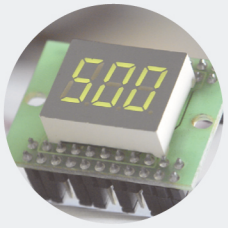
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES






Sistema de Controlo Eletrónico de uma Estufa

RODRIGO SENCADAS DA SILVA
Outubro de 2019

POLITÉCNICO
DO PORTO

SISTEMA DE CONTROLO ELETRÓNICO DE UMA ESTUFA

Rodrigo Sencadas da Silva

Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização de Automação e Sistemas
Departamento de Engenharia Electrotécnica
Instituto Superior de Engenharia do Porto

2019

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

Candidato: Rodrigo Sencadas da Silva, Nº 1110453, 1110453@isep.ipp.pt

Orientação científica: Avelino Marques, aav@isep.ipp.pt

Supervisão: Avelino Marques, aav@isep.ipp.pt

Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

9 de outubro de 2019

Agradecimentos

Agradeço ao orientador científico Professor Doutor Avelino Marques pela sua disponibilidade e ter aceitado o cargo de orientador que possibilitou a realização desta tese.

Resumo

Desde tempos antigos que existe a ideia do cultivo de plantas em estruturas climatéricas controladas e fora da época em que normalmente nascem. As estufas permitem suprimir essa necessidade e com os avanços tecnológicos a sua construção tornou-se mais eficiente e barata, levando também ao inevitável surgimento das estufas inteligentes a permitirem a recolha de informação ambiente e atuação automática de ações de acordo com as necessidades das plantas cultivadas, para além da monitorização e controlo de forma remota.

Conceitos como IoT permitem não só a monitorização do microclima gerado numa estufa através de aplicações móveis e o ajuste dos fatores que afetam o cultivo, como também processar e guardar informação sobre colheitas e clima através de tecnologias de informação de *Machine Learning*.

De forma a validar o conceito das estufas inteligentes, foi desenvolvido um sistema com um módulo local e um módulo remoto (constituído por dois subsistemas: RemoteXY e *ThingSpeak*) que transmitem informação entre si através da comunicação *WiFi*. Este primeiro módulo foi desenvolvido com o intuito de proporcionar uma interface de utilizador local, através de um LCD e um conjunto de botões, que permita recolher e visualizar dados fornecidos pelos sensores presentes e controlar os atuadores de forma manual ou definir as ações de atuação automática. O módulo remoto, por sua vez, recebe a informação ambiental em tempo real numa interface gráfica em ambos os subsistemas constituintes e se necessário, controla os atuadores do módulo local à distância através da aplicação RemoteXY. No final, é avaliado a fiabilidade da solução projetada.

Palavras-Chave

Estufas inteligentes, IoT, automatização, controlo remoto, AWS.

Abstract

Since ancient times there has been the idea of growing plants in controlled climate structures and outside the time in which they are normally born. Greenhouses make it possible to suppress this need and with technological advances their construction has become more efficient and cheaper, also leading to the inevitable emergence of smart greenhouses that allow the collection of environmental information and automatic action according to the needs of cultivated plants, in addition to remote monitoring and control.

Concepts such as IoT not only allow monitoring of greenhouse-generated microclimate through mobile applications and adjusting of factors affecting cultivation, but also process and store crop and climate information through Machine Learning information technologies.

In order to validate the concept of smart greenhouses, a system has been developed with a local module and a remote module (consisting of two subsystems: RemoteXY and ThingSpeak) that transmit information to each other via WiFi communication. This first module was developed to provide a local user interface through an LCD and a set of buttons, allowing to collect and view data provided by sensors and to control the actuators manually or to define the actions of automatic actuation. In turn, the remote module receives real-time environmental information in a graphical interface in both constituent subsystems and, if necessary, controls the actuators of the local module remotely through the RemoteXY application. In the end, the reliability of the designed solution is evaluated.

Keywords

Smart greenhouse, IoT, automation, remote control, AWS.

Résumé

Depuis l’antiquité, on a eu l’idée de cultiver des plantes dans des structures climatiques contrôlées et en dehors de leur période de naissance normale. Les serres permettent de supprimer ce besoin et, avec les avancées technologiques, leur construction est devenue plus efficace et moins chère, ce qui a également entraîné l'émergence inévitable de serres intelligentes permettant la collecte d'informations environnementales et une action automatique en fonction des besoins des plantes cultivées, en plus de la surveillance et du contrôle à distance.

Des concepts tels que l'IoT permettent non seulement de surveiller le microclimat généré par l'effet de serre grâce à des applications mobiles et à l'ajustement de facteurs affectant la culture, mais également de traiter et de stocker des informations sur les cultures et le climat grâce aux technologies de l'information Machine Learning.

Afin de valider le concept de serre intelligente, un système a été développé avec un module local et un module distant (composé de deux sous-systèmes: RemoteXY et ThingSpeak) qui se transmettent des informations via une communication Wi-Fi. Ce premier module a été développé pour fournir une interface utilisateur locale via un écran LCD et un ensemble de boutons, permettant de collecter et de visualiser les données fournies par les capteurs présents et de contrôler les actionneurs manuellement ou de définir les actions de l'actionnement automatique. Le module distant, à son tour, reçoit des informations environnementales en temps réel dans une interface graphique dans les deux sous-systèmes constitutifs et, si nécessaire, contrôle les actionneurs du module local à distance via l'application RemoteXY. À la fin, la fiabilité de la solution conçue est évaluée.

Mots-clés

Smart Serres, IoT, automatisation, commande à distance, AWS.

Índice

Índice de Figuras

Índice de Tabelas

Acrónimos

IDE – Integrated Development Evironment

USB – Universal Serial Bus

LCD – Liquid Crystal Display

IoT – Internet of Things

PVC – PolyVinyl Chloride

UV – Ultravioleta

GSM – Groupe Spécial Mobile

AWS – Amazon Web Services

RTC – Real Time Clock

ICSP – In-Circuit Serial Programming

TWI – Two-Wire Interface

ADC – Analog-to-Digital Converter

A/D – Analógico/Digital

LDR – Light Dependent Resistor

PCB – Printed Circuit Board

NTC – Negative Temperature Coefficient

HR – Humidade Relativa

DC – Direct Current

AO – Analog Output

DO – Digital Output

IR – Infrared

RPM – Rotações Por Minuto

PA – Ponto de Acesso

AP – Access Point

SSID – Service Set IDentifier

TCP – Transmission Control Protocol

1. INTRODUÇÃO

No âmbito de dar a conhecer o estado de arte e formas de implementação de estufas automatizadas foi desenvolvido um projeto para a unidade curricular Tese/Dissertação, do 2º ano do Mestrado em Engenharia Electrotécnica e Computadores (MEEC), do Departamento de Engenharia Electrotécnica (DEE), do Instituto Superior de Engenharia do Porto, com o foco em desenvolver um protótipo de uma estufa inteligente aplicando conceitos aprendidos ao longo do curso e na elaboração da Tese.

1.1. CONTEXTUALIZAÇÃO

Desde tempos antigos que existe a ideia do cultivo de plantas em estruturas climatéricas controladas e fora da época em que normalmente nascem. As estufas vêm suprimir essa necessidade, sendo estruturas que permitem o controlo das condições de temperatura interior através da retenção de calor do sol de acordo com a entrada de radiação solar, mantendo as condições necessárias para o saudável desenvolvimento do cultivo e ao mesmo tempo protegendo as plantas das condições climatéricas exteriores [1].

Desde o século passado até à atualidade, surgiram muitos avanços tecnológicos que ajudaram na construção de estufas mais eficientes e baratas, com o surgimento das estufas inteligentes a permitirem a recolha de informação ambiente e atuação automática de ações

de acordo com as necessidades das plantas cultivadas, para além da monitorização e controlo de forma remota [12] [14].

A expansão das estufas inteligentes tem sido alavancada não só pelos avanços nas tecnologias eletrónicas, mas também no aparecimento de conceitos como *Internet of Things* (IoT) e que abrangem cada vez mais o setor industrial das estufas [13] [14].

IoT permite a monitorização do microclima gerado numa estufa através de aplicações móveis e ajustar os fatores que afetam o cultivo para obter melhores taxas de rendimento, englobando tecnologias de informação de *Machine Learning* para processar e guardar informações sobre colheitas e clima, gerando recomendações com base nas informações recolhidas [14].

No entanto, mão de obra dedicada à plantação e colheita continuam a ter maior preponderância em comparação com a utilização de sistemas de controlo das condições climatéricas e de irrigação [13].

1.2. OBJETIVOS

O objetivo principal deste projeto é a construção e implementação de um sistema automático de controlo do protótipo de uma estufa, quer de forma local, quer de forma remota.

Para tal, ficaram definidos os seguintes pontos principais dentro do objetivo global:

- Leitura do estado de arte referente ao estado atual das estufas automatizadas, o seu contexto dentro da evolução das estufas ao longo da história, os vários tipos de estufa existentes e previsões de mercado sobre este tipo de indústria;

- Construção de um protótipo, onde o módulo local seja capaz de fazer a leitura de vários sensores ambientais (nível de luz, humidade, temperatura) e atuar automaticamente segundo valores definidos pelo utilizador num menu visualizado por um *Liquid Crystal Display* (LCD);

- Implementação de um sistema remoto para leitura dos dados de forma gráfica enviados pelo módulo local e também controlo automático deste último;

- Escrita do relatório.

1.3. CALENDARIZAÇÃO

A **Tabela 1** mostra a planificação final das etapas definidas para a duração da tese. Nesta houve um período de cerca de 1 mês onde não foi possível a realização de qualquer trabalho na tese devido à necessidade de estudo e realização de exames. Também deve ser referido que inicialmente o módulo remoto era para ser constituído por uma interface gráfica construída no *software* LabView. No entanto, a falta de tempo e de conhecimento suficiente referente a este *software* gráfico, por ser o primeiro contato que existiu com este programa, levou ao abandono desta solução por uma mais simples, com a utilização da aplicação de *Smartphone* chamada RemoteXY.

Tabela 1 – Planificação da tese.

ID	Etapas	Abril					Maio				Junho			Julho	Agosto					Set.			Out.
		1	2	3	4	5	6	7	8	9	10	11	12		13	14	15	16	17	18	19	20	21
1	Pesquisa do estado de arte													Pausa para exames									
2	Planeamento da estrutura e materiais do projeto																						
3	Desenvolvimento do código do módulo local																						
4	Construção da estrutura do protótipo																						
5	Desenvolvimento do código do módulo remoto																						
6	Escrita do relatório																						

1.4. ORGANIZAÇÃO DO RELATÓRIO

O Capítulo 1 faz uma pequena introdução ao tema das estufas automatizas e a sua contextualização no âmbito da disciplina de Tese do MEEC e da sua importância na sociedade no global. Este também apresenta um subcapítulo que dará a expor os objetivos que se esperará atingir neste projeto e no desenvolvimento do protótipo final, outro subcapítulo que apresenta a calendarização de todo o projeto, finalizando nesta mesma secção onde é exposta a organização geral deste relatório.

No capítulo seguinte, 2, é apresentada o estado de arte relacionada com as estufas automatizadas, explicando em primeiro lugar em que consiste uma estufa e contextualizar o desenvolvimento destas estruturas ao longo da história até à atualidade. Também serão inumeradas as diferentes caraterísticas e formas das estufas atuais e serão dados a conhecer alguns exemplos de estufas automatizadas e produtos, quer em estado conceptual ou de protótipo, quer em estado de funcionamento e produção. Por último, é exposto o valor do mercado global das estufas inteligentes e principais motivos do seu crescimento.

O 3º capítulo, tem o propósito de expor a arquitetura do protótipo, demonstrando o propósito e características de cada componente utilizado. Também é demonstrado o diagrama de funcionamento do programa desenvolvido para o protótipo e estrutura do menu implementado.

O capítulo 4 por sua vez expõe os passos necessários na implementação da solução proposta, desde a ligação dos componentes e construção do protótipo, assim como explicações das partes principais do código desenvolvido.

A seguir, no capítulo 5, são demonstrados os principais resultados obtidos durante o funcionamento do protótipo.

No último capítulo, o 6º, são reunidas as principais conclusões obtidas pelo desenvolvimento do protótipo e perspetiva de possíveis desenvolvimentos futuros.

2. REVISÃO DO ESTADO DE ARTE

Esta secção, dedicada à revisão do estado de arte atual nas estufas automatizadas, servirá não apenas como uma pequena introdução a exemplo de utilização de uma estufa automatizada e equipamentos utilizados como também no que consiste atualmente uma estufa, a sua evolução ao longo dos tempos e as diferentes formas existentes das estufas consoante o propósito de utilização.

2.1. O QUE É UMA ESTUFA?

Estufas são estruturas que permitem o controlo das condições de temperatura interior através da retenção de calor do sol de acordo com a entrada de radiação solar, mantendo as condições necessárias para o saudável desenvolvimento do cultivo e ao mesmo tempo protegendo as plantas das condições climatéricas exteriores [1]. Estufas feitas de materiais transparentes, como vidro ou plástico, permitem a passagem da radiação solar promovendo o aquecimento do ar da estufa que dar-se-á normalmente por ação de correntes de convecção suprimida (massas de ar quente sobem enquanto massas de ar frio descem), onde a energia que entra pela radiação solar e aquece o interior da estufa não é perdida com correntes ascendentes quando não existe troca de ar entre o seu interior e o exterior (**Figura 1**) [1] [2].

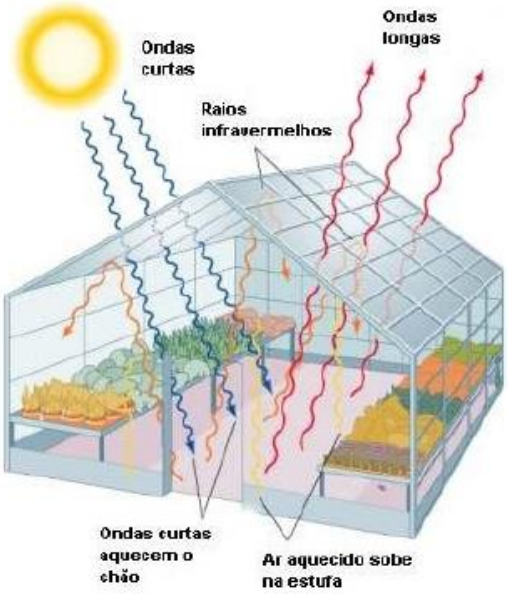


Figura 1 – Exemplo de uma estufa com aquecimento passivo por ação de correntes de convecção [2].

Desta forma, as culturas podem ser protegidas dos efeitos adversos do clima, garantindo uma produção constante ao longo do ano, tendo também em conta de que a luz, calor e humidade nesse ambiente precisa de ser controlado conforme as necessidades das plantas que serão cultivadas e o conjunto de técnicas adotado para estas [4].

2.2. HISTÓRIA DA UTILIZAÇÃO DE ESTUFAS

Desde tempos antigos que existe a ideia do cultivo de plantas em estruturas climatéricas controladas. Uma das referências mais antigas de estruturas que podem ser referidas como estufas provêm do tempo do império Romano, onde pepinos eram cultivados em carroças ficando durante o dia ao sol e depois mantidos quentes durante a noite em edifícios dedicados com coberturas de selenite, uma rocha transparente para deixar passar os raios solares [3] [6].

Mais tarde, com o grande período de exploração, iniciado no século XIII e estendendo-se pelo século XVII, plantas eram trazidas de vários locais exóticos como as Américas, Ásia, Médio Oriente e África. De forma a proteger estas plantas, surgiu em Itália a primeira estufa de nome Jardim Botânico (**Figura 2**). Aqueles que cuidavam destas estufas depressa

aprenderam a ajustar os seus métodos ao movimento solar, utilizando a exposição a Sul para garantir que a máxima radiação solar chegasse às plantas. Esta ideia de criar um espaço protetor para permitir o crescimento de plantas independentemente da altura do ano provou-se bastante popular e depressa se espalhou por toda a Europa, primeiro pela Holanda, depois por Inglaterra e França [7].

Figura 2 – Jardim botânico Orto Botanica di Padova, em Itália, é o jardim botânico universitário mais antigo do mundo, originário a 1545 [46].

Estas primeiras estufas eram restritas aos ricos que as utilizavam para cultivar as suas flores e frutas favoritas, assim como universidades. Por exemplo, a família real francesa adorava tanto laranjas que mandaram a sua plantação nos jardins de Versalhes. No Inverno, estas eram movidas dos jardins para o interior de estruturas aquecidas por fornalhas de carvão (o aquecimento passivo utilizado no Sul da Europa não era suficiente para manter uma temperatura aceitável), desenhadas para conter cerca de 1000 laranjeiras e outras plantas subtropicais, levando a ser chamado de "Laranjeiras" (**Figura 3**). No entanto, este sistema de aquecimento era rudimentar e ineficiente, levando a que algumas plantas tivessem danos provocados por frio, enquanto que outras eram literalmente cozinhadas. Deste modo, muitas das plantas estariam mortas ou bastante danificadas aquando a chegada da Primavera [6] [7].

Figura 3 – As "Laranjeiras" de Versalhes, construídas entre 1684 e 1686, com o jardim em frente [6].

No entanto, a criação da primeira estufa moderna prática é usualmente creditada ao botanista francês Charles Lucien Bonaparte no século XVII com a construção desta estrutura em Leiden, Holanda, para o cultivo de plantas tropicais medicinais [5].

No século XIX, com o abolir de taxas em janelas de vidro, os edifícios começaram a ter janelas de vidro muito maiores permitindo o aparecimento das primeiras estufas quase completamente feitas de vidro (**Figura 4**). Durante este período, as estufas deixam de ser apenas restritas a instituições universitárias e dos mais abastados para se tornar bastante popular o adicionar de uma pequena estufa em qualquer grande casa. Também foi neste período em que industrialistas criaram grandes estufas públicas famosas, conhecidas por conservatórias, com a dupla função de mostrar as várias plantas exóticas ao público e serem centros de convenção. O Palácio de Cristal em Londres foi uma destas grandes estruturas e uma das mais populares, sendo uma prova de conceito de que uma estufa podia ser durável, simples e de rápida construção. No entanto, esta estrutura já não existe atualmente [6] [7].

Figura 4 – A "Nash House" em Londres, Inglaterra, com origem em 1825, é uma das mais antigas estufas de vidro ainda existentes [6].

Com a utilização de estufas a ser bastante comum, principalmente no final do século XIX, continua a existir a experimentação em termos de design, aquecimento, materiais de construção e ventilação utilizada nestas estruturas. Por exemplo, algumas utilizariam fornalhas para aquecimento, enquanto que outras tiravam vantagem da insulação natural proveniente da terra, mas tendo sempre como ponto comum a continuada utilização da exposição solar a sul para ajudar no seu aquecimento de forma passiva através da radiação solar [7].

Desde o século passado até à atualidade, surgiram muitos avanços tecnológicos que ajudaram na construção de estufas mais eficientes e baratas, tornando-as mais acessíveis ao jardineiro e agricultor comum. Um destes avanços, foi o aparecimento e grande disponibilidade durante os anos de 1960 de grandes folhas de polietileno e que acabaram por revolucionar a construção de estufas. Conjugado com o aparecimento de aros de plástico no mesmo período, que também utilizariam extrusões de alumínio ou tubos de policloreto de vinil (*PolyVinyl Chloride* – PVC), ajudaram a manter os custos ainda mais baixos, e com inibidores ultravioleta (UV) desenvolvidos na década de 1970 a aumentarem a durabilidade das folhas de polietileno de 1 ou 2 anos para 4 ou mais, está na origem a que mais de 90% destas estruturas sejam construídas atualmente utilizando estas folhas [7] [8].

Atualmente, estufas encontram-se cada vez mais automatizadas, com sistemas de controlo de temperatura e humidade, sistemas de rega e de abertura ou fecho de condutas automáticas dependente da temperatura, entre vários outros sistemas que permitem poupar tempo e aumentar a eficiência de cultivo. O aparecimento de novos tipos de materiais como fibra de vidro, acrílico e painéis de policarbonato ajudam a aumentar esta eficiência como também a tornar os custos para estas estruturas mais baixos [7].

2.3. TIPO DE ESTUFAS

Consoante a finalidade de uma estufa, como por exemplo o tipo de cultivo desejado, e recursos utilizados, estas terão diferentes classificações. Estas classificações podem depender de fatores como:

- modo de operação;
- tamanho e forma;
- o tipo de construção;
- tecnologia utilizada no crescimento de plantas;
- características técnicas [9].

2.3.1. MODO DE OPERAÇÃO

O regime de utilização de uma estufa permite dividir em dois modos de operação, sendo estes:

- **Sazonal:** onde a operação começa a partir de março até final da estação de Outono. Uma grande vantagem deste modo de operação é a sua facilidade de manutenção e custo reduzido. No entanto, dependendo da sua localização geográfica, o solo nestas estufas podem acabar por congelar no Inverno, tornando em poucos anos o solo de cultivo utilizado menos fértil [9];
- **Integral:** este tipo de regime, onde o cultivo e recolha das plantas pode ser feita em qualquer época do ano, implica a disponibilidade de instalações industriais adicionais para o cultivo. Neste tipo de estufa o custo de construção é muito mais dispendioso,

tendo, no entanto, a vantagem de ser obtido maior lucro e retorno num curto espaço de tempo, fruto do seu funcionamento ao longo de todo o ano [9].

2.3.2. TAMANHO E FORMA

Em termos de tamanho, as estufas podem ser divididas em pequenas, médias e grandes. O tamanho escolhido acaba por ditar a escala de produção e o tipo de tecnologia utilizada, pois este afetará como será feita a manutenção da temperatura, a utilização do equipamento certo para fornecer e regular a luz e aquecimento, como também a otimização dos custos de produção [9].

Para a classificação onde a especificidade da forma utilizada é o fator considerado, as mais populares são:

- **Arqueado:** estufas geralmente com cobertura de folhas de polietileno e um esqueleto composto por arcos de metal ou PVC (**Figura 5**). Apresenta grande resistência a ventos fortes laterais e suportam melhor o peso da precipitação devido à sua forma. [9] [10];

Figura 5 – Estufa do tipo arqueado [9].

- **Lancet:** Eles diferem do telhado pontudo arqueado, que não permite a acumulação de precipitação e não interfere com a penetração da luz solar [9];

- **De encosto:** tal como o nome indica, este tipo de estufa é colocado contra um edifício, utilizando a estrutura existente para um dos seus lados (**Figura 6**). A estufa deve estar exposta às radiações solares ao estar virada na direção Sul, podendo ser tão longa como o edifício a que se encontre encostado. As vantagens neste tipo de estrutura são a sua proximidade a sistemas de água e de eletricidade, tira o melhor partido da luz solar e é uma estrutura de baixo custo por requerer menos materiais na sua construção, como os suportes do teto. No entanto, a altura da parede de suporte limita o tamanho do design, com outras desvantagens como a limitação na receção de luz, ventilação e controlo de temperatura. Este controlo de temperatura é difícil devido à parede com a qual a estufa é adjacente que pode acumular calor, enquanto que o revestimento translúcido a perderá rapidamente [10];

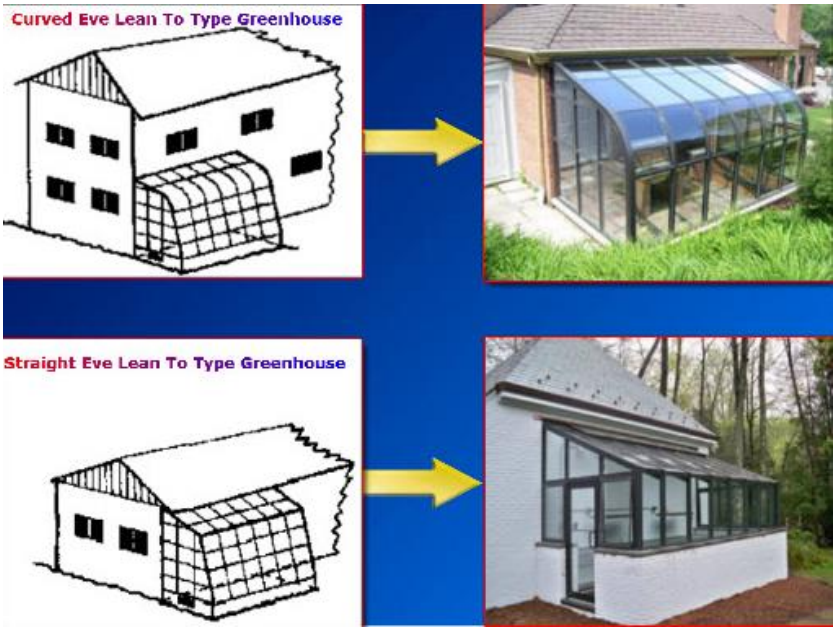


Figura 6 – Estufa do tipo de encosto [10].

- **Empena:** inferior à estrutura arqueada em termos de escala, o que faz com que seja menos popular entre as estufas industriais. Este tipo de estufa deve suportar cargas pesadas (como o vidro utilizado para as paredes e teto), requerendo uma estrutura mais sólida (**Figura 7**) [9]. A forma da estrutura permite uma melhor circulação de ar do que as de encosto, o que ajuda a manter a temperatura uniforme durante o Inverno, onde pode requerer um sistema de aquecimento auxiliar se a estufa não se encontrar próxima a um edifício aquecido. Pelo seu tamanho e área exposta de vidro, também necessitará de um maior custo para o aquecimento [10];

Figura 7 – Estufa do tipo empena [9].

- **Multinave:** composto por vários telhados em forma de "A" interconectados entre eles ao longo do seu comprimento por calhas que transportam a água da chuva ou neve para fora (**Figura 8**). Devido a não existirem paredes laterais dentro da estufa entre as estruturas dos vários telhados, o interior é composto por apenas um grande espaço de cultivo, o que reduz o custo de trabalho e automação, melhora a gestão e reduz o custo no consumo de combustível para o sistema de aquecimento ao reduzir a área exposta da estrutura pelo qual o calor escaparia. Para localizações onde a neve é um fator, este deve ser tido em conta na construção da estrutura visto que a neve que se acumular no topo da estufa não poderá deslizar do topo para o chão, apenas derreter com as calhas a levar a água para fora. Mesmo com esta última desvantagem, este tipo de estufa é utilizado de forma eficaz em países da Europa do Norte (como a Holanda), Canadá e até a Índia [10].

Figura 8 – Estufa do tipo multinave [10].

Para todas estas estufas também é importante a sua orientação em relação ao Sol consoante a localização geográfica. Para estufas localizadas a norte de 60º de latitude, uma orientação longitudinal será a mais eficaz, enquanto que a sul desta latitude uma orientação meridional será a que apresenta melhores resultados [9].

2.3.3. TIPO DE CONSTRUÇÃO

A classificação por tipo de construção pode ser subdividida pelo material utilizado na armação de suporte e pelo seu revestimento. O material da armação influência predominantemente o quão grande a estufa poderá ser, ou seja, quanto maior a estrutura mais forte deverá ser o material utilizado. O revestimento é importante ao nível da influência que tem na criação do efeito de estufa dentro da estrutura, mas também na escolha do material da armação e método de fixação [10].

Baseado no material da armação de suporte, poderão existir os 3 seguintes materiais:

- **Suporte de madeira:** normalmente utilizado em estufas cujo interior seja inferior a 6 metros de comprimento [10]. Este material requer apenas habilidades básicas e um mínimo de ferramentas na sua construção, sendo também o que tem o custo mais reduzido. Tem como desvantagens a propensão para apodrecer (tendo como tempo de vida média de 4 anos, especialmente quando em contacto com o solo) e não ser possível a construção de estruturas arqueadas (**Figura 9**) [11];

Figura 9 – Armação de suporte de madeira [10].

- **Suporte em tubo de plástico/metal:** utilizado para a construção de estufas com espaço interior médio de 12 metros, com colunas, postes e armação cruzada em tubos [10]. Em geral, as estufas de plástico utilizam tubos de PVC que podem ser facilmente dobrados e colocados juntos (**Figura 10**), sendo razoavelmente fácil substituir uma peça deformada por uma nova sem desmontar toda a armação. Este é um material de baixo custo e com baixa condutividade térmica, mas rapidamente acumula rachaduras a baixas temperaturas e também não pode ser utilizado com revestimento de policarbonato, por este último ser mais resistente do que a armação de plástico. Em conjunto com a probabilidade de deformação aquando rajadas de vento fortes, torna a utilização de tubos em PVC pouco confiável [11]. Suportes de metal, como alumínio, são mais resistentes e têm maior duração de vida permitindo a utilização de um revestimento de policarbonato. No entanto, existe uma maior complexidade de instalação quando feita à mão, maior custo do material e particular atenção deve ser dada na construção das fundações devido ao peso da armação metálica. Também é importante ter a habilidade de dobrar o material através de equipamentos especiais, caso se pretenda a criação de uma estufa arqueada, e é possível reduzir a alta condutividade térmica do metal pintando-a com uma cor clara [11];

Figura 10 – Armação de suporte de tubos de plástico ou alumínio [10].

- **Suporte de armação de telhado metálico:** utilizado para a construção de estufas com espaço interior maior de 15 metros (**Figura 11**). Muito utilizado em estufas de tipo empena e multinave devido a serem ideais para prefabricação [10].

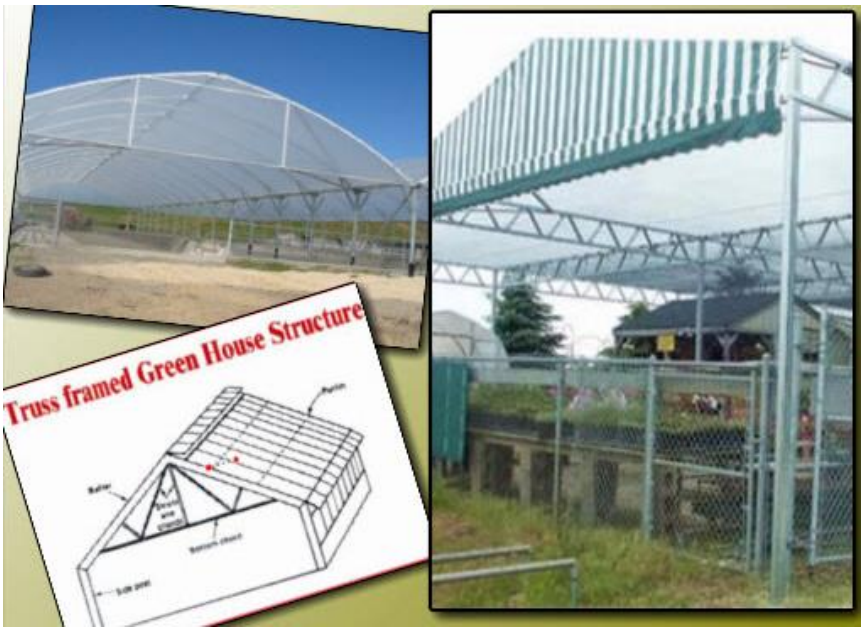


Figura 11 – Quadro de suporte de armação de telhado metálico [10].

Como referido anteriormente, o revestimento é a parte mais importante de qualquer projeto de construção de uma estufa. Assim, baseado no tipo de revestimento existem 3 diferentes materiais possíveis, sendo estes:

- **Revestimento de vidro:** estufas que utilizam este material são aquelas que permitem a entrada interior de luz mais intensa pela sua boa capacidade de transmissão do exterior para o interior. Apresenta bom isolamento térmico e custo relativamente baixo, mas o seu peso requer armações de suporte resistentes e devido às propriedades do vidro, o ar interior aquece rapidamente provocando um sobreaquecimento que pode resultar na perda parcial ou total de uma plantação. Neste último caso, pode se tornar numa vantagem, mas apenas durante a época de Inverno devido às condições climatéricas associadas. Os tipos de estufa mais comuns que utilizam revestimento de vidro são as do tipo de encosto, empena e multinave (**Figura 12**), sendo que estas têm uma taxa de infiltração de ar alta resultando em humidade interior baixa e melhor prevenção de doenças na plantação [9] [10];

Figura 12 - Estufa do tipo multinave com revestimento de vidro [9].

- **Revestimento de filme de plástico:** neste são incluídos os materiais feitos de polietileno (**Figura 13**), poliéster e policloreto de vinil (PVC). Estes plásticos são bastante populares devido ao seu baixo custo e ao menor preço de aquecimento quando comparado com as estufas de revestimento de vidro. No entanto, o tempo de vida deste tipo de revestimento é baixo, com uma duração máxima de 4 anos para o filme de melhor qualidade existente. Estufas do tipo arqueado são os que melhor se adequam a este revestimento [10].

Figura 13 – Exemplo de uma estufa com revestimento de polietileno [10].

- **Revestimento por painéis rígidos:** colocados em estufas arqueadas e multinave, os painéis podem ser de PVC, plástico reforçado de fibra de vidro ou de policarbonato celular (**Figura 14**). Este último é o material mais popular para estufas industriais devido à sua capacidade excelente de isolamento térmico, baixo peso que permite

um trabalho de montagem rápido e dispersão de raios UV nocivos às plantas. Todos estes tipos de painéis são também resistentes a serem partidos e intensidade de luz uniforme quando comparado com os revestimentos referidos nos parágrafos anteriores, sendo que os painéis de melhor qualidade podem durar até 20 anos. Algumas das desvantagens destes painéis são a sua tendência de acumular pó e aparecimento de algas que resulta na redução de transmissão de luz [9] [10].

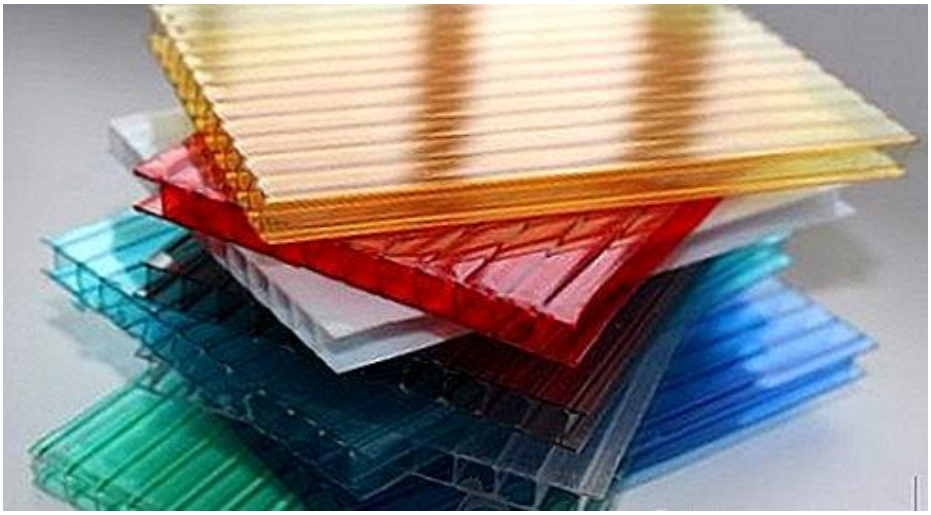


Figura 14 – Painel de policarbonato celular [9].

2.3.4. TECNOLOGIA DE CRESCIMENTO UTILIZADA

Diferentes culturas têm as suas próprias características e necessidades, cada uma podendo ter o seu crescimento afetado de forma positiva e mais eficiente apostando na tecnologia adequada de cultivo para cada planta. Assim, podem ser descritos 3 tipos de tecnologia de cultivo:

- **Solo (terra):** utiliza uma mistura de solo adequado às características e necessidades de cada planta, sem qualquer utilização de bases [9];

- **Prateleiras:** as culturas são cultivadas em prateleiras, equipadas com placas (**Figura 15**). Quando as plantas estão no chão, outra opção seria a utilização de um bezstallable [9];

Figura 15 – Cultivo em prateleiras [9].

- **Hidropónico/aéreo:** numa estufa hidropónica as plantas são cultivadas em soluções aquosas com compostos de nutrientes, sendo impregnados com um substrato de suporte especial que serve de chão (**Figura 16**). Por sua vez, o cultivo por método de cultivo aéreo não utiliza quaisquer substratos, apenas utilizam suportes com braçadeiras para a fixação das plantas. Exemplos de plantas utilizadas em cultivo hidropónico são morangos ou tomates [9].

Figura 16 – Cultivo hidropónico [9].

2.3.5. CARACTERÍSTICAS TÉCNICAS

Para além dos métodos e caraterísticas referidas anteriormente, estufas poderão necessitar de equipamentos adicionais para criar todas as condições necessárias que assegurem o crescimento saudável e eficiente das plantas cultivadas [9].

Uma das condições mais importantes a manter numa estufa é o controlo de temperatura. Para tal, um dos equipamentos necessários a instalar poderá ser caldeiras de alta eficiência, em que o aquecimento se poderá realizar por ar ou a gás. Em contraponto, deve existir o cuidado de a temperatura ambiente não ultrapassar certos limites que tornam as plantas letárgicas e possivelmente levando à sua morte. Por isto mesmo, deve existir um sistema de ventilação localizado na estrutura superior da estufa, automatizado ou manual, que permita ao ar frio entrar e ser ligeiramente aquecido até ao momento em que este chega às plantas (não criando mudanças bruscas de temperatura) [9].

Outra das condicionantes para a boa colheita numa plantação é a administração da quantidade de água correta através de um sistema de rega adequado, quer seja por irrigação automática (**Figura 17**), intrasilar ou por gotejamento, cada uma dependente do tipo de cultivo desejado. Também pode ser incluído um sistema de bombas para permitir a filtragem e economizar de água e fertilizantes, para além de um sistema de drenagem [9].

Figura 17 – Sistema de irrigação automática [9].

Por vezes pode ser instalado também um sistema de iluminação, para certas culturas que exigem o fornecimento de iluminação de alta qualidade de pelo menos 9 a 10 horas por dia, acabando por ser mais relevante a sua utilização durante o Inverno ou em tempo nublado. O tipo de lâmpadas recomendadas na instalação deste sistema são as florescentes, de halogeneto de metal ou as de sódio [9].

2.4. ESTUFAS AUTOMATIZADAS

Estufas automatizadas, também chamadas de estufas inteligentes, permitirem a recolha de informação ambiente, como temperatura, humidade, luminosidade, nível de pH do solo, entre outras, através de vários sensores e a execução automática de ações de acordo com as necessidades das plantas cultivadas por atuadores que garantam o crescimento ideal do cultivo, podendo também ser feita a monotorização e controlo de forma remota [12].

A progressiva automatização de estufas tem sido a natural evolução devido aos avanços nas tecnologias eletrónicas e ao aparecimento de conceitos como *Internet of Things* (IoT) e que abrangem cada vez mais o setor industrial das estufas [13] [14].

Uma plataforma de IoT é caraterizada tipicamente por vários dispositivos (*Things* – Ex: sensores, atuadores) remotos que podem utilizar algum formato de Gateway intermédia para comunicar entre uma rede local até um servidor ou plataforma de *Cloud*, a qual permite o armazenamento, processamento e análise da informação recolhida (**Figura 18**) [15].

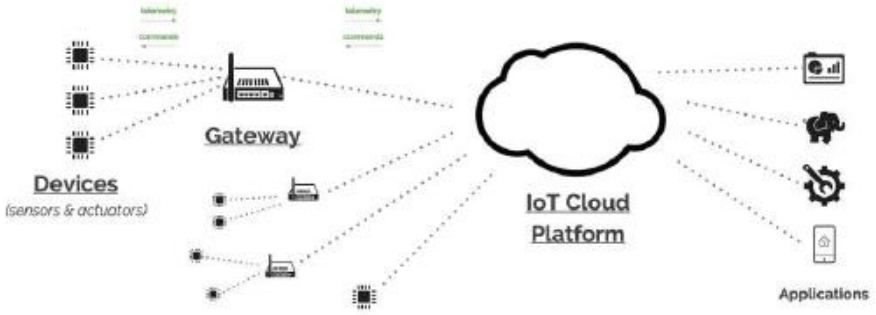


Figura 18 – Estrutura típica de uma plataforma IoT[15].

Conceitos como IoT ajudam na eficiência e o acompanhamento das operações de controlo de tarefas remoto através de decisões que têm por base informação recolhida em tempo real. Desta forma, a produtividade numa estufa pode ser aumentada com o ajuste eficiente dos processos utilizados de cultivo [14]. No entanto, também é importante escolher o tipo de tecnologia e sistema apropriado para cada tamanho e tipo de estufa, desde pequenos jardins a grandes estufas de produção [13].

Atualmente, quase todas as grandes estufas estão automatizadas para a maioria dos seus processos de produção de rotina (como o controlo climático e fertirrigação), o que permite

dispensar muitos dos seus operadores destas mesmas operações, o que lhes permite concentrar nos aspetos fundamentais da exploração de uma estufa [13].

O nível de automatização em estufas inteligentes também é variável podendo ser semiautomáticas ou totalmente automatizadas, esta última a melhor opção para produtores que prefiram a liberdade total de operação, sem se preocupar com suas colheitas. Por exemplo, empresas como a *Emerald Kingdom Greenhouse*, oferecem estufas totalmente automatizadas que permitem a monotorização do microclima gerado através de uma aplicação móvel e ajustar os principais fatores que afetam o cultivo para obter melhores taxas de rendimento através dessa aplicação. Outras aplicações que sirvam as mesmas funções podem utilizar tecnologias de informação de *Machine Learning* para processar e guardar informações sobre colheitas e clima ideal, podendo fazer recomendações com base nas informações recolhidas (**Figura 19**) [14].

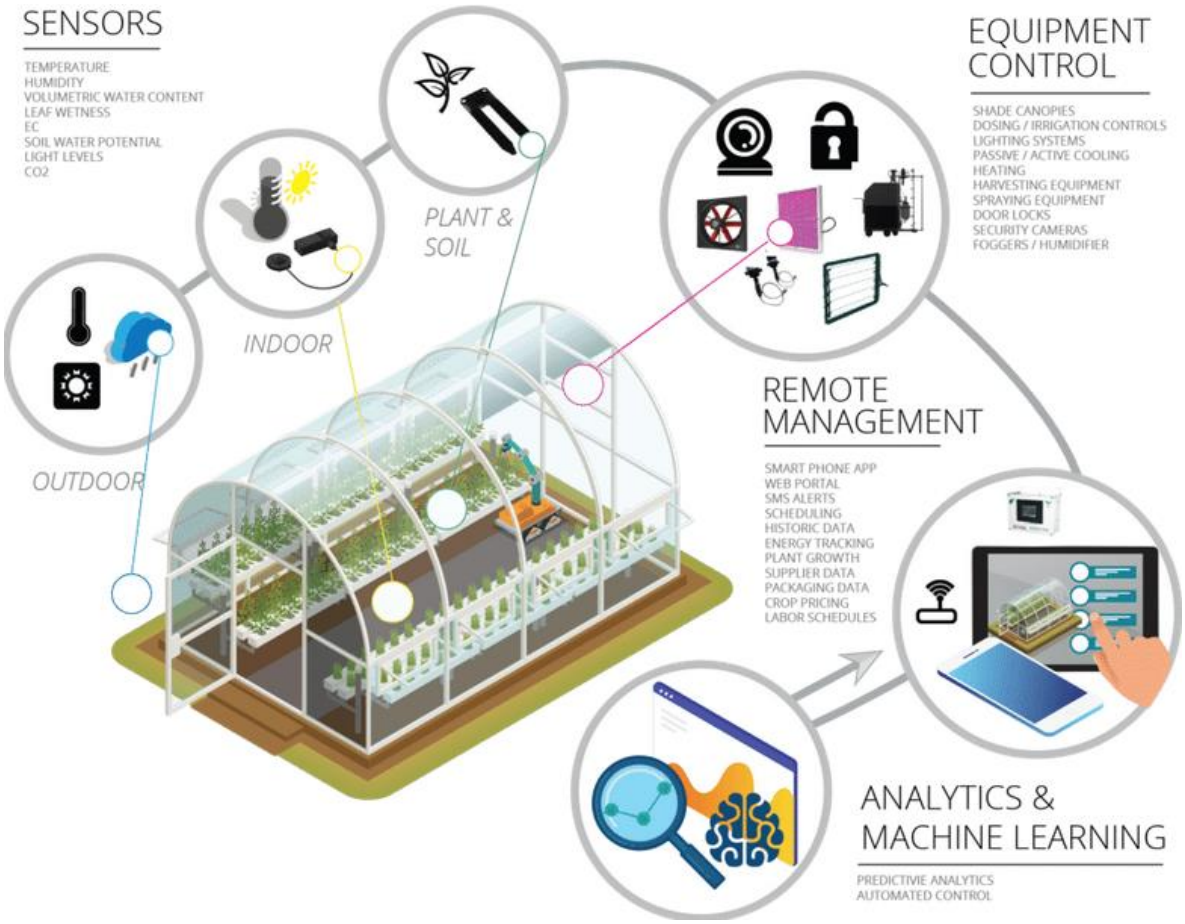


Figura 19 – Esquema de uma possível realização de uma estufa inteligente [17].

2.4.1. CONTROLADORES

Como referido anteriormente, a maioria dos processos de produção de rotina como controlo climático e fertirrigação são automatizadas. Parte importante para esta automatização são os controladores que fazem a ligação entre o processamento de dados enviados pelos sensores e os atuadores existentes. Para a manutenção do clima em estufas de produções ornamentais, a utilização de controladores climáticos é amplamente usada, integrando a informação de uma grande variedade de sensores, como temperatura interior, humidade relativa do ar, velocidade do vento e da chuva no exterior, para depois atuar consoante as condições definidas. Por exemplo, pode ser fechado janelas do telhado em resposta à deteção de ocorrência de chuva, ou em áreas onde o vento atinge velocidades perigosas para estruturas da estufa e detetada por um sensor de vento. Quanto ao controlo da humidade relativa do ar, em geral, as plantas desenvolvem-se corretamente com níveis de humidade entre 60% e 80%, sendo importante para a saúde do cultivo e controlo de pestes. Em estufas não aquecidas, uma das estratégias é a abertura pequena da janela nas primeiras horas da manhã [13].

Para lidar com o controlo dos parâmetros anteriores, estão disponíveis controladores climáticos comerciais de variável complexidade, desde o mais complexo até ao mais simples (**Figura 20**): este último apenas inclui sensores, controladores e contadores que operam janelas e outros equipamentos segundo a ordem do controlador. Um exemplo destes controladores climáticos comerciais é o da empresa ULMA Agrícola que pode integrar informações de uma variedade de sensores desde a temperatura até um piranómetro (medição de radiação) e com diferentes opções para controlar janelas e aquecimento. Como complemento opcional, disponibiliza um módulo de comunicação *Groupe Spécial Mobile* Ceres (GSM Ceres) que permite a monitorização remota, permitindo a consulta de informação e receber alertas por mensagens SMS [13].

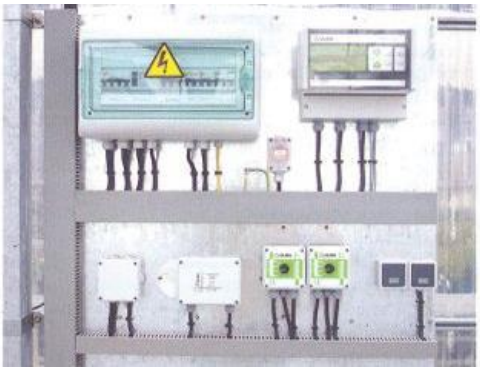


Figura 20 – Um controlador climático simples da ULMA Agrícola [13].

Para o controlo da fertirrigação (irrigação e fertilização), a automatização é normalmente feita de forma independente do controlo climático. Nesta, a rega pode ser controlada em função de vários fatores, onde o método mais simples de controlo é a utilização de tempos de rega fixos para cada cultivo e altura do ano com base na experiência do produtor. Outro método possível é o ajuste da água fornecida relacionando o número de irrigações com a radiação solar acumulada, nível de água dos tabuleiros de cultivo e leituras de drenagem. Para os nutrientes fornecidos através de irrigação, os valores de pH e condutividade da água são recolhidos por sensores apropriados colocados à saída do tanque de mistura, e de acordo com estes valores, a quantidade de nutrientes adicionados aumenta ou diminui [13].

Exemplo de controladores para fertirrigação são os controladores Agrónic, onde os mais comuns para utilização em estufas são Agronic 5000 e 7000 (**Figura 21**), que fertilizam em função da condutividade elétrica (CE) e do pH desejado na água que chegue às plantas [13].

Figura 21 – Controlador de fertirrigação Agrónic 7000 [13].

2.4.2. MAQUINARIA EM ESTUFAS

Não é só o controlo das condições climatéricas e de irrigação que englobam o conceito de uma estufa inteligente, existindo também alguma mecanização e possível automatização na preparação do solo e colheita. No entanto, mão de obra dedicada à plantação e colheita continua a ter a maior preponderância, visto a utilização de veículos de transporte requer planeamento e dimensões adequadas para os percursos nas estufas [13].

Atualmente, a colheita utiliza algum suporte auxiliar de maquinaria, onde a colheita é feita de forma manual com o apoio de plataformas móveis elevatórias, que permitem ao operário ficar à altura certa para executar a sua operação. Estas plataformas podem mover sobre rodas, ou sobre tubos de aquecimento (**Figura 22**), como as existentes no design holandês (multinave) [13].

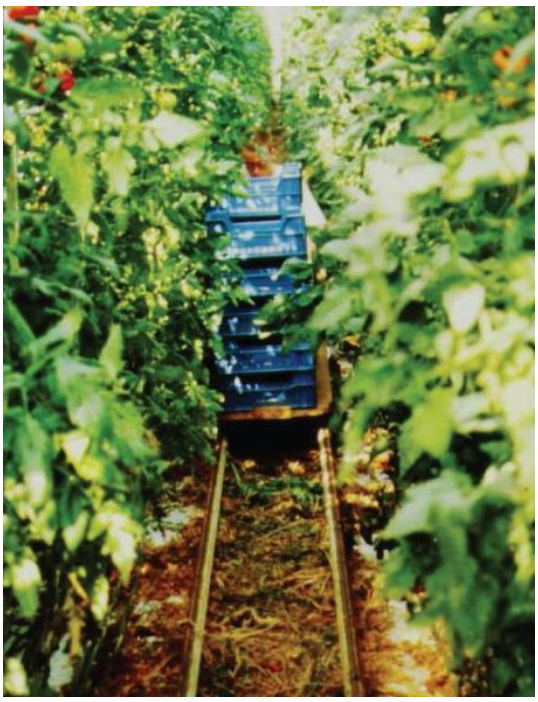


Figura 22 – Plataforma de recolha sobre tubos de aquecimento numa estufa holandesa [13].

2.4.3. EXEMPLOS DE APLICAÇÃO

Existem atualmente vários exemplos de estufas inteligentes com diferentes níveis de automatização disponíveis comercialmente ou como provas de conceito demonstradas em eventos e nesta secção serão enumerados alguns desses produtos para dar a conhecer a extensão do conceito.

Um desses exemplos são as estufas *indoor* na produção de alface pela companhia japonesa, de nome SPREAD, constituído por sistemas hidropónicos ou aeropónicos, que utilizam água corrente com nutrientes dissolvidos e luzes led para o fornecimento de luz (**Figura 23**). O mais especial nestas estufas desenvolvidas pela SPREAD encontra-se na sua completa automatização de todo o processo envolvido através de utilização de robôs. Estes terão o trabalho de plantar as sementes (e quando estas germinarem, transplantá-las para prateleiras maiores), cultivar o novo vegetal e colheita. Os sistemas envolvidos nesta operação são

capazes de monitorizar o crescimento, verificar humidade e temperatura, aumentar ou diminuir as luzes dos leds e ajustar o nível de nutrientes [18].

Figura 23 – Estufa *indoor* da empresa SPREAD [18].

Outra utilização de robôs em estufas, este num espaço mais tradicional, são aqueles que são utilizados para fazer o espaçamento de plantas nas estufas de *Metrolina Greenhouses*, em Huntersville, nos Estados Unidos, que ajudaram a reduzir os custos em trabalhadores e em compensações relacionadas a problemas nas costas causado pelo movimento das plantas durante 8 horas por dia. Os robôs *Harvest Automation* HV-100 (**Figura 24**) ajudaram a evitar que os trabalhadores realizem este tipo de trabalho repetitivo e monótono, aumentando a sua moral e a manter um ciclo de produção mais estável e previsível, porque desde que exista trabalho por fazer os robôs podem trabalhar dia e noite [19].

Figura 24 – Robôs HV-100 a realizar trabalho de espaçamento [19].

Outro exemplo de estufa inteligente mais convencional foi demonstrada no evento Re:Invent 2015, em que esta foi construída do zero no *show floor* de prepósito para a apresentação (**Figura 25**). Esta estufa utiliza o conceito IoT, com vários sensores ligados e atuadores ligados a um microcontrolador Intel Edison que envia dados e recebe comandos de um centro de controlo hospedado na plataforma *cloud* da *Amazon Web Services* (AWS). Os utilizadores podem interagir através de um *dashboard* ou *tablet* para visualizar em tempo real a temperatura, humidade no ar e no solo e luminosidade ou controlar os atuadores diretamente. Também é possível utilizar comandos de voz [12].

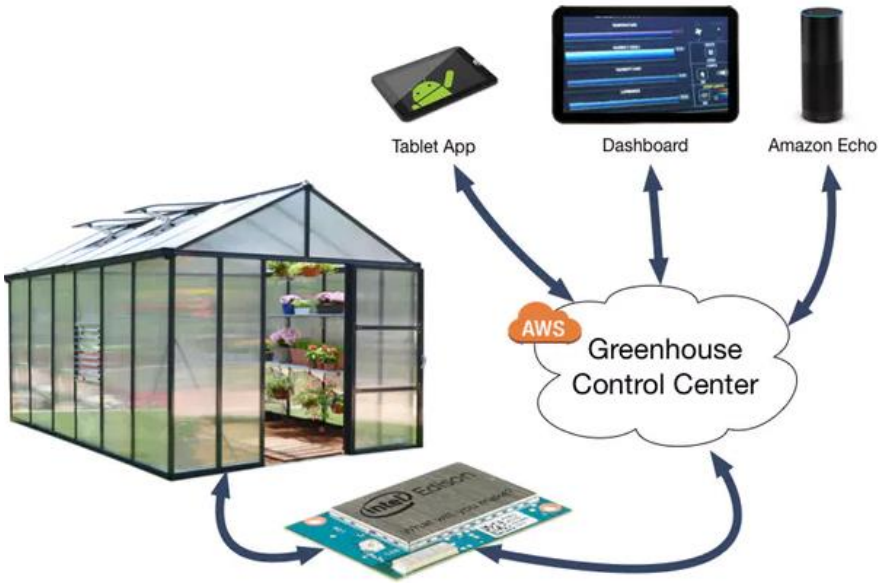


Figura 25 – Estufa inteligente demonstrada no evento Re:Invent 2015 [12].

2.4.4. MERCADO DAS ESTUFAS INTELIGENTES

De acordo com o estudo efetuado pela firma *Verified Market Research*, empresa dedicada à análise de mercados, o mercado das estufas inteligentes encontrava-se avaliado em 0,98 biliões de dólares americanos no ano de 2018, prevendo que este continue a crescer até 2,46 biliões de dólares no ano de 2026 (**Figura 26**). Este crescimento advém substancialmente da alta demanda por alimentos devido à crescente população, a tendência do aumento de agricultura *indoor*, iniciativas governamentais e incentivos relacionados com a utilização de tecnologia de iluminação inteligente em estufas, com todos estes fatores a contribuem para o desenvolvimento e crescimento do mercado das estufas inteligentes mundialmente [16].

Figura 26 – Previsões do mercado de estufas inteligentes [16].

No entanto, a exigência de um investimento inicial elevado e custo de integração da tecnologia são dois dos fatores principais que podem restringir o crescimento do mercado global das estufas inteligentes [16].

3. ARQUITETURA

De modo a simplificar a compreensão da arquitetura do protótipo e também da sua demonstração, esta será dividida em dois módulos chamados de módulo de controlo local e módulo remoto. O módulo de controlo local consiste na definição de valores de controlo para funcionamento automático através de um menu local, podendo também controlar os atuadores de forma manual e definição das horas e data do *Real Time Clock* (RTC). O módulo remoto, subdividido em duas formas de ligação e de aplicações utilizadas, permite utilizar, numa destas aplicações, as mesmas funções que o módulo de controlo local (com a exceção na definição de valores do RTC e máximos e mínimos de controlo), mas de forma remota, permitindo também a visualização gráfica em ambas dos valores recolhidos pelos sensores.

3.1. MÓDULO DE CONTROLO LOCAL

Como referido anteriormente, este módulo tem como função a definição de valores de controlo através de um menu local permitindo o funcionamento automático dos atuadores (existindo também um controlo manual destes), assim como a definição da data e horas do RTC.

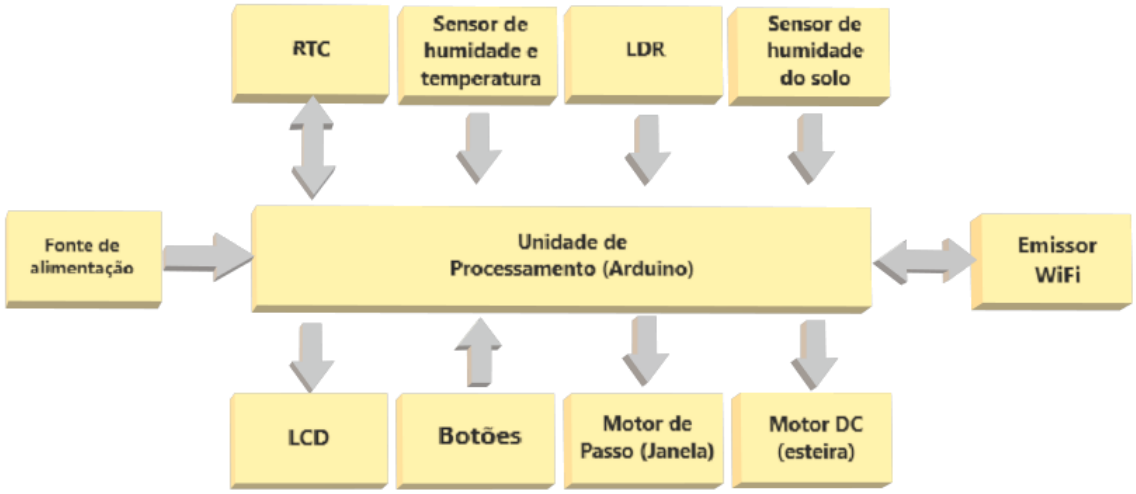


Figura 27 – Arquitetura do módulo de controlo local.

Na figura anterior (**Figura 27**) é possível verificar que a unidade de processamento escolhida foi o Arduíno Uno R3, devido principalmente à necessidade de utilização de um LCD para a visualização e construção de um menu que permita o controlo de todas as funções projetadas. Como este Arduíno contém uma biblioteca compatível com o LCD utilizado, torna-se desnecessário a projeção de uma biblioteca de caracteres de raiz ou a pesquisa de outra que poderia ter problemas de compatibilidade, permitindo assim a poupança de tempo no desenvolvimento de um menu extenso. De referir que o sentido das sentas na figura indica o sentido de comunicação feita.

O Arduíno Uno (**Figura 28**) é uma placa constituída por um microcontrolador ATmega328p que serve como o seu centro de processamento. Este é um microcontrolador desenvolvido pela Atmel e pertencente à família de microcontroladores AVR, definidos pelo facto de utilizarem uma arquitetura computacional modificada de Harvard (permitindo o acesso a conteúdos da memória de instruções como se estes se tratassem de dados) do tipo RISC de 8 bits. Sendo predominantemente de baixa potência e custo torna-o ideal para uma estrutura que necessitará de ter uma fonte de alimentação autónoma. Outras das características mais importantes da placa são: uma ligação *Universal Serial Bus* (USB) que pode ser utilizada para programação do microcontrolador e/ou fornecimento de energia, uma entrada de alimentação que permite o fornecimento de energia através de um adaptador AC/DC, pinos *In-Circuit Serial Programming* (ICSP) que permite a programação do microcontrolador diretamente através de um módulo exterior e um botão de *reset* para realizar uma nova inicialização do programa em memória.

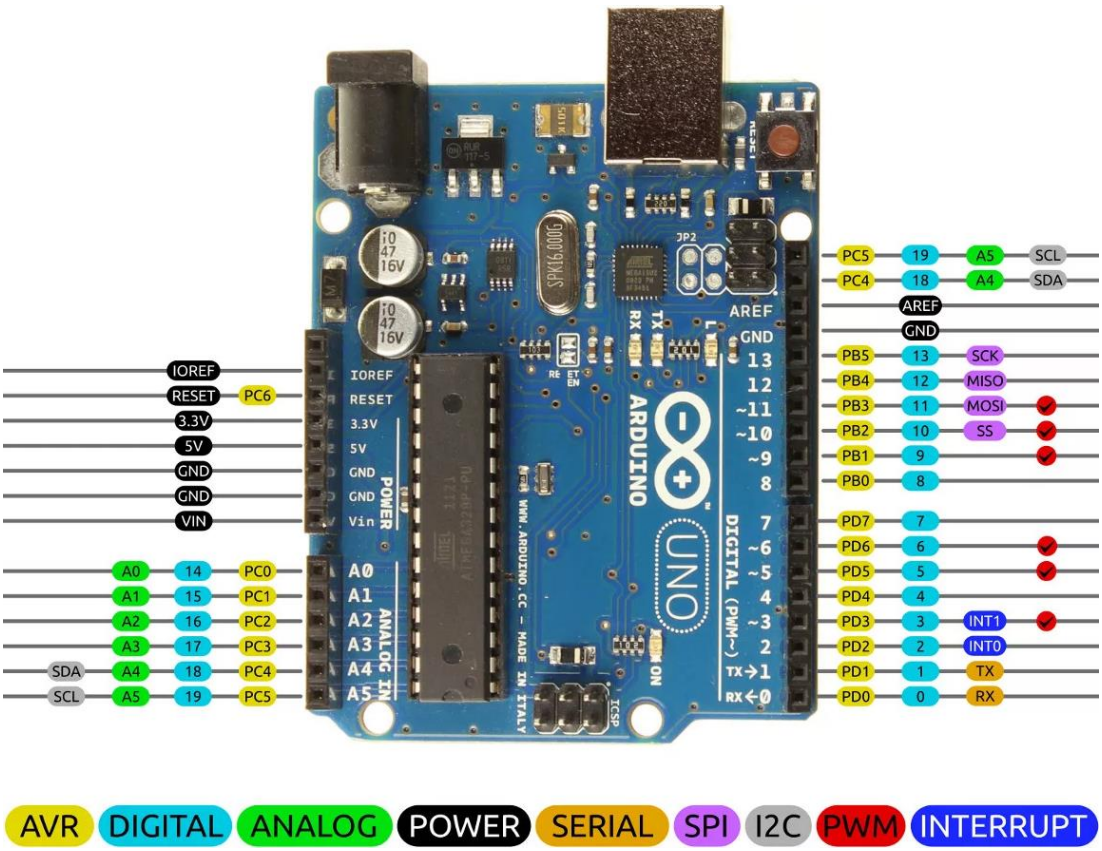


Figura 28 – Esquema de entradas e saídas do Arduíno Uno [20].

Tendo em conta as necessidades de comunicação e de manipulação dos vários componentes projetados para utilização nesta estrutura (sensores, atuadores, RTC, LCD, etc.) é necessária uma unidade de processamento que satisfaça todos esses pré-requisitos. Deste modo, o Arduíno Uno tornou-se na unidade escolhida por satisfazer todas as funcionalidades necessárias, entre elas:

- Utilização da comunicação I^2C para fazer a ligação entre o RTC e a placa Arduíno, conseguido através do microcontrolador ATmega328p (a Atmel chama esta comunicação I^2C de *Two-Wire Interface* – TWI), o que permite utilizar apenas duas linhas *bus* bidirecionais, um para o *clock* (SCL) e outro para os dados (SDA), sendo que o único *hardware* externo necessário para a implementação deste *bus* é uma resistência de *pull-up* para cada uma destas linhas (**Figura 29**). No protótipo projetado estas resistências serão desnecessárias devido ao tipo de RTC escolhido, o qual incluiu as resistências integradas na própria placa. Na comunicação I^2C os dispositivos podem ser mestres ou escravos, com o mestre a iniciar a transferência de dados e os escravos a reagir ao pedido. Os escravos têm um endereço de 7 bits

mais 1 bit que define se a comunicação é de escrita ou leitura (o endereço do RTC DS1307 para escrita é 0xD0 e de leitura 0xD1). Todos os dispositivos compatíveis com o protocolo I^2C utilizam o circuito de interface como coletor aberto, ou seja, as linhas de dados são mantidas passivamente em HIGH pelas resistências de *pull-up* (implementando uma função do tipo AND), com qualquer dispositivo no *bus* a poder colocar a linha de dados a LOW. Para a frequência de *clock* série, o máximo é determinado pelo dispositivo mais lento no *bus*, o que nesta implementação é o RTC com 100 kHz. Este é um protocolo que permite interligar até 128 dispositivos com apenas duas linhas [21].

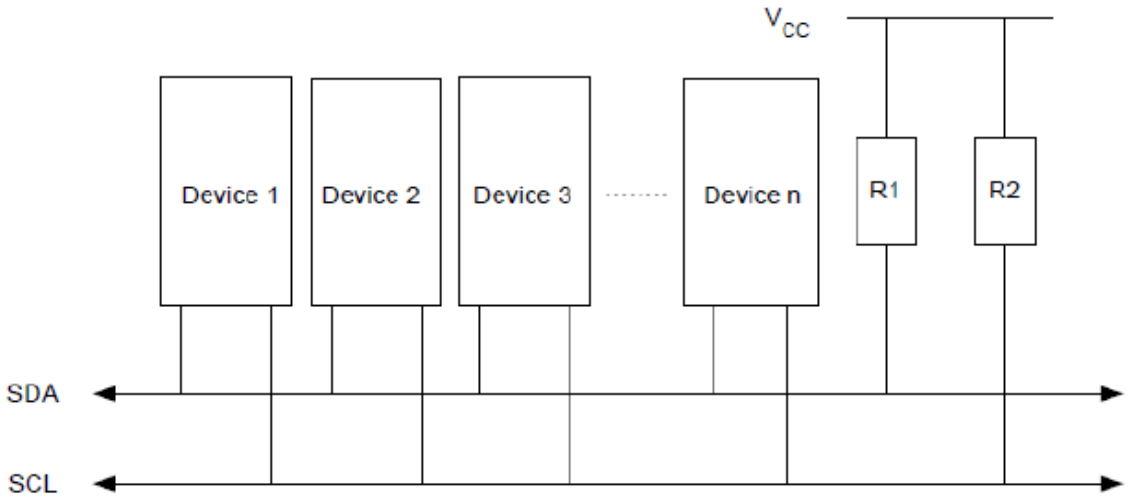


Figura 29 – Ligações de uma comunicação I^2C [21].

- Quantidade adequada de portos de entrada e saída necessárias para o grande número de componentes utilizados, quer para a aquisição de dados, quer para o seu controlo. Por exemplo, dois dos componentes com mais ligações necessárias serão o LCD e o motor de passo, com 6 e 4 ligações respetivamente;

- Portos analógicos para a conversão Analógica/Digital (A/D), também chamado *Analog-to-Digital Converter* (ADC), cada um com uma resolução de 10 bits. De forma predefinida, a leitura é feita de 0 a 5V (com 1024 valores), mas é possível modificar o alcance superior através da utilização do pino AREF e aplicando um valor de tensão diferente. No caso da solução projetada, será utilizada apenas a forma predefinida para a leitura dos sensores de luz, humidade do solo e botão premido;

- Pinos digitais 0 (Rx) e 1 (Tx) para comunicação série com a placa emissora *WiFi*, permitindo dotar o módulo local de um transmissor/recetor sem fios para fazer a ligação com o módulo remoto. O Arduíno inclui também o microcontrolador ATmega16U2, programado para atuar como um conversor USB-série, permitindo a comunicação com o computador através da ligação USB. Isto pode ser utilizado para enviar mensagens de *debug* durante testes no desenvolvimento do código, utilizando o monitor série no ambiente de desenvolvimento Arduíno (em inglês, *Integrated Development Evironment* – IDE).

O Arduíno por sua vez é alimentado por uma fonte de alimentação de 5V, onde neste caso a fonte é a ligação USB com o computador.

3.1.1. RTC

Para realizar a contagem de tempo de forma independente, foi considerada uma placa a utilizar no protótipo (**Figura 30**) que contém um RTC DS1307Z+[1] de baixa potência com 56 bytes de memória não-volátil disponível para utilização, capaz de armazenar e fornecer através da comunicação série I^2C informações completas de data (dia da semana, dia do mês, mês e ano) e horário (horas, minutos e segundos), nos formatos de 12 ou 24 horas. Meses com menos de 31 dias e correções a anos bissextos são ajustados automaticamente [22].

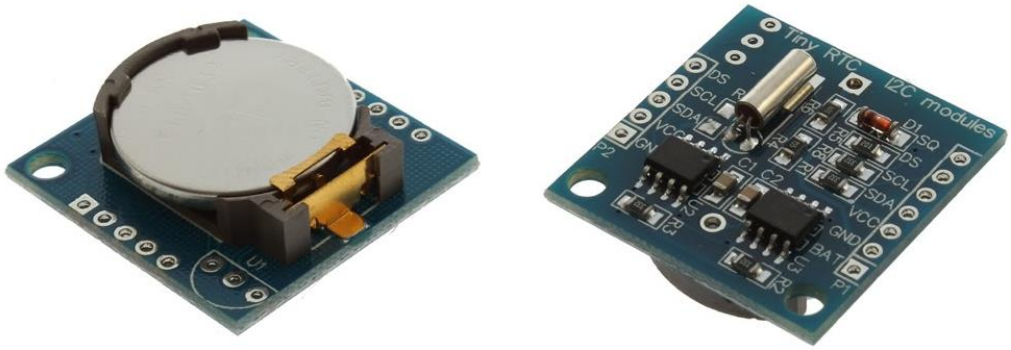


Figura 30 – RTC DS1307 [23].

[1] O "+" no número indicativo do RTC significa que tem um invólucro livre de chumbo [22].

Inclui um suporte para uma bateria de lítio padrão de 3V, garantindo que os dados sejam preservados mesmo quando existe falha da alimentação externa, devido ao circuito de deteção embebido no RTC DS1307Z+ que é acionado automaticamente em caso de falta de energia na placa, trocando para uma fonte de energia alternativa. Neste modo de operação consome menos de 500nA com o oscilador a funcionar, permitindo a contagem de tempo correta por mais de 10 anos na ausência de alimentação externa a 25ºC. O circuito do oscilador consiste num cristal de quartzo externa de 32,768 kHz ligado aos pinos X1 e X2 (**Figura 31**), afetando a precisão do *Clock*. O RTC inclui também um pino de saída de onda quadrática de quatro frequências possíveis (1 Hz, 4 kHz, 8 kHz, 32 kHz) para ligação a uma unidade de processamento [22]. Neste projeto não será necessária a utilização deste pino.

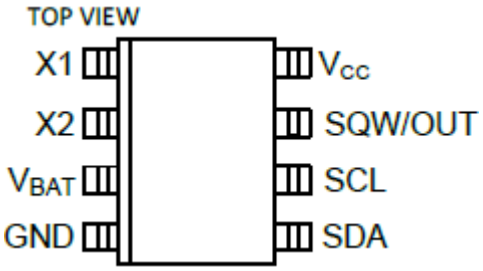


Figura 31 – Entradas e saídas do RTC DS1307Z+ [22].

Outra característica desta placa é a capacidade da utilização opcional de um sensor de temperatura DS18B20 (não incluído), lendo as informações deste sensor a partir do pino DS da placa caso esteja instalado, sendo possível montar um relógio completo com data, horas e temperatura, sem a necessidade de incluir outros componentes. Neste projeto foi feita a opção por não incluir este sensor de temperatura [23].

Por último e como referido anteriormente, esta placa tem incluída 2 resistências de *pull-up*, tendo assim o benefício de apenas necessitar de ligações dos pinos SDA e SCL aos portos apropriados do Arduíno.

3.1.2. SENSORES

O módulo de controlo local utiliza 3 tipos de sensores para a recolha de dados ambientais (**Figura 32**), sendo eles: sensor DHT11 (temperatura e humidade), *Light Dependent Resistor* – LDR – (luz) e MH Sensor-series (humidade do solo).

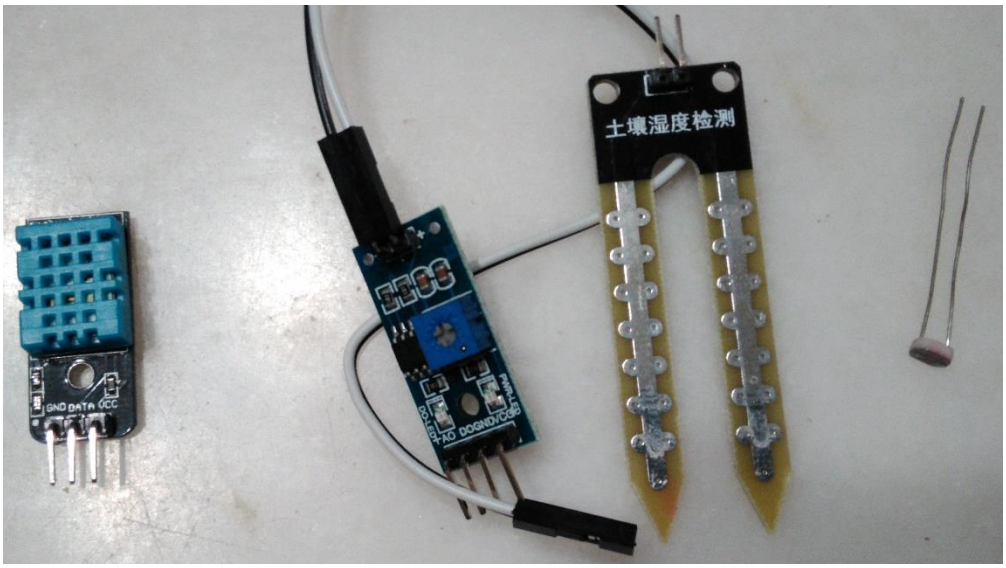


Figura 32 – Sensor DHT11, humidade do solo e LDR (esquerda, centro e direita respetivamente).

O sensor DHT11 existe em duas variantes, de 3 e 4 pinos. O sensor utilizado é a variante de 3 pinos que incorpora numa placa de *Printed Circuit Board* (PCB) uma resistência de *pull-up* para manter o pino de dados no estado alto padrão [24] e um led que mostra visualmente quando transmite informação. Esta versão oferece a vantagem de não necessitar de componentes extra, apenas de fazer as ligações apropriadas ao Arduíno e fonte de alimentação. O sensor propriamente dito (invólucro azul na figura anterior), inclui um componente de medição de humidade do tipo resistivo e um componente semicondutor sensível à temperatura, ou seja, um termístor, do tipo *Negative Temperature Coefficient* (NTC) no qual o coeficiente de variação de resistência com a temperatura é negativo, devido à sua resistência diminuir com o aumento da temperatura [25].

O vapor de água é detetado medindo a resistência elétrica entre dois elétrodos. Este componente do sensor é um substrato de retenção de humidade com elétrodos aplicados à superfície. Quando o vapor de água é absorvido pelo substrato, íons são libertados aumentando a condutividade entre elétrodos. Esta mudança na resistência entre os dois elétrodos é inversamente proporcional à humidade relativa (HR), ou seja, com uma humidade relativa alta a resistência entre elétrodos diminui, enquanto que numa humidade relativa baixa a resistência entre elétrodos aumenta [26].

Com uma resolução de 8 bits e uma voltagem de operação entre 3 e 5,5V, a precisão e alcance do sensor DHT11 em condições atmosféricas de 25ºC são:

- Alcance (humidade): 20-90% HR;
- Precisão (humidade): ±5% HR;
- Alcance (temperatura): 0-50 °C;
- Precisão (temperatura): ±2% °C [24].

O sensor DHT11 mede a humidade relativa, que é a quantidade de vapor de água no ar (ρ_w) versus o ponto de saturação do vapor de água no ar (ρ_s) de acordo com a fórmula (**1**).

$$\mathrm{HR} = (\rho_w / \rho_s) \times 100\ . \qquad (1)$$

No ponto de saturação, o vapor de água condensa e começa a acumular em superfícies formando orvalho, mas o ponto de saturação também muda consoante a temperatura do ar. Assim, isto significa que o ar frio retém menos vapor de água antes de ficar saturado e o ar quente é capaz de reter mais vapor de água antes de ficar saturado [26].

A comunicação e sincronização entre o sensor DHT11 e a unidade de processamento é feita através de uma única ligação ou *bus* (**Figura 33**). Uma transmissão completa consiste em 40 bits de dados, com o sensor a enviar a informação de bits mais significativos primeiro [24].

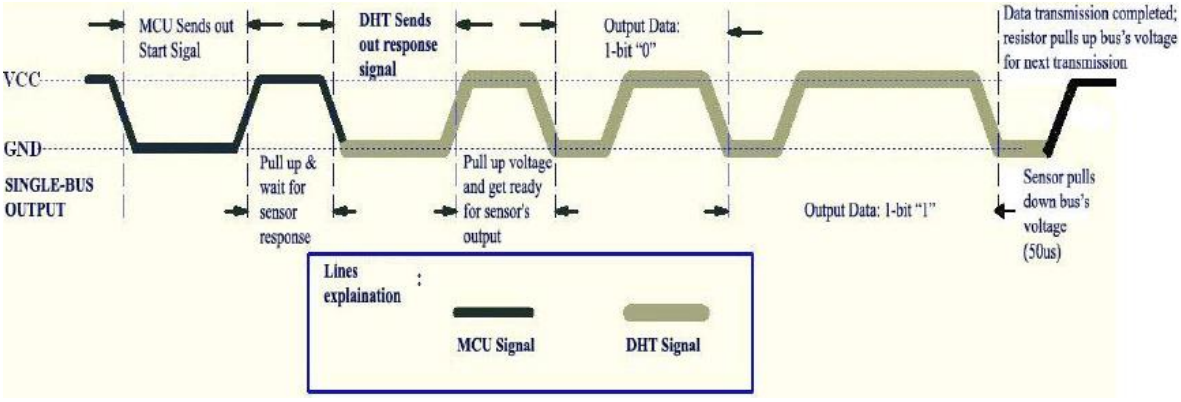


Figura 33 – Processo de comunicação do DHT11 [24].

No início da comunicação, com o *bus* em estado livre (linha em HIGH), quando um sinal de início é enviado por uma unidade de processamento (por exemplo um microcontrolador), o DHT11 muda do modo de baixo consumo energético para o modo de execução, aguardando

que o sinal de início seja concluído. Este sinal de início começa por passar o *bus* de dados de alta para baixa tensão, demorando cerca de 18ms para garantir a sua deteção pelo DHT e depois a passagem do sinal novamente para tensão alta esperando pela resposta do DHT11 durante 20-40us [24].

A seguir, tendo o DHT11 detetado o sinal de início, é enviado um sinal de resposta de dados de 40 bits que inclui as informações de humidade relativa e temperatura à unidade de processamento. Este sinal de resposta consiste, em primeiro lugar, na passagem do *bus* de alto para baixo durante 80us, voltando a alto por mais 80us em preparação para o envio de dados. Quando os dados estão a ser enviados pelo DHT11, cada bit começa com o *bus* em estado baixo por 50us passando a estado alto por um determinado tempo que especifica se o bit é 0 ou 1 (26-28us e 70us respetivamente) (**Figura 34**). Quando o último bit for transmitido, o nível de tensão é colocado pela última vez em baixo por 50us antes de voltar ao nível alto significando que o *bus* se encontra em estado livre. Assim, depois de todos os dados serem recebidos, o DHT11 muda para o modo de baixo consumo de energia até receber novamente um sinal de início da unidade de processamento [24].

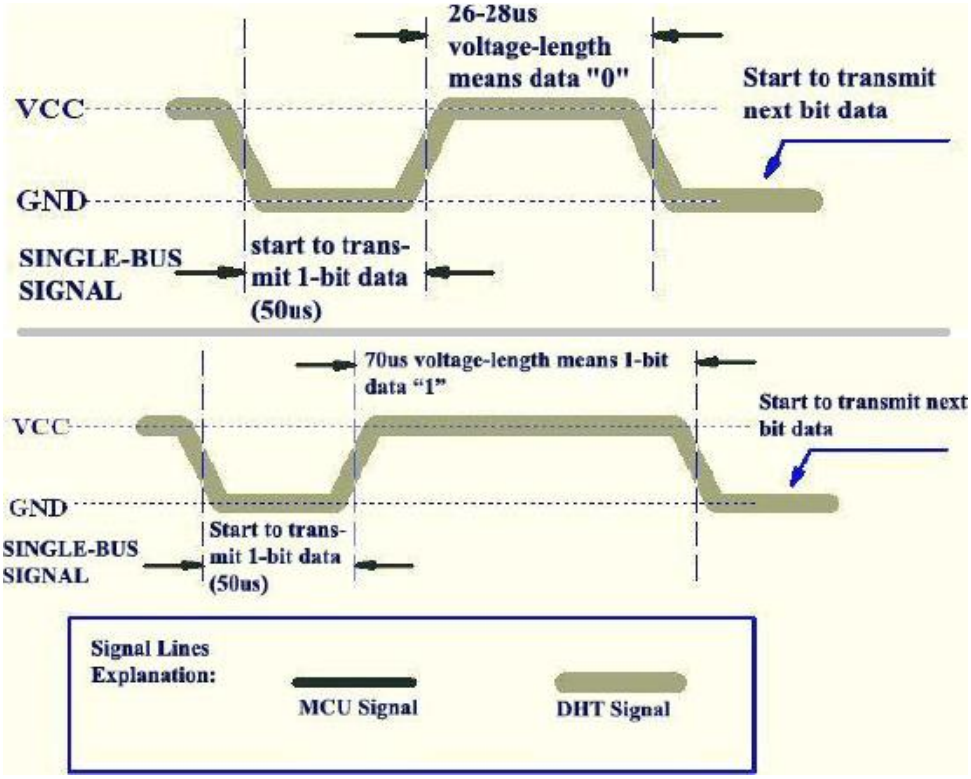


Figura 34 – Nível de sinal no bus para transmitir o bit 0 (cima) e 1 (baixo) [24].

Em termos de caraterísticas elétricas do sensor DHT11, este é alimentado tipicamente por uma fonte de alimentação *Direct Current* (DC) de 5V, não devendo ser alimentado para além do valor máximo de operação ou incorrerá num desfasamento de 3% na leitura da humidade relativa. O intervalo de amostragem do DHT11, necessário para uma recolha de dados efetiva, deve ter um mínimo de 1 segundo. Na próxima tabela (**Tabela 2**) é compilada todas as caraterísticas elétricas do sensor a uma temperatura de 25ºC [24].

Tabela 2 – Características elétricas [24].

	Condições	Mínimo	Típico	Máximo
Fonte de alimentação	DC	3V	5V	5,5V
Fonte de corrente	Medição	0.5mA	-	2,5mA
	Média	0.2mA	-	1mA
	Standby	100uA	-	150uA
Período de amostra	Segundos	1	-	-

O sensor de humidade do solo utilizado, tal como o nome indica foi desenvolvido para detetar as variações de humidade do solo, através de duas sondas que medem a resistividade do solo consoante o conteúdo volumétrico de água. Estas duas sondas permitem que a corrente passe pelo solo e obter o valor de resistência para medir o valor de humidade presente. Quanto maior a quantidade de água no solo, maior é a sua condutividade, o que significa que haverá menos resistência. Deste modo, o nível de humidade é maior. Num solo seco, a quantidade de água no solo é pequena ou inexistente, logo a condutividade é menor e o valor de resistência obtido é maior. Neste caso, o nível de humidade é menor [28].

Este sensor é constituído por uma placa PCB (**Figura 35**) com um comparador LM393, um potenciómetro, 2 leds indicando a saída de dados e o fornecimento de energia, resistências de vários valores, 2 pinos de saída (*Analog Output* – AO; *Digital Output* – DO), 2 pinos de entrada para a recolha dos dados das sondas e 2 pinos de alimentação (VCC e GND). As duas saídas AO e DO oferecem a vantagem de escolha quanto ao modo de funcionamento

utilizado: modo analógico ou digital [28]. Neste projeto será utilizado o modo analógico por oferecer um maior controlo na manipulação dos dados pelo *software*.

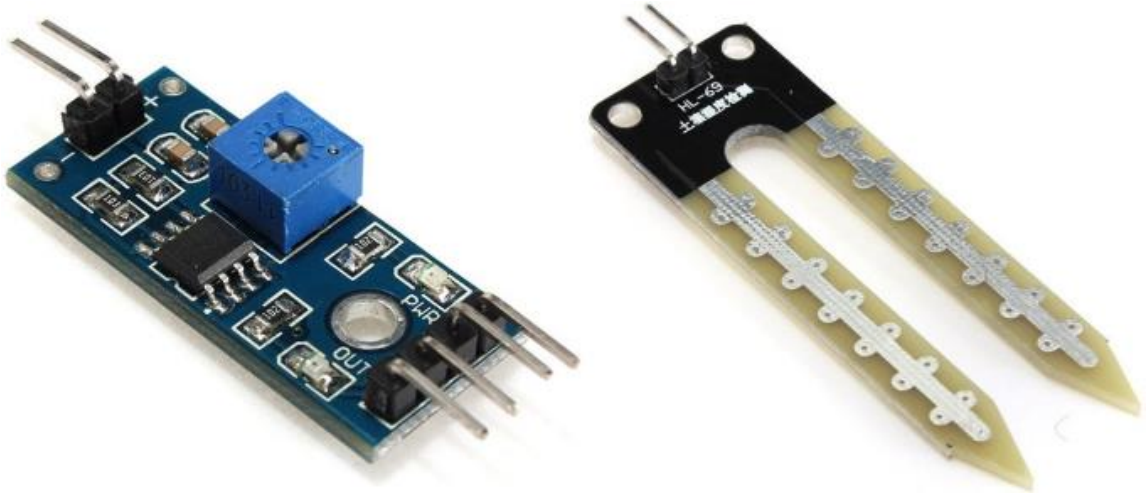


Figura 35 – Componentes da placa do sensor de humidade do solo [27].

O modo digital funciona através da comparação no LM393 do valor obtido pelo sinal das sondas com o valor definido no potenciómetro (valor de referência). Quando o valor do sinal for maior do que o valor de referência, a saída DO terá o valor de 5V e o led de saída acenderá. No entanto, para um valor do sinal menor de que o de referência, DO terá o valor de 0V e o led desligará (**Figura 36**) [28]. Assim, o limite entre o estado do solo seco e húmido pode ser ajustado através do potenciómetro presente para regular a saída digital DO [27].

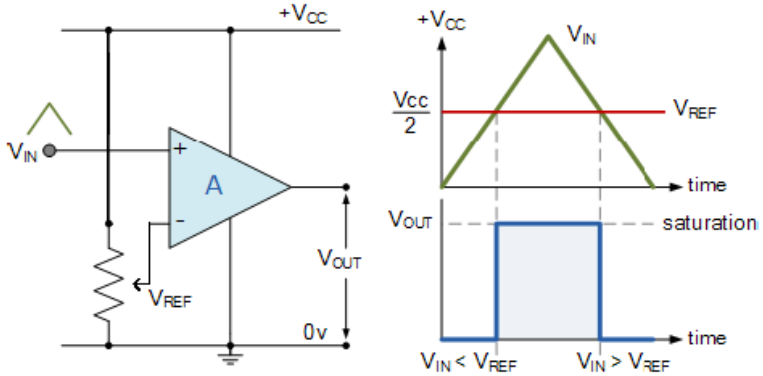


Figura 36 – Princípio de funcionamento do comparador LM393.

O modo analógico apenas necessita da ligação da saída AO com uma unidade de processamento com ADC.

As especificações elétricas do sensor de humidade do solo são:

- **Tensão de operação:** 3,3 – 5V;
- **Corrente de operação:** 35mA;
- **Tensão de saída:** 0 – 4,2V [28].

Por último, um sensor LDR GL5528 de 5mm de diâmetro (**Figura 37**), um componente cuja resistência varia de acordo com a intensidade da luz. Quanto mais luz incidir no sensor, menor a sua resistência e vice-versa [29].

Figura 37 – LDR GL5528 [29].

Os LDR modernos são feitos de materiais fotocondutivos como sulfeto de chumbo, seleneto de chumbo, antimoneto de índio e os tipos mais comuns de sulfeto de cádmio e seleneto de cádmio. O LDR GL5528 utiliza sulfeto de cádmio para a sua camada fotossensível, sendo normalmente indicados como fotorresistencias CdS. Os elétrodos são evaporados a vácuo na superfície de um dos lados para formar pentes de intercalação e pinos de ligação são afixados (**Figura 38**). Para evitar contaminação desta superfície, esta fica encapsulada em plástico transparente [30].

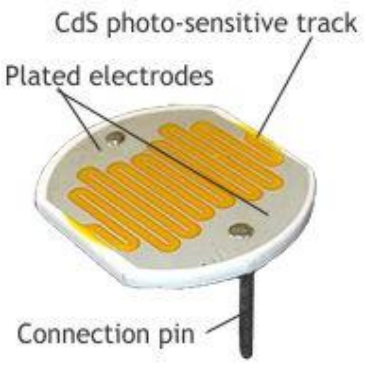


Figura 38 – Componentes de um LDR [30].

Baseado nos materiais utlizados um LDR pode ser de dois tipos: intrínseco e extrínseco. Um LDR intrínseco utiliza materiais sem dopagem como silicone ou germânio. Neste, os fotões que caem no dispositivo excitam os eletrões da banda de valência para a banda de condução, o que resulta em mais eletrões livres, que podem transportar corrente, e assim, menor resistência. Um LDR extrínseco utiliza materiais dopados com impurezas, onde os eletrões precisam de menos energia na transição para a banda de condução, devido a uma menor diferença de energia. Isto resulta num dispositivo sensível a diferentes comprimentos de onda de luz [30].

Como a sensibilidade de um LDR varia com o comprimento de onda da luz incidente, se esse comprimento estiver fora de um determinado intervalo, a sua resistência não será afetada. Consoante o seu material constituinte, o LDR terá curvas de resposta espectral diferentes e únicas de comprimento de onda versus sensibilidade (**Figura 39**). Um LDR extrínseco é geralmente projetado para comprimentos de onda mais longos, com tendência para o infravermelho (*Infrared* – IR) [30].

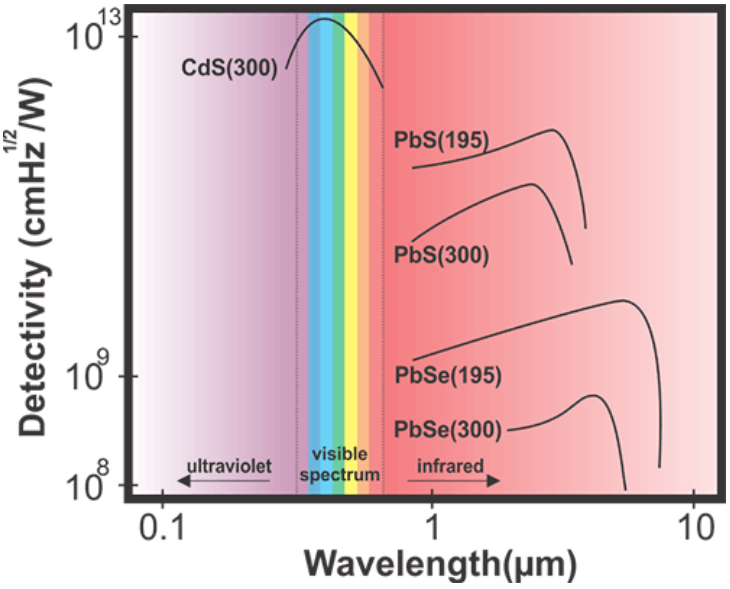


Figura 39 – Curvas espectrais de diferentes materiais constituintes (temperatura expressa em Kelvin entre parênteses) [30].

Como é possível observar na figura anterior, a curva da resposta espectral do sulfeto de cádmio corresponde ao espetro visível.

As especificações técnicas do LDR modelo GL5528 são as seguintes:

- **Tensão máxima:** 150V;
- **Potência máxima:** 100mW;
- **Tensão de operação:** -30°C a 70°C;
- **Espectro:** 540nm;
- **Resistência no escuro:** 1 MΩ (Lux 0);
- **Resistência na luz:** 10-20 KΩ (Lux 10) [29].

3.1.3. ATUADORES (MOTOR DE PASSO E MOTOR DC)

O módulo local utiliza um motor de passo, modelo 28BYJ-48 (**Figura 40**), para a abertura e fecho do que passará como a janela superior de uma estufa no protótipo construído. Este motor foi escolhido devido à sua capacidade precisa de movimento em incrementos nos dois sentidos, mantendo a sua posição especifica quando parado. Isto advém do facto do motor de passo poder ser controlado de forma precisa em sistema de malha aberta. Neste tipo de sistema de controlo nenhuma informação de feedback sobre a posição é necessária, eliminando a necessidade de dispositivos de deteção e feedback, como *encoders* óticos, apenas sendo necessário manter o conhecimento do número de passos incrementados para acompanhar a posição [33].

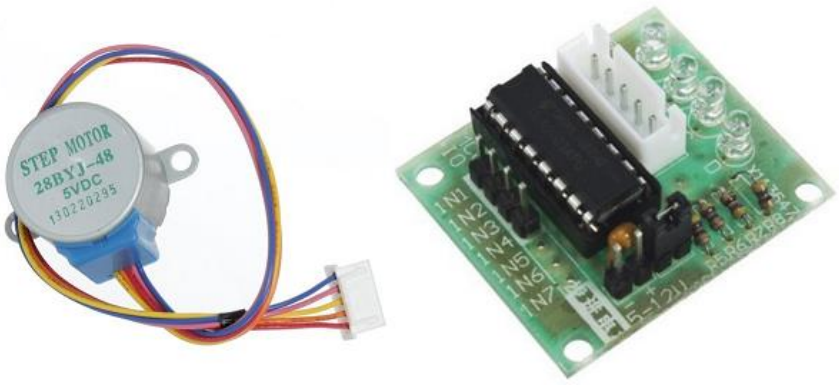


Figura 40 – Motor de passo e Driver ULN2003APG.

A definição de um motor de passo é a de um dispositivo eletromecânico sem escovas que converte a energia de impulsos elétricos em movimentos mecânicos discretos. Quando um impulso ou sequência de impulsos são aplicados através de um circuito externo de controlo (por exemplo, um microcontrolador), o motor gira um certo ângulo onde o ângulo de movimento gerado por um impulso é referido de ângulo de passo (ou *step angle* em inglês). Quanto menor for este ângulo de passo, maior o número de passos para completar uma revolução e assim mais preciso será o motor de passo para se movimentar para uma posição indicada. Alguns dos ângulos de passo mais comuns são de 1,8º, 2,5º e 7,5º [31] [32].

O controlo de direção neste tipo de motor apenas depende da ordem de sequência de impulsos aplicados ao estator ou bobinas, enquanto que a velocidade média do motor é proporcional à frequência desses impulsos. Quanto maior a sua velocidade, também menor será a perceção dos passos ou incrementos do motor [32].

De forma simplificada, o princípio de funcionamento dos motores de passo tem por base a disposição de vários eletroímanes de forma "dentada" em torno de uma engrenagem (**Figura 41**) e com estes eletroímanes a receberem os impulsos do circuito de controlo externo, também chamado de *driver*. Quando o primeiro eletroíman recebe o impulso, os dentes da engrenagem são atraídos magneticamente ao eletroíman ficando os dois alinhados. Quando o próximo eletroíman for ligado e o primeiro desligado, a engrenagem acaba por girar levemente para se alinhar com o próximo, movendo um passo, com este processo a ser repetido até completar uma volta completa [31].

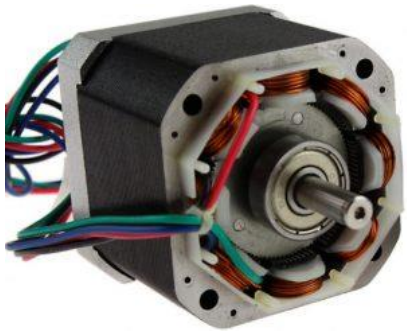


Figura 41 – Disposição interna dos componentes de um motor de passo [32].

Com base no arranjo de enrolamento das bobinas, os motores de passo podem ser unipolares ou bipolares. Num motor unipolar existem dois enrolamentos por fase com um ponto central comum, com o fluxo de corrente a ir numa direção em uma bobina e na direção oposta na outra bobina. Deste modo, esta configuração permite o controlo de direção do motor sem

inverter o sentido do fluxo de corrente, mas trocando os terminais ligados para cada bobina. Num motor bipolar, cada fase é constituída por um único enrolamento, significando que para controlar a direção de rotação do motor é necessário inverter o fluxo de corrente, o que requer um circuito de controlo mais complexo [32]. O motor de passo utilizado, de modelo 28BYJ-48, é um motor unipolar com a configuração do enrolamento demonstrado na figura seguinte (**Figura 42**).

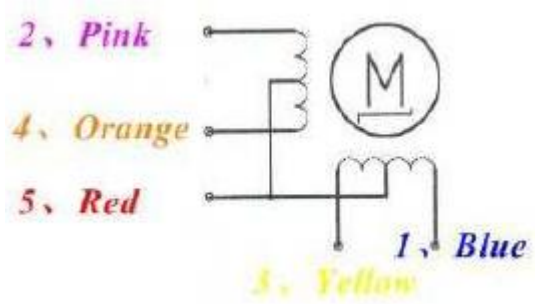


Figura 42 – Disposição das ligações aos enrolamentos do motor de passo 28BYJ-48 [34].

A rotação, e o correspondente incremento de passos do motor, pode ser feita de várias maneiras, dependendo da sequência em que as bobinas são ativadas. As formas mais comuns de operação (**Figura 43**) são:

- ***Wave step*:** o modo mais simples de operação, onde apenas uma fase é ativada de cada vez. Oferece o maior ângulo de passo em comparação com os outros modos, mas o torque é reduzido [32];

- ***Full-step*:** ao longo do tempo, duas fases encontram-se sempre ativas simultaneamente e assim que uma dessas fases desliga outra é ativada. Quando as duas fases estão ativas o rotor está numa posição de equilíbrio entre elas, conseguindo um torque elevado [32];

- ***Half-step*:** uma combinação dos dois modos anteriores, o controlo é alternado entre duas e uma fase ativa. Como o ângulo de passo é metade daquele no modo *full-step*, o número de passos para completar uma rotação será o dobro do modo anterior [32];

- ***Microstepping*:** cada passo do motor é subdividido em vários passos menores aumentado a resolução obtida. Para conseguir esta operação, as duas fases que são ativadas simultaneamente têm valores de corrente variadas diferentes uma da outra. Por exemplo, com uma das fases ativa e corrente constante, a corrente da segunda

fase é incrementada até ao seu máximo (negativo ou positivo). Depois, o inverso acontece com a segunda fase a manter a sua corrente e a corrente da primeira fase a variar [32].

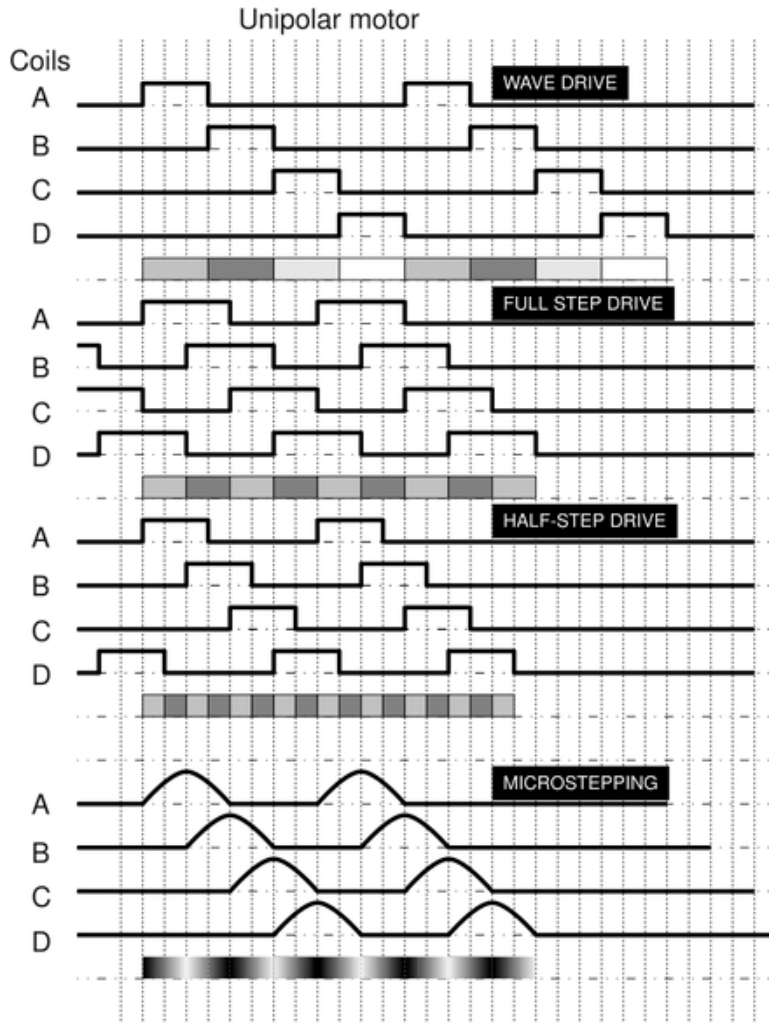


Figura 43 – Modos de operação de um motor de passo unipolar de 4 fases [35].

O motor de passo 28BYJ-48 é um motor unipolar, de tensão de alimentação de 5V, que utiliza o *driver* ULN2003A para o seu controlo e interface com uma unidade de processamento como o Arduíno. Algumas das especificações técnicas mais relevantes deste motor são que tem 4 fases, ângulo de passo de 5,625º e 64 passos para completar uma revolução completa da engrenagem interna [34] [36]. A tabela seguinte, retirada do *datasheet*, mostra as especificações completas.

Tabela 3 – Especificações técnicas do motor de passo 28BYJ-48 [34].

Rated voltage	5VDC
Number of Phase	4
Speed Variation Ratio	1/64
Stride Angle	5.625° /64
Frequency	100Hz
DC resistance	50Ω±7%(25℃)
Idle In-traction Frequency	> 600Hz
Idle Out-traction Frequency	> 1000Hz
In-traction Torque	>34.3mN.m(120Hz)
Self-positioning Torque	>34.3mN.m
Friction torque	600-1200 gf.cm
Pull in torque	300 gf.cm
Insulated resistance	>10MΩ(500V)
Insulated electricity power	600VAC/1mA/1s
Insulation grade	A
Rise in Temperature	<40K(120Hz)

O *Driver* ULN2003A (**Figura 44**) é um *driver* de corrente que permite ao Arduíno controlar os motores com correntes superiores a 50mA, neste caso até 500mA. Esta placa possui leds que indicam a ativação das bobinas e opera com tensões de 5-12V [36].

Figura 44 – *Driver* ULN2003A de controlo de motores de 4 fases [36].

No caso do motor DC (**Figura 45**), este é um motor de tensão nominal 12V, operando entre tensões de 6 e 14V, e com uma velocidade máxima sem carga de 11500 rotações por minuto (rpm). A corrente nominal é de 180mA. De referir que a velocidade num motor DC deste tipo é diretamente proporcional à tensão aplicada, enquanto que o binário é proporcional à corrente [37].

Figura 45 – Motor DC [37].

Este motor utiliza um *driver* L293D (**Figura 46**), controlador capaz de controlar 2 motores DC em simultâneo, assim como a sua velocidade e direção. Este controlador é necessário devido aos portos do Arduíno não fornecerem energia suficiente para fazer o motor DC funcionar, o que faz com que seja necessário a utilização deste controlador externo [38].

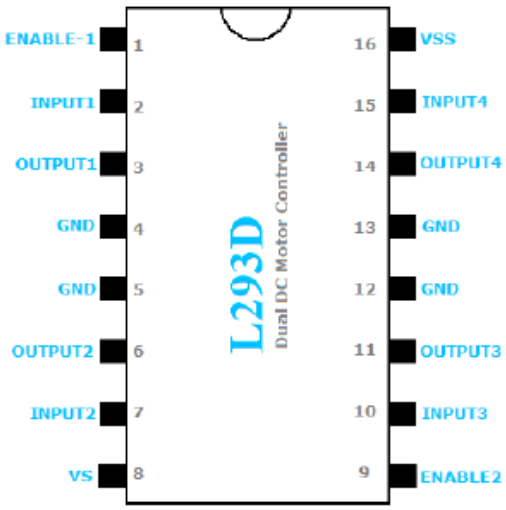


Figura 46 – Driver L293D e as suas entradas e saídas [38].

3.1.4. LCD E BOTÕES

O LCD utilizado é um dispositivo com 2 linhas e 16 colunas, com cada coluna a ser constituída em dimensões de 5 por 8 pontos (**Figura 47**). O *display* tem também a capacidade de variação da luz de fundo (led de baixo consumo) de forma a melhorar a visualização dos caracteres [39]. O LCD será interface de visualização no módulo local com os botões a permitirem a navegação pelo menu que será visualizado.

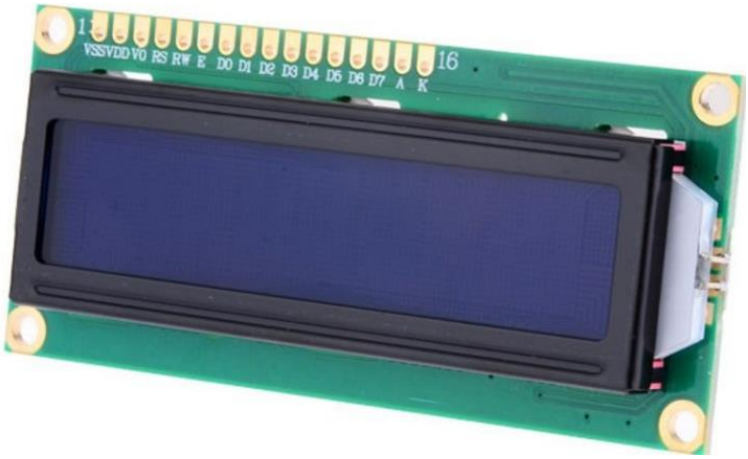


Figura 47 – LCD de 16x2 com fundo azul [39].

Este é um LCD bastante versátil compatível com um grande número de sistemas como Arduíno, PIC, Atmel, entre outros. Este pode também funcionar em dois modos de comunicação, de 4 bits ou 8 bits [39]. Para as caraterísticas técnicas, a próxima tabela detalha as mais importantes.

Tabela 4 – Especificações técnicas do LCD [39].

Tensão de trabalho	4.5V – 5.5V
Corrente de trabalho	1.0mA – 1.5mA
Tensão do LED (backligth)	1.5V – 5.5V
Corrente do LED (backligth)	75mA – 200mA

Os botões de pressão (**Figura 48**) serão o meio de interação entre o utilizador e o módulo local. Para esta interação existirão 4 botões, cada um com as suas funções bem definidas. Dois destes botões têm como funções incrementar e decrementar valores ou posições no

menu (botões “cima” e “baixo”), um é o botão de anulação/retrocesso no menu (botão “Esc”) e o último é o botão de confirmação (botão “OK”).

Figura 48 – Botões de pressão [40].

3.1.5. EMISSOR WIFI

O módulo WiFi ESP8266 (**Figura 49**) permite a ligação sem fios entre o módulo de controlo local com o módulo remoto. O emissor suporta as redes WiFi 802.11 b/g/n, utilizadas na maioria das ligações sem fios como as que são utilizadas num PC portátil ou *Smartphone*, podendo trabalhar como Ponto de Acesso (PA) (em inglês, *Acess Point* – AP) ou como uma Estação, enviando e recebendo dados [42].

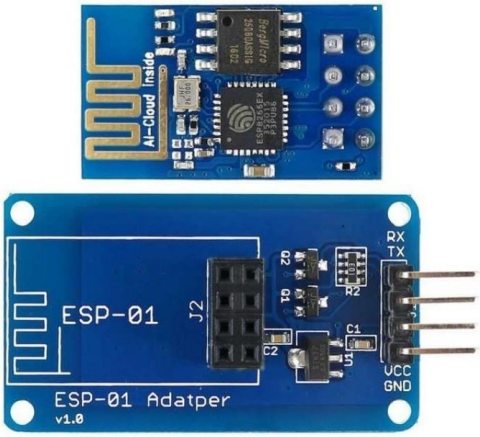


Figura 49 – Emissor WiFi ESP8266 (cima) e o seu adaptador WiFi ESP-01 (baixo) [41].

A comunicação do emissor com o Arduíno pode ser feita via ligação série através dos pinos RX e TX, possível de ser configurada através dos comandos AT. Como a tensão de operação do emissor é de 3,3V é utilizado o adaptador WiFi ESP-01 que possui um conversor de nível e pode ser ligado diretamente ao Arduíno [42]. As especificações técnicas são expostas na tabela seguinte.

Tabela 5 – Especificações técnicas do emissor WiFi ESP8266 [42].

Tensão da operação	3,3V
Suporte de redes	802.11 b/g/n
Alcance	aproximadamente 90m
Comunicação	Série (TX/RX)
Suporta comunicação	TCP e UDP
Conectores	GPIO, I2C, SPI, UART, Entrada ADC, Saída PWM e Sensor de Temperatura interno
Modo de segurança	OPEN, WEP, WPA_PSK, WPA2_PSK, WPA_WPA2_PSK

3.1.6. DISPOSIÇÃO DO MENU E FLUXOGRAMAS

O menu a implementar no módulo local tem 5 menus principais, com 3 deles a terem 3 submenus cada (**Figura 50**), e a sua seleção é realizada através de botões de pressão.

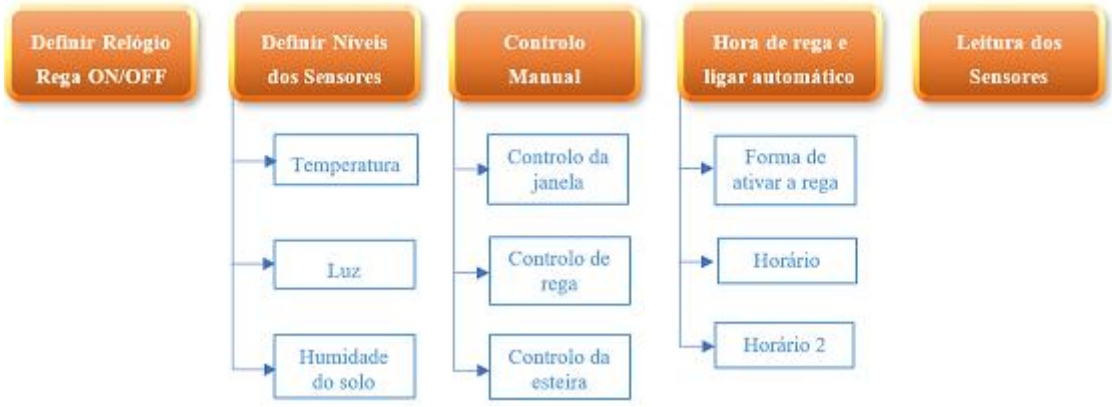


Figura 50 – Esquema do menu no módulo local.

Os menus principais e suas funções são:

- **Definir Relógio:** uma vez selecionado permite ajustar as horas e minutos do RTC, assim como a data (dia, mês e ano). Cada dígito será modificado individualmente, com o cursor no LCD a indicar qual é aquele que se encontra selecionado. Caso o menu não seja selecionado, mostra se a função de rega está ativa ou não;

- **Definir Níveis dos Sensores:** a sua seleção dará acesso a 3 submenus (temperatura, luz e humidade do solo) que permitem modificar os valores de controlo máximos e mínimos, com os dois últimos a serem em forma de percentagem. Estes valores de controlo quando comparados com os valores dos sensores permitirão os atuadores desempenhar a sua função automaticamente;

- **Controlo Manual:** quando selecionado, o módulo local deixa de atuar automaticamente, passando a ser o utilizador a controlar as ações dos atuadores. Cada atuador tem o seu submenu (controlo de janela, controlo de rega e controlo da esteira) que uma vez selecionadas dão acesso a decisões binárias (abrir/fechar, ON/OFF) ao utilizador. Quando o utilizador sair deste menu principal, o módulo passará novamente a modo automático;

- **Hora de rega e ligar automático:** constituído por 3 submenus (forma de ativar a rega, horário e horário 2), consoante a forma de rega em utilização pode ser apenas visualizado 2 submenus (forma de ativar a rega e horário). O primeiro submenu, tal como o nome indica, é dedicado a selecionar a forma de ativar a rega, ou seja, o seu modo de controlo. Existem 4 modos, um onde a rega liga ou desliga consoante o nível do sensor, outro chamado "Diário c/sensor" que tem a mesma função que o modo anterior, mas apenas na hora definida em "Horário", o 3º modo de nome "Diário s/sensor" ativa a rega na hora definida em "Horário" e desliga passado 2 minutos, e por último o modo "Bidiário s/sensor" que funciona como o modo anterior, mas em duas horas diferentes definidas em "Horário" e "Horário 2" (que só aparece se este modo estiver selecionado). Os 2 últimos submenus deste menu principal servem, como referido, para definir as horas de ativação da rega, com cada dígito modificado individualmente e selecionado por um cursor piscante;

- **Leitura dos Sensores:** este menu quando selecionado permite visualizar todos os valores recolhidos pelos sensores (temperatura, humidade do ar, luz e humidade do solo). A temperatura é visualizada em graus Celsius, enquanto que todos os outros se encontram em percentagem.

A arquitetura do programa principal do módulo de controlo local pode ser observada no fluxograma seguinte (**Figura 51**). Neste as horas e data são recolhidas do RTC em cada ciclo, com a leitura dos sensores a ser feita de 2 em 2 segundos. Este intervalo foi escolhido, em primeiro lugar, porque o sensor mais lento (DHT11) requer um intervalo mínimo de 1 segundo para recolher informações e em segundo, por ser necessário que o intervalo não seja demasiado grande para que seja discernível durante o funcionamento do protótipo, de forma atempada, das mudanças nos valores. A seguir, é feito o envio e receção de dados pela placa WiFi, para que hierarquicamente os comandos remotos tomem procedência aos comandos locais do menu. Depois a função do menu que inclui a leitura dos botões e por fim a função que controla o funcionamento dos atuadores (motor de passo, motor DC, rega).

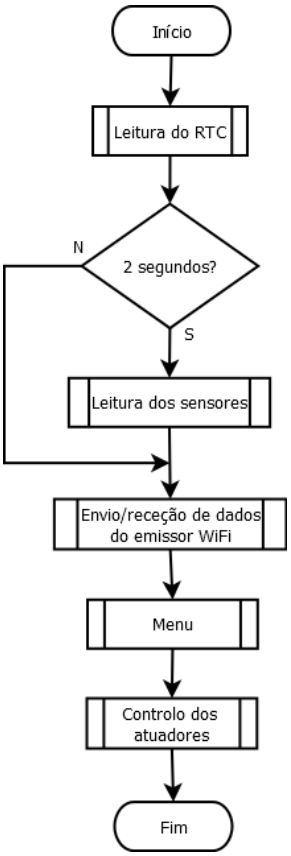


Figura 51 – Fluxograma do programa principal.

3.2. MÓDULO REMOTO

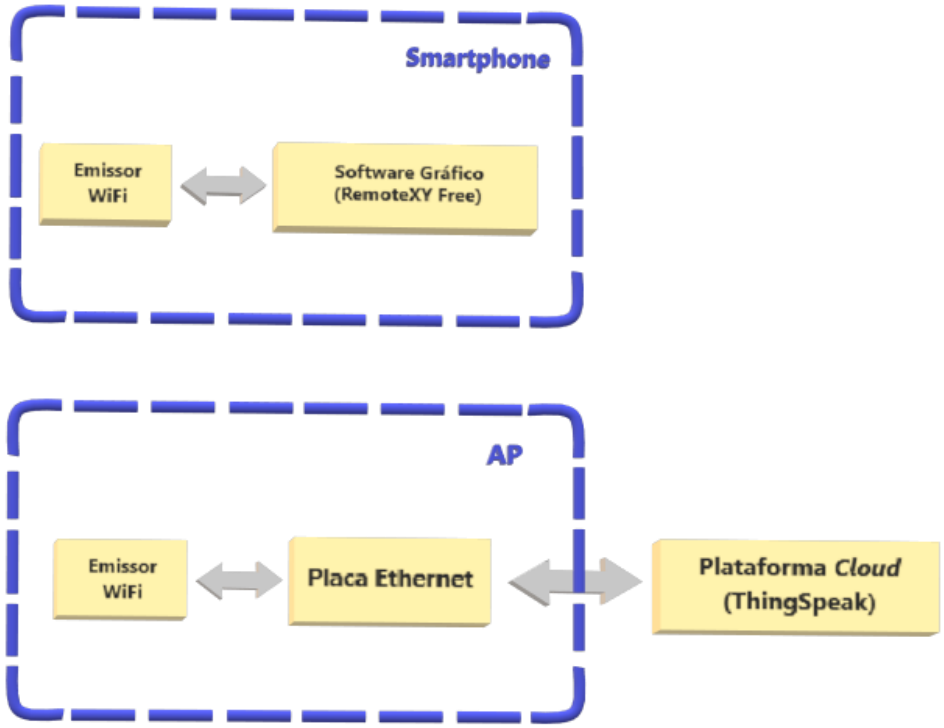


Figura 52 – Arquitetura do módulo remoto.

Como pode ser observado pela figura anterior, o módulo remoto é constituído por dois subsistemas, o primeiro por um *Smartphone* com um emissor/recetor *WiFi* e que tenha instalado o *software* gráfico da RemoteXY e o segundo um servidor *cloud* chamado *ThingSpeak* ligado ao módulo local através de um AP que age como intermediário. O primeiro subsistema, com o *software* RemoteXY permitirá a visualização dos dados recolhidos pelos sensores no módulo de controlo local e o envio de comandos de forma remota, enquanto que o segundo subsistema permite, para além da recolha de dados, a verificação da possibilidade de existência de várias formas de ligação ao módulo local (direto como no primeiro caso ou por um servidor no segundo).

A RemoteXY é um sistema que permite a criação de interfaces gráficas móveis para circuitos de controlo através de um editor localizado na página *web* http://remotexy.com/ e com a utilização de uma aplicação móvel que conecta e controla o módulo remotamente. Após o desenvolvimento da interface gráfica no editor *online*, o código fonte do microcontrolador pode ser obtido e armazenado num microcontrolador para implementar a interface, que permite a fácil integração de um sistema de controlo consoante as necessidades e as tarefas especificas para a qual o dispositivo é desenvolvido [45].

Uma das grandes vantagens de o código fonte da interface ser armazenada no dispositivo do microcontrolador é a utilização de apenas uma aplicação móvel, que pode gerir um grande número de dispositivos com diferentes interfaces de controlo gráfico [45].

A *ThingSpeak* é um serviço de plataforma de análise IoT que permite agregar, visualizar e analisar transmissões de dados em tempo real na *cloud*. Esta plataforma permite o envio de dados a partir de vários dispositivos, criar visualização instantânea de dados em tempo real e enviar alertas. Graças à sua integração com *Matlab*, também permite a análise avançada dos dados recebidos [47].

4. IMPLEMENTAÇÃO

Como referido no capítulo anterior, o protótipo é constituído por duas estruturas fundamentais: o módulo de controlo local e o módulo remoto. Desta forma, e com a arquitetura preferencial do protótipo em mente, é inicialmente indicado os esquemas de montagem e ligação dos componentes para o primeiro módulo, e o esquema gráfico construído para o módulo remoto e sua ligação.

4.1. MÓDULO DE CONTROLO LOCAL

Para a construção e implementação da estrutura projetada são necessários os componentes referidos no capítulo **3.1** em conjunto com os seguintes elementos adicionais:

- 3 *breadboards* (onde os componentes são ligados);
- 4 resistências de 2,2 kΩ, uma de 220 Ω e outra de 10 kΩ;
- 1 potenciómetro de 10 kΩ;
- 1 bateria de 9V e 3 de 1,5V (conjunto de 4,5V);
- Fios de ligação (tipo fêmea-macho e macho-macho).

- 2 copos com terra (1 molhado e 1 seco);
- 16 cm de arame de 3 mm de espessura;
- 26 cm de arame de 1 mm de espessura;
- Duas secções de madeira redonda (17 e 23 cm de comprimento) de 6 mm de espessura;
- Tecido rendilhado que obscura a passagem de luz;
- Fio de cordel;
- Plástico transparente;
- Outros componentes de complemento.

Os 6 penúltimos itens compõem os elementos que são utilizados na construção de um pequeno protótipo em forma de caixa para ser visualizado os atuadores em funcionamento. A sua construção foi um dos últimos passos a realizar, começando primeiro pelas ligações descritas nos próximos capítulos.

4.1.1. CONFIGURAÇÃO DE ESP8266

É necessário ter em conta que antes de completar todas as ligações, o módulo *WiFi* ESP8266 precisa de ter o seu BAUD *rate* alterado da predefinição 115200 para uma taxa de dados menor (neste caso 9600 BAUD). Esta alteração é necessário por causa do tipo de Arduíno em utilização (Arduíno Uno), no qual uma taxa superior a 19200 provocará instabilidade. Deste modo, com o módulo ESP8266 ligado ao adaptador *WiFi* ESP-01, são feitas as conexões de acordo com a **Tabela 6**.

Tabela 6 – Ligação entre os pinos do Arduíno e adaptador ESP-01.

Pino Arduíno UNO	Pino ESP8266
0 (Rx)	Tx
1 (Tx)	Rx
5V	VCC
GND	GND

A seguir, é retirado o microcontrolador ATmega328p da placa do Arduíno Uno de forma a que a programação do módulo ESP8266 seja feita através da ligação com ATmega16U2, que atua como um conversor USB-série, em conjunto com o monitor série no ambiente de desenvolvimento do Arduíno. O monitor série deve estar configurado com 115200 BAUD e definição "Nova linha e retorno de linha" (**Figura 53**). Esta última definição é obrigatória para o funcionamento dos comandos AT utilizados para configurar ESP8266 [43].

Figura 53 – Definições na janela do monitor série.

Utilizando o monitor série é introduzido primeiro o comando "AT" para testar a ligação. Depois, os comandos "AT+RST" e "AT+GMR", onde o primeiro reinicia o módulo (prática inicial recomendada) e o segundo mostra informação sobre a versão de *firmware* do módulo [43], para verificar que conforma com os requerimentos do módulo remoto com aplicação RemoteXY (AT_v0.40 ou acima) [44]. As respostas podem ser observadas na **Figura 54**.

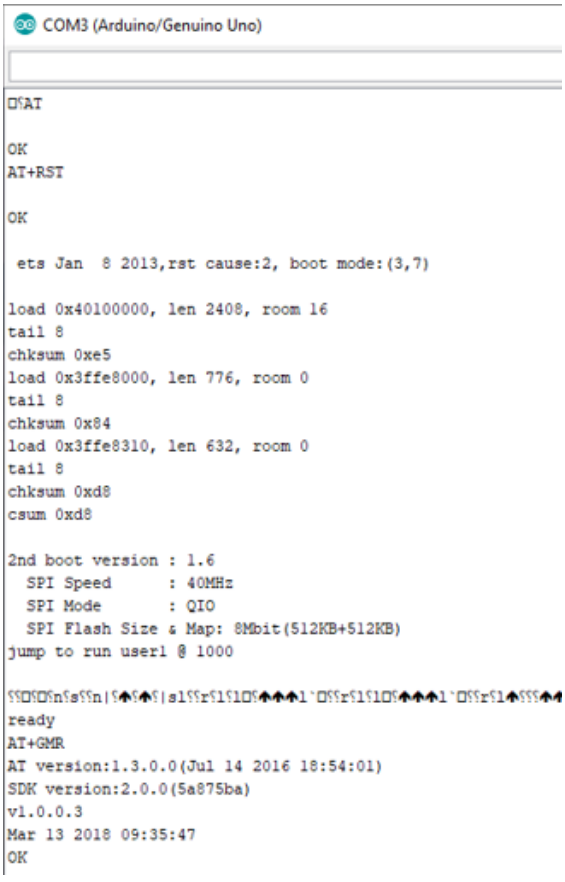


Figura 54 – Verificação inicial do módulo ESP8266.

A seguir, com o comando "AT+CWSAP_CUR" é anotado os parâmetros de ligação atuais para o módulo ESP8266, importante para a ligação do módulo remoto com RemoteXY, para o qual os passos de implementação são descritos no capítulo **4.2.1**. A resposta a este comando na **Figura 55** mostra que por ordem da esquerda para a direita, o *Service Set IDentifier* (SSID) tem o nome "FaryLink_556DB9", sem palavra-passe, canal 1, sem encriptação (*open*), 4 estações é o número máximo que pode se conectar e por último que o nome de SSDI é difundido.

```
AT+CWSAP_CUR?
+CWSAP_CUR:"FaryLink_556DB9","",1,0,4,0

OK
```

Figura 55 – Configurações de ligação de ESP8266.

Por fim, "AT+UART_DEF=9600,8,1,0,0" define a taxa de transmissão para 9600 BAUD, 8 bits de dados, 1 bit *stop*, sem bit de paridade e sem controlo de janela deslizante.

4.1.2. LIGAÇÕES

Com ESP8266 configurado, é possível fazer o resto das ligações de acordo com a **Figura 56**, com exceção das ligações entre o Arduíno com os botões e o *driver* (em conjunto com o motor) que são demonstradas à parte. Na figura seguinte, encontram-se algumas características importantes de notar, sendo estas: ligações com linhas vermelhas correspondem à alimentação de 5V e as linhas negras ao *ground* (GND), o potenciómetro de 10 kΩ permite controlar o contraste no LCD, tornando os caracteres mais visíveis se necessário e por último, que o motor de passo tem a sua própria fonte de alimentação com GND comum ao Arduíno (esta fonte corresponde à ligação em série de 3 pilhas de 1,5V).

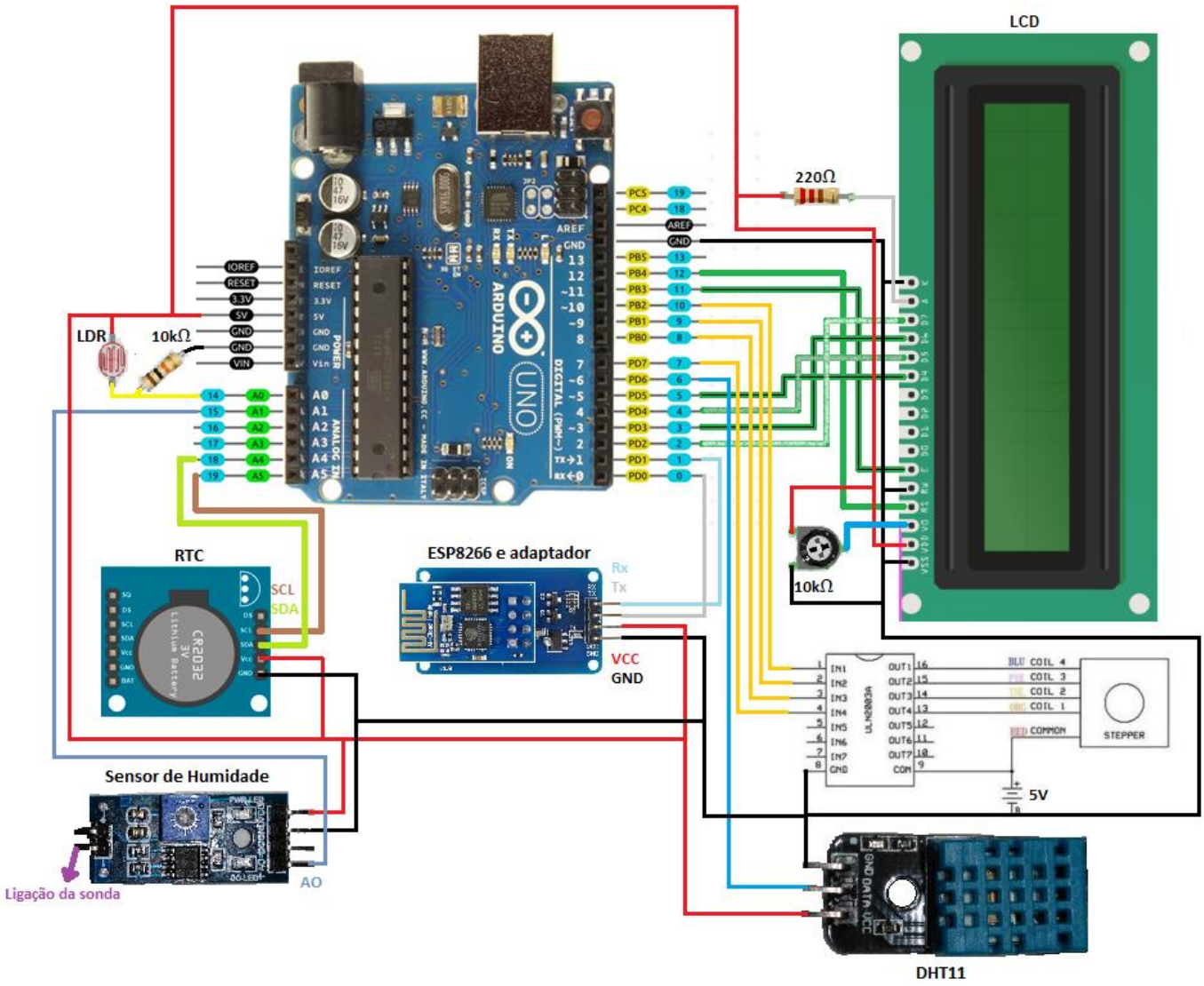


Figura 56 – Ligações completas com a exceção da interface de botões e motor com driver.

A interface do utilizador, que corresponde à utilização de 4 botões de pressão, foi montado de acordo com a **Figura 57**. Esta montagem segue o princípio de um potenciómetro onde, dependendo do botão pressionado, o Arduíno terá uma leitura diferente entre 0 e 1023, com o porto **A2** configurado para ativar a resistência interna de *pull-up* de modo a não existir um curto-circuito. Assim, quanto mais resistências existirem entre o botão pressionado e o porto, maior será o valor lido, como se de um divisor de tensão se tratasse. Este tipo de montagem permite poupar no número de entradas, possibilitando a utilização de apenas 1 porto em vez de 4 como seria necessário se todos os botões tivessem entradas individuais.

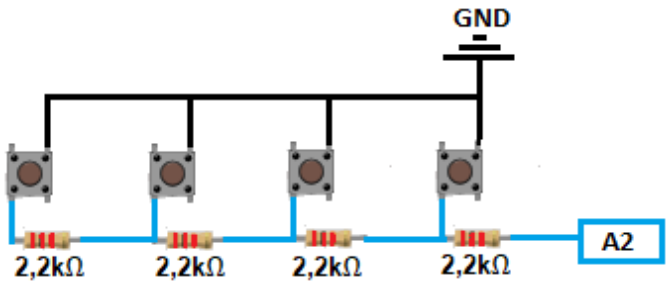


Figura 57 – Ligações dos botões ao Arduíno.

Por último, o motor DC tem as suas conexões representadas pela **Figura 58**. A fonte de alimentação do motor é de 9V com um GND comum ao Arduíno. As saídas **13** e **A3** do Arduíno controlam, através do *driver*, a direção e funcionamento do motor DC.

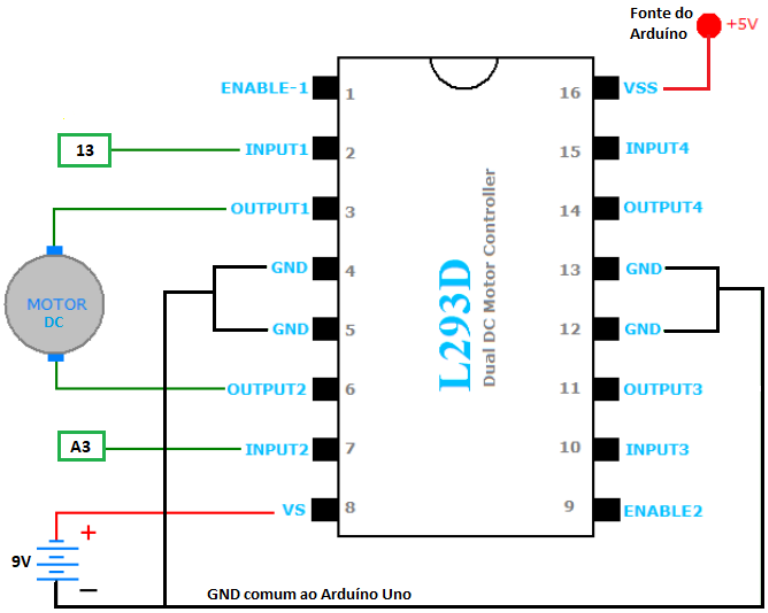


Figura 58 – Ligações do motor DC e *driver* ao Arduíno Uno.

4.1.3. CONSTRUÇÃO DA CAIXA DO PROTÓTIPO

Com a montagem do circuito completa, a última fase no desenvolvimento do protótipo físico passa pela construção de uma pequena caixa de madeira onde são fixados os dois motores existentes e é possível ver os atuadores em ação. Neste, o motor de passo é fixado à vara de madeira de 23 cm formando o eixo da janela. Neste eixo também é fixado o arame de 26 cm na forma de um retângulo e forrado com plástico, formando o equivalente a uma janela de estufa. Este eixo é fixado no centro e transversalmente à caixa.

O motor DC é conectado à vara de madeira de 17 cm formando o eixo da esteira, com o tecido rendilhado afixado neste mesmo eixo. O motor DC é afixado à caixa de forma perpendicular ao eixo do motor de passo num dos lados, enquanto que no lado oposto e de forma paralela é afixado o arame de 16 cm. Este arame servirá para montar uma espécie de correia em cada um dos seus lados com o fio de cordel, que permitirá manter o tecido suspenso e enrolá-lo ou abri-lo consoante o motor rode para um lado ou para o outro. Esta configuração torna desnecessária a utilização de 2 motores DC para abrir e fechar a esteira.

No final da construção e montagem do circuito do protótipo, este tem o aspeto representado na **Figura 59**.

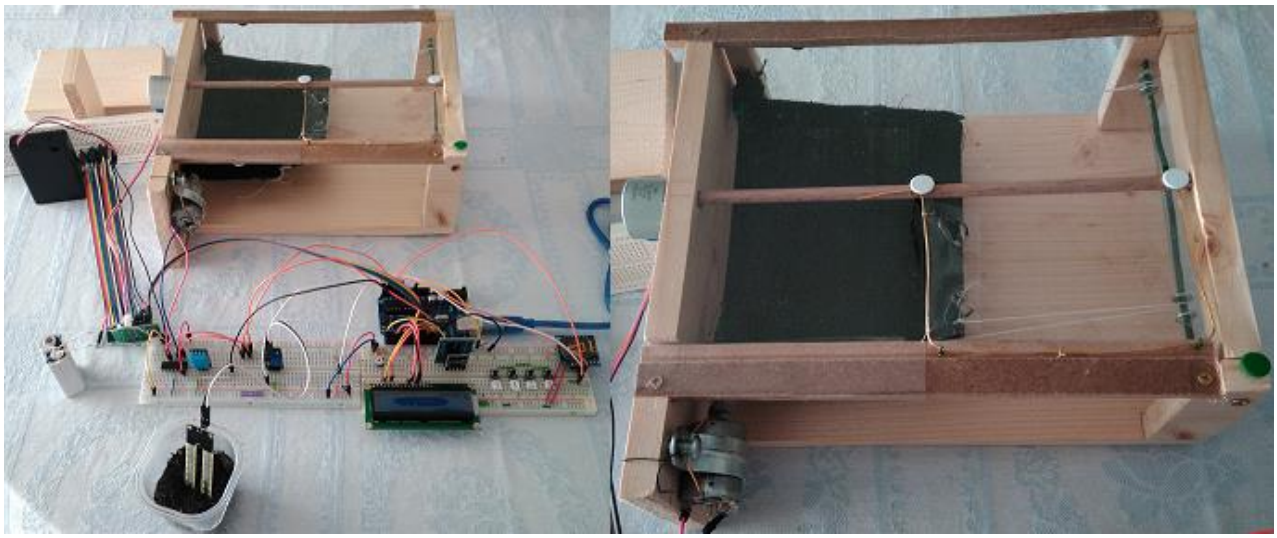


Figura 59 – Caixa do protótipo e circuito final.

4.1.4. DESENVOLVIMENTO DO CÓDIGO

Nesta secção será referido algumas partes extraídas de código relevantes e que possam suscitar alguma explicação detalhada para além dos comentários incluídos neste (Anexo A).

Começando pela inclusão das bibliotecas, deve ser referido que a sua maior parte são bibliotecas que se encontram disponíveis no Arduíno IDE ou que foram adicionadas através da interface disponibilizada no IDE para esta função em Rascunho>Incluir Biblioteca>Gerir Bibliotecas… (como no caso da biblioteca da Adafruit para o sensor DHT11), com duas exceções: da biblioteca do RTC DS1307 que foi extraído a partir do repositório *GitHub* na página https://github.com/filipeflop/DS1307.git, e para o módulo remoto extraído da página

web http://remotexy.com/en/library/. Com estas duas últimas bibliotecas no computador e o IDE do Arduíno aberto, clicar em Rascunho>Incluir Biblioteca>Adicionar Biblioteca .ZIP… e selecionar as bibliotecas a instalar.

A seguir, é apresentado o extrato da definição dos pinos de controlo do motor de passo e dos parâmetros de comunicação WiFi (mais informações no capítulo **4.2** em relação a código relacionado com o módulo remoto).

```
#define REMOTEXY_MODE__ESP8266_HARDSERIAL_POINT
#include <RemoteXY.h>

#define REMOTEXY_SERIAL Serial
#define REMOTEXY_SERIAL_SPEED 9600
#define REMOTEXY_WIFI_SSID "FaryLink_556DB9"
#define REMOTEXY_WIFI_PASSWORD "" //sem password.
#define REMOTEXY_SERVER_PORT 6377

#define WIFI2_SSID "SAPO-1234"
#define WIFI2_PASSWORD "palavra-passe"
#define WRITE_KEY "NM5TBO7KT2AGOXDG"

String tmp_comando = "AT+CWJAP_CUR=\"";
tmp_comando += WIFI2_SSID;
tmp_comando += "\",\"";
tmp_comando +=WIFI2_PASSWORD;
tmp_comando +="\"";
tmp_comando +="\r\n";
sendData(tmp_comando, 4000);

const byte stepsPerRevolution=64;
Stepper motorPasso(stepsPerRevolution,7,9,8,10);
```

A primeira linha deste extrato define qual o tipo de ligação série ao circuito que o módulo *WiFi* tem (a pinos arbitrários ou a pinos dedicados de Rx e Tx como é este o caso), com o segundo "#define" a fazer parte desta configuração para a aplicação RemoteXY. Este tipo de ligação requer alguns cuidados adicionais quando o *sketch* do programa for transmitido para o Arduíno, onde o módulo ESP8266, ou os seus pinos, não podem estar ligados com os pinos Rx e Tx do Arduíno ou ocorre uma mensagem de erro, devido a estes pinos estarem a ser utilizados na programação do ATmega328p. As três linhas seguintes são informações recolhidas aquando da configuração inicial no capítulo **4.1.1** (definição da taxa em 9600 BAUD) e das definições de ligação *WiFi* anotadas (**Figura 55**). O número do porto pode ser escolhido aleatoriamente. O segundo SSID, e respetiva palavra-passe, têm o propósito de criar uma ligação ao AP que permitirá depois a criação de uma ligação *Transmission Control Protocol* (TCP) ao servidor da *ThingSpeak*, onde a chave **WRITE_KEY** permite a atualização dos valores dos sensores na interface fornecida pelo servidor (ver capítulo **4.2.2**

para obtenção desta chave). Para a ligação ao AP, é necessário o envio do comando "AT+CWJAP_DEF" com os parâmetros de ligação. Para tal, é criada uma variável do tipo *string* à qual é feita a concatenação dos parâmetros definidos, mas tendo em conta que caracteres especiais que fazem parte do formato de comando, como o asterisco, necessitam de ser procedidos de "\".

As duas últimas linhas definem que tipo de motor de passo está em uso e os pinos de controlo para a sequência de ativação das bobinas. É importante notar que os pinos centrais não estão por ordem devido a esta biblioteca do motor de passo ter uma sequência de ativação incompatível com o motor utilizado (sequência 1010, 0110, 0101 e 1001). No entanto, ao ter os pinos centrais trocados a sequência torna-se 1100, 0110, 0011 e 1001, correspondente ao modo de operação de *half-stepping*, e o motor de passo passará a funcionar. Esta mudança também podia ter sido efetuada alterando as ligações elétricas em vez da mudança em *software*.

O envio e receção de informação pelo módulo *WiFi* ESP8266 é controlada pela função **RemoteXY_Handler()**, fornecido aquando da extração do código da interface construída (capítulo **4.2.1**). No entanto, o código que controla quais as ações das variáveis recebidas, necessita de ser construída. Com o intuito de apresentar de que forma esse controlo é feito, são apresentados a seguir dois extratos desse código que permite o normal funcionamento do módulo local, com a preparação das variáveis para envio e o seu controlo remoto.

```
 RemoteXY_Handler ();
if(menuLocalManual==1) {
  if(RemoteXY.ControloM_remot==1)  {
    ctrlManual=1;
    menuLocalManual=0;
    MenuFlag=0;
    janela=RemoteXY.Janela_remota;
    esteira=RemoteXY.esteira_remota;
    R=RemoteXY.rega_remota;
  }
}
else {
  if(RemoteXY.ControloM_remot==1) {
    ctrlManual=1;
    janela=RemoteXY.Janela_remota;
    esteira=RemoteXY.esteira_remota;
    R=RemoteXY.rega_remota;
  }
  else {ctrlManual=0;}
}
```

Como é observado, a primeira condição verifica se o utilizador está no menu de controlo manual do módulo local através da variável **menuLocalManual** e caso seja afirmativo em conjunto com a segunda condição, onde o utilizador ativa o controlo manual remoto na aplicação RemoteXY, então existe a passagem dos valores recebidos para as vaiáveis locais que controlam os atuadores (janela, esteira e rega), ativação do controlo manual do módulo local mas de forma remota e a remoção do utilizador dos submenus de "Controlo Manual" do módulo local para o evitar de conflitos, dando controlo total ao utilizador remoto (**MenuFlag**=0). A secção que engloba o código dentro do primeiro *else* toma exatamente as mesmas ações referidas anteriormente, com a exceção da remoção do utilizador dos submenus de "Controlo Manual" por o utilizador não se encontrar nele.

O segundo extrato, relativo ao subsistema do módulo remoto RemoteXY, apresentado a seguir, retirado da função **ler_sensores()** e que ocorre de 2 em 2 segundos, mostra nas duas primeiras linhas os valores de leitura da humidade e temperatura (**hum** e **temp**) a serem guardados em variáveis que estão numa estrutura para posterior transmissão para o módulo remoto RemoteXY (correspondente à visualização gráfica dos valores). As duas últimas linhas convertem os valores de leitura através da função **dtostrf()** em *strings* (ou vetor de *bytes*) e guardam essa informação nas variáveis da estrutura **RemoteXY** que, quando transmitido, é apresentado na interface gráfica do *Smartphone* numa caixa de diálogo os valores exatos que correspondem também ao nível do gráfico apresentado (capítulo **4.2.1** na **Figura 60**). Os valores centrais nesta função são o número de algarismos total possível da variável correspondente (3 ou 2) e o número de algarismos para além do ponto decimal (0).

```
  RemoteXY.Humidade=hum;
  RemoteXY.Temperatura=temp;
 dtostrf(hum, 3, 0, RemoteXY.valorH);
 dtostrf(temp, 2, 0, RemoteXY.valorT);
```

Relativo ao subsistema do módulo remoto com ligação a *ThingSpeak*, que permite comprovar a viabilidade da solução projetada para múltiplas ligações e a um servidor remoto, o extrato seguinte permite visualizar como é feita a conexão e publicação dos valores dos sensores na plataforma *ThingSpeak* através do módulo *WiFi* ESP8266.

```
void mensagemSensores (void){
  String tmp_comando = "AT+CIPSTART=1,\"";
 tmp_comando +="TCP\"";
 tmp_comando +=",\"";
 tmp_comando += "184.106.153.149";
 tmp_comando += "\",80";
 Serial.println(tmp_comando);
 delay(200);
```

```
  String tmp_msg = "GET /update?api_key=";
  tmp_msg += WRITE_KEY;
  tmp_msg += "&field1=";
  tmp_msg += String(temp);
  tmp_msg += "&field2=";
  tmp_msg += String(hum);
  tmp_msg += "&field3=";
  tmp_msg += String(valorLDR);
  tmp_msg += "&field4=";
  tmp_msg += String(valorHumSolo);
  tmp_msg += "\r\n\r\n";
  String tmp_comando = "AT+CIPSEND=1,";
  tmp_comando += String(tmp_msg.length());
  Serial.println(tmp_comando);
  if(Serial.find(">"))  {
     Serial.print(tmp_msg);
  }
  else  {
    Serial.print("AT+CIPCLOSE=1\r\n");  }
}
```

No extrato anterior pode ser observado que primeiro é estabelecido uma ligação TCP ao servidor, com a identificação (ID) 1 e com o porto 80, através do comando "AT+CIPSTART". Com a ligação estabelecida é construída a *string* que conterá a informação dos sensores e da chave do canal criado na interface *web* que será atualizado. Uma vez construída a mensagem com os parâmetros devidos, é transmitido o comando "AT+CIPSEND" que informa o módulo ESP8266 de que a mensagem será para a ligação de ID=1 e com um tamanho especificado. Se o módulo *WiFi* transmitir de volta o caracter ">" significa que está pronto para transmitir a mensagem e ela é transmitida, caso contrário a ligação é fechada e apenas tentada novamente quando o programa voltar a entrar nesta função.

Por último, é necessário colocar algum contexto no funcionamento do menu em termos de *software*. Para tal, o seguinte extrato, encurtado em relação à função real, demonstra uma pequena parte desse menu para a explicação dos princípios básicos com que este se rege e ajudando à sua compreensão.

O menu desenvolvido funciona por níveis, de 0 a 2, que permitem selecionar hierarquicamente o menu principal e os seus submenus e assim imprimir corretamente o texto definido no LCD. A transição do menu principal ou superior para um submenu é feita através dos botões "OK" e "ESC", onde o primeiro botão desce na hierarquia do menu, enquanto que o último sobe nessa hierarquia (ambos modificando **MenuFlag**). A função destes botões também pode ser modificada ou estendida conforme a necessidade do menu

em que o utilizador se encontre. Por exemplo, como o código apresentado é referente ao nível 0, não existe um menu superior a este, e por isso a existência de código referente ao botão "ESC" é desnecessária. Noutra situação, como no nível mais baixo do menu, código do botão "OK" pode ter a função de selecionar individualmente dígitos que são imprimidos no LCD ou então, pode até não existir. Por seu turno, os botões definidos como "cima" e "baixo" permitem percorrer as páginas do menu desse nível, como é neste caso. Em outras situações, estes botões permitem aumentar ou diminuir valores definidos pelo utilizador.

```
if(MenuFlag==0)   //nivel 0 do menu.
{
if((mainMenuPag!=antmainMenuPag)||(MenuFlag!=antM
enuFlag)||(R!=antMenuR))
    {
      lcd.clear();
      lcd.setCursor(0,0);
      antmainMenuPag=mainMenuPag;
      antMenuFlag=MenuFlag;
      antMenuR=R;
      switch (mainMenuPag) {
        case 1: lcd.print("1.DefinirRelogio");
                lcd.setCursor(3,1);
                lcd.print("Rega: ");
                if(R==1){ lcd.print("ON ");}
                else
                {lcd.print("OFF");}
          break;
        case 2: lcd.print("2.Definir niveis");
                lcd.setCursor(2,1);
                lcd.print("dos sensores");
          break; }
    }
    if(SbotMenu!=botMenu)
    {botMenu=SbotMenu;
      if(botMenu== 4)     //OK
      {MenuFlag=1; }
      else
      {
        if(botMenu == 2)    //cima
        {mainMenuPag++;
          if(mainMenuPag > mainMenuTotal)
          {mainMenuPag = 1; }
        }
        else{
          if(botMenu == 1)   //baixo
          {mainMenuPag--;
            if(mainMenuPag == 0) {
             mainMenuPag = mainMenuTotal; }
          }
        }
      }
    }
  }
```

O texto impresso no LCD é apenas impresso uma vez quando existe mudança de nível do menu, página ou valores para evitar distorção. Neste caso, se a rega mudar de estado o texto volta a ser impresso. Um último detalhe relevante no funcionamento do menu, está no facto de que o valor da página é utilizado para selecionar o submenu correto. Por exemplo, se neste menu o utilizador estiver na página 2 e selecionar o botão "OK", o valor guardado em **mainMenuPag** é utilizado para selecionar o submenu 2 quando passar para o nível 1 e assim sucessivamente.

4.2. MÓDULO REMOTO

Como referido, o módulo remoto é constituído por dois subsistemas: da aplicação RemoteXY e do servidor *ThingSpeak*. O módulo de implementação destes dois subsistemas será explicado em detalhe a seguir.

4.2.1. REMOTEXY

Para a elaboração do módulo remoto RemoteXY, foi necessário em primeiro lugar aceder à página *web* http://remotexy.com/en/editor/, onde foi desenvolvida uma interface gráfica para a visualização dos dados transmitidos pelo módulo local e controlo remoto dos atuadores (**Figura 60**).

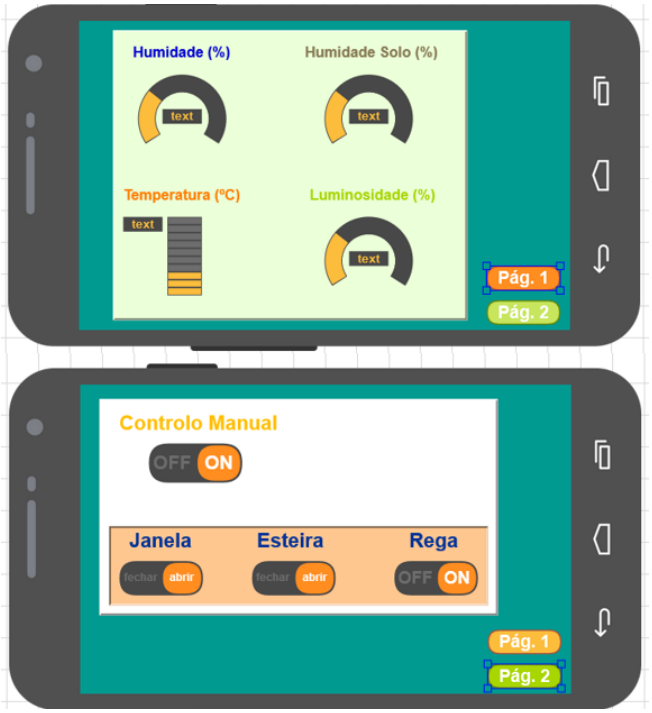


Figura 60 – Grafismo desenvolvido para o módulo remoto RemoteXY.

Esta interface foi elaborada a partir de vários elementos retirados a partir do editor, com cada elemento precisando de ser configurado individualmente, como um elemento aparecer numa página especifica, nome, tamanho e nome de variáveis que são supostos controlar ou ser visualizados. Para a interface ter os dois modos de visualização num *Smartphone* (em pé e tombado) é necessário também criar e dimensionar o segundo modo (**Figura 61**).

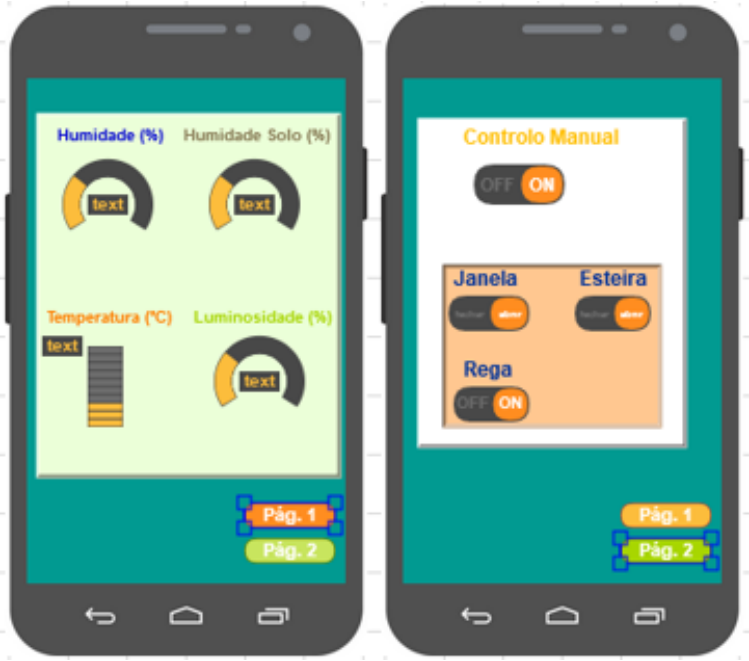


Figura 61 – Interface gráfica do modo "em pé".

Com a interface criada e as variáveis definidas, é necessário configurar a ligação com o ESP8266 com os parâmetros anotados aquando a configuração do módulo no capítulo **4.1.1**. A próxima figura mostra os parâmetros utilizados.

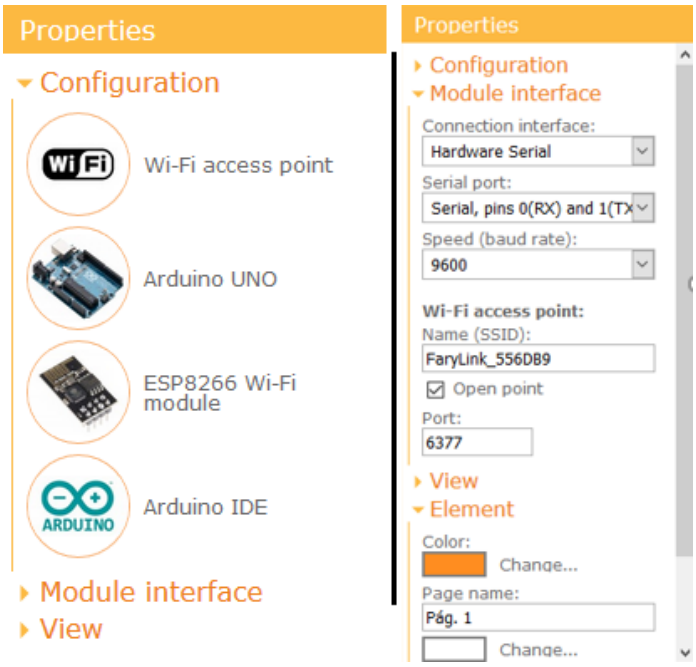


Figura 62 – Parâmetros de configuração da ligação ao módulo ESP8266.

A seguir, basta clicar no botão "*Get source code*" para gerar o código relativo à interface gráfica desenvolvida e incluí-lo no módulo local, configurando as variáveis consoante a sua função.

Este próximo extrato de código, encurtado por razões de espaço neste relatório, corresponde ao grafismo construído para o módulo remoto RemoteXY, que deve ser colocado no *sketch* desenvolvido para o Arduíno do módulo local. Quando o módulo local estiver conectado ao à aplicação RemoteXY, este é transmitido na ligação.

```
uint8_t RemoteXY_CONF[] =
  { 255,4,0,19,0,231,1,8,13,6,
    130,1,4,3,81,44,3,8,53,65,
    2,31,130,2,6,29,77,16,8,37,
  .
  .
  .
  1,2,26,4 };
```

Por último, deve ser feito o *download* e instalação da aplicação "RemoteXY Free" pelo *Smartphone* através da *Play Store* da Google. Com a aplicação iniciada, clicar em *Add new device>Wi-Fi point* e a seguir colocar o nome do SSID e o número do porto (ambos anotados anteriormente e visto na **Figura 62**) e clicar na ligação para estabelecer uma conexão.

4.2.2. THINGSPEAK

Para a implementação da ligação à plataforma disponibilizada pela *ThingSpeak* (e depois da criação de uma conta nesta plataforma), deve ser acedido a página https://thingspeak.com/channels para a criação de um canal que receberá a informação transmitida pelo módulo local e apresenta-la num formato gráfico. A informação relativa à criação do canal pode ser observada na **Figura 63**.

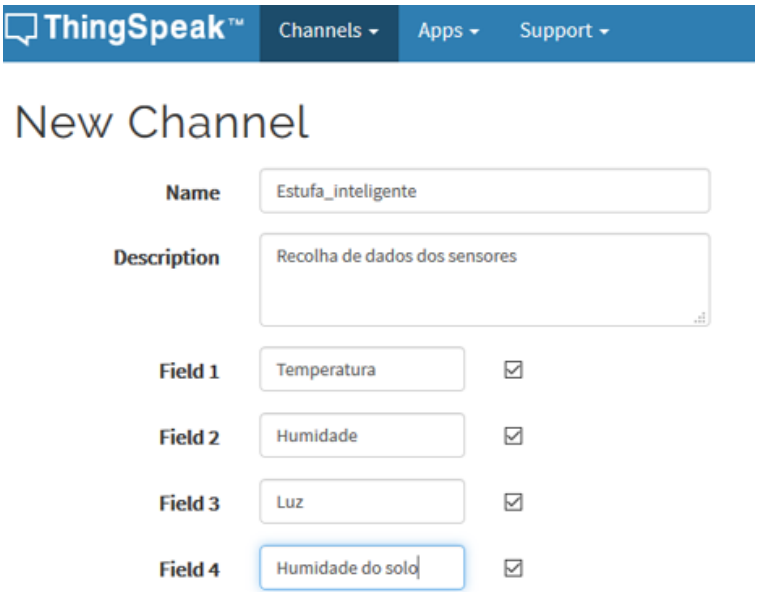


Figura 63 – Criação do canal que receberá os dados transmitidos.

Com o canal criado, o seguinte passo deve ser o acesso a *Channels>My Channels>API Keys* onde são disponibilizadas as chaves de acesso para o envio de informação para este canal específico e a forma de requerer a escrita de dados na plataforma *ThingSpeak* (**Figura 64**).

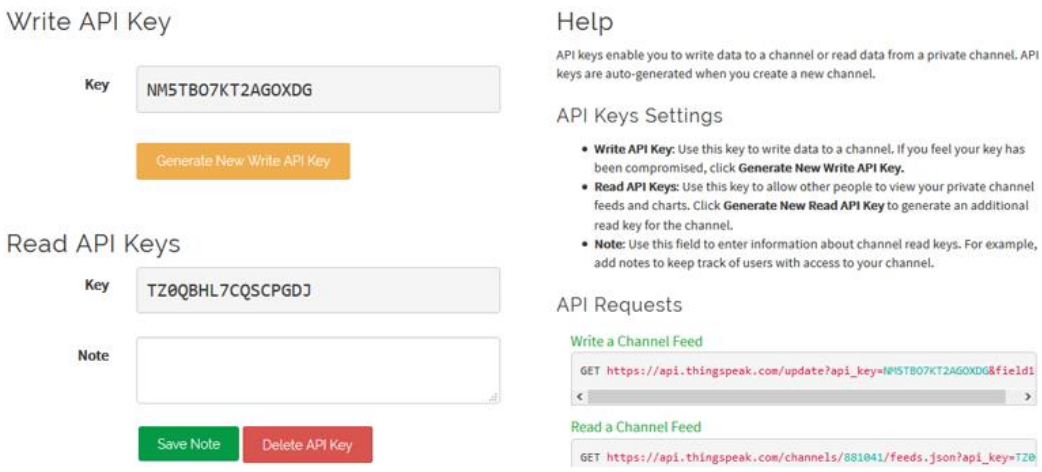


Figura 64 – Chaves de acesso (esquerda) e a forma de requerimento de escrita (direita).

Estes dois elementos são os mesmos que foram referidos e observados nos extratos do capítulo **4.1.4**, essenciais para a elaboração da ligação entre o módulo local e a plataforma remota da *ThingSpeak*. Assim, quando a criação do código de ligação no módulo local for completa, basta aceder ao separador *Private View* para observar as janelas com a informação de cada um dos sensores.

5. RESULTADOS

Neste capítulo encontra-se de forma sintetizada a demonstração de resultados durante o funcionamento do protótipo. É exposta a capacidade de mostrar os valores de leitura dos sensores no LCD e ao mesmo tempo na aplicação do *Smartphone*. Também é demonstrada a capacidade de configuração do RTC a partir dos botões de pressão, a configuração inicial de ligação de ESP8266 e a aparência da interface disponibilizada na plataforma *ThingSpeak* quando a informação dos sensores do módulo local é transmitida.

5.1. INICIALIZAÇÃO DE ESP8266

A **Figura 65** corresponde ao envio de comandos AT (em parte pela função **RemoteXY_Init()**) para configurar os parâmetros de ligação do módulo ESP8266. A execução destes comandos pode ser observada no monitor série do ambiente de desenvolvimento do Arduíno aquando do arranque do programa.

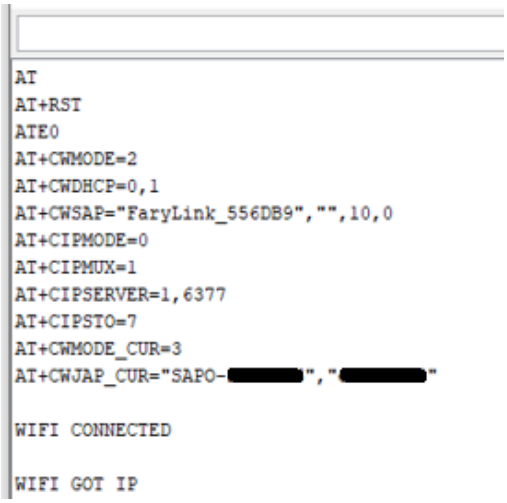


Figura 65 – Configuração inicial de ESP8266 aquando o arranque do programa.

Alguns destes comandos não foram descritos anteriormente (ver capítulo **4.1.1**), como tal será feita uma breve descrição da função de cada um dos comandos cuja função não foi expressa anteriormente.

- **ATE0:** desativa o eco de resposta dos comandos AT;
- **AT+CWMODE = 2:** configura o módulo ESP8266 no modo de ponto de acesso/AP;
- **AT+CWDHCP = 0, 1:** configura ESP8266 como AP e ativa DHCP;
- **AT+CIPMODE = 0:** configura o modo de transmissão como normal, o que significa que se a transmissão falhar, ESP8266 não tentará reconectar;
- **AT+CIPMUX = 1:** ativa a utilização de múltiplas conexões;
- **AT+CIPSERVER = 1, 6377:** cria um servidor TCP com o porto 6377;
- **AT+CIPSTO = 7:** define o tempo de *timeout*, em que desliga a conexão caso o cliente TCP não comunique dentro do tempo definido (7 segundos).
- **AT+CWMODE_CUR = 3:** configura o módulo para agir como AP e estação, mas não grava a configuração em *flash*, logo não é permanente.
- **AT+CWJAP_CUR = ”SSID”, ”Palavra-passe”:** conecta ao ponto de acesso com o SSID e palavra-passe especificado, mas não grava a configuração em *flash*.

5.2. DEMONSTRAÇÃO DE MENUS

A seguir, são demonstrados alguns dos menus imprimidos no LCD aquando do funcionamento do módulo local, de qual o seu aspeto quando o utilizador as opera, como na **Figura 66**, onde pode ser verificado o aspeto da página de configuração das horas e data do RTC.

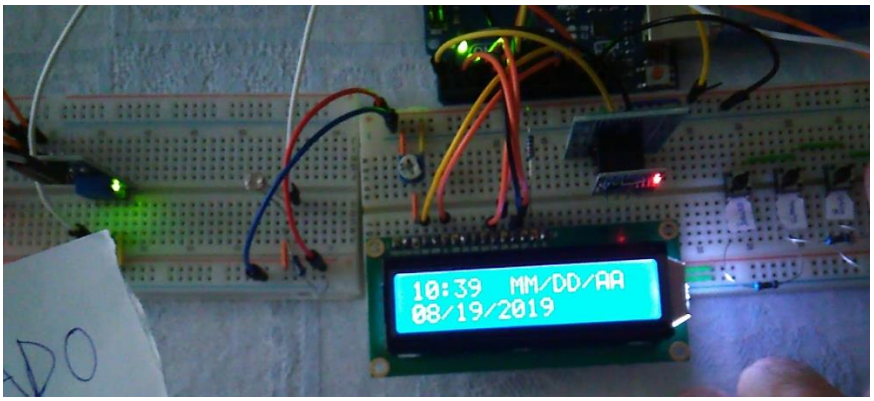


Figura 66 – Página de configuração do RTC.

A página de leitura dos sensores, onde é apresentado os valores de todos os 4 sensores atualizados de 2 em 2 segundos é observável na figura seguinte.

Figura 67 – Página de observação dos valores dos sensores.

A **Figura 68** demonstra simultaneamente os valores de leitura dos sensores no LCD do módulo local e na interface gráfica desenvolvida para o *Smartphone*, onde para esta última os valores são demonstrados quer em modo gráfico (semicírculos e barra graduada) quer numa caixa com o valor numérico.

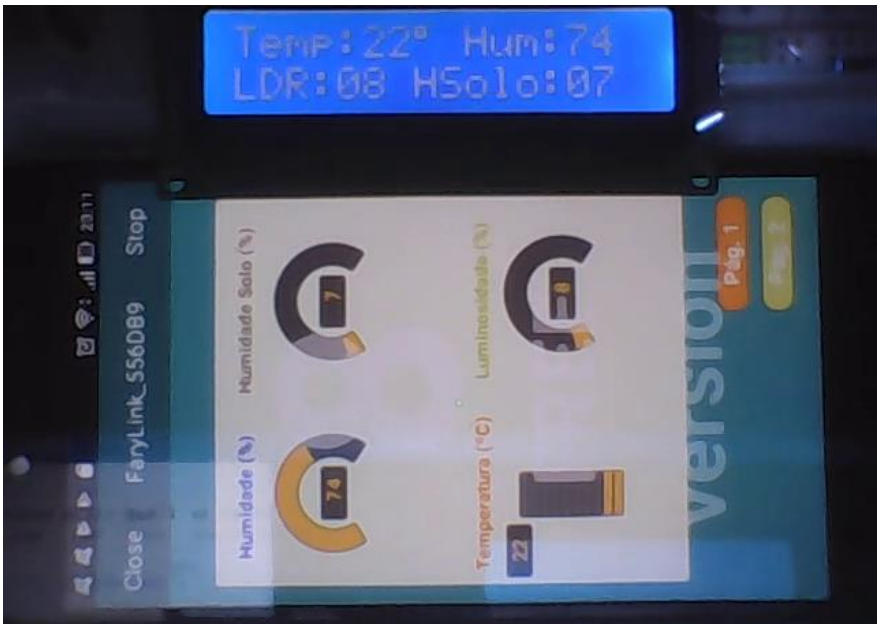


Figura 68 – Valores lidos pelos sensores no LCD e no menu do *Smartphone*.

5.3. PLATAFORMA THINGSPEAK E DEMONSTRAÇÃO DO ENVIO NO MONITOR SÉRIE

A seguir, é apresentado o separador da plataforma que contem várias janelas com o grafismo da informação recebida de cada sensor, quando esta é transmitida pelo módulo local.

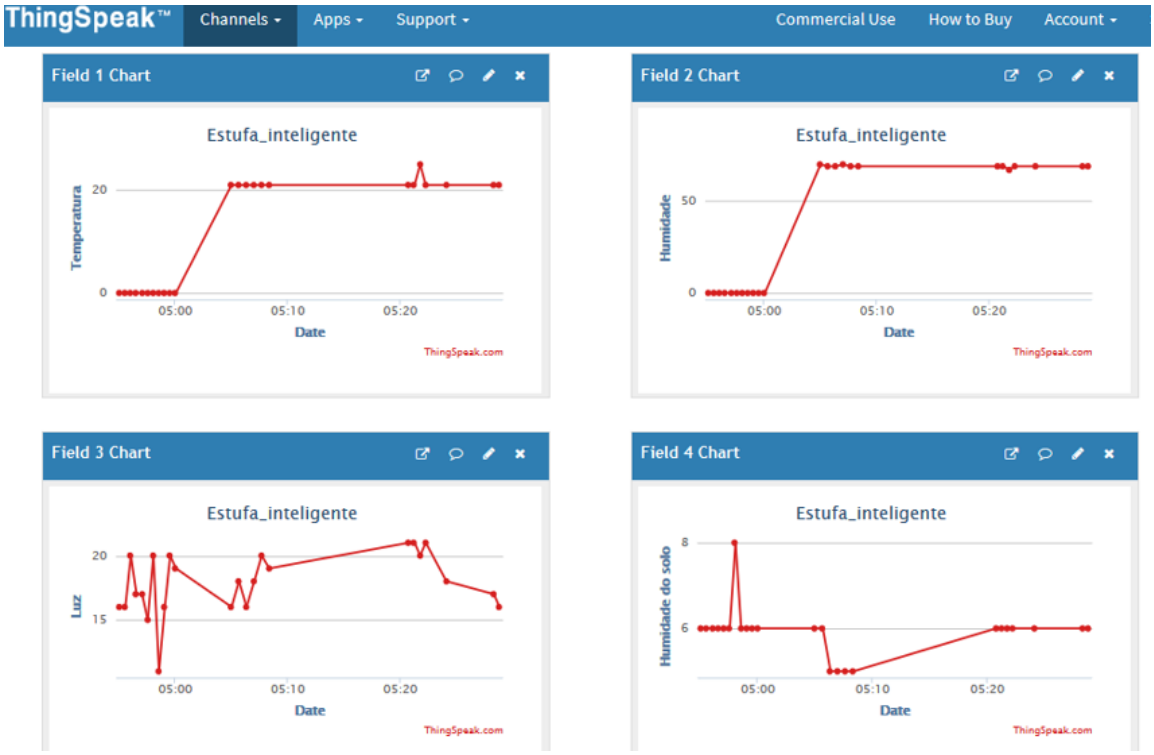


Figura 69 – Dados dos sensores transmitidos pelo módulo local.

Por último, é apresentada a **Figura 70** onde é possível verificar, através do monitor série, os comandos que estão a ser executados durante a transmissão da informação para o módulo RemoteXY e *ThingSpeak*, demonstrando que são executados concorrentemente. Neste, também é possível verificar que o envio é feito para ligações com ID diferentes (0 para RemoteXY e 1 para a plataforma *ThingSpeak*).

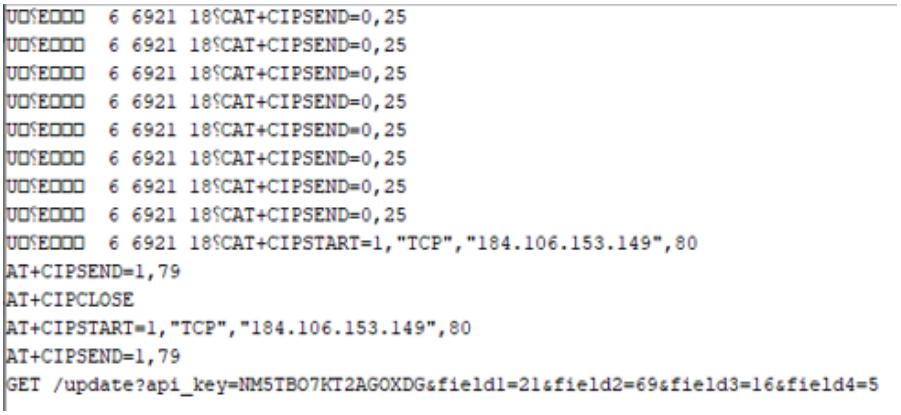


Figura 70 – Monitor série aquando a transmissão de dados.

6. CONCLUSÕES

O objetivo inicial proposto, da construção de um protótipo de uma estufa inteligente capaz de funcionar de modo automático consoante os valores de controlo definidos e de leitura dos sensores, como também do seu controlo de forma manual e remota, foi alcançado. Também ficou demonstrado o desenvolvimento de um menu expansivo no módulo local, capaz de configurar o horário e data do RTC, valores de controlo, controlo manual de atuadores, existência de submenus variável consoante a definição de rega e definição individual de algarismos.

No início, o controlo remoto era para ser mais expansivo e complexo através da elaboração de uma interface gráfica em LabView, mas a falta de tempo e, mais importante, de conhecimento suficiente deste *software* gráfico para a elaboração do módulo remoto, levou ao abandono desta plataforma e a sua substituição por uma plataforma mais simples (o RemoteXY), visto que o objetivo principal era demonstrar o funcionamento remoto do protótipo. Em adição ao aplicativo da RemoteXY, foi elaborada a conexão a uma plataforma *cloud* da *ThingSpeak* para comprovar a viabilidade da solução projetada para múltiplos tipos de conexões, como um servidor remoto e de ligação direta.

Assim, de futuro, uma possibilidade de expansão deste projeto seria o desenvolvimento de uma plataforma gráfica remota mais complexa, com capacidade de gravação dos valores dos

sensores num ficheiro pdf e construção de uma base de dados. Quanto ao módulo local, este pode ser expandido dando diferentes ângulos de abertura à janela (possível pela utilização de um motor de passo no projeto inicial) e o adicionar de um sensor de vento, bastante comum em estufas situadas em zonas ventosas, que fecharia a janela se detetasse uma rajada de vento acima de um valor definido. Outra possibilidade de expansão é fazer com que o módulo local e módulo remoto comuniquem através de uma plataforma *cloud* com capacidade de envio de instruções para o módulo local.

Referências Documentais

[1] REVISTA AGROPECUÁRIA – *Fique por dentro de alguns detalhes sobre a utilização das estufas agrícolas.* [Consult. 10 Abril 2019] Disponível em WWW:<URL:http://www.revistaagropecuaria.com.br/2018/07/04/fique-por-dentro-de-alguns-detalhes-sobre-a-utilizacao-das-estufas-agricolas/>.

[2] PASSINATO, Cristiana – *O que é uma estufa?* [Consult. 10 Abril 2019] Disponível em WWW:<URL:https://pt.slideshare.net/crispassinato/o-que-uma-estufa>.

[3] ROGUECLASSICISM. *Roman Greenhouses?* [Consult. 10 Abril 2019] Disponível em WWW:<URL:http://www.atrium-media.com/rogueclassicism/2004/01/07.html>.

[4] AGÊNCIA FAPESP. *Estufa inteligente melhora produtividade de plantas e frutos*. 30 de Novembro de 2016. [Consult. 10 Abril 2019] Disponível em WWW:<URL:https://revistapegn.globo.com/Banco-de-ideias/noticia/2016/11/estufa-inteligente-melhora-produtividade-de-plantas-e-frutos.html>.

[5] *The History of the Greenhouse*. Cambrige Glasshouse, 2013. [Consult. 12 Abril 2019] Disponível em WWW:<URL:https://web.archive.org/web/20130509082654/http://www.cambridgeglasshouse.co.uk/news/history-of-the-greenhouse>.

[6] HODGSON, Larry. *A Brief History of the Greenhouse*. Laidback Gardener, 2016. [Consult. 12 Abril 2019] Disponível em WWW:<URL:https://laidbackgardener.blog/2016/01/27/a-brief-history-of-the-greenhouse/>.

[7] *The History of Greenhouses*. The Do It Yourself Greenhouse. [Consult. 12 Abril 2019] Disponível em WWW:<URL:http://www.thediygreenhouse.com/the-history-of-greenhouses/>.

[8] *Greenhouse*. Wikidwelling. [Consult. 12 Abril 2019] Disponível em WWW:<URL:https://wikidwelling.fandom.com/wiki/Greenhouse>.

[9] MAD FARMER – *Características e características de estufas industriais*. [Consult. 21 Abril 2019] Disponível em WWW:<URL:https://pt.madlovefarms.com/5483-features-and-characteristics-of-industrial-greenhouses>.

[10] *Lesson 1 History and Types of Greenhouse*. e-Krishi Shiksha, 16 de Dezembro de 2013 [Consult. 22 Abril 2019] Disponível em WWW:<URL:http://ecoursesonline.iasri.res.in/mod/page/view.php?id=1604>.

[11] DECOREXPRO – *Escolhendo um quadro para a estufa*. DecorexPro. [Consult. 30 Abril 2019] Disponível em WWW:<URL:https://pt.decorexpro.com/teplica/karkas/>.

[12] ATUL – *Smart Greenhouse: The future of agriculture*. (2016). [Consult. 10 Maio 2019] Disponível em WWW:<URL:https://www.hackster.io/synergy-flynn-9ffb33/smart-greenhouse-the-future-of-agriculture-5d0e68>.

[13] PERDIGONES, Alicia; BENEDICTO, Susana; LUIS GARCÍA, José – *Automatização de estufas: a evolução das tecnologias*. (2015). [Consult. 10 Maio 2019] Disponível em WWW:<URL:http://www.agronegocios.eu/noticias/automatizacao-de-estufas-a-evolucao-das-tecnologias/>.

[14] O'HANLEY, Elizabeth – *How Smart Greenhouse is Changing Agriculture*. (2018). [Consult. 10 Maio 2019] Disponível em WWW:<URL:https://www.deprolabs.com/blog/smart-greenhouse-changing-agriculture/>.

[15] ECLIPSE IoT WORKING GROUP—*The Three Software Stacks Required for IoT Architectures*. ECLIPSE, Setembro 2016.

[16] VERIFIED MARKET RESEARCH – *Smart Greenhouse Market Size And Forecast*. (Janeiro 2019). [Consult. 10 Maio 2019] Disponível em WWW:<URL:https://www.verifiedmarketresearch.com/product/global-smart-greenhouse-market-size-and-forecast-to-2025/>.

[17] POSTSCAPES – *Smart Greenhouse Remote Monitoring Systems*. [Consult. 15 Maio 2019] Disponível em WWW:<URL:https://www.postscapes.com/smart-greenhouses/>.

[18] MODERN FARMER – *Estufas automatizadas e a alta tecnologia na produção de alface*. (2017). [Consult. 25 Maio 2019] Disponível em WWW:<URL:https://blog.strider.ag/estufas-automatizadas-e-a-alta-tecnologia-na-producao-de-alface/>.

[19] HARVEST AUTOMATION – *Payback: Metrolina Greenhouses realizes a return on their investment in under one year.* 2014.

[20] TAWIL, Yahya – *Understanding Arduino UNO Hardware Design*. 1 de Julho de 2016. [Consult. 30 Agosto 2019] Disponível em WWW:<URL:https://www.allaboutcircuits.com/technical-articles/understanding-arduino-uno-hardware-design/>.

[21] MICROCHIP TECHNOLOGY INC. – *ATmega328/P, AVR® Microcontroller with picoPower® Technology*. Microchip, 2018. DS40001984A.

[22] MAXIM INTEGRATED – *DS1307: 64 x 8, Serial, I2C Real-Time Clock.* Maxim Integrated Products, 2015.

[23] THOMSEN, Adilson – *Relógio com o módulo RTC DS1307*. 11 de Junho de 2014. [Consult. 5 Agosto 2019] Disponível em WWW:<URL:https://www.filipeflop.com/blog/relogio-rtc-ds1307-arduino/>.

[24] *DHT11 Humidity & Temperature Sensor.* D-Robotics, 2010.

[25] LITTELFUSE – *Thermistor Terminology and Technical Vocabulary*. [Consult. 15 Set. 2019] Disponível em WWW:<URL:https://www.littelfuse.com/technical-resources/technical-centers/temperature-sensors/thermistor-info/thermistor-terminology.aspx?utm_source=ussensor.com&utm_medium=redirect&utm_campaign=ussensor-lf>.

[26] *How to Set Up the DHT11 Humidity Sensor on an Arduino.* Circuit Basics. [Consult. 5 Agosto 2019] Disponível em WWW:<URL:http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/>.

[27] ELECTROFUN – *Sensor de Humidade do Solo Terra.* Electrofun, 2019. [Consult. 5 Agosto 2019] Disponível em WWW:<URL:https://www.electrofun.pt/sensores-arduino/sensor-humidade-solo>.

[28] JOJO – *Arduino and Soil Moisture Sensor -Interfacing Tutorial*. 31 de Julho de 2018. [Consult. 5 Agosto 2019] Disponível em WWW:<URL:http://www.circuitstoday.com/arduino-soil-moisture-sensor>.

[29] ELECTROFUN – *Sensor de luz LDR 5mm GL5528.* Electrofun, 2019. [Consult. 5 Agosto 2019] Disponível em WWW:<URL:https://www.electrofun.pt/sensores-arduino/sensor-luz-ldr>.

[30] EETECH MEDIA – *Photo Resistor.* EETech Media, 2019. [Consult. 5 Agosto 2019] Disponível em WWW:<URL:http://www.resistorguide.com/photoresistor/>.

[31] *Stepper Motor – Types, Advantages & Applications*. Elprocus, 2019. [Consult. 10 Agosto 2019] Disponível em WWW:<URL:https://www.elprocus.com/stepper-motor-types-advantages-applications/>.

[32] *What is a Stepper Motor? Types, Construction, Operation & Applications*. Electrical Technology, 2016. [Consult. 10 Agosto 2019] Disponível em WWW:<URL:https://www.electricaltechnology.org/2016/12/stepper-motor-construction-types-and-modes-of-operation.html>.

[33] *Stepper Motor 5V 4-Phase 5-Wire & ULN2003, Driver Board for Arduino*. 2016 [Consult. 10 Agosto 2019] Disponível em WWW:<URL:http://eeshop.unl.edu/pdf/Stepper+Driver.pdf>.

[34] KIATRONICS – *28BYJ-48 – 5V Stepper Motor*. Datasheet.

[35] MISAN2010 - *File:Drive.png*. 2009. [Consult. 10 Agosto 2019] Disponível em WWW:<URL:https://commons.wikimedia.org/wiki/File:Drive.png>.

[36] ELECTROFUN – *Driver ULN2003 para Stepper Motor DC*. [Consult. 10 Agosto 2019] Disponível em WWW:<URL:https://www.electrofun.pt/motores-dc/driver-uln2003>.

[37] ELECTROFUN – *Motor DC 12V DC 180mA 11500rpm*. [Consult. 12 Agosto 2019] Disponível em WWW:<URL:https://www.electrofun.pt/motores-e-bobinas/motor-dc-12v-dc-180ma-11500rpm>.

[38] GUPTA, Avinash – *Controlling DC Motors*. Extreme Electronics, 2008.

[39] ELECTROFUN – *Display LCD 16x2 com fundo azul*. [Consult. 12 Agosto 2019] Disponível em WWW:<URL:https://www.electrofun.pt/display/display-lcd-16x2>.

[40] ELECTROFUN – *Botão de Pressão 6x6x4mm*. [Consult. 12 Agosto 2019] Disponível em WWW:<URL:https://www.electrofun.pt/botoes-e-teclados/botao-de-pressao-6x6x4mm>.

[41] ELECTROFUN – *Adaptador para Módulo WiFi ESP8266 ESP-01*. [Consult. 12 Agosto 2019] Disponível em WWW:<URL:https://www.electrofun.pt/comunicacao/adaptador-wifi-esp8266>.

[42] ELECTROFUN – *Módulo WiFi ESP8266 ESP-01 Wireless*. [Consult. 12 Agosto 2019] Disponível em WWW:<URL:https://www.electrofun.pt/comunicacao/modulo-wifi-esp8266>.

[43] ESPRESSIF SYSTEMS – *ESP8266 AT Instruction Set*. Espressif Systems, 2019.

[44] REMOTEXY – *ESP8266 WiFi module*. [Consult. 20 Setembro 2019] Disponível em WWW:<URL:http://remotexy.com/en/help/esp8266/>.

[45] REMOTEXY – *How it works*. [Consult. 20 Setembro 2019] Disponível em WWW:<URL:http://remotexy.com/en/help/>.

[46] ATLAS OBSCURA – *Botanical Garden of Padua - Orto botanico di Padova*. Atlas Obscura, 2019. [Consult. 12 Abril 2019] Disponível em WWW:URL:https://www.atlasobscura.com/places/botanical-garden-of-padua-orto-botanico-di-padova>.

[47] THINGSPEAK – ThingSpeak for IoT Projects. [Consult. 30 Setembro 2019] Disponível em WWW:<URL:https://thingspeak.com/>.

Anexo A. Código do módulo local

Neste anexo é apresentado o código desenvolvido para o módulo local, cujo nome do ficheiro é “Modulo_local1.ino”.

```
#define REMOTEXY_MODE__ESP8266_HARDSERIAL_POINT    //modo de ligação aos portos dedicados Rx e Tx.
#include <RemoteXY.h>
// configuração da ligação RemoteXY
#define REMOTEXY_SERIAL Serial        //para ligação Rx->0 e Tx->1
#define REMOTEXY_SERIAL_SPEED 9600    //BAUD rate da ligação.
#define REMOTEXY_WIFI_SSID "FaryLink_556DB9"   //nome do SSID da placa wifi
#define REMOTEXY_WIFI_PASSWORD ""       //sem password.
#define REMOTEXY_SERVER_PORT 6377       //porto.


// RemoteXY interface gráfica
#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] =
  { 255,4,0,19,0,231,1,8,13,6,
  130,1,4,3,81,44,3,8,53,65,
  2,31,130,2,6,29,77,16,8,37,
  43,32,2,17,130,1,7,2,72,59,
  2,7,60,72,1,125,66,130,12,11,
  18,14,7,16,18,14,1,2,26,66,
  0,18,40,7,16,12,53,7,16,1,
  2,26,66,130,50,40,18,14,37,51,
  18,14,1,2,26,66,130,50,11,18,
  14,36,16,18,14,1,2,26,2,1,
  14,12,19,8,14,17,18,8,2,2,
  26,31,31,79,78,0,79,70,70,0,
  131,1,83,50,15,5,43,84,18,5,
  1,2,31,80,195,161,103,46,32,49,
  0,131,0,83,57,15,5,43,91,18,
  5,2,106,31,80,195,161,103,46,32,
  50,0,129,0,9,34,24,3,4,46,
  25,3,1,64,84,101,109,112,101,114,
  97,116,117,114,97,32,40,194,186,67,
  41,0,129,0,11,5,20,3,6,10,
  21,3,1,203,72,117,109,105,100,97,
  100,101,32,40,37,41,0,129,0,46,
  5,27,3,31,10,29,3,1,87,72,
  117,109,105,100,97,100,101,32,83,111,
  108,111,32,40,37,41,0,129,0,47,
  34,26,3,33,46,27,3,1,106,76,
  117,109,105,110,111,115,105,100,97,100,
  101,32,40,37,41,0,129,0,8,6,
  32,4,12,10,31,4,2,78,67,111,
  110,116,114,111,108,111,32,77,97,110,
  117,97,108,0,2,1,8,36,17,7,
  9,43,16,7,2,2,26,31,31,97,
  98,114,105,114,0,102,101,99,104,97,
  114,0,67,5,55,18,8,3,41,23,
  8,4,1,2,26,4,2,1,35,36,
```

```
  17,7,34,43,15,7,2,2,26,31,
  31,97,98,114,105,114,0,102,101,99,
  104,97,114,0,2,1,64,36,17,7,
  10,61,15,7,2,2,26,31,31,79,
  78,0,79,70,70,0,129,0,10,30,
  13,4,10,38,13,4,2,188,74,97,
  110,101,108,97,0,129,0,36,30,14,
  4,35,38,13,4,2,188,69,115,116,
  101,105,114,97,0,129,0,67,30,11,
  4,12,56,10,4,2,188,82,101,103,
  97,0,67,5,17,18,8,3,12,23,
  8,4,1,2,26,4,67,5,9,40,
  8,3,3,51,8,4,1,2,26,3,
  67,5,55,47,8,3,42,58,8,4,
  1,2,26,4 };

//estrutura das variáveis da interface gráfica de controlo
struct {
  //variáveis de entrada
  uint8_t ControloM_remot; // =1 se ON e =0 se OFF
  uint8_t Janela_remota;
  uint8_t esteira_remota;
  uint8_t rega_remota;

    //variavéis para ver na interface
  int8_t Humidade;
  int8_t Temperatura;
  int8_t Luminosidade; // =0..100
  int8_t Humidade_Solo;
  char valorHS[4];  // string UTF8 final zero.
  char valorH[4];
  char valorT[3];
  char valorL[4];

    //flag de ligação.
  uint8_t connect_flag;  // =1 se ligado, caso contrário =0

} RemoteXY;
#pragma pack(pop)

#define WIFI2_SSID "SAPO-1234"   //nome do SSID do ponto de acesso (modem).
#define WIFI2_PASSWORD "palavra-passe"     //password do ponto de acesso
(excluído do código por óbvias razões).
String WRITE_KEY = "NM5TBO7KT2AGOXDG";     //chave de escrita no servidor
ThingsSpeak.

#include <LiquidCrystal.h>      //biblioteca do lcd.
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);    //inicialização da libraria com
os números dos pinos de interface.

#include <Adafruit_Sensor.h>
#include <DHT.h>
#define DHTPIN 6            //pino de dados do DHT11.
#define DHTTYPE DHT11       //tipo de DHT (DHT11 ou DHT22).
DHT dht(DHTPIN, DHTTYPE);   //definição dos parâmetros.

#include <DS1307.h>   //Carrega a biblioteca do RTC DS1307.
DS1307 rtc(A4, A5);    //Modulo RTC DS1307 ligado às portas A4 e A5 do
Arduino.

Time t;       //inicialização de uma estrutura de dados-tempo.
```

```
#include <Stepper.h>
const byte stepsPerRevolution=64;      //nº de passos para o rotor interior
completar uma volta.
Stepper motorPasso(stepsPerRevolution, 7, 9, 8, 10);     //pinos 9 e 8
cruzados para permitir a rotação em diferentes direções sem ser necessário
alterar a biblioteca.
//4096 full-stepping ou 2048 em half-stepping para o eixo exterior
completar uma volta.
const int CntPassos = 348;          //348 passos equivale a 67,5º.

////////////////////////////////////////////////////////////
////////////////////  Intervalos de leitura  ///////////////
////////////////////////////////////////////////////////////
const byte intervaloBotao = 100;     //intervalo de leitura do botao para
evitar leituras erradas por causa de mudanças bruscas.
unsigned long antMillisBotao = 0;   //última leitura feita.

const long intervalo = 2000;      //intervalo escolhido (2 segundos) para
fazer a leitura dos sensores (em milisegundos).
unsigned long antMillis = 0;     //última vez que a leitura foi actualizada.

const long intervaloThing = 15000;        //intervalo de envio para a
thingspeak (15 segundos).
unsigned long antMillisThing = 0;

const long intervaloRegaDia = 120000;     //intervalo para parar a rega na
função diária sem sensor (2 minutos).
unsigned long antRegaDia = 0;           //última leitura feita para forma
diária.
unsigned long antRegaDia2 = 0;           //última leitura feita para forma
bidiária.
////////////////////////////////////////////////////////////


////////////////////////////////////////////////////////////
////////////////////////// Sensores ////////////////////////
////////////////////////////////////////////////////////////
const byte LDRpin = A0;      //Pino analógico do LDR.
byte valorLDR = 0;           //Valor inteiro do LDR em percentagem.

const byte HumSolopin = A1;     //Pino analogico do sensor de humidade do
solo.
byte valorHumSolo = 0;           //Valor inteiro do sensor do solo em
percentagem.
////////////////////////////////////////////////////////////


////////////////////////////////////////////////////////////
////////////////////////// Motor DC ////////////////////////
////////////////////////////////////////////////////////////
const int intervaloMotor=500;    //intervalo de funcionamento do motor.
unsigned long antMotor=0;         //última leitura do valor de tempo.
const byte MotorPinA = A3;    //Pinos de controlo do motor DC.
const byte MotorPinB = 13;
////////////////////////////////////////////////////////////


////////////////////////////////////////////////////////////
//////////  Variáveis de controlo e leitura do menu  ///////
////////////////////////////////////////////////////////////
```

```
const byte botMenuPin = A2;       //Pino analógico dos botões de menu.
byte botMenu=0, SbotMenu=0;       //N° do botão de menu e valor anterior,
respectivamente.

const byte mainMenuTotal = 5;
byte MenuFlag=0, mainMenuPag=1,subMenu2Pag=1,subMenu3Pag=1,subMenu4Pag=1,
antMenuFlag=0;
byte      antmainMenuPag=0,      antsubMenu2Pag=0,      antsubMenu3Pag=0,
antsubMenu4Pag=0; //Estado anterior das páginas display.
byte    CursorMainMenu=0,    CursorMainMenuLinha=0,    CursorSubmenu2=0,
CursorSub2menu2=0,          CursorSub3menu2=0,          CursorSub2menu4=0,
CursorSub3menu4=0;  //Variaveis da localizaçao do cursor.
////////////////////////////////////////////////////////////


////////////////////////////////////////////////////////////
/////////////////// Flags e valores do utilizador //////////////
////////////////////////////////////////////////////////////
char  R  =  0,  antMenuR=0,  antR=0,  janela=0,  antJanela=0,  esteira=0,
antEsteira=0,   ctrlManual=0,   biDia=0,   MudaSet=0,   mudaHorario1=0,
mudaHorario2=0, flagTempo=0, func_motor=0, menuLocalManual=0;     //Flags
dos atuadores (Rega, janela e esteira) e de controlo.
byte   tempMax=30,   tempMin=10,   LDRmax=60,   LDRmin=5,   HumSoloMax=70,
HumSoloMin=30;  //Valores máximos e minimos dos sensores.
byte formaRega=1, antformaRega=1;           //Forma de ativaçao da rega.
byte hum=0, temp=0;
////////////////////////////////////////////////////////////


////////////////////////////////////////////////////////////
////////////////////////////  RTC   ////////////////////////////
////////////////////////////////////////////////////////////
int anoRTC=2019, antanoRTC=2019;    //ano retirado do RTC e estado anterior
para comparaçao.
byte  horaRTC=20,  minutoRTC=0,  diaRTC=1,  mesRTC=9,  hSet1=2,  hSet2=0,
minSet1=0,  minSet2=0,  dSet1=0,  dSet2=1,  mSet1=0,  mSet2=9,  aSet1=1,
aSet2=9; //Setup das horas e data.
byte    anthoraRTC=10,    antminutoRTC=0,    antdiaRTC=1,    antmesRTC=9;
//Estado anterior do setup das horas e data.
byte  anoTemp=19,  hora=0,  minuto=0,  hora2=10,  minuto2=0,  horarioH1=0,
horarioH2=0,   horarioM1=0,   horarioM2=0,   horario2H1=1,   horario2H2=0,
horario2M1=0, horario2M2=0;   //Horas para rega.
////////////////////////////////////////////////////////////

void setup()
{
  Serial.begin(9600);     //comunicação série a 9600 bits por segundo.
  RemoteXY_Init ();
  sendData("AT+CWMODE_CUR=3\r\n", 1000);       //Define ESP8266 como AP e
estação.
  String tmp_comando = "AT+CWJAP_CUR=\"";
  tmp_comando += WIFI2_SSID;
  tmp_comando += "\",\"";
  tmp_comando +=WIFI2_PASSWORD;
  tmp_comando +="\"";
  tmp_comando +="\r\n";
  Serial.print(tmp_comando);
  delay(1000);

  rtc.halt(false);        //Aciona o relogio.
```

```
  //As 3 linhas abaixo determinam a data e hora do RTC e são comentadas
apos a primeira utilizacao.
  rtc.setDOW(FRIDAY);      //Define o dia da semana.
  rtc.setTime(20, 37, 0);     //Define as horas.
  rtc.setDate(19, 8, 2019);   //Define o dia, mes e ano

  motorPasso.setSpeed(100);      //100 rotações por minuto.

  pinMode(MotorPinA, OUTPUT);     //pinos de controlo do motor DC definidos
como saída.
  pinMode(MotorPinB, OUTPUT);

  pinMode(LDRpin, INPUT);           //Pino do LDR como entrada, por ser
utilizado apenas como leitura.
  pinMode(HumSolopin, INPUT);
  pinMode(botMenuPin, INPUT_PULLUP);     //pino de entrada analógico dos
botões de menu.

  lcd.begin(16, 2);   // set up the LCD's number of columns and rows.
  lcd.print("Bem vindo");     //imprime a mensagem no lcd.
  delay(1500);
  lcd.clear();
  dht.begin();
}

String sendData(String comando, const int timeout)
{
  // Envio dos comandos AT para o modulo
  String resposta = "";
  Serial.print(comando);
  unsigned long int tempoAtual = millis();
  while ((millis() - tempoAtual) <= timeout)
  {
    if (Serial.available()>0) //O ESP8266 tem dados para mostrar.
    {
      char d = Serial.read(); //ler caracter a caracter.
      resposta += d;        //conectação em string.
    }
  }
  return resposta;
}

void ler_sensores()
{
  valorLDR = ((float)analogRead(LDRpin))*100/1023;     //valor inteiro do
LDR convertido em percentagem.
  valorHumSolo     =     (1023-((float)analogRead(HumSolopin)))*100/1023;
//calculo do valor em percentagem da humidade no solo (1 -> seco, 100->
molhado).

  hum = dht.readHumidity();       //ler humidade no ar em percentagem.
  temp = dht.readTemperature();   //ler temperatura em graus celsius.
  if (isnan(hum) || isnan(temp))        //verificação da leitura do sensor
dht11.
  {
    Serial.println(F("Falha na leitura do sensor DHT11"));
    return;
  }
  RemoteXY.Luminosidade=valorLDR;        //guarda o valor do sensor para
transmitir para a interface.
```

```
  dtostrf(valorLDR, 3, 0, RemoteXY.valorL);    //transforma o valor num
vetor de 3 digitos e 0 pntos decimais.
  RemoteXY.Humidade_Solo=valorHumSolo;
  dtostrf(valorHumSolo, 3, 0, RemoteXY.valorHS);
  RemoteXY.Humidade=hum;
  dtostrf(hum, 3, 0, RemoteXY.valorH);
  RemoteXY.Temperatura=temp;
  dtostrf(temp, 2, 0, RemoteXY.valorT);
}

void mensagemSensores (void)
{
  String tmp_comando = "AT+CIPSTART=1,\"";    //ID da ligação =1.
  tmp_comando +="TCP\"";                  //tipo TCP.
  tmp_comando +=",\"";
  tmp_comando += "184.106.153.149";       //api.thingspeak.com
  tmp_comando += "\",80";                 //porto 80
  Serial.println(tmp_comando);
  delay(200);
  String tmp_msg = "GET /update?api_key=";     //campos para o envio dos
valores dos sensores.
  tmp_msg += WRITE_KEY;                     //chave de acesso ao canal da
thingspeak.
  tmp_msg += "&field1=";
  tmp_msg += String(temp);
  tmp_msg += "&field2=";
  tmp_msg += String(hum);
  tmp_msg += "&field3=";
  tmp_msg += String(valorLDR);
  tmp_msg += "&field4=";
  tmp_msg += String(valorHumSolo);
  tmp_msg += "\r\n\r\n";
  tmp_comando = "AT+CIPSEND=1,";       //comando para envio na ligação TCP
de ID=1.
  tmp_comando  +=  String(tmp_msg.length());       //apende  o  tamanho  da
mensagem.
  Serial.println(tmp_comando);
  if(Serial.find(">"))        //Se encontrar este simbolo manda mensagem.
  {
     Serial.print(tmp_msg);
  }
  else
  {
    Serial.print("AT+CIPCLOSE\r\n");    //fecha ligaçao.
  }
}

byte lerBot_menu(int pino)
{
  byte b;
  int c;
  c=analogRead(pino);
  if(c<100)
  {
    b=4;          //enter
  }
  else
  {
    if((c>130)&&(c<150))
    {
      b=3;        //esc
```

```
    }
    else
    {
      if((c>180)&&(c<195))
      {
        b=2;      //cima
      }
      else
      {
        if((c>215)&&(c<235))
        {
          b=1;    //baixo
        }
        else
        {
          if(c>900)
          {
            b=0;  //nenhum botão a ser premido.
          }
        }
      }
    }
 }
 return b;
}



void display_menu()
{
  if((millis() - antMillisBotao) > intervaloBotao)     //evita que exista
oscilações bruscas dos valores lidos.
  {                                                    //assim a probabilidade
de selecionar um menu errado diminui.
    antMillisBotao = millis();
    SbotMenu=lerBot_menu(botMenuPin);
    //Serial.println(analogRead(A2));
  }
  if(MenuFlag==0)   //nivel 0 do menu.
  {

if((mainMenuPag!=antmainMenuPag)||(MenuFlag!=antMenuFlag)||(R!=antMenuR))
    {
      lcd.clear();
      lcd.setCursor(0,0);
      antmainMenuPag=mainMenuPag;
      antMenuFlag=MenuFlag;
      antMenuR=R;
      switch (mainMenuPag)
      {
        case 1: lcd.print("1.DefinirRelogio");
                lcd.setCursor(3,1);
                lcd.print("Rega: ");
                if(R==1)
                {
                  lcd.print("ON ");
                }
                else
                {
                  lcd.print("OFF");
                }
```

```
          break;
        case 2: lcd.print("2.Definir niveis");
                lcd.setCursor(2,1);
                lcd.print("dos sensores");
          break;
        case 3: lcd.print("3.   Controlo");
                lcd.setCursor(6,1);
                lcd.print("manual");
          break;
        case 4: lcd.print("4.Hora de rega e");
                lcd.setCursor(0,1);
                lcd.print("ligar automatico");
          break;
        case 5: lcd.print("5. Leitura dos");
                lcd.setCursor(4,1);
                lcd.print("sensores");
          break;
      }
    }
    if(SbotMenu!=botMenu)
    {
      botMenu=SbotMenu;
      if(botMenu== 4)     //enter
      {
        MenuFlag=1;
      }
      else
      {
        if(botMenu == 2)    //cima
        {
          mainMenuPag++;
          if(mainMenuPag > mainMenuTotal)
          {
            mainMenuPag = 1;
          }
        }
        else
        {
          if(botMenu == 1)   //baixo
          {
            mainMenuPag--;
            if(mainMenuPag == 0)
            {
              mainMenuPag = mainMenuTotal;
            }
          }
        }
      }
    }
  }
  else
  {
    if(MenuFlag==1)   //nivel 1 do menu.
    {
      switch (mainMenuPag)
      {
        case 1: if((MenuFlag!=antMenuFlag)||(MudaSet==1))
              {
                if(MenuFlag!=antMenuFlag)      //executa apenas se entrar
neste menu pela primeira vez.
                {
```

```
                  antMenuFlag=MenuFlag;
                  CursorMainMenu=0;
                  CursorMainMenuLinha=0;
                  anthoraRTC=horaRTC;
                  antminutoRTC=minutoRTC;
                  antdiaRTC=diaRTC;
                  antmesRTC=mesRTC;
                  antanoRTC=anoRTC;
                  hSet1=horaRTC/10;
                  hSet2=horaRTC%10;
                  minSet1=minutoRTC/10;
                  minSet2=minutoRTC%10;
                  dSet1=diaRTC/10;
                  dSet2=diaRTC%10;
                  mSet1=mesRTC/10;
                  mSet2=mesRTC%10;
                  aSet1=(anoRTC%100)/10;
                  aSet2=anoRTC%10;
                  lcd.clear();
                }
                if(MudaSet==1)          //pára o blink do cursor enquanto
atualiza a informação no display se existir mudança de valor.
                {
                  lcd.noBlink();
                }
                lcd.setCursor(0,0);     //executa se for a 1ª vez no menu
ou se mudar de valor.
                lcd.print(hSet1);
                lcd.print(hSet2);
                lcd.print(":");
                lcd.print(minSet1);
                lcd.print(minSet2);
                lcd.setCursor(7,0);
                lcd.print("MM/DD/AA");
                lcd.setCursor(0,1);
                lcd.print(mSet1);
                lcd.print(mSet2);
                lcd.print("/");
                lcd.print(dSet1);
                lcd.print(dSet2);
                lcd.print("/20");
                lcd.print(aSet1);
                lcd.print(aSet2);
                MudaSet=0;
              }
              lcd.setCursor(CursorMainMenu,CursorMainMenuLinha);
              lcd.blink();
              if(SbotMenu!=botMenu)
              {
                botMenu=SbotMenu;
                if(botMenu== 4)     //enter
                {
                  CursorMainMenu++;             //Muda a seleção do valor
horário a alterar.
                  if(CursorMainMenu==2)
                  {
                    CursorMainMenu=3;
                  }
                  if(CursorMainMenu==5)
                  {
                    if(CursorMainMenuLinha==0)
```

```
                    {
                      CursorMainMenu=0;
                      CursorMainMenuLinha=1;
                    }
                    else
                    {
                      CursorMainMenu=8;
                    }
                  }
                  if(CursorMainMenu==10)
                  {
                    CursorMainMenu=0;
                    CursorMainMenuLinha=0;
                  }
                }
                else
                {
                  if(botMenu== 3)    //esc
                  {
                    MenuFlag=0;
                    horaRTC = (hSet1*10)+hSet2;    //Passagem dos valores
individuais para a variável de setup.
                    minutoRTC = (minSet1*10)+minSet2;
                    diaRTC = (dSet1*10)+dSet2;
                    mesRTC = (mSet1*10)+mSet2;
                    anoRTC = 2000+(aSet1*10)+aSet2;
                    if((horaRTC!=anthoraRTC)||(minutoRTC!=antminutoRTC))
                    {
                      rtc.setTime(horaRTC, minutoRTC, 0);     //Setup do
horário do RTC.
                    }

if((diaRTC!=antdiaRTC)||(mesRTC!=antmesRTC)||(anoRTC!=antanoRTC))
                    {
                      rtc.setDate(diaRTC, mesRTC, anoRTC);
                    }
                    lcd.noBlink();
                  }
                  else
                  {
                    if(botMenu == 2)    //142  cima
                    {
                      switch(CursorMainMenu)
                      {
                        case 0: if(CursorMainMenuLinha==0)    //Linha das
horas e minutos.
                                {
                                  hSet1++;         //Selecionar valor das
dezenas para as horas.
                                  if(hSet1 > 2)   //Algarismo das dezenas
nas horas toma apenas valores de 0, 1 e 2.
                                  {
                                    hSet1 = 0;
                                  }
                                  if((hSet1==2)&&(hSet2       >       3))
//Necessario para que nao ocorra situaçoes como hora 26.
                                  {
                                    hSet2 = 3;
                                  }
                                }
```

```
                                else                    //Linha da data do
calendário (MM/DD/AAAA).
                                {
                                  mSet1++;
                                  if(mSet1>1)     //Algorismo das dezenas
do mes toma os valores 0 e 1.
                                  {
                                    mSet1=0;
                                  }
                                  if(mSet1==1)
                                  {
                                    if(mSet2==1)
                                    {
                                      if((dSet1==3)&&(dSet2==1))
//Evita situaçao de dias errados na passagem de Janeiro para Novembro,
como dia 31.
                                      {
                                        dSet2=0;
                                      }
                                    }
                                    if(mSet2>2)         //Evita situaçoes
erradas como por exemplo mes 16.
                                    {
                                      mSet2=1;
                                      if((dSet1==3)&&(dSet2>0))   //Evita
situaçao de dias errados em meses sem o dia 31.
                                      {
                                        dSet2=0;
                                      }
                                    }
                                  }
                                  if(mSet1==0)
                                  {
                                    if(mSet2==0)                //Evita a
introduçao de um mes zero inexistente.
                                    {
                                      mSet2=1;            //Janeiro.
                                    }
                                    if(mSet2==2)          //Fevereiro
                                    {
                                      if(dSet1>1)           //Evita valor
das dezenas dos dias ser 3.
                                      {
                                        dSet1=2;
                                        if(dSet2>8)
                                        {
                                          anoTemp=(aSet1*10)+aSet2;
//Evita dia inexistentes em Fevereiro
                                          if((anoTemp%4)!=0)
//para anos comuns como dia 29.
                                          {
                                            dSet2=8;
                                          }
                                        }
                                      }
                                    }
                                  }
                                }
                                MudaSet=1;
                              break;
                      case 1: if(CursorMainMenuLinha==0)
```

```
                                {
                                  hSet2++;         //Selecionar valor das
unidades para as horas.
                                  if(hSet1<2)
                                  {
                                    if(hSet2 > 9)    //Entre as horas 0 -
19, as unidades variam entre 0 e 9.
                                    {
                                      hSet2 = 0;
                                    }
                                  }
                                  else
                                  {
                                    if(hSet2 > 3)    //Entre as 20 e 23
horas, as unidades variam de 0 a 3.
                                    {
                                      hSet2 = 0;
                                    }
                                  }
                                }
                                else         //Linha da data do calendário
(MM/DD/AAAA).
                                {
                                  mSet2++;
                                  if(mSet1<1)
                                  {
                                    if(mSet2>9)      //Para mSet1=0 apenas
varia de 1 a 9 (Janeiro a Setembro).
                                    {
                                      mSet2=1;
                                    }
                                    if(mSet2==2)    //Fevereiro.
                                    {
                                      if(dSet1>1)   //Para Fevereiro o dia
máximo é 29 em ano bissexto e 28 nos outros.
                                      {
                                        dSet1=2;
                                        if(dSet2>8)
                                        {
                                          anoTemp=(aSet1*10)+aSet2;
//Evita dia inexistentes em Fevereiro
                                          if((anoTemp%4)!=0)
//para anos comuns como dia 29.
                                          {
                                            dSet2=8;
                                          }
                                        }
                                      }
                                    }
                                    else        //outros meses (Janeiro a
Setembro).
                                    {
                                      if((dSet1==3)&&(dSet2>0))
                                      {
                                        dSet2=0;
                                      }
                                    }
                                  }
                                  else      //Outubro a Dezembro.
                                  {
```

```
                                    if(mSet2>2)    //De Outubro a Dezembro
as unidades variam de 0 a 2.
                                    {
                                      mSet2=0;
                                    }
                                    if((dSet1==3)&&(dSet2>0))
                                    {
                                      dSet2=0;
                                    }
                                  }
                                }
                                MudaSet=1;
                             break;
                     case 3: if(CursorMainMenuLinha==0)
                                {
                                  minSet1++;       //Selecionar valor das
dezenas para os minutos.
                                  if(minSet1>5)
                                  {
                                    minSet1 = 0;
                                  }
                                }
                                else      //Linha da data do calendário
(MM/DD/AAAA).
                                {
                                  dSet1++;
                                  if((mSet1==0)&&(mSet2==2))
//Fevereiro
                                  {
                                    if(dSet1>2)           //Só pode tomar
valores das dezenas até 2.
                                    {
                                      dSet1=0;
                                      if(dSet2==0)     //Evita a situaçao
de ter o dia errado zero.
                                      {
                                        dSet2=1;
                                      }
                                    }
                                    if((dSet1==2)&&(dSet2==9))
                                    {
                                      anoTemp=(aSet1*10)+aSet2;   //Evita
dia inexistentes em Fevereiro
                                      if((anoTemp%4)!=0)           //para
anos comuns como dia 29.
                                      {
                                        dSet2=8;
                                      }
                                    }
                                  }
                                  else      //outros meses
                                  {
                                    if(dSet1>3)         //Todos os outros
meses tomam valores das dezenas até 3.
                                    {
                                      dSet1=0;
                                      if(dSet2==0)     //Evita a situaçao
de ter o dia errado zero.
                                      {
                                        dSet2=1;
                                      }
```

```
                                    }
                                    if((dSet1==3)&&(dSet2>0))     //Evita
situaçoes inexistentes como por exemplo dia 37.
                                    {
                                      dSet2=0;
                                    }
                                  }
                                }
                                MudaSet=1;
                              break;
                        case 4: if(CursorMainMenuLinha==0)
                                {
                                  minSet2++;       //Selecionar valor das
unidades para os minutos.
                                  if(minSet2>9)
                                  {
                                    minSet2 = 0;
                                  }
                                }
                                else       //Linha da data do calendário
(MM/DD/AAAA).
                                {
                                  dSet2++;
                                  if(dSet1==0)
                                  {
                                    if(dSet2>9)     //dias 1 a 9.
                                    {
                                      dSet2=1;
                                    }
                                  }
                                  if(dSet1==1)
                                  {
                                    if(dSet2>9)     //dias 10 a 19.
                                    {
                                      dSet2=0;
                                    }
                                  }
                                  if(dSet1==2)
                                  {
                                    if((mSet1==0)&&(mSet2==2))
//Fevereiro
                                    {
                                      anoTemp=(aSet1*10)+aSet2;   //Evita
dia inexistentes em Fevereiro
                                      if((anoTemp%4)!=0)            //para
anos comuns como dia 29.
                                      {
                                        if(dSet2>8)       //Só pode tomar
valores das unidades até 8.
                                        {
                                          dSet2=0;
                                        }
                                      }
                                      else      //ano bissexto.
                                      {
                                        if(dSet2>9)       //Só pode tomar
valores das unidades até 9.
                                        {
                                          dSet2=0;
                                        }
                                      }
```

```
                                    }
                                    else          //outros meses.
                                    {
                                      if(dSet2>9)         //Só pode tomar
valores das unidades até 9.
                                      {
                                        dSet2=0;
                                      }
                                    }
                                  }
                                  if(dSet1==3)
                                  {
                                    if((mSet1==0)&&(mSet2<8))
                                    {
                                      if((mSet2%2)!=0)    //meses impares
os dias tem valor maximo de 31.
                                      {
                                        if(dSet2>1)
                                        {
                                          dSet2=0;
                                        }
                                      }
                                      else
                                      {
                                        if(dSet2>0)       //meses pares o
dia maximo é 30.
                                        {
                                          dSet2=0;
                                        }
                                      }
                                    }
                                    else             //meses de Agosto a
Dezembro.
                                    {
                                      if((mSet2%2)==0)
                                      {
                                        if(dSet2>1)       //meses pares o
dia maximo é 31.
                                        {
                                          dSet2=0;
                                        }
                                      }
                                      else
                                      {
                                        if(dSet2>0)       //meses impares
o dia maximo é 30.
                                        {
                                          dSet2=0;
                                        }
                                      }
                                    }
                                  }
                                }
                                MudaSet=1;
                              break;
                        case  8:  aSet1++;         //Selecionar valor das
dezenas para o ano.
                                if(aSet1>9)
                                {
                                  aSet1 = 0;
                                }
```

```
                                MudaSet=1;
                              break;
                        case  9:  aSet2++;          //Selecionar  valor  das
unidades para o ano.
                                if(aSet2>9)
                                {
                                  aSet2 = 0;
                                }
                                MudaSet=1;
                              break;
                      }
                    }
                    else
                    {
                      if(botMenu == 1) //207  baixo
                      {
                        switch(CursorMainMenu)
                        {
                          case 0: if(CursorMainMenuLinha==0)
                                  {
                                    hSet1--;        //Selecionar valor das
dezenas para as horas.
                                    if(hSet1 == 255)      //Decrementando
a zero o byte voltará a 255.
                                    {
                                      hSet1 = 2;
                                    }
                                    if((hSet1==2)&&(hSet2      >      3))
//Necessario para que nao ocorra situaçoes como hora 26.
                                    {
                                      hSet2 = 3;
                                    }
                                  }
                                  else     //Linha da data do calendário
(MM/DD/AAAA).
                                  {
                                    mSet1--;
                                    if(mSet1==255)        //Algorismo  das
dezenas do mes toma os valores 0 e 1.
                                    {
                                      mSet1=1;
                                    }
                                    if(mSet1==1)
                                    {
                                      if(mSet2==1)
                                      {
                                        if((dSet1==3)&&(dSet2==1))
//Evita situaçao de dias errados na passagem de Janeiro para Novembro,
como dia 31.
                                        {
                                          dSet2=0;
                                        }
                                      }
                                      if(mSet2>2)                  //Evita
situaçoes erradas como por exemplo mes 16.
                                      {
                                        mSet2=1;
                                        if((dSet1==3)&&(dSet2==1))
//Evita situaçao de dias errados para meses sem o dia 33.
                                        {
                                          dSet2=0;
```

```
                                        }
                                      }
                                    }
                                    if(mSet1==0)
                                    {
                                      if(mSet2==0)              //Evita a
introduçao de um mes zero inexistente.
                                      {
                                        mSet2=1;            //Janeiro.
                                      }
                                      if(mSet2==2)          //Fevereiro
                                      {
                                        if(dSet1>1)         //Evita valor
das dezenas dos dias ser 3.
                                        {
                                          dSet1=2;
                                          if(dSet2>8)
                                          {
                                            anoTemp=(aSet1*10)+aSet2;
//Evita dia inexistentes em Fevereiro
                                            if((anoTemp%4)!=0)
//para anos comuns como dia 29.
                                            {
                                              dSet2=8;
                                            }
                                          }
                                        }
                                      }
                                    }
                                  }
                                  MudaSet=1;
                              break;
                        case 1: if(CursorMainMenuLinha==0)
                                {
                                  hSet2--;       //Selecionar valor das
unidades para as horas.
                                  if(hSet1<2)
                                  {
                                    if(hSet2 == 255)
                                    {
                                      hSet2 = 9;
                                    }
                                  }
                                  else
                                  {
                                    if(hSet2 == 255)
                                    {
                                      hSet2 = 3;
                                    }
                                  }
                                }
                                else       //Linha da data do calendário
(MM/DD/AAAA).
                                {
                                  mSet2--;
                                  if(mSet1<1)
                                  {
                                    if(mSet2<1)         //Para  mSet1=0
apenas varia de 1 a 9 (Janeiro a Setembro).
                                    {
                                      mSet2=9;
```

```
                                    }
                                    if(mSet2==2)    //Fevereiro.
                                    {
                                      if(dSet1>1)    //Para Fevereiro o
dia máximo é 29 em ano bissexto e 28 nos outros.
                                      {
                                        dSet1=2;
                                        if(dSet2>8)
                                        {
                                          anoTemp=(aSet1*10)+aSet2;
//Evita dia inexistentes em Fevereiro
                                          if((anoTemp%4)!=0)
//para anos comuns como dia 29.
                                          {
                                            dSet2=8;
                                          }
                                        }
                                      }
                                    }
                                    else      //outros meses (Janeiro a
Setembro).
                                    {
                                      if((dSet1==3)&&(dSet2>0))
                                      {
                                        dSet2=0;
                                      }
                                    }
                                  }
                                  else      //Outubro a Dezembro.
                                  {
                                    if(mSet2==255)       //De Outubro a
Dezembro as unidades variam de 0 a 2.
                                    {
                                      mSet2=2;
                                    }
                                    if((dSet1==3)&&(dSet2>0))
                                    {
                                      dSet2=0;
                                    }
                                  }
                                }
                                MudaSet=1;
                            break;
                        case 3: if(CursorMainMenuLinha==0)
                                {
                                  minSet1--;     //Selecionar valor das
dezenas para os minutos.
                                  if(minSet1==255)
                                  {
                                    minSet1 = 5;
                                  }
                                }
                                else       //Linha da data do calendário
(MM/DD/AAAA).
                                {
                                  dSet1--;
                                  if((mSet1==0)&&(mSet2==2))
//Fevereiro
                                  {
                                    if(dSet1==255)       //Só pode tomar
valores das dezenas até 2.
```

```
                                        {
                                          dSet1=2;
                                        }
                                        if((dSet1==0)&&(dSet2==0))
//Evita a situaçao de ter o dia errado zero.
                                        {
                                          dSet2=1;
                                        }
                                        if((dSet1==2)&&(dSet2==9))
                                        {
                                          anoTemp=(aSet1*10)+aSet2;
//Evita dia inexistentes em Fevereiro
                                          if((anoTemp%4)!=0)          //para
anos comuns como dia 29.
                                          {
                                            dSet2=8;
                                          }
                                        }
                                      }
                                      else      //outros meses
                                      {
                                        if(dSet1==255)           //Todos os
outros meses tomam valores das dezenas até 3.
                                        {
                                          dSet1=3;
                                        }
                                        if((dSet1==0)&&(dSet2==0))
//Evita a situaçao de ter o dia errado zero.
                                        {
                                          dSet2=1;
                                        }
                                        if((dSet1==3)&&(dSet2>0))    //Evita
situaçoes inexistentes como por exemplo dia 37.
                                        {
                                          dSet2=0;
                                        }
                                      }
                                    }
                                    MudaSet=1;
                                break;
                            case 4: if(CursorMainMenuLinha==0)
                                    {
                                      minSet2--;      //Selecionar valor das
unidades para os minutos.
                                      if(minSet2==255)
                                      {
                                        minSet2 = 9;
                                      }
                                    }
                                    else                //Linha da data do
calendário (MM/DD/AAAA).
                                    {
                                      dSet2--;
                                      if(dSet1==0)
                                      {
                                        if(dSet2<1)     //dias 1 a 9.
                                        {
                                          dSet2=9;
                                        }
                                      }
                                      if(dSet1==1)
```

```
                                    {
                                      if(dSet2==255)     //dias 10 a 19.
                                      {
                                        dSet2=9;
                                      }
                                    }
                                    if(dSet1==2)
                                    {
                                      if((mSet1==0)&&(mSet2==2))
//Fevereiro
                                      {
                                        anoTemp=(aSet1*10)+aSet2;
//Evita dia inexistentes em Fevereiro
                                        if((anoTemp%4)!=0)          //para
anos comuns como dia 29.
                                        {
                                          if(dSet2==255)         //Só pode
tomar valores das unidades até 8.
                                          {
                                            dSet2=8;
                                          }
                                        }
                                        else      //ano bissexto.
                                        {
                                          if(dSet2==255)         //Só pode
tomar valores das unidades até 9.
                                          {
                                            dSet2=9;
                                          }
                                        }
                                      }
                                      else          //outros meses.
                                      {
                                        if(dSet2==255)           //Só pode
tomar valores das unidades até 9.
                                        {
                                          dSet2=9;
                                        }
                                      }
                                    }
                                    if(dSet1==3)
                                    {
                                      if((mSet1==0)&&(mSet2<8))
                                      {
                                        if((mSet2%2)!=0)           //meses
impares os dias tem valor maximo de 31.
                                        {
                                          if(dSet2==255)
                                          {
                                            dSet2=1;
                                          }
                                        }
                                        else
                                        {
                                          if(dSet2==255)           //meses
pares o dia maximo é 30.
                                          {
                                            dSet2=0;
                                          }
                                        }
                                      }
```

```
                                    else            //meses de Agosto a
Dezembro.
                                    {
                                      if((mSet2%2)==0)
                                      {
                                        if(dSet2==255)          //meses
pares o dia maximo é 31.
                                        {
                                          dSet2=1;
                                        }
                                      }
                                      else
                                      {
                                        if(dSet2==255)          //meses
impares o dia maximo é 30.
                                        {
                                          dSet2=0;
                                        }
                                      }
                                    }
                                  }
                                  MudaSet=1;
                                break;
                              case 8: aSet1--;        //Selecionar valor das
dezenas para o ano.
                                  if(aSet1==255)
                                  {
                                    aSet1 = 9;
                                  }
                                  MudaSet=1;
                                break;
                              case 9: aSet2--;        //Selecionar valor das
unidades para o ano.
                                  if(aSet2==255)
                                  {
                                    aSet2 = 9;
                                  }
                                  MudaSet=1;
                                break;
                            }
                          }
                        }
                      }
                    }
                  break;
          case                                                          2:
if((subMenu2Pag!=antsubMenu2Pag)||(MenuFlag!=antMenuFlag))
                  {
                    lcd.clear();
                    lcd.setCursor(0,0);
                    antsubMenu2Pag=subMenu2Pag;
                    antMenuFlag=MenuFlag;
                    switch (subMenu2Pag)
                    {
                      case 1: lcd.print("2.1 Temperatura");
                          break;
                      case 2: lcd.print("2.2 Luz");
                          break;
                      case 3: lcd.print("2.3 Humidade do");
```

```
                    lcd.setCursor(5,1);
                    lcd.print("Solo");
                  break;
              }
            }
            if(SbotMenu!=botMenu)
            {
              botMenu=SbotMenu;
              if(botMenu== 4)     //enter
              {
                 MenuFlag=2;
              }
              else
              {
                if(botMenu== 3)    //esc
                {
                  MenuFlag=0;
                }
                else
                {
                  if(botMenu == 2)    //142  cima
                  {
                    subMenu2Pag++;
                    if(subMenu2Pag > 3)
                    {
                      subMenu2Pag = 1;
                    }
                  }
                  else
                  {
                    if(botMenu == 1)   //207   baixo
                    {
                      subMenu2Pag--;
                      if(subMenu2Pag == 0)
                      {
                        subMenu2Pag = 3;
                      }
                    }
                  }
                }
              }
            }
          break;
        case                                                              3:
if(((subMenu3Pag!=antsubMenu3Pag)||(MenuFlag!=antMenuFlag))&&(RemoteXY.Co
ntroloM_remot==0)) //para não existir conflitos com controlo remoto, este
menu não trabalha com remoto=1.
            {
              lcd.clear();
              lcd.setCursor(0,0);
              antsubMenu3Pag=subMenu3Pag;
              antMenuFlag=MenuFlag;
              ctrlManual=1;
              menuLocalManual=1;
              switch (subMenu3Pag)
              {
                case 1: lcd.print("3.1 Controlo da");
                        lcd.setCursor(4,1);
                        lcd.print("janela");
                break;
                case 2: lcd.print("3.2 Controlo de");
```

```
                        lcd.setCursor(5,1);
                        lcd.print("rega");
                    break;
                    case 3: lcd.print("3.3 Controlo da");
                        lcd.setCursor(3,1);
                        lcd.print("esteira");
                    break;
                  }
                }
                if(RemoteXY.ControloM_remot==0)
                {
                  if(SbotMenu!=botMenu)   //controlo das paginas de seleçao
dos botoes.
                  {
                    botMenu=SbotMenu;
                    if(botMenu== 4)     //enter
                    {
                      MenuFlag=2;
                    }
                    else
                    {
                      if(botMenu== 3)    //esc
                      {
                        MenuFlag=0;
                        ctrlManual=0;
                        menuLocalManual=0;
                      }
                      else
                      {
                        if(botMenu == 2)    //142  cima
                        {
                          subMenu3Pag++;
                          if(subMenu3Pag > 3)
                          {
                            subMenu3Pag = 1;
                          }
                        }
                        else
                        {
                          if(botMenu == 1)   //207 baixo
                          {
                            subMenu3Pag--;
                            if(subMenu3Pag == 0)
                            {
                              subMenu3Pag = 3;
                            }
                          }
                        }
                      }
                    }
                  }
                }
              break;
          case                                                                4:
if((subMenu4Pag!=antsubMenu4Pag)||(MenuFlag!=antMenuFlag))
                {
                  lcd.clear();
                  lcd.setCursor(0,0);
                  antsubMenu4Pag=subMenu4Pag;
                  antMenuFlag=MenuFlag;
                  switch (subMenu4Pag)
```

```
                {
                  case 1: lcd.print("4.1 Forma de");
                          lcd.setCursor(2,1);
                          lcd.print("ativar a rega");
                    break;
                  case 2: lcd.print("4.2 Horario");
                    break;
                  case 3: lcd.print("4.3 Horario 2");
                    break;
                }
              }
              if(SbotMenu!=botMenu)
              {
                botMenu=SbotMenu;
                if(botMenu== 4)     //enter
                {
                  MenuFlag=2;
                }
                else
                {
                  if(botMenu== 3)    //esc
                  {
                    MenuFlag=0;
                  }
                  else
                  {
                    if(botMenu == 2)    //  cima
                    {
                      subMenu4Pag++;
                      if(biDia==1)        //Se a funçao bidiária da rega
estiver ativa mostra horário 2.
                      {
                        if(subMenu4Pag > 3)
                        {
                          subMenu4Pag = 1;
                        }
                      }
                      else              //Caso contrário apenas mostra as
duas primeiras páginas.
                      {
                        if(subMenu4Pag > 2)
                        {
                          subMenu4Pag = 1;
                        }
                      }
                    }
                    else
                    {
                      if(botMenu == 1)   //  baixo
                      {
                        subMenu4Pag--;
                        if(biDia==1)      //Se a funçao bidiária da rega
estiver ativa mostra horário 2.
                        {
                          if(subMenu4Pag == 0)
                          {
                            subMenu4Pag = 3;
                          }
                        }
                        else        //Caso contrário apenas mostra as duas
primeiras páginas.
```

```
                    {
                      if(subMenu4Pag == 0)
                      {
                        subMenu4Pag = 2;
                      }
                    }
                  }
                }
              }
            }
          }
      break;
  case 5: if(MenuFlag!=antMenuFlag)
          {
            antMenuFlag=MenuFlag;
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("Temp:");
            lcd.setCursor(9,0);
            lcd.print("Hum:");
            lcd.setCursor(0,1);
            lcd.print("LDR:");
            lcd.setCursor(7,1);
            lcd.print("HSolo:");
          }
          lcd.setCursor(5,0);
          if(temp<10)
          {
            lcd.print("0");
          }
          lcd.print(temp);
          lcd.print("\xDF");
          lcd.setCursor(13,0);
          if(hum<10)
          {
            lcd.print("0");
          }
          lcd.print(hum);
          lcd.setCursor(4,1);
          if(valorLDR<10)
          {
            lcd.print("0");
          }
          lcd.print(valorLDR);
          lcd.setCursor(13,1);
          if(valorHumSolo<10)
          {
            lcd.print("0");
          }
          lcd.print(valorHumSolo);
          lcd.print(" ");
          if(SbotMenu!=botMenu)
          {
            botMenu=SbotMenu;
            if(botMenu== 3)    //esc
            {
              MenuFlag=0;
            }
          }
      break;
}
```

```
    }
    else        //nivel 2 do menu.
    {
      switch (mainMenuPag)
      {
        case 2:
                switch (subMenu2Pag)
                {
                  case 1: if(MenuFlag!=antMenuFlag)
                          {
                            antMenuFlag=MenuFlag;
                            CursorSubmenu2=0;
                            lcd.clear();
                            lcd.setCursor(0,0);
                            lcd.print("Maximo     ("     "\xDF"     "C):");
//Simbolo de ° no LCD precisa de ser codificada com \xDF para aparecer.
                            lcd.print(tempMax);
                            lcd.setCursor(0,1);
                            lcd.print("Minimo (" "\xDF" "C):");
                            lcd.print(tempMin);
                            lcd.setCursor(12,CursorSubmenu2);
                            lcd.blink();
                          }
                          if(CursorSubmenu2==0)
                          {
                            lcd.setCursor(12,CursorSubmenu2);
                            if(tempMax<10)
                            {
                              lcd.print(" ");
                            }
                            lcd.print(tempMax);
                          }
                          if(CursorSubmenu2==1)
                          {
                            lcd.setCursor(12,CursorSubmenu2);
                            if(tempMin<10)
                            {
                              lcd.print(" ");
                            }
                            lcd.print(tempMin);
                          }
                          if(SbotMenu!=botMenu)
                          {
                            botMenu=SbotMenu;
                            if(botMenu== 4)     //enter
                            {
                              if(CursorSubmenu2==0)
                              {
                                CursorSubmenu2=1;
                              }
                              else
                              {
                                CursorSubmenu2=0;
                              }
                            }
                            else
                            {
                              if(botMenu== 3)    //esc
                              {
                                  MenuFlag=1;
                                  lcd.noBlink();
```

```
                                }
                                else
                                {
                                  if(botMenu == 2)    //142  cima
                                  {
                                    if(CursorSubmenu2==0)      //Cursor  na
primeira linha.
                                    {
                                      tempMax++;
                                      if(tempMax > 40)
                                      {
                                        tempMax = tempMin+5;
                                      }
                                    }
                                    if(CursorSubmenu2==1)      //Cursor  na
segunda linha.
                                    {
                                      tempMin++;
                                      if(tempMin > (tempMax-5))
                                      {
                                        tempMin = 5;
                                      }
                                    }
                                  }
                                  else
                                  {
                                    if(botMenu == 1) //207  baixo
                                    {
                                      if(CursorSubmenu2==0)     //Cursor  na
primeira linha.
                                      {
                                        tempMax--;
                                        if(tempMax < (tempMin+5))
                                        {
                                          tempMax = 40;
                                        }
                                      }
                                      if(CursorSubmenu2==1)     //Cursor  na
segunda linha.
                                      {
                                        tempMin--;
                                        if(tempMin < 5)
                                        {
                                          tempMin = (tempMax-5);
                                        }
                                      }
                                    }
                                  }
                                }
                              }
                            }
                    break;
                    case 2: if(MenuFlag!=antMenuFlag)
                            {
                              antMenuFlag=MenuFlag;
                              CursorSub2menu2=0;
                              lcd.clear();
                              lcd.setCursor(0,0);
                              lcd.print("Maxima (%): ");
                              lcd.print(LDRmax);
                              lcd.setCursor(0,1);
```

```
  lcd.print("Minima (%):  ");
  lcd.print(LDRmin);
  lcd.setCursor(12,CursorSub2menu2);
  lcd.blink();
}
if(CursorSub2menu2==0)
{
  lcd.setCursor(12,CursorSub2menu2);
  if(LDRmax<10)
  {
    lcd.print("  ");
  }
  else
  {
    if(LDRmax<100)
    {
      lcd.print(" ");
    }
  }
  lcd.print(LDRmax);
}
if(CursorSub2menu2==1)
{
  lcd.setCursor(12,CursorSub2menu2);
  if(LDRmin<10)
  {
    lcd.print("  ");
  }
  else
  {
    if(LDRmin<100)
    {
      lcd.print(" ");
    }
  }
  lcd.print(LDRmin);
}
if(SbotMenu!=botMenu)
{
  botMenu=SbotMenu;
  if(botMenu== 4)     //enter
  {
    if(CursorSub2menu2==0)
    {
      CursorSub2menu2=1;
    }
    else
    {
      CursorSub2menu2=0;
    }
  }
  else
  {
    if(botMenu== 3)    //esc
    {
        MenuFlag=1;
        lcd.noBlink();
    }
    else
    {
      if(botMenu == 2)    //142  cima
```

```
                                {
                                  if(CursorSub2menu2==0)      //Cursor  na
primeira linha.
                                  {
                                    LDRmax++;
                                    if(LDRmax > 100)
                                    {
                                      LDRmax = LDRmin+5;
                                    }
                                  }
                                  if(CursorSub2menu2==1)      //Cursor  na
segunda linha.
                                  {
                                    LDRmin++;
                                    if(LDRmin > (LDRmax-5))
                                    {
                                      LDRmin = 1;
                                    }
                                  }
                                }
                                else
                                {
                                  if(botMenu == 1) //207  baixo
                                  {
                                    if(CursorSub2menu2==0)    //Cursor na
primeira linha.
                                    {
                                      LDRmax--;
                                      if(LDRmax < (LDRmin+5))
                                      {
                                        LDRmax = 100;
                                      }
                                    }
                                    if(CursorSub2menu2==1)    //Cursor na
segunda linha.
                                    {
                                      LDRmin--;
                                      if(LDRmin < 1)
                                      {
                                        LDRmin = (LDRmax-5);
                                      }
                                    }
                                  }
                                }
                              }
                            }
                          }
                        }
                  break;
                  case 3: if(MenuFlag!=antMenuFlag)
                          {
                            antMenuFlag=MenuFlag;
                            CursorSub3menu2=0;
                            lcd.clear();
                            lcd.setCursor(0,0);
                            lcd.print("Maximo (%): ");
                            lcd.print(HumSoloMax);
                            lcd.setCursor(0,1);
                            lcd.print("Minimo (%):  ");
                            lcd.print(HumSoloMin);
                            lcd.setCursor(12,CursorSub3menu2);
                            lcd.blink();
```

```
                    }
                    if(CursorSub3menu2==0)
                    {
                      lcd.setCursor(12,CursorSub3menu2);
                      if(HumSoloMax<10)
                      {
                        lcd.print("  ");
                      }
                      else
                      {
                        if(HumSoloMax<100)
                        {
                          lcd.print(" ");
                        }
                      }
                      lcd.print(HumSoloMax);
                    }
                    if(CursorSub3menu2==1)
                    {
                      lcd.setCursor(12,CursorSub3menu2);
                      if(HumSoloMin<10)
                      {
                        lcd.print("  ");
                      }
                      else
                      {
                        lcd.print(" ");
                      }
                      lcd.print(HumSoloMin);
                    }
                    if(SbotMenu!=botMenu)
                    {
                      botMenu=SbotMenu;
                      if(botMenu== 4)     //enter
                      {
                        if(CursorSub3menu2==0)
                        {
                          CursorSub3menu2=1;
                        }
                        else
                        {
                          CursorSub3menu2=0;
                        }
                      }
                      else
                      {
                        if(botMenu == 3)    //esc
                        {
                          MenuFlag=1;
                          lcd.noBlink();
                        }
                        else
                        {
                          if(botMenu == 2)    //142  cima
                          {
                            if(CursorSub3menu2==0)      //Cursor  na
primeira linha.
                            {
                              HumSoloMax++;
                              if(HumSoloMax > 100)
                              {
```

```
                                    HumSoloMax = HumSoloMin+5;
                                  }
                                }
                                if(CursorSub3menu2==1)      //Cursor  na
segunda linha.
                                {
                                  LDRmin++;
                                  if(HumSoloMin > (HumSoloMax-5))
                                  {
                                    HumSoloMin = 1;
                                  }
                                }
                              }
                              else
                              {
                                if(botMenu == 1) //207  baixo
                                {
                                  if(CursorSub3menu2==0)    //Cursor na
primeira linha.
                                  {
                                    HumSoloMax--;
                                    if(HumSoloMax < (HumSoloMin+5))
                                    {
                                      HumSoloMax = 100;
                                    }
                                  }
                                  if(CursorSub3menu2==1)    //Cursor na
segunda linha.
                                  {
                                    HumSoloMin--;
                                    if(HumSoloMin < 1)
                                    {
                                      HumSoloMin = (HumSoloMax-5);
                                    }
                                  }
                                }
                              }
                            }
                          }
                        }
                      break;
                    }
            break;
            case 3: switch (subMenu3Pag)
                    {
                      case 1: if(MenuFlag!=antMenuFlag)
                              {
                                antMenuFlag=MenuFlag;
                                lcd.clear();
                                lcd.setCursor(0,0);
                                lcd.print("Janela:");
                              }
                              if(janela==1)
                              {
                                lcd.setCursor(8,0);
                                lcd.print("aberta ");
                              }
                              else
                              {
                                lcd.setCursor(8,0);
                                lcd.print("fechada");
```

```
        }
        if(SbotMenu!=botMenu)
        {
          botMenu=SbotMenu;
          if(botMenu == 3)    //esc
          {
            MenuFlag=1;
          }
          else
          {
            if(botMenu == 2)  //cima
            {
              janela++;
              if(janela>1)
              {
                janela=0;
              }
            }
            else
            {
              if(botMenu == 1)  //baixo
              {
                janela--;
                if(janela<0)
                {
                  janela=1;
                }
              }
            }
          }
        }
break;
case 2: if(MenuFlag!=antMenuFlag)
        {
          antMenuFlag=MenuFlag;
          lcd.clear();
          lcd.setCursor(0,0);
          lcd.print("Rega:");
        }
        if(R==1)
        {
          lcd.setCursor(6,0);
          lcd.print("ON ");
        }
        else
        {
          lcd.setCursor(6,0);
          lcd.print("OFF");
        }
        if(SbotMenu!=botMenu)
        {
          botMenu=SbotMenu;
          if(botMenu== 3)    //esc
          {
              MenuFlag=1;
          }
          else
          {
            if(botMenu == 2)  //cima
            {
              R++;
```

```
                if(R>1)
                {
                  R=0;
                }
              }
              else
              {
                if(botMenu == 1)  //baixo
                {
                  R--;
                  if(R<0)
                  {
                    R=1;
                  }
                }
              }
            }
          }
break;
case 3: if(MenuFlag!=antMenuFlag)
          {
            antMenuFlag=MenuFlag;
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("Esteira:");
          }
          if(esteira==1)
          {
            lcd.setCursor(8,0);
            lcd.print("aberta ");
          }
          else
          {
            lcd.setCursor(8,0);
            lcd.print("fechada");
          }
          if(SbotMenu!=botMenu)
          {
            botMenu=SbotMenu;
            if(botMenu== 3)    //esc
            {
                MenuFlag=1;
            }
            else
            {
              if(botMenu == 2)  //cima
              {
                esteira++;
                if(esteira>1)
                {
                  esteira=0;
                }
              }
              else
              {
                if(botMenu == 1)  //baixo
                {
                  esteira--;
                  if(esteira<0)
                  {
                    esteira=1;
```

```
                                    }
                                  }
                                }
                              }
                            }
                      break;
                    }
            break;
            case 4: switch (subMenu4Pag)
                    {
                      case                                                    1:
if((formaRega!=antformaRega)||(MenuFlag!=antMenuFlag))
                              {
                                antMenuFlag=MenuFlag;
                                antformaRega=formaRega;
                                lcd.clear();
                                lcd.setCursor(0,0);
                                switch(formaRega)
                                {
                                  case 1: lcd.print("Apenas sensor");
                                          //RSensor=1;
                                          //diaRSensor=0;
                                     break;
                                  case 2: lcd.print("Diario c/sensor");
                                     break;
                                  case 3: lcd.print("Diario s/sensor");
                                     break;
                                  case 4: lcd.print("Bidiario");
                                          lcd.setCursor(0,1);
                                          lcd.print("s/sensor");
                                     break;
                                }
                              }
                              if(SbotMenu!=botMenu)
                              {
                                botMenu=SbotMenu;
                                if(botMenu== 3)    //esc
                                {
                                  MenuFlag=1;
                                  if(formaRega==4)
                                  {
                                    biDia=1;
                                  }
                                  else
                                  {
                                    biDia=0;
                                  }
                                }
                                else
                                {
                                  if(botMenu == 2)    //142  cima
                                  {
                                    formaRega++;
                                    if(formaRega > 4)
                                    {
                                      formaRega = 1;
                                    }
                                  }
                                  else
                                  {
                                    if(botMenu == 1) //207  baixo
```

```
                                {
                                  formaRega--;
                                  if(formaRega == 0)
                                  {
                                    formaRega = 4;
                                  }
                                }
                              }
                            }
                          }
                    break;
                  case 2: if((MenuFlag!=antMenuFlag)||(mudaHorario1==1))
                          {
                            if(MenuFlag!=antMenuFlag)   //Primeira vez que
entra neste submenu.
                            {
                              antMenuFlag=MenuFlag;
                              CursorSub2menu4=0;
                              lcd.clear();
                            }
                            if(mudaHorario1==1)          //Existe mudança
nos valores do horario 1, logo o selector blink é desactivado.
                            {
                              lcd.noBlink();
                            }
                            lcd.setCursor(0,0);
                            lcd.print(horarioH1);
                            lcd.print(horarioH2);
                            lcd.print(":");
                            lcd.print(horarioM1);
                            lcd.print(horarioM2);
                            mudaHorario1=0;
                          }
                          lcd.setCursor(CursorSub2menu4,0);
                          lcd.blink();
                          if(SbotMenu!=botMenu)
                          {
                            botMenu=SbotMenu;
                            if(botMenu== 4)     //enter
                            {
                              CursorSub2menu4++;          //Muda a seleção
do valor horário a alterar.
                              if(CursorSub2menu4==2)
                              {
                                CursorSub2menu4=3;
                              }
                              if(CursorSub2menu4==5)
                              {
                                CursorSub2menu4=0;
                              }
                            }
                            else
                            {
                              if(botMenu== 3)    //esc
                              {
                                MenuFlag=1;
                                lcd.noBlink();
                                hora=(horarioH1*10)+horarioH2;
                                minuto=(horarioM1*10)+horarioM2;
                              }
                              else
```

```
                                {
                                  if(botMenu == 2)    //142  cima
                                  {
                                    switch(CursorSub2menu4)
                                    {
                                      case          0:          horarioH1++;
//Selecionar valor das dezenas para as horas.
                                                if(horarioH1 > 2)
                                                {
                                                  horarioH1 = 0;
                                                }
                                                if((horarioH1==2)&&(horarioH2
> 3))   //Necessario para que nao ocorra situaçoes como hora 26.
                                                {
                                                  horarioH2 = 3;
                                                }
                                                mudaHorario1=1;
                                            break;
                                          case 1: horarioH2++;      //Selecionar
valor das unidades para as horas.
                                                if(horarioH1<2)
                                                {
                                                  if(horarioH2 > 9)
                                                  {
                                                    horarioH2 = 0;
                                                  }
                                                }
                                                else
                                                {
                                                  if(horarioH2 > 3)
                                                  {
                                                    horarioH2 = 0;
                                                  }
                                                }
                                                mudaHorario1=1;
                                            break;
                                          case 3: horarioM1++;      //Selecionar
valor das dezenas para os minutos.
                                                if(horarioM1>5)
                                                {
                                                  horarioM1 = 0;
                                                }
                                                mudaHorario1=1;
                                            break;
                                          case 4: horarioM2++;      //Selecionar
valor das unidades para os minutos.
                                                if(horarioM2>9)
                                                {
                                                  horarioM2 = 0;
                                                }
                                                mudaHorario1=1;
                                            break;
                                    }
                                  }
                                  else
                                  {
                                    if(botMenu == 1) //207  baixo
                                    {
                                      switch(CursorSub2menu4)
                                      {
```

```
                                        case        0:        horarioH1--;
//Selecionar valor das dezenas para as horas.
                                                  if(horarioH1 == 255)
                                                  {
                                                    horarioH1 = 2;
                                                  }

if((horarioH1==2)&&(horarioH2 > 3))    //Necessario para que nao ocorra
situaçoes como hora 26.
                                                  {
                                                    horarioH2 = 3;
                                                  }
                                                  mudaHorario1=1;
                                               break;
                                        case        1:        horarioH2--;
//Selecionar valor das unidades para as horas.
                                                  if(horarioH1<2)
                                                  {
                                                    if(horarioH2 == 255)
                                                    {
                                                      horarioH2 = 9;
                                                    }
                                                  }
                                                  else
                                                  {
                                                    if(horarioH2 == 255)
                                                    {
                                                      horarioH2 = 3;
                                                    }
                                                  }
                                                  mudaHorario1=1;
                                               break;
                                        case        3:        horarioM1--;
//Selecionar valor das dezenas para os minutos.
                                                  if(horarioM1==255)
                                                  {
                                                    horarioM1 = 5;
                                                  }
                                                  mudaHorario1=1;
                                               break;
                                        case        4:        horarioM2--;
//Selecionar valor das unidades para os minutos.
                                                  if(horarioM2==255)
                                                  {
                                                    horarioM2 = 9;
                                                  }
                                                  mudaHorario1=1;
                                               break;
                                          }
                                        }
                                      }
                                    }
                                  }
                                }
                              break;
                        case 3:  if((MenuFlag!=antMenuFlag)||(mudaHorario2==1))
                                 {
                                   if(MenuFlag!=antMenuFlag)   //Primeira vez que
entra neste submenu.
                                   {
                                     antMenuFlag=MenuFlag;
```

```
                                CursorSub3menu4=0;
                                lcd.clear();
                              }
                              if(mudaHorario2==1)
                              {
                                lcd.noBlink();
                              }
                              lcd.setCursor(0,0);
                              lcd.print(horario2H1);
                              lcd.print(horario2H2);
                              lcd.print(":");
                              lcd.print(horario2M1);
                              lcd.print(horario2M2);
                              mudaHorario2=0;
                            }
                            lcd.setCursor(CursorSub3menu4,0);
                            lcd.blink();
                            if(SbotMenu!=botMenu)
                            {
                              botMenu=SbotMenu;
                              if(botMenu== 4)     //enter
                              {
                                CursorSub3menu4++;          //Muda a seleção
do valor horário a alterar.
                                if(CursorSub3menu4==2)
                                {
                                  CursorSub3menu4=3;
                                }
                                if(CursorSub3menu4==5)
                                {
                                  CursorSub3menu4=0;
                                }
                              }
                              else
                              {
                                if(botMenu== 3)    //esc
                                {
                                  MenuFlag=1;
                                  lcd.noBlink();
                                  hora2=(horario2H1*10)+horario2H2;
                                  minuto2=(horario2M1*10)+horario2M2;
                                }
                                else
                                {
                                  if(botMenu == 2)    //142  cima
                                  {
                                    switch(CursorSub3menu4)
                                    {
                                      case         0:         horario2H1++;
//Selecionar valor das dezenas para as horas.
                                                  if(horario2H1 > 2)
                                                  {
                                                    horario2H1 = 0;
                                                  }

if((horario2H1==2)&&(horario2H2 > 3))    //Necessario para que nao ocorra
situaçoes como hora 26.
                                                  {
                                                    horario2H2 = 3;
                                                  }
                                                  mudaHorario2=1;
```

```
                                    break;
                                case 1: horario2H2++;     //Selecionar
valor das unidades para as horas.
                                        if(horario2H1<2)
                                        {
                                          if(horario2H2 > 9)
                                          {
                                            horario2H2 = 0;
                                          }
                                        }
                                        else
                                        {
                                          if(horario2H2 > 3)
                                          {
                                            horario2H2 = 0;
                                          }
                                        }
                                        mudaHorario2=1;
                                    break;
                                case 3: horario2M1++;     //Selecionar
valor das dezenas para os minutos.
                                        if(horario2M1>5)
                                        {
                                          horario2M1 = 0;
                                        }
                                        mudaHorario2=1;
                                    break;
                                case 4: horario2M2++;     //Selecionar
valor das unidades para os minutos.
                                        if(horario2M2>9)
                                        {
                                          horario2M2 = 0;
                                        }
                                        mudaHorario2=1;
                                    break;
                              }
                            }
                            else
                            {
                              if(botMenu == 1) //207  baixo
                              {
                                switch(CursorSub3menu4)
                                {
                                  case        0:        horario2H1--;
//Selecionar valor das dezenas para as horas.
                                        if(horario2H1 == 255)
                                        {
                                          horario2H1 = 2;
                                        }

if((horario2H1==2)&&(horario2H2 > 3))    //Necessario para que nao ocorra
situaçoes como hora 26.
                                        {
                                          horario2H2 = 3;
                                        }
                                        mudaHorario2=1;
                                    break;
                                case        1:        horario2H2--;
//Selecionar valor das unidades para as horas.
                                        if(horario2H1<2)
                                        {
```

```
                                                    if(horario2H2 == 255)
                                                    {
                                                      horario2H2 = 9;
                                                    }
                                                  }
                                                  else
                                                  {
                                                    if(horario2H2 == 255)
                                                    {
                                                      horario2H2 = 3;
                                                    }
                                                  }
                                                  mudaHorario2=1;
                                              break;
                                          case        3:        horario2M1--;
//Selecionar valor das dezenas para os minutos.
                                                  if(horario2M1==255)
                                                  {
                                                    horario2M1 = 5;
                                                  }
                                                  mudaHorario2=1;
                                              break;
                                          case        4:        horario2M2--;
//Selecionar valor das unidades para os minutos.
                                                  if(horario2M2==255)
                                                  {
                                                    horario2M2 = 9;
                                                  }
                                                  mudaHorario2=1;
                                              break;
                                        }
                                      }
                                    }
                                  }
                                }
                              }
                          break;
                      }
                  break;
              }
          }
      }
  }

void controlo_atuadores()
{
  if(ctrlManual==0)
  {
    if(temp>=tempMax)
    {
      if(janela==0)     //caso esteja fechado, abre a janela.
      {
        motorPasso.step(-CntPassos);      //sentido contra-relógio.
        janela=1;     //acima da temperatura máxima a janela abre.
        antJanela=1;    //evita conflitos com controlo manual.
      }
    }
    if(temp<=tempMin)
    {
      if(janela==1)     //caso esteja aberto, fecha a janela.
      {
```

```
        motorPasso.step(CntPassos);   //sentido dos ponteiros do relógio.
        janela=0;     //abaixo da temperatura minima a janela fecha.
        antJanela=0;
      }
    }
    if(valorLDR>=LDRmax)
    {
      if(esteira==1)    //Caso a esteira esteja aberta, fecha.
      {
        esteira=0;      //acima da percentagem de luz fecha a esteira.
        antEsteira=0;   //evita conflitos com controlo manual.
        func_motor=1;       //indica que o motor está em funcionamento.
        antMotor = millis();                  //regista o tempo atual de
ativação.
        digitalWrite(MotorPinA, LOW);     //Sentido horário.
        digitalWrite(MotorPinB, HIGH);
      }
    }
    if(valorLDR<=LDRmin)
    {
      if(esteira==0)    //Caso a esteira esteja fechada, abre.
      {
        esteira=1;      //abaixo da percentagem de luz abre a esteira.
        antEsteira=1;
        func_motor=1;       //indica que o motor está em funcionamento.
        antMotor = millis();                  //regista o tempo atual de
ativação.
        digitalWrite(MotorPinA, HIGH);    //Sentido anti-horário.
        digitalWrite(MotorPinB, LOW);
      }
    }
    switch(formaRega)
    {
      case 1: if(valorHumSolo>=HumSoloMax)        //forma de rega só com
sensor.
              {
                R=0;      //acima da percentagem de humidade no solo fecha
a rega.
              }
              if(valorHumSolo<=HumSoloMin)
              {
                R=1;     //abaixo da percentagem de humidade no solo ativa
a rega.
              }
              break;
      case 2: if((hora==horaRTC)&&(minuto==minutoRTC))    //forma de rega
com sensor numa hora especifica.
              {
                if(valorHumSolo<=HumSoloMin)
                {
                  R=1;       //abaixo da percentagem de humidade no solo
ativa a rega.
                }
              }
              if(valorHumSolo>=HumSoloMax)    //Continua a testar o nivel
de humidade até fechar a rega.
              {
                R=0;      //acima da percentagem de humidade no solo fecha
a rega.
              }
              break;
```

```
      case 3: if((hora==horaRTC)&&(minuto==minutoRTC))    //forma de rega
sem sensor numa hora especifica.
              {
                R=1;
                antRegaDia = millis();      //guarda o valor atual para
comparação futura.

              }
              if(R==1)
              {
                if   ((millis()   -   antRegaDia)   >   intervaloRegaDia)
//Condição para fazer a leitura quando o tempo de intervalo for atingido.
                {
                  R=0;
                }
              }
              break;
      case 4: if((hora==horaRTC)&&(minuto==minutoRTC))    //forma de rega
sem sensor em duas horas especificas.
              {
                R=1;
                if(flagTempo==0)
                {
                  flagTempo=1;
                  antRegaDia2 = millis();     //guarda o valor atual para
comparação futura.
                }
              }
              if((hora2==horaRTC)&&(minuto2==minutoRTC))      //forma  de
rega sem sensor em duas horas especificas.
              {
                R=1;
                if(flagTempo==0)
                {
                  flagTempo=1;
                  antRegaDia2 = millis();     //guarda o valor atual para
comparação futura.
                }
              }
              if(R==1)
              {
                if   ((millis()   -   antRegaDia2)   >   intervaloRegaDia)
//Condição para fazer a leitura quando o tempo de intervalo for atingido.
                {
                  R=0;
                  flagTempo=0;
                }
              }
              break;
    }
    antR=R;
  }
  else        //controlo manual ativado.
  {
    if((janela==1)&&(antJanela==0)) //Caso o estado anterior da janela ser
fechado quando for para abrir, então abre a janela.
    {
      motorPasso.step(-CntPassos);   //abrir janela.
      antJanela=janela;
    }
    if((janela==0)&&(antJanela==1))
```

```
    {
      motorPasso.step(CntPassos);    //fechar janela.
      antJanela=janela;
    }
    if((esteira==1)&&(antEsteira==0))
    {
      func_motor=1;
      antEsteira=esteira;
      antMotor = millis();
      digitalWrite(MotorPinA, HIGH);    //Sentido anti-horário.
      digitalWrite(MotorPinB, LOW);
    }
    if((esteira==0)&&(antEsteira==1))
    {
      func_motor=1;
      antEsteira=esteira;
      antMotor = millis();
      digitalWrite(MotorPinA, LOW);   //Sentido horário.
      digitalWrite(MotorPinB, HIGH);
    }
    if((R==1)&&(antR==0))
    {
      antR=R;
    }
    if((R==0)&&(antR==1))
    {
      antR=R;
    }
  }
  if(func_motor==1)
  {
    if ((millis() - antMotor) > intervaloMotor)   //Condição para parar o
motor quando o intervalo for atingido.
    {
      func_motor=0;
      digitalWrite(MotorPinA, LOW);
      digitalWrite(MotorPinB, LOW);
    }
  }
}

void loop()
{
  t = rtc.getTime();   //recebe dados do rtc.
  horaRTC = t.hour;   //Guarda os valores recebidos do RTC nas variaveis
para manipulação.
  minutoRTC = t.min;
  diaRTC = t.date;
  mesRTC = t.mon;
  anoRTC = t.year;
  if ((millis() - antMillis) > intervalo)   //Condição para fazer a leitura
quando o tempo de intervalo for atingido.
  {
    antMillis = millis();    //guarda o valor atual para comparação futura.
    ler_sensores();
  }
  if((millis()-antMillisThing)>intervaloThing)      //manda dados para o
servidor ThingSpeak de 15 em 15 segundos.
  {
    antMillisThing = millis();       //guarda o valor atual para comparação
futura.
```

```
    mensagemSensores();
  }
  RemoteXY_Handler ();
  if(menuLocalManual==1)        //=1 se estiver no menu do módulo local de
controlo manual.
  {
    if(RemoteXY.ControloM_remot==1)      //=1 se o módulo remoto ativar o
controlo manual.
    {
      ctrlManual=1;
      menuLocalManual=0;             //deixa de estar no menu de controlo
local do módulo local.
      MenuFlag=0;                   //passa para o nível 0 do menu local.
      janela=RemoteXY.Janela_remota;        //passa os dados recebidos do
módulo remoto para as variáveis locais.
      esteira=RemoteXY.esteira_remota;
      R=RemoteXY.rega_remota;
    }
  }
  else    //não está no menu de controlo manual do módulo local.
  {
    if(RemoteXY.ControloM_remot==1)    //o módulo remoto ativa o controlo
manual.
    {
      ctrlManual=1;
      janela=RemoteXY.Janela_remota;      //passa os dados recebidos do
módulo remoto para as variáveis locais.
      esteira=RemoteXY.esteira_remota;
      R=RemoteXY.rega_remota;
    }
    else
    {
      ctrlManual=0;     //o módulo remoto desativa o controlo manual.
    }
  }
  display_menu();
  controlo_atuadores();
}
```