Fatec
Osasco
Prefeito Hirant Sanazar

Automação Industrial

Projeto de Sensoreamento e Instrumentação por controle PID – Parte II

Profº Engº Hermom Leal, Msc.

Versão 2 – Outubro/2019

Fatec

Controladores
PID – Proporcional-Integral-Derivativo -
(Ajuste de Parâmetros)

HERMOM LEAL MOREIRA

• Sumário

1. Introdução;
2. Testando o Controlador - Teste – kp, ki e kd=0;
3. Teste P – tref=30, kp=50, ki e kd=0;
4. Teste P – tref=30, kp=5000, ki e kd=0;
5. Teste PID – tref=30, kp=50, ki=20 e kd=10;

Referência: Movimento, luz e som com Arduino e Raspebery Pi – Simon Monk. Editora NOVATEC – Série Make.

Introdução

- Este experimento tem como base a utilização de um código principal denominado ***/pid_thermostat.ino***.
- O microcontrolador utilizado é o Arduino, sendo que, o código principal realiza a leitura de temperatura do sensor DS18B20, através de uma biblioteca do dispositivo denominado (***DallasTemperature.h***), que tem a função de realizar a leitura de temperatura externa.
- Além disto, o Arduino trabalha com bibliotecas, e para este experimento foram instaladas as bibliotecas denominadas ***OneWire.h***, ***Arduino-Temperature-Control-Library.h e PIDLibrary.h***

• Introdução

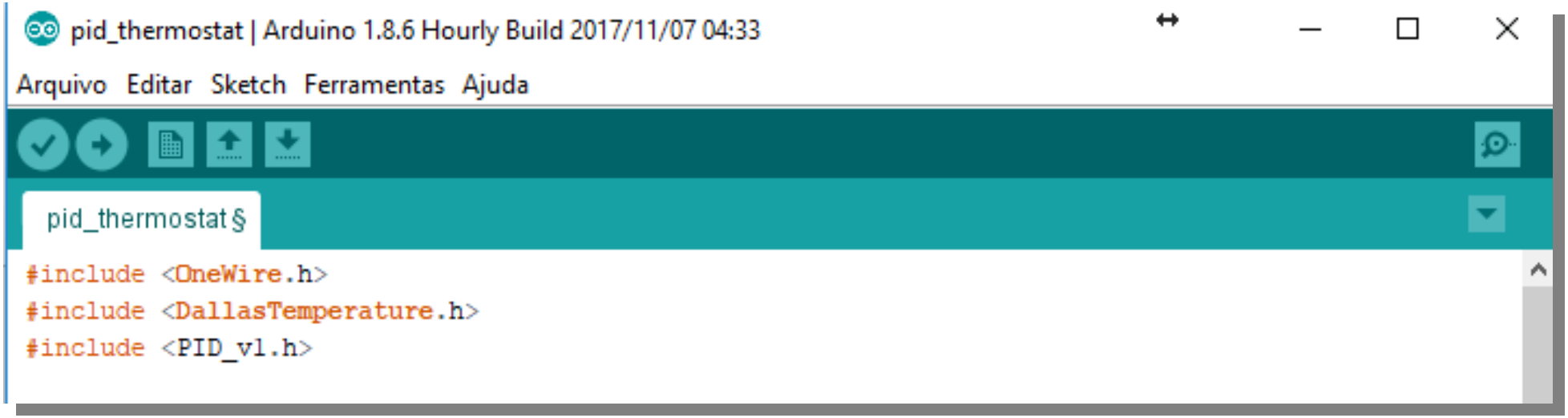


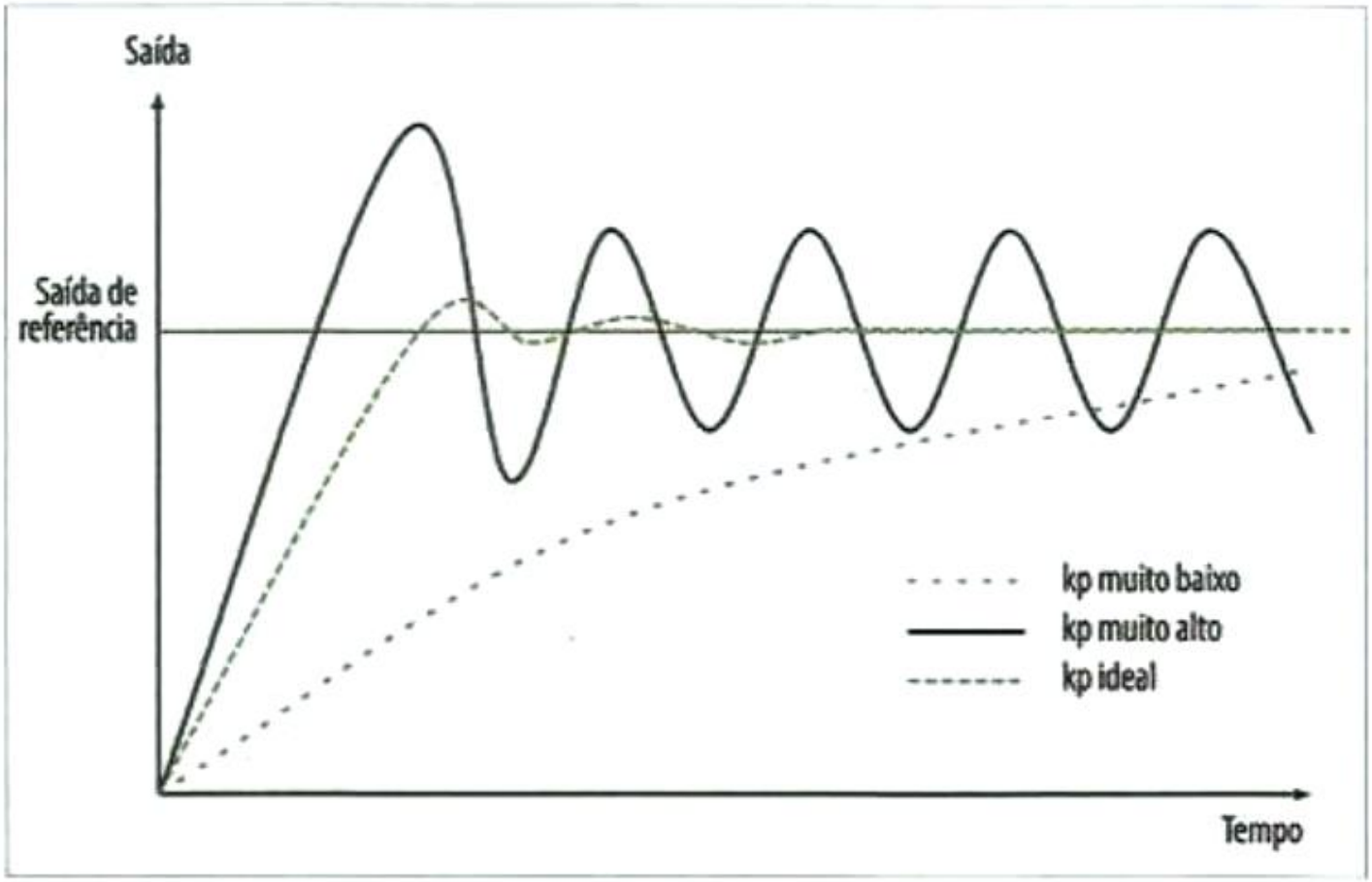
Figura 1. Bibliotecas Utilizadas

O objetivo do experimento é realizar o **teste do controlador PID, para a alteração da temperatura de referência e dos valores de kp, ki e kd.**

Logo após será feito o registro dos dados obtidos na saída a fim de obter bom controle na saída.

2. Testando o Controlador;

Ao ajustar o efeito de ganho kp (ganho) sobre a saída em um controlador proporcional, são obtidos o sinais conforme a Figura 2.



2. Testando o Controlador;

Quando os valores de **kp, ki e kd são iguais a (0)** o controlador não atua. Com isto, somente o sinal de temperatura recebido do sensor DS18B20, através da **entrada (Pin2)** do hardware será computada e consequentemente a **saída (Pin9)** não envia o sinal tipo PWM para a carga, ou seja, não ocorre a chamada retroalimentação conforme a Figura 3.

2. Testando o Controlador;

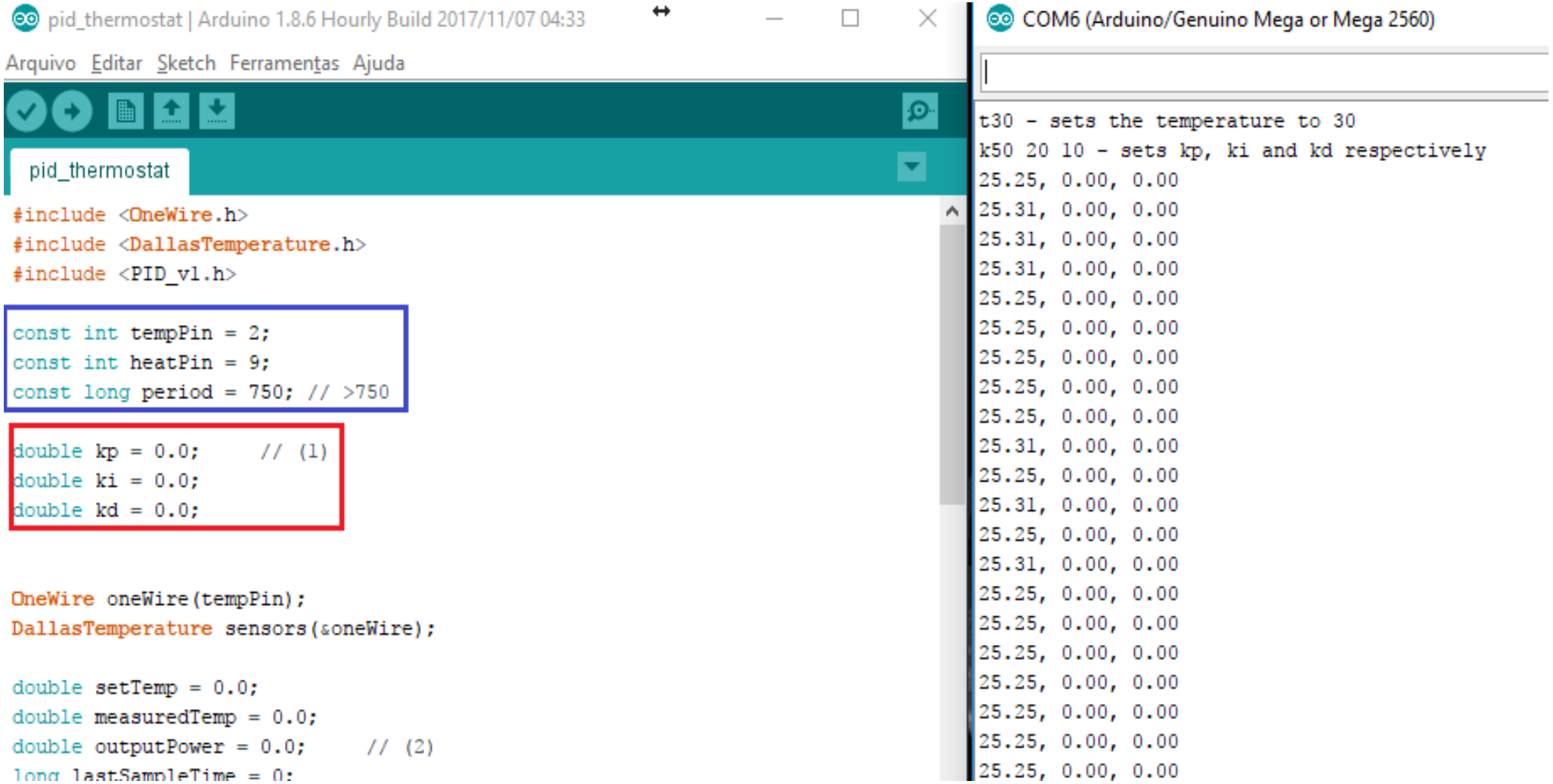


Figura 3 – Dados de entrada (Input)

- Ajuste *Default*

2. Testando o Controlador;

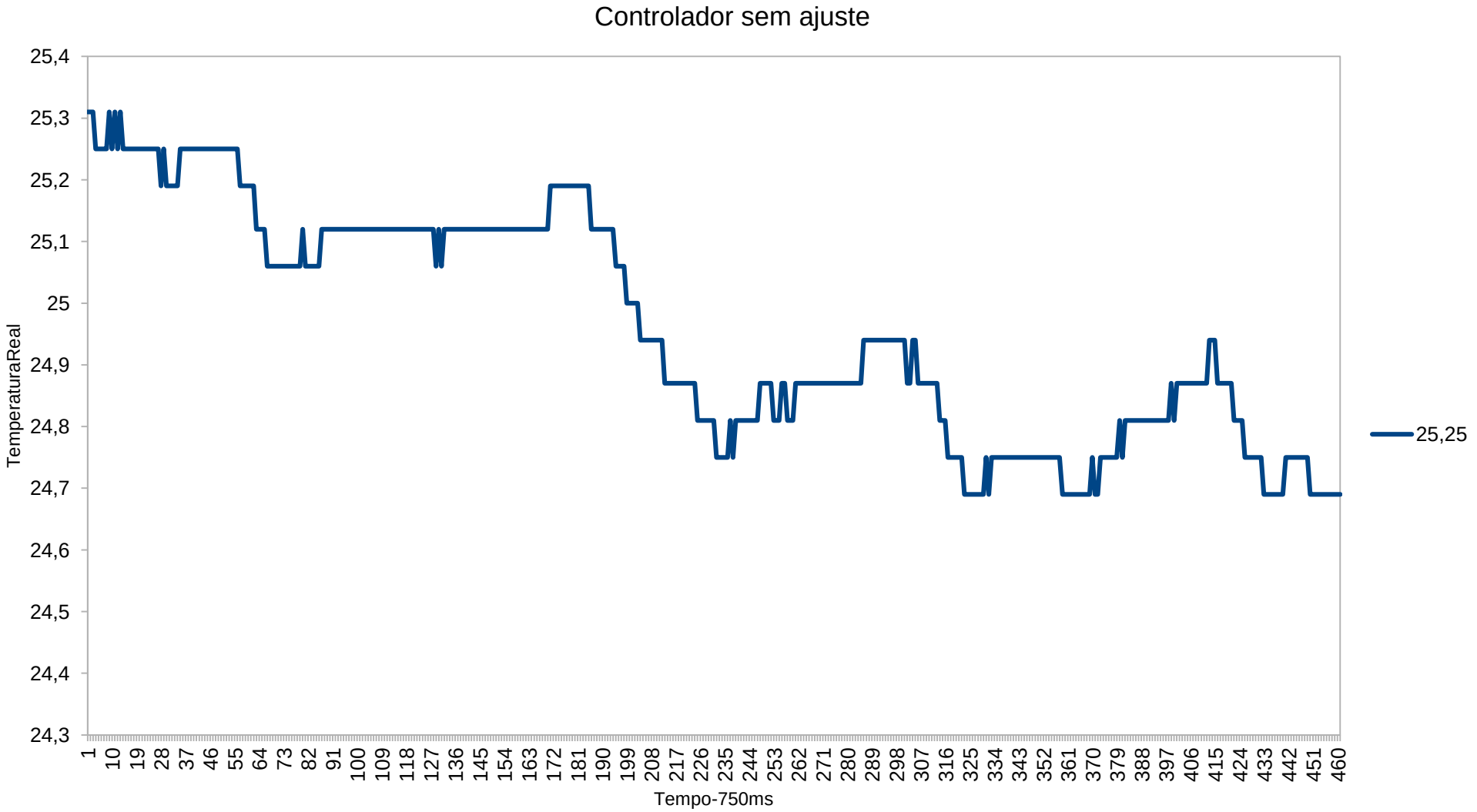


Figura 4 – Dados de saída (Output)

- Ajuste *Default*

3. Teste P – tref=30, kp=50;

Nesta etapa são feitos ajuste no valor de temperatura de referência (tref=30), no valor de kp (k=50). Com isto, é possível observar que a segunda coluna terá um valor fixo de 30 graus e terceira coluna terá um valor variando entre 0 e 255 referente ao sinal PWM, que buscará ajustar o valor da temperatura.

3. Teste P – tref=30, kp=50;

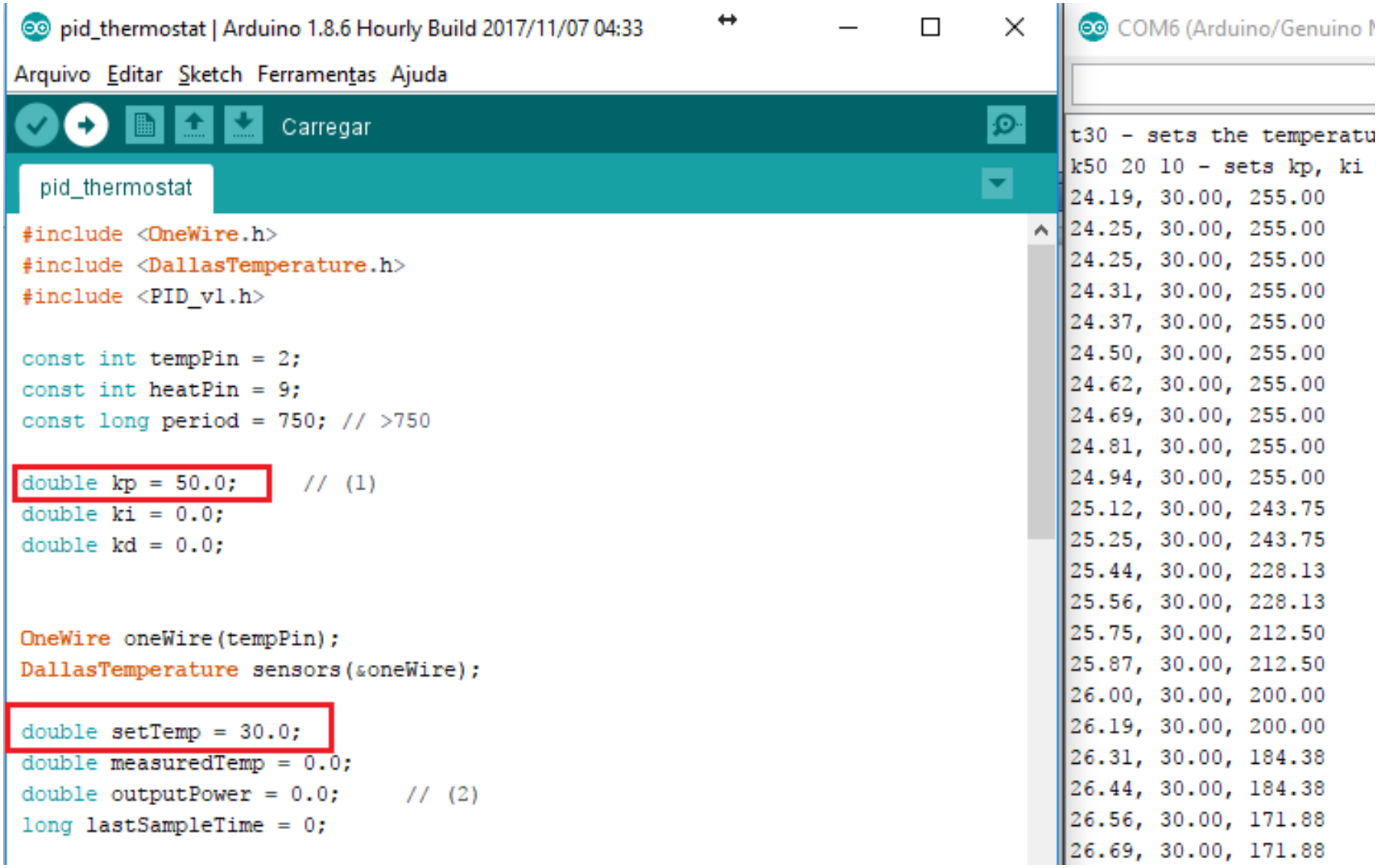


Figura 5 – Dados de entrada (Input)

- Ajuste tref=30, kp=50

3. Teste P – tref=30, kp=50;

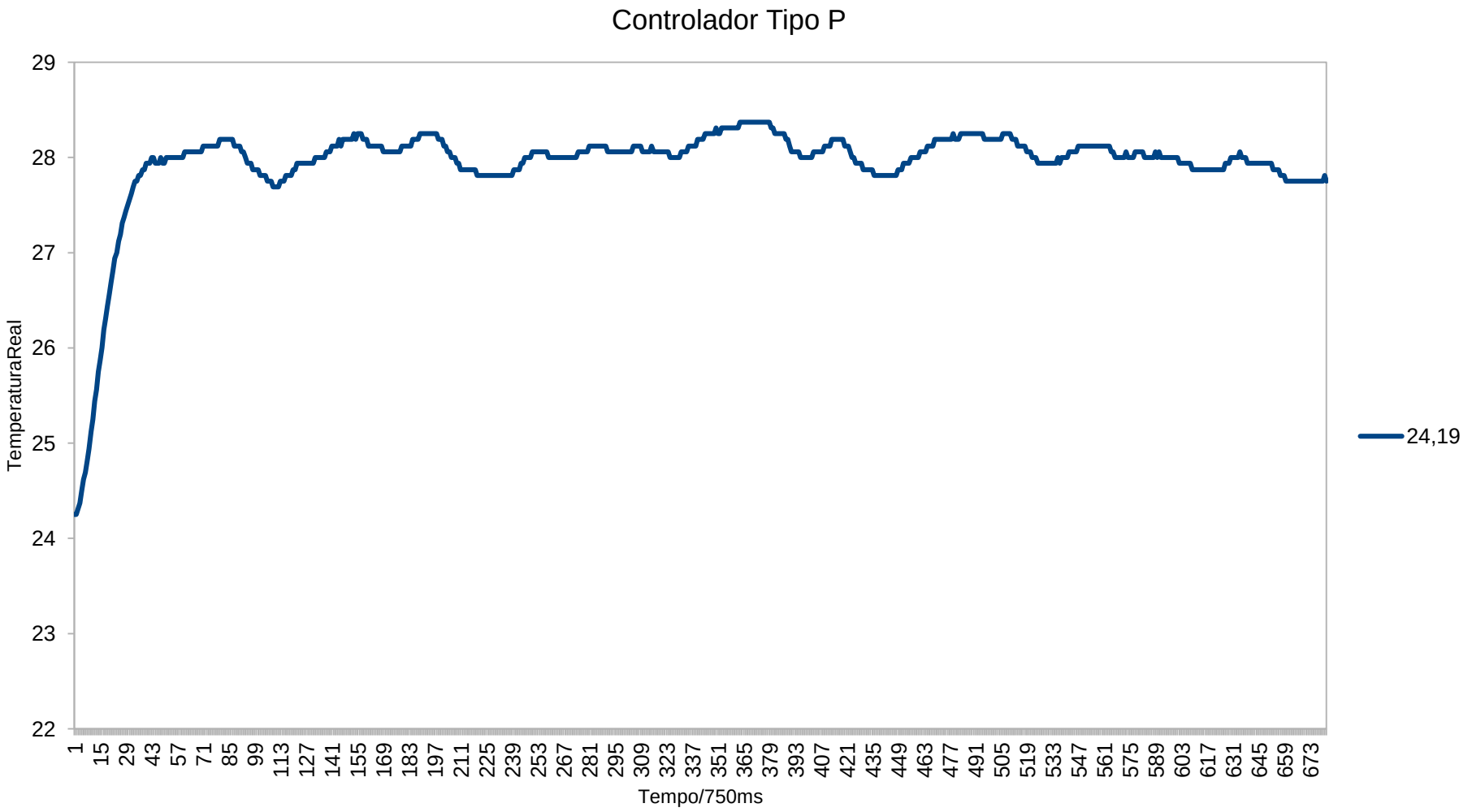


Figura 6 – Dados de saída (Output)

- Ajuste tref=30, kp=50, ki e kd=0

3. Teste P – tref=30, kp=50;

Análise:

4. Teste P – tref=30, kp=5000;

pid_thermostat | Arduino 1.8.6 Hourly Build 2017/11/07 04:33

Arquivo Editar Sketch Ferramentas Ajuda

pid_thermostat Slide 8

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <PID_v1.h>

const int tempPin = 2;
const int heatPin = 9;
const long period = 750; // >750

double kp = 5000.0;     // (1)
double ki = 0.0;
double kd = 0.0;


OneWire oneWire(tempPin);
DallasTemperature sensors(&oneWire);

double setTemp = 30.0;
double measuredTemp = 0.0;
double outputPower = 0.0;     // (2)
long lastSampleTime = 0;

PID myPID(&measuredTemp, &outputPower, &setTemp, kp, ki, kd, DIRECT); // (3)

void setup() {
  pinMode(heatPin, OUTPUT);
```

COM6 (Arduino/Genuino

```
29.56, 30.00, 255.00
29.56, 30.00, 255.00
29.56, 30.00, 255.00
29.62, 30.00, 255.00
29.62, 30.00, 255.00
29.69, 30.00, 255.00
29.69, 30.00, 255.00
29.75, 30.00, 255.00
29.75, 30.00, 255.00
29.81, 30.00, 255.00
29.87, 30.00, 255.00
29.94, 30.00, 255.00
30.00, 30.00, 255.00
30.00, 30.00, 0.00
30.06, 30.00, 0.00
30.12, 30.00, 0.00
30.12, 30.00, 0.00
30.12, 30.00, 0.00
30.12, 30.00, 0.00
30.12, 30.00, 0.00
30.06, 30.00, 0.00
30.00, 30.00, 0.00
29.94, 30.00, 0.00
29.87, 30.00, 255.00
29.81, 30.00, 255.00
29.75, 30.00, 255.00
29.69, 30.00, 255.00
29.69, 30.00, 255.00
```

Figura 7 – Dados de entrada (Input)

- Ajuste tref=30, kp=5000, ki e kd=0

4. Teste P – tref=30,kp=5000;

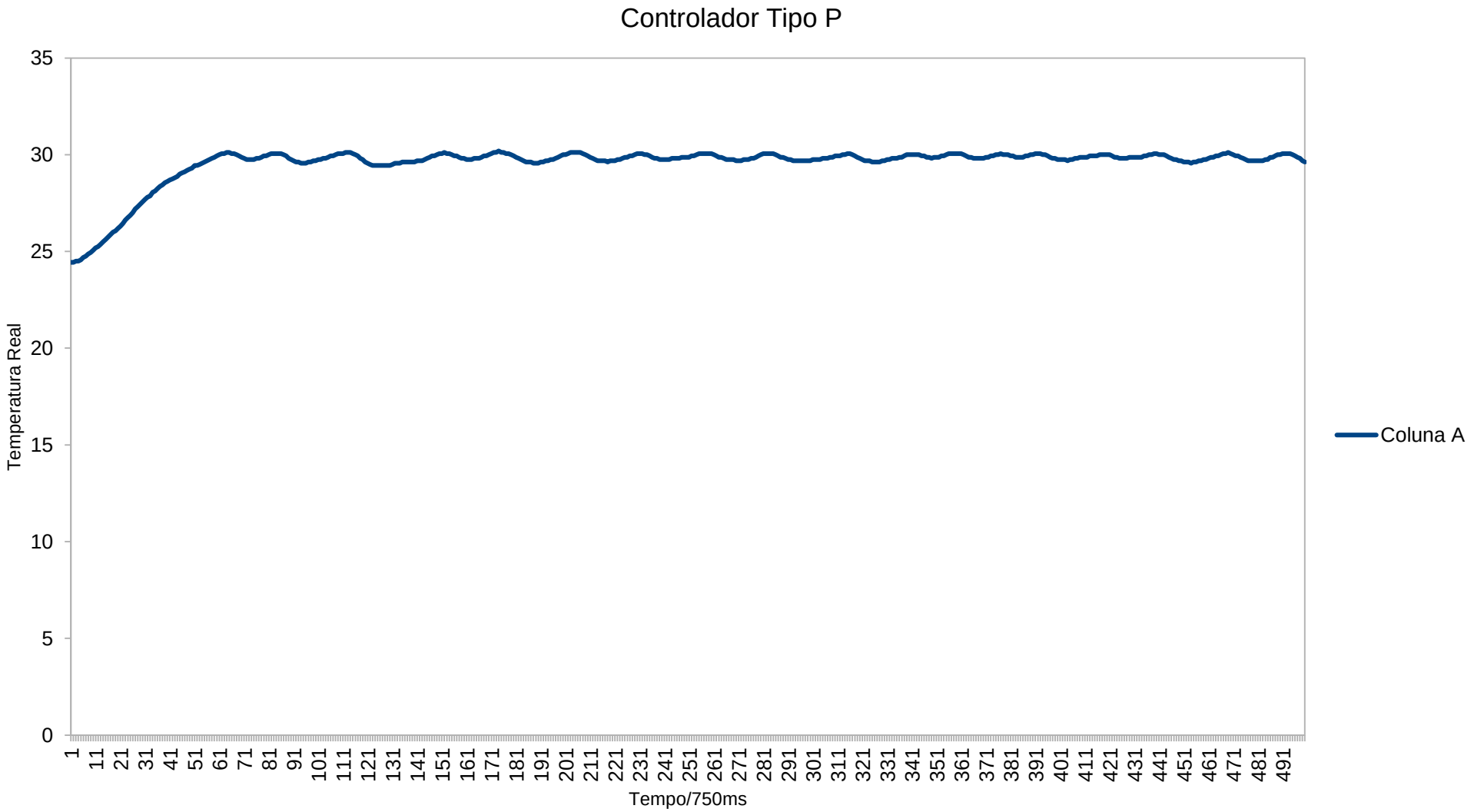


Figura 8 – Dados de saída (Output)

- Ajuste tref=30, kp=5000, ki e kd=0

4. Teste P – tref=30,kp=5000;

Análise:

5. Teste PID – tref=30,kp=50, ki=20, kd=10

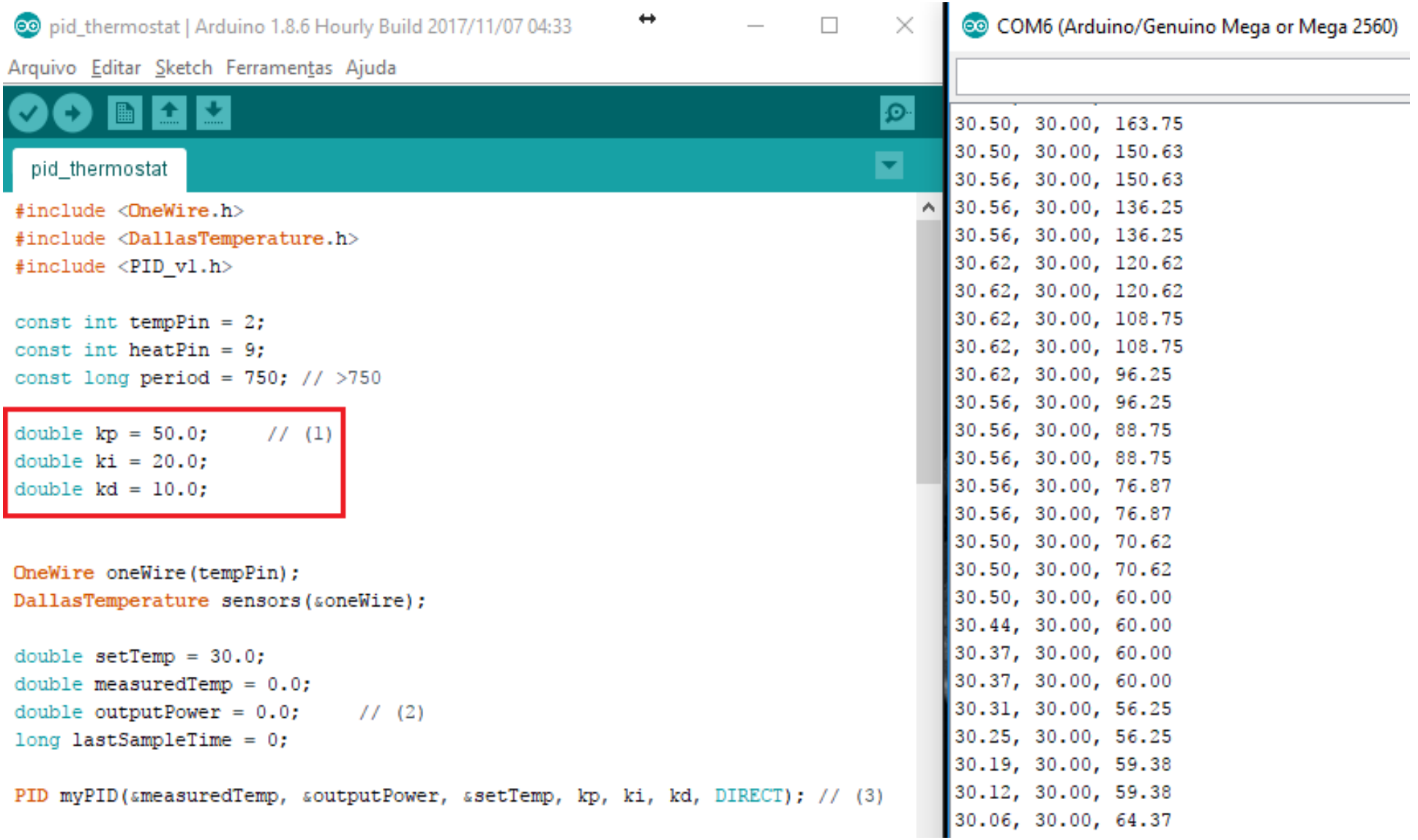


Figura 9 – Dados de entrada (Input)

- Ajuste tref=30, kp=50, ki=20 e kd=10.

5. Teste PID – tref=30,kp=50, ki=20, kd=10

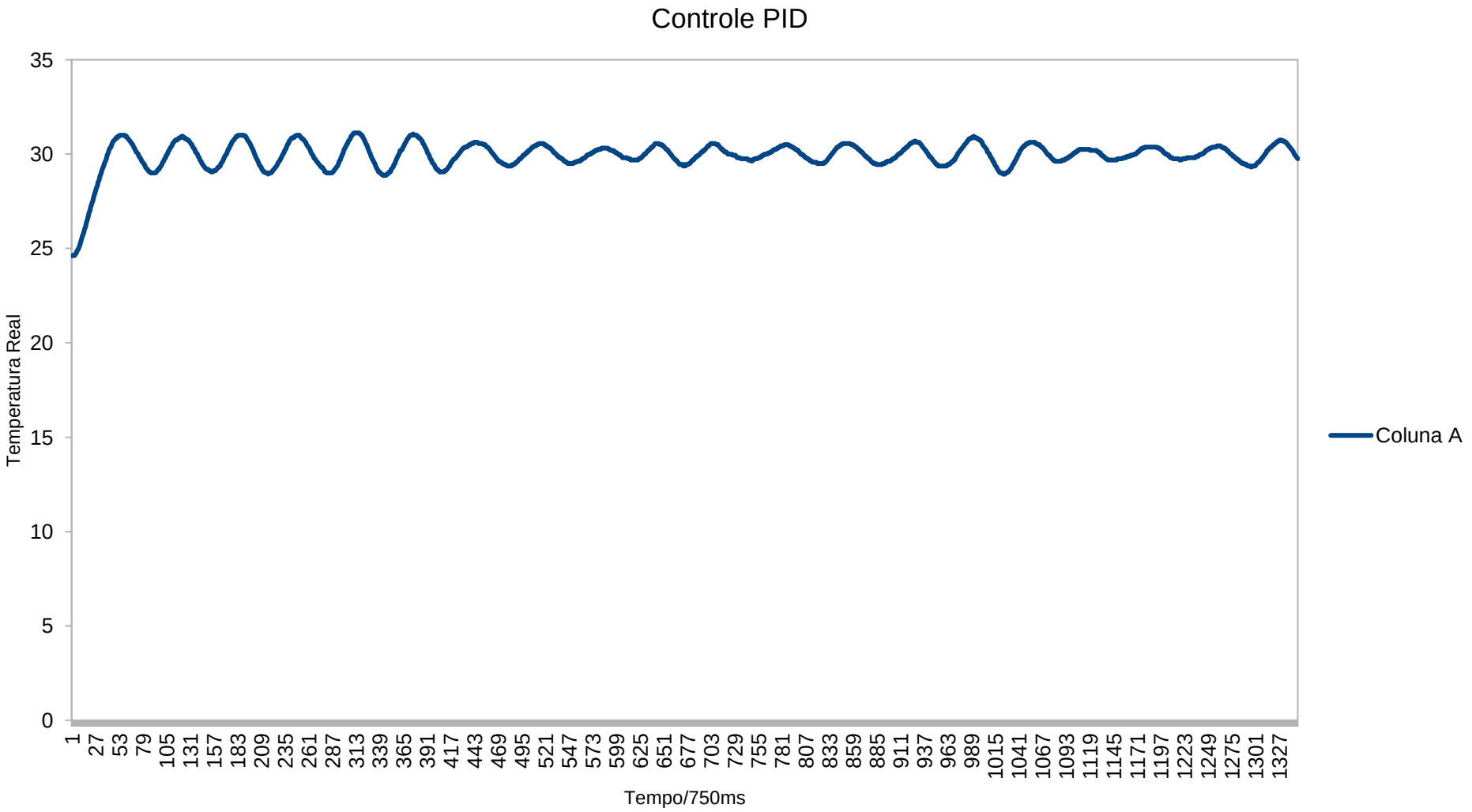


Figura 10 – Dados de saída (Output)

- Ajuste tref=30, kp=50, ki=20 e kd=10.

5. Teste PID – tref=30,kp=50, ki=20, kd=10

Análise: