DESENVOLVIMENTO DE HARDWARE PARA CAIXA DE TRANSPLANTE DE ÓRGÃOS INTELIGENTE COM INTERNET DAS COISAS

Beatriz Nanni dos Santos [1]; Alessandra Dutra Coelho[2]; Wânderson de Oliveira Assis [2]; Fernando de Almeida Martins[2, 3]; Rogério Cassares Pires[3]

[1] Aluno de Iniciação Científica do Instituto Mauá de Tecnologia (IMT);
[2] Professor do Instituto Mauá de Tecnologia (IMT);
[3] Engenheiro do Instituto Mauá de Tecnologia (IMT).

Resumo. *Este trabalho de iniciação científica colaborou com um projeto de pesquisa, com apoio FINEP, voltado para o desenvolvimento de uma caixa de transplante de órgãos e tecidos. A caixa de transplante está sendo desenvolvida pelo Instituto Mauá de Tecnologia em parceria com outras instituições e uma empresa de refrigeração. Neste trabalho foi desenvolvido a arquitetura do hardware do sistema eletrônico da caixa inteligente para transporte de órgãos. Foram inseridos novos sensores na placa primária da caixa, começou-se o desenvolvimento da placa secundária incluindo controle de temperatura e salvamento de dados em cartão micro SDcard. Por fim foi desenvolvido uma interface de visualização dos dados, utilizando internet das coisas.*

Introdução

A embalagem de transporte de órgãos para transplante teve seu desenvolvimento iniciado em dois trabalhos de conclusão de curso por alunos do Instituto Mauá de tecnologia em 2017 e 2018. O primeiro trabalho realizado desenvolveu um protótipo da embalagem para o transporte visando criar algo ergonômico, que proteja e preserve os órgãos da melhor maneira possível, elaborou o sistema de refrigeração utilizando uma tecnologia de placas Peltier para manter a temperatura do órgão na faixa especificada e implementou um sistema de monitoramento da temperatura com visualização num display.

Para tal foram implementados alguns recursos na caixa, tais como vedação da tampa com gaxeta, dobradiça e trava na tampa, berço de acondicionamento do órgão e membrana elástica, que diminuem qualquer impacto causado ao órgão, além de bolsas térmicas e sistema de arrefecimento para manutenção da temperatura interna.

A refrigeração realizada pelas placas peltier, que consiste na junção de dois metais, ocorre pela geração de um delta de temperatura que é produzido quando uma corrente elétrica atravessa a placa, resfriando um dos lados e aquecendo o oposto. Para o controle e visualização da temperatura foi utilizado o microcontrolador Arduino, que por meio do código de programação é capaz de controlar o acionamento e o desligamento das placas peltier de acordo com a medida da temperatura feita pelos sensores, assim como mostrar a temperatura em tempo real num display LCD, e o estado das placas de refrigeração.

No trabalho seguinte foram introduzidas diversas melhorias ao protótipo, a embalagem sofreu mudanças, como alteração do material, criação de uma embalagem interna removível, redução significativa do peso, melhoria e redundância do sistema de resfriamento e mudança do sistema de controle para um microcontrolador PIC, já que anteriormente foi utilizado o Arduino, que não é recomendado para aplicações industriais.

Além dessas consideráveis melhorias, nesse projeto também foram realizados uma série de ensaios para avaliar pontos de falha do protótipo, para garantir que essa tecnologia possa ser oferecida a sociedade, entre eles: queda livre, compressão, vibração e termodinâmico.

Com o desenvolvimento até esse ponto, pode ser considerado que esta embalagem terá um impacto bem positivo para a área da saúde, já que atualmente é utilizado embalagens térmicas simples, com refrigeração feita por gelo, sem nenhum tipo de controle interno, podendo acarretar em algumas perdas de órgãos. Contudo, essa embalagem ainda é um protótipo que precisa de melhorias e testes para ser validada, para implementar mais recursos na caixa de órgãos, foi proposto outro projeto.

O objetivo desta iniciação cientifica consiste em ampliar a quantidade de sensores e recursos junto a um microcontrolador mais adequado para que o monitoramento do transporte seja mais completo, implementar o algoritmo de programação do sistema de controle de temperatura secundário do microcontrolador e criar um sistema de salvamento de dados. Como algo complementar desenvolver um ambiente de visualização dos dados coletados utilizando internet das coisas para monitoramento rápido e fácil dos dados da caixa de órgãos.

Material e Métodos

Sensores

Os sensores que serão introduzidos na placa primária da caixa são sensor de pressão diferencial, de abertura e fechamento de tampa, acelerômetro, giroscópio, módulo GPS e alguns outros recursos adicionais, como led, buzzer, display e botões.

O sensor de pressão diferencial será utilizado para medição da pressão na sucção e descarga do compressor para o controle da temperatura. O modelo a ser utilizado é o XGZP6857100KPGPN, que mede valores de -100kPa a +100KPa (CFSensor, 2022) por meio de uma tecnologia piezo resistiva formada por um diafragma elástico integrado a uma ponte de Wheatstone. Quando o diafragma é pressionado, a ponte produz uma tensão proporcional a pressão aplicada.

Para substituir a utilização de um sensor de corrente elétrica será usado um resistor shunt ligado a carga que se deseja medir a corrente. O resistor shunt possui cerca de 0,01 ohms, e ao medir a queda de tensão, é possível calcular a corrente elétrica na carga. É importante fazer essa medição para verificar se o compressor e as placas de peltier estão operando corretamente para garantir eficiência na refrigeração, além de ser usado para verificar se as baterias estão funcionando de modo correto.

Para medir a tensão elétrica será utilizado um divisor resistivo com um resistor de 10k ohms e um de 100k ohms. A tensão de saída será a tensão de entrada dividida por 10, desse modo é possível ler tensões no circuito em pontos desejados.

Sensores microswitch e reedswitch serão utilizados para monitorar a abertura e fechamento da câmera de transporte de órgãos e da casa de máquinas da caixa. O fechamento é indispensável para garantir vedação entre o ambiente interno e externo da caixa, para que o sistema de refrigeração não seja prejudicado. Esse sensor será anexado a tampa da caixa, assim quando o contato é fechado um sinal é enviado para o terminal, indicando que está destampada.

Módulo GPS com antena será incluído na caixa para a obtenção de coordenadas de latitude e longitude permitindo rastrear a embalagem durante todo o percurso. Variáveis como altitude, data e hora podem também ser exibidas pelo sensor. O módulo GPS GY – NEO6MV2 da Ublox inclui 50 canais e antena GPS, possui bateria para backup, comunicação serial via UART, trabalha com 45mA de corrente, além de ter memória EEPROM para salvamento de dados quando for desligado (ublox, 2011).

O módulo A9G GPRS GSM Pudding possui antenas para GPS e GPRS/GSM. Suporta 4 bandas de frequência GSM/GPRS e tem suporte para chamadas de voz e SMS. Possui duas portas UART e uma USB para comunicação serial, conta com entrada para cartão de memória SD, entrada para chip GSM e microfone para realizar as chamadas de voz. Consegue trabalhar em modo de pouco consumo de energia, consumindo apenas 2mA de corrente (Shenzhen Ai-Thinker, 2017).

Tanto o GPS GY – NEO6MV2 quanto o A9G GPRS GSM Pudding estavam sendo avaliados para inserção no projeto, após análise escolheu-se o módulo A9G GPRS GSM Pudding para ser integrado à caixa. Esse módulo apresenta reduzido consumo de energia e maior quantidade de recursos em comparação com o outro citado.

Sensores inerciais, tais como, acelerômetro e giroscópio serão adicionados no projeto de forma a permitir a identificação de alguma movimentação brusca que a caixa possa ser submetida. Assim, será possível registrar ocorrências de queda ou choque, que pode causar algum dano ao órgão transportado, necessitando ser analisado antes do uso. Um acelerômetro é um instrumento capaz de medir aceleração e a partir disso medir a força em que o objeto foi submetido, enquanto o giroscópio detecta a velocidade angular do movimento. O modulo GY-520 foi escolhido para ser implementado na caixa. Trata-se de uma placa que contêm sensores acelerômetro e giroscópio de alta precisão, controlada por um único circuito integrado MPU6050.

Outros recursos serão inseridos para garantir maior segurança no transporte da caixa de órgãos. Display O-LED para que os usuários possam visualizar informações importantes durante o transporte, botões para navegação, leds e buzzer para sinalizar emergências.

Todos os sensores foram devidamente calibrados pela equipe do centro de pesquisas do Instituto Mauá de Tecnologia e foi desenvolvida uma placa com todos os sensores embutidos juntamente com o microcontrolador esp-32. Essa placa será o controle primário da caixa.

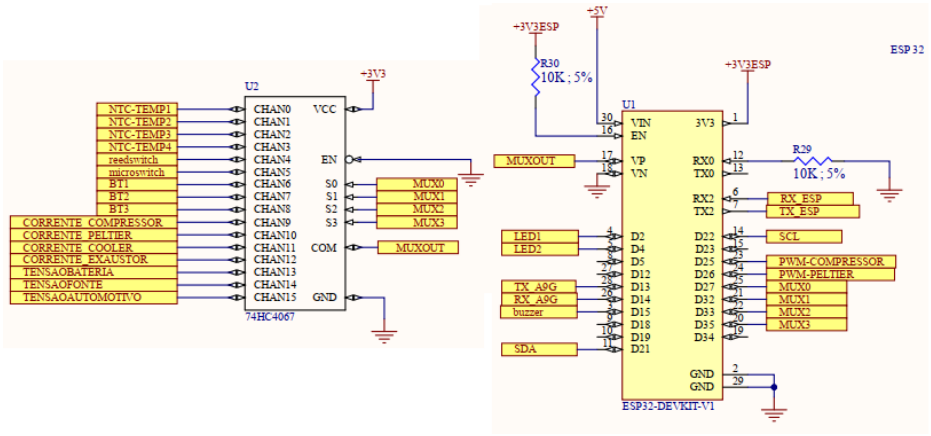
Na Figura 1 é possível visualizar os componentes fundamentais que irão compor a placa. Trata-se do microcontrolador esp-32 e de um multiplexador 74HC4067. O multiplexador comportará os sensores de temperatura interna e externa da caixa, os sensores de tampa, os medidores de corrente e de tensão e botões para navegação do usuário.

Os pinos S0 a S3, são entradas de seleção binárias de 0 a 16, dependendo do valor, uma das variáveis de entrada (CHAN0 A CHAN15) é passada para a saída (COM). As entradas de seleção são definidas no microcontrolador esp32 e enviadas ao multiplexador, e a saída (MUXOUT) é enviada do multiplexador ao esp-32.

Os leds e o buzzer serão conectados ao esp-32, as entradas RX_A9G, TX_A9G, TX_ESP, RX_ESP são as conexões usadas com o módulo GPS A9G e os pinos SDA e SCL são as conexões do módulo MPU6050.

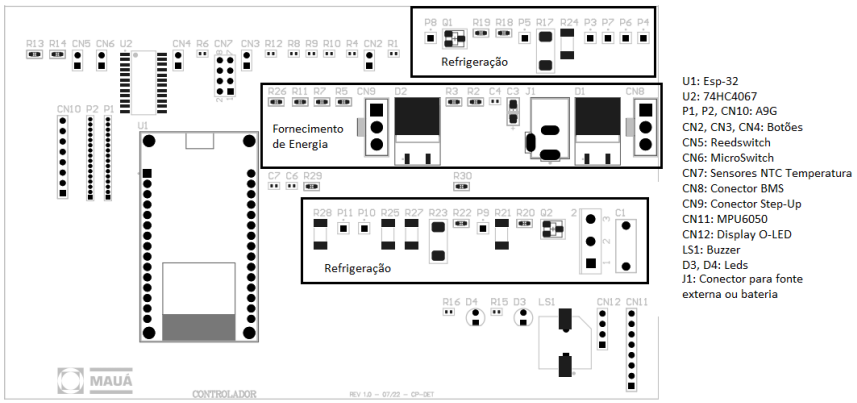
Por fim existem as saídas PWM que controlam as placas peltier e o compressor, para manter a temperatura adequada.

Figura 1: Parte do esquema elétrico da placa primaria da caixa de órgãos



A Figura 2 ilustra o desenho da placa integrada com o microcontrolador, sensores, conexões para fornecimento de energia e ligações para o sistema de refrigeração.

Figura 2: Placa Primária



Controle de Temperatura

Para o controle de temperatura secundário será utilizado um sensor termistor para medição da temperatura, um cooler e uma lâmpada para controle. Todo o processo será feito pelo microcontrolador esp32, que irá receber os dados do sensor e através da implementação de um controle PID enviará os dados de saída para os circuitos do cooler e da lâmpada. Também será implementado um sistema para salvamento dos dados com microsdcard e uma interface de visualização dos dados em tempo real.

Medição da temperatura

Termistores são sensores de temperatura capazes de variar sua resistência proporcionalmente a temperatura, em termistores do tipo NTC, a resistência diminui à medida que a temperatura aumenta.

Através da equação de Steinhart–Hart (Technologies, 2020) é possível relacionar a temperatura (Kelvin) com a resistência (Ω) medida pelo Termistor:

$$\frac{1}{T} = a + b \cdot \ln(R) + c \cdot (\ln(R))^3$$

Na qual a, b e c são variáveis de Steinhart–Hart determinados para cada (Ferreira, 2018). Para um termistor de resistência igual a 10kΩ na temperatura de 298,15K esses parâmetros são:

$a = 0.001129148\,;\ b = 0.000234125\,;\ c = 0.0000000876741$

Foi implementada a conversão no programa do microcontrolador, ilustrado pela Figura 3.

Figura 3: Programa Conversão Resistência para Temperatura

```
// Constantes do termistor
#define A           0.001129148
#define B           0.000234125
#define C           0.0000000876741
#define RES  10000.0
    float res, temp;

    res = RES/(4096.0/((float)analogRead(SENSOR_PIN)) - 1.0);
    res = log(res); // Calcula o ln da resistência para agilizar o cálculo seguinte
    temp = A + B*res + C*res*res*res; // Converte a resistência em temperatura por meio da equação de Steinhart-Hart
    temp = 1.0/temp;// Inverte, pois a equação acima é 1/T
    temp -= 273.15; // Converte de Kelvin para °C
```

Controle PID

Um controle PID é responsável por garantir que o processo ocorra dentro de um padrão especificado, medindo a variável do processo constantemente e comparando-a e corrigindo-a de acordo com a variável especificada.

A medição é feita por um sensor, e o valor desejado, chamado de setpoint, é definido pelo usuário. No programa interno do controlador cria-se uma variável que relaciona o valor medido e o setpoint, chamado erro.

$$e = setpoint - sensor$$

O controlador possui três sinais de entrada, a componente proporcional(kp), a integral (Ti) e a derivativa (Td), as quais se juntam formando um único sinal de saída.

Seu objetivo é que o valor da variável fique o mais próximo possível do setpoint, ou seja, que tenha a mínima variação possível, independente de interrupções ou perturbações que possam ocorrer no processo.

Os coeficientes kp, Ti e Td podem ser ajustados no programa do controlador dependendo da aplicação desejada.

A parte proporcional do controle cria um sinal proporcional a magnitude do sinal de erro, chamada ganho proporcional. Isto faz com que o sistema responda com maior agilidade, mas o sinal de saída pode ultrapassar ou ficar abaixo do setpoint em regime permanente, e o erro não será anulado completamente (Bertoncello, 2020).

A equação da componente proporcional consiste na constante, ganho proporcional, multiplicado pelo erro medido

$$P = kp \cdot e$$

A parte integral cria um sinal proporcional a duração e a magnitude do sinal de erro. Isso faz com que o erro que ficou acumulado pela parte proporcional seja removido.

A equação da componente integral consiste na constante integral ki = 1 /Ti multiplicada pela integral, área do gráfico do erro.

$$I = \frac{1}{T_i} \cdot \int_0^t e dt$$

A parte derivativa cria um sinal proporcional a taxa de variação do sinal de erro, ou seja, ela antecipa a ação do controle para que o processo seja mais rápido ou mais lento, isso aumenta a estabilidade relativa do sistema (Bertoncello, 2020).

A equação da componente derivativa consiste na constante Td multiplicada pela derivada, taxa de variação, do erro.

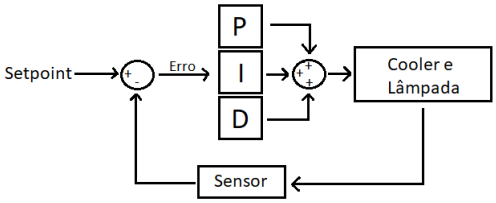
$$D = Td \cdot \frac{de(t)}{dt}$$

A equação a seguir mostra o sinal de saída após somar as três partes do controle (Astrom & Hagglund,1995):

$$Sinal = Kp\left(e(t) + Ki \cdot \int_0^t e(t)dt + Td \cdot \frac{de(t)}{dt}\right)$$

A Figura 4 ilustra o funcionamento do sistema de controle de malha fechada PID. Na qual o setpoint e o sensor são as entradas e o sinal PID é a saída para os atuadores.

Figura 4:Controle PID



É possível ajustar os parâmetros proporcional, integral e derivativo, criando um sinal de saída desejado. Não tem um ajuste correto, depende muito dos requerimentos da aplicação em que ele será utilizado. Cada parte do controlador vai alterar o sinal da saída, podendo assim ter várias possibilidades e várias aplicações.

Para o projeto foi utilizada a expressão do controle de forma discretizada, na qual, se mede o erro a cada tempo de amostragem (T), sendo k a quantidade de amostras. Assim a parte integral é a somatória de todos os erros medidos e a parte derivativa é a variação do erro atual e do erro anterior.

$$Vsaida(k) = kp \cdot \left(e(k) + \frac{T}{Ti} \cdot \big(e(k) + somae(k)\big) + \frac{Td}{T} \cdot (e(k) - e(k-1))\right)$$

Sendo a variável de saída calculada por meio dos parâmetros PID, é necessário controlar o cooler e a lâmpada de modo que mantenha a temperatura por meio da variável de saída. Caso o erro seja positivo, ou seja, o setpoint se encontra num valor maior que o medido pelo sensor, a variável de saída também será positiva e deve-se acionar o cooler, caso o erro seja negativo, a temperatura medida está maior que a desejada, então aciona-se a lâmpada.

Foi adotado 1 segundo como tempo de amostragem, e por meio de testes e considerando o melhor resultado obtido, foi utilizado Kp = 140, Ti = 20, e Td =0 para o projeto.

A Figura 5 ilustra o programa desenvolvido para aplicação do controle PID:

Figura 5:Programa Controle PID

```
dados.sensorData = (int)temp;
dados.controle.ek = dados.setPoint - dados.sensorData;
//Parte proporcional
dados.controle.p = kp*dados.controle.ek;
//Parte integral
dados.controle.i = ((kp*T)/Ti)*(dados.controle.ek+dados.controle.soma_ek);
dados.controle.soma_ek = dados.controle.soma_ek + dados.controle.ek;
//Parte derivativa
dados.controle.d = ((kp*Td)/T)*(dados.controle.ek-dados.controle.ek0);
dados.controle.ek0 = dados.controle.ek;

dados.controle.saida = dados.controle.p + dados.controle.i + dados.controle.d;
if(dados.controle.saida > 0){
    dados.aqueceData = dados.controle.saida;
    dados.coolerData  = 0;
  }
  if(dados.controle.saida < 0){
    dados.coolerData = -dados.controle.saida;
    dados.aqueceData = 0;
  }
```

Gravação dos dados no micro SD card

Será utilizado um micro sdcard em conjunto com o esp32 para a gravação dos dados medidos de temperatura, setpoint e variáveis de controle durante todo o processo, para que depois do transporte os dados possam ser consultados sendo possível verificar se o resfriamento ocorreu de forma correta.

Para efetuar o salvamento de dados foram criadas duas funções dentro do programa, uma para inicializar o cartão sdcard e outra para salvar os dados.

Para o desenvolvimento do código foi necessário o uso de duas bibliotecas: “FS.h” e “SD_MMC.h”.

Na função de inicialização é feita a verificação se existe algum sdcard conectado e se ele foi reconhecido, e dependendo da situação uma mensagem é exibida no monitor serial. Segue o programa na Figura 6.

Figura 6: Programa Inicialização do cartão SdCard

```
void inicializa_sdcard()
{
  if(!SD_MMC.begin("/sdcard", true))
  {
    Serial.println("SD Card ESP-CAM Mount Failed");
    return;
  }
  uint8_t cardType = SD_MMC.cardType();
  if(cardType == CARD_NONE)
  {
    Serial.println("No SD Card ESP-CAM attached");
    return;
  }
  Serial.println("SD Card ESP-CAM OK!");
}
```

Para realizar o salvamento dos dados, é necessário primeiramente abrir um arquivo, nesse caso será utilizado um arquivo .txt, após deve-se escrever os dados que se deseja salvar dentro do arquivo criado, e por fim fechar o arquivo. Se algo der errado ao abrir o arquivo, também é enviada uma mensagem de erro para o monitor serial. É apresentado na Figura 7 a função criada.

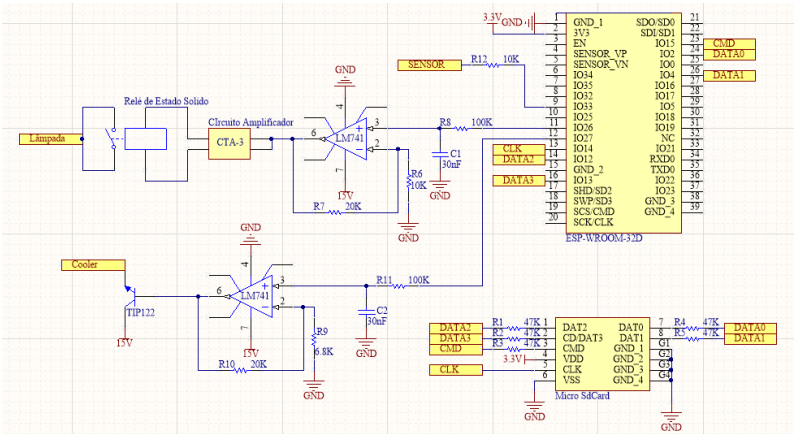
Figura 7: Programa salvamento de dados

```
void salva_sdcard(){
  String path = "/data.txt";
  fs::FS &fs = SD_MMC;
  //Abrir o arquivo
  File file = fs.open(path, FILE_APPEND);
  if(!file)
  {
    Serial.println("Falha ao abrir para gravacao");
    file = fs.open(path, FILE_WRITE);
    if(!file){
      Serial.println("Falha ao abrir para escrita");
      return;
    }
  }
  // Salva no CSV o dado, seguido de uma vírgula.
  char buffer_aux[26];
  strftime(buffer_aux, sizeof(buffer_aux), "%Y:%m:%d %H:%M:%S", &timeinfo);
  file.print(buffer_aux);
  file.write(',');
  file.print("Temp: ");      file.print(dados.sensorData); file.write(',');
  file.print("Setpoint: "); file.print(dados.setPoint);file.write(',');
  file.print("Erro: ");      file.print(dados.controle.ek);file.write(',');
  file.print("Saida: ");    file.print(dados.controle.saida);file.write(',');
  file.print("SaidaLampada: "); file.print(dados.aqueceData);file.write(',');
  file.print("SaidaCooler: ");     file.println(dados.coolerData);
  // Fecha o arquivo
  file.close();
}
```

Esquema elétrico da aplicação

A Figura 8 ilustra o esquema elétrico da aplicação, integrando o microcontrolador, o micro sdcard e os circuitos de acionamento da lâmpada e do cooler.

Figura 8: Esquema Elétrico



Os dois circuitos que integram o microcontrolador com o cooler e a lâmpada serão responsáveis por transformar o sinal enviado do esp32, que é um sinal PWM de 0 a 3,3 V, para um sinal de tensão constante de 0 a 10V por meio de um circuito amplificador além de filtrar valores de alta frequência através de um filtro passa baixa, composto por um resistor de 100kΩ e um capacitor de 30nF.

Frequência de corte do filtro:

$$fc = \frac{1}{2\pi RC} = \frac{1}{2\pi \cdot 100k \cdot 30n} = 53{,}05\ Hz$$

A equação que define o ganho do amplificador operacional é:

$$v_0 = \left(\frac{R1 + R2}{R1}\right) v_i$$

Para o Cooler:

$$v_0 = \left(\frac{6{,}8k + 20k}{6{,}8k}\right) v_i$$

$$v_0 = 3{,}94 v_i$$

Foi necessário um ganho maior para garantir o acionamento do Cooler.

O sinal de tensão que sai do amplificador tem que passar por um transistor TIP122NPN antes de ir para o cooler, pois a corrente do circuito está baixa, ao passar pelo transistor há um aumento de corrente.

Para a lâmpada:

$$v_0 = \left(\frac{10k + 20k}{10k}\right) v_i$$

$$v_0 = 3 v_i$$

O sinal de tensão que sai do circuito amplificador tem que passar por um transmissor CTA - 3 para amplificar a tensão, e por um relé de estado solido responsável por controlar a potência da lâmpada.

Interface de visualização dos dados

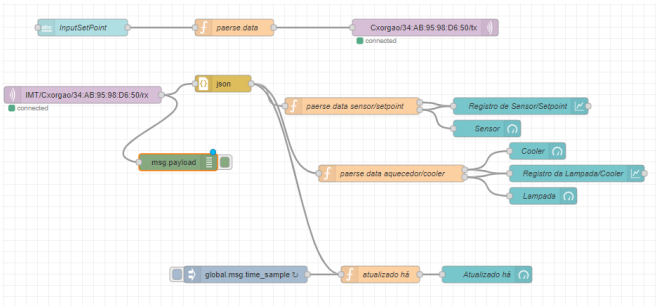
Para a visualização dos dados do projeto foi desenvolvido dashboards que mostram os dados em tempo real. A conexão é transmitida do esp32 via MQTT para o node-red, onde é feita a programação dos dashboards.

Protocolo MQTT (Message Queueing Telemetry Transport) é um sub-servidor de transporte de dados e de fácil implementação, que faz o intermédio de uma conexão entre a origem do sinal e o destino. O recebimento do dado é realizado por meio de um comando de inscrição (subscribe) no mesmo sub-servidor de acordo com um endereçamento de IP (Internet Protocol), porta, nome de usuário e senha inscritos em um tipo de endereçamento denominado tópico (IMT, 2022). Esses parâmetros já foram implementados pelo centro de pesquisas do Instituto Mauá de Tecnologia, dentro do SmartCampus, uma plataforma completa para implementação de internet das coisas.

O site do SmartCampus (IMT, 2022) é aberto a todos, na plataforma estão disponíveis várias aplicações de internet das coisas, com MQTT e node-red, sendo possível acompanhar os dashboards dos projetos em tempo real.

O Node-red é uma ferramenta para programação visual em bloco desenvolvida para conectar hardwares através de internet das coisas e exibir esses dados em interfaces gráficas. Na Figura 9 é possível observar como é desenvolvida a programação de cada painel de informação do dashboard. Cada bloco tem sua programação interna para receber os dados certos enviados do ESP-32. O MQTT é o intermédio que vai conectar o programa desenvolvido no esp-32 com o node-red.

Figura 9: Programação dos Dashboards no Node-red

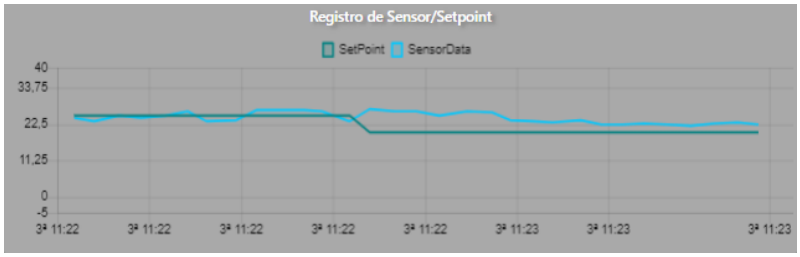


Resultados e Discussão

Em relação ao controle de temperatura, foi realizado testes em uma caixa que possui os atuadores cooler e lâmpada, na qual simulou-se o funcionamento da caixa de órgãos para validação do projeto.

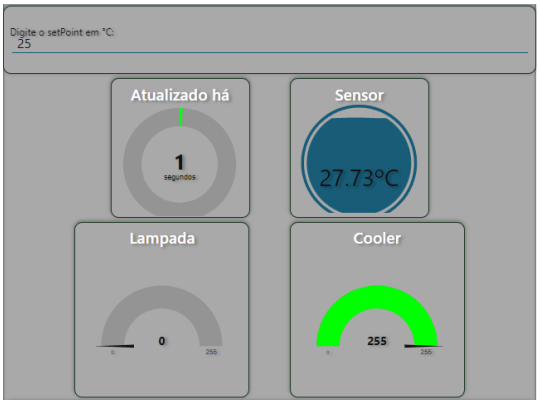
Na Figura 10 é possível visualizar a medida de temperatura em relação ao setpoint escolhido e pode-se verificar que após um tempo a temperatura converge para o setpoint quando alterado.

Figura 10: Gráfico de Registro do sensor e do setpoint



A visualização em tempo real dos dados está ilustrada na Figura 11, a qual mostra a leitura do sensor de temperatura, a quanto tempo os dados foram atualizados, em média 1 segundo, e o estado de funcionamento do cooler e da lâmpada.

Figura 11: Dashbords de visualização dos dados



Durante todo o processo de funcionamento do controle ocorre o salvamento de todos os dados no micro sdcard em forma de texto, na Figura 12 é possível visualizar o arquivo criado.

Figura 12: Arquivo de texto salvo no Micro sdcard

```
2022:10:13 14:07:28,Temp: 26.79,Setpoint: 25.00,Erro: -1.79,Saida: -250.35,SaidaLampada: 0,SaidaCooler: 250
2022:10:13 14:07:28,Temp: 23.53,Setpoint: 25.00,Erro: 1.47,Saida: 206.39,SaidaLampada: 206,SaidaCooler: 0
2022:10:13 14:07:28,Temp: 30.74,Setpoint: 25.00,Erro: -5.74,Saida: -802.97,SaidaLampada: 0,SaidaCooler: 255
2022:10:13 14:07:28,Temp: 24.97,Setpoint: 25.00,Erro: 0.03,Saida: 4.11,SaidaLampada: 4,SaidaCooler: 0
2022:10:13 14:07:28,Temp: 23.24,Setpoint: 25.00,Erro: 1.76,Saida: 245.84,SaidaLampada: 245,SaidaCooler: 0
2022:10:13 14:07:28,Temp: 24.97,Setpoint: 25.00,Erro: 0.03,Saida: 4.11,SaidaLampada: 4,SaidaCooler: 0
2022:10:13 14:07:28,Temp: 26.79,Setpoint: 25.00,Erro: -1.79,Saida: -250.35,SaidaLampada: 0,SaidaCooler: 250
2022:10:13 14:07:28,Temp: 23.24,Setpoint: 25.00,Erro: 1.76,Saida: 245.84,SaidaLampada: 245,SaidaCooler: 0
2022:10:13 14:07:28,Temp: 22.69,Setpoint: 25.00,Erro: 2.31,Saida: 323.80,SaidaLampada: 255,SaidaCooler: 0
2022:10:13 14:07:28,Temp: 24.97,Setpoint: 25.00,Erro: 0.03,Saida: 4.11,SaidaLampada: 4,SaidaCooler: 0
2022:10:13 14:07:28,Temp: 24.10,Setpoint: 25.00,Erro: 0.90,Saida: 126.49,SaidaLampada: 126,SaidaCooler: 0
2022:10:13 14:07:28,Temp: 38.49,Setpoint: 25.00,Erro: -13.49,Saida: -1888.66,SaidaLampada: 0,SaidaCooler: 255
2022:10:13 14:07:28,Temp: 24.39,Setpoint: 25.00,Erro: 0.61,Saida: 86.04,SaidaLampada: 86,SaidaCooler: 0
```

Conclusões

É possível concluir que todos os sensores propostos foram estudados, escolhidos e integrados na placa primaria da caixa de órgãos. Foi iniciado a implementação da placa secundária, a qual inclui controle PID para temperatura que se mostrou eficiente através dos testes e um micro sdcard ativo que salva os dados do controle.

Os dashboards de visualização dos dados foram inseridos no SmartCampus do Instituto Mauá de tecnologia. Sendo possível a consulta dos dados, de qualquer lugar, através de uma visualização online pelo SmartCampus. Assim, este trabalho facilitará o monitoramento à distância da caixa de transplante quando a mesma estiver sendo transportada.

Como sugestão, os demais sensores implementados além dos alarmes e status de bateria podem ser inseridos na visualização através do SmartCampus.

Referências

Astrom, K. & Hagglund, T., 1995. *PID controllers: theory, design, and tuning.* 2nd Edition ed. Research Triangle Park, NC: Instrument society of America.

Bertoncello, S. D., 2020. *Controle PID: rompendo a barreira do tempo.* [Online] Available at: https://www.novus.com.br/blog/artigo-controle-pid-rompendo-a-barreira-do-tempo/ [Acesso em 23 Setembro 2022].

CFSensor, 2022. *XGZP6857A PRESSURE SENSOR.* [Online] Available at: https://www.cfsensor.com/static/upload/file/20220526/XGZP6857A%20Pressure%20Sensor%20V2.4.pdf [Acesso em 18 Outubro 2022].

Ferreira, A. L., 2018. *Compare método Steinhart–Hart - cálculo manual e biblioteca.* [Online] Available at: http://www.squids.com.br/arduino/index.php/desafios/158-desafio22-compare-metodo-para-calcular-temperaturas-com-termistor-ntc-10k [Acesso em 17 Outubro 2022].

IMT, 2022. *SMART CAMPUS.* [Online] Available at: https://smartcampus.maua.br/[Acesso em 15 Outubro 2022].

Shenzhen Ai-Thinker, 2017. *A9G GPRS/GSM+GPS/BDS Module.* [Online] Available at: https://www.usinainfo.com.br/index.php?controller=attachment&id_attachment=701 [Acesso em 18 Outubro 2022].

Technologies, K., 2020. *Practical Temperature Measurements.* [Online] Available at: https://www.keysight.com/us/en/assets/7018-06789/application-notes/5965-7822.pdf[Acesso em 18 Outubro 2022].

ublox, 2011. *NEO-6 u-blox 6 GPS Modules Data Sheet.* [Online] Available at: https://d229kd5ey79jzj.cloudfront.net/976/NEO-6_DataSheet_(GPS.G6-HW-09005).pdf [Acesso em 18 Outubro 2022].