

1 Monitoramento de vagas

1.1 Especificação Estrutural

Nessa seção do documento iremos especificar a estrutura do projeto *Park Monitoring* identificando os componentes a serem utilizados, sua função e sua forma de interação com os demais componentes. O sistema projetado será composto por 2 diferente módulos, um módulo para o conjunto sensor+led e um módulo central para o microcontrolador e o display.

1.1.1 Módulo central

- Microcontrolador ESP32: O Microcontrolador é responsável pela orquestração dos componentes do projeto.
 - Leitura dos sensores infravermelho.
 - Implementação da lógica do programa, fazendo com que as medições sejam interpretadas e ações executadas utilizando as diversas interfaces
 - Envio dos registros de eventos via interface Wi-fi integrada ao microcontrolador via protocolo http.
 - Uso da porta serial I2C para envio das informações para LCD e acender os LEDs RGB.

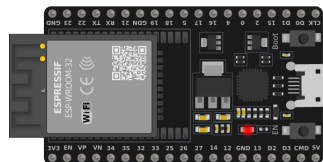


Figura 1: Microcontrolador ESP32

- Display LCD para simular um painel que indica a quantidade de vagas disponível naquele estacionamento. O display tem o controle de vagas totais vs vagas disponíveis no momento. Esse display é atualizado a cada alteração no estado de um led.
 - Características
 - * Comunicação serial via I2C
 - * Display 16x2



Figura 2: Display LCD com módulo de expansão I2C

- Para utilizar a conexão com os diversos sensores precisamos de um módulo I2C IC PCF8574 que fornece expansão de E/S remota de uso geral por meio do relógio serial bidirecional I2C de dois fios (SCL) e dados seriais (SDA).
- Para utilizar o Display LCD precisamos de um módulo I2C IC 16x2 TCA9535DBR.

1.1.2 Módulo sensor

- LED RGB cátodo comum: Permite que com níveis de tensão especificados conseguimos utilizar diferentes cores, ele é formado por um vermelho (R de red), um verde (G de green) e um azul (B de blue - não será utilizado).
- O sensor escolhido *VL53L1X* tem como principal característica a medição de distância através do tempo de viagem de um laser em até 4m. Isso permite flexibilidade na implementação em diferentes ambientes, visto que, podemos calibrar a distância em qualquer ambiente e alterar a lógica adicionando um valor.
- Um mux *PCF8574TS*: Este mux é adequado para ser utilizado em um barramento I2C e cada CI possui um endereço definido via hardware permitindo a expansão do sistema para suportar vários módulos sensores.

1.2 Algoritmo de observação de vagas

- De maneira geral o microcontrolador irá fazer uma varredura em cada um dos dos sensores, em um período definido, verificando se houve uma alteração de estado;
- Ao identificar uma alteração de estado o MC irá guardar o tempo em que essa alteração foi detectada e irá manter uma flag para o endereço daquele sensor indicando uma mudança de estados;
- Caso não haja alteração no estado o valor de t_n será atualizado para garantir que o valor do led desse endereço não seja alterada na etapa de mudança de estado;
- Em um segundo momento o MC irá realizar uma segunda varredura nos endereços dos sensores verificando se o tempo passado desde a alteração de estado ultrapassa o valor definido de t_{cs} (tempo de threshold para mudança de estado de uma vaga)
- Caso o tempo desde a alteração de estado detectada seja maior que t_{cs} o MC irá alterar o valor do led e resetar a flag que indica mudança de estado
- Caso o tempo t_n seja menor que t_{cs} , ou seja, ou não se passou tempo o suficiente desde a última alteração de estado ou o valor medido no sensor seja igual o valor representado no led, não será necessário ocorrer nenhuma mudança

Este comportamento descrito está representado no algoritmo a seguir:

```

1 while true:
2     Fazendo uma varredura por todos os enderecos:
3         Se (valor_sensor != valor_led): % ou seja sensor esta medindo um estado diferente do
         que ta no led
4             se flag_n==0:
5                 t_n = timestamp.now()
6                 flag_n = 1
7             Se valor_sensor == valor_led: % note que caso seja identificada uma mudanca de estado
         que ja foi identificada anteriormente o valor de t_n nao sera alterado
8                 t_n = timestamp.now()
9                 flag_n = 0 % garante que flag de mudanca de estado esteja em zero
10
11 % Para fazer a alteracao de estado da lampada
12 Fazendo uma varredura por todos os enderecos:
13     se (timestamp.now() - t_n) >= tempo threshold:
14         estado_led = estado_sensor
15         flag = 0 % reseta flag
16 % se (timestamp.now() - t_n) < tempo threshold o estado da led e mantido

```

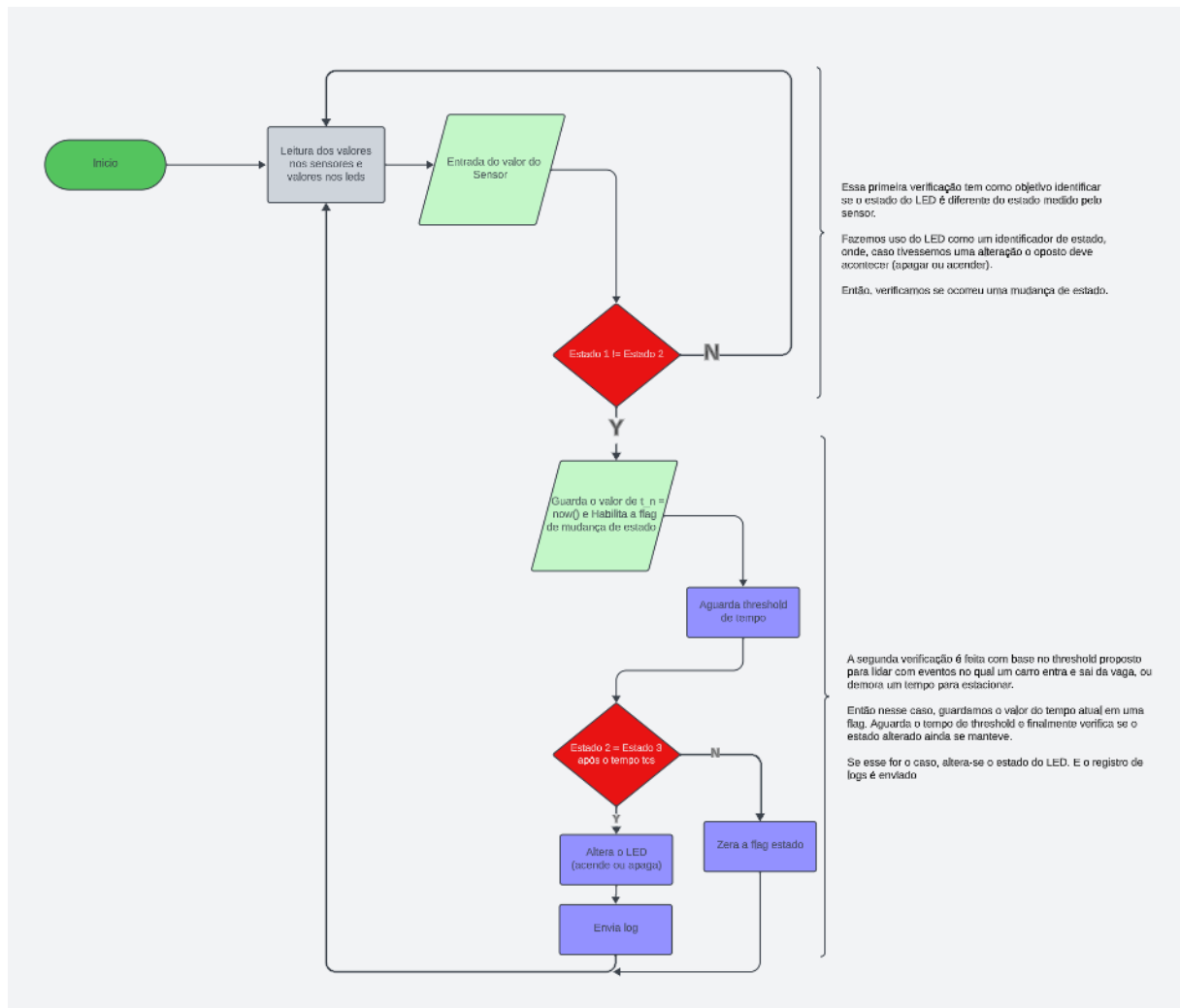


Figura 3: Fluxograma do algoritmo de observação de vagas

2 Envio de dados para análise

Para a realização da análise de fluxo de veículos do sistema, iremos utilizar um software externo, o Splunk Cloud Platform, este software oferece suporte justamente para a análise de dados e geração de relatórios e dashboard para análise de dados de maneira prática. Dessa forma, considerando que o microcontrolador ESP32 escolhido, possui um módulo wifi integrado, a partir de um tempo de coleta de dados t_{dc} definido, o microcontrolador verificaria a relação de vagas naquele instante e bastaria fazer o envio desses dados para o servidor utilizado do Splunk via protocolo http onde esses dados seriam posteriormente analisados via Splunk Processing Language. De maneira geral temos que o algoritmo utilizado para essa comunicação apresentado na seguir

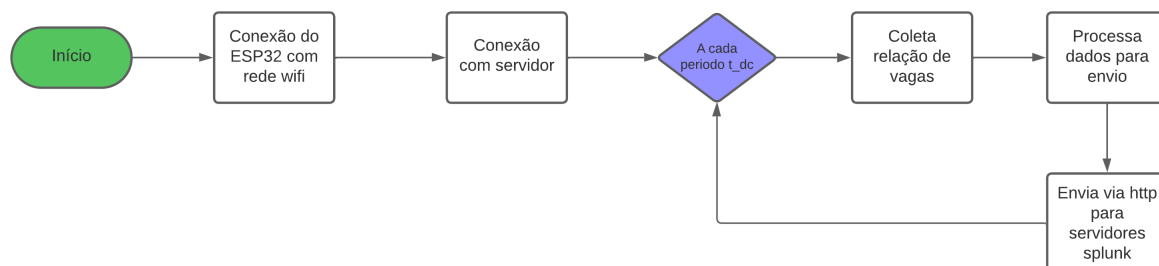


Figura 4: Diagrama de funcionamento do algoritmo de coleta e envio de dados.

3 Projeto

3.1 Módulo central

No módulo central temos o ESP32, microcontrolador responsável por executar o algoritmo mencionado acima e acionar os dispositivos responsáveis pelos indicadores do projeto: LCD e LED RGB (esse, incluído no módulo do sensor).

Foi utilizado um adaptador I2C para o LCD 16x2.

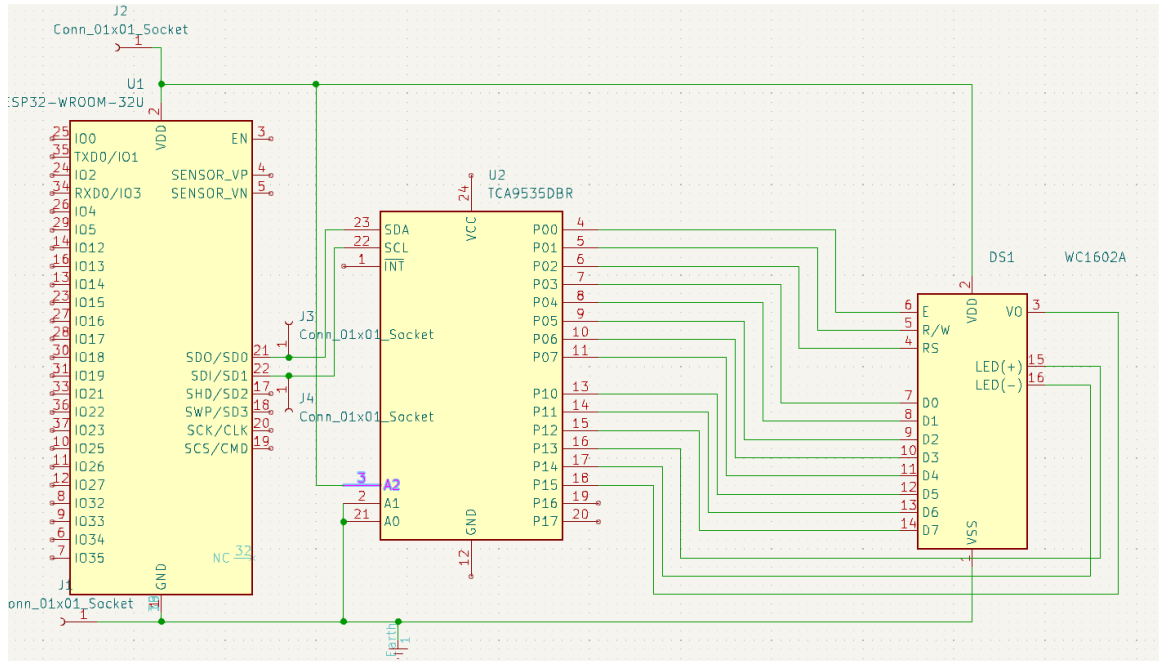


Figura 5: Esquemático do módulo central

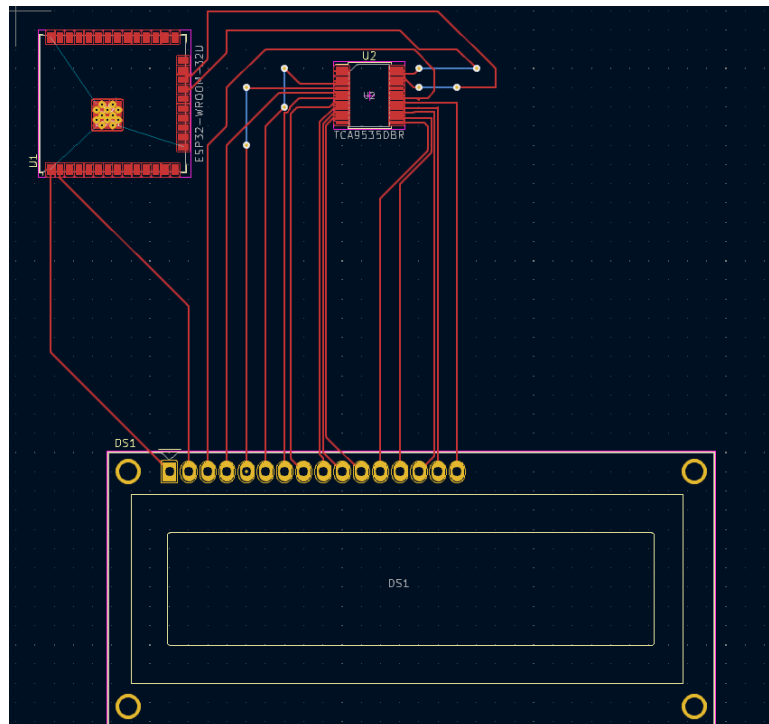


Figura 6: Layout da placa do módulo central

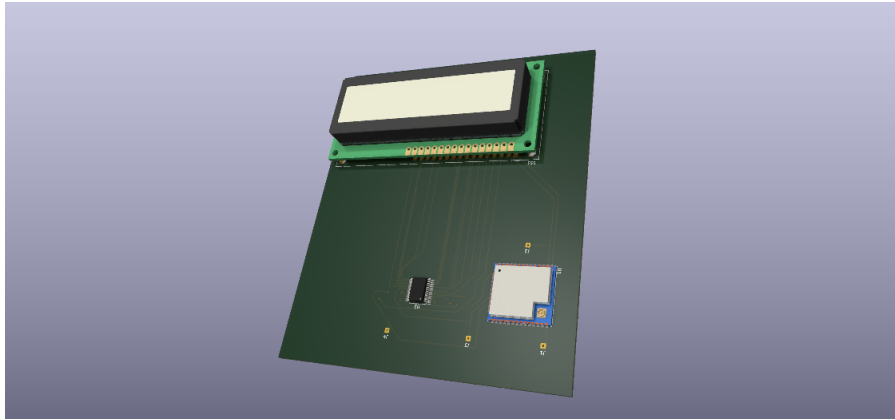


Figura 7: Renderização da placa do módulo central

3.2 Módulo sensor

Cada módulo de sensor seguirá o esquemático da figura 8.

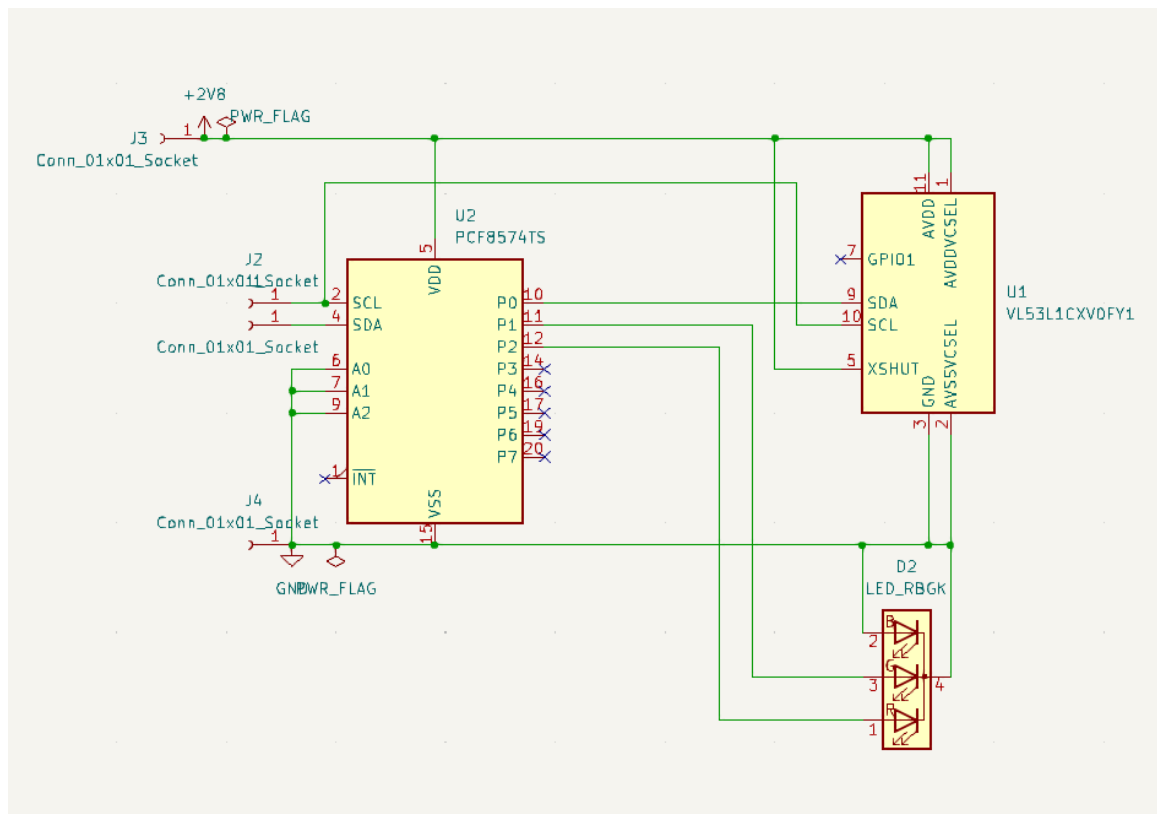


Figura 8: Esquemático do módulo sensor

A partir desse esquemático foi possível projetar o footprint apresentado nas figuras 9, 10 e 11. Podemos também observar a placa desenvolvida no modelo 3D apresentado nas figuras 12, 13 e 14

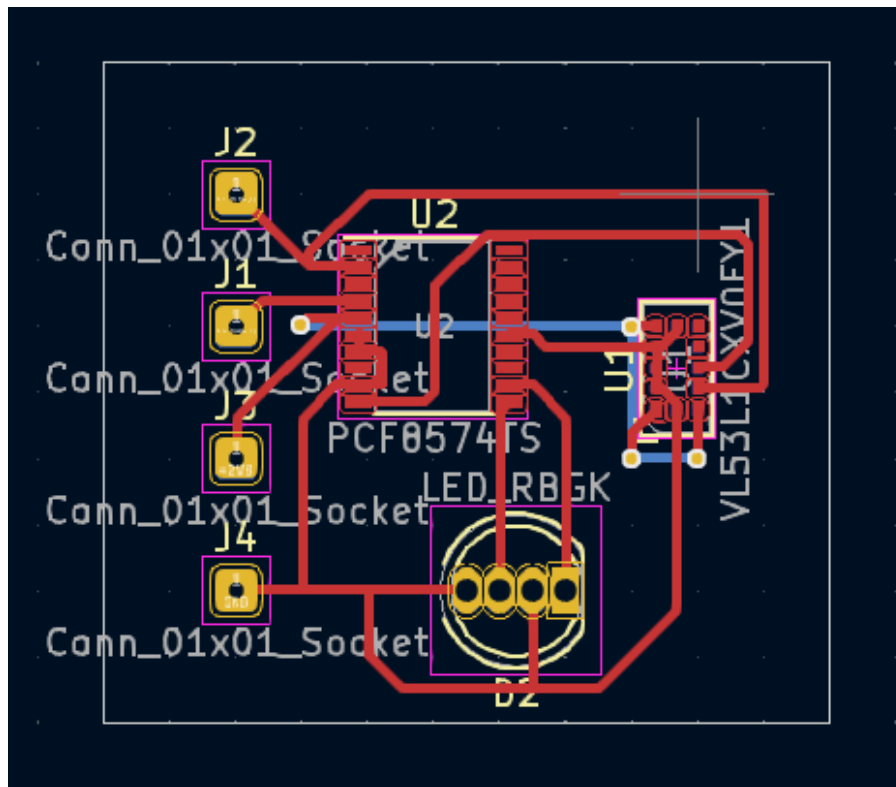


Figura 9: Layout final placa PCB do módulo sensor

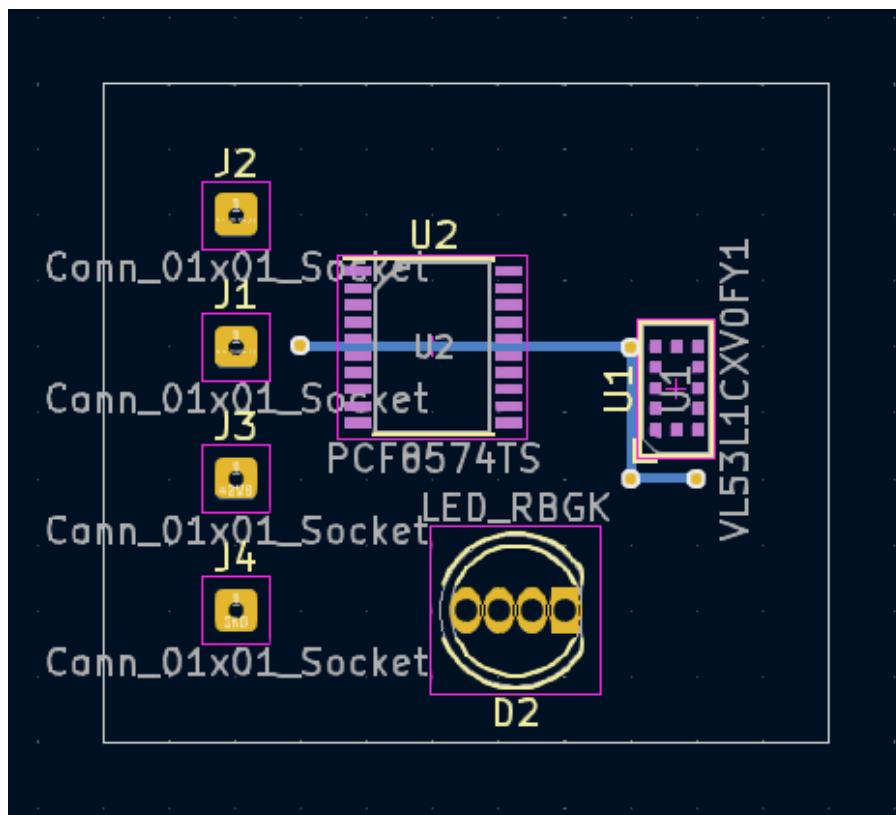


Figura 10: Layout da layer inferior placa PCB do módulo sensor

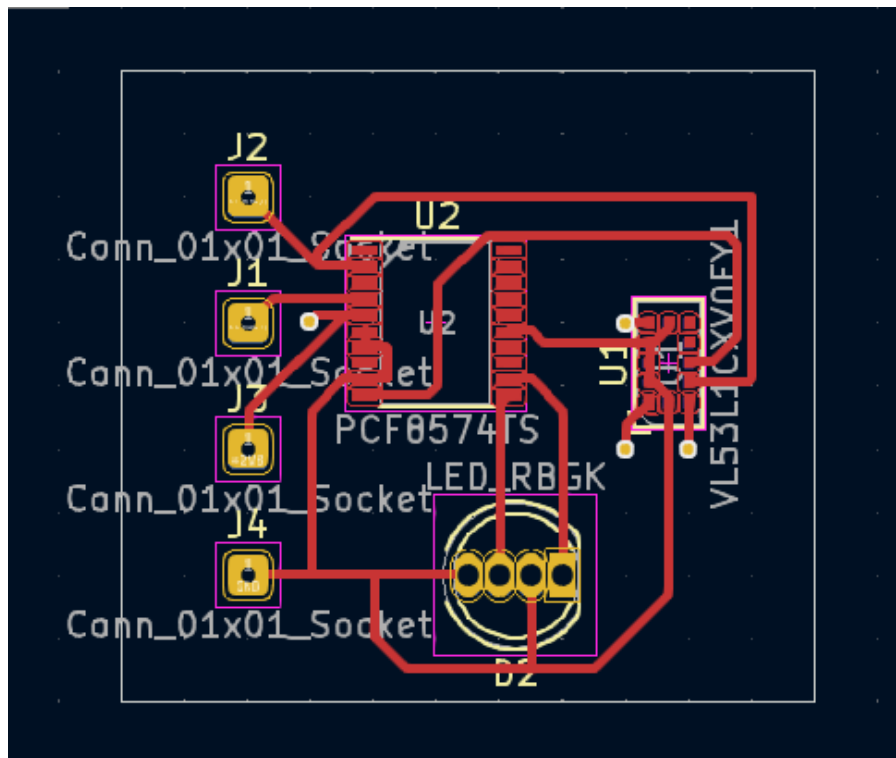


Figura 11: Layout da layer superior placa PCB do módulo sensor

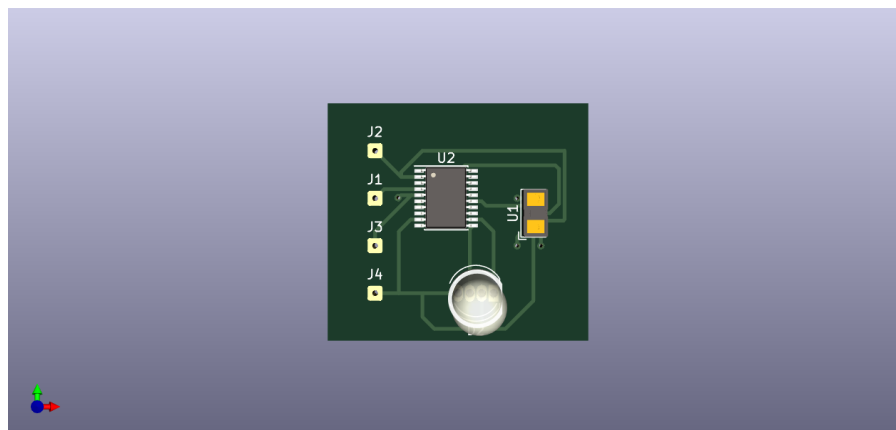


Figura 12: Modelo 3D do modulo sensor

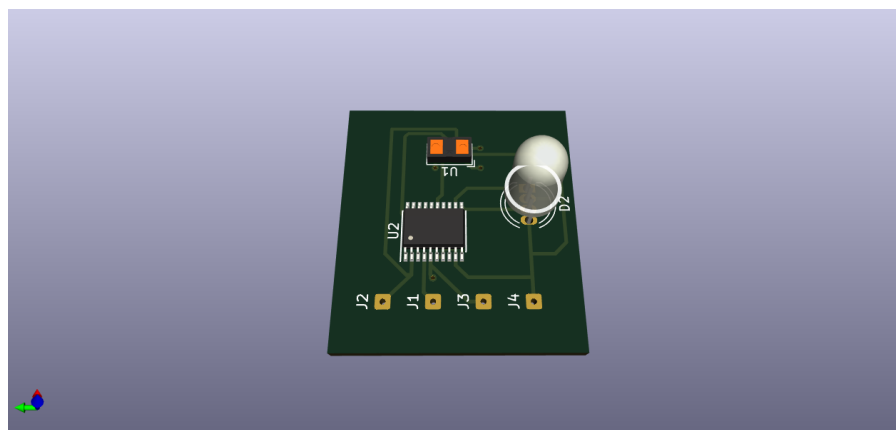


Figura 13: Outra posição do modelo 3D

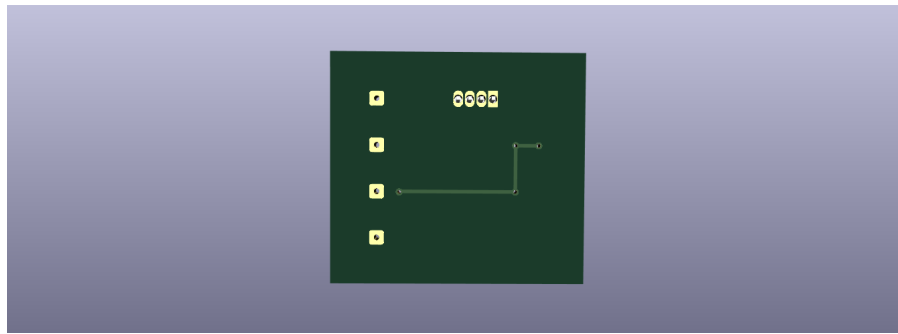


Figura 14: Layer inferior da pcb no modelo 3D