

Sistema de irrigação automática para jardim

THIAGO CORREA E SILVA DE MORAES

MATHEUS ADARIO DI PRADO MILANI

KASSEM MAHMOUD HAMDOUN



Objetivo do Projeto

Desenvolver um **sistema automatizado de irrigação** voltado para plantas decorativas em ambientes residenciais, com foco em **eficiência hídrica, autonomia e sustentabilidade**.

Para isso, o projeto integra:

- **Sensor de umidade do solo** para monitoramento contínuo das condições do solo.
- **Circuito eletrônico de condicionamento de sinal** para garantir leituras estáveis e precisas.
- **Microcontrolador RP2040** para processamento das informações e tomada de decisão.
- **Atuadores mecânicos** (motor, engrenagens e válvula) para controle físico da irrigação.

Processo de desenvolvimento

1. Definição do Problema

- Necessidade de irrigar plantas automaticamente com base na umidade do solo;
- Aplicação escolhida: **Sustentabilidade e automação residencial.**

2. Escolha dos Componentes

- Sensor de umidade analógico;
- Microcontrolador Pi Pico **RP2040**;
- Servo motor MG996R para simular irrigação;
- LED vermelho para sinalização do reservatório;
- Transistor, resistores, capacitores e botão.

3. Montagem do Circuito

Processo de desenvolvimento

4. Programação do Microcontrolador

5. Testes e Ajustes

- Calibração da resistência do solo
- Testes de funcionamento em solo seco e úmido
- Verificação da estabilidade do sistema e resposta do servo
- Calibração do volume de água em relação ao tempo de acionamento

6. Montagem Final e Validação

Fundamentação Teórica - Sensor de Umidade

O sensor de umidade resistivo funciona com base nos princípios da **condutividade elétrica** em meios porosos como o solo. A umidade presente entre duas sondas metálicas atua como um condutor, permitindo a passagem de corrente elétrica.

- Quanto maior a **umidade**, menor a resistência elétrica entre os eletrodos.
- Quando o solo está seco, a condutividade é reduzida, aumentando a resistência.

Esse comportamento segue a **Lei de Ohm**, permitindo que a resistência variável seja convertida em **tensão elétrica**, que pode ser lida por um **conversor analógico-digital (ADC)**.

Para garantir a qualidade do sinal:

- Um filtro RC é utilizado para remover ruídos e flutuações.
- O sinal analógico é interpretado por um microcontrolador (RP2040), que toma decisões com base em um limiar de umidade pré-definido

Fundamentação - Atuadores do Sistema

O sistema de irrigação conta com atuadores responsáveis por realizar a abertura da passagem de água no momento adequado:

- **Válvula solenoide servo atuada:** Controla o volume de água utilizado para irrigação possuindo consumo energético constante independente do volume necessário.

- Servo Motor MG996R
- Registro de esfera de 3/8 pol
- Par de engrenagens com uma relação de 2/1



Fundamentação - Atuadores do Sistema

- **Botão de Acionamento Manual:** permite testes e acionamentos manuais, além de ser usado como recurso de override.

- **Placa PCB e Estrutura Base:** todos os componentes foram fixados em uma base integrada, otimizando o layout e facilitando a montagem do protótipo.

Esses atuadores trabalham de forma coordenada com o microcontrolador, garantindo o controle automático e eficiente da irrigação

Componentes do Circuito

- Pi Pico RP2040;
- Transistor BC547;
- 1 LED;
- 1 push button;
- 3 barramentos com 80 pinos cada;

Resistores:

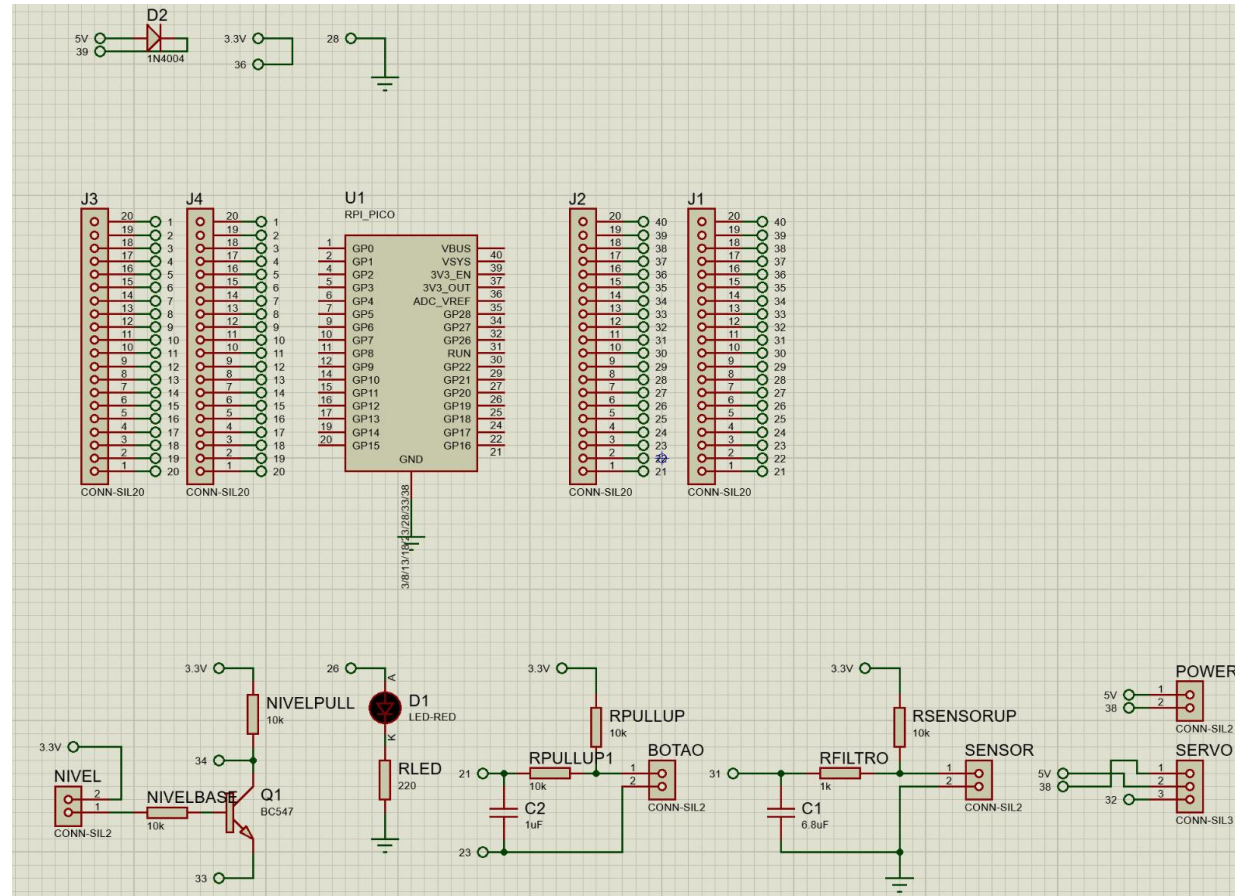
- 5 x 10 k Ω ;
- 1 x 1 k Ω ;
- 1 x 220 Ω .

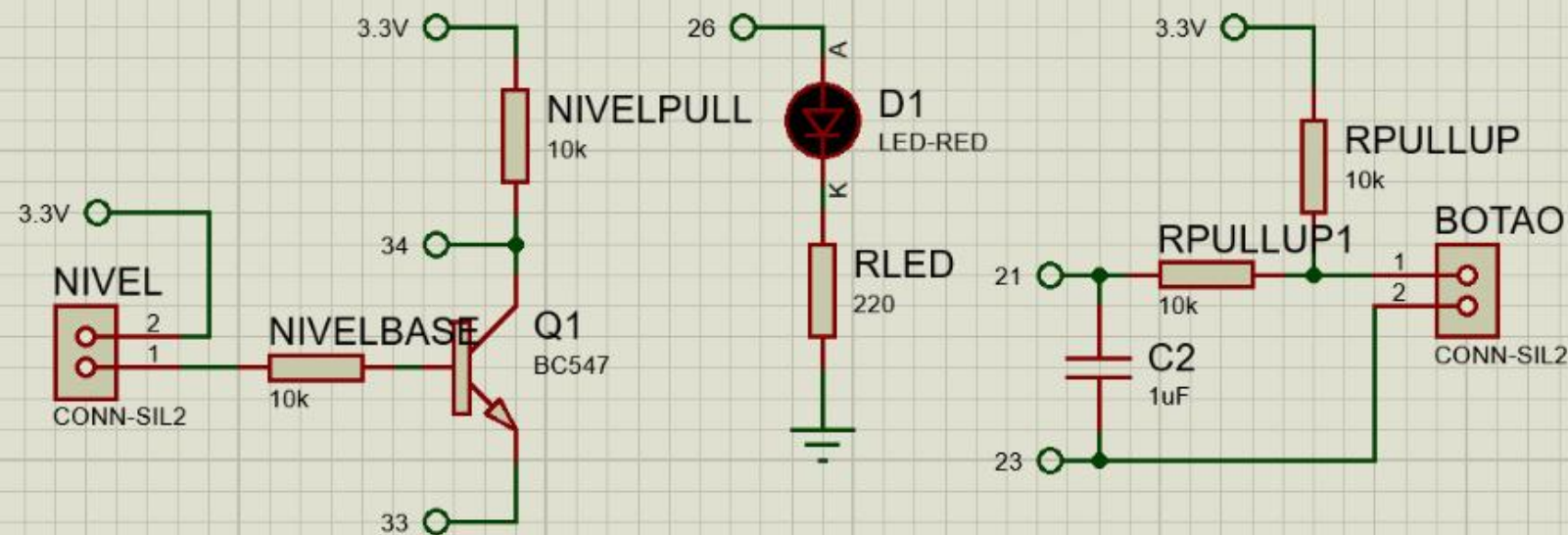
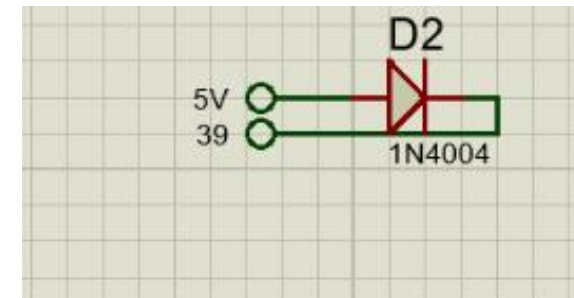
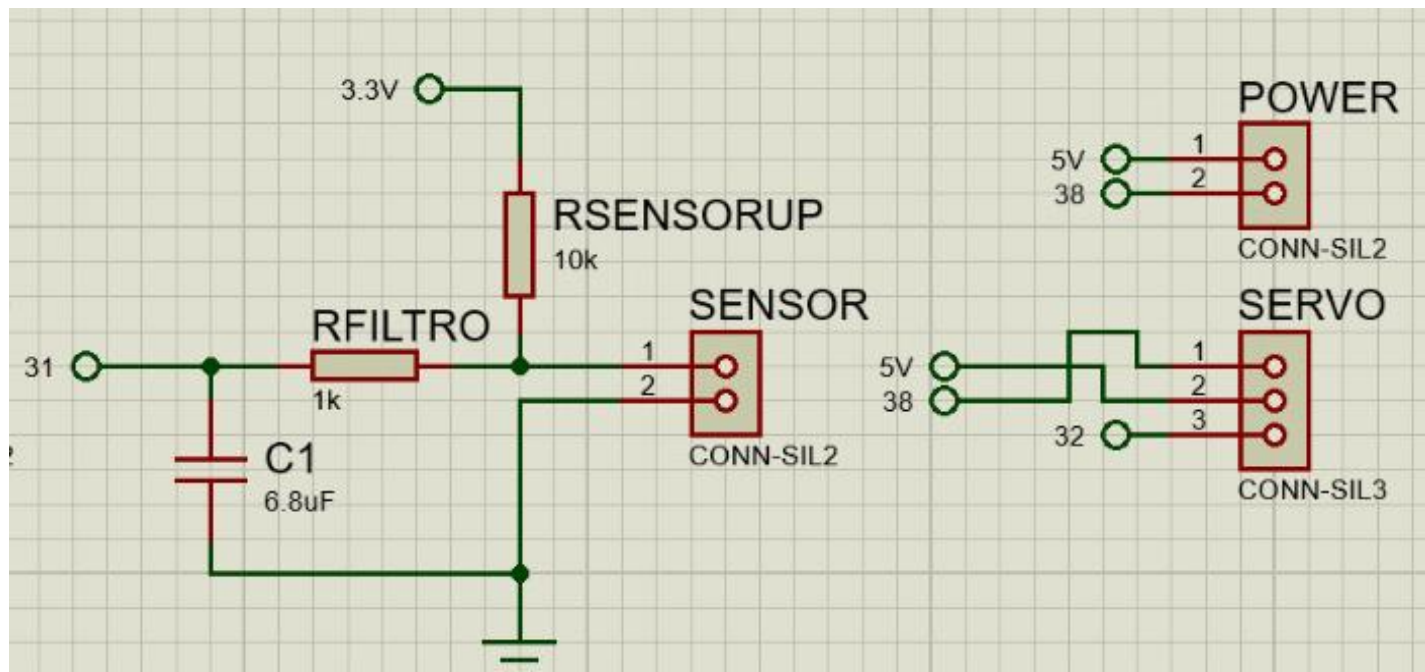
Capacitores:

- 1 x 1 μ F;
- 1 x 4,7 μ F.

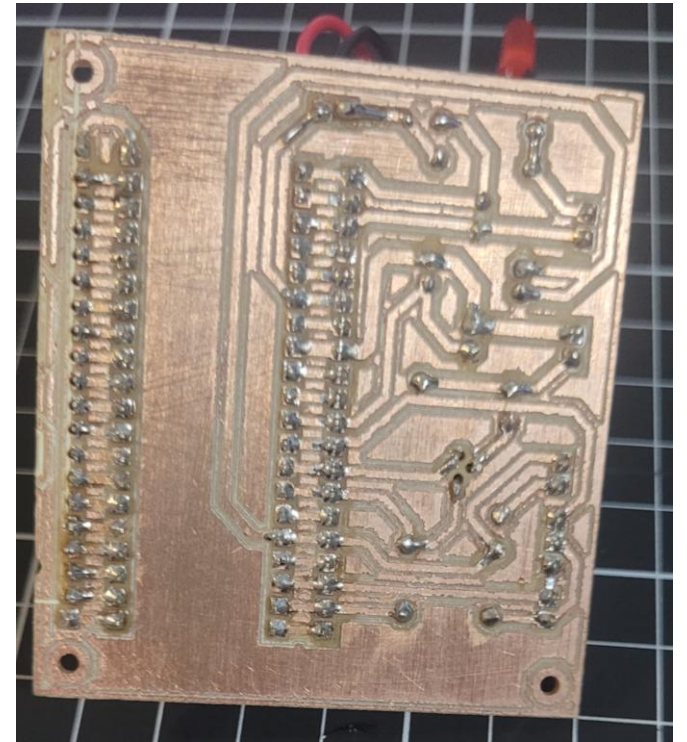
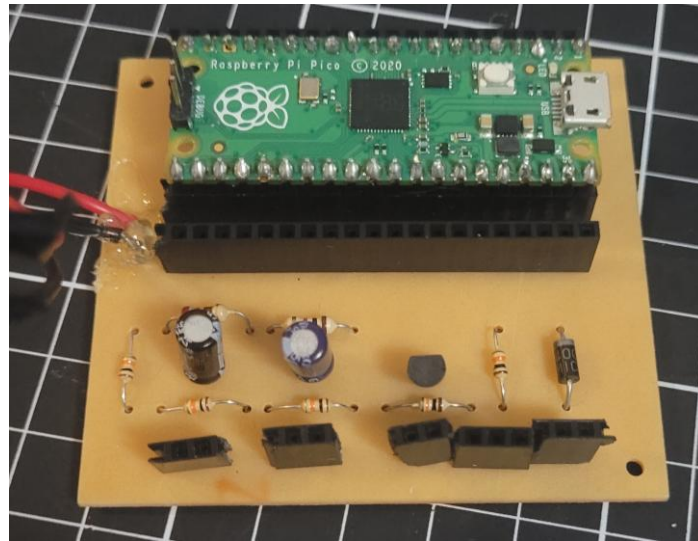
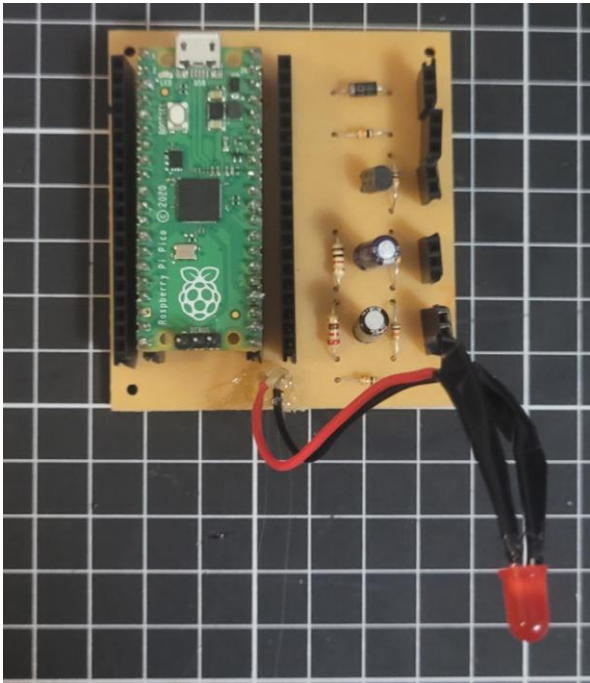


Esquema Eléctrico





PCB



Programação do microcontrolador

Interrupções:

```
gpio_set_irq_enabled_with_callback(BUTTON_PIN,GPIO_IRQ_EDGE_FALL,1,&gpioHandler);

gpio_set_irq_enabled_with_callback(NIVEL_PIN,GPIO_IRQ_EDGE_FALL,1,&gpioHandler);
//gpio_set_irq_enabled_with_callback(NIVEL_PIN,GPIO_IRQ_EDGE_RISE,1,&gpioHandler);

repeating_timer_t timer;
add_repeating_timer_ms(5000, timer_cb, NULL, &timer);
```

Programação do microcontrolador

ADC:

```
bool timer_cb(repeating_timer_t *t) {
    if(gpio_get(NIVEL_PIN)==0){
        gpio_put(LED_PIN,0);
    }else{
        gpio_put(LED_PIN,1);
    }
    //adc_run(true);
    uint16_t raw = adc_read();
    float v = raw * 3.3f / 4095;
    printf("%u,%.3f\n", raw, v); // Inicia 1 conversão single-shot
    if(raw>LIMITE_UMIDADE){
        regar();
    }
    return true; // Repetir indefinidamente
}
```


Programação do microcontrolador

```
void gpioHandler(uint gpio,uint32_t events){
    if(gpio==BUTTON_PIN){

        regar();
    }else if(gpio==NIVEL_PIN){
        if(events == GPIO_IRQ_EDGE_FALL){
            gpio_put(LED_PIN,0);
        }else if(events == GPIO_IRQ_EDGE_RISE){
            gpio_put(LED_PIN,1);
        }
    }
}
```

Programação do microcontrolador

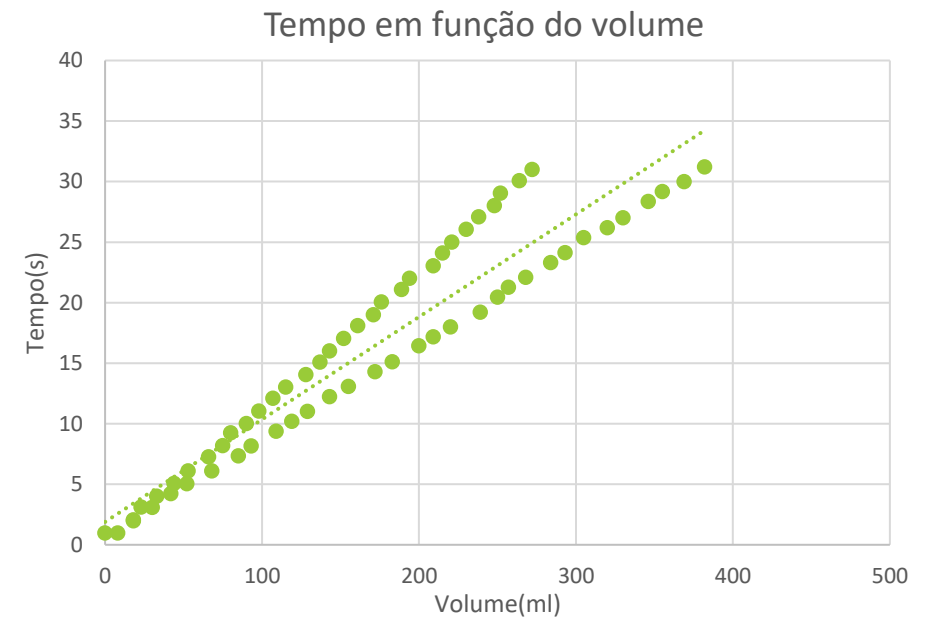
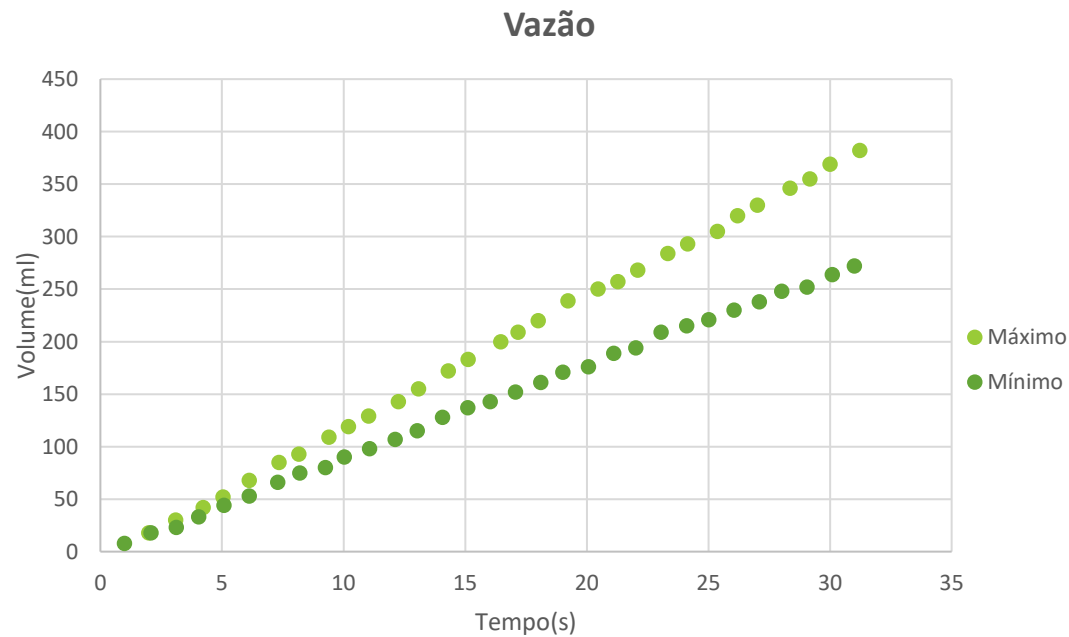
```
#define VOLUME 50
volatile uint32_t DELAYREGAR=VOLUME*84700+1887500;
```

```
void regar(){
    printf("Tentando Regar\n");
    if((gpio_get(NIVEL_PIN)==0) // (DESATIVARNIVEL)){
        uint64_t tempoPassado=time_us_64()-ultimaRega;
        if(tempoPassado>TEMPOULTIMAREGA){
            goServo(SERVO,ON_ANGLE);
            busy_wait_us(DELAYREGAR);
            goServo(SERVO,OFF_ANGLE);
            ultimaRega=time_us_64();
        }
    }
    if(gpio_get(NIVEL_PIN)==0){
        gpio_put(LED_PIN,0);
    }else{
        gpio_put(LED_PIN,1);
    }
}
```

Programação do microcontrolador

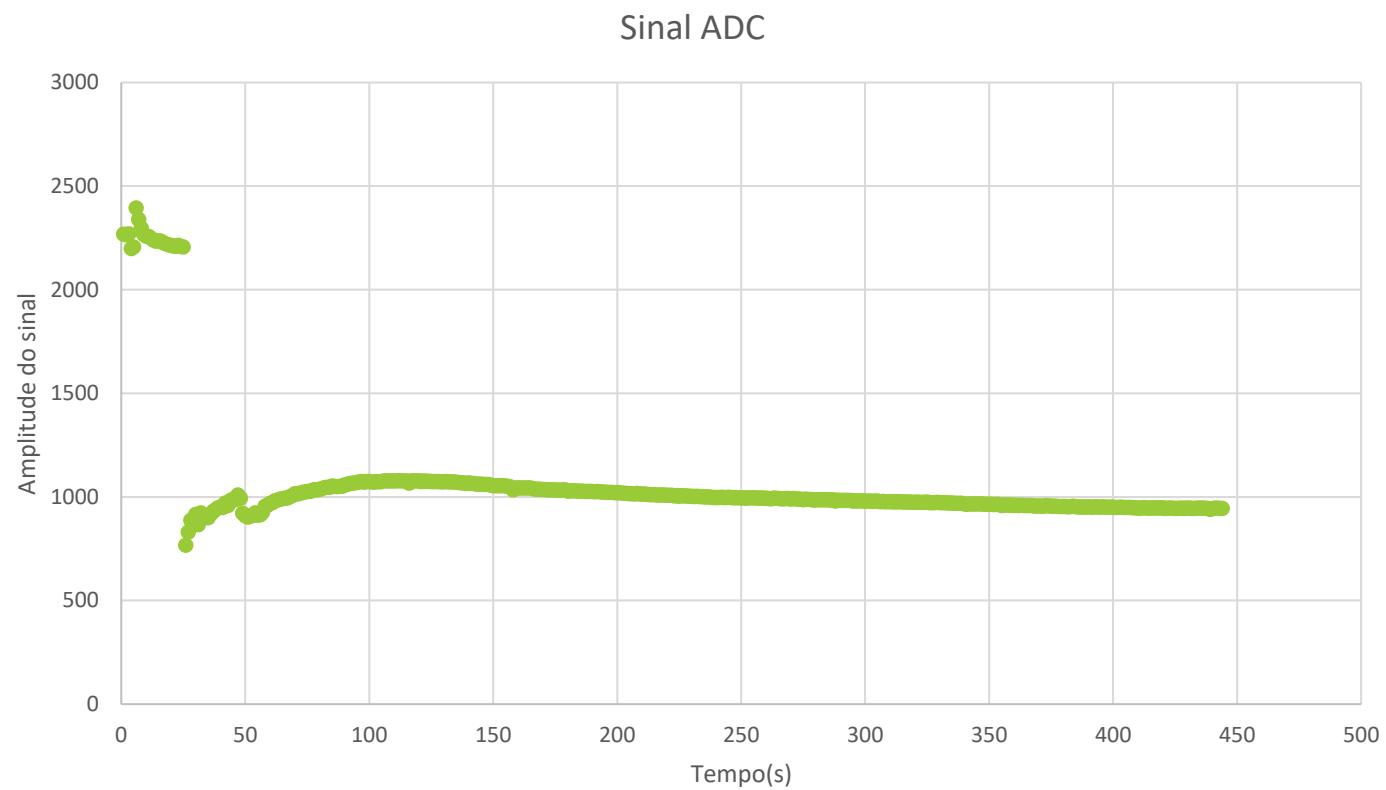
```
void goServo(int servo, int ang){  
    pwm_set_enabled(pwm_gpio_to_slice_num(SERVO), true);  
    int duty = (((float)(SERVO_180 - SERVO_0) / 180.0) * ang) + SERVO_0;  
    pwm_set_gpio_level(servo, duty);  
    busy_wait_us(DELAY_MOV);  
    pwm_set_enabled(pwm_gpio_to_slice_num(SERVO), false);  
  
}
```


Estudos



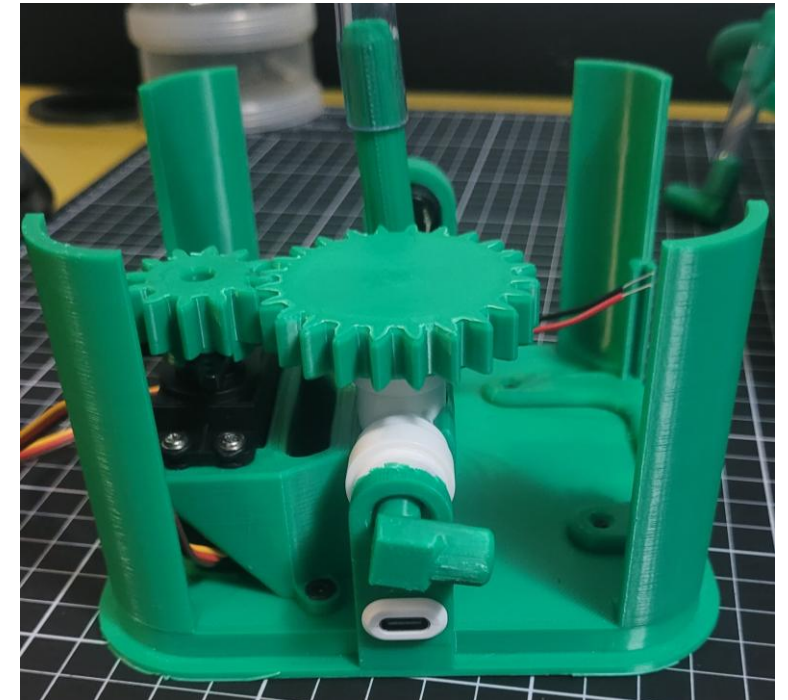
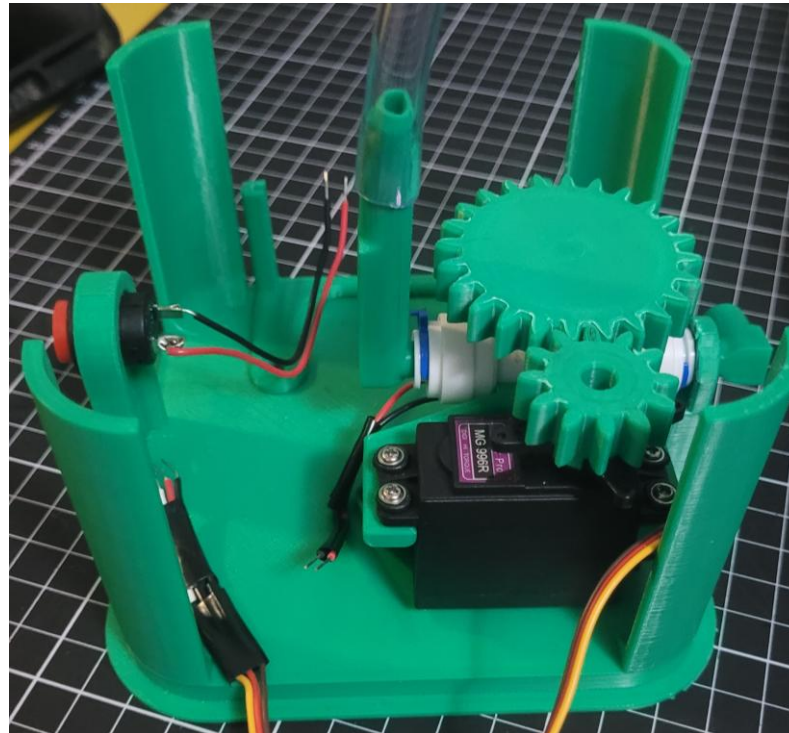
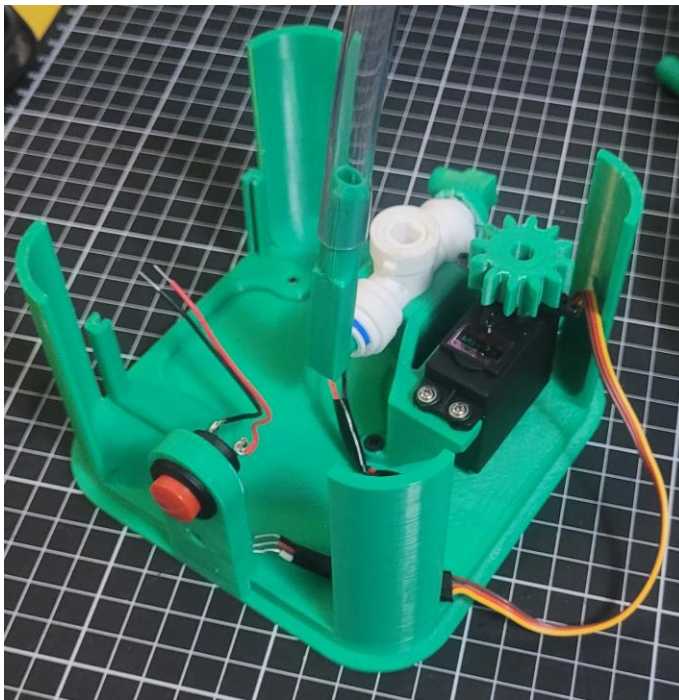
$$\text{Tempo(s)} = 0,0847 * \text{Volume} + 1,8875$$

Estudos



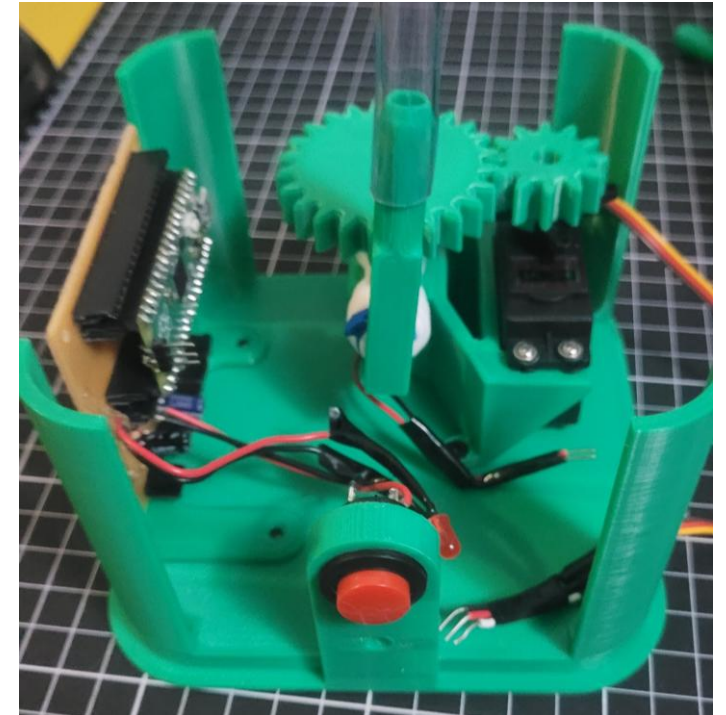
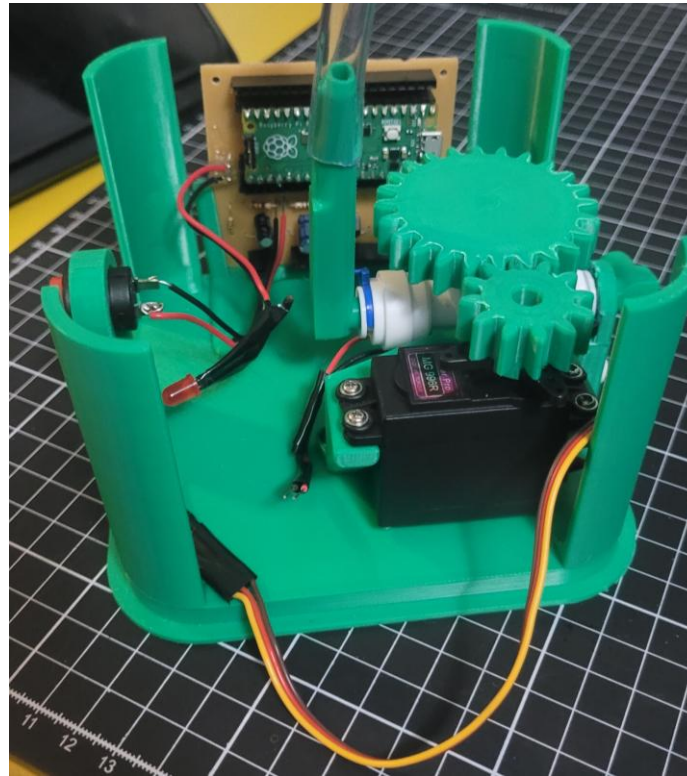
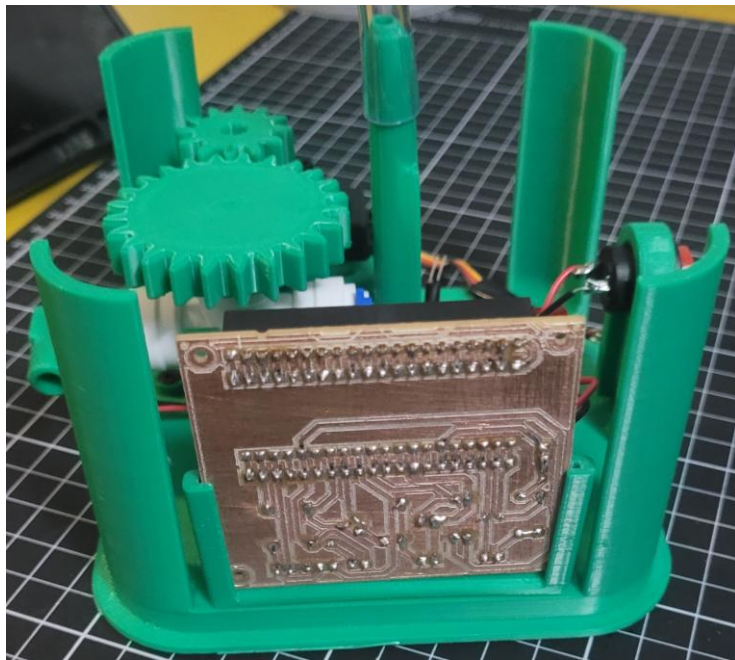
Atuadores

Motor, par de engrenagens e válvula montados na base do reservatório junto com o botão.



Atuadores e PCB

Base do reservatório com o botão, atuadores e PCB.



Reservatório

Hastes do reservatório conectadas à sua tampa inferior;

Reservatório com tampa superior;

Reservatório montado.



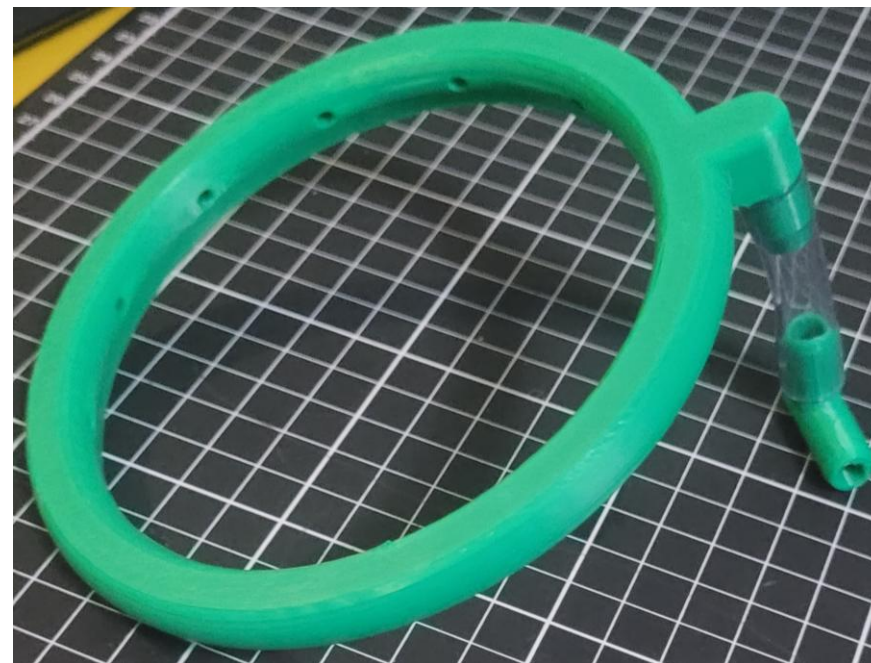
Reservatório montado na base

- Reservatório montado na base fechada.



Distribuição de água do vaso

Peça que recebe e distribui água ao redor do vaso.



Vaso



Base do vaso



Vaso montado

- Sem a peça de irrigação;
 - Com a peça de irrigação.
-





Conclusão
