Sistemas de Computação Móvel e Ubíqua (2020/21)

Relatório Final

Sistema de Monitorização e Rega de medronheiros

Aluno:

Henrique Raposo 57059

Docentes:

Prof. Carmen Pires Morgado

Prof. Nuno Manuel Ribeiro Preguiça

Prof. Cecilia Gomes

1. Introdução

O medronheiro é uma árvore frutífera, que se pode encontrar maioritariamente em zonas de alto declive onde dificilmente outras culturas sobrevivem, sendo os maiores exemplos destas zonas as serras do Caldeirão e Monchique.

O seu fruto é bastante apreciado e utlizado também na produção de licores e aguardentes, mas devido a esta arvore ter melhor sucesso em zonas de mais difícil acesso a cultura desta torna-se difícil. Umas das razões desta dificuldade é a energia e canalizações necessárias para bombear água periodicamente para cada uma destas árvores nestes sítios mais remotos.

É para solucionar este problema e ajudar a cultura deste fruto que proponho a solução detalhada neste relatório.

2. Visão Geral

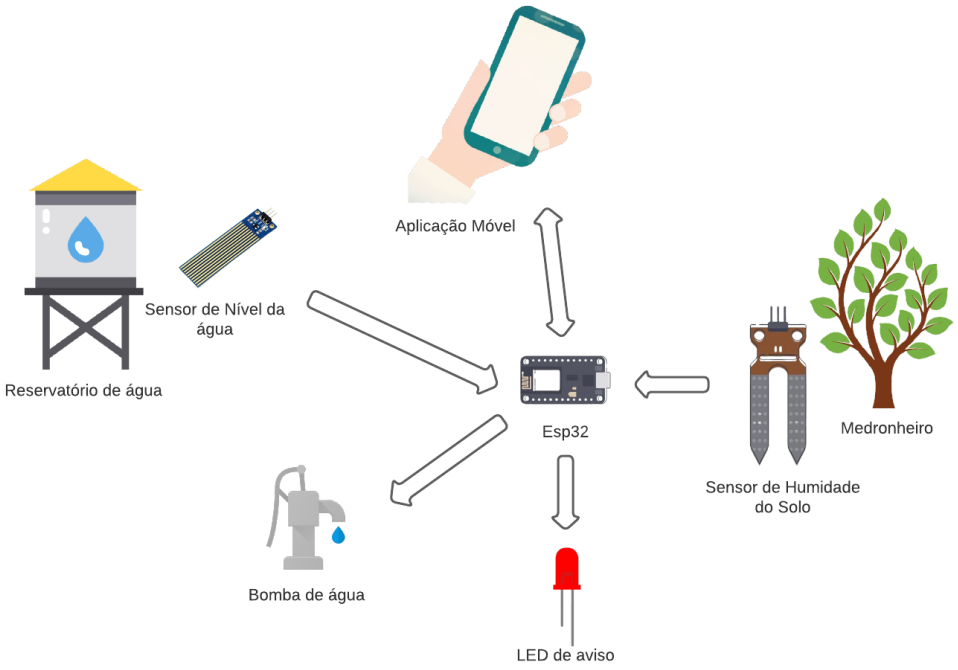

Figura 1 Diagrama da Arquitetura Geral Do Sistema

Como solução para as dificuldades anteriormente apresentadas este sistema pretende dar ao utilizador uma maneira de fornecer água a cada uma das plantas de acordo com a necessidade destas, de uma maneira automática, ou com o premir de um botão através da aplicação se o

utilizador assim achar conveniente. Reduzindo o desperdício de água e as deslocações do utilizador a estes sítios de mais difícil acesso, notificando quando é necessário o reenchimento do reservatório, e oferecendo também um banco de dados sobre cada planta, útil para futuras análises com o intuito de otimização dos recursos e estudo da planta.

Para atingir esta solução seria necessário um reservatório de água com um sensor de nível de água, uma bomba de água, e um led ligados a um microcontrolador com ligação a um servidor. Desta maneira é possível ligar a bomba de água através do servidor, e monitorizar remotamente o nível da água. O led seria vermelho e teria como objetivo indicar visivelmente a quem estiver no local que o nível de água se encontra baixo.

A cada planta também é necessário associar um microcontrolador com capacidade de se ligar através de Wi-Fi ao servidor de maneira a reduzir a dificuldade e custo de instalação. A este microcontrolador estaria ligado um sensor de humidade do solo que seria utilizado para comunicar os valores da humidade do solo perto da planta ao servidor.

O servidor tem como objetivo disponibilizar estes valores ao utilizador através da web para uma aplicação móvel.

Esta aplicação irá e por sua vez mostrar as informações do reservatório indicando o seu nível e notificando o utilizador quando o nível deste se encontra em baixo. Irá também mostras as informações de cada respetiva planta assim como as operações de regar instantaneamente, agendar a rega e ligar ou desligar a rega automática.

3. Arquitetura do Sistema

A arquitetura do sistema como prova de conceito é composta pelos seguintes componentes

- ESP32 – microcontrolador escolhido pelas capacidades de comunicação Wi-Fi, este é responsável pelo processamento dos valores lidos pelos sensores, envio destes, e receção de valores de ativação de atuadores.
- 2 LEDS – Ambos ligados ao microcontrolador. Um led vermelho como indicador de baixo nível de água e um led azul simulador da ativação da bomba de água
- 1 Potenciómetro – Sensor analógico ligado ao ESP32 escolhido para simular o nível de água
- 1 Sensor de humidade de solo - Sensor analógico ligado ao ESP32 escolhido para obter leituras da humidade do solo perto da planta.

- Aplicação Móvel – Interface para o utilizador que mostra os dados lidos pelos sensores, e responsável pelas funções que ligam e desligam a rega automática e manual
- Base de Dados – Base de dados NoSQL escolhida pela maior facilidade em escalabilidade, velocidade de acesso e escrita, e falta de relações que justifiquem uma base de dados relacional

4. Aplicação Móvel

De maneira a disponibilizar os dados para a aplicação móvel é utilizada a base de dados NoSQL em tempo real fornecida pela plataforma Firebase com o modelo de dados representado na seguinte figura:

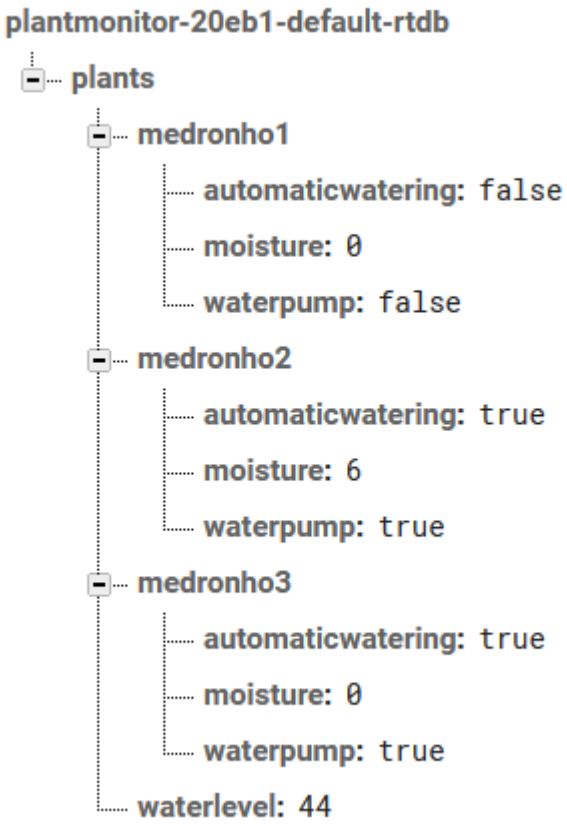

Figura 2 Exemplo de Modelo de Dados guardado no Firebase

Esta disposição dos dados permite distinguir os dados de cada planta individual e o nível do reservatório disponível para o conjunto destas.

A Aplicação foi feita com o uso do Android Studio utilizando Kotlin como linguagem e serve ao utilizador como maneira de interagir remotamente com o sistema.

A tela inicial corresponde a uma lista com os plantas registadas, esta lista é conseguida através de um *RecyclerViewer que* é uma maneira mais eficiente de mostrar listas de maior tamanho ao utilizador, substituindo os componentes *Scrollable*.

Este componente é mais eficiente pois em vez de carregar para uma view cada um dos itens, esta carrega apenas os visíveis e passa os dados dos que vão saindo da tela com o scroll para os novos itens que vão aparecendo.

Figura 3 Screenshot da tela inicial da aplicação

Ao pressionar um dos itens, a aplicação mostra as informações respetivas a planta escolhida. Estes dados são atualizados em tempo real com a utilização de listeners às respetivas referências na base de dados.

Incluído nestes dados estão as informações sobre a humidade do solo, o nível da água, assim como interruptores para alteração do estado da bomba e ligar ou desligar a rega automática.

É também possível encontrar nesta tela um gráfico feito com o uso da biblioteca MPAndroidchart, onde está desenhada uma linha com um conjunto de dados fixo, onde após a correta implementação, seria suposto mostrar os valores do sensor de humidade da planta ao longo do tempo, para uma mais fácil análise.

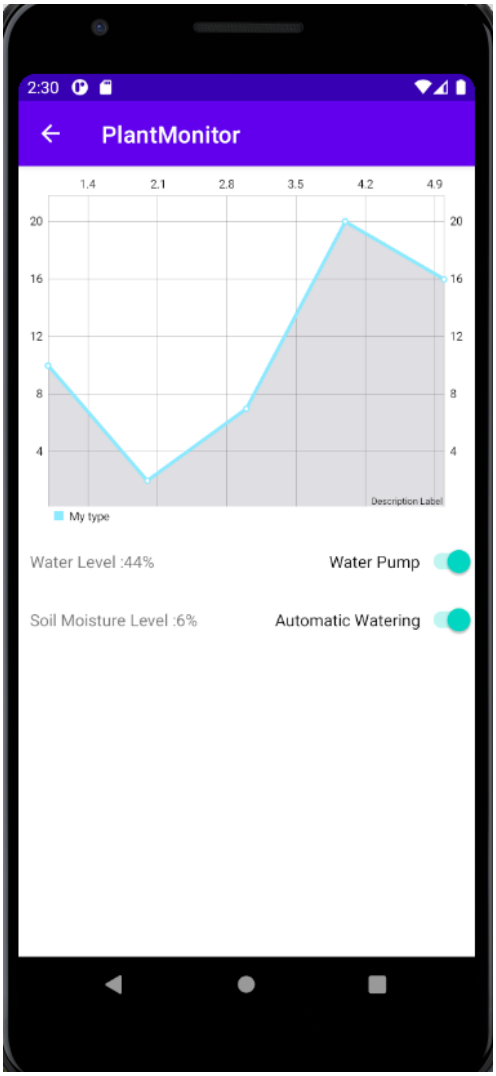

Figura 4 Screenshot da tela correspondente aos detalhes da planta

5. Sensores e Reações

- LEDs

Neste projeto foram utilizados dois LEDs. Um de cor vermelha utilizado como notificação do baixo nível de água, que é ligado quando o sensor do nível de água (simulado com o potenciómetro) está a ler um valor abaixo do pré-determinado na variável *waterLevelThreshold* no código.

O outro LED é de cor azul e é utilizado para simular a ativação da bomba de água. Esta ativação ocorre quando é chamada a função *turnOnPump().*

Cada circuito necessita de estar ligado a um dos pinos do ESP32 capaz de output digital, e ao GND após passar por uma resistência, como demonstrado na figura.

No circuito do trabalho foram utilizados para esta conexão os pinos D22 e D23 e GND.

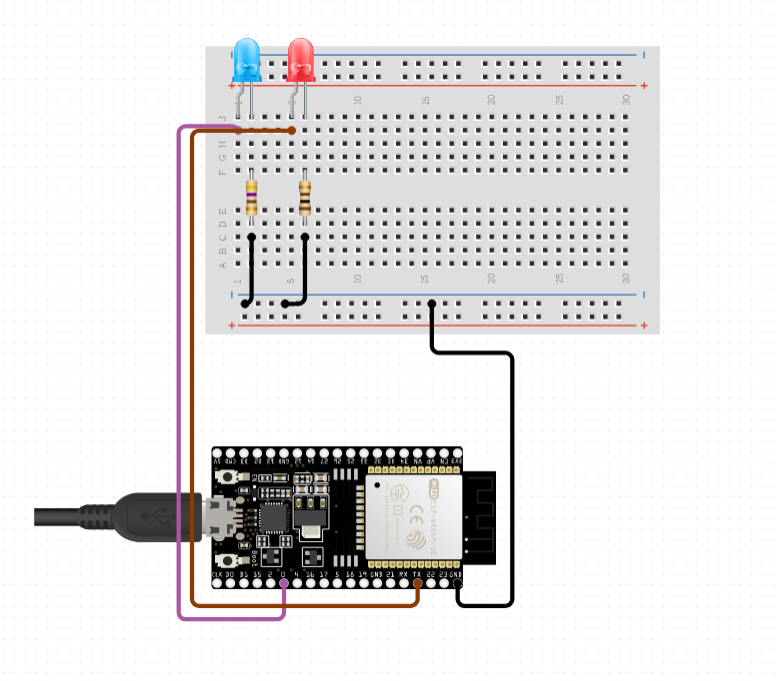

Figura 5 Esquema de demonstração de ligação do ESP32 a LEDs

- Potenciómetro

  Para propósitos de simplificar o processo de testagem neste projeto, foi utilizado um potenciómetro no lugar de um sensor de nível de água devido a similaridade de utilização e conexão, mas mais facilidade na testagem de diferentes valores.

  Este sensor precisa de três ligações, uma ao pino de GND, uma de alimentação e outra de input analógico que vai medir as diferenças de resistência causadas pelo rodar do potenciómetro.

  Quanto menor a resistência feita pelo potenciómetro maior o valor lido pelo pino analógico, maior o nível de água simulado. Este valor para uma melhor usabilidade e facilidade de leitura foi mapeado para um valor percentual: *waterLevelPercent* que em conjunto com o *waterLevelThreshold* vai ditar se o nível da água se encontra alto o suficiente para ligar a bomba em modo automático.

  No circuito implementado o potenciómetro encontra se ligado aos pinos D35, 3V3 e GND

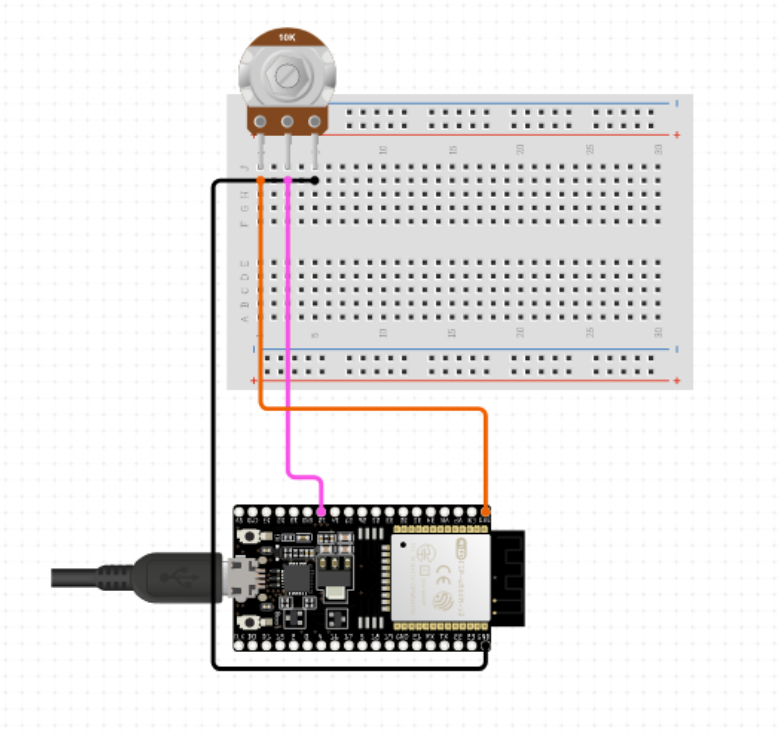

Figura 6 Esquema de demonstração de ligação do ESP32 a potenciómetro

- Sensor de Humidade do solo

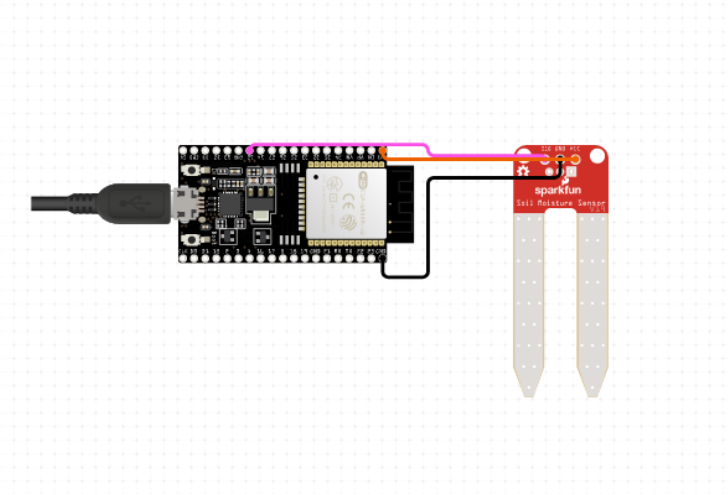

Figura 7 Esquema de demonstração de ligação do ESP32 a sensor de humidade do solo

Ligado de maneira similar ao potenciómetro e com um funcionamento com base nos mesmo princípios, este sensor é utilizado para medir a necessidade da rega da planta e por isso, quando o valor de humidade lido pelo sensor é menor que o valor definido pela constante *waterLevelThreshold* se a rega automática estiver ligada a bomba vai ser ativada.

No circuito implementado o sensor encontra se ligado aos pinos D34, 3V3 e GND

6. Experiências

- Teste de atualização da aplicação:

  Ao ligar o ESP32 e selecionar o Medronho 1, os valores demonstrados pela aplicação devem ser os mesmos lidos e demonstrados no monitor série do Arduino IDE. Isto pode ser testado ao rodar o potenciómetro ou mergulhando o sensor de humidade num copo de água.

- Teste de Rega Automática:

  Com a aplicação e o ESP32 ligado, se na aplicação na tela do Medronho 1 o interruptor de rega automática estiver ligado o sensor de humidade estiver num sítio seco, e o potenciómetro estiver numa posição que simule o nível de água a mais de 10% a luz azul que simula a ativação da bomba deve ligar durante um período de 6 segundos.

- Teste de Rega Manual:

  Com a aplicação ligada e o ESP32 ligado, se na aplicação na tela do Medronho 1 ao ligar o interruptor da bomba de água a luz azul que simula a ativação da bomba deve ligar durante um período de 6 segundos e o interruptor deve voltar ao estado inicial.

- Teste de aviso de baixo nível de água

  Com a aplicação e o ESP32 ligado, se o potenciómetro estiver numa posição que simule o baixo nível de água, a aplicação deve mostrar esse valor e o LED vermelho deve acender.

7. Conclusões

Com a realização deste projeto a solução aparenta ser possível apesar de algumas das funcionalidades pensadas inicialmente terem ficado por implementar por razões de tempo.

Nestas funcionalidades estavam incluídas a opção de agendar uma hora de rega, e a acumulação de dados da humidade do solo da planta ao longo do tempo. Visto que o objetivo seria maximizar o uso da água, o estudo do registo temporal destes valores seria fulcral na definição de uma hora ótima para a rega da planta.

Após este processo seria também benéfico e de semelhante implementação o uso de outros sensores como fotoresistências.

De mais difícil implementação, mas também muito interessante a nível de dados obtidos seria o registo periódico fotográfico da árvore de maneira a avaliar o seu crescimento e o crescimento do seu fruto.