

PULSEIRA INTELIGENTE PARA APLICAÇÕES INDUSTRIAIS E DE LAZER

Carlos Miguel Alves da Silva



Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização de Sistemas Autónomos

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2013

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Disciplina de
Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de
Computadores

Candidato: Carlos Miguel Alves da Silva, N° 1070206, 1070206@isep.ipp.pt

Orientação científica: Eng. Eduardo Silva, eaps@lsa.isep.ipp.pt



Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização de Sistemas Autónomos

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

23 de novembro de 2013

Agradecimentos

Gostaria de começar por agradecer aos meus pais, irmão e namorada por todo o apoio, confiança e, por vezes, sacrifícios que fizeram para me proporcionar melhores condições. Sem eles não seria possível a conclusão desta etapa.

Ao meu orientador, Eng. Eduardo Silva pela boa disposição mantida durante todo o trabalho, o apoio e pela confiança demonstrada desde o início.

Por fim, mas não menos importante, a todos aqueles que direta ou indiretamente estiveram envolvidos nesta fase da minha vida.

Um sincero obrigado a todos!

Resumo

The idea of wearing technology is hardly new. There's armor and swords and many other things that we've worn on our bodies that were the technologies of the day. That can help us think about the current obsession and where these things are going.

Genevieve Bell, diretora de um centro de investigação da Intel

Nos últimos anos, o avanço da tecnologia e a miniaturização de diversos componentes têm permitido o aparecimento de novos conceitos, ideias e projetos, que até aqui não passariam de filmes de ficção científica. Com a tecnologia atual, podem ser desenvolvidos pequenos dispositivos *wearable* com diversas interfaces, múltiplas conectividades, poder de processamento e autonomia. Permitindo desta forma, dar resposta à crescente necessidade de interação com os mais diversos equipamentos eletrónicos do dia-a-dia, melhorando o acesso e o fornecimento de informação.

O principal objetivo deste trabalho passa assim por demonstrar e implementar um conceito que permita estreitar e facilitar a interação entre o utilizador e o mundo que o rodeia, quer em ambientes domésticos quer industriais. Para isso foi projetado e implementado um dispositivo *wearable* (para utilização no pulso) baseado numa arquitetura de *hardware* e *software* capaz de correr diferentes aplicações, tais como extensão de alertas de um *smartphone*, *crowdsourcing* de informações meteorológicas, manutenção e inspeção industrial e monitorização remota de forças de segurança.

Os resultados obtidos demonstram que este conceito é viável tanto do ponto de vista técnico como funcional, evidenciando boas hipóteses para que estes conceitos, métodos e tecnologias possam ser integradas em plataformas robóticas desenvolvidas no âmbito de projetos do Laboratório de Sistemas Autónomos (LSA) bem como nos contextos industrial e de lazer.

Palavras-Chave

Wearable, Alertas de *smartphone*, *Crowdsourcing*, Manutenção e inspeção industrial, Gestão de forças de segurança, pulseira inteligente.

Abstract

The idea of wearing technology is hardly new. There's armor and swords and many other things that we've worn on our bodies that were the technologies of the day. That can help us think about the current obsession and where these things are going.

Genevieve Bell, Intel Investigation Centre Director

In the last years, the technology advance and the miniaturization of several components has created new concepts, ideas and projects that till now just seen in the science fiction movies. Based on the actual technology, small wearable devices would be developed with multiple interfaces and connectivity's, computational power and autonomy. On this way, it enables an effective answer to the growing need of interaction with the diversity of electronic equipment's of the daily activities, improving the information access.

The main goal of this work is to demonstrate and implement a concept that will get closer and easier the interaction between the user and the world surrounding, mainly on the domestic or industrial environments. For that, it was designed and implemented a wearable device (wrist use) based on a hardware and software architecture with the multi-applications capability as the notifications bridge from a smartphone, meteorological crowdsourcing platform, industrial maintenance and inspection and army soldiers remote monitoring.

The results demonstrate that this concept is feasible both from a technical and functional viewpoint, showing a good chance for these concepts, methods and technologies can be integrated into robotic platforms developed within projects of the Laboratory of Autonomous Systems (LSA) and in industrial and entertainment contexts.

Keywords

Wearable, smartphone notifications, crowdsourcing, industrial maintenance and inspection, security forces management, smart strap.

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE	VII
ÍNDICE DE FIGURAS	IX
ACRÓNIMOS	XIII
1. INTRODUÇÃO.....	1
1.1. DISPOSITIVOS <i>WEARABLE</i>	1
1.2. BREVE HISTORIA DOS DISPOSITIVOS <i>WEARABLE</i>	3
1.3. MOTIVAÇÃO - ENQUADRAMENTO	4
1.4. OBJETIVOS	6
1.5. ORGANIZAÇÃO DO RELATÓRIO	7
2. CONCEITOS FUNDAMENTAIS	9
2.1. O QUE É UM DISPOSITIVO <i>WEARABLE</i> ?	10
2.2. CROWDSOURCING	12
2.3. CONCEITOS FUNDAMENTOS DE APLICAÇÕES DE MANUTENÇÃO E GESTÃO INDUSTRIAL	14
3. ESTADO DA ARTE	15
3.1. CONCEITOS	15
3.2. APLICAÇÕES	18
3.3. EQUIPAMENTOS / PROTÓTIPOS	20
4. TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO	27
4.1. SISTEMAS EM TEMPO REAL	27
4.2. RTX - SISTEMA OPERATIVO	29
4.3. EM::BLOCKS (C/C++ IDE)	30
4.4. ALTIUM DESIGNER	31
5. ESPECIFICAÇÃO E DESCRIÇÃO DO SISTEMA	33
5.1. REQUISITOS DO SISTEMA	35
5.2. CENÁRIOS DE APLICAÇÃO	39
6. ARQUITETURA DE HARDWARE	47

6.1.	MÉTODOS DE ENTRADA	49
6.2.	MÉTODOS DE SAÍDA	50
6.3.	LOCALIZAÇÃO/POSICIONAMENTO	51
6.4.	COMUNICAÇÕES	52
6.5.	FONTES DE ENERGIA.....	53
6.6.	DESCRIÇÃO DOS COMPONENTES	54
7.	ARQUITETURA DE SOFTWARE.....	61
7.1.	DEVICE DRIVERS	62
7.2.	SISTEMA OPERATIVO	74
7.3.	MIDDLEWARE.....	81
7.4.	APLICAÇÕES	88
8.	ARQUITETURA MECÂNICA	95
8.1.	DISPOSIÇÃO DA ELETRÓNICA	97
8.2.	DESIGN DO PROTÓTIPO	98
9.	CONCLUSÕES	99
	REFERÊNCIAS DOCUMENTAIS.....	103
	ANEXO A. DIMENSÕES EXTERIORES DO ECRÃ	105

Índice de Figuras

Figura 1	Evolução das invenções criadas por Steve Mann's [6]	3
Figura 2	Exemplo de um cenário de mediação e facilitação de informação	4
Figura 3	Exemplo da troca de informações entre um robô e um bombeiro	5
Figura 4	Ilustração do conceito <i>crowdsourcing</i>	12
Figura 5	Logotipo do OpenStreetMap	13
Figura 6	Processos de manutenção da indústria	14
Figura 7	NIKE FUEL BAND	20
Figura 8	MYO	21
Figura 9	Samsung Galaxy Gear	21
Figura 10	Controlo Remoto no Pulso.....	22
Figura 11	Aparência do Google Glass	23
Figura 12	Interfaces do Google Glass	23
Figura 13	Protótipo de sistema OmniTouch [14]	24
Figura 14	WR1100 - Computador pessoal para utilização no pulso [15]	25
Figura 15	Arquitetura do RTOS RTX.....	29
Figura 16	Aspeto do ambiente de desenvolvimento Em::Blocks	30
Figura 17	Esquema elétrico	31
Figura 18	Modelo 3D do microprocessador.....	32
Figura 19	Interação do dispositivo com o mundo	34
Figura 20	Cenário - Alertas Smartphone.....	39
Figura 21	Cenário – Cenário - Crowdsourcing de Informações Meteorológicas.....	40
Figura 22	Cenário – Manutenção e Inspeção na Industria	42
Figura 23	Cenário – Monitorização Remota de Bombeiros e Militares	43
Figura 24	a) Efeito com isqueiros. b) Efeito aleatório com pulseiras multicolor	45
Figura 25	Padrão criado com recurso a cartão multicolor	45
Figura 26	Funcionamento básico do sistema	48
Figura 27	Arquitetura geral do <i>hardware</i> elétrico.....	48
Figura 28	Diferentes métodos de entrada encontrados no quotidiano	49
Figura 29	Diferentes métodos de saída encontrados no quotidiano	50
Figura 30	Diferentes métodos de localização/posicionamento.....	51

Figura 31	Zonas de comunicação em aplicações <i>wearable</i>	52
Figura 32	Diferentes fontes de energia	53
Figura 33	Broadcom BCM43341	54
Figura 34	Acelerómetro LIS3DSH	55
Figura 35	Diagrama de blocos do microprocessador	56
Figura 36	Aspeto do ecrã	57
Figura 37	Bateria de polímeros de lítio	57
Figura 38	Sensor de temperatura	58
Figura 39	Sensor de humidade Si7005	58
Figura 40	Sensor do índice UV ML8511	59
Figura 41	Relação da tensão com a radiação UV do sensor ML851	59
Figura 42	Arquitetura geral de <i>software</i>	62
Figura 43	Estrutura de comando para apagar o ecrã	64
Figura 44	Estrutura de comando para escrever uma linha	64
Figura 45	Comportamento do sensor capacitivo [23]	66
Figura 46	Fluxograma da rotina de interrupção do TOUCH	68
Figura 47	Exemplo da máquina de estados para a função acordar [24]	69
Figura 48	Fluxograma da comunicação com o MEMS	70
Figura 49	Fluxograma do driver do sensor TMP006	71
Figura 50	Fluxograma do driver do sensor de UV	72
Figura 51	Fluxograma do driver do sensor Si7005	73
Figura 52	Estrutura do CMSIS-RTOS RTX	74
Figura 53	Estados das <i>threads</i>	75
Figura 54	Sequência de operação de um <i>Mutex</i>	76
Figura 55	Operações dos semáforos	77
Figura 56	Estrutura de uma <i>Message Queue</i>	78
Figura 57	Estrutura de um <i>Mail Queue</i>	79
Figura 58	Sequência temporal de uma função <i>timer</i>	80
Figura 59	Gestor de Notificações (<i>Middleware</i>)	81
Figura 60	Gestor de Recursos (<i>Middleware</i>)	83
Figura 61	Exemplo de funcionamento da função <i>Merge Requests</i>	84
Figura 62	Gestor do Interface Gráfico (<i>Middleware</i>)	85
Figura 63	Sequência de interação com o gestor do interface gráfico	86
Figura 64	Gestor de Comunicações (<i>Middleware</i>)	87
Figura 65	Diagrama geral de funcionamento da APP1 (Alertas <i>Smartphone</i>)	88

Figura 66	Exemplo da receção de uma nova SMS	89
Figura 67	Exemplo real da aplicação de alertas de <i>smartphone</i>	89
Figura 68	Diagrama geral de funcionamento da APP2 (<i>CrowdSourcing</i>).....	90
Figura 69	Exemplo da navegação na aplicação.....	91
Figura 70	Exemplo real da aplicação de <i>Crowdsourcing</i>	91
Figura 71	Diagrama geral de funcionamento da APP3 (Manutenção E Inspeção Na Industria) ..	92
Figura 72	Exemplo do interface da interface industrial	92
Figura 73	Diagrama geral de funcionamento da APP4 (Monitorização Remota)	93
Figura 74	Exemplo do interface da interface de monitorização remota	93
Figura 75	Exemplo real da aplicação de monitorização remota	94
Figura 76	Diagrama geral de funcionamento da APP5 (Sincronização e Efeitos Visuais).....	94
Figura 77	Desenhos mecânicos manuais.....	96
Figura 78	Desenho do objeto a três dimensões	96
Figura 79	Desenho das indentações para os componentes	96
Figura 80	<i>Design</i> do protótipo gadget, parte frontal (a) e parte posterior (b)	98
Figura 81	Simulação do aspeto do dispositivo quando em utilização	98
Figura 82	Protótipo de hardware.....	100

Acrónimos

2D	–	Duas Dimensões
3D	–	Três Dimensões
3G	–	Redes Moveis de Terceira Geração
4G	–	Redes Moveis de Quarta Geração
APP	–	Application
API	–	Application Programming Interface
ARM	–	Advanced RISC Machine
BGA	–	Ball Grid Array
CGA	–	Color Graphics Adapter
CNN	–	Cable News Network
DVD	–	Digital Versatile Disc
FM	–	Frequency Modulation
GCC	–	GNU Compiler Collection
GLONASS	–	Global Navigation Satellite System
GPRS	–	General Packet Radio Service
GPS	–	Global Positioning System
GSM	–	Global System for Mobile Communications
HTTP	–	Hypertext Transfer Protocol

IEEE	–	Instituto de Engenheiros Eletricistas e Eletrônicos
ISR	–	Interrupt Service Routine
I2C	–	Inter Integrated Circuit
LED	–	Light Emitting Diode
LESENSE	–	Low Energy Sensor Interface
MPEG	–	Moving Picture Experts Group
MP3	–	MPEG-1/2 Audio Layer 3
NFC	–	Near Field Communication
PDA	–	Personal Digital Assistant
QFN	–	Quad Flat No leads
RTC	–	Real Time Clock
RTOS	–	Real-time operating system
RTX	–	Real Time eXecutive
RAM	–	Random Access Memory
RF	–	Radio Frequency
SPI	–	Serial Peripheral Interface
TETRA	–	Terrestrial Trunked Radio
UMTS	–	Universal Mobile Telecommunication System
UV	–	Radiação Ultravioleta
WLAN	–	Wireless Local Area Network
WLBGA	–	Wafer Level Ball Grid Array

1. INTRODUÇÃO

1.1. DISPOSITIVOS *WEARABLE*

Na atualidade o conceito *wearable* tem sido a base de diversos trabalhos na área da investigação [24]. Seguindo o rumo do que se espera para o futuro, o trabalho desenvolvido visa a criação de uma plataforma base para dispositivos *wearable*. Visto que a área de aplicação tem-se alargado, o objetivo é dar mais ênfase nas aplicações de dispositivos industriais/gadget. Deste modo foram desenvolvidas aplicações específicas de *hardware* e *software*.

"If we think of technology as a runaway monster, we can think of this as a way to tame the beast with a piece of itself."

Steve Mann, inventor of wearable computers, in Cyberman [1].

A citação acima apresentada implica duas coisas: nós não podemos parar a tecnologia, definida como “o monstro fugitivo” [*runaway monster*], mas podemos utilizar dispositivos *wearable* para tornar cada vez mais pequena a lacuna entre os seres humanos e a tecnologia.

Os dispositivos com capacidades de computação estão cada vez mais difundidos, estando atualmente incorporados nos telemóveis, leitores de música, câmaras, na roupa, nos edifícios, nos carros, e em todos os tipos de objetos do nosso quotidiano. Estes não se assemelham à imagem de um PC *desktop* com um monitor, teclado e rato. Como devemos interagir e conviver diariamente com muitos equipamentos, de diferentes tamanhos e com diferentes funções, estando mesmo muitas das vezes inacessíveis ou escondidos para que nem sequer nos apercebamos que existem.

De que forma estes equipamentos podem tornar a nossa vida melhor? A visão da computação ubíqua é que eventualmente, os computadores vão desaparecer e tornar-se parte do nosso meio ambiente, desaparecendo das nossas vidas quotidianas como os conhecemos. Idealmente, existirão mais dispositivos com capacidades de computação e cada vez mais inteligentes que de uma forma invisível tendem a melhorar o nosso dia-a-dia, mas iremos ser menos conscientes da existência deles, concentrando-nos mais nas nossas tarefas em vez da tecnologia que está por detrás delas.

Utilizando tecnologia *wearable* e comunicações sem fios, como WLAN, Bluetooth, RF ou 3G/4G, podemos ter acesso instantâneo a informações, a qualquer momento e em diferentes locais, quer através da rede móvel pessoal, da rede doméstica, da rede da empresa ou através de infraestruturas públicas.

1.2. BREVE HISTORIA DOS DISPOSITIVOS *WEARABLE*

Algumas inovações têm ajudado a definir um caminho para os *wearable computers*. Podemos recuar até à data de 1268 quando se ouviu falar dos óculos *wearable* pela primeira vez, tendo aparecido posteriormente o relógio de bolso em 1762 [24]. O primeiro *wearable computer* foi inventado pelo Ed Thor e Claude Shannon em 1966 [2] com a intenção ser usado por um assistente para definir a velocidade de uma roleta. Nesse mesmo ano foi criado por Sutherland, o primeiro computador baseado num ecrã, para ser colocado na cabeça.

Este tipo de tecnologia tem contribuído para a possibilidade de produção de *wearable computers* úteis e discretos fazendo com que as empresas mostrem algum interesse neste tipo de tecnologia.



Figura 1 Evolução das invenções criadas por Steve Mann's [5]

Entre os anos 1991 e 1993, três *wearable computers* foram construídos como demonstradores de tecnologia pela Universidade de Carnegie Mellon. Estes dispositivos, denominados de VuMan1, VuMan2 e Naigator1, foram construídos como projetos de aulas e não tinham nenhum utilizador definido. Todos eles usaram o *Private Eye* como o dispositivo de visualização [2]. O *Private Eye* é um dispositivo com resolução *Color Graphics Adapter* (CGA) e que é apropriado para desenhos de baixa resolução como pode ser visto à direita da Figura 1.

1.3. MOTIVAÇÃO - ENQUADRAMENTO

No contexto da robótica, uma das grandes áreas de estudo e desenvolvimento dos últimos anos, é a interação dos robôs com os humanos e vice-versa, com o intuito de tornar os robôs cada vez mais parte do quotidiano, simplificando para isso a forma de interação.

Como engenheiros de desenvolvimento nas áreas das tecnologias de computação e de sistemas embebidos, surge naturalmente o desafio de encontrar novas formas de interação com uma nova geração de dispositivos. Uma das formas de melhorar a facilitar a interação é reduzir a quantidade de interação explícita que é necessária para comunicar com eles, e em vez de isso aumentar a quantidade de interação implícita. Um exemplo muito simples de interação implícita seria o seguinte: em vez de carregar num interruptor para acender a luz, poderíamos usar um sensor de movimento que deteta quando alguém entra numa sala, acendendo assim a luz. Assim, reduzimos a quantidade de ações explícitas que o utilizador tem de realizar, utilizando para isso um sensor para controlar a fonte de luz com uma simples regra. Outro exemplo seria um dos cenários explorados no âmbito da tese relacionado com a interação entre o utilizador e o *smartphone*, no qual o dispositivo *wearable* tem um papel de mediador e facilitador de informação como exemplificado na Figura 2.

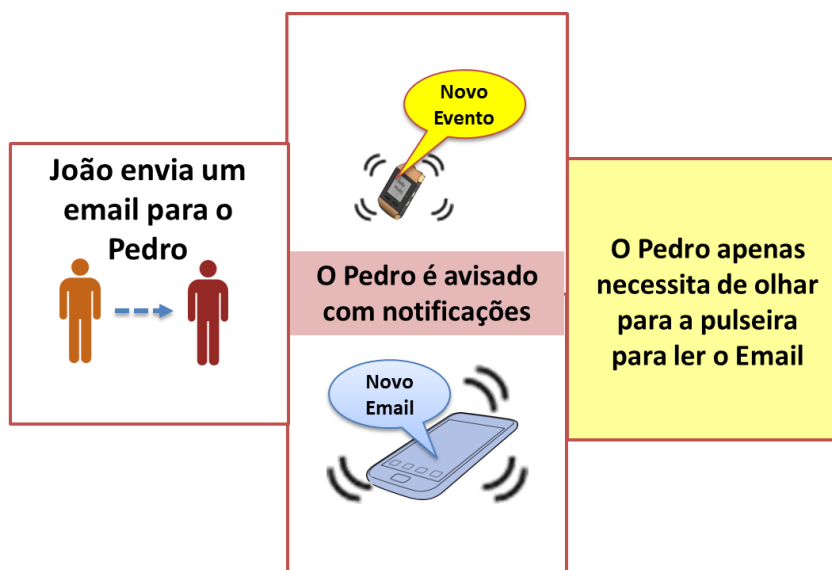


Figura 2 Exemplo de um cenário de mediação e facilitação de informação

No âmbito da robótica um exemplo seria o caso de utilização de robôs para auxílio na busca e salvamento, onde o robô e o bombeiro trocariam informações críticas de uma forma implícita sem existir a necessidade do humano realizar ações complexas para obter informações. Uma situação seria o envio de informações do robô relativas ao terreno para o bombeiro alertando-o de eventuais perigos como ilustrado no exemplo da Figura 3.

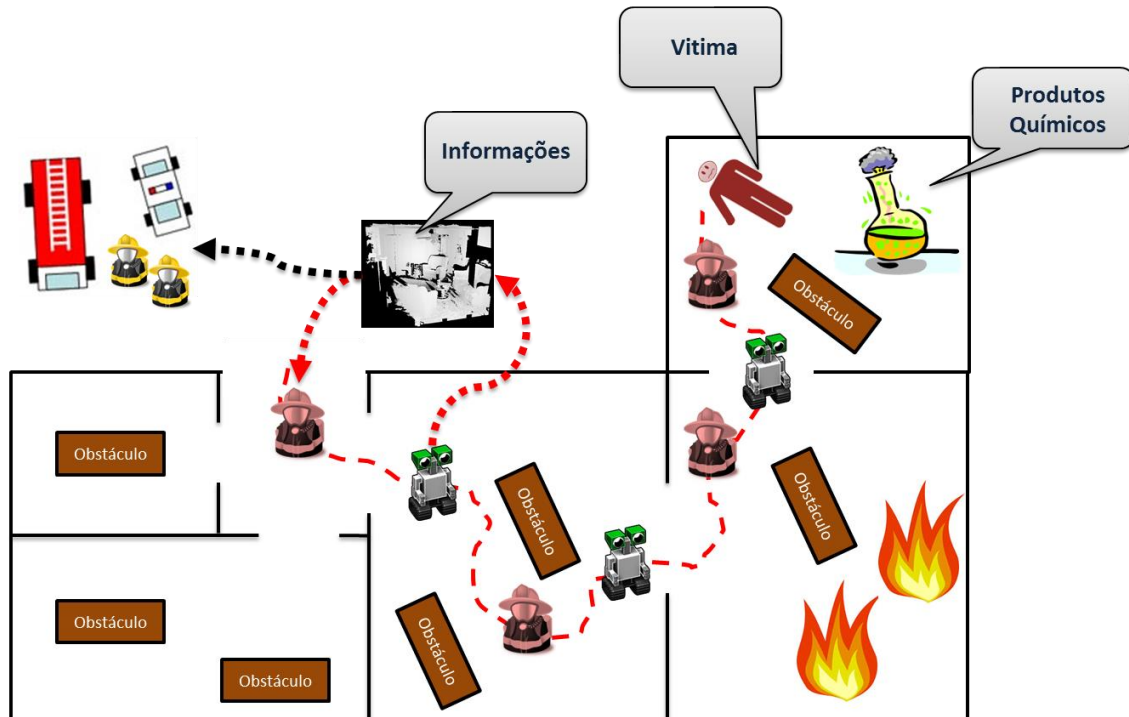


Figura 3 Exemplo da troca de informações entre um robô e um bombeiro

Com isto, perspectivam-se boas hipóteses de estes conceitos, métodos e tecnologia ser integrada em plataformas robóticas desenvolvidas no âmbito de projetos do Laboratório de Sistemas Autônomos (LSA) bem como no âmbito industrial e de lazer.

1.4. OBJETIVOS

Com este trabalho pretende-se estudar, especificar, projetar e implementar um dispositivo/plataforma com o intuito de ser uma base para futuros dispositivos *wearable*, com mais ênfase nas áreas de dispositivos industriais/gadget.

O projeto centraliza-se na criação genérica da plataforma de *hardware* e *software* bem como todas as especificidades associadas aos cenários explorados. Desta forma um dos principais objetivos de um ponto de vista geral é a demonstração do conceito quer a nível funcional quer a nível técnico.

Sendo assim, os objetivos a atingir são:

- Identificação e estudo dos conceitos fundamentais aplicáveis
- Estudo e identificação dos requisitos
- Estudo e especificação da plataforma de *hardware*:
 - Plataforma base e *hardware* específico para aplicações gadget e indústrias
- Desenho e implementação da *framework* de *software*
 - Arquitetura
 - *Drivers*
 - *Middleware*
- Desenho e implementação dos cenários de aplicação:
 - Alertas *Smartphone*
 - *Crowdsourcing* de Informações Meteorológicas
 - Manutenção e Inspeção na Indústria
 - Monitorização Remota de Bombeiros e Militares
 - Sincronização e Efeitos Visuais
- Estudo e desenho da arquitetura mecânica tendo em vista os cenários de aplicação

1.5. ORGANIZAÇÃO DO RELATÓRIO

No primeiro capítulo, maioritariamente de cariz introdutório, é feita uma apresentação e contextualização do tema abordado ao longo deste trabalho. São ainda apresentados os objetivos principais.

No capítulo dois, são introduzidos alguns conceitos fundamentais relevantes para o desenvolvimento do trabalho, abordando essencialmente a área dos gadgets e da indústria.

O capítulo três apresenta o estado da tecnologia para dispositivos *wearable*. Este capítulo introduz alguns conceitos básicos diretamente ligados ao *wearable*, algumas aplicações utilizadas comercialmente e ainda alguns exemplos práticos de dispositivos/protótipos existentes.

No quarto capítulo, são introduzidas algumas tecnologias e ferramentas utilizadas ao longo do trabalho.

No capítulo cinco, é realizada a especificação e descrição do sistema. Os requisitos e os cenários de aplicação explorados são também aqui apresentados.

No sexto capítulo é detalhada a solução de *hardware*. Apresentam-se desta forma as principais funcionalidades da plataforma, explicando o funcionamento de todo o sistema do ponto de vista elétrico. São ainda enumerados e descritos os principais componentes que constituem a plataforma.

No capítulo sete, é apresentada e detalhada a solução de *software*. A solução apresentada subdivide-se em quatro tópicos principais, contemplando o desenho e implementação, dos drivers, do sistema operativo, dos diferentes agentes do *middleware* e ainda as aplicações.

No oitavo capítulo, é realizado o estudo e desenho mecânico do sistema. Desde a disposição da eletrónica ao *design* do protótipo.

No último (9) capítulo, são apresentadas as conclusões sobre o trabalho realizado bem como eventuais trabalhos futuros.

2. CONCEITOS FUNDAMENTAIS

Neste capítulo pretende-se apresentar alguns conceitos fundamentais relativos ao trabalho, importantes para a compreensão das opções tomadas no desenrolar do projeto.

São vários os conceitos abordados ao longo deste trabalho, uma vez que se trata de toda a estrutura da solução, desta forma, são apresentados alguns dos conceitos mais relevantes, relacionados com a visão do projeto e a própria implementação do trabalho desenvolvido.

Gadget (em português: geringonça, dispositivo) é um equipamento que tem um propósito e uma função específica, prática e útil no quotidiano. São normalmente chamados de gadgets dispositivos eletrónicos portáteis como PDA, telemóveis, leitores de MP3, entre outros. Os conceitos associados aos gadgets são intermináveis pelo que vão ser apenas apresentados os mais importantes relacionados com os objetivos propostos neste documento.

2.1. O QUE É UM DISPOSITIVO *WEARABLE*?

Uma forte característica de um dispositivo *wearable* é que este pode permanecer anexado ao corpo do utilizador durante um longo período de tempo sem que se torne incómodo ou condicionador de movimentos. Este é conceito que separa estes equipamentos daquilo que é conhecido como um simples computador, ou equipamento informático portátil.

Por outras palavras um dispositivo *wearable* é por norma um dispositivo pequeno, portátil e interativo que está sempre pronto para utilização. Geralmente este é integrado em objetos do quotidiano como as roupas, pulseiras, relógios, óculos entre outros.

Os dispositivos *wearable* podem ser definidos através do seu modo de operação e os principais atributos [4].

2.1.1. MODOS DE OPERAÇÃO

Os modos de operação definem de que forma o utilizador e as máquinas devem interagir.

Interação

Por norma a interação entre o utilizador-computador é associado ao conceito de sessão, tendo em conta que toda a interação acontece entre o momento em que é iniciada a sessão até ao momento em que é terminada essa mesma sessão.

Por sua vez os dispositivos *wearable* estão constantemente disponíveis, fazendo com que o fluxo de informações entre o dispositivo e o utilizador, ou vice-versa, seja contínuo. A presença contínua do dispositivo no corpo permite uma adaptação a longo prazo, o que leva a que seja criada uma nova forma de sinergia entre o utilizador e o dispositivo.

Aperfeiçoamento

Os computadores tradicionais são normalmente utilizados para a resolução de problemas específicos e a para a comunicação em rede. Em ambos os casos, a utilidade dos computadores está essencialmente ligada à sua eficiência e eficácia na realização dessas atividades. Por outras palavras pode dizer-se que os computadores tradicionais são projetados com o intuito de realizar as tarefas (computação/comunicação) com o maior desempenho possível.

Por outro lado assume-se que a principal tarefa a ser realizada pelos dispositivos *wearable* não está relacionada com o processamento de dados mas sim com os objetivos / necessidades do utilizador. Desta forma, os dispositivos *wearable* devem funcionar de forma a potenciar as capacidades do utilizador, através do aumento dos seus sentidos, memória, comodidade entre outros.

Mediador

Os dispositivos *wearable* têm como característica a capacidade de encapsularem o utilizador, e dessa forma funcionar como um mediador entre o utilizador e o ambiente que o rodeia. Daqui nasce uma consequência básica, que é a possibilidade do dispositivo agir como filtro de informações. Enquanto filtro de informações, os dispositivos *wearable* permitem que o utilizador defina o tipo e a quantidade de informações que pretende receber. Isso permite por exemplo, que o dispositivo filtre de forma personalizada o tipo de alertas de um telemóvel que serão apresentados ao utilizador de acordo com os seus interesses. Além disso, o dispositivo pode, por exemplo, servir como um agente intermediário, ajudando o utilizador em tarefas do dia-a-dia.

2.1.2. ATRIBUTOS DE UM DISPOSITIVO WEARABLE

Os atributos definem as principais características de um dispositivo *wearable*.

- **Não restritivo:** A sua utilização não impede o utilizador de realizar outras tarefas
- **Facilidade de interação:** O interface com o utilizador tem de ser simples e intuitivo.
- **Controlável:** Tem de ser possível o utilizador controlar a forma e a quantidade das informações disponibilizadas.
- **Monitorização do ambiente:** O ambiente em redor do utilizador é constantemente monitorizado, sendo extraídos apenas os dados relevantes definidos pelo utilizador.
- **Capacidade de comunicação:** O dispositivo expande a capacidade de comunicação do utilizador com o ambiente que o rodeia.

2.2. CROWDSOURCING

O *crowdsourcing* é um modelo de produção que utiliza a inteligência e os conhecimentos coletivos e voluntários, geralmente espalhados pela Internet para resolver problemas, criar conteúdo e soluções ou desenvolver novas tecnologias, assim como também para gerar fluxo de informação [6].



Figura 4 Ilustração do conceito *crowdsourcing*

O *crowdsourcing* tem como objetivo prático a redução do custo e o aumento da qualidade de informação para a população em geral. O *crowdsourcing* tem um grande potencial para revolucionar a recolha de informações e os sistemas de processamento, aumentando o custo-benefício da recolha de dados em larga escala e melhorando as técnicas para a extração de informações a partir dos dados. Além disso o *crowdsourcing* fornece um poderoso mecanismo para a criação de dados relacionados com o mundo, principalmente através do uso de dispositivos ricos em sensores (como por exemplo os telemóveis com GPS, acelerómetro, giroscópio, etc). Estes sensores que transportamos no dia-a-dia anexados a diversos dispositivos, podem ser utilizados para proporcionar informação contínua e sem precedentes sobre o estado de todo o mundo em muitas escalas.

Algumas vantagens do *crowdsourcing*:

- A capacidade de recolha de informação deixa de ser limitada pela disponibilidade de fontes de informação: mais fontes, mais informações.
- É relativamente mais barato do que a utilização de agentes especializados para a recolha de informação.
- Permite a triangulação de informações, o que facilita a sua verificação.

Tipologias de *Crowdsourcing*:

- Ilimitado: um grande grupo de agentes envolvidos na transmissão de dados. Este sistema permite um número ilimitado de entrada de informações, no entanto a fiabilidade das informações é reduzida.
- Limitado: os dados são transmitidos apenas por um grupo específico de agentes. Este sistema permite que a informação seja verificada no momento de entrada, no entanto existe o problema relacionado com a limitação das fontes.
- Junção entre *Crowdsourcing Limitado* e *Ilimitado*: as informações são transmitidas por um grupo específico de agentes, mas também pela comunidade em geral. As fontes confiáveis e não confiáveis são combinadas. Este sistema permite:
 - Aumento da quantidade de informação.
 - Aumento da capacidade de validar dados de fontes desconhecidas.

O fenómeno da Wikipedia confirmou que *crowdsourcing* realmente funciona. Muito tempo passou desde que a Wikipedia foi adotada pelas massas. Atualmente existem diversas outras fontes de dados construídos pela população, por exemplo Flickr, Picasa, Instagram e o OpenStreetMaps. Por exemplo, no OpenStreetMap os dados do mapa são publicados sob uma licença de conteúdo aberto, com a intenção de promover a utilização e a redistribuição dos dados gratuitos. O mapa é inteiramente construído por voluntários que podem ou não ser especializados em cartografia.



Figura 5 Logotipo do OpenStreetMap

2.3. CONCEITOS FUNDAMENTOS DE APLICAÇÕES DE MANUTENÇÃO E GESTÃO INDUSTRIAL

Uma planta industrial carece de uma organização dos diferentes processos de gestão, entre os quais os de serviços de manutenção. Os serviços de manutenção em ambiente industrial seguem um conjunto de passos bem definidos ao nível da instituição, podendo ter várias interações com diferentes departamentos ou hierarquias.

Estas interações e procedimentos a seguir podem ser facilitados com a incorporação de tecnologia no terreno.

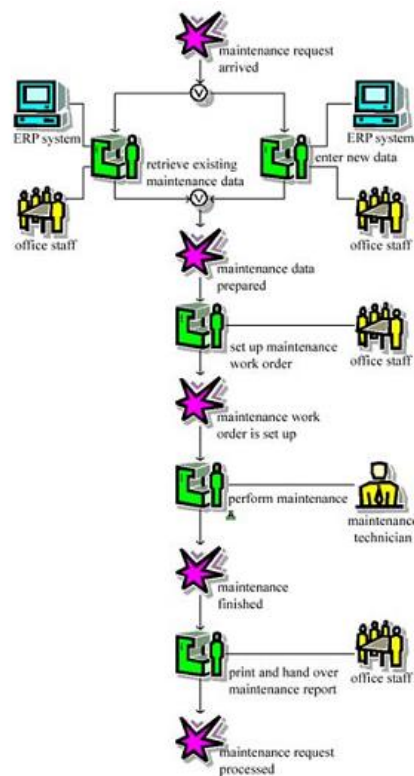


Figura 6 Processos de manutenção da indústria

Os procedimentos de manutenção, bem como registos de inspeção e consequentes distribuições podem ser suportados por dispositivos portáteis de fácil e de utilização.

3. ESTADO DA ARTE

Existem muitas formas dos dispositivos *wearable* melhorarem tanto a vida cotidiana como o trabalho das pessoas, neste capítulo, inicialmente vão ser introduzidos alguns conceitos básicos diretamente ligados ao *wearable*, depois vão ser apresentadas algumas das aplicações utilizadas comercialmente e finalmente alguns exemplos de dispositivos/protótipos específicos de algumas empresas/universidades.

3.1. CONCEITOS

3.1.1. ASSISTENTE INTERATIVO

Os agentes estão normalmente associados à internet, tendo em conta o auxílio que podem dar ao utilizador na realização de melhores e mais completas pesquisas. Pegando neste mesmo conceito mas associado aos dispositivos *wearable*, pode ser obtido um impacto significativo sobre a forma como trabalhamos em geral: agentes inteligentes contextualmente conscientes que nos podem alimentar com informações em tempo real ou mensagens pessoais, tendo em conta o contexto em que o utilizador se encontra.

Com a evolução de toda a envolvente deste tipo de dispositivos, a monitorização das ações do utilizador será constante fazendo com que o dispositivo passe a agir de forma adequada

(por exemplo, um *wearable* não deve interromper as ações do utilizador). Além disso, o *wearable* deve ser capaz de apresentar informações de uma forma que seja apropriada para o utilizador: antes de conduzir, pode ser interessante ver um mapa da distância a ser percorrida, mas durante a viagem, é preferível que os dispositivos leiam em alta voz as direções ("vire à esquerda no próximo cruzamento") ou discretamente apresente setas com as direções. Em última análise, um dispositivo *wearable* irá fornecer as informações mais relevantes da forma mais correta, no lugar apropriado e na hora certa.

3.1.2. REALIDADE AUMENTADA

A realidade aumentada associada a dispositivos *wearable* é um tópico muito pertinente, especialmente quando se fala em dispositivos que permitam ver o mundo com informações relevantes associadas em tempo real. Interpretar os sinais de uma camara montada na cabeça, e posteriormente apresentar num ecrã frontal ao olho informações relacionadas com manuais de reparação, permitiria a um operário desempenhar com muito mais eficiência tarefas em que tem constantemente que parar para olhar para o manual ou instruções. Este conceito assenta sobre um dos principais conceitos dos dispositivos *wearable*, a capacidade de melhorar as capacidades do utilizador sem restringir os seus movimentos.

3.1.3. DOMÓTICA / CASA INTELIGENTE

O contexto casa é um local bastante propício à utilização de dispositivos *wearable* tendo em conta o facto de ser um ambiente com grande concentração e diversidade de equipamentos eletrónicos com os quais necessitamos de interagir diariamente. Desde o simples ato de ligar ou desligar a iluminação de uma divisão ao ato mais complexo de verificar se todos os equipamentos eletrónicos estão desligados no momento em que saímos de casa tendo como objetivo eliminar os consumos desnecessários de energia.

3.1.4. RECONHECIMENTO DO CONTEXTO

Os dispositivos *wearable* permitem recolher diverso tipo de informação do ambiente e do utilizador através de vários tipos de sensores. Estes podem variar, desde simples sensores de temperatura até conjuntos de sensores com dados complexos que necessitam de ser agregados e interpretados.

O objetivo principal é encontrar métodos de reconhecimento de atividades efetuadas pelo utilizador. A interação entre o utilizador e o dispositivo *wearable* pode ser implementada após o sistema ser capaz de reconhecer este tipo de informação.

Os métodos atuais de reconhecimento utilizam técnicas de Inteligência Artificial, no entanto, as mais recentes investigações recorrem ao conceito de aprendizagem supervisionada [25]. Este método baseia-se na análise de dados em bruto para extração de características e preparação de conjuntos de teste, sendo que o objetivo é utilizar a aprendizagem não supervisionada para reconhecimento do contexto. Este tipo de aprendizagem pode permitir o desenvolvimento de novos métodos que podem ser aplicados a vários domínios de aplicações, pois proporciona a aprendizagem de diferentes contextos autonomamente. Ashbrook e Starner apresentam um exemplo para este métodos de aprendizagem utilizando dados GPS.

3.1.5. BOMBEIROS/FORÇAS DE SEGURANÇA

Os dispositivos *wearable* apresentam uma grande utilidade em unidades de proteção (bombeiros, forças de segurança, etc). Estes dispositivos podem ser equipados com vários tipos de comunicações, WLAN, TETRA, GSM, UMTS, etc. Com estas opções em termos de comunicações, é possível ao equipamento selecionar a comunicação que mais se adequa à situação atual.

Outra funcionalidade deste tipo de dispositivos é a transmissão de imagens de vídeo ou até imagens termográficas, que em aplicações relacionadas com bombeiros, permite a visualização de objetos ou pessoas através do fumo. Estes tipos de imagens podem ser bastante úteis a um centro de operações, que poderá tomar decisões sobre as próximas ações das unidades de intervenção.

Com estes dispositivos, poderá ser possível a visualização de mapas dos edifícios ou das áreas onde a equipa de intervenção se encontra, permitindo uma melhor orientação das equipas e um reconhecimento prévio da zona de ação. A possibilidade de monitorização da localização geográfica destes dispositivos, associada aos sinais vitais do utilizador, permite saber o estado atual de cada elemento da equipa, possibilitando a prévia deteção de cansaço ou perda de sentidos.

3.2. APLICAÇÕES

3.2.1. RECOLHA DE INFORMAÇÕES

A gravação de dados e a obtenção de dados através de uma ligação sem fios é possível e comum atualmente em sistemas comerciais, tendo sido já utilizados com sucesso para obtenção de dados em zonas urbanas (dados geofísicos, manutenção de infraestruturas, etc). A recolha de dados com recurso a dispositivos *wearable* permitiria um aumento significativo do tipo e da quantidade de dados recolhidos, possibilitando assim a criação/crescimento de serviços.

3.2.2. FORMAÇÃO / TREINO

Os dispositivos *wearable* podem disponibilizar informações sobre tarefas específicas ou mesmo assistência quando o operador mais precisa. O facto de enviar operadores para ações de formação traduz-se num custo que pode ser ineficaz, isto se comparado com a possibilidade dessa formação ser disponibilizada constantemente no local de trabalho, bastando para isso mudar a abordagem de formação/treino para suporte constante aos operadores enquanto completam as suas tarefas. Integrar as muitas tecnologias educacionais ou de suporte existentes num único sistema diretamente associado às ferramentas de trabalho poderá ser um grande desafio, no entanto será uma forma de demonstrar a potencialidade deste tipo de sistemas.

3.2.3. MANUTENÇÃO / INSPEÇÃO

Existem muitas aplicações para dispositivos *wearable* nos trabalhos de manutenção e inspeção, por exemplo inspeção de linhas de montagem ou de veículos entre outras aplicações.

Manuais de manutenção e operação são normalmente caracterizados por um grande volume de informação que vai sofrendo alteração ao longo do tempo. Por exemplo, um simples avião pode ter um manual com mais de 100.000 páginas [7]. Contudo, devido a alterações operacionais ou simplesmente por atualização, metade dessas páginas podem ficar desatualizadas a cada 6 meses. Em vez da distribuição de material informático (como por exemplo DVDs) por pessoa da manutenção sempre com o risco de um procedimento ser realizado recorrendo a informação desatualizada, as instalações onde decorrem as

manutenções normalmente mantêm a informação centralizada, sendo feito um pedido por parte dos operários sempre que necessário, no entanto em algumas situações são feitas atualizações semanalmente. Disponibilizar esta informação recorrendo a dispositivos *wearable* poderá poupar tempo e dinheiro, disponibilizando ao operador as informações sempre atualizadas, permitindo ainda que este não necessite de sair do local de trabalho podendo dessa forma, manter os olhos e as mãos na máquina a inspecionar.

Apenas como exemplo, atualmente a inspeção de veículos na maioria dos centros de inspeção é apenas indiretamente suportada por computadores. Aquando do início de uma inspeção, o inspetor dirige-se a um computador central e imprime a lista de itens que devem ser inspecionados, transportando este papel durante todo o processo. Todos os resultados e notas devem ser registados no documento, pelo que o inspetor necessita de interromper o que está a fazer sendo ainda necessário no final da inspeção introduzir no computador todos os dados registados no papel. Com a grande evolução das plataformas de computação moveis, tecnologias de reconhecimento de voz e comunicação sem fios é cada vez mais sustentável a ideia de facilitar este tipo de trabalhos fazendo com que o utilizador tenha tudo o que necessita disponível de uma forma simples e eficiente.

3.2.4. MILITAR

Os militares podem usar dispositivos *wearable* como parte integrante do seu vestuário de combate ou uniforme. O pessoal de apoio, de treino, médicos, entre outros, pode também usar este tipo de sistema quando estão a exercer as suas funções, eliminando as perdas de tempo quando têm de se deslocar entre o local de trabalho e o computador.

Atualmente, os militares de combate já estão equipados com armas, mantimentos e equipamentos. A utilização de dispositivos *wearable* dá a uma vantagem em termos de poder computacional e possibilidade de armazenamento de dados por um peso insignificante. Usado para melhorar a decisão de mobilização ou desmobilização das tropas, incluindo as forças especiais, o dispositivo *wearable* é o próximo avanço no campo de batalha digital. Com a utilização do GPS, é possível indicar as localizações das tropas, reportar ameaças críticas e manter contacto próximo com a cadeia de comando, para receber ordens, criar relatórios e pedir reforços.

3.3. EQUIPAMENTOS / PROTÓTIPOS

Os dispositivos *wearable* foram adotados já algum tempo por algumas organizações, utilizando-os de formas inovadoras de modo a melhorar a eficiência e a qualidade nos seus processos. Mais recentemente algumas organizações inovaram ao disponibilizar os dispositivos *wearable* para tarefas comuns do quotidiano.

3.3.1. BOEING

A Boeing tem vindo a equipar os operários das suas fábricas com dispositivos *wearable* desde junho de 1997 para acelerar a produção [8]. Utilizando realidade aumentada os operários podem ver diagramas de montagem, digitalmente adicionados ao seu espaço de trabalho, reduzindo a necessidade de trabalharem com grandes quantidades de diagramas em papel. De acordo com o artigo da CNN, a experiência da Boeing utilizando os dispositivos *wearable* possibilitou um aumento em 50% da produção de cabos, em comparação a métodos alternativos.

3.3.2. NIKE FUEL BAND

Equipada com um acelerômetro a pulseira lançada pela Nike este ano (2013) é capaz de gravar e interpretar os movimentos do utilizador, como caminhar, correr, dançar, jogar golfe, basquete, futebol e muitos outros desportos. Todos esses cálculos são feitos pela pulseira e são transformados em números tais como, calorias e passos, além de mostrar a hora atual.



Figura 7 NIKE FUEL BAND

3.3.3. MYO (*GESTURE ARMBAND CONTROLLER*)

O MYO [9] trata-se de uma pulseira elástica que é colocada no antebraço e que é capaz de ler os impulsos elétricos que movimentam os músculos.



Figura 8 MYO

O sistema consegue detetar a intenção do movimento mesmo antes do músculo começar a reagir. Para além disso, conta ainda com acelerómetros e giroscópios para detetar outros tipos de movimentos. Desta forma o objetivo é permitir a interação com computadores ou outros dispositivos digitais através de uma ligação sem fios Bluetooth.

3.3.4. SAMSUNG GALAXY GEAR

O relógio Galaxy Gear [10] foi anunciado pela Samsung em Setembro deste ano (2013). Este dispositivo está equipado com um ecrã de 41,4 mm, um processador de 800MHz, Bluetooth 4.0, acelerómetro, giroscópio, uma coluna de som e uma bateria de 315 mAh, que permitirá até um dia de utilização.



Figura 9 Samsung Galaxy Gear

O conceito deste dispositivo é ser um complemento ao telemóvel. Este permite reduzir a interação direta com o telemóvel, por exemplo, deixará de ser necessário tirar o telemóvel do bolso para algumas ações básicas, como ler mensagens.

3.3.5. CONTROLO REMOTO

O sistema de controlo remoto no pulso [11] visa permitir uma interface gestual natural quando comparada com os controlos remotos universais, podendo ainda ser utilizado como apontador durante apresentações.

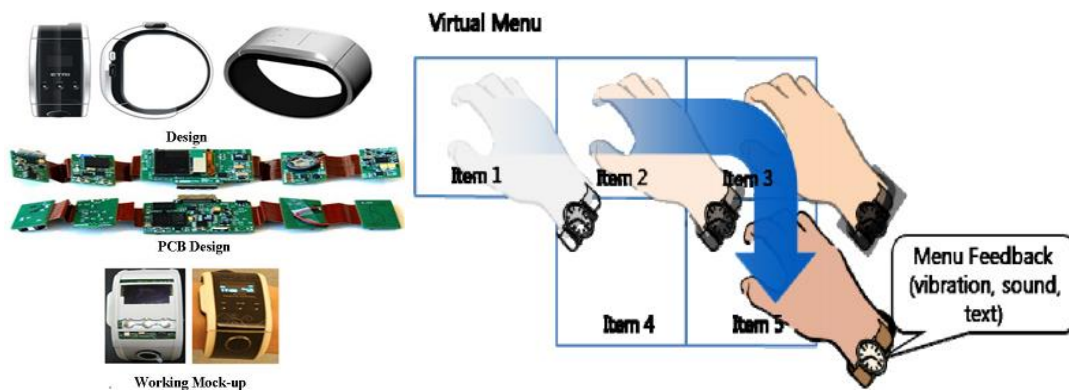


Figura 10 Controlo Remoto no Pulso

O sistema baseia-se num conceito de menu virtual de modo a facilitar o controlo de diferentes equipamentos, sendo que este permite efetuar alguns controlos básicos como navegar num menu ou controlar uma barra de nível. De forma a tudo isto ser possível o sistema monitoriza continuamente os movimentos do braço com o objetivo de identificar, uma de quatro direções (direita, esquerda, cima e baixo), ação de selecionar (*click*) e as rotações do pulso (rotação para a esquerda e rotação para a direita).

3.3.6. GOOGLE GLASS

O Google Glass é um dispositivo *wearable* com um display posicionado numa zona frontal ao olho que integra várias soluções de visualização, áudio e *software* para criar a próxima geração de óculos, que disponibilizam informações do género dum telemóvel mas num formato de mãos livres.



Figura 11 Aparência do Google Glass

O Google Glass projeta as informações num ecrã posicionado logo acima do campo de visão do utilizador, tais como, mapas, músicas, previsão do tempo, fazer chamadas de vídeo e até tirar fotos. O dispositivo pode ser controlado tanto pela haste sensível ao toque quanto por comandos de voz.

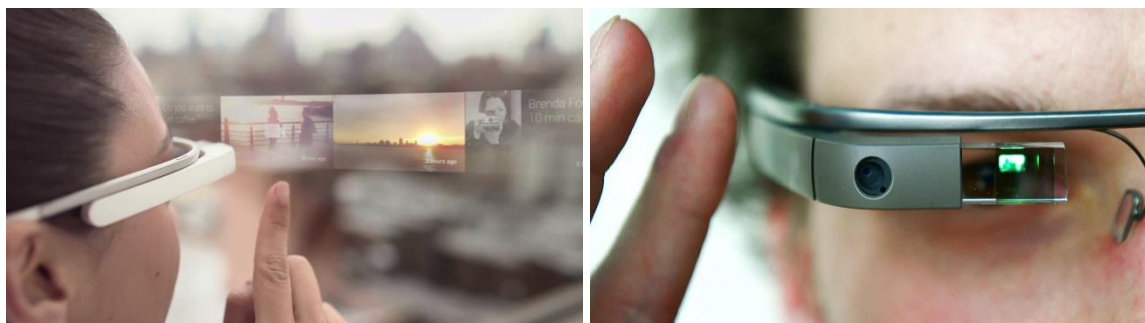


Figura 12 Interfaces do Google Glass

Este equipamento tem ligações Wi-Fi, Bluetooth e pode utilizar o telemóvel como ponto de acesso à internet, inclusive através da rede 3G. Por enquanto, tendo em conta a fase de testes em que se encontra, o dispositivo possui poucas aplicações. No entanto o objetivo do Google é revolucionar o nosso dia-a-dia, assim como fez com a introdução do Android.

3.3.7. OMNITOUCH

O OmniTouch [12] é um *wearable* com capacidades de detecção 3D e projeção que permite aplicações interativas *multitouch* em superfícies do quotidiano.

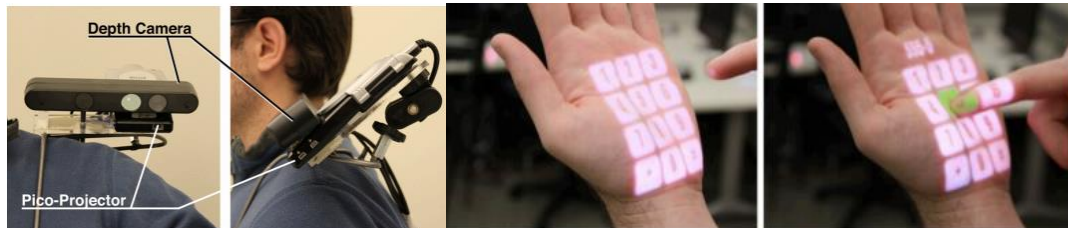


Figura 13 Protótipo de sistema OmniTouch [12]

Além do sistema apoiado no ombro, não existe nenhum outro equipamento instalado. Acima de tudo, o sistema permite que o utilizador utilize as mãos, braços e pernas como superfícies gráficas interativas. Os utilizadores podem ainda, facilmente recorrer a superfícies do ambiente para expandir a área interativa (por exemplo livros, paredes e mesas). Em tais superfícies sem qualquer calibração, o sistema fornece capacidades semelhantes às de um rato ou um ecrã tátil, sendo essas informações a localização 2D (X e Y) e o estado dos dedos, permitindo assim uma grande variedade de interações. Para uma utilização confiável nas mãos por exemplo, os botões devem ter entre 2 a 3 centímetros de diâmetro, sendo assim concebível que grande parte das ações que se fazem com os dispositivos móveis de hoje em dia pudessem ser realizadas na palma da mão.

3.3.8. WR1100 - COMPUTADOR PESSOAL PARA UTILIZAÇÃO NO PULSO

O Zypad WR1100 [13] é um computador pessoal robusto para utilização no pulso. Este foi projetado para condições ambientais adversas. Este dispositivo foi projetado de forma a cumprir a norma MIL-STD-810F e os requisitos da norma MIL-STD-461E (ambas normas de cariz militar), sendo desta forma uma solução ideal para militares, seguranças e outros serviços de emergência.



Figura 14 WR1100 - Computador pessoal para utilização no pulso [13]

Este equipamento pode ser facilmente configurado para aceder a sistemas remotos através das suas interfaces de comunicação com e sem fios, utilizando o sistema operativo Linux. Esta unidade integra um conjunto de características inovadoras, que incluem as interfaces 802.11 / *Bluetooth* / *Zigbee*, um recetor GPS, bússola eletrónica e um sensor biométrico de impressões digitais. Uma das principais características do WR1100 é a sua modularidade, permitindo ao utilizador alterar algumas das funcionalidades do equipamento de uma forma simples, bastando para isso substituir módulos de *hardware*. Por exemplo, se o utilizador pretender abdicar do recetor GPS em função de um módulo GPRS, basta para isso substituir o módulo em causa pelo outro (*plug and play*). O módulo de bateria pode também ser facilmente removido e substituído a qualquer altura. O involucro do equipamento é constituído por fibra de vidro reforçada com *nylon* de modo a maximizar a sua durabilidade e minimizar o seu peso, juntamente com um ecrã sensível ao toque de alta resolução protegido contra a entrada de água e poeiras. O sistema de apoio para o braço permite um posicionamento ergonómico e de fácil fixação, mesmo sobre as roupas de trabalho, garantindo assim, a distribuição de peso ideal e máximo conforto.

4. TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO

Neste capítulo descrevem-se as tecnologias e as ferramentas de desenvolvimento adotadas neste trabalho. São apresentadas as linguagens de programação utilizadas, assim como os conhecimentos considerados necessários.

4.1. SISTEMAS EM TEMPO REAL

Um sistema em tempo real é um sistema que tem de responder a estímulos externos ou a serviços durante um tempo específico, independentemente da carga do sistema, a ação a ser executada tem de ser previsível. Também é desejável que um sistema em tempo real atinga a sua correta funcionalidade e cumpra em termos temporais enquanto é utilizado.

4.1.1. TERMINOLOGIA DOS SISTEMAS EM TEMPO REAL

Existem alguns termos comuns dentro dos sistemas em tempo real, tais como:

- Job: Um *job* é uma unidade de trabalho que pode ser executada ou escalonada pelo sistema. Um exemplo disso é uma função de leitura de uma entrada digital.
- Task: Uma tarefa é uma lista de trabalhos que estão relacionados e que podem ser agrupados numa única função. Exemplos disso são um conjunto de trabalhos, por exemplo: ler uma posição de memória, fazer alguns cálculos e escrever o resultado numa outra posição de memória. Uma tarefa pode ser escalonada para ser periódica se tiver de ser executada em intervalos de tempo regulares.
- Tempo de resposta: O tempo de resposta de uma tarefa é o tempo entre o início de execução da tarefa até ao seu término.
- Atraso: Refere-se ao atraso na execução de uma tarefa relativamente ao tempo em que deveria ser executada. Pode-se dizer que uma tarefa tem um atraso de zero se esta terminar a sua execução antes ou no tempo em que deveria ter terminado.

4.1.2. CLASSIFICAÇÃO DOS SISTEMAS EM TEMPO REAL

Os sistemas em tempo real podem ser classificados em duas categorias, baseadas nos requisitos temporais e na criticidade de cumprimento do seu tempo de execução. As duas categorias são:

- Hard Real-Time Systems: Um sistema em tempo real é considerado de hard real time quando tem restrições temporais de execução muito apertadas. Estes sistemas têm obrigatoriamente de executar tarefas críticas, em que o não cumprimento do seu tempo de execução pode ter graves consequências. Tipicamente os sistemas hard real time encontram-se em equipamentos tais como sistema de travagem ABS, *pace makers* e controladores dos aviões.
- Soft real-time systems: Os *soft real-time systems* estão mais preocupados em ter uma boa estabilidade do sistema e permitir que todas as tarefas sejam executadas. Este tipo de sistemas são utilizados para executar tarefas menos críticas e que podem tolerar pequenos atrasos na conclusão da execução de uma tarefa. Ao contrário dos hard real time, os atrasos nos tempos de execução destes sistemas não têm consequências graves. Em geral este tipo de sistemas tem uma probabilidade 95% para que uma tarefa seja executada no tempo pedido.

4.2. RTX - SISTEMA OPERATIVO

O RTX (*Real Time eXecutive*) é um RTOS determinístico, projetado para ARM e dispositivos CortexM. O RTX permite criar programas que executam várias funções em simultâneo (tarefas ou processos criados estaticamente) e ajuda a criar aplicações melhor estruturadas e de mais fácil manutenção. Às tarefas podem ser atribuídas prioridades de execução. O *kernel* RTX utiliza as prioridades de execução para seleccionar a próxima tarefa a ser executada (agendamento de preferência). Este sistema operativo contém ainda funções adicionais para as comunicações entre tarefas, gestão de memória e de periféricos.

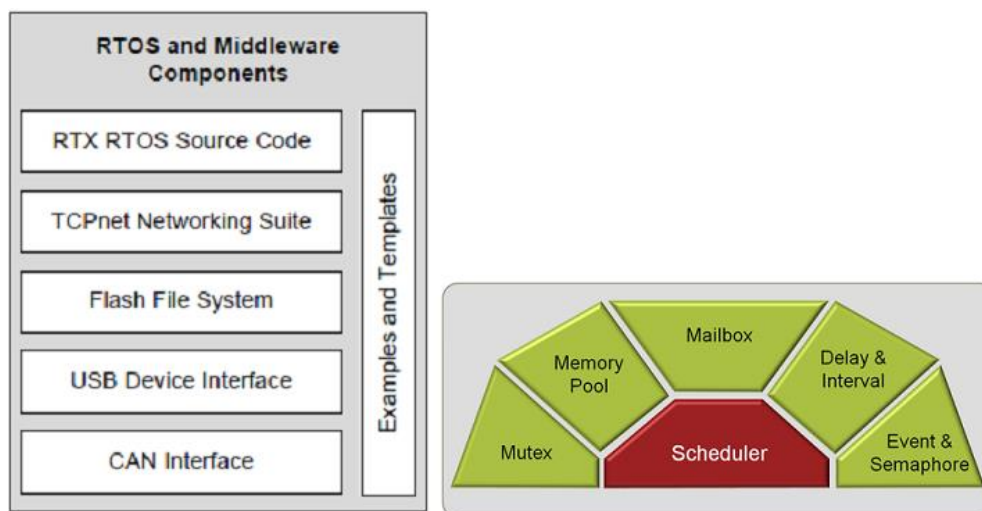


Figura 15 Arquitetura do RTOS RTX

As principais características do RTX incluem:

- É um RTOS *royalty-free*
- Programação flexível: *round-robin*, *preemptiva* e de colaboração
- Operações em tempo real com baixa latência de interrupção
- Tamanho reduzido para sistemas com recursos limitados
- Número ilimitado de tarefas, cada uma com 254 níveis de prioridade
- Número ilimitado de filas de mensagens, semáforos, *mutex* e temporizadores
- Suporte para *multithreading* e operações *thread-safe*.

4.3. EM::BLOCKS (C/C++ IDE)

O Em::Blocks [14] é um *software* livre e de código aberto, criado para responder às diversas necessidades dos utilizadores que trabalham na área de desenvolvimento de *software* embebido. Este *software* é bastante extensível e totalmente configurável, apresentando um estilo semelhante ao Visual Studio, sendo no entanto baseado em CodeBlocks [15] e no GCC [16].

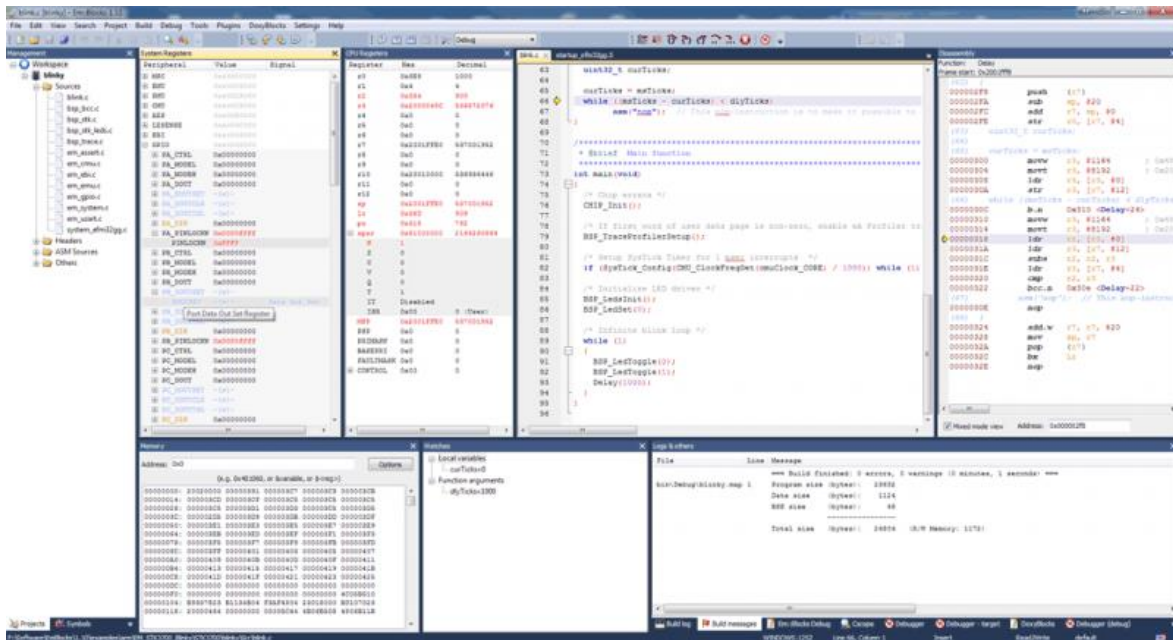


Figura 16 Aspetto do ambiente de desenvolvimento Em::Blocks

Características principais:

- Múltiplos dispositivos (ARM, PIC, AVR, entre outros).
- Compilador ARM GNU "*bare-metal*" incorporado com diferentes bibliotecas otimizadas.
- Depurador totalmente otimizado para desenvolvimento de sistemas embebidos.
- Escrito em C++. Não necessita de linguagens interpretadas (Java e NET) ou de bibliotecas proprietárias.
- Permite a instalação de vários *plugins* (Doxygen, FilleDif, etc).

4.4. ALTIUM DESIGNER

O Altium Designer é um *software* de desenvolvimento utilizado para a criação de placas de circuito impresso. Este permite desenhar o esquemático e posteriormente a placa de circuito impresso. Uma das suas principais vantagens é a capacidade de incorporar o modelo 3D dos componentes, o que é importante quando se está a lidar com espaços extremamente reduzidos, algo inerente a um dispositivo que se considere *wearable*.

4.4.1. ESQUEMÁTICO

O esquemático é a parte do ambiente de desenvolvimento onde se criam as ligações elétricas. Para tal é necessário colocar os símbolos dos componentes e efetuar as ligações pretendidas. Na imagem que se segue está representada uma pequena parte do esquemático criado para o protótipo do equipamento. Aqui podem visualizar-se alguns componentes que foram criados, e as respetivas ligações.

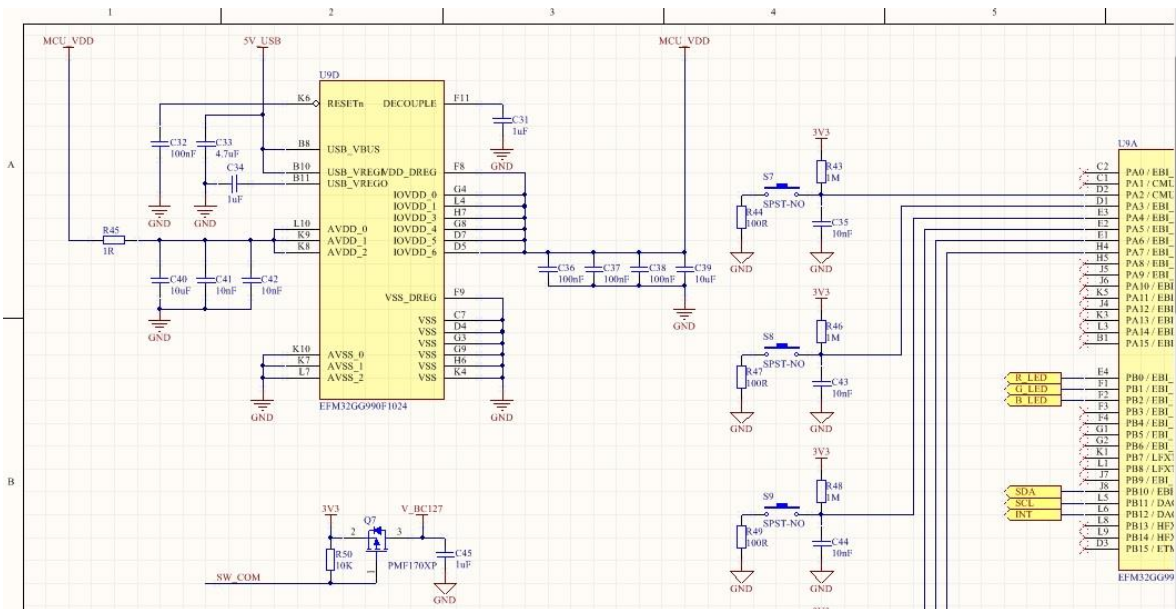


Figura 17 Esquema elétrico

4.4.2. DEFINIÇÃO DE MODELO 3D

O modelo 3D dos componentes tem que ser criado aquando da criação do componente. Este deve ser o mais fiel possível, seguindo os valores fornecidos pelo fabricante. No exemplo que se segue está o modelo do microprocessador.

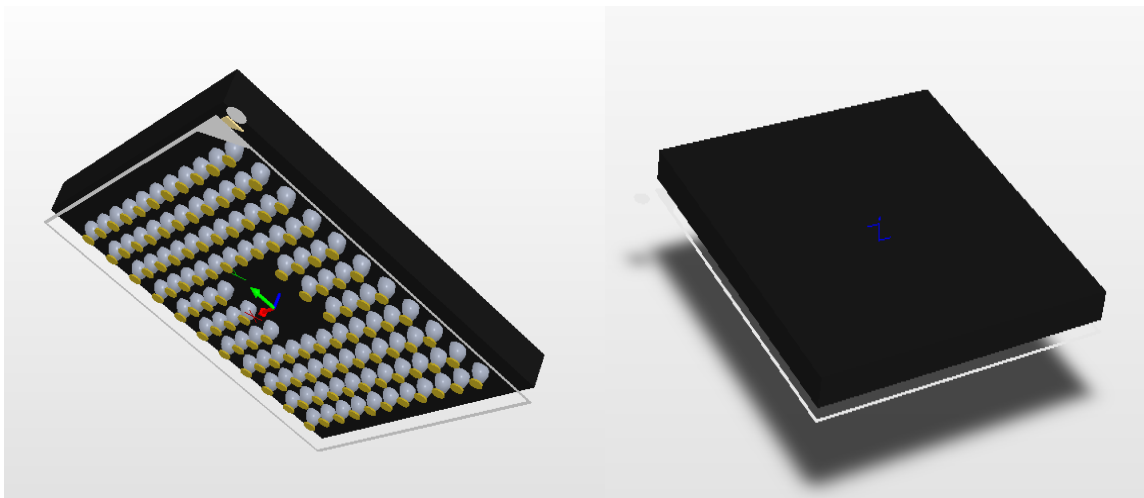


Figura 18 Modelo 3D do microprocessador

5. ESPECIFICAÇÃO E DESCRIÇÃO DO SISTEMA

O sistema proposto procura responder às diversas necessidades que acompanham a evolução da tecnologia e do ser humano. Este sistema pode ser visto como um prestador/facilitador de serviços que pretende estender e facilitar a forma como o utilizador interage com o mundo bem como dotar o utilizador de novas capacidades.

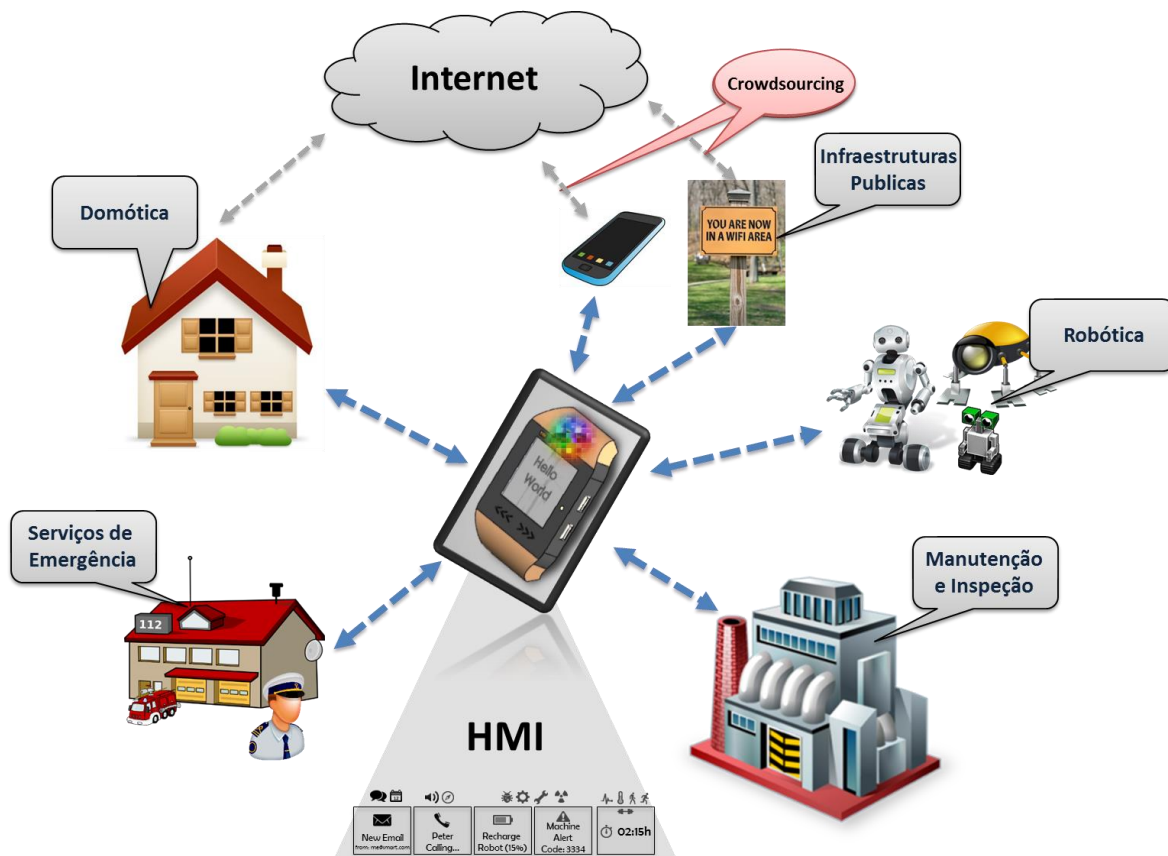


Figura 19 Interação do dispositivo com o mundo

De uma forma geral o sistema proposto poderá virtualmente interagir com tudo o que o rodeia, bastando para isso que exista uma compatibilidade protocolar nas comunicações.

O sistema permite assim a extensão de aplicações e serviços para um contexto de utilização mais fácil e rápido. Esta extensão só é possível com recurso a facilidades de comunicação, nomeadamente Bluetooth, WIFI e eventualmente RF FM.

Estas facilidades de comunicação permitem alavancar a utilização do sistema em diversos cenários, tais como:

- Extensão do telemóvel;
- Comunicação com aplicações de redes locais, como exemplo uma instalação industrial, em ambiente doméstico, com uma infraestrutura local do género de um carro ou ambulância;
- Publicação de dados do ambiente para uma plataforma em nuvem para tratamento das grandes quantidades de dados.

- Monitorização do portador do dispositivo, quanto a sensores de informação vital assim como de atividade física ou até da exposição ambiental como ruído, UVs, etc.
- Terminal de acesso a informação, derivada da internet ou outros sistemas associados

O interface do dispositivo multimodal permite ao utilizador uma interação mais rica e bem conseguida. A interação com o dispositivo nos seus diversos cenários recorre a um display e sinais sonoros e de vibração como outputs da interface, podendo este ser controlado com recurso a alguns botões, de pressão e deslizantes, bem como pequenos toques.

5.1. REQUISITOS DO SISTEMA

Este capítulo apresenta um conjunto genérico de necessidades independentes dos casos de aplicação aqui explorados. Os requisitos mais específicos de cada caso de utilização são apresentados junto da explicação desses mesmos casos.

5.1.1. INTERFACE

A arquitetura do sistema deverá permitir a implementação de interfaces *user-friendly* cujo *design* considere a capacidade de aprendizagem de utilização rápida e interface intuitivo, facilidade em lembrar como se usa, eficiência e produtividade da utilização, tolerância a erros do utilizador. Os interfaces multimodais (exemplos: áudio, visual, toque, toque deslizante e reconhecimento gestual) também devem ser privilegiados nas diferentes aplicações de forma a criar uma interação mais rica com o utilizador.

5.1.2. MULTI-APLICAÇÃO

O desenho do sistema deve assentar na decomposição e suportar modularidade para permitir a criação de um ambiente multi-aplicação, permitindo o desenho e desenvolvimento independentes de aplicações e subsistemas.

Para tal, a arquitetura deve garantir isolamento de interferências não esperadas entre os diferentes subsistemas integrados.

A especificação da interface da arquitetura tem de permitir a integração de novos subsistemas sem a ser necessário perceber o funcionamento interno dos componentes da arquitetura.

5.1.3. DETERMINISMO TEMPORAL

A arquitetura do sistema deverá garantir que o tempo de arranque e execução dos componentes de *software* é conhecido e limitado, de forma a permitir implementar tolerâncias a falhas, reiniciando o componente após a deteção da falha.

Adicionalmente, a arquitetura deverá garantir a sincronização computacional e de comunicações de serviços distribuídos e validação temporal dos dados de tempo real gerados pelos diferentes subsistemas. Para tal, um serviço de tempo global deve ser distribuído a todos os serviços.

5.1.4. TRANSFERÊNCIA DE MENSAGENS

A arquitetura deverá permitir transferência de mensagens despoletadas por eventos, desenhada tendo em conta uma carga de comunicação média, permitindo falha de comunicação durante cenários de elevada procura de comunicação.

A transferência de mensagens em tempo real também deverá ser permitida, suportando o transporte de mensagens de forma determinística temporal.

A infraestrutura de comunicação física deverá permitir ser partilhado pelos diferentes serviços da arquitetura.

5.1.5. GESTÃO DE RECURSOS

A arquitetura deverá permitir a escalabilidade dinâmica da performance dos recursos de *hardware* com base no nível atual de energia disponível, nomeadamente controlo de distribuição de energia para componentes não usados ou até reconfiguração e ajuste da frequência de relógio.

A arquitetura do sistema deverá permitir a reconfiguração tal que possibilite um conjunto alargado perfis de performance a poderem ser usados em diferentes aplicações.

Para a implementação de serviços e aplicações críticas e relacionadas com segurança, a arquitetura garantir a coexistência de mecanismos de alocação de recursos estáticos e dinâmicos e recursos de comunicações mesmo que estes recursos sejam partilhados.

5.1.6. EFICIÊNCIA ENERGÉTICA

Os consumos e eficiência energética são atributos de qualidade bastante importantes, especialmente para os dispositivos portáteis, sendo assim o desenho de baixo consumo e medição dos próprios consumos deverão também fazer parte da arquitetura do sistema.

A gestão dos consumos deve incluir a adaptação e ajuste de componentes a cada aplicação e performance de um uso específico, para que as aplicações corram com os menores recursos e energia possível. Os consumos de *standby* devem ser de muito baixo consumo, fazendo parte da gestão dos consumos.

A arquitetura do sistema deverá permitir a recolha e armazenamento da informação da energia fornecida e energia remanescente, consumos atuais, consumos de pico e temperatura. Estas informações devem ser apresentadas com um interface adequado a utilizadores e desenvolvedores. No entanto, a arquitetura também deverá permitir configurações por defeito, para que o utilizador final não necessite de fazer nada relacionado com a eficiência energética para usar o dispositivo e as diferentes aplicações.

5.1.7. SEGURANÇA E CONFIDENCIALIDADE DA INFORMAÇÃO

A arquitetura deverá fornecer mecanismos e serviços para garantir os conceitos mais básicos de segurança: integridade, disponibilidade e confidencialidade assim como conceitos mais secundários como autenticação e autorização de acesso ao sistema.

Para garantir a integridade, a arquitetura deverá ter mecanismos para prevenir modificações de *hardware* e software por pessoal não autorizado. Mecanismos para a prevenção de disponibilização de informação a pessoas ou sistemas não autorizados também deverão ser considerados.

5.1.8. DIAGNÓSTICO E TESTE DO SISTEMA

O sistema deverá conter serviços de diagnóstico que forneçam informação consistente sobre o estado do sistema.

A arquitetura deverá considerar um desenho para a testabilidade, possibilitando uma interface de teste estandardizado para o teste dos componentes durante o funcionamento do sistema.

5.2. CENÁRIOS DE APLICAÇÃO

Neste capítulo vão ser apresentados os cenários específicos que vão ser explorados no decorrer deste documento.

5.2.1. ALERTAS SMARTPHONE

Atualmente a demanda por facilidade de utilização de qualquer dispositivo é cada vez maior. As pessoas querem a informação que pretendem sem ter que a procurar, e esta tem que estar disponível com a menor interação possível. Uma demonstração prática disso é a leitura de mensagens, ou outras notificações em telemóveis. Tome-se o exemplo da receção de uma mensagem durante uma reunião, é necessário retirar o telemóvel do bolso, pegar nele com ambas as mãos, dada a tendência para o aumento de tamanho deste tipo de dispositivos, desbloquear o ecrã, e só depois ler a notificação. Após este processo, é necessário voltar a guardar o telemóvel, sendo por vezes incomodativo.

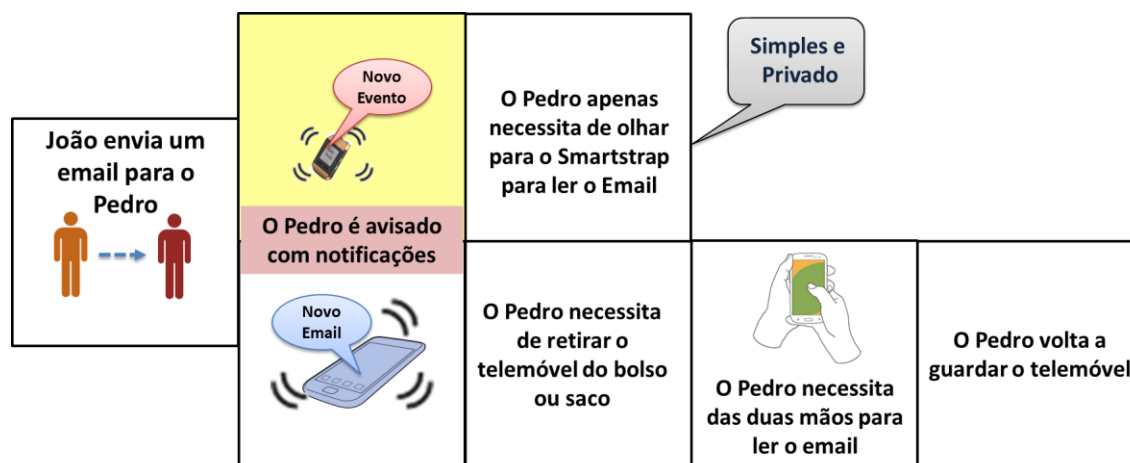


Figura 20 Cenário - Alertas Smartphone

Este ato que é replicado dezenas ou centenas de vezes por milhões de pessoas, pode ser melhorado, e ter menos impacto no quotidiano, se a informação que quisermos visualizar estiver ao alcance de um simples rodar de pulso, o que se traduz numa maior facilidade de acesso à informação pretendida. Esta mesma informação não necessita do ato de ocupar as mãos ao pegar num objeto, uma vez que já está colocado no corpo. Adiciona ainda a vantagem de aceder á informação de forma discreta porque, como o dispositivo está muito próximo do corpo, exige pouca vibração para alertar o utilizador.

Necessidades da arquitetura para o cenário alertas *smartphone*

A utilização do sistema para alertas provenientes do *smartphone* envolve um conjunto de necessidades de interação com o *smartphone* ao nível das comunicações e interoperabilidade de aplicações. Assim, o sistema deverá permitir a interação com várias aplicações do *smartphone*, nomeadamente chamadas, mensagens e correio eletrónico, redes sociais, agenda, etc.

O interface com o utilizador deverá refletir a interface mais usual apresentada no *smartphone*, permitindo incluir as funcionalidades mais básicas e as mais utilizadas com a mesma similaridade de utilização.

5.2.2. CROWDSOURCING DE INFORMAÇÕES METEOROLÓGICAS

Atualmente os serviços de meteorologia disponíveis *online* são essencialmente vocacionados para a previsão da temperatura, disponibilizando a informação de uma forma genérica que pode por vezes induzir o utilizador em erro. Muito devido à dificuldade/incapacidade da instalação de diversas estações meteorológicas numa só zona geográfica (por exemplo uma freguesia), por vezes as informações apresentadas para essa mesma zona são generalizadas e dificilmente correspondem aos valores atuais reais (os serviços online utilizam normalmente previsões e não valores em tempo real).

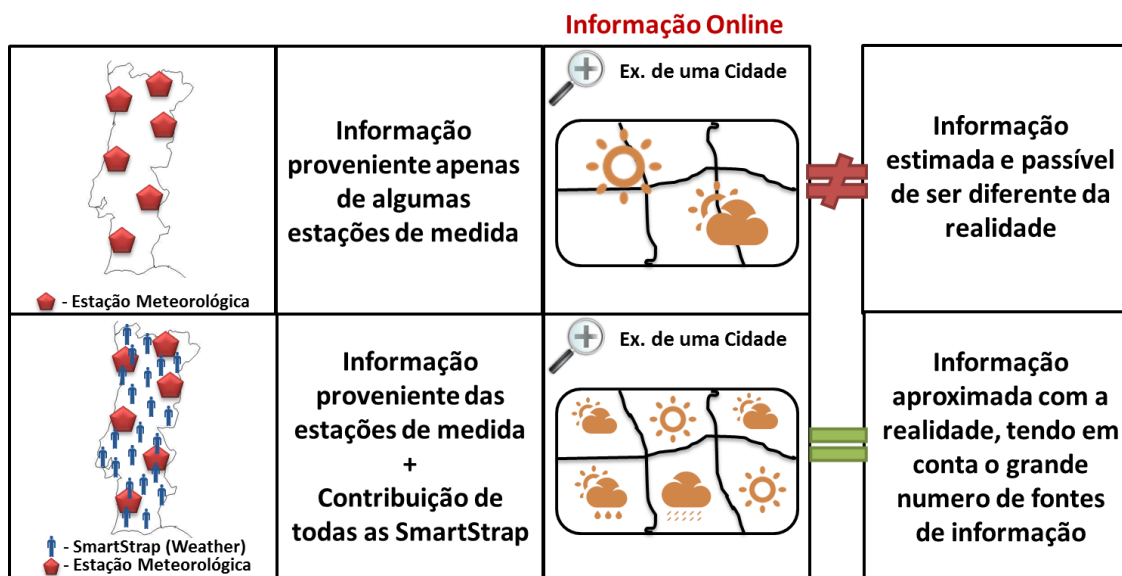


Figura 21 Cenário – Cenário - Crowdsourcing de Informações Meteorológicas

O cenário apresentado procura responder a estas limitações utilizando o conceito de *crowdsourcing* [ver 2.2] aplicado aos dados meteorológicos. A solução proposta passa pelo desenvolvimento de um dispositivo para o pulso que possa ser utilizado no dia-a-dia, idealmente por grande parte da população, que funcione como uma mini estação meteorológica (temperatura ambiente, humidade e radiação ultravioleta). Desta forma a capacidade de recolha de informação deixa de ser limitada pela disponibilidade de fontes de informação (estações meteorológicas comuns) além de permitir a triangulação de informações, facilitando a sua verificação. Por sua vez os serviços *online* poderão disponibilizar a informação com maior resolução geográfica e ainda dados bastante próximos do estado real atual (dados em tempo real).

Necessidades da arquitetura para o cenário *crowdsourcing* de informações meteorológicas

Para este caso de utilização, será necessário que o sistema incorpore os diferentes sensores para recolha de informações de meteorologia e a sua georreferenciação.

A arquitetura deverá conseguir lidar com sensores de temperatura ambiente, humidade relativa e exposição a raios ultravioleta. A georreferenciação poderá ser efetuada com recurso a sistemas de posicionamento por satélite, triangulação da rede GSM ou ainda por inferência de redes WIFI a que o dispositivo se possa ligar.

A transferência da informação poderá ser efetuada por intermédio de uma *gateway*, o *smartphone*, ou diretamente caso o sistema disponha de meios de comunicação adicionais tais como 3G ou WIFI.

A informação meteorológica deverá assim ser enviada para um sistema agregador de informação na nuvem, permitindo depois o tratamento da grande quantidade de dados.

5.2.3. MANUTENÇÃO E INSPEÇÃO NA INDÚSTRIA

Cada vez mais a indústria em geral lida com grandes quantidades de informação relacionadas com a manutenção/inspeção de equipamentos. Apesar de existirem já alguns métodos para otimizar os processos de inspeção, como é o caso das inspeções automáticas recorrendo a máquinas que substituem os operadores humanos, continuam a existir situações tanto de inspeção como manutenção que necessitam de intervenção humana, que por sua vez necessitam de suporte. Um exemplo atual e recorrente é a utilização de manuais de manutenção impressos (papel) normalmente caracterizados por um grande volume de informação sempre sujeitos a alterações/atualizações ao longo do tempo. Além disso estes manuais têm o inconveniente de prejudicar o desempenho do operador, quer por serem demasiado extensos e complexos, o que pode originar diferentes tipos de enganos, quer pelo impedimento que incute ao operador na impossibilidade de efetuar uma intervenção contínua a duas mãos.

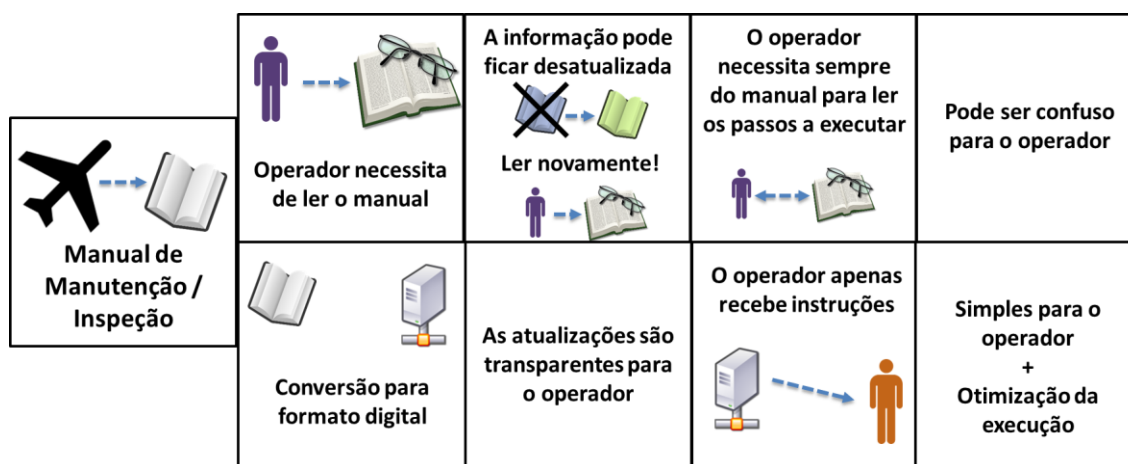


Figura 22 Cenário – Manutenção e Inspeção na Indústria

A solução proposta procura responder a estas limitações disponibilizando as informações relacionadas com os passos necessários para a realização da manutenção ou mesmo da inspeção recorrendo a um dispositivo portátil e ergonómico. A utilização de um dispositivo deste género permitirá a poupança de tempo e dinheiro, disponibilizando ao operador as informações estritamente necessárias sempre atualizadas, permitindo ainda que este não necessite de sair do local de trabalho além de poder manter os olhos e as mãos livres para realizar a manutenção ou inspeção.

Necessidades da arquitetura para o cenário manutenção e inspeção na indústria

A execução no terreno dos processos de manutenção e inspeção com a utilização deste sistema requerem que a arquitetura se interligue com o sistema de gestão da manutenção dessa indústria.

A interação com a plataforma de gestão da manutenção carece que seja possível a visualização da informação relativa aos processos de manutenção e a possibilidade de preenchimento de formulários e relatórios de manutenção.

A arquitetura deverá também permitir a recepção de notificações relativas eventos de erros e falhas de máquinas e sistemas instaladas na planta industrial. A notificação de eventos agendados deverá também ser possível.

5.2.4. MONITORIZAÇÃO REMOTA DE BOMBEIROS E MILITARES

Nos dias de hoje os incêndios são uma realidade incontornável levando a consequências inevitáveis como as diversas lesões a que os bombeiros estão sujeitos. Numa situação de emergência como são os incêndios e outras realidades, a gestão dos operacionais é extremamente importante, no entanto é uma tarefa de muito difícil execução tendo em conta os meios atuais. A gestão dos operacionais é feita tendo em conta algumas ordens que são acatadas pelos mesmos, e mantem essa gestão através de comunicações de voz (walkie-talkie) o que acaba por ser insuficiente em situações onde é necessário gerir tantos operacionais.



Figura 23 Cenário – Monitorização Remota de Bombeiros e Militares

A solução apresentada procura dar resposta a estas insuficiências utilizando para isso um dispositivo individual capaz de permitir a criação de um centro operacional com informações detalhadas de cada um dos operacionais envolvidos. Esta solução permitirá a monitorização dos dados vitais (como o batimento cardíaco e temperatura corporal), a localização GPS e das condições ambientais associadas a cada operacional. Além dos dados relacionados com a monitorização individual com maior interesse para o centro de controlo, o sistema permitira o envio de alertas e/ou ordens diretamente para o operacional, permitindo assim uma rápida disseminação de novas abordagens perante o cenário proporcionado.

Necessidades da arquitetura para o cenário de monitorização remota de bombeiros e militares

A monitorização remota de bombeiros e militares em situações operacionais de alto risco implica que a arquitetura contenha um conjunto de sensores da condição física do operacional com georreferenciação. Estas informações devem ser recolhidas num centro agregador de informação na nuvem e utilizada na gestão operacional.

O sistema também deverá permitir comunicar com o operacional, enviando-lhe informações em forma de mensagem escrita, bem como mensagem de áudio ou vídeo. A conversação com um centro operacional deverá também ser possível.

A robustez do sistema é também uma especificidade desta aplicação, dado que os cenários de utilização exigem um esforço mecânico e eletrónico acrescido.

5.2.5. SINCRONIZAÇÃO E EFEITOS VISUAIS

A área dos espetáculos é uma área que move multidões e onde todas as inovações que proporcionam uma melhoria do espetáculo são facilmente absorvidas. A uns anos atrás era bastante comum a utilização de um isqueiro (chama luminosa) para criar um padrão luminoso em espetáculos, este movimento, além de motivar a multidão acrescentava algo mais ao próprio espetáculo. Hoje em dia a utilização de pulseiras luminosas com cores aleatórias é uma realidade em muitos dos grandes espetáculos pelo mundo fora, no entanto o efeito criado por elas é apenas a variação luminosa completamente aleatória.

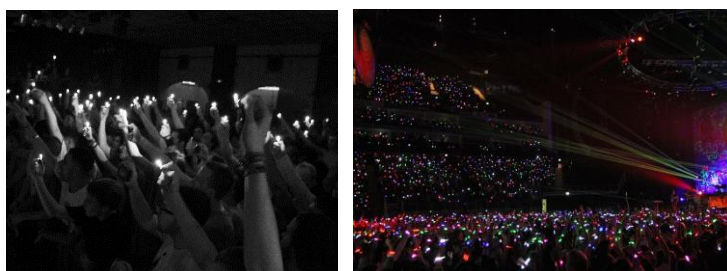


Figura 24 a) Efeito com isqueiros. b) Efeito aleatório com pulseiras multicolor

O próximo passo deste tipo de aplicações para espetáculos poderá passar pelo cenário aqui apresentado, que recorrendo à sincronização e controlo de vários dispositivos individuais com LED multicolor. Este possibilitaria a criação de um padrão ou imagem, como se faz atualmente recorrendo a cartões de múltiplas cores em estádios de futebol, como apresentado na figura seguinte.



Figura 25 Padrão criado com recurso a cartão multicolor

Necessidades da arquitetura para o cenário sincronização de efeitos visuais em multidões

A sincronização de efeitos visuais implica que a arquitetura possa incluir um *buffer* de efeitos visuais e que esta tenha a capacidade de processamento despoletado por um referencial temporal partilhado por todos os dispositivos da multidão.

A arquitetura poderá também permitir um *streaming* em tempo real de efeitos visuais e processamento dos mesmos.

Os efeitos visuais poderão surgir na animação do ecrã ou de dispositivos emissores de luz incorporados no sistema.

6. ARQUITETURA DE HARDWARE

De acordo com as várias características dos dispositivos *wearable*, desde os elementos de detecção e atuação, o armazenamento e processamento de dados, as fontes de energia e comunicação são todos igualmente importantes e necessários.

- As fontes de energia são necessárias tendo em conta a necessidade de energia por parte dos restantes componentes.
- O armazenamento e processamento de dados são necessários para as funcionalidades e implementação de inteligência.
- Os sensores e atuadores completam o ciclo do sistema de controlo.
- É necessária a comunicação entre todos os componentes do dispositivo, entre o dispositivo e outros equipamentos bem como entre o dispositivo e as fontes de informação.

O funcionamento base do dispositivo é apresentado na Figura 26.

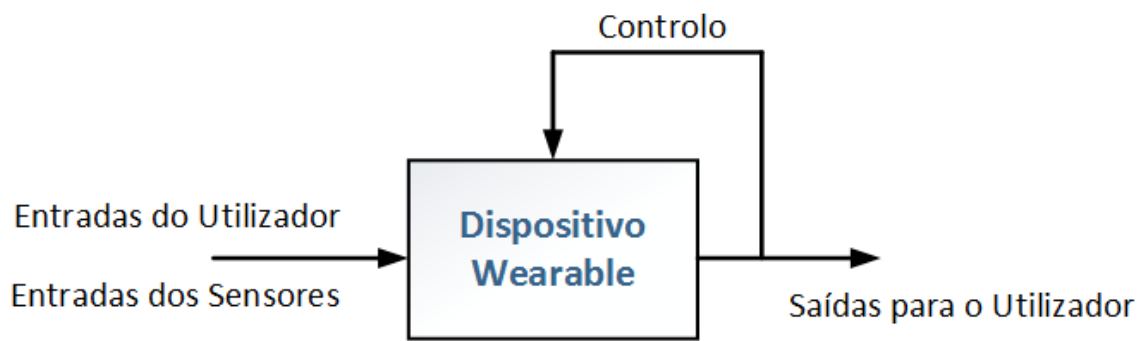


Figura 26 Funcionamento básico do sistema

As entradas para o dispositivo são fornecidas pelo utilizador ou pelos diferentes sensores monitorizando tanto o utilizador como o ambiente. Tendo este tipo de informação como base da lógica de decisão, o sistema proporciona informação ao utilizador e controla o desempenho das funções automatizadas. A seleção dos componentes de *hardware* para este dispositivo baseiam-se nas aplicações pretendidas e nas condições do ambiente que se pretendem monitorizar.

A arquitetura geral do *hardware* elétrico do dispositivo é ilustrada na Figura 27.

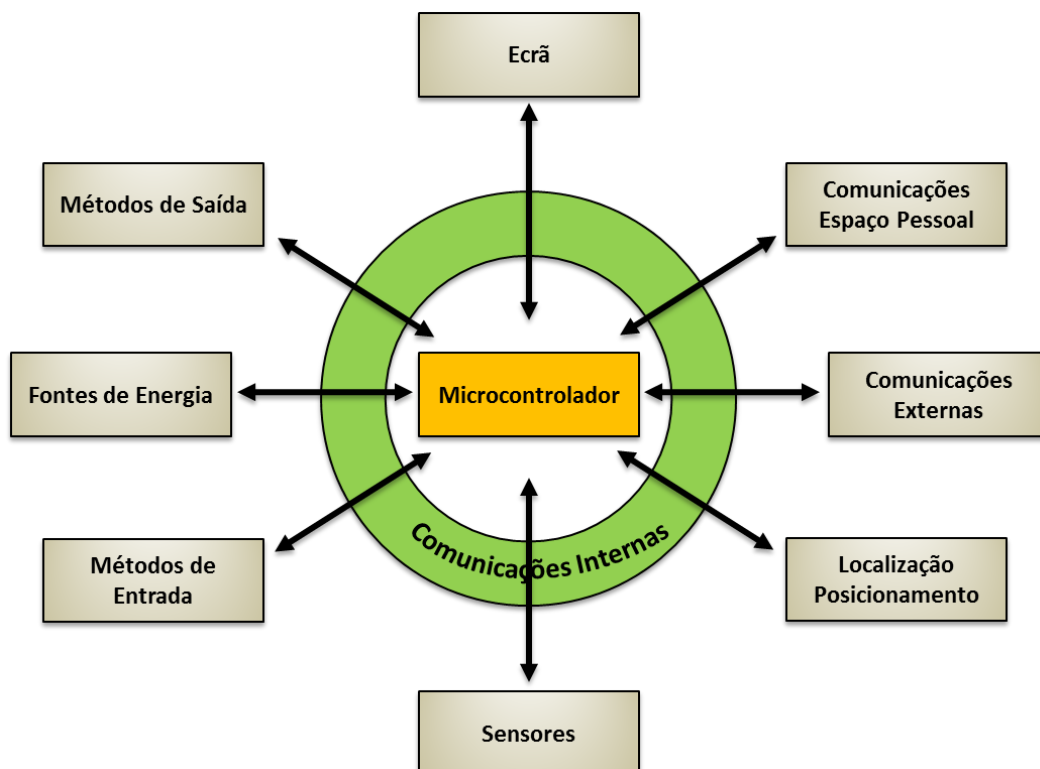


Figura 27 Arquitetura geral do *hardware* elétrico

6.1. MÉTODOS DE ENTRADA

Os dispositivos de entrada convencionais como o rato e o teclado, não são adequados para dispositivos móveis, tendo em conta que a sua utilização necessita de espaço e são em si, muito grandes para serem utilizados e transportados durante o dia-a-dia. Portanto, em aplicações *wearable* são tipicamente utilizados métodos de entrada alternativos. Estes incluem botões simples e ecrãs táteis além dos métodos indiretos como o reconhecimento de gestos e comandos de voz entre outros relacionados com o corpo. No entanto, o uso destes novos métodos requer por vezes treino específico, já que os utilizadores estão mais familiarizados com o teclado e rato presentes no computador.



Figura 28 Diferentes métodos de entrada encontrados no quotidiano

Além dos métodos mais comuns que requerem interação física como os botões ou ecrãs táteis, os métodos baseados nos recursos disponibilizados pelo corpo humano são ainda uma novidade que requer adaptação por parte dos utilizadores, no entanto mostram ser um grande avanço no âmbito dos dispositivos *wearable* muito devido á facilidade de interação com o equipamento mantendo ambas as mãos livres. Exemplos destes métodos são a fala, o movimento dos olhos, expressões faciais, emoções e os próprios movimentos corporais. A entrada através de fala é considerada um método de entrada eficaz apesar da dificuldade de perceção genérica, uma vez que não tem o problema de escalabilidade que quase todos os dispositivos de entrada física têm e é um método natural para as pessoas [17].

Apesar de todos estes novos métodos de entrada os dispositivos *wearable* caminham no sentido de realizar algumas ações baseadas no contexto em que o utilizador se encontra, evitando assim qualquer tipo de ação por parte do mesmo.

Durante este trabalho vão ser focados os métodos baseados em interação física e movimento corporal.

6.2. MÉTODOS DE SAÍDA

Os ecrãs são o principal dispositivo de saída dos computadores e em aplicações *wearable* estes desempenham também um papel importante. As pessoas estão familiarizadas com a utilização de ecrãs no dia-a-dia logo este método de saída é natural também em aplicações *wearable*. A aplicação e os dados a serem apresentados determinam o tipo e as características do ecrã a utilizar (cores ou monocromático, tamanho, etc). Os ecrãs podem ser utilizados de uma forma bastante intuitiva, no entanto, não são adequados para todas as tarefas e são também necessários métodos de saída alternativos e mais diretos.



Figura 29 Diferentes métodos de saída encontrados no quotidiano

Além dos métodos de saída para informações complexas como os ecrãs os métodos de saída para notificações simples por exemplo podem passar pela utilização de sinais luminosos (LED's), sinais sonoros (colunas de som) ou mesmo através de sinais vibratórios (vibrador).

Durante este trabalho vão ser focados os métodos baseados em ecrã gráfico, sinais luminosos, vibratórios e sonoros.

6.3. LOCALIZAÇÃO/POSICIONAMENTO

Os métodos de posicionamento e localização são tecnologias bastante úteis em dispositivos *wearable* tendo em conta as possíveis necessidades de navegação e orientação. As técnicas de posicionamento podem ser categorizadas de acordo com a precisão, fiabilidade, disponibilidade, latência e adequação para a aplicação em causa [18]. A última categoria inclui o consumo de energia, o tamanho e o peso, bem como a dependência em relação às infraestruturas disponíveis no ambiente. Outra possibilidade para a classificação dos métodos de posicionamento baseia-se nas técnicas de localização. Três das principais técnicas são a triangulação, a análise do contexto e a proximidade [19].

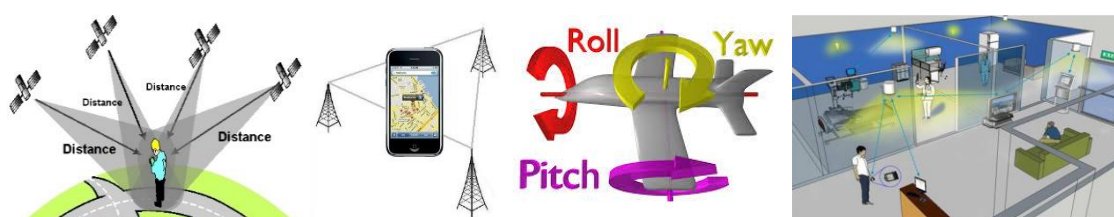


Figura 30 Diferentes métodos de localização/posicionamento

Para a localização/posicionamento exterior as técnicas baseadas em satélite são tradicionalmente utilizadas devido à sua disponibilidade global, recorrendo a serviços como o GPS e o GLONASS. No entanto este método não é adequado para a localização em interiores e por vezes também em algumas zonas exteriores devido à sua fraca intensidade de sinal, como por exemplo em áreas urbanas com grandes prédios que tendem a bloquear o sinal entre o satélite e o recetor. Portanto além da localização para exteriores baseada em satélites, pode ser utilizado o princípio baseado na proximidade das células da rede móvel, a partir das quais pode ser estimada uma localização.

Em relação à localização em interiores os principais métodos utilizados envolvem soluções de RF, utilizando redes locais sem fios (WLANs) ou Bluetooth para deteção da localização. O posicionamento inercial que recorre à utilização de sensores de aceleração e giroscópios, também é adequado para aplicações de posicionamento e de navegação.

6.4. COMUNICAÇÕES

As comunicações nos dispositivos *wearable* são de veras importantes, em primeiro lugar para interligar componentes distribuídos numa área próxima do utilizador (zona pessoal), como por exemplo uma rede de sensores específicos espalhados por todo o corpo. Este tipo de comunicação é considerado interno na medida em que tem lugar dentro da zona de ação do utilizador. Em segundo lugar, a comunicação externa é necessária para a transferência de dados entre o dispositivo *wearable* e outros equipamentos e redes de informação externas. Estas camadas de comunicação são ilustradas na Figura 31.

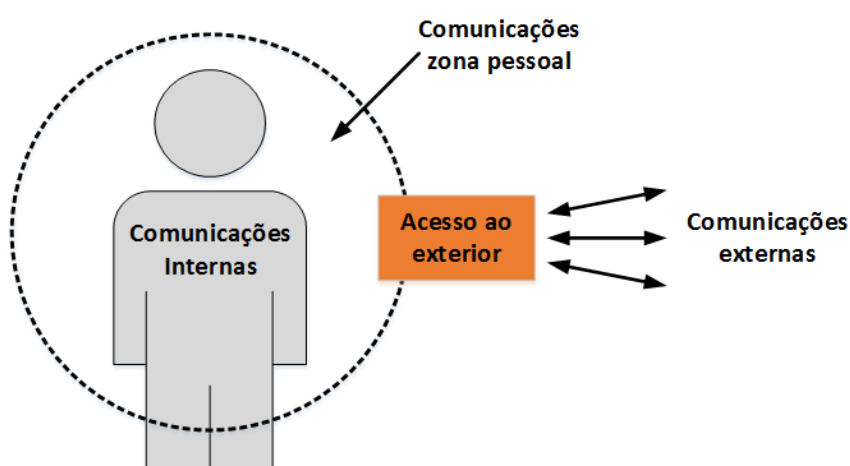


Figura 31 Zonas de comunicação em aplicações *wearable*

Neste modelo de comunicação em existe geralmente um ponto de acesso para permitir comunicações externas. Este ponto de acesso pode, por exemplo, ser uma interface de rede de dados móveis (através da ligação a um telemóvel), ou então diretamente a um ponto de rede sem fios.

Para a execução da comunicação estão disponíveis várias técnicas. As técnicas mais adequadas são seleccionadas tendo em conta as suas necessidades de comunicação, que são determinadas pelo tipo de dados, as taxas de transferência, periodicidade, fiabilidade, segurança e custo, bem como o consumo de energia.

6.5. FONTES DE ENERGIA

As fontes de energia são deveras importantes em dispositivos *wearable* tendo em conta o importante objetivo da longa duração entre recarregamentos. Além de tecnologias primárias com base em baterias, métodos alternativos de captação de energia são utilizados como por exemplo as células solares. Um dos grandes desafios existentes é razão implícita entre a capacidade energética versus o tamanho necessário. Tendo em conta este desafio, além das evoluções ligadas ao armazenamento da energia, muitas melhorias foram feitas de forma a otimizar o consumo de diversos componentes tanto a nível de *hardware* como software, entre os quais as unidades de processamento e as unidades de comunicação.

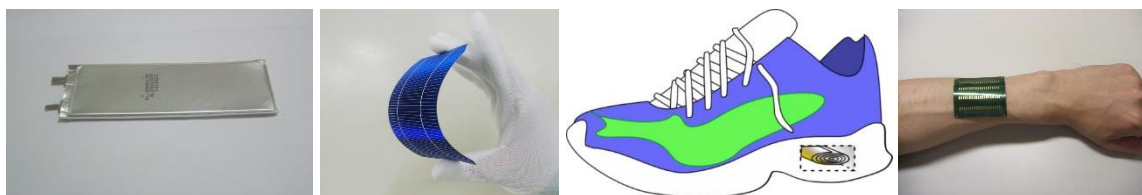


Figura 32 Diferentes fontes de energia

Em ambientes móveis qualquer peso adicional pode prejudicar o conforto de utilização. Normalmente a solução mais comum para aplicações *wearable* é a utilização de baterias de polímero de lítio, que possibilitam a construção de baterias flexíveis capazes de encaixar naturalmente na estrutura do dispositivo. Outras fontes de energia podem ser obtidas a partir do utilizador ou do ambiente. Por exemplo a utilização de materiais piezoelétricos para a geração de energia através de movimentos do utilizador (como os passos).

6.6. DESCRIÇÃO DOS COMPONENTES

Nesta secção vão ser apresentados os componentes mais preponderantes para o sistema. De notar que todos os componentes foram selecionados tendo em conta o seu consumo energético e o seu tamanho.

6.6.1. CONECTIVIDADE MÚLTIPLA

Uma vez que se trata de um dispositivo *wearable*, ter a capacidade de interagir com outros equipamentos é uma mais-valia. Para responder a essa necessidade foi escolhido o Broadcom BCM43341, pois é um integrado que suporta comunicações NFC, IEEE 802.11 a/b/g/n, Bluetooth 4.0, e rádio FM num único componente.

Algumas das suas principais características são:

- Bluetooth 4.0 + EDR
- Amplificador de potência integrado para “Classe 1”
- Potência de saída programável
- NFC com colheita de energia para transações mesmo sem energia do dispositivo
- Tensão de alimentação de até 4.8V
- Muito baixo consumo
- WLPGA package

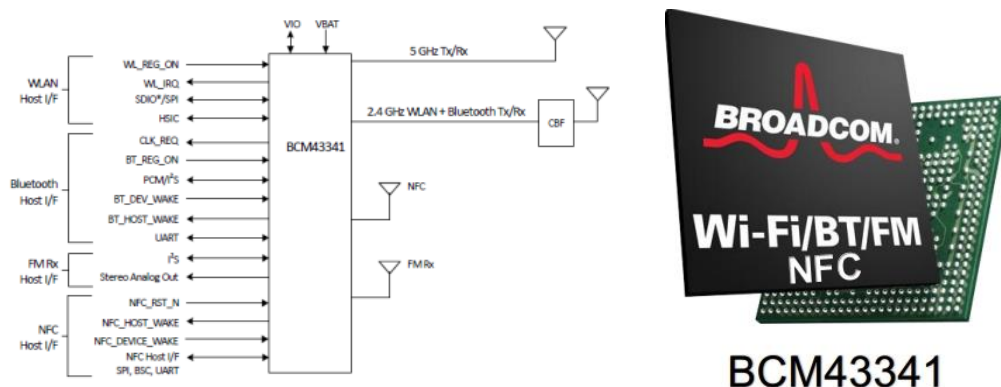


Figura 33 Broadcom BCM43341

6.6.2. ACELERÓMETRO

Atualmente existe uma vasta gama deste tipo de equipamentos, desde os analógicos aos digitais. Tendo em conta a necessidade do acelerómetro ser de três eixos, foram comparadas as características ao nível do consumo energético, tamanho e funcionalidade de vários de modo a escolher o que melhor se enquadrava. As principais características do escolhido (LIS3DSH) são:

- Ampla gama de alimentação, desde 1.7V a 3.7V
- Sensor de temperatura embebido
- LGA-16 package (3mm x 3 mm)
- Máquinas de estado programáveis. Permitem fazer a deteção de algumas situações como quedas, acordar de um estado parado, contar passos, entre outros.

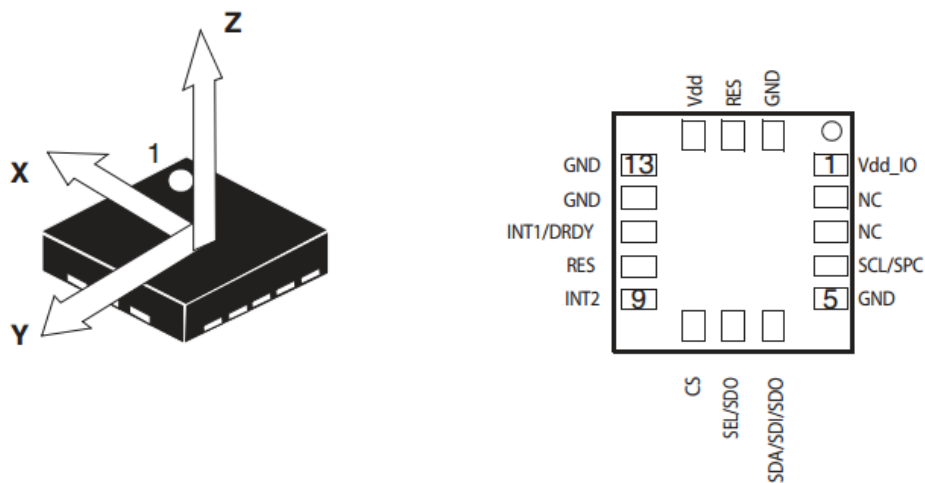


Figura 34 Acelerómetro LIS3DSH

6.6.3. MICROPROCESSADOR

O microprocessador é um componente crítico do sistema, uma vez que é necessário ser potente o suficiente para tornar a usabilidade o mais fluida possível, e ao mesmo tempo ter um baixo consumo de energia. O escolhido foi o Energy Micro EFM32GG990 que conta com as seguintes principais características:

- 32bit ARM Cortex-M3 @ até 48MHz
- Consumo de energia de:
 - 20nA @ 3V *Shutoff Mode*
 - 0.4uA @ 3V *Shutoff Mode* com RTC
 - 200 µA/MHz @ 3 V *Run Mode*
- 1024KB Flash
- 128KB RAM
- USB 2.0
- Package BGA 112, com a dimensão de 10x10 mm

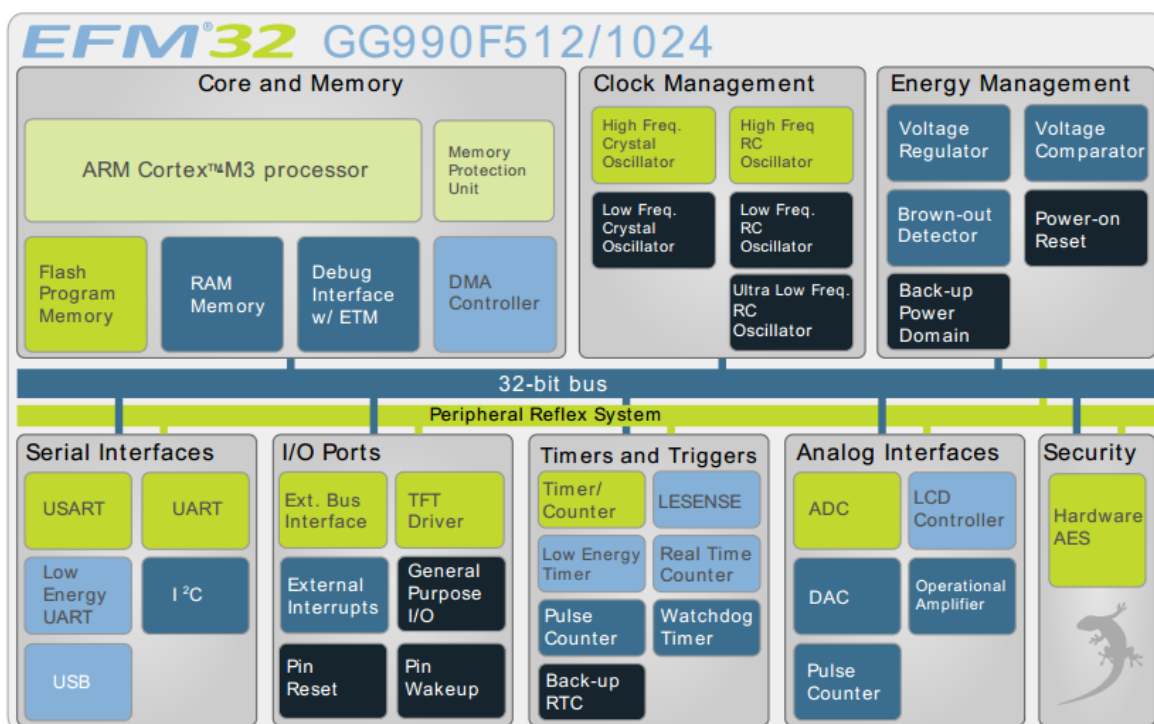


Figura 35 Diagrama de blocos do microprocessador

6.6.4. ECRÃ

A escolha do ecrã teve em conta vários fatores, como o tamanho reduzido para o género de dispositivo, o baixo consumo de energia e a capacidade de poder visualizar o seu conteúdo em diferentes ambientes. Para tal foi escolhido o Sharp LS013B7DH03, que possui as seguintes características:

- Utilizável tanto em ambientes interiores como exteriores.
- Monocromático (modo transmissivo melhorado com luz de fundo)
- Tamanho do ecrã de 1.28” de forma a se adaptar facilmente ao pulso.



Figura 36 Aspeto do ecrã

6.6.5. BATERIA

Devido às restrições de espaço a bateria escolhida é de polímero de lítio, uma vez que estas têm uma grande densidade de potência, o que permite ter uma vida útil maior mantendo um tamanho reduzido.

As principais características são:

- Auto descarregamento muito baixo, comparativamente as baterias de níquel.
- Não é necessário fazer descargas periódicas porque a bateria não tem memória.
- Mais caras aproximadamente 40% em relação as baterias de níquel.



Figura 37 Bateria de polímeros de lítio

6.6.6. SENSOR DE TEMPERATURA

O sensor de temperatura para um equipamento *wearable* deve ser o mais pequeno possível. Para tal, o sensor selecionado (TMP006) tem um formato bastante reduzido, 1,7mm por 1,3mm, e mede a temperatura através da radiação infravermelha emitida pelos objetos. Através deste método o sensor pode ser colocado dentro do equipamento e evitar a necessidade de existirem aberturas no involucrio do dispositivo.

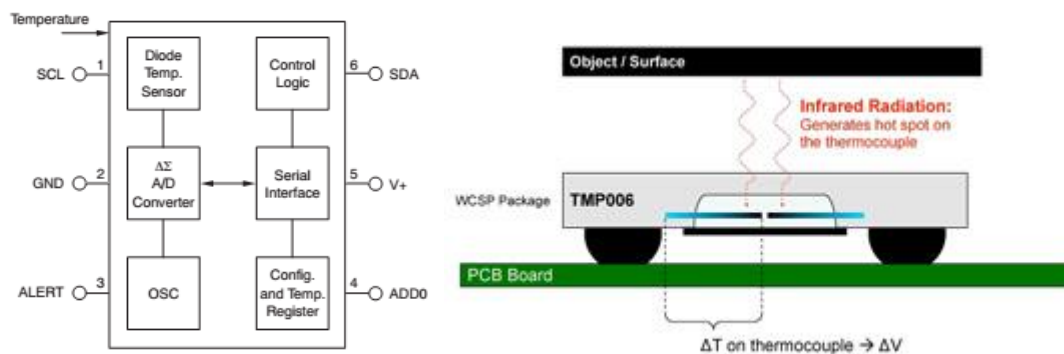


Figura 38 Sensor de temperatura

6.6.7. SENSOR DE HUMIDADE

O sensor de humidade Si7005 foi escolhido tendo em conta o formato mais pequeno possível. Este tem uma membrana hidrofóbica para proteção, que evita possíveis danos em ambientes mais húmidos. O sensor tem as seguintes características:

- Gama de operação de 0 a 100% de humidade relativa
- Precisão de 0,5%
- *Package* QFN 4x4mm



Figura 39 Sensor de humidade Si7005

6.6.8. ÍNDICE UV

O sensor de índice UV ML8511 consegue captar radiação com comprimento de onda até 365nm, o que equivale a região de raios UV-A (315nm a 400nm) e UV-B (280nm a 315nm). A caixa tem a dimensão de 4,0mm x 3,7mm x 0.73mm.

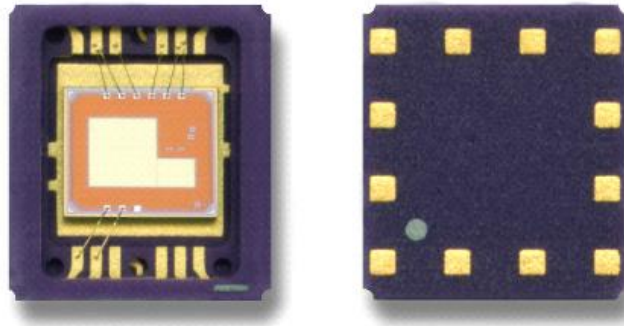


Figura 40 Sensor do índice UV ML8511

Conforme o aumento da incidência de raios UV, o valor de tensão é proporcional à intensidade da luz como apresentado na Figura 41.

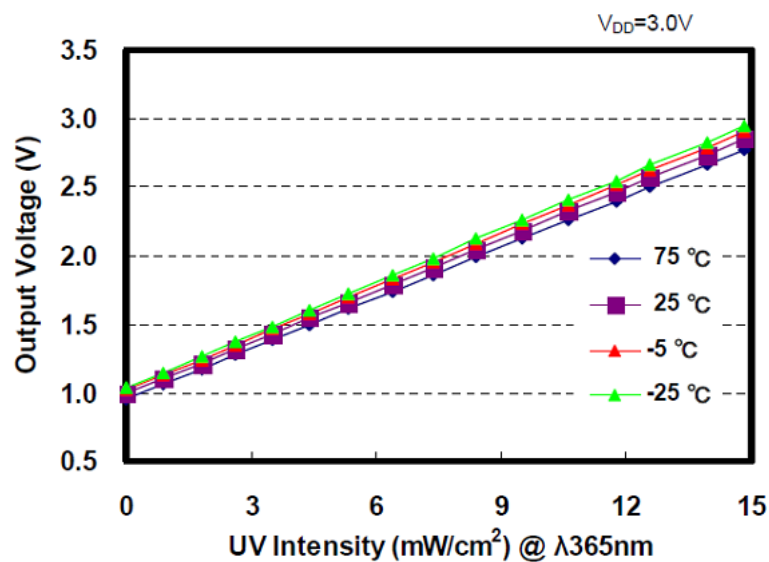


Figura 41 Relação da tensão com a radiação UV do sensor ML851

7. ARQUITETURA DE SOFTWARE

No decorrer deste capítulo as figuras/esquemas utilizam essencialmente o inglês, tendo em conta a similaridade com a linguagem de software.

A arquitetura de *software* segue uma abordagem orientada a notificações e eventos, recorrendo ainda frequentemente ao esquema de *publisher/subscriber*, um conhecido paradigma para o desenvolvimento de sistemas distribuídos e caracterizado pelo desacoplamento das partes e pela comunicação assíncrona entre componentes.

A Figura 42 seguinte mostra os diversos componentes da arquitetura de *software* e a sua distribuição pelos diversos grupos existentes (*Device Drivers*, Sistema Operativo, *Middleware* e Aplicações).

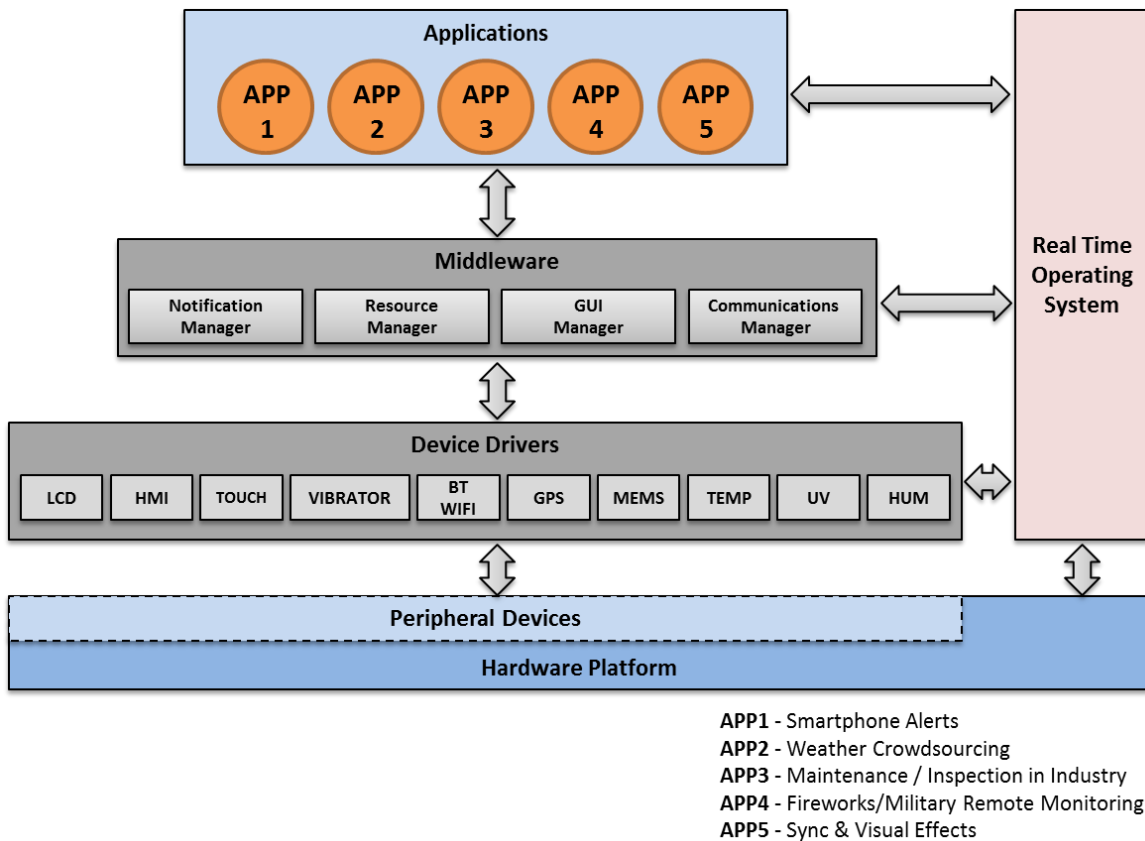


Figura 42 Arquitetura geral de *software*

7.1. DEVICE DRIVERS

Esta camada é constituída por um conjunto de drivers (código) que simplificam a programação das camadas superiores, agindo como um tradutor entre o componente de *hardware*, as aplicações e/ou o sistema operativo. Desta forma o código necessário para as camadas de nível superior deve ser feito independentemente de qualquer *hardware* específico. Por exemplo, um componente de comunicação sem fios WIFI necessita de lidar com protocolos de comunicação convencionais, tais como HTTP que é comum a todos os equipamentos que falam esta língua. No entanto, a camada física precisa de comunicar com o componente específico WIFI. Esta camada aborda estas variações específicas dos componentes periféricos.

7.1.1. LCD

A forma como a matriz de pixéis deste ecrã é utilizada para armazenar os dados da imagem deriva do seu nome (Sharp's Memory LCD). Cada *pixel* da matriz corresponde a 1 bit (apenas para escrita), correspondendo no total a um único buffer de gravação para uma imagem completa. Este recurso alivia o processador e barramento da sobrecarga de dados contínuos transferidos para atualizar a imagem, tendo em conta que a imagem só precisa de ser escrita uma vez (a imagem é retida indefinidamente). Quando é necessário atualizar a imagem exibida, não há necessidade de limpar e reescrever a imagem inteira, basta apenas reescrever as linhas que se pretende alterar.

Os dados são enviados para o ecrã no formato *little-endian*. O tamanho da trama de dados por linha a escrever depende da resolução do ecrã. Neste caso específico a resolução do ecrã é de 128x128 o que representa um tamanho de 128 bits.

Inicialização dos recursos necessários para interagir com o ecrã:

```
EMSTATUS MEMLCD_Init(MEMLCD_Config *cfg)
{
    /* Setup clocks */
    CMU_ClockEnable(cmuClock_GPIO, true );
    CMU_ClockEnable(cfg->timerClock, true );
    CMU_ClockEnable(cfg->usartClock, true );
    /* Setup GPIO's */
    GPIO_PinModeSet(cfg->sclk.port, cfg->sclk.pin, gpioModePushPull, 0 );
    GPIO_PinModeSet(cfg->si.port,  cfg->si.pin,   gpioModePushPull, 0 );
    GPIO_PinModeSet(cfg->scs.port,  cfg->scs.pin,  gpioModePushPull, 0 );
    GPIO_PinModeSet(cfg->extcomin.port, cfg->extcomin.pin, gpioModePushPull, 0 );
    GPIO_PinModeSet(cfg->extmode.port,  cfg->extmode.pin,  gpioModePushPull, 0 );
    GPIO_PinModeSet(cfg->disp.port,    cfg->disp.pin,    gpioModePushPull, 0 );
}
```

Existem três comandos que podem ser enviados para o ecrã que foram implementados no driver:

- Apagar o ecrã (*Clear Screen*)
- Escrever uma linha (*Write Line*)
- Escrever múltiplas linhas (*Write Multiple Lines*)

Comando: Apagar Ecrã

Este comando limpa todo o ecrã, escrevendo 0 em todas as posições da memória. A estrutura de comando a tem o formato apresentado na figura seguinte.

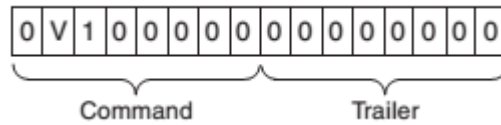


Figura 43 Estrutura de comando para apagar o ecrã

Implementação do comando apagar ecrã:

```
void MEMLCD_Clear( void )
{
    uint8_t cmd;
    /* Set SCS */
    GPIO_PinOutSet( pConfig->scs.port, pConfig->scs.pin );
    /* SCS setup time: min 6us */
    usDelay(6);
    /* Send command */
    cmd = (MEMLCD_CMD_ALL_CLEAR | comPolarity);
    USART_TxDouble( pConfig->usart, cmd );
    /* Wait for transfer to finish */
    while ( !(pConfig->usart->STATUS & USART_STATUS_TXC) );
    /* Clear SCS */
    GPIO_PinOutClear( pConfig->scs.port, pConfig->scs.pin );
}
```

Comandos: Escrever uma linha e Escrever múltiplas linhas

A estrutura do comando para escrever uma linha é a representada na figura seguinte.

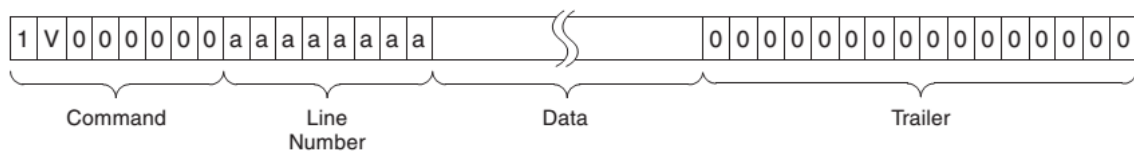


Figura 44 Estrutura de comando para escrever uma linha

O comando para escrever múltiplas linhas inicia-se da mesma forma do anterior utilizando a mesma estrutura inicial. No entanto após os 8 bits de dados devem ser enviados 8 bits de trailer (ao invés de 16bits) e repetir esta sequência para todas as linhas que se queria atualizar. No envio dos dados da última linha deve ser enviado o trailer com 16 bits para indicar o final da transmissão.

Tendo em conta a similaridade dos comandos de escrita, foi criada uma função que tanto permite a escrita de apenas uma linha como de várias. A implementação pode ser vista no código apresentado abaixo:

```
EMSTATUS MEMLCD_Update(uint16_t *data, int firstLine, int lastLine)
{
    int i,j;
    /* Assert SCS */
    GPIO_PinOutSet( pConfig->scs.port, pConfig->scs.pin );
    /* SCS setup time: min 6us */
    usDelay(6);
    /* Send update command and first line address */
    USART_TxDouble(pConfig->usart, MEMLCD_CMD_UPDATE | (firstLine + 1) << 8);
    /* Get start address to draw from */
    uint16_t *p = (uint16_t *)data;
    p += firstLine * 10;
    for ( i=firstLine; i<=lastLine; i++ ) {
        /* Send pixels for this line */
        for ( j=0; j<9; j++ ) {
            USART_TxDouble(pConfig->usart, *p);
            p++;
        }
        /* Skip padding data in frame buffer */
        p += 1;
    }
    /* Wait for USART to finish */
    while (!(pConfig->usart->STATUS & USART_STATUS_TXC)) ;
    /* De-assert SCS */
    GPIO_PinOutClear( pConfig->scs.port, pConfig->scs.pin );
    return MEMLCD_OK;
}
```

7.1.2. TOUCH

A detecção de contacto capacitiva é uma tecnologia baseada na medição de uma alteração na capacidade. O driver desenvolvido utiliza a interface de baixo consumo LESENSE que se baseia nos comparadores analógicos do microcontrolador EFM32 para medir a capacidade entre o pino sensor e a referência. Mais especificamente, é medida a capacidade entre o pino sensor e a referência através de um circuito oscilador RC (este circuito é intrínseco ao microcontrolador). Desta forma a frequência vai mudar em função da capacidade presente no pino sensor, como ilustrado na figura seguinte.

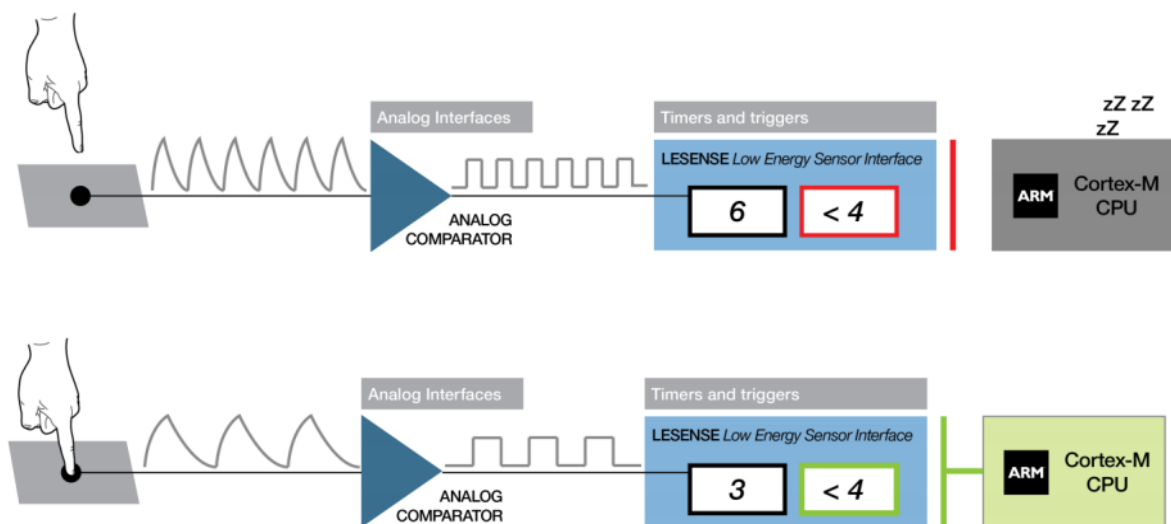


Figura 45 Comportamento do sensor capacitivo [21]

Para uma correta aquisição dos segmentos capacitivos foi necessário implementar a seguinte sequência:

- Configuração dos comparadores analógicos para o modo de toque capacitivo.
- A saída do comparador analógico é encaminhada para o registo de LESENSE que inicia a contagem. O contador do tempo de amostragem é iniciado ao mesmo tempo.
- Quando o contador de tempo de amostragem atinge o valor configurado, o valor de contagem do LESENSE é armazenado e comparado com o valor limite (*threshold*).
- Se a comparação do valor de LESENSE for superior ao valor limite, é despoletada uma interrupção, e posteriormente processada a informação relevante (qual o valor presente em cada segmento). De notar que o microcontrolador nunca é acordado durante um ciclo de medição, ajudando deste modo na poupança de energia.

Tendo em conta as características intrínsecas do microcontrolador, o circuito RC apenas precisa de ser associado aos pinos que se pretendem utilizar para os segmentos capacitivos (neste caso concreto são 4 segmentos) como ilustrado no seguinte bloco de código.

```
void CAPLESENSE_setup(void)
{
    /* Configure ACMP locations, ACMP output to pin disabled. */
    ACMP_GPIOSetup(ACMP1, 0, false, false);
    /* Initialize ACMPs in capacitive sense mode. */
    ACMP_CapsenseInit(ACMP1, &initACMP);
    /* Configure the drive strength of the ports for the light sensor. */
    GPIO_DriveModeSet(CAPLESENSE_SLIDER_PORT0, gpioDriveModeStandard);
    /* Initialize the 4 GPIO pins of the touch slider for using them as LESENSE
     * scan channels for capacitive sensing. */
    GPIO_PinModeSet(CAPLESENSE_SLIDER_PORT0, CAPLESENSE_SLIDER0_PIN,
pioModeDisabled, 0);
    GPIO_PinModeSet(CAPLESENSE_SLIDER_PORT0, CAPLESENSE_SLIDER1_PIN,
gpioModeDisabled, 0);
    GPIO_PinModeSet(CAPLESENSE_SLIDER_PORT0, CAPLESENSE_SLIDER2_PIN,
gpioModeDisabled, 0);
    GPIO_PinModeSet(CAPLESENSE_SLIDER_PORT0, CAPLESENSE_SLIDER3_PIN,
gpioModeDisabled, 0);
}
```

Depois de configurados todos os parâmetros relativos à aquisição capacitiva (pinos e valores limite para comparação), é apenas necessário garantir que na rotina de interrupção (despoletada apenas quando pelo menos um dos segmentos é superior ao limite) são lidos e guardados para processamento todos os valores.

O fluxograma da Figura 46 ilustra a implementação desta rotina.

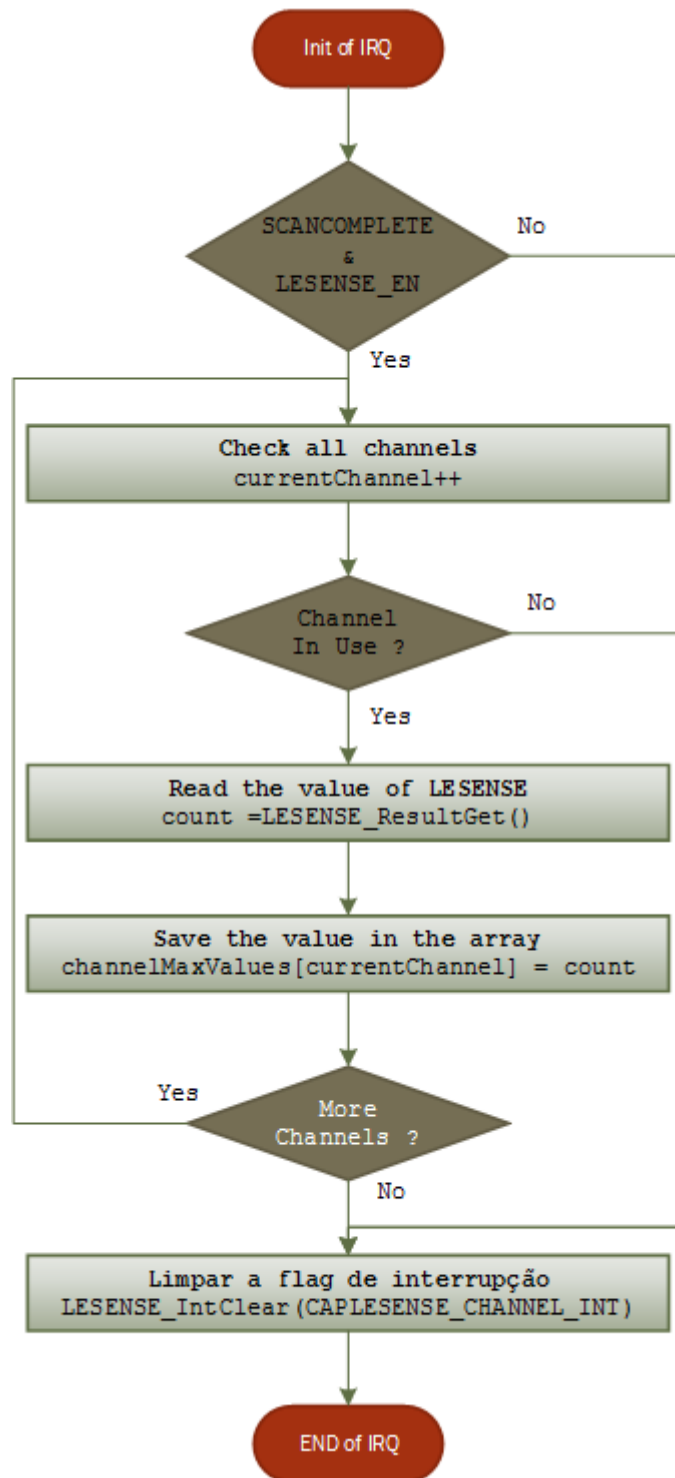


Figura 46 Fluxograma da rotina de interrupção do TOUCH

7.1.3. MEMS

Uma das características mais interessantes do sensor escolhido (LIS3DSH) é a possibilidade da configuração de máquinas de estados independentes da intervenção do microprocessador. Este sensor possui duas máquinas de estados, cada uma com a possibilidade de configuração de 16 estados, tendo como parâmetros:

- 4 Temporizadores independentes
- 2 Mascaras independentes (X, Y, Z, V)
- 3 Limites de aceleração independentes
- Função de detecção de picos

O próprio sensor contém já algumas definições para eventos específicos como:

- Clique (*Toggle*)
- Duplo clique (*Double Tap*)
- Acordar do estado parado (*Wake Up*)

Exemplo da função de acordar do estado parado, que é bastante interessante tendo em conta o objetivo de poupança de energia. Com este evento o microprocessador apenas necessita de realizar as tarefas cíclicas se o dispositivo não estiver parado.

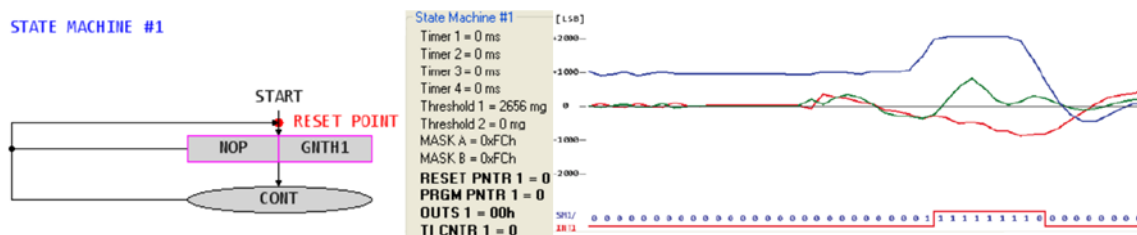


Figura 47 Exemplo da máquina de estados para a função acordar [22]

O fluxograma seguinte ilustra o driver implementado tendo em conta o protocolo específico do LIS3DSH.

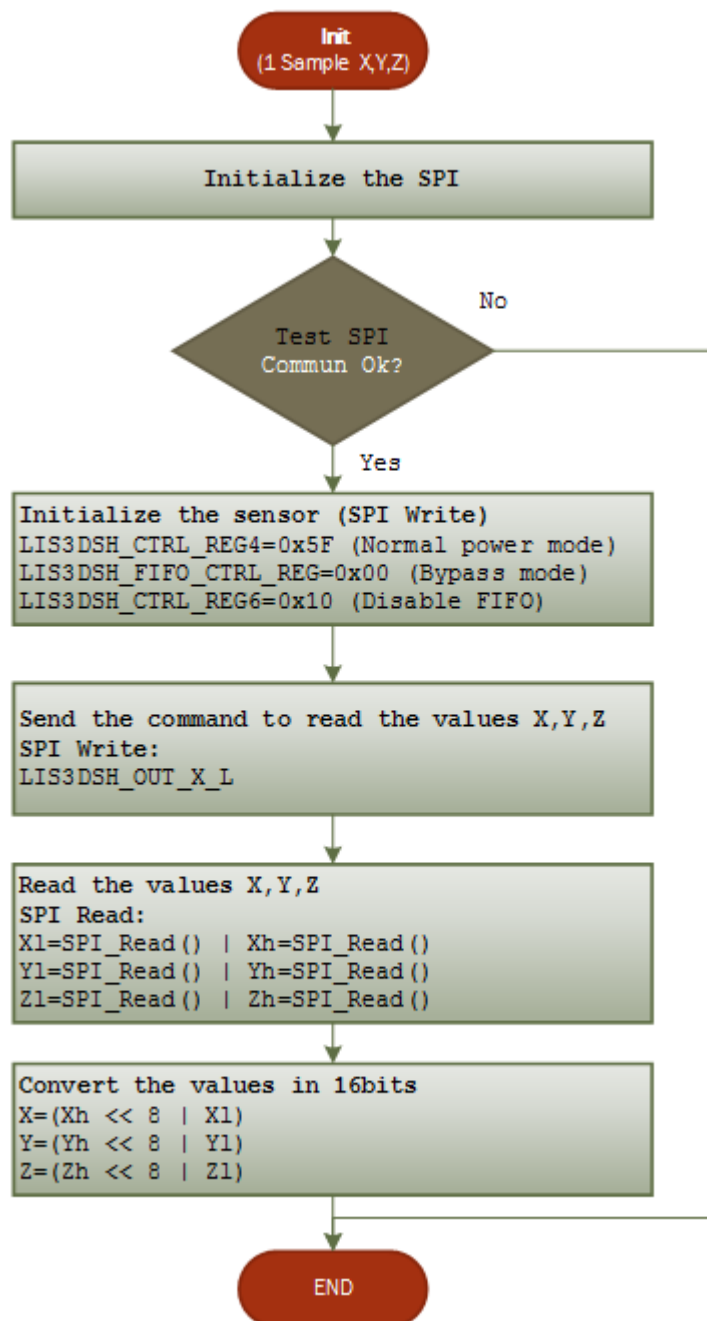


Figura 48 Fluxograma da comunicação com o MEMS

7.1.4. TEMPERATURA

O sensor de temperatura TMP006 é o primeiro de uma série de sensores de temperatura que medem a temperatura de um objeto sem a necessidade de entrar em contato com o mesmo. Este utiliza um sensor para absorver a energia infravermelha emitida a partir do objeto a ser medido.

A implementação do driver está representada no fluxograma da Figura 49. Este driver recorre ao protocolo I2C para comunicar com o sensor e posteriormente calcula tanto a temperatura ambiente como a do objeto.

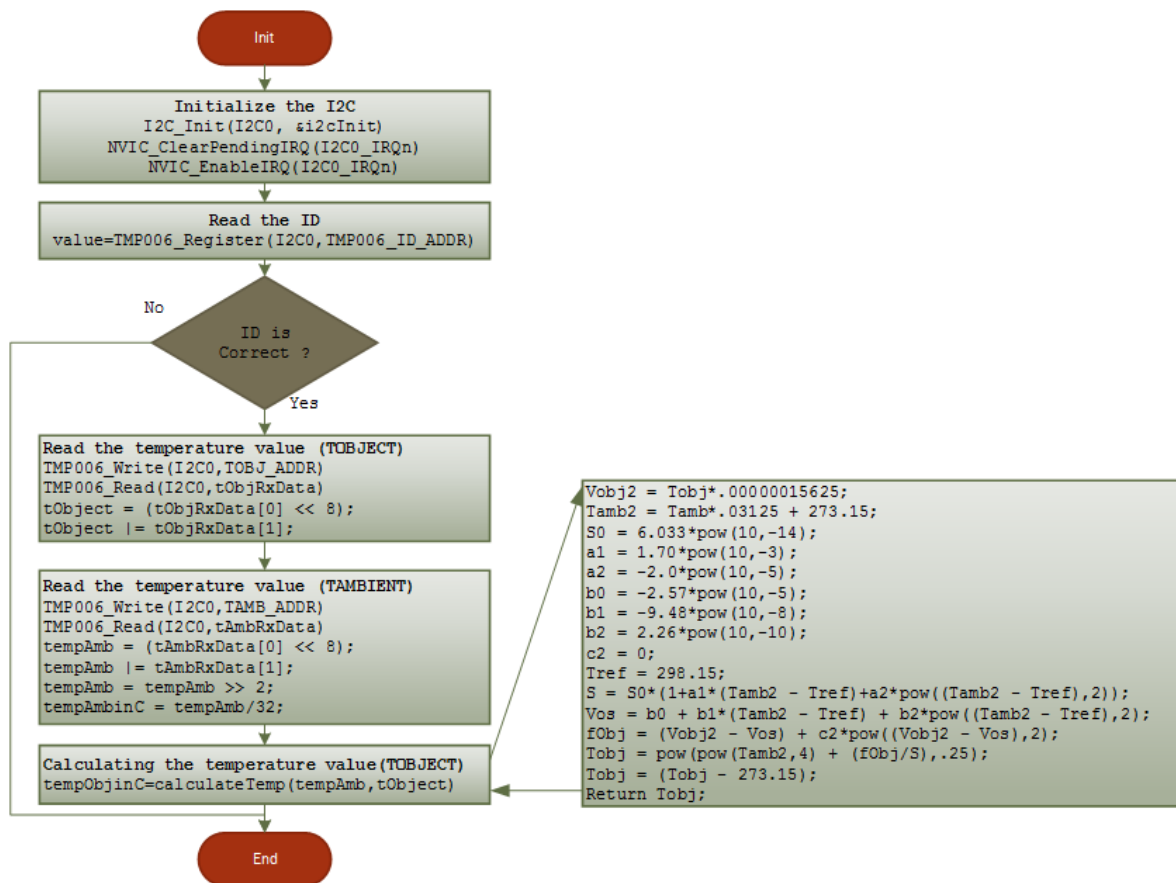


Figura 49 Fluxograma do driver do sensor TMP006

7.1.5. UV

O ML8511 [23] é um sensor de luz adequado para a medição de intensidade da radiação UV tanto em interiores como em exteriores. Este sensor possui internamente um amplificador, que é responsável pela conversão da radiação UV para tensão.

O driver implementado representado pelo fluxograma da Figura 50 utiliza um ADC do microprocessador para ler o valor de tensão presente na saída do sensor e é também responsável pela conversão do valor de tensão na unidade relativa à radiação UV.

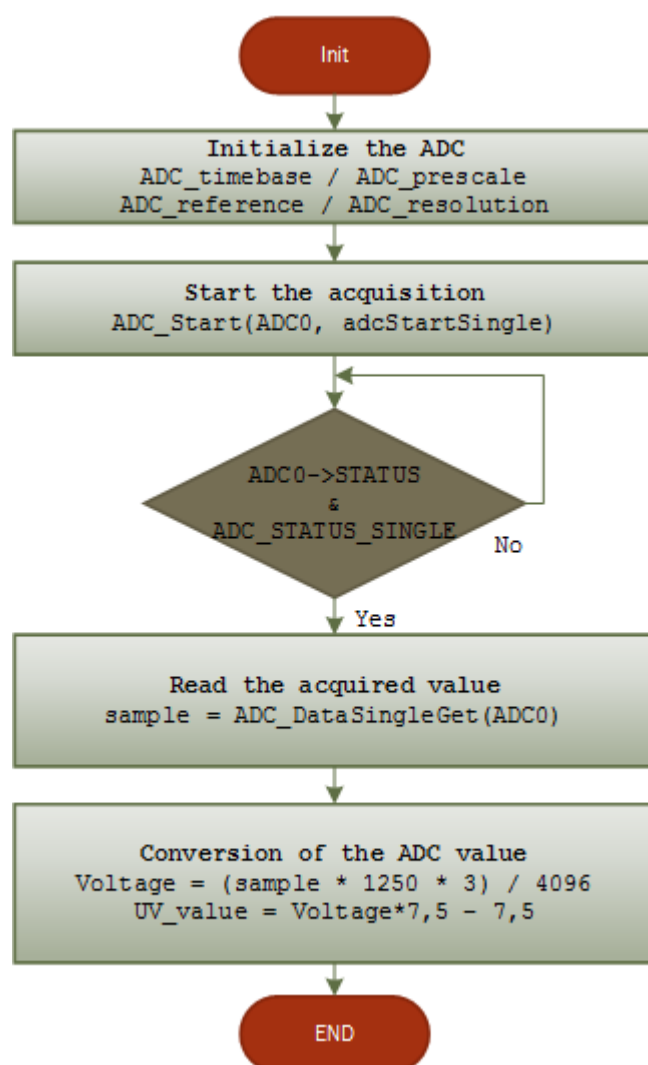


Figura 50 Fluxograma do driver do sensor de UV

7.1.6. HUMIDADE

O Si7005 é um sensor de humidade relativa e um sensor de temperatura no mesmo componente. Ambos os sensores de humidade e de temperatura estão calibrados de fábrica e os dados de calibração são armazenados na memória não volátil do próprio sensor. A comunicação com o sensor é feita através de do protocolo I2C, que permite a ligação de até 128 dispositivos endereçáveis individualmente.

O driver implementado tem como output o valor da humidade (%) e da temperatura (°C). O fluxograma da Figura 51 ilustra a implementação necessária para obter estes dados.

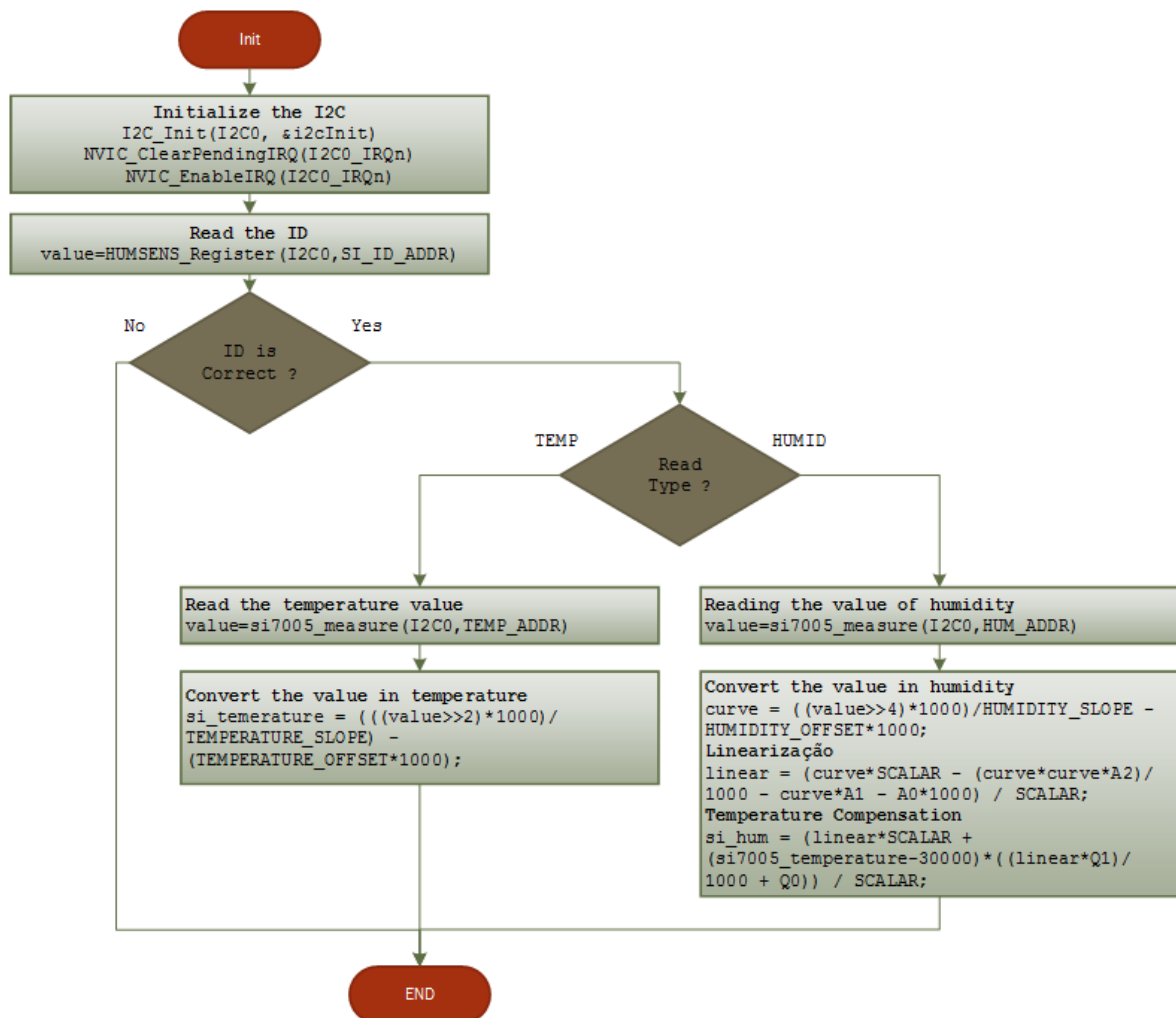


Figura 51 Fluxograma do driver do sensor Si7005

7.2. SISTEMA OPERATIVO

Esta camada é onde estão contidas todas as rotinas e funcionalidades pertencentes ao funcionamento do sistema operativo.

O RTX implementa a interface genérica de RTOS para processadores Cortex-M através da API CMSIS-RTOS. Esta API disponibiliza uma interface uniforme para qualquer componente de *software* que necessite de implementar funcionalidades de RTOS.

Cada aplicação que é criada utilizando o RTX tem de incluir o ficheiro “cmsis-os.h” que contém a API de interface ao RTX CMSIS-RTOS. A configuração do sistema operativo e de algumas das suas funcionalidades encontram-se no ficheiro “RTX_Conf_CM.c”.

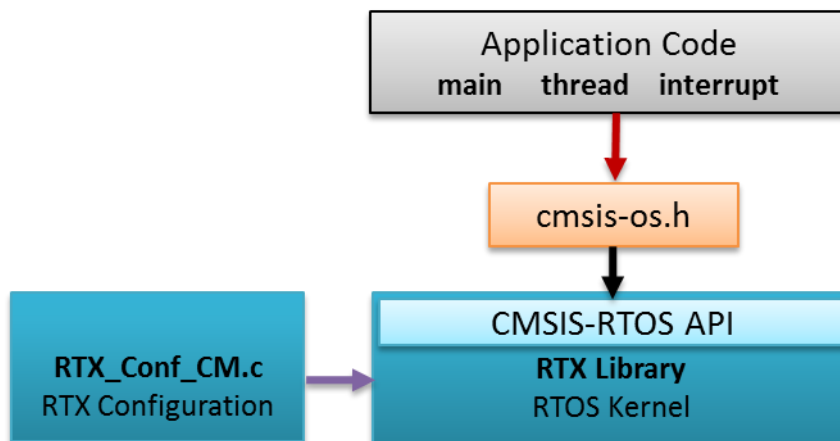


Figura 52 Estrutura do CMSIS-RTOS RTX

O RTX disponibiliza um conjunto de componentes de *software* que podem ser utilizados para a criação das várias aplicações constituintes do sistema. Este tipo de componentes está dividido nos seguintes grupos:

- *Thread*
- *Mutex*
- *Semaphore*
- *Message queue*
- *Memory pool*
- *Mail Queue*
- *Timer*

7.2.1. THREAD

Este conjunto de funções permite criar, definir e controlar funções de *thread*. Um exemplo de uma *thread*, é a função *main* que é iniciada quando o sistema é iniciado.

Uma *thread* pode ter os seguintes estados:

- **Running** – A *thread* está ativa e está a correr, sendo que apenas uma *thread* pode estar ativa de cada vez.
- **Ready** – Após uma *thread* ter terminado a sua execução ou passar para o estado *Waiting*, a *thread* que tem maior prioridade que se encontra no estado *Ready* passa a ficar ativa.
- **Waiting** – As *threads* que se encontram neste estado são *threads* que estão à espera que ocorra um evento.
- **Inactive** – *Threads* que ainda não foram criadas ou que foram terminadas, passam para este estado.

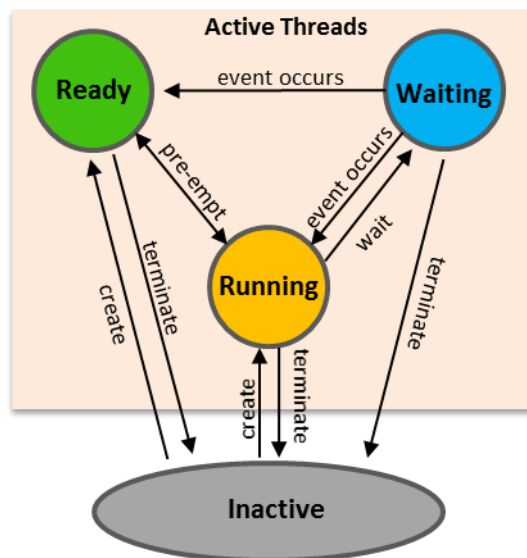


Figura 53 Estados das *threads*

Exemplo da utilização de *threads*:

```
#include "cmsis_os.h"

DigitalOut led1(LED1);
DigitalOut led2(LED2);

void led2_thread(void const *args) {
    while (true) {
        led2 = !led2;
        osDelay(1000);
    }
}
```

```

}
osThreadDef(led2_thread, osPriorityNormal, DEFAULT_STACK_SIZE);

int main() {
    osThreadCreate(osThread(led2_thread), NULL);

    while (true) {
        led1 = !led1;
        osDelay(500);
    }
}

```

7.2.2. MUTEX

As funções do grupo *Mutex* são utilizadas para sincronizar a execução de *threads*. Podem ser utilizadas para a proteção do acesso a recursos compartilhados entre duas *threads*.

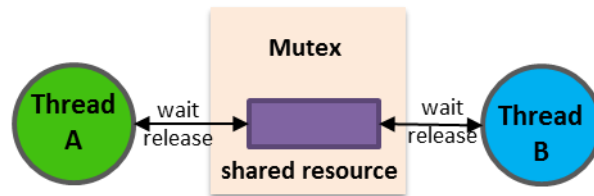


Figura 54 Sequência de operação de um *Mutex*

Exemplo da utilização de *mutex*:

```

#include "cmsis_os.h"

osMutexId stdio_mutex;
osMutexDef(stdio_mutex);

void notify(const char* name, int state) {
    osMutexWait(stdio_mutex, osWaitForever);
    printf("%s: %d\n\r", name, state);
    osMutexRelease(stdio_mutex);
}

void test_thread(void const *args) {
    while (true) {
        notify((const char*)args, 0); osDelay(1000);
        notify((const char*)args, 1); osDelay(1000);
    }
}

void t2(void const *argument) {test_thread("Th 2");}
osThreadDef(t2, osPriorityNormal, DEFAULT_STACK_SIZE);

void t3(void const *argument) {test_thread("Th 3");}
osThreadDef(t3, osPriorityNormal, DEFAULT_STACK_SIZE);

int main() {
    stdio_mutex = osMutexCreate(osMutex(stdio_mutex));

    osThreadCreate(osThread(t2), NULL);
    osThreadCreate(osThread(t3), NULL);
    test_thread((void *)"Th 1");
}

```

7.2.3. SEMAPHORE

O grupo de funções relacionadas com os semáforos é utilizado para a gestão e proteção de recursos partilhados. Um exemplo da sua utilização é a gestão de acesso a um grupo de periféricos, em que o nº de recursos disponíveis é especificado por um parâmetro na criação dos semáforos.

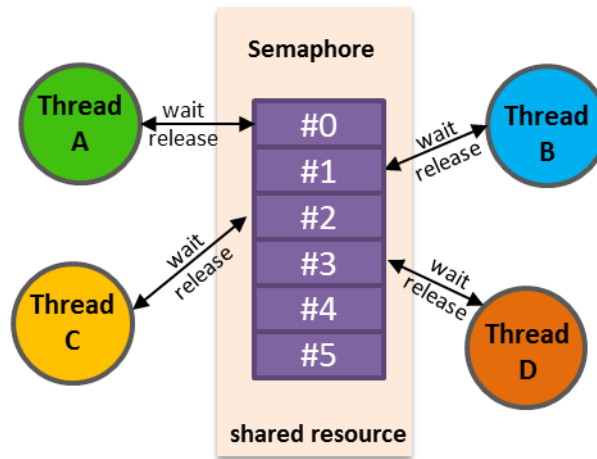


Figura 55 Operações dos semáforos

Exemplo da utilização de *semaphores*:

```
#include "cmsis_os.h"

osSemaphoreId two_slots;
osSemaphoreDef(two_slots);

void test_thread(void const *name) {
    while (true) {
        osSemaphoreWait(two_slots, osWaitForever);
        printf("%s\n\r", (const char*)name);
        osDelay(1000);
        osSemaphoreRelease(two_slots);
    }
}

void t2(void const *argument) {test_thread("Th 2");}
osThreadDef(t2, osPriorityNormal, DEFAULT_STACK_SIZE);

void t3(void const *argument) {test_thread("Th 3");}
osThreadDef(t3, osPriorityNormal, DEFAULT_STACK_SIZE);

int main (void) {
    two_slots = osSemaphoreCreate(osSemaphore(two_slots), 2);

    osThreadCreate(osThread(t2), NULL);
    osThreadCreate(osThread(t3), NULL);

    test_thread((void *)"Th 1");
}
```

7.2.4. MESSAGE QUEUE

As funções incluídas no grupo de *Message Queue* permitem criar, controlar, enviar, receber ou esperar por mensagens.

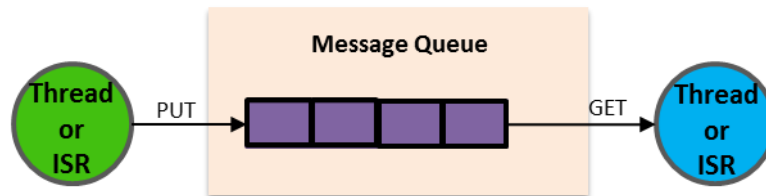


Figura 56 Estrutura de uma *Message Queue*

7.2.5. MEMORY POOL

O *memory pool* são partições de memória com um tamanho fixo. Ao contrário da memória dinâmica, a memória é sempre fixa trazendo benefícios em termos de tempos de execução, pois sabemos o tamanho de todos os blocos de memória.

7.2.6. MAIL QUEUE

O *Mail* (correio) é um bloco de memória que é enviado para uma *thread* ou rotina de interrupção. Este grupo de funções permite criar, controlar, enviar e receber correio.

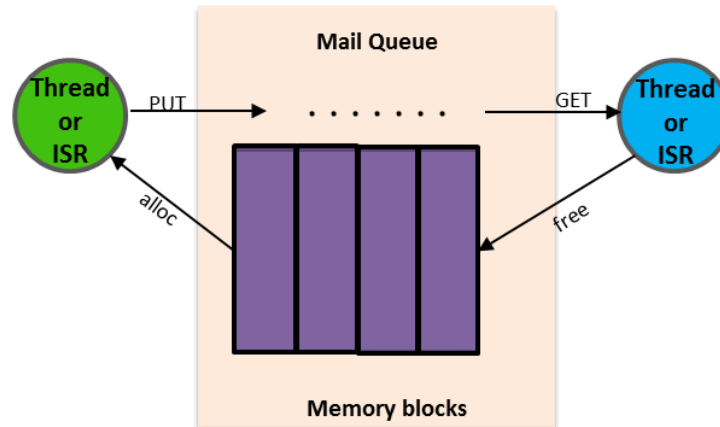


Figura 57 Estrutura de um *Mail Queue*

Exemplo da utilização de uma *Mail Queue*:

```
#include "cmsis_os.h"

typedef struct {
    float    voltage; /* AD result of measured voltage */
    float    current; /* AD result of measured current */
    uint32_t counter; /* A counter value */
} mail_t;
osMailQDef(mail_box, 16, mail_t);
osMailQId mail_box;
void send_thread (void const *args) {
    uint32_t i = 0;
    while (true) {
        i++; // fake data update
        mail_t *mail = (mail_t*)osMailAlloc(mail_box, osWaitForever);
        mail->voltage = (i * 0.1) * 33;
        mail->current = (i * 0.1) * 11;
        mail->counter = i;
        osMailPut(mail_box, mail);
        osDelay(1000);
    }
}
osThreadDef(send_thread, osPriorityNormal, DEFAULT_STACK_SIZE);
int main (void) {
    mail_box = osMailCreate(osMailQ(mail_box), NULL);
    osThreadCreate(osThread(send_thread), NULL);
    while (true) {
        osEvent evt = osMailGet(mail_box, osWaitForever);
        if (evt.status == osEventMail) {
            mail_t *mail = (mail_t*)evt.value.p;
            printf("\nVoltage: %.2f V\n\r", mail->voltage);
            printf("Current: %.2f A\n\r", mail->current);
            printf("Number of cycles: %u\n\r", mail->counter);
            osMailFree(mail_box, mail);
        }
    }
}
```

7.2.7. TIMER

A função *timer* é chamada quando um período de tempo expira. O *timer* pode ser iniciado, terminado e parado.

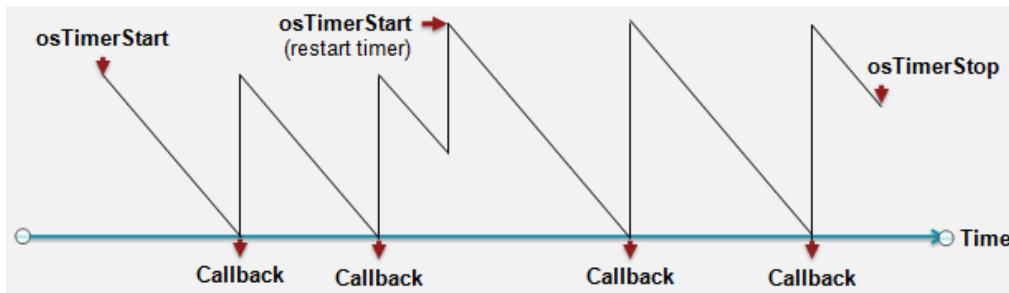


Figura 58 Sequência temporal de uma função *timer*

Exemplo da utilização do *timer*:

```
#include "cmsis_os.h"

DigitalOut LEDs[4] = {
    DigitalOut(LED1), DigitalOut(LED2), DigitalOut(LED3), DigitalOut(LED4)
};

void blink(void const *n) {
    LEDs[(int)n] = !LEDs[(int)n];
}

osTimerDef(blink_0, blink);
osTimerDef(blink_1, blink);
osTimerDef(blink_2, blink);
osTimerDef(blink_3, blink);

int main(void) {
    osTimerId timer_0 = osTimerCreate(osTimer(blink_0), osTimerPeriodic, (void *)0);
    osTimerId timer_1 = osTimerCreate(osTimer(blink_1), osTimerPeriodic, (void *)1);
    osTimerId timer_2 = osTimerCreate(osTimer(blink_2), osTimerPeriodic, (void *)2);
    osTimerId timer_3 = osTimerCreate(osTimer(blink_3), osTimerPeriodic, (void *)3);

    osTimerStart(timer_0, 2000);
    osTimerStart(timer_1, 1000);
    osTimerStart(timer_2, 500);
    osTimerStart(timer_3, 250);

    osDelay(osWaitForever);
}
```

7.3. MIDDLEWARE

A camada de *middleware* é responsável por gerir todos os acessos ao *hardware* (através dos drivers) de forma a evitar concorrência e prevenir uma sobrecarga de acessos a determinado recurso por parte das várias aplicações além de garantir uma interoperabilidade entre as várias camadas de software.

7.3.1. GESTOR DE NOTIFICAÇÕES (*NOTIFICATION MANAGER*)

O gestor de notificações é o agente responsável por fazer o encaminhamento de todos os eventos (receção de uma nova comunicação, evento de um botão, etc.) presentes no sistema, evitando por exemplo a necessidade de as aplicações estarem constantemente a monitorizar os drivers.

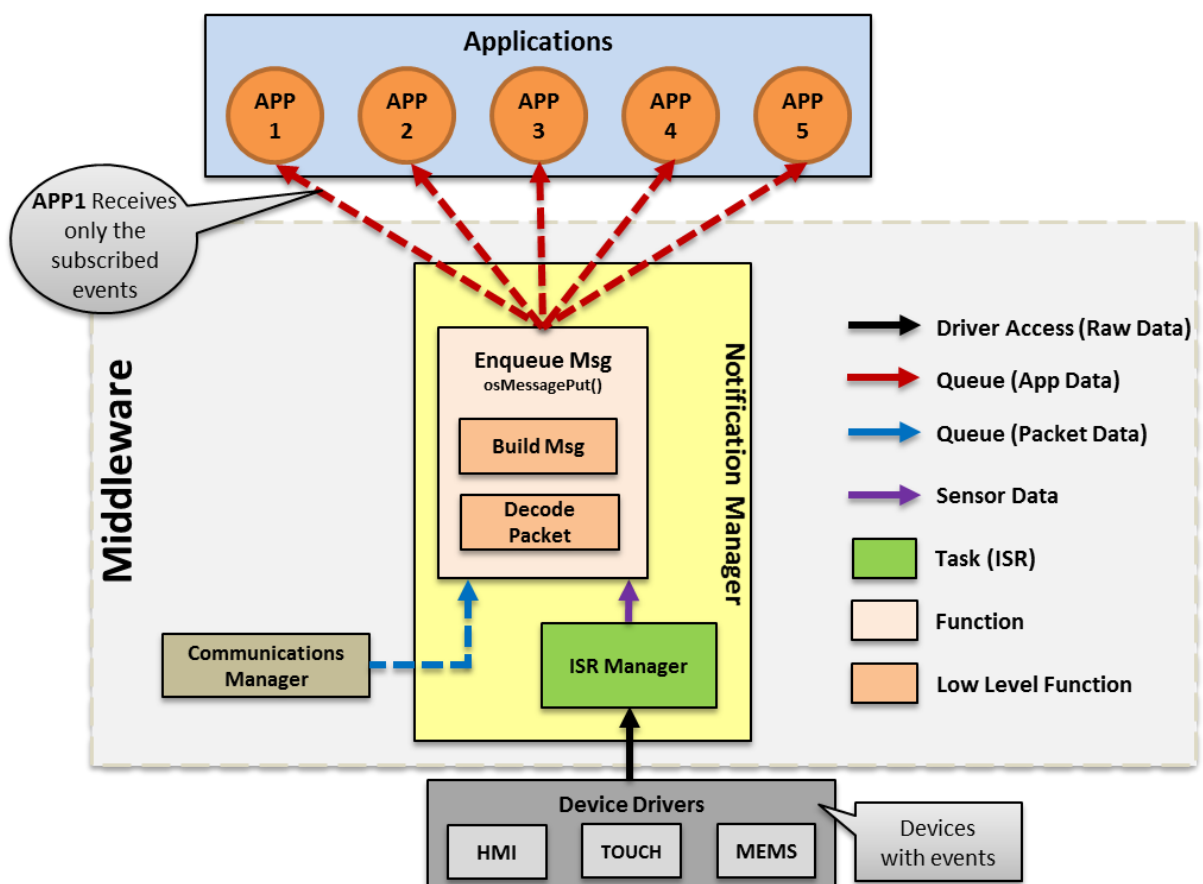


Figura 59 Gestor de Notificações (*Middleware*)

A implementação deste agente do *middleware* é constituído por diversas entidades, no entanto duas delas são essenciais ao seu funcionamento:

- **ISR Manager** – Esta tarefa subscreve e gere todas as interrupções relacionadas com o *hardware* de entrada, como os botões de pressão e de toque (*touch*) e os eventos programados do acelerómetro, tais como, acordar do estado parado e rodar de pulso.
- **Enqueue Msg** – Esta função tem como principal objetivo descodificar, reconstruir e empacotar os dados já referenciando-os a uma aplicação específica, garantindo desta forma a notificação apenas da aplicação à qual os dados dizem respeito. Além dos eventos/dados recebidos pelo *ISR Manager* esta função trabalha também em conjunto com o gestor de comunicações garantindo assim a entrega dos pacotes de comunicação diretamente à aplicação correta.

7.3.2. GESTOR DE RECURSOS (*RESOURCE MANAGER*)

O gestor de recursos é uma parte importante do *middleware*, responsável pela abstração do *hardware* (drivers de sensores), garantindo sempre a acessibilidade aos diversos recursos mesmo que existam pedidos simultâneos de varias aplicações. Esta funcionalidade é garantida recorrendo à metodologia de *publish/subscribe*.

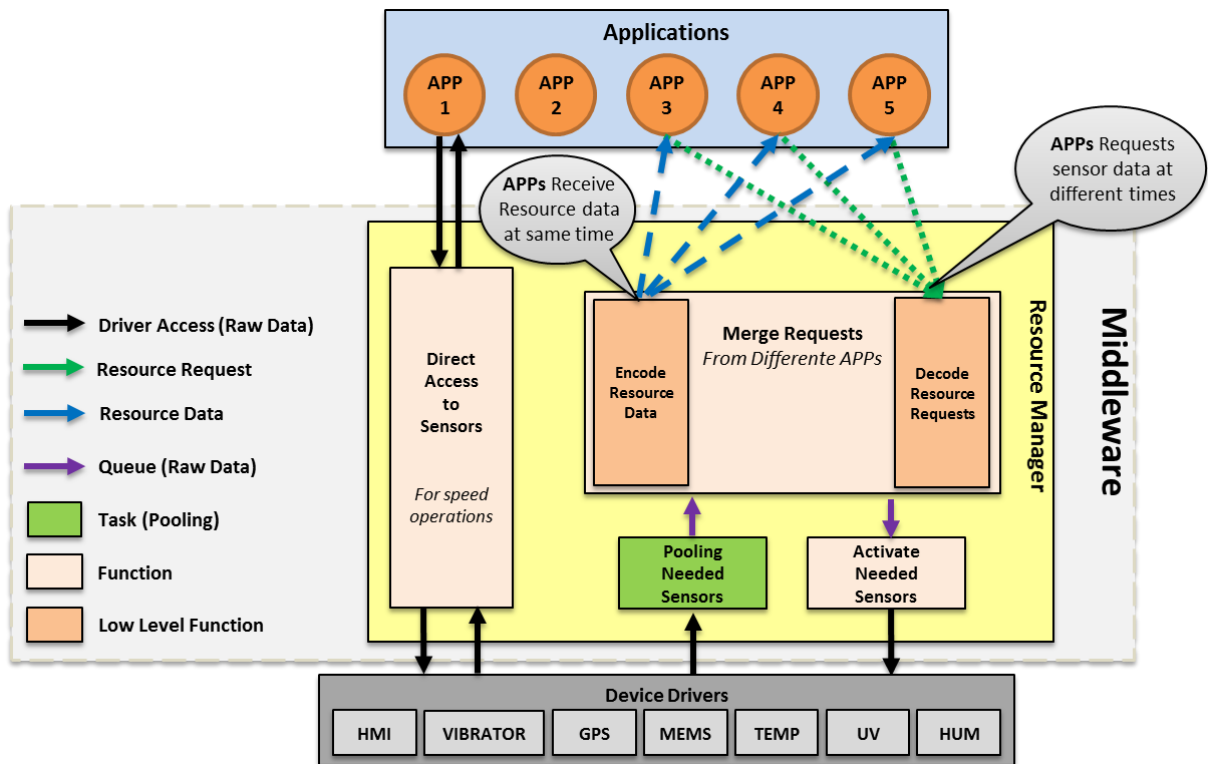


Figura 60 Gestor de Recursos (*Middleware*)

De modo a garantir uma arquitetura flexível do ponto de vista das aplicações, este agente permite o acesso aos recursos de duas formas distintas:

- **Acesso direto aos sensores** – Este método permite o acesso direto das aplicações aos drivers, garantindo assim que cenários com necessidade de informações temporalmente recorrentes possam aceder aos dados sem tratamento nem gestão por parte de entidades externas. Por exemplo, um cenário que necessite dos dados do acelerómetro com uma grande taxa de aquisição pode ao invés de subscrever os dados ao *Merge Requests* que implicava uma cadência de dados fixa, aceder diretamente ao sensor e recolher ela mesmo os dados.

- **Merge Requests** – Este segundo método por sua vez garante uma grande abstração dos sensores, sendo apenas necessário que as aplicações subscrevam o tipo de dados que desejam receber. Após uma subscrição a aplicação irá receber as informações a uma cadência fixa, até a mesma cancelar a subscrição. Desta forma é assegurada a transparência e facilidade no acesso aos dados.

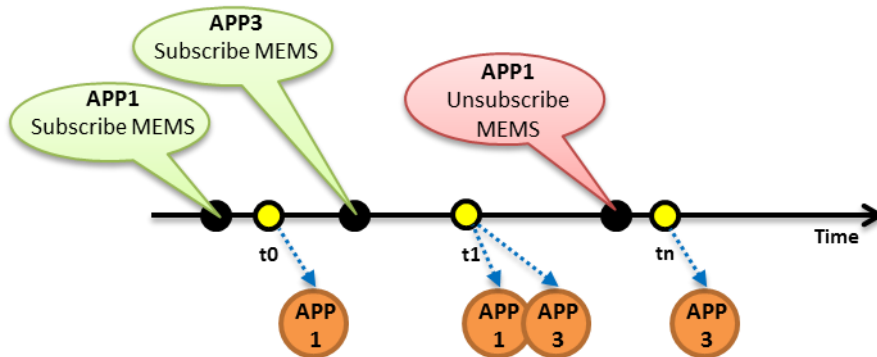


Figura 61 Exemplo de funcionamento da função *Merge Requests*

7.3.3. GESTOR DO INTERFACE GRÁFICO (UI MANAGER)

O gestor do interface gráfico é responsável pela gestão de todos os acessos ao ecrã, bem como pela transparência como são feitas as escritas no mesmo.

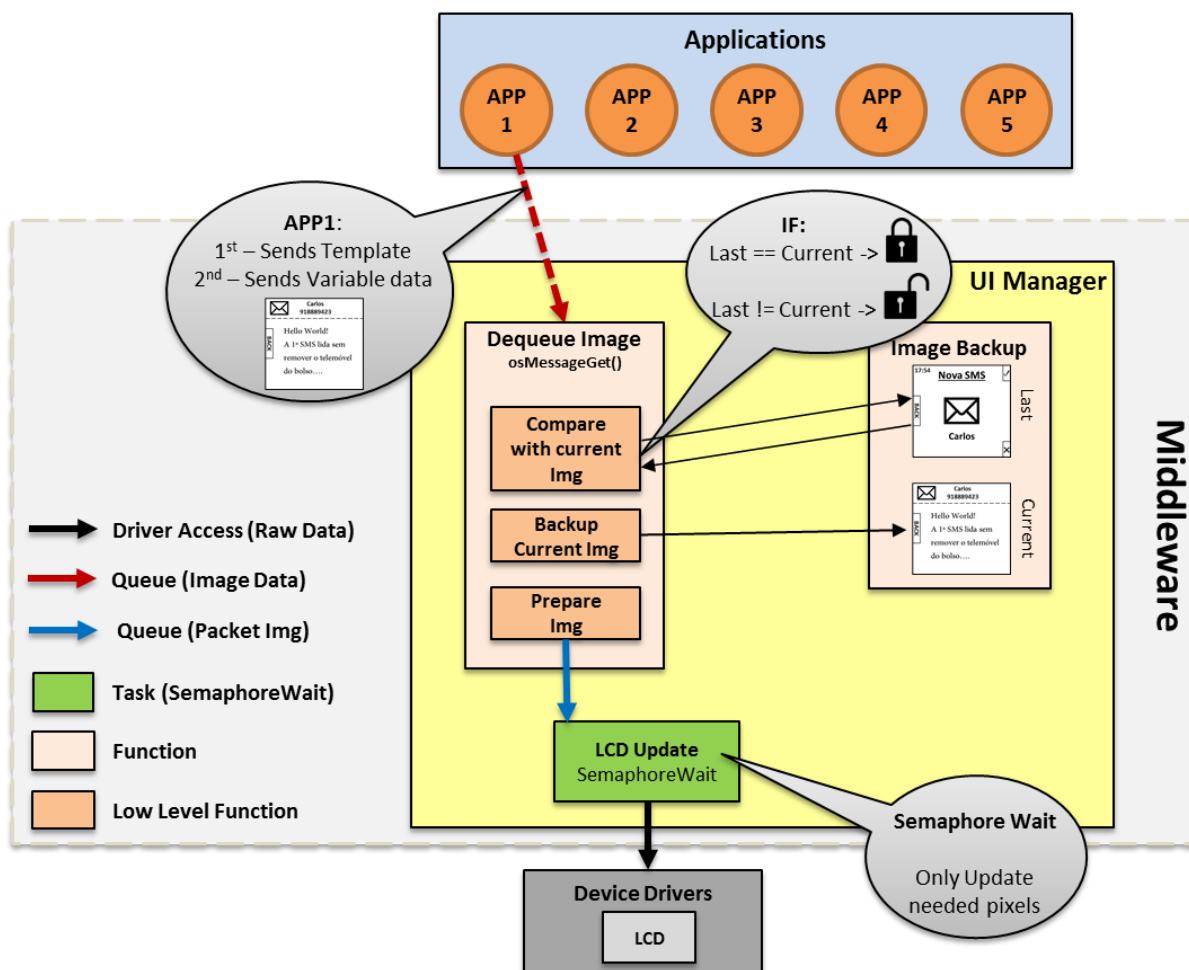


Figura 62 Gestor do Interface Gráfico (Middleware)

As aplicações que desejem escrever no ecrã necessitam apenas de enviar para um dos seguintes recursos para o gestor:

- **Img Template** – Este recurso diz respeito ao modelo da imagem que é constituído por uma estrutura de pixéis completa (128x128). Desta forma o gestor apenas necessita de escrever diretamente todos os pixéis no ecrã (isto se a imagem for diferente da imagem atual presente no ecrã).
- **Variable Data** – Sempre que for necessária a utilização de ecrãs dinâmicos, a aplicação deve utilizar este recurso, bastando para isso indicar qual a zona de

pixels a atualizar. Desta forma é reduzida a quantidade de informação a ser escrita no ecrã bem como a quantidade de recursos transferidos pelo sistema.

A Figura 63 ilustra um exemplo da interação de uma aplicação com o gestor do interface gráfico:

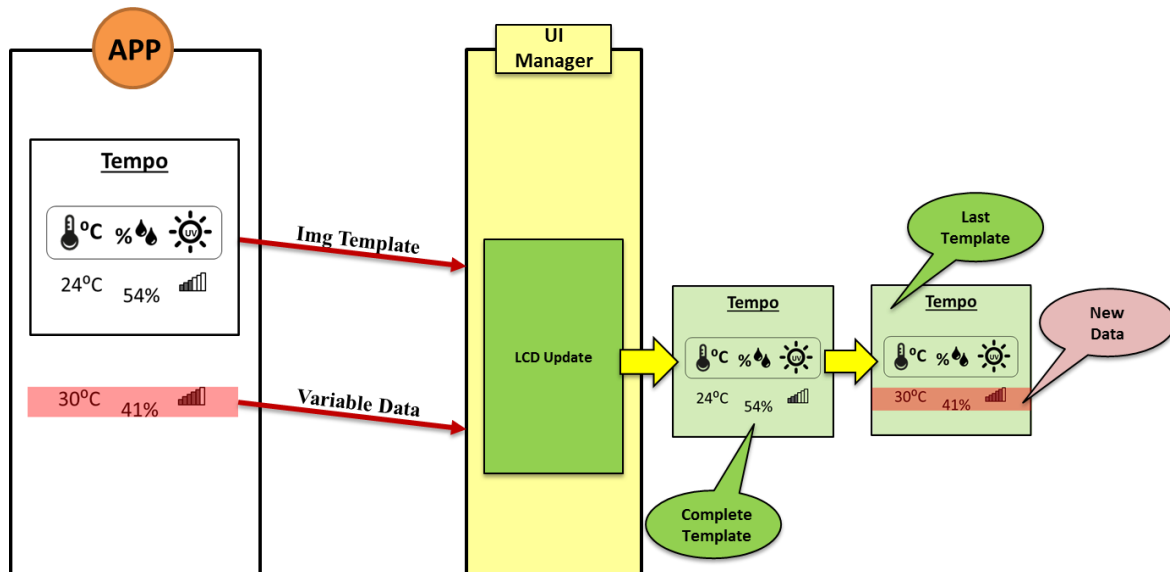


Figura 63 Sequência de interação com o gestor do interface gráfico

7.3.4. GESTOR DE COMUNICAÇÕES (*COMMUNICATIONS MANAGER*)

O gestor de comunicações é a entidade do *middleware* responsável pela abstração das comunicações. Esta abstração garante a todas as aplicações um alto nível de transparência no momento da transmissão de informações para o exterior do dispositivo, não interessando à mesma qual as interfaces disponíveis.

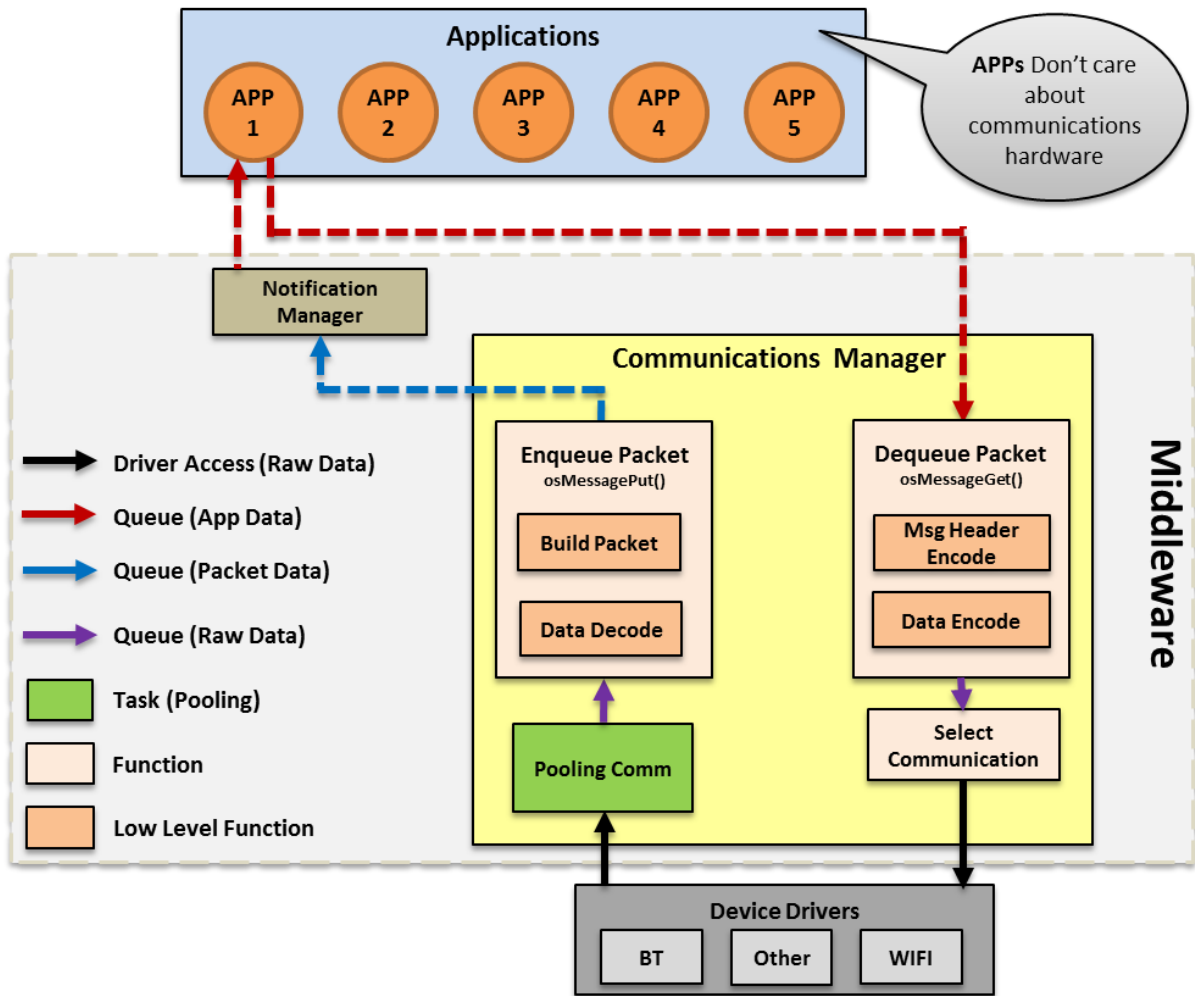


Figura 64 Gestor de Comunicações (*Middleware*)

7.4. APLICAÇÕES

Neste capítulo vão ser apresentadas as aplicações específicas tendo em conta todas as funcionalidades oferecidas pelas camadas inferiores (sistema operativo, *middleware* e *device drivers*).

7.4.1. APP 1 - ALERTAS SMARTPHONE

A aplicação para alertas provenientes do *smatphone* é caracterizada funcionalmente na Figura 65. A aplicação é genérica do pondo de vista de notificações permitindo que o utilizador receba diferentes tipos, nomeadamente chamadas, mensagens e correio eletrónico, redes sociais, agenda, entre outros, através da interação do dispositivo com o *smartphone*.

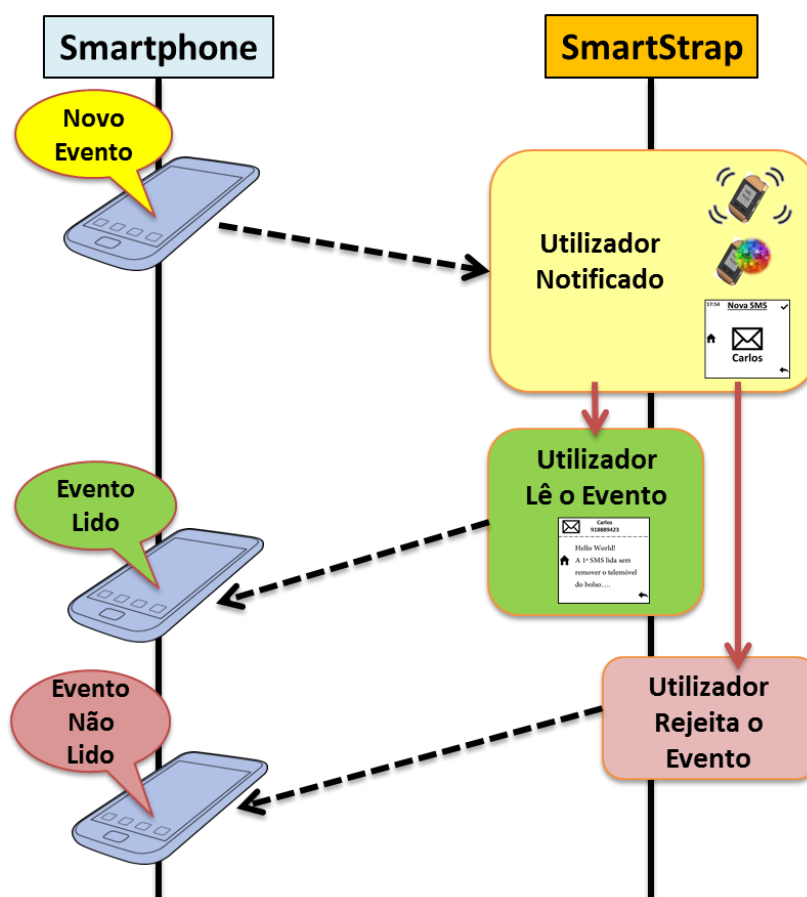


Figura 65 Diagrama geral de funcionamento da APP1 (Alertas Smartphone)

O interface com o utilizador reflete a interface mais usual apresentada no *smartphone*, permitindo incluir as funcionalidades mais básicas e as mais utilizadas com a mesma similaridade de utilização. No exemplo da Figura 66 pode ver-se a interação aquando da receção de uma nova SMS.

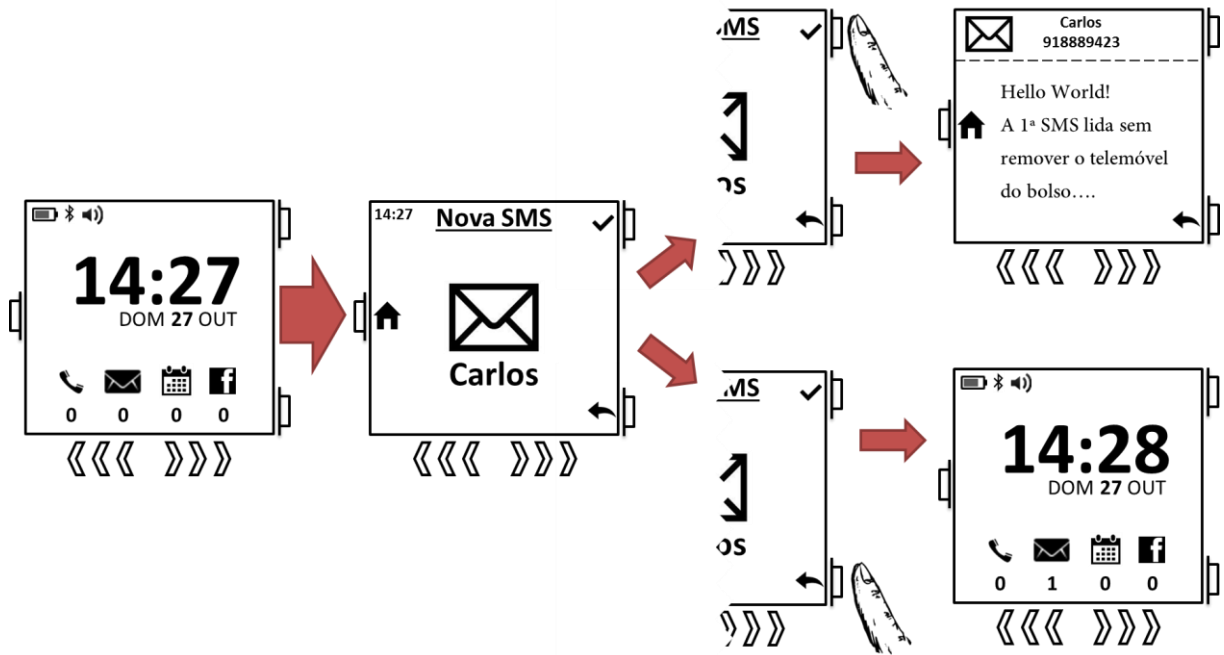


Figura 66 Exemplo da receção de uma nova SMS

Na Figura 67 podem ver-se alguns dos ecrãs apresentados no exemplo anterior da receção de um SMS em ambiente real.

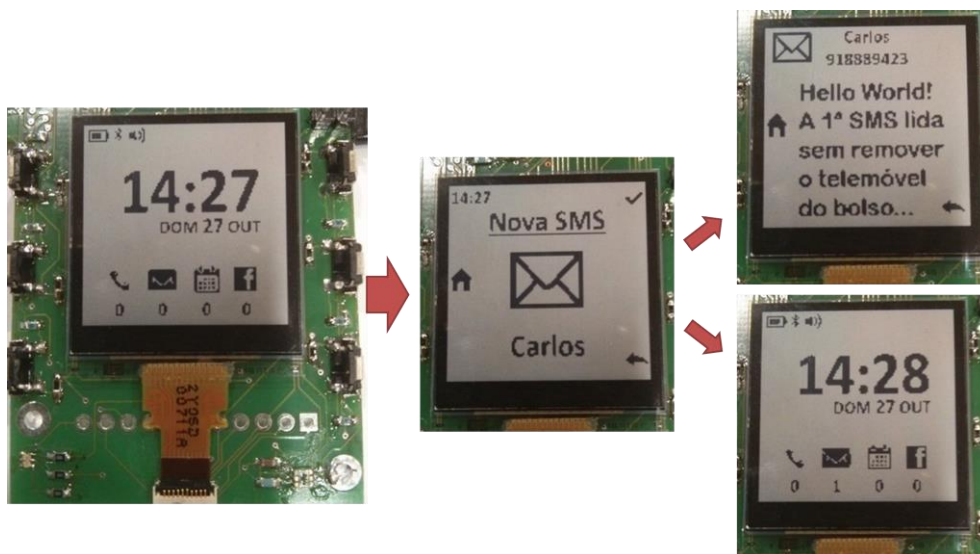


Figura 67 Exemplo real da aplicação de alertas de *smartphone*

7.4.2. APP 2 – CROWDSOURCING DE INFORMAÇÕES METEOROLÓGICAS

A arquitetura de funcionamento desta aplicação é apresentada na Figura 68. Funcionalmente a aplicação adquire os vários parâmetros meteorológicos (temperatura ambiente, humidade relativa e exposição a raios ultravioleta), enviando-os posteriormente para um sistema agregador (*online*). A georreferenciação é obtida com recurso as infraestruturas que utiliza para transmitir os dados.

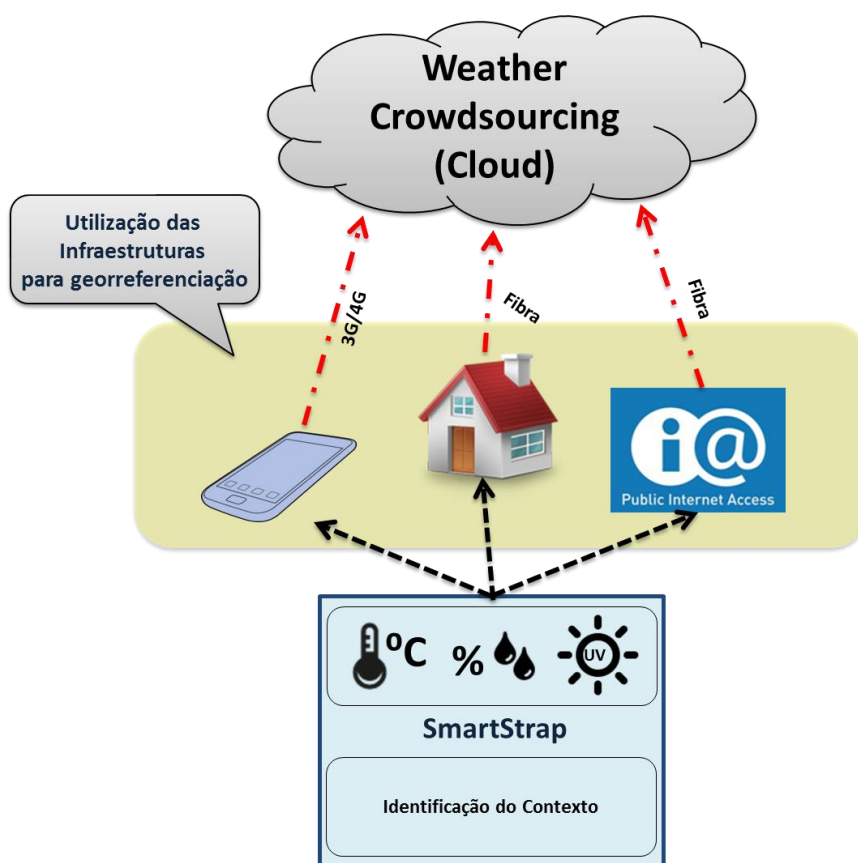


Figura 68 Diagrama geral de funcionamento da APP2 (*CrowdSourcing*)

Além da funcionalidade do envio de dados para uma entidade agregadora através dos recursos de comunicação (Bluetooth e Wifi) a aplicação disponibiliza um interface que reflete o estado atual do estado do tempo. No exemplo da Figura 69 pode ver-se a interação necessária para aceder a estes dados.

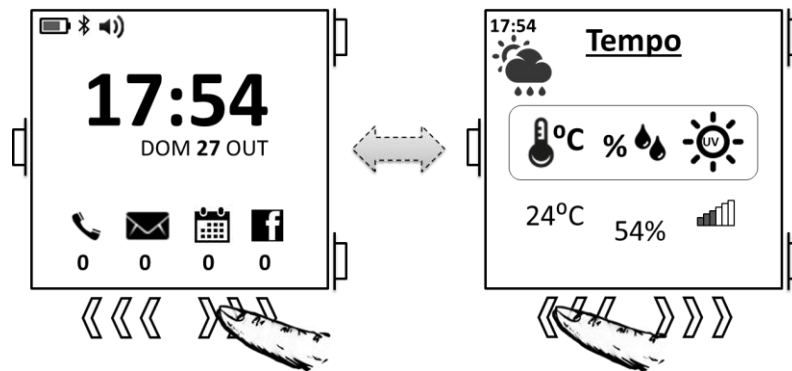


Figura 69 Exemplo da navegação na aplicação

Na Figura 70 podem ver-se alguns ecrãs relativos à aplicação de *Crowdsourcing* em ambiente real.

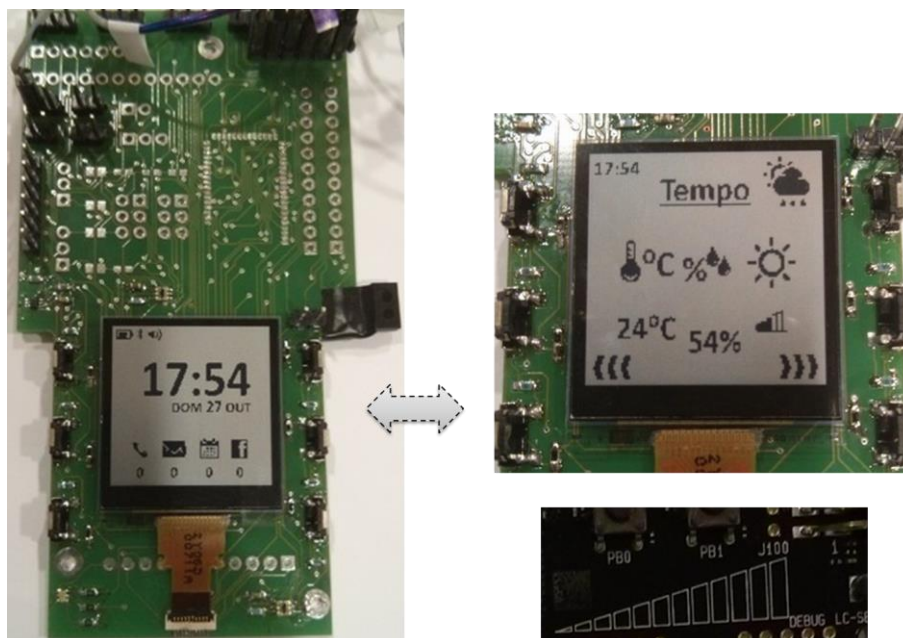


Figura 70 Exemplo real da aplicação de *Crowdsourcing*

7.4.3. APP 3 – MANUTENÇÃO E INSPEÇÃO NA INDUSTRIA

O funcionamento da aplicação de manutenção e inspeção na indústria é demonstrado pela Figura 71. Através da interação com a plataforma de gestão da manutenção é possível a visualização da informação relativa aos processos de manutenção e o preenchimento de formulários simples.

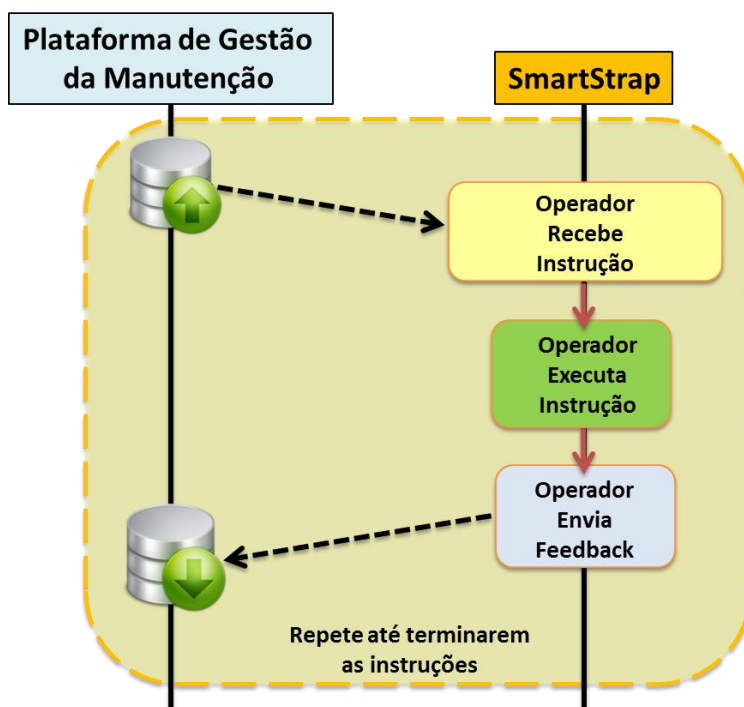


Figura 71 Diagrama geral de funcionamento da APP3 (Manutenção E Inspeção Na Industria)

Tendo em conta a natureza do ambiente industrial o interface com o utilizador prevê uma melhor interação através de mais botões e de imagens de tamanho mais generoso. Um exemplo de um interface industrial é apresentado na Figura 72.

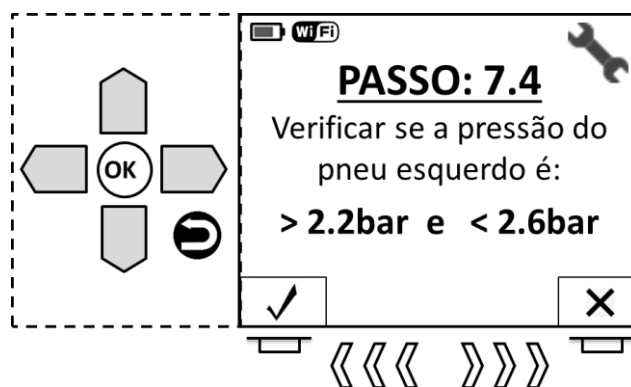


Figura 72 Exemplo do interface da interface industrial

7.4.4. APP 4 – MONITORIZAÇÃO REMOTA DE BOMBEIROS E MILITARES

A arquitetura de funcionamento da aplicação de monitorização remota de bombeiros e militares é representada na Figura 73. Esta aplicação adquire um conjunto de informações relativas à condição física do operacional e a sua georreferenciação. Depois de recolhidas, estas informações são enviadas para o centro operacional. Além deste sentido de informação (entre o operacional e o centro de operações) é também possível o envio de alertas ou outros avisos para o operacional.

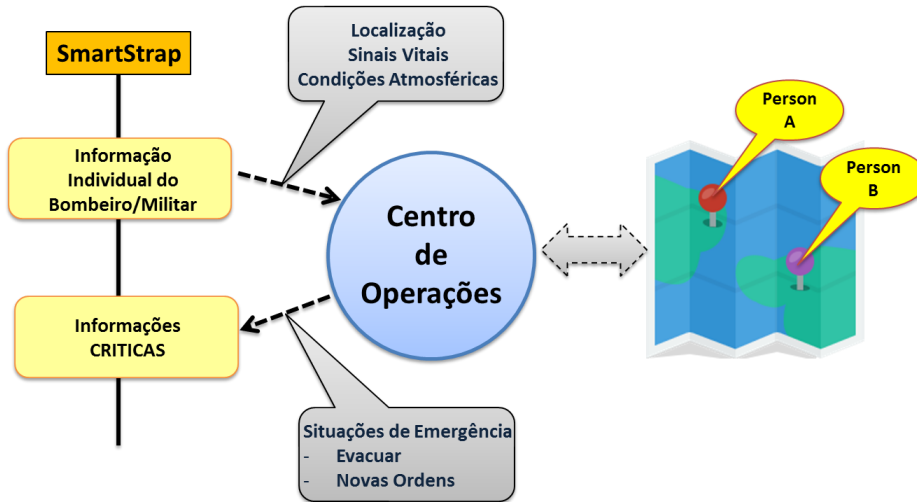


Figura 73 Diagrama geral de funcionamento da APP4 (Monitorização Remota)

Um exemplo conceptual da interface de monitorização remota é apresentado na Figura 74.

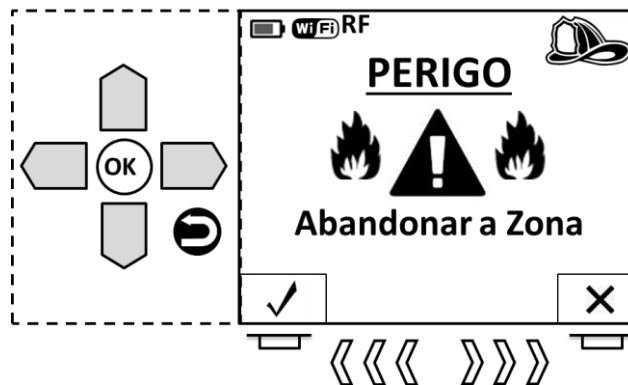


Figura 74 Exemplo do interface da interface de monitorização remota

Na Figura 75 podem ver-se alguns ecrãs relativos à aplicação em ambiente real.



Figura 75 Exemplo real da aplicação de monitorização remota

7.4.5. APP 5 – SINCRONIZAÇÃO E EFEITOS VISUAIS

A Figura 76 apresenta a arquitetura funcional da aplicação de sincronização e efeitos visuais. A sincronização de efeitos visuais implica que o centro de controlo conheça a disposição do espaço, tendo assim as condições necessárias para partilhar os efeitos visuais por todos os dispositivos da multidão. Os efeitos visuais poderão surgir como animação no ecrã ou como pontos emissores de luz.

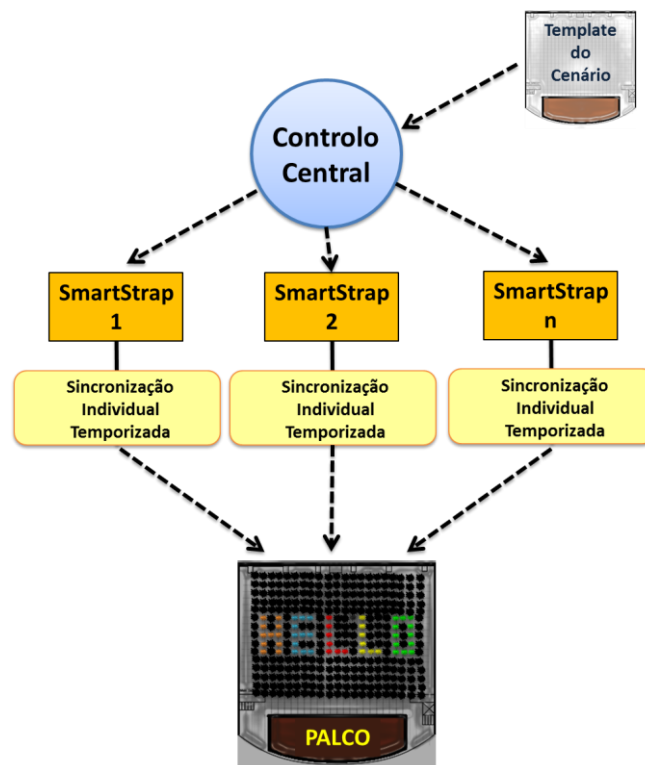


Figura 76 Diagrama geral de funcionamento da APP5 (Sincronização e Efeitos Visuais)

8. ARQUITETURA MECÂNICA

A mecânica está presente não apenas como uma vertente estrutural do equipamento mas também de *design* e de organização dos componentes eletrônicos. Deste modo, a componente da mecânica revelou-se um aspeto importante para apresentar neste trabalho, pois é através desta que é possível criar o interface físico entre o equipamento e os utilizadores. Note-se que o método de fabrico não é uma preocupação, sendo o modelo apresentado apenas para demonstração do aspeto final aproximado. Uma das preocupações iniciais e que está presente ao longo do desenvolvimento mecânico é a sua adaptabilidade aos diversos pulsos.

Nesta fase o principal interesse encontra-se em transmitir o aspeto visual desejado para o equipamento, para tal foram realizados alguns desenhos preliminares à mão (Figura 77) posteriormente passados para o *software* de desenho mecânico *SolidWorks*.

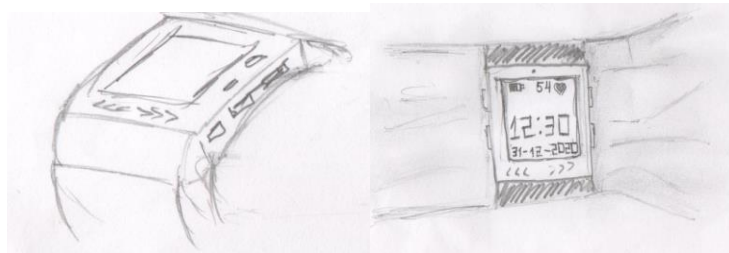


Figura 77 Desenhos mecânicos manuais

Tornou-se um desafio aprender [20] a trabalhar com um *software* de uma área completamente nova, no entanto revelou-se uma ferramenta crucial para traduzir os componentes eletrónicos em um equipamento. O primeiro passo foi a obtenção do objeto a três dimensões (ferramenta *Sketch* e *Extruded*), como ilustrado pela Figura 78.

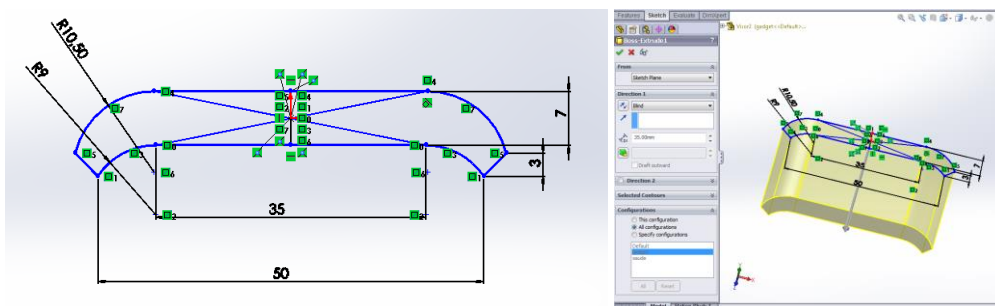


Figura 78 Desenho do objeto a três dimensões

De forma a simular o espaço necessário para a eletrónica foram criadas indentações e espaços vazios para a colocação dos diversos componentes (ferramenta *CutExtrude*) como visíveis na Figura 79.

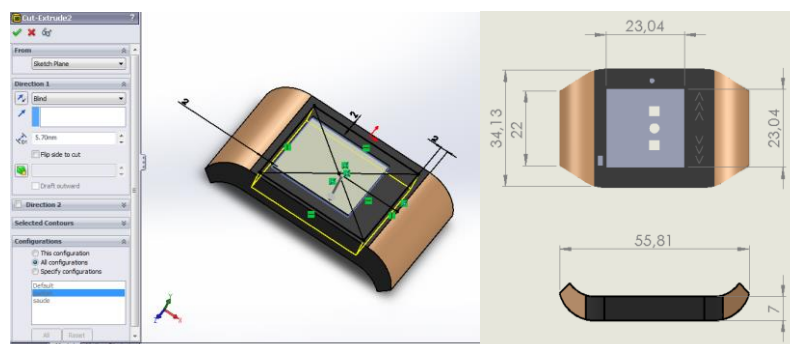


Figura 79 Desenho das indentações para os componentes

8.1. DISPOSIÇÃO DA ELETRÓNICA

Uma das principais preocupações da arquitetura mecânica é o agrupamento dos componentes eletrónicos, tendo em conta que estes induzem algum atravancamento, sendo necessário um estudo prévio da sua disposição. Sendo o tema desta tese abrangente em dois diferentes pontos de utilização, comum e industrial, torna-se necessário adaptar o *design*.

Em comum, os principais componentes do sistema em análise são:

- Ecrã
- Bateria (15 x 28 x 4 mm)
- Diversos dispositivos de medição
- Botões de interface (principalmente no dispositivo industrial)

O estudo da disposição da eletrónica iniciou-se com o gadget havendo depois uma substituição e adição de funcionalidades para o dispositivo industrial. Esta diferenciação também é visível no que trata a seleção do ecrã, visto que se optou por um modelo maior no dispositivo industrial, de modo a otimizar a interação entre o utilizador e o equipamento. Para além disso permite que a disposição de informação seja mais alargada e mais visível.

8.2. DESIGN DO PROTÓTIPO

Para o *design* inicial optou-se por uma abordagem adaptativa, que garanta a funcionalidade conjugada com um design moderno.

Na parte frontal do gadget existem apenas quatro interfaces (Figura 80 a), um LED, um IR, o ecrã e o *slide*, este último permite ao utilizador navegar entre os vários menus existentes. Na lateral existem ainda mais três botões, para a interação do utilizador com o *software*.

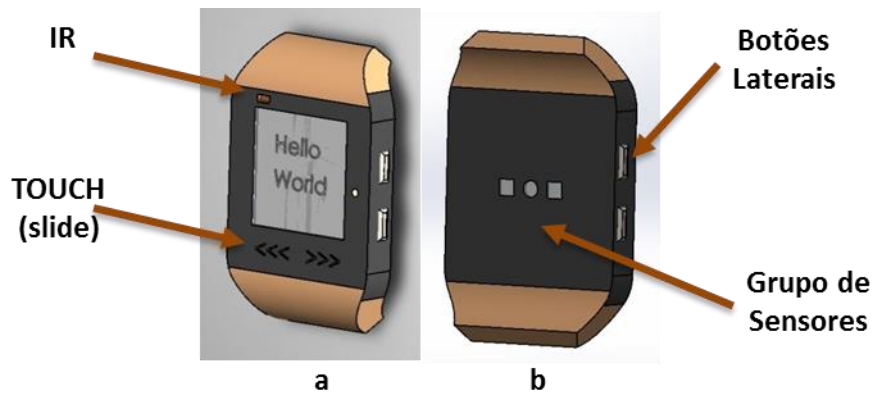


Figura 80 *Design* do protótipo gadget, parte frontal (a) e parte posterior (b)

Na parte posterior do dispositivo estão localizados alguns sensores que interagem com o utilizador (Figura 80 b), sendo de grande importância garantir o bom contato entre estes. Assim sendo o *design* final tem que ter em conta a usabilidade do dispositivo por diferentes pessoas, diferença essa que se reflete nas dimensões do pulso (Figura 81).

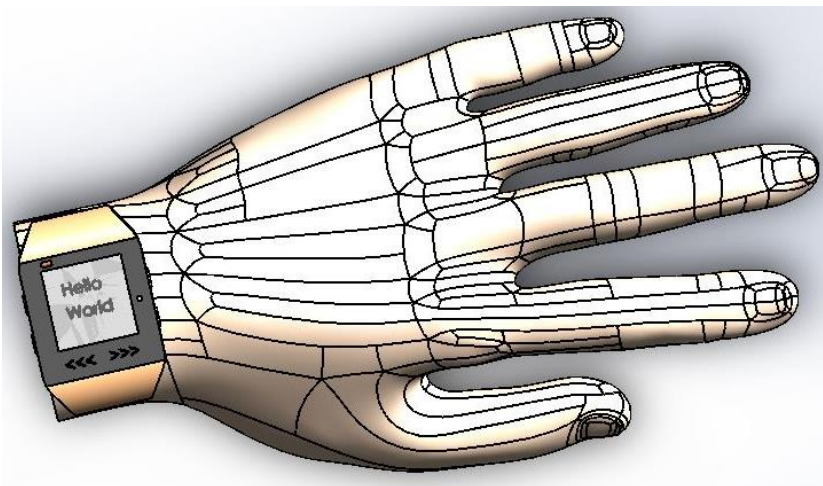


Figura 81 Simulação do aspeto do dispositivo quando em utilização

9. CONCLUSÕES

Ao longo deste texto foram apresentadas análises e estudos que permitiram sustentar as opções de desenvolvimento efetuadas durante o projeto. Assim, nesta última secção é realizada uma conclusão, abordando as consequências, relevância do trabalho realizado e eventuais desenvolvimentos futuros.

O desenvolvimento de dispositivos *wearable* está em franco crescimento, e no espaço de alguns anos, espera-se que o mercado deste tipo de equipamentos passe dos 14 milhões registados para cerca de 171 milhões em 2016. Estes factos só salientam a facilidade de absorção deste tipo de tecnologia/equipamentos, e a abertura dos utilizadores para aplicações que melhorem o seu dia-a-dia.

Desta forma, os cenários de aplicação explorados neste trabalho mostram um grande potencial de crescimento e uma forte possibilidade de absorção por parte dos utilizadores. Atualmente no mundo das tecnologias móveis, um dos fatores a que mais se dá importância, é à forma de disponibilização das informações e o tipo de ações necessárias para obtenção dessa mesma informação. De maneira a sustentar algumas opções foram identificados e estudados alguns conceitos fundamentais (capítulo 2).

Assim, de maneira a criar uma plataforma de *hardware* e *software* capaz de ostentar um ecossistema de aplicações [Figura 42], onde estão incluídas as aplicações dos cenários explorados, foram tidos em conta alguns requisitos e necessidades específicas considerados preponderantes neste tipo de equipamentos como apresentado no capítulo 5.1 (Requisitos do sistema).

A arquitetura de *hardware* tenta ser o mais abrangente possível, disponibilizando à camada aplicacional uma panóplia de sensores e interfaces, como são exemplo, os sensores de temperatura, humidade, acelerómetros e os interfaces gráfico, sonoro e luminoso como apresentado ao longo do capítulo 6 (Arquitetura de Hardware). Apesar da integração desta multiplicidade de componentes, nunca foi deixada de parte a vertente energética do equipamento, tendo sido escolhidos componentes [subcapítulo 6.6] de baixo consumo aliados ao facto da arquitetura implementada permitir uma gestão dinâmica dos recursos ativos. A implementação do *hardware* seguiu uma linha de demonstração do conceito, possuindo ainda dimensões excessivas para uma utilização no pulso [Figura 81], apresentado desta forma um aspeto de desenvolvimento como visível na Figura 82.

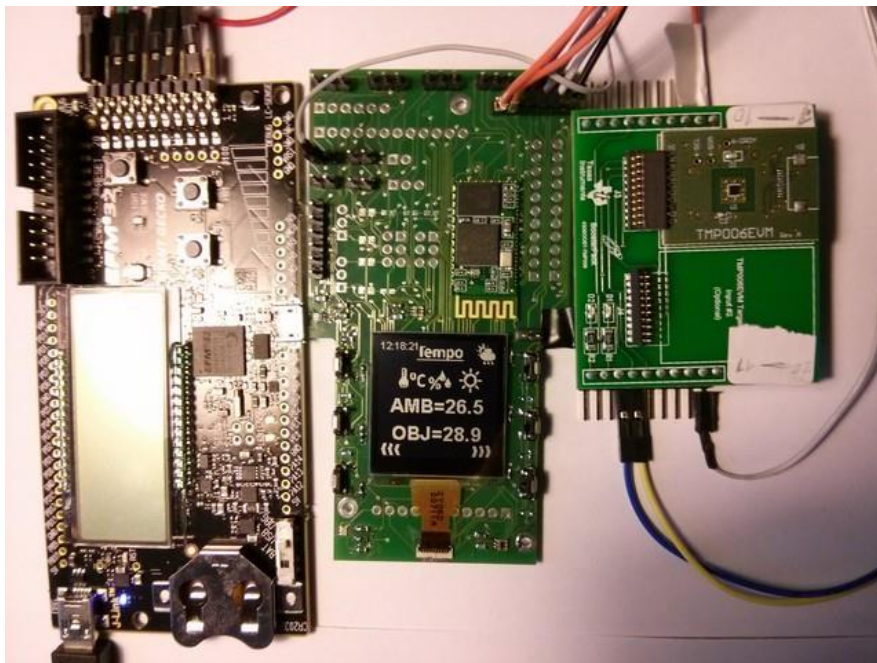


Figura 82 Protótipo de hardware

Da mesma forma que a arquitetura de *hardware* prevê uma diversidade de componentes elétricos, com o intuito de aumentar as possibilidades de aplicação, o software foi desenhado igualmente a pensar na facilidade e transparência para a criação de aplicações.

Esta arquitetura [Figura 42] permite assim a coexistência de várias aplicações no mesmo dispositivo, recorrendo para isso numa primeira instância às potencialidades do sistema operativo em tempo real [subcapítulo 7.2] e posteriormente às camadas de *software* implementadas, nomeadamente o *middleware* [subcapítulo 7.3] e os *device drivers* [subcapítulo 7.1]. Foram assim implementados os diferentes agentes constituintes da camada de *middleware*, responsáveis pela ponte entre as aplicações e os *device drivers*. No que respeita a estes últimos (*device drivers*) foram implementados os necessários para a exploração do conceito dos diferentes cenários de aplicação propostos [subcapítulo 7.4], nomeadamente, extensão do *smartphone*, manutenção e inspeção industrial, *crowdsourcing* de dados meteorológicos, monitorização de forças de segurança e ainda sincronização de informação para efeitos visuais.

Nesta fase são já perceptíveis alguns pontos que podem ser melhorados num futuro próximo. A incorporação de interfaces mais ricas seria uma mais valia tendo em conta a melhoria da experiência do utilizador. A expansão dos tipos de conectividades disponíveis permitiria acima de tudo uma maior compatibilidade com os diversos equipamentos do dia-a-dia. O último ponto, mas não menos importante prende-se com o aspeto visual do equipamento, que está diretamente ligado ao *hardware*, cuja preocupação esteve maioritariamente virada para a funcionalidade, mas que pode ser trabalhado com o objetivo de o tornar mais atrativo e de menores dimensões.

O estado atual do projeto faz ver que o conceito apresentado e desenvolvido é viável tanto do ponto de vista técnico como funcional. Desta forma um dos principais objetivos que se prendia com a demonstração do conceito foi cumprida, e perspetivam-se assim boas hipóteses de estes conceitos, métodos e tecnologia ser integrada em plataformas robóticas desenvolvidas no âmbito de projetos do Laboratório de Sistemas Autónomos (LSA) bem como no contexto industrial e de lazer.

Referências Documentais

- [1] Edward O. Thorp, “The Invention of the First Wearable Computer”, ISWC 98
- [2] A brief history of wearable computing,
<http://www.media.mit.edu/wearables/lizzy/timeline.html>
- [3] Steve Mann, http://wearcam.org/cyberman_sxsw_wired0_1282_50976_00.html.
- [4] Steve Mann, "Wearable computing as means for personal empowerment", ICWC-98, Fairfax VA, May 12-13 1998.
- [5] Steve Mann, http://en.wikipedia.org/wiki/Steve_Mann
- [6] Crowdsourcing, <http://pt.wikipedia.org/wiki/Crowdsourcing>
- [7] The Challenge of Wearable Computer Design,
<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/vuman/www/general>
- [8] "Augmented" wiring speeds up plane production,
http://edition.cnn.com/TECH/9711/21/t_t/jet.set.wiring/index.html
- [9] MYO, <https://www.thalmic.com/en/myo/>
- [10] Samsung GALAXY Gear SM-V700, <http://www.samsung.com/pt/consumer/mobile-phone/accessory/galaxy-gear/SM-V7000ZGABTU>
- [11] Dong-woo Lee; Jeong-mook Lim; John Sunwoo; Bae-sun Kim; Cheol-hoon Lee, "Actual Remote Control: A Wearable Remote Control on Wrist", Electronics and Telecommunications Research Institute, Daejeon, Korea
- [12] Chris Harrison; Hrvoje Benko; Andrew D. Wilson, “OmniTouch: Wearable Multitouch Interaction Everywhere”, Microsoft Research, Redmond, WA 98052, USA.
- [13] WR1100 Rugged Wrist Wearable Wireless Computer,
<http://www.parvus.com/Product/overview.aspx?prod=WR1100>
- [14] EmBlocks IDE, <http://www.emblocks.org>
- [15] Codeblocks IDE, <http://www.codeblocks.org>
- [16] GCC - the GNU Compiler Collection, <http://gcc.gnu.org/>
- [17] Sawhney, N.; Schmandt, C., “Speaking and Listening on the Run: Design for Wearable Audio Computing”, SecondIEEE International Symposium on Wearable Computers, Pittsburgh, PA, USA, October 19-20, 1998
- [18] Syrjärinne, J., “Studies of Modern Techniques for Personal Positioning”, Doctor of Technology Thesis, Tampere University of Technology

- [19] Hightower, J., Borriello, G., “A Survey and Taxonomy of Location Sensing Systems for Ubiquitous Computing”, Technical Report, UW CSE 01-08-03, University of Washington, Department of Computer Science and Engineering, Seattle, USA
- [20] SolidWorks Tutorials, <http://www.solidworks.com/sw/resources/solidworks-tutorials.htm>
- [21] Application Note Energy Micro - Low Energy Sensor Interface
- [22] LIS3DSH - MEMS digital output motion sensor ultra low-power high performance three-axis “nano” accelerometer
- [23] ML8511 - UV Sensor IC with Voltage Output
- [24] Marios Samdanis; Yikyung Kim; Soo Hee Lee, “The Emergence of Wearable Space: A Review and Research Implications”, CHI’13, April 27 – May 2, 2013, France
- [25] Oscar D. Lara and Miguel A. Labrador, “A Survey on Human Activity Recognition using Wearable Sensors”, Department of Computer Science and Engineering, University of South Florida

Anexo A. Dimensões exteriores do ecrã

