cartão SD e transferindo a informação para um computador pessoal. Os dados foram então incorporados numa folha de Excel, para que nesta fossem feitos tratamentos e cálculos estatísticos.

As fases de tratamento foram as seguintes:

- Conversão os dados de formato *counts* para aceleração g.
- Calculada a média, RMS e desvio padrão de acordo com as fórmulas apresentadas.
- Apresentação dos resultados em formato de tabelas.

5.2.6. Apresentação de resultados

A tabela 5.1 apresenta os resultados dos ensaios estáticos para cada um dos três sensores selecionados.

Tabela 5.1 - Resultados dos ensaios estáticos

| | MMA8451Q | | | MPU6050 | | | LIS331HH | | |
|---------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | X [mg] | Y [mg] | Z [mg] | X [mg] | Y [mg] | Z [mg] | X [mg] | Y [mg] | Z [mg] |
| Média | 35,62 | 21,34 | 987,34 | -24,98 | 37,75 | 984,71 | -77,74 | 81,59 | 974,11 |
| RMS | 35,65 | 21,41 | 987,34 | 25,20 | 37,90 | 984,72 | 78,35 | 82,40 | 974,24 |
| Desvio Padrão | 1,47 | 1,65 | 1,84 | 3,31 | 3,30 | 4,39 | 9,77 | 11,52 | 15,96 |

5.2.7. Conclusões sobre o ensaio estático

As primeiras observações feitas aos resultados obtidos foram sobre a da média e RMS, nos eixos X e Y. Observou-se que os valores apresentados pelos sensores, nestes dois eixos em particular, não apresentam qualquer relação entre eles, sendo incoerentes. A razão destes resultados, deveu-se ao mau nivelamento dos sensores durante cada ensaio, originando desvios nos eixos X e Y com o plano perpendicular ao eixo da gravidade. Sem equipamento metrológico foi muito difícil, nivelar a superfície e obter os resultados esperados, que deveriam ser idênticos em ambos os eixos e perto de zero.

Os resultados esperados das médias e RMS, numa situação ideal em que o ensaio é feito numa superfície perfeitamente nivelada e sem ruído, são valores aproximaram-se de zero. Com a existência de ruído no sistema, os resultados da média deviam aproximar-se de zero visto que não houve movimento, é os valores do RMS deviam de ser ligeiramente acima de zero, assim indicando a presença de ruído existente.

O RMS sendo uma média quadrática os valores são sempre positivos e relativamente superiores a média. Ao analisar os valores da tabela 5.1, verificou-se essa situação. Portanto os resultados RMS obtidos neste cenário, dizem respeito apenas ao ruído total do sistema, uma vez que a quantidade obtida do RMS é inferior a obtida pelo desvio padrão.

Os valores apresentados no eixo dos Z, aproximam-se dos valores esperados de 1000mg ou 1g, representando uma unidade de aceleração gravítica terrestre.

Pelo facto de se não conseguir um nivelamento idêntico entre todos os sensores, recorreu-se aos resultados obtidos do desvio padrão de cada sensor para comparar resultados entre cada sensor.

Assim verificou-se que o sensor MMA8451Q tem o melhor desempenho, seguindo do MPU6050 e por fim o LIS331HH. Era esperado que o LIS331HH tivesse a pior prestação, pois tem a maior escala dinâmica dos três sensores.

Foi possível através da norma euclidiana obter um vetor normalizado, e verificar o valor da aceleração g obtido em cada sensor, assim verificando o erro do fator de escala. Abaixo apresenta-se a tabela 5.2 com os resultados.

Tabela 5.2 - Normalização de resultados dos ensaios estáticos

| | MMA8451Q | MPU6050 | LIS331HH |
|---------------|-----------|-----------|-----------|
| | X [mg] | X [mg] | X [mg] |
| Média | 988,21 | 985,75 | 980,61 |
| RMS | 988,22 | 985,77 | 980,85 |
| Desvio Padrão | 2,88 | 6,41 | 21,97 |

Verificou-se que os valores obtidos afastaram-se dos valores esperados, verificando-se pequenos erros de fator de escala. O sensor MMA8451Q apresentou o melhor resultado dos três sensores, pois aproximando-se mais de 1000mg.

A solução para contrariar este problema e outros que não abordados neste trabalho, é executar ensaios metrológicos que permitem identificar e quantificar cada um dos erros e desvios. Este processo tem algumas complicações implementação práticas e morosas.

Existem técnicas de calibração simples, compiladas em notas técnicas produzidas pelos vários fabricantes, que ajudam a ultrapassar alguns dos problemas evidenciados. Estas técnicas têm como objetivo normalizar os dados obtidos do sensor, por meio de um processo que envolve muitas equações em cadeia.

Os processos de calibração sugeridos pelos fabricantes, requerem a utilização de planos de calibração, sobre os quais se apoiam os sensores ou dispositivos, por forma a adquirir valores, que por sua vez são transformados com recursos a fórmulas específicas para se obtenham valores corretos de aceleração. O processo de calibração mais simples é feita apenas em dois planos, designando-se por calibração *standard*, existe outro com 6 planos para casos que se pretenda uma calibração mais precisa.

O processo de calibração implica que o próprio dispositivo tenha uma superfície plana, e que o sensor esteja alinhada com essa superfície. É também necessário para o processo, um objeto tridimensional com superfícies planas e com ângulos regulares entre cada superfície, como por exemplo um cubo ou prima hexagonal. Esse objeto deve ter ângulos retificados, para uma melhor precisão da calibração.

Os dados obtidos do processo de calibração, necessários à correção dos desalinhamentos, erros São incorporados num logaritmo de correção dentro da rotina do *software* do microprocessador. Há fabricantes que disponibilizam soluções incorporadas no próprio sensor de auto nivelamento, bastando ativar a função ao iniciar o sensor.

5.3. Ensaios Dinâmicos

5.3.1. Introdução

Num sistema pendular gravítico simples, com um movimento harmónico simples, são conhecidas equações que permitem descrever esse movimento, em termos de velocidade, período, aceleração, amplitude de oscilação, posição no espaço bidimensional, ao longo do tempo.

Neste sistema pode ter dois comportamentos ao longo do tempo, caso seja amortecido ou não amortecido. No caso não amortecido o movimento é constante e perpétuo, não sofrendo alterações comportamentais, sendo este apenas um caso hipotético. No caso do sistema amortecido o movimento tende para o repouso, sendo a situação que se aproxima da realidade.

A característica que se pretende observar com este ensaio é a variação da amplitude de oscilação ao longo do tempo em função do período do movimento. Num sistema não amortecido a amplitude e o período são constantes ao longo do tempo, enquanto num

sistema amortecido o período do movimento é constante mas a amplitude diminui com o tempo.

O sistema pendular simples amortecido o movimento tende a diminuir com o tempo, assim também as acelerações, velocidades e amplitudes de oscilação, mas o período permanece constante.

Com o Dispositivo 1 foi possível registar ao longo do tempo, todo o comportamento do movimento harmónico em termos de acelerações. Registando dados de ciclos completos desde o impulso inicial até ao repouso. Através de pós-processamento dos dados foi possível determinar a frequência do sistema, lembrando esta é inversa ao período, em intervalos de tempo discretos. O mesmo pós-processamento permitiu a determinação da amplitude da frequência nesses mesmos intervalos.

Para melhor entendimento do comportamento de cada sensor selecionado em condições diferentes, utilizou-se dois sistemas pendulares gravíticos, assim obtendo dados para duas frequências diferentes e amplitudes de oscilação diferentes.

5.3.2. Objetivos

O objetivo foi executar um ensaio dinâmico para cada um dos sensores selecionados em dois sistemas distintos, em cada ensaio executou-se recolha de dados, e posterior tratamento análise desses dados. Por fim através de interpretação e comparação dos resultados, tirou-se conclusões que ajudaram a determinar qual dos sensores inerciais inicialmente selecionados, tem a melhor sensibilidade.

5.3.3. Sistema pendular gravítico

O sistema pendular gravítico é composto por uma massa que oscilou em torno de ponto fixo. Na posição Inicial, o sistema está em equilíbrio, isto porque o centro de massa e ponto fixo encontram-se perfeitamente alinhados com a direção da força gravítica.

Quando a massa foi sujeita a uma força impulsiva, esta deslocou-se para fora do alinhamento inicial criando um braço que em combinação com a força gravítica, resultou numa força de restauração do equilíbrio, puxando a massa para o alinhamento inicial.

A massa ao aproximar-se do alinhamento inicial faz com que a força de restauração diminua até zero, mas em contrapartida, esta contribui para o aumento da velocidade e da energia cinética da massa. No momento em que a massa passa pelo alinhamento inicial, a energia cinética acumulada na massa, instabiliza o sistema e puxa a massa novamente para fora do alinhamento inicial.

Estas duas forças são produto da energia mecânica, instalada no sistema pendular, e que resultam em movimento oscilatório. Se não houvessem mais interações neste sistema, este oscilaria por um tempo infinito, mas devido ao atrito existe amortecimento e perda desta energia mecânica.

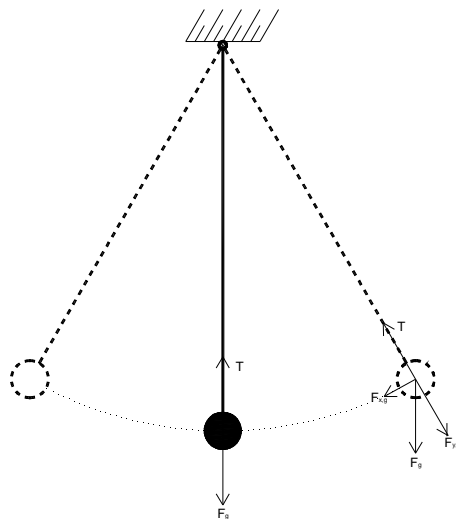


Figura 5.1 - Sistema pendular gravítico

5.3.4. Composição do sistema

O sistema foi constituído por simples objeto com uma massa conhecida, suspensa por um elemento de ligação flexível, de elevada resistência à tração e uma baixa deformação elástica, que uniu a massa a um ponto de apoio.

As ligações entre o elemento flexível, a massa e o apoio, é do tipo simples com três graus de liberdade, neste caso é permitido a rotação com uma amplitude de rotação de 180° nas zonas de apoios, nas três direções Z,Y,Z, num referencial local.

O objeto foi composto por uma embalagem cilíndrica plástica com um volume de 1,07 litros com uma altura de 18cm e 8,7cm de diâmetro, na qual se introduziu no fundo uma massa, por cima desta colocou-se o Dispositivo 1 e por fim, no topo um diafragma com o sensor inercial a avaliar no momento.

O diafragma é um elemento plástico com uma forma de taça, que é usado para fixar e posicionar o sensor inercial, assim permitindo o posicionamento de um eixo do sensor com o eixo da gravidade. Esta adaptação surgiu porque na solução proposta inicial verificou-se que o sensor aplicado na *breadboard*, é uma situação aceitável na maioria dos casos, mas neste em particular foi necessário alinhar corretamente um eixo vertical com o eixo da gravidade, o que não seria possível aplicando aleatoriamente o sensor na *breadboard*.

A figura 5.2 ilustra o Dispositivo 1 dentro da embalagem, com o sensor aplicado no diafragma, mas com a vista também para o interior da embalagem par se poder-se visualizar todo o aparato. A restante sequência de montagem foi a seguinte: primeiro aplicou-se o diafragma no topo de Dispositivo 1 o mais alinhado possível com o plano horizontal, por fim selou-se a embalagem com a tampa que estava fixa à ligação flexível.



Figura 5.2 - Embalagem com o Dispositivo 1 e diafragma com sensor

A massa total do objeto era de 0,593kg com um peso de 5,81N.

A ligação flexível consistiu num fio de aço com 1mm de diâmetro, fixo ao topo da embalagem de plástico, no centro geométrico da tampa e a na outra extremidade fixo a face inferior de uma laje de um edifício.

Como foi necessário dois sistemas pendulares com frequência de oscilação diferentes, foi preciso alterar as variáveis do sistema, que constituem a estrutura. Para determinar o período matematicamente aplicaram-se as equações do pêndulo, em concreto a expressão conhecida como a Lei de Huygens que permite determinar o período, em amplitude pequenas.

$$T_0 = 2\pi \sqrt{\frac{l}{g}} \quad (8)$$

Em que l é comprimento do pêndulo, g é a aceleração da gravidade e T é o período.

Ao verificar a Equação 4 a única variável possível de ser alterada é o comprimento do pêndulo. Visto que o período é inverso da frequência, ao alterar o comprimento do pêndulo, modifica-se a período do sistema.

Para obter dois sistemas pendulares arranjou-se dois fios de aço, um com 2,03m e outro com 0,05m. O comprimento l é determinado pela distância entre o apoio e centro de massa. O centro de massa do objeto encontrou-se a 14cm abaixo do topo da embalagem, onde se inseriu o fio de aço. Assim os dois comprimentos l são 217cm e 19cm, obtendo-se dois sistemas com duas frequências, sendo o valor calculado para o primeiro sistema de 0,338 Hertz Hz e o segundo de 1,143Hz.

5.3.5. Desenvolvimento do ensaio

Como referido desenvolveu-se dois sistemas pendulares para avaliar três sensores inerciais MEMS, foram assim executados 6 ensaios dinâmicos.

Os procedimentos de ensaio para cada um dos dois sistemas pendulares foram os seguintes:

- Primeiro instalou-se no Dispositivo 1 o sensor a avaliar.

- Conectou-se o Dispositivo 1 ao computador pessoal pela ligação USB;
- Criou-se um *sketch* na plataforma IDE Arduino, indicado para operar cada sensor instalado placa de desenvolvimento;
- Alterou-se o nome do ficheiro .txt, dentro *sketch*, assim possibilitando criar um novo registo com uma nova referência;
- Faz-se a *upload* do *sketch* para o Dispositivo 1, após o *upload* desconectou-se o Dispositivo 1 do computador;
- Instalou-se se um cartão SD no computador, limpou-se toda a informação, desconectou-se o cartão do computador pessoal e instalou-se cartão no módulo de cartões SD.
- Aplicou-se o Dispositivo 1 sem a fonte de alimentação dentro da embalagem.
- Aplicou-se o sensor no diafragma e conseqüentemente na embalagem.
- Ligou-se a bateria 9V a placa Arduino, iniciando o programa e o registo.
- Colocou-se a embalagem sobre uma superfície plana e faz-se um registo de 60 segundos em repouso para recolher dados, que foram utilizados para avaliar o nível de ruído no momento, e serem utilizados numa fase posterior, em pós-processamento.
- Aplicou-se a embalagem no pêndulo, deslocou se a embalagem para uma distância pré estabelecida. (a distancia estabelecida para o ensaio 1 foi aproximadamente 156cm fora do alinhamento inicial e para o ensaio 2 foi aproximadamente 15cm).
- Libertou-se a embalagem, que iniciando o movimento oscilatório.
- Após 36 minutos do início do movimento oscilatório, registou-se aproximadamente o movimento oscilatório, e de seguida retirou-se do pêndulo.
- Retirou-se o diafragma que contem o sensor, expondo o Dispositivo 1, de seguida desconectou-se a pilha de 9V e retirou-se o cartão SD.
- Desinstalou-se o sensor, disponibilizando o Dispositivo 1 para um novo ensaio com outro sensor.

5.3.6. Verificação de conformidade dos resultados do ensaio

Após cada recolha de dados verificou-se a qualidade e conformidade dos dados registados. Para tal procedeu-se à extração dos dados localizados dentro do ficheiro .txt, no cartão SD.

Transferiu-se os dados para uma folha de Excel, que de seguida procedeu-se à localizar dos registos feitos durante o período de repouso. Com os quais se elaborou um gráfico com os registos dos *counts* por tempo e por eixo. Por fim procurou-se no gráfico, zonas lineares onde os registos sejam praticamente constantes ou lineares, indicando o repouso do dispositivo em cima da plataforma plana, esse dados foram extraídos e executou-se uma análise estática.

A análise dos dados do tempo de repouso foi feita com recursos a métodos estatísticos, como a média dos valores, *Root Mean Square* (RMS) e desvio padrão, comparando os com valores determinados nessa amostra com valores padrão, já determinados durante o ensaio estático.

5.3.7. Pós-processamento de dados

Em nenhum dos casos verificou-se anomalias dos dados, sendo todos aceites para o pós-tratamento de dados, que consistiu nos seguintes passos:

- Conversão os dados em formato *counts* para aceleração unidades g;
- Regularizam os dados para uma nova taxa amostragem;
- Normalização dos vetores X e Y;
- Transformação de dados por algoritmo *Fast Fourier Transformation* (FFT);
- Execução de gráficos com histórico da amplitude de Fourier.

5.3.8. Regularização dos dados numa nova taxa amostragem.

Após observação da base de dados verificou-se que o período entre cada amostra, não é constante. Este fenómeno deveu-se a duas situações, a primeira à necessidade de tempo por parte microcontrolador para processar informação acumulada, gerando uma interrupção momentânea no ciclo ou *loop* do programa. Este problema pode ser ultrapassado com um microcontrolador com maior capacidade de processamento.

A segunda situação deveu-se ao processo de gravação dos dados no cartão, este precisa de bastante tempo para ser executar esta operação, assim verificando pequenas falhas periódicas no registo dos dados.

Pode-se verificar esta situação calculando na base de dados, dentro da folha de Excel, o período entre cada amostragem. Verificou-se que o valor do período entre amostras ao longo do tempo foi irregular, havendo por vezes lapsos bastante prolongados, na ordem dos centésimas ou milésimas de segundo. Mas a taxa de amostragem média a longo prazo foi alta e dentro dos critérios estabelecidos.

Então o segundo passo do processo de pós-tratamento dos dados, foi a conversão destes dados, num registo com período irregular para uma nova base de dados com período regular.

A nova base de dados corrigida, foi concebida a partir da base dados original. Na mesma folha Excel criou-se uma coluna com uma nova série de tempo. O passo usado para criar a nova série, foi determinado pela média dos valores do período entre cada amostra no total de valores registados, por fim arredondou-se o valor calculado em 5 casa decimais.

Abaixo são apresentadas duas tabelas 5.3 e 5.4, a tabela da esquerda apresenta dados da base original do Dispositivo 1 e a tabela da direita apresenta uma base de dados corrigida. Na tabela da esquerda o problema é evidenciado na coluna dos períodos, observa-se que o período entre cada amostra é irregular enquanto os valores apresentados na tabela da direita para a mesma coluna são regulares.

As acelerações apresentadas na tabela da direita foram obtidas a partir da tabela da esquerda por meio de interpolação entre períodos de tempo e *counts*. Todo o processo de cálculo foi executado numa folha de cálculo no programa Excel 2013, a metodologia de cálculo por interpolação esta descrita no *site* desenvolvido pela empresa Draga especializada em programas de conversão de bases de dados em gráfico, neste site encontra-se uma folha de cálculo com o exemplo de cálculo de interpolação. [29]

Tabela 5.4 - Base de dados do Dispositivo 1

Amostra de uma base de dados de registos do sensor MMA8451Q

| Tempo | Período | Aceleração | | | | | |
|----------|----------|------------|--------|--------|---------|---------|---------|
| | | X | Y | Z | X | Y | Z |
| s | s | counts | counts | counts | g | g | g |
| 0 | | 137 | 86 | 4045 | 0,03344 | 0,02099 | 0,98743 |
| 0,004156 | 0,004156 | 138 | 79 | 4038 | 0,03369 | 0,01928 | 0,98572 |
| 0,008308 | 0,004152 | 141 | 83 | 4061 | 0,03442 | 0,02026 | 0,99133 |
| 0,012456 | 0,004148 | 141 | 81 | 4053 | 0,03442 | 0,01977 | 0,98938 |
| 0,016612 | 0,004156 | 148 | 87 | 4055 | 0,03613 | 0,02124 | 0,98987 |
| 0,02076 | 0,004148 | 135 | 82 | 4051 | 0,03295 | 0,02002 | 0,98889 |
| 0,024912 | 0,004152 | 141 | 91 | 4050 | 0,03442 | 0,02221 | 0,98865 |
| 0,029072 | 0,00416 | 142 | 90 | 4055 | 0,03466 | 0,02197 | 0,98987 |
| 0,034032 | 0,00496 | 135 | 86 | 4038 | 0,03295 | 0,02099 | 0,98572 |
| 0,038184 | 0,004152 | 143 | 87 | 4040 | 0,03491 | 0,02124 | 0,98621 |
| 0,042336 | 0,004152 | 146 | 91 | 4037 | 0,03564 | 0,02221 | 0,98548 |
| 0,046496 | 0,00416 | 139 | 80 | 4045 | 0,03393 | 0,01953 | 0,98743 |
| 0,050652 | 0,004156 | 143 | 88 | 4036 | 0,03491 | 0,02148 | 0,98523 |
| 0,054808 | 0,004156 | 138 | 84 | 4036 | 0,03369 | 0,02051 | 0,98523 |
| 0,058956 | 0,004148 | 138 | 84 | 4040 | 0,03369 | 0,02051 | 0,98621 |
| 0,063108 | 0,004152 | 138 | 85 | 4049 | 0,03369 | 0,02075 | 0,98840 |
| 0,06726 | 0,004152 | 144 | 88 | 4039 | 0,03515 | 0,02148 | 0,98596 |
| 0,071408 | 0,004148 | 144 | 89 | 4050 | 0,03515 | 0,02173 | 0,98865 |
| 0,07556 | 0,004152 | 141 | 90 | 4033 | 0,03442 | 0,02197 | 0,98450 |
| 0,079712 | 0,004152 | 135 | 76 | 4033 | 0,03295 | 0,01855 | 0,98450 |

Média do período de amostragem 0,0042

Tabela 5.3 - Base de dados regularizada

Base de dados regularizada

| Tempo | Período | Aceleração | | | |
|--------|---------|------------|---------|---------|--------------------|
| | | X | Y | Z | $\sqrt{X^2 + Y^2}$ |
| s | s | g | g | g | g |
| 0 | | 0,03344 | 0,02099 | 0,98743 | 0,03949 |
| 0,0042 | 0,0042 | 0,03370 | 0,01930 | 0,98578 | 0,03883 |
| 0,0084 | 0,0042 | 0,03442 | 0,02025 | 0,99129 | 0,03993 |
| 0,0126 | 0,0042 | 0,03448 | 0,01982 | 0,98940 | 0,03977 |
| 0,0168 | 0,0042 | 0,03598 | 0,02118 | 0,98983 | 0,04176 |
| 0,021 | 0,0042 | 0,03304 | 0,02014 | 0,98888 | 0,03870 |
| 0,0252 | 0,0042 | 0,03444 | 0,02220 | 0,98873 | 0,04097 |
| 0,0294 | 0,0042 | 0,03455 | 0,02191 | 0,98959 | 0,04091 |
| 0,0336 | 0,0042 | 0,03310 | 0,02108 | 0,98608 | 0,03924 |
| 0,0378 | 0,0042 | 0,03473 | 0,02122 | 0,98616 | 0,04069 |
| 0,042 | 0,0042 | 0,03558 | 0,02214 | 0,98553 | 0,04190 |
| 0,0462 | 0,0042 | 0,03405 | 0,01972 | 0,98729 | 0,03935 |
| 0,0504 | 0,0042 | 0,03485 | 0,02136 | 0,98536 | 0,04088 |
| 0,0546 | 0,0042 | 0,03375 | 0,02055 | 0,98523 | 0,03951 |
| 0,0588 | 0,0042 | 0,03369 | 0,02051 | 0,98617 | 0,03944 |
| 0,063 | 0,0042 | 0,03369 | 0,02074 | 0,98835 | 0,03956 |
| 0,0672 | 0,0042 | 0,03513 | 0,02147 | 0,98600 | 0,04117 |
| 0,0714 | 0,0042 | 0,03515 | 0,02173 | 0,98864 | 0,04132 |
| 0,0756 | 0,0042 | 0,03441 | 0,02194 | 0,98450 | 0,04080 |

5.3.9. Normalização dos vetores X e Y.

No movimento pendular ou movimento harmónico simples ideal, o movimento desenvolve-se num único plano, nesse devem-se registar acelerações apenas em dois eixos, não devendo existir mais acelerações em planos transversais.

Mas não é o que acontece na realidade, devido as condições imprecisas de lançamento do pêndulo, existem rotações e translações, existindo sempre acelerações e movimento em todos eixos, devido ao movimento descontrolados

A figura 5.3 abaixo ilustra o problema.

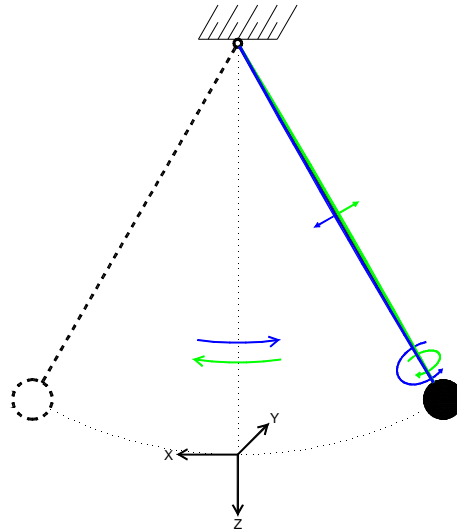


Figura 5.3 - Movimentos verificados um pêndulo livre não controlado

Esses movimentos foram verificados no pós-processamento dos dados recolhidos de cada ensaio.

Verificou-se nos dados obtidos pelo programa de pós-processamento, que nos primeiros minutos, existem vibrações que não dizem respeito ao movimento oscilatório, mas sim a rotações e a vibrações produzidas pelo elemento de ligação objeto-apoio.

Verificou-se que as vibrações do elemento de ligação objeto-apoio dissipam-se rapidamente, mas as rotações sobre este eixo mantém ao longo do tempo mesmo que impercetíveis.

Outra situação que se verificou foi o desalinhamento dos eixos do sensor plano do movimento. Esta situação não possível controlar, por falta de elemento estrutural que force o alinhamento do pendulo assim evitando que este saia do plano.

Devido ao problema do desalinhamento surgiu um problema na comparação de resultados, porque cada sensor tem os eixos X e Y tem os seus próprios alinhamentos em posições diferente, assim solução encontra foi a normalização destes vetores X e Y perpendiculares ao eixo vertical do elemento de ligação objeto-apoio. Desta forma continua-se a preservar a informação contida nestes dois eixos, e obtendo-se um resultado claro possibilitando uma melhor avaliação.

Na tabela 5.4 apresentada anteriormente pode-se verificar essa normalização, observando a coluna mais à direita.

5.3.10. Transformação de dados pelo método *Fast Fourier Transformation* (FFT)

O algoritmo *Fast Fourier Transformation*, permite converter uma amostra limitada de dados que se encontram no domínio do tempo para o domínio da frequência. Em engenharia esta técnica é usada transformar valores de registo de movimento de uma estrutura ou sistema, em frequências. Assim determinando características estruturais dessa estrutura. Ao avaliar as alterações das frequências a longo prazo, permitem determinar a evolução das características estruturais.

O algoritmo FFT é bastante complexo e utilizam-se programas próprios para executar essas transformações. O programa usado neste trabalho foi o SeismoSignal, que para além deste também disponibiliza outros métodos da transformação.

Algumas das especificações e particularidades relevantes, deste programa, para este trabalho são as seguintes:

- Este programa aceita apenas no máximo 32768 linhas, com uma resolução de ADC a 215. No caso de taxa de amostragem a 250Hz, este programa apenas aceita \pm 130 segundos de amostra.
- Este programa tem a possibilidade de fazer a regularização do periódico de amostragem, assim dispensado um dos passos anteriores. Mas em determinadas circunstâncias, quando o período é muito irregular, o programa dá erro. A solução preconizada na ajuda do *software* é o passo referido anteriormente. A opção no programa que permite executar este processo encontra-se na caixa de diálogo *Input Files Parameters*. A opção por definição é a *single acceleration value per line*, e a opção que executa este processo é *time acceleration values per line*. Com a opção *single acceleration value per line* é necessário fornecer o período na caixa *time step dt*.
- Permite fazer uma *Baseline Corretion*, ou correção da linha base. Caso o sensor estiver desnivelado, esse fato reflete-se nos resultados, dificultando a compreensão. Esta opção nivela esses dados facilitando o processo de pós-tratamento.
- Este programa após a transformação apresenta os resultados numa tabela de valores e também de uma forma gráfica. Os valores da tabela podem ser copiados e transportados para uma folha de Excel. Os dados apresentados nesta tabela são frequência, período, amplitude de Fourier, fase de Fourier e *Power Spectrum*.

Abaixo a apresenta a figura 5.4 que ilustra o programa Seismosignal com a caixa de diálogo *Input Files Parameters* aberta.

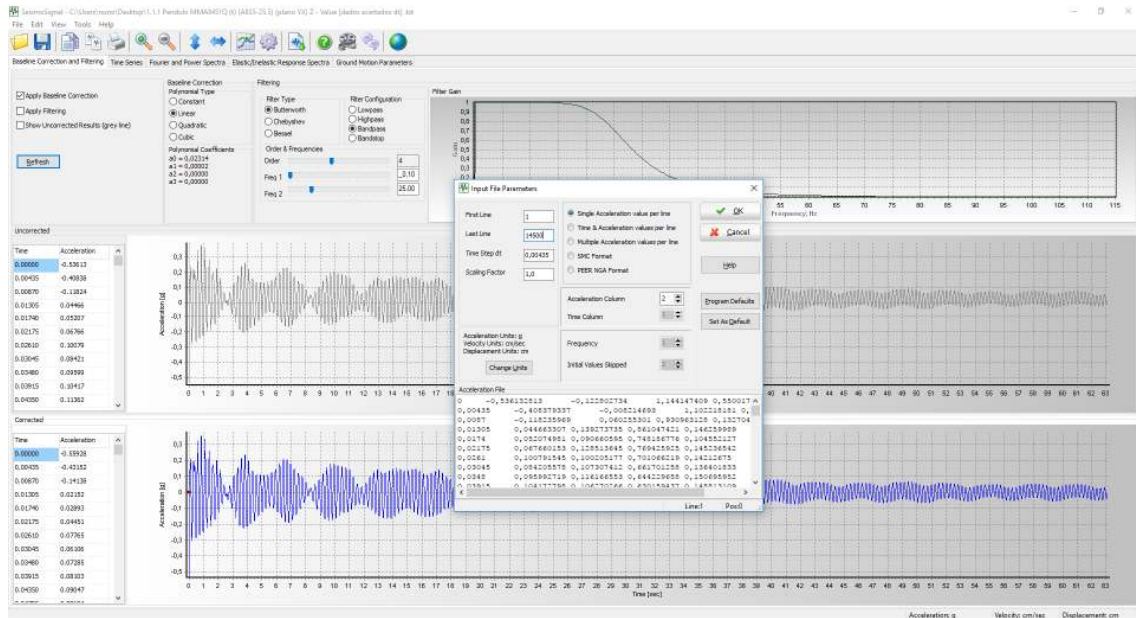


Figura 5.4 - Programa Seismosignal

Devido a limitação da quantidade de amostra aceites pelo programa Seismosignal, apenas foram usadas amostras de 60 segundos.

5.3.11. Execução de gráfico com histórico da amplitude de Fourier

Por forma a compreender a evolução da amplitude do movimento ao longo do tempo, do movimento pendular, desenvolveu-se um gráfico com base nos valores da amplitude Fourier obtidos pela FFT. Visto que o programa tem uma limitação em termos de quantidade de informação importada para este, apenas permitindo ± 90 segundos de amostra, reduziu-se o valor para 60 segundo, evitando erros de sobrecarga de dados.

Inicialmente pensou-se fazer uma avaliação em períodos de 60 em 60 segundos ao longo dos 36 minutos de cada ensaio, o representava uma sobrecarga em termo de tempo de pós-tratamento muito grande, decidiu-se aumentar de avaliação de amostra para intervalos de 4 minutos, assim são necessários gerar seis pós-processamento FFT por cada ensaio, necessários para gerar um gráfico com histórico da evolução da amplitude de Fourier.

Após regularização de dados, foram introduzidos no programa Seismosignal, e os resultados produzidos transformação pela FFT foram copiados para a Folha de Excel.

Os dados usados para produzir os gráficos foram apenas a amplitude de Fourier de uma frequência. A frequência usada foi a mais alta registada durante o ensaio, que representa a frequência natural do sistema pendular.

5.3.12. Apresentação de resultados

Como pré-definido no procedimento de ensaio, a cada ensaio o pêndulo foi afastado do eixo de equilíbrio vertical a uma distância definida e libertado, podendo oscilar livremente durante 36 minutos. Ao fim desse tempo, verificou-se que no ensaio 1, ainda se verificava movimento, aproximadamente com 2,5cm. No ensaio 2 o pêndulo ao fim de 36 minutos encontrava-se aparentemente imóvel. Nesta situação foi bem evidente nos valores determinados da amplitude de Fourier entre os minutos 35-36 em todos os gráficos do ensaio 2.

São apresentados de seguida os resultados dos dados processados, em cada ensaio, em tabelas e gráficos, para cada um por cada um dos eixos dos sensores, e um plano XY.

Ensaio 1 – frequência calculada 0,338Hz

Tabela 5.5 - Tabela de variação de amplitude Fourier ao longo do ensaio - Ensaio 1

| Sensor Inercial | Frequência | Eixo | Amplitude de Fourier | | | | | | | |
|-----------------|------------|------|--|--------|--------|---------|---------|---------|---------|---------|
| | | | Intervalos de tempo [m] utilizados para determinar as amplitudes de Fourier através de FFT | | | | | | | |
| | | | 0 - 1 | 5 - 6 | 10 -11 | 15 - 16 | 20 - 21 | 25 - 26 | 30 - 31 | 35 - 36 |
| MMA8451Q | 0,337 | X | 0,2682 | 0,0815 | 0,0324 | 0,0158 | 0,0052 | 0,0036 | 0,0044 | 0,0035 |
| | | Y | 0,2766 | 0,1148 | 0,0698 | 0,0491 | 0,0390 | 0,0316 | 0,0252 | 0,0222 |
| | 0,673 | Z | 2,1377 | 0,1093 | 0,0308 | 0,0146 | 0,0072 | 0,0053 | 0,0035 | 0,0030 |
| | 0,337 | XY | 0,1990 | 0,1373 | 0,0769 | 0,0514 | 0,0380 | 0,0304 | 0,0247 | 0,0217 |
| MPU6050 | 0,332 | X | 0,2893 | 0,0719 | 0,0366 | 0,0271 | 0,0216 | 0,0172 | 0,0171 | 0,0167 |
| | | Y | 0,3173 | 0,0765 | 0,0459 | 0,0295 | 0,0227 | 0,0125 | 0,0094 | 0,0051 |
| | 0,678 | Z | 1,8186 | 0,1293 | 0,0386 | 0,0187 | 0,0117 | 0,0048 | 0,0051 | 0,0051 |
| | 0,332 | XY | 0,1615 | 0,0890 | 0,0469 | 0,0336 | 0,0266 | 0,0197 | 0,0188 | 0,0174 |
| LIS331HH | 0,342 | X | 0,0547 | 0,0154 | 0,0133 | 0,0119 | 0,0099 | 0,0178 | 0,0063 | 0,0033 |
| | | Y | 0,3952 | 0,1363 | 0,0638 | 0,0510 | 0,0354 | 0,0280 | 0,0319 | 0,0110 |
| | 0,670 | Z | 2,0869 | 0,1070 | 0,0278 | 0,0207 | 0,0051 | 0,0095 | 0,0107 | 0,0075 |
| | 0,342 | XY | 0,0442 | 0,0101 | 0,0041 | 0,0077 | 0,0020 | 0,0021 | 0,0057 | 0,0033 |

Gráfico 5.1 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaio 1 – Eixo X

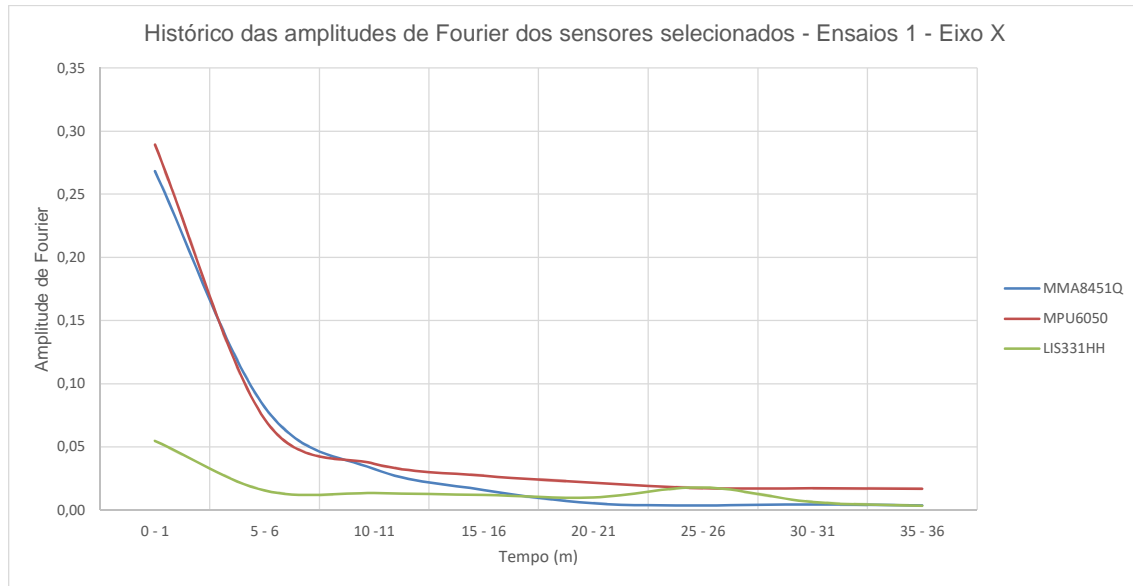


Gráfico 5.2 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaio 1 – Eixo Y

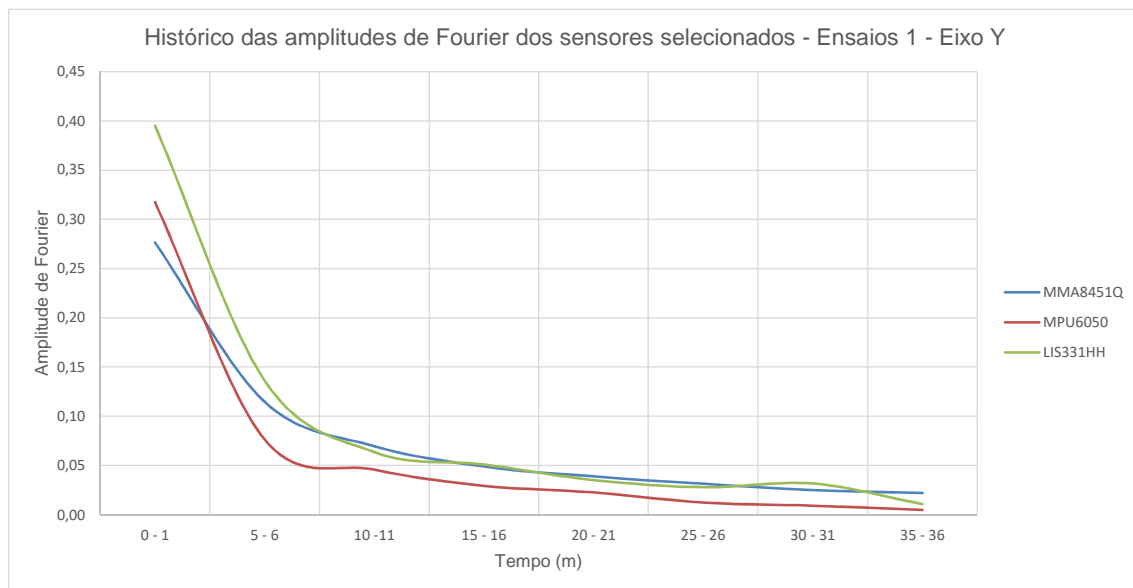


Gráfico 5.3 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaios 1 – Eixo Z

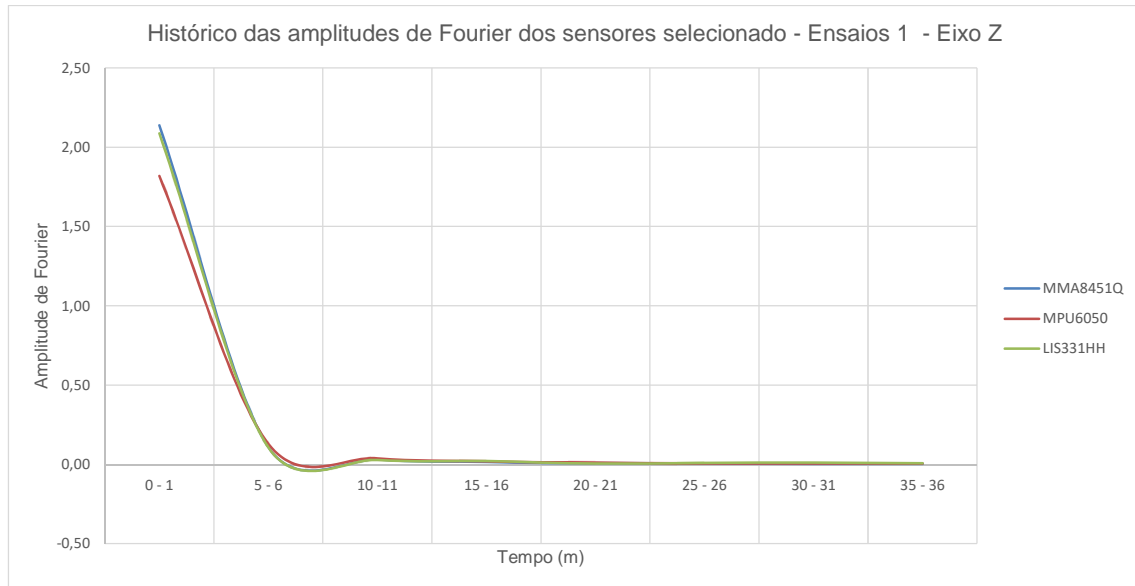
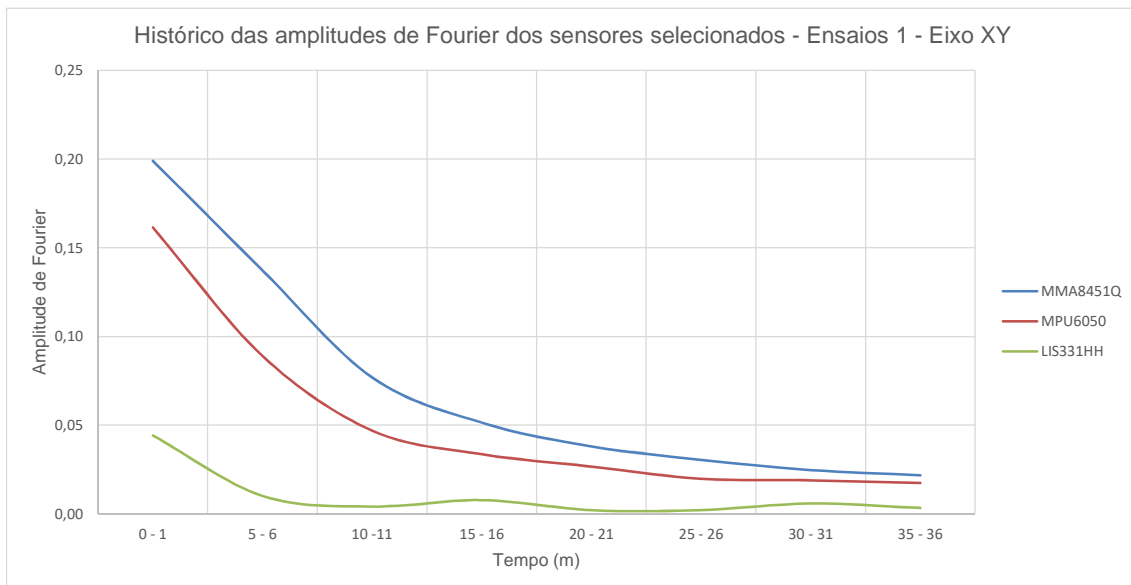


Gráfico 5.4 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaios 1 – Eixo XY



Ensaio 2 – frequência calculada 1,143Hz

Tabela 5.6 - Tabela de variação de amplitude Fourier ao longo do ensaio - Ensaio 2

| Sensor Inercial | Frequência | Eixo | Amplitude de Fourier | | | | | | | |
|-----------------|------------|------|---|--------|--------|---------|---------|---------|---------|---------|
| | | | Intervalos discretos utilizados para determinar as amplitudes de Fourier através de FFT | | | | | | | |
| | | | 0 - 1 | 5 - 6 | 10 -11 | 15 - 16 | 20 - 21 | 25 - 26 | 30 - 31 | 35 - 36 |
| MMA8451Q | 1,157 | X | 0,3417 | 0,0590 | 0,0173 | 0,0097 | 0,0032 | 0,0040 | 0,0014 | 0,0010 |
| | | Y | 2,1922 | 0,5841 | 0,3615 | 0,2042 | 0,1161 | 0,0646 | 0,0353 | 0,0192 |
| | | Z | 0,3974 | 0,1020 | 0,0629 | 0,0370 | 0,0202 | 0,0109 | 0,0054 | 0,0040 |
| | | XY | 2,0179 | 0,5776 | 0,3590 | 0,2028 | 0,1155 | 0,0641 | 0,0352 | 0,0190 |
| MPU6050 | 1,175 | X | 0,1786 | 0,0525 | 0,0252 | 0,0115 | 0,0052 | 0,0040 | 0,0056 | 0,0026 |
| | | Y | 2,1719 | 0,3526 | 0,1428 | 0,0645 | 0,0315 | 0,0173 | 0,0096 | 0,0065 |
| | | Z | 0,0505 | 0,0078 | 0,0038 | 0,0008 | 0,0025 | 0,0034 | 0,0006 | 0,0028 |
| | | XY | 0,4495 | 0,1362 | 0,0384 | 0,0033 | 0,0017 | 0,0026 | 0,0034 | 0,0012 |
| LIS331HH | 1,176 | X | 0,5150 | 0,2216 | 0,1179 | 0,0440 | 0,0231 | 0,0173 | 0,0113 | 0,0055 |
| | | Y | 2,2550 | 0,6527 | 0,2111 | 0,0834 | 0,0420 | 0,0226 | 0,0185 | 0,0062 |
| | | Z | 0,1101 | 0,0495 | 0,0223 | 0,0052 | 0,0083 | 0,0009 | 0,0048 | 0,0046 |
| | | XY | 0,5795 | 0,3560 | 0,1631 | 0,0620 | 0,0320 | 0,0215 | 0,0150 | 0,0065 |

Gráfico 5.5 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaio 2 – Eixo X

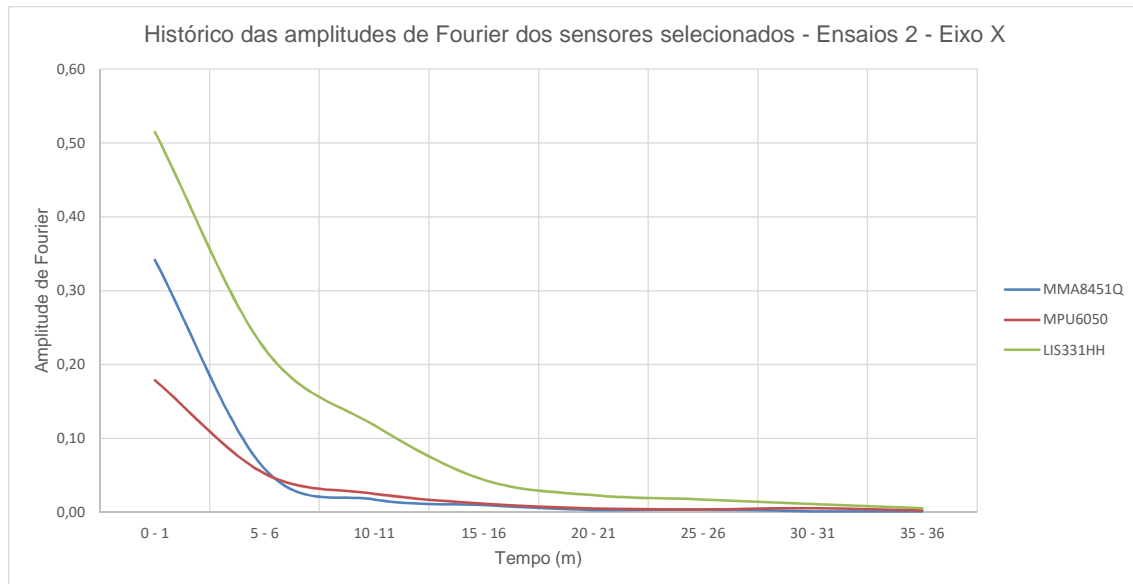


Gráfico 5.6 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaios 2 – Eixo Y

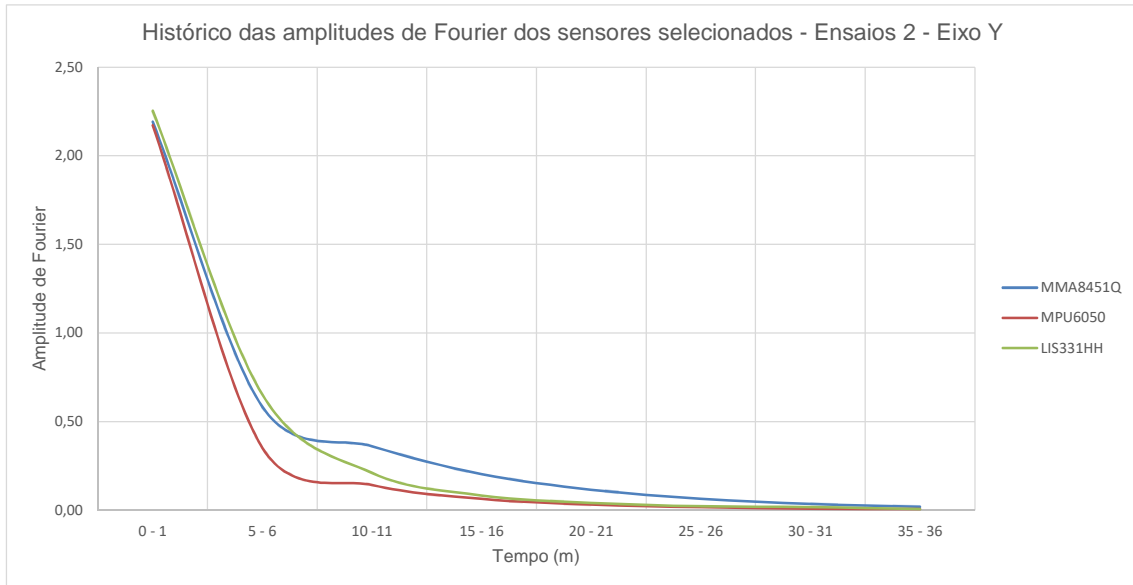


Gráfico 5.7 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaios 2 – Eixo Z

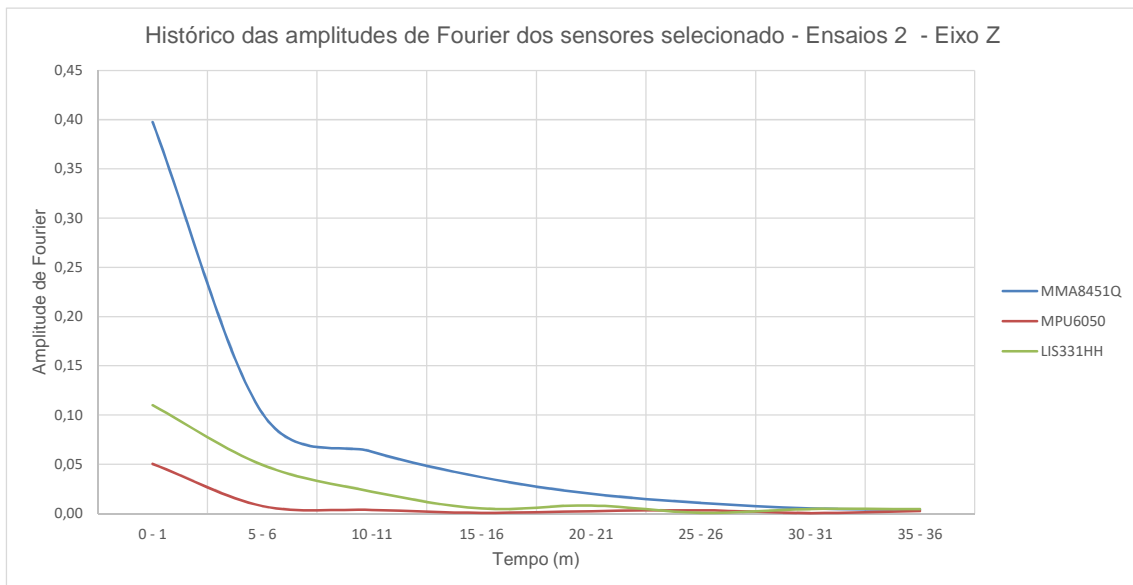
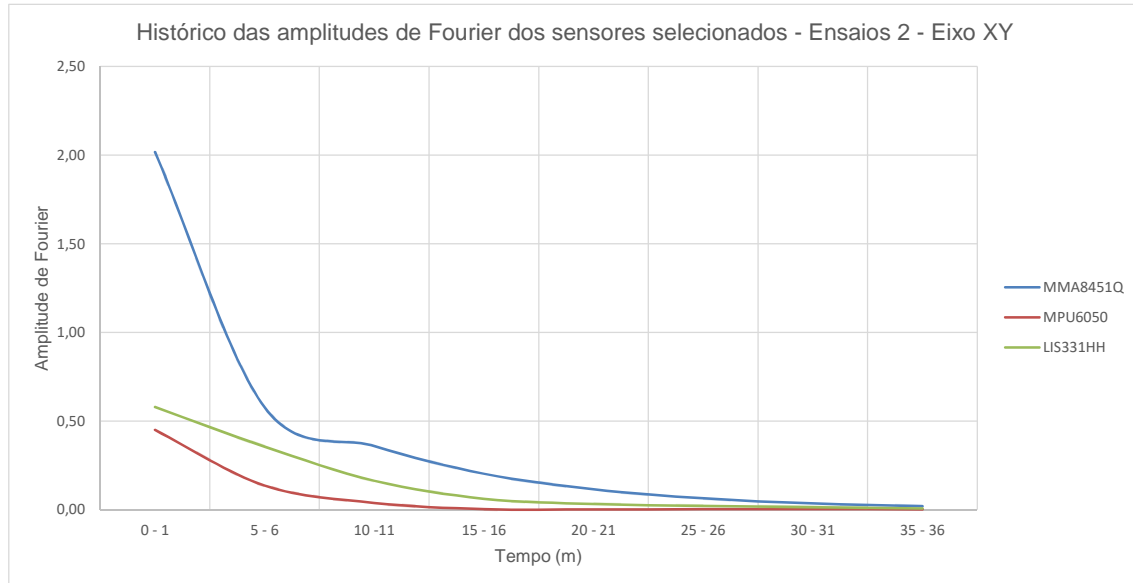


Gráfico 5.8 - Histórico das amplitudes de Fourier dos sensores MEMS - Ensaio 2 – Eixo XY

5.3.13. Conclusões sobre os resultados do ensaio dinâmico

Ensaio 1

Durante o ensaio 1 fez-se uma contagem visual do número de ciclos oscilatórios num determinado período de tempo e calculou-se a frequência deste sistema que foi 0,338Hz, igual ao valor calculado pelas equações do pêndulo.

Após o tratamento de dados pelo *software* Seimosignal, identificou-se os valores das frequências com maiores amplitudes de Fourier. Esses valores são os indicados nas tabelas acima.

Verificou-se que são muito similares aos valores determinados por cálculo e pelo processo de contagem visual, que oferecem uma boa confiança da informação produzida.

Incorreta interpretação da frequência do movimento pendular

Na tabela de dados produzida pelo programa Seimosignal, verifica-se apenas no eixo Z (eixo vertical), que a frequência tem a maior amplitude de Fourier, neste caso o dobro do valor real, situação verificada em todos sensores testados.

A razão para este fenómeno, deveu-se à simetria do movimento oscilatório no eixo vertical, e aos valores dos sinais de aceleração registados, terem sempre o mesmo sinal. Os valores de aceleração no eixo z são sempre positivos nunca passando pelo valor neutro e invertendo o sinal.

A figura abaixo representa uma função hipotética do movimento pendular, num gráfico cartesiano, os pontos assinalados a verde indicam a aceleração registada na passagem pelo ponto de equilíbrio. Os pontos azuis e vermelhos indicam a aceleração em cada extremo do movimento pendular. É indicado a quantidade de tempo do período da função, também é indicada o período do movimento pendular de um ciclo completo que é o dobro da função.

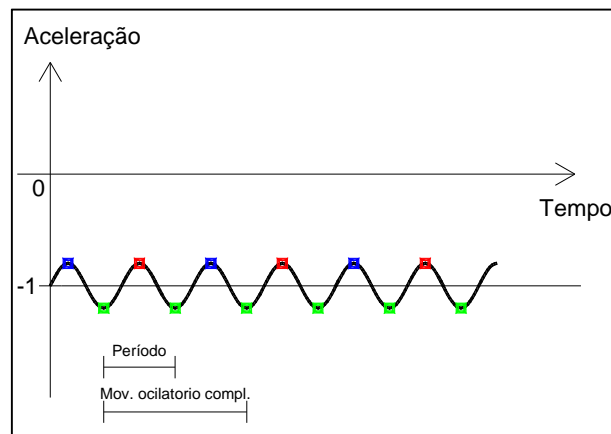


Figura 5.5 - Representação do movimento pendular em tempo/aceleração do eixo Z

Como o resultado da simetria do sistema e os valores de aceleração serem sempre com o mesmo sinal, o programa Seismosignal determinou que a frequência pela variação da aceleração, e não pela inversão do sinal, assim duplicando o valor da frequência.

No ensaio 2 este problema já não se verificou, obtendo os mesmos valores de frequências com amplitudes máximas em todos os eixos e sensores. Presume-se, que deve ao facto do sistema ter menor energia mecânica, e uma maior desaceleração e variação angular comparativamente com o sistema 1, traduzindo em valores de aceleração com características mais pronunciadas, que contribuem para uma melhor interpretação deste tipo de movimento e sistema pelo *software* Seismosignal.

Desalinhamentos dos eixos

Devido ao tipo de sistema usado nestes ensaios, que não permite uma estabilização da rotação sobre o eixo vertical, é notório o desalinhamento dos eixos dos sensores com o plano de movimento pendular.

Nos gráficos dos eixos X e Y do ensaio 1, verificou-se este fenómeno. Os sensores MMA8451Q e MPU6050, apresentam valores relativamente simétricos em ambos os eixos, que indica um desalinhamento. No entanto o sensor LIS331HH isso já não aconteceu, os valores apresentados no eixo Y, são significativamente mais elevados que o eixo X, indicado que este eixo está mais alinhado com o plano do movimento pendular.

Ao avaliar os três sensores apenas por observação dos gráficos X ou Y, não obtemos conclusões comparativas corretas, pois existem sensores que apresentam bons resultados num eixo e relativamente pobres no eixo oposto. Assim para facilitar este processo, normalizou-se os valores destes dois eixos, apresentando-os num gráfico do plano XY de valores normalizados, que produz melhores resultados de comparação

Ensaio 2

Após o pós-processamento dos dados do ensaio 2, com o *software* FFT, determinou-se as frequências com os valores maiores valores de amplitude Fourier, e em todos os sensores ensaiados, os valores das frequências forma superiores a frequência determinada pela equação do pendulo. Duas situações que podem justificar esta discrepância, a primeira é a própria equação do movimento pendular, a equação apresentada é válida apenas para pequenas amplitudes. Neste ensaio devido ao comprimento do pêndulo ser relativamente pequeno, o movimento oscilatório verificado no sistema 2, teve relativamente uma grande amplitude de movimento.

A segunda situação deveu-se à imprecisão da medição do comprimento do pêndulo. Contribuindo para uma determinação menos precisa da frequência quando usada equação da frequência do movimento pendular, resultando na diferença observada entre o valor calculado e valor verificado após o tratamento dos dados.

Verificou-se também na tabela de resultados, que a frequência do sensor MMA8451Q é diferente dos outros dois sensores.

Esta diferença deveu-se a uma pequena alteração no sistema pendular, durante o ensaio com o sensor MMA8451Q, nomeadamente o posicionamento do sensor dentro da embalagem e uma pequena variação verificada no comprimento do pendulo. Com a diferença verificada entre este sensor e os restantes não é muito significativa aceitou os resultados obtidos

Ao analisar os gráficos obtidos pela normalização dos valores dos eixos X e Y e os gráficos do eixo Z, há um sensor que sobressai acima dos restantes, que é o MMA8451Q, indicando que uma a melhor performance em termos de sensibilidade ao movimento, apesar da menor resolução de ADC que os sensores MPU6050 e LIS331HH.

5.4. Conclusão das avaliações dos sensores MEMS

Os resultados da avaliação estática indicam que o sensor MMA8451Q tem a melhor performance, nomeadamente o menor variação de resultados indicando que produz a menor quantidade de ruído dos três sensores avaliados.

No ensaio dinâmico, que teve como objetivo identificar o sensor com maior sensibilidade ao movimento, novamente o sensor MMA8451Q é indicado com a melhor opção para um dispositivo de monitorização de vibrações.

6. Protótipo de aquisição de vibrações e monitorização de vibrações

6.1. Introdução

Uma vez concebido, testado e validado o dispositivo 1 para medição e registo de vibrações, de acordo com pré-requisitos e especificações, prosseguiu-se para a fase seguinte. Esta consistiu em dotar o dispositivo de capacidade de monitorização de vibrações, através da análise dos registos obtidos em tempo real.

A capacidade de monitorizar as vibrações e tomar decisões com base na análise de dados, automaticamente em tempo real, é realizada por *trigger action algorithm* ou algoritmo de decisão automática. Assim para concretizar este objetivo foi necessário desenvolver um algoritmo que cumprisse o objetivo, e injetando no código no *sketch* desenvolvido na fase anterior.

Com a introdução de mais código, as capacidades do dispositivo 1 foram afetadas, particularmente a taxa de amostragem, que baixou significativamente entrando em incumprimento das especificações funcionais. O algoritmo obriga a uma maior quantidade de processamento e memória a cada ciclo, assim foi necessário redimensionar o *hardware* para acomodar estas alterações, tendo como suporte o desenvolvimento e conhecimentos adquiridos durante o desenvolvimento do Dispositivo 1

A capacidade de monitorizar vibrações, e o desenvolvimento algoritmo foi um assunto não previsto na fase de pesquisa literária, foi necessário pesquisar informações adicionais sobre a concepção de um algoritmo de análise e decisão automática por regras. Esse algoritmo ou processo foi depois programado em linguagem usada pelo IDE da Arduino e integrado no *sketch* ou programa já desenvolvido.

6.2. Especificações do sistema

Para satisfazer o novo objetivo foram introduzidas novas especificações as já referidas no ponto 4.1.2. Estas focaram-se no aumento das capacidade do *hardware*, para acomodar a necessidade extra de capacidade de processamento e memória, essenciais para o correr algoritmo e manter a taxa de amostragem dentro dos parâmetros. Com a introdução de um algoritmo foi necessário rever algumas especificações ao *software*.

Especificações do *Software* / programa

- O programa deve ter a capacidade para detetar variações nos valores obtidos do sensor, em tempo real, em intervalos periódicos de tempo.
- Deve automaticamente tomar a decisão de guardar ou descartar registos de dados, com base em regras e/ou condições.
- Deve guardar pelo menos 60 segundos ou mais de registos, aquando o acionamento de uma regra, numa unidade de armazenamento à semelhança das especificações descritas no ponto 4.1.2.
- Deve registar o tempo e data em cada um dos ficheiros criados.

Especificações do *Hardware*

- O algoritmo requiere uma capacidade adicional de processamento e memória, para analisar dados em tempo real, assim é necessário um microprocessador com maior velocidade de processamento e memória, e que ao mesmo tempo seja compatível tecnologia Arduino e o *software* implementado no dispositivo 1.
- O dispositivo 2 é para ser aplicado em locais não vigiados, assim sujeito a todo o tipo de agressões, especialmente do meio ambiente, assim deve possuir uma caixa que albergue todos os componentes, e os proteja de poeiras, agressões físicas, queda de partículas de água (chuva), e humidade.
- Para poder registar o tempo e data em cada um dos ficheiros, é necessário incorporar um módulo com relógio e calendário.
- O dispositivo 2 deve obter energia de fontes de alimentação externa, sobre a forma de corrente elétrica alternada 220V, também deve possuir uma fonte interna de backup.

6.3. Aquisição peças, procedimento de montagem do protótipo 2

6.3.1. Placa de desenvolvimento – Microcontrolador

Para compensar o crescimento de processamento e necessidade de memória volátil adicional para armazenar a dados necessários ao algoritmo, foi necessário encontrar uma nova placa de desenvolvimento para o novo dispositivo 2.

Fez-se um levantamento nos *sites* de fabricantes de placas de desenvolvimento, compatíveis com tecnologia Arduino, e comparou-se os produtos encontrados pelas seguintes características: capacidade de processamento; voltagem; memória disponível EEPROM e SDRAM e Flash, e placas de expansão ou *shields* e opções extras.

Foi encontrada uma que reuniu todas as características solicitadas, com um preço bastante apelativo e disponível no mercado *online*, do fabricante Adafruit, modelo Adafruit Feather M0 Basic Proto - ATSAMD21 Cortex M0.

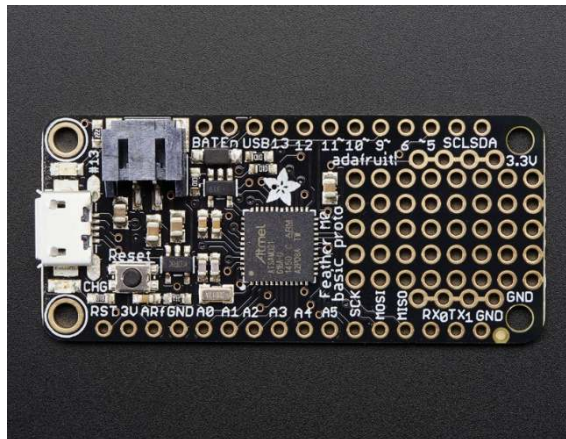


Figura 6.1 - Placa de desenvolvimento Adafruit Feather M0 Basic Proto [30]

Por comparação direta destas características da Arduino UNO R3 com esta placa, verificou-se que esta possui especificações superiores à da Arduino UNO R3, oferecendo ainda novas características e possibilidade.

O resumo das características da placa de desenvolvimento Adafruit, modelo Feather M0 Basic Proto, encontram-se na tabela 6.1.

Tabela 6.1 - Especificações da placa Adafruit, Feather M0 Basic, Proto Cortex M0

| Fabricante Modelo | processador | | | Voltagem | | Velocidade | Ligações - Conceções | | | | Memória | | Portas | | |
|---|-------------|-------------|----------------------|-----------------|----------------|--------------|----------------------|-------|----------|-----|----------------|---------------|---------------|------------|---|
| | Fabricante | Modelo | arquitetura [bit] | Operação [V] | Entrada [V] | CPU [MHz] | Analogicas | | Digitais | | EEPROM [kB] | SDRAM [kB] | Flash [kB] | USB | |
| | | | | | | | Entrada | Saída | IO | PWR | | | | UART | |
| Adafruit - Feather M0 Basic Proto | ARM | ATSAMD21G18 | 32 | 3,3 | 3,3-5 | 48 | 6 | 1 | 13 | 7 | | 32 | 256 | Tipo B (F) | 6 |

Existem outras especificações, não mencionadas na tabela 6.1, que contribuirão também para a escolha desta placa em detrimento de outras, que são apresentadas de seguida:

- A placa Adafruit, Feather M0 Basic, é 3 vezes mais pequena que a Arduino UNO R3.
- Esta placa é 1,33 vezes mais barata que a Arduino UNO R3.
- Possui uma memória *flash* de 256 Kb, que permite o armazenamento 46 segundos de dados, considerando uma taxa de amostragem 250Hz, registo de três eixos e registo de tempo em nano segundos.
- Possui 4 orifícios com 2mm de diâmetro para montagem numa estrutura de apoio.
- Possui um regulador de voltagem a 3,3V que permite um consumo energético reduzido, em relação às placas a 5V com um máximo de intensidade de corrente de 0,5A.
- Possui 6 canais de comunicação Serial internos, invés de um só da Arduino UNO, assim é possível ter o protocolo Serial instalado e a enviar informação, situação que não é possível com a Arduino UNO.
- Permite ligar uma bateria de polímeros de lítio de 3,7V ou 4,2V.
- Quando uma bateria de lítio é ligada a placa, esta automaticamente faz a gestão de energia, alterando entre fonte externa e interna.
- Possui uma área de prototipagem de 40 *pin's*, com as dimensões iguais ao sensor NPX MMA8451Q, sendo possível aí instalar o sensor.

6.3.2. Leitor de cartões micro SD e Relógio digital com calendário

À semelhança da placa Arduino UNO R3, a placa Adafruit, Feather M0 Basic tem um relógio interno ou *real time clock* RTC, capaz de medir o tempo a curto prazo mas não possui calendário, não sendo possível registar data e horas.

Para compensar esta lacuna pesquisou-se por este módulo no *site* do fabricante da placa de desenvolvimento e encontrou-se uma placa de expansão ou *shield*, com este módulo e em simultâneo um módulo e interface para cartões micro USB. O produto designa-se por *Adalogger FeatherWing - RTC + SD Add-on For All Feather Boards*, que resolve duas questões no mesmo produto. O funcionamento do módulo micro SD, é igual ao módulo da DFRobot, não havendo qualquer alteração de programação ou novas bibliotecas.

Esta solução reúne duas funções num único componente, que se torna monetariamente mais económica, em relação a aquisição dos componentes separados. Também não há perdas de tempo com ligações complicadas com *jumpers* à placa de desenvolvimento, ou material extra de fixação à caixa externa, pois esta empilha-se diretamente sobre a placa de desenvolvimento. Esta solução não retira acesso as funções dos *pin's* da placa de desenvolvimento.

O relógio e calendário encontrado no *shield*, é NPX modelo PCF8523, este permite uma resolução de segundos, minutos, horas, dias, meses e anos. Este RTC necessita de uma bateria CR1220 de 3V, para fornecer energia interna de *backup* assim mantendo o registo do tempo em caso de falha de energia externa.

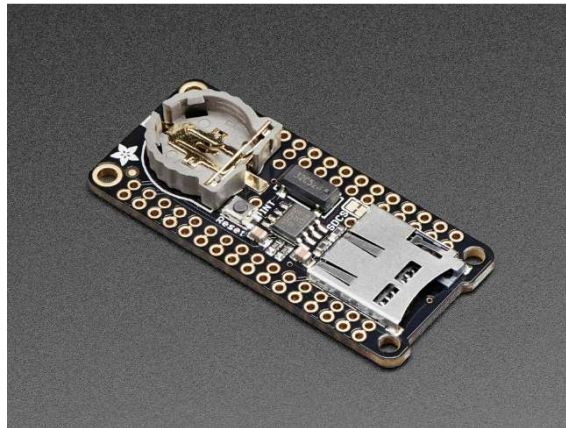


Figura 6.2 - Adafruit Adalogger FeatherWing [30]

6.3.3. Fonte de alimentação

Este dispositivo terá com que operar continuamente num mesmo local, durante grandes períodos de tempo, sem interrupções e com manutenções periódicas bastante espaçadas, por isso foi necessário conceber um sistema de fornecimento de energia que garantisse essa continuidade sem interrupções.

O fornecimento de energia ao Dispositivo 2 foi concebido para operar com duas fontes elétricas uma fonte externa e uma interna. A fonte externa fornece energia interruptamente, e uma fonte de energia interna de curta duração que compensa as falha ou interrupções da energia externa, servido assim de *backup*.

A energia externa deve ter origem de uma tomada elétrica de baixa tensão, no local de monitorização. Essa energia externa deve ser de corrente alternada (AC) de 220V, que é convertida por um transformador de energia elétrica de AC em corrente contínua (DC) a 5V e pelo menos 0,5 Amperes. O sistema elétrico é composto então por uma ficha elétrica à qual se liga ao transformador que por sua vez liga à placa de desenvolvimento por meio de um cabo USB.

O sistema de gestão da energia é feito pela própria placa de desenvolvimento, que possui *software* próprio de fábrica. Este sistema gere todo o processo automaticamente, dando prioridade à fonte de energia externa, sobre a interna, quando essa existe. Também tem a capacidade de recarregar a bateria de polímeros de lítio automaticamente, sem programação adicional, sendo por isso considerada *plug and play*.

Com referido a placa de desenvolvimento permite a ligação a uma bateria de polímeros de lítio, de 3,7V ou 4,2V. O fabricante da placa não dá informações sobre os limites máximos e mínimos da capacidade de armazenamento da bateria, assim optou por uma bateria de polímeros de lítio 3,7V com 500mAh, que permite que o dispositivo opere por várias horas, desligado de uma fonte externa.

Quando ligada a bateria à placa, esta começa imediatamente a correr o programa instalado, pois a placa não tem um botão de on/off, que pode determinadas condições uma inconveniência. A ligação da bateria à placa é feita por um ficha ou *jack* do tipo JST 2 com 2mm de largura. Devido às pequenas dimensões da ficha, a operação de conectar e desconectar a ficha é muito dificultada. Para ultrapassar este problema, a solução foi montar um interruptor do tipo *click clack*, no fio positivo, entre a ficha JST e a bateria, assim instalando um botão *on/off*.

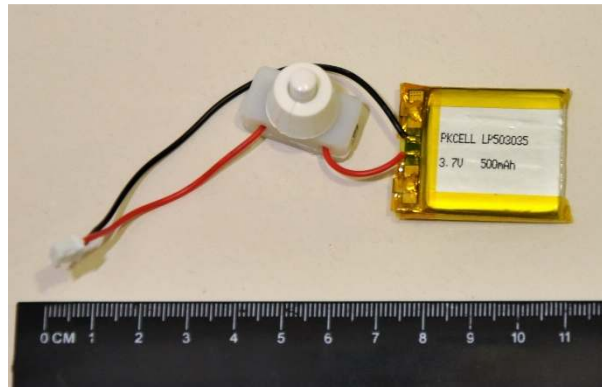


Figura 6.3 - Bateria de lítio 3,7V 500mAh e botão *click-clak* instalado

6.3.4. Caixa de Proteção

Sensor de temperatura

Por forma monitorizar a temperatura interna, dentro da caixa externa e temperatura externa do ambiente, foram aplicados dois termómetros digitais. Foram encontrados vários sensores de temperatura, optando-se pelo fabricante Maxim Integrated, modelo Dallas DS18B20, com uma resolução de 0,06 °C, de baixo custo. Este tipo de termómetro opera com um protocolo diferente dos referidos, este protocolo é o One Wire, e permite a comunicação entre a placa controladora e o sensor usando apenas um fio.

O sensor opera com um tensão de correntes entre os 3,3V e os 5V, no entanto o canal comunicação necessita de ser ligado polo positivo da placa de 3,3V, por intermédio de uma ligação com redução de tensão, feita por uma resistência de 4,7k Ohm, como ilustrado na figura 4. Por fim liga-se um fio eléctrico de comunicação, entre uma entrada digital da placa de desenvolvimento e o fio a jusante da resistência.

O termómetro interno foi fixo num pequeno circuito criado para o efeito dentro da caixa, o termómetro externo ligado ao pequeno circuito por meio de um cabo eléctrico com 1 metro de comprimento.

6.3.5. Caixa de Proteção

Para instalar e proteger todos os componentes do dispositivo 2, reunindo-os e fixando-os num única estrutura, optou-se por uma caixa proteção. Dentro das ofertas disponíveis no

mercado *online* encontrou-se uma caixa externa do fornecedor Sparkfun [25] com volume necessário para albergar todos os componentes.

A caixa é em ABS, composta por duas partes, que se encaixam numa ligação macho/fêmea, isoladas por um vedante de borracha contínuo, que previne a entrada de água, poeira ou humidade para dentro da caixa.

As dimensões internas da caixa são 146x84x69 mm. Para evitar a abertura acidental esta possui 8 parafusos de fixação na tampa. O fundo da caixa tem 10 furações com 2mm de diâmetro para fixação de equipamento. A base externa da caixa tem duas abas com furações para permitir a fixação a elemento externo.



Figura 6.4 - Caixa ABS

As furações da caixa externa, não coincidem com as furações da placa de desenvolvimento, foi necessário projetar e fabricar uma placa de transição entre estes dois elementos. A placa desenvolvida foi em acrílico com 5mm de espessura, com a forma interna da caixa.

Na mesma placa de acrílico foram executadas 3 grupos de 4 furações de 2mm para fixação da placa de desenvolvimento. A configuração de um grupo de 4 furações, é idêntica às furações da placa Adafruit, Feather M0 Basic.

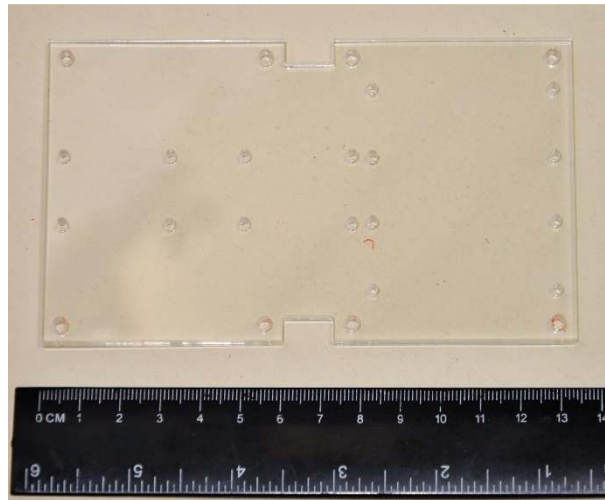


Figura 6.5 - Placa em acrílico para fixação de equipamento

Foi necessário abrir dois negativos na parede lateral da caixa com um diâmetro 12mm, para passar o cabo elétrico que fornece energia aos transformador e um cabo elétrico para ligar um sensor de temperatura à placa de desenvolvimento. A abertura dos negativos para a passagem de cabos, comprometem a integridade da resistência a ataques ambientais externos. Para garantir a integridade na passagem de cabos foram usados bucins com vedantes.

Usaram-se dois conetores ou bucins do tipo Gland, modelo PG7, que permitem o isolamento perfeito entre a superfície da caixa e os cabos.



Figura 6.6 - Bucins desmontado

6.3.6. Sensor de temperatura

Para registar a temperatura ambiental encontrou-se um sensor de temperatura do Maxim modelo DS18B20. Um sensor com uma resolução até 12bit excelente para este tipo de uso, necessita só de fio para transmitir e receber dados, e com custo unitário muito reduzido.

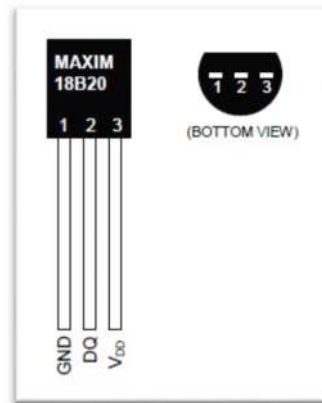


Figura 6.7 - Sensor DS18B20

6.3.7. Diagrama de ligações e montagem do Dispositivo 2

Para ilustrar as ligações entre os vários componentes do Dispositivo 2, a figura 45 mostra o diagrama de ligações. No Anexo A2 apresenta-se um procedimento de montagem do dispositivo 2.

Nesta ilustração a placa de expansão de capacidades ou *shield*, Adafruit Adalogger RTC + SD ADD-on, não é mostrado pois empilhou-se sobre a placa de desenvolvimento Adafruit, Feather M0 Basic, não ficando visível, todas as ligações pin's são diretas entre ambas as placas.

Também nesta ilustração o sensor MMA8451Q da NXP não aparece na sua posição final, que é sobre a área inferior de prototipagem, assim esta ilustração mostra com as ligações executadas de uma forma mais perceptível.

Todos elementos apresentados têm a mesma relação em termos de escala com exceção do interruptor *click clack* e os sensores de temperatura.

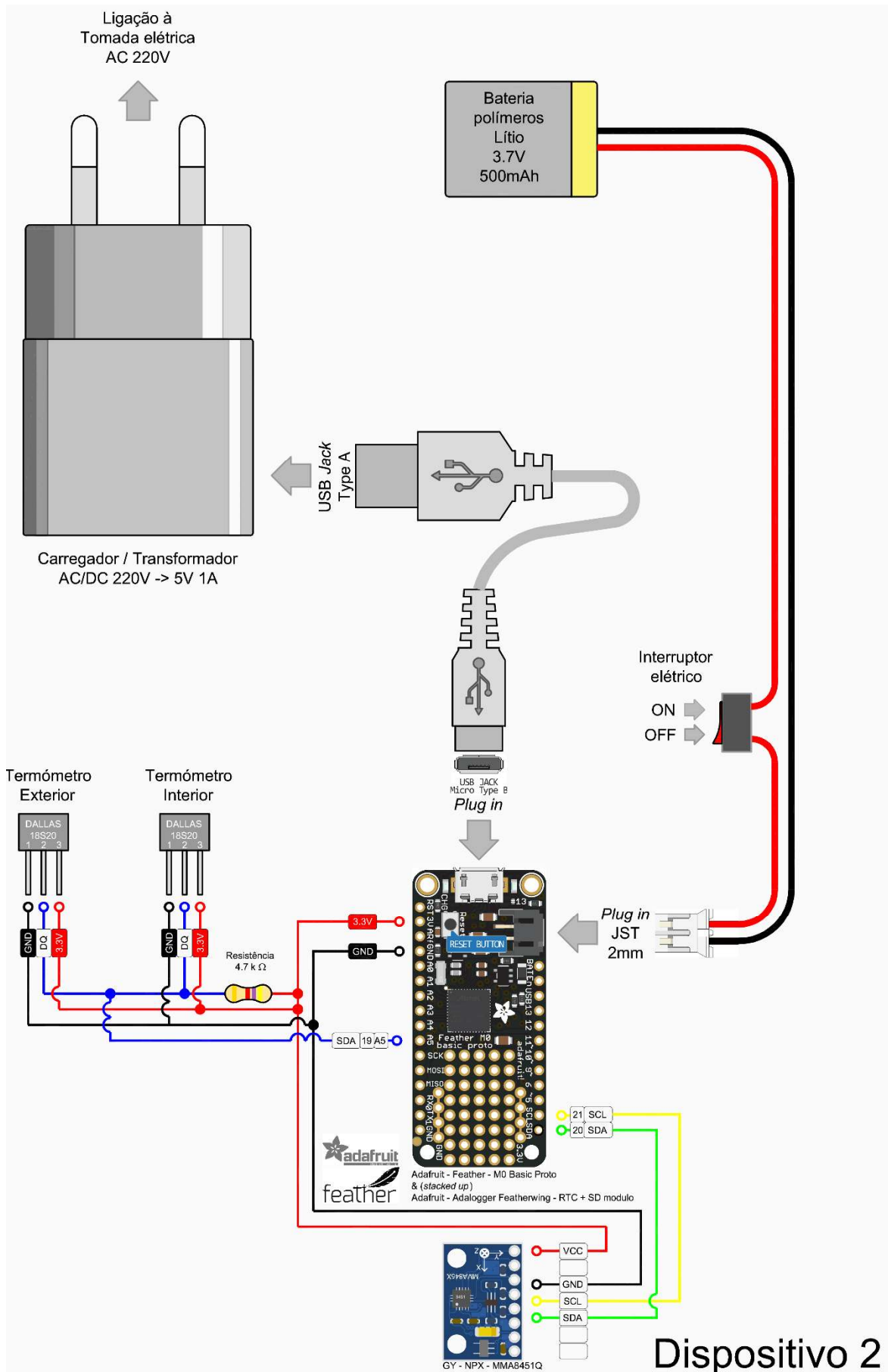


Figura 6.8 - Diagrama de ligações e montagem do Dispositivo 2

6.4. Desenvolvimento de Sketch (Programa)

6.4.1. Introdução

O desenvolvimento do *sketch* do dispositivo 2 tem como ponto de partida o *sketch* do dispositivo 1. Com o *upgrade* de alguns componentes não é necessário de alterar o código, mas é eventualmente necessário instalar algumas bibliotecas. Com introdução de novos componentes obrigatoriamente é necessário adicionar código e bibliotecas para opera-los.

O *trigger action algorithm* é o elemento crucial do sistema de monitorização, o seu desenvolvimento mereceu de pesquisa complementar, sobre assunto tendo atenção ao tipo sistema desenvolvido e testado até ao presente capítulo. O algoritmo deve ser nível de sofisticação ajustado ao tipo de processador, sensor, memória, e demais capacidades disponíveis.

6.4.2. Algoritmo de *Trigger action*

Um algoritmo é um processo ou fórmula para resolver um problema matemático, composto por um número finito de instruções lógicas numa sequência específica.

Um *Trigger Action* numa tradução livre quer dizer gatilho de ação. Em linguagem de programação é executado sobre a forma de um algoritmo responsável por desencadear uma ação mediante um conjunto condições que podem regras, operadores lógicos, ciclo de repetição, etc.

O objetivo do algoritmo é evitar a sobrecarga de dados, resultantes da recolha contínua de dados na unidade de armazenamento, e a aumentar a periodicidade entre cada recolha de dados, visto que a gravação contínua de dados esgota rapidamente os recursos da base de armazenamento. O excesso de dados torna o pós-processamento moroso, devido a quantidade de informação recolhida, que no caso em particular do dispositivo 1 pode rondar os 600Mb por dia.

A solução encontrada para minimizar a excessiva acumulação de dados foi a criação de um algoritmo de decisão automático (*trigger action*), injetando-o dentro do *sketch* já

desenvolvido, assim delegando para o programa a decisão de salvar os dados em função de resultados de análise em tempo real.

Existem vários algoritmos de decisão automático, a ser utilizados no presente, dos mais simples com limites fixados pela amplitude aos mais sofisticados que reconhecem padrões de comportamento, a adaptativos a redes neurais. [31]

Algoritmos de decisão são relativamente ineficientes comparados com a capacidade de um ser humano treinado e com experiência em reconhecimento de padrões. [31] Os algoritmos avaliados em casos práticos, em certas condições, perdem parte dos dados relevantes e noutras condições geram falsas decisões. As razões que levam-nos a descartar resultados devem-se à falta de sensibilidade dos algoritmo e no caso dos falsos *triggers* devem-se a irregularidades nos dados provenientes de fonte de ruído.

No artigo *Understanding and parameter setting of STA/LTA trigger algorithm*, do autor Amadej Trnkoczy,[31] é proposto um algoritmo de decisão automático de simples implementação usado em sismógrafos, para deteção de eventos sísmico, suficientemente sensível para detetar movimentos fracos.

Um sismógrafo é um dispositivo de aquisição vibrações sísmicas, comparável em alguns aspetos com o dispositivo proposto neste trabalho. As diferenças prendessem sobretudo no tipo no tipo de sensor, especializado na aquisição de vibrações típicas de ondas sísmicas, com taxas de amostragem superiores ao dispositivo desenvolvido.

A principal relação entre este artigo e o trabalho realizado, está no comportamento das ações sísmicas de baixa vibração que o algoritmo pretende detetar, as mesmas ações que também afetam as construções civis com comportamentos similares. Assim foram utilizados alguns princípios, estruturas, regras e valores, como base no desenvolvimento do algoritmo de decisão automático.

O algoritmo de decisão proposto tem por princípio a comparação de um rácio determinado em intervalos de tempo periódicos e um parâmetro fixo, que o resultado da comparação entre ambos, determina a ação a desenvolver.

O rácio é conseguido através de uma relação de dois valores, por sua vez obtidos por tratamento estáticos de duas amostras com tamanhos diferentes. Neste caso, a recolha de duas mostras de registos do sensor, uma de curto prazo e outra de longo prazo de tempo.

É definido a utilização da média com ferramenta estatística, são calculadas duas médias, uma a curto prazo definida por *Short-Time-Average* (STA) e uma a longo prazo, definida *Long-Time-Average* (LTA). Os tempos das janelas temporais, proposto nesse artigo, para o cálculo das médias a curto STA entre 0,3s a 3s e longo prazo STA entre os 30s a 100s.

Após determinado o rácio das médias, o algoritmo proposto define como segundo passo a tomada de decisão, em dois níveis diferentes. Num primeiro nível a decisão feita através da introdução de limites ou tectos máximos e mínimos sobre os valores do rácio calculados entre médias, verificada em intervalos de tempo regulares e independentemente em cada um dos eixos ou direções. Num segundo nível a decisão é tomada com base num conjunto de regras, que verifica o rácio de cada um dos eixos, impõem limites mínimos e máximos, mas estabelecer regras de grupo ou conjunto (e.g. $X > 1$ e $Y > 1$ então executa uma pré-determinada ação).

Os valores ou parâmetros, para os tectos mínimos, máximos, regras e tempos para cada janela temporal, são definidos em função da origem da ação (longe ou perto), duração da ações, curta ou longa duração, tipo de ondas, etc.

Problemas associados a falsos *triggers*, originados por fontes ruído externas, são resolvidos por se ajustar os tempos das janelas temporais, e os parâmetros dos tectos mínimos e máximos e atribuição de regras.

Foi verificado que os valores obtidos variavam em função do nivelamento da superfície em que o dispositivo se aplicava. Esta situação pode ser ultrapassa com um algoritmo de auto nivelamento. Mas Esta solução não foi implementada, por várias razões, sendo uma delas de tornar o programa mais complexo, obrigado a ainda a um algoritmo auto calibração e um protocolo de cada vez que se parava o dispositivo. Outra razão é o caso de se mover no dispositivo em pleno funcionamento, as medições de nivelamento obtidas pelo nivelamento feito antes de iniciar as leituras perderiam o seu efeito, podendo a partir desse momento gerar falsas informações ao algoritmo de decisão automática.

A solução foi encontra após a análise dos valores e resultados das relações entre eles. A média e RMS são valores com posições absolutas ao um eixo, assim as relações são diretamente influenciadas pelas distâncias a um determinado eixo. De cada vez que o aparelho obtém uma nova inclinação essas relações mudam, que também dificulta a atribuição dos valores de tectos.

A solução é obter valores com posições relativas que não se alteram a cada novo plano de nivelamento. Assim foi utilizada uma variação da *root mean square* RMS Expressão 4, para analisar as leituras do dispositivo, abaixo apresenta-se a expressão usada.

$$Função_1 = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - x_{i-1})^2} \quad (9)$$

A solução proposta elimina a relação ao eixo e foca-se apenas no desvio entre valores sucessivos, acumulados. Numa situação normal os valores obtidos desta função tendem para ser manter próximo de um valor que se mantém constante. Numa situação anormal o sensor pode tender para um *bias* ou desvio, que pode ser linear, neste caso o algoritmo tem a capacidade de compensar, mas ser for irregular, causa falsos disparos, neste caso a melhor solução é substituir o sensor.

Janelas temporais

O algoritmo foi definido apenas um valor fixo na janela temporal de longo prazo, o valor definido foi de 30 segundos. A janela temporal de curto prazo que pode assumir vários valores que variam entre 0,5s a 3s, bastando para isso alterar algumas linhas do *sketch*. O optado na janela temporal a curto prazo, determina a taxa de amostragem, quanto menor for este valor, menor é a taxa de amostragem.

O rácio entre janelas temporais é feito em função da escolha da janela temporal a curto prazo. Por exemplo, se optarmos por um janela temporal de curto prazo com 3 segundos, o rácio é determinado em períodos também de 3s.

Ajustes à janela temporal de curto prazo são feitos no próprio *sketch*, pela alteração dois parâmetros encontrados nas linhas 129 e 130, podendo variar a janela temporal entre valores recomendados que variam entre 0,5s a 3s. Lembrando que a alteração destes dois parâmetros implica um acréscimo ou decréscimo da taxa de amostragem.

A cada alteração do parâmetro da janela temporal a curto prazo, é também necessário alterar o valor localizado na linha 122, do *sketch*, que define o valor de *loop* necessários a janela temporal. Este valor obtido empiricamente visto que o tempo de cada *loop* varia. Nas linhas 119 a 121 encontram-se exemplos de valores já determinados.

Tectos máximos individuais e regras de conjunto

O mecanismo de ação foi elaborado com regras lógicas, que avaliam rácios entre média a curto prazo e longo prazo. O mecanismo tem dois níveis de decisão, o primeiro nível é feito pela avaliação de rácios de médias em cada eixo individual. Num segundo nível de decisão é feita uma avaliação entre a relação dos resultados obtidos nos eixos X e Y, este de seguida é comparado com um parâmetro pré-definido e um conjunto de regras, que fazem decidir a ação a executar.

Neste algoritmo de decisão automática por opção, apenas são usados os dados dos eixos X e Y. Estes eixos definem o plano horizontal, enquanto o eixo Z é perpendicular, encontrado alinhado com o eixo da gravidade.

No primeiro nível de decisão do algoritmo avalia o rácio entre os valores obtidos de curta duração sobre os de longa duração e comparando com um parâmetro definido, para cada um dos eixos. Caso em algum dos eixos for ultrapassado o valor parametrizado são guardados os dados.

O ruído tem um comportamento aleatório, podendo surgir picos muito acima dos valores obtidos da média de longo prazo, por esta razão o parâmetro definido neste nível deve ser muito alto. Como referência os picos do desvio padrão de curta duração podem ser 5 a 6 vezes superiores ao desvio padrão de longa duração.

Por esta razão criou-se um segundo nível de decisão, com valores de parâmetros mais baixos, mas com uma regra de conjunto que define que os resultados obtidos, em ambos os eixos X e Y, devem ultrapassar um parâmetro pré-definido, acionando assim a decisão de guardar os dados.

Com referido o ruído é aleatório, a probabilidade de picos em ambos os eixos X e Y em simultâneo é relativamente baixa. Mas quando existe uma ação dinâmica, por norma surgem registos dessa ação em ambos os eixos do sensor, mas com intensidades diferentes, dependendo da direção da ação em relação aos eixos. Se o parâmetro for definido para obter uma boa sensibilidade, esta regra pode ser capaz de detetar ações dinâmicas de baixa intensidade, a definição desse parâmetro depende em muito da

O *sketch* está programado para enviar informações relativas aos eixos X e Y, a cada instante. Bastando para isso ligar o dispositivo 2 a um computador pessoal pela portas

USB. O IDE Arduino possui um monitor de portas COM, não é necessário definir a velocidade de transmissão pois esta define-se automaticamente, esta situação acontece apenas com este Dispositivo 2, numa situação normal tem que se definir velocidade de transmissão manualmente.

A informação disponibilizada pelo programa é em tempo real, apresenta os valores tratados de cada janela temporal, o rácio, para cada um dos eixos X e Y. Esta informação ajuda a definir os valores de cada um dos parâmetros, no próprio local de instalação. A definição dos valores dos parâmetros depende muito do níveis de ruído (interno e externo), devendo fazer uma avaliação rápida e defini-los no próprio local da aplicação.

A alteração dos parâmetros é feita no próprio *sketch*, através do IDE da Arduino. Estes localizam-se nas linhas 144 a 146, abaixo da linha comentada que diz *Definition of individual parameters and group rule*. O *sketch* completo encontra-se no Anexo B2.

Tempo de armazenamento de informação

A informação proveniente do sensor é gravada dentro do cartão SD em blocos de 30 segundos, dentro de um ficheiro com extensão txt, temporariamente na raiz do cartão, que são descartados caso o algoritmo de decisão automático tome essa opção. A situação oposta, em que o algoritmo aciona a opção de gravar os dados, automaticamente o tamanho do bloco de 30 segundos duplica, e caso o algoritmo for acionado um segunda vez este então triplica o tempo de armazenamento. O ficheiro temporário txt nunca ultrapassa os 90 segundos de dados, gravando automaticamente quando chega a este limite, para garantir que não haja perda de informação.

Durante o processo de gravação, o ficheiro txt é renomeado, dando-lhe um nome com 6 a 8 caracteres, que resultam da combinação dos números do mês, dia, hora e minutos em que o ficheiro foi gravado (e.g. 04/08/2017 (dia/mês/ano) 15:03:47 traduz-se num nome 84153.txt).

Durante o mesmo processo de gravação, o ficheiro é transferido da raiz do cartão para uma pasta com nome de Acc_Log.

6.4.3. Relógio e calendário - *Real Time Clock* RTC

Como foram acrescentados mais dois novos componentes, neste caso um RTC externo à placa de desenvolvimento e dois termómetros, foi necessário encontrar e instalar bibliotecas para operar estes componentes.

A biblioteca para operar o RTC encontra pré instalada no IDE Arduino, podendo-se encontrar em sessão Ficheiro -> Exemplos -> RTCLib -> PCF8523. Existe um *sketch*, com todas as instruções necessárias para operar o RTC. Quando instalada a bateria do RTC, este inicia com um tempo pré-definido, no *sketch* referido existe um comando que acerta o RTC automaticamente, através de uma sub-rotina que vai solicita o tempo ao computador pessoal e atualiza no RTC.

6.4.4. Termómetro

No caso do termómetro DS18B20 da Maxin, também existe no IDE Arduino uma biblioteca pré-instalada, no entanto optou-se por uma biblioteca mais completa da Adafruit para sensores de temperatura da NPX. Esta possui comandos mais simplificados com menos linhas de programação.

O protocolo *One Wire* usado para operar o sensor de temperatura, permite ligar no mesmo canal de comunicação, vários sensores de temperatura idênticos. Para obter a temperatura de um determinado sensor, é preciso aplicar um comando próprio, e indicar a ordem do sensor pretendido. Na biblioteca encontram-se exemplos e explicações sobre todo o processo, entre outras mais informações.

6.4.5. Bibliotecas

Tabela 6.2 - Biblioteca para sensor de temperatura NPX em linguagem Arduino

| | |
|---------------|-----------------------|
| Componente: | Sensor de temperatura |
| Biblioteca: | Dallas temp MAX31850 |
| Contribuidor: | Adafruit |

| Localização |
|-------------|
|-------------|

| |
|---|
| https://github.com/adafruit/MAX31850_DallasTemp |
|---|

6.4.6. Sketch

Declarações

Dispositivo 2

Declaração de bibliotecas

- Arduino - Wire (*comunication protocol*)
- Arduino - SPI (*comunication protocol*)
- Adafruit - MMA8451
- Adafruit - Sensors
- Bill Greiman - SDFat
 - Definição do sistema de ficheiro usado
- Arduino - Real Time Clock
 - Definição do modelo do RTC
- Arduino - One Wire (*comunication protocol*)
 - Adafruit - Dallas Temperature

Declaração de constantes

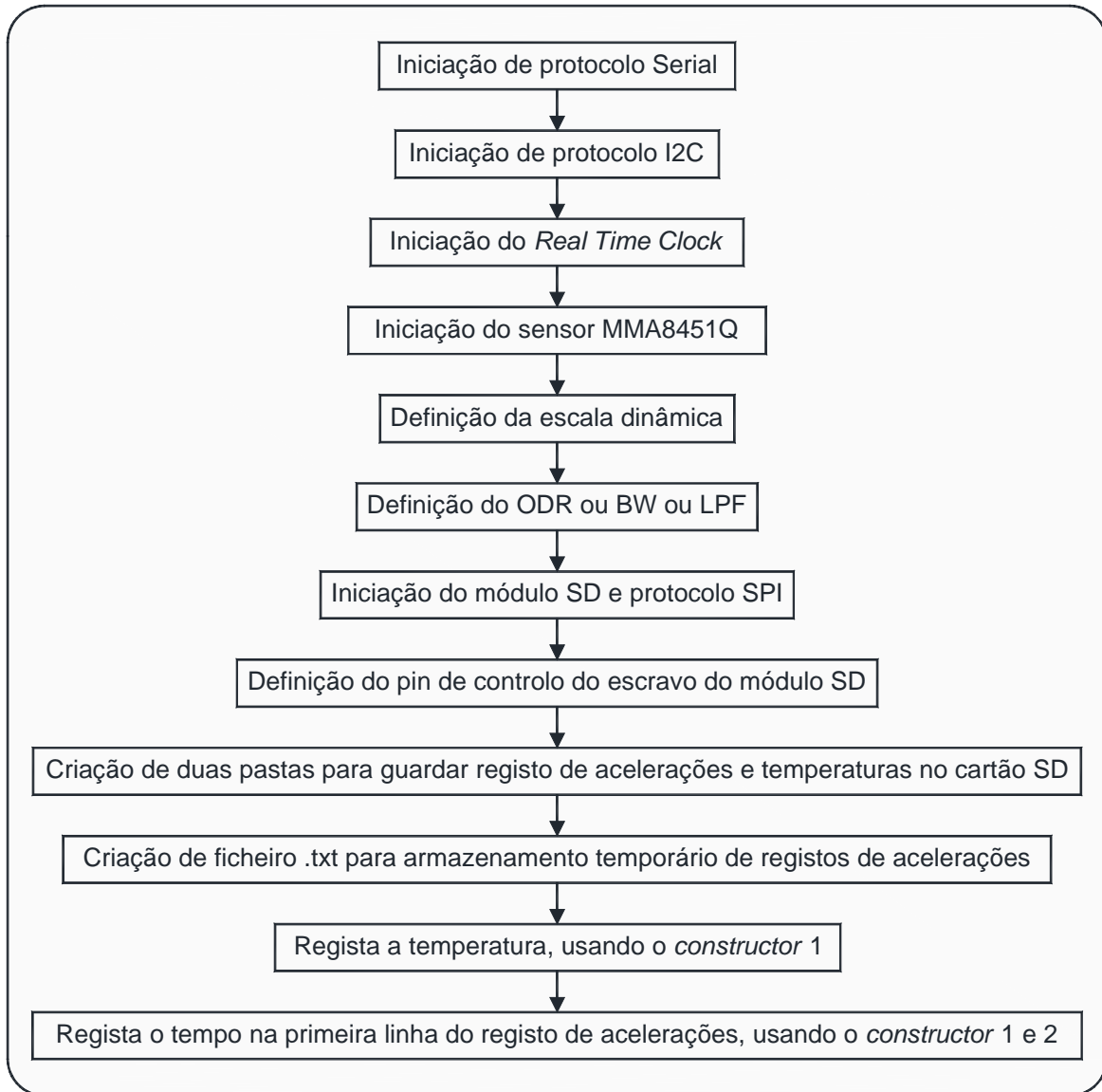
- Definição da morada I2C do sensor inercial
- Pin usado pelo protocolo SPI para controla módulo leitor de cartões SD
- Pin usado pelo protocolo One Wire para comunicar com os sensores de temperatura
- Setup de configuração para o sensor DS18B20
- Cruzamento de referências entre bibliotecas One Wire e Adafruit - Dallas Temperature

Declaração de variáveis

- Tempo
- Registo para o valor da aceleração eixo X
- Registo para o valor da aceleração eixo Y
- Short Time (ST) RMS eixo X
- Short Time (ST) RMS eixo Y
- Root Mean Square (RMS) de 30s eixo X
- Root Mean Square (RMS) de 30s eixo Y
- Rácio entre ST RMS e 30s de RMS eixo X
- Rácio entre ST RMS e 30s de RMS eixo Y
- Ciclo 1 (número de contagem tempo da janela temporal de curto prazo)
- Ciclo 2 (Determina o rácio entre ST RMS e 30s RMS e executa o *trigger action logaritm*)
- Ciclo 3 (Ações a executar após o *trigger action logaritm*)
- Número de contagem de registo (**n**), dentro do ciclo 1
- Número de contagem de registo (**q**), dentro do ciclo 2
- Número para contagem (**i**) a ser utilizado pelo comando for()
- Número de repetições do ciclo 3
- Número de ciclos ST (definem o tempo do Short Time RMS)
- Definição da matriz de armazenagem de registos de aceleração de uma janela temporal do eixo X
- Definição da matriz de armazenagem de registos de aceleração de uma janela temporal do eixo Y
- Número de blocos (*block*) (fator de multiplicação, depende tempo da janela temporal)
- Número que define o período de amostragem da temperatura em minutos
- Número do período de amostragem da temperatura em ciclo (obtido por conversão)
- Número que define o parâmetro de individual do *trigger action*
- Número que define a regra de conjunto do *trigger action*

Void setup()

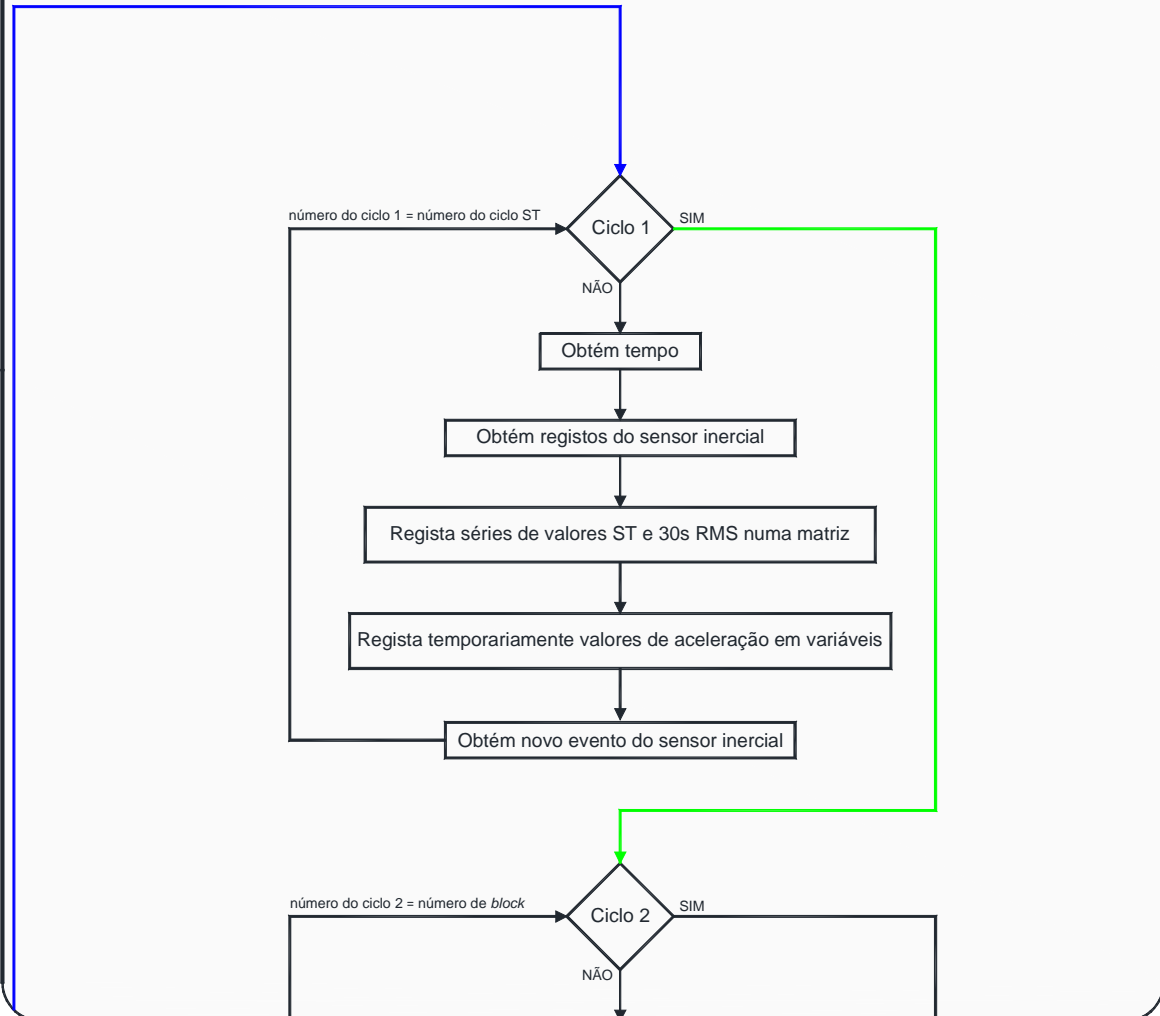
Dispositivo 2



Void loop()

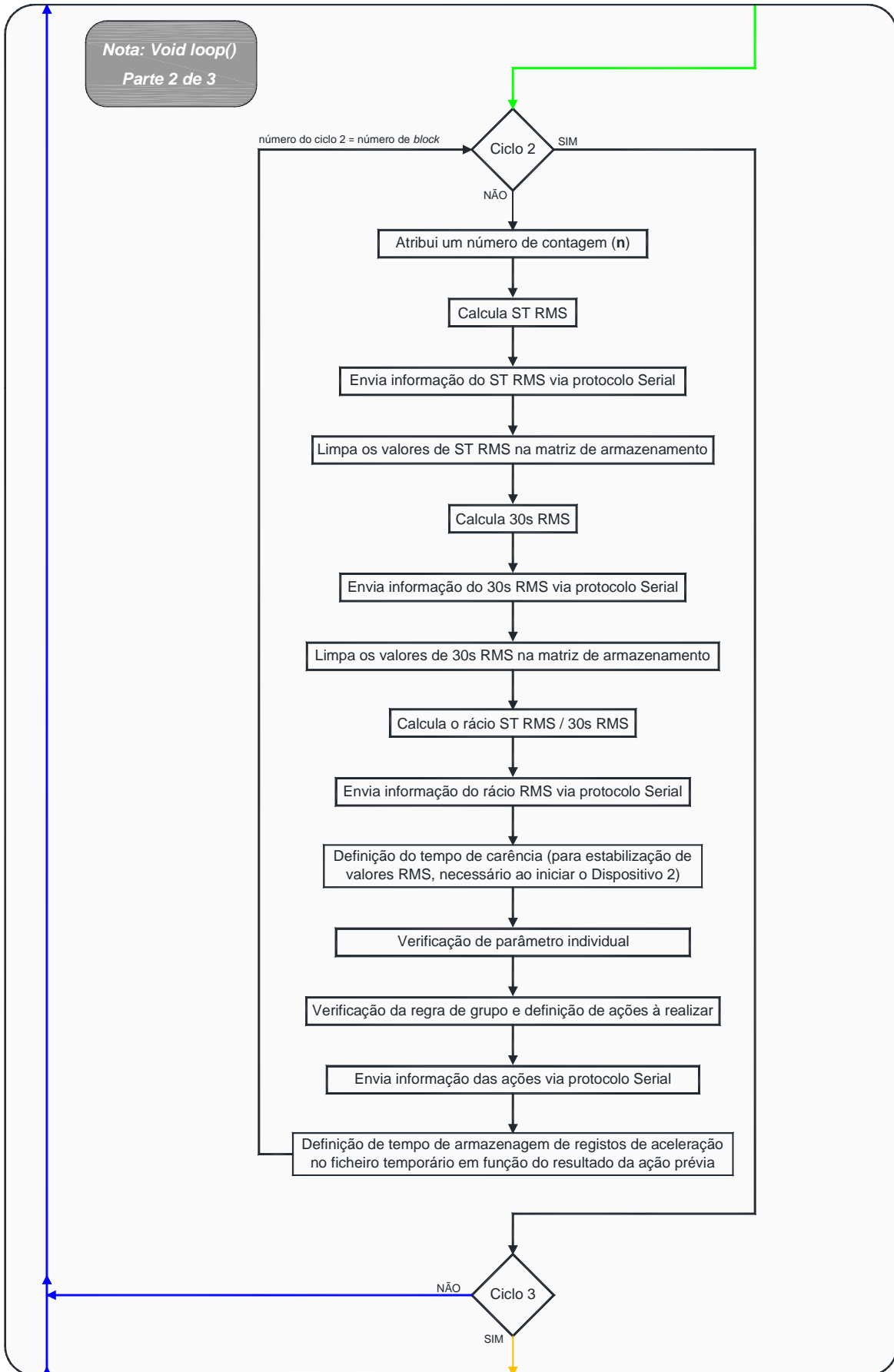
Dispositivo 2

Nota: Void loop()
Parte 1 de 3



Void loop()

Dispositivo 3



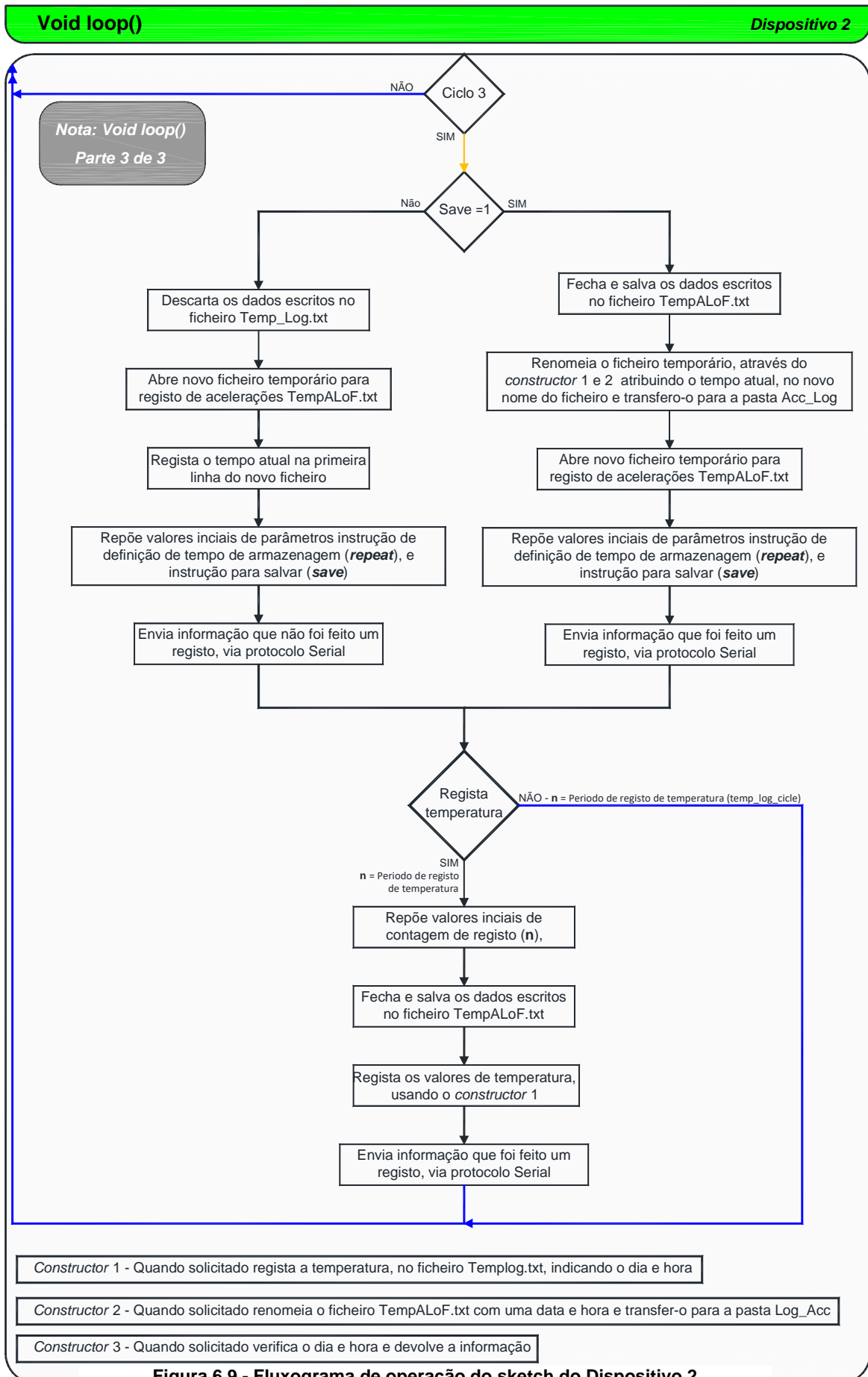


Figura 6.9 - Fluxograma de operação do sketch do Dispositivo 2

6.4.7. Teste de avaliação do *Sketch*

Foram feitos ensaios elementares para avaliar o comportamento e performance do *sketch* ou programa do Dispositivo 2, para assim verificando a conformidade dos resultados registados com os critérios estabelecidos e encontrar eventuais problemas

Relógio e calendário RTC

O relógio em operação contínua atrasou aproximadamente 6 minutos em 30 dias de operação, este ensaio realizou-se com temperaturas média que rondou os 16°C. De cada vez que se removem dados é conveniente ligar o dispositivo a um computador e correr o *sketch* de atualização existente no IDE da Arduino.

Bateria de polímeros de lítio

Executou-se um teste à bateria de lítio de 500mAh, que consistiu utilizar o dispositivo em modo de registo contínuo sem o algoritmo de decisão automática, e verificar quanto tempo ele consegue operar nestas condições. O resultado foi 15 horas de operação contínua.

Algoritmo de aquisição automático

Executou-se um teste de avaliação contínua do dispositivo 2, por um período de 8 horas, num 3º piso de um edifício. Utilizou-se o *sketch* com o algoritmo e parametrizou-se a regra individual com um valor de 1,10 e a regra de conjunto de 1,07. O aparelho operou normalmente sem falhas, obtiveram-se 7 eventos, dispersos ao longo do tempo em média 1 por hora.

7. Ensaios de validação do dispositivo 2

7.1. Introdução

Para determinar a credibilidade, estabelecer confiança dos registos obtido do sensor inercial MEMS instalado no dispositivo, e validar todo o sistema, foi desenvolvido um ensaio utilizando um outro aparelho de registo de vibrações com qualidade laboratorial, em condições reais de campo.

Os objetos de estudo selecionados que foram recolhidos e analisados dados de vibrações, com o fim de identificar as características dinâmicas, foram estruturas de construção civil e acessíveis ao público

O objetivo do ensaio foi recolher dados das estruturas selecionadas, em modo de registo contínuo por um período de alguns minutos, operando com fontes de energia interna, e durante esse tempo provocar ações de forma que a estrutura vibrem, assim obtendo dados de vibrações.

Posteriormente em gabinete, executou-se um processamento dos dados recolhidos, utilizando a metodologia descrita no ponto 5.3.10, transformando-os com recurso a FFT, com o objetivo fazer análises comparativas entre resultados obtidos dos dois dispositivos, atendendo aos seguintes aspetos:

- Diferença entre valores de aceleração e espectros de frequências.
- Avaliação da sensibilidade do sensor MMA8451Q em comparação com o dispositivo padrão;
- Diferenças entre localização de picos de frequências e respetivos valores da amplitude.

7.2. Comparação de especificações dos dispositivos

Descrevem-se de seguida as principais especificações e diferenças que distinguem os dois dispositivos usados nos ensaios.

Dispositivo 2 - Dispositivo de aquisição e monitorização de vibrações de baixo custo

- Sensor inercial MEMS capacitivo da marca NPX, modelo MMA8451Q triaxial;
- Microcontrolador – “Adafruit feather M0 basic proto” com um processador ATSAM21G18 ARM Cortex M0 de 32bit e velocidade de relógio a 48MHz;
- Suporte estrutural – Todos os equipamentos são ficados por parafusos sobre uma placa acrílica de 5mm e que esta fixa a uma caixa em ABS;
- As condições de aplicação do dispositivo no local de estudo foi feito aplicando-o sobre a superfície da estrutura sem recurso sistema de fixação, meramente apoiados sobre a superfície;
- Programa – De média complexidade, tem como funções leitura de dados do sensor de vibrações, do sensor de temperatura, e registo de dados condicionados a um algoritmo de decisão automática numa unidade de armazenamento de dados;
- Taxa de aquisição dos dados – $\pm 296\text{Hz}$;
- Peso - 0,346kg.

Dispositivo padrão - Dispositivo de aquisição e monitorização de vibrações com características laboratoriais

- Sensores piezoelétricos unidirecionais, da marca PCB Piezotronics, modelo 352C03 [32]
- Placa de aquisição de dados – de 4 canais com uma taxa máxima de 51200 amostras por segundo, da marca National Instruments, modelo NI-9234 [33]
- Dispositivo de armazenamento de dados – A gestão e armazenamento de dados foi feita com recurso a computador pessoal e um *software* desenvolvido para este efeito, designado por LabVIEW da marca National Instruments. [34]
- Estrutura de suporte – Os sensores foram montados num cubo de aço por meio de parafuso, simplesmente apoiados sobre a superfície das estruturas.
- A taxa de aquisição de dados usada nos ensaios foi aproximadamente 825Hz

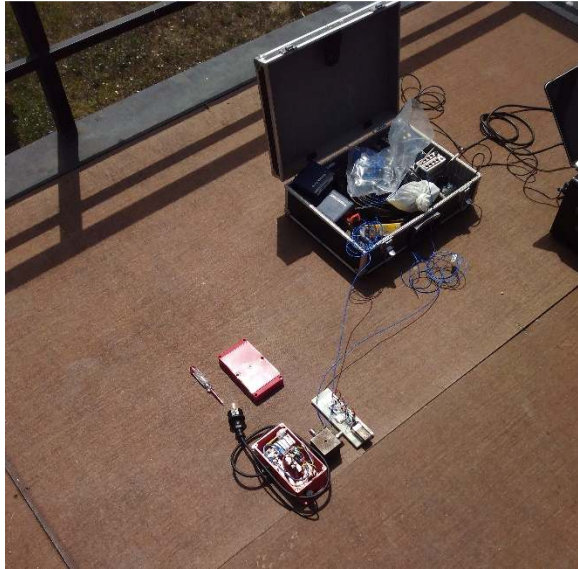


Figura 7.1 - Dispositivo 2 e dispositivo padrão

7.3. Estruturas de ensaio

A seleção e escolha nas estruturas de estudo teve em conta alguns aspetos técnicos, estas tinham que ter características estruturais simples, em que o comportamento dinâmico fossem fácil de compreensão e fáceis de serem excitadas dinamicamente.

7.3.1. Miradouro

A primeira estrutura foi uma plataforma de observação, que se designou-se por Miradouro. Tem a particularidade de ter um comportamento similar um pêndulo invertido. É composta por quatro pilares H metálicos, uma plataforma em contraplacado que forma o único piso pedonal e encontra-se em bom estado de conservação.

O miradouro é uma plataforma para observação do espaço envolvente ao circuito pedonal POLIS de Leiria, localizado nas seguintes coordenadas 39°45'5.20"N, 8°48'59.10"W. Esta situa-se no extremo norte do percurso pedonal POLIS que acompanha o curso do Rio Lis junto ponte A19 sobre rio Lis.

É composta por uma estrutura metálica tendo por quatro pilares metálico que suportam um piso em contraplacado marítimo, este acessível por uma escada do lado oeste da estrutura. A estrutura tem uma forma paralelepipedal com 4m de alto e 3m de largura e comprimento.



Figura 7.2 - Miradouro

7.3.2. Ponte Pedonal

A segunda estrutura foi uma Ponte Pedonal de simples conceção, em bom estado de conservação, composta por treliças metálicas com perfis quadrangulares, com uma plataforma pedonal formada por travessas de madeira, o seu comportamento é similar a uma corda vibrante.

Junto ao açude do Rio Lis, localizado a Norte em Leiria, localiza-se uma ponte pedonal sobre o Rio Lis com as coordenadas 39°45'4.46"N, 8° 48'44.20"W.

A sua estrutura é de construção metálica com uma forma trapezoidal, constituída por elementos retangulares, o piso é construído com travessas em madeira maciça. As dimensões da estrutura são as seguintes, o vão é aproximadamente 32m, a largura aproximada é 3,20m e altura aproximada é 3,0m.

Os acelerómetros dos dois dispositivos foram aplicados a meio da ponte e no centro do piso, orientados com os eixos da estrutura. O eixo X ficou orientado Sul-Norte, o eixo Y Oeste-Este e o eixo Z ficou orientado paralelamente com o eixo da gravidade da terra.

As referências locais Y e Z dos sensores dos dispositivos 2 e padrão estão trocadas, devido a aplicação sensor do dispositivo 3 estar na parte inferior do microcontrolador, que resultou numa inversão dos sentidos dos eixos.



Figura 7.3 - Ponte Pedonal

7.4. Caracterização dos ensaios

7.4.1. Miradouro

Os sensores de vibração de ambos os dispositivos foram colocados em paralelos, sobre a plataforma de contrapalado, junto ao pórtico mais a sul da estrutura, estes ficaram alinhados com eixos principais da própria estrutura. O eixo X foi definido no eixo este-oeste, com sentido positivo este, o eixo Y foi definido norte-sul com o sentido positivo sul. O dispositivo padrão só tem dois sensores, portanto neste ensaio ignorou-se o eixo Z do sensor do dispositivo 2. O sentido do eixo Y do dispositivo 2 está invertido, por que o sensor está fixo na parte inferior da placa de desenvolvimento.

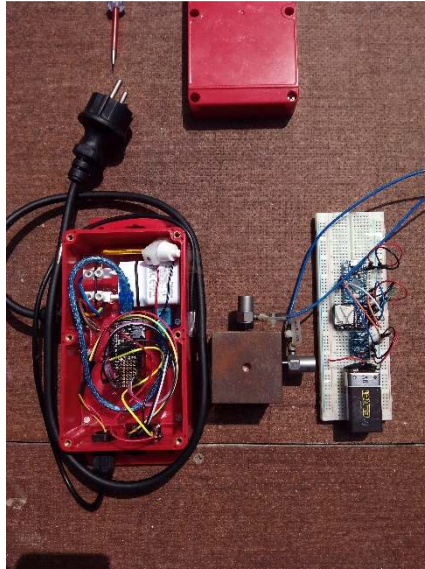


Figura 7.4 - Dispositivo 2 (esquerda), Dispositivo Padrão (centro) sobre o miradouro

Após feitos fazer trabalhos preparatórios, inerentes a montagem e ligações, ambos os aparelhos foram iniciados em simultâneo. Logo que se iniciaram foi feitos alguns impulsos sobre a estrutura, para obter um registo forte que serviria como *baseline* para ambos os aparelhos e necessário posteriormente para alinhar ambos registos de aceleração.

Foram necessário apenas alguns minutos para obter dados necessários para fazer as comparações necessárias, assim que se decidiu que já haveria tempo suficientes desligaram-se ambos aparelhos e moveram-se para o novo campo de ensaio.

7.4.2. Ponte Pedonal

Os sensores foram colocados sobre pavimento pedonal de madeira, no meio e centro da estrutura. De modo idêntico ao ensaio anterior, ambos os sensores ficaram alinhados com eixos principais da própria estrutura. Definiu-se com eixo X o eixo longitudinal da ponte, com alinhamento norte-sul e sentido positivo para sul. O Y com eixo transversal, com alinhamento este-oeste e sentido positivo este. O eixo Z ficou alinhado verticalmente com sentido positivo para cima.

De igual forma que no ensaio anterior o dispositivo padrão só tem dois sensores assim foram medidos os eixos Y e Z. O sentido do eixo Z do dispositivo 2 está invertido com o do dispositivo padrão, por estar fixo na parte de debaixo da placa de desenvolvimento.

O ensaio decorreu nos mesmos parâmetros do ensaio anterior, após iniciados ambos os dispositivos em simultâneo, geraram-se impulso na estrutura para obter um registo forte que serviria como *baseline* para ambos os aparelhos e necessário posteriormente para alinhar ambos registos de aceleração.



Figura 7.5 - Dispositivo 2 (esq.), Dispositivo Padrão (centro) sobre a ponte pedonal

7.5. Apresentação de resultados

Para conseguir executar análises comparativas entre dados recolhidos de ambos os dispositivos, determinou-se que deviam ser apresentadas comparações em termos de diferenças de valores de aceleração, e frequências detetas e amplitude de sinal.

Para desenvolver gráficos de tempo/aceleração, primeiro teve que ser converter os dados em acelerações, nivela-los e sincroniza-los por um momento chave, que se designou por *baseline*. Desenvolveram-se gráficos para ambos os sensores, ambos locais ensaiados e para os eixos. Depois fizeram-se sobreposições entre dados dos dois dispositivos por cada eixo e local, assim produzindo quatro gráficos tempo/aceleração. Apenas se analisou um porção do tempo total recolhido, o momento escolhido o imediatamente a seguir ao primeiro impulso executado em cada estrutura.

Fez-se também um levantamento da aceleração máxima positiva e negativa, gerada pelo primeiro impulso efetuado em cada estrutura. Calculou-se a relação entre o valor do dispositivo padrão e dispositivo 2 para determinar diferença entre ambos os dispositivos.

Para produzir gráficos logarítmicos frequência/amplitude utilizou-se o método descrito no ponto 5.3.10, empregando o programa Seismosignal para gerar gráfico e extrair valores das frequência e amplitudes detetadas. A semelhança da análise anterior usou-se apenas uma porção do tempo total, os dados usados no programa foram os mesmos usados na análise anteriormente de tempo/aceleração.

7.5.1. Resultados do Miradouro

Apresenta-se de seguida os gráficos resultantes dos valores de aceleração ao longo do tempo, obtidos dos dispositivos 2 e padrão, durante o ensaio efetuado sobre a estrutura do Miradouro.

Gráfico 7.1 - Tempo/aceleração ensaio sobre Miradouro, eixo X.

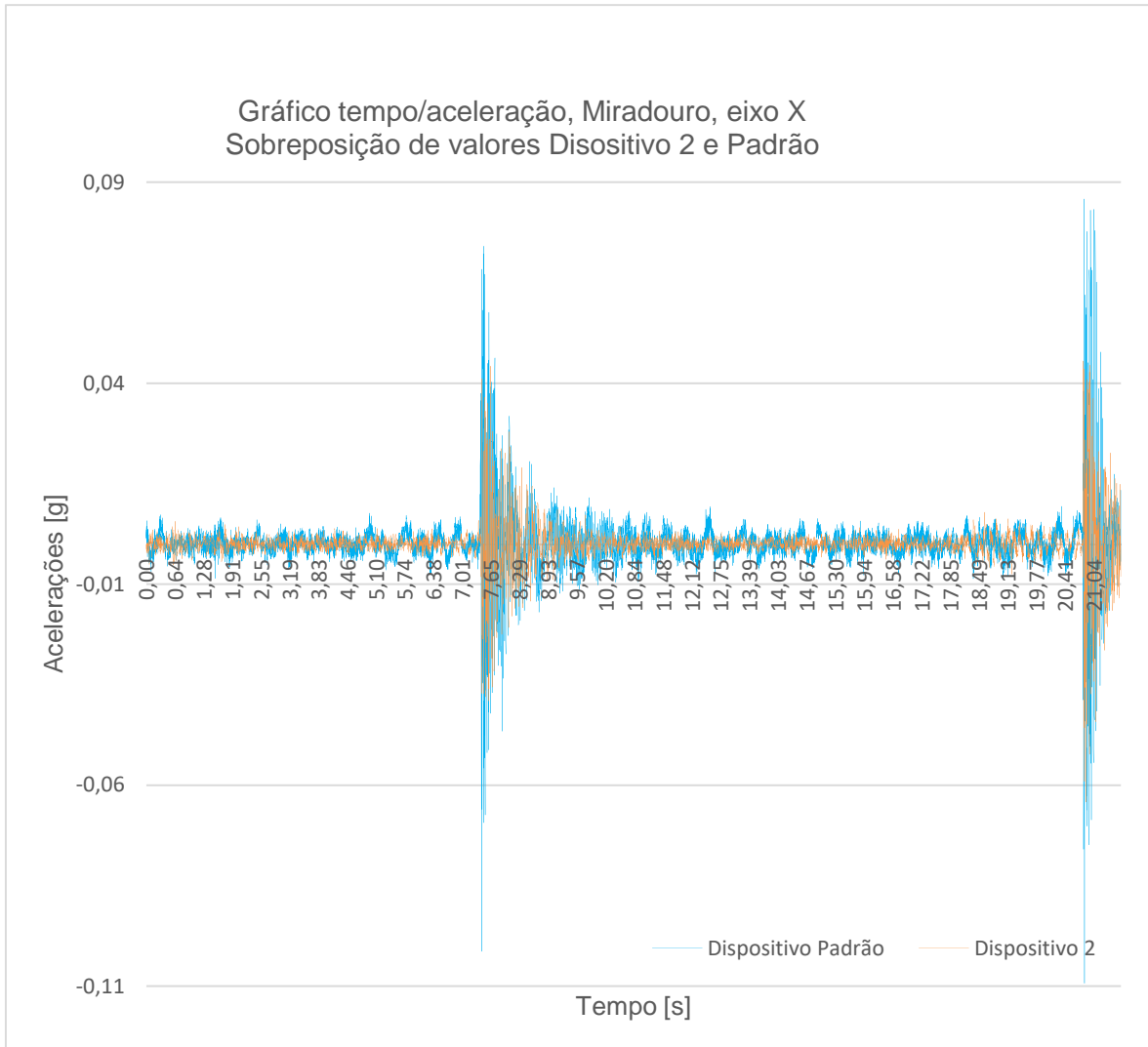
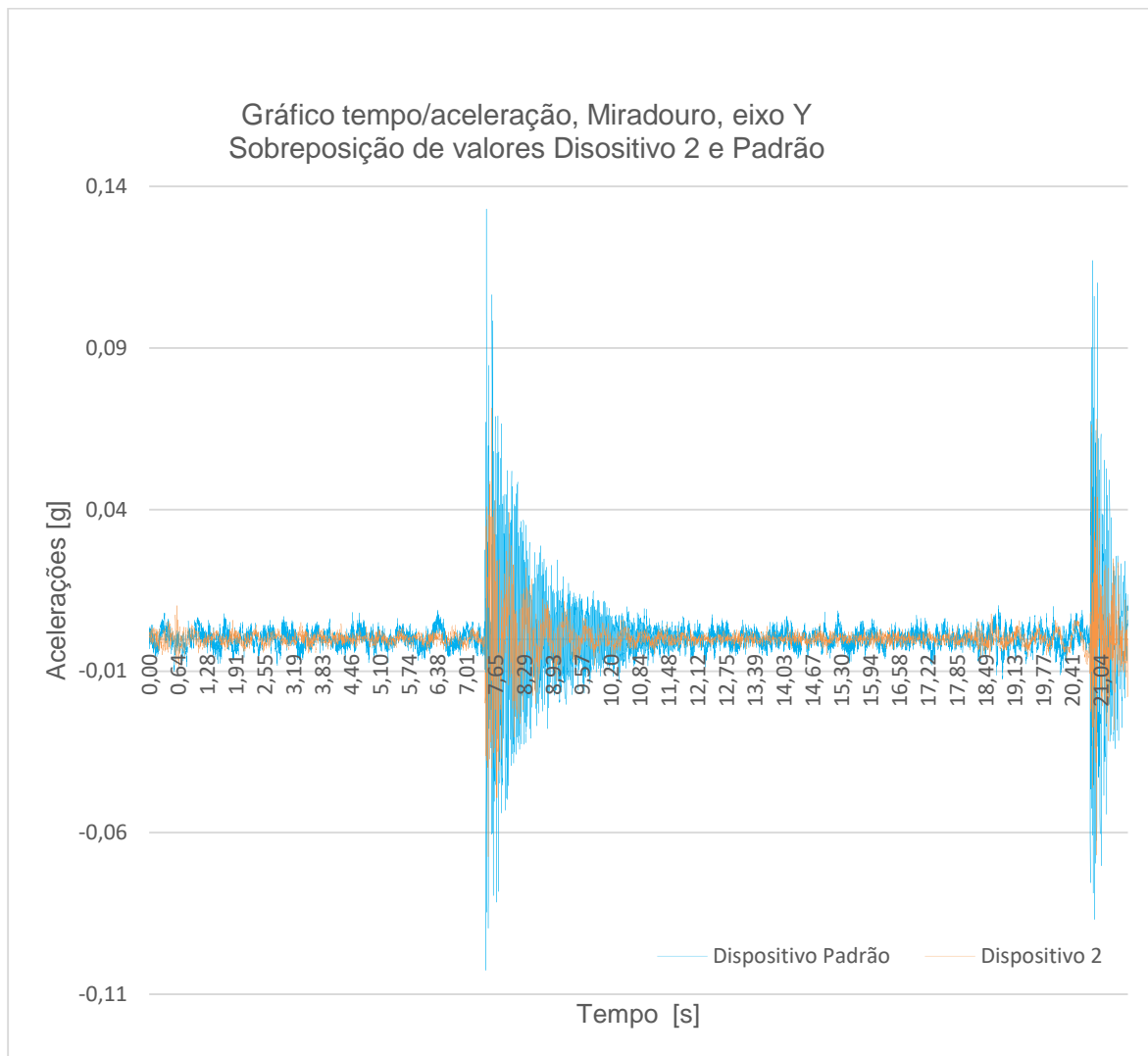


Gráfico 7.2 - Tempo/aceleração ensaio sobre Miradouro, eixo Y.

Valores de aceleração máxima e mínima de pico, do ensaio da estrutura do Miradouro, de ambos os dispositivos por eixo.

Tabela 7.1 - Valores de aceleração de pico, do ensaio feito no Miradouro

| Dispositivo | Eixo | Acelerações máx. | |
|--------------------|------|------------------|-----------------|
| | | Positiva [g] | Negativa [g] |
| Dispositivo padrão | X | 0,0858 | -0,1092 |
| | Y | 0,1330 | -0,1025 |
| Dispositivo 2 | X | 0,0455 | -0,0642 |
| | Y | 0,0860 | -0,0951 |

Relação entre valores de aceleração máxima de pico, do dispositivo padrão sobre os dispositivos 2, aceleração máxima e mínima, por eixo.

Tabela 7.2 - Relação entre valores de pico dos dispositivos 2 e padrão, ensaio do Miradouro

| Relações | | |
|----------|---------|----------|
| eixo | máximas | (mínimas |
| | [g] | [g] |
| X | 1,89 | 1,70 |
| Y | 1,55 | 1,08 |

Gráficos de frequência e amplitude de Fourier e identificação dos principais picos de frequência, na direção X.

Gráfico 7.3 - Esp. freq. eixo X disp. 2, ensaio Miradouro

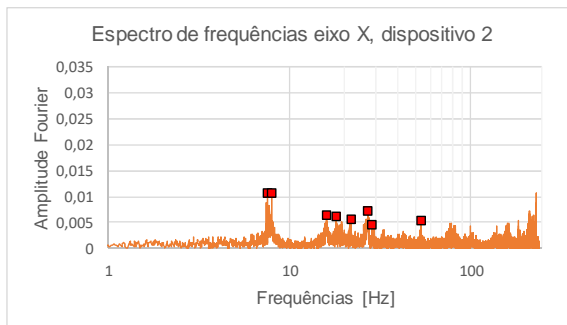


Gráfico 7.4 - Esp. freq. eixo X disp. padrão, ensaio Miradouro

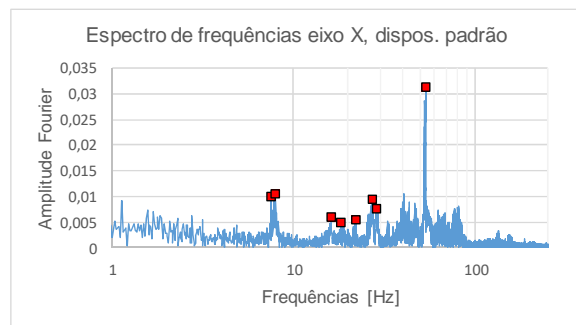


Tabela de comparação de picos detetados dos dispositivos 2 e padrão com a determinação de variações entre amplitude por frequência, na direção X.

Tabela 7.3 - Valores de frequências e amplitude, obtidos do eixo X, no Miradouro

| Dispositivo 2 - Eixo X | | | Dispositivo Padrão - Eixo X | | | Δf |
|------------------------|----------|--------------|-----------------------------|----------|--------------|------------|
| Ordem | f [Hz] | Amp. Fourier | Ordem | f [Hz] | Amp. Fourier | |
| 1 | 7,76483 | 0,00934 | 36 | 7,49068 | 0,00997 | 0,274 |
| 3 | 7,96346 | 0,00847 | 34 | 7,96988 | 0,01047 | -0,006 |
| 100 | 16,08945 | 0,00584 | 265 | 16,09109 | 0,00598 | -0,002 |
| 115 | 18,09385 | 0,00568 | 540 | 18,134 | 0,00483 | -0,040 |
| 248 | 21,94015 | 0,00464 | 364 | 21,94239 | 0,00552 | -0,002 |
| 7 | 26,99632 | 0,00814 | 39 | 26,98662 | 0,00943 | 0,010 |
| 41 | 28,69375 | 0,00674 | 84 | 28,7521 | 0,00767 | -0,058 |
| 13 | 53,18004 | 0,00795 | 1 | 53,2166 | 0,03135 | -0,037 |

Gráficos de frequência e amplitude de Fourier e identificação dos principais picos de frequência, na direção Y.

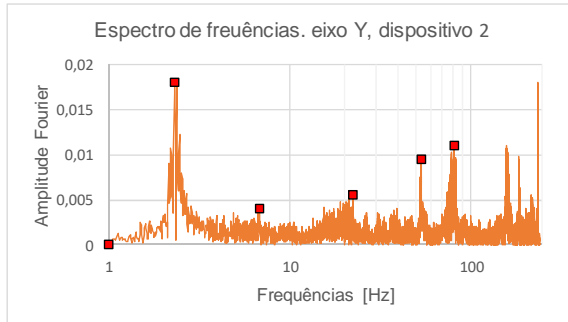
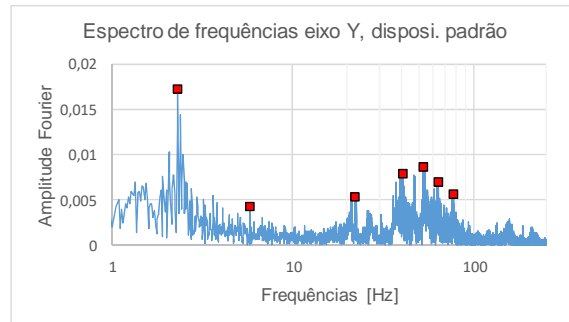
Gráfico 7.5 - Esp. freq. eixo Y disp. 2, ensaio Miradouro**Gráfico 7.6 - Esp. freq. eixo Y disp. padrão, ensaio Miradouro**

Tabela de comparação de picos detetados dos dispositivos 2 e padrão com a determinação de variações entre amplitude por frequência, na direção Y.

Tabela 7.4 - Valores de frequências e amplitude, obtidos do eixo Y, no Miradouro

| Dispositivo 2 - Eixo Y | | | Dispositivo padrão - Eixo Y | | | Δf |
|------------------------|----------|--------------|-----------------------------|----------|--------------|------------|
| Ordem | f [Hz] | Amp. Fourier | Ordem | f [Hz] | Amp. Fourier | |
| 1 | 2,38362 | 0,01685 | 1 | 2,32034 | 0,01725 | 0,063 |
| 1203 | 6,77165 | 0,00361 | 386 | 5,80086 | 0,00422 | 0,971 |
| 65 | 22,46383 | 0,00806 | 173 | 22,24504 | 0,00535 | 0,219 |
| | | | 14 | 40,42949 | 0,00794 | |
| 13 | 53,16198 | 0,01083 | 7 | 53,06528 | 0,00863 | 0,097 |
| | | | 45 | 63,48161 | 0,00696 | |
| | | | 136 | 76,94969 | 0,00567 | |
| 73 | 81,38623 | 0,00766 | | | | |

7.5.2. Resultados da Ponte Pedonal

Apresenta-se de seguida os gráficos resultantes dos valores de aceleração ao longo do tempo, obtidos dos dispositivos 2 e padrão, durante o ensaio efetuado sobre a estrutura da Ponte Pedonal.

Gráfico 7.7 - Tempo/aceleração ensaio sobre Ponte Pedonal, eixo Y.

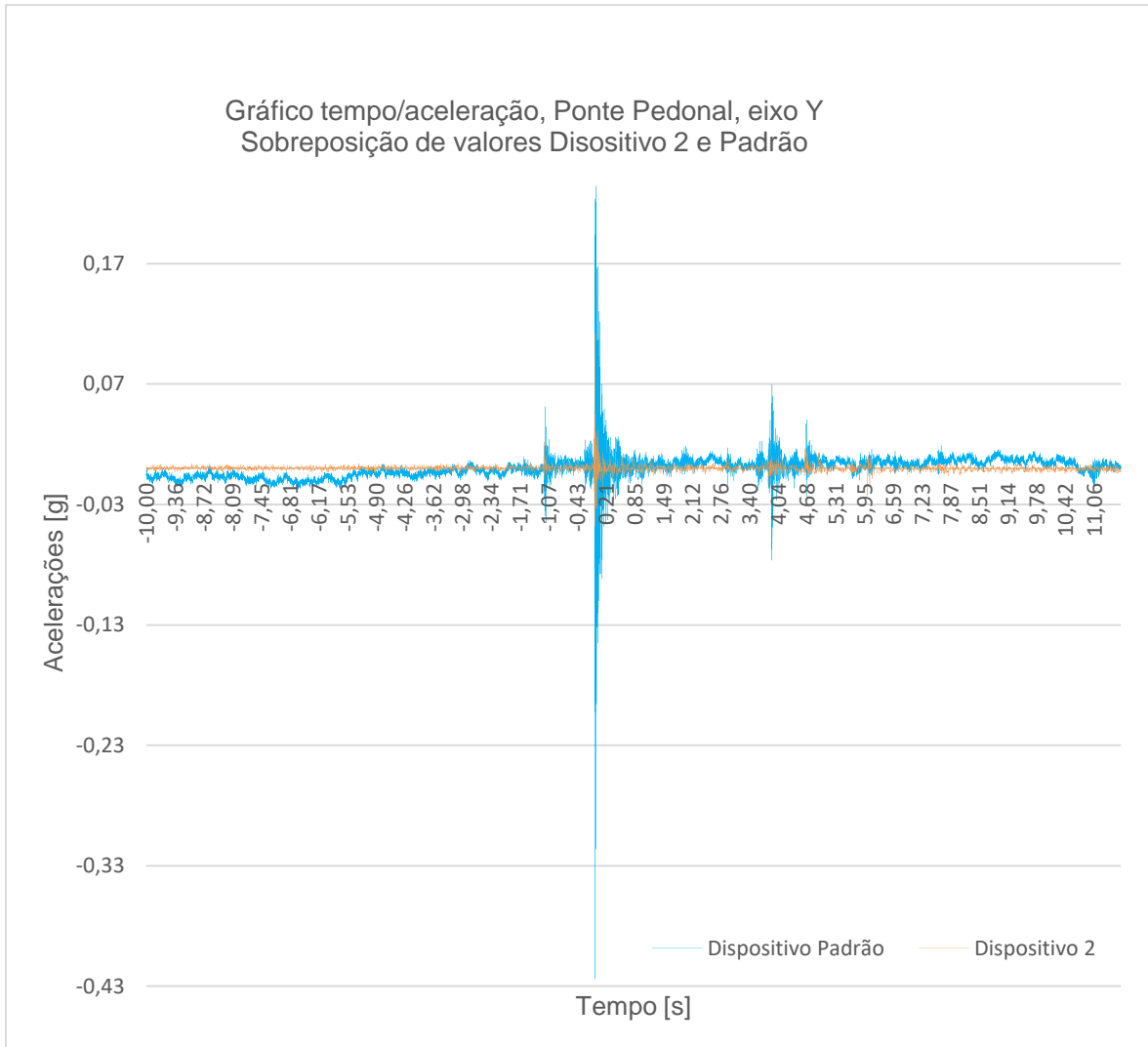
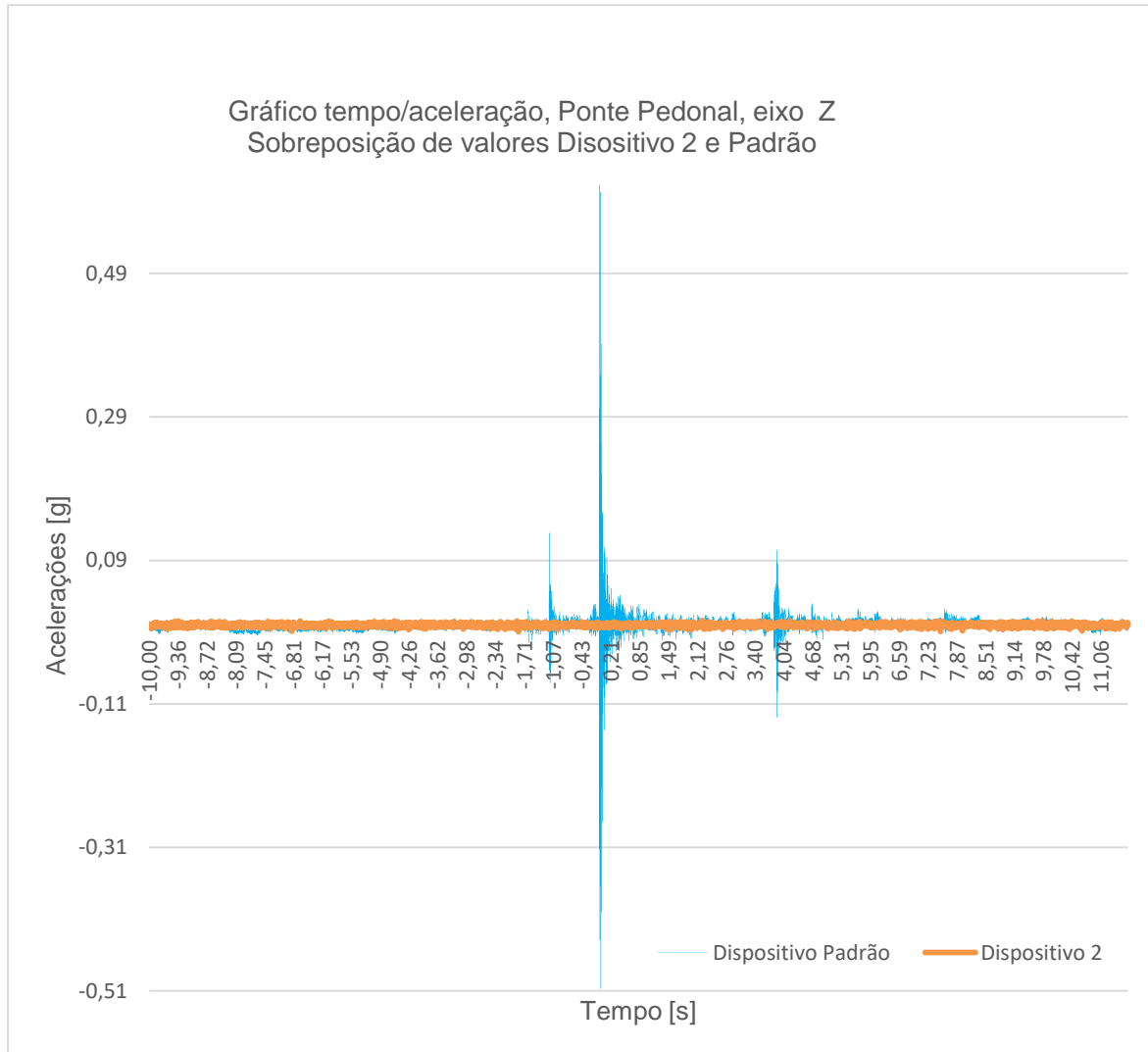


Gráfico 7.8 - Tempo/aceleração ensaio sobre Ponte Pedonal, eixo Z.



Valores de aceleração máxima e mínima de pico, do ensaio da estrutura Ponte Pedonal, de ambos os dispositivos por eixo.

Tabela 7.5 - Valores de aceleração de pico, do ensaio feito na Ponte Pedonal

| Dispositivo | Eixo | Acelerações máx. | |
|--------------------|------|------------------|-----------------|
| | | Positiva [g] | Negativa [g] |
| Dispositivo padrão | Y | 0,2345 | -0,4239 |
| | Z | 0,6131 | -0,5068 |
| Dispositivo 2 | Y | 0,1123 | -0,0482 |
| | Z | 0,0061 | -0,0095 |

Relação entre valores de aceleração máxima de pico, do dispositivo padrão sobre os dispositivos 2, aceleração máxima e mínima, por eixo.

Tabela 7.6 - Relação entre valores de pico dos dispos. 2 e padrão, ensaio na Ponte Pedonal

| Relações | | |
|----------|---------|----------|
| eixo | máximas | (mínimas |
| | [g] | [g] |
| X | 2,09 | 8,80 |
| Y | 99,84 | 53,11 |

Gráficos de frequência e amplitude de Fourier e identificação dos principais picos de frequência, na direção Y.

Gráfico 7.9 - Esp. freq. eixo Y disp. 2, ensaio Ponte Pedonal

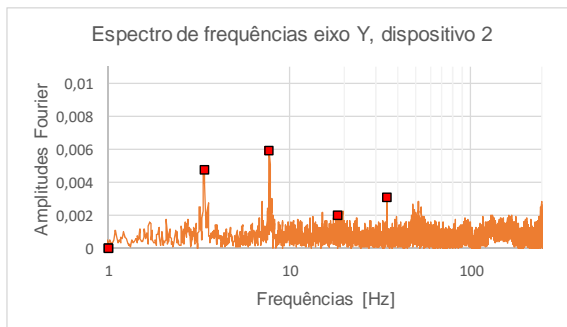


Gráfico 7.10 - Esp. freq. eixo Y disp. padrão, ensaio Ponte Pedonal

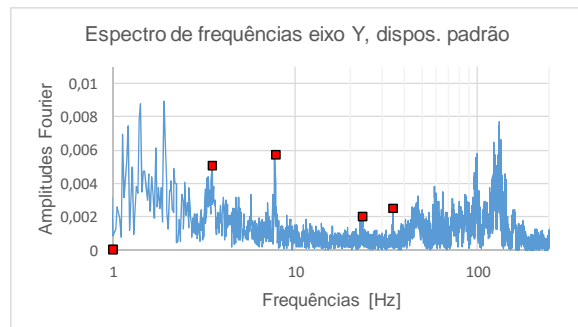


Tabela de comparação de picos detetados dos dispositivos 2 e padrão com a determinação de variações entre amplitude por frequência, na direção Y.

Tabela 7.7 - Valores de frequências e amplitude, obtidos do eixo Y, na Ponte Pedonal

| Dispositivo 2 - Eixo Y | | | Dispositivo padrão - Eixo Y | | | Δf |
|------------------------|----------|--------------|-----------------------------|----------|--------------|------------|
| Ordem | f [Hz] | Amp. Fourier | Ordem | f [Hz] | Amp. Fourier | |
| 7 | 3,3969 | 0,00477 | 149 | 3,53096 | 0,00505 | -0,134 |
| 1 | 7,77472 | 0,00593 | 67 | 7,79333 | 0,00573 | -0,019 |
| 107 | 18,60119 | 0,00204 | 1733 | 23,5061 | 0,00203 | -4,905 |
| 25 | 34,49576 | 0,00309 | 1095 | 34,50252 | 0,00250 | -0,007 |

Gráficos de frequência e amplitude de Fourier e identificação dos principais picos de frequência, na direção Z.

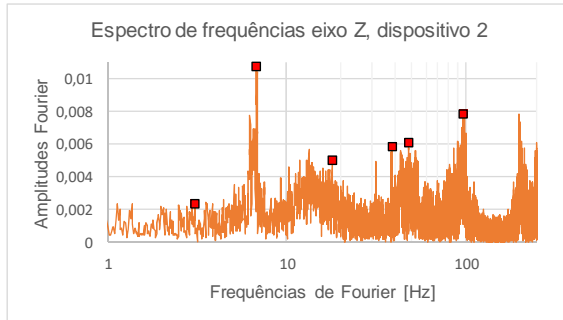
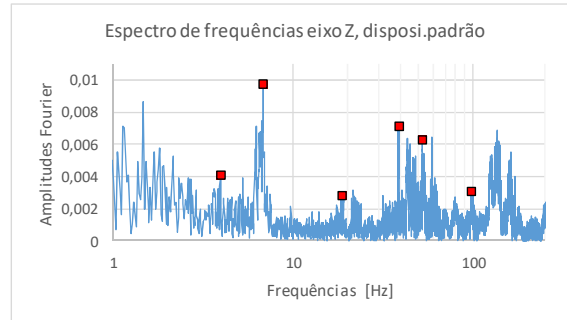
Gráfico 7.11 - Esp. freq. eixo Z disp. 2, ensaio Ponte Pedonal**Gráfico 7.12 - Esp. freq. eixo Z disp. padrão, ensaio Ponte Pedonal****Gráfico 19 - Esp. freq. eixo Z disp. 2, ensaio P. Pedonal** **Gráfico 20 - Esp. freq. eixo Z disp. padrão, ensaio P. Pedonal**

Tabela de comparação de picos detetados dos dispositivos 2 e padrão com a determinação de variações entre amplitude por frequência, na direção Z.

Tabela 7.8 - Valores de frequências e amplitude, obtidos do eixo Z, na Ponte Pedonal

| Ordem | f [Hz] | Amp. Fourier | Ordem | f [Hz] | Amp. Fourier | Δf |
|-------|-------------|--------------|-------|-------------|--------------|------------|
| 4189 | 3,06992 | 0,00232 | 826 | 3,95972 | 0,00410 | -0,890 |
| 1 | 6,79379 | 0,01072 | 1 | 6,83493 | 0,00971 | -0,041 |
| 389 | 17,91091 | 0,00500 | 2111 | 18,81497 | 0,00279 | -0,904 |
| 155 | 38,78276 | 0,00586 | 5 | 38,73967 | 0,00711 | 0,043 |
| 129 | 48,01069 | 0,00610 | 25 | 52,43475 | 0,00630 | -4,424 |
| 9 | 97,56542 | 0,00786 | 1768 | 98,76598 | 0,00306 | -1,201 |

7.6. Comparação de resultados

Os valores apresentados nos gráficos tempo/aceleração, registados pelos dispositivos ensaiados, apresentam diferenças evidentes. As diferenças verificam-se sobre tudo nas trajetórias de valores registados. A trajetória apresentada pelo dispositivo 2 é menos dinâmicas, em comparação com o dispositivo padrão, que deteta movimentos mais subtis. Existem também diferenças entre os valores de máximos e mínimos de pico. O dispositivo padrão, para o ensaio sobre o Miradouro registou valores globais médios, aproximadamente 1,55 vezes superiores, assim demonstrando uma maior sensibilidade.

Os resultados apresentados pelo dispositivo 2, no eixo Z do ensaio efetuado sobre a Ponte Pedonal, aparentemente são quase estáticos em comparação com o do dispositivo padrão. Verificou-se no sensor MEMS do dispositivo 2, a existência de diferenças de resultados

obtidos pelo eixo Z em comparação com os eixos X e Y. Uma das razões é que este sensor MEMS, possui uma estrutura mecânica simétrica entre os eixos X e Y, e assimétrica em relação ao eixo Z, produzindo resultados com menor sensibilidade neste eixo, esta situação condiciona a comparação de resultados.

Da análise dos dados obtidos pela transformação rápida de Fourier, verificou-se que o dispositivo 2 obteve menos sucesso na determinação de frequências e amplitudes acima dos 50Hz. Contudo o sensor MEMS conseguiu detetar as primeiras frequências naturais das estruturas com pouca diferença. Em relação as diferenças entre valores de amplitude de Fourier, do sensor MEMS e dispositivo padrão, verificou-se que não há grandes diferenças para também primeiras frequências até aos 50Hz.

O dispositivo 2 teve menos sucesso em identificar frequências e amplitudes de Fourier acima dos 50Hz. A razão para este facto deve-se á própria taxa de amostragem, que no dispositivo 2 foi perto de 300Hz em comparação com o dispositivo padrão que foi de 825Hz.

Se considerarmos a taxa de amostragem de 50Hz, o dispositivo 2 para esta frequência obtém sensivelmente 6 amostras, já no caso do dispositivo padrão consegue obter sensivelmente 16 amostras.

Quanto maior o número de amostras recolhidas, melhor a reconstituição do sinal efetuada pela transformação FFT, para uma determinada frequência. A utilização de uma taxa de amostragem mínima para uma frequência de referência, pode originar uma fraca ou nula reconstituição do sinal original. A causa deve-se á dessincronização da fase taxa de amostragem em relação á fase do sinal analógico. Assim o dispositivo padrão tem uma maior capacidade de identificação de frequências e amplitudes de Fourier, em relação ao dispositivo 2.

8. Avaliação de performance

8.1. Introdução

Pretendeu-se avaliar o funcionamento do dispositivo 2 durante um longo prazo de tempo, sem verificação periódica e numa situação controlada. Assim verificando a capacidade da placa de desenvolvimento em operar continuamente o programa desenvolvido, durante um período de tempo bastante lato sem intervenção humana.

Para realizar esta avaliação procurou-se um local com condições particular, que fosse isolado, possuísse pouco ruído provindo de atividade humana e que fosse um pavimento térreo o mais interior no solo possível, com é caso das caves.

O local encontrado foi a cave do laboratório de estruturas do departamento de engenharia civil (EC) da Escola Superior de Tecnologia e Gestão (ESTG) de Leiria. Escolheu-se o período de férias de verão, pois nesse todas atividades na ESTG foram reduzidas ao mínimo.

Programou-se o ensaio para inicial no dia 04 de agosto de 2017 e concluir no dia 28 de agosto de 2017.

8.2. Objetivos

O primeiro objetivo foi avaliar o comportamento a longo prazo do Dispositivo 2, verificando á posteriori falhas de funcionamento. O segundo objetivo foi avaliar a quantidade de registos feitos pelo dispositivo por dia, assim definido uma quantidade de registo por dia base. Um terceiro objetivo foi verificar quais desses se deveram a eventos sísmicos, assim verificando a sensibilidade do Dispositivo 2. Para obter dados de registos sismos válidos, acompanhou-se a atividade sísmica, durante o período de avaliação no portal do Instituto do Mar e da Atmosfera IPMA.

8.3. Definições do *Trigger action*

Antecipadamente á avaliação da performance, foram realizados ensaios preliminares para ajudar a definir os parâmetros de tecto individual e de conjunto, a usar durante a avaliação

de performance de modo que o dispositivo regista-se o menor número possível de eventos. O dispositivo foi aplicado num 3º piso de um prédio com 4 pisos, durante os períodos noturnos de menor atividade humana, em intervalos de 8 horas, numa série consecutiva de dias, com vários parâmetros diferentes, até que as configurações produzissem o menor número de resultados possível, procurando sempre a maior sensibilidade possível.

Nas experiências efetuadas nas condições referidas, obtiveram-se as seguintes configurações, para regra individual 1,10 e para a regra de conjunto 1,07 e a média de curto prazo STA de 3 segundos. Com estes parâmetros, nas condições referidas, potencialmente obter-se-iam 19 registos por dia.

Considerou-se que 19 registos por dias seria um número elevado de registos, para avaliar em pós processamento, assim para o ensaio de performance usou-se para a regra individual 1,12 e para a regra de conjunto 1,08 com a média de curto prazo de 3 segundos

8.4. Local de aplicação

O local de aplicação do Dispositivo 2 foi a junto à parede norte da cave do laboratório de estruturas. A localização do laboratório estruturas de engenharia civil da ESTG de Leiria situa-se nas seguintes coordenadas 39°44'5.30"N e 8°49'13.28"W.

O dispositivo foi fixo a uma viga metálica H com 10cm de altura e largura, e aproximadamente 1 metro de comprimento, por meio de uma série de ímanes de 4mm de diâmetro colados na base da caixa externa do Dispositivo 2. A viga serve de massa ao dispositivo, e foi aplicada sobre a laje da cave com ilustrado na imagem abaixo.

A orientação dos eixos X e Y com norte geográfico, situação que se refere aos azimutes, foram 329,5° e 59,5° respetivamente para os eixos X e Y.



Figura 8.1 - Localização do Dispositivo 2, cave no laboratório de estruturas EC da ESTG

8.5. Eventos sísmicos

Durante o período que decorreu este ensaio foram registados vários eventos sísmicos. A informação sobre estes eventos foi recolhida no *site* do Instituto Português do Mar e da Atmosfera [35], na sessão dedicada aos sismos e atividade sísmica.

Durante o período de ensaio foram registados vários eventos, sendo dois deles sentidos pela população local.

O quadro abaixo resume os registos sísmicos verificados num raio de 50km, em redor do laboratório de engenharia civil, com a exceção do sismo localizado em Sobral de Monte Agraço, porque este foi sentido pela população da região de Leiria.

Tabela 8.1 - Quadro resumo da atividade sísmica

| Data | Hora | Magnitude | localização geográfica do sismo | | | Distância entre epicentro e lab. Eng. Civil km | Azimute graus | Observações |
|------------|----------|-----------|---------------------------------|--------------|---------------------------------|--|---------------|-------------|
| | | Mercalli | | | | | | |
| 04/08/2017 | 15:54:23 | 1,8 | 40° 1'30.00"N | 8°38'56.40"W | Sobra l de Baixo - Soure | 35,4 | 24,33 | |
| 12/08/2017 | 00:32:46 | 0,8 | 39°37'30.00"N | 8°52'19.20"W | Casais dos Matos - Porto de Mós | 12,9 | 199,91 | |
| 17/08/2017 | 06:44:59 | 4,3 | 39° 1'33.60"N | 9° 7'15.60"W | Sobra l de Monte Agraço | 88,8 | 198,6 | Sentido |
| 19/08/2017 | 06:14:06 | 1,5 | 39°28'44.40"N | 9° 0'36.00"W | Vimeiro - Alcobaça | 32,7 | 209,67 | |
| 25/08/2017 | 14:01:44 | 2 | 39°22'55.20"N | 8°55'8.40"W | Casalinho - Rio Maior | 40 | 192,14 | |
| 27/08/2017 | 00:01:07 | 1,8 | 39°23'6.00"N | 9° 7'26.40"W | Gaieiras - Caldas da Rainha | 46,7 | 213,68 | |
| 28/08/2017 | 01:42:28 | 2,7 | 39°47'38.40"N | 8°50'20.40"W | Campos do Lis - Leiria | 6,7 | 346,43 | Sentido |

8.6. Resultados

Os resultados apresentados representam o número de ficheiros gravados pelo Dispositivo 2, durante o período de tempo, iniciado às 12:00:00 do dia 04 de agosto de 2017 e termino às 12:10:00.

Tabela 8.2 - Resultados obtidos do ensaio performance

| Data | | Número de eventos registados |
|------------|-----|------------------------------|
| 04/08/2017 | sex | 2 |
| 05/08/2017 | sáb | 3 |
| 06/08/2017 | dom | 3 |
| 07/08/2017 | seg | |
| 08/08/2017 | ter | 1 |
| 09/08/2017 | qua | 1 |
| 10/08/2017 | qui | 3 |
| 11/08/2017 | sex | 2 |
| 12/08/2017 | sáb | 1 |
| 13/08/2017 | dom | 1 |
| 14/08/2017 | seg | 2 |
| 15/08/2017 | ter | 4 |
| 16/08/2017 | qua | 6 |
| 17/08/2017 | qui | 4 |
| 18/08/2017 | sex | 58 |

| Data | | Número de eventos registados |
|------------|-----|------------------------------|
| 19/08/2017 | sáb | |
| 20/08/2017 | dom | 3 |
| 21/08/2017 | seg | |
| 22/08/2017 | ter | 2 |
| 23/08/2017 | qua | 4 |
| 24/08/2017 | qui | |
| 25/08/2017 | sex | 2 |
| 26/08/2017 | sáb | 2 |
| 27/08/2017 | dom | 1 |
| 28/08/2017 | seg | 1 |

Total de eventos 91
 Eventos por dia (*) 2

(*) Não se contabilizou o dia 18 na determinação da média

8.7. Conclusões do ensaio de performance

Ao analisar os resultados produzidos pelo ensaio de performance, o dispositivo 2 trabalhou corretamente sem falhas durante todo o período, avaliando assim o ensaio um sucesso. Analisando os registos obtido, por dia durante o período de ensaio, verificou-se que houve em média dois registos por dia, considerando uma situação normal e esperada. Verificou-se também que no dia 18 houve uma quantidade elevada de registos. Foi inquirido aos responsáveis pelos laboratórios do departamento Engenharia Civil (EC) acerca da atividade realizada nos laboratórios e neste dia em particular, decorrem vários ensaios no laboratório de geotecnia da EC, localizado ao lado do laboratório de estruturas, um piso acima.

Relativamente aos registos de atividade sísmica, procurou-se pelos registos no cartão de memória SD do Dispositivo 2, pelo dia e hora dos sismos registados e verificou-se que não ficaram registados.

Não é possível concluir pela razão do algoritmo não ter acudado. Entre várias hipóteses, as três mais prováveis são a falta de sensibilidade do sensor, a frequência do sismo, ou os parâmetros de acionamento da ação do algoritmo. Uma solução para discernir qual das hipóteses é a correta é capturar um evento sísmico. Para realizar esta tarefa é necessário registar continuamente vibrações até que um evento ocorra. Após a recolha de registos de um evento ter ocorrido, devesse verificar se o sensor tem capacidade de o identificar. Caso afirmativo procedesse á simulação do algoritmo com várias parametrizações diferentes. Para que tudo isto seja possível ser realizado será necessário desenvolver um programa específico para esta função, que não está no âmbito deste trabalho.

9. Caso prático – Mosteiro da Batalha

9.1. Introdução

Concluído todo o processo de desenvolvimento, decidiu-se aplicar o dispositivo 2 no Mosteiro da Batalha com intuito de obter dados de vibrações deste espaço emblemático.

A escolha do local exato dentro do Mosteiro da Batalha, teve como a única condicionante a existência de uma fonte elétrica externa em corrente AC 220V. Assim o local escolhido foi antigo refeitório do mosteiro, agora convertido em loja de venda de acesso ao público que visita o monumento.

A abaixo é apresentada a planta do Mosteiro da Batalha, onde é definida a localização do antigo refeitório pelo número 6, também assinado com um ponto vermelho está indicada a localização onde foi aplicado o dispositivo 2.

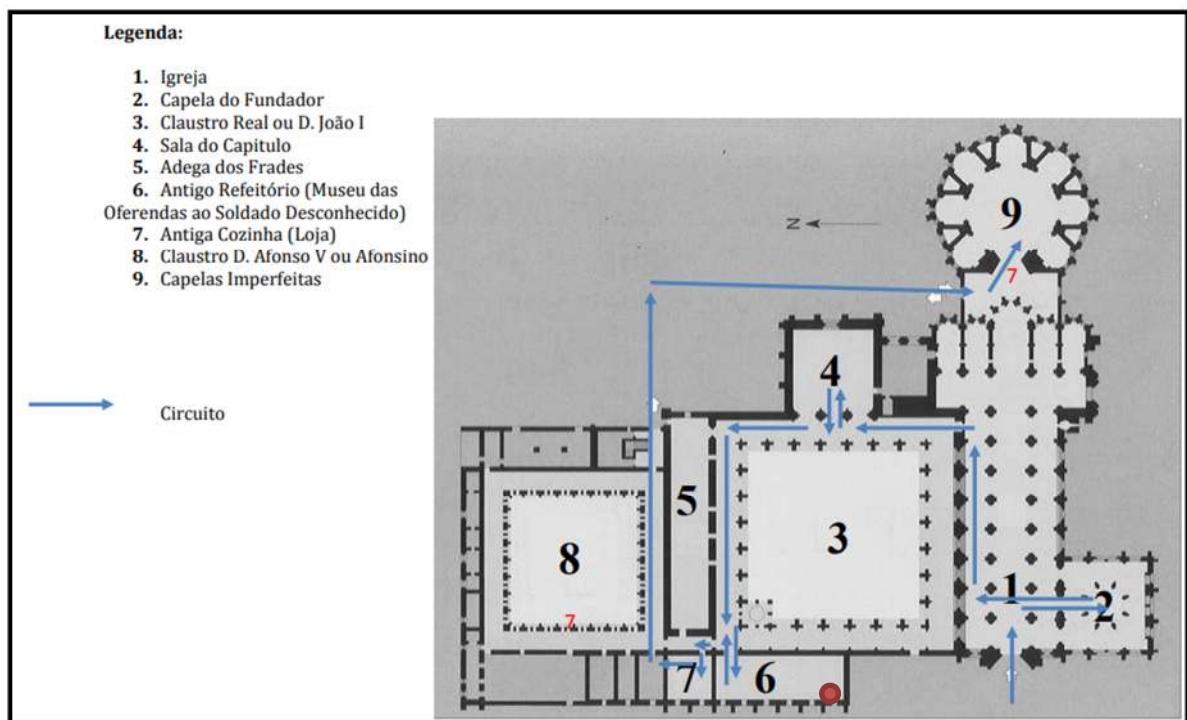


Figura 9.1 - Planta do Mosteiro da Batalha [36]do Mosteiro da Batalha [36]

9.2. Desenvolvimento dos trabalhos

O período escolhido para fazer a monitorização do Mosteiro da Batalha foi entre 13 a 23 de setembro de 2017.

O sítio de aplicação do dispositivo 2 foi junto à parede longitudinal oeste, por de trás de um biombo, que ocultou o dispositivo da vista do público, sensivelmente a 1 metro do corredor de passagem pedonal.

O dispositivo foi aplicado no pavimento térreo, com o eixo X orientado paralelamente com a parede longitudinal, com sentido positivo virado para sul. O eixo Y ficou perpendicular à parede longitudinal, com sentido positivo apontado para este.

Iniciou-se o dispositivo com novos parâmetros, procurando torna-lo mais sensível a deteção de vibrações. Assim os valores dos parâmetros individuais no eixo X foi 1,16 e eixo Y foi 1,17, para a regra de conjunto foi 1,125 e tempo para media de curta duração 1,5 segundos. Ao fim de três dias voltou-se ao local para proceder a uma quantificação de registos, assim averiguado se os parâmetros usados estavam ajustados as condições do local.

Verificou-se que, durante o tempo que o aparelho esteve ligado, este registou 574 eventos, em média 190 por dia. Esta quantidade de resultados é muito superior ao inicialmente esperado, e de difícil pós processamento. Assim ajustou os parâmetros dos tectos individuais para 1,12 em ambos os eixos, de conjunto para 1,12 com um tempo da média de curto prazo de 3 segundos. Com este aumento dos parâmetros esperou-se filtrar mais ruído de vibração ambiental.

Após a alteração dos parâmetros do programa, reiniciou-se o dispositivo no mesmo local e nas mesmas condições anteriores. Passados 8 dias retornou-se ao local, para recolher os dados da monitorização de vibrações ambientais.

9.3. Resultados

Foram registados durante o tempo de funcionamento 21 registos, 2 dos quais são referentes ao manuseamento do dispositivo, a quantidade de dados é aceitável para proceder a uma avaliação.

Tabela 9.1 - Resultados obtidos da monitorização do Mosteiro da Batalha

| Data | | Número de eventos registados |
|------------|-----|------------------------------|
| 16/09/2007 | sáb | 1 |
| 17/09/2007 | dom | 5 |
| 18/09/2007 | seg | 1 |
| 19/09/2007 | ter | |
| 20/09/2007 | qua | 2 |
| 21/09/2007 | qui | 3 |
| 22/09/2007 | sex | 7 |
| 23/09/2007 | sáb | |

Analisaram-se os dados e obtiveram-se resultados das frequências e amplitudes de Fourier. Foram verificados seis registos de eventos significantes e outros trezes que aparentam ser ruído.

Foram processados os seis registos significativos, obtendo dados das frequências e suas amplitudes. Abaixo é apresentado o pós processamento de um dos seis registos, neste caso, o evento mais forte que ocorreu no dia 17 de setembro de 2017.

Gráfico 9.1 - Tempo/aceleração de evento a 17/09/2019 08:51, Mosteiro da Batalha eixo X

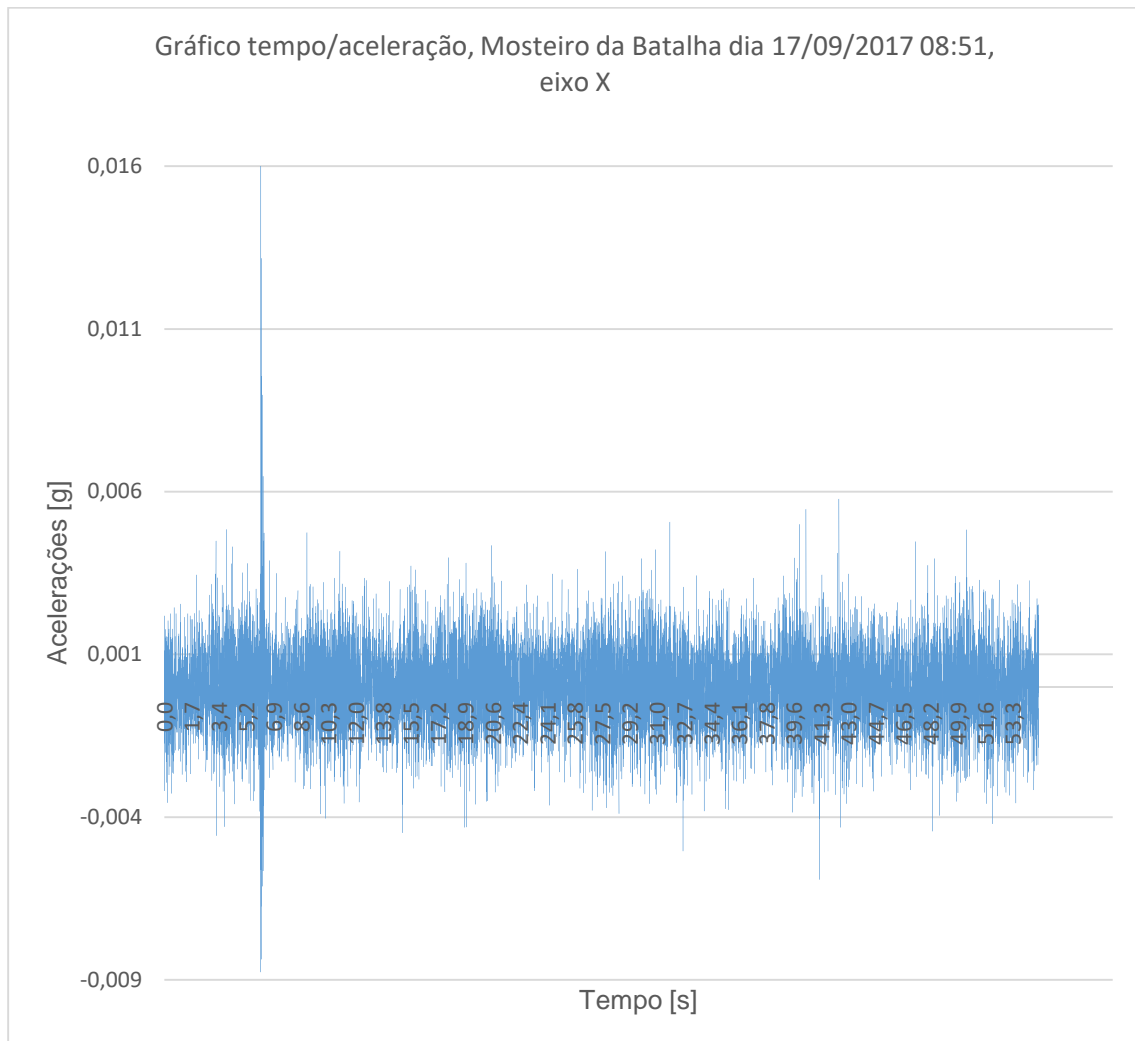


Gráfico 9.2 - Tempo/aceleração de evento a 17/09/2019 08:51, Mosteiro da Batalha eixo Y

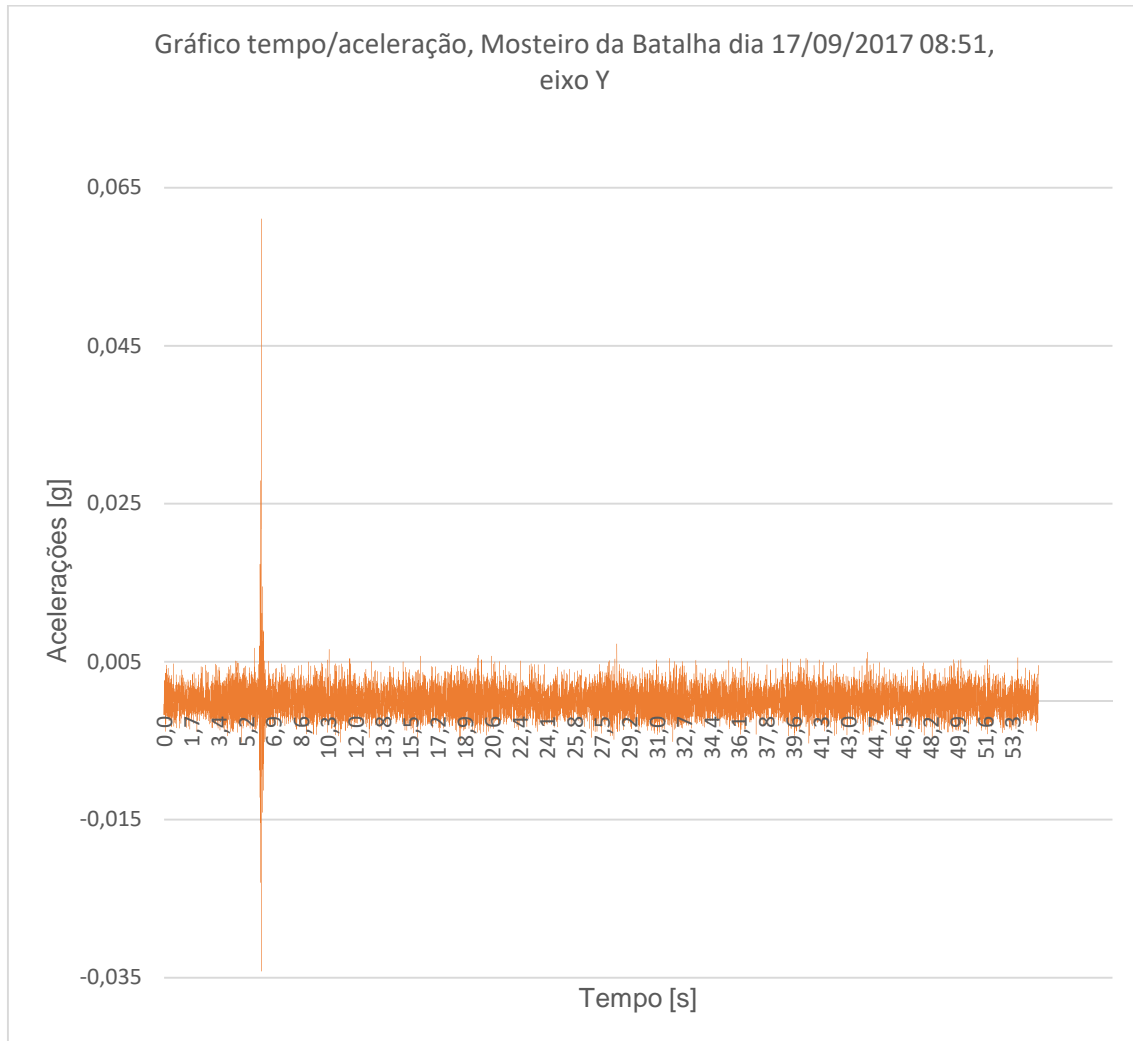


Gráfico 9.3 - Espectro de freq. eixo X, Mosteiro da Batalha evento a 17/08/2017 08:51

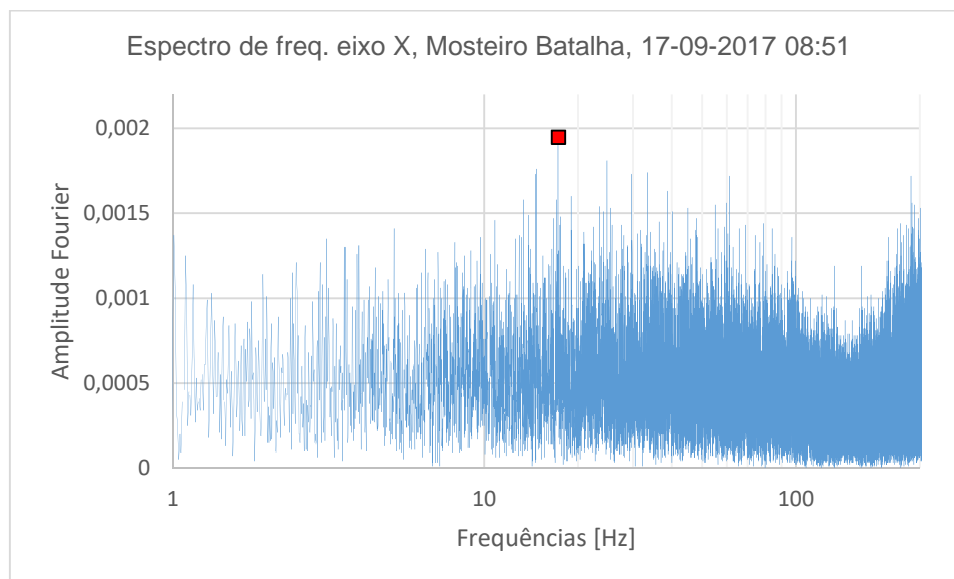
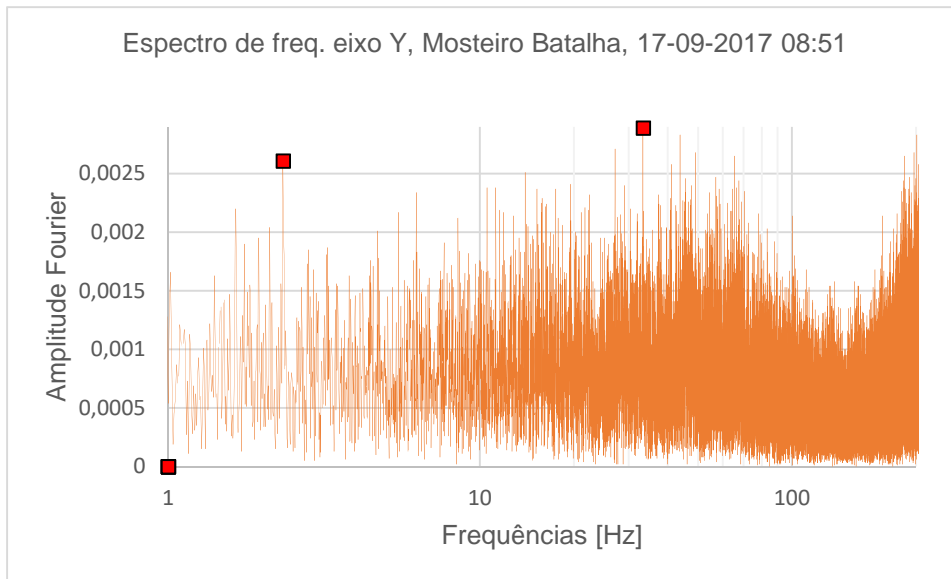


Gráfico 9.4 - Espectro de freq. eixo Y, Mosteiro da Batalha evento a 17/08/2017 08:51**Tabela 9.2 - Resultados obtidos da monitorização do Mosteiro da Batalha, por dia**

| Dispositivo 2 - eixo X | | |
|------------------------|-------------|--------------|
| Ordem | f [Hz] | Amp. Fourier |
| 1 | 17,28 | 0,00195 |

| Dispositivo 2 - eixo Y | | |
|------------------------|-------------|--------------|
| Ordem | f [Hz] | Amp. Fourier |
| 1 | 33,33 | 0,00289 |
| 7 | 2,34 | 0,00261 |

Dos seis registos comparam-se os espectros de frequências, identificando aquelas com maiores amplitudes, verificou-se que não houve correlações entre os registos.

Verificou-se a aceleração máxima de pico foi de 0,017g no eixo X, 0,061g em Y. Os valores mínimos de pico foram de -0,009g no eixo X e -0,034g em Y. As frequências de máxima amplitude de Fourier encontradas foram 17,28Hz no eixo X e 33,32Hz no eixo Y.

10. Conclusões ou Conclusão

Sobre o desenvolvimento do dispositivo de monitorizações de vibrações debaixo custo, conclui-se é que é possível projetar, desenvolver, e construir um dispositivo, testar, com materiais e recursos disponíveis na rede global de informação, sem que à partida, se tenha um conhecimento profundo e especializado sobre o assunto. Uma realidade que era muito pouco provável á 15 anos atrás.

A tecnologia desenvolvida pelo sistema *open-source* da Arduino, atingiu um patamar de desenvolvimento bastante considerável. Durante o desenvolvimento das componentes *hardware* e *software*, todas as dificuldades que surgiram durante esse processo foram ultrapassas com ajuda e contribuições de outros entusiastas da área que foi desenvolvida. Também foi surpreendente a transição sem sobre saltos entre placas de desenvolvimento e componentes, mostrando o bom desenvolvimento das bibliotecas.

Os resultados obtidos pelo sensor MEMS de baixo custo, utilizados e existentes neste momento não se podem comparar aos sensores de nível profissional de alta precisão, que utilizam uma tecnologia já bem amadurecida. Mas estes sensores MEMS, apresentam um bom nível de confiança, tendo em conta o baixo preço. É possível imaginar o desenvolvimento de rede de monitorização numa estrutura, num sistema a recolher dados em simultâneo e interligados, com um valor de projeto muito mais acessível do que se empregassem sensores de alta precisão.

Num futuro que avança para *internet of thing* IoT e para *internet of everything* IoE, são necessários sensores com maior qualidade, a baixo preço, para satisfazer a necessidade de estabelecer a ponte entre objetos, estruturas, etc, á rede de informação. Espera-se então que num futuro próximo que a tecnologia MEMS, entre outras tecnologias de baixo custo, atinjam o patamar dos sensores de alta precisão.

Referências Bibliográficas

[1] Bisby, L.A.; Briglio, M.B. - ISIS Educational Module 5: An Introduction to Structural Health Monitoring, Department of Civil Engineering - Queen's University, Agosto 2004.

[2] Farrar, Charles R; Worden, Keith – Philosophical Transactions of The Royal Society A (2007) 365, 303-315: An introduction to structural health monitoring, 15 fevereiro 2007.

[3] Franden, Jacob - Handbook of Modern Sensors: Physics, Designs, and Applications Fifth Edition”, Springer, 2016, Disponível na WWW:
<https://www.ordemengenheiros.pt/fotos/dossier_artigo/chedassampaio_20150603_18561814795575a4962b3f1.pdf>

[4] Sampaio, Chedas - Medição de vibrações em equipamentos e/ou estruturas usando sensores low cost”, maio 2015.

[5] Marques, Leandro Nunes Martins Vieira - Tese de Mestrado: Monitorização Estática e Dinâmica: Aplicações, Universidade do Minho, dezembro de 2007.

[6] Aktan, A. Emin; Catbas, F. Necati; Grimmelsman, Kirk A.; Pervizpour, Mesut - Report: Development of a Model Health Monitoring Guide for Major Bridges, Drexel Intelligent Infrastructure and Transportation Safety Institute, julho 2003

[7] HBM - White Paper: Optical Fiber Sensors vs. Conventional Electrical Strain Gauges for Infrastructure Monitoring Applications, Disponível na WWW:
<<https://www.hbm.com/pt/6482/sensores-opticos-vs-strain-gauges-eletricos-convencionais/>>

[8] Azzler, Craig - Acceleration, Shock and Vibration Sensors, PCB Piezotronics, Inc

[10] Tianliang Li; Yuegang Tan and Zude Zhou: “A Fiber Bragg Grating Sensing-Based Micro-Vibration Sensor and Its Application”, School of Mechanical and Electronic Engineering, Wuhan University of Technology, abril 2016

- [11] ST Microelectronics - Product Technologies - MEMS: Micro-Electro-Mechanical System, Disponível na WWW: <https://www.st.com/content/st_com/en/about/innovation---technology/mems.html> 2019
- [12] Cigada, Alfredo; Lurati, Massimiliano; Redaelli, Matteo, Vanali, Marcello - Mechanical Performance and Metrological Characterization of MEMS Accelerometers and Application in Modal Analysis, Politecnico di Milano, Dipartimento di Ingegneria Meccanica, Milano, Italy, janeiro 2007
- [13] i-micronews – Product report: Sensors for mobile phones and tablets 2016
- [14] Kavitha,S. ^a; Daniela, R. Joseph; Sumangala, K. ^b - Design and Analysis of MEMS Comb Drive Capacitive Accelerometer for SHM and Seismic Applications, ^a NPMaSS National MEMS Design Centre, ^b Department of Electronics and Instrumentation Engineering, Department of Civil and Structural Engineering, Annamalai University, Annamalainagar, Chidambaram, Tamil Nadu, India, 2016
- [15] Octopart, Disponível na WWW: <<https://octopart.com/>>
- [16] Digikey, Disponível na WWW: <<https://www.digikey.com/>>
- [17] Krishnan, Girish; Kshirsagar, Chaitanya U.; Ananthasuresh, G. K.; Navakanta Bhat, Journal of Indian Institute of Science: “Micromachined High-Resolution Accelerometers” setembro 2007
- [18] Santos, Antônio Fernando Moura, Dissertação de Pós-graduação em Engenharia Mecânicas - Caracterização de um Sistema de Medição de Vibrações de Baixo Custo para Aplicações Gerais, Universidade Federal de Uberlândia, 2011
- [19] NXP Freescale Semiconductor, AN4075 Rev 1, 09/2010 - How Many Bits are Enough?: The Trade-off Between High Resolution and Low Power Using Oversampling Modes
- [20] Cemer, Izudin - Noise Measurement, Disponível na WWW: <<https://www.fierceelectronics.com/embedded/noise-measurement>>, fevereiro 2011

[21] Analog Devices - Accelerometer Specifications: Quick Definitions, Disponível na WWW: <<https://www.analog.com/en/products/landing-pages/001/accelerometer-specifications-definitions.html#>>

[22] Github, Disponível na WWW: <<https://github.com/>>

[23] Bell, Charles A. - Beginning Sensor Networks with Arduino and Raspberry Pi, 2013

[24] Arduino, Disponível na WWW: <<https://www.arduino.cc/>>

[25] Sparkfun, Disponível na WWW: <<https://www.sparkfun.com/>>

[26] DFRobot, Disponível na WWW: <https://www.dfrobot.com/>

[27] Gammon, Nick, artigo: "I2C - Two-Wire Peripheral Interface - for Arduino", Disponível na WWW: <<http://www.gammon.com.au/i2c>>

[28] Stack Exchange - maximum I2C BUS length?, Disponível na WWW: <<https://electronics.stackexchange.com/questions/106265/maximum-i2c-bus-length>>

[29] Dagra - Linear Interpolation with Excel, Disponível na WWW: <<http://www.datadigitization.com/dagra-in-action/linear-interpolation-with-excel/>>

[30] Adafruit, Disponível na WWW: <<https://www.adafruit.com/>>

[31] Trnkoczy, Amadej - Understanding and parameter setting of STA/LTA trigger algorithm, setembro 1999

[32] PCB Piezotronic - specification model 352C03, Disponível na WWW: <<https://www.pcb.com/products?m=352C03>>

[33] National Instruments, product specification NI-9234, Disponível na WWW: <<http://www.ni.com/pt-pt/support/model.ni-9234.html>>

[34] National Instruments, product specification LabVIEW, Disponível na WWW: <<http://www.ni.com/pt-pt/shop/labview.html>>

[35] Instituto Português do Mar e da Atmosfera, Disponível na WWW:

<<https://www.ipma.pt>>

[36] Mosteiro da Batalha, Planta, Disponível na WWW

<<http://www.mosteirobatalha.gov.pt/data/planta%20do%20mosteiro%20sem%20adega.pdf>>

Anexos

ANEXO A1 – Instruções de montagem dispositivo 1

Instruções de montagem do dispositivo aquisição de vibrações Dispositivo 1

1. Introdução

Este documento são indicadas instruções que auxiliam a montagem do Dispositivo 1, com recurso a uma lista de componentes e figuras ilustrativas.

É apresentada uma lista de componentes usados para construir o Dispositivo 1, e um orçamento com discriminação de preços parciais e totais. Os preços apresentados apenas referem o valor de aquisição dos componentes, incluem o valor do imposto no local de expedição do componente, mas não incluem o valor do transporte, montagem ou lucro.

Na lista de componentes, cada material e componente está referenciando com um índice, indicado na coluna da esquerda da lista. Os componentes apresentados nas figuras, também têm um índice que é o mesmo indicado na lista de componentes, assim melhorando a compreensão da montagem do Dispositivo 1.

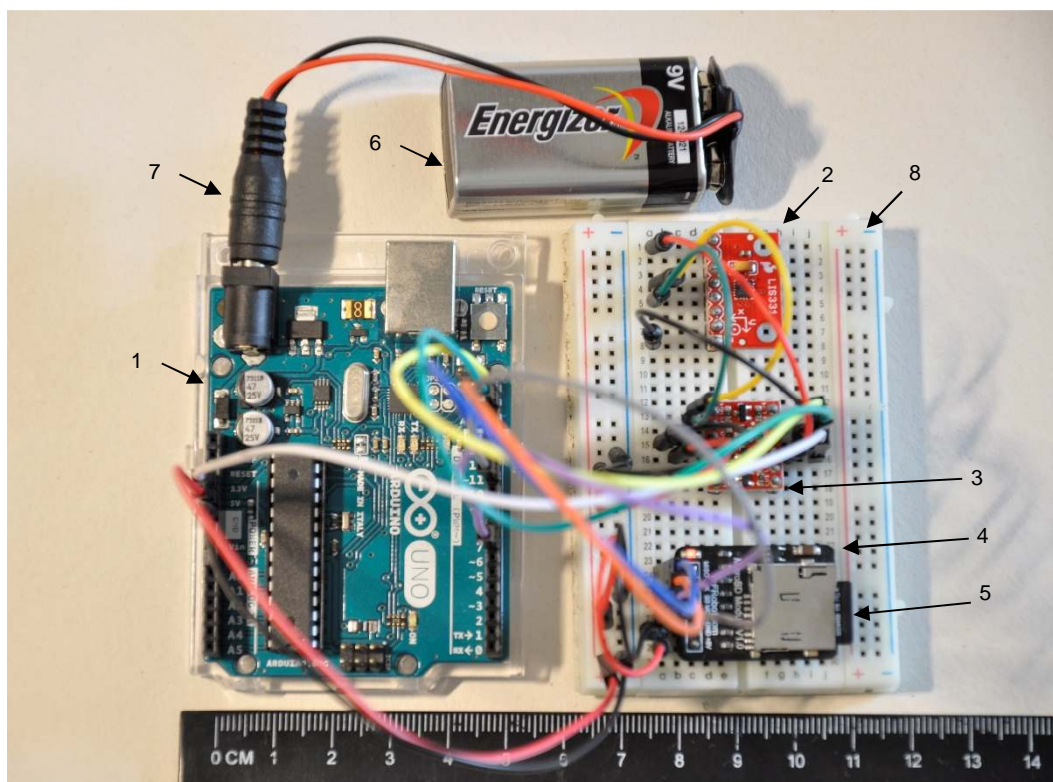
2. Lista de componentes

| Índice | Componente | Descrição do componente | Fornecedor | País de origem |
|--------|-------------------------|---|---------------------------|----------------|
| 1 | Microcontrolador | Arduino UNO R3 | Sonigate | Portugal |
| 2 | Suporte | Breadboard 400 pontos | Ebay - xiumeche-0 | China |
| 4 | Datalogger | Módulo para cartões micro SD | DFRobot | China |
| 5 | Cartão de memória | Cartão de memória 4Gb classe 4 | PTRobotics | Portugal |
| 3 | Componente eletrónico | Logic Level Converter Bi-Directional | PTRobotics | Portugal |
| 2 | Sensor | Breakout DFRobot - Sensor Inercial MPU6050 | DFRobot | China |
| 2 | Sensor | Breakout GY 45 - Sensor Inercial MMA8451Q | Ebay - czb6721960 | China |
| 2 | Sensor | Breakout Sparkfun - Sensor Inercial LIS331HH | RoboSavvy | UK |
| | Jumper | Jumper MM 10cm | PTRobotics | Portugal |
| | Jumper | Jumper MM 20cm | Ebay - satisfyelectronics | China |
| 6 | Fornecimento de energia | Pilha alcalina 9V | | |
| 7 | Conexões | Cabo de conexão entre a pilha de 9V e barrel jack | Ebay - xiumeche-0 | China |
| | Conexões | Cabo de ligação USB Type A/B | Pcdiga - Leiria | Portugal |

3. Orçamento

| Índice | Componente | Descrição do componente | Quantidades un | Valor unitário €/un | valor parcial € |
|--------|-------------------------|---|----------------|---------------------|-----------------|
| 1 | Microcontrolador | Arduino UNO R3 | 1 | 24,60 | 24,60 |
| 2 | Suporte | Breadboard 400 pontos | 1 | 1,69 | 1,69 |
| 4 | Datalogger | Módulo para cartões micro SD | 1 | 5,35 | 5,35 |
| 5 | Cartão de memória | Cartão de memória 4Gb classe 4 | 1 | 5,80 | 5,80 |
| 3 | Componente eletrónico | Logic Level Converter Bi-Directional | 1 | 2,83 | 2,83 |
| 2 | Sensor | Breakout DFRobot - Sensor Inercial MPU6050 | 1 | 9 | 9 |
| 2 | Sensor | Breakout GY 45 - Sensor Inercial MMA8451Q | 1 | 2,14 | 2,14 |
| 2 | Sensor | Breakout Sparkfun - Sensor Inercial LIS331HH | 1 | 8,54 | 8,54 |
| | Jumper | Jumper MM 10cm | 8 | 0,02 | 0,20 |
| | Jumper | Jumper MM 20cm | 9 | 0,00 | 0,03 |
| 6 | Fornecimento de energia | Pilha alcalina 9V | 1 | 2,29 | 2,29 |
| 7 | Conexões | Cabo de conexão entre a pilha de 9V e barrel jack | 1 | 0,44 | 0,44 |
| | Conexões | Cabo de ligação USB Type A/B | 1 | 1,90 | 1,90 |
| | | | | Total | 64,81 |

4. Instruções de montagem para o Dispositivo 1

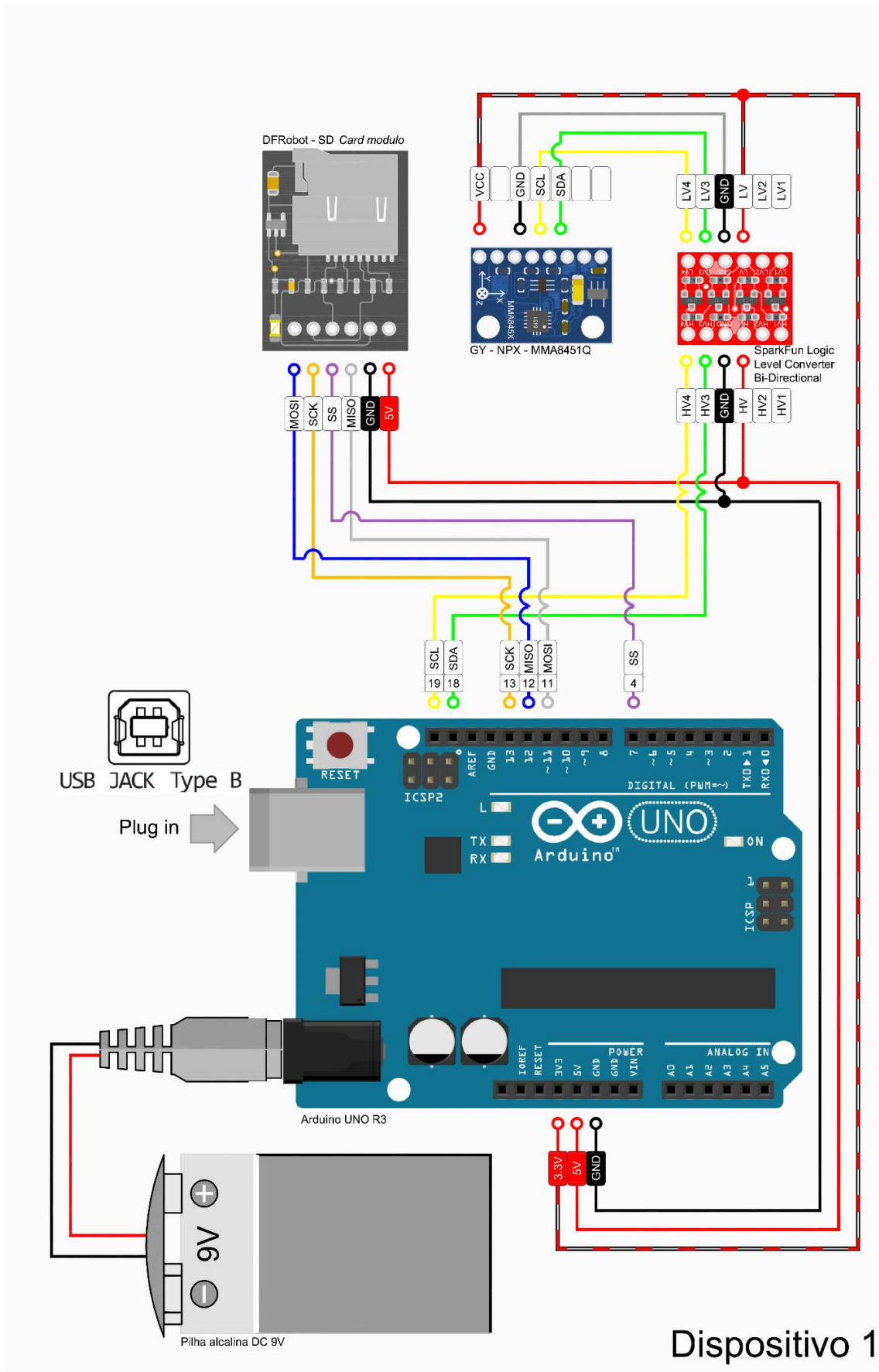


1. Placa de desenvolvimento Arduino UNO R3
2. Sensor inercial MEMS
3. *Logic level converter bi-directional*
4. Módulo de armazenamento para cartões micro SC
5. Cartão micro SD
6. Pilha alcalina de corrente contínua 9V
7. Jack 3.5
8. *Breadboard 400 pin's*

Os módulos 4, 3 e 2, são instalados na *breadboard* conforme indicado na figura mostrada acima.

As ligações entre os vários componentes deve obedecer a seguinte configuração indicada na figura no ponto 5 desta instrução de montagem.

5. Esquema de ligações para o Dispositivo 1



Dispositivo 1

ANEXO A2 - Instruções de montagem dispositivo 2

Instruções de Montagem do Dispositivo Monitorização de Vibrações – Dispositivo 2

1. Introdução

Neste documento são documentadas as instruções, sobre o processo de montagem do Dispositivo 2, passo a passo, com recurso a uma lista de componentes e uma sequência de figuras de todo o processo.

É apresentada uma lista de componentes, usados para construir o Dispositivo 2, um orçamento com discriminação de preços parciais e totais. Os preços apresentados apenas referem o valor de aquisição dos componentes, incluem o valor do imposto no local de expedição do componente, mas não incluem o valor do transporte, montagem ou lucro.

Na lista de componentes, cada material e componente está referenciado com um índice, indicado na coluna da esquerda da lista. Os componentes apresentados nas figuras, também têm um índice que é o mesmo indicado na lista de componentes, assim melhorando a compreensão da montagem do Dispositivo 2.

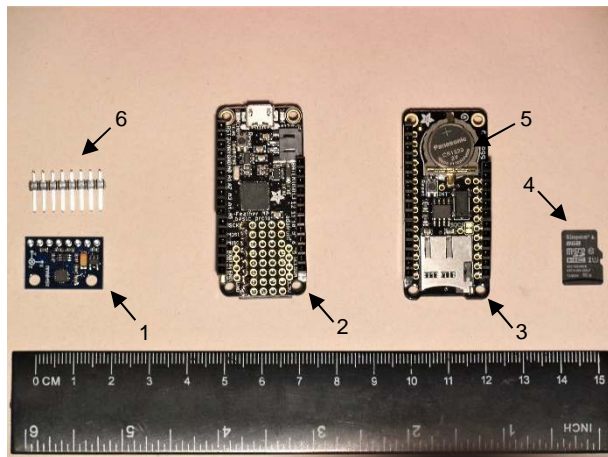
2. Lista de componentes

| Índice | Componente | Descrição do componente | Fornecedor | Pais de origem |
|-------------|-------------------------|--|---------------------------------|----------------|
| | Acessórios | Stackable Header Pins 6 Pins for Arduino Shield UNO MEGA DUE 2.54mm Pitch | Ebay - czb6721960 | China |
| 35 | Acessórios | Stackable Header Pins 10 Pins for Arduino Shield UNO MEGA DUE 2.54mm Pitch | Ebay - czb6721960 | China |
| 7, 8, 9, 15 | Acessórios | 120PCS M2 Nylon Hex Spacers Screw Nut Assortment Kit Stand off Accessories Set | Ebay - cskwin2015 | China |
| 5 | Baterias | CR1220 Coin Cell Battery 3V 35mA | E.S. - Electronic Shop | Luxemburgo |
| 30 | Baterias | Lithium Ion Polymer Battery - 3.7v 500mAh | E.S. - Electronic Shop | Luxemburgo |
| 32 | Cabos | 25cm USB 2.0 A Male to Micro USB 5 pin Male Data Charge Cable Cord | Ebay - czb6721960 | China |
| 18, 19 | Caixa externa | Enclosure - Flanged (Red) - 74x158x90mm | E.S. - Electronic Shop | Luxemburgo |
| 4 | Cartão de memória | Transcend 8Gb micro SD HC class 4 | Alibaba - Impressão e Papelaria | Portugal |
| 36 | Componente eletrónico | Resistência 4,7 Ohm | Sonigate | Portugal |
| 14 | Conexões | Barra de junção 10mm2 branca | AKI - Leira | Portugal |
| 22 | Conexões | Super Strong Round Disc 5x2mm Magnets Rare-Earth Neodymium N35 | Ebay - huanshop2015 | Sigapura |
| 23 | Conexões | Screws 2,7mm x 12mm | | |
| 25, 26 | Conexões | PG7 Black Plastic Waterproof Connector Gland 3-6.5mm Dia Cable | Ebay - kingelectronics15 | China |
| 24 | Conexões | Junta de neoprene 12x18x2 | AKI - Leira | Portugal |
| 3 | Datalogger + RTC | Adalogger FeatherWing - RTC + SD Add-on For All Feather Boards | E.S. - Electronic Shop | Luxemburgo |
| 13 | Fornecimento de energia | Carregador Samsung USB 5W 1A branco | Podiga - Leiria | Portugal |
| 27 | Fornecimento de energia | Cabo electrico com ficha eletrica 1m (reciclado) | | |
| 31 | Interruptor | Interruptor unipolar 2A | AKI - Leira | Portugal |
| 33 | Jumper | Dupont Wire Male to Female M/F Jumper Cable 20cm | Ebay - satisfyelectronics | China |
| 34 | Jumper | Dupont Wire Female to Female F/F Jumper Cable cm | Ebay - satisfyelectronics | China |
| 2 | Microcontrolador | Adafruit Feather M0 Basic Proto - ATSAMD21 Cortex M0 | E.S. - Electronic Shop | Luxemburgo |
| 1 | Sensor | Breakout GY 45 - Sensor Inercial MMA8451Q | Ebay - czb6721960 | China |
| 10, 11, 12 | Suporte e fixação | Placa de acrilico de suporte com furações 16x2mm 8x3mm | Plexicril | Portugal |
| 28 | Termómetro | Waterproof Digital Temperature Temp Sensor Probe DS18B20 | Ebay - czb6721960 | China |
| 29 | Termómetro | DALLAS DS18B20 18B20 TO-92 Thermometer Temperature Sensor | Ebay - czb6721960 | China |

3. Orçamento

| Índice | Componente | Descrição do componente | Quantidades un | Valor unitário €/un | valor parcial € |
|--------------|-------------------------|--|----------------|---------------------|-----------------|
| | Acessórios | Stackable Header Pins 6 Pins for Arduino Shield UNO MEGA DUE 2.54mm Pitch | 6 | 0,09 | 0,54 |
| 35 | Acessórios | Stackable Header Pins 10 Pins for Arduino Shield UNO MEGA DUE 2.54mm Pitch | 5 | 0,15 | 0,75 |
| 7, 8, 9, 15 | Acessórios | 120PCS M2 Nylon Hex Spacers Screw Nut Assortment Kit Stand off Accessories Set | 1 | 3,42 | 3,42 |
| 5 | Baterias | CR1220 Coin Cell Battery 3V 35mA | 1 | 1 | 1,00 |
| 30 | Baterias | Lithium Ion Polymer Battery - 3.7v 500mAh | 1 | 14,8 | 14,80 |
| 32 | Cabos | 25cm USB 2.0 A Male to Micro USB 5 pin Male Data Charge Cable Cord | 1 | 0,94 | 0,94 |
| 18, 19 | Caixa externa | Enclosure - Flanged (Red) - 74x158x90mm | 1 | 15 | 15,00 |
| 4 | Cartão de memória | Transcend 8Gb micro SD HC class 4 | 1 | 5,8 | 5,80 |
| 36 | Componente eletrónico | Resistência 4,7 Ohm | 1 | 0,08 | 0,08 |
| 14 | Conexões | Barra de junção 10mm2 branca | 2 | 0,1 | 0,20 |
| 22 | Conexões | Super Strong Round Disc 5x2mm Magnets Rare-Earth Neodymium N35 | 26 | 0,03 | 0,78 |
| 23 | Conexões | Screws 2,7mm x 12mm | 8 | 0,01 | 0,08 |
| 25, 26 | Conexões | PG7 Black Plastic Waterproof Connector Gland 3-6.5mm Dia Cable | 2 | 0,19 | 0,38 |
| 24 | Conexões | Junta de neoprene 12x18x2 | 2 | 0,09 | 0,18 |
| 3 | Datalogger + RTC | Adalogger FeatherWing - RTC + SD Add-on For All Feather Boards | 1 | 10 | 10,00 |
| 13 | Fornecimento de energia | Carregador Samsung USB 5W 1A branco | 1 | 3,9 | 3,90 |
| 27 | Fornecimento de energia | Cabo electrico com ficha eletrica 1m (reciclado) | 1 | 0 | 0,00 |
| 31 | Interruptor | Interruptor unipolar 2A | 1 | 1,49 | 1,49 |
| 33 | Jumper | Dupont Wire Male to Female M/F Jumper Cable 20cm | 4 | 0,0258 | 0,10 |
| 34 | Jumper | Dupont Wire Female to Female F/F Jumper Cable cm | 6 | 0,0258 | 0,15 |
| 2 | Microcontrolador | Adafruit Feather M0 Basic Proto - ATSAMD21 Cortex M0 | 1 | 22,8 | 22,80 |
| 1 | Sensor | Breakout GY 45 - Sensor Inercial MMA8451Q | 1 | 2,20 | 2,20 |
| 10, 11, 12 | Suporte e fixação | Placa de acrilico de suporte com furações 16x2mm 8x3mm | 1 | 10,71 | 10,71 |
| 28 | Termómetro | Waterproof Digital Temperature Temp Sensor Probe DS18B20 | 1 | 1,96 | 1,96 |
| 29 | Termómetro | DALLAS DS18B20 18B20 TO-92 Thermometer Temperature Sensor | 1 | 0,67 | 0,67 |
| Total | | | | 97,27 | |

4. Instruções de montagem para o Dispositivo 2

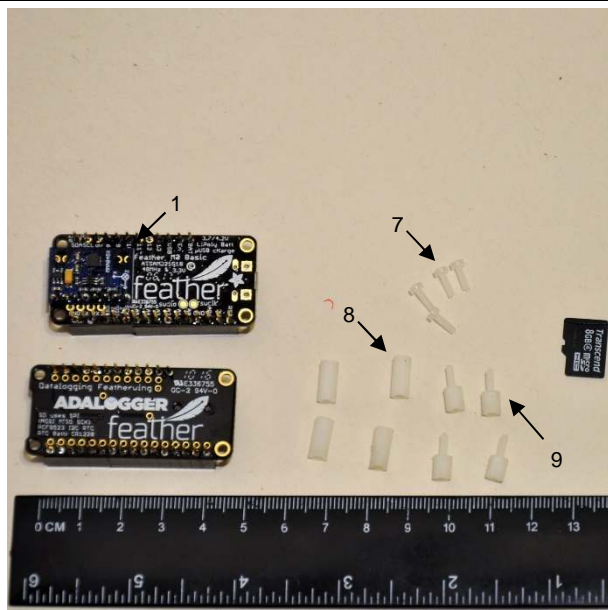


Materiais

- 1 – Sensor MMA8451Q
- 2 – Adafuit Feather M0 Proto
- 3 – Adafuit Adalogger+RTC PCF8523
- 4 – Cartão Micro SD 8Gb
- 5 – Pilha CR1220 3V
- 6 – 8x Pin

Tarefa (1)

Soldar com solda de estanho e chumbo, o Sensor na parte inversa da área proto da placa Adafuit M0, usando os pins com ponte de ligação. Aplicar pilha CR1220 na placa

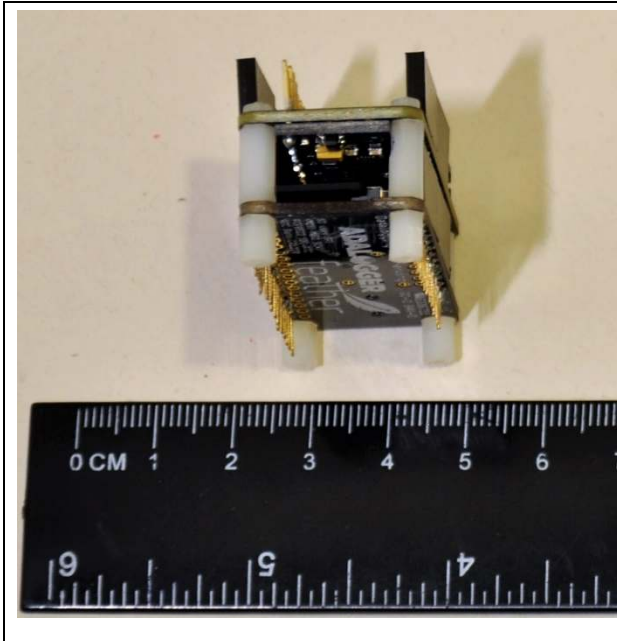


Materiais

- 1 – Sensor soldado na parte inversa
- 7 – Parafusos 6mm nylon M2
- 8 – Espaçador 10mm nylon M2 F/F
- 9 – Espaçador 6mm nylon M2 M/F

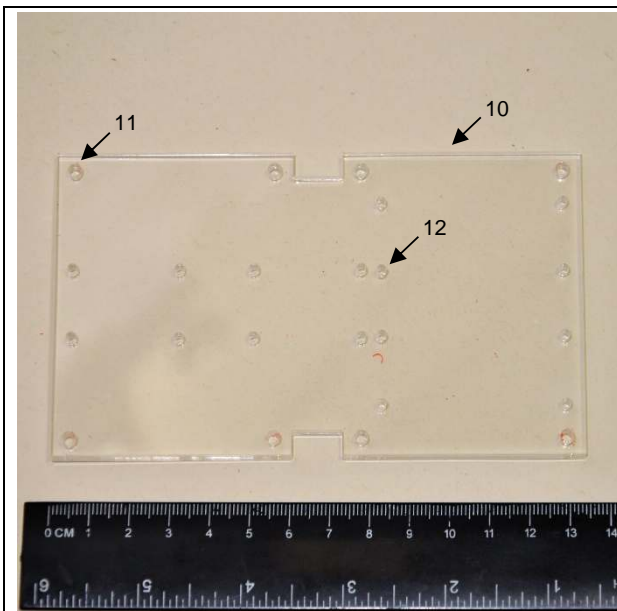
Tarefa (2)

Aplicar cartão micro SD no módulo micro SD.
Empilhar a placa sobre o escudo.
Aplicar espaçadores (8) entre placas.
Aplicar parafusos (7) no topo da placa, fixando parafusos e espaçadores.
Aplicar espaçadores (9) na parte inferior da pilha.



Observações

Perspetiva da Placa de desenvolvimento Adafruit Feather M0 Proto, empilhada sobre o escudo Adafruit Feather Adalogger, fixa por espaçadores e parafusos.



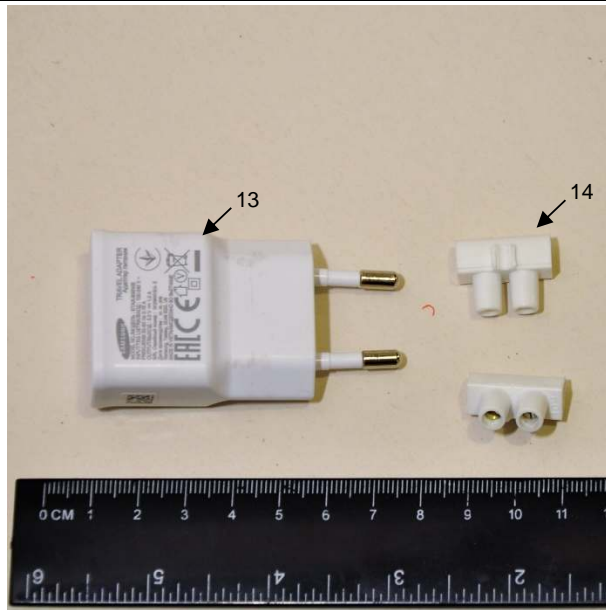
Materiais

- 10 – Placa de acrílico com 5mm
- 11 – 8x Furações ϕ 3mm
- 12 – 16x Furações ϕ 2mm

Observações

As furações que se encontram nos cantos são furações de ϕ 3mm. Estas são usadas para conectar a placa de acrílico à caixa externa de ABS.

As furações de ϕ 2mm, estão dispostas de modo a permitir vários esquemas de montagem da placa de desenvolvimento.



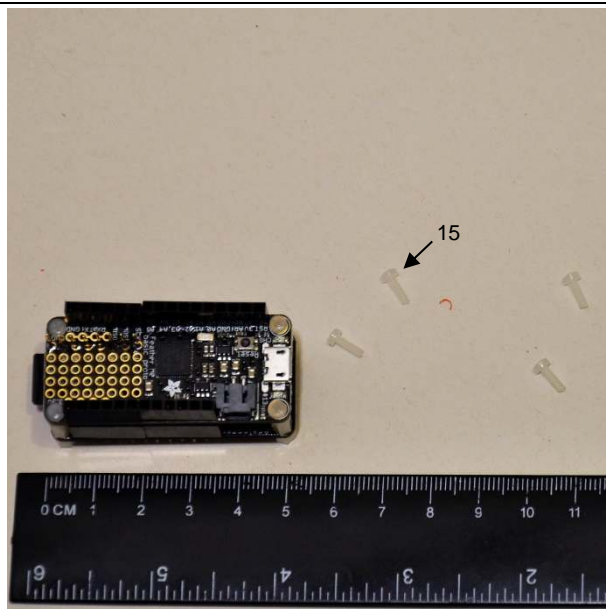
Materiais

13 – Transformador AC/DC 5V 1A

14 – Caixa de união

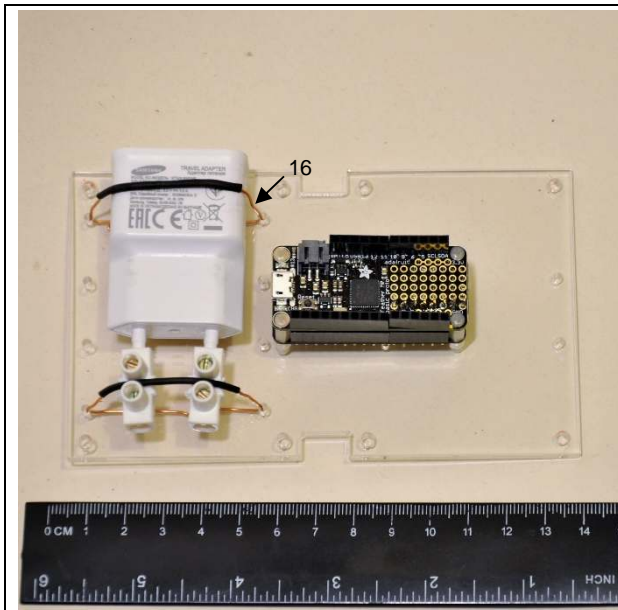
Tarefa (3)

Aplicar caixas de união ϕ 10mm² (14) nos terminais do transformador (13).



Materiais

15 – Parafusos 6mm nylon M2



Materiais

16 – Fio de cobre

Tarefa (4)

Fixar placas com os parafusos (15), no grupo central de furações ϕ 2mm.

Fixar transformador (13) ao grupo de furações mais à esquerda, com recurso a fio de cobre (16).

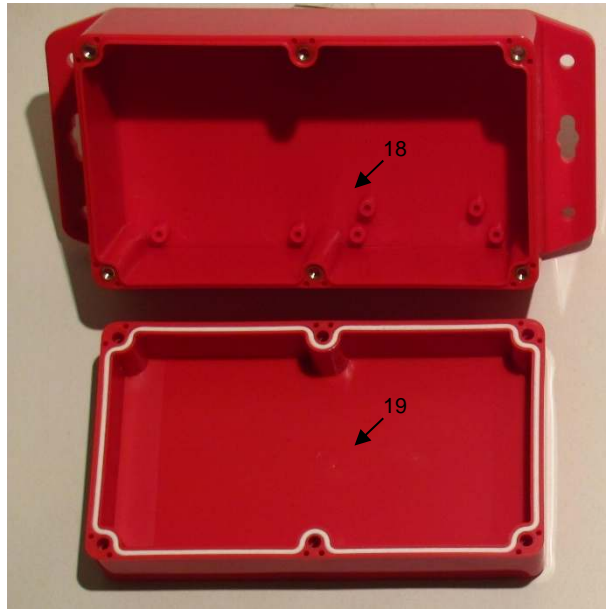


Materiais

17 – Calço de nivelamento do transformador (17)

Observações

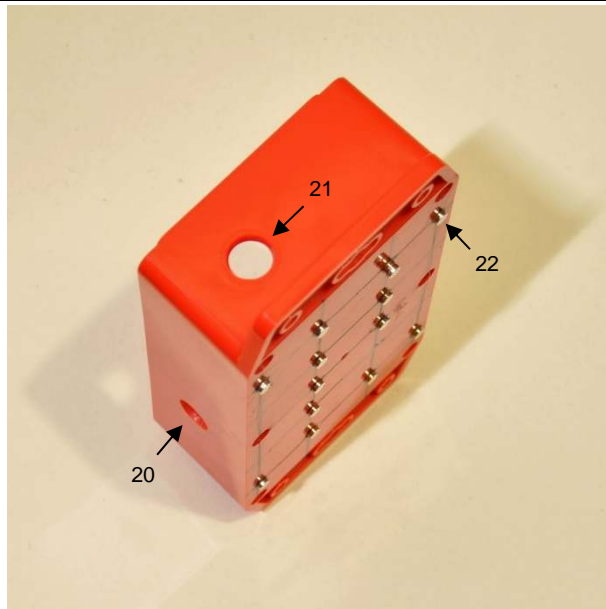
Perspetiva inversa do esquema de montagem amostrado anteriormente.



Materiais

18 – Fundo de caixa 74x158x90mm em ABS, com múltiplas furações de ϕ 3mm.

19 – Tampa de caixa em ABS



Materiais

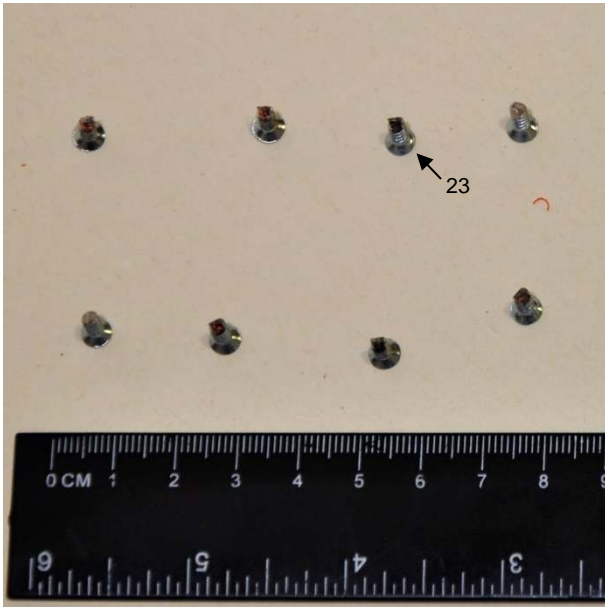
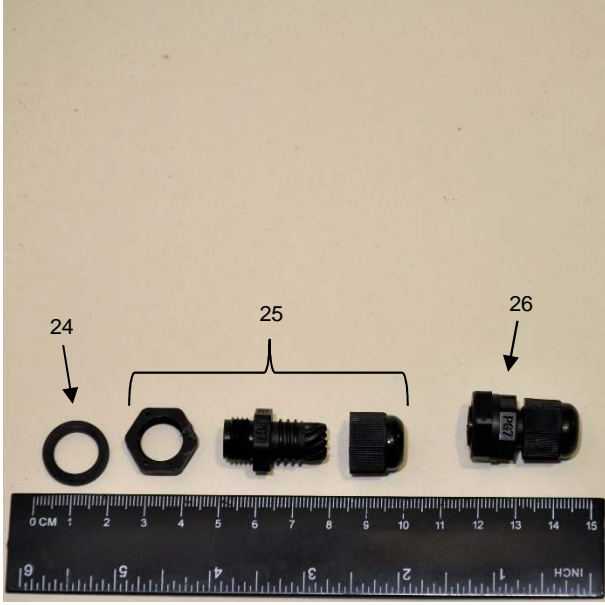
22 - Imãs 5x2mm

Tarefa (5)

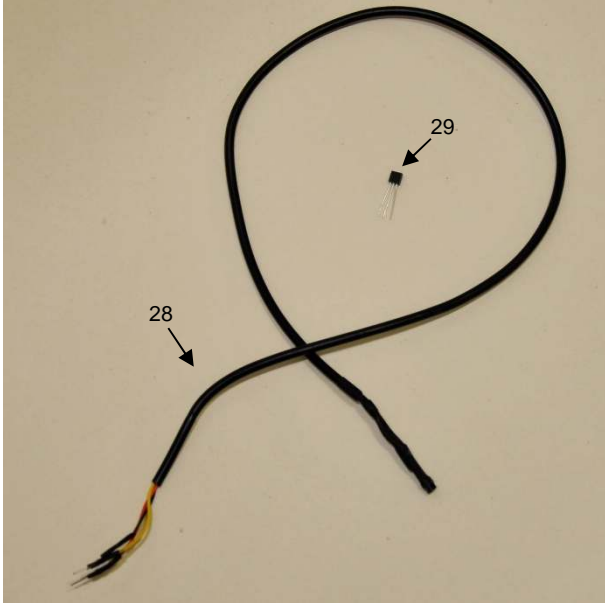
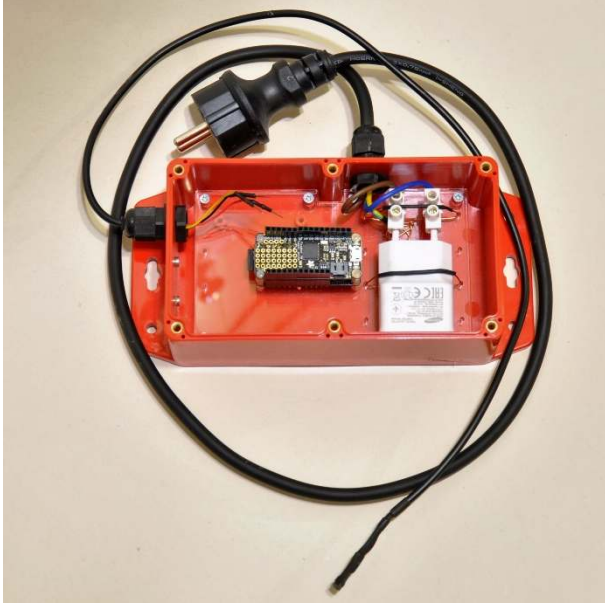
Execução de furação de ϕ 12mm para passagem de cabo de alimentação elétrica externa (20).

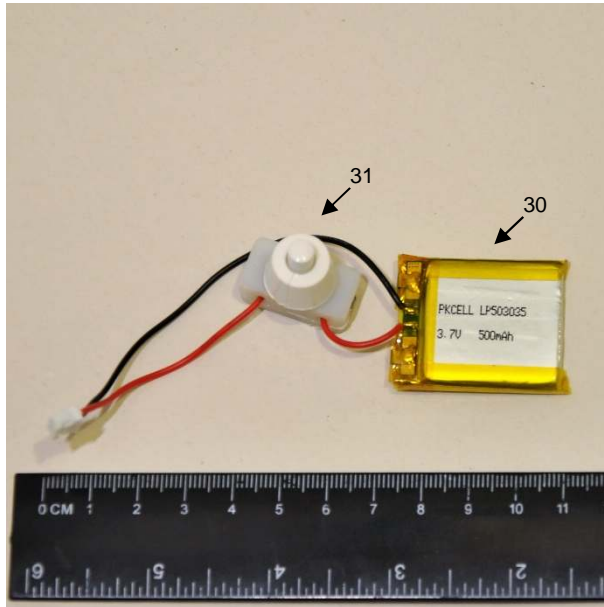
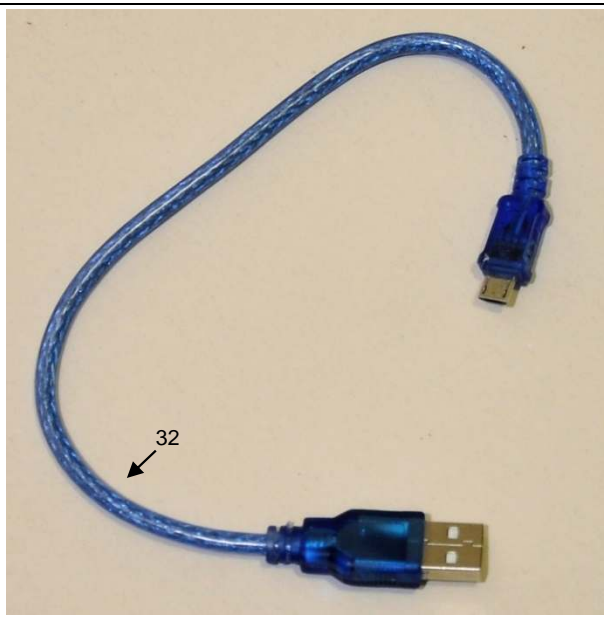
Execução de furação de ϕ 12mm para passagem de cabo comunicação externa a um sensor (temperatura, humidade, etc).

Aplicar na base da caixa ABS, múltiplos imãs 5x2mm (22), para habilitar a conexão magnética a superfícies metálicas.

| | |
|--|--|
|  <p>23</p> | <p><u> Materiais </u></p> <p>23 – 8x Parafusos ϕ 3mm</p> |
|  <p>24</p> <p>25</p> <p>26</p> | <p><u> Materiais </u></p> <p>24 – Junta de neopreno 12x18x2mm</p> <p>25 – Conector <i>Gland</i> PG7 3-6,5mm, desmontado (bucins).</p> <p>26 - Conector <i>Gland</i> PG7 3-6,5mm (bucins), montado com junta de neopreno (24)</p> |

| | |
|--|--|
|  | <p><u>Tarefa (6)</u></p> <p>Montagem do equipamento previamente preparado na tarefa (4), usando os parafusos (23).</p> <p>Aplicação dos dois conectores <i>Gland</i>, nas furações (20) e (22) previamente executadas na caixa em ABS.</p> |
|  | <p><u>Materiais</u></p> <p>27 – Cabo de alimentação elétrica com 1m e ficha</p> |

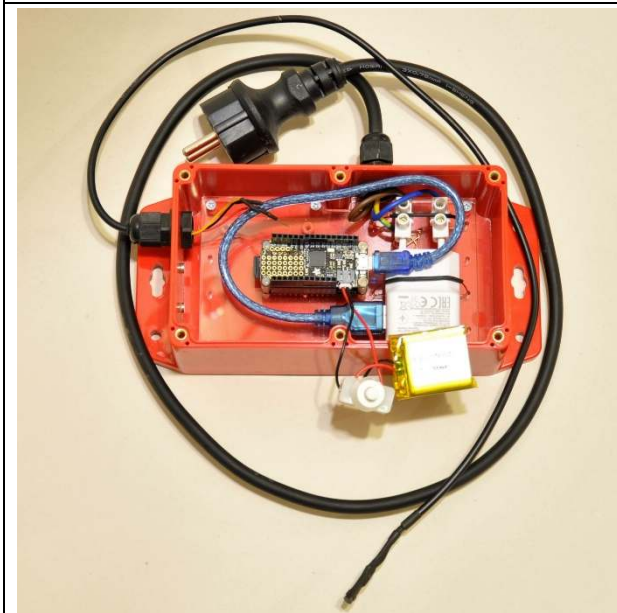
| | |
|---|---|
|  | <p><u> Materiais </u></p> <p>28 – Cabo de comunicação com três fios, com 1m de comprimento</p> <p>29 – Sensor de temperatura Dallas DS18B20</p> <p><u> Tarefa (7) </u></p> <p>Aplicar o sensor de temperatura DS 18B20 (29) no cabo de comunicação (28).</p> <p><u> Observações </u></p> <p>O cabo de comunicação mostrado na imagem, já possui o sensor montado, num terminal.</p> |
|  | <p><u> Tarefa (8) </u></p> <p>Passagem do cabo de alimentação (27) por um conector <i>Gland</i> e ligação dos seus terminais às caixas (14) de união, previamente montadas no transformador (13), ver tarefa (3).</p> <p>Passagem do cabo de comunicação por um conector <i>Gland</i>.</p> |

| | |
|--|---|
|  | <p><u> Materiais </u></p> <p>30 – Bateria polímeros de lítio de 500mAh</p> <p>31 – Interruptor click-clack</p> <p><u> Tarefa (9) </u></p> <p>Aplicação de interruptor (31) no fio de positivo da bateria de polímeros de lítio.</p> |
|  | <p><u> Materiais </u></p> <p>32 – Cabo USB 2.0 com conectores <i>type A</i> e <i>micro type B</i>, com 25cm</p> |



Tarefa (10)

Aplicação do cabo USB, conectando-o à placa Adafruit Feather M0 Proto ao transformador AC/DC 5V 0,5A.

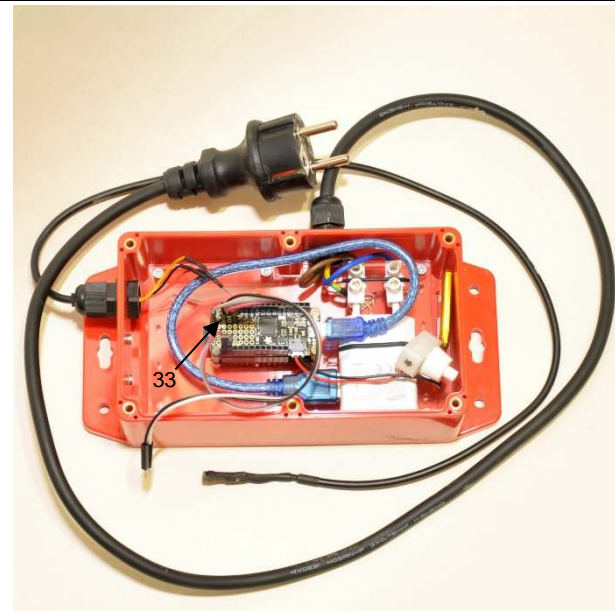


Tarefa (11)

Conectar a bateria de polímeros de lítio, à placa Adafruit Feather M0 Proto, pelo terminador JST-PH de 2 *pin*.

Observações

Antes de aplicar a bateria na placa, verificar se o botão está desligado, assim prevenindo o acionamento da placa.

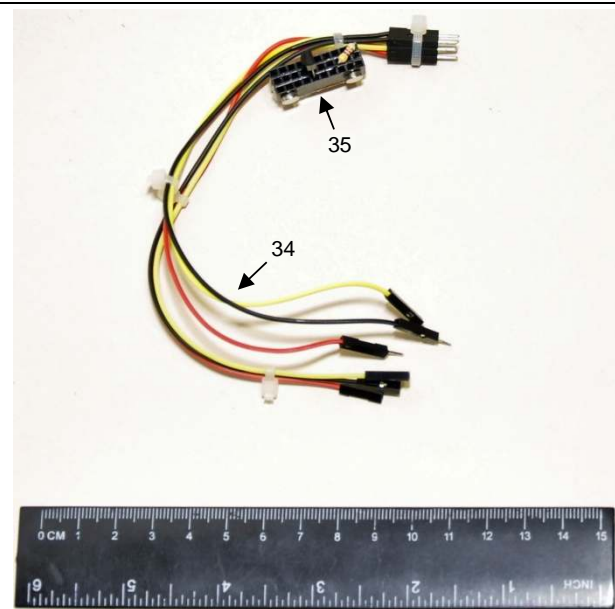


Materiais

33 – *Jumper* M/F com 20cm

Tarefa (11)

Aplicar 4 *jumpers* (33) no sensor (1) de acordo com o diagrama de ligações e montagem do dispositivo 2, ponto 5.



Materiais

34 – 3x jumper M/F 20cm + 3x jumper M/M 20cm

35 – 3x *headers* + Sensor temperatura Dallas DS18B20 + resistência 4,7k Ohms + 2x imãs 5x2mm

Tarefa (11)

Unir os terminais de 3 *headers* com fio e solda de estanho.

Colar os 3 *headers* lado a lado, para criar um pequeno circuito.

Aplicar um sensor DS18B20 (29), no circuito.

Ligar o pin do meio do sensor DS18B20 (29), a pista ou *header* positivo.

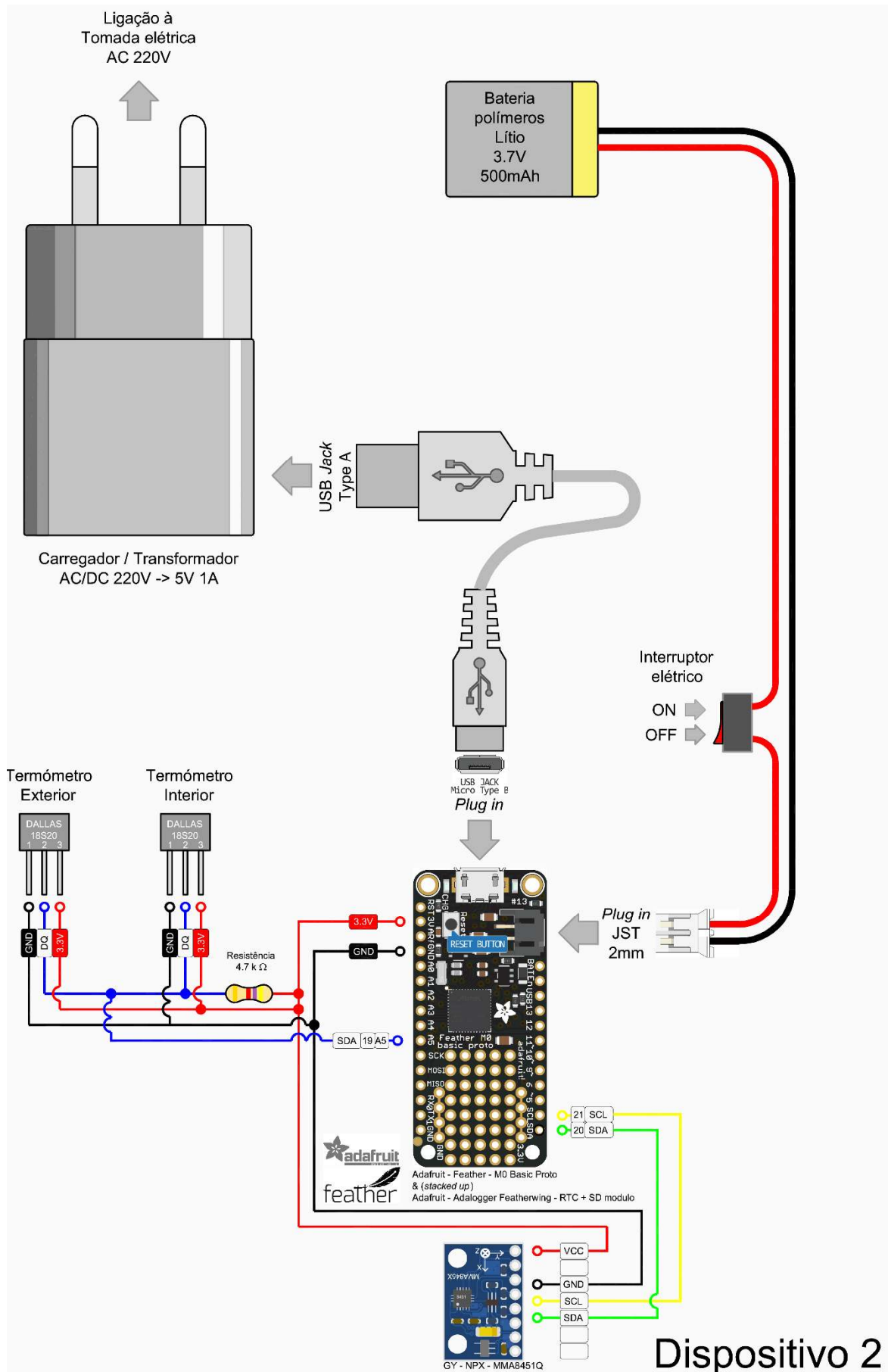
| | |
|--|---|
|  | <p><u> Materiais </u></p> <p>36 – Resistência 4,7k Ohm</p> <p><u> Tarefa (12) </u></p> <p>Colar 2 imãs (22) na parte lateral de <i>header</i>.</p> <p>Juntar 6 <i>jumpers</i> no lado em encontram os 6 <i>pin</i> machos, com recurso a uma abraçadeira de plástico.</p> <p>Agrupar 3 jumpers M/M e 3 jumpers M/F, com recurso a abraçadeira de plástico.</p> |
|  | <p><u> Tarefa (13) </u></p> <p>Aplicar o circuito, executado na tarefa (12), dentro da caixa ABS e conectar os terminais (<i>pin</i>) nos respetivos <i>headers</i>, conforme é indicado no diagrama de ligações e montagem do dispositivo 2, ponto 5.</p> <p>Instalar o programa ou <i>sketch</i> desenvolvido para dentro da placa Adafruit Feather M0 Proto, através da porta micro USB.</p> |



Tarefa (14)

Aplicar tampa da caixa ABS e fechar com os parafusos

5. Esquema de ligações para o Dispositivo 2



ANEXO B1 – *Sketchs* para o dispositivo 1

Sketch para o Dispositivo 1 com o sensor MMA8451Q

| | |
|----|--|
| 1 | /* |
| 2 | * Configurações |
| 3 | * Este sketch NÃO TEM o monitor serial ativo |
| 4 | * ODR do sensor MMA8451Q está configurado para 400Hz |
| 5 | * O Protocolo I2C está com um relógio de 100Hz |
| 6 | * O Protocolo SPI está com um relógio de 1000Hz |
| 7 | * Esta solução propõe uma taxa de escrita na ordem dos 237Hz |
| 8 | * n representa o número de ciclo em que o cartão está aberto para |
| 9 | * escrita e a cada ciclo fecha o cartão guardando a informação e |
| 10 | * abrindo este novamente para um novo ciclo. |
| 11 | * O ideal é gravar a cada 1:30 minuto, que a uma taxa +-21330 ciclos |
| 12 | * A cada novo ciclo resulta numa interrupção momentânea, entre 650ms |
| 13 | * a 850ms dependo de vários factores. |
| 14 | _____ |
| 15 | _____ |
| 16 | Ported to SdFat from the native Arduino SD library example by Bill Greiman |
| 17 | On the Ethernet Shield, CS is pin 4. SdFat handles setting SS |
| 18 | _____ |
| 19 | _____ |
| 20 | _____ |
| 21 | SD card read/write |
| 22 | _____ |
| 23 | This example shows how to read and write data to and from an SD card file |
| 24 | The circuit: |
| 25 | * SD card attached to SPI bus as follows: |
| 26 | * MOSI - pin 11 |
| 27 | * MISO - pin 12 |
| 28 | * CLK - pin 13 |
| 29 | * CS - pin 7 |
| 30 | _____ |
| 31 | _____ |
| 32 | _____ |
| 33 | @file Adafruit_MMA8451.h |
| 34 | @author K. Townsend (Adafruit Industries) |
| 35 | @license BSD (see license.txt) |
| 36 | _____ |

```
37 This is an example for the Adafruit MMA8451 Accel breakout board
38 ----> https://www.adafruit.com/products/2019
39
40 Adafruit invests time and resources providing this open source code,
41 please support Adafruit and open-source hardware by purchasing
42 products from Adafruit!
43
44 @section HISTORY
45
46 v1.0 - First release
47 _____
48
49 Modificado por Nuno Albuquerque Dezembro de 2016
50
51 */
52
53
54 #include <Wire.h>
55 #include <Adafruit_MMA8451.h>
56 #include <Adafruit_Sensor.h>
57
58 #include "SdFat.h"
59 SdFat sd;
60 SdFile myFile;
61
62 Adafruit_MMA8451 mma = Adafruit_MMA8451();
63
64 unsigned long time;
65
66 const int chipSelect = 7;
67
68 int n = 0;
69
70 int q = 21330; // 237Hz x 90s = 21330 ciclo, o ficheiro é guardado a cada 1:30
71
72 void setup(void) {
73
```

```
74 //Serial.begin(250000);
75
76 Wire.begin();
77 //TWBR = 12; // I2C communication at TWBR=32=200 kHz; TWBR=12=400 kHz
  (Arduino Uno)
78
79 mma.begin();
80 /*
81 if (! mma.begin()) {
82   Serial.println("Couldnt start");
83   while (1);
84 }
85 Serial.println("MMA8451 found!");
86 */
87
88 mma.setRange(MMA8451_RANGE_2_G);
89
90 /*
91 // Start I2C Transmission
92 Wire.beginTransmission(0x1C);
93 // Select data configuration
94 Wire.write(0x0E);
95 //Select range 0x00 [2g]; 0x01 [4g]; 0x02 [8g]
96 Wire.write(0x00);
97 // Stop I2C Transmission
98 Wire.endTransmission();
99 */
100
101 // In other to aply new chances in control register 1 it as to be reseted, put it in
  stanby by mode
102 // Start I2C Transmission
103 Wire.beginTransmission(0x1C);
104 // Select control register 1
105 Wire.write(0x2A);
106 Wire.write(0x00);
107 // 0x0D Select low noise configutation and an ODR 400hz and range 4g (only
  option)
108 // 0x09 Select normal mode and an ODR 400Hz
109 // 0x11 Select normal mode and an ODR 200Hz
```

```
110 // Stop I2C Transmission
111 Wire.endTransmission();
112
113 // Apply the new changes to control register 1
114 // Start I2C Transmission
115 Wire.beginTransmission(0x1C);
116 // Select control register 1
117 Wire.write(0x2A);
118 Wire.write(0x0D);
119 // 0x0D Select low noise configuration and an ODR 400hz and range 4g (only
    option)
120 // 0x09 Select normal mode and an ODR 400Hz
121 // 0x11 Select normal mode and an ODR 200Hz
122 // Stop I2C Transmission
123 Wire.endTransmission();
124
125
126 // Initialize SdFat or print a detailed error message and halt
127 // Use half speed like the native library.
128 // change to SPI_HALF_SPEED or SPI_FULL_SPEED for more performance.
129 if (!sd.begin(chipSelect, SPI_FULL_SPEED)) {
130     sd.initErrorHalt();
131 }
132 // open the file for write at end like the Native SD library
133 if (!myFile.open("MMA_01.txt", O_RDWR | O_CREAT | O_AT_END)) {
134     sd.errorHalt("opening test.txt for write failed");
135 }
136
137 myFile.println("_____MMA8451Q_____
    _____");
138
139 }
140
141 void loop() {
142
143     if(n!=q)
144     {
145         n=n+1;
```

```
146 }
147 else
148 {
149     myFile.close();
150
151     myFile.open("MMA_01.txt", O_WRITE | O_AT_END);
152
153     n=0;
154 }
155
156 time = micros();
157
158 // Read the 'raw' data in 14-bit counts
159 mma.read();
160
161 // Read the 'raw' data in 14-bit counts
162 // if the file opened okay, write to it:
163 myFile.print(q); myFile.print("\t");
164 myFile.print(time); myFile.print("\t");
165 myFile.print(mma.x); myFile.print("\t");
166 myFile.print(mma.y); myFile.print("\t");
167 myFile.println(mma.z);
168
169 /*
170 Serial.print(time); Serial.print("\t");
171 Serial.print(mma.x); Serial.print("\t");
172 Serial.print(mma.y); Serial.print("\t");
173 Serial.println(mma.z);
174 */
175
176 // Get a new sensor event
177 sensors_event_t event;
178 mma.getEvent(&event);
179
180 }
```

Sketch para o Dispositivo 1 com o sensor LIS331HH

| | |
|----|--|
| 1 | /* |
| 2 | * Configurações |
| 3 | * Este sketch NÃO TEM o monitor serial ativo |
| 4 | * ODR esta configurado para 400Hz |
| 5 | * O Protocolo I2C está com um relógio de 100Hz |
| 6 | * O Protocolo SPI está com um relógio de 1000Hz |
| 7 | * Esta solução propõe uma taxa de escrita na ordem dos 245Hz |
| 8 | * n representa o número de ciclo em que o cartão está aberto para |
| 9 | * escrita e a cada ciclo fecha o cartão guardando a informação e |
| 10 | * abrindo este novamente para um novo ciclo. |
| 11 | * O ideal é gravar a cada 1:30 minuto, que a uma taxa +-22050 ciclos |
| 12 | * A cada novo ciclo resulta numa interrupção momentânea, entre 650ms |
| 13 | * a 850ms dependo de vários factores. |
| 14 | _____ |
| 15 | _____ |
| 16 | Ported to SdFat from the native Arduino SD library example by Bill Greiman |
| 17 | On the Ethernet Shield, CS is pin 7. SdFat handles setting SS |
| 18 | _____ |
| 19 | _____ |
| 20 | _____ |
| 21 | SD card read/write |
| 22 | _____ |
| 23 | This example shows how to read and write data to and from an SD card file |
| 24 | The circuit: |
| 25 | * SD card attached to SPI bus as follows: |
| 26 | * MOSI - pin 11 |
| 27 | * MISO - pin 12 |
| 28 | * CLK - pin 13 |
| 29 | * CS - pin 7 |
| 30 | _____ |
| 31 | _____ |
| 32 | _____ |
| 33 | created Nov 2010 |
| 34 | by David A. Mellis |
| 35 | updated 2 Dec 2010 |
| 36 | by Tom Igoe |
| 37 | modified by Bill Greiman 11 Apr 2011 |

```
38   This example code is in the public domain.
39
40
41
42   Modificado por Nuno Albuquerque Dezembro de 2016
43
44   */
45
46   #include<Wire.h>
47   #include <SPI.h>
48   #include "SdFat.h"
49   SdFat sd;
50   SdFile myFile;
51
52   // LIS331HH I2C address is 0x18(pg24)
53   #define Addr 0x19
54
55   unsigned long time;
56
57   const int chipSelect = 7;
58
59   int n = 0;
60
61   int q = 22050; // 245Hz x 90s = 22050 ciclo, o ficheiro é guardado a cada 1:30
62
63   void setup()
64   {
65
66     /* Initialise Serial communication */
67
68     //Serial.begin(115200);
69
70
71     /* Initialise I2C communication */
72
73     Wire.begin();
74
```

| | |
|-----|--|
| 75 | /* I2C communication at TWBR=32=200 kHz; TWBR=12=400 kHz */ |
| 76 | |
| 77 | //TWBR = 32; |
| 78 | |
| 79 | /* |
| 80 | Opções do Registos de Controlo n.º1 (0x20) |
| 81 | 0x27 Enable X, Y, Z axis, power on mode, data rate selection = 50Hz, LPF 37Hz |
| 82 | 0x2F Enable X, Y, Z axis, power on mode, data rate selection = 100Hz, LPF 74Hz |
| 83 | 0x37 Enable X, Y, Z axis, power on mode, data rate selection = 400Hz, LPF 292Hz |
| 84 | 0x3F Enable X, Y, Z axis, power on mode, data rate selection = 1000Hz, LPF 780Hz |
| 85 | */ |
| 86 | // Start I2C Transmission with LIS331HH at 0x19 |
| 87 | Wire.beginTransmission(Addr); |
| 88 | // Select control register 1 |
| 89 | Wire.write(0x20); |
| 90 | Wire.write(0x37); |
| 91 | // Stop I2C Transmission |
| 92 | Wire.endTransmission(); |
| 93 | |
| 94 | |
| 95 | /* |
| 96 | Opções do Registos de Controlo n.º4 (0x23) |
| 97 | 0x00, Enable X, Y, Z axis at 6g |
| 98 | 0x10, Enable X, Y, Z axis at 12g |
| 99 | 0x30 Enable X, Y, Z axis at 24g |
| 100 | */ |
| 101 | // Start I2C Transmission |
| 102 | Wire.beginTransmission(Addr); |
| 103 | // Select control register 4 |
| 104 | Wire.write(0x23); |
| 105 | // Set Continuous update, +/- 6g, self-test enabled, LSB first, |
| 106 | Wire.write(0x00); |
| 107 | // Stop I2C Transmission |
| 108 | Wire.endTransmission(); |
| 109 | |
| 110 | |

```
111 // Initialize SdFat or print a detailed error message and halt
112 // Use half speed like the native library.
113 // change to SPI_HALF_SPEED or SPI_FULL_SPEED for more performance.
114 if (!sd.begin(chipSelect, SPI_FULL_SPEED)) {
115     sd.initErrorHalt();
116 }
117 // open the file for write at end like the Native SD library
118 if (!myFile.open("LIS_01.txt", O_RDWR | O_CREAT | O_AT_END)) {
119     sd.errorHalt("opening test.txt for write failed");
120 }
121
122 myFile.println("_____ LIS331HH _____");
123
124 }
125
126 void loop()
127 {
128     if(n!=q)
129     {
130         n=n+1;
131     }
132     else
133     {
134         myFile.close();
135
136         myFile.open("LIS_01.txt", O_WRITE | O_AT_END);
137
138         n=0;
139     }
140     time = micros();
141
142     unsigned int data[6];
143
144     for (int i = 0; i < 6; i++)
145     {
146         // Start I2C Transmission
```

```
147 Wire.beginTransmission(Addr);
148 // Select data register
149 Wire.write((40 + i));
150 // Stop I2C Transmission
151 Wire.endTransmission();
152
153 // Request 1 byte of data
154 Wire.requestFrom(Addr, 1);
155
156 // Read 6 bytes of data
157 // xAccl lsb, xAccl msb, yAccl lsb, yAccl msb, zAccl lsb, zAccl msb
158 if (Wire.available() == 1)
159 {
160     data[i] = Wire.read();
161 }
162 }
163
164 // Convert the data
165 int xAccl = ((data[1] * 256) + data[0]);
166 int yAccl = ((data[3] * 256) + data[2]);
167 int zAccl = ((data[5] * 256) + data[4]);
168
169
170 // if the file opened okay, write to it:
171 myFile.print(n); myFile.print("\t");
172 myFile.print(time); myFile.print("\t");
173 myFile.print(xAccl); myFile.print("\t");
174 myFile.print(yAccl); myFile.print("\t");
175 myFile.println(zAccl);
176
177 }
```

Sketch para o Dispositivo 1 com o sensor MPU6050

| | |
|----|--|
| 1 | /* |
| 2 | * Configurações |
| 3 | * Este sketch NÃO TEM o monitor serial ativado |
| 4 | * ODR esta configurado para 400Hz |
| 5 | * O Protocolo I2C está com um relógio de 200Hz |
| 6 | * O Protocolo SPI está com um relógio de 1000Hz |
| 7 | * Esta solução propõe uma taxa de escrita na ordem dos 252,7Hz |
| 8 | * q representa o número de ciclo em que o cartão está aberto para |
| 9 | * escrita e a cada ciclo fecha o cartão guardando a informação e |
| 10 | * abrindo este novamente para um novo ciclo. |
| 11 | * O ideal é gravar a cada 1:30 minuto, que a uma taxa +-22770 ciclos |
| 12 | * A cada novo ciclo resulta numa interrupção momentânea, entre 650ms |
| 13 | * a 850ms dependendo de vários factores. |
| 14 | |
| 15 | _____ |
| 16 | |
| 17 | Ported to SdFat from the native Arduino SD library example by Bill Greiman |
| 18 | On the Ethernet Shield, CS is pin 7. SdFat handles setting SS |
| 19 | |
| 20 | _____ |
| 21 | |
| 22 | SD card read/write |
| 23 | |
| 24 | This example shows how to read and write data to and from an SD card file |
| 25 | The circuit: |
| 26 | * SD card attached to SPI bus as follows: |
| 27 | * MOSI - pin 11 |
| 28 | * MISO - pin 12 |
| 29 | * CLK - pin 13 |
| 30 | * CS - pin 7 |
| 31 | |
| 32 | _____ |
| 33 | |
| 34 | MPU-6050 Short Example Sketch |
| 35 | By Arduino User JohnChi |
| 36 | August 17, 2014 |
| 37 | Public Domain |

38

39

40

41 Modificado por Nuno Albuquerque Dezembro de 2016

42

43 */

44

45

46 #include<Wire.h>

47 #include <SPI.h>

48 #include "SdFat.h"

49 SdFat sd;

50 SdFile myFile;

51

52 const int MPU_addr=0x68; // I2C address of the MPU-6050

53

54 // For the SD Shield Plus

55 const int chipSelect = 7;

56

57

58 //int16_t AcX,AcY,AcZ,Tmp;

59 int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;

60

61 unsigned long time;

62

63 int n = 0;

64

65 int q = 22770; // 252Hz x 90s = 22770 ciclo, o ficheiro é guardado a cada 1:30

66

67 void setup()

68 {

69 Wire.begin();

70 // TWBR = 32; // I2C communication at TWBR=32=200 kHz; TWBR=12=400 kHz

71

72 /*

73 * Register 107 – Power Management 1 (pg41) (0x6B)

74 * set to zero (wakes up the MPU-6050) [pg8]

| | |
|-----|---|
| 75 | * 0x00 reset value for all registers except 107 & 117 |
| 76 | |
| 77 | * PWR_MGMT_1 register 107 [pg41] , |
| 78 | * 0 bit7 don't reset device |
| 79 | * 0 bit6 sleep mode is off |
| 80 | * 0 bit5 cycle is off mode |
| 81 | * 0 bit4 is reserved |
| 82 | * 0 bit3 enable temp |
| 83 | * 000 bit3 2 1 set internal clk 8Mhz |
| 84 | */ |
| 85 | Wire.beginTransmission(MPU_addr); |
| 86 | Wire.write(0x6B); |
| 87 | Wire.write(0); |
| 88 | Wire.endTransmission(true); |
| 89 | |
| 90 | /* |
| 91 | * Register 28 – Accelerometer Configuration (0x1C) |
| 92 | * range selection 0x00 [2G] |
| 93 | * range selection 0x08 [4G] |
| 94 | * range selection 0x10 [8G] |
| 95 | * range selection 0x18 [16G] |
| 96 | */ |
| 97 | Wire.beginTransmission(MPU_addr); |
| 98 | Wire.write(0x1C); |
| 99 | Wire.write(0x00); |
| 100 | Wire.endTransmission(true); |
| 101 | |
| 102 | |
| 103 | /* |
| 104 | * Register 25 – Sample Rate Divider |
| 105 | * 19 = 13 = 400Hz |
| 106 | * 30 = 3E = 258.06hZ |
| 107 | * 31 = 3F = 250Hz |
| 108 | */ |
| 109 | Wire.beginTransmission(MPU_addr); |
| 110 | Wire.write(0x19); |
| 111 | Wire.write(0x13); |

```
112   Wire.endTransmission(true);
113   /*
114   Serial.begin(115200);
115
116   // Wait for USB Serial
117   while (!Serial) {
118     SysCall::yield();
119   }
120
121   Serial.println("Type any character to start");
122   while (!Serial.available()) {
123     SysCall::yield();
124   }
125   */
126   // Initialize SdFat or print a detailed error message and halt
127   // Use half speed like the native library.
128   // change to SPI_HALF_SPEED or SPI_FULL_SPEED for more performance.
129   if (!sd.begin(chipSelect, SPI_FULL_SPEED)) {
130     sd.initErrorHalt();
131   }
132   // open the file for write at end like the Native SD library
133   if (!myFile.open("MPU_01.txt", O_RDWR | O_CREAT | O_AT_END)) {
134     sd.errorHalt("opening test.txt for write failed");
135   }
136
137   myFile.println("_____MPU6050_____");
138
139   }
140
141   void loop()
142   {
143     if(n!=q)
144     {
145       n=n+1;
146     }
147     else
148     {
```

```
149     myFile.close();
150
151     myFile.open("MPU_01.txt", O_WRITE | O_AT_END);
152
153     n=0;
154 }
155
156     time = micros();
157
158     Wire.beginTransmission(MPU_addr);
159     Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
160     Wire.endTransmission(false);
161
162     Wire.requestFrom(MPU_addr,14,true); // request a total of 14 registers
163     AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C
    (ACCEL_XOUT_L)
164     AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E
    (ACCEL_YOUT_L)
165     AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
166     Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
167
168     GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
169     GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
170     GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
171
172     // if the file opened okay, write to it:
173     //myFile.print(q); myFile.print("\t");
174     myFile.print(time); myFile.print("\t");
175     myFile.print(AcX); myFile.print("\t");
176     myFile.print(AcY); myFile.print("\t");
177     myFile.print(AcZ); myFile.print("\t");
178     myFile.print(Tmp/340.00+36.53); myFile.print("\t");
179
180     myFile.print(GyX); myFile.print("\t");
181     myFile.print(GyY); myFile.print("\t");
182     myFile.println(GyZ);
183     /*
184     // Serial.print(n); Serial.print("\t");
```

```
185 Serial.print(time); Serial.print("\t");
186 Serial.print(AcX); Serial.print("\t");
187 Serial.print(AcY); Serial.print("\t");
188 Serial.print(AcZ); Serial.print(F("\t"));
189 Serial.println(Tmp/340.00+36.53);
190 //Serial.print("\t"); //equation for temperature in degrees C from datasheet
191
192 //Serial.print(GyX); Serial.print("\t");
193 //Serial.print(GyY); Serial.print("\t");
194 //Serial.println(GyZ);
195 */
196 }
```

ANEXO B2 – *Sketch* para o dispositivo

2

| | |
|----|--|
| 1 | /* |
| 2 | * Configurações |
| 3 | * ODR do sensor MMA8451Q esta configurado para 400Hz |
| 4 | * As janelas temporais são de 3s a curto praxo 30s a longo prazo |
| 5 | *Com esta definição a taxa de amostragem é na ordem dos 294Hz |
| 6 | *Caso a janela temporal a curto prazo seja alterada para de 1,5s a taxa de amostragem passa para 264Hz |
| 7 | **Caso a janela temporal a curto prazo seja alterada para de 1s a taxa de amostragem passa para 237Hz |
| 8 | |
| 9 | |
| 10 | |
| 11 | Ported to SdFat from the native Arduino SD library example by Bill Greiman |
| 12 | On the Ethernet Shield, CS is pin 4. SdFat handles setting SS |
| 13 | |
| 14 | |
| 15 | |
| 16 | SD card read/write |
| 17 | |
| 18 | This example shows how to read and write data to and from an SD card file |
| 19 | The circuit: |
| 20 | * SD card attached to SPI bus as follows: |
| 21 | * MOSI - pin 11 |
| 22 | * MISO - pin 12 |
| 23 | * CLK - pin 13 |
| 24 | * CS - pin 10 |
| 25 | |
| 26 | |
| 27 | |
| 28 | @file Adafruit_MMA8451.h |
| 29 | @author K. Townsend (Adafruit Industries) |
| 30 | @license BSD (see license.txt) |
| 31 | |
| 32 | This is an example for the Adafruit MMA8451 Accel breakout board |
| 33 | ----> https://www.adafruit.com/products/2019 |
| 34 | |
| 35 | Adafruit invests time and resources providing this open source code, |

| | |
|----|---|
| 36 | please support Adafruit and open-source hardware by purchasing |
| 37 | products from Adafruit! |
| 38 | |
| 39 | @section HISTORY |
| 40 | |
| 41 | v1.0 - First release |
| 42 | |
| 43 | |
| 44 | OneWire DS18S20, DS18B20, DS1822 Temperature Example |
| 45 | |
| 46 | http://www.pjrc.com/teensy/td_libs_OneWire.html |
| 47 | |
| 48 | The DallasTemperature library can do all this work for you! |
| 49 | http://milesburton.com/Dallas_Temperature_Control_Library |
| 50 | |
| 51 | |
| 52 | |
| 53 | Date and time functions using a DS1307 RTC connected via I2C and Wire lib |
| 54 | |
| 55 | |
| 56 | |
| 57 | Sketch criado Nuno Albuquerque julho 2017 |
| 58 | |
| 59 | */ |
| 60 | |
| 61 | #include <Wire.h> |
| 62 | #include <SPI.h> |
| 63 | |
| 64 | #include <Adafruit_MMA8451.h> |
| 65 | #include <Adafruit_Sensor.h> |
| 66 | |
| 67 | Adafruit_MMA8451 mma = Adafruit_MMA8451(); |
| 68 | |
| 69 | #include "SdFat.h" |
| 70 | SdFat sd; |
| 71 | SdFile myFile; |
| 72 | |

| | |
|-----|---|
| 73 | #include "RTClib.h" |
| 74 | |
| 75 | RTC_PCF8523 rtc; |
| 76 | |
| 77 | #include <OneWire.h> |
| 78 | #include <DallasTemperature.h> |
| 79 | |
| 80 | const int chipSelect = 10; |
| 81 | |
| 82 | unsigned long time; |
| 83 | |
| 84 | // Data wire is plugged into pin 19 on the Adafruit Feather M0 |
| 85 | #define ONE_WIRE_BUS 19 |
| 86 | |
| 87 | // Setup a oneWire instance to communicate with any OneWire devices (not just Maxim/Dallas temperature ICs) |
| 88 | OneWire oneWire(ONE_WIRE_BUS); |
| 89 | |
| 90 | // Pass our oneWire reference to Dallas Temperature. |
| 91 | DallasTemperature sensors(&oneWire); |
| 92 | |
| 93 | // definition of the integers needed |
| 94 | int ac_x; |
| 95 | int ac_y; |
| 96 | |
| 97 | float RMS_ST_X; |
| 98 | float RMS_ST_Y; |
| 99 | |
| 100 | float RMS_30s_X; |
| 101 | float RMS_30s_Y; |
| 102 | |
| 103 | float rac_X; |
| 104 | float rac_Y; |
| 105 | |
| 106 | int cicle_1 = 1; |
| 107 | int cicle_2 = 1; |
| 108 | int cicle_3 = 1; |
| 109 | |

| | |
|-----|--|
| 110 | |
| 111 | int q; |
| 112 | int i; |
| 113 | int n; |
| 114 | int save; |
| 115 | int repeat = 1; |
| 116 | |
| 117 | |
| 118 | // Definition of Short Time window |
| 119 | // cicle_ST = 882 -> Short Time window of 3s -> $\pm 294\text{Hz} \times 3 = 882$ |
| 120 | // cicle_ST = 396 -> Short Time window of 1,5s -> $\pm 264\text{Hz} \times 1,5 = 396$ |
| 121 | // cicle_ST = 237 -> Short Time window of 1s -> $\pm 237\text{Hz}$ |
| 122 | int cicle_ST = 882; |
| 123 | |
| 124 | // Definition of number of blocks needed to store the RMS parcels information |
| 125 | // RMS_#[11] -> 3s |
| 126 | // RMS_#[16] -> 2s |
| 127 | // RMS_#[21] -> 1,5s |
| 128 | // RMS_#[31] -> 1s |
| 129 | int RMS_X[11]; |
| 130 | int RMS_Y[11]; |
| 131 | // RMS_#[11] -> int block = 10 |
| 132 | // RMS_#[16] -> int block = 15 |
| 133 | // RMS_#[21] -> int block = 20 |
| 134 | // RMS_#[31] -> int block = 30 |
| 135 | int block = 10; |
| 136 | |
| 137 | // Period between temperature logs |
| 138 | int temp_log = 15; // in minutes |
| 139 | |
| 140 | // Converts the value attributed to the period between temperature logs in to number of cicles |
| 141 | int temp_log_cicle = ((temp_log*60)/(30/block)); |
| 142 | |
| 143 | |
| 144 | // Definition of individual parameters and group rule |
| 145 | float individual = 1.12; |
| 146 | float rule = 1.08; |

| | |
|-----|---|
| 147 | |
| 148 | |
| 149 | |
| 150 | void setup() { |
| 151 | |
| 152 | Serial.begin(250000); |
| 153 | |
| 154 | Wire.begin(); |
| 155 | |
| 156 | rtc.begin(); |
| 157 | |
| 158 | rtc.initialized(); |
| 159 | |
| 160 | mma.begin(); |
| 161 | |
| 162 | mma.setRange(MMA8451_RANGE_2_G); |
| 163 | |
| 164 | // In other to aply new chances in control register 1 it as to be reseted, put it in stanby by mode |
| 165 | // Start I2C Transmission |
| 166 | Wire.beginTransmission(0x1C); |
| 167 | // Select control register 1 |
| 168 | Wire.write(0x2A); |
| 169 | Wire.write(0x00); |
| 170 | // 0x0D Select low noise configutation and an ODR 400hz and range 4g (only option) |
| 171 | // 0x09 Select normal mode and an ODR 400Hz |
| 172 | // 0x11 Select normal mode and an ODR 200Hz |
| 173 | // Stop I2C Transmission |
| 174 | Wire.endTransmission(); |
| 175 | |
| 176 | // Apply the new changes to control register 1 |
| 177 | // Start I2C Transmission |
| 178 | Wire.beginTransmission(0x1C); |
| 179 | // Select control register 1 |
| 180 | Wire.write(0x2A); |
| 181 | Wire.write(0x0D); |
| 182 | // 0x0D Select low noise configutation and an ODR 400hz and range 4g (only option) |
| 183 | // 0x09 Select normal mode and an ODR 400Hz |

| | |
|-----|--|
| 184 | // 0x11 Select normal mode and an ODR 200Hz |
| 185 | // Stop I2C Transmission |
| 186 | Wire.endTransmission(); |
| 187 | |
| 188 | /***** initiates the SD Card at selected speed *****/ |
| 189 | |
| 190 | // Initialize SdFat or print a detailed error message and halt |
| 191 | // Use half speed like the native library. |
| 192 | // change to SPI_HALF_SPEED or SPI_FULL_SPEED for more performance. |
| 193 | if (!sd.begin(chipSelect, SPI_FULL_SPEED)) { |
| 194 | sd.initErrorHalt(); |
| 195 | } |
| 196 | |
| 197 | /***** Two new directories for Acceleration Log call Acc_Log and for Temperature Log cal Temp_Log *****/ |
| 198 | |
| 199 | sd.mkdir("Acc_Log"); |
| 200 | sd.mkdir("Temp_Log"); |
| 201 | |
| 202 | /***** Create a new data file for logging temperature and a new directory call Acceleration Log or Acc_Log *****/ |
| 203 | |
| 204 | // open the file for write at end like the Native SD library |
| 205 | if (!myFile.open("Temp_Log/Templog.txt", O_RDWR O_CREAT O_AT_END)) { |
| 206 | sd.errorHalt("opening test.txt for write failed"); |
| 207 | } |
| 208 | |
| 209 | myFile.close(); |
| 210 | |
| 211 | |
| 212 | /***** log a temperature in the open myFile and then close it afterwards *****/ |
| 213 | |
| 214 | logtemperature(); |
| 215 | |
| 216 | |
| 217 | /***** Create a new data file for logging vibrations *****/ |
| 218 | |
| 219 | // open the file for write at end like the Native SD library |
| 220 | if (!myFile.open("TempALoF.txt", O_RDWR O_CREAT O_AT_END)) { |
| 221 | sd.errorHalt("opening test.txt for write failed"); |
| 222 | } |
| 223 | |
| 224 | |
| 225 | /***** log a TimeStamp in the firth line of the vibration log *****/ |
| 226 | |
| 227 | timestamp(); |
| 228 | |

| | |
|-----|--|
| 229 | } |
| 230 | |
| 231 | |
| 232 | void loop() { |
| 233 | |
| 234 | /****** levels definition *****/ |
| 235 | |
| 236 | for (cicle_3=1; cicle_3<=repeat; cicle_3++){ |
| 237 | |
| 238 | for (cicle_2=1; cicle_2<=block; cicle_2++){ |
| 239 | |
| 240 | for (cicle_1=1; cicle_1<=cicle_ST; cicle_1++){ |
| 241 | |
| 242 | /****** 1st level *****/ |
| 243 | |
| 244 | //Read tima a store |
| 245 | time = micros(); |
| 246 | |
| 247 | // Read the 'raw' data in 14-bit counts |
| 248 | mma.read(); |
| 249 | |
| 250 | // Read the 'raw' data in 14-bit counts |
| 251 | // if the file opened okay, write to it: |
| 252 | myFile.print(time); myFile.print("\t"); |
| 253 | myFile.print(mma.x); myFile.print("\t"); |
| 254 | myFile.print(mma.y); myFile.print("\t"); |
| 255 | myFile.println(mma.z); |
| 256 | |
| 257 | // Store RMS partials and sum |
| 258 | for (i = 0; i <= block; i++){ |
| 259 | |
| 260 | RMS_X[i] = RMS_X[i] + pow((mma.x-ac_x),2); |
| 261 | RMS_Y[i] = RMS_Y[i] + pow((mma.y-ac_y),2); |
| 262 | } |
| 263 | |
| 264 | // Store aceleration |
| 265 | ac_x = mma.x; |
| 266 | ac_y = mma.y; |
| 267 | |
| 268 | // Get a new sensor event |
| 269 | sensors_event_t event; |
| 270 | mma.getEvent(&event); |
| 271 | |
| 272 | } |
| 273 | |
| 274 | /****** 2st level *****/ |
| 275 | |

| | |
|-----|---|
| 276 | // numeral index |
| 277 | n = n + 1; |
| 278 | |
| 279 | // Print numeral index |
| 280 | Serial.print(n); Serial.print("\t"); |
| 281 | |
| 282 | |
| 283 | |
| 284 | RMS_ST_X = sqrt(RMS_X[0]/cicle_ST); |
| 285 | RMS_ST_Y = sqrt(RMS_Y[0]/cicle_ST); |
| 286 | |
| 287 | Serial.print(RMS_ST_X); Serial.print("\t"); |
| 288 | Serial.print(RMS_ST_Y); Serial.print("\t"); |
| 289 | |
| 290 | RMS_X[0] = 0; |
| 291 | RMS_Y[0] = 0; |
| 292 | |
| 293 | |
| 294 | RMS_30s_X = sqrt(RMS_X[cicle_2]/(cicle_ST*block)); |
| 295 | RMS_30s_Y = sqrt(RMS_Y[cicle_2]/(cicle_ST*block)); |
| 296 | |
| 297 | |
| 298 | Serial.print(RMS_30s_X); Serial.print("\t"); |
| 299 | Serial.print(RMS_30s_Y); Serial.print("\t"); |
| 300 | |
| 301 | RMS_X[cicle_2]=0; |
| 302 | RMS_Y[cicle_2]=0; |
| 303 | |
| 304 | rac_X = RMS_ST_X / RMS_30s_X; |
| 305 | rac_Y = RMS_ST_Y / RMS_30s_Y; |
| 306 | |
| 307 | Serial.print(rac_X); Serial.print("\t"); |
| 308 | Serial.print(rac_Y); Serial.print("\t"); |
| 309 | |
| 310 | /****** Verification of racios Definition of individual parameters and group rule ******/ |
| 311 | |
| 312 | |
| 313 | /****** Trigger action ******/ |
| 314 | |
| 315 | // q define the number of cicles, that are associate to the grace period before start to evaluate de racio |
| 316 | |
| 317 | if(q <= block+1){ |
| 318 | |
| 319 | q = q + 1; |
| 320 | } |

| | |
|-----|---|
| 321 | |
| 322 | if(q<=block){ |
| 323 | } |
| 324 | |
| 325 | //Evaluation of the individual axes |
| 326 | |
| 327 | else if(rac_X >= individual rac_Y >= individual){ |
| 328 | |
| 329 | save = save + 1; |
| 330 | } |
| 331 | |
| 332 | //Evaluation of the group of axes |
| 333 | else if(rac_X >= rule && rac_Y >= rule){ |
| 334 | |
| 335 | save = save + 1; |
| 336 | } |
| 337 | |
| 338 | // Print the result of the evaluation |
| 339 | Serial.print(save); Serial.print("\t"); |
| 340 | |
| 341 | |
| 342 | |
| 343 | /****** Actions performed when the Trigger action is activated *****/ |
| 344 | |
| 345 | // Define the quantity of save data time save = 1 -> 60s; save = 2 -> 90s |
| 346 | if(save == 1){ |
| 347 | |
| 348 | repeat=2; |
| 349 | } |
| 350 | |
| 351 | else if (save == 2){ |
| 352 | |
| 353 | repeat=3; |
| 354 | } |
| 355 | |
| 356 | // Print the quantity factor of save data time |
| 357 | Serial.print(repeat); Serial.println("\t"); |
| 358 | |
| 359 | } |
| 360 | |
| 361 | /****** 3th level *****/ |
| 362 | |
| 363 | } |
| 364 | |
| 365 | |
| 366 | if(save>=1){ |
| 367 | |

| | |
|-----|---|
| 368 | myFile.close(); |
| 369 | |
| 370 | rename_and_timestamp(); |
| 371 | |
| 372 | myFile.open("TempALoF.txt", O_RDWR O_CREAT O_AT_END); |
| 373 | |
| 374 | timestamp(); |
| 375 | |
| 376 | repeat=1; |
| 377 | |
| 378 | save=0; |
| 379 | |
| 380 | Serial.println("log acceleration"); |
| 381 | |
| 382 | if(n >= temp_log_cicle){ |
| 383 | |
| 384 | n = 0; |
| 385 | |
| 386 | myFile.close(); |
| 387 | |
| 388 | logtemperature(); |
| 389 | |
| 390 | myFile.open("TempALoF.txt", O_RDWR O_CREAT O_AT_END); |
| 391 | |
| 392 | Serial.println("log temperature"); |
| 393 | |
| 394 | } |
| 395 | |
| 396 | } |
| 397 | |
| 398 | /****** Actions performed when the Trigger action is not activated *****/ |
| 399 | |
| 400 | else{ |
| 401 | |
| 402 | |
| 403 | myFile.remove(); |
| 404 | |
| 405 | myFile.close(); |
| 406 | |
| 407 | myFile.open("TempALoF.txt", O_RDWR O_CREAT O_AT_END); |
| 408 | |
| 409 | timestamp(); |
| 410 | |
| 411 | repeat=1; |
| 412 | |
| 413 | save=0; |
| 414 | |

| | |
|-----|--|
| 415 | Serial.println("din't save to log"); |
| 416 | |
| 417 | if(n >= temp_log_cicle){ |
| 418 | |
| 419 | n = 0; |
| 420 | |
| 421 | myFile.close(); |
| 422 | |
| 423 | logtemperature(); |
| 424 | |
| 425 | myFile.open("TempALoF.txt", O_RDWR O_CREAT O_AT_END); |
| 426 | |
| 427 | Serial.println("log temperature"); |
| 428 | |
| 429 | } |
| 430 | |
| 431 | } |
| 432 | |
| 433 | } |
| 434 | |
| 435 | /****** End of loop *****/ |
| 436 | |
| 437 | |
| 438 | |
| 439 | |
| 440 | /****** Section of the sketch destined for the Constructors or subroutines ******/ |
| 441 | |
| 442 | |
| 443 | |
| 444 | |
| 445 | /****** Constructor 1 *****/ |
| 446 | |
| 447 | void logtemperature(){ |
| 448 | |
| 449 | /****** loges the date and temperature on a file that exists in folder called Temp_Log *****/ |
| 450 | |
| 451 | myFile.open("Temp_Log/Templog.txt", O_WRITE O_AT_END); |
| 452 | |
| 453 | // Send the command to get time for timestamp |
| 454 | DateTime now = rtc.now(); |
| 455 | |
| 456 | myFile.print(now.day(), DEC); |
| 457 | myFile.print('/'); |
| 458 | myFile.print(now.month(), DEC); |
| 459 | myFile.print('/'); |

```

460 |   myFile.print(now.year(), DEC);
461 |   myFile.print("\t");
462 |   myFile.print(now.hour(), DEC);
463 |   myFile.print(':');
464 |   myFile.print(now.minute(), DEC);
465 |   myFile.print(':');
466 |   myFile.print(now.second(), DEC);
467 |   myFile.print("\t");
468 |   myFile.print("Temp exterior:");
469 |   myFile.print("\t");
470 |
471 | // Send the command to get temperatures
472 |   sensors.requestTemperatures();
473 |
474 |   myFile.print(sensors.getTempCByIndex(0));
475 |   myFile.print("\t");
476 |   myFile.print("Temp interior:");
477 |   myFile.print("\t");
478 |   myFile.println(sensors.getTempCByIndex(1));
479 |
480 |   myFile.close();
481 | }
482 |
483 |
484 |
485 |
486 | /***** Constructor 2 *****/
487 |
488 | void rename_and_timestamp(){
489 |
490 | /***** Rename the temporary acceleration log file using the present date and
491 | relocate it in the folder Acc_log *****/
492 |   DateTime now = rtc.now();
493 |
494 |   String nyear = String(now.year(), DEC);
495 |   String nmonth = String(now.month(), DEC);
496 |   String nday = String(now.day(), DEC);
497 |   String nhour = String(now.hour(), DEC);
498 |   String nminute = String(now.minute(), DEC);
499 |
500 |   String timestamp1 = nmonth + nday + nhour + nminute + ".txt";
501 |   String timestamp2 = "Acc_Log/" + timestamp1;
502 |
503 |   int str_len = timestamp2.length() + 1;
504 |
505 |   char fileName[str_len];

```

| | |
|-----|---|
| 506 | |
| 507 | timestamp2.toCharArray(fileName, str_len); |
| 508 | |
| 509 | |
| 510 | SdFile file("TempALoF.txt", O_WRITE); |
| 511 | file.isOpen(); |
| 512 | |
| 513 | file.rename(sd.vwd(), fileName); |
| 514 | |
| 515 | file.close(); |
| 516 | |
| 517 | } |
| 518 | |
| 519 | |
| 520 | |
| 521 | /****** Constructor 3 *****/ |
| 522 | |
| 523 | void timestamp(){ |
| 524 | |
| 525 | /****** Create a timestamp for the acceleration log file *****/ |
| 526 | |
| 527 | DateTime now = rtc.now(); |
| 528 | |
| 529 | String nyear = String(now.year(), DEC); |
| 530 | String nmonth = String(now.month(), DEC); |
| 531 | String nday = String(now.day(), DEC); |
| 532 | String nhour = String(now.hour(), DEC); |
| 533 | String nminute = String(now.minute(), DEC); |
| 534 | String nsecond = String(now.second(), DEC); |
| 535 | |
| 536 | String timestamp = nday + '/' + nmonth + '/' + nyear + "\t" + nhour + ':' + nminute + ':' + nsecond; |
| 537 | |
| 538 | myFile.print("----"); |
| 539 | myFile.print(timestamp); |
| 540 | myFile.println("----"); |
| 541 | |
| 542 | } |
| 543 | |
| 544 | |