

Microcontroladores

Tem o papel fundamental de receber os dados vindos do sensor, interpretá-lo e enviar o resultado para o display;

Microcontroladores são amplamente utilizados em produtos e dispositivos automatizados, como sistemas de controle automotivo, dispositivos médicos implantáveis, controles remotos, máquinas de escritório, eletrodomésticos, ferramentas elétricas, brinquedos e diversos outros sistemas embarcados.

Entre eles temos alguns exemplos abaixo:

Arduino Uno

É uma plataforma que possibilita o desenvolvimento de projetos eletrônicos;

É usado para o controle de portas, motores, brinquedos, entre outros objetos e equipamentos;

O Arduino Uno coleta os dados de cada componente e os envia para a ESP32, que por sua vez encaminha esses dados recebidos para o banco de dados.

Especificações:

O Arduino Uno possui voltagem de 5V, possui 14 pinos de I/O, dos quais 6 possuem capacidade para PWM e 6 entradas analógicas.

- A tensão de operação é 5V
- A tensão de entrada recomendada varia de 7V a 12V
- A tensão de entrada varia de 6V a 20V
- 14 pinos de entrada/saída digital
- 6 pinos analógicos
- A corrente DC para cada pino de entrada/saída é de 40 mA
- A corrente DC para o pino de 3,3 V é 50 mA
- A memória flash é de 32 KB
- SRAM é 2KB
- EEPROM é 1 KB
- A velocidade do CLK é de 16 MHz

ESP32

É uma placa de desenvolvimento de hardware de código aberto, baseada em um processador dual-core de 32 bits, com 520 KB de memória flash.

A ESP32 é utilizada para controlar diversos componentes eletrônicos, como sensores, displays, motores, entre outros.

No contexto atual, a ESP32 está sendo utilizada para coletar os dados recebidos do Arduino e enviá-los para o banco de dados.

Especificações:

A ESP32 conta com cinco modos de consumo de energia: Ativo, Modem Sleep, Light Sleep, Deep Sleep e Hibernate.

Abaixo outras características:

- Processador dual-core de 32 bits
- 520 KB de memória flash
- Wi-Fi e Bluetooth integrados
- Suporte a interfaces para câmera, UART, SPI, I2C, I2S, ADC, DAC, PWM
- Arquitetura Dual-Core, que permite executar várias tarefas simultaneamente
- Baixo consumo de energia
- Clock máximo de 240 MHz
- Wireless padrão 802.11 b/g/n
- Conexão Wifi 2.4Ghz (máximo de 150 Mbps)
- Antena embutida
- Conector micro-usb
- Bluetooth BLE 4.2

Sensoriamento

Sensor DSB18B20:

O sensor de temperatura DS18B20 é um termômetro digital que fornece leituras de temperatura de forma precisa e com apenas um fio de comunicação.

Especificações:

- Modelo: DSB18B20;
- Tensão de operação: 3-5,5V
- Faixa de medição: -55°C a +125°C
- Precisão: $\pm 0.5^{\circ}\text{C}$ entre -10°C e +85°C
- Possui: Ponta de aço inoxidável
- Dimensão ponta de aço: 6 x 50mm

Sensor ST100:

O sensor de turbidez da água é formado por um par de sensores infravermelhos, um emissor e um receptor. Conforme a quantidade de luz recebida pelo receptor, a tensão de saída do sensor varia: ela aumenta ou diminui, sendo que a turbidez é inversamente proporcional à tensão de saída. Ou seja, quanto menor for a tensão de saída, maior será a turbidez da água.

Especificações:

- Modelo: ST100;
- Chip Board: MPC6021
- Tensão: 5VDC;
- Corrente: 30mA (MAX);
- Tempo de resposta: <500ms;

- Resistência de isolamento: 100MΩ (Min);
- Saída: Analógica (0-4.5V) ou Digital (alto 5V / baixo 0V);
- Temperatura de operação: -30°C a 80°C;
- Comprimento do jumper: 20cm;
- Dimensões do sensor (Cx D): ~34x30mm;
- Dimensões do módulo (Cx L x E): ~30x20x9mm;
- Peso: 15g.

Sensor PH4502C:

A sonda de vidro do sensor é projetada para medir a concentração de íons de hidrogênio (H⁺) em soluções aquosas, garantindo leituras precisas de pH.

Especificações:

- Modelo: PH-4502C;
- Marca: OEM;
- Tensão de aquecimento: 5 ±0.2V (AC/DC);
- Corrente de trabalho: 5-10mA;
- Faixa de temperatura: 0~60°C;
- Tempo de resposta: 5s;
- Tempo de sedimentação: 60s;
- Componente Potência: 0,5 W;
- Saída: Analógica;
- Faixa de medição: 0,00 ~ 14,00 pH;
- Zero pontos: 7 +- 0.5ph;
- Erro alcalino: 0.2pH;
- Temperatura de Operação: -10~50°C(Temperatura Nominal 20°C);
- Umidade de Operação: 95%RH (Umidade Nominal 65%RH);
- Vida Útil: 3 anos;
- Resistência interna: <250 MOhms;
- Comprimento do cabo: 90cm;
- Conector: Plugue BNC;
- Material: Termoplástico / Metal / Placa de Fenolite;
- Dimensões: 66mm Largura x 33mm Profundidade x 20mm Altura (Módulo);

- Tamanho: 160mm Largura x 26mm Profundidade x 26mm Altura (Sensor);
- Peso: 82g.

Sensor HC-SR04:

O sensor emite pulsos ultrassônicos quando o pino Trigger é ativado em nível alto por mais de 10 microsegundos. Esses pulsos são refletidos por obstáculos e retornam ao módulo, fazendo com que o pino Echo permaneça em nível alto.

Especificações:

- Sensor de Distância Ultrassônico
- Modelo: HC-SR04 ([Datasheet](#))
- Tensão de operação: 5 V
- Corrente de operação: 15 mA
- Ângulo de detecção: 15 graus
- Distância de medição: 2 cm a 4 m
- Precisão: ± 3 mm
- Frequência de operação: 40 kHz

Ciclo PDCA

O PDCA é uma metodologia de melhoria contínua aplicada à gestão empresarial, estruturada em quatro fases: Planejar (Plan), Executar (Do), Verificar (Check) e Agir (Action). Seu principal objetivo é promover o aprimoramento contínuo de processos e produtos, garantindo um ciclo constante e ininterrupto de melhorias.

O ciclo PDCA é usado para:

- Organizar e direcionar a criação de planos de ação
- Melhorar a gestão de processos, tornando-os mais eficientes e claros

- Controlar e obter resultados eficazes e confiáveis nas atividades de uma organização
- Padronizar as informações do controle da qualidade
- Evitar erros lógicos nas análises
- Tornar as informações mais fáceis de entender

No projeto isso significa que nos testamos se der algum erro a gente reve os códigos encontra o erro.

PMBOK

É um guia que serve para orientar profissionais na gestão de projetos, independentemente do setor em que seja aplicado.

O PMBOK pode ser usado para:

- Padronizar e difundir as melhores práticas de gestão de projetos
- Aumentar a eficiência das atividades realizadas
- Melhorar a gestão de custos
- Conduzir os gestores e as equipes pelas principais fases de cada projeto

Código:

```
// Inicializa os sensores
sensors.begin();

// Configuração do sensor ultrassônico
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
}

void loop() {
    // Lê os dados dos sensores
    float temperature = readTemperature();
    float pHValue = readPH();
    float turbidity = readTurbidity();
    float distance = readUltrasonicDistance();

    // Cria o JSON para transmitir
    String jsonData = createJSON(temperature, pHValue, turbidity, distance);

    // Envia os dados para o ESP32
    espSerial.println(jsonData);
    Serial.println(jsonData);

    // Aguarda 10 segundos antes de enviar novamente
    delay(10000);
}
```

Aqui está o procedimento resumido do código acima:

- 1. Inicialização dos sensores:** Os sensores são ativados e os pinos do sensor ultrassônico são configurados (um para envio do sinal e outro para recebimento).
- 2. Leitura de dados:** No loop principal, os valores de temperatura, pH, turbidez e distância são coletados a partir de suas respectivas funções de leitura.

3. **Criação de um JSON:** Um objeto JSON contendo esses valores é gerado para transmitir os dados de forma estruturada.

4. **Envio dos dados:** Os dados são enviados via comunicação serial para um ESP32 e também exibidos no monitor serial.

5. **Intervalo:** O processo é repetido a cada 10 segundos.

Código:

```
// Função para ler a temperatura
float readTemperature() {
    sensors.requestTemperatures();
    return sensors.getTempCByIndex(0);
}

// Função para ler o pH
float readPH() {
    // Lê o valor do sensor de pH no pino A0
    int sensorValue = analogRead(phPin);

    // Converte o valor analógico (0 a 1023) para pH
    float voltage = sensorValue * (5.0 / 1023.0); // 5V referência
    // Ajuste a fórmula conforme necessário para seu sensor de pH
    float phValue = (3.5 * voltage); // Exemplo de conversão
    return phValue;
}

// Função para ler a turbidez
float readTurbidity() {
    // Lê o valor do sensor de turbidez no pino A1
    int sensorValue = analogRead(turbidityPin);

    // Converte o valor analógico (0 a 1023) para tensão
    float voltage = sensorValue * (5.0 / 1023.0); // Converte para volts (assumindo 5V de referência)

    // Calcula o valor de turbidez baseado na tensão lida
    float turbidityValue;
    if (voltage >= 2.5) {
        turbidityValue = 3000.0 - (1120.0 * voltage); // Para voltagens maiores que 2.5V
    } else {
        turbidityValue = 2200.0 - (1100.0 * voltage); // Para voltagens menores que 2.5V
    }

    return turbidityValue;
}
```


Resumo do código acima:

1. **Leitura de Temperatura:** A função ``readTemperature()`` solicita e retorna a temperatura em graus Celsius usando um sensor de temperatura.
2. **Leitura de pH:** A função ``readPH()`` lê um valor analógico do sensor de pH e converte esse valor para um nível de pH usando uma fórmula baseada na tensão.
3. **Leitura de Turbidez:** A função ``readTurbidity()`` lê o valor analógico do sensor de turbidez e converte a leitura em tensão. Dependendo da tensão, a turbidez é calculada com diferentes fórmulas.
4. **Leitura de Distância (Ultrassônico):** A função ``readUltrasonicDistance()`` mede a distância usando um sensor ultrassônico, calculando a distância com base no tempo que o som leva para viajar até o objeto e voltar.

Citação:

<https://www.makerhero.com/produto/sensor-de-distancia-ultrassonico-hc-sr04/>

[https://www.usinainfo.com.br/outros-sensores-arduino/modulo-sensor-de-ph-bnc-ph4/502c-sem-sonda-8023.html#:~:text=%2D%20Tempo%20de%20Estabilidade%3A%2060%20s,nominal%20de%2065%25%20RH\)%3B](https://www.usinainfo.com.br/outros-sensores-arduino/modulo-sensor-de-ph-bnc-ph4/502c-sem-sonda-8023.html#:~:text=%2D%20Tempo%20de%20Estabilidade%3A%2060%20s,nominal%20de%2065%25%20RH)%3B)

<https://www.usinainfo.com.br/outros-sensores-arduino/sensor-de-turbidez-arduino-st100-modulo-de-leitura-original-4539.html>

<https://www.eletrogate.com/sensor-de-temperatura-ds18b20-a-prova-dagua>

<https://www.sta-eletronica.com.br/artigos/arduinos/caracteristicas-principais-do-arduino-uno>