

PROJETO HORTA

Cláudio Henrique Carvalho Bacelete;

Leandro Martins Ferreira de Oliveira;

Mateus Henrique Ramalho Lima;

Mikael José Araya Bruno;

Victor Sousa Rocha

Engenharia de Computação (PUC Minas);

Professor: Mário Guimarães Buratto;

Sumário

- Introdução
- Desenvolvimento
 - 1) Projeto de Hardware
 - 2) Esquema Elétrico
 - 3) Projeto de Software
 - 4) Funcionalidades
 - 5) Arquitetura
- Conclusão
- Referências e Bibliografias

Introdução

Inicialmente, o nosso projeto foi desenvolvido no intuito de criar uma horta inteligente, que por meio do sensor de umidade do solo e dos horários definidos para iluminação, seja capaz de realizar irrigação e iluminação automaticamente. Além disso, por meio de um aplicativo com conexão bluetooth, o usuário é capaz de acionar manualmente a iluminação e a irrigação e verificar a umidade do solo.

Para o desenvolvimento da horta, foi utilizado o ESP32 com FreeRTOS, permitindo a utilização de múltiplas threads e tasks para cada componente do projeto, garantindo assim melhor performance nas leituras e execuções do sistema embarcado.

Já para o aplicativo, utilizamos o MIT APP INVENTOR para criar um aplicativo simples para conexão com bluetooth já presente na placa do ESP32.

Desenvolvimento

1) Projeto de Hardware:

Na parte física do projeto, utilizamos o microcontrolador ESP32 no qual realizamos toda a programação e as conexões entre os componentes.

Para os sensores, implementamos somente o sensor de umidade de solo, que utilizando da resistência elétrica do meio, podemos ler um valor de umidade do solo.

Como atuadores, utilizamos de uma bombinha de aquário para irrigar a planta quando o valor de umidade estiver abaixo do limite mínimo, que é acionada utilizando de um transistor NPN BC337 para aumentar o valor de corrente fornecida pelo ESP32. Além disso, utilizamos uma lâmpada incandescente junto com um relé para fazer o acionamento da iluminação.

Para a interação do usuário com a horta utilizamos de um display LCD com I2C para reduzir o número de conexões e um teclado matricial para exibir e permitir a configuração do ambiente da horta.

2) Esquema Elétrico:

O esquema elétrico está representado na figura 1.

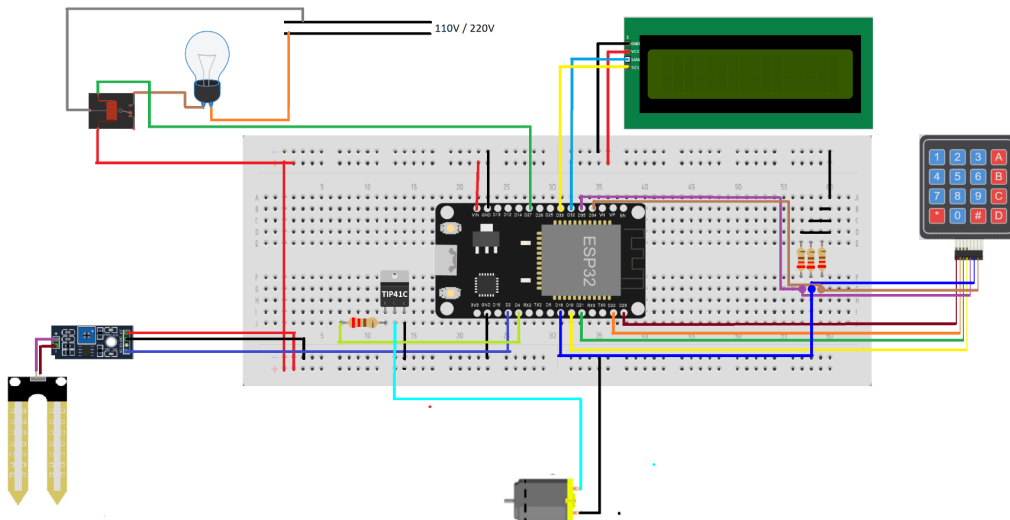


Figura 1: Esquema Elétrico.

3) Projeto de Software:

Para o desenvolvimento da aplicação, utilizamos da linguagem C++ para microcontroladores utilizando do FreeRTOS, um sistema operacional de tempo real que permite melhor gerenciamento de dispositivos, melhor portabilidade e confiabilidade e o suporte ao paralelismo por mais de uma thread simultânea.

Conforme mostrado na figura 2, consta o código para a criação de todas as tasks, sendo elas:

- vTaskConfiguracao: responsável por configurar todas as informações iniciais,
- vTaskSensorUmidade: é responsável por fazer leitura no sensor de umidade e se esse valor lido for menor que o valor padrão configurado é acionado o timer xTimerIrrigação por cerca de 10 segundos.
- vTaskLampada: é responsável por mandar um sinal para o relé caso o chegue no horário programado para acionar a lâmpada.
- vTaskRelogio: é responsável pelo relógio do sistema, sendo uma task que é acionada a cada 60 segundos, dessa forma é possível trabalhar com tempo real, nessa mesma tarefa é verificado se o está na hora de ligar ou desligar a lâmpada, caso positivo, é enviado um sinal através da fila xQueue_Lampada, essa mesma fila fica na task vTaskLampada aguardando receber um sinal.
- vTaskApp: é responsável por receber sinal do aplicativo e acionar os componentes.

```
xTaskCreate(vTaskConfiguracao, "Configuracao", configMINIMAL_STACK_SIZE + 2048, NULL, 1, &xTaskHandleConfiguracao);
xTaskCreate(vTaskSensorUmidade, "SensorUmidade", configMINIMAL_STACK_SIZE + 2048, NULL, 1, &xTaskHandleSensorUmidade);
xTaskCreate(vTaskLampada, "Lampada", configMINIMAL_STACK_SIZE + 2048, NULL, 1, &xTaskHandleLampada);
xTaskCreate(vTaskRelogio, "Relogio", configMINIMAL_STACK_SIZE + 2048, NULL, 1, &xTaskHandleRelogio);
xTaskCreate(vTaskApp, "App", configMINIMAL_STACK_SIZE + 2048, NULL, 1, &xTaskHandleApp);
```

Figura 2: Criação das Tarefas.

Já para o desenvolvimento do aplicativo de celular, utilizamos o MIT APP INVENTOR.

O aplicativo bluetooth possui duas telas, a tela inicial e a tela de ajuda. A tela inicial possui os botões de ligar e desligar para as ações de irrigação e iluminação, onde o usuário pode manualmente definir o tempo de funcionamento dos componentes. Além disso, a tela possui o botão para conexão bluetooth, botão de interrogação para a seção de ajuda e botão de configuração. Na tela de ajuda, há uma breve explicação acerca do funcionamento do aplicativo.

4) Funcionalidades:

Após ligarmos o projeto, o usuário será informado pelo display LCD para inserir por meio do teclado matricial as configurações iniciais, como hora atual, hora de início de iluminação, hora de fim de iluminação e valor mínimo de umidade. Em seguida, todas as funções da horta começam a funcionar efetivamente.

Se, em algum momento, a umidade lida pelo sensor for inferior ao valor mínimo definido pelo usuário, a bomba de irrigação vai ser acionada até o valor ser superior ao valor mínimo. Quando a hora atual entrar no intervalo de horas de iluminação, o ESP32 vai acionar a lâmpada até o horário sair desse intervalo.

Além disso, o usuário pode a qualquer momento se conectar ao ESP32 por meio de um aplicativo bluetooth no intuito de controlar manualmente a irrigação e iluminação das plantas, permitindo monitoramento customizável.

5) Arquitetura:

Na figura 3, tem-se a arquitetura do projeto.

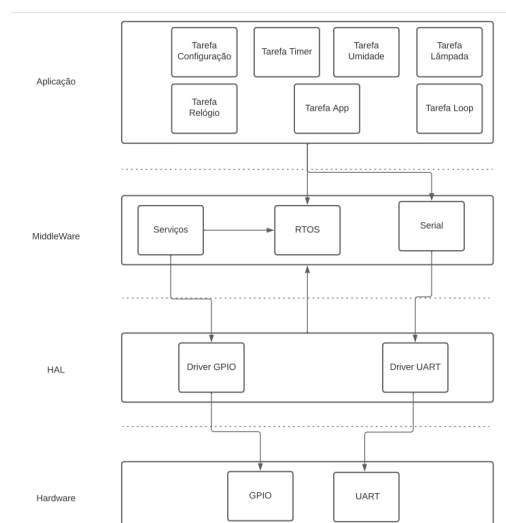


Figura 3: Arquitetura do projeto.

6) Metodologia:

No decorrer do projeto seguimos o cronograma estabelecido no início, as tarefas ou sprints foram divididas entre os membros do grupo e a cada 15 dias era realizado a daily para verificar se o progresso de cada aspecto do projeto.

Product Owner: Mateus Henrique;

Scrum Master: Victor Sousa;

A seguir o cronograma seguido.

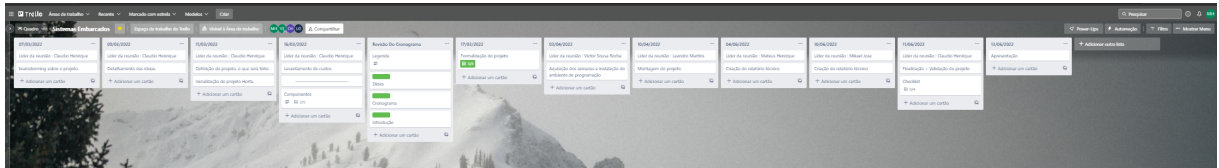


Figura 4: Cronograma.

Fonte:

<https://trello.com/invite/b/rPYa50zq/82cc006344bd6a1847be1c7add246377/sistemas-embarcados>

Cada sprint representada no cronograma corresponde a uma tarefa, que seria uma procedimento individual programado como task no FreeRTOS.

Conclusão

Com o mundo cada vez mais conectado, fica evidente que é necessário automatizar tarefas domésticas, tais como uma horta, nesse sentido, o projeto Smart Garden busca otimizar essa atividade, além disso, sua construção trouxe conhecimentos práticos de planejamento, eletrônicos e de programação a todos os integrantes, contribuindo para a finalização de um projeto robusto, de utilidade prática no mundo.

Referências Bibliográficas

<https://www.udemy.com/course/primeiros-passos-com-o-freertos/>

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/index.html>

<https://www.freertos.org/>

<https://www.dobitaobYTE.com.br/geral/esp32/>