Projeto Controlador de Temperatura

ES670 - Projeto de Sistemas Embarcados

Autores:

João Victor Evangelista Matoso	176293
Renato Pepe	176677

Campinas, 19 de junho de 2021

1. Resumo

Neste projeto, buscou-se implementar um sistema embarcado de controle de temperatura utilizando um resistor como elemento de aquecimento e um cooler para resfriamento.

O software foi desenvolvido visando o sistema embarcado presente nos laboratórios da Faculdade de Engenharia Mecânica da Unicamp. Este sistema é composto por um microcontrolador ARM Cortex-M0+ de 32 bits fornecido pela Freescale no seu kit de desenvolvimento KL25Z Kinetis KL2x. Esta placa está conectada ao kit McLab2 que fornece alguns periféricos como um elemento de aquecimento resistivo, sensor de temperatura, cooler para resfriamento, um display LCD, botões, entre outros. Mais detalhes sobre o kit estão listados na seção “Documentação”.

O objetivo do sistema desenvolvido neste projeto é controlar a temperatura medida pelo sensor do kit McLab2 utilizando o resistor e o cooler presentes. Este controle deve responder o mais rapidamente possível e o overshoot não deve ultrapassar 1°C. Para isto, foi desenvolvido um controlador PID para regular a temperatura, uma interface local que possibilite a operação pelo usuário e uma interface para configuração via comunicação UART.

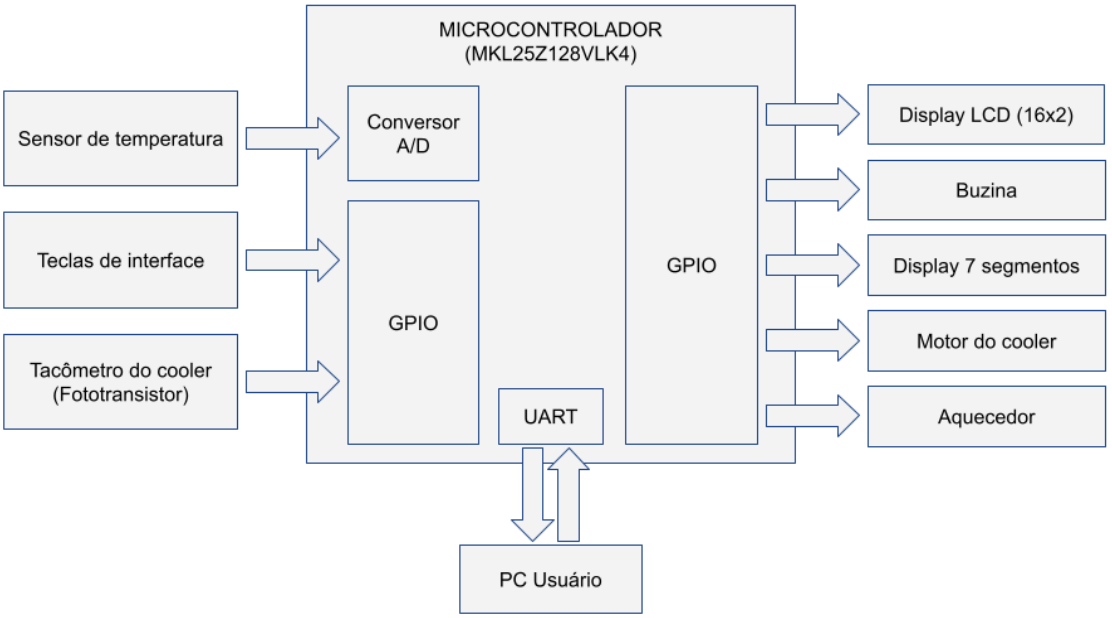
O sistema possibilita ao usuário a configuração da temperatura desejada dentro do intervalo da temperatura ambiente, considerada nesse projeto como 23ºC, até 74°C, assim como também possibilita a configuração de um timer para ligar ou desligar o controle de temperatura. Todas as configurações podem ser feitas tanto pela interface local, quanto pela comunicação UART, como mostrado na seção “Manual de utilização”.

2. Documentação

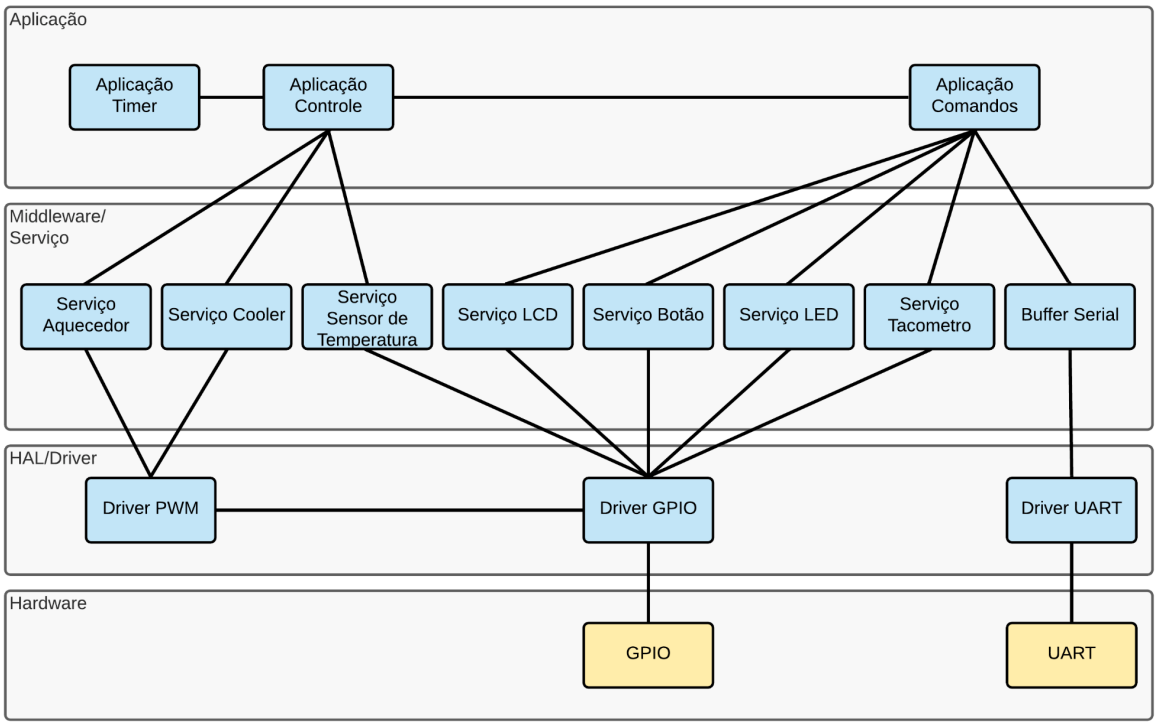
2.1. Requisitos do sistema

Requisitos funcionais	
RF1	Manter a temperatura medida em um valor determinado pelo usuário não ultrapassando a temperatura máxima de 74°C
RF2	Oferecer ao usuário a função de resfriamento rápido até a temperatura ambiente.
RF3	Possibilitar a programação agendada do aquecimento
Requisitos não funcionais	
NF1	Deve atingir a temperatura determinada pelo usuário o mais rápido possível
NF2	Deve fornecer uma interface para operação local
NF3	Deve controlar a temperatura com um overshoot máximo de 1°C
NF4	Deve exibir a temperatura atual do resistor
NF5	Deve controlar a temperatura através de uma interface UART
NF6	Deve alertar o usuário caso haja falha no sistema
NF7	Deve indicar através de um LED se está aquecendo ou resfriando
NF8	Deve controlar o duty cycle do aquecedor através de uma interface UART
NF9	Deve controlar o duty cycle do cooler através de uma interface UART

2.2. Diagrama de blocos do hardware



2.3. Diagrama de camadas



2.4. Especificações de Hardware

Freescale KL25Z Kinetis KL2x

- Microcontrolador: MKL25Z128VLK4

    - Núcleo ARM® Cortex™-M0+ de 32 bits
    - Frequência de operação 48MHz
    - 16KB de memória RAM
    - 128KB de memória FLASH
    - Core pipeline de dois estágios
    - Consumo de energia: 9.8uW/MHz
    - Interfaces de controle e comunicação:
        - USB (Host/Device)
        - SPI (x2)
        - I2C (x2)
        - UART (x3)
        - Real time clock (RTC)
        - PWM (Timer/PWM Module)
        - ADC (16 bit)
        - DAC (x1 12bit)
        - Touch Sensor
        - GPIO (x66)

- LED: CLV1A-FKB-CJ1M1F1BB7R4S3
    - Marca: Cree, Inc.
    - Cor: RGB
    - Corrente direta - If: 20 mA
    - Tensão direta - Vf (R,G,B): 2 V, 3.2 V, 3.2 V
    - Comprimento de onda (R,G,B): 624 nm, 540 nm, 480 nm
    - Ângulo de visão: 120°
    - Dimensões (CxLxA): 3.2 mm x 2.8 mm x 1.9 mm
    - Temperatura mínima de operação: - 40° C
    - Temperatura máxima de operação: +100° C
    - Tipo de montagem: SMD/SMT

Kit Mclab2

- Sensor temperatura
    - Modelo: Lm431
    - VO: 2.5 V
    - Reference voltage Adjustable
    - Initial accuracy (Max) (%) 0.5, 1, 2
    - Rating Catalog
    - VO adj (Min) (V) 2.5
    - VO adj (Max) (V) 36
    - Iz for regulation (Min) (uA) 400
    - Temp coeff (Max) (ppm/ degree C) 97
    - Operating temperature range (C) -40 to 85, 0 to 70
    - Iout/Iz (Max) (mA) 100

- Outros recursos:
    - LCD alfanumérico;
    - Displays de leds de 7 segmentos;
    - Teclas e leds;

- Buzzer;
- Memória serial EEPROM 24C04 (protocolo I²C);
- Comunicação serial RS232;
- Conversão A/D;
- Aquecedor;
- Ventilador;
- Tacômetro;
- Leitura de jumpers;
- Conector de expansão contendo 15 I/O's;
- Botão de reset manual;
- Gravação in-circuit compatível com McFlash+ ou ICD2BR

- Características:
  - 8K de memória de programa;
  - 368 bytes de memória de dados volátil (RAM);
  - 256 bytes de memória de dados não volátil (E²PROM);
  - 14 interrupções;
  - 33 I/O's;
  - 3 timers (2 de 8 bits, 1 de 16 bits);
  - 2 Capture/Compare/PWM;
  - USART;
  - MSSP (PSI e I²C);
  - PSP;
  - 8 canais de conversão A/D com 10 bits cada

2.5. Máquina de estados

Diagrama Geral

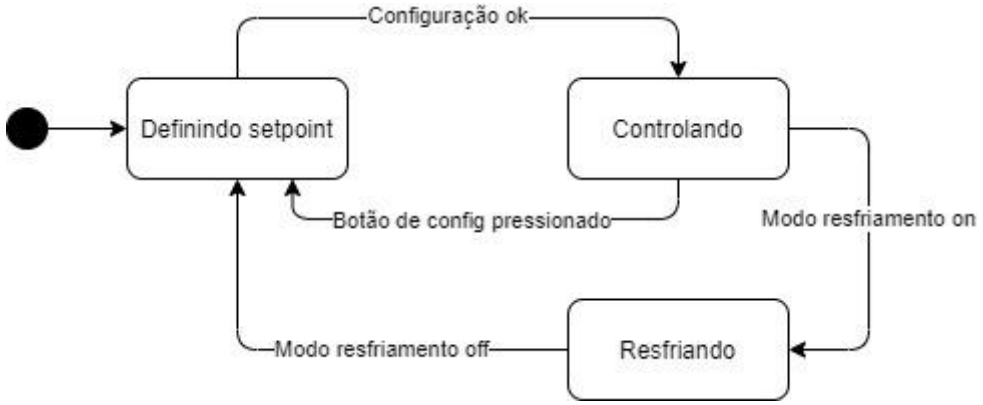
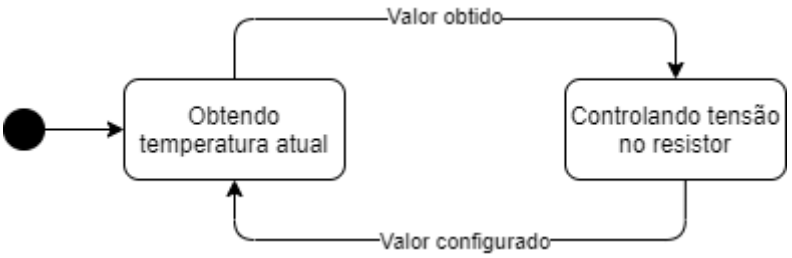
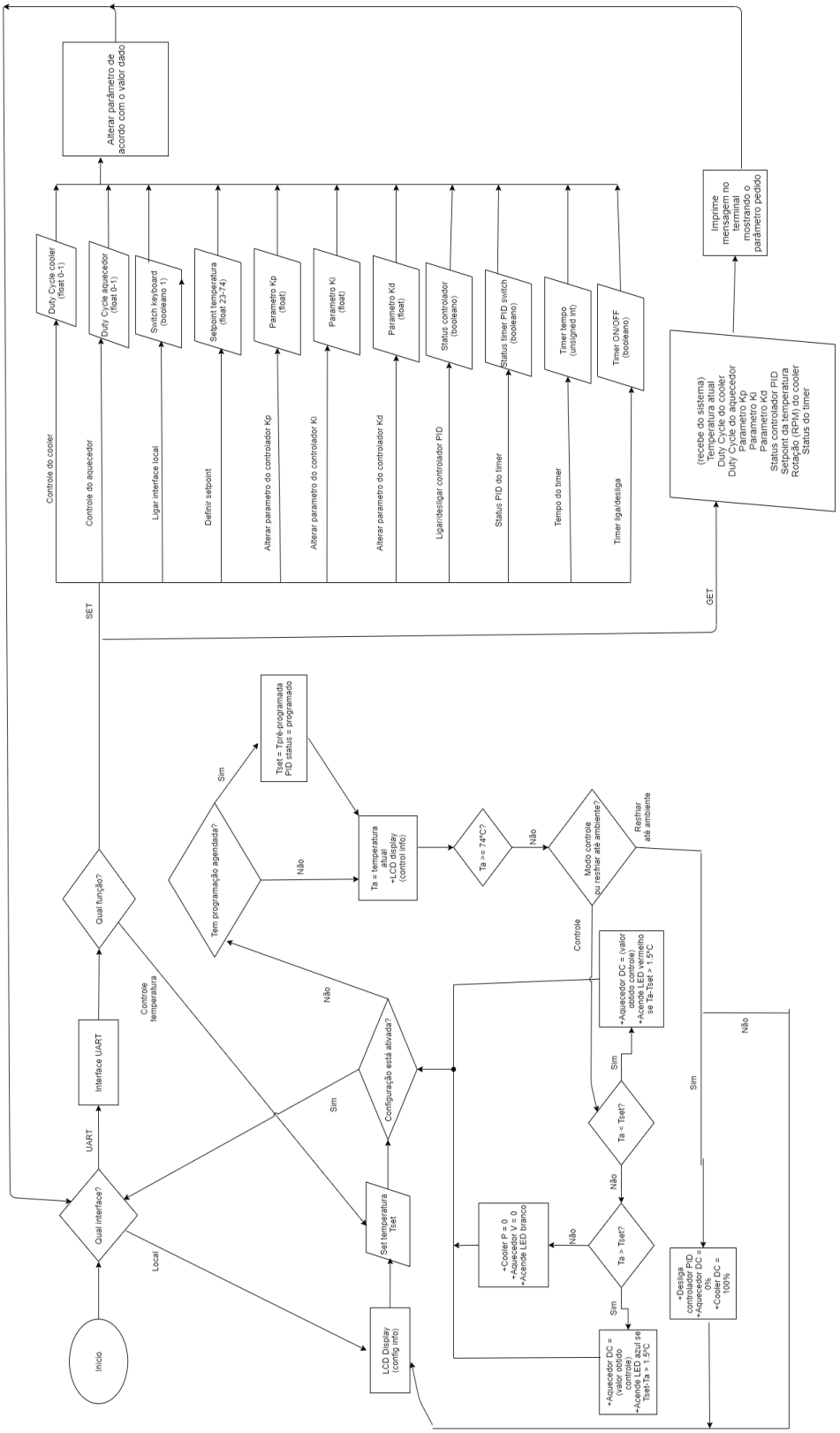


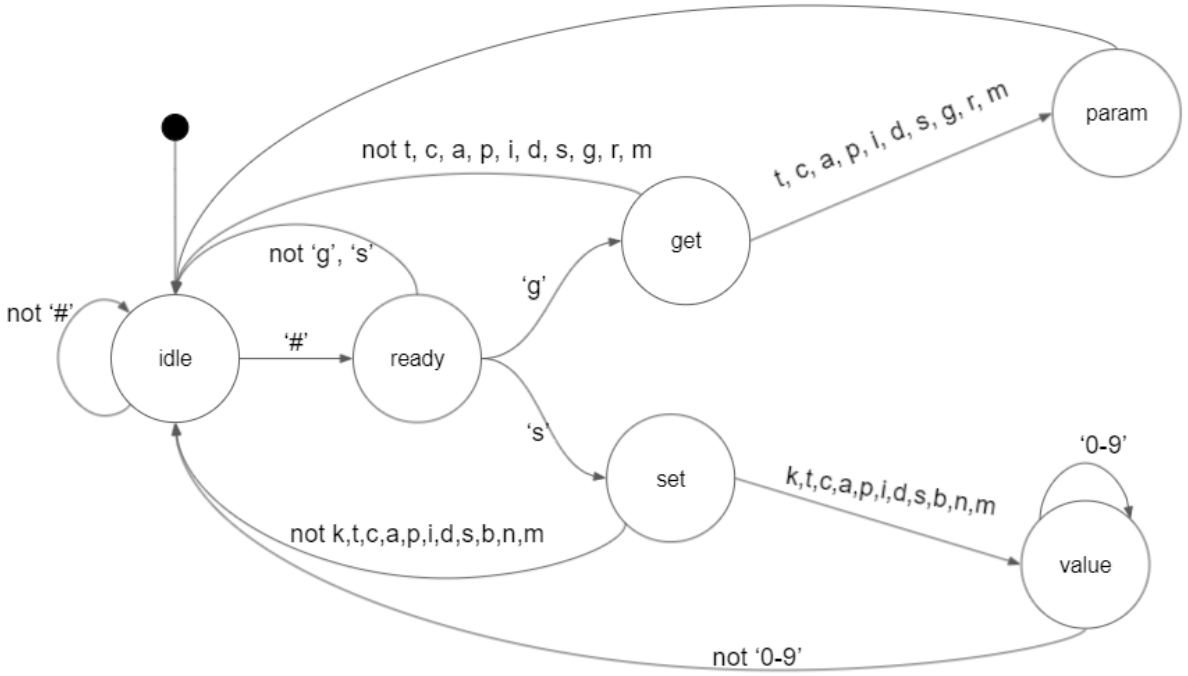
Diagrama do estado “controlando”



2.6. Fluxograma do sistema



2.6 Diagrama máquina de estados UART



3. Sintonização do controlador

Para realizar o controle de temperatura do sistema, foi escolhido um controlador do tipo PID, um controlador robusto e utilizado largamente na indústria, que pode ser facilmente implementado em sistemas embarcados, já que não exige muito poder de cálculo e performance por parte do hardware.

O controlador PID é composto por três partes: proporcional, integral e derivativa. Para implementar a parte derivativa, é necessário reduzir o ruído do sinal de leitura do sensor de temperatura. Para isto, foi implementado um filtro digital, usando a média móvel exponencial dupla (do inglês *Double Exponential Moving Average*). Esse filtro alisa a curva de temperatura, reduzindo a amplitude do ruído e melhorando a qualidade do cálculo do PID. Na figura 3.1, é possível notar o efeito do filtro implementado, amenizando a amplitude do ruído.

O controlador atua sobre o Duty Cycle do aquecedor, inicialmente foi testado o controle com o cooler constantemente ligado com DC = 50%, porém notamos que ele não estava contribuindo para o sistema, estava atrasando a velocidade de aquecimento e aumentando a oscilação, por fim foi decidido que o controle seria feito com o cooler desligado, porém é possível controlar o Duty Cycle do cooler pelas interfaces, foram feitos testes com o controlador ligado e o cooler setado para DC = 40% e o desempenho se manteve com apenas uma demora a mais para aquecer até o setpoint e com um pouco de oscilação depois de estabilizar.

A fim de sintonizar os ganhos do controlador, inicialmente foi utilizado o método de Ziegler-Nichols em malha fechada. Porém, esse método não apresentou bons resultados. Devido às oscilações presentes tornou-se difícil determinar o ganho crítico do sistema.

Em seguida, decidiu-se realizar uma identificação do sistema, a fim de obter a ordem de grandeza dos ganhos do controlador PID. Primeiro, um duty cycle de 20% foi aplicado no aquecedor, e após a estabilização da temperatura, em $t = 0$, aumentou-se para 40%. A temperatura observada foi a seguinte:

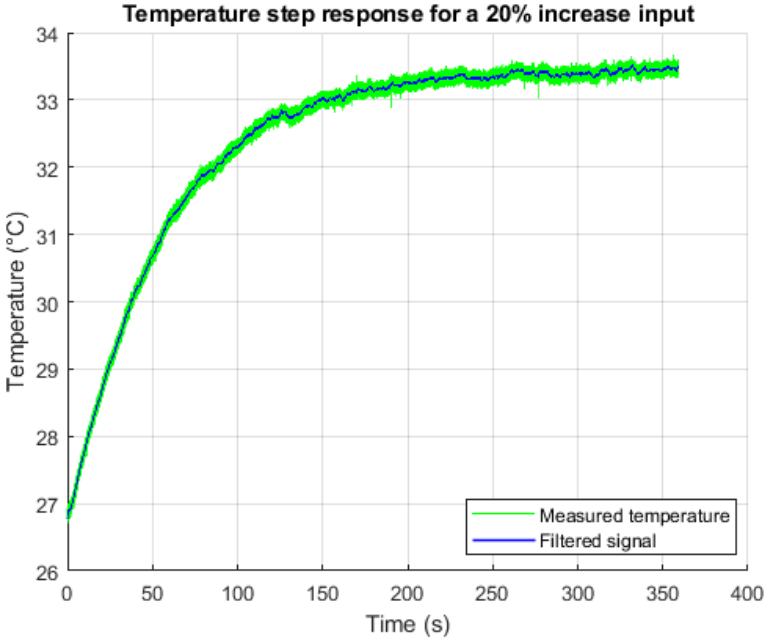


Figura 3.1: Resposta da temperatura em malha aberta.

Observando a forma da resposta, o sistema foi identificado para um sistema de primeira ordem com delay puro, e o seguinte sistema foi identificado:

$$G(s) = exp(-2s) \cdot \frac{6.591}{1117s+20}$$

A figura abaixo compara a saída do modelo e os dados reais para uma mesma entrada e condições iniciais.

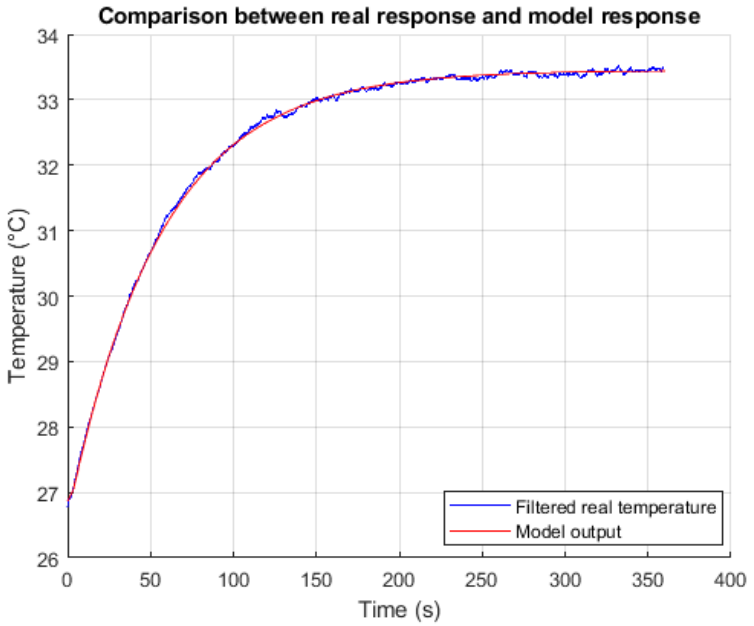


Figura 3.2: Comparação entre os dados obtidos e o modelo identificado

Após identificação do modelo, analisou-se a planta utilizando o *Matlab* e após simulações com possíveis controladores, conclui-se que os ganhos do PID são da seguinte ordem de grandeza: $k_p = 10$, $k_i = 0,1$ e $k_d = 1$.

A partir dos valores estimados no Matlab, realizou-se diversos testes e ajustes empíricos dos ganhos utilizando o kit instalado no computador 3 do laboratório. Chegou-se em um resultado muito satisfatório com os ganhos a seguir. A figura abaixo mostra a performance do sistema quando o *setpoint* de temperatura é 70°C.

$$k_p = 10$$
$$k_i = 0,1$$
$$k_d = 20$$

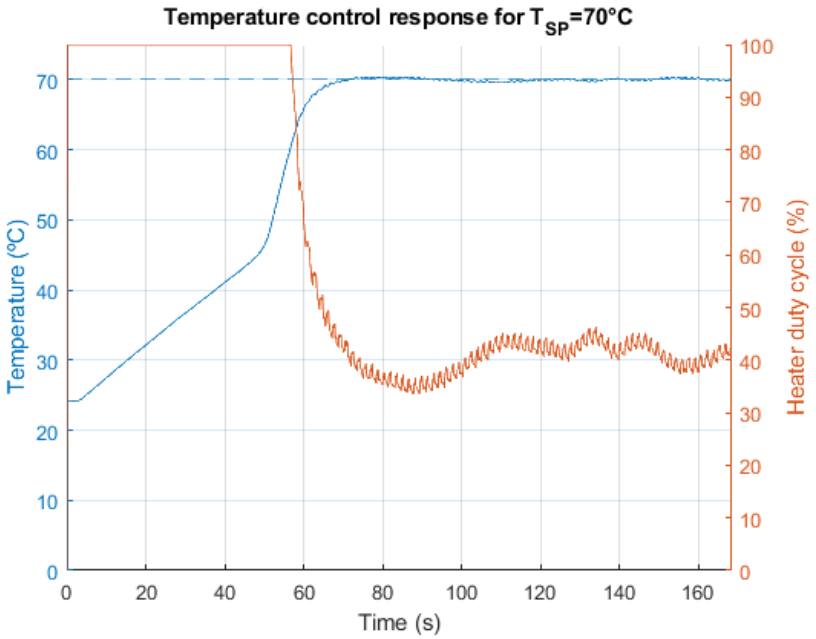


Figura 3.3: Controle da temperatura em 70°C

Observa-se que o sistema apresenta um overshoot pequeno de menos de 1%, importante observar que o valor observado de overshoot está dentro do valor no qual o sistema oscila quando estabilizado (em torno de 0,5ºC). Além disso, é possível verificar que o sistema responde o mais rapidamente possível, uma vez que a saída se mantém saturada durante boa parte da curva de subida. É importante notar o papel do sistema *anti-windup* implementado no controlador, que evita o overshoot devido a saturação do sinal de controle.

Obs: resultados obtidos pelo computador 3 do laboratório. A performance varia entre os kits McLab2.

4. Manual de utilização

O sistema possui duas interfaces por onde o usuário pode interagir com seus parâmetros e funções, uma interface local que consiste de três botões (botões 2, 3, e 4) e um display LCD, e uma interface UART por onde o usuário pode dar instruções ou checar parâmetros através do terminal. Devido a impossibilidade de acesso aos botões, o sistema se inicia com a interface UART ligada.

Interface local:

A interface terá os parâmetros mostrados no display LCD, a atualização do LCD e a leitura dos botões é realizada a cada 0.5 segundos, essa interface tem 12 menus diferentes e ao pressionar o botão 4 ele muda para o próximo menu, de maneira geral o botão 2 diminui os parâmetros/desliga funções e o botão 3 aumenta os parâmetros/liga funções. Na segunda linha do LCD sempre será mostrado a temperatura atual medida e a temperatura de set point, a primeira linha irá variar de acordo com o menu atual.

-Menu padrão: Neste menu serão mostrados constantemente os valores do duty cycle do aquecedor e a rotação em RPM do cooler, os botões 2 e 3 não desempenham nenhuma função nesse menu.

-Menu tempSet: Neste menu será possível controlar a temperatura de set point, o botão 2 diminui esse set point em 0.5ºC e o botão 3 aumenta esse valor em 0.5ºC.

-Menu Cooldown: Neste menu poderá ser ativada a função de resfriar o sistema até a temperatura ambiente, ao ligar esse modo o controlador será desligado e o duty cycle to cooler será colocado no máximo, o botão 2 desativa essa função e o botão 3 ativa essa função, o status desse modo será mostrado no LCD.

-Menu PIDswitch: Neste menu será possível ativar e desativar o controlador PID, o botão 2 desliga o controlador e o botão 3 liga o controlador, o status do controlador será mostrado no LCD

-Menu Kp: Menu para controlar o parâmetro Kp do controlador, esse parâmetro será mostrado no LCD e poderá ser alterado com os botões, o botão 2 diminui o parâmetro em 1 e o botão 3 aumenta o parâmetro em 1.

-Menu Ki: Menu para controlar o parâmetro Ki do controlador, esse parâmetro será mostrado no LCD e poderá ser alterado com os botões, o botão 2 diminui o parâmetro em 0.05 e o botão 3 aumenta o parâmetro em 0.05

-Menu Kd: Menu para controlar o parâmetro Kd do controlador, esse parâmetro será mostrado no LCD e poderá ser alterado, o botão 2 diminui o parâmetro em 1 e o botão 3 aumenta o parâmetro em 1.

-Menu Cooler: Neste menu o duty cycle e a rotação do cooler serão mostradas no LCD, o botão 2 diminui o duty cycle em 5% e o botão 3 aumenta o duty cycle em 5%.

-Menu Aquecedor: Neste menu será mostrado o duty cycle do aquecedor, o botão 2 diminui o duty cycle em 5% e o botão 3 aumenta o duty cycle em 5%.

-Menu TimerSet: Neste menu será possível controlar os parâmetros do timer, no LCD será mostrado o tempo que será programado caso o timer seja ativado e o status para qual esse timer irá mudar o controlador. O botão 2 diminui o tempo do timer (inicialmente de 5s em 5s, se o tempo for maior que 5min será de 1min em 1min, e se o tempo for maior que 1h será de 5min em 5min), o botão 3 irá aumentar o tempo do timer (seguindo o mesmo padrão do botão 2). Ao apertar os botões 2 e 3 ao mesmo tempo o status para qual o controlador irá mudar de ON/OFF para OFF/ON.

-Menu TimerStatus: Neste menu o LCD irá mostrar se a função do timer está ligada/desligada e quanto tempo falta caso ela esteja ligada, o botão 2 desativa o timer e o botão 3 ativa o timer (com os parâmetros configurados no menu anterior).

-Menu UART: Esse menu mostra se a interface UART está ligada ou desligada. Se a interface local estiver desligada ao pressionar o botão 2 a UART será ligada (e os botões 2 e 3 desligados), se a interface local estiver desligada o botão 4 irá desligar a UART e ligar a interface local (se a interface local estiver ligada o botão 4 terá sua funcionalidade padrão de mudar para o próximo menu, como este menu é o último ele retornará para a interface padrão)

obs: Não dar comandos através da interface UART enquanto a interface local estiver ativada, os comandos poderão ser interpretados erroneamente como pressionamento do botão 2.

obs2: Os valores para alterar os valores de Kp, Ki, e Kd foram escolhidos levando em consideração as ordens de grandeza encontradas para o controlador estável.

Interface UART:

A interface UART permite controlar o sistema através do terminal Putty, os comandos tem o formato #(carácter parâmetro); para os comandos de GET, e #(carácter parâmetro)(valor); para os comandos de SET, os valores a serem passados na função SET permitem valores de até 7 dígitos (incluindo a vírgula no caso de um float), as tabelas abaixo descreve esses comandos:

SET:

#st<numero_real>;	Definir setpoint da temperatura
#sk<booleano>;	Definir modo de operação da interface: ● ‘1’ : Habilitar botões (habilita interface local e desabilita UART)

#sc<numero_real>;	Definir duty cycle do cooler (entre 0 e 1)
#sa<numero_real>;	Definir duty cycle do aquecedor (entre 0 e 1)
#sp<numero real>; #si<numero real>; #sd<numero real>;	Definir parâmetro Kp Definir parâmetro Ki Definir parâmetro Kd
#ss<booleano> ;	Ligar/Desligar controlador PID ● ‘0’ : Desliga PID ● ‘1’ : Liga PID
#sb<booleano>;	Definir se timer vai ligar ou desligar PID ‘0’ : Timer desligará PID ‘1’ : Timer ligará PID
#sn<unsigned int>;	Definir tempo do timer em segundos (Max = 599940 segundos)
#sm<booleano>;	Switch do timer ‘0’ : Aborta o timer ‘1’ : Liga o timer

GET:

#gt;	Obter temperatura atual
#gc;	Obter duty cycle cooler
#ga;	Obter duty cycle aquecedor
#gp;	Obter Kp
#gi;	Obter Ki
#gd;	Obter Kd
#gs;	Obter status do PID (on/off)
#gg;	Obter setpoint da temperatura
#gr;	Obter rotação do cooler (RPM)
#gm;	Obter status do timer

obs: O BAUD rate utilizado é de 115200.

5. Problemas identificados e futuras melhorias

5.1. Conflito UART e Interface local:

A interface local utiliza os botões 2, 3, e 4 do keyboard, porém a interface UART compartilha um dos seus pinos de comunicação com o botão 2, quando a interface local estiver desativada não haverá conflito, pois a leitura do botão irá retornar o valor -1 e o sistema indentificará que esse botão está desativado, mas no caso de a interface local estiver ativada e o usuário enviar um comando através da UART a leitura do botão 2 poderá retornar um falso positivo, o sistema achará que o usuário pressionou esse botão mesmo que o usuário não tenha pressionado, portanto o usuário deve evitar enviar comandos pela UART quando a interface local estiver ativada.

É possível também que ao pressionar o botão 2 a UART interprete o sinal que está passando pelo fio como um comando, porém é extremamente improvável que um comando completo seja dado através do botão. Devido a impossibilidade de testar o pressionamento dos botões é possível também que exista algum erro não identificado aqui.

Escolhemos utilizar três botões pois esse efeito colateral (onde o usuário tenta passar uma instrução pela UART quando a interface local está ativada) pode ser facilmente evitado pela parte do usuário, e o botão extra aumenta significativamente a liberdade e facilidade de controlar os parâmetros pela interface local.

6. Autoria dos códigos

Todo o software fornecido foi escrito pelo grupo, com exceção dos arquivos fornecidos previamente pelo próprio professor durante os laboratórios da disciplina.

O código do filtro digital que utiliza a média móvel exponencial foi escrito pelo grupo com base no seguinte artigo online:

- Double Exponential Moving Average Filter – Speeding up the EMA. Norwegian Creations, 31 de agosto de 2016.
  Disponível em:
  https://www.norwegiancreations.com/2016/08/double-exponential-moving-average-filter-speeding-up-the-ema/.