

Wireless Data Transfer

Vinícius Inacio Breda

4 de abril de 2022

Resumo

Este documento visa descrever a solução apresentada para o desafio proposto, bem como as justificativas para as tomadas de decisão relevantes para a solução final. Apresentando esquemáticos eletrônicos e design de placa de circuito impresso simples para integração da solução modular. Ao final, tem-se uma solução simples para a transferência de arquivos de, aproximadamente, 500 Kb, entre dois microcontroladores totalmente sem fios, com ao menos 100 metros de separação entre os sistemas e um dos dois sendo alimentado a partir de baterias.

1 Introdução

Dispositivos *Data Loggers* são os responsáveis pela captação e armazenamento de dados de sensores. Estes dados podem ser recolhidos fisicamente ou transmitidos por cabos ou por meio de tecnologias sem fio.

A transferência sem fio de arquivos possui vantagens práticas como, por exemplo, o monitoramento e controle sensoriado de equipamentos em grandes distâncias em locais ou ambientes em que passar cabos é imprático, difícil ou indesejado. O envio dos dados dos sensores ainda é crítico nessas situações, sendo necessária uma solução rápida e simples, facilmente expansível, com componentes relativamente baratos e amplamente disponíveis no mercado.

Os passos do projeto são:

- Definição de Microcontrolador;
- Definição de método de transmissão e recepção de dados;
- Definição de método de armazenamento de dados a serem transmitidos e recebidos;
- Definição da bateria para um dos circuitos;
- Desenvolvimento de esquemático de componentes eletrônico do circuito;
- Desenvolvimento de Placa de Circuito Impresso para ambos os sistemas;
- Desenvolvimento do código do sistema;
- Testes com componentes reais para validação.

	Consumo (mA)		Clock Base (MHz)	Pinos			Wi-Fi/BT	Preço (R\$)
	<i>Deep Sleep</i>	Máximo		Digitais	PWM	Analógicos		
Arduino UNO Rev.3	34	410	16	14	6	8	N/N	50~90*

* Dependendo da qualidade da placa, SMD ou PTH, original ou paralela

Tabela 1: Características Principais do Arduino

2 Microcontrolador

A escolha de microcontroladores é diretamente relacionada à atividade a ser executada. Temos os microcontroladores de empresas como a Texas Instruments ou ST Microelectronics, que são muito capazes de executar as tarefas, porém com uma dificuldade maior de obtenção no mercado brasileiro ao se comparar com placas Arduino ou ESP, mais difundidas no mercado e facilmente encontradas na maioria das lojas de componentes eletrônicos, tanto físicas quanto virtuais.

Levando isto em consideração e, também, o fator familiaridade, pode-se limitar a escolha às placas Arduino e ESP.

2.1 Arduino

A comunidade de desenvolvedores de sistemas baseados nos chips Atmel, utilizados nas placas Arduino, é muito ativa e projetos de tudo que se possa imaginar utilizam estes microcontroladores como central de processamento. Pode-se tomar isso como um fator positivo para o Arduino, por ter-se uma grande comunidade de apoio e muitas fontes de informação para desenvolvimento. Existem muitas bibliotecas de apoio para os vários módulos que podem ser utilizados nos projetos, porém, muitas dessas bibliotecas levam em consideração outros microcontroladores. Sendo assim, não se pode tomar isso como ponto para a definição do controlador.

Tem-se então, que recorrer à outros pontos característicos como, por exemplo, recursos presentes na placa base, preço, consumo de energia, etc. Os fatores considerados mais importantes estão apresentados na tabela 2.1. Um ponto importante a se considerar é que as placas Arduino UNO Wi-Fi R2 são consideravelmente mais caras, portanto foram desconsideradas.

2.2 ESP

Os microcontroladores da Espressif, mais conhecidos como ESP, também têm se destacado bastante no mundo embarcado, muito por até os modelos mais básicos apresentarem conexão Wi-Fi, podendo ser utilizados mais facilmente para aplicações de internet das coisas (IoT, *Internet of Things*). Este fator tem um peso grande quando se considera um projeto com transmissão de dados, já que pode-se criar pontos de acesso centrais e conectar diversos microcontroladores neles, que enviam dados periodicamente sem a necessidade da adição de módulos externos extras. Existem ainda, modelos de microcontroladores ESP que possuem bluetooth integrado, o que é mais um ponto positivo para estes modelos.

Dos modelos de ESP a ESP32 é a que melhor se equipara em preço com o Arduino UNO Rev3. Em testes realizados comparando as placas mais comuns encontradas no mercado entre si, tem-se que a ESP32 possui o melhor Wi-Fi, com a ESP-02 sendo

	Consumo (A)		Clock Base (MHz)	Pinos			Wi-Fi/BT	Preço (R\$)
	<i>Deep Sleep</i>	Máximo		Digitais	PWM	Analógicos		
ESP32	5u	240m	80~240	36	36	15	S/S	45~80*
ESP32 DevKit v1	5u	240m	80~240	25	21	15	S/S	45~80*
* Dependendo do DevKit e da placa selecionada								

Tabela 2: Características Principais da ESP32

a segunda melhor devido a sua antena externa. Para aplicações de DataLogger, a ESP32 apresenta muita flexibilidade por ter um grande número de portas com conversão analógica, possuir uma alta frequência de funcionamento e um baixo consumo de energia, podendo chegar na casa dos micro ampères, como pode ser visto na tabela 2.2. Possui alguns modelos com conector para antena externa, porém, instalar uma antena em um modelo que não possui conector é relativamente simples, basta isolar ambas as saídas para a antena presente na placa, remover sua tinta e soldar um conector SMA.

2.3 Definição de Microcontrolador

Para esta aplicação o microcontrolador ESP32 apresenta muitos benefícios que não podem ser ignorados. A presença de Bluetooth e Wi-Fi na mesma placa abrem grandes oportunidades para desenvolvimento e flexibilizam bastante como o projeto pode ser desenvolvido. A possibilidade de se utilizar o Wi-Fi LR para distâncias de até 1 Km em campo aberto, sem a necessidade de módulos ou antenas adicionais torna o controlador ainda mais atrativo. Sendo assim, já define-se o microcontrolador e a forma de transmissão de dados ao mesmo tempo.

O bluetooth pode ser uma forma de se enviar ao microcontrolador o arquivo que deve ser transmitido pela rede, por exemplo. Há muito o que se explorar com esta solução.

O código será desenvolvido levando em consideração o módulo DOIT ESP32 Dev-Kit1, com 30 pinos.

3 Armazenamento

Tendo em vista que um dos objetivos do projeto é o envio de arquivos de 500Kb, é necessária a definição de um local armazená-los. Algumas opções disponíveis são: Um cartão SD, um cartão microSD e a própria memória interna por meio de bibliotecas como a LITTLEFS.

O uso da memória interna é interessante pois não acrescenta custo algum ao projeto, simplificando-o bastante, porém limitando o tamanho dos arquivos que podem ser utilizados. Os módulos SD e microSD são bastante baratos, porém, exigem um cartão adicional, que normalmente não é. Seu uso é recomendado somente caso se deseje armazenar de 4MB até 4GB de dados.

3.1 Definição do Armazenamento

Nesta solução a memória interna será utilizada. Caso haja tempo, testes com um módulo microSD serão realizados para comparação de tempo de escrita e leitura.

O armazenamento inicial do arquivo de teste pode ser feito por bluetooth a partir do aplicativo Android *Serial Bluetooth Terminal*, ou de alguma variante para IOS. Pode-se, também, realizar o upload diretamente pela interface de programação do Arduino ou a partir do PlatformIO.

4 Transmissão de Dados

A transmissão de dados foi feita a partir da criação de um servidor e um cliente HTTP. O controlador utilizado como Base de recepção abre um ponto de acesso Wi-Fi para que outros controladores se conectem, este ponto de acesso possui um cliente HTTP que manda *GET requests* periodicamente a todos os controladores a ele conectados, um a um.

Cada controlador envia o arquivo armazenado em sua memória flash, não volátil, e os dados enviados são anexados ao final do arquivo presente no controlador base. Várias formas de manipular os dados podem ser implementadas, como uma marcação de tempo utilizando o RTC presente na placa. Também pode-se inserir o endereço único MAC de cada controlador para identificar qual enviou o dado. É possível, também, enviar uma linha que possui um identificador a partir do servidor HTTP em cada controlador.

No momento em que os controladores são programados, deve-se escolher qual modo de transmissão de dados será utilizado. O Wi-Fi Long Range só é ativado caso seja desejado. Este modo garante transmissões de até 1Km, comprometendo a velocidade de transmissão. Testes de envio em ambos os modos foram feitos e ambos geraram resultados esperados, com o arquivo sendo enviado rapidamente, em questão de segundos.

5 Alimentação

A alimentação do sistema não precisa ser muito complexa. Levando em consideração o uso dos modos de baixo consumo de energia do ESP32, uma única bateria seria capaz de mantê-lo ativo por horas. Como a maior parte da energia do sistema será consumida pelo próprio microcontrolador, pode-se levar em conta somente sua corrente máxima de 240mA para o dimensionamento de baterias. O consumo sendo mantido ao máximo por uma hora requer uma bateria de 240mAh, ou seja, para mantê-lo com, aproximadamente, 5 horas de autonomia, 2 baterias de 9V de 600mAh em paralelo seriam suficientes.

6 Esquemático

Para efeitos de aprendizado, foi utilizado o esquemático de uma das placas de desenvolvimento da ESP32, que utiliza o chip ESP32-WROOM-32. Algumas modificações foram feitas para adequá-la ao projeto de um DataLogger de sensores I²C, bem como para aumentar a eficiência do envio de energia.

O esquemático dos circuitos se encontram nas figuras 1 e 2. Com uma pequena correção: o botão "SEND" será utilizado como um chaveador que liga e desliga o bluetooth.

6.1 Placa de Circuito Impresso

As placas desenvolvidas para receptor e transmissor estão nas figuras 3 e 4.

Levando em consideração que a ESP32 possui todos os componentes necessários para armazenar um arquivo de 500KB e servir de base para recepção de dados, o circuito torna-se bem simples, sendo quase uma cópia idêntica dos kits de desenvolvimentos, com algumas alterações para eficiência.

Já o circuito de transmissão sofre algumas alterações para que seja possível alimentá-lo com baterias de 9V e manter o pino V_{in} como fonte de alimentação 5V para periféricos ou sensores.

Também foi adicionado um circuito conversor de nível lógico para módulos I²C que possuem saída lógica em 5V, transformando-a em 3.3V, como é suportado pelo ESP32.

7 Código

Os códigos estão armazenados no [GitHub do projeto](#), utilizado para facilitar o versionamento e controle de alterações do código.

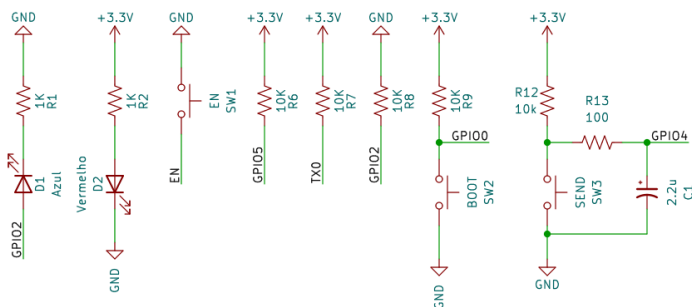
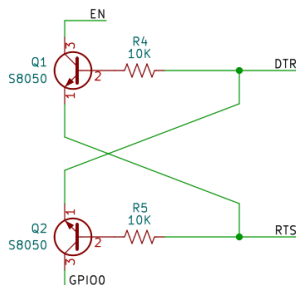
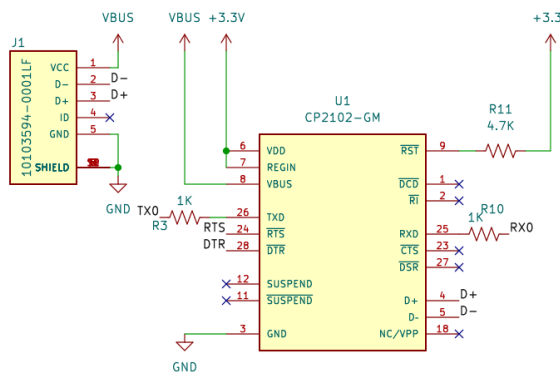
8 Testes

Os testes foram realizados em curta distância devido à dificuldade de se realizar testes de longo alcance sem que fossem utilizadas outras formas de alimentação, o que não era possível durante o desenvolvimento. A garantia de distância mínima maior que 100 metros é dada pelo modo de transmissão Wi-Fi Long Range. Porém, pode-se dizer que é possível garantir uma transmissão para mais de 100 metros utilizando o Wi-Fi comum ao se realizar alterações na antena da placa ou utilizar uma placa com conector externo de antenas.

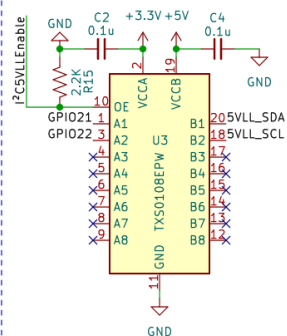
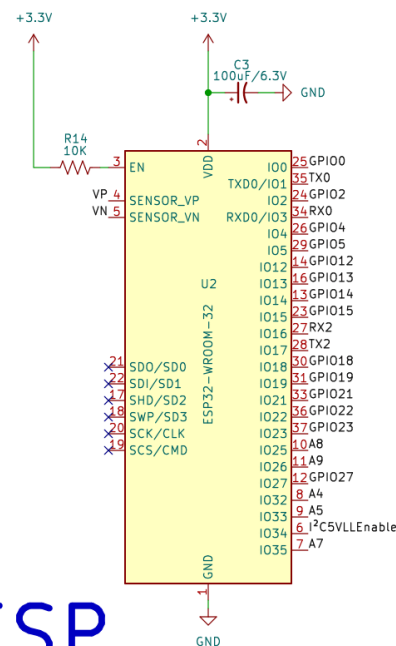
Foi gerado um arquivo chamado "loremipsum.txt" com pouco mais de 500KB de dados que foi armazenado no sistema interno de um dos controladores, considerado o Transmissor. Este arquivo é enviado por Wi-Fi em poucos segundos.

Não foi utilizada uma forma de medir o tempo de envio, porém isso seria de fácil implementação utilizando as funções de tempo das bibliotecas do Arduino e do ESP32.

USB

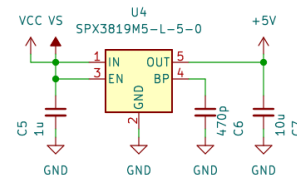
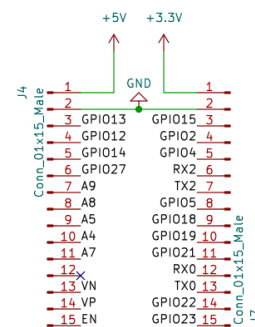


ESP

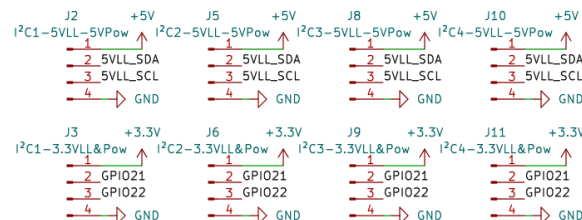
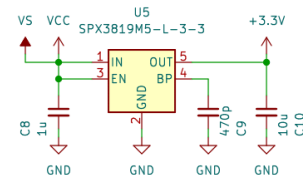
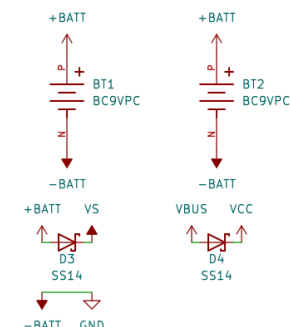


I²C Bus

GPIO



Power



Circuito de transmissão de dados.

Sheet: /
File: traction-esp.kicad_sch

Title: DataLogger de sensores I²C - ESP32 DEVKIT 1

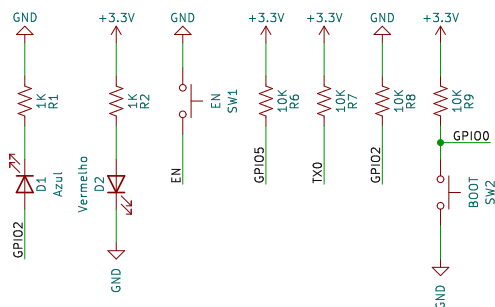
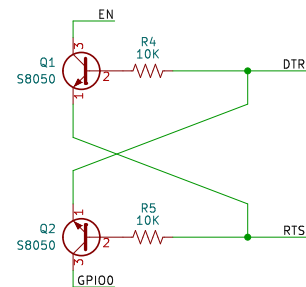
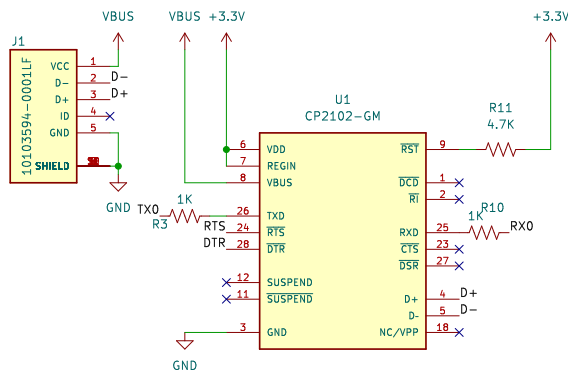
Size: A4 Date: 2022-03-26

KiCad E.D.A. kicad (6.0.4)

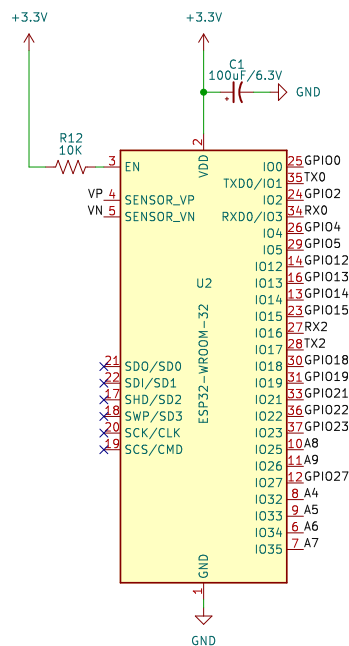
Rev: 1

Id: 1/1

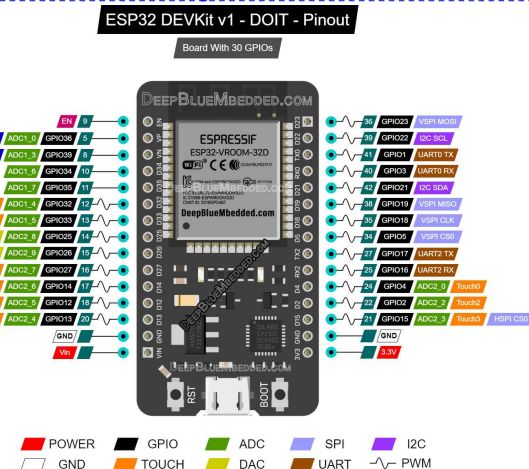
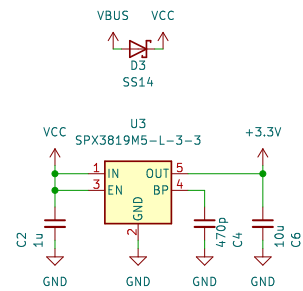
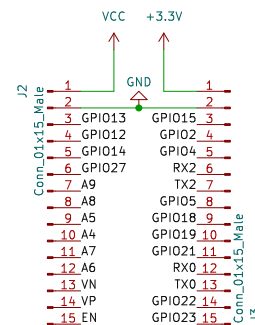
USB



ESP



GPIO Power



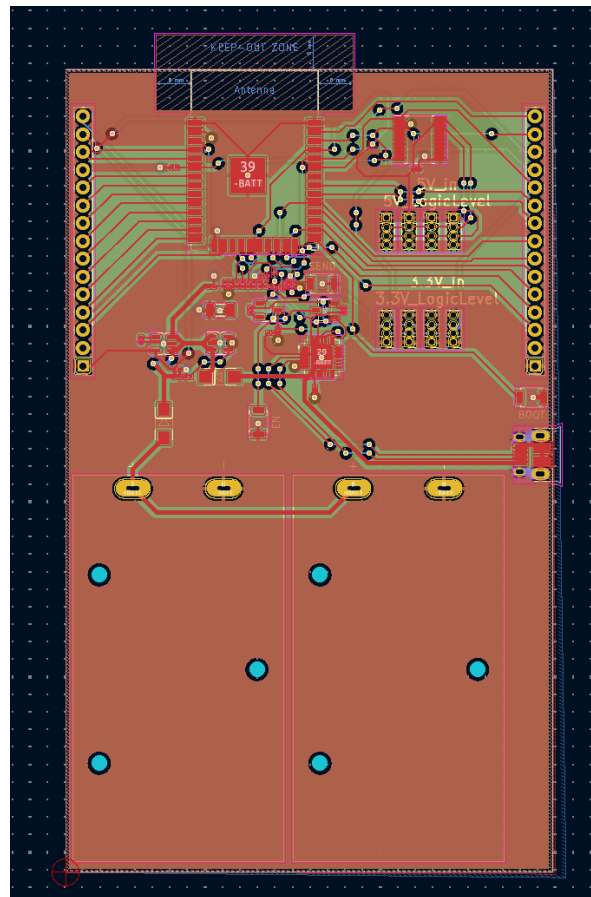


Figura 3: PCI - Transmissor

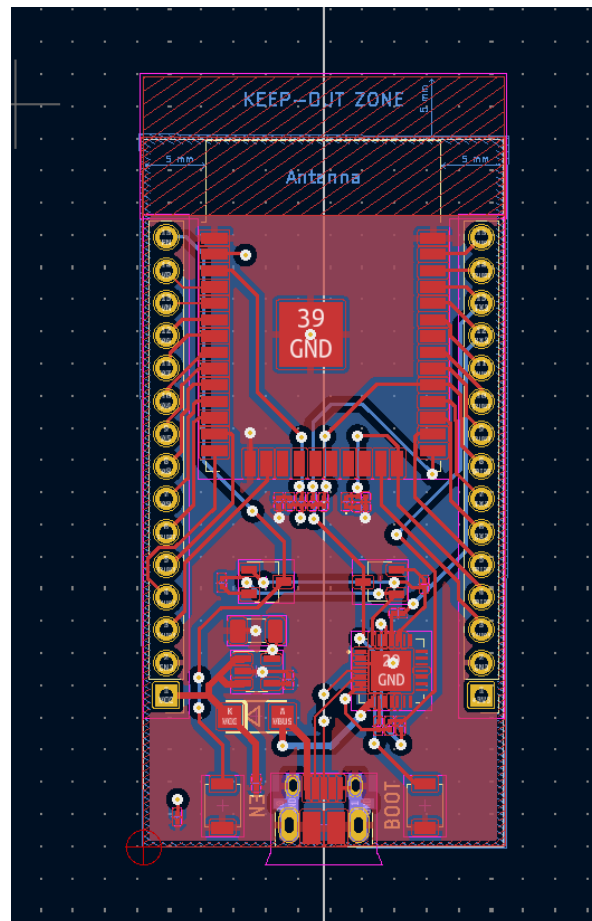


Figura 4: PCI - Receptor

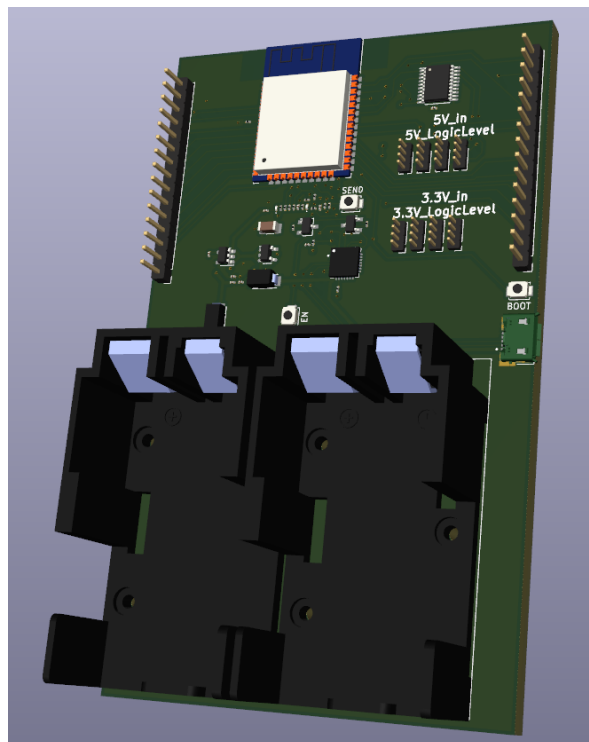


Figura 5: PCI - Transmissor

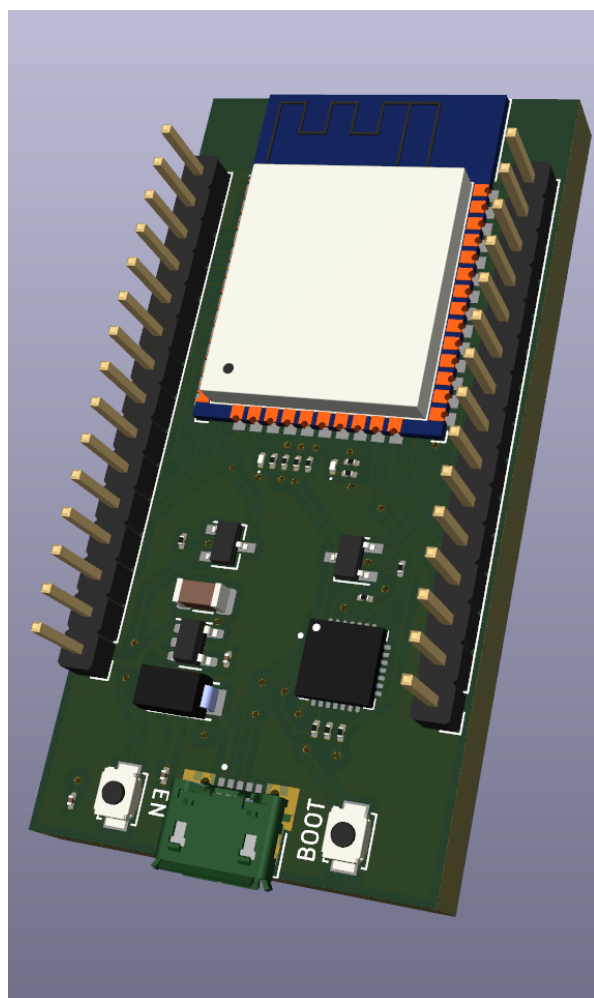


Figura 6: PCI - Receptor