

# OPTA-PROG-0-10VTEMPERATURA

PT-BR

## Programação para medição de temperatura com transmissor PT100 para 0-10V com OPTA

### 1. Informações Gerais

- Versão / Revisão: Revisão 1
- Data de Desenvolvimento: 20/06/2023
- Autor: Daniel Arcos
- Linguagem: C++
- Plataforma / Hardware: Finder OPTA

### 2. Objetivo da Programação

Essa programação tem como objetivo ler um sinal 0-10V em uma das entradas do OPTA e converter para temperatura, posteriormente disponibilizando na Cloud da Arduino

### 3. Requisitos e Dependências

- OPTA
- Transmissor de temperatura PT100 para 0-10V

- Fonte de alimentação

## **Configuração na Arduino IoT Cloud:**

1. Crie uma Thing com três variáveis:

- `sensorValueA0` (float, READ\_WRITE) - Valor bruto do ADC
- `voltageA0` (float, READ\_WRITE) - Tensão calculada
- `voltageA1` (float, READ\_WRITE) - Temperatura convertida

## **4. Passo a Passo da Implementação**

### **Conexões Físicas:**

1. **Conexão do Transmissor PT100:**

- Sinal de saída 0-10V → Pino A6 do OPTA
- GND do transmissor → GND do Arduino
- Alimentação do transmissor → Fonte 24V DC (verificar especificação)

2. **Alimentação:**

- Conecte o OPTA à fonte de alimentação
- Alimente o transmissor PT100 conforme especificação

3. **Configuração do Transdutor:**

- Verifique a faixa de temperatura do transdutor (ex: 0-200°C)
- Confirme a saída linear 0-10V correspondente

### **Configuração do Software:**

1. **Arduino IoT Cloud:**

- Crie a Thing e gere o código base
- Adicione as variáveis especificadas

## 2. Configuração dos Pinos:

- Pino A6 configurado como entrada analógica

## 5. Código completo

Todo o arquivo da programação está disponível para download nos arquivos.

## Visão Geral

### Variáveis Globais:

```
float teste = 0;    // Variável temporária para testes
float voltteste = 0; // Variável temporária para conversão
```

### Função setup (Configuração Inicial):

```
void setup() {
  // Inicialização da comunicação serial
  Serial.begin(9600);
  delay(1500); // Espera pelo Serial Monitor

  // Configuração do pino analógico
  pinMode(A6, INPUT); // Pino A6 como entrada analógica

  // Inicialização Arduino IoT Cloud
  initProperties(); // Definido em thingProperties.h
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  // Configuração de debug
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}
```

### Função loop (Loop Principal):

```
void loop() {  
  ArduinoCloud.update(); // Atualiza conexão com a nuvem  
  
  // Leitura e processamento do sinal  
  sensorValueA0 = analogRead(A6); // Leitura bruta do ADC (0-4095)  
  voltageA0 = sensorValueA0 * (3.0 / 4095.0) / 0.3 + 0.827; // Conversão para tensão  
  voltteste = voltageA0 * 20; // Conversão para temperatura  
  voltageA1 = voltteste; // Atribuição à variável da nuvem  
  
  delay(5); // Pequeno delay entre leituras  
}
```

## Análise detalhada do processamento de Sinais

### Conversão ADC para Tensão:

```
voltageA0 = sensorValueA0 * (3.0 / 4095.0) / 0.3 + 0.827;
```

### Explicação passo a passo:

1. **sensorValueA0**: Valor lido do ADC (12 bits → 0-4095)
2. **3.0 / 4095.0**: Converte para tensão considerando referência 3V
  - $4095 = 2^{12} - 1$  (máximo valor do ADC de 12 bits)
  - 3.0V = tensão de referência assumida
3. **/ 0.3**: Considera divisor de tensão ou ganho do circuito
  - Provavelmente há um circuito condicionador com ganho  $1/0.3 \approx 3.33$
4. **+ 0.827**: Offset de calibração
  - Compensa tensão residual ou offset do circuito

## Conversão para Temperatura:

```
voltteste = voltageA0 * 20;
```

### Relação Linear Assumida:

- $0V = 0^{\circ}C$
  - $10V = 200^{\circ}C$
  - Portanto:  $1V = 20^{\circ}C$  ( $200^{\circ}C / 10V = 20^{\circ}C/V$ )
- 
- 

EN

# Programming for temperature measurement with a PT100 transmitter for 0-10V with OPTA

## 1. General Information

- Version / Revision: Revision 1
- Development Date: 06/20/2023
- Author: Daniel Arcos
- Language: C++
- Platform / Hardware: Finder OPTA

## 2. Programming Objective

This program aims to read a 0-10V signal from one of the OPTA inputs and convert it to temperature, later making it available on the Arduino Cloud.

### 3. Requirements and Dependencies

- OPTA
- PT100 temperature transmitter for 0-10V
- Power supply

#### Arduino IoT Cloud Setup:

1. Create a Thing with three variables:
  - `sensorValueA0` (float, READ\_WRITE) - Raw ADC value
  - `voltageA0` (float, READ\_WRITE) - Calculated voltage
  - `voltageA1` (float, READ\_WRITE) - Converted temperature

### Implementation Step-by-Step

#### Physical Connections:

1. **PT100 Transmitter Connection:**
  - 0-10V output signal → OPTA pin A6
  - Transmitter GND → Arduino GND
  - Transmitter power → 24V DC power supply (check specifications)
1. **Power Supply:**
  - Connect the OPTA to the power supply
  - Power the PT100 transmitter according to specifications
1. **Transducer Configuration:**
  - Check the transducer temperature range (e.g., 0-200°C)
  - Confirm the corresponding 0-10V linear output

#### Software Configuration:

1. **Arduino IoT Cloud:**
  - Create the Thing and generate the base code
  - Add the specified variables
1. **Pin Configuration:**

- Pin A6 configured as input Analog

## 5. Complete Code

The entire programming file is available for download in the archives.

### Overview

#### Global Variables:

```
float teste = 0;    // Variável temporária para testes
float voltteste = 0; // Variável temporária para conversão
```

#### Setup function (Initial Configuration):

```
void setup() {
  // Inicialização da comunicação serial
  Serial.begin(9600);
  delay(1500); // Espera pelo Serial Monitor

  // Configuração do pino analógico
  pinMode(A6, INPUT); // Pino A6 como entrada analógica

  // Inicialização Arduino IoT Cloud
  initProperties(); // Definido em thingProperties.h
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);

  // Configuração de debug
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}
```

#### Loop function (Main Loop):

```
void loop() {
  ArduinoCloud.update(); // Atualiza conexão com a nuvem
}
```

```
// Leitura e processamento do sinal
sensorValueA0 = analogRead(A6); // Leitura bruta do ADC (0-4095)
voltageA0 = sensorValueA0 * (3.0 / 4095.0) / 0.3 + 0.827; // Conversão para tensão
voltteste = voltageA0 * 20; // Conversão para temperatura
voltageA1 = voltteste; // Atribuição à variável da nuvem

delay(5); // Pequeno delay entre leituras
}
```

## Detailed Analysis of Signal Processing

### ADC to Voltage Conversion:

```
voltageA0 = sensorValueA0 * (3.0 / 4095.0) / 0.3 + 0.827;
```

### Step-by-step explanation:

1. **sensorValueA0**: Value read from the ADC (12 bits → 0-4095)
2. **3.0 / 4095.0**: Converts to voltage considering a 3V reference
  - $4095 = 2^{12} - 1$  (maximum 12-bit ADC value)
  - 3.0V = assumed reference voltage
1. **/ 0.3**: Considers voltage divider or circuit gain
  - There is probably a conditioning circuit with a gain of  $1/0.3 \approx 3.33$
1. **+ 0.827**: Calibration offset
  - Compensates for residual voltage or circuit offset

### Conversion to Temperature:

```
voltteste = voltageA0 * 20;
```

### Assumed Linear Relationship:

- $0V = 0^{\circ}C$



- $10V = 200^{\circ}C$
- Therefore:  $1V = 20^{\circ}C$  ( $200^{\circ}C / 10V = 20^{\circ}C/V$ )