



Departamento de Física

Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Instrumentação para Estudo da Pressão dos Dedos no Teclado do Computador e Relação com Patologias Associadas ao Acto de Dactilografar

João Pedro Almeida Meneses

Professores Orientadores: Doutor António Miguel Lino Santos Morgado e Doutora
Paula Cristina Vaz Bernardo Tavares

Mestrado Integrado em Engenharia Biomédica

Setembro de 2009

“Apenas desistir é que faz de nós fracos!”

Eu próprio.

Agradecimentos

Aos meus orientadores, Professor Miguel Morgado e Professora Paula Tavares, cuja disponibilidade, dedicação e apoio ao longo deste projecto, foram fundamentais para a conclusão deste.

Ao GEI, por acolher este projecto, pela cedência do espaço e de diverso material indispensável ao desenvolvimento deste projecto.

Ao Professor Jorge Landeck, pela ideia de se utilizar o PSoC® neste projecto.

Ao pessoal do Serviço de Apoio a Cliente da Cypress™, nomeadamente aos engenheiros de aplicação Snehal Acharekar e Abhijeet, pelo esclarecimento de dúvidas no desenvolvimento do projecto em PSoC®.

A todos os amigos que fiz durante esta jornada de 5 anos, com quem partilhei inúmeros momentos, para os quais não há palavras que expressem a enorme gratidão que lhes tenho.

Ao pessoal de Porto de Mós, em especial aos amigos que permanecem presentes na minha vida.

A toda a minha família, em especial aos meus pais, cujo esforço e dedicação permitiu que eu chegasse até este ponto. A eles, devo tudo o que sou. E a eles, dedico este trabalho.

Índice

Capítulo 1: Introdução, o Estado da Arte e os Objectivos da Instrumentação	9
Génese das Patologias Associadas ao Acto de Dactilografar	9
A Influência do Acto de Dactilografar no Aparecimento de MSDs	10
Os Equipamentos Existentes para Estudar a Força Aplicada ao Dactilografar	11
Motivação e Objectivos de uma Nova Instrumentação	12
Capítulo 2: Desenvolvimento do Hardware	14
O Sensor	14
Introdução às Medições de Força	14
Os Tipos de Transdutores de Força	15
Critérios de Selecção do Sensor de Força para esta Aplicação	19
Estudo Comparativo das Tecnologias	20
Estudo Comparativo dos Sensores Piezoresistivos	21
Propriedades do Sensor Seleccionado	22
Calibração Estática no Sensor Seleccionado	22
Seleção do Sistema de Aquisição de Dados e do Sistema de Comunicação <i>Wireless</i>	27
Propriedades do Sistema de Aquisição de Dados	27
SADs Estudados e a Solução Encontrada	28
Introdução ao conceito de Microcontrolador e a escolha do PSoC®	29
O Sistema de Comunicação <i>Wireless</i> Escolhido	30
Esquema do Projecto	33
O PSoC®	34
A Arquitectura Geral do PSoC®	34
O Núcleo do PSoC®	35
O Sistema Digital	39
Os Blocos Analógicos	41
Os Recursos Adicionais do PSoC®	45
Como Desenvolver o Projecto e Programar o PSoC®	45
A Plataforma de Desenvolvimento da Instrumentação	47
O <i>Hub</i>	50
O <i>hardware</i> do <i>Hub</i>	50
O <i>firmware</i> do <i>Hub</i>	50
O Software Cypress Sense and Control Dashboard (SCD)	51
O <i>Nodo</i>	53

O <i>hardware</i> do Nodo	53
O Sistema a Configurar no PSoC®	54
Criar o Projecto no Nível Chip do PSoC Designer™	55
A Selecção e Configuração dos <i>User Modules</i>	57
Os recursos globais do PSoC®	64
O <i>firmware</i> do Nodo	65
Gerar e programar o Nodo	66
Capítulo 3: Desenvolvimento do Software	68
A SCD	68
Iniciar a Ligação <i>Hub</i> - Nodo	68
Configurar os Dados do Nodo.....	69
Monitorizar o valor de Força na SCD	69
Exportar os dados obtidos pela SCD	71
A Aplicação de Análise e Processamento de Dados.....	72
Descrição da Aplicação Gráfica.....	72
Capítulo 4: Testes, Discussão de Resultados e Conclusões Finais.....	76
Testes Realizados	76
Calibração Estática.....	77
Tempo de Aquisição de 50 minutos contínuos.....	81
Multiplexagem	82
Operar com dois Nodos em simultâneo	82
Conclusões Finais	84
Trabalho Futuro.....	85
Anexos	86
Anexo I – O código do <i>firmware ForceFingerTips</i>	86
Anexo II – O código da Aplicação Gráfica <i>ForceGlv</i>	91
Anexo III – O código das Funções criadas para a Aplicação Gráfica <i>ForceGlv</i>	97
Anexo IV – A Data Sheet do Sensor FSS	100
Bibliografia e Webgrafia.....	103

Índice de Figuras

Figura 1 – Esquema básico de um sistema de medição de forças. F_0 é a força aplicada. Retirada de [1]. **(pág. 15)**

Figura 2 – No diagrama da esquerda encontra-se uma *strain gauge* do tipo resistência eléctrica em forma de chapa. No diagrama da direita encontram-se *strain gauges* do mesmo tipo dispostas de forma a poder obter informações angulares (a, b, c, d), estes arranjos de strain gauges designam-se rosettes. Imagens retiradas de [1]. **(pág. 17)**

Figura 3 – *Strain gauges* ligadas a uma unidade estrutural. F_0 é a força aplicada. Na figura (a) as strain gauges vão sofrer uma tensão, enquanto na figura (b) vão sofrer uma compressão. Imagem retirada de [1]. **(pág. 18)**

Tabela 1 – Comparação de algumas propriedades das tecnologias utilizadas em *strain gauges*. **(pág. 20)**

Tabela 2 – Comparação dos sensores passíveis de serem utilizados neste projecto. **(pág. 22)**

Figura 4 – À direita, foto da montagem para calibrar o sensor FSS. À esquerda, foto das massas de referência utilizadas para a calibração. **(pág. 23)**

Tabela 3 – Massas de referência utilizadas. **(pág. 24)**

Tabela 4 – Resultados da calibração estática do sensor FSS. **(pág. 25)**

Gráfico 1 – Resultados da calibração estática do sensor FSS. **(pág. 25)**

Tabela 5 – Estudo da repetitividade do sensor FSS. **(pág. 25)**

Figura 5- (a) Espaço de memória ocupado por diferentes tecnologias *wireless*; (b) Resposta do CyFi™ às interferências do meio; (c) Comparativo entre tecnologias da imunidade às interferências e agilidade na alteração do canal de comunicação; Direitos das imagens detidos por: © 2008 Cypress Semiconductor Corporation. **(pág. 32)**

Figura 6 – Esquema da modelização por *Direct Sequence Spread Spectrum* (DSSS); Direitos da imagem detidos por: © 2008 Cypress Semiconductor Corporation. **(pág. 33)**

Figura 7 – Esquema da instrumentação. **(pág. 33)**

Figura 8 – Diagrama de blocos sobre a arquitectura geral da família de microcontroladores PSoC®. Imagem retirada do *Technical Reference Manual* [29]. **(pág. 34)**

Figura 9 – Esquemático do núcleo do PSoC. Imagem retirada do *Technical Reference Manual* [29]. **(pág. 35)**

Figura 10 – Diagrama de blocos de cada GPIO, esquemático dos drive modes. Imagem retirada do *Technical Reference Manual* [29]. **(pág. 38)**

Figura 11 – Diagrama do Sistema Digital. Imagem retirada do *Technical Reference Manual* [29]. **(pág. 39)**

Figura 12 – Diagrama da arquitectura de um Bloco Digital. Imagem retirada do *Technical Reference Manual* [29]. **(pág. 40)**

Figura 13 – Diagrama do sistema analógico. Retirada do *Technical Reference Manual* [29]. **(pág. 41)**

Figura 14 – Esquemático do bloco analógico CT. Retirada do *Technical Reference Manual* [29]. **(pág. 43)**

Figura 15 – Esquemático do bloco analógico SC do tipo C. Retirada do *Technical Reference Manual* [29]. **(pág. 44)**

Figura 16 – Esquemático do bloco analógico SC do tipo C. Retirada do *Technical Reference Manual* [29]. **(pág. 44)**

Figura 17 – Diagrama dos recursos adicionais do PSoC®. Retirada do *Technical Reference Manual* [29]. **(pág. 45)**

Figura 18 – Ambiente de desenvolvimento do PSoC Designer™. **(pág. 47)**

Figura 19 – As cinco placas disponíveis no kit first touch CY3271. [33] **(pág. 48)**

Figura 20 – As placas do kit CY3271 utilizadas no projecto: (20a) A placa FTPC; (20b) A placa FTRF; (20c) A placa AAA. **(pág. 49)**

Figura 21 – Foto do Hub do projecto. Formado pela placa FTPC do kit CY3271 e por um cabo USB. **(pág. 51)**

Figura 22 – O ambiente de monitorização da SCD. Imagem retirada de [33]. **(pág. 52)**

Figura 23 – Foto do Nodo do projecto composto pela placa FTRF, pela placa AAA e pelo sensor de força FSS. **(pág. 54)**

Figura 24 – Diagrama de blocos da arquitectura do Nodo com os principais elementos utilizados para processar o sinal do sensor de força e a sua localização no dispositivo. **(pág. 54)**

Figura 25 – As propriedades do multiplexador 4:1 na instrumentação final. Imagem retirada do PSoC Designer™. **(pág. 58)**

Figura 26 – Topologia do amplificador diferencial implementado na instrumentação final e as equações que descrevem o seu funcionamento. Retirado da AN2367. [38] **(pág. 58)**

Figura 27 – Propriedades dos dois PGAs (em cima), o seu posicionamento e ligações (em baixo) na instrumentação final. Retirado do PSoC Designer™. **(pág. 60)**

Figura 28 – Esquemático do ADCINC com um modulador de 2ª ordem. Retirada da *Data Sheet* [41]. **(pág. 62)**

Figura 29 – Propriedades do ADCINC na instrumentação final. Retirado do PSoC Designer™. **(pág. 62)**

Figura 30 – Parâmetros do CyFi™ SNP na instrumentação final. Retirado do PSoC Designer™. **(pág. 63)**

Figura 31 – As propriedades do temporizador na instrumentação final. Retirado do PSoC Designer™. **(pág. 64)**

Figura 32 – Propriedades dos recursos globais do PSoC® na instrumentação final. Retirada do PSoC Designer™. **(pág. 64)**

Figura 33 – O aspecto do PSoC Designer™ após a configuração dos módulos para este projecto. **(pág. 65)**

Figura 34 – A estrutura do *firmware*. **(pág. 67)**

Figura 35 – Janela para a ligação no Nodo na rede. Retirado da SCD. **(pág. 68)**

Figura 36 – Janela para configurar os Dados provenientes de cada sensor do Node. Retirado da SCD. **(pág. 69)**

Figura 37 – Janela para definir a calibração do sensor na SCD. Retirado da SCD. **(pág. 70)**

Figura 38 – Janela para escolha da origem dos dados a carregar no MATLAB®. Encontram-se carregados os dados do dedo indicador direito. **(pág. 72)**

Figura 39 – Janela de pop-up para selecção do objecto de estudo. **(pág. 73)**

Figura 40 – A Aplicação Gráfica ForceGlv em MATLAB®. **(pág. 75)**

Tabela 6 – Propriedades da instrumentação. **(pág. 77)**

Tabela 7 - Resultados da calibração estática do sensor FSS, a operar a 5V e ligado á instrumentação desenvolvida no PSoC®. **(pág. 78)**

Gráfico 2 - Resultados da calibração estática do sensor FSS, a operar a 5V e ligado á instrumentação desenvolvida no PSoC®. **(pág. 78)**

Tabela 8 - Resultados da calibração estática do sensor FSS, a operar a 3,3V e ligado á instrumentação desenvolvida no PSoC®. **(pág. 80)**

Gráfico 3 - Resultados da calibração estática do sensor FSS, a operar a 3,3V e ligado á instrumentação desenvolvida no PSoC®. **(pág. 80)**

Figura 41 – A monitorização na SCD com dois Nodos ligados ao *Hub*. Aqui o *Hub* encontra-se a operar a 3 MHz. **(pág. 83)**

Capítulo 1:

A Introdução, o Estado da Arte e os Objectivos da Instrumentação

Génese das Patologias Associadas ao Acto de Dactilografar

Actualmente o uso de computadores encontra-se generalizado a nível mundial. Estimou-se em 2008, que o número de computadores instalados já superava a marca de 1 bilião de unidades e previu-se que até ao ano de 2014, esse volume iria duplicar [1]. A confirmar estes dados, encontram-se os indicadores crescentes da utilização das novas tecnologias de comunicação global, como a Internet. Levantamentos estatísticos apontam que para uma população mundial (estimada em 2008) de 6,710,029,070 habitantes, 23.8% utilizavam a Internet, tendo-se registado um crescimento no número de utilizadores entre os anos de 2000 e 2008, na ordem dos 342.2% [2].

A *Wellnomics* apresentou em 2007 uma análise ao comportamento de 50,000 utilizadores de computadores, pertencentes a um total de 95 organizações espalhadas pelo mundo [3]. Este estudo revelou que a maioria dos trabalhadores, durante as horas de trabalho, utiliza o computador uma média de 2.4 horas por dia, o que dá, em média, 12.4 horas por semana, numa semana com 5 dias de trabalho. O mesmo estudo reporta também que 12% dos utilizadores excede as 20 horas semanais de utilização e que 1% excede as 30 horas de utilização do computador. Dados que não têm em conta a utilização do computador nos períodos de lazer, factor que irá certamente aumentar a média da sua utilização. Ainda neste estudo são apontados, como factores

responsáveis pelo aparecimento de uma RSI (*Repetitive Strain Injury*), a postura estática e a tensão muscular, especialmente as que estão ligadas ao uso do rato do computador, e numa escala menor as associadas ao teclado no computador.

Este uso crescente do computador, associado por vezes a uma utilização excessiva e a um uso incorrecto relacionado quer com a ergonomia dos dispositivos (computador, teclado, rato, local de trabalho), quer com a postura do sujeito (posição do tronco, braço, pulso, mão) [4,5,6,7], levou a um agravar no número de doentes com patologias músculo-esqueléticas (MSD) das extremidades superiores. Concluiu-se num estudo de 2002 [8] que mais de 50% dos utilizadores de computadores apresentam sintomas de doenças músculo-esqueléticas (afectando principalmente mãos, braços, pescoço e ombros) durante o primeiro ano de um novo trabalho relacionado com o uso de computadores. Também se concluiu que as MSDs são comuns entre utilizadores de computadores. Procurou-se então demonstrar que o risco de MSDs, como é o caso das tendinites e da síndrome do túnel cárpico (CTS), aumentam com o trabalho prolongado num computador [9], facto que se verificou.

A Influência do Acto de Dactilografar no Aparecimento de MSDs

Encontram-se na Literatura alguns estudos relativos à influência da pressão exercida no acto de dactilografar no aumento da pressão no túnel cárpico e portanto no nervo mediano, factor reconhecido como sendo um dos possíveis causadores do CTS. Um destes estudos [10] refere, com base em imagiologia MRI, uma relação entre o CTS e a contracção dos tendões flectores dos dedos e a flexão/extensão do pulso, na qual uma posição flectida do pulso com a contracção dos tendões flectores leva a valores elevados e eventualmente perigosos de pressão no nervo mediano. Este estudo revela uma interacção que pode ser uma das contribuições significativas para o aparecimento do CTS. Outros estudos apoiam este anterior e demonstram que a actividade de teclado eleva a pressão no túnel cárpico, independentemente da posição do pulso [11,12,13].

Embora se reconheça que existe uma relação entre o acto de dactilografar e o aparecimento de patologias como o CTS ou tendinites, não existem estudos relacionando as forças repetitivas aplicadas ao teclado durante largos períodos de tempo e as consequências que esta tarefa repetitiva pode ter ao longo do tempo, ao nível da pressão no túnel cárpico, na fadiga muscular e em outros parâmetros que contribuam para o aparecimento de desordens músculo-esqueléticas.

A relevância do estudo da força aplicada ao teclado ao longo do tempo e a sua relação com factores conhecidos por desencadear MSDs, como o conhecimento da postura do sujeito, a pressão do túnel cárpico, dados provenientes da actividade muscular (EMG), ou outros, podem contribuir para o conhecimento dos mecanismos que geram estas patologias, e para a melhoria da qualidade de vida através do desenvolvimento e melhoramento dos equipamentos relacionados com as tarefas de dactilografar, tornando-os mais ergonómicos.

Os Equipamentos Existentes para Estudar a Força Aplicada ao Dactilografar

Resultados obtidos por Rempel et al. (1994) determinam que o pico de força típico, aplicado por um sujeito ao dactilografar, varia entre 1,6N e 5,3N. Valor que depende quer das características únicas do sujeito em estudo, quer de propriedades intrínsecas do teclado em utilização. [14]

Para efectuar estudos deste género foram desenvolvidos alguns equipamentos, tendo todos eles como base uma luva onde são dispostos os elementos sensores de força. A espessura da luva e a colocação dos sensores, assim como a tecnologia utilizada variam bastante em cada caso, revelando a especificidade da criação dessa instrumentação com vista à sua utilização num ensaio bem definido.

Em [15] desenvolveu-se uma luva fina para testes clínicos, com elementos sensíveis de fibra óptica. Estes possuíam uma resolução de 0,4N e uma gama dinâmica de 2N a 95N. Outros métodos mais elaborados [16] procuram criar uma malha artificial de fibras ocas, calculando a força através de alterações na capacidade em pontos de

cruzamento das fibras. Também tirando partido da variação da capacidade, o estudo [17], calcula o valor da força explorando uma relação não-linear entre a força aplicada e a alteração da frequência de oscilação de uma onda quadrática. A alteração de frequência da onda reflecte por sua vez a variação do valor da capacidade. Este é um método que utiliza um gerador de ondas quadráticas, e traduz-se num modo eficaz de converter uma alteração do valor da capacidade num sinal. Outro equipamento encontrado na literatura [18] utiliza uma luva mais espessa e strain gauges de resistência eléctrica enquanto elementos sensores de força, apresentando uma resolução de 0,38N.

Um estudo brasileiro [19] tenta uma abordagem algo diferente para estudar a força aplicada ao teclar utilizando uma célula de carga na base de uma tecla vulgar e pedindo aos sujeitos para teclar uma sequência alfanumérica que repetidamente recorra à tecla sensível à força. Este estudo pode incorrer em alguns erros ao alterar a força necessária para activar a tecla sensível. Este facto faz com que o sujeito se veja obrigado a executar uma força imposta pela construção da própria tecla, ao invés do esforço natural que se pretende estudar. É demonstrado por [20] que o design das teclas influencia a força aplicada pelo sujeito ao teclar.

Motivação e Objectivos de uma Nova Instrumentação

Pretende-se com este trabalho desenvolver uma luva com sensores de pressão que permita registar a força aplicada no acto de dactilografar. Esta instrumentação deve superar as limitações associadas às instrumentações disponíveis no mercado, nomeadamente possuir uma resolução superior, ter uma gama dinâmica adequada ao tipo de estudos pretendido e eliminar, por construção, constrangimentos técnicos e motores encontrados noutros modelos, permitindo obter resultados fiáveis. Factores que retiram habilidade motora ao sujeito, como luvas espessas e/ou produzidas em materiais pesados, e a existência de vários cabos ou fios que para além de alterarem a destreza motora, ainda por vezes produzem efeitos psicológicos limitadores no sujeito, são factores a ter em conta no desenvolvimento desta instrumentação.

Os objectivos a atingir com o desenvolvimento desta instrumentação estão sumariados de seguida:

- Desenvolver um sistema, para aplicar numa luva, capaz de registar a força aplicada ao teclado pelos dedos indicador, médio, anelar e mínimo;
- Utilizar um sensor de pressão fiável, capaz de operar numa gama de forças de 0 a 10N;
- Desenvolver um sistema para aquisição e condicionamento do sinal, que prime pela portabilidade e leveza, eliminando vários cabos e fios dos designs normais, que alteram as habilidades motoras típicas do indivíduo;
- Utilizar um sistema de comunicação *Wireless*, para enviar os dados provenientes do sistema de aquisição e condicionamento para o PC para posterior armazenamento e tratamento dos dados;

Capítulo 2:

Desenvolvimento do Hardware.

O Sensor

Introdução às Medições de Força [1;2]

O principal elemento num sistema destinado a medir forças é o transdutor de força ou célula de carga (do inglês “*load cell*”). O transdutor possui um elemento sensível que produz um sinal eléctrico relacionado com a deformação sofrida (compressão ou alongamento), permitindo desse modo estabelecer a relação com a força aplicada no transdutor. Por fornecer uma resposta à deformação convencionou-se chamar a este tipo de transdutores *strain gauges*. De acordo com a tecnologia utilizada, estes dispositivos podem ser mecânicos, ópticos, acústicos e eléctricos. Também se podem classificar as *strain gauges* como dispositivos passivos ou activos. Os dispositivos passivos, como as resistências eléctricas *strain gauges*, ou as *strain gauges* que possuem elementos piezoresistivos, necessitam de uma fonte de energia externa para funcionar. Os dispositivos activos, como é o caso das *strain gauges* que utilizam elementos piezoeléctricos, não requerem uma fonte de energia externa para laborar.

As medidas de uma *strain gauge* são usualmente dadas em unidades de *microstrain* ($\mu\epsilon$), onde $1 \mu\epsilon = 1 \times 10^{-6} \epsilon$. A *strain* (ϵ) consiste numa unidade adimensional, que exprime a razão entre a variação do comprimento do objecto segundo um eixo e o seu comprimento inicial. A *strain* é portanto uma alteração de

comprimento normalizada ao comprimento inicial. O alongamento/compressão das *strain gauges* (alteração proporcional á força aplicada) quando registado em unidades de ϵ permite a comparação de diferentes tipos de *strain gauges*.

A resposta duma *strain gauge* é usualmente pós-processada de modo a obter-se o sinal de interesse. Para tal recorre-se a amplificadores, filtros e a um sistema de aquisição de dados. Um esquema típico da instrumentação utilizada em medições de força é apresentado em baixo, na Figura 1.

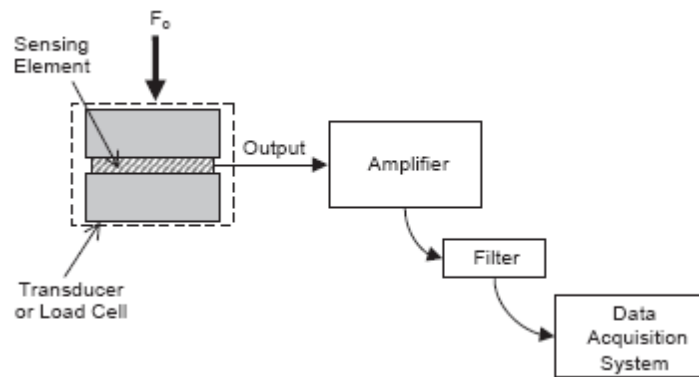


Figura 1 – Esquema básico de um sistema de medição de forças. F_0 é a força aplicada. Retirada de [1].

Os Tipos de Transdutores de Força [1;2;3]

Apresentam-se agora sumariamente as *strain gauges* eléctricas, mecânicas, ópticas e acústicas.

(a) *Strain Gauges* Mecânicas.

São o tipo de *strain gauges* mais antigo e, devido às suas grandes dimensões e à sua baixa exactidão, a sua utilização tem sido preterida em relação aos outros tipos de *strain gauges*. São utilizadas essencialmente em aplicações industriais, onde o tamanho do dispositivo não tem importância e onde o facto de não usar instrumentação electrónica associada é considerado vantajoso. O mecanismo associado a estas *strain gauges* baseia-se num sistema de alavancas que amplifica e permite a leitura do valor de interesse. Permite obter leituras com uma resolução de $500 \mu\epsilon$.

(b) Strain Gauges Ópticas.

A utilização de LASERs como fonte de luz colimada levou ao desenvolvimento de várias *strain gauges* ópticas, incluindo as baseadas em princípios de difracção e as *strain gauges* interferométricas. As mais simples funcionam com princípios semelhantes às do tipo mecânico, substituindo aqui as alavancas mecânicas por raios de luz. Permite obter valores com uma resolução de $0,1 \mu\epsilon$.

(c) Strain Gauges Acústicas.

As *strain gauges* acústicas utilizam na realidade duas *strain gauges* semelhantes, uma que entra em contacto com a força a medir (*gauge* de teste) e cujo valor da frequência de vibração é alterado pela força aplicada. E outra *strain gauge* que se encontra ligada a um micrómetro (*gauge* de referência), e cuja frequência é alterada propositadamente através da manipulação do micrómetro, até se igualar a frequência da *gauge* de teste, obtendo-se a partir do valor da variação do micrómetro, o valor força aplicada. Esta técnica acústica pode obter valores de $1 \mu\epsilon$ de resolução.

[3]

(d) Strain Gauges Eléctricas.

Existem vários tipos de transdutores de força, como já referi anteriormente, sendo mais utilizados os do tipo eléctrico. Este tipo de *strain gauges* explora propriedades como a capacidade, a indutância, a resistência e as propriedades dos semicondutores para obter uma resposta à deformação.

(d1) Capacidade e Indutância

A *strain gauge* que utiliza a capacidade como forma de medir a deformação aplicada fá-lo através de um condensador de placas paralelas onde a distância entre as duas placas é proporcional ao valor da capacidade. Logo, alterando a distância entre as duas placas aquando da deformação, o valor da capacidade do condensador também vai variar permitindo medir a deformação exercida. Também existem vários tipos de

strain gauges que utilizam a indutância, como é o caso do LVDT (*linear variable differential transformer*). Este dispositivo consiste num núcleo de ferro esférico em contacto com o que se pretende medir, rodeado por uma bobina primária e duas bobinas secundárias idênticas. O deslocamento do núcleo faz variar a tensão nas bobinas, o que permite calcular o deslocamento e a deformação sofrida.

(d2) Resistência

As *strain gauges* do tipo resistências eléctricas são fabricadas a partir de materiais condutores (metais) que alteram a sua resistência quando são deformados. Estão disponíveis sob as mais diversas formas, desde simples fios a filmes finos, sendo mais popular a forma em chapa. Neste último tipo, o fio é disposto em grelha maximizando a resistência eléctrica e minimizando o comprimento e a espessura do sensor. Para se obter informação acerca de forças com diferentes orientações angulares utilizam-se várias resistências deste tipo, orientadas em ângulos diferentes. (Ver Figura 2).

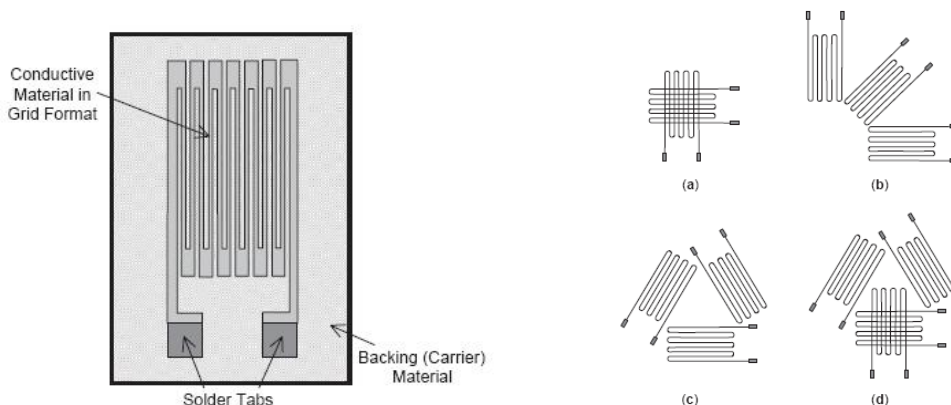


Figura 2 – No diagrama da esquerda encontra-se uma *strain gauge* do tipo resistência eléctrica em forma de chapa. No diagrama da direita encontram-se *strain gauges* do mesmo tipo dispostas de forma a poder obter informações angulares (a, b, c, d). Estes arranjos de *strain gauges* designam-se *rosettes*. Imagens retiradas de [1].

A sensibilidade das *strain gauges* do tipo resistência eléctrica é dada pelo “Gauge Factor” (GF), um valor adimensional que expressa a razão entre a variação relativa da propriedade mensurável (resistência eléctrica) com a variação relativa da dimensão do sensor (*strain*). Quanto maior for o GF maior a sensibilidade do sensor. A expressão do GF encontra-se em baixo:

$$GF = \frac{\frac{\Delta R}{R}}{\frac{\Delta L}{L}} = \frac{\Delta R}{R} \cdot \frac{L}{\Delta L}$$

O valor típico GF para uma *strain gauge* do tipo resistência eléctrica é aproximadamente 2. Factores como a temperatura, o material adesivo utilizado, e as propriedades do material condutor podem afectar a resistência medida, pelo que podem ser necessárias compensações. Como o valor de variação das resistências é muito pequeno, normalmente recorre-se à colocação deste tipo de sensor numa configuração em ponte de Wheatstone. Os transdutores de força que utilizam *strain gauges* do tipo resistências eléctricas encontram-se normalmente ligados a uma unidade estrutural (Ver Figura 3). Este tipo de transdutores de força exhibe uma sensibilidade linear elevada na gama elástica, boa exactidão e repetitividade, baixa histerese e boa durabilidade.

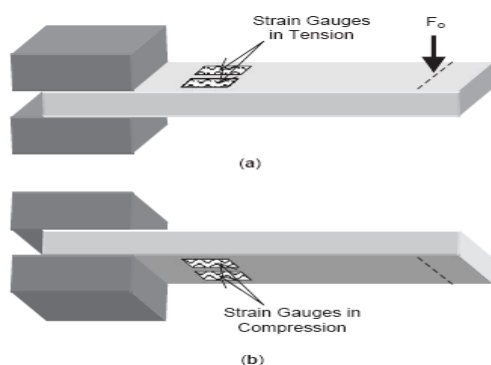


Figura 3 – *Strain gauges* ligadas a uma unidade estrutural. F_0 é a força aplicada. Na figura (a) as *strain gauges* vão sofrer uma tensão, enquanto na figura (b) vão sofrer uma compressão. Imagem retirada de [1].

(d3) Semicondutores

As *strain gauges* baseadas nas propriedades dos semicondutores são bastante utilizadas na produção de células de carga, transdutores miniaturizados, e outros dispositivos para aplicações que requerem a medição de deformações muito pequenas. São tipicamente construídas em silício ou germânio.

Os sensores baseados na tecnologia piezoresistiva dos semicondutores, um tipo de dispositivo passivo, têm uma reacção semelhante às *strain gauges* resistências eléctricas. Isto porque com a alteração de volume do elemento piezoresistivo varia também a resistência eléctrica desse elemento. Estes sensores apresentam um GF entre 50 e 200. As vantagens deste tipo de sensores são: elevada sensibilidade às

deformações, baixa histerese, boa durabilidade e tamanho reduzido. Têm como desvantagens as derivas da sensibilidade, pois são muito sensíveis à variação de temperatura, luz e humidade, para além de uma elevada não linearidade na relação resistência/deformação. As desvantagens descritas são colmatáveis em parte com uma compensação de software e/ou com uma construção apropriada.

Os dispositivos activos incluem a tecnologia piezoelétrica dos semicondutores. Neste caso a deformação do sensor leva à criação de cargas nas faces do elemento piezoelétrico. É usualmente comparado a um condensador com uma carga na ordem de grandeza do pico – Coulomb. Este dispositivo com uma impedância de saída elevada é normalmente ligado por um cabo de resistência elevada a um amplificador, para transformar o sinal de carga num sinal de tensão. Também existem sensores deste tipo com um amplificador incorporado, excluindo assim os problemas de ligação ao cabo, mas necessitando de energia eléctrica para o amplificador funcionar. Podem ser usados para medir sinais com frequências na ordem dos 100kHz. A perda exponencial de carga ao longo do tempo, nas interfaces do elemento piezoelétrico, faz com que seja quase impraticável realizar medições estáticas. No entanto com condicionamento e tratamento do sinal adequados é possível efectuar medições com uma frequência inferior a 1 Hz.

Crítérios de Selecção do Sensor de Força para esta Aplicação

Para a aplicação neste projecto é preciso ter em conta o tamanho do sensor, que deve ser bastante reduzido e leve de modo a que cause o mínimo de obstrução ao operador. Procuramos um sensor sensível à compressão. É também necessário ter em conta o erro de aplicação de uma carga fora do eixo de leitura do sensor. Existem sensores com soluções de engenharia para minimizar este erro (*off-axis loading*).

Os erros que afectam o desempenho do sensor (deriva da sensibilidade devido á temperatura, não linearidade, histerese, etc.) devem ser naturalmente os menores possíveis. Pretende-se então um sensor que vá de encontro às especificações ergonómicas do projecto, que possua as melhores características de desempenho

possíveis e que possua o menor custo possível associado. Um sensor que cumpra os objectivos traçados para o dispositivo final.

Estudo Comparativo das Tecnologias [4;5;6]

Apresenta-se agora uma tabela com a comparação entre as diferentes tecnologias de *strain gauges*:

Tabela 1 – Comparação de algumas propriedades das tecnologias utilizadas em *strain gauges*.

Tecnologia da <i>Strain Gauge</i>	PROPRIEDADES				
	Dimensão Típica	Resolução	Gauge Factor (GF)	Necessidade de suporte estrutural na aplicação	Medições quase-estáticas
Mecânicas	Grande/Média	500 $\mu\epsilon$	ND*	Sim	Sim
Ópticas	Média/Pequena	0.1 $\mu\epsilon$	ND*	Não	Sim
Acústicas	Grande/Média	1 $\mu\epsilon$	ND*	Não	Sim
Eléctricas:					
i) Capacitância e Indutância	Média/Pequena	ND*	15 - 30	Não	Sim
ii) Resistência	Média/Pequena	ND*	aprox. 2	Sim	Sim
iii) Semicondutores	Média/Pequena	ND*	50 - 200	Não	a) Piezoresistivo: Sim
*ND - Não Disponível/Não aplicável					b) Piezoeléctrico: Não

Procuramos assim o sensor mais leve possível, com uma dimensão inferior às dimensões típicas da ponta do dedo mais pequeno a ser monitorizado. Elimina-se de imediato a hipótese de utilizar a tecnologia mecânica e acústica devido às suas dimensões elevadas. As tecnologias que permitem uma miniaturização do transdutor/sensor de força são a eléctrica e a óptica.

Os sensores ópticos são sensíveis e exactos, mas são muito susceptíveis a interferências do meio. Funcionam bem em condições laboratoriais apropriadas, não sendo muito usados em ambientes menos controlados como o ambiente industrial. Os sensores ópticos apresentam custos elevados. Este conjunto de factores desaconselha a sua utilização nesta aplicação.

As *strain gauges* mais usadas no mercado são as de tecnologia eléctrica. As *strain gauges* que exploram a resistência eléctrica dos metais têm um GF muito baixo pelo que requerem uma grande amplificação do sinal. Também o facto de necessitarem, na maioria das vezes, de um suporte estrutural para a sua correcta aplicação leva a um aumento da dimensão e peso dos sensores que usam esta

tecnologia. São portanto preteridas quando se trata da miniaturização de sistemas de medição de força face a *strain gauges* que utilizam as propriedades dos semicondutores ou a capacidade e indutância eléctricas. No entanto as *strain gauges* que usam a capacidade e indutância eléctricas vêm também a sua aplicação limitada devido aos requisitos de montagem, à complexidade dos circuitos associados e à sua grande sensibilidade à vibração. [6]

Todos estes motivos levaram a uma utilização crescente das *strain gauges* que utilizam as propriedades dos semicondutores, que embora possuam um GF elevado apresentam elevada não linearidade e sensibilidade a alterações de temperatura. Estes factores podem ser colmatados em parte por construção, permitindo obter sensores fiáveis e sistemas miniaturizados para medição de forças. Entre os semicondutores, os piezoresistivos são os que permitem realizar medições a baixa frequência com maior exactidão e também os que permitem obter sistemas mais pequenos. Os piezoeléctricos necessitam de um amplificador incorporado para transformar o sinal de carga num sinal de tensão, ou em alternativa necessitam de cabos de elevada impedância para transportar o sinal até um amplificador distante. Esses cabos são por natureza largos e pesados o que iria certamente alterar a mobilidade do sujeito. O amplificador incorporado iria aumentar também a dimensão do sensor. Opta-se assim pela escolha da tecnologia dos piezoresistivos para as *strain gauges* deste projecto.

Estudo Comparativo dos Sensores Piezoresistivos [7,8,9,10]

Na procura da *strain gauge* piezoresistiva ideal para este projecto, tentou-se obter um compromisso entre dimensão/peso e as propriedades gerais de desempenho do sensor, tendo sempre em vista o ambiente em que será utilizado e os objectivos estipulados. Foram encontrados quatro possíveis sensores piezoresistivos, dos quais aplicando os critérios de menor dimensão/peso, linearidade, capacidade de compensação da aplicação de uma carga fora do eixo de leitura do sensor, e gama dinâmica de entrada mínima de 0 – 10 Newton, resultou na escolha do sensor *FSS Low Profile Force Sensors* fabricado pela *Honeywell*.

Apresenta-se de seguida uma tabela com a descrição sumária dos quatro sensores analisados:

Tabela 2 – Comparação dos sensores passíveis de serem utilizados neste projecto. **Nota:** ND – Não Disponível; As dimensões dos sensores são apresentadas em milímetros.

Sensor	Produtor	Sensibilidade	Gama de entrada	off axis loading	Dimensões (cxlxa)	Linearidade
FSS Low Profile Force Sensors	HONEYWELL	0.12 mV/g	0 - 1500 g	Sim	9.14x5.59x3,75	± 1,5 % FSS
HFD-500	HOKURIKU	20 mV/N	0 - 5 N	ND	7.0x5.2x2.9	± 3 % FSS
FSG-15N1A	HONEYWELL	0.24 mV/g	0 - 1500 g	Sim	12.7x8.0x10.3	± 1,5 % FSS
FS01/FS03 Force Sensors*	HONEYWELL	4,41 mV/g	0 - 680 g	Sim	25.1x17.27x8.26	± 1.0 % FSS

* Com amplificador incorporado

Propriedades do Sensor Seleccionado [7]

O sensor *FSS Low Profile Force Sensors* utiliza o silício como elemento sensível piezoresistivo, e é produzido através de micromaquinação. O projecto deste transdutor dispõe o elemento sensível num circuito em ponte de Wheatstone não compensada, permitindo assim uma saída estável em tensão para toda a gama de entrada. Este sensor é de baixo consumo, possui uma saída ratiométrica e não amplificada. A saída deste sensor deve ser medida como uma tensão diferencial.

A folha de especificações (Data Sheet) deste sensor encontra-se no anexo IV.

Calibração Estática no Sensor Seleccionado

Para comprovar algumas propriedades do sensor, ele foi montado num circuito de teste que inclui um amplificador operacional *INA126P* [11] configurado como amplificador diferencial. Este amplificador está configurado com o ganho mínimo permitido por construção do próprio, que é 5, e está alimentado com tensões de alimentação $V_s = \pm 15V$ através duma fonte de tensão DC. O circuito também possui um divisor de tensão para obter a partir dos 15V, os 5V necessários para alimentar o sensor, de acordo com as especificações deste. O circuito contém ainda um circuito tampão (buffer) criado a partir de um amplificador operacional *TL081CP* [12] configurado como um seguidor de tensão para alimentar o sensor com os 5V desejados e dois condensadores de 100 nF para minorar o ruído nas tensões de alimentação. A saída do amplificador operacional *INA126P* foi obtida e registada

através de um osciloscópio da *Pico*® modelo ADC – 200/20 com 8 bits de resolução e uma incerteza associada de $\pm 3\%$ da medição.

Procedeu-se á calibração estática do sensor para comprovar algumas propriedades descritas na *Data Sheet*. A calibração foi realizada tomando por referência “*A Code of Practice for the Calibration of Industrial Process Weighing Systems*”, nomeadamente a descrição do processo de calibração através de massas de referência. [13]



Figura 4 – À direita, foto da montagem utilizada para calibrar o sensor *FSS*. À esquerda, foto das massas de referência utilizadas para a calibração.

O conjunto de massas utilizado foi pesado com uma balança *PK – 601* da *Denver Instrument*, capaz de pesar um máximo de 600 grama com uma resolução de 0,1 grama. As massas utilizadas no processo de calibração estão disponíveis na tabela 3. A relação entre força e massa é dada através da lei do movimento de Newton que estabelece a relação $\mathbf{F} = \mathbf{ma}$, onde \mathbf{a} é a aceleração da massa. O peso de um objecto é então o produto da sua massa pela aceleração da gravidade, cujo valor padrão é $\mathbf{g} = 9,80665 \text{ m/s}^2$. Esta relação permite utilizar as massas como referência para calibrar o sensor de força, permitindo posteriormente a passagem para o valor correspondente em Newton utilizando o valor de \mathbf{g} . [14]

Tabela 3 – Massas de referência utilizadas.

Objecto	Massa em grama
Aplicador	21,8
Massas Grandes (MG)	
MG1	133,4
MG2	133,3
MG3	132,7
Massas Pequenas (MP)	
MP1	91,1
MP2	92,4
MP3	95,2
MP4	94,9

Para se efectuar a calibração estática recorreu-se a um suporte universal com uma garra, permitindo assim aplicar as massas de referência directamente no sensor utilizando o aplicador das massas (dispositivo visível na Figura 4 ao lado das massas de referência). A montagem criada tenta garantir o equilíbrio das massas em cima do sensor e também que a totalidade da massa de referência seja suportada pelo sensor. Na calibração com massas de referência foram obtidos dez pontos à temperatura ambiente, numa gama de 0 até 571 grama. O sensor está alimentado a 5V, e a sua resposta é condicionada pelo circuito de teste descrito anteriormente. O valor da resposta do sensor é obtido através do osciloscópio e é considerada como resposta do sensor o valor médio registado no osciloscópio num intervalo de 80 milisegundos após a aplicação da massa de referência. A tabela com os resultados da calibração estática do sensor seleccionado e o gráfico correspondente, assim como a tabela com o teste da repetitividade, encontram-se na Tabela 4.

Tabela 4 – Resultados da calibração estática do sensor FSS.

Massa Aplicada (gramas)	Valor médio registado (mV)	Intervalos massa (gr)	Intervalos (mV)	Sensibilidade (mV/g)
0	27,88	21,8	13,19	0,605045872
21,8	41,07	91,1	57,7	0,633369923
112,9	98,77	42,3	24,93	0,589361702
155,2	123,7	90,4	57,2	0,632743363
245,6	180,9	54,9	30	0,546448087
300,5	210,9	78,4	54	0,68877551
378,9	264,9	57,5	36,9	0,64173913
436,4	301,8	75,9	45,7	0,602108037
512,3	347,5	58,7	38	0,647359455
571	385,5			
Sensibilidade média:				0,620772342
				0,124154468

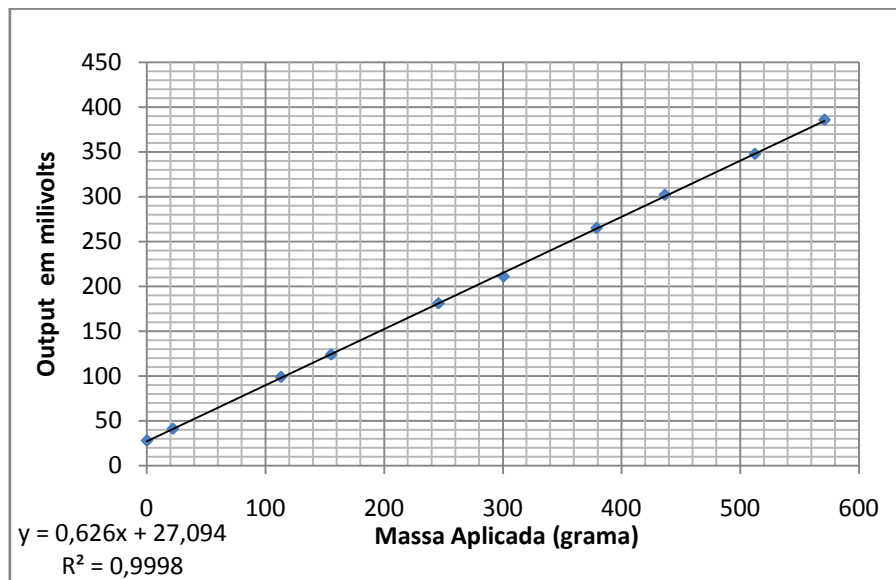


Gráfico 1 – Resultados da calibração estática do sensor FSS.

Tabela 5 – Estudo da repetitividade do sensor FSS.

Repetitividade a 300.2 g (mV)	
Ensaio 1	216,8
Ensaio 2	213,3
Ensaio 3	212
Ensaio 4	213,6
Ensaio 5	213,3
Ensaio 6	214,2
Ensaio 7	211,4
Ensaio 8	212,4
Ensaio 9	213,6
Desvio Padrão (mV):	1,551612065
Repetitividade (grama):	6,928223

A não-linearidade foi calculada com base nos dados apresentados na Tabela 4, através da obtenção de uma recta (BFSL) por aplicação do método dos mínimos quadrados, e calculando o desvio máximo entre o valor de tensão previsto pela recta e o valor de tensão observado. A não-linearidade apresenta um valor de $\pm 0,4785\%$ FSS, valor dentro do estimado na Data Sheet de $\pm 1,5\%$ FSS. A repetitividade foi calculada com base na definição do British Standard Institution [15], que estabelece que em 95% dos casos, medições repetidas desta mensuranda realizadas num conjunto de condições, que inclui o mesmo procedimento de medição, o mesmo operador, o mesmo sistema de medição, as mesmas condições operativas e a mesma localização, não diferirão mais do que 6,93 grama. Valor este que se apresenta dentro dos limites esperados de ± 10 grama que constam da folha de especificações do sensor. A sensibilidade provou também estar de acordo com as especificações com um valor igual a 0,12 mV/g.

Seleção do Sistema de Aquisição de Dados e do Sistema de Comunicação *Wireless*

Propriedades do Sistema de Aquisição de Dados [16;17;18;19;20]

Procuramos um sistema de aquisição de dados (SAD) que cumpra os critérios de portabilidade formulados para este projecto. Deve portanto ser um sistema leve, portátil, alimentado por bateria(s) e que cause o mínimo de obstrução ao operador. Um sistema que permita a recolha de sinais dos sensores de força, o acondicionamento desses sinais analógicos diferenciais, a sua digitalização e envio posterior por transmissão *wireless* para um computador pessoal (PC) onde serão armazenados, analisados e apresentados.

O sistema deve então possuir por cada sensor de força, dois canais para entradas analógicas do tipo diferenciais, capazes de suportar uma gama de entrada de 0 a 180 mV. A taxa de amostragem típica deste tipo de sinais de força é baixa, estando relacionada com a velocidade máxima de um sujeito a dactilografar. Para se obter um sinal representativo a taxa de amostragem deve seguir o teorema de Nyquist, que de forma resumida estabelece o valor da frequência de amostragem de um sinal como sendo o dobro da frequência máxima registada nesse mesmo sinal. Isto claro, para que o sinal seja perfeitamente reconstruído. A velocidade máxima a dactilografar está registada no “THE GUINNESS BOOK OF WORLD RECORDS”, e foi atribuída pelo feito de teclar 37 500 vezes em 50 minutos, atingindo uma velocidade máxima de 212 palavras por minuto num teclado DSK (*Dvorak Simplified Keyboard*). Estes dados apontam para uma velocidade de 12,5 teclas por segundo, pelo que seguindo o teorema de Nyquist, a frequência de amostragem deve ser no mínimo o dobro desta registada, ou seja, no mínimo 25 S/s (amostras por segundo).

O ADC incluído no sistema deve ter o melhor desempenho possível com a maior resolução possível, isto, para permitir uma boa resolução final no valor de força adquirido. Relembro que o sensor escolhido tem, quando alimentado a 5V, uma sensibilidade típica de 0,12 mV/g. Portanto, para se tirar proveito da totalidade da

gama de entrada do ADC, poderá ser necessário amplificar previamente o sinal, pelo que deve ser tido em conta o ganho nos sistemas de aquisição para uma escolha adequada do ADC.

A instrumentação a desenvolver deve ser capaz de monitorizar em tempo real a força exercida pelo sujeito, mas também guardar esses dados para posterior análise. Os ensaios típicos para os quais esta instrumentação está a ser projectada, consistem em estudos clínicos de 150 minutos de utilização, definidos por 3 partes (de 50 minutos) onde o sujeito irá transcrever um texto predefinido, com intervalos de 10 minutos entre as 3 partes. A autonomia da instrumentação nesse espaço de tempo deve ser um factor tomado em consideração.

SADs Estudados e a Solução Encontrada [21;22;23;24;25;26]

Realizou-se uma pesquisa de mercado na tentativa de encontrar um Sistema de Aquisição de Dados com as funcionalidades desejadas. Foram encontrados dois tipos de Sistemas de Aquisição de Dados. Um dos tipos, os chamados Sistemas de Aquisição em tempo real, permite a aquisição e utilização simultânea dos dados e em tempo real (alguns destes também permitem o armazenamento dos dados para posterior processamento). O outro tipo consiste nos sistemas que adquirem os dados e os armazenam imediatamente para análise e uso posterior (envolvem necessariamente o uso de algum tipo de memória), designados por *Data Loggers*. [21]

Na categoria de Sistemas de Aquisição de Dados em tempo real foram estudadas as soluções portáteis da *National Instruments™* (NI), nomeadamente as *NI USB-6008*, *NI USB-6009*, *NI USB-621x* e as *Portable E Series*. Embora sejam soluções portáteis e de dimensão relativamente pequena, não utilizam comunicação *wireless* e são alimentadas por cabo USB ou por outro tipo de cabo de ligação (ex: BNC) e não por baterias, factores que retiram a mobilidade desejada e que as excluem como sistemas de aquisição de dados para este projecto. Soluções da NI com comunicação *wireless*, como a *NI Wi-Fi Data Acquisition*, apresentam-se com dimensões demasiado grandes para a sua utilização neste projecto.

Na categoria de *Data Loggers* foram estudadas as soluções da *dataTaker*[®] e da *DataQ*[®], dispositivos que embora sejam alimentados a bateria, têm limitações quanto à taxa de amostragem, que normalmente é de 1 a 2 amostras por segundo. Existem no entanto alguns dispositivos que adquirem a 100 S/s, e até 500 S/s, uma melhoria que vem associada a um custo bastante elevado. Também a memória destes dispositivos ronda usualmente as 130.000 amostras, o que a uma taxa de 25 S/s (a taxa de amostragem prevista) dá para um pouco menos que 1,5 horas de funcionamento contínuo. Outra questão que limita a utilização dos Loggers é que normalmente não existe uma preocupação com as dimensões do aparelho, são grandes, sendo que os modelos com maiores Logging rates (taxas de amostragem) até apresentam ecrãs para mostrar os recursos desses Loggers. O custo normal associado a estes dispositivos também é alto.

Após alguma pesquisa concluiu-se que a solução final passa pela criação de um Sistema de Aquisição de Dados específico para este projecto, uma vez que não foi encontrada no mercado uma solução satisfatória capaz de assegurar todas as funcionalidades desejadas com a dimensão adequada. Esta solução própria passa pela possível criação de uma PCB (*printed circuit board*) com os dispositivos electrónicos que permitam condicionar, digitalizar e enviar para o PC os dados provenientes do sensor.

Introdução ao conceito de Microcontrolador e a escolha do PSoC[®] [27;28]

Um microcontrolador ou MCU consiste num circuito integrado que contém um microprocessador e todos os periféricos necessários para o seu funcionamento. Esses elementos periféricos podem ser memórias voláteis e não voláteis de leitura, escrita, ou de armazenamento permanente de dados (memórias EEPROM), osciladores, fontes de relógio (cristais de sincronização ou outros) e interfaces de entrada e saída de dados (UART, I2C...). Em alguns casos já se incluem nos periféricos conversores analógico/digital (ADC) e/ou conversores digital/analógico (DAC). Os microprocessadores trabalham a frequências de poucos MHz ou inferiores, possuem baixo consumo, o que adicionado à capacidade de entrar em modo de espera (*Sleep*)

torna-os dispositivos ideais para operar em sistemas portáteis remotos. São alimentados usualmente a 3,3V ou 5V e têm uma dimensão reduzida sendo utilizados em instrumentações de pequena dimensão. Existem vários produtores de microcontroladores, desde os mais simples aos mais complexos (*AMCC, Atmel, Cypress, MicroSystems, Freescale Semiconductor, Fujitsu, Holtek, Intel, Microchip Technology*, entre outros).

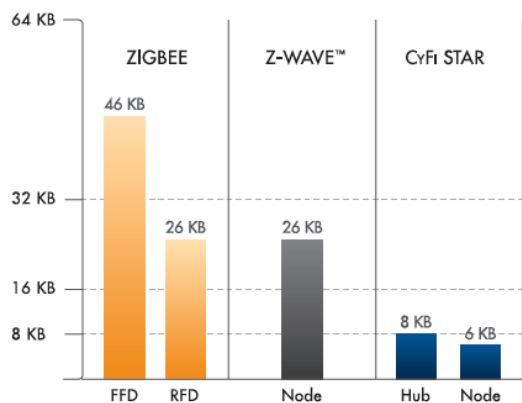
Uma nova tecnologia no universo dos sistemas embutidos foi criada em 2000 pela *Cypress Semiconductor Corporation*, uma tecnologia que integra num só chip conceitos provenientes de outros dispositivos digitais como microcontroladores, DSPs e FPGAs. A essa tecnologia a *Cypress* deu o nome de PSoC® - *Programmable System on Chip*. Como o próprio nome indica trata-se de um microcontrolador que contém tudo o que é preciso para construir um sistema completo de aquisição, tratamento e comunicação de dados. Os periféricos incluídos como a memória Flash e RAM, portas, blocos configuráveis analógicos e digitais, comunicam com o microprocessador incluído como sistemas separados embora estando todos convenientemente acondicionados num só chip, no PSoC®. O PSoC® apresenta-se como uma solução de pequena dimensão, que através do sistema de blocos configuráveis implantado e das interconexões entre estes blocos, permite obter um sistema bastante versátil e miniaturizado que inclua por exemplo: amplificadores, portas série, geradores PWM, temporizadores, ADC, DAC, entre outros. Por estas propriedades o PSoC® afirma-se como uma base ideal para o desenvolvimento do sistema de aquisição, tratamento e comunicação de dados requerido para este projecto. Este chip permite condicionar o sinal analógico proveniente do sensor de força, transformá-lo num sinal digital, e como iremos ver de seguida, permite implementar o protocolo de comunicação *Wireless* (CyFi™). Tudo num só chip, formando um sistema pequeno, leve e com baixo consumo.

O Sistema de Comunicação *Wireless* Escolhido [29]

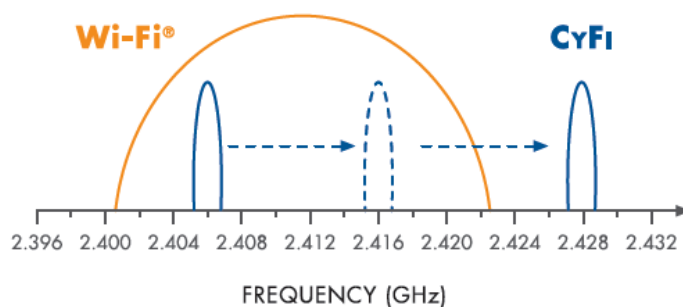
A *Cypress* desenvolveu uma solução para criar redes *Wireless*, designada por *Cypress CyFi™ Low-Power RF*, que é prática e de simples implementação em conjunto com o PSoC®. Esta solução *wireless* actua na banda ISM de 24 GHz, criando uma rede

sem fios com uma topologia em estrela, combinando apenas o PSoC®, onde corre um protocolo de comunicação leve, e um *transceiver* CyFi™ que aplica a tecnologia de modulação *Direct Sequence Spread Spectrum* (DSSS), para obter imunidade às interferências, maior alcance e uma co-localização melhorada, constituindo assim uma solução simples, viável e energeticamente eficiente.

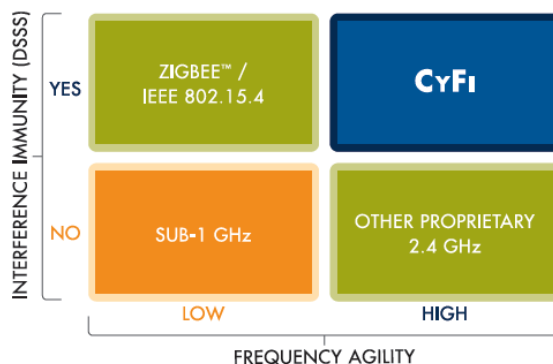
O protocolo de comunicação CyFi™ permite formar uma rede em estrela com um máximo de 250 nodos por *hub*. Ocupa pouco espaço de memória, factor visível quando comparado com outros protocolos de comunicação *wireless* (ver figura 5a). O espaço ocupado varia consoante a implementação final: um hub ocupa 8Kbytes enquanto um nodo ocupa entre 4,5Kbytes e 7,5Kbytes. Suporta comunicação assíncrona e bidireccional, assim como dispositivos de alta latência (alimentados a bateria) e de baixa latência. As redes que utilizam esta tecnologia, quando detectam interferências variam constantemente o canal de comunicações onde operam, a taxa de dados na comunicação e a potência emitida, até encontrarem um canal estável sem interferências para continuar a comunicação. Esta capacidade está exemplificada na Figura 5b, onde a fonte de interferências é outra rede, neste caso uma rede Wi-Fi®. Como as comunicações da tecnologia CyFi™ apenas ocupam 1MHz de largura de banda, existem 5 vezes mais canais para escolher que nas tecnologias usuais que se regem pelos padrões IEEE 802.15.4. (ver Figura 5c)



(a)



(b)



(c)

Figura 5- (a) Espaço de memória ocupado por diferentes tecnologias *wireless*; (b) Resposta do CyFi™ às interferências do meio; (c) Comparativo entre tecnologias da imunidade às interferências e agilidade na alteração do canal de comunicação; Direitos das imagens detidos por: © 2008 Cypress Semiconductor Corporation.

Como já foi referido, o *transceiver* CyFi™ opera na banda ISM dos 2,4GHz e utiliza modulação DSSS. Este tipo de modulação permite evitar a corrupção dos dados durante a transferência, e é realizada naturalmente antes da transferência dos dados. A desmodulação é realizada por outro *transceiver* na extremidade receptora, recuperando-se assim os dados originais (ver Figura 6). Este dispositivo opera com uma alimentação de 1,8V a 3,3V e é capaz de atingir taxas de comunicação de 250kbps/125Kbps (com modulação DSSS).

Por fim, a ligação em rede destes dispositivos CyFi™ permite minimizar os períodos de consumo de picos de corrente e maximizar os períodos de *sleep*. Por vezes isso significa transmitir dados o mais rápido possível e outras vezes accionar o DSSS e aumentar a potência das ondas rádio, a fim de eliminar a necessidade de retransmissões. Estas decisões são realizadas em tempo real pelo *transceiver*, permitindo uma maior economia energética.

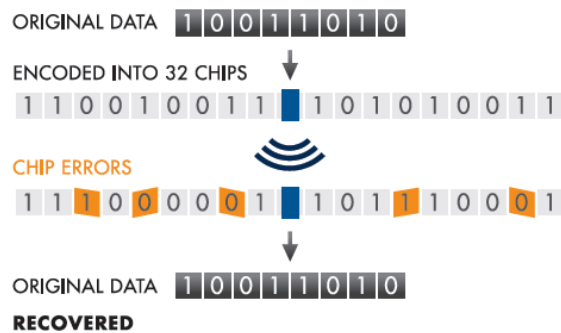


Figura 6 – Esquema da modulação por *Direct Sequence Spread Spectrum* (DSSS); Direitos da imagem detidos por: © 2008 Cypress Semiconductor Corporation.

Esquema do Projecto

Apresenta-se agora na Figura 7 um esquema resumindo a estrutura da instrumentação deste projecto com a especificação das escolhas que foram sendo tomadas:

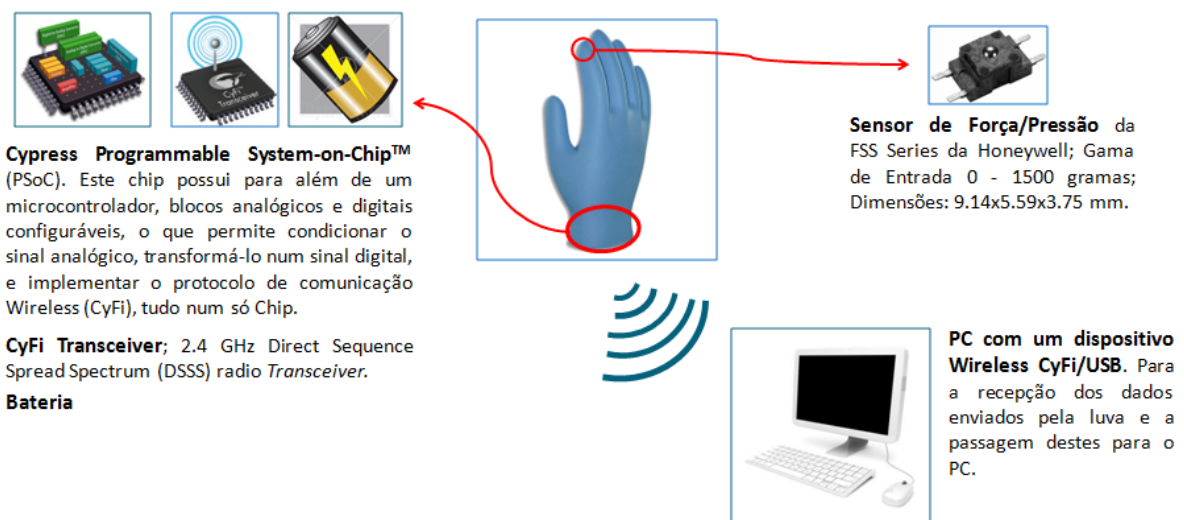


Figura 7 – Esquema da instrumentação;

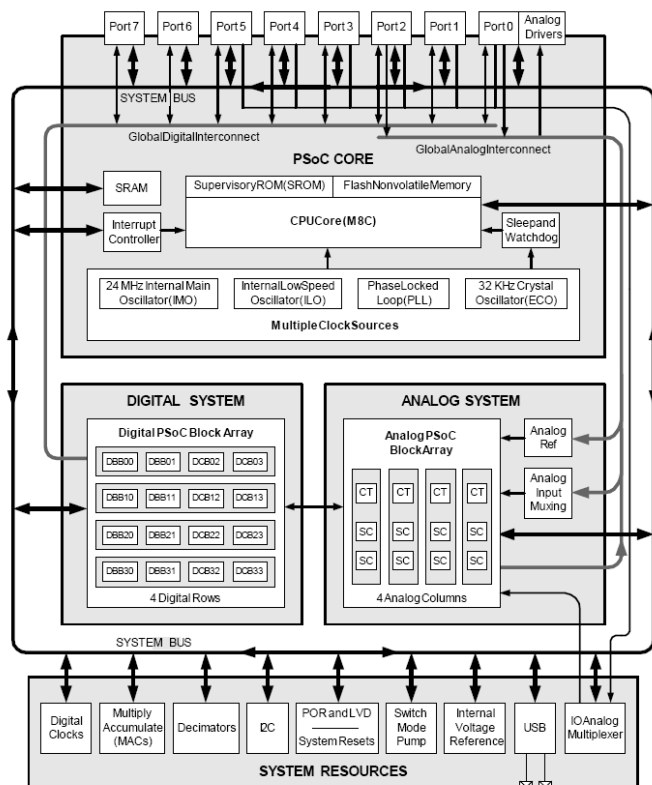
O PSoC®

A Arquitectura Geral do PSoC® [28;30]

Vamos agora observar mais minuciosamente a estrutura do PSoC®. Como já foi introduzido, o PSoC® consiste essencialmente num conjunto de sistemas analógicos e digitais independentes com interconexões programáveis, sistemas estes que são constituídos respectivamente por blocos de circuitos analógicos e por blocos de lógica digital de configuração programável. Os blocos analógicos e digitais, assim como as suas ligações são acondicionados no mesmo *chip* juntamente com um microprocessador e os seus periféricos (microcontrolador M8C), somando-lhes ainda alguns recursos adicionais. Forma-se deste modo uma unidade de processamento de sinal de pequena dimensão e elevada versatilidade. Este tipo de arquitectura permite ao utilizador final obter um sistema personalizado que vá de encontro às suas necessidades.

O diagrama de blocos da família dos microcontroladores PSoC®, retirado do *Technical Reference Manual* encontra-se na Figura 8.

Figura 8 – Diagrama de blocos sobre a arquitectura geral da família de microcontroladores PSoC®. Imagem retirada do *Technical Reference Manual* [30].



O Núcleo do PSoC® [27;29]

O núcleo do PSoC® (ver Figura 9) contém:

O núcleo da CPU (M8C) - Equipado com um microcontrolador M8C que pode operar com velocidades desde 93.7 kHz até 48 MHz, capaz de executar 4 MIPS, o núcleo da CPU do PSoC® consiste num microprocessador de 8-bits com arquitectura física Harvard e arquitectura lógica CISC. O microprocessador do M8C possui cinco registos internos, a saber: o Acumulador (Accumulator - A), o Index (X), o Contador de Programa (Program Counter - PC), o Ponteiro de Stack (Stack Pointer - SP) e o Registo de Sinalização (Flags - F). Os registos têm todos 8 bits com excepção do registo PC que possui 16 bits. O M8C contém ainda três espaços para endereçamento: a ROM, a RAM e os registos. A ROM contém a SROM e a memória Flash. Esta última encontra-se organizada em blocos de 64-bytes podendo alcançar um total de 64 Kbytes de memória. A ROM pode ser acedida a partir dos seus próprios endereços ou através do *data bus*. O espaço de endereçamento dos registos que são utilizados para configurar os blocos programáveis do PSoC® encontra-se dentro da memória RAM, e consiste em dois bancos de 256-bytes cada. A SROM por sua vez contém todo o código necessário para o iniciar PSoC®, configurar os circuitos e executar as operações sobre a memória Flash. As funções disponibilizadas pela SROM são chamadas pelo código armazenado na memória Flash ou pelo código criado pelos utilizadores/programadores finais do dispositivo.

A **SRAM** é acessível através do *bus* de endereços de 8 bits, pelo que o M8C apenas pode aceder apenas a 256 bytes de SRAM. Portanto para aumentar a SRAM disponível e preservar a linguagem *assembly* do M8C, a memória SRAM foi paginada em páginas de 256 bytes.

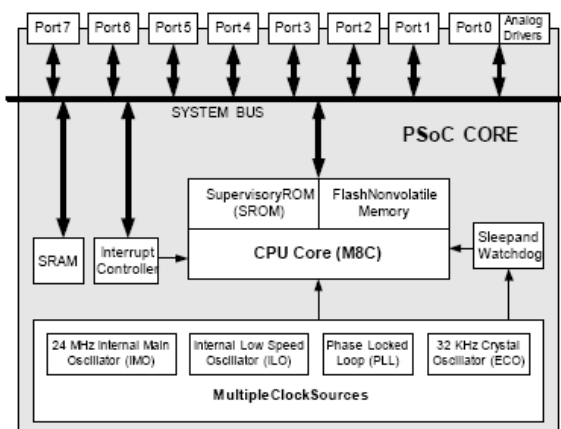


Figura 9 – Esquemático do núcleo do PSoC. Imagem retirada do *Technical Reference Manual* [30].

O Controlador de interrupções é utilizado para alterar de modo fácil a execução de um programa para um novo endereço. As interrupções estão sujeitas a uma activação organizada e controlada de acordo com a prioridade de cada uma, para que quando ocorram duas interrupções em simultâneo, seja primeiro activada a resposta à interrupção mais prioritária. A ordem das prioridades das interrupções é definida previamente.

Um temporizador de *Watchdog* e de *Sleep*. A operação de *Sleep* (modo de espera) e *Watchdog* envolvem a utilização de: uma fonte de relógio de 32 kHz, o temporizador de *Sleep*, o bit de *Sleep* do registo CPU_SCR0, um circuito de *Sleep* para sequenciar a entrada e a saída do modo de espera, um circuito para estabilizar a tensão de referência *bandgap* durante o modo de espera, e o temporizador *Watchdog*. O objectivo da operação de *Sleep* é reduzir o consumo energético do dispositivo o mais possível, através do abrandamento da CPU e da redução da alimentação do IMO, do módulo de memória Flash e da referência de tensão *bandgap*. Apenas permanecem ligados o oscilador de 32 kHz (interno ou externo), os blocos cuja fonte de relógio é este oscilador e o circuito de monitorização da tensão de alimentação. O sistema pode despertar do modo de *Sleep* através de uma interrupção ou de uma reinicialização, regressando ao funcionamento normal. O circuito do temporizador *Watchdog* é preparado para realizar um *reset* ao *hardware* após um intervalo de tempo pré-programado a não ser que a operação de *reset* seja executada antes pelo *firmware*. Serve essencialmente para evitar os bloqueios da execução do código, reiniciando todo o sistema sempre que se ultrapassa o tempo definido para a execução do *reset* pelo *firmware*.

Múltiplas fontes de relógio que incluem: O *phase locked loop* (PLL) que gera o relógio do sistema (SYSCLK) sincronizando-o com um cristal externo (ECO), permitindo obter um oscilador com 23.986 MHz. Este *external crystal oscillator* (ECO) de 32.768 kHz pode ser ligado ao PSoC® através de duas portas de I/O específicas para cada dispositivo, para se obter maior precisão e exactidão nas fontes de relógio internas; O *internal main oscillator* (IMO) gera duas fontes de relógio, a primeira é o SYSCLK que pode provir do oscilador interno de 24 MHz (um circuito RC com uma precisão de

±2,5%) ou como já vimos antes pode provir de um oscilador externo (ECO), e a segunda, o SYSCLKX2 que fornece sempre o dobro da frequência do SYSCLK; O *internal low speed oscillator* (ILO) é um oscilador com uma frequência nominal de 32 kHz, e é utilizado para gerar as interrupções no modo *Sleep*, para temporizar o *Watchdog* e alguns blocos digitais. Estas fontes de relógio juntamente com os divisores programáveis de fontes de relógio (VC1, VC2, VC3) permitem obter os temporizadores necessários para a maioria dos eventos no PSoC®.

Os GPIOs (General Purpose I/O) permitem a ligação do universo externo ao PSoC®, e deste modo à CPU, às unidades de memória e aos recursos analógicos e digitais do PSoC®. Os GPIOs contêm *buffers* de entrada, *drivers* de saída, memória para armazenamento do bit associado ao estado lógico do GPIO e a lógica necessária para preparar a configuração de cada pino para a sua ligação com o exterior (*drive mode*). Cada porta de I/O do PSoC® possui um registo associado de 8 bits, ou seja cada porta possui oito blocos idênticos de GPIO interligados (cada pino possui um bloco de GPIO) mas de modo a que cada bloco possua o seu endereço, assim como o seu bit de registo próprio. Cada bloco pode suportar entradas e saídas analógicas ou digitais, assim como vários *drive modes* e tipos de interrupções. Todas estas propriedades são configuráveis através de registos próprios para cada bloco de GPIO.

Uma das operações básicas com as portas GPIO consiste em receber ou enviar informação. Esse processo faz-se através dos registos PRTxDR (x representa o número da porta). Neste registo cada bit representa o estado de um pino da porta em questão, tendo sempre em conta o *drive mode* programado para cada GPIO. O diagrama de blocos para cada GPIO está descrito na Figura 10, assim como as configurações e esquemáticos de cada *drive mode*.

Os *drive modes* disponíveis são, sumariamente:

High Impedance Analog: este modo de condução é utilizado para conectar sinais analógicos. Neste caso todas as ligações entre o registo PTRxDR (responsável pela porta x) e o pino são desligadas, para que não existam interferências no sinal analógico que conduz;

Strong mode: é utilizado quando é necessário conectar o estado do registro PTRxDR diretamente para os pinos;

Pull up ou *Pull down:* são utilizados para garantir que as entradas de sistemas lógicos se ajustem a níveis lógicos esperados. No caso do *pull up* mantém-se uma tensão alta no condutor a que a entrada está ligada, no caso do *pull down* a tensão no condutor é mantida a zero. Este tipo de *drive modes* serve para ligar interruptores ou outros dispositivos que necessitem desta componente. As resistências associadas (visíveis na configuração 3 e 0, da Figura 10) definem o estado da entrada quando o botão ou outro dispositivo não é premido/ativado;

Open Drain: este modo é utilizado quando é necessário ligar vários dispositivos na mesma linha. O termo *open drain* refere-se ao terminal *drain* de um MOSFET. No estado lógico 0 a entrada encontra-se ligada á terra, enquanto no estado lógico 1 a entrada passa a possuir alta impedância.

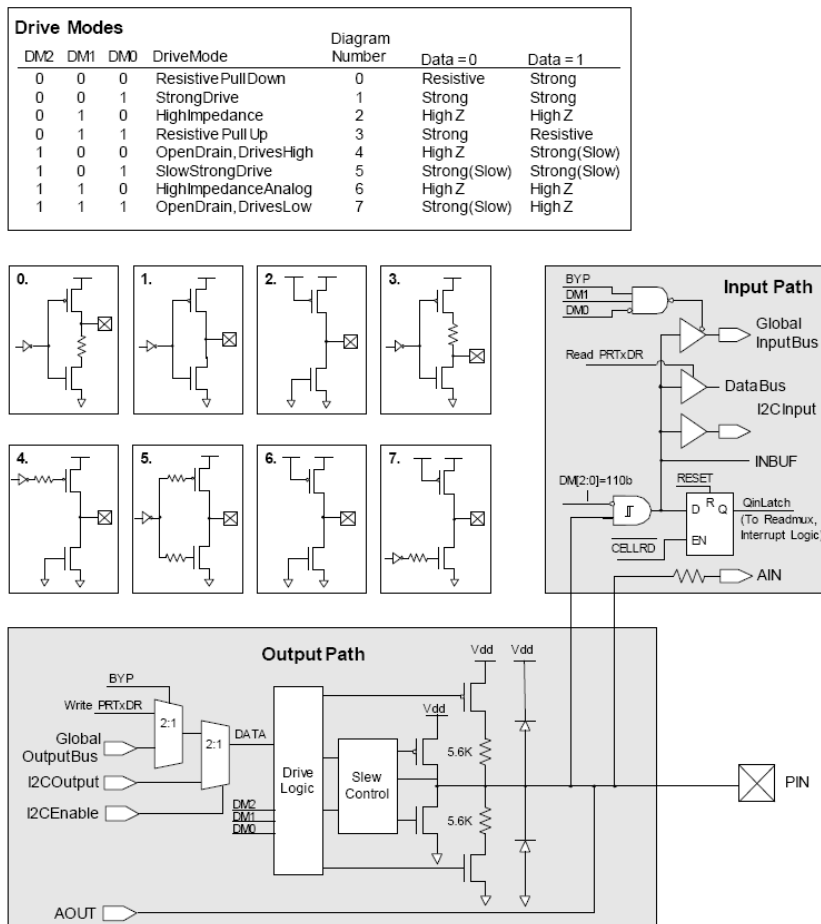


Figura 10 – Diagrama de blocos de cada GPIO, esquemático dos *drive modes*. Imagem retirada do *Technical Reference Manual [30]*.

O Sistema Digital [28;30]

O sistema digital é composto pelos blocos digitais programáveis e pelas interconexões entre estes blocos e com o M8C. O diagrama que mostra a arquitectura do sistema digital está representado na figura 11.

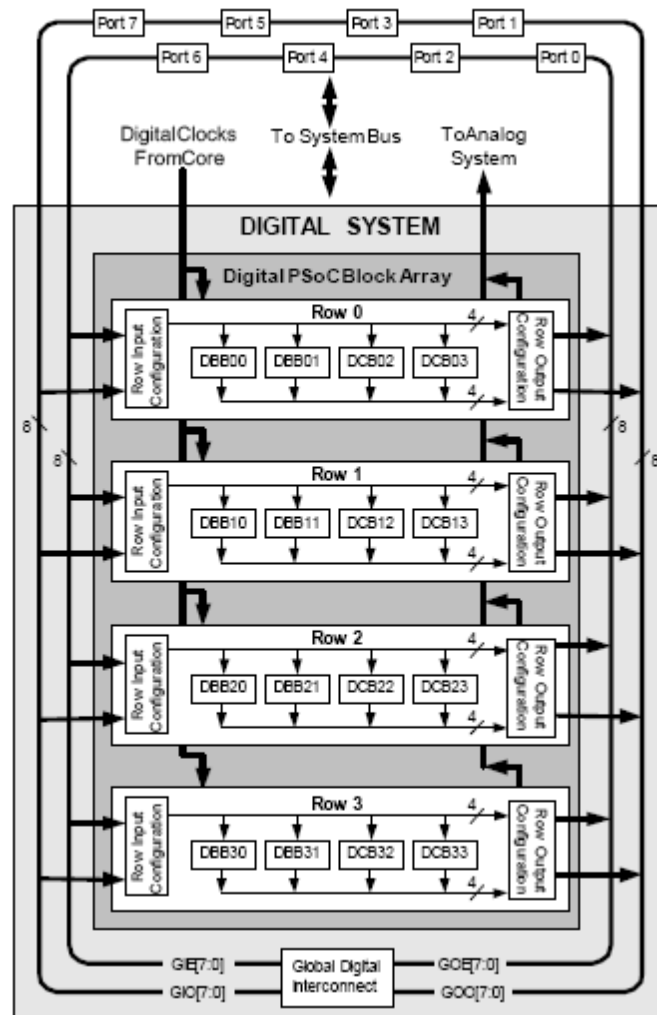


Figura 11 – Diagrama do Sistema Digital. Imagem retirada do *Technical Reference Manual* [30].

As interconexões digitais presentes no sistema digital do PSoc® podem ser de três tipos, as *Global Digital Interconnect* (GDI), as *Array Digital Interconnect* (ADI) e as *Row Digital Interconnect* (RDI). As GDI consistem em quatro *buses* de 8 bits cada. São estas que permitem a troca de dados entre os blocos digitais e os pinos das portas do PSoc®. As GDI estão divididas entre as que são destinadas a entradas ou a saídas

(*Global Input* e *Global Output*), e dentro desta categoria são divididas ainda entre as que ligam a portas pares ou a portas ímpares (GOO - *Global Output Odd*, GOE - *Global Output Even*, GIO - *Global Input Odd*, GIE - *Global Input Even*). As ADI asseguram as ligações entre as diferentes linhas da matriz de blocos digitais. As RDI estabelecem as ligações entre os blocos digitais de uma mesma linha e as ligações entre os blocos e as ADI e as GDI.

Os blocos digitais do PSoC® são dispostos em linhas de quatro blocos cada, sendo que o número de linhas de blocos digitais é variável de modelo para modelo. Todos os blocos digitais podem ser configurados para desempenhar qualquer uma destas cinco funções básicas: temporizador, contador, PWM - *pulse width modulator*, PRS - *pseudo random sequence*, ou CRC - *cyclic redundancy check*. Estas funções podem ser utilizadas quer configurando apenas um bloco digital quer ligando em cadeia vários blocos digitais, permitindo assim formar funções com mais que 8 bits. Existem dois tipos de blocos digitais: os blocos digitais básicos (DBB) que podem correr as cinco funções já descritas, e os blocos digitais de comunicação (DCB) que para além das cinco funções básicas, possuem a capacidade de desempenhar mais funções, como a de mestre ou escravo SPI ou a de transmissor/receptor UART. A função desempenhada por um bloco digital é completamente independente da função desempenhada pelos outros blocos. São necessários cerca de sete registos para estabelecer completamente a função e o estado de cada bloco digital. Cada bloco digital inclui multiplexadores de entrada, desmultiplexadores de saída, re-sincronizadores e circuitos de lógica digital para permitir o desempenho das funções descritas. A arquitectura de cada bloco digital está representada na figura 12.

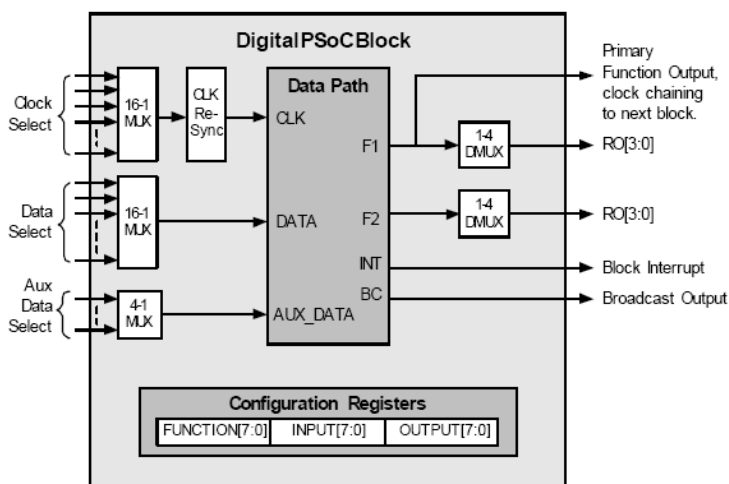


Figura 12 – Diagrama da arquitectura de um Bloco Digital. Imagem retirada do *Technical Reference Manual* [30].

Os Blocos Analógicos [28;30]

O sistema analógico é composto por colunas de blocos analógicos programáveis, referências de tensão para os sistemas analógicos, multiplexadores de entrada e condutores analógicos. Cada coluna de blocos analógicos contém: um bloco analógico contínuo no tempo (CT), designado também como bloco do tipo B (ACB), por não possuir condensadores comutados, ou seja por operar continuamente; e dois blocos analógicos com condensadores comutados, um do tipo C (ASC) e outro do tipo D (ASD). O diagrama de blocos do sistema analógico está representado na Figura 13.

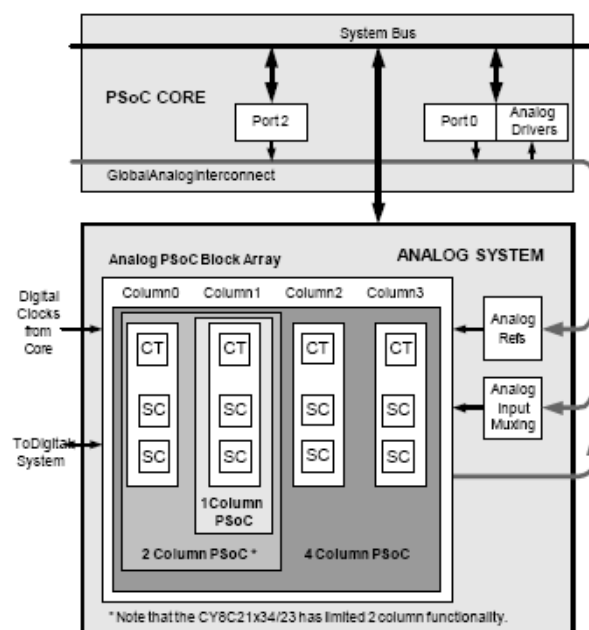


Figura 13 – Diagrama do sistema analógico. Retirada do *Technical Reference Manual* [30].

Cada bloco analógico tem várias entradas potenciais, pelo que os sinais analógicos podem ser provenientes de fontes externas, dos blocos analógicos contíguos a esse, ou ainda das várias referências de tensão internas. Embora sendo variável entre as várias famílias de dispositivos PSoC®, as portas que se encontram preparadas para receber sinais analógicos são normalmente as de ordem mais baixa (porta 0, 1, 2). Todos os blocos da mesma coluna analógica são temporizados a partir do mesmo relógio e cada coluna também partilha um único *bus* de dados de saída. Existem três tipos de saídas para cada bloco analógico, sendo que os blocos CT possuem ainda mais duas saídas. O primeiro tipo é através do bus analógico de saída

(ABUS). Existe apenas uma saída deste tipo por coluna e apenas um bloco pode utilizá-la num determinado tempo. É também a única saída que pode ser directamente roteada para um pino. O segundo tipo de saída dos blocos analógicos é o *bus* do comparador. Consiste num recurso digital e, do mesmo modo que a ABUS, também só existe um *bus* deste tipo por coluna conduzindo, num dado momento, apenas a saída de um bloco analógico. Este *bus* do comparador permite comparar as saídas seleccionadas entre duas colunas adjacentes. O terceiro tipo de saída analógica possível consiste nos vários multiplexadores do sistema analógico (NMux, PMux, RBotMux, AMux, BMux, e CMux), que aceitam a saída dos vários blocos analógicos (OUT) como entradas, permitindo ligar os blocos analógicos entre si. Os blocos CT possuem adicionalmente as saídas locais GOUT e LOUT. As propriedades dos blocos analógicos assim como das suas entradas e saídas são configuráveis e manipuláveis através dos registos associados.

Como o PSoC é alimentado apenas com uma tensão positiva, ele não dispõe de qualquer fonte de tensão negativa. A tensão de referência do PSoC® para as funções analógicas designa-se por AGND (*analog ground*) e é construída para ter um valor a rondar a metade do valor da alimentação do PSoC®, embora em alguns dispositivos estejam disponíveis outras opções para tensão de referência. Também é criada uma tensão de referência alta (RefHi) e uma baixa (RefLo), com vista a definir a gama de entrada de alguns blocos analógicos. Todas estas tensões de referência são roteadas até todos os blocos analógicos disponíveis.

Os blocos analógicos CT são construídos com entradas e saídas rail-to-rail, de baixo offset e com um amplificador operacional de baixo ruído. Possuem vários multiplexadores configuráveis que através dos bits dos registos específicos de cada bloco permitem obter topologias variáveis. Cada bloco CT contém também uma matriz de resistências de precisão controladas por um registo próprio, assim como um comparador de baixo consumo (Ver figura 14).

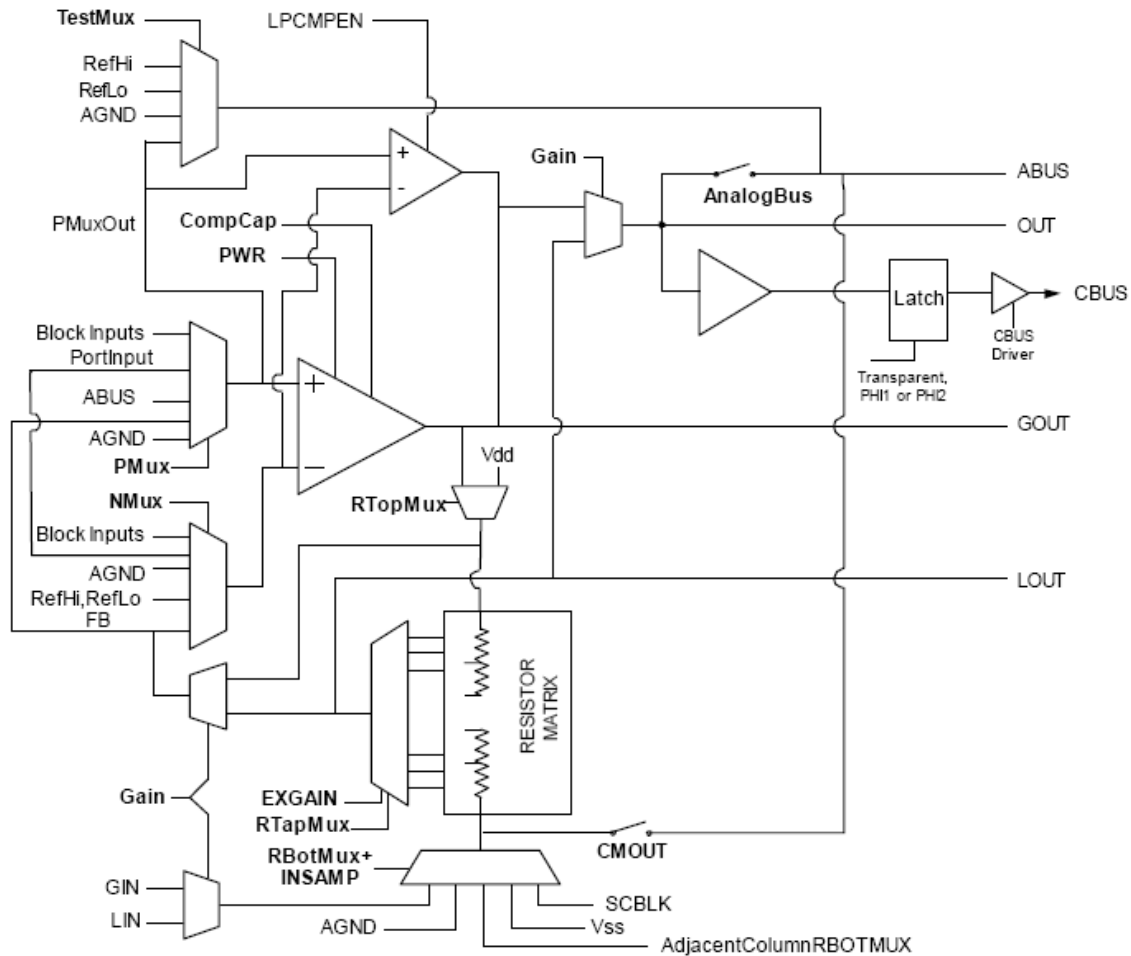


Figura 14 – Esquema do bloco analógico CT. Retirada do *Technical Reference Manual* [30].

À semelhança dos blocos CT, os blocos analógicos com condensadores comutados, ou blocos SC (*Switch Cap*), são construídos com entradas e saídas rail-to-rail de baixo offset, um amplificador operacional de baixo ruído, vários multiplexadores e também com um comparador. Como elementos diferenciadores, possuem quatro condensadores configuráveis ligados ao amplificador operacional. Cada um destes condensadores define um ramo específico, sendo os ramos designados por *A Cap*, *B Cap*, *C Cap*, e *F Cap*. À semelhança dos blocos CT, também aqui os multiplexadores controlam a topologia interna dos blocos SC. Adicionalmente existe um grupo específico de multiplexadores utilizados para o processamento do sinal que sincronizam actividades dentro do bloco, baseados nas fases do temporizador associado ao bloco SC (fases PH1 e PH2). Como tinha referido, cada coluna analógica possui dois tipos de blocos SC, os tipo C e os tipo D. As diferenças

estão relacionadas com as ligações internas do ramo do C Cap e na posição de alguns elementos internos, permitindo no caso do tipo D maior flexibilidade na comutação do B Cap. Esquemáticos dos blocos SC do tipo C e D encontram-se na Figura 15 e 16 respectivamente.

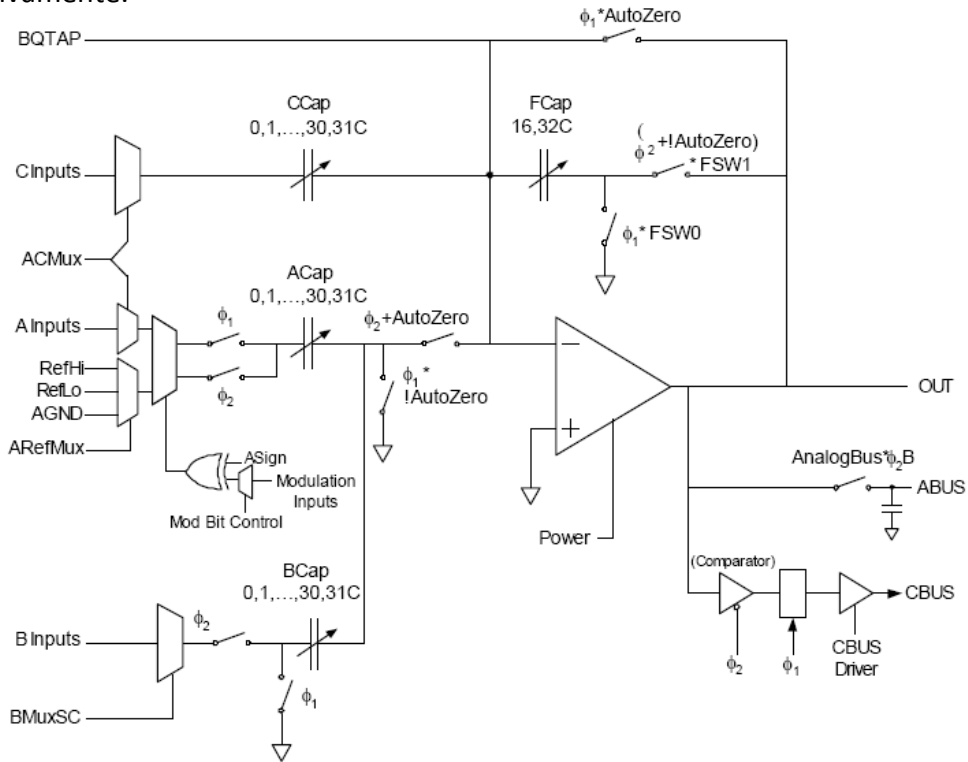


Figura 15 – Esquemático do bloco analógico SC do tipo C. Retirada do *Technical Reference Manual* [30].

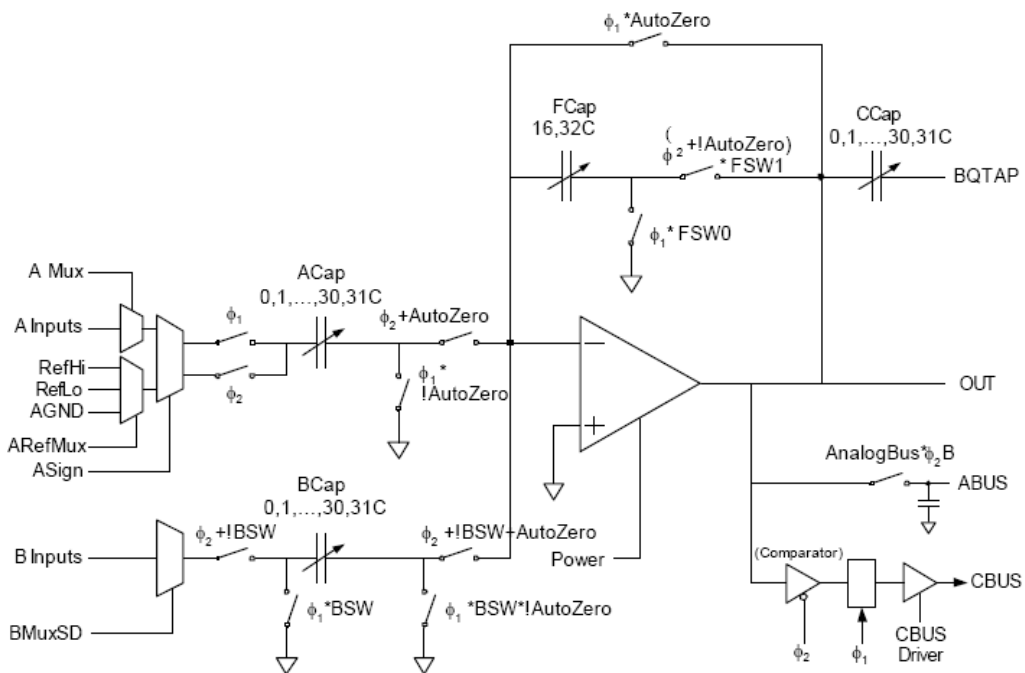


Figura 16 – Esquemático do bloco analógico SC do tipo D. Retirada do *Technical Reference Manual* [30].

As diferentes configurações de blocos analógicos possíveis, associadas às várias possibilidades de interligação entre os três tipos de blocos analógicos (CT e SC do tipo C e D), permitem obter uma grande variedade de funções de processamento de sinal como vários tipos de ADC, filtros, PGAs, entre outros.

Os Recursos Adicionais do PSoC® [28;30]

Os recursos adicionais do PSoC® dão mais capacidades ao dispositivo. Estes recursos adicionais variam entre as famílias do PSoC® e estão incluídos no PSoC®. A sua arquitectura geral é descrita na figura 17.

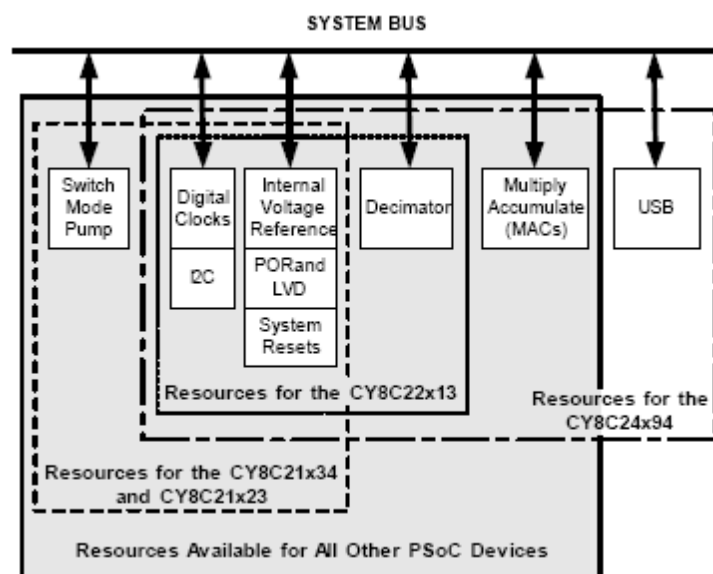


Figura 17 – Diagrama dos recursos adicionais do PSoC®. Retirada do *Technical Reference Manual* [30].

Os recursos disponíveis incluem: temporizadores digitais, registos multiplicador – acumulador (MAC - Multiply Accumulate), decimadores, I2C, referências de tensão, unidade de reinicialização do sistema, *Switch Mode Pump* (SMP), circuitos de *Power on Reset* (POR) e *Low Voltage Detect* (LVD), multiplexadores de entrada e saída e interface USB.

Como Desenvolver o Projecto e Programar o PSoC® [30;31;32]

Os dispositivos da família PSoC® são programados através do protocolo ISSP™ (*In-System Serial Programming*). Este protocolo permite, através de um dispositivo programador externo do tipo ICE (*In-Circuit Emulation*) e de software adequado programar ou reprogramar o PSoC® quando este já se encontra instalado num circuito

próprio (PCB). O programador, comunicando em série com o dispositivo alvo, fornece os comandos e o temporizador necessários ao procedimento de programação, e descarrega para o dispositivo PSoC® alvo o projecto criado num ficheiro hexadecimal.

O software que permite programar o PSoC® com recurso ao ICE designa-se por PSoC Programmer™ e encontra-se disponível no site da Cypress juntamente com o manual do utilizador [32].

A Cypress criou uma IDE (*integrated development environment*) que permite ao utilizador/programador configurar o seu dispositivo PSoC® de uma forma prática e simples poupando tempo e trabalho. Esta ferramenta chama-se PSoC Designer™ e junta ao IDE uma interface de programação visual onde todos os projectos com PSoC® são criados, editados, construídos e testados. Esta ferramenta suporta dois tipos de níveis de desenvolvimento do projecto dos quais se deve optar por apenas um. O chamado de nível do *chip* permite actuar ao nível mais baixo do PSoC® permitindo aceder e especificar todas as características do dispositivo desde o roteamento e a gestão do uso de recursos, até ao desenvolvimento do *firmware*. O outro nível chama-se nível de sistema, e só permite actuar a um nível alto no PSoC® onde o utilizador apenas pode definir a arquitectura alta do dispositivo a fim de cumprir os objectivos desejados. Enquanto o utilizador adiciona blocos e funções predefinidas ao seu projecto, o PSoC Designer™ escreve o *firmware* automaticamente. Naturalmente ao criar um projecto ao nível do sistema perde-se a capacidade de alterar algumas propriedades, o controlo sobre o projecto diminui, e alguns recursos tornam-se inacessíveis. Este projecto de desenvolvimento de instrumentação foi realizado recorrendo a este *software* e foi desenvolvido no nível do *chip*.

O PSoC Designer™ possui um ambiente de desenvolvimento que permite ver a disposição interna dos blocos analógicos e digitais do dispositivo escolhido, assim como controlar e definir as ligações quer entre os diferentes blocos do PSoC®, quer entre os diferentes blocos e o universo exterior ao dispositivo. O utilizador apenas tem de seleccionar o bloco com a função desejada, rotear o sinal de interesse até ele e dele até ao próximo bloco ou pino e especificar as propriedades relacionadas com cada parte, isto é, o utilizador tem de definir a estrutura interna do PSoC® de acordo com o

projecto a realizar. O *software* configura automaticamente os registos associados a cada bloco, ligação ou pino, de acordo com as selecções do utilizador, ao mesmo tempo que este vai ligando e especificando as partes do PSoC® para a sua aplicação. A versão do PSoC Designer™ utilizada neste projecto foi o PSoC Designer™ 5.0. O ambiente de desenvolvimento é descrito em pormenor em [33], e encontra-se representado figura 18.

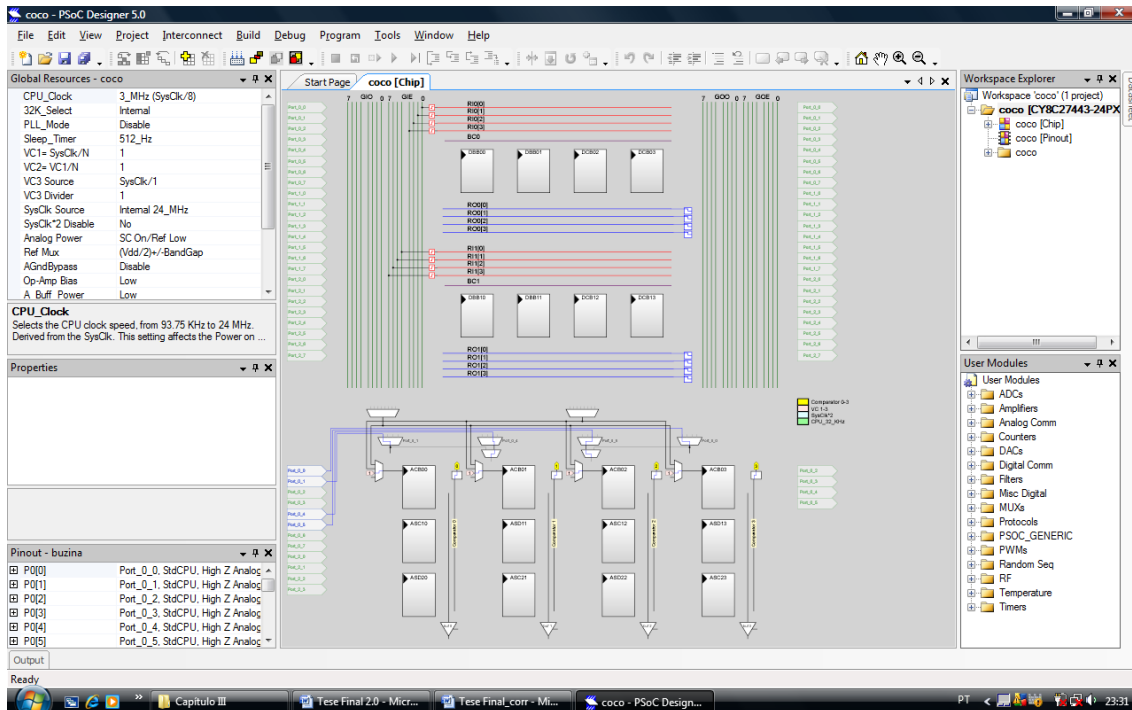


Figura 18 – Ambiente de desenvolvimento do PSoC Designer™.

A partir da especificação dos blocos, das ligações, das suas propriedades e da escrita do Firmware necessário para o bom funcionamento dos sistemas (analógicos e digitais) e para a implementação da solução desejada pelo utilizador, o PSoC Designer™ cria um ficheiro hexadecimal que contém toda a informação do projecto desenvolvido pelo utilizador assim como código essencial (pré-definido) para a correcta operação com o PSoC®. Este ficheiro hexadecimal irá posteriormente programar o dispositivo PSoC® do modo já descrito anteriormente.

A Plataforma de Desenvolvimento da Instrumentação [34]

Para tomar contacto com a tecnologia do PSoC® e para desenvolver a instrumentação final, foi necessário optar por uma plataforma de apresentação e

desenvolvimento desta tecnologia tendo em conta, no processo de escolha, critérios como o custo, o nível de conhecimentos do utilizador e as especificações do projecto a implementar no PSoC®. Relembro que a plataforma deve permitir o desenvolvimento de um sistema portátil, capaz de implementar um sistema de aquisição, tratamento e comunicação de dados através de um sistema sem fios. A plataforma escolhida foi o kit first touch CY3271 [34] que apesar de ser destinado a utilizadores principiantes, permite um primeiro contacto a tecnologia do PSoC®, inclui a tecnologia wireless CyFi™ e tem baixos custos associados, indo de encontro aos objectivos da instrumentação a desenvolver. Este kit CY3271 (ver Figura 19) inclui ainda exemplos de projectos desenvolvidos para PSoC®, destacando-se o exemplo *RF Expansion Card Ultra Low Power Temperature Sensor*, que serviu como base de aprendizagem para o desenvolvimento deste projecto. Uma das desvantagens deste kit consiste no facto dos dispositivos PSoC® nele incluídos estarem já embutidos numa placa PCB que visa a execução de demonstrações específicas. Isto significa que a maioria das portas dos PSoC® disponíveis encontram-se ligadas a dispositivos como termistores, LEDs, botões, sensores de luminosidade, entre outros, deixando poucas GPIOs livres para utilizar (apenas cinco GPIO livres). Mesmo assim apresenta-se como sendo um kit capaz de suportar o desenvolvimento desta instrumentação.



Figura 19 – As cinco placas disponíveis no kit first touch CY3271. [34]

O kit CY3271 possui cinco placas. Para o desenvolvimento deste projecto foram utilizadas apenas três dessas placas, a saber: A placa FTPC (Figura 20a), através da qual se programam todos os PSoC® do kit, o que significa que pode actuar como ICE. Possui também uma interface I2C para USB para realizar a ligação ao PC através de uma porta

USB. Contém dois PSoC®, um, ao qual não é possível aceder, que estabelece a comunicação com o PC e actua como programador e outro que comunica com o primeiro e que contém as ligações ao CyFi™ *transceiver*, permitindo a esta placa actuar como *Hub* no protocolo de comunicação sem fios CyFi™. É possível aceder e alterar este segundo PSoC® da placa FTPC; A placa FTRF (Figura 20b) que é alimentada pela placa AAA (Figura 20c). Estas duas placas permitem formar o Nodo do protocolo CyFi™.

É neste conjunto de placas, nomeadamente no PSoC® da placa FTRF, que se encontra a pedra basilar deste projecto. É onde se realiza a aquisição e condicionamento dos dados provenientes dos sensores de força. As cinco GPIO disponíveis na placa FTRF apenas permitem ligar dois sensores de força, uma vez que cada sensor de força necessita de duas entradas para se obter uma medição diferencial. A instrumentação a desenvolver consistirá na adaptação deste kit de demonstração da *Cypress* de modo provar o conceito para apenas dois sensores de força, sendo que, como se verificará, a instrumentação facilmente se estenderá para englobar os quatro sensores de força por mão que constam nos objectivos.

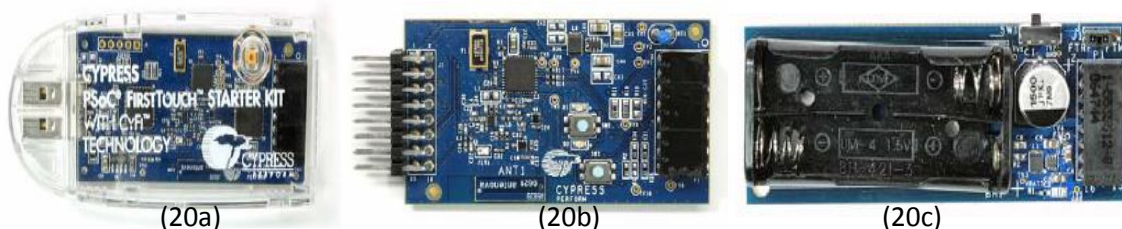


Figura 20 – As placas do kit CY3271 utilizadas no projecto: (20a) A placa FTPC; (20b) A placa FTRF; (20c) A placa AAA.[34]

O *hardware* deste projecto divide-se então em duas partes lógicas de acordo com o papel desempenhado no protocolo de comunicação *wireless* e pelas funções distintas desempenhadas no âmbito geral do projecto: o *Nodo* e o *Hub*.

O Hub.

O hardware do Hub [34]

Como já havia referido, o *Hub* da instrumentação a desenvolver baseia-se na placa FTFC do kit CY3271 da Cypress. Esta placa possui dois PSoC®, ambos CY8C24894. O primeiro actua como microcontrolador mestre providenciando a ponte entre o porto USB do PC e o porto I2C, de comunicação entre os dois PSoC®, e controlando a programação de todos os outros dispositivos PSoC® existentes no kit CY3271. O segundo PSoC® actua como escravo no protocolo I2C e encontra-se ligado a um *transceiver* CYRF7936 2.4GHz CyFi™, o que permite a este segundo PSoC® actuar também como *Hub* no protocolo de comunicação *wireless* CyFi™. A placa FTFC contém um conector de 16 pinos que permite ligar às outras placas do kit a fim de se efectuar uma transferência de dados normal ou a programação ISSP dos dispositivos PSoC® das outras placas. O PSoC® que actua como Hub também comunica com o software SCD da Cypress através da interface I2C com o PSoC® mestre. Em resumo, a placa FTFC estabelece a interface entre o PC e o dispositivo remoto.

O firmware do Hub [34]

O *firmware* do *Hub* deste projecto corresponde a uma solução de *firmware* criada pela Cypress para configurar o segundo PSoC® como *Hub* genérico que comunica com qualquer dos Nodos presentes no kit CY3271, e que passa essa informação através da I2C para o PSoC® mestre e deste para o PC, onde o software SCD fará a monitorização e armazenamento em tempo real dos dados.

Este Firmware está incluído no ficheiro “RF_HUB.hex”, disponível com o kit CY3271. A descrição pormenorizada ao nível da arquitectura do *firmware* do *Hub* está descrita no guia do *PSoC FirstTouch Starter Kit* CY3271. [34]

Devido à especificidade do kit para trabalhar com o software SCD visando demonstrar as tecnologias *wireless* do PSoC®, a solução de *firmware* do *Hub* inclui

várias variáveis destinadas a ser utilizadas pelo SCD, assim como várias rotinas de pedidos de dados por parte desse software. Este factor, aliado à falta de informação sobre o software SCD e sobre o próprio *Hub* e à impossibilidade de reprogramar o PSoC® mestre da placa FTFC (principal responsável pela comunicação de dados para o PC) tornam difícil e pouco prática a adaptação e a criação de um *Hub* personalizado para este projecto, com base no kit CY3271. Por isto optou-se por manter no seu todo a solução de demonstração da *Cypress* do *Hub*. Esta solução revelou-se capaz de suportar as velocidades de transferência de dados necessárias à instrumentação a desenvolver, estabelece correctamente o protocolo CyFi™ com o Nodo desenvolvido e permite a correcta monitorização dos dados do sensor de forças no software SCD (boa comunicação *Hub* – PC). A solução do *Hub* genérico da *Cypress* cumpriu assim todos os objectivos traçados para o *Hub* do projecto, mostrando ser uma solução possível para esta instrumentação. A imagem do *Hub* utilizado no projecto constitui a figura 21.



Figura 21 – Foto do Hub do projecto, formado pela placa FTFC do kit CY3271 e por um cabo USB.

O Software Cypress Sense and Control Dashboard (SCD) [34;35]

O SCD permite o armazenamento e a monitorização de sensores ou redes de sensores ligados a dispositivos PSoC®. Este *software* comunica por USB com a placa FTFC do kit CY3271. Ele permite ainda iniciar a ligação do *Hub* ao Nodo no protocolo *Wireless* CyFi™, através de um evento accionado pelo utilizador, e ainda possibilita o controlo das redes *Wireless* CyFi™ disponíveis. Permite também configurar os pacotes de dados recebidos de acordo com a sua dimensão, forma e modo de interpretação

dos dados recebidos, dispondo-os sob a forma de um gráfico no tempo, ou de um mostrador. Adicionalmente o software SCD permite calibrar, agregar dados de vários sensores e configurar alarmes. O ambiente da SCD encontra-se representado na Figura 22, exemplificando uma aquisição de temperaturas.

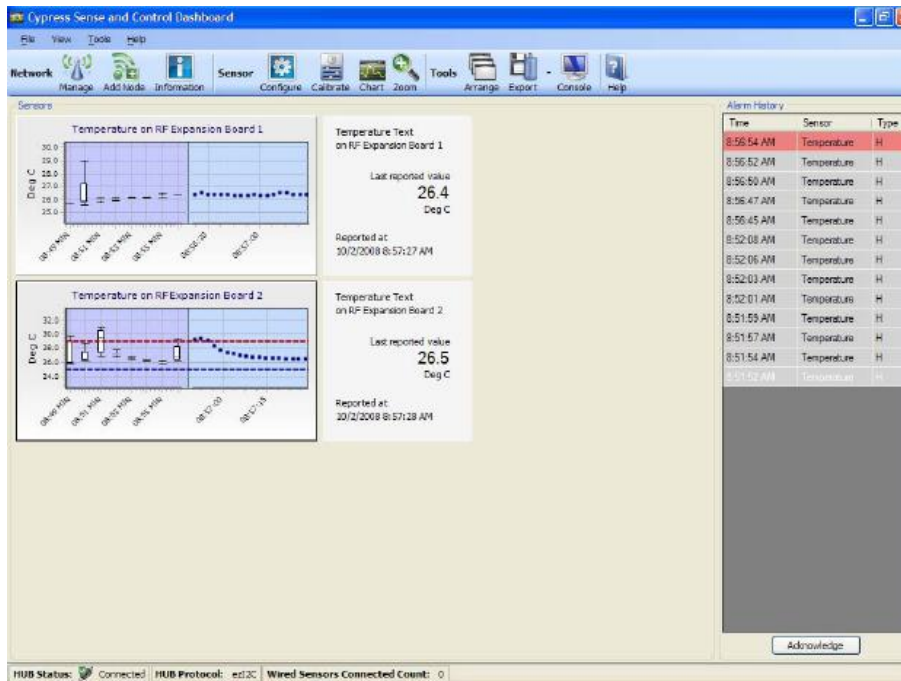


Figura 22 – O ambiente de monitorização da SCD. Imagem retirada de [34].

O Nodo.

O *hardware* do Nodo [34,36]

O *hardware* do Nodo consiste, como já referi em duas placas do kit CY3271, a placa FTRF e a placa AAA, a que se juntam o(s) sensore(s) de força FSS (ver figura 23). Para o desenvolvimento da instrumentação, o sensor de força foi alimentado por uma fonte de tensão externa, uma vez que a placa FTRF não permite alimentar o sensor com a tensão recomendada de 5V (a placa apresenta apenas uma saída 3,3V DC a um máximo de 100 mA no conector de expansão de 16 pinos), tensão com que queremos testar o sensor uma vez que foi com estas condições que realizámos as calibrações anteriores. Também foi cedida às placas e ao sensor de força a mesma terra de referência da fonte de tensão externa, para evitar os erros associados a *ground loops*.

Em resumo, a placa FTRF possui: um PSoC® CY8C27443 – 24PVXI que controla o *transceiver* rádio CYRF7936 (também incluído e com ligação a uma antena também incluída na placa), dois botões, dois LEDs, a alimentação, e o conector de 16 pinos. Esta placa liga à placa FTFC para que este PSoC® possa ser reprogramado. Como também já foi dito no capítulo dedicado ao PSoC®, esta placa apenas possui cinco GPIO livres. Todos os outros GPIO encontram-se ligados aos dispositivos de *hardware* já referidos, ou destinam-se à programação ISSP, ao controlo da alimentação, ou à ligação I2C para comunicar com a placa FTMF, que não é utilizada neste projecto.

O PSoC® CY8C27443 – 24PVXI possui oito blocos digitais dispostos em duas linhas com quatro blocos digitais cada. Também possui doze blocos configuráveis analógicos dispostos em 4 colunas de 3 blocos. Este PSoC® possui doze GPIOs que aceitam sinais analógicos, enquanto todos os 28 GPIOs disponíveis podem aceitar sinais digitais. O CY8C27443 – 24PVXI está contido num invólucro do tipo SSOP. Contém uma memória Flash de 16 Kbytes e 256 bytes de RAM.

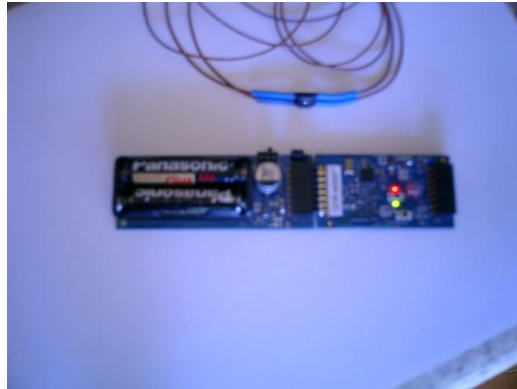


Figura 23 – Foto do Nodo do projecto composto pela placa FTRF, pela placa AAA e pelo sensor de força FSS.

O Sistema a Configurar no PSoC®

O diagrama dos recursos da placa FTRF/PSoC® utilizados para adquirir e processar o sinal proveniente do sensor de força, tal como os recursos utilizados para estabelecer a comunicação de dados encontra-se na figura 24.

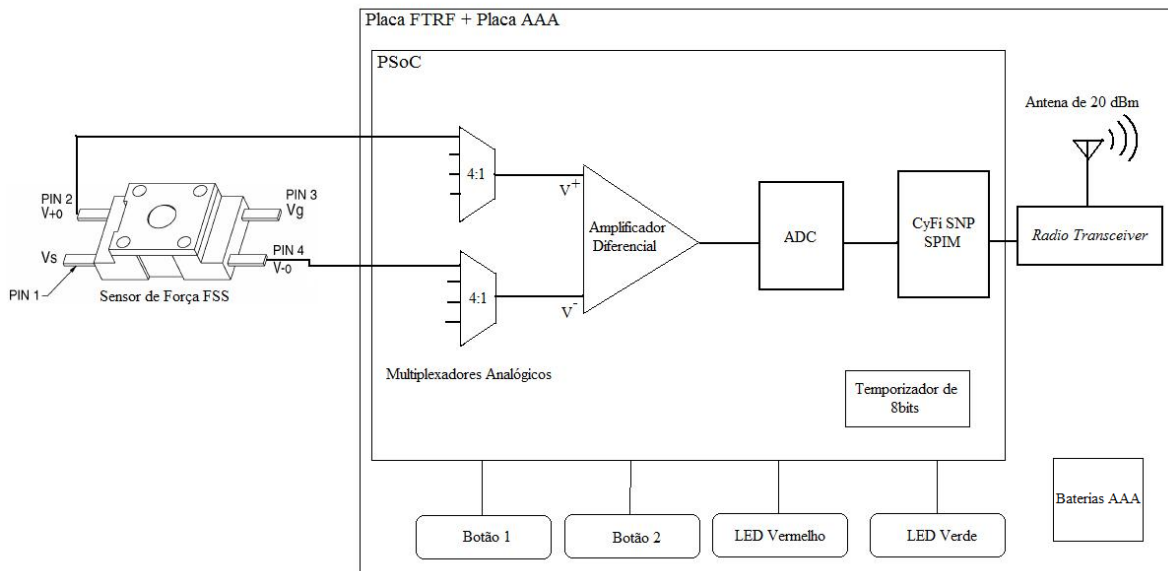


Figura 24 – Diagrama de blocos da arquitectura do Nodo com os principais elementos utilizados para processar o sinal do sensor de força e a sua localização no dispositivo.

Portanto, para ler o sinal diferencial do sensor de força temos de primeiro rotear esse sinal através dos GPIO disponíveis, devidamente configurados como de impedância analógica elevada. Só então os sinais V^+ e V^- provenientes do sensor são encaminhados para um multiplexador com quatro entradas e uma saída. O multiplexador tem como função comutar o sinal vindo dos GPIOs de entrada, permitindo ter em funcionamento mais de um sensor em simultâneo com apenas um ADC e um amplificador diferencial. Se esta comutação entre os sinais vindos dos diferentes sensores for suficientemente rápida, podemos possuir os quatro sensores em funcionamento com este dispositivo sem perda de informação relevante. Após a passagem pelos multiplexadores, o sinal irá para um sistema de blocos analógicos do PSoC[®] programados como amplificador diferencial, para obter a diferença de potencial entre V^+ e V^- . Essa diferença é amplificada neste sistema de modo a utilizar a gama de entrada disponível no ADC. O ADC utilizado é um ADC incremental, que, para ser implementado utiliza dois blocos analógicos e um bloco digital do PSoC[®]. A comunicação do valor de saída do ADC é realizada pelo sistema de comunicação CyFi[™] SNP, que para ser implementado no Nodo requer a programação de um bloco digital (é este bloco que comunica com o CyFi[™] *transceiver* e através desse com a antena). O temporizador utilizado ocupa também um bloco digital e é usado para a calibração do ILO. Fora do PSoC[®] mas necessitando do controlo deste encontram-se os botões e os LEDs, que também serão usados neste projecto.

Criar o Projecto no Nível Chip do PSoC Designer[™] [33,36]

Para acelerar o processo de desenvolvimento de soluções, a Cypress disponibiliza na ferramenta do PSoC Designer[™] uma biblioteca de funções periféricas, pré-testadas e pré-construídas, designadas por *user modules*. Os *user modules* tornam a selecção e implementação dos sistemas analógicos e digitais bastante mais simples, pois enquanto o utilizador selecciona e posiciona o sistema desejado (UART, ADC, DAC...) através do IDE, está a seleccionar um *user module* que estabelecerá automaticamente os registos básicos necessários à implementação dessa função, mobilizando os blocos analógicos e digitais necessários e fornecendo uma lista de parâmetros que o utilizador pode particularizar de modo a ir de encontro aos

objectivos da sua instrumentação. Adicionalmente, cada *user module* providencia ao utilizador funções de alto nível para controlar e responder a eventos no *hardware*, designadas por API (*Application Programming Interface*). As API são utilizadas na construção do *firmware* e para cada *user module* contêm funções como ligar/desligar, o nível de alimentação do *user module*, e outras mais específicas.

Após seleccionar e posicionar os *user modules* na matriz de blocos analógicos e digitais disponíveis no PSoC Designer™, o utilizador deve rotear os sinais entre os blocos utilizados e entre os GPIOs e os blocos, de acordo com o processamento de sinal a realizar. Também nesta altura devem-se seleccionar e ligar correctamente as fontes de relógio aos dispositivos que elas vão temporizar. Os recursos globais também devem ser definidos no nível de chip do projecto, assim como a personalização dos nomes dos GPIOs (*pinout*) e dos blocos. Quando o *hardware* estiver totalmente seleccionado, configurado e no posicionamento final, o utilizador deve executar o gerador de aplicação (*Generate Application*), passo que irá permitir ao PSoC Designer™ gerar o código fonte que configurará o dispositivo para o sistema especificado pelo utilizador. Isto ao mesmo tempo que gera também as API relacionadas com os *user modules* seleccionados, permitindo ao utilizador passar para a fase seguinte de desenvolvimento onde irá criar o *firmware*.

O passo seguinte para criar um projecto no nível chip do PSoC Designer™ consiste em criar o *main* e as suas subrotinas que definem o funcionamento do PSoC® e dos seus blocos. Este passo pode ser desenvolvido em linguagem Assembly ou C. No desenvolvimento desta instrumentação optou-se pela utilização da linguagem de programação C. O compilador de C incluído no PSoC Designer™ é o compilador ImageCraft [38]. É nestas rotinas do *firmware* que está todo o código necessário para estabelecer a solução que pretendemos. O PSoC Designer™ permite-nos aceder aos ficheiros que contêm o código fonte, aos *headers* e às APIs criadas, de modo a que possamos aproveitar e incluir o código já criado na nossa solução.

Por último, após o utilizador ter desenvolvido todo o *firmware* e após ter corrigido todos os erros (quer de sintaxe, quer na configuração dos recursos do PSoC®), o utilizador corre o gerador de projecto e o PSoC Designer™ constrói o ficheiro

hexadecimal que irá programar o dispositivo PSoC® alvo, neste caso o CY8C27443 – 24PVXI da placa FTRF.

A Selecção e Configuração dos *User Modules*

Para implementar o sistema de processamento de sinal da Figura 1, necessitamos de seleccionar dois multiplexadores, um amplificador diferencial, um ADC, o protocolo de comunicação CyFi™ SNP, e um temporizador. De seguida vamos analisar estas escolhas e a sua configuração no universo de possibilidades disponíveis no interior do PSoC® CY8C27443.

O Multiplexador 4 para 1

Existem no PSoC® dois tipos de multiplexadores analógicos. O primeiro tipo tem 4 entradas para 1 saída, o segundo tipo tem 8 entradas para 1 saída. A implementação dos multiplexadores de qualquer um dos tipos não utiliza nenhum bloco dos recursos do PSoC®. É apenas utilizada a memória Flash associada à implementação das API para executar a multiplexagem nos multiplexadores já existentes no PSoC®. Estes dois tipos de multiplexadores têm propriedades em tudo semelhantes, como a elevada impedância de entrada, ou a capacidade dos sinais de entrada poderem ser *rail-to-rail*. Diferem apenas no número de entradas que varia de 4 para 8. Como para implementar a instrumentação desejada apenas necessitamos de multiplexar os sinais de 4 sensores por mão, é suficiente optar pelo multiplexador 4:1. Assim, para ler o sinal diferencial necessitamos de dois multiplexadores 4:1, um para as quatro entradas V^+ , outro para as quatro entradas V^- . Este multiplexador 4:1 liga directamente ao PMux dos blocos analógicos CT. [38]

As propriedades dos dois multiplexadores 4:1 utilizados neste projecto encontram-se definidas nas tabelas da figura 25.

Os pinos deste PSoC® ligados aos multiplexadores (neste projecto) estão configurados como entrada analógica com alta impedância (*high Z analog drive mode*).

Properties - AMUX4_1		Properties - AMUX4_2	
Name	AMUX4_1	Name	AMUX4_2
User Module	AMUX4	User Module	AMUX4
Version	1.4	Version	1.4
Analog Column Mux	AlnMux_0	Analog Column Mux	AlnMux_1

Figura 25 – As propriedades do multiplexador 4:1 na instrumentação final. Imagem retirada do PSoC Designer™.

O Amplificador Diferencial

A tarefa de obter a diferença de potencial entre dois sinais é tradicionalmente levada a cabo por um amplificador de diferencial. De acordo com a nota de aplicação AN2367 [39], para se obter um amplificador diferencial no PSoC®, é preferível construir uma solução com dois PGAs e um ADC com entradas diferenciais, do que optar pelos amplificadores de instrumentação disponíveis como *user modules* no PSoC® associados a um ADC comum. Isto porque esta última configuração apenas pode trabalhar com frequências de relógio de 1 MHz, enquanto que a solução com dois PGAs permite velocidades oito vezes superiores e utiliza menos recursos do PSoC® (não utiliza blocos SC). Ainda de acordo com a nota de aplicação AN2367, vamos implementar o amplificador diferencial com a topologia descrita na Figura 26, seguido de um ADC com entradas diferenciais.

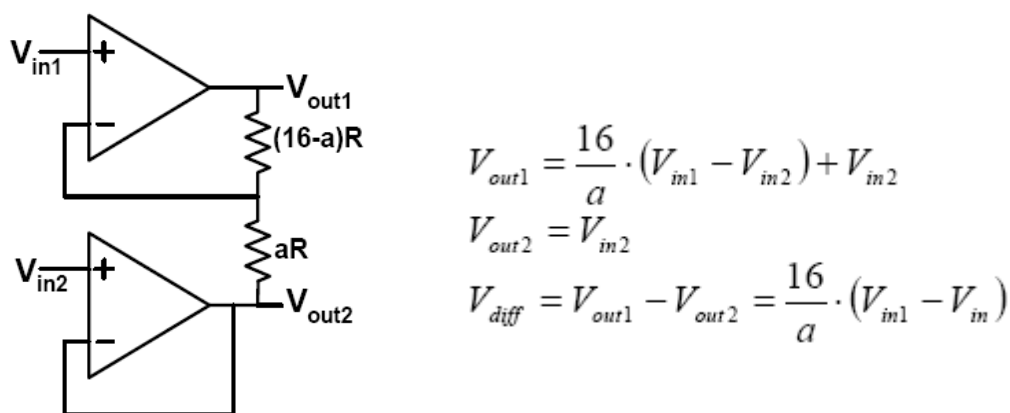


Figura 26 – Topologia do amplificador diferencial implementado na instrumentação final e as equações que descrevem o seu funcionamento. Retirado da AN2367. [39]

Uma vez que as referências analógicas seleccionadas para este projecto foram $V_{dd}/2 \pm V_{dd}/2$ (AGND \pm VRef), o ADC tem uma gama de entrada de $-V_{dd}/2$ até $V_{dd}/2$. Logo, para uma alimentação do PSoC® a 3,3V o ADC terá uma gama de entrada de -1,65V até 1,65V. Esta gama de entrada poderia ser alterada através do VRef, colocando a tensão de referência desejada no pino 2[6] (porta 2, pino 6), mas tal não é possível na placa FTRF, uma vez que esta porta foi utilizada para outro propósito e não se encontra disponível. Assim opta-se pela tensão de referência $V_{dd}/2 \pm V_{dd}/2$, uma vez que é a que permite maior gama de entrada do ADC e portanto uma maior amplificação do sinal diferencial. Como o sinal proveniente do sensor de força é positivo e varia de 0 a 180 mV (0 a 1500g), utilizamos apenas a meia escala positiva do ADC para a conversão. Como estamos apenas interessados em forças até aos 10N (1019,72 g), a resposta típica máxima dada pelo sensor é de 122,37 mV (quando alimentado a 5V). Este dado permite que o sinal proveniente do sensor seja amplificado 13,48 vezes. Como os PGAs do PSoC® [40] apenas permitem determinados valores discretos de amplificação (4, 16, 24, 48) optamos pelo valor de amplificação mais próximo de 13,48 que é 16. Esta opção faz com que o valor máximo da saída do sensor antes de o ADC saturar seja de 103,125 mV, ou de 859,375 g (utilizando o valor de sensibilidade típica do sensor FSS 0,12 mV/g a 5V). Passando o sistema a registar forças até aos 8,428 N em vez dos 10 N que consta nos objectivos. Apesar de tudo esta perda na gama de entrada é aceitável e não afecta de modo significativo os objectivos do projecto, permitindo obter maior resolução no ADC (menor número de grama por bit do ADC).

Assim o posicionamento (na estrutura do PSoC®) dos blocos responsáveis pela implementação do amplificador diferencial descrito e as suas propriedades encontram-se na Figura 27.

Properties - PGA_1		Properties - PGA_2	
Name	PGA_1	Name	PGA_2
User Module	PGA	User Module	PGA
Version	3.2	Version	3.2
Gain	16.00	Gain	1.000
Input	AnalogColumn_InputMUX_0	Input	AnalogColumn_InputSelect_1
Reference	ACB01	Reference	VSS
AnalogBus	Disable	AnalogBus	Disable

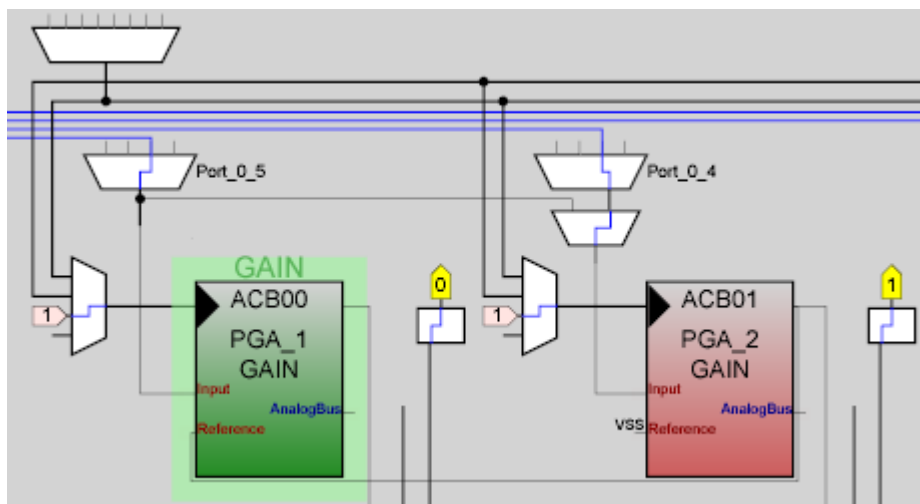


Figura 27 – Propriedades dos dois PGAs (em cima), o seu posicionamento e ligações (em baixo) na instrumentação final. Retirado do PSoC Designer™.

O ADC

O PSoC® permite a implementação de vários tipos de conversor analógico/digital:

ADC SAR – Registo de Aproximações Sucessivas. Este tipo de ADC utiliza 100% da CPU durante a conversão pelo que é desaconselhado quando se pretende utilizar a CPU em simultâneo com a conversão.

ADC Incremental. Este ADC utiliza um integrador, um comparador e um par de referências de tensão. Com base nesta topologia existem vários outros tipos de ADC especificados no PSoC, como ADCINCVR, DualADC, TriADC, ADCINCVR, entre outros. A diferença entre eles assenta no número de bits, na taxa de amostragem, ou em expansões com inclusão de mais blocos analógicos e digitais no mesmo ADC por motivos de maior precisão, menores tempos de conversão, ou simplesmente por serem aplicações que ocupam menor espaço na memória e utilizam menos CPU.

ADC Delta Sigma. Este ADC utiliza uma topologia semelhante ao incremental, mas difere no processamento dado à saída do comparador. Em vez desse resultado ser integrado num contador, o que sucede no ADC incremental, neste, a saída do comparador é processada no decimador.

Com base na nota de aplicação AN2239 da Cypress [41], o conversor incremental permite uma resolução alargada (de 6 a 14 bits), uma taxa de amostragem suficiente, possui entradas diferenciais, utiliza pouco a CPU libertando-a para as tarefas de comunicação, está adaptado à multiplexação (no caso do Delta Sigma as duas primeiras respostas após multiplexação não possuem significado), possui baixos erros de não linearidade diferencial e integral, e apresenta baixo ruído de quantização. Portanto o ADC incremental é o ADC escolhido para este projecto.

Neste projecto, o ADC implementado no PSoC® é o ADCINC com um modulador de 2ª ordem. Este ADC utiliza 2 blocos analógicos SC, um bloco digital básico e o decimador. O esquemático deste ADC encontra-se na Figura 28. A construção deste ADC necessita que dois blocos SC sejam configurados como um integrador. De acordo com a polaridade à saída do integrador, o sinal de referência é adicionado ou subtraído à entrada do integrador, fazendo com que sucessivamente a saída do integrador se aproxime do valor de AGND. O ADCINC necessita de 2^n ciclos de integração para obter um resultado com n bits de resolução. Para o funcionamento deste ADC é necessário um modulador PWM, para contar o número de ciclos de integração, e um decimador configurado no modo incremental para acumular o número de ciclos em que a saída do comparador é positiva. [42]

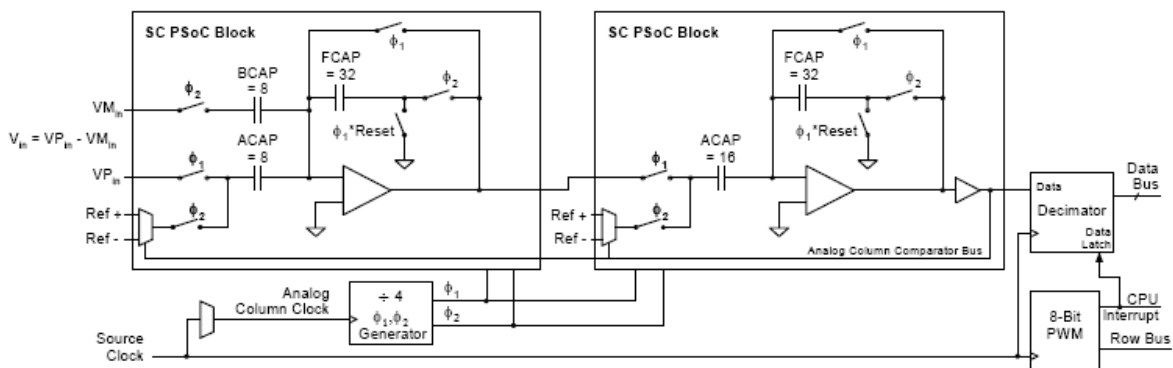


Figura 28 – Esquemático do ADCINC com um modulador de 2ª ordem. Retirada da Data Sheet [42].

As entradas diferenciais do ADC devem ser ligadas aos PGAs configurados anteriormente como amplificador diferencial, sendo que a entrada positiva liga ao PGA_1 e a entrada negativa ao PGA_2. As propriedades do ADCINC especificadas para o projecto encontram-se na Figura 29.

Properties - ADCINC_1	
Name	ADCINC_1
User Module	ADCINC
Version	1.1
DataFormat	Unsigned
Resolution	8 Bit
Data Clock	VC1
PosInput	ACB00
NegInput	ACB01
NegInputGain	1.000
ClockPhase	Normal
PulseWidth	1
PWM Output	None

Figura 29 – Propriedades do ADCINC na instrumentação final. Retirado do PSoC Designer™.

O CyFi™ SPN SPIM

O módulo utilizado para implementar o CyFi™ SNP num Node necessita de um bloco digital básico, 7416 bytes de memória Flash, 87 bytes de RAM e 5 GPIOs (2[0], 2[3], 2[5], 2[6] e 2[7]) devidamente configurados para permitir as comunicações entre o PSoC® e o CyFi™ rádio *Transceiver*. Este módulo requer que a CPU do PSoC® funcione a uma frequência superior a 3 MHz e que um bloco da memória Flash seja marcado como não protegido a fim de se guardar no Node parâmetros de rede, como a

identificação dada ao dispositivo. A descrição do protocolo com os diferentes modos de funcionamento, assim como a descrição da estrutura dos diferentes pacotes de dados e das API encontram-se na folha de especificações [43]. As propriedades definidas para este bloco encontram-se na Figura 30.

Properties - CYFISNP_1	
Name	CYFISNP_1
User Module	CYFISNP
Version	1.0
Node Power Supply Type	Low impedance battery
EEPROM Block	255
External PA	Yes
MUX IRQ on MOSI	Disabled
Clock	VC3
MISO	Row_0_Input_0
MOSI	Row_0_Output_1
SClk	Row_0_Output_3
IRQ_Port	Port_2
IRQ_Pin	Port_2_6
nSS_Port	Port_2
nSS_Pin	Port_2_7
First Channel	10 (2.412 GHz)
Last Channel	58 (2.460 GHz)

Figura 30 – Parâmetros do CyFi™ SNP na instrumentação final. Retirado do PSoC Designer™.

O temporizador de 8 bits

O módulo para implementar um temporizador (*timer*) de 8 bits utiliza um bloco digital básico e 70 bytes de Flash. É utilizado no projecto para a calibração do ILO e, desse modo, calibrar a fonte de relógio que regula a comunicação CyFi™ SNP (o ILO sem este processo de calibração possui uma precisão que pode variar de -50% a +100% no pior caso [45]). Informações sobre o funcionamento deste módulo, assim como as API e os registos associados encontram-se na data sheet [44]. O método aplicado para calibrar e compensar o ILO foi baseado na nota de aplicação NA14278 da Cypress [45], e no exemplo de projecto *RF Expansion Card Ultra Low Power Temperature Sensor* disponível com o kit CY3271. Este método de calibração do ILO consiste em contar o número de impulsos gerados pelo *sleep timer*, durante um período de tempo definido por um relógio que tenha como base o IMO. Este método permite comparar o número de impulsos que o sleep timer gerou durante esse período, com o valor de impulsos

que deveria ter gerado, procedendo-se de seguida de modo a corrigir essa variação. As propriedades definidas no temporizador seleccionado para este projecto estão apresentadas na Figura 31.

Properties - Timer8_1	
Name	Timer8_1
User Module	Timer8
Version	2.6
Clock	CPU_32_KHz
Capture	Low
TerminalCountOut	None
CompareOut	None
Period	255
CompareValue	0
CompareType	Less Than Or Equal
Interrupt Type	Terminal Count
ClockSync	Sync to SysClk
TC_PulseWidth	Full Clock
InvertCapture	Normal

Figura 31 – As propriedades do temporizador na instrumentação final. Retirado do PSoC Designer™.

Os recursos globais do PSoC®

As especificações dos parâmetros dos recursos globais do PSoC® para este projecto encontram-se na Figura 32. A descrição de cada parâmetro está presente na referência [33].

Global Resources - forcefingertips	
CPU_Clock	12_MHz (SysClk/2)
32K_Select	Internal
PLL_Mode	Disable
Sleep_Timer	64_Hz
VC1= SysClk/N	6
VC2= VC1/N	1
VC3 Source	SysClk/1
VC3 Divider	6
SysClk Source	Internal 24_MHz
SysClk*2 Disable	No
Analog Power	SC On/Ref High
Ref Mux	(Vdd/2)+/-(Vdd/2)
AGndBypass	Disable
Op-Amp Bias	Low
A_Buff_Power	Low
SwitchModePump	OFF
Trip Voltage [LVD (SM)	3.02V (3.09V)
LVDThrottleBack	Disable
Supply Voltage	3.3V
Watchdog Enable	Disable

Figura 32 – Propriedades dos recursos globais do PSoC® na instrumentação final. Retirada do PSoC Designer™.

No final da configuração de todos os parâmetros e do posicionamento dos blocos e das ligações estar devidamente estabelecido, o projecto ao nível do *hardware* encontra-se completo. Resta gerar a aplicação final em *firmware* que irá definir o funcionamento quer do *hardware* seleccionado, quer das tarefas a executar com os dados na solução final. O aspecto do ambiente de desenvolvimento do PSoC Designer™ no final deste processo de configuração do *hardware* (módulos) do PSoC® está apresentado na Figura 33.

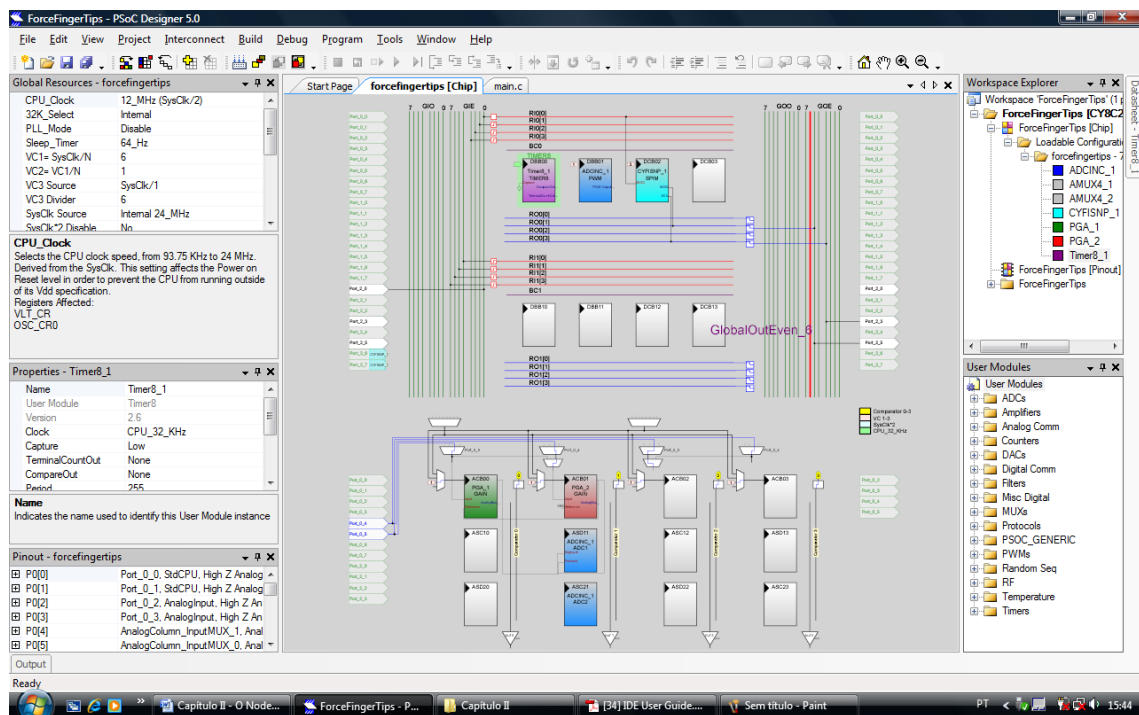


Figura 33 – Aspecto do PSoC Designer™ após a configuração dos módulos para este projecto.

O *firmware* do Nodo

O *firmware* do Nodo foi criado utilizando o compilador de linguagem C ImageCraft [37] presente como ferramenta dentro do PSoC Designer™. O *firmware* do Nodo do projecto foi desenvolvido em linguagem de programação C, e após ser criado e se ter verificado e corrigido os erros de programação (a ferramenta de *debugger* também está presente no PSoC Designer™) foi transformado num ficheiro hexadecimal com o nome do projecto. Esse ficheiro é que irá, como já referi anteriormente, programar o PSoC® da placa FTRF.

Para a construção do *firmware* foram utilizadas as APIs que constam das *data sheets* dos módulos utilizados. O diagrama de blocos apresentado (ver figura 34) tem a finalidade de ajudar a perceber as funções e a estrutura do *firmware* criadas para o Nodo do projecto. A totalidade do código desenvolvido encontra-se disponível no Anexo I.

Para o desenvolvimento do Nodo foram criadas 5 funções no *firmware* deste projecto: duas destinadas a verificar se utilizador premia os botões SW1 e SW2 (Switch 1 e 2), designadas respectivamente de *Check_bind_SW1* e *Check_bind_SW2*, uma destinada a calibrar o ILO com o IMO para permitir maior precisão no ILO e na comunicação *wireless* CyFi™ (Calibrate_ILO), uma para processar os dados recebidos pelo Nodo (função *Check_Data_Rx*), e, finalmente, uma função para obter o valor de força dos diferentes sensores e carregar essa informação no pacote de dados a enviar (função *GetForceReport*).

A função *delay* importada no início do código foi retirada do *site* da Cypress onde se encontra disponível para *download*. Esta função é utilizada para produzir um atraso nas funções *Check_bind_SW1* e *Check_bind_SW2* permitindo verificar se o botão foi realmente premido.

Gerar e programar o Nodo

No final do desenvolvimento do *firmware* do Nodo, gerou-se a aplicação no PSoC Designer™, criando-se o ficheiro hexadecimal *ForceFingerTips.hex*, com o qual se programou o Node através do software PSoC Programmer™ 3.05 e da placa FTPC.

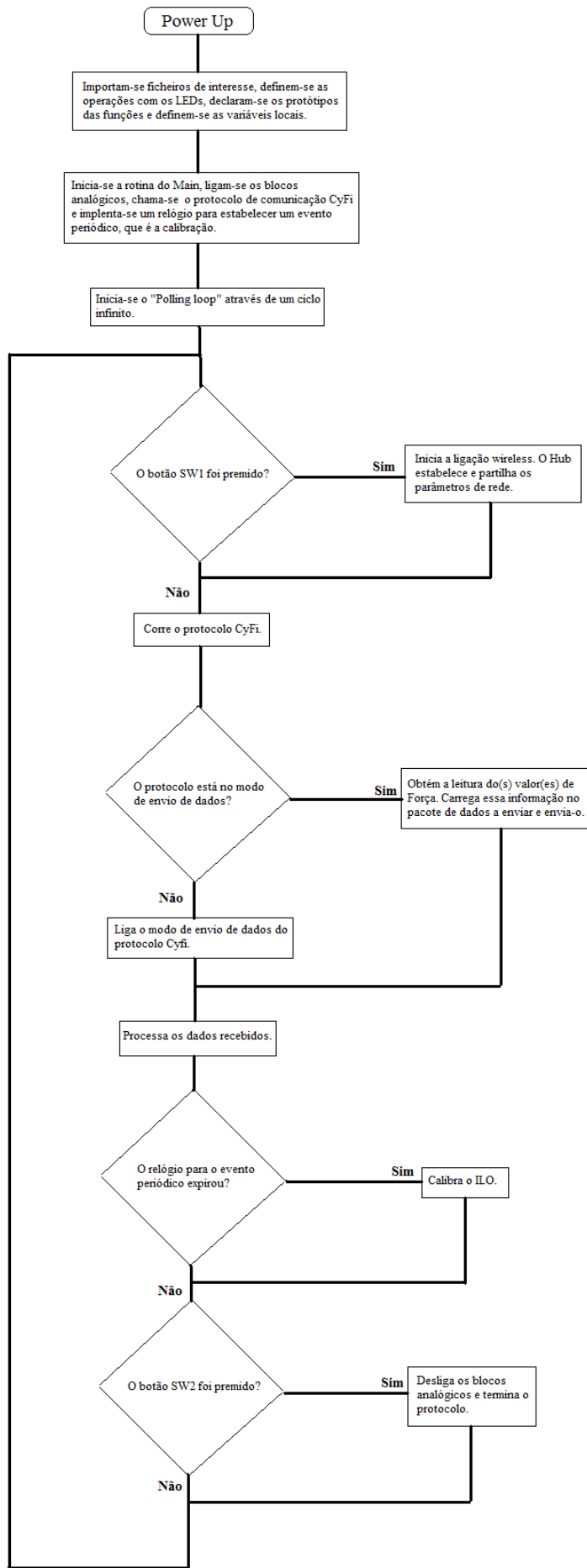


Figura 34 – A estrutura do *firmware*.

Desenvolvimento do Software

A SCD

Iniciar a Ligação Hub - Nodo [34]

A SCD, como já havia referido, permite monitorizar e armazenar os dados enviados pelo(s) Nodo(s) ao *Hub*. É também através deste programa que é possível criar redes e controlar o *Hub* do kit CY3271. Para iniciar a monitorização e aquisição dos dados de força provenientes do sensor necessitamos de ligar o *Hub* ao PC e iniciar a aplicação *Sense and Control Dashboard* (SCD). Após ligar a alimentação do(s) Nodo(s) adicionamo-lo(s) à rede através da SCD. (figura 35). Após o Hub e o Nodo terem partilhado os parâmetros de rede e da comunicação ser estabelecida, é apresentada uma mensagem de sucesso. É posteriormente pedido para configurar o tipo de informação que o Nodo envia.

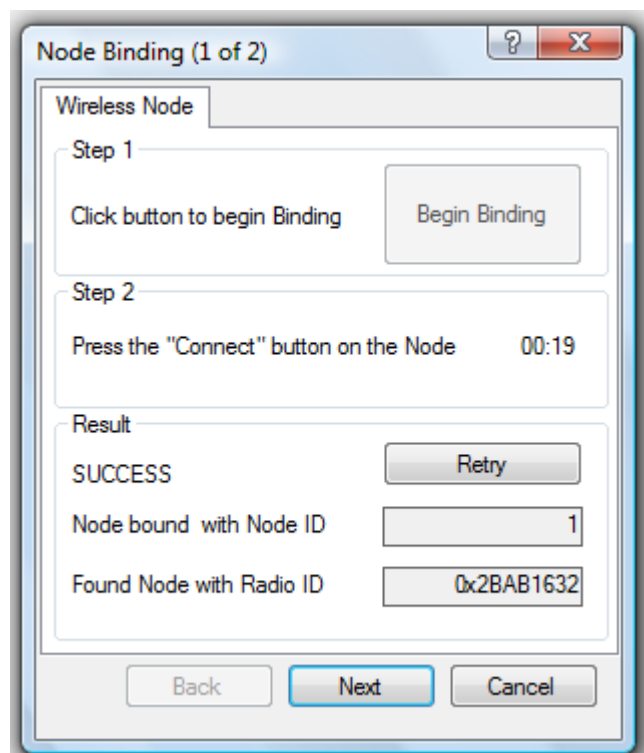


Figura 35 – Janela para a ligação no Nodo na rede.

Retirado da SCD.

Configurar os Dados do Nodo

Os pacotes de dados provenientes do Nodo têm de ser correctamente interpretados pela SCD a fim de serem monitorizados e amostrados. A janela a exemplificar os parâmetros seleccionados para cada sensor do(s) Nodo(s) deste projecto encontra-se na Figura 36.

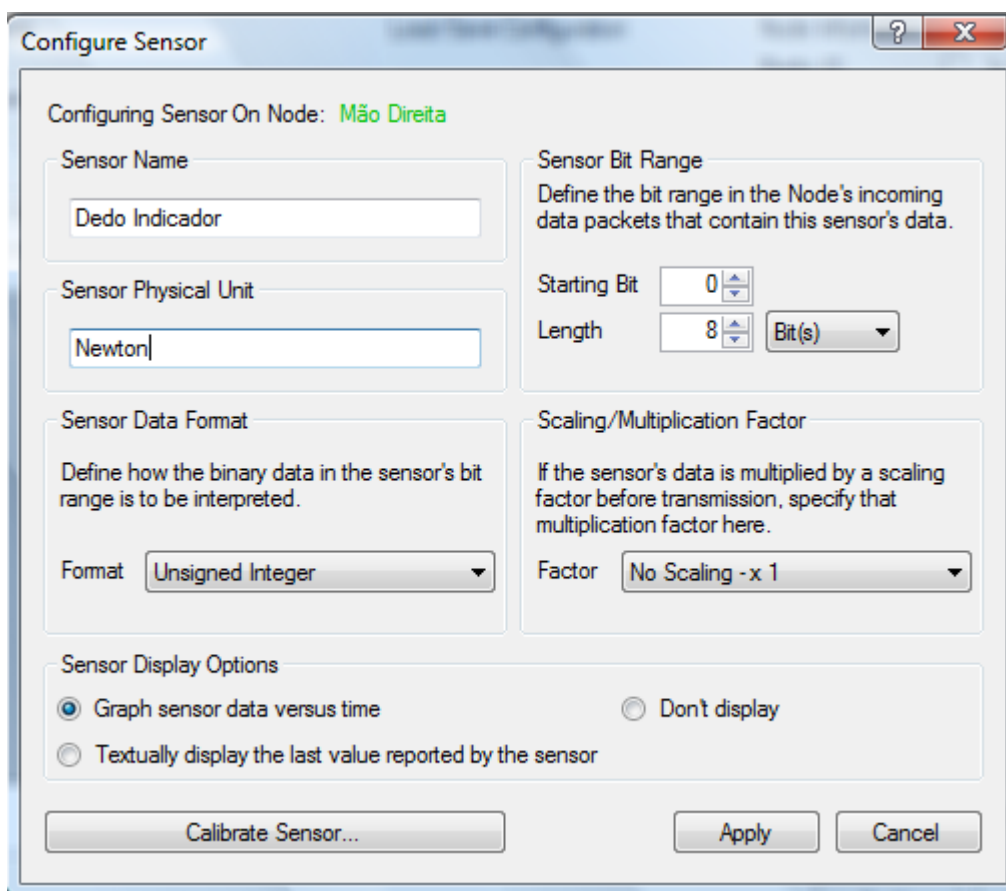


Figura 36 – Janela para configurar os Dados provenientes de cada sensor do Nodo. Retirado da SCD.

Monitorizar o valor de Força na SCD

Os pacotes de dados provenientes dos Nodos contêm o valor da saída do ADC para cada sensor ligado. A SCD permite, através da calibração de cada sensor, transformar o valor de saída do ADC no valor de interesse que pretendemos visualizar. Os parâmetros para esta curva de calibração foram obtidos através do cálculo de um

parâmetro que permite transformar a saída do ADC em Newton. Para tal foram utilizados o valor da sensibilidade do sensor de força FSS (0,0795 mV/g quando alimentado a 3,3V – o motivo para alimentação do sensor ser de 3,3V, e não os tradicionais 5V é discutido em pormenor no capítulo 4), o valor da resolução do ADC (12,89 mV/bit), o valor típico de *Offset* (o *Offset* é usualmente 130 ± 1 bit), a amplificação (16 vezes), e a taxa de conversão de grama em Newton que é 0,00980665 N/g. A fórmula para o cálculo do valor calibrado encontra-se em baixo.

$$\text{Valor calibrado} = \text{Conversor}(N/\text{bit}) \times \text{Valor do ADC} - \text{Offset}$$

$$\text{Conversor}(N/\text{bit}) = \frac{\text{Resolução do ADC}(mV/\text{bit}) \times \text{Taxa de conversão}(N/g)}{\text{Amplificação} \times \text{sensibilidade do FSS}(mV/g)}$$

$$\text{Valor calibrado} = 0,09934(N/\text{bit}) \times \text{Valor do ADC} - \text{Offset}$$

A Calibração dos sensores de força na SCD foi realizada (ver Figura 37) assumindo o *Offset* como 130, e que 1 bit na saída do ADC representa 0,09934 Newton.

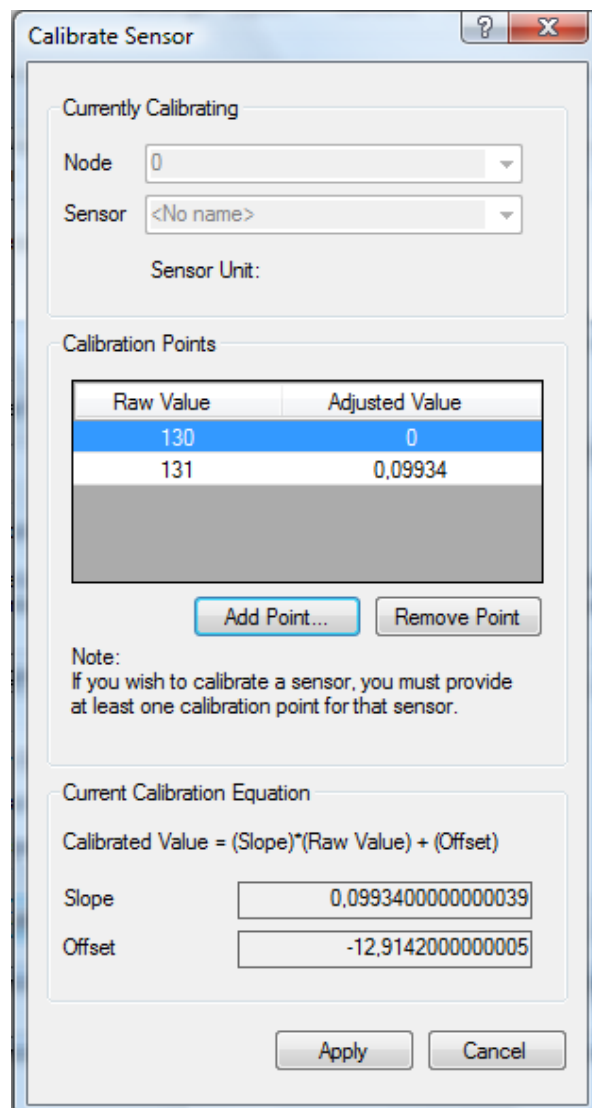


Figura 37 – Janela para definir a calibração do sensor na SCD. Retirado da SCD.

Para este projecto foi criado um *template* que configura automaticamente a SCD para receber valores de dois sensores de força por Nodo. Precisamente o número máximo de sensores de força que são permitidos para cada placa FTRF. Basta indicar o ficheiro *template* correcto imediatamente após estabelecer a ligação com o Nodo. De outro modo, para adicionar manualmente mais do que um sensor é necessário indicar na janela de configuração do sensor, o bit no pacote de dados onde começa o Byte respeitante a esse novo sensor (o bit de começo para o segundo sensor neste projecto é o oitavo bit). Para ligar mais do que um Nodo ao *Hub* é necessário voltar a executar todo este processo outra vez para o novo Nodo.

Exportar os dados obtidos pela SCD

A SCD inclui uma ferramenta que permite exportar os dados adquiridos. Basta, para tal, seleccionar o sensor desejado e executar a opção *export*. A SCD exporta a aquisição respeitante a esse sensor até ao instante presente, num ficheiro de formato *.CSV (Ficheiro de Valores Separados por Vírgulas). Para analisar e processar os dados exportados pela SCD foi desenvolvida neste projecto uma aplicação gráfica em Matlab.

A Aplicação de Análise e Processamento de Dados

Descrição da Aplicação Gráfica [47]

A aplicação gráfica foi criada com o propósito de analisar e processar os dados provenientes da SCD correspondentes aos valores de força registados pelos sensores FSS. Esta aplicação, designada “ForceGlv” foi desenvolvida em MATLAB® R2008b. Trata-se de uma aplicação que adquire oito sinais de força em formato *.CSV e processa-os de modo a estabelecer relações e assegurar o tratamento estatístico destes oito sinais (embora cada placa FTRF só suporte dois sensores FSS, ou seja 2 sinais de força, a aplicação está preparada para receber e processar a totalidade dos sinais de força previstos nos objectivos).

A aplicação permite ao utilizador carregar os dados de força de acordo com o dedo da mão que os originou. Isto permite separar a origem dos dados e realizar mais tarde estudos tendo como base apenas um dedo específico de uma mão específica, ou estudar apenas todos os dedos de uma mão. Após a selecção da origem dos dados pelo utilizador, a aplicação executa uma função destinada a abrir e transformar os dados do ficheiro *.CSV num vector de dados capaz de ser manipulado pelo MATLAB® (função *opendata*). Os dados que constam no vector criado à saída da função *opendata*, são os valores da saída do ADC multiplicados pela taxa de conversão 0,09934 N/bit (obtida para o sensor FSS alimentado a 3,3V). Após os dados estarem carregados no MATLAB®, a interface apresenta um sinal de “confere” junto do nome da origem dos dados carregados (ver Figura 38).

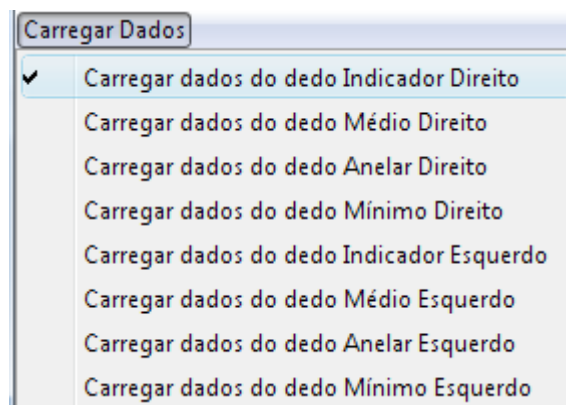


Figura 38 – Janela para escolha da origem dos dados a carregar no MATLAB®. Encontram-se carregados os dados do dedo indicador direito.

Após carregar todos os dados que se pretendem analisar, o utilizador deve seleccionar o objecto de estudo, sob o qual vão incidir a análise e o tratamento posterior. Esta selecção é realizada numa janela *pop-up* (ver Figura 39). Para cada caso seleccionado aplicam-se as funções *Avaliar* e *removezeros*. No final da selecção do objecto de estudo, a função *Avaliar* verifica se os dados foram efectivamente carregados, e no caso de não terem sido carregados apresenta uma mensagem de erro. Caso os dados tenham sido correctamente carregados a função *removezeros* constrói um vector com os dados de força sem os tempos mortos (tempos onde não foi registada qualquer força) e sem o *offset* (o valor do *offset* é calculado pela média das primeiras 60 amostras adquiridas, no início da utilização da instrumentação a parte sensível dos sensores não deve tocar em nada durante esses dois segundos iniciais).

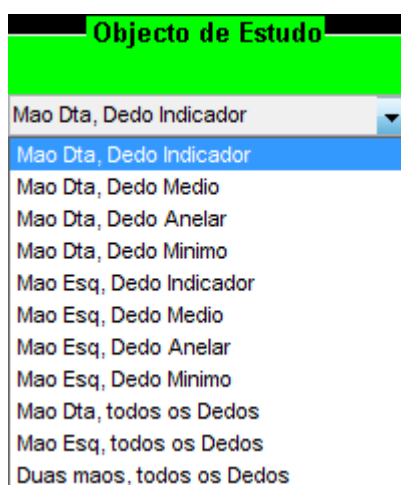


Figura 39 – Janela *pop-up* para selecção do objecto de estudo.

Ao seleccionar o objecto de estudo, é apresentado ao utilizador, na janela “Sinal Inicial Seleccionado”, o gráfico da variação da força no tempo, sem qualquer tipo de tratamento do sinal de força, incluindo os tempos mortos e o *offset*.

As colunas do painel de estatística permitem obter uma análise estatística do conjunto de dados seleccionados. A selecção dos dados é realizada duplamente: uma parte diz respeito ao objecto de estudo, como já vimos anteriormente, e a outra parte diz respeito às opções disponíveis para tratar os dados do objecto de estudo seleccionado (remover tempo morto, remover *offset*, etc.), opções de selecção que

constam no painel “Tratamento”. É sobre este segundo conjunto de dados escolhidos que incide a análise estatística. Foi criada uma função designada por *estatistica*, que utiliza as ferramentas estatísticas do MATLAB® e aplica-as ao sinal seleccionado. O tratamento estatístico disponível consiste em medidas de tendência central (média, mediana, moda, etc.), medidas de dispersão (MAD, desvio padrão, variância, etc.) e medidas de forma da distribuição (percentis).

Os últimos dois painéis da aplicação dizem respeito ao tratamento dos dados e à sua visualização gráfica. Para tratamento dos dados estão disponíveis as seguintes opções:

- 1) Remover os tempos mortos e o *offset*, operação designada por OT;
- 2) Determinar o valor rms (*roote mean square*) do sinal de força a cada 15 amostras (função *RMS15sp*), sob este sinal já foi realizada também a OT;
- 3) Realizar a OT e obter o valor máximo do sinal de força de 15 em 15 amostras (função *MAX15sp*);
- 4) Apresentar o gráfico de quartis respeitante aos dados seleccionados (mais OT);
- 5) Realizar o histograma dos dados seleccionados (mais OT);
- 6) Apresentar o gráfico de probabilidade acumulada assumindo a distribuição dos dados como sendo normal (mais OT);
- 7) Análise das médias ao longo do tempo onde para tal divide-se o tempo de aquisição em dez parcelas e apresenta-se sob a forma de gráfico a média de cada uma dessas fracções temporais (estudo realizado sobre o sinal inicial, apenas é removido o offset com a função *removeoffset*).

O aspecto completo da *ForceGlv* encontra-se na figura 40. O código para criar esta interface encontra-se nos Anexos II e III.

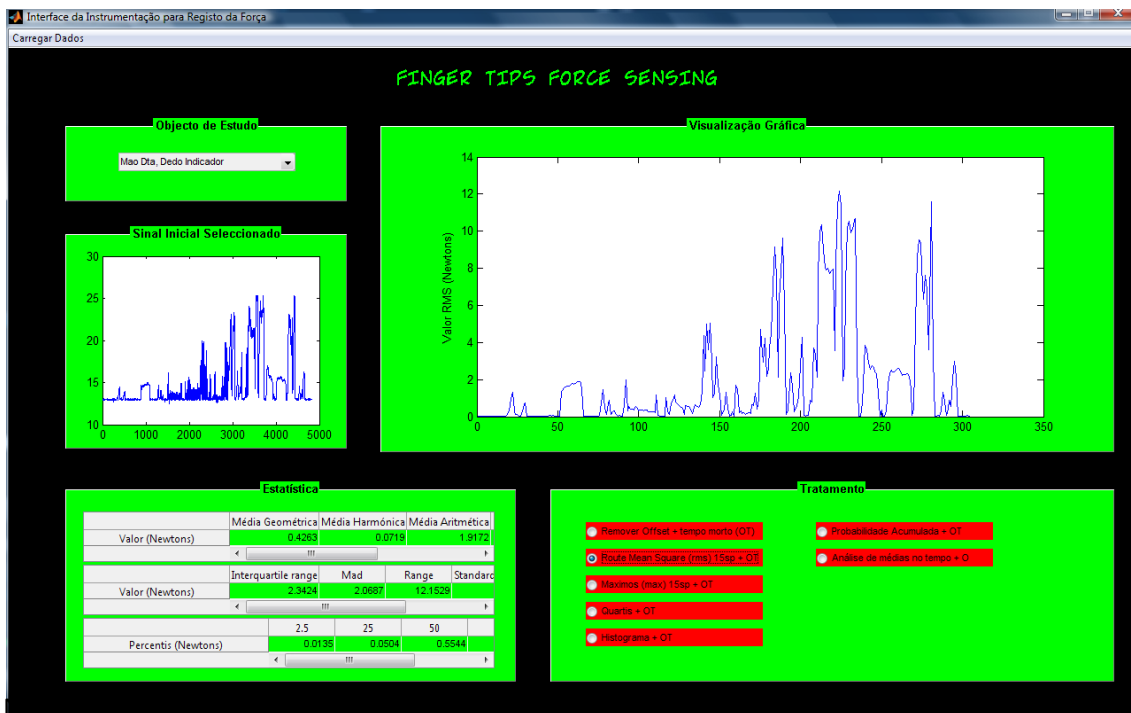


Figura 40 – A Aplicação Gráfica ForceGlv em MATLAB®.

Testes, Discussão de Resultados e Conclusões Finais

Testes Realizados

Por fim foram realizados alguns testes sobre a instrumentação desenvolvida. Para efectuar estes testes foi ligado o sensor de força *FSS* da *Honeywell* à placa *FTRF* (devidamente configurada como Nodo *ForceFingerTips*). Relembro que o sensor *FSS* se encontra alimentado para efeitos de teste por uma fonte de tensão externa que não a placa *FTRF*. O Hub está configurado como *RF_HUB*, uma solução disponível pela *Cypress*, e encontra-se ligado a um PC onde a monitorização e armazenamento dos dados adquiridos é efectuada através do software *SCD* também da *Cypress*. Os dados armazenados no PC são depois exportados para a ferramenta *ForceGlv*, uma aplicação gráfica criada em *MATLAB*® para analisar e processar os dados dos sinais de força adquiridos pela instrumentação.

Os testes realizados sobre o sistema foram:

- 1) Duas calibrações estáticas com o sensor *FSS* ligado à instrumentação acima descrita e alimentado numa calibração a 3,3V e noutra a 5V. A calibração foi realizada segundo o processo de calibração já referido anteriormente no subcapítulo “O Sensor”;
- 2) Foram realizadas aquisições de 50 minutos seguidos, para verificar se o sistema suportava o tempo de aquisição contínuo pedido para este projecto e o tratamento da quantidade de dados resultante. O tempo de operação contínua relembro ser 3x50 minutos de aquisição com intervalos de 10 minutos entre as aquisições, isto para cada sujeito em estudo;

- 3) Para testar o funcionamento da multiplexagem, foi ligado no lugar dum segundo sensor de força na FTRF uma fonte de tensão e monitorizaram-se simultaneamente dois sensores na SCD;
- 4) Foram testados dois Nodos (ambos configurados como *ForceFingerTips*) em funcionamento simultâneo com apenas um *Hub*, para verificar se a instrumentação desenvolvida tem viabilidade para adquirir das duas mãos com a velocidade de 30 amostras por segundo por mão.

Calibração Estática

Na calibração estática com massas de referência foram obtidos dez pontos à temperatura ambiente, numa gama de 0 até 571 gramas. O sensor está alimentado inicialmente a 5V e encontra-se montado na instrumentação já descrita. O valor da resposta do sensor é obtido recorrendo á monitorização com o software SCD. É registada como sendo a resposta do sensor, o valor médio da saída do ADC registado na SCD logo após a aplicação da carga e após a remoção do offset presente.

Para a calibração com o sensor *FSS* a operar a 5V obteve-se os resultados descritos na Tabela 7 e no Gráfico 2. Algumas propriedades da instrumentação que foram utilizadas para o cálculo dos valores presentes na calibração encontram-se na Tabela 6.

Tabela 6 – Propriedades da instrumentação.

Amplificação do input Diferencial	16
Meia escala do ADC (volts)	1,65
Número de bits de ½ escala do ADC	128
Patamar do ADC (volts/bit)	0,01289
Patamar do ADC (mV/bit)	12,89

Tabela 7 - Resultados da calibração estática do *FSS Low Profile Force Sensors*, a operar a 5V e ligado á instrumentação desenvolvida no PSoC®.

Massa Aplicada (gramas)	Valor ADC	Valor ADC - Offset	Valor (mV)	Intervalo de massa (gramas)	Intervalo (mV)	Sensibilidade (mV/g)	
0	130	0	0,00	21,8	38,67	1,77	
21,8	133	3	38,67	91,1	141,80	1,56	
112,9	144	14	180,47	42,3	116,02	2,74	
155,2	153	23	296,48	90,4	167,58	1,85	
245,6	166	36	464,06	54,9	103,13	1,88	
300,5	174	44	567,19	78,4	154,69	1,97	
378,9	186	56	721,88	378,9			
						Sensibilidade Media:	2,02
							0,13

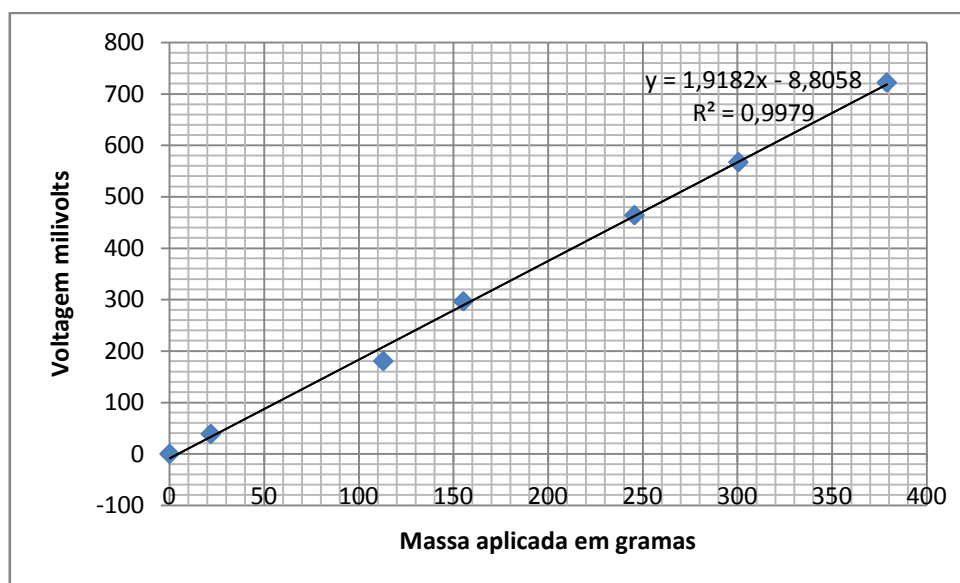


Gráfico 2 - Resultados da calibração estática do *FSS Low Profile Force Sensors*, a operar a 5V e ligado á instrumentação desenvolvida no PSoC®.

A sensibilidade registada nesta calibração foi de 0,126 mV/g o que vai de encontro às especificações do sensor *FSS* (0,12 mV/g típicos, podendo esse valor variar entre 0,1 e 0,14) e também de encontro à calibração apresentada no subcapítulo “O Sensor”, onde o valor calculado para esta propriedade foi de 0,124 mV/g. A variação do sinal de tensão com a carga aplicada também se apresenta com uma linearidade elevada, o que é visível pelo elevado factor de correlação linear 0,9979. A não linearidade calculada é de 0,95% FSS, valor que também está dentro dos valores descritos na *data sheet* do sensor como tendo uma não linearidade de $\pm 1,5\%$ FSS. Prova-se assim que o sistema de aquisição desenvolvido não interfere com relevância

nas principais propriedades do sensor escolhido. O erro associado a cada medição é de 1 LSB ou seja de $\pm 0,0066$ Newtons ($\pm 6,567$ grama).

Esta calibração não utiliza os dez pontos definidos numa gama de 0 a 571 grama, devido a limitações do PSoC[®]. Isto porque, para uma amplificação de 16 vezes do sinal diferencial, é ultrapassada, um pouco antes da aplicação das 378,9g a limitação de tensões internas do dispositivo que é V_{ss} (3,3V neste projecto). Este facto acontece porque a tensão do sinal diferencial amplificada é somada ao valor de tensão comum de entrada no PSoC[®] antes da ligação ao ADC. Isto faz com que aos cerca de 2,6V de tensão comum seja somada uma tensão diferencial igual a 16 vezes $0,12 (mV/g) \times 378,9 g$, ultrapassando o V_{ss} . Logo, o sistema para um sensor *FSS* alimentado a 5V, satura por volta das 365 grama, não permitindo sequer utilizar a totalidade da meia escala disponível do ADC. Para operar a 5V e obter a gama de entrada especificada para este projecto é necessário baixar a amplificação para um valor menor do que 16 vezes, valor mais baixo que, associado à obrigatoriedade de manter as voltagens de referência do ADC, compromete a resolução final da instrumentação. Isto levou-nos a optar por outro caminho para adequar este sensor ao PSoC[®] placa FTRF.

Este outro caminho traduz-se em alimentar do sensor a 3,3V. Deste modo para além de baixarmos a tensão comum, utilizamos a totalidade da meia escala disponível do ADC. Outra vantagem da alimentação a 3,3V assenta no facto de ser mais comum e acessível encontrar baterias que alimentem a 3,3V do que a 5V. Repetiu-se assim o procedimento da calibração estática, desta vez para uma calibração de 3,3V. Os resultados desta calibração encontram-se na Tabela 8 e no Gráfico 3.

Tabela 8 - Resultados da calibração estática do *FSS Low Profile Force Sensors*, a operar a 3,3V e ligado á instrumentação desenvolvida no PSoC®.

Massa Aplicada (gramas)	Valor do ADC	valor ADC rectificado	Valor voltagem (mV)	Intervalo de massa (g)	Intervalos mV	Sensibilidade (mV/g)
0	131	0	0,00	21,8	25,78	1,18
21,8	133	2	25,78	91,1	90,23	0,99
112,9	140	9	116,02	42,3	77,34	1,83
155,2	146	15	193,36	90,4	103,13	1,14
245,6	154	23	296,48	54,9	38,67	0,70
300,5	157	26	335,16	78,4	141,80	1,81
378,9	168	37	476,95	57,5	64,45	1,12
436,4	173	42	541,41	75,9	103,13	1,36
512,3	181	50	644,53	58,7	77,34	1,32
571	187	56	721,88			
Sensibilidade Media:						1,27
						0,08

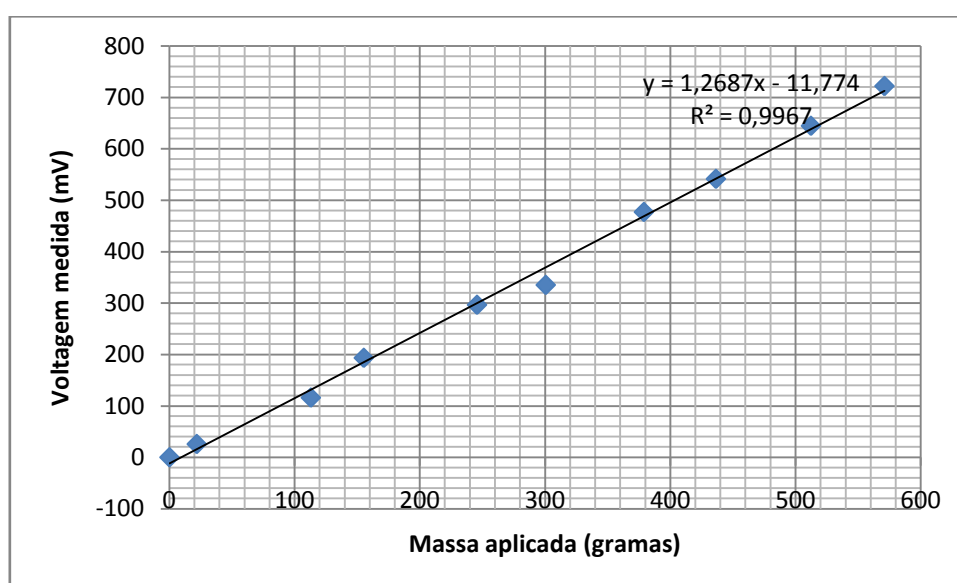


Gráfico 3 - Resultados da calibração estática do *FSS Low Profile Force Sensors*, a operar a 3,3V e ligado á instrumentação desenvolvida no PSoC®.

A sensibilidade estática para o sensor *FSS*, quando alimentado a 3,3V desce para 0,0795 mV/g, o que sucede devido à saída ratiométrica do sensor (saída proporcional ao valor de alimentação). No entanto, apesar da sensibilidade do sensor ter baixado, baixou também a tensão do sinal comum nas entradas. Assim, o sinal de saída do sensor alimentado a 3,3V permite na configuração criada utilizar a totalidade da meia escala disponível no ADC e utilizar um ganho de 16 vezes sob o sinal diferencial. Podemos então registar até 12,72N com um ganho de 16 vezes, um valor de força que excede ligeiramente os 10N pedidos para este projecto.

A resposta do sensor perante a carga aplicada mantém-se linear, factor mais uma vez provado pelo elevado valor do factor de correlação linear 0,996 e também

pela não linearidade de $\pm 1,79\%$, que se mostra um pouco superior á obtida quando o sensor é alimentado a 5V. O erro associado a cada medição é 1 LSB, ou seja $\pm 0,099$ Newton ($\pm 10,13$ grama), um valor também mais alto do que quando o sensor é alimentado a 5V. Estes resultados são justificados pelo facto de as tensões de referência do ADC permanecerem as mesmas, o que faz com que a resolução do ADC de 8 bits permaneça igual. Ao diminuir a sensibilidade do sensor FSS, o mesmo patamar de 12,89 mV/bit do ADC vai necessitar de uma resposta mais alta por parte do sensor para ser alcançado, fazendo com que a resolução final do dispositivo seja agora de 10,13 grama.

Tempo de Aquisição de 50 minutos contínuos

Para este teste da instrumentação desenvolvida, a aquisição foi monitorizada na SCD e, após 50 minutos de aquisição e monitorização, o sistema foi parado e os dados obtidos pela SCD exportados para a interface *ForceGlv* onde foram analisados. O objectivo era verificar se a instrumentação criada suporta o armazenamento e tratamento de um conjunto de dados resultante de uma aquisição de 50 minutos. O teste teve um resultado positivo. A SCD armazena o conjunto de dados adquirido durante os 50 minutos num ficheiro *.CSV e permite exportá-lo para o MATLAB® que também consegue manipular matrizes com estas dimensões. A taxa de amostragem da instrumentação desenvolvida ronda as 30 amostras por segundo, o que numa aquisição de 50 minutos resulta em 90 000 amostras. O tempo de aquisição não foi testado para valores superiores a 50 minutos, pelo que é recomendado exportar os dados obtidos após os períodos de 50 minutos previstos para os períodos de estudo contínuo com esta instrumentação.

Multiplexagem

Para testar a multiplexagem foi ligado o sensor FSS num dos dois lugares disponíveis na placa FTRF. No outro lugar disponível para um segundo sensor foi ligada uma fonte de tensão DC. Este teste tem a finalidade de demonstrar que um Nodo pode suportar mais do que um sensor e também que pode ser construída no PSoC® uma plataforma que adquira os dados de quatro sensores de força, indo de encontro aos objectivos desta instrumentação. Mais uma vez a aquisição foi obtida através da SCD. O resultado deste teste foi positivo. A multiplexagem funcionou, embora se tenha registado algum *cross-talk* entre os dois canais testados, cuja origem não foi confirmada.

Operar com dois Nodos em simultâneo

Foram configuradas duas placas FTRF com o *firmware ForceFingerTips* e foram ambas ligadas ao mesmo *Hub*, configurado com a solução da Cypress RF_HUB. Como era previsto o Hub permite ligar os dois Nodos necessários para esta aplicação. No entanto foi notória uma diminuição na taxa de amostragem de cada sinal de força quando comparada com a taxa de amostragem quando está apenas um Node ligado ao Hub. Com dois Nodos, o Hub consegue obter apenas 15 amostras por segundo (ver Figura 41) por Nodo, enquanto com um Nodo consegue obter 30 amostras por segundo. Ora para colmatar esta quebra é necessário aumentar a velocidade a que o Hub processa os dados para o dobro. Verificou-se que ao aumentar a velocidade para o dobro, ou seja 6 MHz, o Hub passa a adquirir 60 pacotes de dados por segundo no lugar dos anteriores 30 pacotes de dados, permitindo assim que cada Nodo mantenha a sua taxa de amostragem de 30 amostras por segundo.

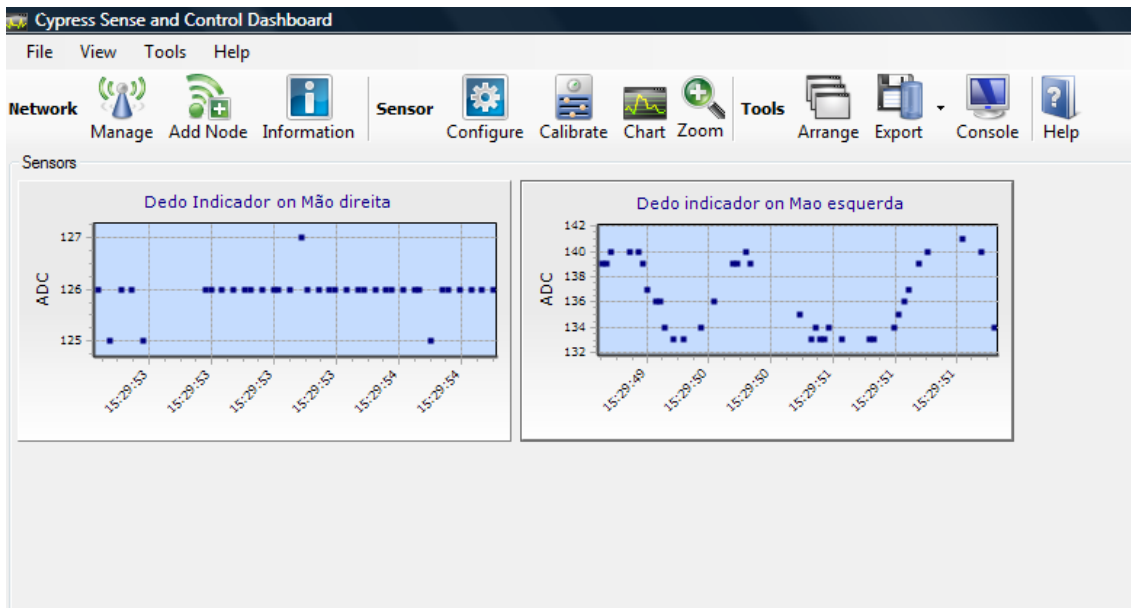


Figura 41 – A monitorização na SCD com dois Nodos ligados ao *Hub*. Aqui o *Hub* encontra-se a operar a 3 MHz.

Conclusões Finais

A instrumentação desenvolvida prova servir como base para se obter uma solução personalizada, capaz de superar os constrangimentos técnicos e motores associados às soluções comuns existentes no mercado para este tipo de aplicação. Através da implementação do sistema de acondicionamento e aquisição dos sinais de força no PSoC®, reduziu-se o número de componentes na PCB, reduzindo-se também a dimensão e o peso desta solução. A implementação do protocolo de comunicação sem fios CyFi™ SNP, permite substituir os vários cabos que em soluções anteriores ligariam directamente os sensores a um módulo de aquisição pesado junto de um PC, e que retiravam mobilidade ao sujeito em estudo. Nesta solução é apenas necessário ligar um *Hub* de pequenas dimensões numa porta USB do PC. O *Hub* obterá os dados enviados pelos Nodos (terminais de aquisição em cada uma das mãos do sujeito em estudo), através da rede sem fios. A aplicação criada em MATLAB® trata os dados adquiridos, que podem ser monitorizados e armazenados em tempo real no PC através do software SCD.

Os objectivos propostos inicialmente para este projecto, apesar de não terem culminado na construção do protótipo de uma luva que adquira o valor de força ao teclear, foram cumpridos por meio do desenvolvimento da base tecnológica com a qual se poderá obter a solução final. A solução criada peca apenas por possuir recursos limitados, o que resulta em grande parte por consistir numa adaptação de um kit de apresentação da Cypress e não numa solução particularizada criada de raiz. No entanto, foi desenvolvida uma solução no kit CY3271, capaz de obter o valor de força de 2 sensores FSS (por Nodo) alimentados a 3,3V, e que provou registar forças de 0 a 12 Newton com um erro associado de $\pm 0,099$ Newton. O sistema criado é leve e portátil, alimentado por duas pilhas AAA. Permite também monitorizar os dados adquiridos através do *Hub* e do software SCD de onde são exportados para futuro tratamento numa interface criada em MATLAB®.

Trabalho Futuro

A dimensão dos Nodos pode ser ainda mais reduzida do que a solução aqui apresentada na placa FTRF do kit CY3271. Esta redução é atingível através de uma tecnologia recente também criada pela Cypress e designada por P_{RoC}[®] (*Programmable Radio on Chip*). Esta tecnologia consiste num *chip* único que inclui um P_{SoC}[®] e um *transceiver* CyFi™, que para operar correctamente apenas necessita externamente de uma antena, de algumas resistências, condensadores e da fonte de alimentação. O desenvolvimento deste projecto no P_{SoC}[®] CY8C27443 mostra que este projecto pode ser facilmente generalizado para os 4 sensores pedidos por mão (Nodo), cumprindo os objectivos traçados. Falta portanto construir uma solução particularizada para este projecto, contendo um P_{SoC}[®] e um CyFi[®] SNP, ou como já vimos um P_{RoC}[®] desde que estejam disponíveis todos os recursos internos do *chip*, o que não acontece nas placas FTRF. A solução final ao possuir todos os recursos do P_{SoC}[®], poderá para além de generalizar a solução para 4 sensores, otimizar o ganho no sinal diferencial para a gama de forças de interesse e aumentar a resolução do ADC por manipulação externa dos valores das tensões de referência deste, diminuindo assim o erro associado a cada medição nesta instrumentação.

Para se obter uma solução final mais robusta deve criar-se também uma solução de Hub particularizada, que não necessite do software SCD e que portanto permita a aquisição e monitorização directamente por uma aplicação MATLAB[®]. No que ao P_{SoC}[®] diz respeito, existem soluções que incluem já como recurso a ligação a uma porta USB e às quais facilmente se adapta o protocolo CyFi™ através de um *radio transceiver*. Fica também o desafio de criar na ponta do dedo um engenho/plataforma de pequenas dimensões que permita ajustar o posicionamento do sensor para cada sujeito, para que a força seja aplicada o menor número de vezes possível fora do eixo principal de medição do sensor, otimizando assim a aquisição do valor de força por parte do sensor FSS.

Anexo I – O código do *firmware ForceFingerTips*

```
//-----  
// C main file  
//-----  
  
#include <m8c.h>  
// part specific constants and macros  
#include "PSOCAPI.h"  
// PSoC API definitions for all User Modules  
#include "delay.h"  
// header file for delay functions, I use the function Delay50uTimes()  
// for debouncing  
#include "PSOCGPIOINT.h"  
  
//-----  
// Define's  
//-----  
  
// code for LED's manipulation  
#define RED_LED_ON (PRT1DR |= 0x04)  
#define RED_LED_OFF (PRT1DR &= ~0x04)  
#define GREEN_LED_ON (PRT1DR |= 0x08)  
#define GREEN_LED_OFF (PRT1DR &= ~0x08)  
  
//-----  
// Functions Prototypes  
//-----  
BOOL Check_bind_SW1(void);  
BOOL Check_bind_SW2(void);  
void Calibrate_ILO(void);  
void Check_Data_Rx(void);  
void GetForceReport(void);  
  
//-----  
// Local variables  
//-----  
WORD sleepTicksPerSec = 64; // sleep timer ticks to reach 1 second  
WORD oneSecTimer;  
WORD timerCount;  
static CYFISNP_1_API_PKT *pRxApiPkt; // pointer to (Rx Packet API)  
static CYFISNP_1_API_PKT txApiPkt; // (Tx Data Buffer)  
int ADCForce1;  
int ADCForce2;  
int NewtonForce1;  
int NewtonForce2;  
static char txPktSz;  
char pktType = CYFISNP_1_API_TYPE_CONF_BCDR;
```

```

//-----
// Main Routine
//-----
void main()
{
PRT1DR=0x00;           // pull down enabled
RED_LED_ON;
M8C_EnableGInt;      // enable global interrupts
// start blocks
CYFISNP_1_Start();
PGA_1_Start(PGA_1_MEDPOWER);
PGA_2_Start(PGA_2_MEDPOWER);
ADCINC_1_Start(ADCINC_1_MEDPOWER);
//clocks to establish periodic events (Calibration)
CYFISNP_1_TimeSet(&oneSecTimer,sleepTicksPerSec);

//-----
// Polling loop
//-----
while(1)
{
    if (Check_bind_SW1())
    {
        CYFISNP_1_Start();
        CYFISNP_1_BindStart(0x00);
        GREEN_LED_ON;
    }
    CYFISNP_1_Run();
    if (CYFISNP_1_eProtState == CYFISNP_1_DATA_MODE)
    {
        RED_LED_OFF;
        GetForceReport();
    }
    if (CYFISNP_1_TimeExpired (&oneSecTimer)==TRUE)
    {
        Calibrate_ILO();
    }
    Check_Data_Rx();
    if (Check_bind_SW2())
    {
        PGA_1_Stop();
        PGA_2_Stop();
        ADCINC_1_StopADC();
        CYFISNP_1_Stop();
        GREEN_LED_OFF;
    }
}
}

//-----
// Functions
//-----

//-----
// BOOL Check_bind_SW1(void)
// Routine to verify if the SW1 was pressed
// (returns TRUE if it was pressed, returns FALSE if not)
//-----

```

```

BOOL Check_bind_SW1(void)
{
    if (PRT1DR & 0x40)
    {
        Delay50uTimes(100);           // Debouce time, 5ms
        if (PRT1DR & 0x40)
        {
            while(PRT1DR & 0x40)
            {
                // Waits for button release
            }
            return (TRUE);
        }
        else
            return (FALSE);
    }
}

//-----
// BOOL Check_bind_SW2(void)
// Routine to verify if the SW2 was pressed
// (returns TRUE if it was pressed, returns FALSE if not)
//-----
BOOL Check_bind_SW2(void)
{
    if (PRT1DR & 0x10)
    {
        Delay50uTimes(100);           // Debouce time, 5ms
        if (PRT1DR & 0x10)
        {
            while(PRT1DR & 0x10)
            {
                // Waits for button release
            }
            return (TRUE);
        }
        else
            return (FALSE);
    }
}

//-----
// void Calibrate_ILO(void)
// Routine to calibrate the ILO that is used as reference for the
// sleep timer, and so by CYFISNP, this routine calibrates the ILO
// with the IMO.
// (does not return any value)
//-----
void Calibrate_ILO(void)
{
    // Get number of ILO ticks in 1 mS (as measured by IMO) (assuming
    // ILO(wake) = ILO(sleep))
    Timer8_1_WritePeriod(255);
    // maximun value allowed for the period
    M8C_DisableGInt;
    Timer8_1_Start();

    for (timerCount=0; timerCount != 10; ++timerCount)// Delay 1ms
    {
        CYFISNP_1_Delay100uS();
        // Function created to generate the delays (based on the IMO)
    }
}

```



```

    }
    timerCount = Timer8_1_bReadTimer();
    // reads the actual value in Count register
    timerCount = 255 - timerCount;
    // reads how many periods are present in lms timer, it should read
    // 32 periods, since the clock source is 32kHz
    Timer8_1_Stop();
    M8C_EnableGInt;
    sleepTicksPerSec = timerCount<<1;
    sleepTicksPerSec -= timerCount>>1;
    // Decrease by 25% to estimate ILO
    CYFISNP_1_TimeSet(&oneSecTimer,sleepTicksPerSec);
}

//-----
// void Check_Data_Rx(void)
// Function to check and process the data received
// (does not return any value)
//-----
void Check_Data_Rx(void)
{
    if (CYFISNP_1_RxDataPend() == TRUE)
    {
        pRxApiPkt = CYFISNP_1_RxDataGet();
        CYFISNP_1_RxDataRelease();
    }
}

//-----
// void GetForceReport(void)
// Routine to obtain the Force value and to send it to the Hub
// (does not return any value)
//-----
void GetForceReport(void)
{
    //-----
    // transmission packet Size
    //-----
    txPktSz = 5;
    //-----
    // function
    //-----
    if (CYFISNP_1_eProtState == CYFISNP_1_DATA_MODE)
    {
        // Analog Multiplexing to Finger 1
        AMUX4_1_InputSelect(AMUX4_1_PORT0_5);
        AMUX4_2_InputSelect(AMUX4_2_PORT0_4);
        ADCINC_1_GetSamples(1);
        while (ADCINC_1_fIsDataAvailable() == 0);
        // waiting while the sample is being processed
        ADCForce1 = ADCINC_1_wClearFlagGetData();
        // cleans the data ready flag and gets the ADC result

        // Analog Multiplexing to Finger 2
        AMUX4_1_InputSelect(AMUX4_1_PORT0_3);
        AMUX4_2_InputSelect(AMUX4_2_PORT0_2);
        ADCINC_1_GetSamples(1);
        while (ADCINC_1_fIsDataAvailable() == 0);

        ADCForce2 = ADCINC_1_wClearFlagGetData();
    }
}

```

```

// Code to calculate the Force in Newtons

NewtonForce1=ADCForce1;
NewtonForce2=ADCForce2;

// Load tx buffer payload bytes with force value to be
// transmitted
txApiPkt.length = txPktSz;
// Specify Tx length
txApiPkt.type    = pktType;

// Payload
int *loadValue1 = &NewtonForce1;
int *loadValue2 = &NewtonForce2;

txApiPkt.payload[1] = *(loadValue2+0);
txApiPkt.payload[0] = *(loadValue1+0);

CYFISNP_1_TxDataPut (&txApiPkt);
}
else
{
    CYFISNP_1_Jog();
}
}

//-----
// End of C main file
//-----

```

Anexo II – O código da Aplicação Gráfica *ForceGlv*

```
function ForceGlv
% ForceGlv é uma interface que pretende adquirir oito sinais de
% força em formato *csv e tratá-los de modo a estabelecer relações
% e oferecer tratamento estatístico destes oito sinais.

% Iniciar variáveis
scrsz = get(0,'ScreenSize');
Dta_Ind_Tempo = [];
Dta_Med_Tempo = [];
Dta_An1_Tempo = [];
Dta_Min_Tempo = [];
Esq_Ind_Tempo = [];
Esq_Med_Tempo = [];
Esq_An1_Tempo = [];
Esq_Min_Tempo = [];
row1 = {'Valor (Newtons)'};
cnames1 = {'Média Geométrica', 'Média Harmónica', 'Média
           Aritmética', 'Mediana', 'Moda', 'Trimed Mean'};
row2 = {'Valor (Newtons)'};
cnames2 = {'Interquartile range', 'Mad', 'Range', 'Standard
           deviation', 'Variance'};

row3={'Percentis (Newtons)'};
cnames3 = {'2.5', '25', '50', '75', '97.5'};
actual_selected_data = [];
actual_zerofree = [];

% constrói o GUI
% Lista dos Handles

background = figure('Visible','on','color',[0 0 0],
    'DockControls','off','menubar','none','name','Interface da
    Instrumentação para Registo da Força', 'NumberTitle',
    'off','resize','off','outerposition',
    [1 1 scrsz(3) scrsz(4)]);
panell = uipanel('parent',background,'visible','on',
    'BackgroundColor',[0 1 0],'bordertype','beveledout',
    'fontsize',10,'FontWeight','bold','ForegroundColor',
    [0 0 0],'title','Objecto de Estudo','TitlePosition',
    'centertop','position',[0.05 0.77 1/4 1/8]);
titulo = uicontrol('parent',background,'style','text','string',
    'FINGER TIPS FORCE SENSING','fontsize',17,'fontname',
    'comic geek','backgroundcolor',[0 0 0],'ForegroundColor',
    [0 1 0],'position',[420 700 400 30]);
Objpopup = uicontrol('callback',{@popup_ObjEst_Callback},'parent',
    panell,'style','popupmenu','string',{'Mao Dta, Dedo
    Indicador' 'Mao Dta, Dedo Medio' 'Mao Dta, Dedo Anelar'
    'Mao Dta, Dedo Minimo' 'Mao Esq, Dedo Indicador' 'Mao Esq,
    Dedo Medio' 'Mao Esq, Dedo Anelar' 'Mao Esq, Dedo Minimo'
    'Mao Dta, todos os Dedos' 'Mao Esq, todos os Dedos' 'Duas
    maos, todos os Dedos'},'position',[60 35 200 20],
    'fontsize',8);
menu = uimenu('parent',background,'label','Carregar Dados',
    'separator','on');
Dta1 = uimenu(menu,'label','Carregar dados do dedo Indicador
    Direito','callback',{@Dta1_Callback});
```

```

Dta2 = uimenu(menu,'label','Carregar dados do dedo Médio
Direito','separator','off','callback',{@Dta2_Callback});
Dta3 = uimenu(menu,'label','Carregar dados do dedo Anelar
Direito','separator','off','callback',{@Dta3_Callback});
Dta4 = uimenu(menu,'label','Carregar dados do dedo Mínimo
Direito','separator','off','callback',{@Dta4_Callback});
Esq1 = uimenu(menu,'label','Carregar dados do dedo Indicador
Esquerdo','separator','off','callback',{@Esq1_Callback});
Esq2 = uimenu(menu,'label','Carregar dados do dedo Médio
Esquerdo','separator','off','callback',{@Esq2_Callback});
Esq3 = uimenu(menu,'label','Carregar dados do dedo Anelar
Esquerdo','separator','off','callback',{@Esq3_Callback});
Esq4 = uimenu(menu,'label','Carregar dados do dedo Mínimo
Esquerdo','separator','off','callback',{@Esq4_Callback});
panel2 = uipanel('parent',background,'visible','on',
'BackgroundColor',[0 1 0],'bordertype','beveledout','fontsize',
10,'FontWeight','bold','ForegroundColor',[0 0 0],'title','Sinal
Inicial Seleccionado','TitlePosition','centertop','position',
[0.05 0.40 1/4 1/3]);
grafico1 = axes('parent',panel2,'ActivePositionProperty',
'position','color',[0 1 0]);
panel3 = uipanel('parent',background,'visible','on',
'BackgroundColor',[0 1 0],'bordertype','beveledout','fontsize',
10,'FontWeight','bold','ForegroundColor',[0 0 0],'title',
'Estatística','TitlePosition','centertop','position',
[0.05 0.05 0.40 0.30]);
tabela1 = uitable('parent',panel3,'BackgroundColor',[0 1 0],
'ColumnName',cnames1,'rowname',row1,'position',[20 135 465 57]);
tabela2 = uitable('parent',panel3,'BackgroundColor',[0 1 0],
'ColumnName',cnames2,'rowname',row2,'position',[20 75 465 57]);
tabela3 = uitable('parent',panel3,'BackgroundColor',[0 1 0],
'ColumnName',cnames3,'rowname',row3,'position',[20 15 465 57]);
panel4 = uipanel('parent',background,'visible','on',
'BackgroundColor',[0 1 0],'bordertype','beveledout',
'fontsize',10,'FontWeight','bold','ForegroundColor',[0 0 0],
'title','Visualização Gráfica','TitlePosition','centertop',
'position',[0.33 0.395 0.65 0.50]);
grafico2 = axes('parent',panel4,'ActivePositionProperty',
'position','color',[0 1 0]);
grupobutton = uibuttongroup('parent',background,'BackgroundColor',
[0 1 0],'bordertype','beveledout','fontsize',10,'FontWeight',
'bold','ForegroundColor',[0 0 0],'title','Tratamento',
'TitlePosition','centertop','position',[0.48 0.05 0.50 0.30],
'SelectionChangeFcn',@butoes);
botao_zero = uicontrol('parent',grupobutton,'style','radiobutton',
'string','Remover Offset + tempo morto (OT)','backgroundcolor',
[1 0 0],'position',[40 160 200 20],'tag','zero');
botao_rms = uicontrol('parent',grupobutton,'style','radiobutton',
'string','Route Mean Square (rms) 15sp + OT','backgroundcolor',
[1 0 0],'position',[40 130 200 20],'tag','rms');
botao_max = uicontrol('parent',grupobutton,'style','radiobutton',
'string','Maximos (max) 15sp + OT','backgroundcolor',[1 0 0],
'position',[40 100 200 20],'tag','max');
botao_quartis =
uicontrol('parent',grupobutton,'style','radiobutton','string',
'Quartis + OT','backgroundcolor',[1 0 0],'position',
[40 70 200 20],'tag','quart');
botao_hist = uicontrol('parent',grupobutton,'style','radiobutton',
'string','Histograma + OT','backgroundcolor',[1 0 0],
'position',[40 40 200 20],'tag','hist');
botao_acum = uicontrol('parent',grupobutton,'style','radiobutton',

```

```

    'string','Probabilidade Acumulada + OT','backgroundcolor',
    [1 0 0],'position',[300 160 200 20],'tag','PA');
botao_temp = uicontrol('parent',grupobutton,'style','radiobutton',
    'string','Análise de médias no tempo + 0','backgroundcolor',
    [1 0 0],'position',[300 130 200 20],'tag','medtime');

% Inicializa o GUI

%-----
% Callbacks para o menu
%-----

function Dta1_Callback(Dta1,eventdata)
    [FileName,PathName,FilterIndex] = uigetfile('*.csv','Selecione
        os dados do dedo Indicador Direito');
    Dta_Ind_Tempo = opendata(FileName,PathName);
    if isempty(Dta_Ind_Tempo)==0
        set(Dta1,'Checked','on');
    end
end

function Dta2_Callback(Dta2,eventdata)
    [FileName,PathName,FilterIndex] = uigetfile('*.csv','Selecione
        os dados do dedo Médio Direito');
    Dta_Med_Tempo = opendata(FileName,PathName);
    if isempty(Dta_Med_Tempo)==0
        set(Dta2,'Checked','on');
    end
end

function Dta3_Callback(Dta3,eventdata)
    [FileName,PathName,FilterIndex] = uigetfile('*.csv','Selecione
        os dados do dedo Anelar Direito');
    Dta_An1_Tempo = opendata(FileName,PathName);
    if isempty(Dta_An1_Tempo)==0
        set(Dta3,'Checked','on');
    end
end

function Dta4_Callback(Dta4,eventdata)
    [FileName,PathName,FilterIndex] = uigetfile('*.csv','Selecione
        os dados do dedo Mínimo Direito');
    Dta_Min_Tempo = opendata(FileName,PathName);
    if isempty(Dta_Min_Tempo)==0
        set(Dta4,'Checked','on');
    end
end

function Esq1_Callback(Esq1,eventdata)
    [FileName,PathName,FilterIndex] = uigetfile('*.csv','Selecione
        os dados do dedo Indicador Esquerdo');
    Esq_Ind_Tempo = opendata(FileName,PathName);
    if isempty(Esq_Ind_Tempo)==0
        set(Esq1,'Checked','on');
    end
end

function Esq2_Callback(Esq2,eventdata)
    [FileName,PathName,FilterIndex] = uigetfile('*.csv','Selecione
        os dados do dedo Médio Esquerdo');
    Esq_Med_Tempo = opendata(FileName,PathName);
    if isempty(Esq_Med_Tempo)==0
        set(Esq2,'Checked','on');
    end
end

```

```

end
end
function Esq3_Callback(Esq3,eventdata)
    [FileName,PathName,FilterIndex] = uigetfile('*.csv','Selecione
                                                os dados do dedo Anelar Esquerdo');
    Esq_An1_Tempo = opendata(FileName,PathName);
    if isempty(Esq_An1_Tempo)==0
        set(Esq3,'Checked','on');
    end
end
function Esq4_Callback(Esq4,eventdata)
    [FileName,PathName,FilterIndex] = uigetfile('*.csv','Selecione
                                                os dados do dedo Mnimo Esquerdo');
    Esq_Min_Tempo = opendata(FileName,PathName);
    if isempty(Esq_Min_Tempo)==0
        set(Esq4,'Checked','on');
    end
end

% Objecto de Estudo pop-up Callback
function popup_ObjEst_Callback(Objpopup,eventdata)
    str=get(Objpopup,'string');
    val=get(Objpopup,'value');
    switch str{val};
        case 'Mao Dta, Dedo Indicador'
            axes(grafico1);
            Avaliar(Dta_Ind_Tempo);
            actual_selected_data = Dta_Ind_Tempo;
            actual_zerofree = removezeros(actual_selected_data);
            plot(actual_selected_data);
        case 'Mao Dta, Dedo Medio'
            axes(grafico1);
            Avaliar(Dta_Med_Tempo);
            actual_selected_data = Dta_Med_Tempo;
            actual_zerofree = removezeros(actual_selected_data);
            plot(actual_selected_data);
        case 'Mao Dta, Dedo Anelar'
            axes(grafico1);
            Avaliar(Dta_An1_Tempo);
            actual_selected_data = Dta_An1_Tempo;
            actual_zerofree = removezeros(actual_selected_data);
            plot(actual_selected_data);
        case 'Mao Dta, Dedo Minimo'
            axes(grafico1);
            Avaliar(Dta_Min_Tempo);
            actual_selected_data = Dta_Min_Tempo;
            actual_zerofree = removezeros(actual_selected_data);
            plot(actual_selected_data);
        case 'Mao Esq, Dedo Indicador'
            axes(grafico1);
            Avaliar(Esq_Ind_Tempo);
            actual_selected_data = Esq_Ind_Tempo;
            actual_zerofree = removezeros(actual_selected_data);
            plot(actual_selected_data);
        case 'Mao Esq, Dedo Medio'
            axes(grafico1);
            Avaliar(Esq_Med_Tempo);
            actual_selected_data = Esq_Med_Tempo;
            actual_zerofree = removezeros(actual_selected_data);
            plot(actual_selected_data);
    end
end

```

```

        case 'Mao Esq, Dedo Anelar'
            axes(grafico1);
            Avaliar(Esq_An1_Tempo);
            actual_selected_data = Esq_An1_Tempo;
            actual_zerofree = removezeros(actual_selected_data);
            plot(actual_selected_data);
        case 'Mao Esq, Dedo Minimo'
            axes(grafico1);
            Avaliar(Esq_Min_Tempo);
            actual_selected_data = Esq_Min_Tempo;
            actual_zerofree = removezeros(actual_selected_data);
            plot(actual_selected_data);
        case 'Mao Dta, todos os Dedos'
            axes(grafico1);
            Avaliar4(Dta_Ind_Tempo,Dta_Med_Tempo,Dta_An1_Tempo,Dta_Min_Tempo);
            actual_selected_data =
                [Dta_Ind_Tempo;Dta_Med_Tempo;Dta_An1_Tempo;Dta_Min_Tempo];
            actual_zerofree =
                [removezeros(Dta_Ind_Tempo);removezeros(Dta_Med_Tempo);removezer
                os(Dta_An1_Tempo);removezeros(Dta_Min_Tempo)];
            plot(actual_selected_data);
        case 'Mao Esq, todos os Dedos'
            axes(grafico1);
            Avaliar4(Esq_Ind_Tempo,Esq_Med_Tempo,Esq_An1_Tempo,Esq_Min_Tempo);
            actual_selected_data =
                [Esq_Ind_Tempo;Esq_Med_Tempo;Esq_An1_Tempo;Esq_Min_Tempo];
            actual_zerofree =
                [removezeros(Esq_Ind_Tempo);removezeros(Esq_Med_Tempo);removezer
                os(Esq_An1_Tempo);removezeros(Esq_Min_Tempo)];
            plot(actual_selected_data);
        case 'Duas maos, todos os Dedos'
            axes(grafico1);

            Avaliar8(Dta_Ind_Tempo,Dta_Med_Tempo,Dta_An1_Tempo,Dta_Min_Tempo,Esq_I
            nd_Tempo,Esq_Med_Tempo,Esq_An1_Tempo,Esq_Min_Tempo);
            actual_selected_data =
                [Dta_Ind_Tempo;Dta_Med_Tempo;Dta_An1_Tempo;Dta_Min_Tempo;Esq_Ind
                _Tempo;Esq_Med_Tempo;Esq_An1_Tempo;Esq_Min_Tempo];
            actual_zerofree =
                [removezeros(Dta_Ind_Tempo);removezeros(Dta_Med_Tempo);removezer
                os(Dta_An1_Tempo);removezeros(Dta_Min_Tempo);removezeros(Esq_Ind
                _Tempo);removezeros(Esq_Med_Tempo);removezeros(Esq_An1_Tempo);re
                movezeros(Esq_Min_Tempo)];
            plot(actual_selected_data);
    end
end

% selection change Fnc callback para detectar as mudançãs no
%painel de botoes
function botoes(source,eventdata)
    switch get(eventdata.NewValue,'Tag')
    % Get Tag of selected object.
    case 'zero'
        axes(grafico2);
        plot(actual_zerofree),ylabel('Newtons');
        [MCT_stat,MD_stat,Percentis] = estatistica
            (actual_zerofree);

        set(tabela1,'data',MCT_stat);
        set(tabela2,'data',MD_stat);
        set(tabela3,'data',Percentis);
    case 'rms'

```

```

rms = RMS15sp(actual_zerofree);
axes(grafico2);
plot(rms),ylabel('Valor RMS (Newtons)');
[MCT_stat,MD_stat,Percentis] = estatistica(rms);
set(tabela1,'data',MCT_stat);
set(tabela2,'data',MD_stat);
set(tabela3,'data',Percentis);
case 'max'
max = MAX15sp(actual_zerofree);
axes(grafico2);
plot(max),ylabel('Valor Máximo (Newtons)');
[MCT_stat,MD_stat,Percentis] = estatistica(max);
set(tabela1,'data',MCT_stat);
set(tabela2,'data',MD_stat);
set(tabela3,'data',Percentis);
case 'quart'
axes(grafico2);
boxplot(actual_zerofree),ylabel('Newtons');
[MCT_stat,MD_stat,Percentis] = estatistica
(actual_zerofree);
set(tabela1,'data',MCT_stat);
set(tabela2,'data',MD_stat);
set(tabela3,'data',Percentis);
case 'hist'
axes(grafico2);
hist(actual_zerofree,120),ylabel('Frequência
Absoluta'),xlabel('Newtons');
[MCT_stat,MD_stat,Percentis] = estatistica
(actual_zerofree);
set(tabela1,'data',MCT_stat);
set(tabela2,'data',MD_stat);
set(tabela3,'data',Percentis);
case 'PA'
axes(grafico2);
cdfplot(actual_zerofree),ylabel('Probabilidade');
[MCT_stat,MD_stat,Percentis] = estatistica
(actual_zerofree);
set(tabela1,'data',MCT_stat);
set(tabela2,'data',MD_stat);
set(tabela3,'data',Percentis);
case 'medtime'
axes(grafico2);
medtempo =
[mean(separa(removeoffset(actual_selected_data),1)),mean(separa(remove
offset(actual_selected_data),2)),mean(separa(removeoffset(actual_selec
ted_data),3)),mean(separa(removeoffset(actual_selected_data),4)),mean(
separa(removeoffset(actual_selected_data),5)),mean(separa(removeoffset
(actual_selected_data),6)),mean(separa(removeoffset(actual_selected_da
ta),7)),mean(separa(removeoffset(actual_selected_data),8)),mean(separa
(removeoffset(actual_selected_data),9)),mean(separa(removeoffset(actua
l_selected_data),10))];
plot(medtempo),ylabel('Média dos Períodos
(Newtons)'),xlabel('Espaços de tempo');
[MCT_stat,MD_stat,Percentis] = estatistica(medtempo);
set(tabela1,'data',MCT_stat);
set(tabela2,'data',MD_stat);
set(tabela3,'data',Percentis);
end
end
end

```


Anexo III – O código das Funções criadas para a Aplicação Gráfica

ForceGlv

```
function Avaliar(parametro)
% Avalia se um ficheiro foi carregado ou não
    if isempty(parametro)
        h = msgbox('Estes dados não constam (total ou parcialmente) na
                    base de dados, verifique se carregou todos os
                    dados.','Falta de dados','warn');

        uiwait(h);
    end
end
```

```
function Avaliar4(dedo1,dedo2,dedo3,dedo4)
% Avaliar a existencia de informação dos 4 dedos
    if (isempty(dedo1) || isempty(dedo2) || isempty(dedo3) ||
        isempty(dedo4))
        h = msgbox('Estes dados não constam (total ou parcialmente) na
                    base de dados, verifique se carregou todos os
                    dados.','Falta de dados','warn');

        uiwait(h);
    end
end
```

```
function Avaliar8(dedo1,dedo2,dedo3,dedo4,dedo5,dedo6,dedo7,dedo8)
% Avaliar a existencia ded dados dos 8 dedos
    if (isempty(dedo1) || isempty(dedo2) || isempty(dedo3) ||
        isempty(dedo4) || isempty(dedo5) || isempty(dedo6) || isempty(dedo7)
        || isempty(dedo8))
        h = msgbox('Estes dados não constam (total ou parcialmente) na
                    base de dados, verifique se carregou todos os
                    dados.','Falta de dados','warn');

        uiwait(h);
    end
end
```

```
function [MCT_stat,MD_stat,Percentis] = estatistica(current_data)
% Função que executa o tratamento estatístico dos dados provenientes
% do sensor de força.
    MCT_stat = [geomean(current_data) harmmean(current_data)
                mean(current_data) median(current_data) mode(current_data)
                trimmean(current_data,30)];
    MD_stat = [iqr(current_data) mad(current_data) range(current_data)
                std(current_data) var(current_data)];
    Percentis = prctile(current_data,[2.5 25 50 75 97.5]);
end
```

```
function [maximos15sp] = MAX15sp(Finger_data_final)
% maximos da força aplicada
% analisa 15 samples (500ms) e junta o max a uma lista de maximos
    maximos15sp=[];
    for n=1:(length(Finger_data_final)/15)-1
        segmentos=[];
        id=1;
```

```

        for ss=n*15:n*15+15
            if ss<length(Finger_data_final)
                segmentos(id)=Finger_data_final(ss,1);
                id=id+1;
            end
        end
        maximos15sp(n)=max(segmentos);
    end
end

function [s_offset] = removeoffset(c_offset)
% Remove o offset em Newtons
    offset=round(mean(c_offset(1:60)));
    s_offset=c_offset-offset;
end

function [Finger_data_final] = removezeros(Finger_data_nonsupressed)
% Função que remove ao conjunto de dados o valor registado pelo
dispositivo
% quando o sujeito não exerce qualquer tipo de força.
% O valor zero é obtido na média das primeiras 60 amostras (2
segundos).
% NOTA: O sujeito não deve tocar em nada nos primeiros segundos em que
se
% inicia a aquisição.

    %Obter o valor ADC para o zero
    valorzero = round(mean(Finger_data_nonsupressed(1:60)));
    offset=valorzero;
    %Obter os dados finais o valor ADC para o zero
    Finger_data_final = [];
    idx=1;
    treshold=min(Finger_data_nonsupressed)+1;
    for x=1:length(Finger_data_nonsupressed)
        if Finger_data_nonsupressed(x,1)>valorzero
            Finger_data_final(idx,1)=Finger_data_nonsupressed(x,1);
            idx=idx+1;
        end
    end
    Finger_data_final=Finger_data_final-offset;
end

function [valorms15sp] = RMS15sp(Finger_data_final)
% Função que calcula o valor rms de 15 em 15 amostras e retorna um
% vector com os resultados.
    valorms15sp=[];
    for n=1:(length(Finger_data_final)/15)-1
        segmentos=[];
        id=1;
        for ss=n*15:n*15+15
            if ss<length(Finger_data_final)
                segmentos(id)=Finger_data_final(ss,1);
                id=id+1;
            end
        end
        somatorio=0;
        for inc=1:length(segmentos)
            somatorio=somatorio+segmentos(1,inc)^2;
        end
    end
end

```

```

        end
        valorms15sp(n)=sqrt(somatorio/length(segmentos));
    end
end

function [Finger_data_nonsupressed] = opendata(filename,currentpath)
% Função que abre e prepara os ficheiros com extensão *csv para o
% tratamento.
% Utiliza o valor de conversão 0.09934 válido para o FSS a 3,3V de
% alimentação
    H = waitbar(0, 'A processar os dados seleccionados. Por favor
    aguarde.', 'name', 'Estado');
    try
        completepath = strcat(currentpath,filename);
        Finger_data = csvimport(completepath);
        Finger_data_nonsupressed=[];
        z=length(Finger_data)-8;
        for x=1:z
            y=rsplit(';', char(Finger_data(8+x)));
            Finger_data_nonsupressed(x,1)=0.09934*str2num(char(y(2)));
            waitbar(x/z);
        end
    catch
        delete(H);
    end
    delete(H);
end

```

Force Sensors

FSS Low Profile Force Sensors

FS Series

FEATURES

- True Surface Mount Technology
- Maximum peak reflow temperature of 260 °C [500 °F]
- Compact, commercial grade package
- Robust performance characteristics
- Adaptable package design
- Precision force sensing
- Reliability rated at 20 million MCTF at 25 °C [77 °F]
- Electrically ratiometric output
- Extremely low deflection (30 microns typ. @ Full Scale)
- High ESD resistance 8 kV
- Available signal conditioning
- Optional terminal configurations

TYPICAL APPLICATIONS

- Medical infusion pumps
- Ambulatory noninvasive pump pressure
- Occlusion detection
- Kidney dialysis machines
- Load and compression sensing
- Variable tensions control
- Robotic end-effectors
- Wire bonding equipment



The FS Series sensors provide precise reliable force sensing performance in a compact commercial grade package at a cost effective price. The sensor features a proven sensing technology that uses a specialized piezoresistive micromachined silicon sensing element. The low power, unamplified, uncompensated Wheatstone bridge circuit design provides inherently stable mV outputs over the force range.

Force sensors operate on the principle that the resistance of silicon-implanted piezoresistors will increase when the resistors flex under any applied force. The sensor concentrates force from the applications, through the stainless steel ball, directly to the silicon-sensing element. The amount of resistance changes in proportion to the amount of force being applied. This change in circuit resistance results in a corresponding mV output level change.

The sensor package design incorporates patented modular construction. The use of innovative elastomeric technology and engineered molded plastics result in load excitation capacities of 4.5/5.5 kg over-force. The stainless steel ball provides excellent mechanical stability and is adaptable to a variety of applications. The FSS sensor delivered 20 million operations in Mean Cycles to Failure (MCTF) reliability testing at 50 °C [122 °F]. This test determines the number of possible sensor operations at full scale until failure. Various electric interconnects can accept prewired connectors, printed circuit board mounting, and surface mountings. The unique sensor design also provides a variety of mounting options that include mounting brackets, as well as application specific mounting requirements.

⚠ WARNING
PERSONAL INJURY
 DO NOT USE these products as safety or emergency stop devices or in any other application where failure of the product could result in personal injury. Failure to comply with these instructions could result in death or serious injury.

⚠ WARNING
MISUSE OF DOCUMENTATION

- The information presented in this product sheet is for reference only. Do not use this document as a product installation guide.
- Complete installation, operation, and maintenance information is provided in the instructions supplied with each product.

Failure to comply with these instructions could result in death or serious injury.

Force Sensors

FSS Low Profile Force Sensors

FS Series

PERFORMANCE CHARACTERISTICS @ 5.0 ± 0.01 Vdc Excitation*, 25 °C [77 °F]

Parameter	Min.	Typical	Max.	Units
Null Offset	-15	0	+15	mV
Operating Force	0	-	1500	grams
Sensitivity	0.1	0.12	14	mV/gram
Linearity (B.F.S.L.)**	-	± 1.5	-	% span
Repeatability @ 900 g	-	± 10	-	grams
Null Shift				
25 °C to 2 °C [77 °F to 35.6 °F]	-	± 0.5	-	mV
25 °C to 40 °C [77 °F to 104 °F]	-	± 0.5	-	mV
Sensitivity Shift				
25 °C to 50 °C [77 °F to 122 °F]	-	5.5	-	% span
25 °C to 0 °C [77 °F to 32 °F]	-	5.5	-	% span
Input Resistance	4.0 K	5.0 K	6.0 K	Ohms
Output Resistance	4.0 K	5.0 K	6.0 K	Ohms
Overforce	-	-	4,500	grams
ESD (direct contact, terminals and plunger)	8	-	-	KV

* Non-compensated force sensors, excited by constant current (1.5 mA) instead of voltage, exhibit partial temperature compensation of Span.

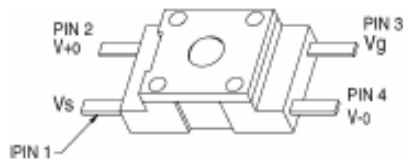
** B.F.S.L.: Best Fit Straight Line

ENVIRONMENTAL SPECIFICATIONS

Operating temperature	-40 °C to 85 °C [-40 °F to 185 °F]
Storage temperature	-40 °C to 100 °C [-40 °F to 212 °F]
Shock	Qualification tested to 150 g
Vibration	Qualification tested to 0 to 2 kHz, 20 g sine
MCTF	20 million at 25 °C [77 °F]
Solderability	5 sec at 315 °C [599 °F] per lead
Output ratiometric	Within supply range

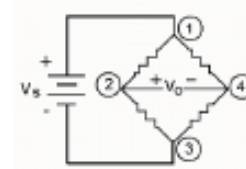
Note: All force related specifications are established using dead weight or compliant force.

SENSOR PINOUT



EXCITATION SCHEMATIC

Excitation 5 Vdc Typ., 12 Vdc Max.



FS SERIES CIRCUIT

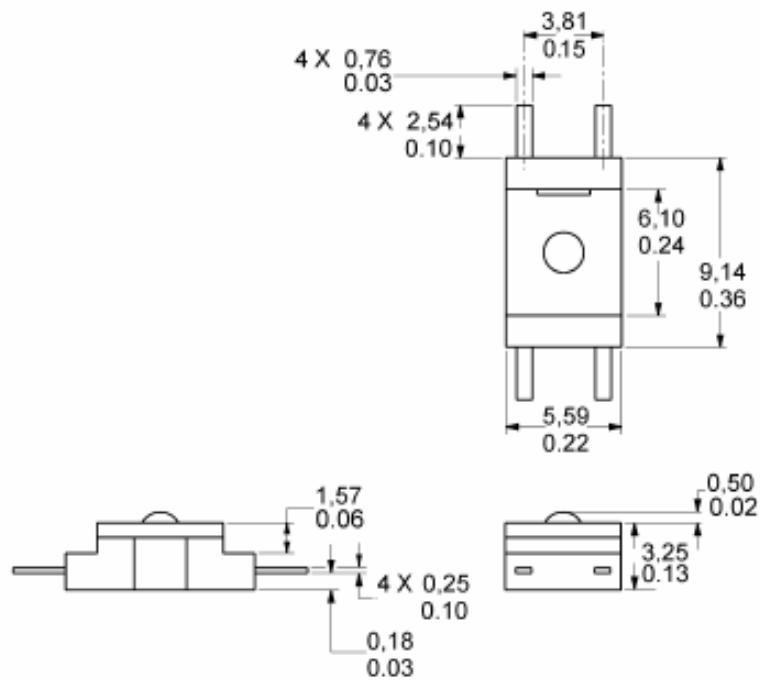
1. Circled numbers refer to sensor terminals (pins).
Pin 1 = Supply V_s (+)
Pin 2 = Output V_o (+)
Pin 3 = Ground V_g (-)
Pin 4 = Output V_o (-)
2. The force sensor may be powered by voltage or current. Maximum supply voltage is not to exceed 12 volts. Maximum supply current is not to exceed 1.6 mA. Power is applied across Pin 1 and Pin 3.
3. The sensor output should be measured as a differential voltage across Pin 2 and Pin 4 ($V_o = V_2 - V_4$). The output is ratiometric to the supply voltage. Shifts in supply voltage will cause shifts in output. Neither Pin 2 nor Pin 4 should be tied to ground or voltage supply.

Force Sensors

FSS Low Profile Force Sensors

FS Series

MOUNTING DIMENSIONS (for reference only) mm/in



DESCRIPTION

Catalog Listing	Packing Style
FSS1500NST	Tube
FSS1500NSB	Bubble Pack
FSS1500NSR	Tape and Reel

Bibliografia e Webgrafia

Capítulo 1:

- [1] Reuters. (23 Junho de 2008). Computers In Use Pass 1 Billion Mark, Gartner Says. *InformationWeek* .
- [2] Miniwatts Marketing Group. (s.d.). www.internetworldstats.com. Obtido em 10 de Junho de 2009.
- [3] Media release . (6 de Dezembro de 2007). *World Leading Research into Computer Released at International Conference*. Christchurch, New Zealand.
- [4] Marcus M, Gerr F., Monteilh C, Ortiz D, Gentry E, Cohen S, Edwards A, Ensor C, Kleinbaum D. (2002). A prospective study of computer users: II Postural Risk Factors for Musculoskeletal Symptoms and Disorders. *American Journal of Industrial Medicine* 41 , pp. 236-249.
- [5] Gerr F, M. M. (2004). Epidemiology of musculoskeletal disorders among computer users: lesson learned from the role of posture and keyboard use. *Journal of Electromyography and Kinesiology* 14 , pp. 25-31.
- [6] S Ijmker, M. A. (2006). Should office workers spend fewer hours at their computer? A systematic review of the literature. . *Occup. Environ. Med.* 64 , pp. 211-222.
- [7] Jason A. Bower, G. J. (2006). MRI evaluation of carpal tunnel dimensions in healthy wrists: Implications for carpal tunnel syndrome. *Clinical Biomechanics* 21 , pp. 816–825.
- [8] Gerr F, M. M. (2002). A prospective study of computer users: I. Study design and incidence of musculoskeletal symptoms and disorders. *American Journal of Industrial Medicine* 41 , pp. 221-235.
- [9] Ali AYDENIZ, S. G. (2008). Upper Extremity Musculoskeletal Disorders among Computer Users. . *Turk J Med Sci* 2008 , pp. 235-238.
- [10] Peter J. Keir, R. P. (1999). Changes in geometry of the Finger Flexor tendons in the carpal tunnel with wrist posture and tendon load: an MRI study on normal wrists . *Clinical Biomechanics* 14 , pp. 635-645.
- [11] David Rempel, P. J. (1997). Effects of Static Fingertip Loading on Carpal Tunnel Pressure. *Journal of Orthopaedic Research* 15 , pp. 422-426.
- [12] David M. Rempel, P. J. (Setembro de 2008). Effect of Wrist Posture on Carpal Tunnel Pressure while Typing. *J Orthop Res.* , pp. 1269–1273.

- [13] Rempel D, K. P. WRIST POSTURE INFLUENCES CARPAL TUNNEL PRESSURE DURING KEYBOARD USE. *PREMUS 2007*.
- [14] Rempel et al. (1994). VDT – related musculoskeletal symptoms: Interactions between work posture and psychosocial work factors. *American Journal of Industrial Medicine*.26(5).
- [15] David J. Nuckley, D. R.-c. *DEVELOPMENT OF A FIBER-OPTIC FORCE SENSING GLOVE TO PROVIDE CLINICAL BIOMECHANICS MEASUREMENTS*.
- [16] Y. Hasegawa, M. S. (2007). GLOVE TYPE OF WEARABLE TACTILE SENSOR PRODUCED BY ARTIFICIAL HOLLOW FIBER. *14th International Conference on Solid-State Sensors* . Lyon, France.
- [17] Paul HChappell, J. A. (2003). Contact force sensor for artificial hands with a digital interface for a controller. *Meas. Sci. Technol.* 14 , pp. 1275–1279.
- [18] Kostas N. Tarchanidis, J. N. (Junho de 2003). Data Glove With a Force Sensor. *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT* , VOL. 52 .
- [19] CARLOS GERTZ, M. A. *ESTUDO DA FORÇA APLICADA NA TECLA DURANTE DIGITAÇÃO USUAL E COM ACESSÓRIOS ERGONÔMICOS*. . Laboratório de Medições Mecânicas da Universidade Federal do Rio Grande do Sul – UFRGS.
- [20] Lee D. L., K. P. (2007). EFFECT OF KEYBOARD KEYSWITCH DESIGN ON FINGER MUSCLE ACTIVITY DURING TAPPING ON ISOLATED KEYSWITCHES. *PREMUS 2007*.

Capítulo 2, Capítulo 3 e Capítulo 4:

- [1] I. A. ANDERSON, K. L. FORCE MEASUREMENT. In Wiley, *Encyclopedia of Biomedical Engineering*, Vol. 6. (pp. 1664 – 1670).
- [2] PAUL C. DECHOW, Q. W. STRAIN GAGES. In Wiley, *Encyclopedia of Medical Devices and Instrumentation*, Vol. 6. (pp. 282 – 290).
- [3] R. S. JERRETT, B. (1945). The Acoustic Strain Gauge. *J. Sci. Instrum.* , 22-29.
- [4] <http://www.techbriefs.com/component/content/article/2027>. (s.d.). Obtido em 25 de Julho de 2009, de www.techbriefs.com.
- [5] Jingtiao Li a, J. P. (2007). Interdigital capacitive strain gauges fabricated by direct-write thermal spray and ultrafast laser micromachining. *Sensors and Actuators A* 133 , pp. 1–8.
- [6] OMEGA. (s.d.). <http://www.omega.com/Literature/Transactions/volume3/strain.html>; Obtido em 24 de Julho de 2009, de www.omega.com.
- [7] Honeywell. (s.d.). FSS Low Profile Force Sensors. *Data Sheet* .

- [8] HOKURIKU. (s.d.). HFD-500. *Data Sheet* .
- [9] Honeywell. (s.d.). FSG15N1A. *Data Sheet* .
- [10] Honeywell. (s.d.). FS01/FS03 Force Sensors. *Data Sheet* .
- [11] Brown, B. . (s.d.). INA126, MicroPOWER INSTRUMENTATION AMPLIFIER Single and Dual Versions. *Data Sheet* .
- [12] JFET-INPUT OPERATIONAL AMPLIFIERS, SLOS081A–D2297, FEBRUARY 1977–REVISED NOVEMBER 1992. *Data Sheet*.
- [13] Institute of Measurement and Control. (Originally Published 1996. Reviewed and Re-issued 2003). *A Code of Practice for the Calibration of Industrial Process Weighing Systems*.
- [14] Erdem, U. (1982). Force and weight measurement. *J. Phys. E: Sci. Instrum., Vol. 15* .
- [15] British Standards Institution: Precision of Test Methods 1: Guide for the Determination and Reproducibility for a Standard Test Method (BS 597, Part 1). London, BSI, 1975.
- [16] National Instruments Corporation. All rights reserved. (2006). *Acquisition, Introduction to Data*. This tutorial is part of the National Instruments Measurement Fundamentals series.
- [17] National Instruments Corporation. (2006). *Data Acquisition Fundamentals. NI*. National Instruments Corporation. All rights reserved.
- [18] Olshausen, B. A. (October 10, 2000). *PSC 129 - Sensory Processes* .
- [19] McWhirter, N. (1985). *THE GUINNESS BOOK OF WORLD RECORDS, 23rd US edition*. New York: Sterling Publishing Co., Inc.
- [20] L. M. V. Serrano, C. J. (s.d.). *Sistemas de Aquisição, Processamento e Armazenamento de Dados* .
- [21] DataTaker®. *Data acquisition and data logging. Some basics* . Powerpoint.
- [22] National Instruments. *Portable E Series*. Data Sheet.
- [23] National Instruments. *USB-2008 NI*. Data Sheet.
- [24] National Instruments. (s.d.). <http://sine.ni.com/nips/cds/view/p/lang/en/nid/205686>;. Obtido em 23 de Julho de 2009, de <http://sine.ni.com>.
- [25] DataTaker®. (s.d.). Obtido em 23 de Julho de 2009, de www.datataker.com.
- [26] DataQ®. (s.d.). <http://www.dataq.com>. Obtido em 24 de Julho de 2009
- [27] <http://www.psoc-chile.es.tl>. (s.d.). Obtido em 24 de Julho de 2009

- [28] Ashby, R. (Copyright © 2005, Elsevier Inc. All rights reserved.). *Designer's Guide to the Cypress PSoC™* .
- [29] Cypress®. *CYPRESS CyFi™ LOW-POWER RF SOLUTIONS brochure*.
- [30] Cypress®. *PSoC® Mixed-Signal Array Technical Reference Manual (TRM)* . Document No. 001-14463 Rev. *A.
- [31] Jeffrey Stewart, Cypress®. (March 17, 2004). *In-System Serial Programming (ISSP™) Protocol*.
- [32] Cypress®. *Programmer Guide 3.06* . Spec. # 001-51796 Rev. *A.
- [33] Cypress®. *IDE Guide Document*. # 001-42655 Rev **.
- [34] Cypress®. *PSoC FirstTouch Starter Kit with CyFi Low-Power RF* . Spec. # 001-48286 Rev. **.
- [35] Cypress®. *Sense and Dashboard Control Guide*. Spec. # 001-47654 Rev. **.
- [36] © Cypress MicroSystems. (2002 – 2004). *CY8C27443 Data Sheet*. Document No. 38-12012 Rev. *I.
- [37] © Cypress Semiconductor Corporation. (2005-2008). *ImageCraft C Compiler Guide Document* . # 001-44476 Rev **.
- [38] Cypress Semiconductor Corporation. (2001-2008). *4 to 1 Analog Multiplexer Data Sheet Copyright* .
- [39] Ess, D. V. (April 14, 2006). *AN2367 Analog – Differential Amplifier* . Document No. 001-25559 Rev. **.
- [40] © Cypress Semiconductor Corporation. . (*B Revised October 7, 2008). *Programmable Gain Amplifier Data Sheet* . Document Number: 001-13575 Rev. .
- [41] Seguire, D. *AN2239 ADC Selection* . 2005 Revision A, Cypress Semiconductor Corporation.
- [42] Cypress Semiconductor Corporation. (*B Revised October 7, 2008). *Incremental ADC Data Sheet* . Document Number: 001-13251 Rev. .
- [43] Cypress Semiconductor Corporation. (Revised October 8, 2008). *CyFi™ Star Network Protocol Stack Data Sheet* . Document Number: 001-44749 Rev. **.
- [44] Cypress Semiconductor Corporation. (Revised October 7, 2008). *8-Bit Timer Data Sheet* . Document Number: 001-13625 Rev. *C.
- [45] Young, K. *Measurement and Compensation of the PSoC Internal Low Speed Oscillator AN14278* . Document No. 001-14278 Rev. ** .
- [46] Horton, I. (2006). *Beginning C: From Novice to Professional*. Fourth Edition.

- [47] Kalechman, M. (2009). *Practical MATLAB basics for engineers* . ISBN 978-1-4200-4774-5.
- [48] Sadanandan, A. (Copyright (c) 2008). csvimport function.
- [49] Dalley, I.B. (2007).rsplit function.