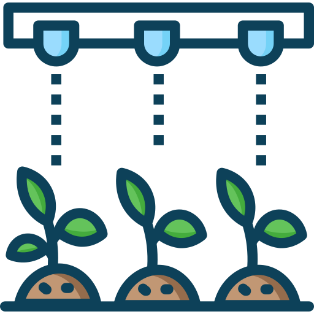
**ROTEIRO DO RELATO INDIVIDUAL**

SISTEMA DE IRRIGAÇÃO AUTOMATIZADO - ARDUINO

Equipe:2

membros: LUCAS, ÍCARO, MARCILIO, NICHOLAS, FELIPE ROCHA

ESTACIO

Professor: Abraão | Aplic. de Cloud, Iot e Indústria 4.0 em Python | 28/11/2024

**Contexto e Objetivos**

O projeto consiste no desenvolvimento de um sistema de irrigação automatizado utilizando Arduino para coleta de dados de umidade e controle de bomba d'água, e Python para o processamento e visualização dos dados. A irrigação automatizada é uma solução amplamente utilizada em sistemas de agricultura e jardinagem, onde a precisão e o monitoramento de umidade são essenciais para garantir a saúde das plantas, reduzindo desperdícios e otimizando o uso de recursos hídricos.

Neste projeto, o sistema monitoraria os níveis de umidade do solo e ligaria ou desligaria automaticamente a bomba d'água com base em um valor pré-definido. Além disso, o sistema é projetado para registrar os dados de umidade e status da bomba em um arquivo CSV para análise futura e para atualizar um arquivo HTML, oferecendo uma interface de monitoramento acessível. Ao final, um gráfico com os níveis de umidade é gerado para uma visão clara das variações ao longo do tempo.

**Participação no Projeto**

Planejamento e Implementação Técnica

Definição de Requisitos e Seleção de Componentes: Participei ativamente na definição dos requisitos do sistema, especificando componentes de hardware (Arduino, sensor de umidade, relé para a bomba d'água) e as bibliotecas necessárias para a comunicação serial e visualização de dados em Python.

**Desenvolvimento do Código em Python e Integração com Arduino**: Desenvolvi o código em Python para comunicação com o Arduino via porta serial, permitindo a coleta contínua de dados de umidade. Para melhorar a interface de usuário, criei uma página HTML que exibe os dados coletados em uma tabela organizada com estilo CSS. Também implementei uma função de geração de gráficos usando a biblioteca Matplotlib, que gera automaticamente um gráfico de umidade ao final do ciclo de monitoramento.

**Simulação Inicial e Teste de Comunicação:** Para simular o sistema antes de conectar o Arduino, criei um simulador que gerava dados de umidade aleatórios, permitindo testar o funcionamento do sistema e corrigir eventuais problemas de visualização e armazenamento de dados. Essa simulação foi removida posteriormente, substituída pela comunicação real com o Arduino.

**Configuração e Teste do Sistema em Ambiente de Produção**: Quando o Arduino estava disponível, executei o código em ambiente de produção, garantindo que os dados fossem coletados e armazenados corretamente. Testei a funcionalidade de controle da bomba para acioná-la apenas quando a umidade do solo estava abaixo do limite.

Aprendizados e Impacto do Projeto Esse projeto proporcionou uma experiência rica na integração de hardware e software, bem como na criação de uma interface visual para monitoramento. Foi uma oportunidade de trabalhar em um sistema de IoT prático, aplicando habilidades de programação em Python e C++ para Arduino.

Objetivos do Projeto de Sistema Automatizado de Irrigação com Arduino e Python

Automatizar a Irrigação com Controle por Umidade

Desenvolver um sistema que monitore os níveis de umidade do solo e acione ou desligue automaticamente uma bomba d'água, garantindo que o solo se mantenha na faixa ideal de umidade para o crescimento saudável das plantas.

**Registrar Dados para Análise e Otimização**

Criar um registro dos dados coletados (data, hora, nível de umidade e status da bomba) em um arquivo CSV, para análise de desempenho do sistema e avaliação dos níveis de umidade ao longo do tempo.

**Fornecer uma Interface Visual de Monitoramento**

Implementar uma página HTML que atualize os dados em tempo real, permitindo ao usuário visualizar o histórico de umidade e o status da bomba, oferecendo uma interface de fácil acesso e uso.

**Facilitar a Visualização de Tendências e Comportamento do Solo**

Gerar gráficos ao final de cada ciclo, mostrando as variações de umidade do solo ao longo do tempo para facilitar a interpretação dos dados, ajudando na identificação de padrões que auxiliem no ajuste dos parâmetros de irrigação.

**Desenvolver um Sistema Adaptável e de Fácil Manutenção**

Criar um sistema que possa ser ajustado para diferentes níveis de umidade desejados, sendo de fácil manutenção e expansão, atendendo a diferentes cenários e requisitos hídricos.

**Metodologia do Projeto de Sistema Automatizado de Irrigação com Arduino e Python**

Local e Período

Local: O projeto foi desenvolvido em um ambiente de laboratório, utilizando uma bancada para montagem de protótipos com Arduino e um computador para programar e monitorar o sistema. A parte de simulação e desenvolvimento do código Python foi realizada em uma máquina com acesso à rede, onde o arquivo de dados foi salvo em um diretório compartilhado.

Período: O projeto foi desenvolvido ao longo de algumas semanas, abrangendo desde a fase de planejamento e simulação inicial até os testes finais e ajustes em ambiente real com o Arduino conectado.

**Sujeitos e Públicos Envolvidos**

O projeto foi conduzido com um foco experimental, direcionado principalmente a profissionais de TI interessados na integração de sistemas IoT com Python e Arduino. Embora o projeto não envolvesse diretamente outros sujeitos, ele foi desenvolvido com o objetivo de servir como um modelo para estudantes e profissionais de agricultura ou jardinagem, oferecendo uma solução de irrigação prática e automatizada.

**Etapas do Projeto**

Planejamento e Levantamento de Requisitos

Identificou-se a necessidade de automatizar o sistema de irrigação e de monitorar o solo em tempo real. Foi definido o escopo do projeto, incluindo a especificação de componentes como o Arduino, o sensor de umidade, o módulo de relé para a bomba e as bibliotecas de software necessárias.

**Montagem e Configuração do Hardware**

Montou-se o circuito com o Arduino, sensor de umidade e relé, configurando a comunicação serial entre o Arduino e o computador. O sensor de umidade foi calibrado para garantir a precisão na leitura dos valores de umidade.

**Desenvolvimento do Código em Python e Simulação Inicial**

O código foi inicialmente desenvolvido com uma simulação de valores de umidade, utilizando a biblioteca random para gerar dados. Essa simulação permitiu testar o sistema de atualização do CSV e HTML, verificar a lógica de controle da bomba e ajustar a interface visual antes de integrar o Arduino real.

**Integração do Arduino e Testes com Dados Reais**

Após a validação com a simulação, a integração com o Arduino foi feita. Implementou-se o código para ler dados de umidade diretamente do Arduino via comunicação serial, verificando o funcionamento correto do controle da bomba com base nos valores de umidade lidos em tempo real.

**Desenvolvimento da Interface HTML e Geração de Gráficos**

Criou-se uma página HTML que exibe os dados de umidade e status da bomba em uma tabela organizada, com estilo CSS para uma visualização

**Resultados e Discussão**

Expectativas e Resultados Alcançados

O objetivo inicial era criar um sistema de irrigação automatizado que monitorasse a umidade do solo em tempo real, acionando a bomba de água quando necessário e registrando esses dados para análise futura. A expectativa era não só implementar essa funcionalidade básica, mas também adicionar uma interface acessível e intuitiva que permitisse monitorar e visualizar os dados de forma prática. Os resultados finais superaram essas expectativas, pois além do controle da bomba, foi possível integrar uma interface HTML que atualiza automaticamente os registros de umidade, e um gráfico que oferece uma visão clara das variações de umidade ao longo do tempo.

**Observações Durante o Desenvolvimento**

Simulação Inicial e Integração Real com Arduino: A simulação de dados no código Python permitiu realizar ajustes sem a necessidade do hardware, o que facilitou a identificação de problemas na atualização de dados e geração de gráficos antes de conectar o Arduino. Quando o sistema foi conectado ao Arduino, o funcionamento manteve-se consistente, validando a abordagem inicial.

**Interface Visual e Geração de Gráficos:** A interface HTML se mostrou eficiente para monitorar o status de irrigação. A geração de gráficos também facilitou a análise dos períodos de irrigação e a relação entre umidade e acionamento da bomba, contribuindo para ajustes de parâmetros e calibração do sensor.

**Reflexões Pessoais: Aprendizagens e Desafios**

Descobertas e Aprendizados: O projeto reforçou habilidades de integração de hardware e software, especialmente no uso de Python para coleta e visualização de dados em tempo real. A experiência também destacou a importância de criar uma interface visual, que melhora significativamente a compreensão dos dados e facilita o monitoramento e a manutenção do sistema.

**Facilidades e Dificuldades**: A simulação de dados foi uma grande vantagem inicial, permitindo realizar ajustes no código antes da etapa de testes com o Arduino. No entanto, desafios surgiram com a configuração precisa do sensor de umidade e o gerenciamento de atrasos e intervalos no controle da bomba, o que exigiu ajustes para evitar falsos positivos e garantir um acionamento estável da bomba.

**Recomendações**

Para futuros desenvolvimentos, seria interessante:

Implementar uma conexão com uma plataforma IoT para armazenar e visualizar dados de forma remota, acessível em tempo real a partir de qualquer dispositivo.

Integrar alertas automáticos, como notificações de baixa umidade, enviadas ao usuário para melhorar a eficácia do monitoramento e manutenção do sistema.

A experiência prática deste projeto de irrigação automatizada com Arduino e Python proporcionou uma compreensão profunda e integrada de vários conceitos teóricos, especialmente aqueles ligados à automação, IoT (Internet das Coisas) e análise de dados em tempo real.

**1. Automação e Controle de Sistemas**

Na teoria, a automação é abordada como uma forma de controlar processos com base em parâmetros definidos. Neste projeto, a teoria foi aplicada na prática ao programar o sistema para monitorar a umidade do solo e acionar automaticamente a bomba quando necessário. A lógica de controle embutida no código se baseou diretamente em conceitos de controle de sistemas, como o uso de condicionais para ligar ou desligar a bomba com base na leitura do sensor. Implementar e ajustar esse controle prático trouxe à tona a complexidade de criar sistemas que reagem de maneira precisa e confiável a variáveis ambientais dinâmicas.

**2. Conceitos de IoT e Comunicação entre Dispositivos**

O projeto também destacou o conceito de IoT, onde dispositivos coletam e trocam dados, criando uma rede de dispositivos interconectados. O Arduino e o sistema Python interagiam por meio da comunicação serial, prática que ilustra a teoria de redes de dispositivos e transmissão de dados em tempo real. A experiência trouxe à tona os desafios de manter essa comunicação estável e precisa, permitindo o funcionamento conjunto do hardware e do software em sincronia para realizar as tarefas de controle e monitoramento.

**3. Programação e Manipulação de Dados**

Na teoria, Python é frequentemente apresentado como uma linguagem ideal para processamento de dados. No projeto, essa capacidade foi fundamental para:

* Registrar dados de umidade e status da bomba em um arquivo CSV.
* Atualizar e formatar automaticamente esses dados em uma interface HTML.
* Gerar gráficos que visualizam variações de umidade ao longo do tempo, aplicando os conceitos teóricos de análise e interpretação de dados.

Trabalhar diretamente com dados permitiu explorar como técnicas de manipulação e visualização, vistas em teoria, são aplicadas para melhorar a interpretação de informações e a tomada de decisões.

**4. Desafios Práticos e Calibração de Sensores**

Enquanto a teoria sobre sensores aborda a importância de calibração para dados confiáveis, o projeto revelou os desafios práticos dessa tarefa. Ajustar o sensor de umidade para que ele interpretasse corretamente o nível do solo exigiu paciência e experimentação para atingir um equilíbrio entre a sensibilidade do sensor e o comportamento esperado da bomba. Esse processo de calibração trouxe insights sobre como dados de sensores, se mal interpretados, podem afetar a confiabilidade de sistemas automatizados.

**5. Importância da Interface Visual e da Usabilidade**

A teoria muitas vezes menciona a importância da interface do usuário, mas desenvolver uma página HTML funcional com CSS, que exibe dados de forma organizada e clara, tornou real essa importância. A experiência prática mostrou que uma interface bem desenhada facilita a interação e a compreensão do sistema, reduzindo o tempo necessário para o monitoramento e tornando o sistema mais acessível para os usuários.

**Conclusão**

Em resumo, a prática deste projeto reforçou como a teoria e a prática se complementam. A implementação trouxe uma compreensão mais sólida sobre automação, IoT, e manipulação de dados, enquanto os desafios e ajustes necessários fortaleceram habilidades de resolução de problemas e inovação. Trabalhar com a teoria na prática evidenciou que, embora os conceitos sejam fundamentais, é o refinamento em situações reais que torna a tecnologia aplicável e eficaz no mundo prático.