



RELATÓRIO DO PROJETO

Instrumento de Aferição do Potenciometro de Joystick

Orientador: Elias Guimarães Miranda Barbosa da Silva

Aluno: Leonardo Romão Queiroz Araújo

Matrícula: TIC370100406

Vitoria da Conquista

27 de fevereiro de 2025

Sumário

1	Escopo do Projeto	2
1.1	Apresentação do projeto	2
1.2	Objetivos do projeto	2
1.3	Descrição de funcionamento	2
1.4	Justificativa	3
1.5	Originalidade	3
2	Especificação do Hardware	5
2.1	Descrição de cada bloco e os pinos	5
3	Especificação do Firmware	6
3.1	Funções	6
3.2	Fluxograma	6
3.3	Descrição do Código	7
4	Execução do Projeto	10
4.1	Metodologia	10
4.2	Testes e Validação	11
4.3	Discursão dos Resultados	11
5	Referências	12

1 Escopo do Projeto

1.1 Apresentação do projeto

Neste trabalho, foi desenvolvido um equipamento para garantir o conserto, calibração e substituição do módulo joystick. Este dispositivo eletromecânico sofre, naturalmente, desgaste ou acúmulo de poeira em seus potenciômetros. Isso ocorre devido ao uso, o que, por consequência, pode causar mau funcionamento. Além disso, em dispositivos mais modernos, os joysticks utilizam o efeito Hall, sendo mais resistentes ao uso, pois não empregam potenciômetros. Contudo, é necessária calibração prévia para maior precisão. Esses controles são geralmente utilizados em diversas máquinas, como: drones, cadeiras elétricas, equipamentos médicos, videogames e brinquedos.

Dessa forma, foi desenvolvido um software para a placa de desenvolvimento **Bitdoglab**, pois nela é possível construir um protótipo funcional com os componentes nela presentes, como: o módulo joystick, o display SSD1306, o buzzer e os botões. O software, desenvolvido na linguagem C, permite a modulação e a comunicação entre os componentes.

1.2 Objetivos do projeto

O principal objetivo do projeto é a criação do protótipo funcional de um equipamento que auxilia o usuário a identificar ou garantir o funcionamento do módulo joystick. Para isso, é necessário cumprir os seguintes requisitos nos periféricos:

- Uso do periférico ADC para captura de valores do modulo joystick.
- Uso do Display ssd1306 para auxiliar o usuário utilizar o equipamento.
- Uso dos botões A e B para as ações do programa.
- Uso do Buzzer para sinalizar o usuário.

1.3 Descrição de funcionamento

O equipamento informa ao usuário, por meio do display, que ele deve mover o joystick para a esquerda, direita, cima e baixo. O usuário, então, posicionará o joystick na direção indicada e apertará o botão A da placa **Bitdoglab**. Após informar essas quatro direções, serão exibidos os valores lidos pelo ADC. Em seguida, a posição do joystick referente aos eixos X e Y será ilustrada no display. Também será possível selecionar, individualmente, qual eixo será exibido ao pressionar os botões A e B. Dessa forma, é possível testar o módulo joystick de maneira prática, identificando qual potenciômetro precisa ser trocado ou garantindo seu bom funcionamento antes de soldá-lo.

1.4 Justificativa

Quando o módulo joystick de um controle é danificado, é necessária a substituição do módulo ou do controle. Por ser mais barato, costuma-se realizar apenas a troca do módulo por um novo ou a substituição do potenciômetro, conforme ilustrado na figura a seguir. O problema ocorre no processo de troca, no qual, muitas vezes, é necessário soldar e dessoldar o componente, o que é um processo tedioso e demorado. Surge, então, a necessidade de testar os potenciômetros antes de soldá-los na placa do controle, para garantir seu funcionamento correto.

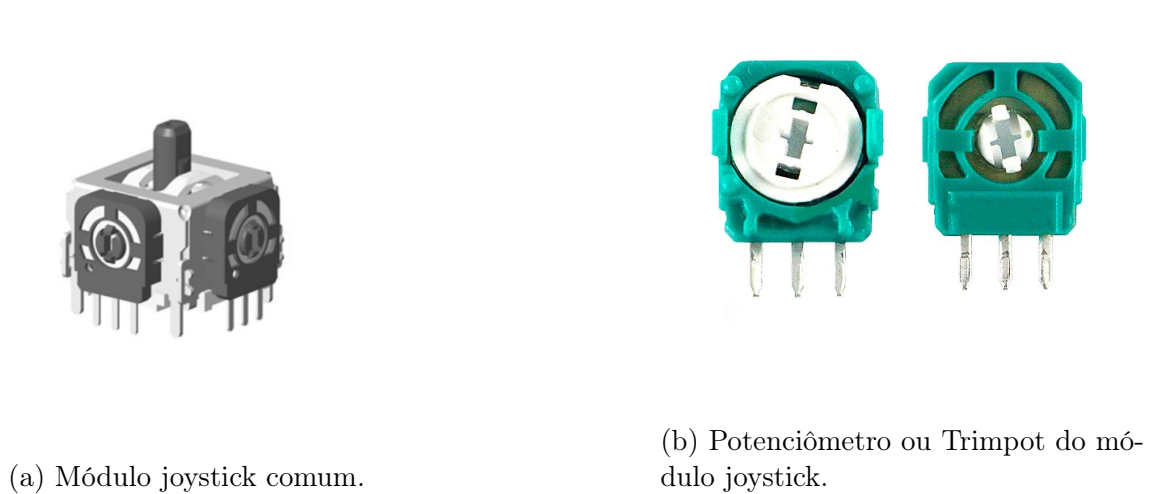


Figura 1: Exemplo dos dispositivos centrais

Dada essa necessidade específica, pensou-se em desenvolver um aparelho para testar esse equipamento antes de fixá-lo definitivamente no controle. Caso contrário, não seria possível comprovar seu bom funcionamento, pois ainda não foi criado um equipamento comercial específico para essa finalidade. Outra razão é o fato de os fabricantes de controles, muitas vezes, não terem interesse em reparo e criarem produtos descartáveis, incentivando a compra de um controle novo. O real problema ocorre quando um produto deixa de ser fabricado, elevando o custo ou tornando impossível a aquisição de um controle novo, tornando-se, então, indispensável o reparo.

Por fim, os módulos joystick de efeito Hall precisam ser calibrados, normalmente via software, para acompanhar o posicionamento ideal do ímã interno, o que também pode ser aplicado neste equipamento.

1.5 Originalidade

O equipamento desenvolvido neste trabalho é para reparo, conserto ou substituição de potenciômetro ou de todo o módulo joystick. Portanto, é destinado a um público

bastante específico e não possuem trabalhos diretamente correlatos que fazem uso de microcontroladores para auxiliar nesta atividade. O dispositivo comercial mais próximo que encontrei encontra-se na figura 2 a baixo. Possui pouquíssima informação sobre este equipamento, então supõe que ele seja apenas uma ponte que facilite a conexão do osciloscópio com a placa do controle, para, então, obter os dados do módulo joystick ligado diretamente na placa, apenas.



Figura 2: Dispositivo comercial mais próximo em função do tema do projeto.

Contudo, existem alguns softwares muito próximos da funcionalidade deste trabalho. Ferramentas como o *Gamepad Tester* são utilizadas para verificar o funcionamento dos módulos de joysticks. Além disso, os videogames mais recentes possuem a função de testar o funcionamento correto dos joysticks em seu firmware. Entretanto, os demais tipos de controle carecem dessa função de verificação.



Figura 3: Ilustração do software Gamepad Tester, funciona apenas para controles para jogos de computador.

2 Especificação do Hardware

A figura 4 mostra os periféricos utilizados neste projeto. Como fora proposto todos encontram-se embarcados na placa **Bitdoglab**.

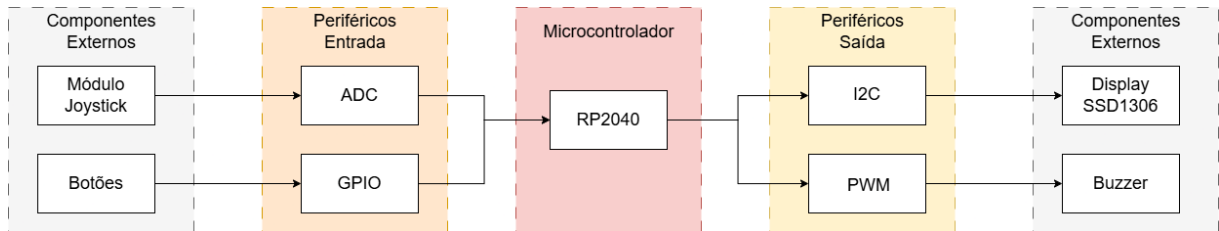


Figura 4: Diagrama em bloco dos hardware utilizado

2.1 Descrição de cada bloco e os pinos

- O módulo joystick foi utilizado para simular outro módulo para verificar a existência de defeitos. Foi utilizado os pinos 26 para o potenciômetro do eixo x e pino 27 para o eixo y. Além do 3.3 V e GND da placa.
- Os botões A e B foram utilizados para o usuário interagir com o programa ao seguir as informações exibidas no display. Foi utilizado os pinos 5, 6 e GND e o resistor "pullup" interno da placa.
- O ADC é ligado ao joystick pelos pinos 26 e 27. É empregado ao converter a tensão analógica do joystick em sinais digitais a serem processados.
- Os botões são conectados pela gpio 5 e 6. É utilizado para identificar a alteração dos níveis alto e baixo ao apertar os botões A e B.
- O microcontrolador utilizado durante o curso presente na **Bitdoglab**.
- O periférico I2C é utilizado para o microcontrolador conversar com o display. Os pinos empregados foram 14 para o "serial data" e 15 para o "serial clock".
- O PWM foi empregado para controlar o buzzer. Foi utilizado no pino 21.
- O buzzer é utilizado para sinalizar que a ação do usuário foi executada corretamente. É utilizado ao configurar o PWM no pino 21.
- O **Display SSD1306** é utilizado para interagir com o usuário e auxilia na calibração do módulo joystick. Os pinos utilizados foram 14, 15, GND e 3.3 V da **Bitdoglab**.

A seguir a imagem representando todo o circuito do projeto desenhado no programa do Wokwi disponível online.

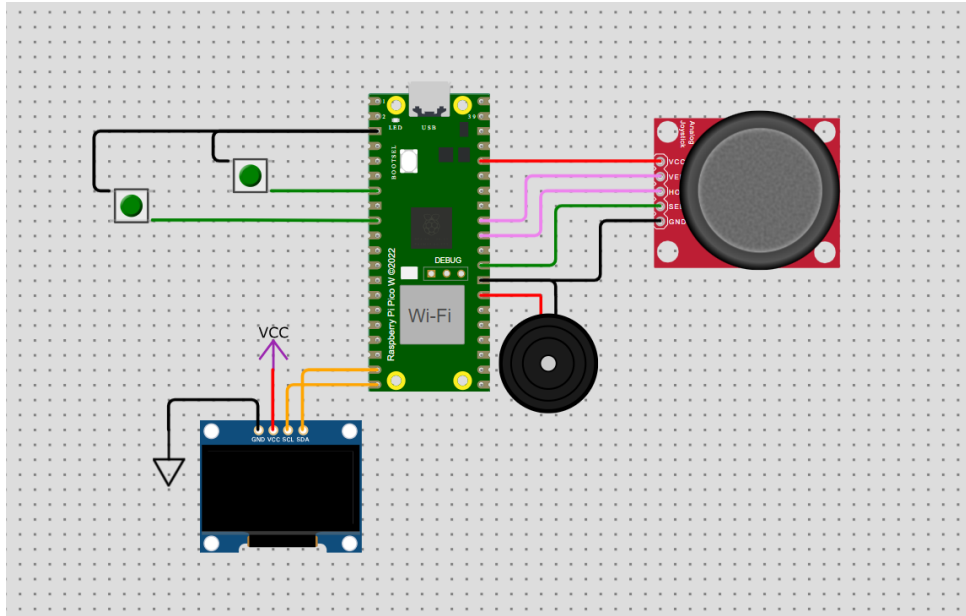


Figura 5: Circuito completo do projeto desenhado no Wokwi.

3 Especificação do Firmware

3.1 Funções

A seguir temos os blocos funcionais e as funcionalidades do projeto.

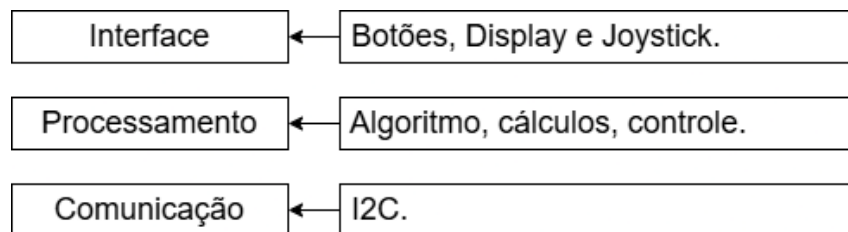


Figura 6: Blocos Funcionais do Projeto.

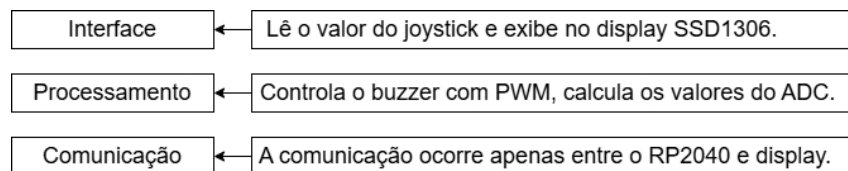


Figura 7: Descrição das funcionalidades do projeto

3.2 Fluxograma

Aqui será apresentado o fluxograma do trabalho e posteriormente descrição de suas partes.

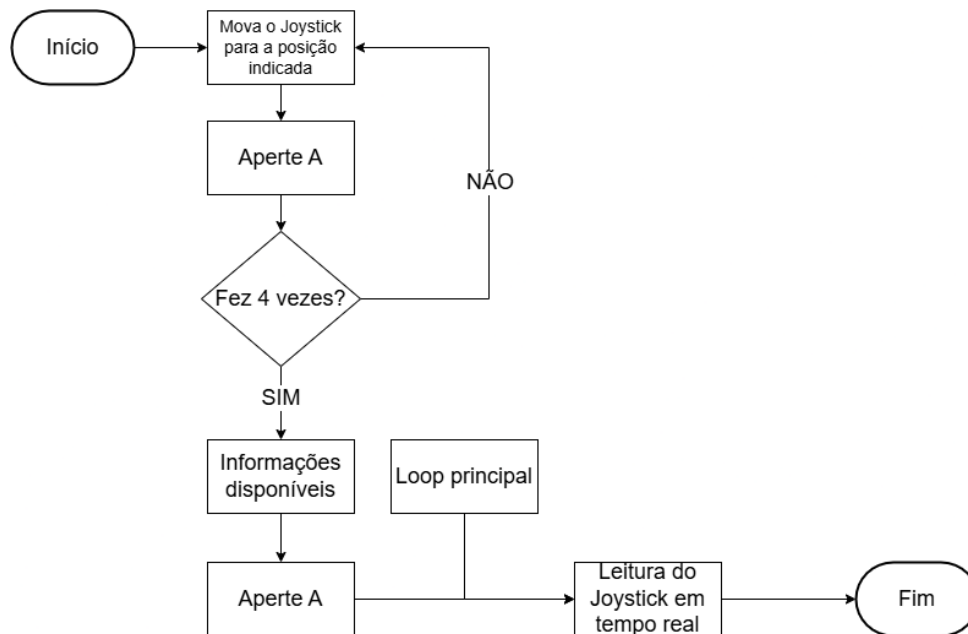


Figura 8: Caption

O trabalho inicia pedindo ao usuário calibrar o joystick indicando no display sua posição e para capturar o dado aperte o botão A. Após calibrar as 4 direções indicadas as informações destas posições são mostradas, também, no display. Apertando A novamente entra no loop principal nele as informações da posição do joystick são convertidas e representada no display.

3.3 Descrição do Código

A seguir será mostrado a função de configuração dos botões A, B e do Joystick. Utiliza a função para abilitar o resistor de 'pullup' da GPIO correspondente.

Código 1: Função de configuração dos Botões

```

1 void initButton(uint botao) {
2     gpio_init(botao);
3     gpio_set_dir(botao, GPIO_IN);
4     gpio_pull_up(botao);}
  
```

A próxima função inicia a transmissão I2C entre o display e o RP2040. Utiliza o endereço do display e os pinos de 'serial clock' e 'serial data'.

Código 2: Função do display SSD1306 com I2C.

```

1 void ssd1306_conf_init(ssd1306_t *ssd){
2     i2c_init(PORTA_I2C, 400 * 1000);
3     gpio_set_function(SDA, GPIO_FUNC_I2C);
4     gpio_set_function(SCL, GPIO_FUNC_I2C);
  
```



```

5  gpio_pull_up(SDA);
6  gpio_pull_up(SCL);
7  ssd1306_init(ssd, 128, 64, false, ENDERECO_DISPLAY, PORTA_I2C);
8  ssd1306_config(ssd);
9  ssd1306_send_data(ssd);
10 ssd1306_fill(ssd, false);
11 ssd1306_send_data(ssd);}

```

A seguir também temos a configuração do PWM padrão para o controle do Buzzer. Este último é empregado como sinalizador de que o programa está funcionando como esperado.

Código 3: Configuração do PWM para o controle do Buzzer.

```

1 void pwm_init_buzzer(uint pin){
2     // Configuração do GPIO para o buzzer como saída
3     gpio_init(BUZZER_PIN);
4     gpio_set_dir(BUZZER_PIN, GPIO_OUT);
5     // Configurar o pino como saída de PWM
6     gpio_set_function(pin, GPIO_FUNC_PWM);
7     // Obter o slice do PWM associado ao pino
8     uint slice_num = pwm_gpio_to_slice_num(pin);
9     // Configurar o PWM com frequência desejada
10    pwm_config config = pwm_get_default_config();
11    pwm_config_set_clkdiv(&config,
12        clock_get_hz(clk_sys) / (BUZZER_FREQUENCY * 4096)); // Divisor de clock
13    pwm_init(slice_num, &config, true);
14    // Iniciar o PWM no nível baixo
15    pwm_set_gpio_level(pin, 0);}

```

O ADC é configurado iniciando e direcionando os pinos 26 e 27 correspondente ao eixo x e y.

Código 4: Configuração do conversor ADC.

```

1  adc_init();
2  adc_gpio_init(EIXO_HORIZONTAL);
3  adc_gpio_init(EIXO_VERTICAL);

```

No início do loop temos a variável cont (uma flag) dentro da comparação com o valor inicial de zero. Ao entrar nesta comparação temos a função 'telaEsquerda' que sera discutida em breve. e a seleção do adc correspondente ao eixo x do joystick.

Código 5: Início do loop.

```

1  if (cont == 0){

```

```

2         telaEsquerda(&tela);
3         adc_select_input(1);}

```

A função abaixo orienta, com uma mensagem no display, o usuário em qual posição ele deve deslocar o joystick. Nesta função escreve no display a seguinte frase: Mova para esquerda. Ao mover para posição indicada o usuário deve também apertar o botão A, que será explicado a seguir.

Código 6: Função das 4 primeiras telas.

```

1 void telaEsquerda(ssd1306_t *ssd)
2 { ssd1306_fill(ssd, false);
3   ssd1306_rect(ssd, 3, 3, 122, 58, true, false);
4   ssd1306_draw_string(ssd, "Mova para", 25, 18);
5   ssd1306_draw_string(ssd, "a Esquerda", 25, 35);
6   ssd1306_send_data(ssd); }

```

Ao apertar o botão A inicia-se a função de interrupção e dentro dela, temos: interrupção para o botão B, para o do joystick e para o botão A todos com debouncing via software. Ao apertar o botão A o ADC captura o valor da tensão naquele momento do joystick e incrementa em um o valor da flag (cont). Se apertar o botão B apenas decrementa a flag, sua função consiste em refazer a leitura do ADC caso o usuário cometa algum erro. E por fim, o botão do joystick serve apenas para testa-lo muda o preenchimento do retângulo na ultima tela da programa.

Código 7: Função de interrupção completa.

```

1 void handler_interrupcao(uint gpio, uint32_t eventos){
2     uint32_t tempo_atual = to_us_since_boot(get_absolute_time());
3     if (gpio == botao_B && tempo_atual - tempo_ultimo_clique > 200000){
4         tempo_ultimo_clique = tempo_atual;
5         if (cont > 0){
6             cont--;}}
7     else if (gpio == BOTAO_JOYSTICK){
8         if (tempo_atual - tempo_ultimo_clique > 200000)
9         { // debounce de 200ms
10            tempo_ultimo_clique = tempo_atual;
11            quadrado = !quadrado;}}
12     else if (gpio == botao_A){
13         if (tempo_atual - tempo_ultimo_clique > 200000)
14         { // debounce de 200ms
15            tempo_ultimo_clique = tempo_atual;
16            limitsADC[cont] = adc_read();
17            if (cont < 5){
18                cont++;}

```

```
19         buzzer_active = true;}}}
```

Depois de obter o valor das quatro posições do joystick (a flag cont está com valor 4) temos a função que mostra os valores obtidos no display.

Código 8: Função que obtém a posição do joystick e converte para uma animação no display.

```
1         else if (cont == 4){
2             sprintf(buffer_xmax, "%d", limitsADC[1]);
3             sprintf(buffer_ymax, "%d", limitsADC[2]);
4             sprintf(buffer_xmin, "%d", limitsADC[0]);
5             sprintf(buffer_ymin, "%d", limitsADC[3]);
6             valores(&tela, buffer_xmax, buffer_ymax, buffer_xmin, buffer_ymin);
```

E por último temos a função que obtém a posição do joystick em tempo real. Ela obtém o valor de tensão deste componente convertido pelo ADC e calcula a posição x e y para posicionar um retângulo que será desenhado no display.

Código 9: Função que obtém a posição do joystick e converte para uma animação no display.

```
1         else {
2             adc_select_input(1);
3             valor_adc_x = adc_read();
4             adc_select_input(0);
5             valor_adc_y = adc_read();
6             posX = PosiX(valor_adc_x, limitsADC[1], limitsADC[0]);
7             posY = PosiY(valor_adc_y, limitsADC[2], limitsADC[3]);
8             ssd1306_fill(&tela, !inverter_cor);
9             ssd1306_rect(&tela, 10, 39, 45, 45,
10             inverter_cor, !inverter_cor);
11             ssd1306_hline(&tela, 31, 95, 31, true);
12             ssd1306_vline(&tela, 62, 5, 60, true);
13             ssd1306_rect(&tela, posY, posX, 5, 5,
14             inverter_cor, quadrado);
15             ssd1306_send_data(&tela);}
```

4 Execução do Projeto

4.1 Metodologia

Neste projeto, inicialmente, foi elaborada a seção "Escopo do Projeto", com o objetivo de definir e delimitar o trabalho. Em seguida, foi realizada uma pesquisa sobre temas

relacionados e equipamentos comerciais semelhantes.

Foi proposto o uso da placa **Bitdoglab** para simular um protótipo funcional do projeto. As funcionalidades foram definidas com base nas seções anteriores. Além disso, foi sugerido o uso da IDE **VS Code** com a extensão **Pico SDK**, amplamente utilizada durante o curso. A documentação do curso foi consultada continuamente para garantir a programação correta dos periféricos, uma vez que os exemplos fornecidos foram de grande auxílio. Por fim, quando ocorreram erros no programa, estes foram corrigidos com o auxílio do terminal da IDE.

4.2 Testes e Validação

Nesta parte foi realizado a validação em relação aos objetivos definidos do projeto. Foi concluído de forma crescente um de cada vez depois de programado foi testado na **Bitdoglab**.

4.3 Discursão dos Resultados

O programa conseguiu atingir os objetivos especificados na sua seção. Utilizou o ADC para capturar o valor do módulo do joystick. O display, em conjunto com os botões, interage com o usuário. O buzzer é utilizado quando o botão A da placa é pressionado.

Ao longo do trabalho, identificou-se a necessidade de criar um perfil para cada tipo de joystick, pois eles apresentam valores de potenciômetro que, conforme a aplicação, podem variar significativamente. Alguns joystick não deslocam sua resistência para próximo de zero, mantendo sempre um determinado nível de tensão.

Ao realizar este projeto, fica clara a diferença entre os conceitos teóricos e a realidade, pois, ao utilizar componentes reais, surgem situações que não ocorreriam em teoria. Como, por exemplo, erro no conversor ADC ou o joystick não utilizando todos os níveis de tensão da placa. Além disso, observou-se limitações nos dispositivos. Por exemplo, o display possui 64 pixels na vertical, e o ADC tem até 4096 níveis de tensão, o que impede a representação de todos esses níveis ao mover o joystick no display. Por outro lado, é complicado exibir muitas informações no display devido à sua resolução.

Por fim, o projeto apresenta alguns bugs e erros, que são descritos a seguir. Caso o joystick não seja calibrado corretamente, pode ocorrer de o programa travar, pois não há limitação quanto ao movimento do bloco no display. Além disso, ao apertar o botão de reset em um momento específico, ocorre um erro que congela o display, sendo necessário desligar a placa para corrigir o problema.

5 Referências

COMPONENTS 101. Joystick Module, 2018. Disponível em: <https://components101.com/module/joystick-module>. Acesso em: 22 fev. 2025.

ELETRONICS, Soundwell. RJ13 analog joystick potentiometer, Disponível em: <https://www.soundwell.com/products/rj13-analog-joystick-potentiometer.html>. Acesso em: 22 fev. 2025.

Hardware Tester. Gamepad Tester. Disponível em: <https://hardwaretester.com/gamepad>. Acesso em: 20 fev. 2025.

Para Ps5 Switch Xbox 13mm Hall Joystick Test Effect. Disponível em: <https://produto.mercadolivre.com.br/MLB-3624374319-para-ps5-switch-xbox-13mm-hall-joystick-test-effect-J-M?matttool=18956390>