

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE SISTEMAS ELÉTRICOS DE AUTOMAÇÃO E ENERGIA
ENG10032 – MICROCONTROLADORES

PROJETO FINAL

Gabriel J.A Grabher - 244073
João Pedro G. de Souza - 273166

Porto Alegre, 12 de maio de 2021

1. Introdução

Este relatório tem como objetivo apresentar o desenvolvimento do projeto de um aquecedor elétrico para a disciplina ENG10032 Microcontroladores. O objetivo do projeto é desenvolver um shield para a Galileo capaz de controlar a temperatura de um aquecedor elétrico através do sinal de PWM (Pulse Width Modulation) e leitura de sensores de temperatura e nível de água.

2. Projeto de Hardware

Foi feito um projeto de controle para o aquecedor, onde realizamos a leitura de um sensor de temperatura da água, e através de um controlador PID (Proportional Integral Derivative) definimos a tensão média que deve ser aplicada no aquecedor.

Foi utilizado um aquecedor ligado à rede 127V, onde o circuito de acionamento do aquecedor foi isolado da galileo por um optoacoplador. Foi utilizado o método de PWM para o ajuste da tensão da tensão média aplicada no aquecedor.

Utilizamos um sensor de nível da água, para garantir que o sistema nunca será acionado sem a quantidade de água suficiente para o funcionamento correto.

Para garantir o funcionamento do circuito, o sensor de temperatura foi isolado através de um termo retrátil, para garantir que a água não conduza corrente ao entrar em contato com o sensor.

3. Projeto de Software

Foi desenvolvida uma API para realizar a leitura dos sensores e controle do PWM. A estrutura de diretórios da API foi organizada da seguinte forma:

- include/ : contém todos os arquivos de header;
- lib/ : contém todos os arquivos fonte da biblioteca dos sensores e PWM;
- src/ : contém o arquivo main.c

A API desenvolvida conta com as seguintes funções:

- calc_temp()
 - Calcula o valor da temperatura com base na tensão sobre o NTC lida em AD0, usando um divisor de tensão e propriedade específicas do termistor.
- pid_controller()
 - Calcula a tensão a ser aplicada no sinal de PWM com base na diferença entre a temperatura deseja e a temperatura medida, juntamente com sua derivade e integral multiplicados por coeficientes obtidos anteriormente.
- pwm_init()

- Inicializa o pwm, com um período de 1MHz. Retorna um handler para o duty cycle.
 - Neste projeto foi utilizado o pwm3.
- `pwm_end()`
 - Fecha o handler do arquivo duty cycle, e desativa pwm3.
- `pwm_tension()`
 - Recebe um valor para a tensão média e calcula o valor do duty cycle com base no período que foi definido para o PWM.
 - Escreve o valor calculado no handler do duty cycle.
- `R_thermistor()`
 - Calcula a resistência do NTC através de um divisor de tensão conhecido.
- `store_temperature()`
 - Salva a temperatura atual lida pelo sensor em um buffer contendo as últimas 10000 leituras.
- `init_temp_sensor()`
 - Cria um handler para o arquivo A0.
- `temperature()`
 - Lê a tensão do circuito do sensor de temperature em AD0 e retorna a temperatura medida em C°.
- `end_temp_sensor()`
 - Fecha o handler do arquivo A0.
- `init_water_sensor()`
 - Cria um handler para o arquivo IO2.
- `water_level()`
 - Realiza a leitura do sensor de nível da água, que está conectado no pino IO2. A função é utilizada para verificar se o nível da água está atingindo o valor mínimo para o correto funcionamento do circuito.
- `end_water_lvl_sensor()`
 - Fecha o handler do arquivo IO2.

Com o valor obtido pela leitura do sensor de temperatura da água, foi feito o cálculo do PID. O PID possui três coeficientes: proporcional, integral e derivativo. Cada coeficiente tem o seu ganho, e somando-os chegamos ao valor de saída do PID. O valor dos ganhos deve ser ajustado, para o sistema de controle funcionar de forma correta, e com um bom tempo de resposta.

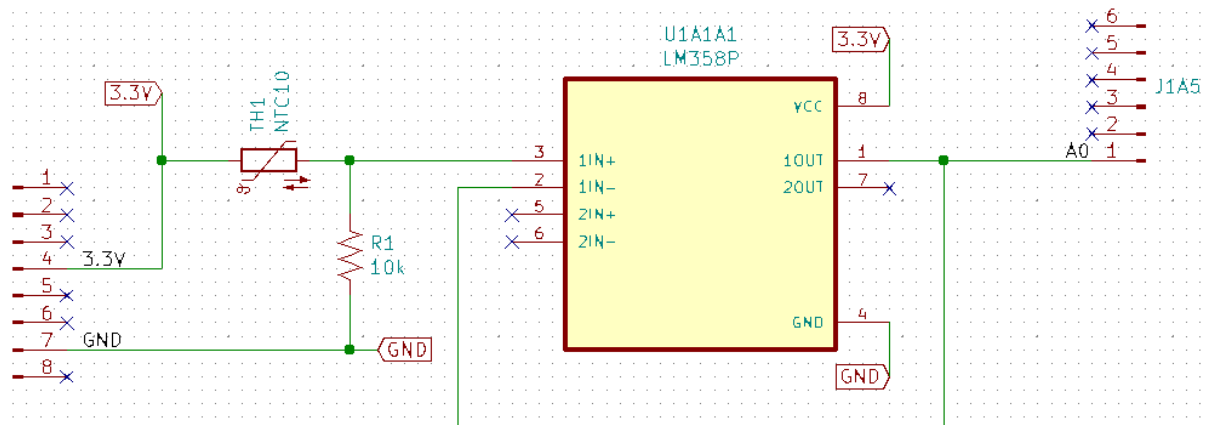
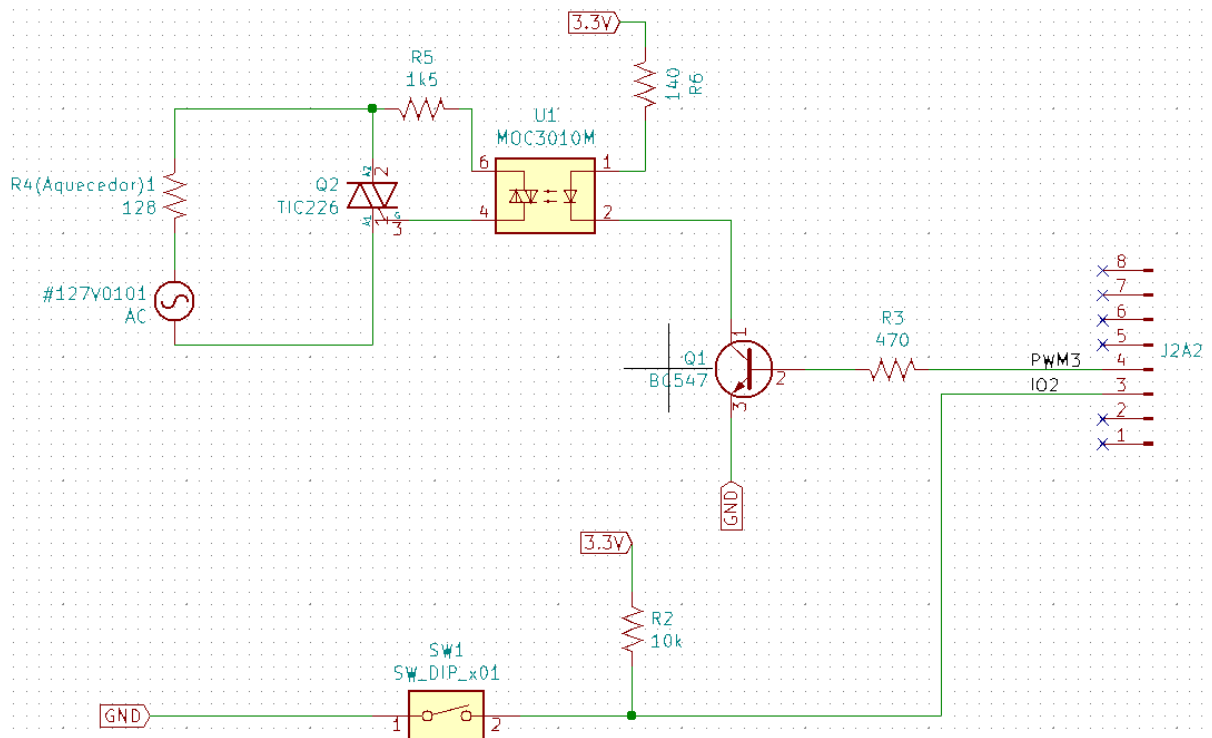
O coeficiente proporcional é a diferença entre o valor esperado, e o valor que é lido no sensor de temperatura. Já o coeficiente derivativo é a taxa de variação do erro, enquanto o coeficiente integral soma o termo de erro ao longo do tempo.

4. Documentação

Para gerar a documentação do software, foi utilizado o Doxygen, programa sugerido pelo professor. Através de comentários específicos, o Doxygen é capaz de gerar uma página HTML contendo toda a documentação do software, com descrição das funções e lógica implementada.

5. Anexos

Esquemático do circuito:



Projeto da PCB:

