Instituto Politécnico de Viseu
Escola Superior de Tecnologia e Gestão de Viseu

João Carlos Dias Abrunhosa Araújo

Sistema de Aquisição e Monitorização de Dados para Indústria Alimentar e Laboratorial

Dissertação de Mestrado em Engenharia Eletrotécnica
Energia e Automação Industrial

Professor Doutor Miguel Francisco Martins Lima
Professor Bruno Filipe Lopes Garcia Marques

Setembro de 2012

RESUMO

A garantia das boas condições de armazenamento e conservação dos produtos é uma preocupação cada vez mais rigorosa, nomeadamente com a conservação de produtos alimentares.

Os requisitos exigentes da conservação alimentar, a necessidade de criar maior confiança na conservação de produtos e, consequentemente, na melhoria da qualidade de vida das populações em geral, levaram a que se criassem normas específicas para o efeito.

A série ISO 9000 define explicitamente a relação entre garantia da qualidade e a metrologia, incluindo diretrizes para se manter um controlo sobre os instrumentos e equipamentos, tornando assim necessária a implementação de um processo metrológico que permita obter ou manter uma certificação da empresa.

Tem-se vindo a verificar, por parte das empresas e técnicos responsáveis, uma maior consciencialização para este tema. Nas empresas surgem as equipas de controlo de qualidade e os instrumentos que lhes permitem efetuar continuamente as medições e registos requeridos pelas normas aplicáveis.

Com o objetivo de ir ao encontro destas necessidades, foi desenvolvida ao longo deste projeto uma ferramenta de trabalho que permite facilitar a atividade das equipas de controlo de qualidade, podendo mesmo ser transferido algum controlo a entidades externas que se responsabilizam por monitorizar várias instalações em simultâneo.

Nesse âmbito, revelou-se uma mais valia o desenvolvimento de uma aplicação Web, que permite remotamente ou localmente ter acesso a temperaturas de equipamentos de acondicionamento, temperaturas e humidades relativas dos locais, assim como das condições de luminosidade.

O fundamento deste projeto consiste na monitorização das condições de armazenamento de produtos, sobretudo nas áreas laboratoriais de análises clínicas e restauração. Para tal, idealizou-se um sistema com base num microcontrolador e sensores que permitem medir três tipos de variáveis físicas (temperatura, humidade e luminosidade), através de comunicação sem fios.

O sistema desenvolvido, que se descreve e fundamenta ao longo deste documento, pode facilmente ser expandido futuramente com a introdução de outros sensores.

ABSTRACT

The guarantee of good storage conditions and product conservation is an increasingly stringent concern, particularly with the conservation of food.

The demanding food conservation requirements, the need to sustain the general public trust on it, with the goal to improve life quality, has let to the creation of specific standards for this purpose.

The ISO 9000 series defines explicitly the relationship between quality assurance and metrology, including guidelines to keep a check on the instruments and equipment, thus making it necessary to implement a metrological process that allows a company to obtain or maintain their certification.

A greater awareness for this issue by companies and responsible technicians has been observed. Quality control teams are created and there is a selection of tools that allow them to continuously measure and maintain the records required by the rules.

Aiming to meet these needs, we developed throughout this project a tool that facilitates the activity of the quality control teams and can even transfer control to external entities, that are responsible for simultaneously monitoring multiple facilities.

The development of a web application that allows locally or remotely to measure temperatures of conservation equipment, temperatures and relative humidities of storage areas and the lighting conditions, proved to be a real asset in this framework.

The aim of this project lays on the monitoring of storage conditions of products, particularly in the areas of laboratorial clinical analysis and food industry. This system has been designed based on a microcontroller and sensors that can measure three types of physical variables (temperature, humidity and light), through wireless communication.

The system that was developed and is described throughout this document, may readily be expanded in the future with the introduction of other sensors.

PALAVRAS CHAVE

Monitorização metrológica

Aquisição de temperatura

Sensores sem fios

Sensores de luminosidade

Sensores de humidade

Comunicação sem fios XBee

Aquisição dados com Arduino

AGRADECIMENTOS

Este trabalho não teria sido possível sem o apoio e colaboração de diversas pessoas e instituições.

A todos aqueles que de alguma forma contribuíram para a realização deste Projeto Final e Dissertação, gostaria de expressar o meu agradecimento e reconhecimento profundo, pelo seu importante contributo e ajuda para a ultrapassar as dificuldades por mim sentidas.

Agradeço especialmente ao meu orientador Professor Doutor Miguel Francisco Martins de Lima, pela sua disponibilidade e apoio prestado.

Ao meu coorientador Professor Bruno Filipe Marques, igualmente pela disponibilidade e apoio prestado.

À Engidom, Engenharia, Lda, representada pelo Eng. Hurbano Mendonça e Eng. José Manuel Chambino, pelo acolhimento e pelas longas horas dispensadas para que fosse possível a realização deste projeto.

A todos os colegas e colaboradores da Engidom, que nas minhas horas de ausência, mantiveram e desenvolveram todos os trabalhos em curso.

Ao colega Eng. Augusto Casais pela paciência e longas horas de trabalho conjunto passadas na sala de investigação para que este projeto se tornasse realidade.

Ainda ao Eng. Estagiário Nuno Silva, pelo seu apoio no desenvolvimento da aplicação web.

Por fim, e não menos importante, às pessoas e familiares mais próximas pela compreensão e paciência, pelos longos períodos de ausência e projetos familiares adiados.

ÍNDICE GERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

ABREVIATURAS E SIGLAS

AD	Analógic Digital
ADC	Analog Digital Converter
AREF	Analog Reference, pino especifico da placa Arduino
CO_2	Dióxido de Carbono
DC	Direct current
DFU	Device Firmware Update
DSSS	Direct-sequence Spread Spectrum
EEPROM	Electrically-Erasable Programmable Read-Only Memory
FFD	Full Function Device
FTDI	Future Technology Devices
GSM	Global System for Mobile Communications
HDMI	High-Definition Multimedia Interface
I2C	Inter-Integrated Circuit
IC	Integrated Cirquito
ICSP	In Circuit Serial Programming
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Eletronics Engineers
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
MAC	Medium Acess Contrl
NTC	Negative Temperature Coefficient
OSI	Open Systems Interconnection
PHY	Physical - Camada física do protocolo ZigBee
PWM	Pulse Width Modulation
RDF	Reduce Function Device
SAP	Service Access Point
SMS	Short Message Service
SNMP	Simple Network Management Protocol
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
TTL	Transistor-Transistor Logic
TWI	Two Wire Interface
UART	Universal Asynchronous Receiver/Transmitter
ZDO	ZigBee Device Object

Capítulo 1 - Introdução

1.1. Contexto

Num ambiente de laboratório, num local de armazenamento e conservação ou em locais de preparação de produtos que necessitem de condições especiais, somos obrigados a cumprir regulamentos específicos e a respeitar os planos diretores de validação das grandezas ambientais, tais como a temperatura, a humidade e luminosidade.

A metrologia, como ciência das medições, está na base de qualquer atividade técnica e proporciona o desenvolvimento tecnológico associado à qualidade de produtos e serviços.

A crescente exigência dos mercados consumistas tem vindo a originar uma uniformização na qualidade dos produtos e serviços, que vão desde o processo de fabrico, passando pelo embalamento e transporte dos mesmos, até à forma como estes são colocados nos expositores de venda. Em qualquer uma destas etapas está presente uma premissa que é “Garantia de satisfação e qualidade”.

Ao longo do trajeto de produção, transporte e distribuição de um produto, ou mesmo na prestação de determinados serviços especializados, é necessário desenvolver mecanismos de controlo de processos, normalmente associados a grandezas mesuráveis, nomeadamente temperaturas, humidades, densidades, luminosidade, velocidades, entre outras.

Desta forma, as tecnologias modernas tendem a adaptar-se e a evoluir tecnologicamente, com o objetivo de facilitar todos os processos de controlo e garantias de qualidade. Nesse âmbito, as comunicações sem fios têm vindo a ocupar um espaço importante no que concerne aos processos de medição, quer ao nível da readaptação de processos, quer ao nível de novas unidades de produção/prestação de serviços. A facilidade de adaptação dos sistemas de comunicação sem fios, aliada à grande fiabilidade dos sistemas de *hardware* e protocolos de

comunicação, dão garantias de maior simplicidade na implementação, assim como maior flexibilidade.

1.2. Motivação

Pretendeu-se neste projeto desenvolver um sistema capaz de ir ao encontro dos requisitos, de fiabilidade na conservação de produtos segundo as normas aplicáveis.

O sistema irá permitir recolher em tempo real os valores de diversas grandezas, quer em locais de ambiente controlado, quer em equipamentos, nomeadamente câmaras frigoríficas e estufas que se destinam especificamente à conservação de reagentes utilizados em laboratórios de análises clínicas, assim como de colheitas para análise.

Adicionalmente, o sistema permitirá monitorizar também espaços restritos onde são desenvolvidas atividades específicas da mesma área.

A aplicabilidade e versatilidade do sistema permite também a adaptação a outros tipos de atividades, nomeadamente na restauração, no que se refere ao acondicionamento de produtos confecionados com necessidades de condições específicas de conservação antes de serem servidos, ou no armazenamento de matéria-prima para confeção alimentar. O sistema pode ainda utilizar-se na indústria e comércio de congelados.

O sistema permitirá a supervisão em tempo real, local e remotamente, de um conjunto de dados, nomeadamente de temperatura, humidade relativa e iluminação em locais de armazenamento, câmaras frigoríficas, estufas e locais de trabalhos específicos. O sistema permitirá ainda a visualização de históricos e de alarmes em tempo real.

Para além do interface homem/máquina, efetua-se um registo em base de dados de toda a informação recolhida em tempo real, assim como o tratamento gráfico dessa informação, podendo ainda ser disponibilizada em suporte de papel.

1.3. Objetivos e Requisitos do Sistema

Tendo como base os procedimentos existentes nos laboratórios de análises clínicas, no que concerne ao tratamento e conservação, quer de amostras recolhidas de pacientes, quer de alguns tipos de produtos e reagentes utilizados nos vários exames a realizar, é de máxima importância manter monitorizadas as condições ambientais e de conservação dos mesmos.

O mesmo se passa nos processos de produção alimentar. O acompanhamento dos produtos para confeção alimentar é extremamente sensível no que diz respeito ao armazenamento em câmaras frigoríficas ou estufas de conservação e secagem, humidificação, entre outras.

Para um controlo correto e adequação das características do produto final, quer ao nível do resultado das análises clínicas, quer ao nível dos produtos alimentares, existem diversos

sensores aplicáveis ao longo dos processos, nomeadamente, temperatura, humidade e luminosidade.

Não sendo o objetivo desta dissertação abranger exaustivamente todas as situações possíveis, pretende-se desenvolver um conjunto de:

- Dispositivos ("nó"), com comunicação sem fios ("*rooter/end device*"), tendo em vista o acoplamento de diversos tipos de sensores, possibilitando assim uma comunicação, com ausência de fios, dos dados recolhidos pelos sensores;
- Uma unidade central principal ("*coordinator*") ligada a uma unidade de processamento (CPU), que permite em tempo real processar a informação, monitorizar e efetuar a gestão de alarmes.

Nos processos produtivos existem condições particulares de implantação dos sensores, nomeadamente no que se refere à sua mobilidade, embora, na maioria dos casos, os sensores permanecerão fixos e estáticos.

No entanto, por vezes, devido às condições de produção ou reorganização frequente dos espaços de armazenamento ou transporte de produtos, é necessária uma total autonomia energética dos sensores, sendo, por isso, necessária a utilização de pilhas e/ou baterias. Os dispositivos devem também ser suficientemente robustos, tendo em conta a sua aplicabilidade em ambientes industriais.

Assim, para garantir a flexibilidade e fácil integração do sistema, os dispositivos ou módulos de comunicação deverão caracterizar-se por:

- Independentemente das condições exteriores, poderem ser integrados, no que diz respeito aos aspetos construtivos, em invólucros estanques e fisicamente robustos, não criando assim restrições na sua implantação;
- Sendo desejável a sua fácil integração, possuírem um interface e protocolos de comunicação normalizados, possibilitando a ligações a dispositivos já existentes;
- Possuírem tamanho reduzido, para maior facilidade de implantação;
- Apresentarem baixa complexidade, que estará associada também a um baixo consumo energético, conseguindo-se assim uma autonomia alargada, no caso dos dispositivos associados a baterias ou pilhas;
- Possuírem baixos custos construtivos e de implementação;
- Possuírem fiabilidade elevada, logo ter-se-á de ponderar a relação preço/ qualidade/fiabilidade. No que se refere à implantação, esta deve ser tão fácil quanto possível, de modo a que qualquer pessoa possa, após a primeira instalação, ser capaz de proceder a alterações e montagem, evitando custos adicionais com equipas externas especializadas, que, na maioria dos casos, são bastante onerosas.

1.4. Organização da Dissertação

A organização da dissertação encontra-se dividida em quatro capítulos.

No Capitulo 1 – Introdução - é feita uma contextualização do tema da dissertação na atualidade, é abordada a necessidade contínua de medição e controlo de todas as fases produtivas, de forma a garantir as necessidades impostas pelos sistemas de qualidade.

No Capítulo 2 – Sistemas de aquisição de dados - é feita uma abordagem sumária à Norma ISO 9000, assim como é dada uma panorâmica geral dos sistemas de monitorização e controlo atualmente comercializados, fazendo referência às principais características e formas de integração. Faz-se ainda uma abordagem ao protocolo de comunicação ZigBee.

No Capítulo 3 – Implementação de um sistema de aquisição de dados - efetua-se uma abordagem à tecnologia utilizada, para além de serem apresentadas as razões que levaram à sua escolha para o desenvolvimento deste projeto. É descrito todo o equipamento, sensores, módulos de comunicação e placas adaptadoras utilizadas para o desenvolvimento do projeto, assim como se descreve o modo de funcionamento do sistema desenvolvido. Abordam-se ainda os resultados obtidos.

No Capítulo 4 – Conclusão e desenvolvimentos futuros - fazem-se as considerações finais de todo o projeto, onde se realçam as qualidades e vantagem do sistema desenvolvido, bem como são tecidas algumas sugestões para desenvolvimentos futuros.

Nesta dissertação, pelo facto de se abordar uma área tecnológica em que a literatura anglo saxónica é abundante, e com o objetivo de facilitar o entendimento dos assuntos por parte do leitor, optou-se por utilizar algumas expressões em inglês.

Capítulo 2 - Sistemas de Aquisição de Dados

2.1. Introdução

Para um correto transporte e manuseamento de produtos/amostras nos seus processos de fabrico ou em laboratórios deverão ser garantidas as condições que evitem toda e qualquer modificação do número de microrganismos presentes. Isto é, deve ser garantido que as características iniciais dos produtos não sejam alteradas. Para isso, e de acordo com inúmeros documentos específicos para cada produto, são definidas para a sua conservação as condições de temperatura e humidade, entre outras grandezas.

A título de exemplo, e de acordo com o procedimento descrito na Norma Francesa – NF ISO 7218, de Dezembro de 2001, Microbiologia dos Alimentos – *Regras gerais para análises microbiológicas,* deve ser dada uma atenção especial às temperaturas de conservação, com os seguintes tipos de produtos:

a) Produtos estáveis: temperatura ambiente;

b) Produtos frescos e refrigerados: entre 0^{o}C e $+4^{o}$C;

c) Produtos congelados ou ultracongelados: inferior a -18^{o}C;

d) Produtos pasteurizados e similares: entre 0^{o}C e $+4^{o}$C;

e) Produtos alimentares sensíveis (por exemplo: carnes cruas, produtos da pesca) devem ser conservados a uma temperatura entre os 0^{o}C e $+ 2^{o}$C.

A monitorização e controlo de equipamentos são de máxima importância e, nesse âmbito, tem-se vindo a dotar os equipamentos e instalações de sistemas, mais ou menos sofisticados, de forma a garantir as condições específicas necessárias.

A utilização destes sistemas permitem, de uma forma rápida, evitar a utilização de produtos que, devido ao seu estado de conservação, podem, por vezes, ser nocivos à saúde ou atempadamente gerar alarmes que evitem os processos de deterioração desses mesmos produtos.

Os sistemas de monitorização e controlo possuem a capacidade de recolher e interpretar sinais obtidos pelos diversos sensores instalados e torná-los visíveis, e facilmente interpretáveis pelos operadores, para que se tomem medidas corretivas.

Para responder aos requisitos anteriormente referidos, existe já no mercado um grande número de sistemas que podem ir desde os instrumentos de medição portáteis até sistemas integrados inteligentes, usados de acordo com a especificidade em causa e de acordo com a disponibilidade económica.

2.2. Generalidades da Norma ISO 9000

As normas da família ISO 9000 são referenciais para a implementação de sistemas de gestão da qualidade (SGQ) que representam um consenso internacional sobre boas práticas de gestão e com o objetivo de garantir o fornecimento de produtos que satisfaçam os requisitos dos clientes, estatutários e/ou regulamentares, bem como a prevenção dos problemas e a enfase na melhoria contínua.

A definição de produto deverá ser interpretada de acordo com a ISO 9000:2000 e respetivas notas associadas (definição 3.4.1). O produto é definido como o resultado de um processo e divide-se em quatro categorias genéricas: Serviços (ex. transportes); *Software* (ex. aplicações informáticas, dicionários); *Hardware* (ex. equipamentos mecânicos de máquinas) e Materiais Processados (ex. lubrificantes) [1].

Estas boas práticas são compiladas num conjunto de requisitos normativos (ISO 9001:2000) e orientações para a melhoria do desempenho (ISO 9004:2000), cuja implementação é independente do tipo, dimensão e setor de atividade das organizações. Estas duas normas formam um “par consistente”, com estruturas e formatos alinhados.

A adoção generalizada destas normas é justificada, quer por fatores de competitividade, quer por exigências formais dos clientes e da sociedade. Constituindo um referencial aceite a nível multissetorial e internacional, disponibiliza um “mapa” para a definição do SGQ que potencia a satisfação dos diversos clientes de uma organização.

A família ISO 9000 é constituída por 4 normas principais, a saber:

- *ISO 9000:2000* (sistemas de gestão da qualidade. Fundamentos e vocabulário);

  Trata-se de uma norma de base que descreve os princípios básicos, terminologia e definições nas quais as restantes normas da família se fundamentam.

- *ISO 9001:2000* (sistemas de gestão da qualidade. Requisitos);

  Esta norma é a mais conhecida e na qual se assenta a certificação. O seu objetivo é fornecer um conjunto de requisitos que permitam a uma organização demonstrar a sua capacidade e aptidão de forma consistente, proporcionar produtos com qualidade que vão ao encontro dos requisitos do cliente e regulamentação aplicável.

- *ISO 9004:2000* (sistemas de gestão da qualidade. Melhoria do desempenho);

  É nesta norma que são fornecidas as recomendações para os organismos que pretendam ir para além dos requisitos da ISSO 9001 e que queiram desenvolver um sistema de gestão de qualidade que melhore a eficiência e a eficácia organizacional.

- *ISO 19011:2002* (sistemas de gestão da qualidade. Orientação para auditorias);

  Nesta norma são definidas linhas de orientação para auditorias a sistemas de gestão da qualidade e/ou de gestão ambiental, podendo as organizações usá-las para desenvolverem os seus próprios programas de auditorias e/ou programas de avaliação de fornecedores.

Um vasto número de normas é baseado na ISO 9000. No caso deste projeto o objetivo será de ir ao encontro dos requisitos da ISO 9001:2000 e da sua revisão realizada em 14-11-2008 originando a ISO 9001:2008.

Para a atual revisão de 2008, a ISO estabeleceu como objetivo apenas fazer clarificações do texto da norma, com base nos 8 anos de experiência da sua utilização em todo o mundo, não introduzindo novos requisitos.

Embora existam algumas alterações em relação ao texto de 2000, a SGS-ICS (Société Générale de Surveillance S.A. - Serviços Internacionais de Certificação Lda) considera que as clarificações ora formalizadas coincidem com o seu entendimento, desde sempre adotados na prática de Auditoria e Certificação dos sistemas de Gestão da Qualidade dos seus Clientes.

Por esta razão, é expectável que a ISO 9001:2008 terá impactos relativamente reduzidos nos SGQ adequadamente implantados e já certificados pela SGS ICS segundo a ISO 9001:2000.

Neste sentido, seguidamente referem-se alguns aspetos da ISO 9001:2000.

2.3. Enquadramento da ISO 9001:2000

Como já vimos anteriormente, a ISO 9001:2000 estabelece o propósito de definir requisitos para o sistema da qualidade, no intuito de permitir à organização fornecer, de forma consistente, produtos que vão ao encontro das necessidades dos seus clientes e dos requisitos estatutários/regulamentares relevantes.

Nesse sentido, seguidamente abordam-se os aspetos relacionados com o processo de produção, na vertente do controlo dos dispositivos de monitorização e medição nos SGQ. Uma análise mais pormenorizada de todos os requisitos pode efetuar-se consultando o documento original da norma.

As normas ISO 9000 destinaram-se sempre a uma aplicação de acordo com a natureza da organização, sendo que todas as atividades que afetam a qualidade do produto que fornece têm de ter cobertura total. Se, por exemplo, uma organização for responsável pela transformação das necessidades, expectativas e/ou requisito do cliente num conjunto de características ou especificações de produto ou serviço, então, de acordo com a definição 3.4.4 da ISO 9000:2000, trata-se de "conceção e desenvolvimento de produto" que pode ter impacto direto na satisfação do cliente. Se, por algum motivo, a organização deixa este processo fora do âmbito do SGQ, há o risco, mesmo que o produto "cumpra as especificações", de que não possa ser atingida a satisfação do cliente, porque as especificações poderão não refletir de forma adequada as suas reais necessidades, expectativas e/ou requisitos.

Para o sistema que aqui se pretende desenvolver, vamo-nos debruçar num requisito de máxima importância no processo de produção numa organização no âmbito do SGQ, referenciado na ISO 9001:2000, pela CLÁUSULA 7.6 "Controlo dos dispositivos de monitorização e de medição". Adicionalmente a CLÁUSULA 6.4 – "Ambiente de Trabalho" que define o conjunto de condições sob as quais o trabalho é executado. Vamos abordar esta última apenas no seu essencial, tendo em conta os nossos objetivos.

CLÁUSULA 6.4 - AMBIENTE DE TRABALHO

A ISO 9000:2000 define "ambiente de trabalho" como (definição 3.3.4 da norma) o conjunto de condições sob as quais o trabalho é executado.

Nota: Incluem-se nestas condições os fatores físicos, sociais, psicológicos e ambientais (tais como a temperatura), os sistemas de reconhecimento, os aspetos ergonómicos e a composição do ar atmosférico.

Os aspetos do ambiente de trabalho que podem afetar a qualidade do produto podem variar consideravelmente, dependendo da natureza das atividades da organização. Numa situação de manufatura convencional, podem incluir: níveis de ruído, limpeza/arrumação, vibração, iluminação, frequências de oscilação, temperatura, poeira e humidade.

CLÁUSULA 7.6 - CONTROLO DOS DISPOSITIVOS DE MONITORIZAÇÃO E MEDIÇÃO

O objetivo principal desta cláusula é determinar que qualquer equipamento ou Dispositivos de Monitorização e Medição (DMM) utilizados para monitorizar e/ou medir a conformidade do produto estejam aptos a fornecer resultados válidos.

A conformidade do produto pode ser realizada por um dos métodos seguintes:

- Por observação direta (dispositivos de monitorização), que pode incluir a utilização de dispositivos, como câmaras de vídeo, equipamento de gravação, entre outros, que podem necessitar de manutenção periódica e verificação de funcionamento para assegurar a sua adequação continuada;
- Por medições periódicas das características (dispositivos de medição) do produto ou de parâmetros do processo, utilizando equipamento de medição. Se for este o caso, estes dispositivos devem cumprir os requisitos desta cláusula da norma ou, ainda, quando aplicável, o cumprimento dos requisitos regulamentares.

Sempre que necessário a calibração dos DMM deve ser realizada para fornecer resultados válidos sobre a conformidade do produto em relação aos requisitos especificados. Em alguns casos (e necessariamente para “Processos Especiais” – subcláusula 7.5.2 da norma), isso pode requerer a calibração do equipamento utilizado para medir parâmetros do processo.

Devem ser identificados quais os DMM que servem para inspecionar, medir e ensaiar características consideradas relevantes dos produtos recebidos, em curso de produção ou finais, bem como os DMM que medem parâmetros essenciais para o controlo dos processos com influência na conformidade do produto (desde que os resultados dessas atividades / processos sejam necessários para proporcionar evidência da conformidade do produto com os requisitos aplicáveis). Só é necessário aplicar a confirmação metrológica aos DMM cujos resultados sirvam para proporcionar a evidência da conformidade e tomada de decisão de aceitação/rejeição de produto.

Uma consideração importante é que não é necessário calibrar todos os DMM. A organização deve suportar as suas decisões com base na análise, estudos e considerações relevantes, decorrentes, quer da experiência existente, quer das necessidades dos processos produtivos e resultados finais, como, por exemplo, através de estudos estatísticos. Os custos associados às calibrações só se justificam pela exigência do resultado final.

Os custos inerentes à calibração são normalmente elevados, pelo que a organização deve assegurar a identificação das necessidades de calibração dos DMM para dar confiança às medições e satisfazer os requisitos de qualidade dos seus produtos.

De igual forma, devem, ser contemplados na calibração os programas desenvolvidos e materiais de ensaio, bem como os padrões de referência.

Os DMM selecionados como estando sujeitos a calibração devem estar identificados. Outros equipamentos de medição existentes, que não estão sujeitos a calibração, devem, igualmente, ser identificados por forma a possibilitarem a sua gestão em termos de manutenção.

Para os DMM selecionados é necessário estabelecer como se vai efetuar a confirmação metrológica, responsabilidades associadas e periodicidade, assegurando a rastreabilidade a padrões de medição internacionais ou nacionais.

As calibrações devem ser realizadas por pessoal com competência adequada (subcláusula 6.2.2), em condições ambientais apropriadas, cuja evidência é particularmente relevante no caso de calibrações internas ou externas, quando efetuadas por entidades ou laboratórios não acreditados (conforme a norma ISO / IEC 17025). Informação adicional pode ser consultada na subcláusula 7.4.1. da norma.

Nas calibrações externas, os laboratórios acreditados que prestam o serviço de calibração devem emitir certificados de acordo com o estabelecido na ISO /IEC 17025. Estes devem indicar os valores de comparação com os padrões e a incerteza expandida de calibração.

Nas calibrações internas (e externas, quando realizadas por entidades ou laboratórios não acreditados) é necessário a emissão de documentos que traduzam a atividade desenvolvida, bem como os valores obtidos, incluindo a incerteza da calibração e a rastreabilidade dos DMM utilizados, a padrões internacionais ou nacionais reconhecidos, isto é que os resultados obtidos possam ser relacionados com determinadas referências (geralmente padrões nacionais ou internacionais) por intermédio de uma cadeia ininterrupta de comparações, tendo todas as incertezas determinadas (cadeia de rastreabilidade) [2].

É expectável que a organização evidencie que analisou todo o equipamento e dispositivos de monitorização e de medição que utiliza para verificar ou assegurar a conformidade do produto, e que determinou a necessidade de calibração, de modo a assegurar resultados válidos. Quaisquer reclamações de clientes ou não conformidades do produto, que possam ser atribuídas a problemas associados aos DMM, devem ser investigadas e deve ser criada a ação corretiva apropriada.

2.4. Redes sem Fios

As recomendações do IEEE[1], nomeadamente as recomendações da série IEEE 802.11, são o exemplo mais conhecido para os protocolos de redes sem fio, sendo considerada a existência de quatro grandes grupos:

- **WPAN (Wireless Personal Area Network) –** Neste grupo podem-se encaixar as tecnologias sem fios de pequeno alcance (entre 10 e 100 metros). É um protocolo para redes locais, definido pelo IEEE 802.15, para o endereçamento de redes sem

[1] IEEE - Institute of Electrical and Eletronics Engineers

fio que utilizam dispositivos portáteis ou móveis, tais como, PCs, PDAs, periféricos, telefones sem fios, entre outros;

- **WLAN (Wireless Local Area Network) –** Enquadram-se aqui as tecnologias sem fio destinadas à interligação de redes locais com alcance entre 100 e 300 metros. Este protocolo complementa as redes tradicionais por cabo (par de cobre ou fibra ótica) funcionando como extensão ou alternativa;
- **WMAN (Wireless Metropolitan Area Network) -** Neste grupo tem-se as tecnologias utilizada pelos operadores de telecomunicações que tratam dos acessos de banda larga, para redes em áreas metropolitanas, com alcance aproximado de 6 a 10km, conhecidas também por anel local;
- **WWAN (Wireless Wide Area Network) –** Este grupo engloba as tecnologias voltadas para as redes de longa distância para telecomunicações. É utilizada pelos operadores de telecomunicações para criar as redes de interligação para os serviços de voz e alguns serviços de dados.

A Figura 2-1 mostra as diversas redes sem fios.

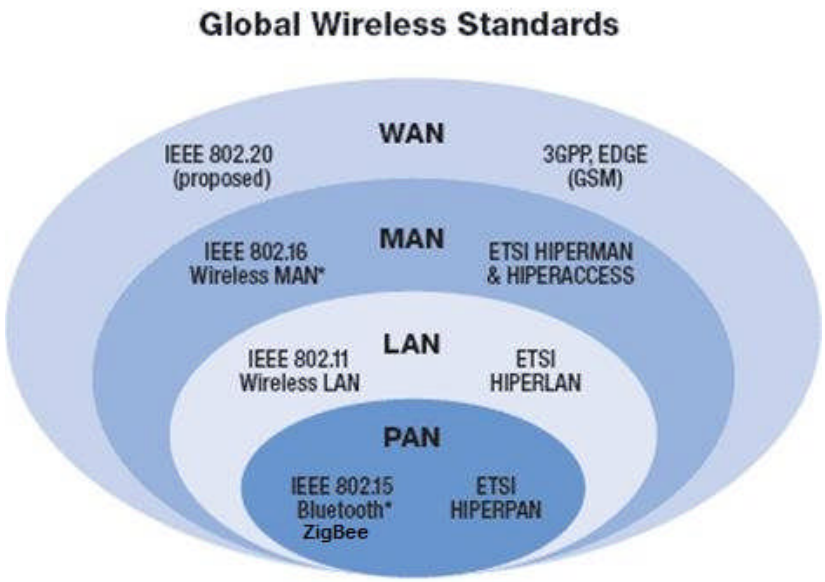

Figura 2-1: Redes sem fios normalizadas

Já existem implementações de redes de comunicação sem fios há alguns anos, como são os casos das redes WLANs, WMANs e WWANs, todas vocacionadas para os utilizadores finais, PME e mesmo grandes empresas, onde o objetivo é a transferência de grandes volumes de dados e voz em altas velocidades.

São poucas as redes sem fios destinadas exclusivamente ao controlo de dispositivos como relés, dispositivos eletromagnéticos, ventilação, aquecimento e motores. De igual forma,

mesmo ao nível doméstico, são poucas as redes sem fio utilizadas no controlo de pequenos eletrodomésticos, brinquedos e na aquisição de dados de sensores, como temperatura, luminosidade, humidade, pressão, entre outras grandezas.

No seio das redes WPAN (Wireless Personal Area Network) existentes, a mais recente e promissora é a que usa o protocolo ZigBee[2] IEEE 802.15.4.

A ZigBee Alliance é quem desenvolve o protocolo ZigBee junto do IEEE, através da associação de várias empresas. Trabalham em conjunto para proporcionar e desenvolver tecnologias para criar um protocolo de baixo consumo de energia, baixo custo, com segurança, fiabilidade, e com funcionamento em rede sem fios baseado numa norma aberta global.

2.4.1. Protocolo ZigBee

O protocolo ZigBee permite comunicações robustas e opera na frequência ISM[3], não requerendo, por isso, qualquer tipo de licenciamento. Abrange a faixa de 2,4 GHz em todo o mundo, 915Mhz na América e 868Mhz na Europa.

As taxas de transferência de dados são de 250kbps, 40kbps e 20kbps para 2,4GHz, 915Mhz e 868Mhz, respetivamente.

As redes com base no protocolo ZigBee oferecem uma excelente imunidade a interferências e dispõem de uma grande capacidade de hospedar milhares de dispositivos ligados na mesma rede (65.000 dispositivos por canal).

O protocolo ZigBee é destinado a aplicações industriais, tendo em conta que as características destas redes, normalmente, não necessitam de grandes velocidades, o que se enquadra perfeitamente nas suas especificações.

O protocolo ZigBee (IEEE 802.15.4) foi planeado e desenvolvido, tendo como objetivo preferencial apresentar as seguintes características:

- Consumo energético baixo e implementação simples, com interfaces de baixo custo;
- Dois estados de funcionamento: "active" para transmissão e receção e "sleep", quando inativo;
- Simplicidade de configuração e redundância de dispositivos (operação segura);
- Densidade elevada de nós suportados pela rede. As camadas PHY[4] e MAC[5] permitem que as redes funcionem com grande número de dispositivos ativos. Este atributo é crítico para aplicações com sensores nas redes de controlo;

[2] ZigBee – Protocolo de comunicação (analogia entre o modo de funcionamento da rede em malha e o modo como as abelhas trabalham e se locomovem)
[3] ISM - Industrial, Scientific and Medical
[4] PHY – Physical - Camada física do protocolo ZigBee

- Protocolo simples que permite a transferência fiável de dados com níveis apropriados de segurança.

A arquitetura do protocolo ZigBee, tal como em outros protocolos, é composta por camadas, formando uma estrutura hierárquica. Cada entidade de serviço fornece uma interface para a camada superior através do ponto de acesso ao serviço *SAP*[6]. *C*ada *SAP* suporta um determinado número de primitivas de serviço para ativar a funcionalidade, que será solicitada pela camada imediatamente superior.

Apesar do protocolo ZigBee se basear no modelo *OSI*[7], que é definido por sete camadas (Figura 2-2), a arquitetura do protocolo ZigBee apenas define as camadas necessárias para atingir um conjunto de funcionalidades desejadas (Figura 2-3).

Figura 2-2: Camadas do modelo OSI

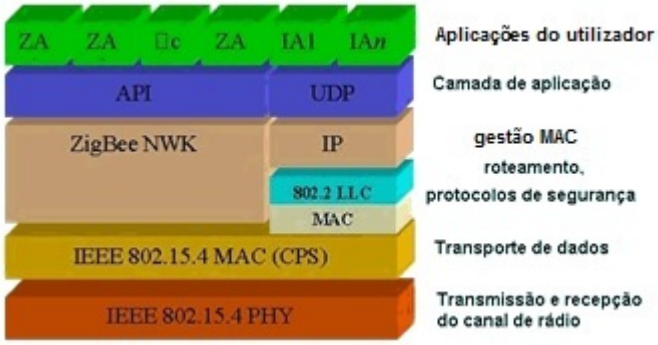

Figura 2-3: Estrutura de camadas do protocolo ZigBee

As duas camadas inferiores, a camada física (*PHY*) e a camada de controlo de acesso ao meio (*MAC*), foram definidas pelas norma IEEE 802.15.4. As restantes camadas de rede foram concebidas especificamente para o protocolo ZigBee. Tais camadas são a camada de rede (*NWK*) e o *Framework* para a camada de aplicação (*API)*. Nesta última camada estão incluídas a subcamada de suporte aplicacional (*APS*), que asseguram a ligação da camada de aplicação e o objeto de dispositivo ZigBee (*ZDO - ZigBee Device Object*).

Seguidamente descreve-se as camadas do protocolo ZigBee.

- ***IEEE 802.15.4 PHY*** - A camada física (PHY) foi projetada para acomodar as necessidades de interfaces de baixo custo, permitindo níveis elevados de integração. O uso da técnica de transmissão de Sequência Direta[8] permite que os equipamentos sejam muito simples, possibilitando implementações com custos reduzidos.

[5] MAC – Medium Acess Control – camada de controlo de acesso ao meio
[6] SAP - Service Access Point
[7] OSI - Open Systems Interconnection
[8] DSSS - Transmissão de sequencia direta (Direct-sequence Spread Spectrum)

- ***IEEE 802.15.4 MAC -*** A camada de controlo de acesso ao meio foi prevista para permitir topologias múltiplas com baixa complexidade, onde a gestão de energia, por exemplo, não requer modos de operação complexos. Esta camada também permite que um dispositivo com funcionalidade reduzida (RDF)[9] opere na rede sem a necessidade de grandes quantidades de memória disponíveis, podendo controlar também um grande número de dispositivos sem a necessidade de colocá-los "em espera", como ocorre em algumas tecnologias sem fio.
- ***ZigBee NWK*** - A camada de rede utiliza um algoritmo que permite implementações da pilha de protocolos, visando equilibrar os custos das unidades em aplicações específicas. É o caso do consumo das baterias, procurando produzir soluções específicas para a aplicação, com boa relação preço qualidade.
- ***API*** – É formada pela subcamada de suporte à aplicação, que fornece uma interface entre a camada de rede e a camada de aplicação através de um conjunto geral de serviços que são usados pelo ZDO[10] e os objetos da aplicação definidos pelo fabricante. Os serviços fornecidos no suporte à aplicação são:
  - *Descovery* - este serviço verifica a existência de outros pontos ativos na área de alcance do dispositivo, para serviços de troca de dados;
  - *Binding* - este serviço interliga os vários dispositivos, tendo em conta as necessidades e serviços.

2.4.2. Topologia da Rede

Com uma vasta área de possibilidade de aplicação, desde o controlo industrial à automação de residências (domótica), o protocolo ZigBee possui determinadas características que o tornam absolutamente distinto dos restantes. Algumas dessas características já foram referenciadas anteriormente e sustentam os motivos que levaram à sua criação, nomeadamente no que se refere à estrutura da rede, admitindo diferentes topologias da rede:

Estrela (*star*), malha (*mesh*) ou árvore (*cluster tree*), permitindo o estabelecimento de redes de nós "*ad-hoc*[11]".

O ZigBee possui um tempo de ligação à rede menor que os outros protocolos e apresenta maior rapidez na passagem do modo *standby* a ativo. O ZigBee apresenta também uma latência baixa (*low latency*[12]), características estas que dão grande vantagem na estruturação das redes.

O protocolo disponibiliza ainda suporte para duas classes de dispositivos físicos (definidos na norma IEEE 802.15.4), podendo ambos coexistir numa mesma rede:

[9] RDF – Dispositivo com funcionalidade reduzida (Reduce Function Device)
[10] ZDO - ZigBee Device Object
[11] Ad-Hoc - rede espontânea
[12] low latency – tempo necessário de espera entre um pedido e o seu atendimento.

- Full Function Device (FFD) – pode funcionar em qualquer que seja a topologia da rede, desempenhando a função de coordenador da rede e consequentemente ter acesso a todos os outros dispositivos. Por isso, trata-se de dispositivos de construção mais complexa;
- Reduced Function Device (RFD) – é limitado a uma configuração com topologia em estrela, não podendo atuar como coordenador da rede. Pode apenas comunicar com um coordenador de rede. São dispositivos de construção mais simples;

Às duas classes anteriores de dispositivos físicos correspondem três tipos de dispositivos lógicos: Coordinator, Router e Endpoint como se descreve na Tabela 1 – Classes e tipos de dispositivos lógicos.

Dispositivo	Coordinator	Router	Endpoint
Tipo de dispositivo físico associado (IEEE 802.15.4)	FFD	FFD	RFD ou FFD
Função	Forma a rede, atribui endereços, suporta binding table. Existe apenas um por rede.	Permite que mais nós se juntem à rede, ao aumentar o seu alcance físico. Pode também efetuar funções de controlo ou monitorização. A sua existência é opcional.	Efetua ação de controlo ou monitorização através de dispositivo que lhe esteja associado (sensor, controlador, atuador…).

Tabela 1 – Classes e tipos de dispositivos lógicos

2.4.2.1. Formação da Rede

A formação das redes ZigBee, pode ser comparada com a atividade das abelhas. De facto, as abelhas que vivem em colmeia voam em ziguezague e, dessa forma, durante um voo de trabalho em busca de néctar, trocam informações com outros membros da colmeia sobre distância, direção e localização de alimentos.

Também uma Malha ZigBee dispõe de vários caminhos possíveis entre os nós da rede para a passagem da informação, sendo possível, assim, eliminar falhas se um nó estiver inoperante, simplesmente mudando o percurso da informação.

Uma vez que este protocolo perspetiva a rede de uma forma espontânea, não existe uma topologia pré determinada, nem um controlo obrigatoriamente centralizado.

É de salientar que, apesar desta característica dinâmica de configuração da rede, em qualquer uma das topologias ZigBee o coordenador (*Coordinator*) é o dispositivo responsável por iniciar a rede.

É de salientar igualmente que, apesar de existirem diferentes formações topológicas, é possível que estas possam coexistir numa mesma rede.

De modo a facilitar a descrição das diferentes topologias de rede ZigBee é importante conhecer alguns conceitos utilizados neste tipo de redes.

- **Coordenador ZigBee** é o coordenador (*coordinator*) geral de toda a rede. Trata-se de um dispositivo com recursos completos, conforme a norma IEEE 802.15.4- (FFD[13])
- **Roteador ZigBee** o roteador (*router*) é o elemento da rede ZigBee que pode agir como coordenador no seu espaço particular de operação, sendo capaz de gerir (rotear) mensagens entre dispositivos e suportar associações. Trata-se de um dispositivo com recursos completos, conforme a norma IEEE 802.15.4- (FFD).
- **Escravo** ou **dispositivo final** (*Endpoint*) ZigBee é o elemento da rede ZigBee que não é coordenador ou roteador e encontra-se nos extremos da rede. É o responsável pela comunicação com os dispositivos de aquisição e atuação com o meio físico. Trata-se de um dispositivo com recursos reduzidos, conforme a norma IEEE 802.15.4- (FFD).

Seguidamente descrevem-se as topologias de rede que podem ser utilizadas nas redes ZigBee.

Topologia em estrela

Na topologia em estrela (*Star*) é ao coordenador que cabe iniciar a rede, assim como ter todo o controlo da mesma, assumindo um papel central e de comunicação direta com todos os dispositivos escravos ou dispositivos finais. Assim, toda a informação que circula na rede passa pelo nó coordenador (Figura 2-4).

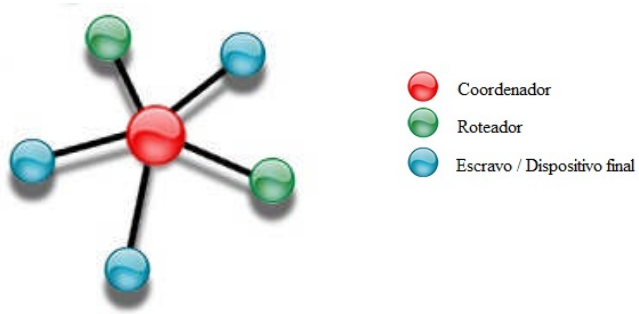

Figura 2-4: Topologia estrela

Esta topologia deve ser implementada em locais que não ofereçam muitos obstáculos à transmissão e receção.

[13] FFD - Full Function Device

Topologia em árvore

Nesta estrutura de rede (*Cluster Tree*) o coordenador assume o papel de "nó nuclear" da rede, originando uma distribuição de dados e mensagens de controlo, numa estrutura hierárquica, em que o Coordenador apenas se liga ao roteador (*router*) e este é que assume a ligação com os escravos. Esta topologia está ilustrada na Figura 2-5.

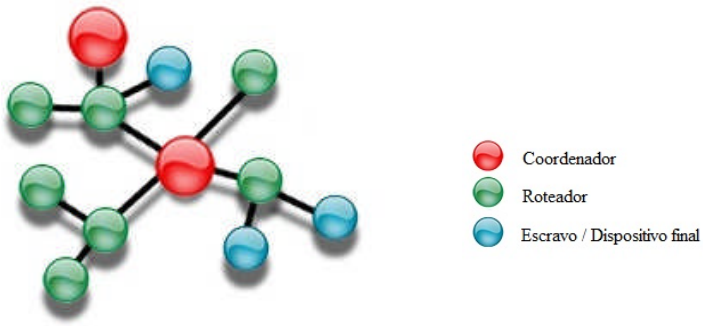

Figura 2-5: Topologia em árvore

Topologia em malha

Numa topologia em malha (*Mesh*) os dispositivos do tipo FFD (Coordenadores/Roteadores) são livres de comunicar com outro dispositivo FFD. Isto permite a expansão física da rede (maior alcance) quando necessária. O coordenador regista toda a entrada e saída de dispositivos, mas não assume um papel tão preponderante em termos de fluxo de informação como na configuração em estrela, podendo deixar parte do controlo ao cuidado dos roteadores. A Figura 2-6 mostra a configuração desta topologia.

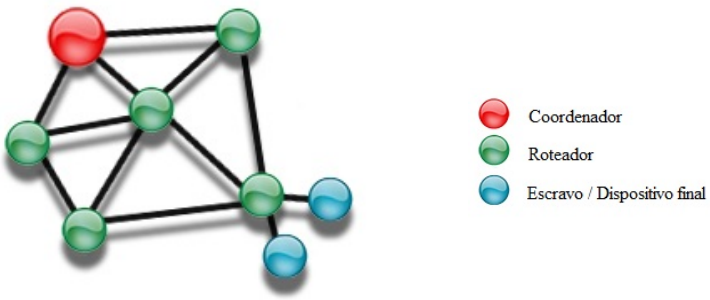

Figura 2-6: Topologia em malha

Esta rede auto-organiza-se de forma a otimizar o tráfego de dados, podendo abranger áreas geográficas relativamente extensas como, por exemplo, um prédio de grandes dimensões.

2.4.3. Modo de Operação da Rede

Devido ao protocolo ZigBee ser relativamente simples, fica facilitado o desenvolvimento do código o que contribui para os custos reduzidos no desenvolvimento de aplicações. Outro fator que simplifica o desenvolvimento de aplicações é o facto de haver apenas dois modos de funcionamento (*active* e *sleep*), tanto para envio como para receção.

Adicionalmente o protocolo possui também um tempo reduzido de ligação à rede e uma transição rápida entre modos de funcionamento, fazendo com que o ZigBee apresente uma latência baixa.

Considerando as características atrás referidas, os módulos podem ser configurados de modo a operarem em modo de sinalização (*beaconing*), em que os nós ZigBee roteadores transmitem periodicamente sinalização (*beacons*) a confirmar a sua presença, como nós que formam a rede. Os restantes nós só necessitam de estar ativos no momento da sinalização. Este mecanismo permite mantê-los no modo *sleep* entre sinalizações, com evidente vantagem em termos de consumo energético, diminuindo o seu fator de serviço (*duty cycle*), e consequentemente, aumentará a autonomia da fonte de energia (bateria/pilha) a que possam estar ligados.

O intervalo entre sinalizações pode variar entre os 15.36ms e os 251.65s, para uma taxa de transmissão de 250kbit/s. No entanto, há que ter em conta que a operação com fator de serviço reduzido, ou seja, intervalos prolongados entre sinalizações, requer uma temporização de elevada precisão. Este aspeto origina alguns problemas no funcionamento em rede dos dispositivos, requerendo maior cuidado na sua escolha, podendo, por essa razão, não se enquadrar em termos de custos.

No modo de não sinalização (*non-beaconing)*, a maioria dos dispositivos mantém os seus recetores permanentemente ativos, sendo o consumo energético mais significativo. Normalmente neste modo são utilizadas fontes de alimentação permanentes ou unidades de alimentação mais robustas e com maior autonomia.

2.4.3.1. Tramas do Protocolo ZigBee

Após a análise da arquitetura, topologia e operação protocolar do ZigBee vão-se analisar resumidamente os tipos de tramas utilizadas no protocolo que se podem observar, de forma simples, no esquema da Figura 2-7.

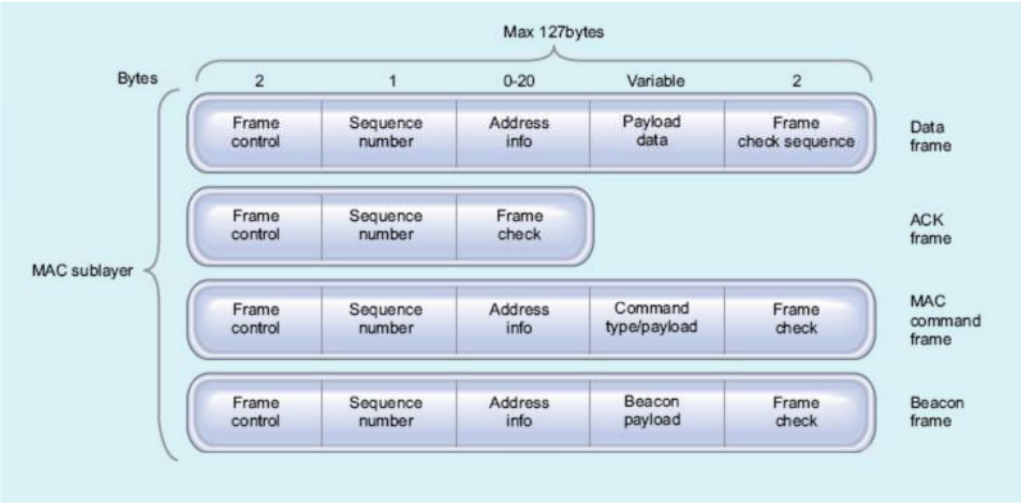

Figura 2-7: Tramas do comando MAC

Existem quatro tipos de Tramas:

- ***Tramas de comando MAC*** (*MAC command*) – Estas tramas são utilizadas para efetuar o controlo dos nós clientes;
- ***Tramas de dados*** – Estas tramas são as que mais interessam ao utilizador, pois são usadas para todo o tipo de transferência de dados. Podem ter até 104 bytes e estão numeradas, a fim de manter alguma fiabilidade na comunicação, pois a existência de uma sequência de *frame-check* permite assegurar uma transmissão fiável e sem erros;
- ***Tramas de reconhecimento*** *(acknowledgement -ACK*) – São utilizadas para confirmar a receção bem sucedida de pacotes. O *acknowledgement* faz-se no tempo livre de comunicações existente entre o envio de tramas;
- ***Tramas de sinalização*** (*beacon*) – são utilizadas pelos ZigBee coordenador e roteador para efetuar a transmissão de *beacons*.

2.5. Sistemas de Controlo e Monitorização Existentes

No mercado é fácil encontrar uma gama variada de sistemas que podem ser utilizados em função das necessidades. No entanto, são na generalidade sistemas fechados, possibilitando apenas algumas configurações por parte do utilizador. A maioria são dispositivos acoplados por intermédio de cabos, necessitando, para tal, de técnicos especializados, quer ao nível da montagem, quer ao nível da manutenção ao longo da sua vida útil, tornando-se assim bastante onerosos e dependentes.

2.5.1. Características Gerais

Na sua maioria os sistemas de monitorização e controlo de temperaturas, humidade e luminosidade baseiam-se no mesmo princípio. Aquisição de dados faz-se por intermédio de sondas (ex: sondas de temperatura, humidade, entre outras), ligadas geralmente por cabo a um computador ou autómato que acondiciona os dados recolhidos e os processa. Os dados podem depois ser observados e analisados pelos operadores, sabendo-se dessa forma o estado de conservação e as condições ambientais de trabalho. Adicionalmente os sistemas permitem o registo e tratamento desses dados.

Os sistemas podem existir isolados ou constituírem sistemas complexos integrados e inteligentes e utilizarem tecnologias diversificadas em função da sua aplicação. Podem utilizar sensores analógicos ou digitais, para medir apenas uma ou diversas grandezas de natureza diferente.

A titulo de exemplo serão aqui apresentados alguns sistemas comerciais, pois a gama de opções é muito vasta.

2.5.2. Exemplos de Sistemas

a) Monitorização de temperatura via WEB[14].

O HWg-STE do fabricante HW group S.R.O. é um termómetro com interface IP[15] que permite a monitorização remota da temperatura/humidade através da WEB. Possui um método de alerta por correio eletrónico de SNMP[16]-Trap, sempre que os limites definidos forem ultrapassados. Fornecido com aplicação para Windows, permite a criação de gráficos e a exportação dos dados para folha de cálculo. A Figura 2-8 representa o sistema HWg-STE.

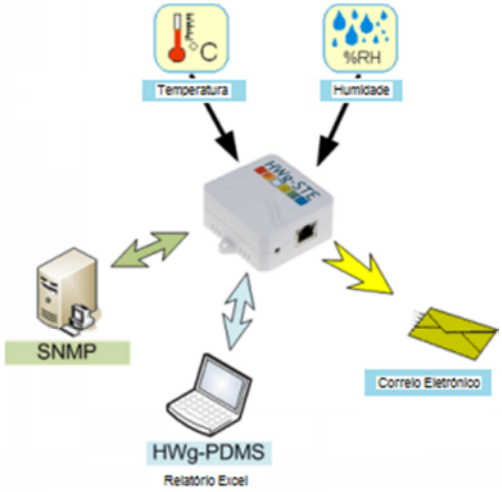

Figura 2-8: Sistema HWg-STE

Este sistema apresenta diversas potencialidades, das quais se destacam as seguintes:

- Alerta por *E-mail* (correio eletrónico) no caso de se verificar anomalias dos equipamentos monitorizados;
- Alerta por sobreaquecimento de equipamentos;
- Alerta pelo aumento de temperatura numa zona;
- Instalação rápida e fácil (suporta DHCP);
- Ligação a sistema de monitorização SNMP;

Este sistema de monitorização, está vocacionado essencialmente para espaços habitacionais e escritórios. Monitoriza temperaturas e humidades podendo ter diversas aplicações, como a monitorização de compartimentos climatizados ou dar informação sobre a temperatura no exterior. Pode estar associado diretamente a equipamentos de climatização.

[14] Web, também conhecida por WWW – World Wide Web, Rede de alcance mundial.
[15] IP – Internet Protocol
[16] SNMP - Simple Network Management Protocol

b) Monitorização da temperatura em sistemas de Refrigeração/Congelação.

O Poseidon 2250 do fabricante HW group S.R.O., pode monitorizar as condições de armazenamento de alimentos ou produtos farmacêuticos e permite gerir até 30 sensores. Os sensores podem ser analógicos e medir grandezas, tais como: temperatura, humidade e pressão, suporta também entradas digitais, como se representa esquematicamente na Figura 2-9.

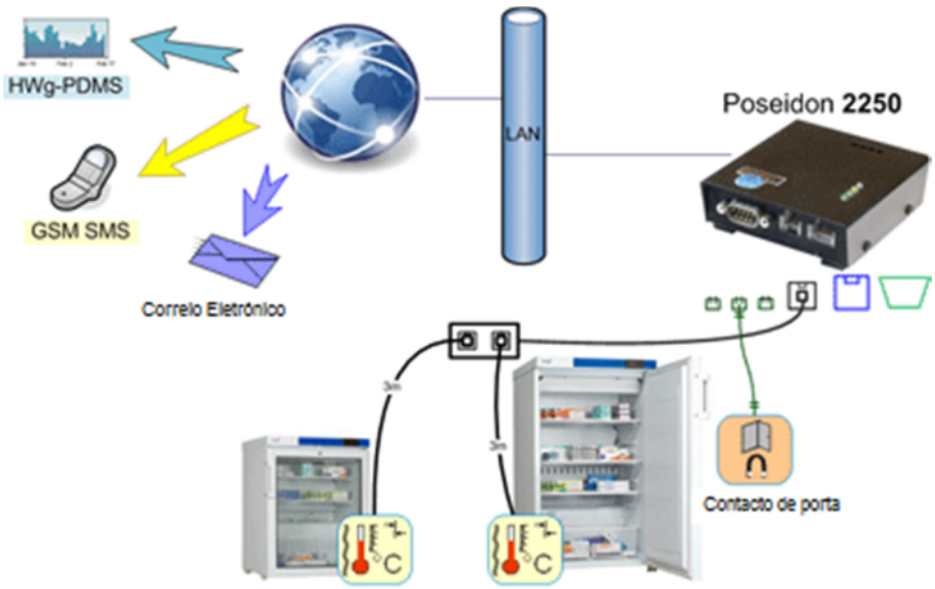

Figura 2-9: Sistema Poseidon 2250

Apresentam-se seguidamente algumas características principais:

- Os valores recolhidos podem ser guardados na memória interna e tem capacidade para armazenar 40.000 registos;
- Possui um modo de alerta por E-mail ou SMS[17], sempre que o estado de um sensor exceda os limites estabelecidos;
- Permite a ligação a sistemas de monitorização SNMP;
- Suporta a ligação a um modem GSM[18] e envia alertas SMS, em caso de desvio ou erro na leitura de um sensor.

c) Sistema de monitorização e controlo integrado de temperatura

O Cryocell / DD86-750 P do fabricante Norconcessus (Figura 2-10) é um sistema composto por uma câmara frigorífica vertical de duas portas e um microprocessador ao qual estão associados todos os sensores e atuadores.

[17] SMS – Short Message Service
[18] GSM - Global System for Mobile Communications

O microprocessador promove a monitorização e ajuste contínuos das condições de operação da câmara, assegurando o controlo otimizado da temperatura e minimizando o consumo de energia.

Figura 2-10: Sistema Cryocell /DD86-750

Este sistema possui algumas características interessantes, das quais se destacam:

- Possui uma função de segurança com proteção para arranque, "*Set point*" e definições de alarmes;
- Possibilita a programação de alarmes independentes para temperatura máxima e mínima;
- Permite ajustes de temperaturas em intervalos de 1ºC:
- Possui uma função de proteção do compressor, que monitoriza e ajusta automaticamente as condições do equipamento, aumentando a sua vida útil.
- Permite carga automática e contínua da bateria de *backup*, que possibilita o funcionamento integral do sistema durante falhas de energia;
- Monitoriza o estado da grelha de condensados;
- Monitorização do sistema de *backup* de CO_2 [19], isto é, o sistema permite identificar a falta de fornecimento de energia elétrica e começa automaticamente o fornecimento de CO_2 líquido ou nitrogénio líquido dentro da câmara de refrigeração, mantendo-a na temperatura previamente selecionada pelo utilizador.

[19] CO_2 – Dióxido de Carbono

2.6. Considerações

Pode-se concluir que as soluções para a monitorização de ambientes controlados ou de equipamentos de conservação e armazenamento de produtos, quer alimentares, quer para a indústria farmacêutica/saúde, são vastas.

Alguns dos equipamentos que existem no mercado permitem apenas a monitorização de um número limitado de grandezas, como o referido na secção 2.5.2 - a), o que restringe bastante a sua aplicabilidade a processos mais variados.

Outros são bastante mais versáteis, com a possibilidade de medição de grande número de grandezas. No entanto, são pouco atrativos do ponto de vista económico e, normalmente, apresentam uma grande dependência técnica, no que se refere à instalação, conservação e ampliação. Têm montagens complexas e baseiam-se em redes de cabos.

Por outro lado existem sistemas muito específicos, vocacionados unicamente para determinadas atividades produtivas ou económicas, com características extremamente rígidas.

Capítulo 3 - Implementação de um Sistema de Aquisição de Dados

3.1. Introdução

Neste capítulo descreve-se a tecnologia utilizada e apresentam-se as razões que levaram à sua escolha para o desenvolvimento deste projeto. Serão ainda descritas todas as características relevantes dos dispositivos utilizados. Considerando que, dentro da mesma tecnologia, a gama de soluções é bastante vasta, procurou-se focalizar a atenção nas soluções mais viáveis para se atingirem os objetivos pretendidos.

A indústria laboratorial e alimentar são os potenciais utilizadores do sistema de aquisição de dados "TLab" desenvolvido no âmbito deste trabalho. As necessidades específicas dessas indústrias determinaram algumas decisões com vista a atingir os objetivos deste projeto. Essas decisões determinaram a elaboração de um protótipo funcional, com grande capacidade adaptativa, de fácil instalação e conceção e, acima de tudo, de baixo custo, sem nunca esquecer a qualidade e fiabilidade dos seus resultados.

Nesse sentido escolheu-se o micro controlador Arduino Mega – Atmega 2560, devido às suas características, que iremos abordar neste capítulo, assim como a existência de algumas placas que facilitam a integração dos módulos de comunicação e dos sensores de aquisição.

O Arduino Mega disponibiliza um grande número de entradas e saídas, o que é conveniente numa fase de desenvolvimento, considerando futuras ampliações. No entanto, um qualquer outro modelo do Arduino seria aplicável e, de uma forma geral, até mais económico.

Para a comunicação sem fios selecionaram-se os módulos XBee da série II e XBee Pro. Genericamente estes módulos são iguais, diferindo apenas na potência do sinal de transmissão.

O protocolo ZigBee utilizado por estes módulos permite, só por si, a criação de uma rede de comunicação de dados. No entanto, para que se possa tirar proveito da sua utilização, no que se refere à visualização e gestão de dados que circulam na rede, devem ser implementadas aplicações, que disponibilizem o acesso através da internet utilizando um navegador Web ou outro aplicativo, assim como ter acesso aos dados através de uma rede local. Porém, não faz parte do âmbito da especificação do ZigBee a definição de tais aplicações, nem de tais dispositivos. Contudo, o protocolo define uma estrutura de trabalho (*framework*) cujo objetivo é facilitar o desenvolvimento de aplicações, além de incentivar um certo nível de padronização das mesmas.

Por outro lado, no âmbito deste projeto desenvolveram-se aplicações para a utilização efetiva da tecnologia ZigBee. Para isso, criou-se o "Interface TLab" do qual fazem parte alguns elementos sensoriais que serão integrados nos equipamentos industriais.

Este dispositivo deverá, por sua vez, conter um *software* capaz de empregar a tecnologia ZigBee de forma a disponibilizar os benefícios desejados do sistema. A união de *hardware* e *software*, feita de acordo com esta premissa, constituirá a aplicação que irá gerir o fluxo de dados proveniente dos sensores, assim como da própria rede. Os seus elementos, nas suas componentes de *hardware* e *software*, serão descritos detalhadamente mais à frente neste capítulo.

Serão também descritos os diversos tipos de sensores utilizados neste projeto.

Numa primeira fase, fizeram-se algumas tentativas de utilização de sensores analógicos simples como PT100 para medição de temperaturas e LDR para a luminosidade. No entanto, devido às suas características, para serem integradas com o Arduino necessitavam de eletrónica adicional chegando mesmo a ser desenvolvidas algumas placas de acoplamento. Não se obtiveram resultados satisfatórios, pois afastavam-se do objetivo proposto inicialmente de simplicidade da instalação.

Optou-se então por selecionar alguns sensores digitais de interface direto com o Arduino. A escolha determinou a utilização dos seguintes sensores: Sensores de temperatura DS18B20, sensores de luminosidade BH1750FVI e o sensor duplo DHT11, com capacidade para medir temperatura e humidade simultaneamente.

3.2. Razões Para a Escolha do ARDUINO

Este projeto consiste no desenvolvimento de um sistema de monitorização de temperatura e outras grandezas e é designado por "TLab". Possui alguns sensores de diferentes tecnologias, colocados em equipamentos e/ou espaços de trabalho que fornecem informação a um

dispositivo de processamento. Este dispositivo é constituído por uma placa de *hardware* que suporta entradas e saídas, tem capacidade ainda de efetuar algum processamento baseado numa linguagem de programação conhecida.

O TLab possibilita a visualização da informação recolhida, recorrendo a aplicações gráficas. OTLab visa monitorizar as grandezas envolvidas, sendo elas, a temperatura, a humidade relativa e os níveis de luminosidade do espaço envolvente, assim como a aquisição de um conjunto de temperaturas de equipamentos específicos.

Considerando os requisitos do sistema e por outro lado, a versatilidade e aplicabilidade pretendidas, optou-se pela escolha da plataforma "Arduino".

O grande potencial do Arduíno reside no facto de ser um sistema "Aberto", tanto em *hardware* como em *software*. Por essa razão a informação disponível é vasta e em constante evolução.

Para o desenvolvimento de aplicações, existem plataformas de programação para os diversos sistemas operativos mais comuns (Windows, Linux e MAC OS). O *hardware* está disponível em grande variedade e especificidade e com custos relativamente baixos. Esta característica leva a que o Arduino se apresente como uma solução interessante para o desenvolvimento experimental.

Muito embora apresente algumas fragilidades no que concerne à sua aplicabilidade industrial, tem-se revelado de grande eficácia quando tomadas escolhas adequadas no desenvolvimento das aplicações. As suas características de comunicação e interligação permitem o acoplamento de diferentes tipos de sensores com diferentes tecnologias, graças às bibliotecas de programação[20] já concebidas, que se podem obter facilmente e usá-las de acordo com a aplicação. Outra grande vantagem que o Arduíno apresenta é a diversidade de protocolos com que pode comunicar, o que é interessante para o desenvolvimento de projetos de grande complexidade, não estando, deste modo, dependentes de fabricantes ou de sistemas comerciais já desenvolvidos e de custos, normalmente, elevados.

O Arduíno disponibiliza um Interface de Desenvolvimento (IDE[21]) muito simples onde se faz toda a programação, sendo esta uma aplicação multiplataforma escrita em Java. Com a biblioteca "*Wiring*" é possível programar nas linguagens C e C++. Com as funções *setup*() e *loop*() é possível criar várias operações de entrada e saída. Existe alguma sintaxe própria de funções já concebidas para simplificar a programação. Também neste interface IDE, e após a programação concluída, pode-se fazer a transferência do programa para o Arduíno (*upload*) através de um cabo USB. Seguidamente podem realizar-se testes de funcionamento, obtendo-se resultados imediatos dessa programação.

Como ferramenta é usualmente associado à filosofia de *Physical Computing*, ou seja, ao conceito que engloba a criação de sistemas físicos através do uso de *Software* e *Hardware* capazes de responder a "estímulos" vindos do mundo real.

[20] Bibliotecas de programação - Na literatura anglo-saxónica é conhecida por "Library"
[21] IDE - Integrated Development Environment - Interface de Desenvolvimento

Através de uma aprendizagem simples e dedicada, o estudo do Arduino aliado à imaginação, leva à compreensão de uma ferramenta importante de desenvolvimento que abre portas à criação de inúmeras aplicações.

3.3. Hardware do Arduino

A tecnologia que aqui se descreve teve como origem a ousadia de um professor de artes Italiano (Prof. Massimo Banzi), que pretendia que os seus alunos desenvolvessem projetos de arte com interatividade. Assim, em conjunto com um seu aluno (David Mellis, ano 2005) desenvolveram uma plataforma extremamente económica que lhes permitia uma programação simples, uma vez que se tratava de alunos de artes, mas com grandes potencialidades.

O facto de ser uma plataforma aberta, através da sua divulgação na Web permitiu um rápido desenvolvimento e constantes melhorias, assim como, ao nível programação, surgiram soluções e aplicações partilhadas por todos. Desta forma, gerou-se uma onda de partilha de conhecimentos e experiências, que aliado ao facto de se tratar de código aberto[22], deu confiança aos utilizadores, pois não estariam dependentes de fabricantes para desenvolverem os seus próprios sistemas.

Desde 2008, com a venda de milhares de placas de controlo a estudantes e projetistas do mundo inteiro, apareceram inúmeras aplicações que vão desde luminárias inteligentes até aviões auto pilotados.

Dada a existências de diversas plataformas disponíveis no mercado, com várias alternativas no que diz respeito ao *hardware*, são as necessidades das aplicações e os recursos necessários que determinam a escolha da placa. Os critérios de decisão baseiam-se no número de entradas e saídas, necessidade de portas de comunicação série, portas analógicas, tamanho físico, consumo energético, capacidade de memória onde é gravado o programa do utilizador (*memory flash*) que é executado ciclicamente (RAM[23]) e tamanho do espaço de memória não volátil (EEPROM[24]), que armazena a informação mesmo após a falha de energia de alimentação.

Considerando o referido anteriormente, a escolha do *hardware* recaiu sobre as plataformas do Arduino Uno e Mega que apresentam características mais do que suficientes para o projeto, como se descreve adiante. No entanto, para além destas plataformas, seguidamente também se descrevem outras, pois no futuro poderão ser integradas em posteriores expansões deste projeto.

[22] Codigo Aberto – Na literatura anglo-saxónica é conhecida por "*Open-Source*"
[23] RAM – Random Access Memory - Memória de acesso aleatório
[24] EEPROM - Electrically-Erasable Programmable Read-Only Memory

3.3.1. Arduino Uno

A plataforma Arduino Uno é uma placa baseada no microcontrolador ATmega 168 ou ATmega328 . Possui 14 pinos de entradas / saídas digitais, dos quais 6 podem ser usados como saídas PWM[25], 6 entradas analógicas, um cristal oscilador de 16 MHz, uma ligação USB[26], uma ligação para alimentação, uma ligação ICSP[27] e um botão de *reset*. A placa dispõe de um vasto conjunto de dispositivos de apoio ao microcontrolador. Para começar a utilizar o Arduino Uno basta ligá-lo a um computador através um cabo USB ou alimentá-lo através de uma fonte de alimentação DC[28] ou bateria. No manual técnico do fabricante podem-se consultar mais especificações do dispositivo [3].

Na Figura 3-1 está representada a plataforma do Arduino Uno, onde se identificam os componentes principais e os pinos de ligação com o exterior.

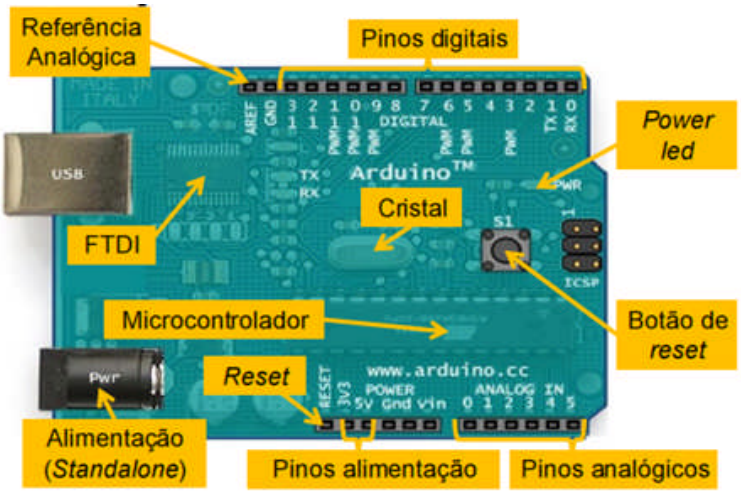

Figura 3-1: Arduino UNO

A plataforma "*Uno*" difere de todas as antecessoras na medida em que não utiliza o *chip* conversor USB-para-série FTDI[29]. Em vez disso, utiliza o Atmega16U2 (Atmega8U2 até à versão R2) programado como um conversor USB para série. Na revisão 2 a plataforma Uno tem uma resistência que efetua a ligação do HWB 8U2 à massa, tornando-a mais fácil de colocar em modo DFU[30]. Este modo permite, de forma fácil, uma atualização do *firmware* da placa.

Na revisão 3, para além da introdução de mais alguns pinos de utilização específica, tais como o SDA[31] e SCL[32], destinados a comunicação por BUS, foi ainda introduzida a possibilidade

[25] PWM - Pulse-width Modulation
[26] USB - Universal Serial BUS
[27] ICSP - In Circuit Serial Programming
[28] DC - Direct current,
[29] FTDI – Future Technology Devices – conversores RS232 para USB
[30] DFU - Device Firmware Update
[31] SDA – Serial Data Line

de referenciar valores de tensão para dispositivos alimentados pela própria plataforma. Esta alteração prende-se com a utilização do microprocessador Atmega 16U2 que substitui o 8U2.

No que se refere à comunicação, o Arduino Uno apresenta uma facilidade de comunicação podendo, através da porta série UART[33] TTL[34] (5V), comunicar facilmente com um computador, com outro Arduino ou com outro microcontrolador. O ATmega168 disponibiliza esta comunicação através dos pinos RX / TX e é possível ligar o Arduino a um computador, pois o *software* do integrado disponibiliza uma porta série virtual.

3.3.2. Arduino Mega

O Arduino Mega 2560 é uma atualização do Arduino Mega 1280, em que a principal diferença é o tamanho da memória. Esta atualização baseia-se no microcontrolador ATmega 2560 [4]. É composto por 54 pinos digitais de entrada/saída, 16 entradas analógicas, 4 portas de comunicação série (UARTs), um oscilador de cristal de 16 MHz, uma ligação USB, uma entrada de alimentação, uma ligação ICSP e um botão de *reset*, como se representa na Figura 3-2. Contem portanto os dispositivos essenciais para suportar o microcontrolador. Conforme as versões antecessoras, basta um cabo de ligação USB e o *software* de desenvolvimento para poder utilizar-se. É de referir que a capacidade de memória é o dobro da versão anterior, passando a dispor de 256Kb. Esta placa é compatível com a maioria dos *shields* (ver Item 3.3.8) existentes para o Arduíno.

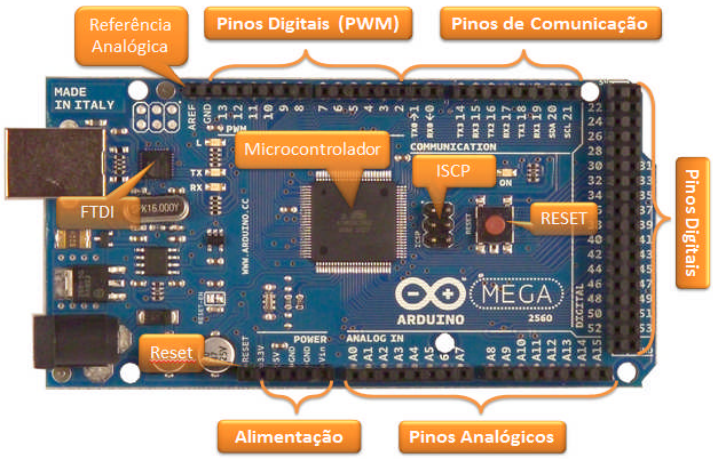

Figura 3-2: Arduino Mega

Relativamente aos portos digitais, alguns têm aplicações adicionais, pois 14 desses pinos podem ser usados como saídas de sinal PWM. As 4 UART portas de comunicação série com o *hardware* que se dividem em: i) portas Série: pino 0 (RX) e pino1 (TX); Série 1: pino 19 (RX) e pino 18 (TX); Série 2: pino 17 (RX) e pino 16 (TX); Série 3: pino 15 (RX) e pino 14 (TX). Usados para receber (RX) e transmitir (TX) dados série TTL. Os pinos 0 e 1 estão

[32] SCL – Serial Clock Line
[33] UART – Universal Asynchronous Receiver/Transmitter
[34] TTL - Transistor-Transistor Logic

também ligados aos pinos correspondentes do chip serie FTDI USB-to-TTL. ii) duas para comunicações série síncrona SPI[35] pinos 50 a 53, para comunicar com periféricos de grande velocidade, mas a curta distância, iii) comunicação I^2C[36] que no Arduino é denominado de TWI[37] pinos 20 e 21. Para este tipo de comunicação o Arduino implementa internamente as resistências de *pull-up*.

3.3.3. Arduino FIO

O Arduino FIO foi desenhado e criado especialmente para ser utilizado em comunicações sem fios com os módulos de comunicação XBee. Utiliza um microcontrolador ATmega 328P a 8Mhz e funciona com uma alimentação de 3,3V. É composto por 14 pinos digitais de entrada/saída, em que 6 podem ser usados como saídas PWM, 8 entradas analógicas e um botão de *reset*. Disponibiliza ainda uma ligação para uma bateria de lítio e um circuito de carga através de uma ligação USB. Incorpora um XBee socket na própria placa, não sendo, por isso, necessário qualquer outro acessório para ligação a um módulo XBee (Figura 3-3).

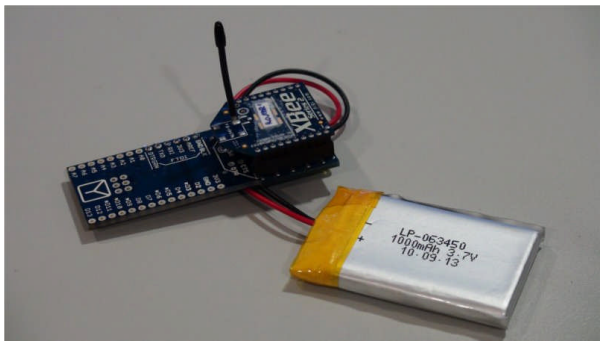

Figura 3-3: Arduino Fio

O Arduino Fio tem igualmente uma série de facilidades de comunicação: com outros dispositivos, a computadores, outros Arduinos ou outros microcontroladores. O microcontrolador fornece uma comunicação série UART TTL, a qual está disponível nos pinos digitais 0 (RX) e 1 (TX).

Tendo em conta as suas características um pouco mais limitadas em alguns aspetos e no seu preço, esta plataforma não foi usada neste projeto. No entanto, e após o desenvolvimento do sistema não será de excluir uma possível integração de placas deste tipo. Pois podem, em determinadas circunstâncias, ser mais adaptáveis devido ao seu tamanho e ao facto de disporem de facilidades no que se refere à alimentação do sistema, nomeadamente no que se refere à sua autonomia.

[35] SPI - Serial Peripheral Interface
[36] I2C - Inter-Integrated Circuit
[37] TWI - Two Wire Interface

3.3.4. Outras Plataformas Arduino

Para além das plataformas já referidas, existem inúmeras outras com aplicações mais específicas. No entanto, a característica de base, a sua programação e a estrutura funcional baseia-se nos mesmos princípios e tecnologia, variando apenas as suas capacidades e especificidade. A título de exemplo, apresentam-se sucintamente no ANEXO 1, mais algumas dessas plataformas.

3.3.5. Funcionamento do Arduino

Como já foi referido, a plataforma Arduino simplifica o processo de trabalhar com microcontroladores, agrupando os detalhes confusos e complexos de programação num pacote fácil de utilizar, oferecendo grande vantagem no desenvolvimento de sistemas físicos de automação.

O Arduino baseia-se numa placa de desenvolvimento que interpreta as variáveis ambientais que nos rodeiam, através de sensores ligados aos terminais de entrada. Depois de processar as variáveis pode controlar ou acionar alguns elementos eletrónicos ligados aos terminais de saída (Figura 3-4).

Figura 3-4: Esquema funcional do Arduino

O Arduino possui um microcontrolador, desenhado e construído de forma a integrar diversos componentes num único circuito integrado, evitando, assim, a necessidade de adicionar componentes externos ao microcontrolador. Esta abordagem é diferente dos microprocessadores clássicos, que normalmente são constituídos por um circuito integrado com a capacidade de executar determinadas instruções e os componentes que garantem a sua funcionalidade são externos ao processador, estando interligados pelos barramentos como se pode ver na Figura 3-5. Esta abordagem clássica tem as desvantagens inerentes à comunicação, sendo a sua velocidade de processamento determinada por um circuito externo que produz um determinado *clock*.

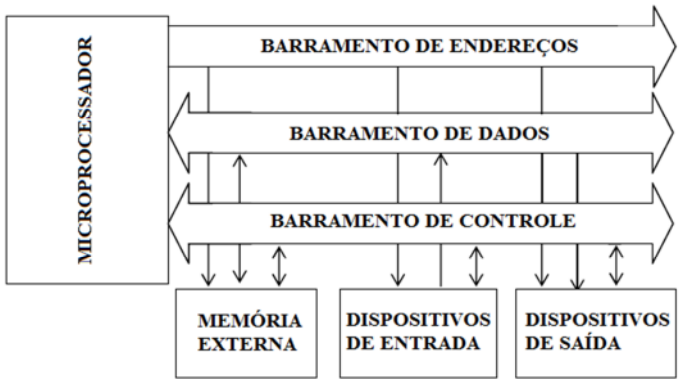

Figura 3-5: Microprocessador clássico e seus periféricos

O poder de processamento dos microprocessadores depende de características como o comprimento da palavra processada (em bits), quantidade de núcleos, arquitetura apresentada, tipo de instruções, entre outras. Como fator de grande importância tem-se ainda a existência de memória externa, onde estão armazenados os programas que serão executados.

Na Figura 3-6 podemos ver exemplos de alguns componentes que se encontram disponíveis num único circuito integrado do microcontrolador, conseguindo reunir uma grande quantidade de recursos.

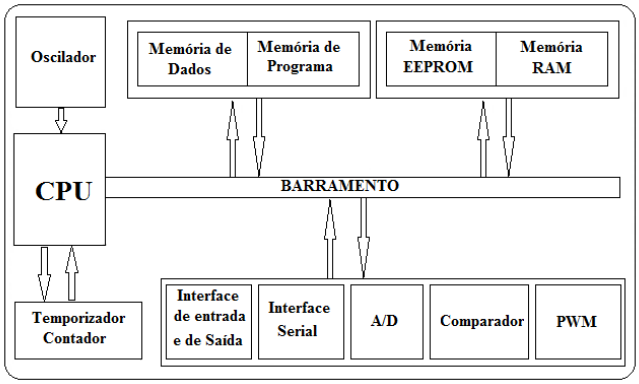

Figura 3-6: Arquitetura do microcontrolador Arduino

No diagrama de blocos que se apresenta na Figura 3-7 e que representa o microcontrolador ATmega2560 [4] é possível identificar todos os seus componentes e as suas relações funcionais.

O Arduino é constituído por vários componentes. Relativamente aos componentes que constituem o seu núcleo, destacam-se alguns de grande importância e que é fundamental conhecer para desenvolver aplicações. A explicação de todos os componentes que compõem o Arduinos está disponível na informação técnica do equipamento disponibilizado pelo fabricante [3]. Neste sentido, seguidamente descreve-se a memória e alguns terminais (*pinout*) do equipamento.

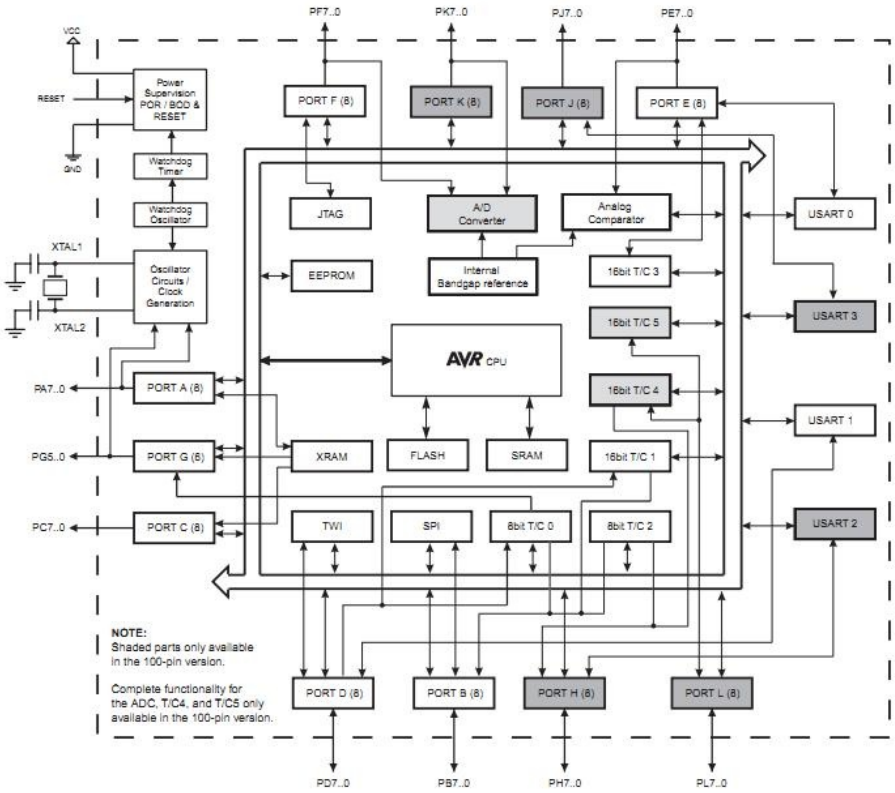

Figura 3-7 Diagrama de blocos ATmega2560

- Memória
  Podem ser consideradas três tipos de memórias, em que cada uma terá funções destintas no desenvolvimento e execução dos processos, nomeadamente:

  Memória Flash que permite o armazenamento do *bootloader* e do programa – *sketch* - a ser executado, é capaz de preservar os dados armazenados por um longo tempo sem a presença de corrente elétrica. O *bootloader* é a parte do programa que garante todas as premissas de arranque do Arduino e o *sketch* é a aplicação desenvolvida pelo utilizador.

  Memória SRAM [38] é utilizada pelo programa que ao ser executado cria e modifica os valores das variáveis necessárias ao seu funcionamento e é limpa todas as vezes que é iniciado o dispositivo. Este tipo de memória é idêntica à memória RAM dos computadores.

  Memória EEPROM é utilizada para armazenar variáveis e informação não voláteis.

Por vezes, devido à necessidade de preservar o espaço de memória para processamento, é utilizada, quer a memória *Flash*, quer a EEPROM, para guardar informação com alguma

[38] SRAM - Static Random Access Memory

dimensão como, por exemplo, uma constante numérica (Ex: o número π). Para tal, recorre-se a bibliotecas próprias que possibilitam essa leitura e escrita. Para a EEPROM, pode-se utilizar a biblioteca "EEPROM.h" e para a memória *Flash* pode-se recorrer à biblioteca "Pgmspace.h" ou "Flash.h".

- *Pinout* do Arduino

  Como já foi referido anteriormente na secção 3.3.2 – Arduino Mega, o Arduino disponibiliza 54 pinos digitais de entrada/saída em que 14 desses pinos podem ser utilizados como saídas PWM. Eles operam à tensão de 5 V e cada pino pode fornecer ou receber, no máximo, 40 mA de corrente e possuem uma resistência interna de 20 a 50 KΩ.

  São ainda disponibilizados 16 pinos para entradas analógicas. Para este efeito o microprocessador possui internamente um conversor ADC[39] de 10 bits. Assim, o número de combinações possíveis é $2^{10} = 1024$ e como a tensão máxima é de 5V, correspondendo ao valor 1023, obtém-se a seguinte resolução (1):

  $$\frac{5}{1024} = 0{,}00488\,V \cong 5mV \quad (1)$$

  Deste modo, só se conseguem detetar variações superiores a 5 mV, ou seja, o valor lido pelo Arduino só se altera a cada 5 mV de variação do sinal analógico de entrada [5].

  Em determinadas aplicações, a resolução do Arduino não é aceitável, como, por exemplo, quando se utilizam determinados tipos de sensores analógicos, em que desse modo existe uma grande perda de resolução. Nestes casos, sem recorrer a eletrónica externa, pode-se utilizar o pino AREF[40] do Arduino para melhorar a resolução. Este pino permite alterar a referência analógica habitual (5V) para o valor de tensão adequado. Todas as entradas analógicas ficam afetadas com a nova referência introduzida.
  Exemplificando, se for introduzido no pino AREF a tensão de 2V obtém-se a seguinte resolução (2):

  $$\frac{2}{1024} \cong 0{,}00195\,V \cong 2mV \quad (2)$$

  É de referir que, após configurar o *Arduino* para o uso do pino AREF, deixam de estar disponíveis os pinos de 3.3V e 5V. Neste caso é necessário recorrer a alimentação externa, se se pretendessem utilizar essas tensões de alimentação.

- *Alimentação* do *Arduino*

  No que se refere a alimentação da placa é disponibilizada uma ligação USB com alimentação proveniente de um PC ou através de uma fonte externa. A entrada da

[39] ADC - Analog Digital Converter
[40] AREF – Analog Reference

alimentação é selecionada automaticamente, isto é, a própria placa faz a gestão da sua alimentação, podendo estar as duas fontes de alimentação ligadas em simultâneo. Adicionalmente, a placa pode ser alimentada por intermédio de baterias externas com a tensão máxima de 5V As baterias são ligadas aos terminais GND e V_{in} da placa do Arduíno. A alimentação externa não deverá ter uma tensão menor do que 6V DC e no máximo 20V DC, sendo aconselhável uma tensão entre os 7V e os 12V DC para evitar problemas de instabilidade ou sobreaquecimento do regulador de tensão.

Os pinos de alimentação são:

- **VIN** - Destinado à alimentação da placa Arduino quando se usa alimentação externa, como alternativa aos 5V fornecidos pela ligação USB ou outra fonte de alimentação regulada. É possível fornecer alimentação através deste pino ou funcionar como saída de tensão quando a placa estiver a ser alimentanda por outra das formas de alimentação;
- **5V -** Fornecimento de alimentação regulada a 5V para dispositivos externos;
- **V3 -** Alimentação de 3,3V para dispositivos externos, gerada pelo circuito integrado FTDI. A corrente máxima que pode fornecer é 50 mA.
- **GND.** Pinos terra (-).

- *Pino Digital 13* do *Arduino*

  O Arduíno possui ainda um led ligado ao pino da saída digital 13 incorporado na própria placa, podendo servir para testes ou outras finalidades, sem necessidade de recorrer a montagens externas. Este pino revelou-se bastante útil.

3.3.6. Desenvolvimento de Aplicações no Arduino

A linguagem de programação utilizada com o Arduino é uma implementação do "*Wiring*" (sub conjunto da linguagem C++), e baseada no ambiente de programação "*Processing*". Trata-se de uma linguagem de programação de código aberto e ambiente de desenvolvimento integrado que suporta programação gráfica. A filosofia subjacente à plataforma Arduino é a sua fácil utilização, até por pessoas que não são das áreas de eletrónica e informática. Assim, a grande contribuição do projeto em si é justamente a facilidade de utilizar a eletrónica e a informática como meio para o desenvolvimento de objetos interativos, sem que se precise conhecer aprofundadamente as partes internas de um microcontrolador e como aceder aos seus periféricos.

A ferramenta IDE é fornecida pelo próprio fabricante do Arduino e descarregado do próprio sítio. Na Figura 3-8, mostra-se o Interface IDE. Este ambiente de desenvolvimento é construído em Java, compatível com qualquer sistema operativo existente no mercado. Particularmente para quem se inicia em programação de microcontroladores, é um ambiente amigável, muito simples e intuitivo.

Quando executamos o IDE temos de ter em conta qual a placa que estamos a usar, para que o IDE reconheça o *bootloader* e consiga enviar o seu código corretamente para a placa. Depois, seleciona-se a porta série virtual que está ligada ao Arduino. Após estas configurações, pode-se então começar a utilizar o IDE para o desenvolvimento da aplicação. Uma vez concluída a aplicação, será apenas necessário efetuar o *upload* para o Arduíno e visualizar o respetivo resultado.

O *software* Arduino inclui um monitor série, o que permite que dados simples de texto possam ser enviados de e para a placa Arduino através de uma ligação série externa e ainda como é de desenvolvimento livre, tem a particularidade de possuir uma grande coleção de bibliotecas. Algumas das bibliotecas já vêm com o IDE e outras podem ser descarregadas do próprio sítio do Arduino, ou de outra fonte que esteja disponível. Adicionalmente, o utilizador pode criar as suas próprias bibliotecas. Para se poder utilizar as novas bibliotecas apenas se têm que adicionar à pasta das bibliotecas do IDE.

Ao realizar o *upload* da aplicação, o IDE faz uma verificação de possíveis erros de sintaxe compila o código para linguagem máquina, procedendo de seguida ao envio do ficheiro resultante para o microcontrolador, sem necessidade de mais intervenção ou recurso a qualquer outro programa [6].

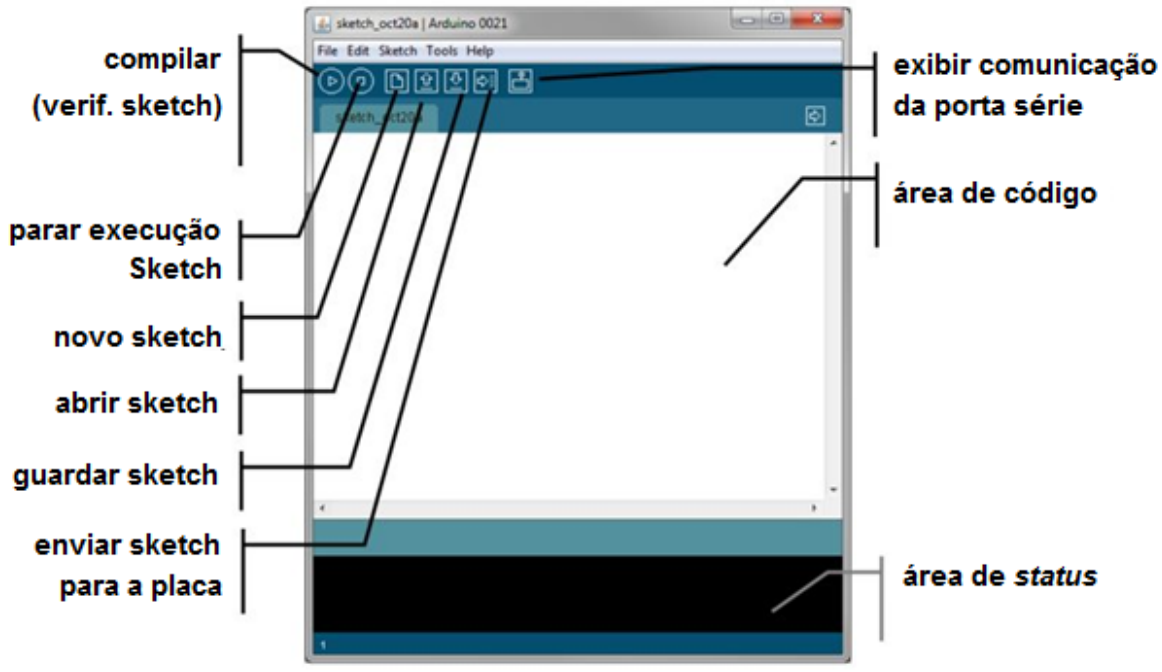

Figura 3-8: *Interface* IDE

A estrutura de um programa Arduino é designada por "*sketch*" que é composto por duas funções principais: ***setup*()** e ***loop*()**.

- **A função *setup*()** tem o código de inicialização do programa e apenas é executada uma única vez, no início da execução da aplicação. Nesta função definem-se os pinos de entrada e saída, os valores inicias das variáveis, faz-se a inicialização da utilização das bibliotecas e da comunicação série, entre outros procedimentos.
- **A função *loop*()** é executada de forma contínua em ciclo, tal como indica o nome da função. É formada pelo código principal do programa, o que pode permitir a leitura sucessiva de portas, e de valores provenientes de sensores externos, e atuar de acordo com as condições estabelecidas, entre muitas outras aplicações.

A declaração de variáveis globais ou objetos e a inclusão de bibliotecas, caso o nosso programa necessite, deverão ser feitas logo no início, formando um cabeçalho da aplicação. Na Figura 3-9 pode-se ver a estrutura de um sketch simples.

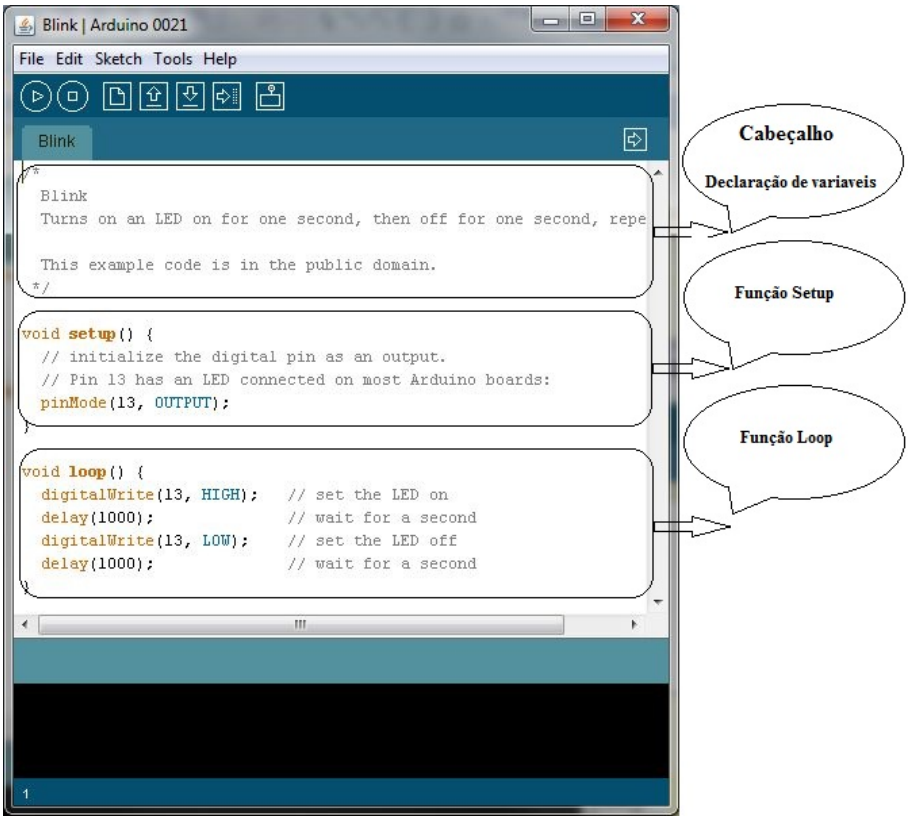

Figura 3-9: Estrutura do sketch no IDE

3.3.7. Shields Arduino

A plataforma de desenvolvimento Arduino permite ser expandida e utilizada para funções específicas, permitindo dessa forma uma maior amplitude de utilização, nomeadamente no controlo de motores, na interligação com redes *Ethernet*, ou redes sem fios, equipamentos específicos de medição ou sensores, entre outras utilizações. Para diversificar essa utilização

são utilizados *shields* que existem no mercado para as várias aplicações. Normalmente são preparados para serem acoplados ao Arduino de uma forma fácil, encaixando sobre a placa base. Através das suas bibliotecas incluídas na aplicação, esta fica de imediato pronta a funcionar.
Normalmente a arquitetura dos *shields* permite ainda a utilização dos pinos do Arduino e, na maioria dos casos, podemos encaixar vários *shields*, uns por cima dos outros.
Não querendo alargar em demasia os objetivos deste projeto, serão abordados aqui, de uma forma simples, apenas os *shields* utilizados para o acoplamento dos módulos de comunicação XBee.

3.3.8. XBee *Shield*

Este Xbee *shield* permite que uma placa Arduino comunique sem fios usando o protocolo ZigBee, e baseia-se no módulo de Xbee MaxStream. O módulo pode comunicar a uma distância máxima de 1,6 km ao ar livre, desde que haja linha de vista, mas estes valores estarão dependentes do tipo de módulo Xbee utilizado. Este assunto será abordado na secção 3.7. deste documento.
O XBee *shield* pode ser usado como alternativa à comunicação série ou USB, ou pode ser colocado em modo de comando e ser programado para uma rede de transmissão com diversas configurações. A Figura 3-10 mostra um exemplo de módulo XBee *shield*.

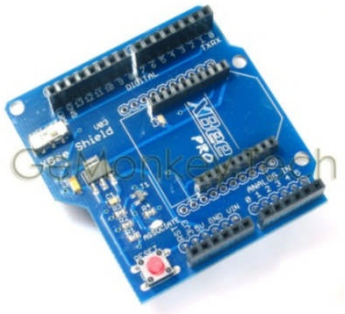

Figura 3-10: Exemplo de módulo Xbee *Shield* V03

Apenas por motivos meramente experimentais, foi ainda testado e usado neste trabalho um outro *Shield* (Figura 3-11) com características idênticas, só variando na sua arquitetura. Devido à sua forma de encaixe no Arduino permite uma maior acessibilidade aos pinos livres. Adicionalmente, utiliza a ligação ICSP e disponibiliza uma ligação fácil a todos os terminais do módulo XBEE.

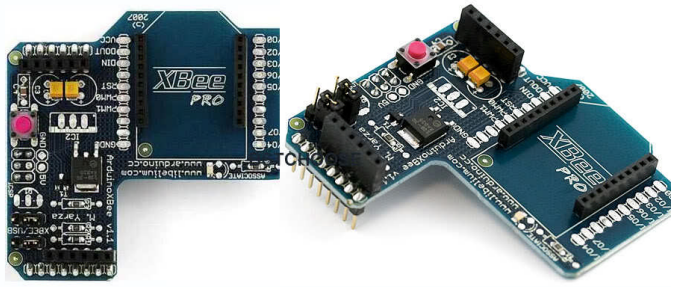

Figura 3-11: Exemplo de módulo Shield XBee Pro V1.1

3.4. Sensor Digital de Temperatura DS18B20

O sensor digital de temperatura DS18B20 internamente tem um ADC que permite a medição de temperatura com uma resolução ajustável de 9 a 12 bits, que permite ajustar e aumentar a precisão do sensor. Quanto maior a resolução escolhida, também maior é o tempo de conversão. Tem ainda incorporadas funções não-voláteis, programáveis pelo utilizador, para alarme de temperatura máxima e mínima.

A comunicação com este sensor é do tipo *One-Wire*[41] que, por definição, requer apenas uma linha de dados para a comunicação com um microprocessador central, no pressuposto de haverem linhas de alimentação a ligar os dois dispositivos.

Este protocolo de comunicação foi desenvolvido pela "Dallas Semiconductor Corp" [6], que é um fabricante de componentes eletrónicos, nomeadamente de uma grande variedade de sensores. Consiste num BUS de dados (dois fios), partilhado por vários sensores a ele ligados, um dos fios designado por "*ground*" e outro onde fluem simultaneamente os dados e a alimentação.

Na Figura 3-12 ilustra-se a ligação destes dispositivos.

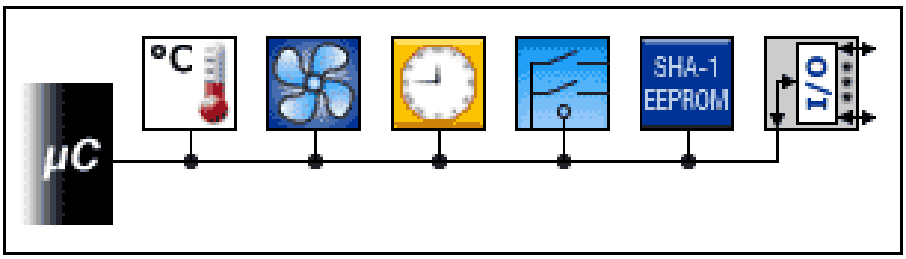

Figura 3-12: Ligação de dispositivos One-Wire

Os dispositivos *One-wire* têm como principal característica, necessitarem de uma corrente de alimentação muito reduzida. Desta forma, conseguem alimentar-se do próprio BUS de dados, carregando um condensador interno quando o BUS está com nível de tensão alto. Este modo de operação é chamado modo parasita[42], podendo assim evitar o uso de fontes de alimentação externas.

Este protocolo é viável para a utilização de diversos sensores para funções diferentes como por exemplo, medição de temperaturas, acelerações, humidade, entre outros. O protocolo de comunicação entre os dispositivos e o microcontrolador é do tipo mestre/escravo. Os dispositivos, uma vez ligados por intermédio de um BUS, funcionam como escravo de um sistema, existindo no mesmo BUS um dispositivo microcontrolador mestre que efetuará a identificação de todos os dispositivos ligados, por intermédio da leitura dos endereços

[41] One-Wire – Um fio
[42] Modo parasita – na literatura anglo saxónica é referido como "*Parasite Power*"

internos. Esta identificação dos sensores é única e é composta por 64 bits gravada numa ROM[43] quando fabricados e podem ser instalados até 127 dispositivos no mesmo Bus.

A informação circula no BUS em forma de trama contendo os dados, a identificação do dispositivo, assim como bits de verificação (*Checksum*) para que o mestre possa validar a informação recolhida. Desta forma é assegurada a fiabilidade da comunicação dos dados.

A faixa de temperatura de funcionamento é de -55 ° C a 125 ° C e uma precisão de ± 0,5 ° C no intervalo de -10 ° C a 85 ° C.

3.4.1. Características Principais

Seguidamente enumeram-se algumas características dos sensores de temperatura DS18B20 que determinaram a sua utilização neste trabalho:

- O interface de Comunicação One-Wire requer apenas uma porta (um pino) para a Comunicação;
- Cada dispositivo tem um código único de série de 64-Bits armazenado numa ROM On-Board;
- A capacidade de multiponto simplifica a aplicabilidade de sensores de temperatura nas aplicações;
- Não necessita de componentes externos;
- Pode ser alimentado a partir da linha de dados; a faixa de alimentação é de 3.0V a 5.5V;
- Mede temperaturas entre -55 ° C a +125 ° C (-67 ° F a 257 ° F);
- Tem uma precisão de ± 0,5 ° C entre -10 ° C a +85 ° C;
- A resolução do termómetro é selecionável pelo utilizador a partir de 9 a 12 Bits;
- Converte valores digitais de temperatura de 12-Bits num tempo máximo de 750ms;
- Configuração não volátil, feita pelo utilizador;
- Para gerir os alarmes de temperatura, existe um comando de pesquisa de alarmes, que identifica e aborda os dispositivos cuja temperatura está fora dos limites programados;
- *Software* compatível com sensores da mesma família DS1822,
- As aplicações mais comuns são o controlo termostático, sistemas industriais, produtos de consumo, termômetros, ou qualquer outro sistema termicamente sensíveis;
- Existem várias configurações físicas disponíveis, capsulagem de 8 Pin-SO (150 mils), 8-Pin μSOP, e 3 Pin-TO-92 Pacotes, como se pode observar na Figura 3-13.

[43] ROM - Read Only Memory

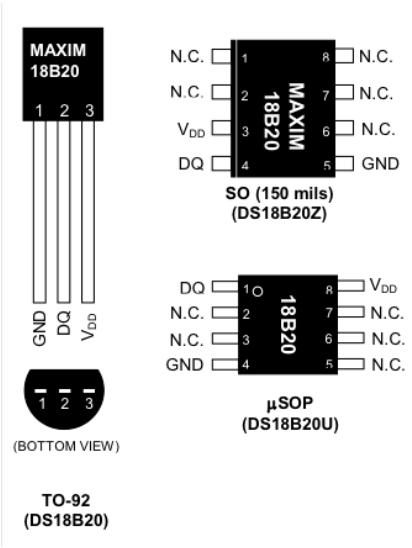

Figura 3-13: Configurações e pinos dos sensores DS18B20 [7]

No ANEXO 2 referem-se as especificações do sensor de temperatura DS18B20.
Na Figura 3-14 mostra-se o diagrama de blocos do sensor de temperatura DS18B20 que é útil para perceber o seu funcionamento de modo a poder programá-lo corretamente.

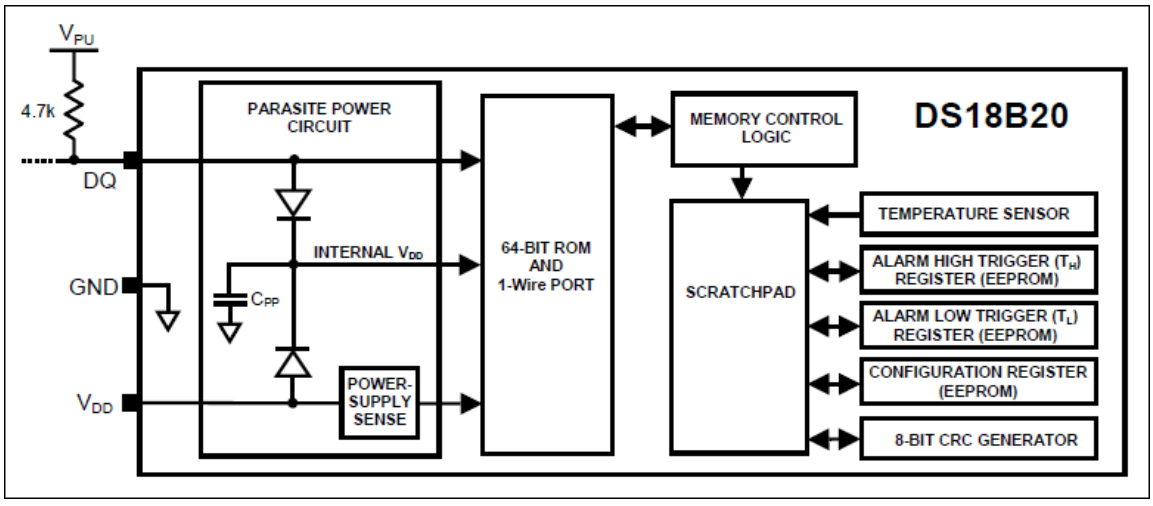

Figura 3-14: Diagrama de Blocos do DS18B20

3.4.2. Implementação e Funcionamento

Como já foi referido anteriormente, os sensores da família DS18x20 são relativamente precisos e fáceis de montar fisicamente, através do BUS que os interliga. Eles podem obter a energia que necessitam, assim como comunicar através do protocolo *one-wire*. A Figura 3-15, representa o esquema de ligações elétricas utilizado [8].

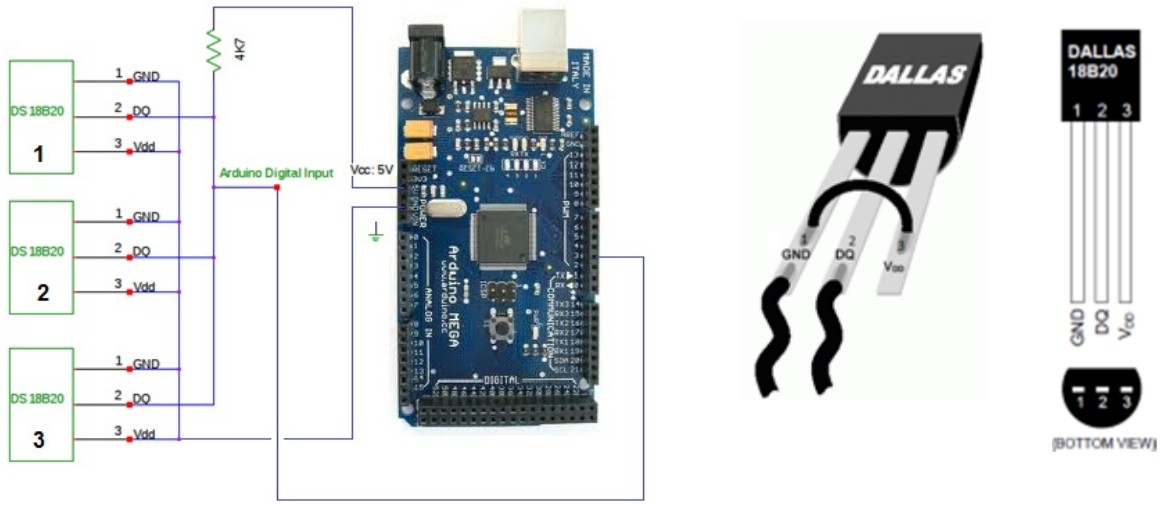

Figura 3-15: Ligação Elétrica dos Dispositivos DS18B20

Para se obterem dados dos dispositivos instalados o micro controlador mestre inicia com um envio do comando "*reset*" recebendo dos dispositivos escravos uma resposta de presença. Então o microcontrolador seleciona o dispositivo, analisando a trama que contém os dados de identificação do dispositivo e através do comando "MATCH ROM [55h]" inicia o processo de comunicação com o dispositivo pretendido.

Para obter uma medição de temperatura, o microcontrolador emite o comando "CONVERT [44h]". Quando o sensor recebe este comando, inicia o processo de conversão de dados, gerando 2 bytes com o valor da temperatura medido. Este é um processo relativamente lento e pode demorar até 750 milissegundos, por isso temos de esperar algum tempo após a emissão do comando.

Todas as medições são armazenadas num espaço de memória RAM do sensor, designado "SCRATCHPAD", a estrutura desta zona da memória mostra-se na Figura 3-16. Podemos lê-lo para obter os dados, assim como podemos escrever para definir os limites de alarme e ainda especificar a resolução do sensor.

Para ler a partir da "SCRATCHPAD", emite-se um comando "READ SCRATCHPAD [BEh]"e na resposta recebem-se 9 bytes de dados (ver Figura 3-16). Então, podemos obter uma temperatura de acordo com a seguinte fórmula (3):

$$\textbf{Temp = ((HighByte << 8) + LowByte) *0.0625} \qquad (3)$$

Em que <<8 é a posição do bit significativo (<< = *bitshift left*) e 0,0625 é um coeficiente de conversão entre os valores internos do sensor de temperatura e a temperatura real, de acordo com a resolução 12-bit para este exemplo. Cada degrau na escala do sensor representa 0,0625° C.

SCRATCHPAD
(POWER-UP STATE)

Byte 0	Temperature LSB (50h)	(85°C)
Byte 1	Temperature MSB (05h)	
Byte 2	T_H Register or User Byte 1*	
Byte 3	T_L Register or User Byte 2*	
Byte 4	Configuration Register*	
Byte 5	Reserved (FFh)	
Byte 6	Reserved	
Byte 7	Reserved (10h)	
Byte 8	CRC*	

Figura 3-16: Estrutura da Memória SCRATCHPAD

Todo este processo está já desenvolvido e disponível em forma de biblioteca, bastando, para isso, inclui-la na aplicação. De uma forma perfeitamente transparente, a aplicação apenas efetua os pedidos de identificação e leitura, obtendo de imediato os resultados. Esta metodologia facilita bastante o desenvolvimento de aplicações, sem requerer um conhecimento detalhado dos comandos para ler ou escrever na "SCRATCHPAD".
A Figura 3-17 mostra um extrato do cabeçalho e inclusão das definições das bibliotecas.

```
/* Aplicação para Monitorização de Ambientes Controlados TLab
 Light Sensor BH1750 V1.0 - medição de luminosidade
 DHT11 Sensor de Humidade e Temperatura
 DS18 Sensor Digital de Temperatura
   Comunicação wireless XBee Series 2 e Pro
*/

/* ////////////////( Biblioteca )////////////////////////*/

#include <dht11.h> // DHT11
#include <Wire.h> // BH1750 I2C Mode
#include <math.h> // Clac. Matemático
#include <OneWire.h> // comunicação do BUS
#include <DallasTemperature.h> // Sensor DS18
```

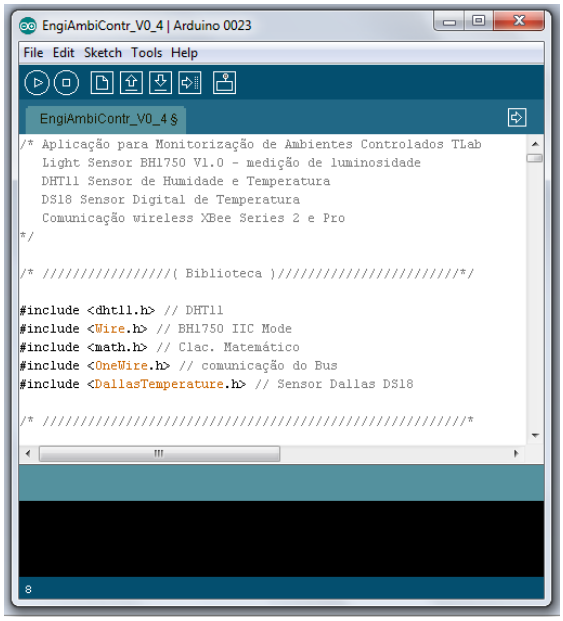

Figura 3-17: Extrato do Cabeçalho

Como se mostra neste pequeno extrato, a forma como se obtém o valor de temperatura é extremamente simplificada com o uso das bibliotecas, apenas necessitamos de identificar o sensor que se pretende obter o valor de temperatura e através do seu endereço (variável *tempDeviceAddress*) e o comando *requestTemperatures*, da biblioteca DallasTemperature obtemos de imediato a resposta ao pedido.

```
…
void LerDS18()
    {
        Serial.println("Temperatura ");

 // Faz o pedido de temperatura a todos os dispositivos do BUS
        Serial.print("Lendo Temperaturas...");
        sensors.requestTemperatures(); // envia o comando de pedido de temperatura
        Serial.println("Concluido");

// Loop a todos os dispositivos para obter as temperaturas
       for(int i=0;i<numberOfDevices; i++)
       {
// Procura no BUS todos os endereços
        if(sensors.getAddress(tempDeviceAddress, i))
           {
// Escreve número de ordem do dispositivo
            Serial.print("Temperatura do dispositivo: ");
            Serial.println(i,DEC);

// Resposta imediata dos sensores
            printTemperature(tempDeviceAddress); // Chama função para escrever os dados
          }
         }
        }
```

3.5. Sensor de Luminosidade BH1750FVI

O sensor digital de luminosidade que foi selecionado para este projeto baseia-se num módulo sensor de intensidade de luz BH1750FVI que integra um conversor ADC de 16 bits. O resultado dos seus dados é devolvido em Lux[44] e apresenta uma sensibilidade espectral elevada, semelhante à do olho humano. Esta característica é importante, pois os comprimentos de onda fora do alcance da visão humana, tais como os ultravioleta e os infravermelhos, podem dar origem a leituras imprecisas dos sensores de luz [9].

A comunicação com o microcontrolador é feita através de um BUS de comunicação do tipo I^2C.

[44] Lux – Unidade de medida da densidade luminosa e representa 1 lúmen por metro quadrado

Este circuito integrado é um dos mais utilizados no controlo de luminosidade de dispositivos como, por exemplo, LCD[45]s e telemóveis. Com o objetivo de reduzir o consumo energético, são utilizados para detetar o nível de luminosidade. A Figura 3-18 apresenta um exemplo do módulo sensor de luminosidade BH1750FVI.

Figura 3-18: Módulo sensor de luminosidade BH1750FVI

3.5.1. Comunicação I²C

A comunicação série síncrona I²C, é implementada por apenas dois fios condutores. Num condutor faz-se a transmissão dos bits de dados e é conhecido por SDA e o outro condutor designado por SCL é utilizado como um sinal de relógio. O pacote de dados enviado pela linha do I²C contém o endereço, que permite definir qual o dispositivo que deve responder a determinada solicitação.

O barramento I²C foi definido de tal forma, que permitiu que diversos circuitos integrados, de conceção e modelos diferentes, pudessem ser integrados numa mesma aplicação. Foram introduzidos pequenos filtros na parte recetora dos circuitos integrados para reduzir problemas com os ruídos. Foi ainda desenvolvido de forma a possibilitar que circuitos integrados mais lentos possam adiar, por breves instantes, o envio do pacote de dados, por forma a libertar as linhas de comunicação para dispositivos mais rápidos.

Este barramento de comunicação é amplamente usado em sistemas eletrónicos relacionados com imagem e som, no controlo e processamento de sinais, por exemplo, VGA[46] e HDMI[47]. O sucesso do I²C deve-se principalmente à versatilidade do seu BUS.

[45] LCD – Liquid Cristal Display
[46] VGA - Video Graphics Array
[47] HDMI - High-Definition Multimedia Interface

3.5.2. Características Principais

Seguidamente referem-se algumas características do sensor de luminosidade BH1750FVI que são importantes para este trabalho:

- A tensão de alimentação usada 4,5V a 6V ou 2,3V;
- Interface do tipo BUS I^2C possui dois endereços alternativos;
- A sua resposta espetral é aproximada à do olho humano;
- Apresenta uma sensibilidade cuja gama de medição pode ir dos 1 aos 65535 lux;
- Não depende do tipo de luz. Responde de igual forma para os diversos tipos de iluminação. Por exemplo lâmpada incandescente, lâmpada fluorescente, lâmpada de halogéneo, LED branco e luz solar;
- Pode-se ajustar o resultado da medição quando o sensor de luminosidade é afetado por uma janela ótica. É possível detetar no mínimo 0,11 lx e máximo 100.000 lx, usando esta função;
- A influência de infravermelhos é baixa;
- Apresenta uma variação de medição da sensibilidade de ± 20%.

3.5.3. Diagrama de Blocos e Configuração Física

Na Figura 3-19 mostra-se o diagrama de blocos do sensor que ilustra o seu funcionamento.

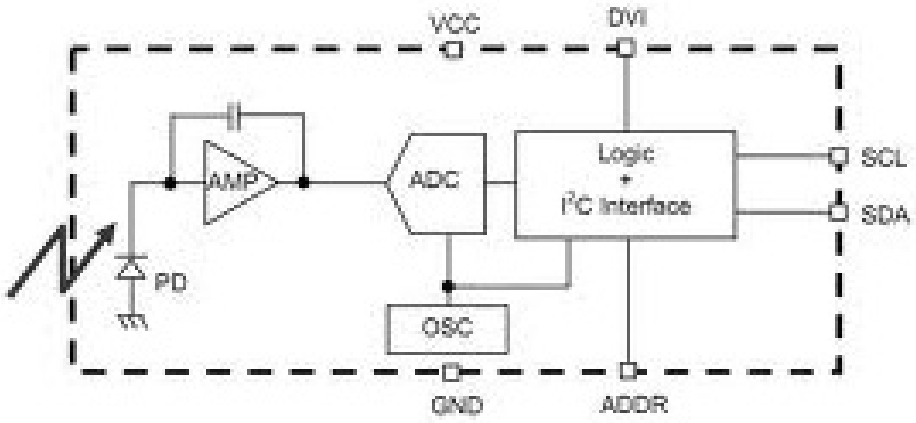

Figura 3-19: Diagrama de Blocos do sensor BH1750FVI

Os blocos têm a seguinte funcionalidade [8]:

- **PD** – é o foto díodo com resposta idêntica à sensibilidade da visão humana:
- **AMP** – é o amplificador operacional integrador para conversão da corrente em tensão elétrica, gerada pelo foto díodo;
- **ADC** – é o conversor analógico para digital, que converte os valores de tensão obtidos em dados na forma digital de 16bit;

- **LOGIC + I2C INTERFACE** – é a lógica de processamento do cálculo da luminosidade e interface I2C, onde são guardados os seguintes registo: DATA REGISTER – que é o valor digital da medição da iluminação, com valor inicial de "0000_0000_0000_0000", MEASUREMENT TIME REGISTER – que é o tempo durante o qual foi realizada a medição, com valor inicial de "0100_0101";
- **OSC** – é o relógio oscilador interno (320kHz).

A Figura 3-20 mostra a configuração física do sensor BH1750FVI e o seu *pinout.*
Para além da configuração original aqui representada, existem outras adaptadas a diferentes aplicações.
O BH1750FVI normalmente é montado em pequenas placas eletrónicas associado a outros componentes de modo a satisfazer, da forma mais adequada, as necessidades de interligação com os controladores, resultando assim em diversos dispositivos semelhantes comercializados.

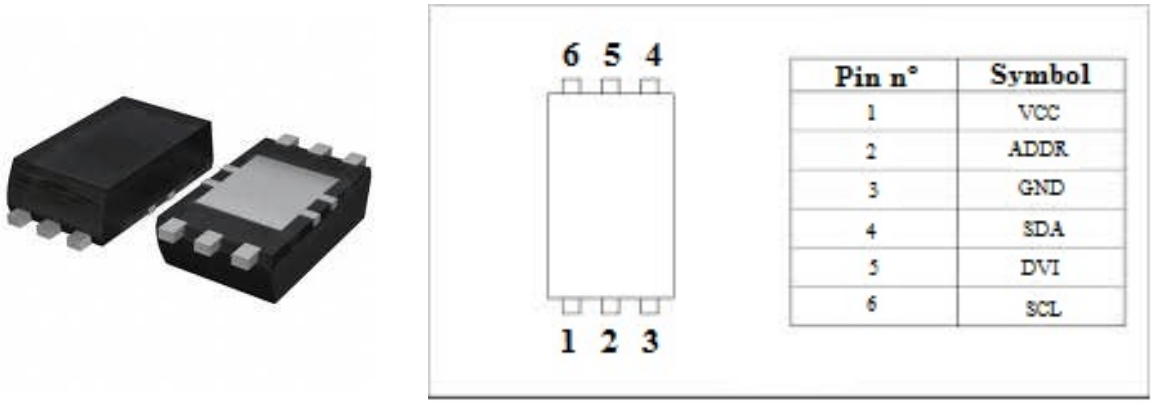

Figura 3-20: Configuração do sensor BH1750FVI e o *Pinout*

3.5.4. Implementação e Funcionamento

Como já foi referido anteriormente, este dispositivo tem uma implementação muito simplificada ao utilizar a comunicação BUS I2C. Assim, o sensor tem um par de fios para comunicação e outros dois que se destinam à sua alimentação. Deste modo, no nosso caso, podemos ligar diretamente ao microcontrolador Arduino utilizando os seus terminais de alimentação (5V e GND), para alimentar o dispositivo, e utilizar dois terminais analógicos, por exemplo, os terminais 4 e 5 para ligar SDA e SCL do sensor (BUS I2C). Pode-se ainda, caso se disponha de um *shield* para ligação de sensores, ligar diretamente a uma das entradas deste, do tipo I2C, como se ilustra na Figura 3-21.

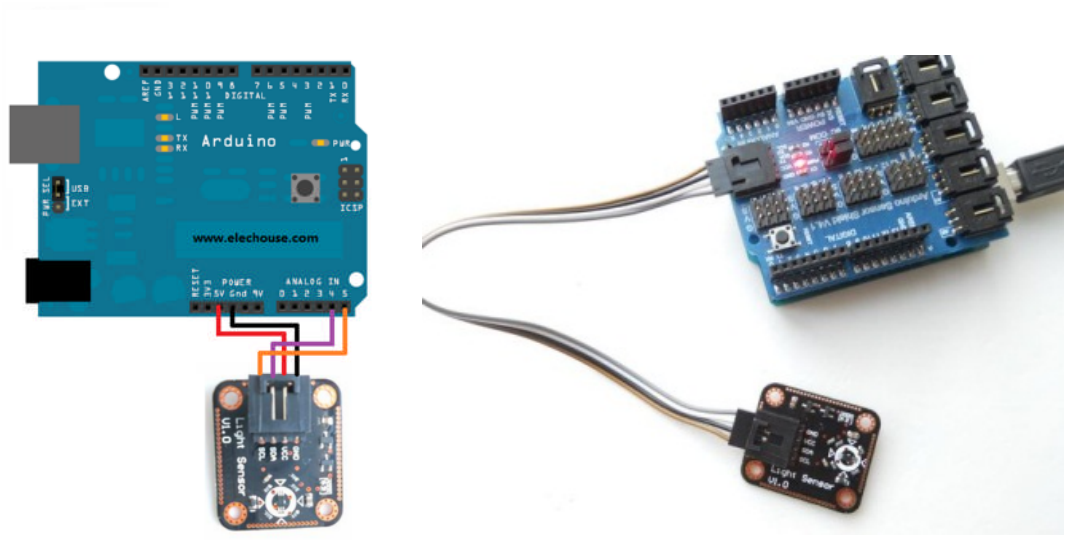

Figura 3-21: Interligação do sensor BH1750FVI com o Arduino

A utilização destes dispositivos integrada em diversos equipamentos obriga, frequentemente, à utilização de janelas óticas, isto é, aberturas para a entrada de luz que, por vezes, são protegidas com materiais transparentes. Estes materiais alteram um pouco a quantidade de luz que chega até ao sensor, torna-se necessário efetuar um ajuste da sensibilidade do sensor. A anulação da influência da janela ótica é possível através do ajuste do tempo de medição. Por exemplo, quando a taxa de transmissão da janela ótica é de 50%, ou seja, existe uma redução de 50% do valor de medição em relação à medição sem janela, então a influência da janela ótica é anulada alterando a sensibilidade padrão para o dobro. Se o tempo padrão de medição rondar os 120 ms, tempo que é comum para estes sensores, após a calibração, o tempo passa então para 240 ms. Esta calibração consegue-se alterando o valor do registo MTreg[48]. Este procedimento pode ser consultado no manual técnico do fabricante.

Na integração deste sensor de luminosidade como Interface TLab, foi usada uma placa de acoplamento para sensores que, como já se referiu anteriormente, disponibiliza um terminal especifico com ligadores de quatro fios, facilitando muito na fase de desenvolvimento.

Os valores de luminosidade obtidos resultam de procedimentos que fazem parte da aplicação desenvolvida e instalada no Arduino, em conjunto com a biblioteca Wire.h disponibilizada pelo fabricante.

[48] MTreg – Measurement Time Register

3.6. Sensor de Temperatura e Humidade DHT11

Este sensor DHT11 possui um sensor complexo de temperatura e humidade, com uma saída de sinal digital. Usa uma tecnologia sensorial de aquisição de sinal digital para temperatura e humidade, que assegura uma fiabilidade e estabilidade a longo prazo. A Figura 3-22 mostra o sensor DHT11, evidenciando os três pinos para ligação.

Figura 3-22: Sensor DHT11

Este sensor inclui um elemento sensorial de humidade tipo resistivo e um do tipo NTC que faz a medição da temperatura, associado a um microcontrolador rápido de 8 bits. Este conjunto permite uma resposta rápida, com alguma imunidade a interferências e com uma boa relação custo-benefício.

Cada componente DHT11 é calibrado em laboratório, o que o torna bastante preciso na medição de humidade, conforme se pode ver nas características técnicas no ANEXO 3 . Os coeficientes de calibração são armazenados na memória interna do sensor, para serem utilizados no processamento inerente ao processo de medição.

O interface série de um único fio torna a sua integração, muito rápida e fácil. O seu tamanho reduzido, o baixo consumo de energia e a transmissão do sinal a mais de 20 metros apresentam-no como a melhor escolha para várias aplicações.

3.6.1. Comunicação Série

O interface série de dados através de um único fio é usado para comunicação e sincronização entre o micro controlador e o sensor DHT11. Este processo de comunicação é feito em cerca de 4ms, como se ilustrado na Figura 3-23. O processo de comunicação inicia-se quando o microcontrolador envia o sinal de início "*start signal*" para o sensor, alterando o estado da linha para o modo de funcionamento. O microcontrolador fica então a aguardar a comunicação de resposta do sensor. Este envia ao microcontrolador um sinal de resposta com 40 bits de dados, que incluem o valor da humidade relativa (HR) (8 bits parte inteira e 8 bits

parte decimal) e da temperatura (8 bits parte inteira e 8 bits parte decimal). Fazem parte ainda da resposta 8 bits de verificação (*check sum*). Uma vez concluído este procedimento o dispositivo irá colocar a linha em modo latente[49] (*low-power-consumption mode*) até que se verifique novo pedido por parte do microcontrolador. Está representado na Figura 3-23 o comportamento do sinal em tensão (VCC) na linha de comunicação.

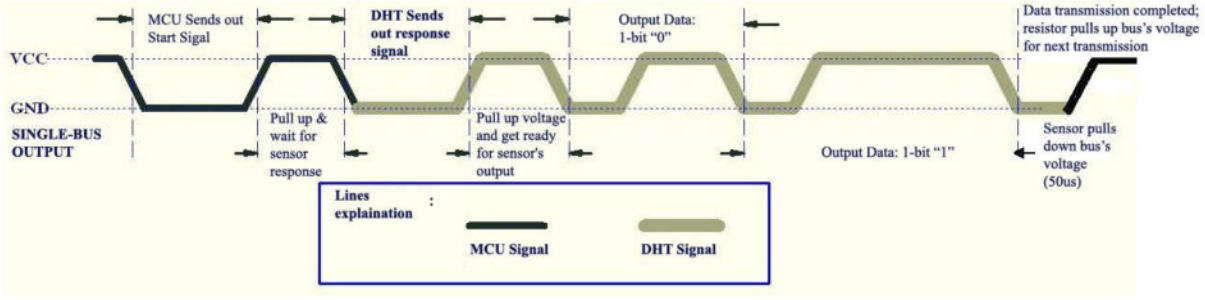

Figura 3-23: Comunicação Single-Wire Two-Way

No processo de comunicação de dados entre o micro controlador e o sensor DHT11, a duração de estado de tensão da linha (*low or high power*) determinam o estado de transmissão.

Após detetado um sinal "*Start Signal*" o dispositivo irá enviar um sinal de resposta de baixa tensão de nível de linha (VCC ≈ GND), que dura 80μs. Em seguida, o programa do dispositivo inverte a tensão do barramento de baixo para alto e mantem-na durante mais 80μs, ficando preparado para o envio de dados.

Quando o dispositivo sensor está a enviar dados para o micro controlador, cada bit de dados começa com o nível de tensão baixo durante 50μs. O sinal de nível de tensão seguinte determina o estado dos bits de dados se é "0" ou "1". Caso o estado da linha permaneça durante mais de 50μs a um nível de sinal baixo, então significa que ocorreu um erro na comunicação.

3.6.2. Características

Avaliando as características técnicas do sensor DHT11, verifica-se que apresenta propriedades muito interessantes, nomeadamente no que diz respeito à HR sendo possível uma medição entre os 20% e os 90% de humidade relativa (HR), com uma grande fiabilidade e uma resolução de uma unidade. No que se refere à medição da temperatura é limitado relativamente à gama de valores, não sendo possível medir valores negativos. Nas temperaturas entre o 0º e os 50º C, apresenta um erro máximo a 50º de 2º Cº. No entanto, como a sua utilização no âmbito deste trabalho estará vocacionada para a medição de

[49] *low-power-consumption mode* – modo de baixo consumo de energia

humidade e temperatura em espaços de trabalho, as suas características são aceitáveis. As características técnicas do sensor podem ser consultadas no ANEXO 3.

3.6.3. Ligações e Configuração

De uma forma genérica a interligação entre um micro controlador e o sensor DHT11 deve ser efetuada conforme se indica na Figura 3-24.

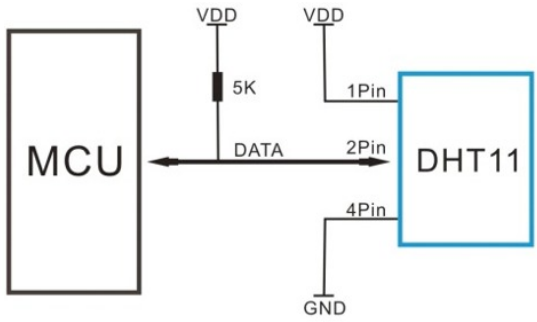

Figura 3-24: Diagrama de ligações do dispositivo DHT11

Quando a distância entre o microcontrolador e o sensor é inferior 20 metros deve ser montada uma resistência de 5 kΩ (*pull-up resistor*). Para distancias superiores o valor da resistência deve ser calculado conforme o manual técnico do fabricante.
Muito embora a configuração do dispositivo DHT11 surja com quatro pinos de ligação, apenas são utilizados três, pois o pino três não está ligado internamente (Figura 3-25). Assim sendo, na maioria das placas comercializadas apenas disponibilizam os três pinos para interligação com o microcontrolador.

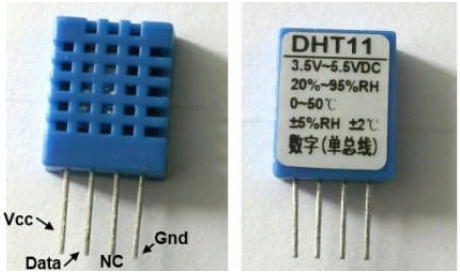

Figura 3-25: Configuração Do Sensor DHT11

3.6.4. Implementação e Funcionamento

No desenvolvimento do Interface TLab foi prevista a possibilidade de ligação de apenas um sensor DHT11, tendo em conta que este dispositivo destina-se a medição das condições de

temperatura e humidade de um espaço de trabalho. Assim não será previsível a utilização de mais do que um sensor por interface TLab.

De acordo com o esquema de ligações referido anteriormente, aproveitou-se a alimentação disponível no Arduino, terminal de 5V e ligou-se um fio para o vcc do sensor, assim como se interligaram os GND. Para fluxo de dados utilizou-se o terminal digital 18 como linha de dados, como se mostra na Figura 3-26.

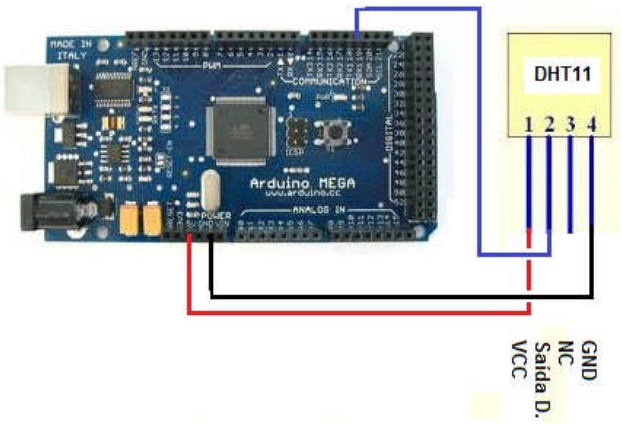

Figura 3-26: Ligação do sensor DHT11 ao Arduino

A biblioteca do sensor DHT11 permite efetuar a gestão de erros como, por exemplo, a validação dos dados. Isto é, se a resposta a um pedido de leitura de temperatura devolver uma resposta válida, a variável de controlo toma o valor de zero, que significa que os dados estão corretos, isto é importante e servirá futuramente para controlar possíveis falhas do sensor, como se mostra no seguinte extrato do programa.

```
/* ////////////////////////////////// (Validação dos Dados da Sonda DHT11) ////////////////////*/

    switch (chk)
    {
       case 0: Serial.println("Dados Corretos");
               break;
       case -1: Serial.println("Erro nos Dados");
               break;
       case -2: Serial.println("Tempo de leitura excedido");
               break;
       default: Serial.println("Erro desconhecido");
               break;
    }
```

A obtenção do valor de temperatura no sensor de DTH11 é feita pela chamada da função de leitura do terminal da linha de dados. A partir daí todo o processo se desenrola de forma semelhante como foi explicado anteriormente. É a biblioteca que gere todo esse processo e mais uma vez se mostra aqui a simplicidade de utilização deste dispositivo.

3.7. Módulo de Comunicação XBee

3.7.1. Introdução

O módulo de transmissão e receção sem fios é um componente fundamental do sistema TLab. Durante a elaboração do projeto optou-se pela utilização de um módulo compatível com a norma IEEE 802.15.4, pois a rede de sensores ZigBee é baseada neste protocolo. Há vários modelos disponíveis no mercado. Contudo, após uma análise onde foram ponderadas as funcionalidades disponíveis e o custo associado, selecionou-se o módulo XBee que satisfaz as necessidades do projeto.

Este dispositivo desenvolvido pela empresa Digi International Inc. a partir de 2005, é um módulo que se baseia no protocolo de comunicação ZigBee. O XBee é disponibilizado em dois modelos, sendo eles, o Xbee e o XBee-Pro, em que as funcionalidades são as mesmas, diferindo entre eles apenas a potência do sinal de transmissão. A forma como se apresentam também tem várias vertentes, nomeadamente no que diz respeito ao tipo de antena, como se mostra na Figura 3-27.

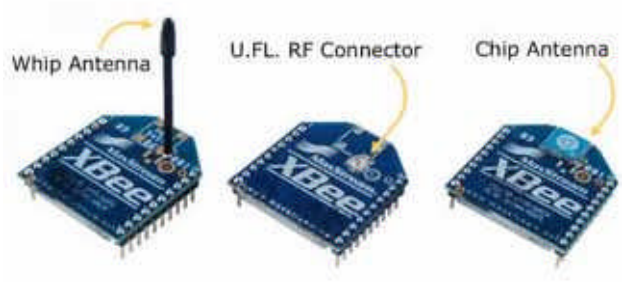

Figura 3-27: Módulo Xbee

3.7.2. Características

Ambos os modelos XBee apresentam dimensões muito reduzidas. O XBee mede aproximadamente 2,7 cm de comprimento e o XBee-Pro 3,3 cm e ambos têm uma largura aproximada de 2,4 cm, sendo esta característica bastante importante para o projeto.

Como referido anteriormente, a principal diferença entre os dois modelos é a potência em termos de sinal de transmissão, e consequentemente, o respetivo alcance. Assim, o XBee dispõe de uma potência de 1mW e um alcance que varia entre os 100m e os 30m, em campo aberto ou no interior de edifícios, respetivamente. Enquanto o XBee Pro opera com uma potência de 60mW, o que lhe permite alcançar um raio de transmissão até 1,6km ou 100m, tratando-se de campo aberto ou no interior de edifícios, respetivamente.

Em termos de consumo, são dispositivos que apresentam um consumo muito reduzido (inferior a 10μA) quando em estado adormecido (*sleep*). As necessidades de alimentação aumentam quando entram em modo de operação (transmissão/receção). No âmbito deste parâmetro, e como se torna evidente, é o XBee quem apresenta menor consumo energético,

com uma corrente de aproximadamente 50mA, para uma tensão de alimentação de 3,3V. No ANEXO 4 faz-se uma breve comparação das características dos modelos do XBee.

Cada módulo disponibiliza um conjunto de terminais de entrada e saída (I/O), configuráveis conforme a aplicação. Esses terminais são o meio de comunicação com o ambiente externo. No caso deste projeto, os terminais servem para comunicar, quer com o computador, quer com o Arduino, utilizando, respetivamente, uma placa adaptadora para USB e um *Shield* de adaptação ao Arduino.

3.7.3. Formato de Dados

Na comunicação entre os XBees, o formato de dados enviados é especificado por uma API, o que facilita a organização dos módulos.
Cada mensagem difundida na rede ZigBee é estruturada por uma trama, que contém informações pertinentes para uma garantia de comunicação na rede. Esta estrutura está representada na Figura 3-28.

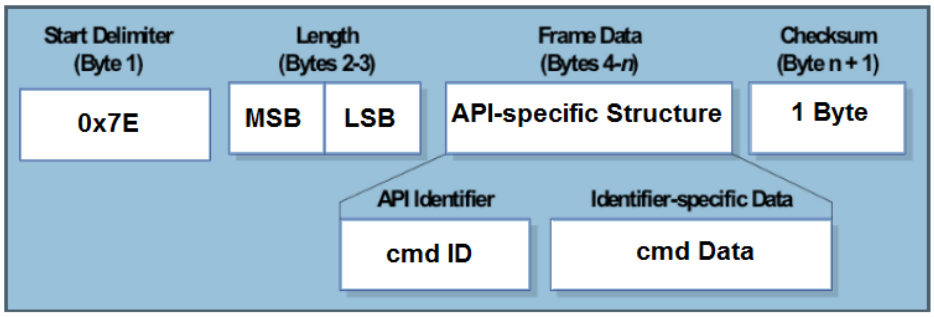

Figura 3-28: Estrutura da *trama*

Assim, o conjunto de dados difundidos na rede é composto pelos seguintes campos:

- Byte indicador de início da trama: - *Start Delimiter*:
- Dois bytes que indicam o tamanho da mensagem: *Frame Data Length*
- Estrutura composta pelo tipo de comando e mensagem: *Frame Data*
    - Tipo de comando – *API Identifier* (*AT, Remote Request, Remote Response,* entre outras);
    - Conteúdo da trama com a estrutura interna de dados que varia de acordo com o tipo de comando que é utilizado – *Identifier-specific Data*
- Byte que valida a integridade da trama - *Checksum*

3.7.4. Comandos AT/API

De acordo com a documentação disponibilizada pelo fabricante, as mensagens podem ser de dois tipos e cada tipo de mensagem possui diferentes tipos de comandos, que podem ser usados para interagir com os módulos.

- "**AT *Command***" este tipo de comando usa-se quando se pretende enviar diretamente uma mensagem para o coordenador da rede. Isto é, tratar-se-á de um dispositivo final, em que a sua mensagem terá apenas como destinatário o coordenador. Por isso, não é necessário especificar o endereço do destinatário, tornando a trama da mensagem bastante simples.
- "***Remote Command Request***" é usado este tipo de mensagem quando se pretende enviar um comando para um módulo remoto. Neste caso é necessário enviar o endereço do destinatário ou efetuar uma transmissão em difusão (*broadcast*), em que todos os módulos da rede poderão receber a mensagem. As mensagens chegarão sempre ao coordenador, que será o responsável por fazê-la chegar ao destinatário.

É o próprio *firmware* do XBee que realiza o tratamento necessário para que as mensagens sejam difundidas com sucesso entre os módulos, de acordo com o protocolo ZigBee. É também o coordenador o responsável pelos avisos, caso uma mensagem não tenha chegado ao destino. Esta funcionalidade é de máxima importância para garantir a confiança no sistema.

A comunicação necessária entre os módulos para a gestão e controlo dos dispositivos faz-se através dos comandos disponibilizados pela API do XBee.

Para um conhecimento da totalidade dos comandos disponibilizados, pode ser consultado o manual do XBee disponibilizado pelo fabricante [9].

Na Figura 3-29 está representada a estrutura dos comandos para serem enviados aos módulos XBee e seguidamente são apresentados alguns dos comandos AT.

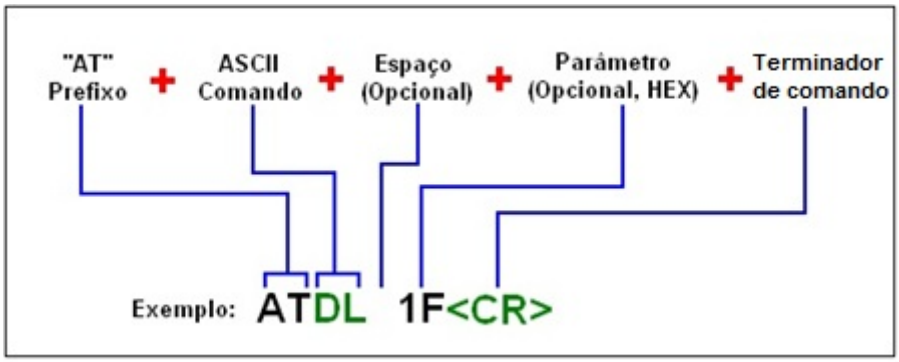

Figura 3-29: Formato dos comandos AT

Para se configurar o módulo XBee pode-se fazer de diversas formas. No entanto, a mais utilizada é através da ferramenta X-CTU usando a janela "terminal".

Para que o XBee entre em modo de comando é necessário enviar a instrução "+++" e nos 10 segundos seguintes tem de ser enviado o comando AT que se pretende. Assim, para ler o endereço de destino, por exemplo, tem-se de escrever **ATDL<ENTER>,** e aguardar pela resposta do XBee. Alguns comandos usados para a configuração do XBee, são agora descritos.

- **ATDL**5001<ENTER> Altera o endereço de destino para 5001 do XBee, devolve "OK".
- **ATMY**5000<ENTER> Altera o próprio endereço para 5000, devolve "OK".
- **ATWR**<ENTER> Grava as alterações feitas na memória *flash* do XBee.

3.7.5. Endereçamento

Para existir comunicação dentro de uma rede entre vários módulos XBee é necessário efetuar os endereçamentos dos vários módulos XBee. Existem três formas de endereçamento que são:

- Trabalhar no mesmo canal – comando ATCH;
- Trabalhar no mesmo PAN ID – comando ATID;
- Trabalhar com um endereço Fonte – comando ATSH, ATSL ou ATMY.

Nesta última forma de endereçamento são indicados ao módulo XBee qual o endereço do módulo XBee de destino.

No Interface TLab, os módulos XBee estão configurados para utilizar o PAN ID 0 (Zero) e a trabalharem no canal 12. Ainda foi usado, no caso dos dispositivos terminais, o endereço de destino, optando-se pelo endereço fixo de fábrica de 64 bits do XBee coordenador, uma vez que é este que recebe os dados de todos os Interfaces TLab, como se mostra na Figura 3-30.

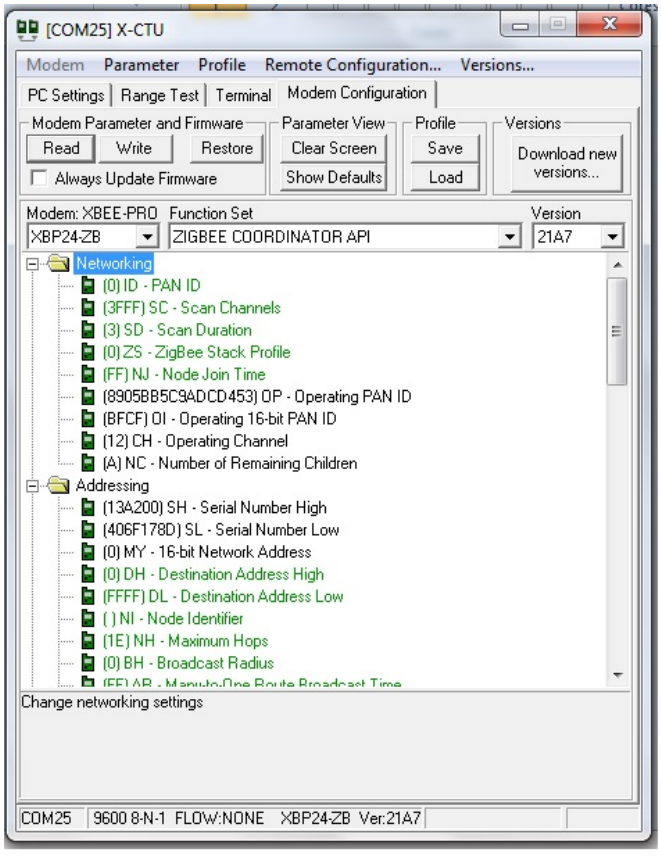

Figura 3-30: Configuração do XBee coordenador utilizando a ferramenta X-CTU

3.8. Descrição e Funcionamento Geral do Interface TLab

Este sistema divide-se em três módulos funcionais distintos, com desenvolvimentos também distintos:

- ***Módulo Hardware*** é o núcleo do sistema. Baseia-se no Arduino e na aplicação desenvolvida com vista à aquisição de grandezas recorrendo a dispositivos sensoriais. Foi no desenvolvimento desta parte do sistema que se investiu mais tempo e tecnologia.
- ***Módulo Sincronização*** é responsável pela sincronização dos sinais obtidos pelos elementos sensoriais, difundidos pelos módulos de comunicação, e a plataforma de monitorização.
- ***Módulo Monitorização*** permite, através do interface homem/máquina e do *software* desenvolvido, facilitar a interpretação dos dados recolhidos.

A Figura 3-31 mostra a configuração do sistema.

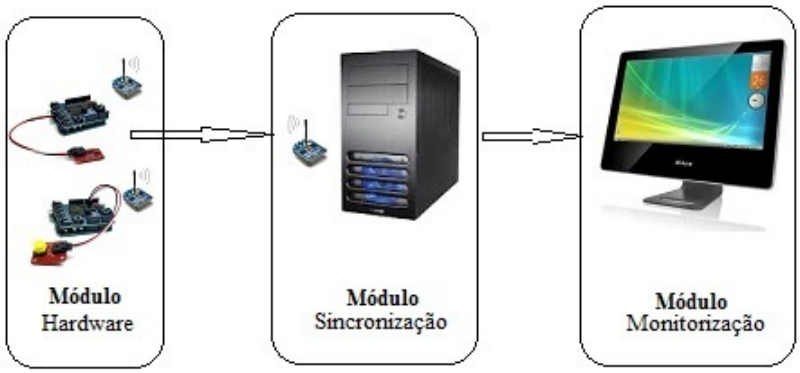

Figura 3-31: Módulos do sistema

O módulo hardware baseia-se num conjunto de sensores ligados ao micro controlador Arduino que, através de uma aplicação desenvolvida e adequada a cada tipo de sensor, irá recolher os dados, processá-los, guardá-los e, por fim, transmiti-los através do módulo de comunicação sem fios.
No módulo sincronização está instalado um módulo de comunicação sem fios, que se destina a coordenar a receção de dados. Através de uma aplicação instalada num computador ou servidor faz-se a deteção dos dispositivos pertencentes à rede, bem como dos sensores a eles acoplados, sendo-lhes atribuídos identificadores. Uma vez em modo de sincronização, os dados são enviados para uma base de dados local ou externa.
O módulo monitorização baseia-se numa janela Web desenvolvida para o efeito, com ligação configurável para aceder à base de dados. Na base de dados define-se, configura-se e distribuem-se os equipamentos. Por exemplo, Sensor com ID 1 está localizado na Zona X da Instituição Y. Esta atribuição de localizações, irá facilitar a visualização dos dados, quer em tempo real, quer no registo histórico. A visualização dos dados faz-se graficamente, podendo-se visualizar em simultâneo, um grande número de sensores.

O esquema de blocos representado pela Figura 3-32 mostra a forma como todo o "Interface TLab" é constituído fisicamente.

O Tlab dispõe de um núcleo formado pelo microcontrolador gerido pelo programa de aquisição de dados. Todos os sensores estão ligados ao micro controlador, usando uma ligação física por cabo adequada a cada sensor, que lhes permite comunicar com este e fornecer as leituras executadas.

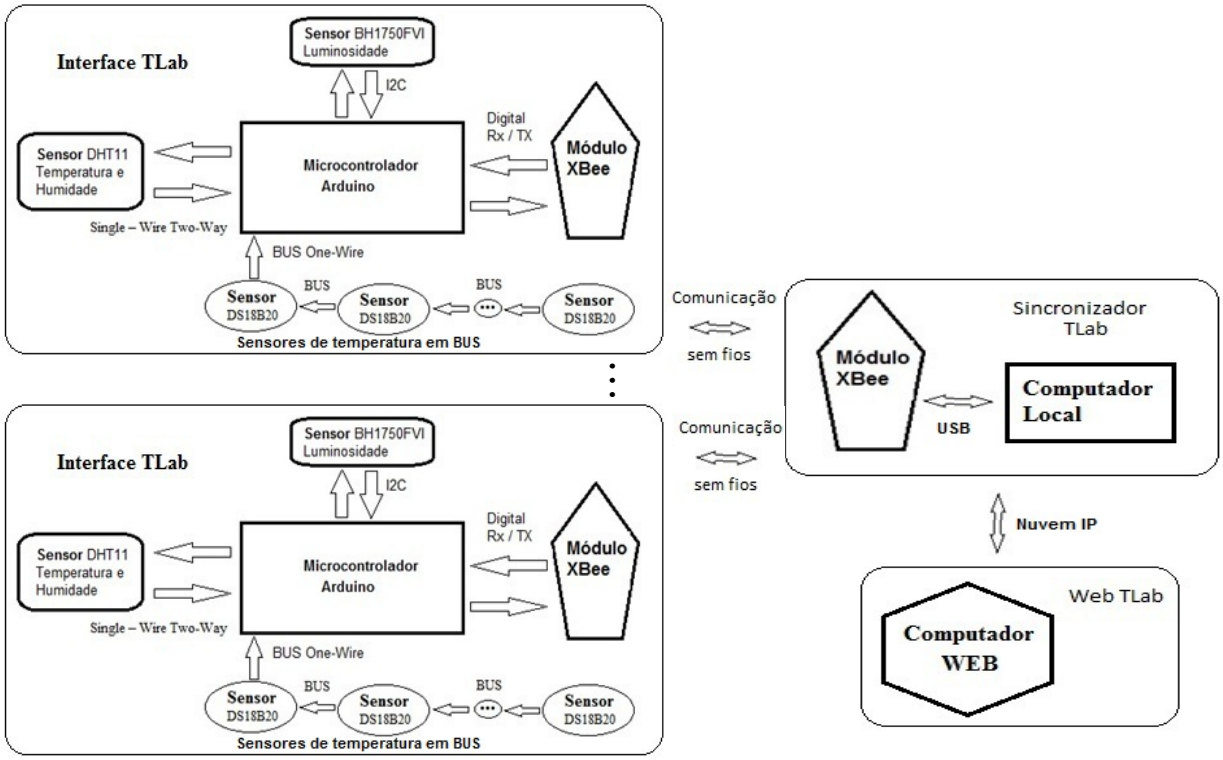

Figura 3-32: Diagrama de blocos da rede do sistema TLab

O programa de aquisição de dados é formado por vários procedimentos que solicitam, de forma sequencial e em ciclo fechado, as leituras dos elementos sensoriais. Os dados são armazenados em variáveis locais, para posterior envio.

Os valores armazenados não são instantâneos, eles resultam de uma média de leituras. O número de leituras que contribui para a média pode ser configurado em função das necessidades de cada instalação.

Após um ciclo de leituras, os dados são acondicionados de forma a serem enviados pelo comunicador XBee. É criado um pacote de dados que será integrado na *Identifier-specific Data* da trama que o comunicador envia ao coordenador.

O pacote de dados é constituído por:

- Identificador alfanumérico – este identificador pode ser numérico para os sensores ligados em BUS, em que o número indicará o número de ordem que o sensor assume na sequência do BUS, ou alfabético, para os restantes sensores.
- Carater de separação de dados – este conjunto de carateres são acrescentados à mensagem para que a mensagem possa ser mais facilmente interpretada e é formada por "- V:" onde a letra é distinta para cada tipo de sensor. Este conjunto de caracteres é desprezado após a validação da mensagem, pelo coordenador.
- Valor – é a parte da mensagem onde seguem os dados, formados por um número decimal com duas casas decimais.
- Separador de dados – é utilizado o carater "," para indicar que termina o conteúdo de dados de um sensor.

O *software desktop* sincronizador TLab, instalado no computador e ligado ao módulo XBee coordenador, tem como função o reconhecimento dos equipamentos instalados e sincronizar o fluxo de dados (Figura 3-33).

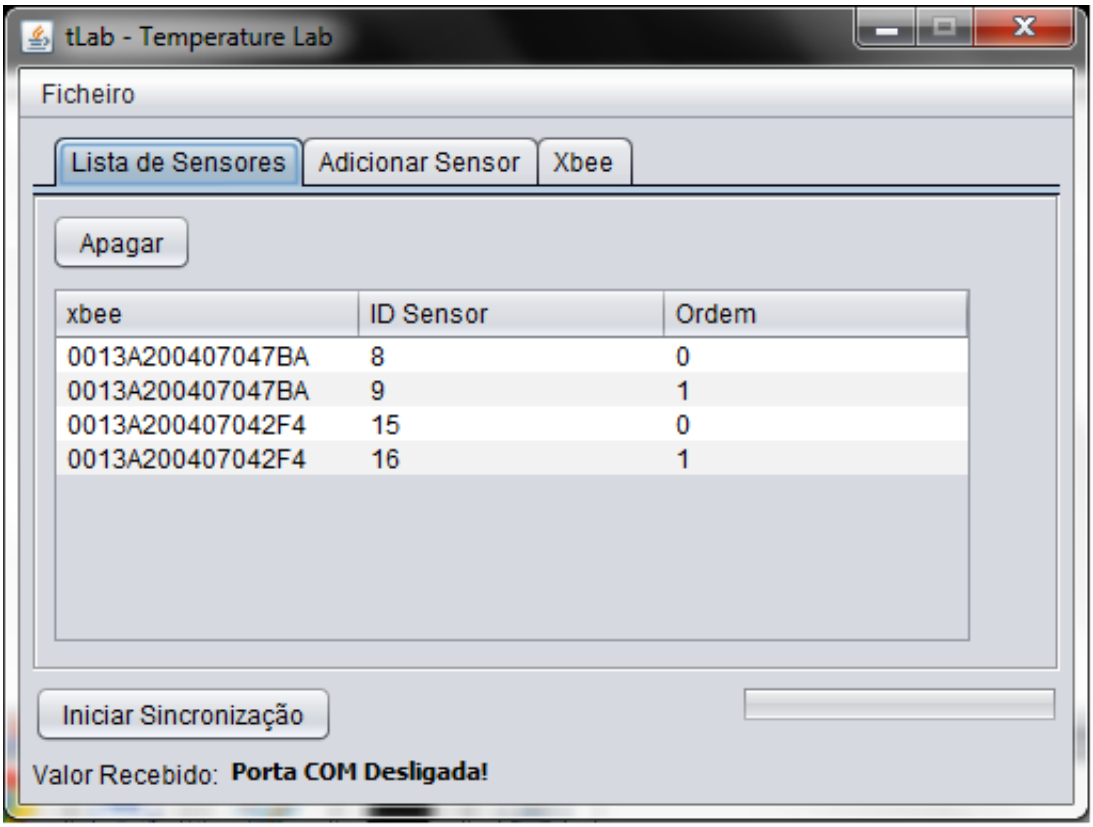

Figura 3-33: Aplicação e sincronização TLab

Numa primeira fase, o sincronizador TLab identifica os comunicadores XBee que fazem parte da rede, gravando o número de cada XBee num ficheiro, assim como os sensores a ele associados. Esta identificação serve para, numa segunda fase, atribuir um identificador a cada dado que entra, para um correto preenchimento da base de dados, dados esses que são depois encaminhados para a aplicação Web TLab.

A aplicação Web TLab recebe e grava cada dado (valor) que recebe numa base de dados MySQL.

O modelo de entidade relacionamento, representa as entidades e os relacionamentos envolvidos na especificação de um sistema. A Figura 3-34 ilustra o modelo físico gerado através da ferramenta *Power Designer*, que representa as entidades e relacionamentos utilizados no sistema TLab, mostrando também os atributos de cada entidade.

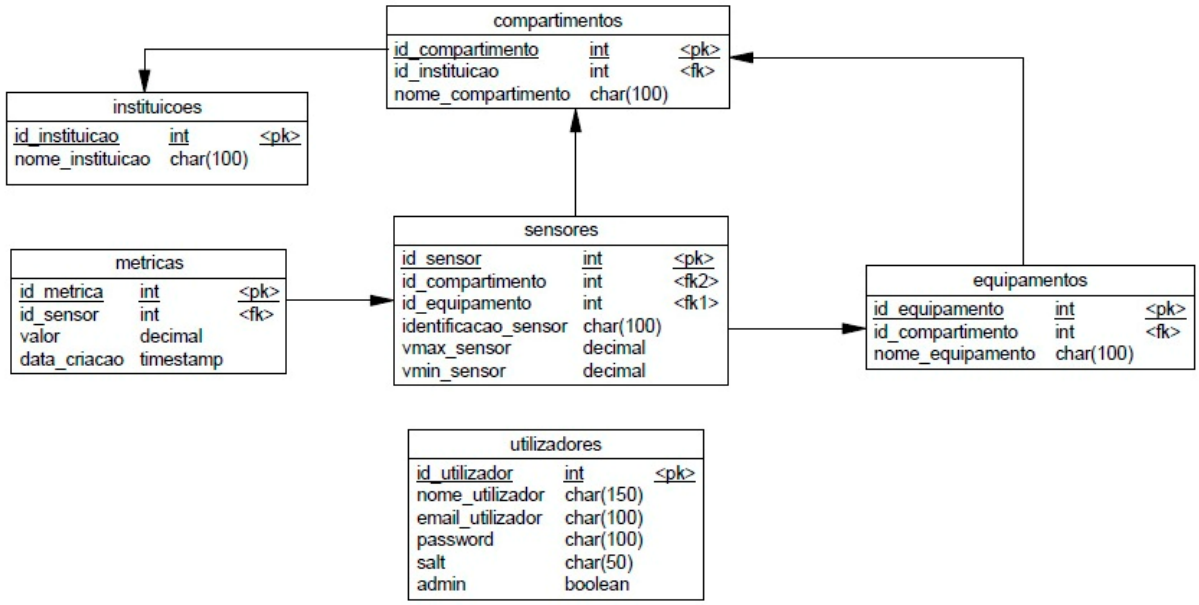

Figura 3-34: Modelo entidade relacionamento da base de dados

Através deste modelo, pode-se verificar que a base de dados da aplicação é muito simples, tendo apenas cinco entidades que se relacionam entre si. A entidade “Utilizadores” não se relaciona com mais nenhuma outra entidade, porque serve apenas para a autenticação de utilizadores. Enquanto que as restantes entidades, se interligam para se poder relacionar os registos que são inseridos na entidade “Metricas”. Estes registos representam os valores que são obtidos para cada sensor, a cada instante do processo de sincronização.

É através da plataforma web que se tem acesso graficamente aos dados recebidos, quer em tempo real, quer, posteriormente, aos históricos registados em base de dados. É ainda nesta plataforma que se definem alguns parâmetros que permitem a gestão de alarmes (Figura 3-35).

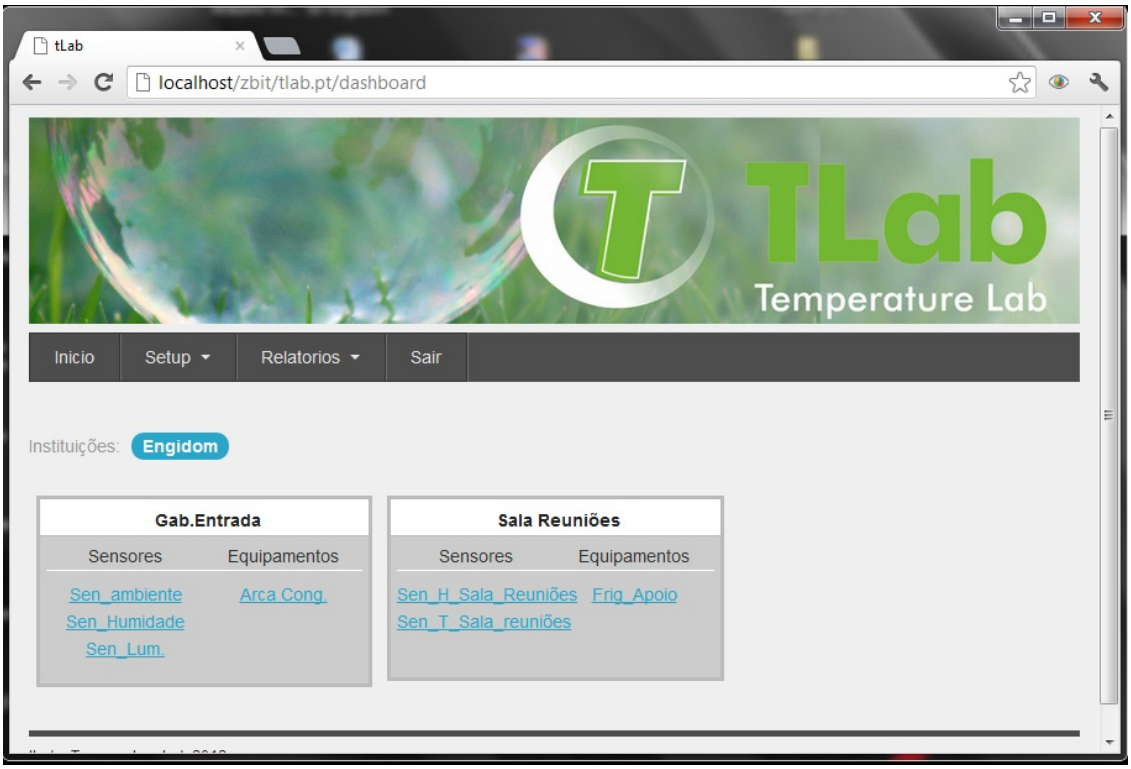

Figura 3-35: Plataforma Web TLab

3.9. Resultados Obtidos

Seguidamente são apresentadas algumas figuras que ilustram os resultados obtidos após alguns ensaios do sistema de aquisição de dados Tlab.

Foram instalados dois interfaces Tlab em locais diferentes e com equipamentos diferentes, assim:

Zona designada por Gab. Entrada – foi instalado um sensor de temperatura e humidade DTH11, um sensor de luminosidade BH1750FVI e dois sensores de temperatura DS18B20 ligados em BUS.

Zona designada por Sala Reuniões – foi instalado um sensor de temperatura e humidade DTH11 e dois sensores de temperatura DS18B20 ligados em BUS.

Pela observação do gráfico da Figura 3-36 que representa os dados recolhidos pelo sensor 1 e sensor 2 do tipo DS18B20 (temperaturas), no período decorrido aproximadamente entres as

9:30 e as 10:30 horas do dia 17 de Setembro de 2012. Os valores estão dentro dos padrões normais de temperatura, por isso não foram gerados alarmes.

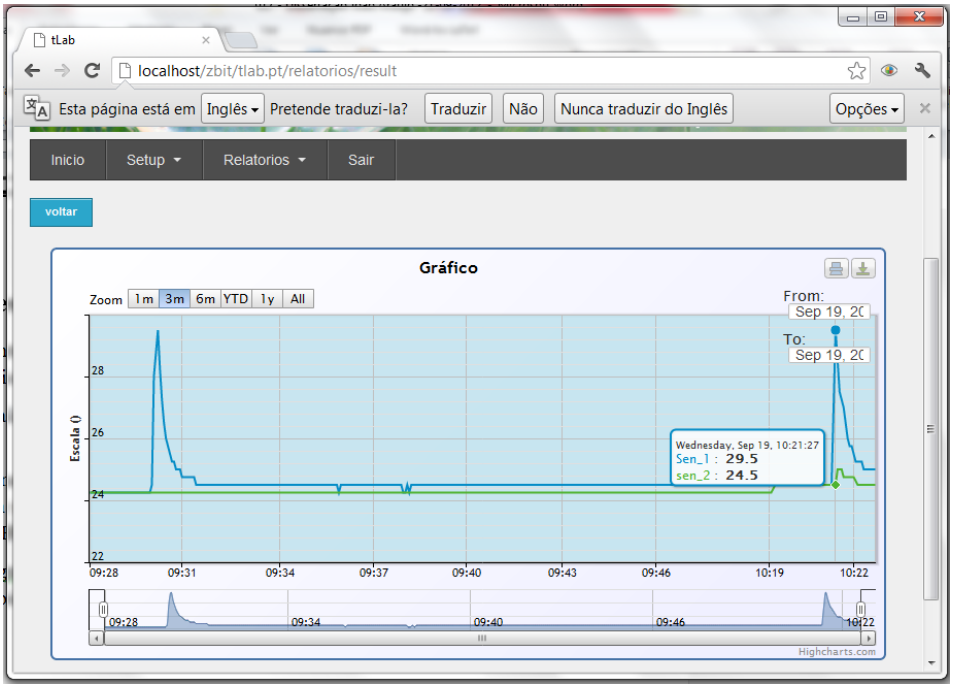

Figura 3-36: Gráfico de temperaturas captadas pelo sensor DS18B20

Podemos observar no gráfico da Figura 3-37 a variação da humidade relativa do ar ao longo de um período de duas horas no dia 6 de Setembro de 2012 e que representa o ensaio do sensor de temperatura e humidade do tipo DHT11.

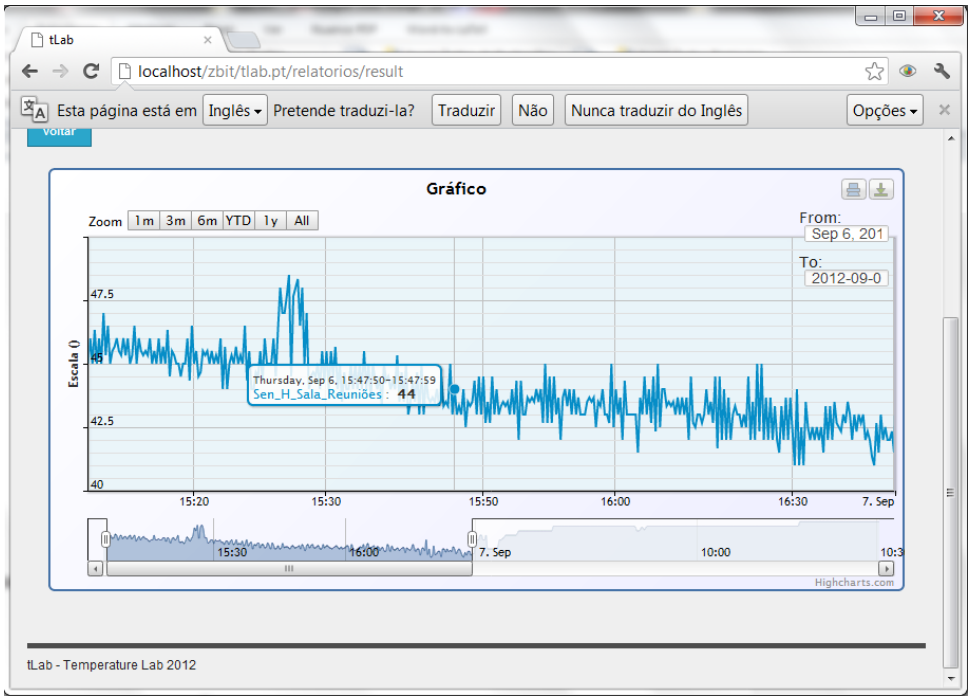

Figura 3-37: Gráfico de humidade relativa

No gráfico da humidade relativa pode observar-se alguma instabilidade dos valores. Esta instabilidade pode ser ultrapassada através de uma filtragem obtida a partir do cálculo da média. Se se definir um número maior de amostras que contribui para a média este problema é resolvido.

No gráfico da luminosidade (Figura 3-38) medida aproximadamente entre as 9 horas e as 10:30 horas do dia 7 de Setembro de 2012 pode-se ver claramente que no período entre as 9 e as 9:45 horas, o nível de luminosidade excedeu o limite máximo configurado (zona demarcada pelo sombreado mais azul da Figura 3-38).

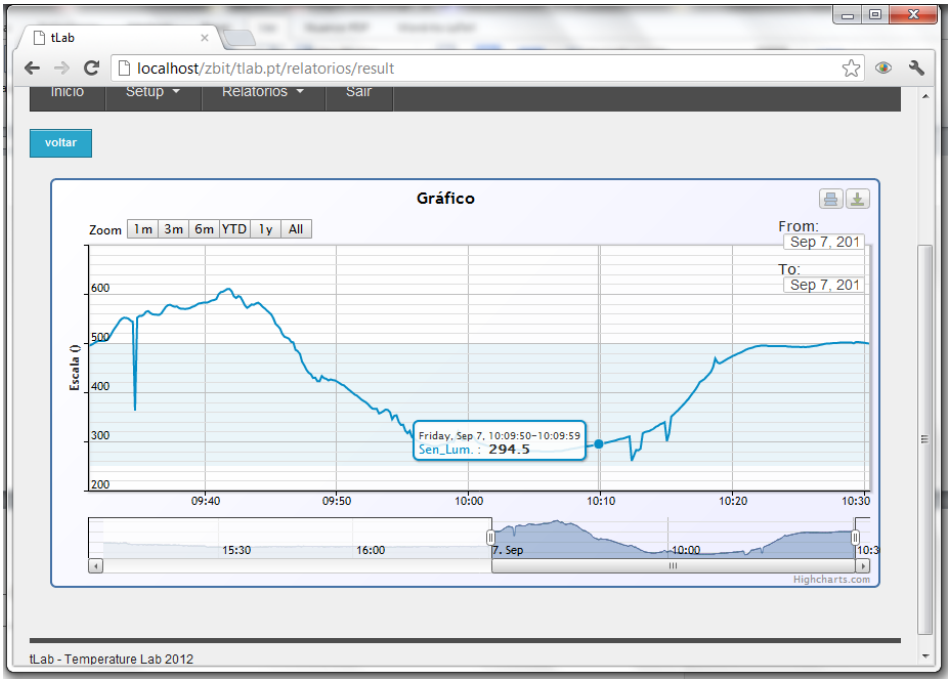

Figura 3-38: Gráfico da luminosidade

O TLab faz uma verificação em tempo real, dos limites configurados para cada sensor. Caso se esteja a monitorizar um determinado sensor e os valores obtidos estejam fora dos parâmetros estipulados pelas configurações, é gerada na janela do próprio gráfico uma mensagem, em roda pé de cor vermelha.

Contudo, se o desvio se verificar num sensor que não esteja a ser monitorizado nesse instante é gerada, igualmente, uma mensagem que é gravada em ficheiro do tipo CSV[50]. Este ficheiro pode ser consultado diretamente através da plataforma Web do TLab (Figura 3-39), ou ser usada por qualquer folha de cálculo compatível com o tipo de ficheiro, para posterior tratamento.

[50] CSV – Comma Separated Values

A mensagem é composta pela data e hora da ocorrência, pela identificação do sensor e o valor de leitura, antecedido do limite configurado, como se mostra na Figura 3-39: Lista de mensagens de alarmes.

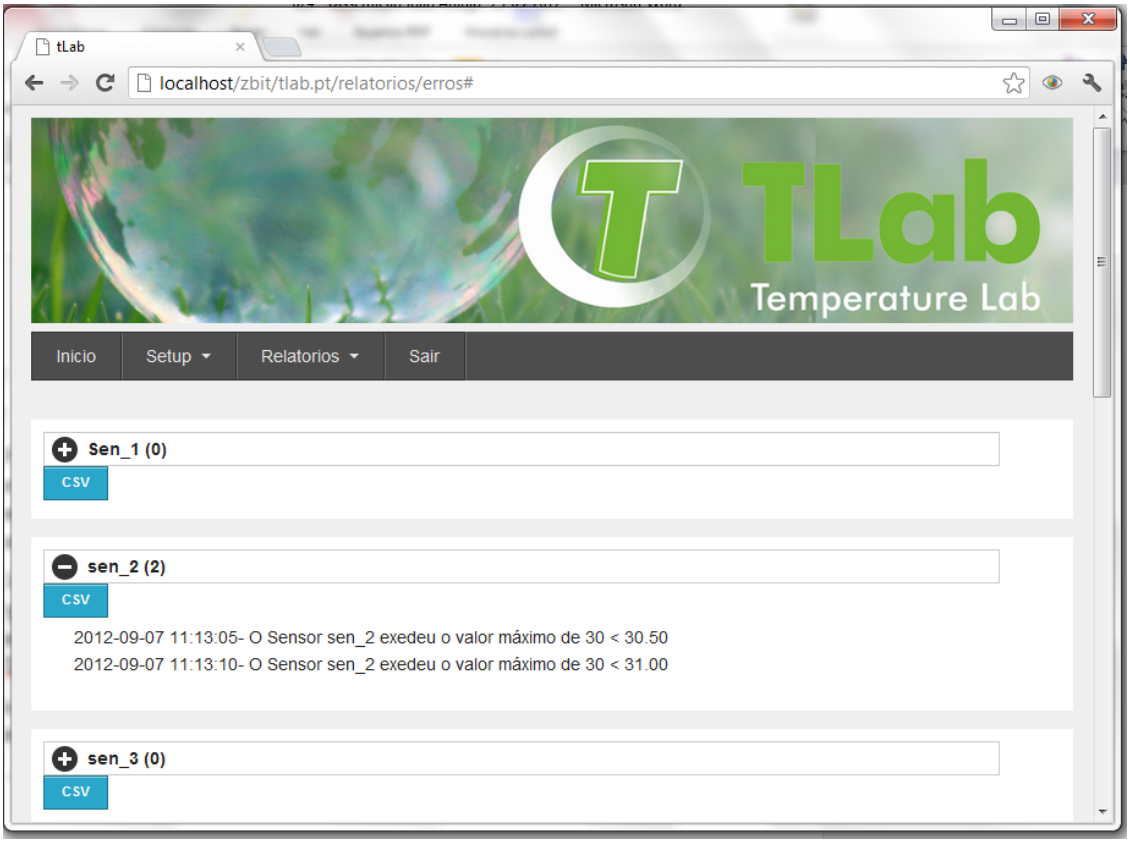

Figura 3-39: Lista de mensagens de alarmes

Capítulo 4 - Conclusões e Trabalhos Futuros

A partir das necessidades verificadas em algumas atividades económicas, nomeadamente na indústria laboratorial e indústria de conservação alimentar, desenvolveu-se um sistema de aquisição de dados e monitorização. Com este sistema verificou-se que é possível monitorizar ambientes de trabalho e equipamentos específicos de conservação ou tratamento, de uma forma económica e com grande facilidade de instalação, uma vez que, se trata de um sistema sem fios. Para além disso, também é possível medir uma grande variedade de grandezas e em número bastante extenso.

Por outro lado desenvolveu-se uma plataforma web, que permite centralizar facilmente todos os dados recolhidos, mesmo de instalações completamente distintas. A base de dados pode estar localizada fora das instalações e ser monitorizada, quer por responsáveis internos das instituições, quer por equipas externas de manutenção, utilizando-se o conceito atual de nuvem de dados "*CLOUD*".

Este sistema permite ainda, através da análise histórica de alarmes de ocorrências, perspetivar e planear intervenções de manutenção corretiva e periódicas, facilitando o planeamento de atividades produtivas e garantindo resultados de qualidade dentro dos parâmetros desejado.

Um dos pontos a considerar em desenvolvimentos futuros, relaciona-se com a gestão de alarmes. O sistema deverá permitir a ligação a um sistema de comunicação telefónica, de modo a poder emitir mensagens de aviso.

Outro ponto de grande importância a considerar no futuro é o controlo de falhas técnicas do sistema, embora se tenham desenvolvido alguns mecanismos e procedimentos para prevenir e avisar possíveis falhas. No entanto, a sua utilização plena foi deixada para futuros desenvolvimentos.

Foi ainda deixado para futuros desenvolvimentos o tratamento dos relatórios escritos. Estes relatórios deverão ser dinâmicos e personalizados, considerando as especificidades de cada instalação.

Outro ponto importante a considerar em futuros desenvolvimentos é a criação de uma plataforma eletrónica com base nos princípios do Arduino, mas mais específica e personalizada para o Interface TLab.

Por último, o Interface TLab possibilita a condicionamento dos ambientes que estão a ser monitorizados. Assim, através da utilização de saídas (digitais ou analógicas) é possível controlar equipamentos que permitam a alteração das grandezas analisadas, de forma que estas se mantenham dentro de parâmetros previamente configurados. Esta funcionalidade será implementada futuramente.

REFERÊNCIAS

[1] International Organisation for Standardisation (ISO), "ISO 9000," no. Implementação de sistemas de gestão da qualidade (SGQ), 1º Publicação 1987.

[2] Instituto Português da Qualidade. (2003, Novembro) Politica de Acreditação - Rastreabilidade das Medições.

[3] Atmel Corporation. (2011, Setembro) [Online]. http://www.atmel.com/Images/doc8161.pdf - Acedido em Agosto 2012

[4] Atmel Corporation. (2012) ATmega640/1280/1281/2560/2561.

[5] Nuno Pessanha Santos - ASPOF EN-AEL - Escola Naval - Marinha. (2009) [Online]. http://www.isegi.unl.pt/docentes/vlobo/escola_naval/MFC/Tutorial%20Arduino.pdf - Acedido em Abril 2012

[6] Brian W. Evans, Arduino Programming Notebook, 2008.

[7] Dallas Semicondutor Corp. [Online]. http://www.maxim-ic.com/products/1-wire/ - Acedido em Abril 2012

[8] Maik Schmidt, *Arduino a Quick Start Guide*.: Susannah Davidson Pfalzer, 2008.

[9] Jonathan Oxer / Hugh Blemings, *Practical Arduino*.: Technology In Action.

[10] ROHM -Semiconductor. (Rev 2011, Outubro) BH1750FVI - Digital 16 bit Serial OutPut Type Ambiente Light Sensor IC.

[11] Digi Internacional Inc. (2008, Setembro) XBee OEM RF Modules - Product Manual. [Online]. http://ftp1.digi.com/support/documentation/90000982_A.pdf - Acedido em Junho 2012

[12] APCER - Associação Portuguesa de Certificação, "Guia Interpretativo ISO 9001:2000," 2003.

[13] Inc. Maxim Integrated Products, Datasheet DS18B20, DS18B20 Programmable Resolution 1-Wire Digital Thermometer.

[14] Pessanha Santos. http://www.isegi.unl.pt/docentes/vlobo/escola_naval/MFC/Slides%20Arduino.pdf. [Online]. http://www.isegi.unl.pt/docentes/vlobo/escola_naval/MFC/Tutorial%20Arduino.pdf - Acedido em Abril 2012

[15] José Maurício Santos Pinheiro -, "As Redes Com ZigBee," no. Redes sem fios, Julho 2004.

ANEXOS

ANEXO 1 – Modelo de Arduino Existentes no Mercado

Neste anexo apresentam-se alguns modelos de Arduino que existem no mercado, alguns destes modelos têm aplicações muito específicas, como é o caso do Arduino LilyPad (Figura Anexo 1- 2)

1) Arduino LEONARDO (Figura Anexo 1- 1)

- Baseada no microcontrolador ATmega32u4
- Tem 20 entradas / saídas digitais (das quais 7 podem ser usadas como saídas PWM e 12 como entradas analógicas)
- Um oscilador de cristal de 16 Mhz
- Uma ligação USB, uma entrada de alimentação, uma ligação ICSP
- Um botão de "*reset*"

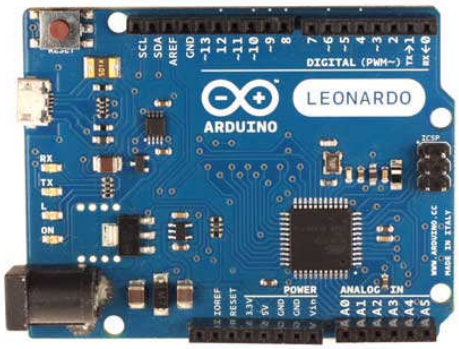

Figura Anexo 1- 1: Arduino Leonardo

2) Arduino LilyPad (Figura Anexo 1- 2)

- Microcontrolador projetado para indústria têxtil (vestuário).
- Pode ser costurada ao tecido e da mesma forma integrar fontes de alimentação, sensores e atuadores
- Baseada no microcontrolador ATmega168V ou ATmega328V

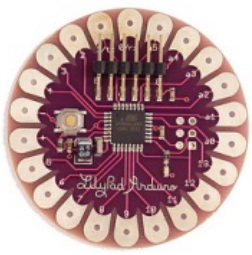

Figura Anexo 1- 2: Arduino Lily Pad

3) Arduino Ethernet (Figura Anexo 1- 3)

- Baseada no microcontrolador ATmega328
- Tem 14 entradas / saídas digitais e 6 entradas analógicas)
- Um oscilador de cristal de 16 Mhz
- Uma ligação RJ45, uma entrada de alimentação, uma ligação ICSP
- Um botão de “reset”

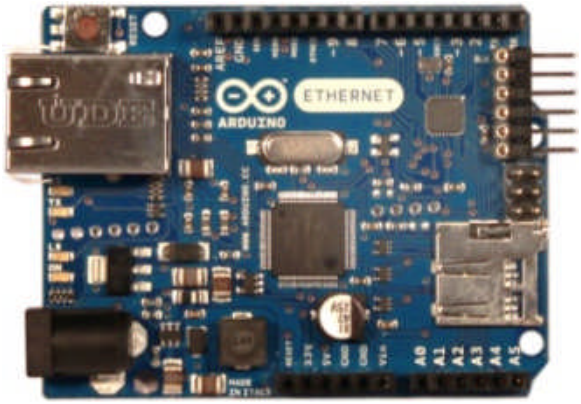

Figura Anexo 1- 3: Arduino Ethernet

ANEXO 2 – Características Sensor DS18B20

Especificações do Sensor DS18B20 (Tabela 2)

Temperature Sensors									
Part Number	Sensor Type	Interface	Accuracy (±°C)	Parasite Pwr.	Temp. Thresh.	Temp. Resolution (bits)	Multi Droppable	Oper. Temp. (°C)	Smallest Available Pckg. (mm2) max w/pins
DS18B20	Local	1-Wire	0.5	Yes	Programmable (NV)	9	Yes	-55 to +125	15.6
						10			
						11			
						12			

Tabela 2 – Características do sensor DS18B20

Modelos e características especificas

PART	TEMP RANGE	PIN-PACKAGE	TOP MARK
DS18B20	-55°C to +125°C	3 TO-92	18B20
DS18B20+	-55°C to +125°C	3 TO-92	18B20
DS18B20/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20-SL/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20-SL+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20U	-55°C to +125°C	8 µSOP	18B20
DS18B20U+	-55°C to +125°C	8 µSOP	18B20
DS18B20U/T&R	-55°C to +125°C	8 µSOP (3000 Piece)	18B20
DS18B20U+T&R	-55°C to +125°C	8 µSOP (3000 Piece)	18B20
DS18B20Z	-55°C to +125°C	8 SO	DS18B20
DS18B20Z+	-55°C to +125°C	8 SO	DS18B20
DS18B20Z/T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20
DS18B20Z+T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20

Tabela 3 – Modelos e características

ANEXO 3 – Características do Sensor DHT11

Especificações do Sensor DHT11 (Tabela 4)

Item	Medição	Gama de Medição	Precisão	Resolução	Ligação
DHT11	Humidade Relativa	20-90% HR	±5%RH	1	Linha única com 3 Fios
	Temperatura	0-50 ℃	±2℃	1	

Tabela 4 – Características Gerais DHT11

Na seguinte tabela disponibilizada pelo fabricante podem ser analisadas mais em pormenor as características do sensor DHT11.

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25℃		±4%RH	
	0-50℃			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0℃	30%RH		90%RH
	25℃	20%RH		90%RH
	50℃	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25℃, 1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
Temperature				
Resolution		1℃	1℃	1℃
		8 Bit	8 Bit	8 Bit
Repeatability			±1℃	
Accuracy		±1℃		±2℃
Measurement Range		0℃		50℃
Response Time (Seconds)	1/e(63%)	6 S		30 S

Tabela 5 – Especificações Técnicas do DHT11

ANEXO 4 – Características dos Módulos Xbee

Especificações dos módulos XBee (Tabela 6)

	XBee	XBee-Pro
Potência de saída	1 mW (0 dBm)	60 mW (18 dBm)
Alcance interior	até 30 m	até 100 m
Alcance Exterior	até 100 m	até 1600 m
Sensibilidade do receptor	-92 dBm	-100 dBm
Frequência de operação	ISM 2.4000 a 2.4835 GHz	
Taxa de transmissão	250 kbps	
Taxa de dados da Interface	115.2 kbps	
Tensão de alimentação	2.8V a 3.4V	
Corrente de transmissão (típica)	45 mA @ 3.3 V	215 mA @ 3.3 V
Corrente de Recepção (típica)	50 mA @ 3.3 V;	55 mA @ 3.3 V
Corrente em modo *Sleep*	<10 µA	
Dimensões	2.438cm x 2.761cm	2.438 cm x 3.294 cm
Peso	3 g	4 g
Temperatura de operação	-40 a 85º C (industrial)	

Tabela 6 – Características módulo XBee

Modelos e características específicas dos módulos XBee.

	Specifications*				Networking Features		
Product	Frequency	Power Output	Maximum Range	RF Data Rate	Protocol	Multipoint	Mesh
XBee ZB	2.4 GHz	1.25/2 mW	120 m	250 Kbps	ZigBee		✔
XBee-PRO ZB	2.4 GHz	63 mW*	3.2 km	250 Kbps	ZigBee		✔
XBee ZB SMT	2.4 GHz	3.1/6.3 mW	1200 m	250 Kbps	ZigBee		✔
XBee-PRO ZB SMT	2.4 GHz	63 mW	3.2 km	250 Kbps	ZigBee		✔

XBee 802.15.4	2.4 GHz	1 mW	90 m	250 Kbps	802.15.4	✔	
XBee-PRO 802.15.4	2.4 GHz	63 mW*	1.6 km	250 Kbps	802.15.4	✔	
XBee-PRO XSC	900 MHz	100 mW	24 km**	9.6 Kbps	Proprietary	✔	
XBee-PRO 900	900 MHz	50 mW	10 km**	156 Kbps	Proprietary	✔	
XBee-PRO DigiMesh 900	900 MHz	50 mW	10 km**	156 Kbps	DigiMesh		✔
XBee DigiMesh 2.4	2.4 GHz	1 mW	90 m	250 Kbps	DigiMesh		✔
XBee-PRO DigiMesh 2.4	2.4 GHz	63 mW*	1.6 km	250 Kbps	DigiMesh		✔
XBee-PRO 868	868 MHz	500 mW	80 km**	24 Kbps	Proprietary	✔	
XBee Wi-Fi	2.4 GHz	16 dBm	300 m	65 Mbps	802.11bgn	✔	
XBee 865LP	865 MHz	12dBm	4 km	80 kbps	Proprietary (Multipoint and DigiMesh)	✔	✔

ANEXO 5 – Código para Arduino do TLab

Neste anexo está transcrito o código em linguagem C para Arduino, instalado nos Interfaces TLab. Existem algumas funções e procedimentos que se destinam a melhorias futuras e outras que servem para verificações de funcionamento na fase de instalação.

CODIGO:

```
/* Aplicação para Monitorização de Ambientes Controlados
 Para Interface TLab
 Light Sensor BH1750 V1.0 - medição de luminosidade
 DHT11 Sensor de Humidade e Temperatura
 DS18 Sensor Digital de Temperatura
 Comunicação wireless XBee Series 2 e Pro
 */

/* ////////////////////////////( Bibliotecas )/////////////////////////////////////////*/

#include <dht11.h> // DHT11
#include <Wire.h> // BH1750 IIC Mode
#include <math.h> // Clac. Matemático
#include <OneWire.h> // comunicação do BUS
#include <DallasTemperature.h> // Sensor Dallas DS18

/* ////////////////////////( Declaração de objectos )////////////////////////////////*/

dht11 DHT11;


/* ////////////////////( Declaração Constantes, Pinos de Entrada )////////////////////*/

#define DHT11PIN 18
int BH1750address = 0x23; // Definição de endereço I2C
byte buff[2];

/* /////////////////( Ligação do BUS na entrada 3 Digital do Arduino) ///////////////*/

#define ONE_WIRE_BUS 3
#define TEMPERATURE_PRECISION 10

/* /////////////( Ativação do BUS de comunicação "oneWire" para qualquer dispositivo do tipo OneWire */

OneWire oneWire(ONE_WIRE_BUS);

/* //////////( passagem de referencia oneWire para sensores de temperatura Dallas ) */
```

```
DallasTemperature sensors(&oneWire);

int numberOfDevices; // Variavel para armazenar o nº de sensores encontrados
int numberOfSample; // Variavel para armazenar o nº de ciclos de amostras
void Menuentrada();
DeviceAddress tempDeviceAddress; // Variavel para armazenar os endereços dos dispositivos encontrados
float myTemperatura [10];
int nDevice;       // Variavel para contador sensores encontrados
float tempC;
/* ////////////////////////////////// ( Void Setup ) ////////////////////////////////*/
/* ////////////////////////////////////////////////////////////////////////////////*/

void setup()
{
 Serial.begin(9600); // Iniciar a Porta serie

 // inicializar a Bibliotecas
 Wire.begin();
 sensors.begin();
 // Atribui á variavelo valor da contagem de dispositivos
 numberOfDevices = sensors.getDeviceCount();

 // Localiza os dispositivos no BUS
 Serial.print("Dispositivos localizados...");

 Serial.print(numberOfDevices, DEC);
 Serial.println(" Sensores.");

 // Verificação do estado da alimentação "parasite power" dos Sensores
 Serial.print("Alimentacao dependente ");
 if (sensors.isParasitePowerMode())
 {
  Serial.println("Ligada");
  Serial.println("Sensores alimentados pelo cabo de BUS");
 }
 else
 {
  Serial.println("Desligada");
  Serial.println("Sensores alimentados por fonte externa");
 }

 // Loop atravéz dos dispositivos para recolher os endereços
 for(int i=0;i<numberOfDevices; i++)
 {
  // procura os endereços no BUS
  if(sensors.getAddress(tempDeviceAddress, i))
  {
   Serial.print("Dispositivo ");
```

```
      Serial.print(i, DEC);
      Serial.print(" com endereco: ");
      printAddress(tempDeviceAddress);
      Serial.println();

      Serial.print("Ajuste de resolucao para ");
      Serial.println(TEMPERATURE_PRECISION,DEC);

      // Ajusta a resolução para 9 bit (Cada sensor Dallas/Maxim podem ter diferentes resoluções)
      sensors.setResolution(tempDeviceAddress, TEMPERATURE_PRECISION);

      Serial.print("Resolucao atual ajustada: ");
      Serial.print(sensors.getResolution(tempDeviceAddress), DEC);
      Serial.println();
    }
    else{
      Serial.print("Encontrado um suposto dispositivo em ");
      Serial.print(i, DEC);
      Serial.print(" mas nao foi possivel verificar o endereço. Verifique a alimentacao e cabos");
    }
  }
  //  Menuentrada(); só usado para testes
}
/* ////////////////////////////////// (Fim setup )//////////////////////////////*/

/* ////////////////////////////////// (Void loop; Corpo do programa) ////////////*/

void loop()

{
  numberOfSample = 10;
    LerDHT11();
    LerBHL750();
    LerDS18();

  /* ////////////////////// (Zona de Menuentrada só para testes de instalção)////

   if (Serial.available() < 0)
    {
     uint8_t letra = Serial.read();
     switch (letra)
     {

   switch (letra)
   {
     case 'h':
     LerDHT11();
     Menuentrada();
     break;
```

```
    case 'l':

    LerBHL750();
    Menuentrada();
    break;
    case 't':
    LerDS18();
    Menuentrada();
    break;

    case 'r':
    LerDHT11();

    LerBHL750();
    LerDS18();

    Menuentrada();
    break;
    default: Serial.println("Opcao incorreta");
   }

   */
}

/* ////////////////////////////////(Fim do corpo do programa principal ) ////////////////////// */

/* ////////////////////////////////(Procedimentos de Leitura das sondas)/////////////////////////*/

/* ///////(Leitura das sondas DHT11 Temperatura e Humidade Relativa)/////////////*/


void LerDHT11()
{

  Serial.println("Temperatura e Humidade");
  Serial.println("\n");

  int chk = DHT11.read(DHT11PIN);

//  Serial.print("Ler Sensor Humidade e Temperatura (DHT11): ");

  /* ////////////// (Validação dos Dados da Sonda DHT11) ///////////////*/

  switch (chk)
  {
  case 0:
    Serial.println("Dados Corrtos");
    break;
  case -1:
```

```
    Serial.println("Erro nos Dados");
    break;
  case -2:
    Serial.println("Tempo de leitura excedido");
    break;
  default:
    Serial.println("Erro desconhecido");
    break;
  }

  */

  /* /////////////// (Apresentação dos Dados) ////////////////*/

  /* //////// ( Temperatura e Humidade Relativa)////////*/

 // Serial.print("Humidade (%): ");
  Serial.print ("h");
  Serial.print ("-H:");
  Serial.print((float)DHT11.humidity, 2);
  Serial.print(",");

//  Serial.print("Temperatura (oC): ");
  Serial.print ("t");
  Serial.print ("-T:");
  Serial.print((float)DHT11.temperature, 2);
  Serial.print(",");

  /*
  //////// Desenvolvimento futuro //////////
  Apresentação de dados em unidades de temperatura diferentes

  Serial.print("Temperatura (oF): ");
   Serial.println(Fahrenheit(DHT11.temperature), 2);

   Serial.print("Temperatura (K): ");
   Serial.println(Kelvin(DHT11.temperature), 2);

   Serial.print("Ponto de Orvalho (oC): ");
   Serial.println(dewPoint(DHT11.temperature, DHT11.humidity));

   Serial.print("Ponto de Orvalho Rapido (oC): ");
   Serial.println(dewPointFast(DHT11.temperature, DHT11.humidity));
   */

  delay(1000);

}
```

```
/* /////////(Leitura das sondas BHL 750 Luminosidade)//////////*/

void LerBHL750()
{
  //Serial.println("Luminosidade");

  /* //////// ( Luminosidade)////////*/

  int i;
  uint16_t val=0;
  BH1750_Init(BH1750address);

  if(2==BH1750_Read(BH1750address))
    {
      val=((buff[0]<<8)|buff[1])/1.2;
      // Serial.println(" ");
      // Serial.print("Luminosidade do Local (Lx): ");
      Serial.print ("l");
      Serial.print ("-L:");
      Serial.print(val,DEC);
      Serial.print (",");
    }
  delay(1000);
}

/* /////////////////(Leitura das sondas DS18B20 Temperatura)///////////////*/

void LerDS18()
{
  float acomulado;
  float tempArmazenada;
  float valor;
  float myTemperatura[]={0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

  //Serial.println("Temperatura DS18B20");

  /* //////// ( Temperaturas Sensores Dallas DS18 )////////*/

  // Faz o pedido de temperatura a todos os dispositivos do BUS
  // ----- Serial.print("Lendo Temperaturas...");

  sensors.requestTemperatures(); // envia o comando de pedido de temperatura

  // ----- Serial.println("Concluido");

  // Loop a todos os dispositivos para obter as temperaturas
   for(int contador=0;contador<numberOfSample; contador++)
    {
    for(int nDevice=0;nDevice<numberOfDevices; nDevice++)
```

```
      {
      // Search the wire for address
      if(sensors.getAddress(tempDeviceAddress, nDevice))
      {
/* ///////// (linhas comentadas usadas para testes iniciais) /////////*/
      // Escreve o ID do dispositivo
      //Serial.print("Temperatura do dispositivo: ");
      // Serial.println(nDevice,DEC);
      // mediaTemperature(tempDeviceAddress); // calcula a Média
      // Resposta imediata dos sensores

      printTemperature(tempDeviceAddress); //  para ler os dados

      //Serial.print("temperatura armazenada :");
      //Serial.println(tempC);

      tempArmazenada = myTemperatura[nDevice]+tempC;
      myTemperatura[nDevice] = tempArmazenada;
      valor = myTemperatura[nDevice];

      //Serial.print ("coluna ");
      //Serial.print (nDevice);
      //Serial.print (" Acomulado: ");
      //Serial.println (valor);

      }
      //verifica o estado das ligações através da resposta do endereço
      else
      {
        Serial.println("Ocorreu um problema – verifique cabos e ligações dos sensores (Erro 01) ");
      }
    }
  }
// Serial.println (" ******** ****** Para valores medios ****** *******");
  for(int nDevice=0;nDevice<numberOfDevices; nDevice++)
    {
      // Serial.print ("Contador ");
      // Serial.println (contador);
      float acomulado = myTemperatura[ nDevice];
      //    Serial.print ("Sensor ");
      Serial.print (nDevice);
      Serial.print ("-T:");
      Serial.print(acomulado/numberOfSample);
      Serial.print(",");
    }
}


/*////////////////////////////////// ( Bloco de Funções  ) //////////////////////////////////*/
```

```
//

void Menuentrada()
{
  Serial.println("");
  Serial.println("");
  Serial.println("Para leitura dos sensore digite uma das letras ");
  Serial.println("");
  Serial.println("Para leitura de Humidade Relativa e temperatura do Local - Tecla [h] ");
  Serial.println("Para leitura da luminosidade do Local                - Tecla [l] ");
  Serial.println("Para leitura das Temperaturas das Sondas             - Tecla [t] ");
  Serial.println("Para leitura de Todas as Sondas                      - Tecla [r] ");
  //  Serial.println("");
}


/* ////////////////////////////////////(Conversão de Graus Celsius em Fahrenheit) ////////////////////*/

double Fahrenheit(double celsius)
{
  double fahrenheit = 1.8 * celsius + 32;
  return fahrenheit;
}

/* ////////////////////////////////////(Conversão de Graus Celsius em Kelvin) ////////////////////*/

double Kelvin(double celsius)
{
  double kelvin = celsius + 273.15;
  return kelvin;
}

/* ////////////////////////////////////// (Função calculo do Ponto de Orvalho ////////////////////////////*/
/* ////////////// (segundo a National Oceanic and Atmospheric Administration ) /////////////*/
/* ////////////// referencia: http://wahiduddin.net/calc/density_algorithms.htm ///////////*/

double dewPoint(double celsius, double humidity)
{
  double Porvalho;
  double A0= 373.15/(273.15 + celsius);
  double SUM = -7.90298 * (A0-1);
  SUM += 5.02808 * log10(A0);
  SUM += -1.3816e-7 * (pow(10, (11.344*(1-1/A0)))-1) ;
  SUM += 8.1328e-3 * (pow(10,(-3.49149*(A0-1)))-1) ;
  SUM += log10(1013.246);
  double VP = pow(10, SUM-3) * humidity;
  double T = log(VP/0.61078);   // variavel de temperatura
  Porvalho = (241.88 * T) / (17.558-T);
  return Porvalho;
```

```
}

/* ////////////////////////// (Função calculo do Ponto de Orvalho Rápido //////////////////*/
/* ////////////////////////// (Variação max = 0.6544 wrt Ponto de Orvalho())//////////////*/
/* ////////////////////////// (5x mais rápido doque o Ponto de Orvalho())//////////////////*/
/* ////////////////////////// referencia: http://en.wikipedia.org/wiki/Dew_point //////////*/

double dewPointFast(double celsius, double humidity)
{
  double a = 17.271;
  double b = 237.7;
  double temp = (a * celsius) / (b + celsius) + log(humidity/100);
  double Td = (b * temp) / (a - temp);
  return Td;
}

/* /////////// (Funções Leitura dos dados do sensor de Luminosidade //////////////////*/

int BH1750_Read(int address) //
{
  int i=0;
  Wire.beginTransmission(address);
  Wire.requestFrom(address, 2);
  while(Wire.available()) //
  {
    buff[i] = Wire.receive();  // receive one byte
    i++;
  }
  Wire.endTransmission();
  return i;
}

void BH1750_Init(int address)
{
  Wire.beginTransmission(address);
  Wire.send(0x10);//1lx reolution 120ms
  Wire.endTransmission();
}

/* ////////////////////// Função de apresentação da temperatura do dispositivo//////////////////*/

float printTemperature(DeviceAddress deviceAddress)
{
    tempC = sensors.getTempC(deviceAddress);

   // Usado para testes
   // Serial.print("Temp C: ");
   // Serial.print(tempC);
   // Serial.print(" Temp F: ");
```

```
  // Serial.println(DallasTemperature::toFahrenheit(tempC)); // Converte temperatura Cº em Fahrenheit

   return tempC;
}

void mediaTemperature(DeviceAddress deviceAddress)
{
 float acomulado;
 for(int contador=0;contador<numberOfSample; contador++)
   {
    acomulado = acomulado+sensors.getTempC(deviceAddress);
   }
 float media = acomulado/numberOfSample;
  // Serial.print("Temp C: ");
  // Serial.println(media);
}

/* ///////////////// (Função apresentar o endereço dos dispositivos)///////*/

void printAddress(DeviceAddress deviceAddress)
{
 for (uint8_t i = 0; i < 8; i++)
 {
   if (deviceAddress[i] < 16) Serial.print("0");
   Serial.print(deviceAddress[i], HEX);
 }
}
/* ( THE END ) */
```