

RELATÓRIO DE ATIVIDADES PRIMEIRA SEMANA - ÁREA IOT

**PONTOS DA APRESENTAÇÃO**

**Desenvolvimento do DLV1.**

O DLV1 é a primeira versão do microcontrolador de baixo custo que iremos desenvolver utilizando a placa (Arduino/Esp32), nele gostaríamos de implementar as funções necessárias para o decorrer do desenvolvimento da plantação:

* Interface gráfica para controle de erros e notificações sobre o funcionamento das bombas.
* Entradas de conexão para recebimento de dados via cabo (SDI12).
* Módulos de conexão via Bluetooth/Wifi.

Placa Esp32:

* Conectividade nativa WIFI e Bluetooth.
* Processamento de 240 MHz.
* Tensão baixa de 3.3V, exige conversores para módulos de 5V.
* Poucos pinos analógicos.

Placa Arduino:

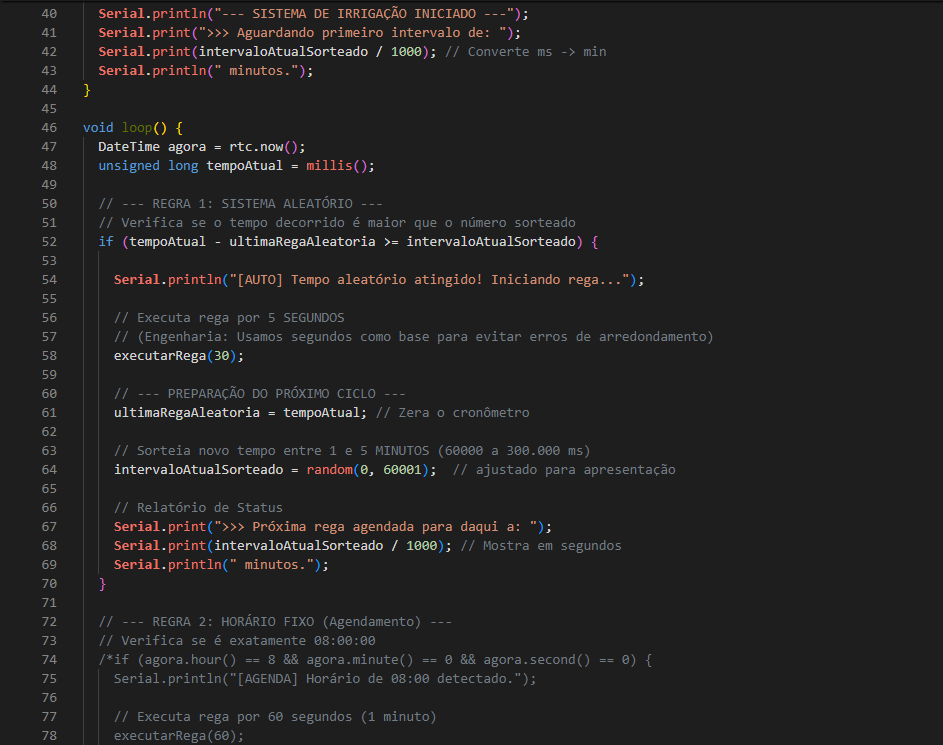
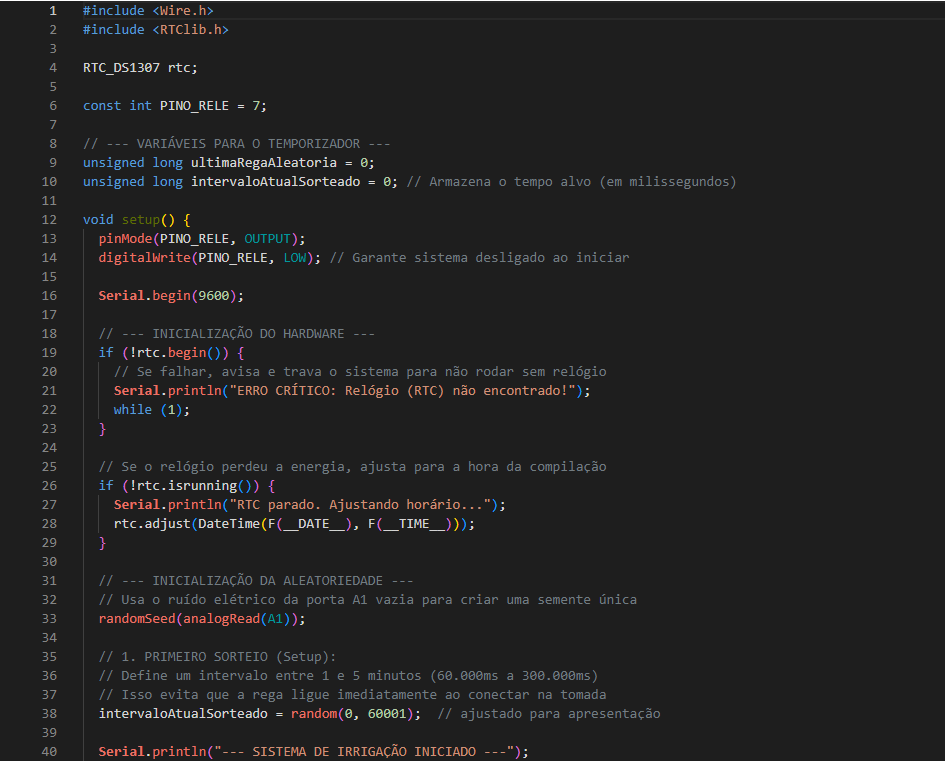
* Simplicidade para utilização.
* Robusto contra erros de hardware.
* Maior número de pinos 5V.
* Comunidade maior.
* Sem conectividade nativa a WIFI e Bluetooth.
* Mais lento, apenas 8 bits e baixo clock 16 MHz.
* Pouca memória disponível.

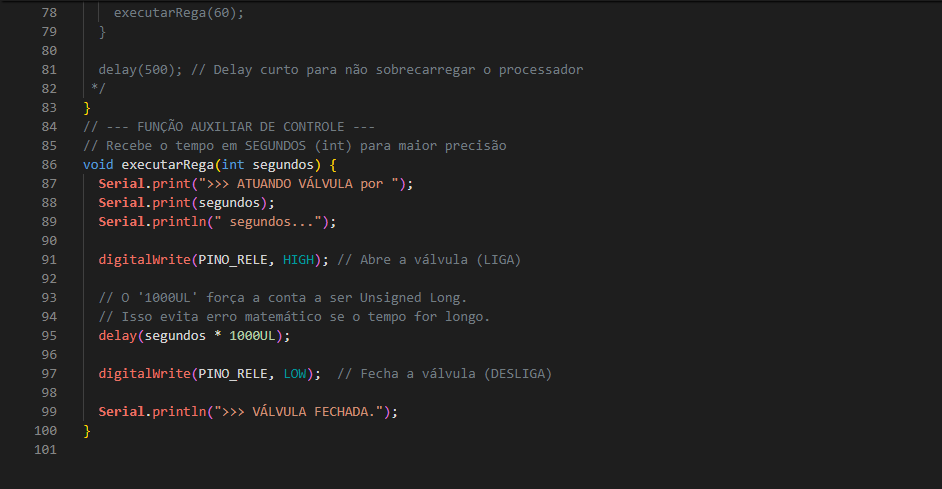
No primeiro protótipo do DLV1 encontramos algumas limitações para aplicação, pois, os materiais necessários para comunicação entre placas estava faltando, seja a placa de comunicação loRa, o RTC para utilização na placa arduino e a falta de um Step-Down para integração na placa ESP32.

Contudo, nossos trabalhos são dar continuidade com os estudos de avanço literário e tecnológico para o aprimoramento do projeto. No protótipo de conexão e leitura de dados, temos uma conexão simples via jumpers TX, RX. Essa conexão já permite a simulação do recebimento e envio de dados entre o microcontrolador e o dataloger, contando também com a tradução/conversão de dados em valores inteiros.

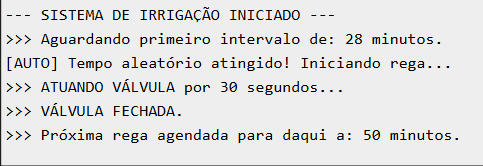
Para próximas aplicações, desejamos implementar a conexão via loRa para termos uma simulação mais próxima da realidade.

Ao final da semana, concluimos os dois protótipos iniciais do DLV1, o primeiro focao nos intervalos aleatórios de irrigação, seguindo com o código referente:





Logo abaixo temos o Output do código:



Em resumo, o código inicia escolhendo um valor aleatório para o intervalo de espera inicial, passado esse intervalo, novamente o algoritimo escolhe um valor aleatório, mas dessa vez para o tempo de irrigação. Após esse intervalo chegar ao fim, o algoritimo programa a próxima rega e assim sucessivamente.

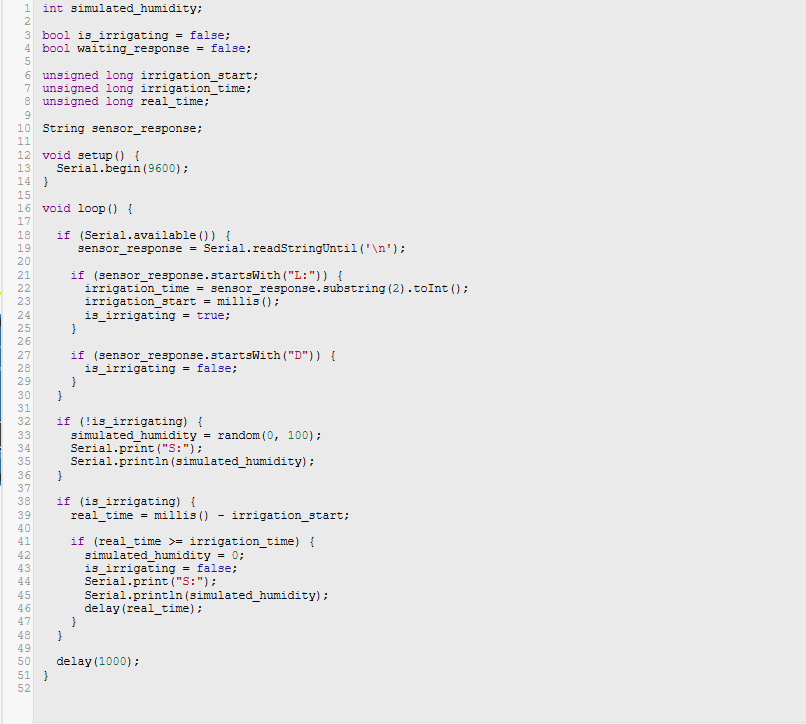
Já o segundo focado na simulação de comunicação (por enquanto via conexão física) e o tratamento de dados, com um arduino recebendo valores e irrigando, enquanto o outro envia valores aleatórios com intervalos aleatórios de tempo. O primeiro arduino recebendo esses dados, faz um pequeno cálculo e retorna o tempo necessário para a irrigação, o segundo arduino recebendo essa informação espera o proximo sinal para continuar o código.

Em seguida uma imagem do código referente ao primeiro arduino:

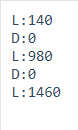
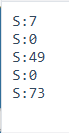


OBS: O tempo de irrigação na linha 22, esta sendo multiplicado por 20, apenas para consumir menos tempo durante os teste, tendo em vista que a vazão selecionada como base seja 1L/s, o certo seria multiplicar por 1000, já que trabalhamos com milisigundos.

Logo após, uma imagem do código referente ao segundo arduino:



Em seguida as telas de Output de cada código respectivamente:

Iniciando da direita para a esquerda, o arduino 2 seleciona um numero aleatório, nesse caso o 7, e envia para o arduino 1, que calcula o tempo necessario em segundos e envia junto de um sinal "L" indicando que a bomba deve ser ligada.

O arduino 2 recebe esses valores, aguarda os 140 ms e retorna um sinal "0", indicando que não necessita mais da irrigação. Chegando no arduino 1 ele reconhece que não precisa continuar com a irrigação e retorna um sinal "D", encerrando a irrigação.

Após esses passos, o arduino 2 fica um intervalo aleatório em "repouso", e logo em seguida novamente seleciona um numero aleatório, iniciando o ciclo novamente.