

**UNIVERSIDADE DO VALE DO PARAÍBA
FACULDADE DE ENGENHARIAS, ARQUITETURA E URBANISMO
CURSO DE ENGENHARIA ELÉTRICA**

João Pedro Cruz Gonçalves

TRABALHO DE GRADUAÇÃO

Automatização de um Módulo para Cardioplegia (CMCAP 001).

Trabalho de Graduação apresentado no Curso de Bacharel em Engenharia Civil da Faculdade de Engenharias, Arquitetura e Urbanismo - Univap como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientadora: Prof. Me. Hélio Lourenço
Esperidião Ferreira
Co-orientador: Eng. Rodislei Prado Gonçalves

SÃO JOSÉ DOS CAMPOS
2025

RESUMO

A cardioplegia é um procedimento essencial em cirurgias cardíacas, permitindo a parada temporária e controlada do miocárdio enquanto as funções fisiológicas do paciente são mantidas por equipamentos de suporte. O Módulo de Cardioplegia CMCAP, desenvolvido pela Cardio Medical Industrial, utiliza atualmente um sistema baseado em CLP com painel touchscreen para controle de temperatura, fluxo e tempo de perfusão. Este trabalho apresenta um estudo de viabilidade técnica para substituir o CLP existente por um microcontrolador ESP32, utilizando comunicação Bluetooth para integração com um aplicativo móvel desenvolvido na plataforma App Inventor. A proposta visa reduzir custos, aumentar a portabilidade e facilitar a manutenção do equipamento, mantendo a confiabilidade do sistema. Foram realizados testes de comunicação e simulações de funcionamento dos sensores e atuadores, verificando-se estabilidade na transmissão de dados e respostas rápidas aos comandos enviados pelo aplicativo. Os resultados demonstraram que o uso do ESP32 associado à conectividade Bluetooth é tecnicamente viável para modernizar o controle do equipamento.

Palavras-chave: Cardioplegia. Automação biomédica. ESP32. Bluetooth. Microcontrolador.

ABSTRACT

Cardioplegia is an essential procedure in cardiac surgery, allowing temporary and controlled cessation of myocardial activity while the patient's physiological functions are maintained by life support equipment. The CMCAP Cardioplegia Module, developed by Cardio Medical Industrial, currently uses a PLC-based system with a touchscreen panel to control temperature, flow, and perfusion time. This work presents a technical feasibility study to replace the existing PLC with an ESP32 microcontroller, using Bluetooth communication for integration with a mobile application developed on the App Inventor platform. The proposal aims to reduce costs, increase portability, and facilitate equipment maintenance while maintaining system reliability. Communication tests and simulations of the operation of the sensors and actuators were performed, verifying stability in data transmission and rapid responses to commands sent by the application. The results demonstrated that the use of ESP32 associated with Bluetooth connectivity is technically feasible to modernize the control of the equipment.

Keywords: Cardioplegia. Biomedical automation. ESP32. Bluetooth. Microcontroller.

Sumário

1. ACRÔNIMOS	4
2. INTRODUÇÃO.....	5
3. MATERIAIS E MÉTODOS	9
3.1. Microcontrolador (ESP 32).....	9
3.2. Comunicação ESP 32 e Dispositivo móvel	10
3.3. Sensores.....	12
3.3.1. Sensor Magnético – Sensor de abertura de Tampa	12
3.3.2. Sensor de nível de água no tanque.....	13
3.3.3. Sensor ótico – Sensor de rotação do tipo encoder.....	14
3.3.4. Sensor de Temperatura (PT100)	16
3.4. Atuadores	18
3.4.1. Resistência	19
3.4.2. Bomba d'água	20
3.4.3. Bomba Peristáltica	21
3.5. Construção	24
3.6. Aplicativo para substituição de painel LCD.....	25
3.7. Programação da ESP 32.....	29
4. RESULTADOS E DISCUSSÕES	32
5. CONCLUSÃO.....	35
REFERÊNCIAS.....	36
ANEXOS.....	37
5.1. Programação Aplicativo	37
5.2. Programação na ESP 32.....	42

1. ACRÔNIMOS

- MQTT - Message Queue Telemetry Transport (Transporte de Telemetria de Fila de Mensagens);
- CEC - Circulação Extracorpórea;
- IoT - *Internet of Things* (Internet das Coisas);
- IDE - *Integrated Development Environment* (Ambiente de Desenvolvimento Integrado);
- BLE - *Bluetooth Low Energy* (Bluetooth de baixa energia);
- RPM – Rotações por minuto;
- LED - *Light Emitting Diode* (Diodo Emissor de Luz);
- CLP - Controladores Lógicos Programáveis;
- RTD - *Resistance Temperature Detector* (Detector de Temperatura por Resistência);
- FHSS - *Frequency Hopping Spread Spectrum* (Espectro Espalhado por Salto de Frequência);
- Banda ISM – Faixa de banda usada para usos industriais, científicos e médicos (industrial, scientific and medical);
- GFSK - *Gaussian Frequency Shift Keying* (Modulação por deslocamento de frequência gaussiana);
- BR PHY - *Physical Layer* (“Camada Física” do modelo OSI);
- OSI - *Open Systems Interconnection* (Interconexão de Sistemas Abertos);
- ABNT – Associação Brasileira de Normas Técnicas;
- NBR – Norma Brasileira;
- IEC – *International Electrotechnical Commission* (Comissão Eletrotécnica Internacional);
- INMETRO – Instituto Nacional de Metrologia, Qualidade e Tecnologia.
- ANATEL – Agência Nacional de Telecomunicações;
- ANVISA - Agência Nacional de Vigilância Sanitária;
- PCB - Printed Circuit Board (Placa de Circuito Impresso).

2. INTRODUÇÃO

A Circulação Extracorpórea abriu as portas para a cura de doenças cardíacas, já que a possibilidade de corrigir defeitos do coração visualizando de forma direta. Na máquina de CEC, uma bomba peristáltica faz a função de bombear o sangue no lugar do coração e as trocas gasosas por uma membrana possibilitando uma parada temporária das funções fisiológicas do paciente, tanto do miocárdio como dos pulmões [1]. Já na cardioplegia é somente a parada cardíaca controlada é o procedimento que permite a interrupção temporária e segura da atividade do miocárdio [2], com essa interrupção temporária permite o cirurgião fazer reparação de defeitos complexos ou a substituição de válvulas. Algumas Cirurgias ao longo da história que possa destacar o uso dessa técnica são mencionadas nas **Tabela 1**, **Tabela 2**, **Tabela 3** e **Tabela 4**.

Tabela 1 – Cirurgias no início da Cardioplegia

Ano	Cirurgião	Cirurgia	Descrição
1952	John Lewis	Fechamento de Comunicação Interatrial (CIA) com Hipotermia	Primeiro fechamento de CIA sob visão direta, usando hipotermia moderada e oclusão das veias cavas
1953	John Gibbon	Fechamento de CIA com Circulação Extracorpórea (CEC)	Primeira operação cardíaca com sucesso usando uma máquina coração-pulmão, apesar de não ter sido inicialmente recebida com entusiasmo.

Fonte: Pequena história da cirurgia cardíaca [3]

Tabela 2 – Cirurgias de Implante de Próteses Valvares

Ano	Cirurgião	Cirurgia	Descrição
1960	Dwight Harken	Implante de Prótese Valvar Aórtica	Primeira prótese aórtica implantada em posição subcoronária com o auxílio da CEC.
1961	Albert Starr e Lowell Edwards	Implante de Prótese Valvar Mitral	Marcou o início de uma nova era no tratamento das valvopatias. A prótese Starr-Edwards se tornou popular

Fonte: Pequena história da cirurgia cardíaca [3]

Tabela 3 – Cirurgia de Revascularização do Miocárdio (Ponte de Safena):

Ano	Cirurgião	Cirurgia	Descrição
1967	René Favaloro	Revascularização do Miocárdio com Ponte de Safena Aortocoronarian	Operação pioneira utilizando a veia safena para tratar obstruções coronarianas. Dudley Johnson também é um pioneiro importante
1967	Christiaan Barnard	Transplante Cardíaco Humano para Humano	Um marco histórico na cirurgia cardíaca, apesar dos resultados iniciais insatisfatórios devido à rejeição e infecção

Fonte: Pequena história da cirurgia cardíaca [3]

Tabela 4 – Cirurgia de Correção de Defeitos Congênitos Complexos

Ano	Cirurgião	Cirurgia	Descrição
1975	Adib Jatene	Correção da Transposição dos Grandes Vasos em Nível Arterial (Jatene)	Publicação da primeira operação de correção no nível arterial para essa cardiopatia congênita complexa

Fonte: Pequena história da cirurgia cardíaca [3]

Nessas operações é necessário a presença de um profissional especialista na operação dessa máquina, tanto a de CEC como a de cardioplegia que é o foco desse trabalho, seguindo a RESOLUÇÃO COFEN Nº 528/2016. Esse profissional é o perfusionista que tem a responsabilidade durante o procedimento cirúrgico, pois de certa forma terá em suas mãos e sob seus olhos a vida do paciente que está sendo operado [1].

Durante esse processo da cardioplegia, é necessário que o paciente tenha suas funções fisiológicas mantidas por equipamentos que desempenham de forma controlada a função cardíaca, conforme as especificações médicas e técnicas [2]. Entre as várias marcas de equipamentos médicos deste tipo (cardioplegia) utilizados no Brasil temos o Módulo de Cardioplegia CMCAP (O equipamento está apresentado na **Figura 1**), desenvolvido pela Cardio Medical Industrial de Registro na ANVISA: 81466610006, que integra bomba peristáltica que fará a função de bombear o sangue do paciente [4], sistema de troca de calor para controlar a temperatura do sangue através de resistência [5] e sistema de arrefecimento e cronometragem

para acompanhar o tempo de cirurgia, operando por meio de um painel touchscreen retroiluminado capaz de monitorar temperatura, fluxo e tempo de procedimento.

Figura 1 -Foto do equipamento CMCAP



Fonte: CARDIO MEDICAL INDUSTRIAL (2024) [2]

A motivação desse trabalho é a redução de custo da máquina, já que a diferença de valores entre uma ESP 32 (microcontrolador) e um CLP é considerável, e o aumento do espaço interno para favorecer a manutenção do equipamento, o CLP ocupa um espaço maior no gabinete comparado a ESP 32.

Embora o painel sensível ao toque proporcione uma interface centralizada e multifuncional, ele apresenta limitações práticas que impactam sua usabilidade e benefícios. Entre essas limitações estão a restrição de mobilidade do operador no ambiente cirúrgico e o alto preço de um CLP com uma tela touchscreen. Diante disso, e considerando o avanço das tecnologias embarcadas e de comunicação sem fio, torna-se relevante investigar soluções mais flexíveis, econômicas e acessíveis, capazes de substituir sistemas de automação tradicionais baseados em CLPs por microcontroladores modernos, como o ESP32, com conectividade Bluetooth integrada.

Nesse contexto, este Trabalho de Graduação tem como objetivo geral realizar um estudo de viabilidade técnica da substituição do CLP existente no Módulo de Cardioplegia CMCAP por um microcontrolador ESP32, estabelecendo a comunicação com uma interface móvel via Bluetooth, em caso de falha segue a recomendação existente no manual do equipamento para

operação manual realizada pelo perfusionista. Essa proposta busca replicar as funções de controle e monitoramento originalmente presentes no painel físico, utilizando um smartphone como interface operacional, com a finalidade de tornar o sistema mais acessível, portátil e de fácil manutenção. Ressalta-se que este trabalho não tem por finalidade a criação de um produto nem de um protótipo clínico, limitando-se à análise de viabilidade e experimentação de comunicação e controle entre os dispositivos.

Os principais objetivos desse trabalho, então, são:

- Mapear as funções atualmente desempenhadas pelo CLP no módulo CMCAP;
- Projetar e implementar um firmware em ESP 32 que realize as funções básicas de controle e monitoramento;
- Desenvolver uma interface móvel para comunicação via Bluetooth com o ESP 32;
- Avaliar a comunicação entre ESP 32 e smartphone em termos de estabilidade em cenário laboratorial.

A relevância deste estudo está na possibilidade de modernizar equipamentos biomédicos com recursos de baixo custo, promovendo uma atualização disponível na família de equipamentos do CMCAP. A utilização de microcontroladores como o ESP32, aliada à conectividade sem fio e ao uso de aplicativos móveis demonstra uma agilidade e flexibilidade para o operador do dispositivo que deve ter total controle. Dessa forma, este trabalho contribui para a exploração de soluções que podem reduzir custos, aumentar a flexibilidade de operação e possibilitar atualizações contínuas de software, ampliando as possibilidades de aplicação em ambientes hospitalares.

3. MATERIAIS E MÉTODOS

Durante o desenvolvimento foram utilizados diversos materiais, desde microcontroladores, atuadores e sensores e diversos fatores como programação do microcontrolador, do aplicativo e até sua construção. Fazendo brevemente uma lista de sensores utilizados:

- ESP 32 DevKit V1 (Microcontrolador);
- *Bluetooth Classic* (Comunicação);
- Sensor de abertura de Tampa (Sensor Magnético);
- Medidor de rotação (Sensor ótico – tipo *encoder*);
- PT 100 (Sensor de Temperatura);
- Resistência (Atuador);
- Bomba d'água (Atuador);
- Bomba peristáltica (Atuador);
- App Inventor (Aplicativo Celular);
- Arduino IDE (Programação ESP 32).

3.1. Microcontrolador (ESP 32)

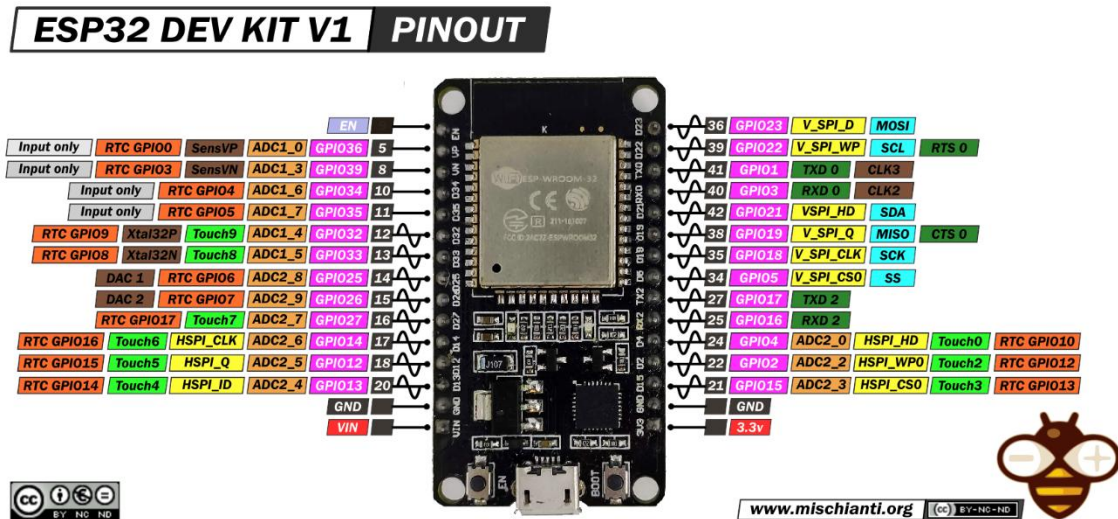
A ESP 32,

Figura 2, é um dos dispositivos no mercado mais utilizados para as atuais automações IoT. Sua fácil programação utilizando IDE simples que é a do Arduino e sua já integração na própria placa com Wifi e Bluetooth ajudam a utilizar protocolos e projetos usando MQTT, por exemplo [6].

Observando esses fatores, foi escolhido esse microcontrolador já observando a integração ao aplicativo móvel sem a necessidade de comprar um módulo separado para a comunicação, como por exemplo o HC-05 que é um módulo amplamente utilizado em projetos com Arduino com integração Bluetooth.

Das várias versões de ESP 32, foi utilizado a ESP 32 DevKit V1 que era a disponível.

Figura 2 – ESP 32 do modelo utilizado



Fonte: Renzo Mischianti [7]

3.2. Comunicação ESP 32 e Dispositivo móvel

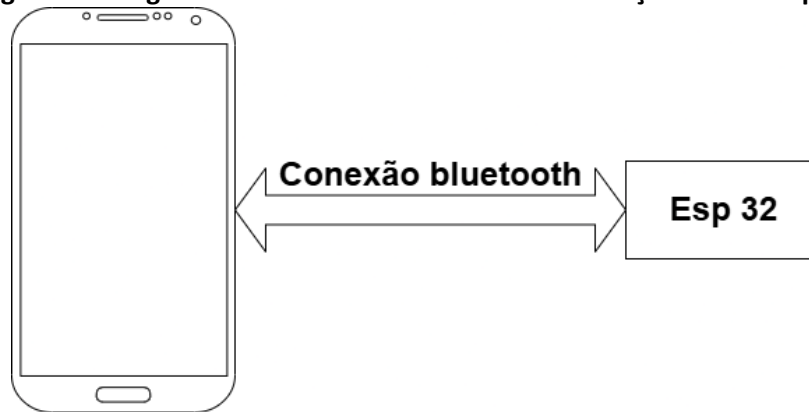
Foram testadas duas estratégias de comunicação entre os dispositivos móvel e o microcontrolador ESP32. A primeira abordagem utilizou do protocolo MQTT, que é reconhecido por sua capacidade de facilitar a comunicação ponto a ponto e por ser uma escolha popular em aplicações de IoT, conforme descrito no site Newton C. Braga [8]. No entanto, a necessidade de uma conexão contínua a uma rede e as implicações de segurança associadas levaram à rejeição dessa opção.

A segunda forma testada, e escolhida, foi o Bluetooth que apesar das dificuldades [9] para pareamento e a possibilidade de interferência com outros celulares com o *bluetooth* ativado, entretanto as probabilidades são baixas já que a distância do celular com a máquina será pequena e a comunicação será com um dispositivo móvel e aplicativo específico, além disso o equipamento passa por ensaios de compatibilidade eletromagnética, o que mostra que a decisão favorável para com essa tecnologia. Além disso ela tem um estilo de conexão bem simples e direta do tipo ponto a ponto como visto na Figura 3, sendo ligado de forma de ida e volta com dois pontos de comunicação de emissor e receptor (podendo comutar). Algumas características relevantes do Bluetooth utilizado no projeto:

- Banda de frequência: Banda ISM de 2,4 GHz (2,402 – 2,480 GHz utilizados);
- Canais: 79 canais com espaçamento de 1 MHz;
- Uso do Canal: Espectro de propagação de salto de frequência (FHSS);
- Modulação: GFSK;
- Taxa de dados: BR PHY: 1 Mb/s;
- Potência de Transmissão: ≤ 100 mW (+20 dBm);
- Sensibilidade Rx: ≤ -70 dBm;

As características do *Bluetooth* [10] utilizado na ESP 32 [11] e na programação utilizando a biblioteca *BluetoothSerial.h*. Com isso em vista e após apresentar os prós e contras a empresa parceira desse trabalho, chegou-se à conclusão de que a melhor escolha seria o *Bluetooth*.

Figura 3 - Diagrama de blocos Bluetooth da comunicação celular-Esp 32



Fonte: elaborado pelo próprio autor

Na ESP 32 utiliza-se o *bluetooth Classic*, que é o mais utilizado entre os dispositivos em geral que não precisam de economia de bateria, e o BLE, um especializado em eficiência e voltado para dispositivos que geralmente têm um fornecimento energético menor, [7] onde o escolhido pela simplicidade de configuração foi o *bluetooth Classic*.

A Comunicação é feita da seguinte forma: do Celular para a ESP 32 tem uma formatação com a temperatura configurada seguida de botões de controle pressionados, “_00.0,0,0,0” e, pelo caminho contrário, mudamos a formatação para não ocorrer confusões, tendo a configuração com a temperatura medida, rotação em RPM e alarme, “:00.0,000,0”. Podemos ver isso de maneira mais evidente na

Figura 4, essa imagem foi tirada do monitor serial da ESP onde vemos o pacote de dados recebidos, em cima, e o pacote de dados enviados, em baixo.



Fonte: próprio autor

3.3. Sensores

3.3.1. Sensor Magnético – Sensor de abertura de Tampa

Para saber se a tampa da bomba peristáltica está aberta usa-se de um sensor que funciona como uma chave onde tendo a presença de um ímã ele se fecha dando contato igual um *Switch on/off*, sem a presença de campo ele tem nível lógico 0 com a presença o nível lógico passa para 1.

Assim tendo um pequeno ímã de neodímio na extremidade da porta, quando fechada, ela aproxima esse ímã no sensor fechando o contato (nível lógico 1) e se aberta o ímã se afasta abrindo o contato (nível lógico 0).

Esse sensor permite ao microcontrolador saber se a porta da bomba de roletes está aberta ou não e, caso aberta, deve parar o funcionamento da mesma por questão de segurança do operador.

Na Figura 5 visto na foto do sensor usado em testes, já na

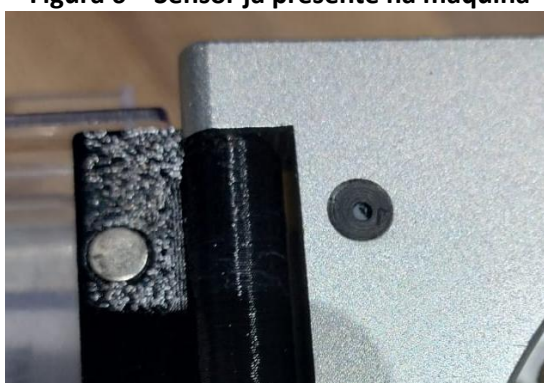
Figura 6 observa-se na direita o sensor já no equipamento e na esquerda um pequeno ímã para identificar quando está aberta a porta.

Figura 5 – Sensor usado em testes



Fonte: próprio autor

Figura 6 – Sensor já presente na máquina



Fonte: próprio autor

3.3.2. Sensor de nível de água no tanque

Quando o nível está alto a boia sobe fechando o contato (Nível lógico 1) e se o nível entra em estado crítico a boia desce e abre o contato (Nível lógico 0).

Tendo em vista que a água ajuda a ajustar a temperatura do sangue, o nível do tanque deve sempre estar cheio para um melhor funcionamento do equipamento e execução da cirurgia, por isso o monitoramento do nível é imprescindível.

Na Figura 7 observa-se o sensor usado durante o desenvolvimento, já na

Figura 8 nota-se o sensor já implementado dentro do tanque

Figura 7 – Sensor usado em testes



Fonte: próprio autor

Figura 8 – Sensor presente no reservatório de água

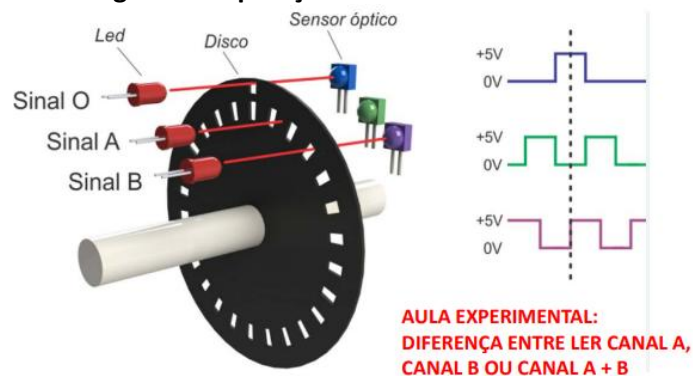


Fonte: próprio autor

3.3.3. Sensor ótico – Sensor de rotação do tipo encoder

“O *encoder* rotativo é um dispositivo que converte o movimento de rotação do eixo em sinais elétricos digitais ou analógicos. Ele permite medir Ângulo de rotação, Velocidade angular (em RPM) e Direção de rotação” [12]. O dispositivo *encoder* tem feixes de luz transpassando por orifícios podendo medir a rotação pela variação do tempo. Na Figura 9 podemos ver que há aberturas que quando alinhadas aos sensores ópticos como os LEDs emitem um sinal permitindo saber a velocidade e o ângulo. Essas cavidades se chamam pulsos, pois “emitem” os pulsos do sinal do sensor. O equipamento mede a rotação do eixo da bomba peristáltica para estimar o fluxo sanguíneo.

Figura 9 – Operação básica de um encoder.



Fonte: Pedro Luiz Golobovante Loureiro (2025) [12]

Equação 1 – Medição de velocidade

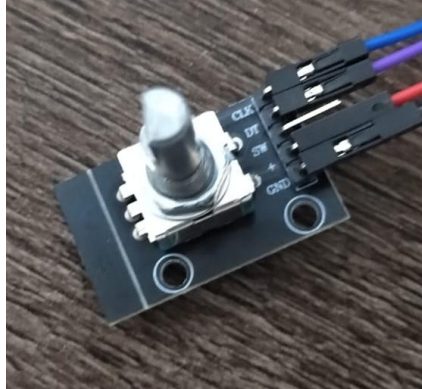
$$Velocidade = \frac{\Delta\theta}{\Delta t} = \frac{18^\circ}{\Delta t} * \frac{\pi}{180} = \frac{\pi}{10 * \Delta t}$$

Na Equação 1 é possível visualizar a dedução do método de medir a velocidade angular (em RPM) usando do *encoder*, onde a variação angular de um orifício para outro dividido pelo tempo em que isso ocorreu conseguimos calcular a velocidade.

Como o modelo utilizado KY-040 tem 20 pulsos, isso é, em uma volta temos 20 aberturas, isso dá que a variação angular entre cada abertura será de 18° , fazendo a conversão de graus para radianos termos $\pi/10$.

O *encoder* utilizado nos testes está presente na Figura 10, ele se apresentou eficaz para os testes. Entretanto para a implementação do projeto recomenda-se o modelo: e38s6g5-400b-g24n, devido a uma melhor precisão e robustez.

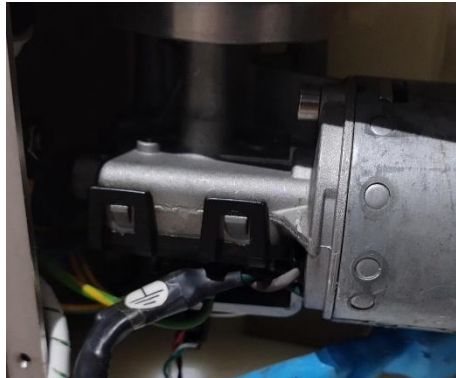
Figura 10 – Sensor usado em testes



Fonte: próprio autor

Na Figura 11 verifica-se o atual sensor presente na máquina, o princípio de funcionamento é o mesmo, porém o encoder indicado é de mais fácil instalação.

Figura 11 – Sensor acoplado ao eixo da bomba de roletes



Fonte: próprio autor

3.3.4. Sensor de Temperatura (PT100)

O sensor é utilizado para a medição da temperatura da água que passa pelo trocador de calor (Consumível descartável) e aquece ou resfria a corrente sanguínea do paciente. Os valores de temperatura são definidos perfusionista (profissional devidamente capacitada para o uso da máquina), além disso a máquina possui alarmes e circuitos de segurança para temperaturas altas e baixas, ver item 3.6.

“A variação da resistência com a temperatura para a maioria dos materiais metálicos pode ser representada por uma equação da forma:” [13]

$$R = R_0 (1 + a_1 T^2) + \dots + a_n t^n$$

Os materiais metálicos como a platina, que é utilizado no PT100, devem ter uma constante “a” que é um coeficiente de relação entre Temperatura e Resistência. Essas grandezas estão diretamente ligadas como podemos observar na Equação de Callender, Equação 2, sabendo disso podemos fazer diversos circuitos para determinar que essa pequena variação de resistência possa determinar uma variação na temperatura.

Equação 2 - Equação de Callender

$$R_T = R_0 + R_0 \alpha \left[T - \delta \left(\frac{T}{100} - 1 \right) \left(\frac{T}{100} \right) - \beta \left(\frac{T}{100} - 1 \right) \left(\frac{T}{100} \right)^3 \right]$$

Fonte: Unicamp [14]

Onde:

R_T é a resistência na temperatura T

R_0 é a resistência na temperatura 0°C

α é o Coeficiente de temperatura em 0°C (Tipicamente +0,00392 W/W/°C para Pt

β é um Coeficiente de se caso $T > 0$ ele vale 0, caso contrário 0,11 para $T < 0$

Assim funciona os sensores RTD ou de resistência,

Figura 12, que foi utilizado, com a variação dele e um módulo, que veremos adiante, é possível determinar a temperatura tendo somente a resistência medida.

Figura 12 – PT100



Fonte: próprio autor

3.3.4.1. Amplificador MAX 31865

O Amplificador usado foi o MAX 31865, Figura 13, esse módulo permite uma comparação utilizando um divisor de tensão [15]. Internamente ele possui um resistor de referência que utiliza para calcular a resistência do PT100, por consequência a temperatura através da **Equação 2**. Este amplificador possui uma peculiaridade que quando usado para resistências a 3 fios, que foi o caso, é necessário colocar em curto um terminal conforme indicado pelo documento da biblioteca utilizado por esse módulo [16].

Como o microcontrolador não mede resistência de forma direta, este amplificador é necessário para a conversão da resistência em sinais que a ESP possa processar.

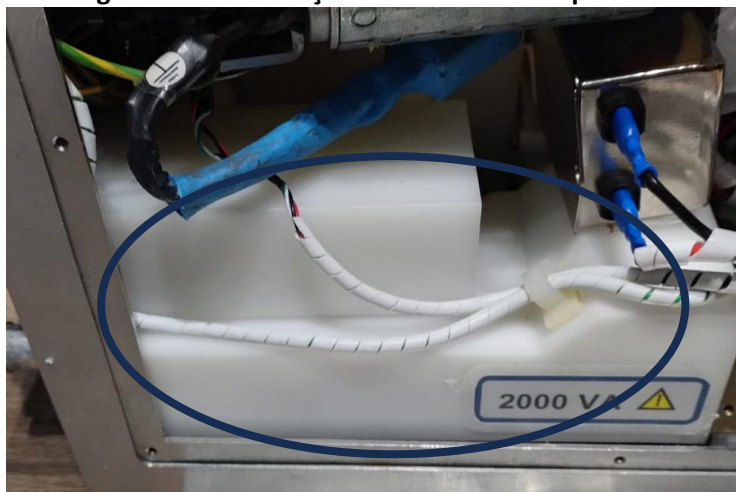
Figura 13 - MAX 31865



Fonte: próprio autor

Na **Figura 14** temos a localização do PT100 dentro da marcação azul escuro, ele está exatamente dentro do tanque, para medir a temperatura da água que é utilizada para controlar a temperatura do sangue.

Figura 14 – Localização do sensor de temperatura



Fonte: próprio autor

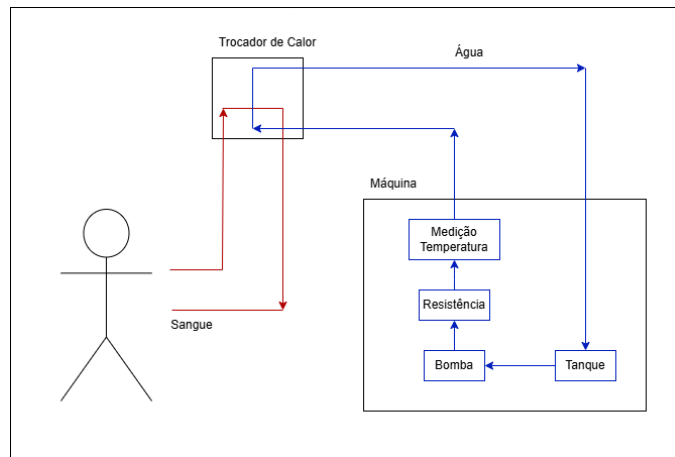
3.4. Atuadores

Os atuadores servem para o controle da temperatura do sangue e para bombear o sangue, temos basicamente três tipos de atuadores que serão apresentados a seguir.

Na

Figura 15 observa-se que o sangue entra na máquina, como o diagrama é só de fluido não aparece a bomba peristáltica, vê o sangue entrando e saindo pelo equipamento e passa pelo local onde a água que tem o sistema de controle de Temperatura, pode notar que tem uma resistência para o aumento da mesma, já a bomba serve para manter a pressão no sistema e liberar os sistema de arrefecimento quando a temperatura está elevada.

Figura 15 – Sistemas de Fluidos do equipamento



Fonte: próprio autor

3.4.1. Resistência

A resistência elétrica utilizada pelo equipamento de modelo Tubular em aço inoxidável, blindada, 2000W / 220Vac é utilizada para o aquecimento da água e possibilitar o controle da temperatura sanguínea pelo perfusionista. Uma vez que a resistência não pode entrar em contato direto com o sangue afim de evitar contaminações.

Na

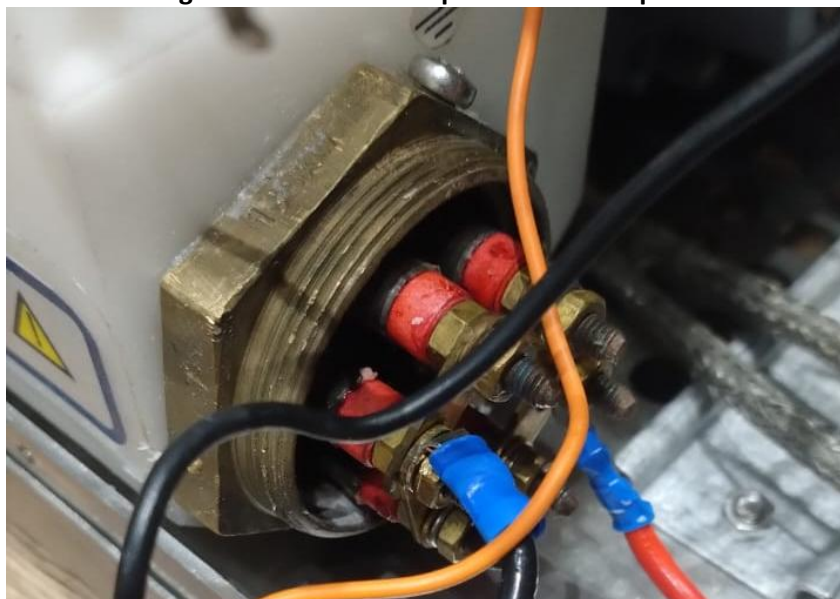
Figura 15 nota-se claramente que o sangue não entra em contato com o sistema e temos um sistema de aquecimento através de uma resistência. Isso ocorre devido ao efeito Joule: “é o nome dado ao fenômeno o aquecimento de um material devido à passagem de uma corrente elétrica.” [5], com essa corrente podemos aquecer a água que envolve essa resistência, como no chuveiro.

Na

Figura 16 observa-se a resistência presente na máquina, mas não foi possível executar essa parte da automação, mas tendo em vista que o trabalho se trata num estudo de caso para ver a viabilidade a resistência foi trocada por um LED, como vemos na

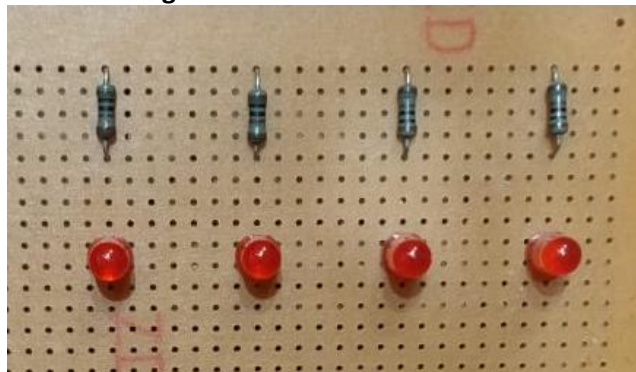
Figura 17.

Figura 16 – Resistência presente na máquina



Fonte: próprio autor

Figura 17 – Usado no Trabalho



Fonte: próprio autor

3.4.2. Bomba d'água

A bomba d'água utilizada é do tipo Centrífuga de pressão interna máxima de 30 kPa e tem por objetivo pressurizar todo o sistema de água da máquina. Sistema esse que mantém a temperatura conforme a indicação do perfusionista controlando o aumento (Resistência) ou a diminuição (água gelada presente no tanque) do sistema. Na

Figura 18 observa-se ver a entrada e saída para o trocador de calor.

Figura 18 – Entrada e saída de água do equipamento



Fonte: próprio autor

Na Figura 19 observa-se a Bomba presente na máquina, mas, novamente, não foi possível executar essa parte da automação, mas tendo em vista que o trabalho se trata num estudo de caso para ver a viabilidade assim, como a resistência foi trocada, por um LED.

Figura 19 – Bomba d'água presente dentro do equipamento



Fonte: próprio autor

3.4.3. Bomba Peristáltica

Primeiro devemos compreender seu funcionamento básico, ela é um tipo especial de bomba ideal para aplicações médicas, já que ela não tem contato com o fluido que ela bombeia, diferente da bomba d'água que no maquinário dela passa a água, a bomba peristáltica permite o bombear sem o contato.

“Uma bomba peristáltica é uma bomba que utiliza um processo de compressão e relaxamento para mover fluidos de um lugar para outro. A bomba é composta por um tubo flexível (ou mangueira) que é comprimido por roletes, que ao rodar, criam um vácuo que puxa o fluido para dentro do tubo. Quando a rotação continua, o fluido é empurrado para fora do tubo. Este processo é repetido continuamente, permitindo um fluxo constante de fluido.” [4].

Vemos na Imagem do próprio equipamento os roletes que pressionam o tubo criando o sistema peristáltico (

Figura 20).

Figura 20 – Bomba de Rolete ou bomba peristáltica

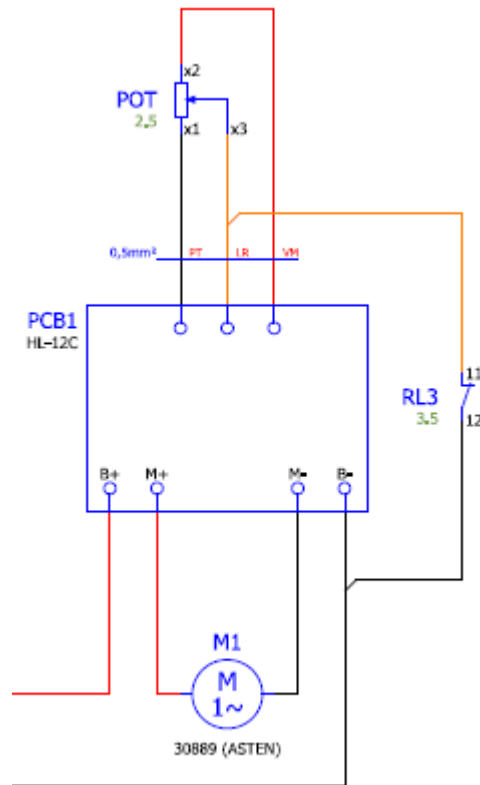


Fonte: próprio autor

O sistema dessa bomba está diretamente ligado a função vital que o aparelho vai fazer, logo essa função não está ligada ao controlador, mas sim diretamente ao motor, como podemos ver na Figura 21, mas ainda assim há um momento que o microcontrolador intervém nesse dispositivo, quando a tampa é aberta, para evitar esmagar o dedo do operador. Na figura citada observa-se que quando a chave RL3 for acionada o potenciômetro pode controlar livremente a velocidade, caso o contrário o pino será aterrado não gerando a diferença de potencial

necessária para controlar o motor. Ele é alimentado com 24 V através de um PCB desenvolvido pela própria empresa fornecedora da máquina. O CLP e futuramente Microcontrolador atua num Relé mostrado na **Figura 21**.

Figura 21 - Motor da bomba no esquema elétrico

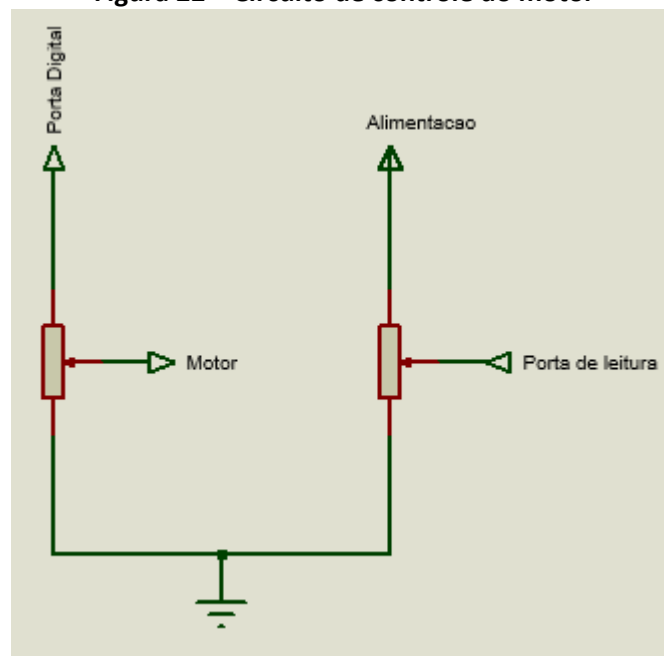


Fonte: CARDIO MEDICAL INDUSTRIAL (2021) [17]

No circuito presente nesse trabalho encontramos exatamente conforme a

Figura 22 em que o microcontrolador controlará a alimentação do potenciômetro que controla a bomba, mas percebe-se um segundo potenciômetro, pois quando ocorre o desligamento desse motor num caso de temperatura excedida o permitido ao corpo humano (entorno de 45°C) ele desliga o equipamento e para volta a operar deve zerar o potenciômetro para a segurança do paciente que está ligado a máquina.

Figura 22 – Circuito de controle do motor



Fonte: próprio autor, usando o Software Proteus

Novamente não foi possível fazer todos os sistemas de controle operacional como um equipamento final, mas para a validação que realmente está funcionando foi utilizado LEDs, veja a

Figura 17.

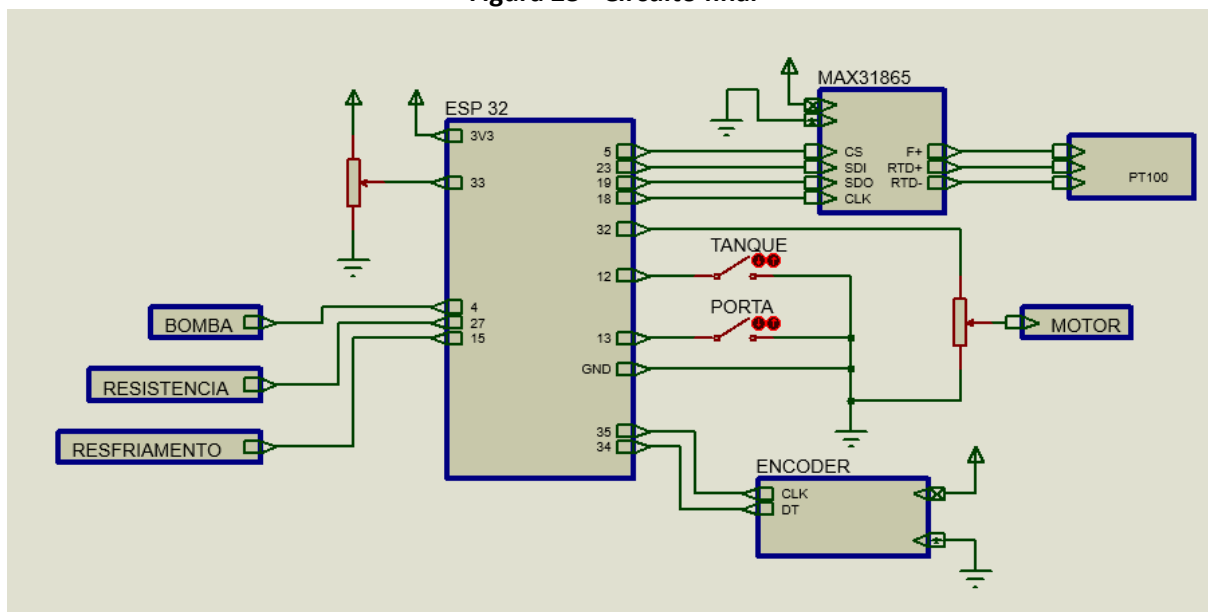
3.5. Construção

A montagem segue como base a estrutura já existente da máquina, com a diferença da ESP 32 controlando tudo ao invés do CLP. Mantendo a aparência do equipamento como podemos ver na Figura 1.

A montagem do circuito final utilizando cada conceito apresentado obteve-se a

Figura 23. Cada sensor e atuador está presente mostrando sua conexão com a ESP 32.

Figura 23 - Circuito final



Fonte: próprio autor

A princípio os LEDs foram utilizados como símbolo da Bomba, Resistência e Resfriamento, mas estes não serão mais utilizados quando forem substituídos por reles de controle de resistência e bomba d'água.

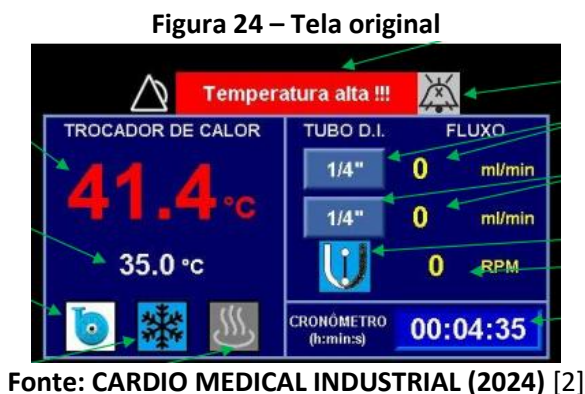
3.6. Aplicativo para substituição de painel LCD.

Em paralelo com a construção física do equipamento, foi desenvolvido um aplicativo para celular na plataforma App Inventor do MIT [18], esse ambiente de desenvolvimento é online e toda lógica feita em blocos, não é uma das melhores cabendo um futuro desenvolvimento de um aplicativo melhor.

Optou-se por ser uma maneira simples e online, fora ter uma vasta documentação presente na internet para consultas. A programação estará em anexo com breves comentários a respeito da funcionalidade de cada trecho.

Na

Figura 24 pode-se comparar a Tela desenvolvida originalmente pela empresa, algumas animações presentes na tela original não foram possíveis reproduzir, mas como são detalhes além da funcionalidade não foi um problema.



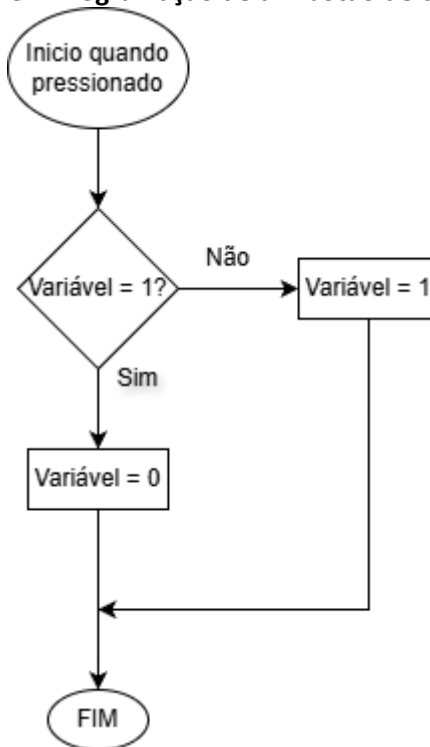
Nos últimos testes o aplicativo se mostrou satisfeito para a execução futuramente desse equipamento para um produto de mercado, podendo se tornar um equipamento da família CMCAP.

Antes de qualquer coisa, por segurança para o aplicativo não enviar dados que possam prejudicar o paciente, todos os botões estão desabilitados para alterações, impedindo de configurar eles sem estar conectado ao equipamento.

No aplicativo temos três botões de controle localizados na metade da esquerda inferior, na programação desses botões temos o fluxograma presente na

Figura 25, nota-se aqui de uma forma genérica a programação de todos os botões, onde temos uma variável que controla a função do botão e quando pressionada ele comuta o valor entre On/Off. Essas variáveis de tempos em tempos serão colocadas conforme o padrão de envio “_00.0,0,0,0”, onde o primeiro limitador é armazenado numa caixa de texto e os outros três números são justamente os botões de controle, em ordem, Bomba, Arrefecimento e Aquecimento, que controla dois atuadores que estão conectados ao equipamento.

Figura 25 – Programação de um botão de comando

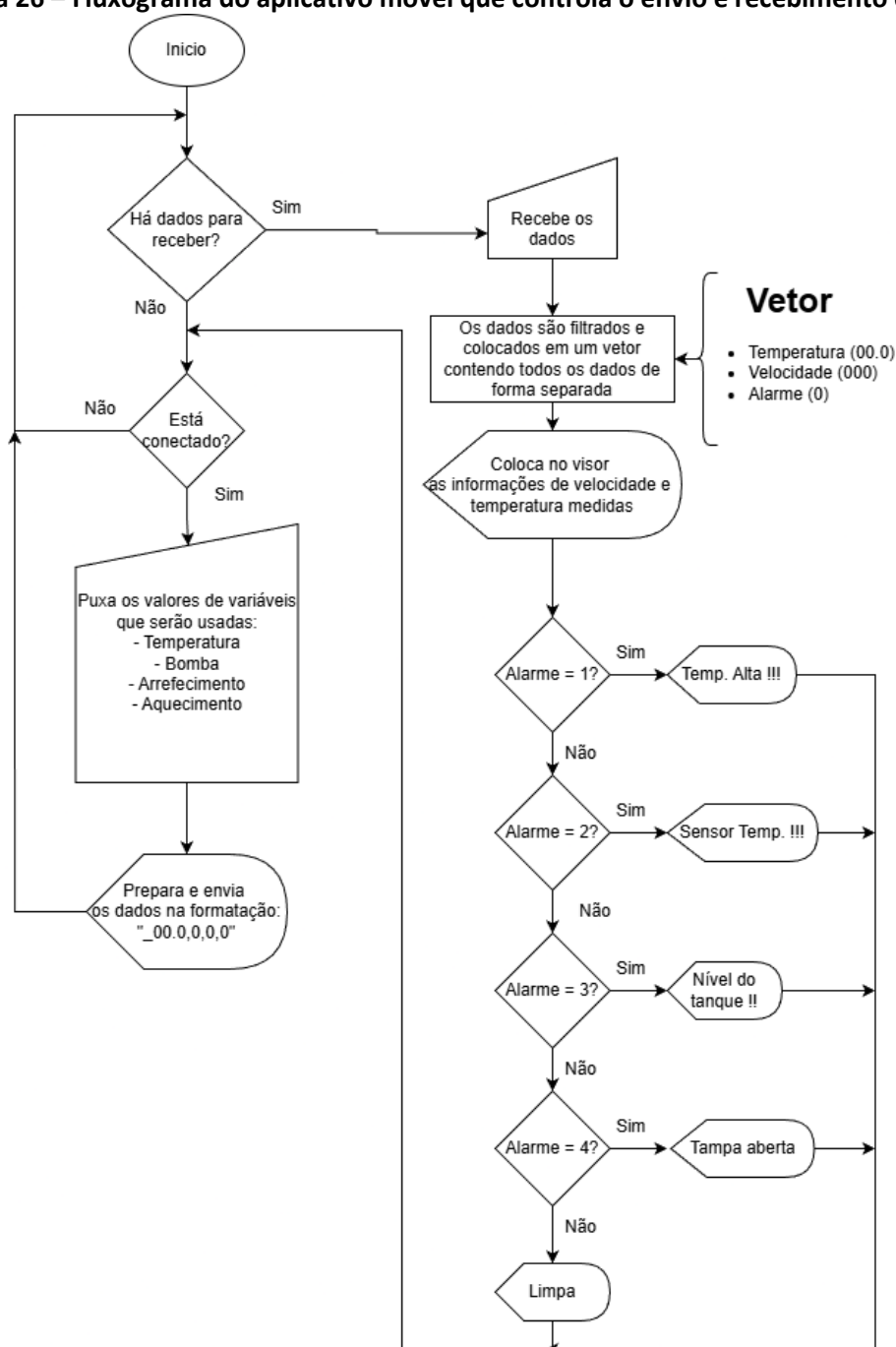


Fonte: próprio autor

Além disso, constantemente há um loop realizado pelo celular que recebe os dados e prepara para enviar ao ESP 32 como podemos ver na

Figura 26, o ciclo representado acontece a cada 20 ms garantindo o envio ao celular dos dados, observa-se que quando inicia o loop ele analisa se há mensagens chegando via *bluetooth* (ESP 32 para Celular) caso chegue, vai separar as informações vindas do pacote num vetor onde teremos a Temperatura, Velocidade e Alarme, colocando as informações na tela e em suas respectivas áreas e fazendo um filtro identificando o código do alarme (Tabela 5 e Tabela 6).

Figura 26 – Fluxograma do aplicativo móvel que controla o envio e recebimento de dados



Fonte: próprio autor

Os alarmes representados seguem uma lógica simples e numérica para identificar falhas que podem ser prejudiciais à saúde do paciente como para o operador, podemos observar nas Tabela 5 e Tabela 6 que existem quatro tipos de alarmes e entre eles alertas que são avisos que não tem grande importância para a cirurgia, mas deve ter um aviso sobre, média que já requer uma atenção e resolução de problema ocasionado e por fim os de alta prioridade que requer uma ação a ser tomada pra a segurança do paciente.

Tabela 5 - Alarmes

Número do alarme	Texto apresentado	Tipo de falha	Prioridade
0	-	Não há alarme	-
1	Temp. Alta !!!	Temperatura excedeu o máximo permitido de 43 °C	Alta
2	Sensor Temp. !!!	Falha na medição da temperatura ou no próprio sensor	Alta

Fonte: próprio autor

Tabela 6 - Alarmes

Número do alarme	Texto apresentado	Tipo de falha	Prioridade
3	Nível do tanque !!	Nível do tanque está baixo podendo trazer riscos a cirurgia	Média
4	Tampa aberta	Tampa de proteção da bomba peristáltica aberta	Alarme

Fonte: próprio autor

Além desses, que são a prioridade do aplicativo tem um cronômetro que na tela original fica no canto inferior direito (

Figura 24) que funciona quando pressionado inicia o cronômetro e ao pressionar novamente ele pausa, novamente reinicia a contagem. Ter um cronômetro para acompanhar o tempo de cirurgia do paciente e ver quanto tempo foi necessário para completar o procedimento. Ele funciona como um básico de relógio digital a cada 1 segundo ele soma 1 na variável de segundos que quando completa 60, soma 1 na variável de minuto e o resto consecutivamente.

3.7. Programação da ESP 32

Todo desenvolvimento da programação foi baseado em uma programação simplificada do CLP já existente no projeto original. Como observa na **Figura 27**, percebe-se as contadoras de um CLP fazendo exatamente o que foi projetado para esse equipamento fazer, onde tem-se um

sistema de travamento em casos de alarme e outros erros que possam prejudicar a operação, para exemplificar pode observar que o nível do tanque quando acionado deve barrar certas operações sendo uma da primeiras contadoras normalmente fechada que aparece para a continuação dos comandos da máquina, isso serve para ver da onde vem a proposta original da máquina. Para os alarmes que estão descritos na programação Ladder abaixo, todos estão descritos no manual no equipamento, **Figura 28**.

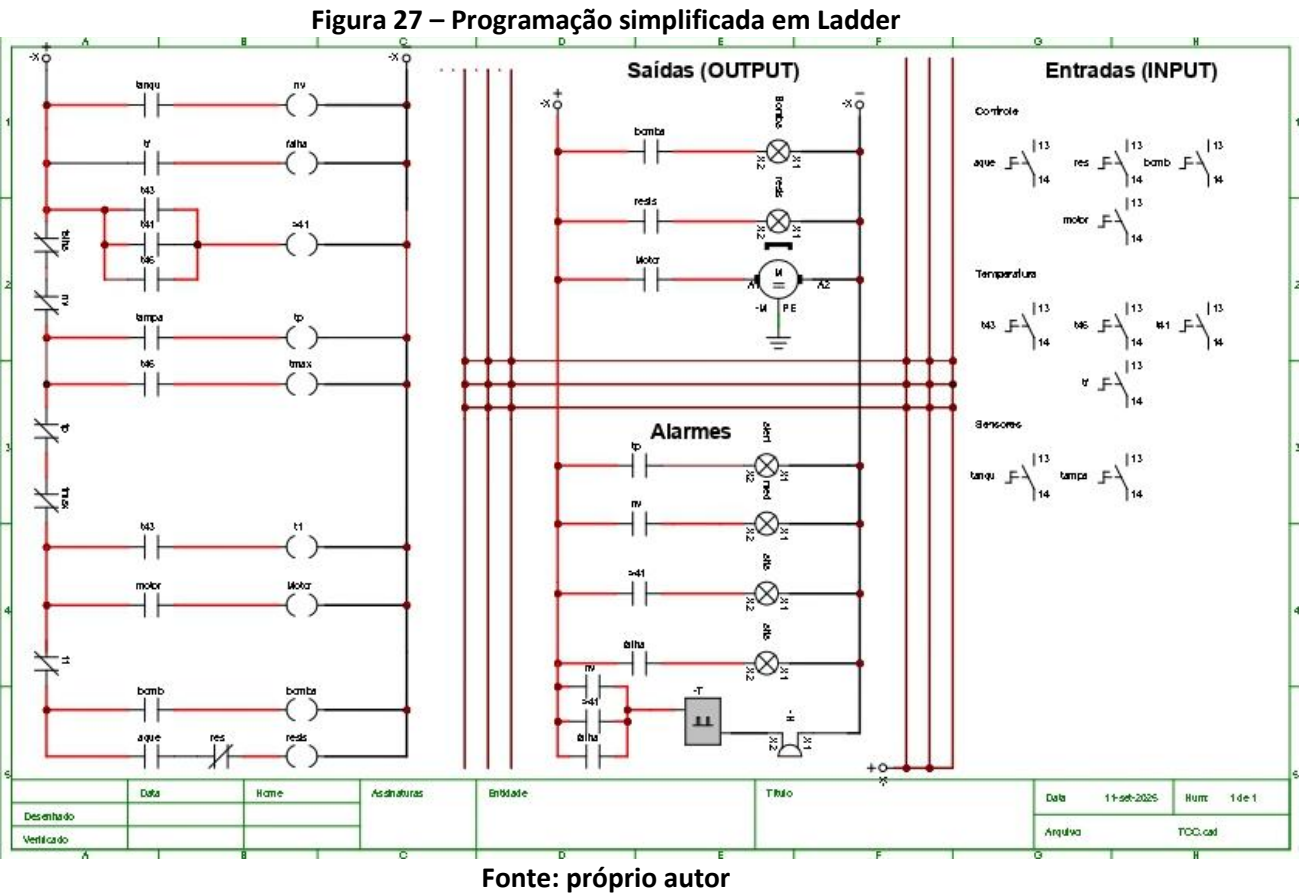


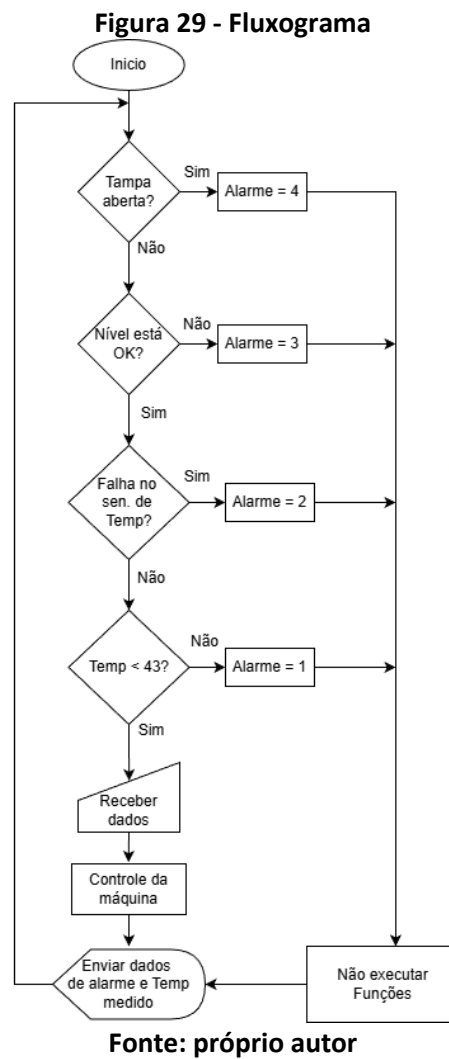
Figura 28 – Alarmes no manual

OCORRÊNCIA	INDICAÇÃO VISUAL
Temperatura d'água maior que 41°C	 Temp. Alta !!! Pisca na frequência de 2Hz
Falha no sensor de temperatura	 Sensor Temp. !!! Pisca na frequência de 2Hz
Nível do tanque d'água abaixo do mínimo	 Nível do tanque !! Pisca na frequência de 0,5Hz
Tampa da capapa	Tampa aberta Constante

Fonte: CARDIO MEDICAL INDUSTRIAL (2024)

A partir da programação em Ladder mencionada na

Figura 24 foi possível transcrever ela para C considerando a linguagem do Arduino e seguindo o fluxograma, **Figura 29**. A programação estará em anexo.



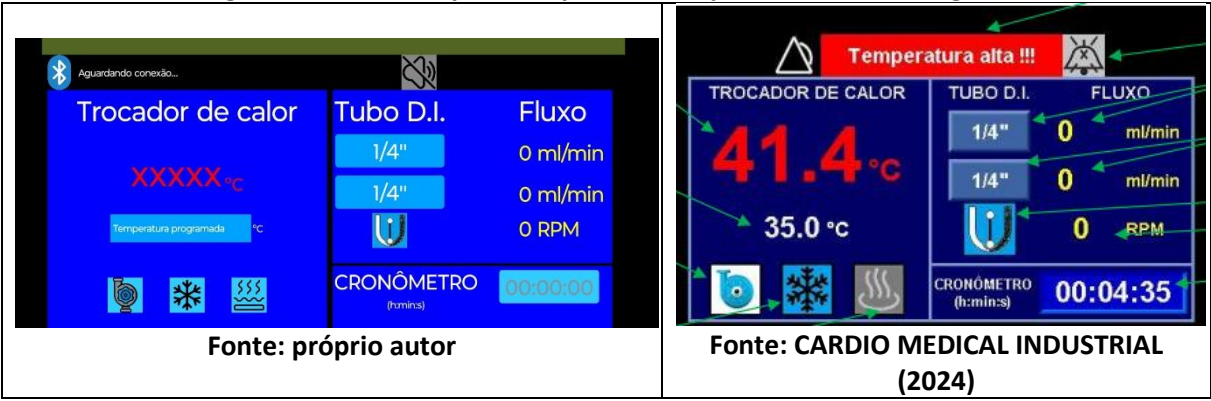
Fonte: próprio autor

4. RESULTADOS E DISCUSSÕES

Durante o desenvolvimento do projeto, os testes realizados confirmaram a viabilidade da substituição do CLP do Módulo de Cardioplegia CMCAP por um microcontrolador ESP32, utilizando comunicação Bluetooth para controle e monitoramento via aplicativo móvel, na Figura 30 observa-se a tela principal desse aplicativo desenvolvido que resultou em algo parecido com o que já tinha

Figura 24. O sistema apresentou desempenho estável nas simulações e respostas rápidas aos comandos, mantendo o fluxo de dados constante entre o aplicativo e a placa. O que demonstra que o ESP32 é capaz de atender aos requisitos de tempo real necessários para o controle do equipamento.

Figura 30 - Tela do Aplicativo para celular junto com a tela original



A etapa de comunicação foi um dos pontos de maior relevância do projeto. A transmissão de dados via Bluetooth Classic mostrou-se eficaz para curtas distâncias, apresentando estabilidade durante os ensaios, ainda que tenha sido observada uma leve interferência em determinados momentos, ocasionada por dispositivos próximos operando na mesma faixa de 2,4 GHz. Essa limitação, contudo, não comprometeu o funcionamento geral do sistema, já que o equipamento se encontra numa sala cirúrgica que são ambientes controlados e que não será possível a entrada de outros celulares ou fontes de interferência para essa comunicação. Assim,

verificou-se que a escolha do Bluetooth foi adequada para a proposta de mobilidade e simplicidade operacional do projeto.

O aplicativo desenvolvido no App Inventor demonstrou funcionalidade satisfatória, replicando as principais operações do painel físico original, como controle da bomba peristáltica, ativação do sistema de aquecimento e resfriamento e monitoramento da temperatura. Embora o ambiente de desenvolvimento utilizado apresente restrições gráficas e de desempenho, o aplicativo cumpriu seu papel como interface de teste, evidenciando o potencial de substituição do painel touchscreen original por uma interface móvel de baixo custo. Porém, para futuras mudanças é viável o estudo a respeito de um aplicativo feito de uma maneira mais eficiente e com maior robustez que não possa travar durante a operação, já que o App Inventor pode dar erros com certas facilidades.

De modo geral, os resultados obtidos demonstram que o uso do microcontrolador ESP32, aliado a uma interface móvel, é uma solução tecnicamente viável e economicamente vantajosa para modernizar o sistema de controle do CMCAP. Ainda que não tenha sido implementado o controle físico de todos os atuadores do equipamento real, as simulações e os testes realizados comprovaram a eficiência do modelo proposto.

No que tange à segurança, qualquer outro erro que possa ser ocasionado sem ser por comunicação não foi avaliado visto que esse equipamento já possui a certificação nas normas:

- ABNT NBR IEC 60601-1:2010 + Emenda 1:2016: Requisitos gerais para segurança básica e desempenho essencial;
- ABNT NBR IEC 60601-1-2:2017: Requisitos gerais para segurança básica e desempenho essencial – Norma colateral: Compatibilidade eletromagnética - Requisitos e ensaio;
- ABNT NBR IEC 60601-1-6:2011: Requisitos gerais para segurança básica e desempenho essencial – Norma colateral: Usabilidade;
- ABNT NBR IEC 60601-1-8:2014: Requisitos gerais para segurança básica e desempenho essencial – Norma colateral: Requisitos gerais, ensaios e diretrizes para sistemas de alarme em equipamentos eletromédicos e sistemas eletromédicos;

- ABNT NBR IEC 60601-1-9:2014: Requisitos gerais para segurança básica e desempenho essencial – Norma colateral: Prescrições para um projeto eco responsável.
- IEC 62304:2006: Medical device software Software life cycle processes.

Conforme o que está no manual de operação do equipamento [2], logo não é necessário refazer análise de risco desses itens, mas, quando for realmente virar um produto já que se mostrou viável, deve ser adicionada uma camada de falha de comunicação que deve ser tão importante quanto a falta da energia elétrica e atender as normas para tecnologia bluetooth legislada pela ANATEL:

- Resolução ANATEL nº 680, de 27 de junho de 2017;
- Ato ANATEL nº 14448, de 4 de dezembro de 2017;
- Ato ANATEL nº 237, de 7 de janeiro de 2022.

O equipamento deverá passar por uma nova certificação, passando por ensaios, gerenciamento de risco e novo registro.

Também é interessante ressaltar que o equipamento deve ser operado sempre com a presença e uma pessoa capacitada e que entenda sobre cardioplegia e como funciona o dispositivo. Essa pessoa tem toda a autoridade para controlar manualmente, se necessário, a máquina, até mesmo em leituras que ele acredita estar erradas no visor, que agora se tornou o celular.

5. CONCLUSÃO

O presente trabalho alcançou o objetivo de estudar a viabilidade técnica da substituição do CLP do Módulo de Cardioplegia CMCAP por um microcontrolador ESP32, integrando-o a uma interface móvel via comunicação Bluetooth. Observando os resultados obtidos e discutidos, podemos concluir que é possível criar um equipamento visando a retirada do CLP e, no lugar, colocar o ESP 32, já que ele demonstrou uma boa estabilidade na transmissão de dados, resposta rápida aos comandos e integração eficiente com o aplicativo desenvolvido.

A avaliação do sistema mostrou que o uso do Bluetooth Classic é uma alternativa prática e funcional para o controle de equipamentos biomédicos. Essa tecnologia proporcionou uma comunicação confiável e de fácil emparelhamento, eliminando a necessidade de infraestrutura de rede e favorecendo a mobilidade do operador, além de ser simples a programação para desenvolvimento de novos equipamentos. Apesar das limitações associadas à faixa de frequência de 2,4 GHz, como interferências ocasionais, os resultados demonstraram que o desempenho obtido é suficiente para as exigências do módulo de cardioplegia, equilibrando simplicidade, eficiência e segurança.

Conclui-se, portanto, que a substituição do CLP por um microcontrolador como o ESP32 representa um avanço significativo na automação de equipamentos médicos, oferecendo uma alternativa de baixo custo e fácil manutenção. A proposta apresentada pode contribuir para futuras atualizações na linha de produtos da Cardio Medical Industrial (CMI), servindo de base para estudos posteriores que visem aprimorar a robustez, a segurança e a certificação do sistema em ambiente hospitalar. Recomenda-se, como trabalhos futuros, a implementação prática em um protótipo clínico, a validação em situações reais de operação.

REFERÊNCIAS

- [1] P. D. D. M. Braile, “SciELO,” 02 Março 2011. [Online]. Available: <https://www.scielo.br/j/rbccv/a/Jjj7GqLXqgLG75Vd3Vts3fz/?lang=pt>. [Acesso em 09 Dezembro 2025].
- [2] CARDIO MEDICAL INDUSTRIAL, *Manual de Operação do Módulo de Cardioplegia CMCAP*, Piracicaba - SP, 2024.
- [3] D. P. PRATES, “Pequena história da cirurgia cardíaca: e tudo aconteceu diante de nossos olhos...,” *Rev Bras Cir Cardiovasc*, pp. 177-184, 1999.
- [4] MILAN EQUIPAMENTOS, “MILAN EQUIPAMENTOS,” MILAN EQUIPAMENTOS, Agosto 2024. [Online]. Available: <https://milan-ec.com.br/lorem-ipsum-dolor-sit-amet-consectetur-adipiscing-elit-2/>. [Acesso em 10 Novembro 2025].
- [5] A. C. d. Lourenço, E. C. A. Cruz e S. C. júnior, “Efeito Joule,” em *Circuitos em Corrente Contínua*, São Paulo, Érica LTDA, 1996, p. 110.
- [6] J. G. A. Murta, “Conhecendo o ESP32 – Introdução (1),” Blog Eletrogate, 21 Fevereiro 2024. [Online]. Available: <https://blog.eletrogate.com/conhecendo-o-esp32-introducao-1/>. [Acesso em 17 Maio 2025].
- [7] R. Mischianti, “Mischianti,” Renzo Mischianti , 6 Agosto 2025. [Online]. Available: <https://mischianti.org/doit-esp32-dev-kit-v1-high-resolution-pinout-and-specs/>. [Acesso em 12 Novembro 2025].
- [8] N. C. BRAGA, “NEWTON C. BRAGA,” 17 Maio 2025. [Online]. Available: <https://www.newtoncbraga.com.br/microcontroladores/143-tecnologia/17117-comunicando-se-via-mqtt-com-o-esp32-mic404.html>.
- [9] C. THORBECKE, “Décadas após criação, Bluetooth ainda é “uma tecnologia complicada”,” CNN Brasil, 11 Julho 2022. [Online]. Available: <https://www.cnnbrasil.com.br/tecnologia/decadas-apos-criacao-bluetooth-ainda-e-uma-tecnologia-complicada-entenda/>. [Acesso em 28 Outubro 2025].
- [10] Bluetooth, “Bluetooth,” Bluetooth, [Online]. Available: <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/>. [Acesso em 17 Novembro 2025].
- [11] Espressif Systems , “ESP32-WROOM-32 Datasheet Version 3.6,” Espressif Systems , Shanghai, 2025.
- [12] P. L. G. Loureiro, “Aula de Sistemas Robóticos,” em *Notas de Aula*, São José dos Campos, 2025.
- [13] K. A. R. Ismail, “Instrumentação Básica para Engenharia,” em *Instrumentação Básica para Engenharia*, Campinas - SP, UNICAMP, 1998, p. 315 até 320.
- [14] Unicamp, “Instrumentação e Controle,” Unicamp, [Online]. Available: <https://sites.fem.unicamp.br/~instrumentacao/termometroderesistencia04.html>. [Acesso em 04 Novembro 2025].
- [15] maxim integrated, “Datasheet MAX 31865,” 2015.
- [16] l. ada, “Adafruit MAX31865 RTD PT100 or PT1000 Amplifier,” Adafruit Industries, 2020.
- [17] Cardio Medical Industrial Ltda, “Diagrama de Ligações Elétricas - Cardioplegia,” Cardio Medical Industrial Ltda, Piracicaba - SP, 2021.

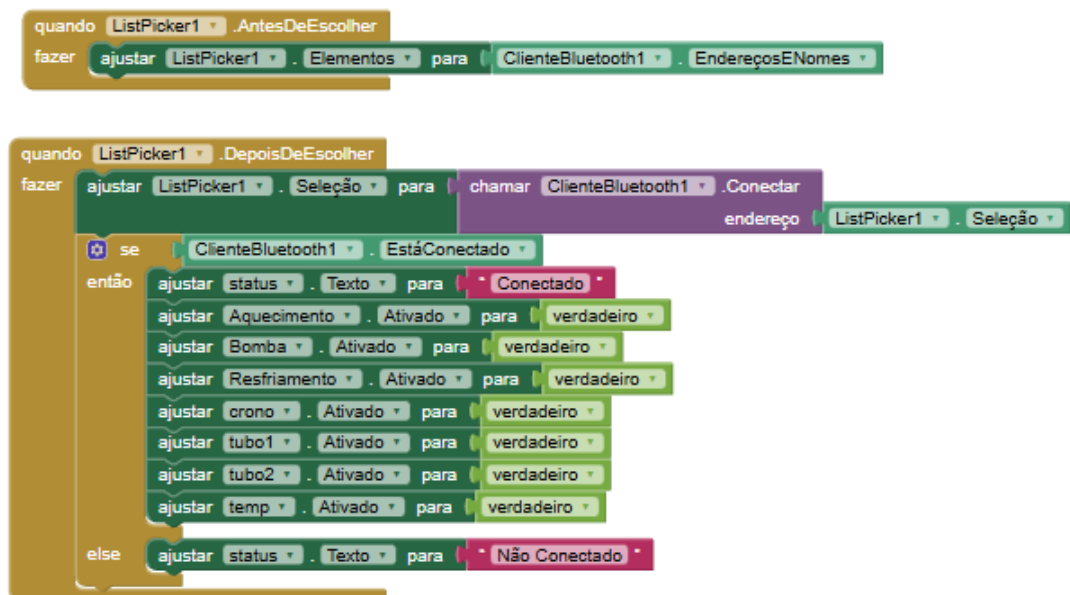
[18] MIT, “App Inventor,” MIT, Julho 2015. [Online]. Available: <https://appinventor.mit.edu/explore/library>. [Acesso em 17 maio 2025].

ANEXOS

5.1. Programação Aplicativo

Na **Figura 31** temos a parte em que cria a lista de dispositivos Bluetooth e, quando ele for selecionado, vai liberar os botões do aplicativo.

Figura 31 – Seleção do Bluetooth.



Fonte: próprio autor

Na **Figura 32** temos a inicialização das variáveis utilizadas ao longo do resto da programação.

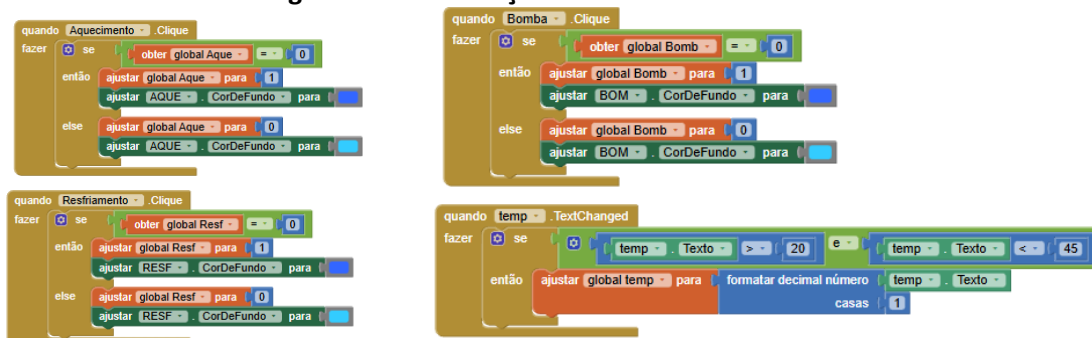
Figura 32 – Inicialização de Variáveis



Fonte: próprio autor

Na **Figura 33** vemos o controle dos botões de maneira independente atualizando os valores de variáveis.

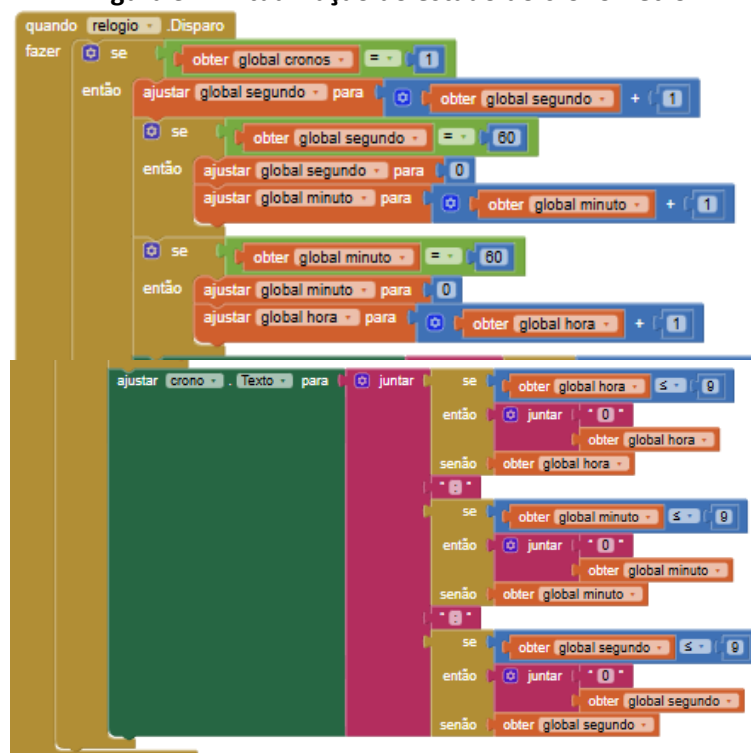
Figura 33 – Atualização do estado das Variáveis



Fonte: próprio autor

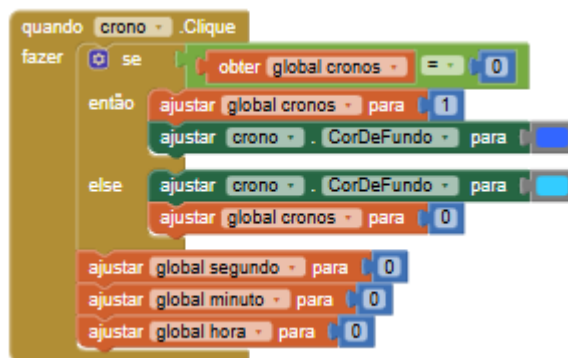
Durante as operações é importante saber o tempo transcorrido, logo a presença de um cronômetro é essencial, na **Figura 34** temos a atualização que ocorre com periodicidade sendo só habilitada para mostrar ao usuário caso o botão que habilita isso for pressionado, programação está que está na **Figura 35**.

Figura 34 – Atualização do estado do cronômetro



Fonte: próprio autor

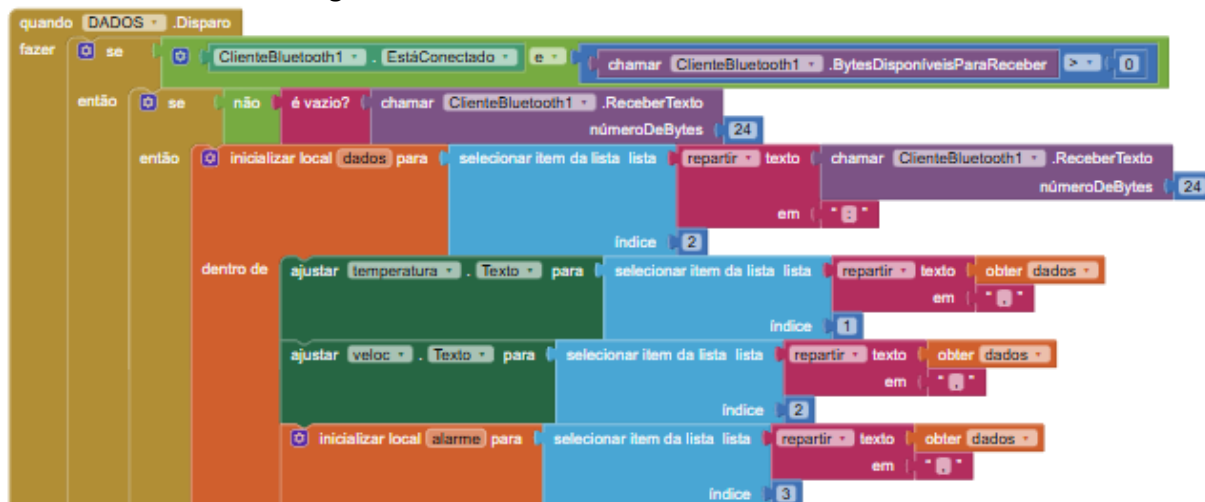
Figura 35 – Atualização do estado do cronômetro no visor



Fonte: próprio autor

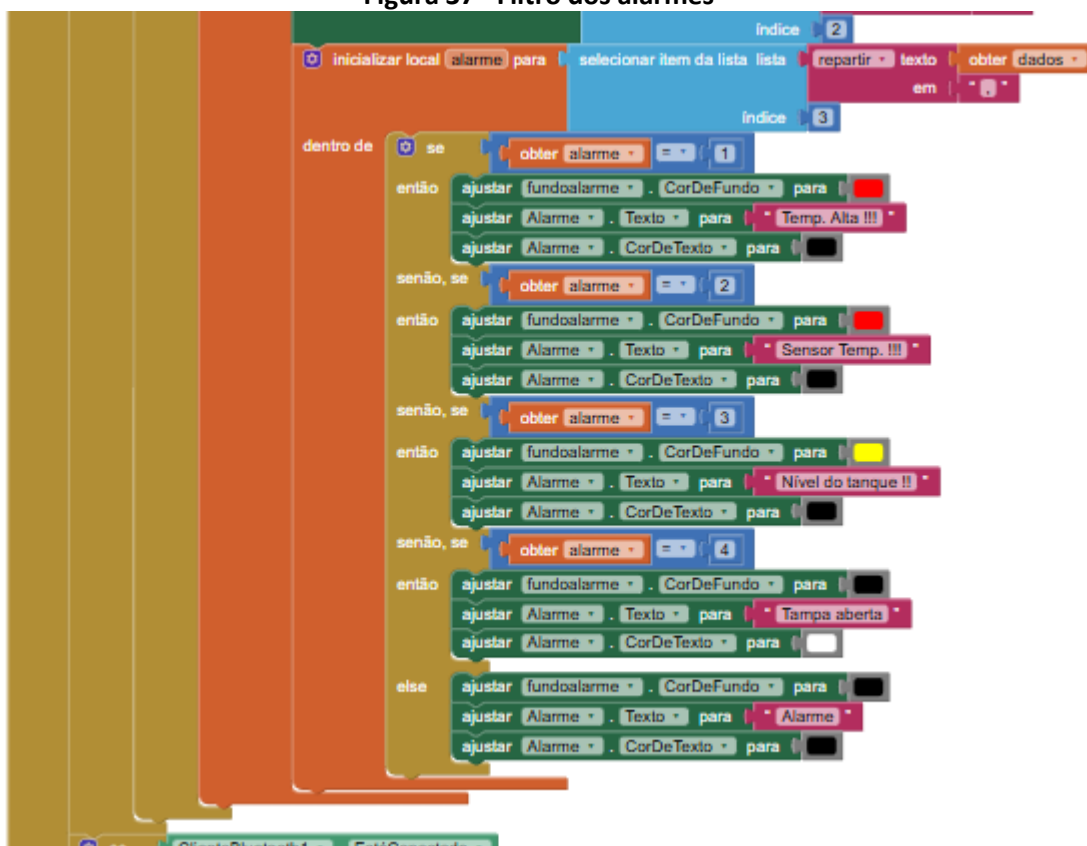
E para finalizar temos nas Figuras a seguir os envios e recebimento dos dados via bluetooth. Na **Figura 36** vemos o recebimento no formato “:00.0,000,0” onde temos o separador de pacotes “:”, o delimitador de informação “,” e três tipos de informações: Temperatura, Rotação e Alarme. Já na **Figura 37** observamos o filtro para alarmes e mostrar no visor.

Figura 36 – Recebimento e tratamento dos dados



Fonte: próprio autor

Figura 37– Filtro dos alarmes



Fonte: próprio autor

Na **Figura 38** tem a parte da programação dedicada ao envio de dados do aplicativo a ESP, mudando um pouco a formatação para evitar confusões “_00.0,0,0,0”, nessa formatação temos 4 variáveis importantes para o microcontrolador, a temperatura setada e a seleção para acionar ou não a resistência e a bomba de água para controle de temperatura.

Figura 38 – Envio dos dados



Fonte: próprio autor

5.2. Programação na ESP 32

```
#include "BluetoothSerial.h"
#include <Adafruit_MAX31865.h>

// Configuração do sensor MAX31865 (SPI por software)
Adafruit_MAX31865 thermo = Adafruit_MAX31865(5, 23, 19, 18);

// Constantes do sensor PT100
#define RREF 430.0
#define RNOMINAL 100.0

// Pinos digitais
int tanque = 12;
int porta = 13;
int led_aviso = 2;

// Temperatura limite máxima permitida
float temp_max = 50;

// Verificação de configuração Bluetooth
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

char recebida[24];
BluetoothSerial SerialBT;

// Pinos do encoder
#define CLK 35
#define DT 34

// Variáveis de leitura e controle
int pulseCount = 0;
unsigned long lastTime = 0;
float rpm = 0.0;
int velocidade = 0;
int alarme = 0;
const int PULSES_PER_REV = 20;

int controle_pot_alarme4 = 0;

// Pinos de atuadores
int motor = 32;
int potPin = 33;
int bomb = 4;
int resf = 15;
int resis = 27;
```

```

void setup() {
  Serial.begin(115200);
  delay(5000);

  Serial.println("Starting..");
  SerialBT.begin("MyESP32");
  Serial.println("The device started, now you can pair it with Bluetooth!");

  // Configuração de entradas e saídas
  pinMode(porta, INPUT_PULLUP);
  pinMode(tanque, INPUT_PULLUP);
  pinMode(led_aviso, OUTPUT);
  pinMode(bomb, OUTPUT);
  pinMode(resf, OUTPUT);
  pinMode(resis, OUTPUT);

  // Configura o MAX31865 para sensor de 3 fios
  thermo.begin(MAX31865_3WIRE);

  // Configuração do encoder
  pinMode(CLK, INPUT_PULLUP);
  pinMode(DT, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(CLK), handleEncoder, RISING);

  pinMode(motor, OUTPUT);
  pinMode(potPin, INPUT_PULLUP);
}

void loop() {
  float temperatura;
  alarme = 0;

  // Alarme 4 – Porta da bomba aberta
  if (digitalRead(porta) == 1) {
    alarme = 4;
    digitalWrite(led_aviso, 1);
    controle_pot_alarme4 = 1;
  } else {
    if (controle_pot_alarme4 == 0) {
      digitalWrite(motor, 1);

      // Alarme 3 – Nível do tanque baixo
      if (digitalRead(tanque) == 1) {
        alarme = 3;
        digitalWrite(led_aviso, 1);
      } else {
        // Medição de temperatura
        temperatura = medir_temperatura();
        Serial.println(falha_temperatura());
      }
    }
  }
}

```

```

// Alarme 2 – Falha de leitura ou temperatura muito alta
if (falha_temperatura() || temperatura >= 60) {
    alarme = 2;
    digitalWrite(led_aviso, 1);
    temperatura = 99.9;
} else {
    // Alarme 1 – Temperatura acima do limite configurado
    if (temperatura >= temp_max) {
        alarme = 1;
        digitalWrite(led_aviso, 1);
    } else {
        // Medição de RPM e operação normal
        velocidade = medir_rpm();
        digitalWrite(led_aviso, 0);

        // Se temperatura ultrapassar 45°C, o motor é parado
        if ((temperatura >= 45)) {
            pwm = 0;
            alarme = 1;
        } else {
            // Recebe e trata a mensagem Bluetooth
            strcpy(recebida, receber_mensagem());
            Serial.println(recebida);
            char msg[25];

            filtro(recebida, msg);

            float temp;
            int bomba, resfrige, resistencia;
            separarValores(msg, &temp, &bomba, &resfrige, &resistencia);

            Serial.print("Temperatura setada: ");
            Serial.println(temp);
            Serial.print("Bomba: ");
            Serial.println(bomba);
            Serial.print("resfrigeradora: ");
            Serial.println(resfrige);
            Serial.print("Aquecimento: ");
            Serial.println(resistencia);

            // Controle dos atuadores conforme comandos recebidos
            if (bomba == 1) digitalWrite(bomb, 1);
            else digitalWrite(bomb, 0);

            if (resfrige == 1) digitalWrite(resf, 1);
            else digitalWrite(resf, 0);

            if (resistencia == 1) digitalWrite(resis, 1);
            else digitalWrite(resis, 0);
        }
    }
}

```

```

    }
    }
} else {
    // Controle do motor via potenciômetro (modo de alarme 4)
    if (digitalRead(potPin) == 1) {
        digitalWrite(motor, 1);
        controle_pot_alarme4 = 0;
    } else {
        digitalWrite(motor, 0);
        controle_pot_alarme4 = 1;
    }
}
}

// Monta e envia mensagem de status via Bluetooth
char msg[12];
if (temperatura < 10) sprintf(msg, ":0%.1f,%03d,%d", temperatura, velocidade,
alarme);
else sprintf(msg, ":%.1f,%03d,%d", temperatura, velocidade, alarme);

Serial.println(msg);
enviar_mensagem(msg);

delay(20);
}

// --- Funções auxiliares ---

float medir_temperatura() {
    uint16_t rtd = thermo.readRTD();
    float ratio = rtd;
    ratio /= 32768;
    return thermo.temperature(RNOMINAL, RREF);
}

bool falha_temperatura() {
    uint8_t falha = thermo.readFault();
    if (falha) {
        Serial.print("Falha 0x");
        Serial.println(falha, HEX);
        return true;
    } else {
        digitalWrite(led_avisos, 0);
        return false;
    }
}

void enviar_mensagem(char var[15]) {
    for (int i = 0; i <= 10; i++) {

```

```

    SerialBT.write(var[i]);
}
}

char* receber_mensagem() {
    static char var[24];
    for (int i = 0; i <= 23; i++) {
        char c = SerialBT.read();
        if (((c >= '0') && (c <= '9')) || (c == '.') || (c == '_') || (c == ',')) {
            var[i] = c;
        }
    }
    var[24] = '\0';
    return var;
}

int medir_rpm() {
    if (millis() - lastTime >= 1000) {
        detachInterrupt(digitalPinToInterrupt(CLK));
        float pulsesPerSecond = (float)pulseCount / 1.0;
        rpm = (pulsesPerSecond / PULSES_PER_REV) * 60.0;
        pulseCount = 0;
        lastTime = millis();
        attachInterrupt(digitalPinToInterrupt(CLK), handleEncoder, RISING);
    }
    return int(rpm);
}

void handleEncoder() {
    pulseCount++;
}

void filtro(char msgrecebida[], char msg[]) {
    int controle = 0;
    int n = 0;
    for (int i = 0; i < strlen(msgrecebida); i++) {
        if (controle == 1 || msgrecebida[i] == '_') {
            if (controle == 1 && msgrecebida[i] == '_') {
                controle = 0;
                msg[n] = '\0';
            } else {
                msg[n] = msgrecebida[i];
                n++;
                controle = 1;
            }
        }
    }
}

void separarValores(char *mensagem, float *v1, int *v2, int *v3, int *v4) {

```

```
if (mensagem[0] == '_') {  
    mensagem++;  
}  
  
char *parte = strtok(mensagem, ",");  
if (parte != NULL) *v1 = atof(parte);  
parte = strtok(NULL, ",");  
if (parte != NULL) *v2 = atoi(parte);  
parte = strtok(NULL, ",");  
if (parte != NULL) *v3 = atoi(parte);  
parte = strtok(NULL, ",");  
if (parte != NULL) *v4 = atoi(parte);  
}
```