

MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E
TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
(Real Academia de Artilharia, Fortificação e Desenho, 1792)

SEÇÃO DE ENSINO DE ENGENHARIA ELÉTRICA (SE/3)
PROJETO DE SISTEMAS EMBARCADOS
IMPLEMENTAÇÃO DE SENSOR DE OBSTÁCULO COM ESP-32

PROFESSOR: CAP QEM HERBERT AZEVEDO SÁ

1° TEN FRANCISCO **NAZÁRIO** PEREIRA JÚNIOR - 20023
1° TEN RUAN DE SOUSA **MONTENEGRO** - 21056

Sumário

1	Introdução	2
2	Metodologia	2
2.1	Materiais Utilizados	2
2.2	Objetivo	2
3	Código-Fonte	2
4	Esquemático	5
5	Resultados	7
6	Conclusão	9

1 Introdução

Este projeto didático foi desenvolvido com o objetivo de se ter uma experiência prática no uso de sistemas embarcados. Para isso, utilizando a plataforma ESP-32, foi implementado um sistema capaz de receber dados de um sensor de obstáculo e transmití-los via conexão Wi-Fi para a plataforma ThingSpeak. Tal abordagem permitiu a aplicação de conceitos teóricos de eletrônica, programação e comunicação de dados em um contexto prático e altamente passível de emprego em situações do cotidiano do engenheiro.

2 Metodologia

O projeto foi dividido em etapas, desde a escolha dos componentes, montagem do circuito, desenvolvimento do código, até a transmissão de dados para a plataforma online ThinkSpeak.

2.1 Materiais Utilizados

- Microcontrolador ESP-32
- Sensor de obstáculo KY-032
- Bateria de 11V
- Jumpers diversos para conexões
- Protoboard
- Computador para programação e monitoramento

O sensor KY-032 foi conectado ao ESP-32, que, por sua vez, foi programado para ler os dados do sensor e enviar essas informações para a nuvem via Wi-Fi.

2.2 Objetivo

Com a implementação de um circuito simples envolvendo os materiais listados e o código que será apresentado na próxima seção, busca-se uma aplicação em que o usuário, acessando a sua página do ThingSpeak, consiga, de onde estiver, ver quantas vezes algo passou na frente de seu sistema embarcado. Para isso, o ESP-32 contará quantas vezes o KY-032 detectou um obstáculo no intervalo de 15s demandados para o envio de um novo dado para a web. Após o envio, o contador é zerado. Desta forma, é possível saber quantas vezes um obstáculo é detectado, ou algo passa na frente do sensor, através do tempo. Tal aplicação pode ser usada para se investigar quantas vezes uma porta é aberta, ou como um contador em geral, bastando que passe em sua frente para que a contagem seja realizada.

3 Código-Fonte

O código-fonte utilizado neste projeto foi desenvolvido em C++, utilizando o ambiente Arduino IDE. Abaixo será inserido o trecho do código utilizado para leitura dos dados do sensor e envio para o ThingSpeak:

Listing 1: Código ESP-32 para envio de dados ao ThingSpeak

```

1 // Inclusão das bibliotecas necessárias para conexão Wi-Fi e
  // requisitos HTTP
2 #include <WiFi.h>
3 #include <HTTPClient.h>
4
5 // Credenciais da rede Wi-Fi
6 const char* ssid = "S23 de Francisco";
7 const char* password = "pyx728fkehiwwdp";
8
9 // Pino digital conectado ao sensor KY-032
10 const int digPin = 12;
11
12 // URL da API do ThingSpeak para envio de dados
13 const char* server = "http://api.thingspeak.com/update";
14 const char* apiKey = "Z9IG5H47M54IJRKC"; // Chave da API do canal
15
16 // Variável para contar quantas vezes um obstáculo foi detectado
17 int countDetections = 0;
18
19 // Variáveis para controle de tempo
20 unsigned long previousMillis = 0;
21 const long interval = 15000; // Envia dados a cada 15 segundos
22
23 void setup() {
24     // Configura o dos pinos
25     pinMode(digPin, INPUT); // Pino do sensor como entrada
26     pinMode(13, OUTPUT);    // LED no pino 13 como saída (
        indicador)
27
28     Serial.begin(115200);    // Inicializa comunicação serial para
        depuração
29
30     // Conexão com a rede Wi-Fi
31     WiFi.begin(ssid, password);
32     Serial.print("Conectando-se ao Wi-Fi");
33     while (WiFi.status() != WL_CONNECTED) {
34         delay(500);
35         Serial.print(".");
36     }
37     Serial.println("\nWiFi conectado!");
38 }
39
40 void loop() {
41     // Leitura do sensor de obstáculo (LOW indica obstáculo
        detectado)
42     int digValue = digitalRead(digPin);
43
44     // Se obstáculo detectado, incrementa o contador
45     if (digValue == LOW) {
46         Serial.println("Objeto detectado");

```

```

47     countDetections++;
48 } else {
49     Serial.println("Nenhum objeto detectado");
50 }
51
52 // Acende o LED como indicador de leitura ativa
53 digitalWrite(13, HIGH);
54
55 // Mostra valores lidos no monitor serial
56 Serial.print("Valor lido do sensor: ");
57 Serial.println(digValue);
58 Serial.print("Contador de detecções: ");
59 Serial.println(countDetections);
60
61 // Verifica se já passou o intervalo de tempo para envio
62 unsigned long currentMillis = millis();
63 if (currentMillis - previousMillis >= interval) {
64     previousMillis = currentMillis;
65
66     // Se estiver conectado ao Wi-Fi, envia os dados
67     if (WiFi.status() == WL_CONNECTED) {
68         HTTPClient http;
69         // Monta a URL com os dados a serem enviados
70         String url = String(server) + "?api_key=" + apiKey + "&
71             field1=" + String(countDetections);
72         http.begin(url); // Inicializa a requisição HTTP
73         int httpResponseCode = http.GET(); // Envia a requisição
74         GET
75         countDetections = 0; // Zera o contador após o envio
76
77         // Exibe resposta da requisição
78         if (httpResponseCode > 0) {
79             Serial.print("Código de resposta HTTP: ");
80             Serial.println(httpResponseCode);
81         } else {
82             Serial.print("Erro ao enviar dados: ");
83             Serial.println(http.errorToString(httpResponseCode).c_str
84                 ());
85         }
86
87         http.end(); // Finaliza conexão HTTP
88     } else {
89         // Caso Wi-Fi desconecte, tenta reconectar
90         Serial.println("Wi-Fi desconectado. Tentando reconectar...");
91         ;
92         WiFi.begin(ssid, password);
93     }
94 }
95
96 delay(100); // Pequeno atraso para evitar sobrecarga no loop
97 }

```

4 Esquemático

Nesta seção, estão apresentados os esquemas do circuito: esquemático - que será posteriormente adicionado em anexo em uma imagem maior, esquema PCB e modelo 3D da PCB montada. Todos esses esquemas foram feitos no software CAD indicado pelo professor: EasyEDA.

Por orientação do professor, em vez de se representar uma entrada para o sensor KY-032 como apenas espaços para encaixe dos seus 4 terminais, implementou-se componente a componente do sensor na placa.

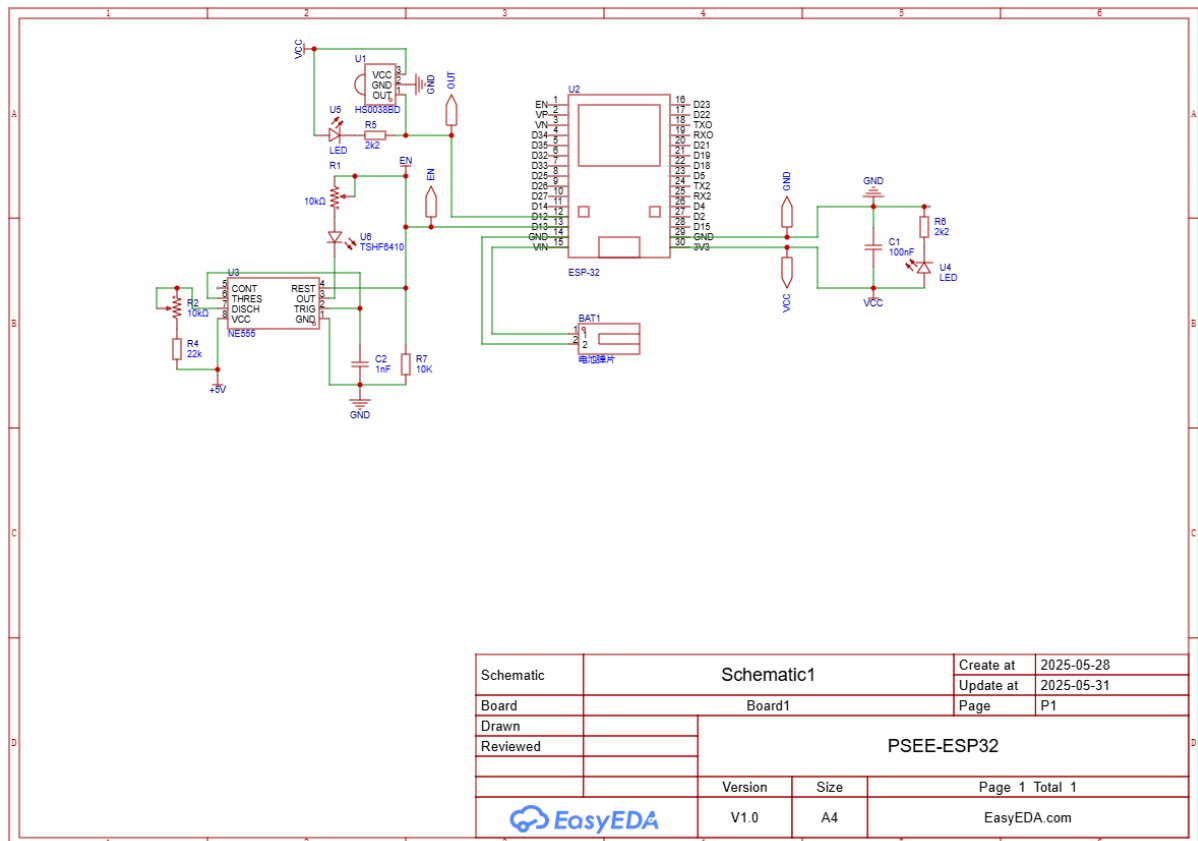


Figura 1: Esquemático do circuito feito em EasyEDA

Na Figura 2, as rotas em vermelho são no top layer, enquanto as rotas em azul são no bottom layer.

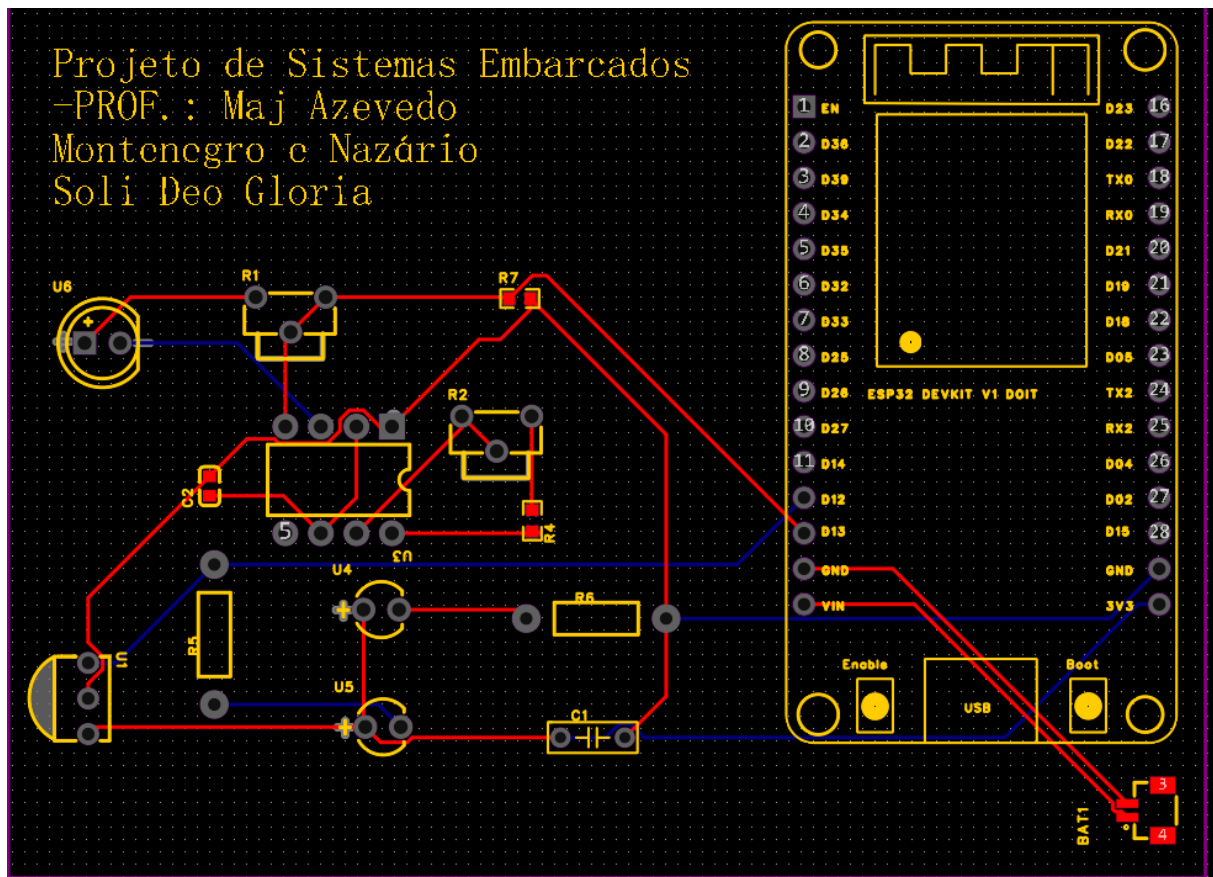


Figura 2: Esquema da PCB em EasyEDA

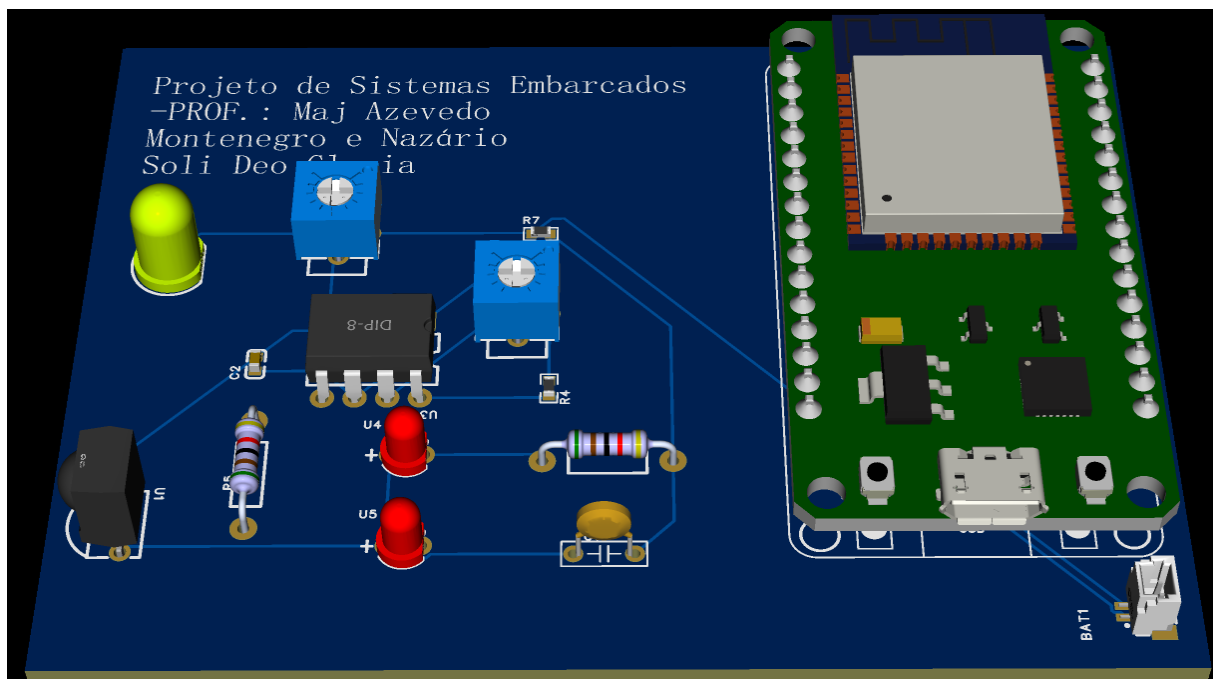


Figura 3: Representação 3D da placa feita em EasyEDA - top layer

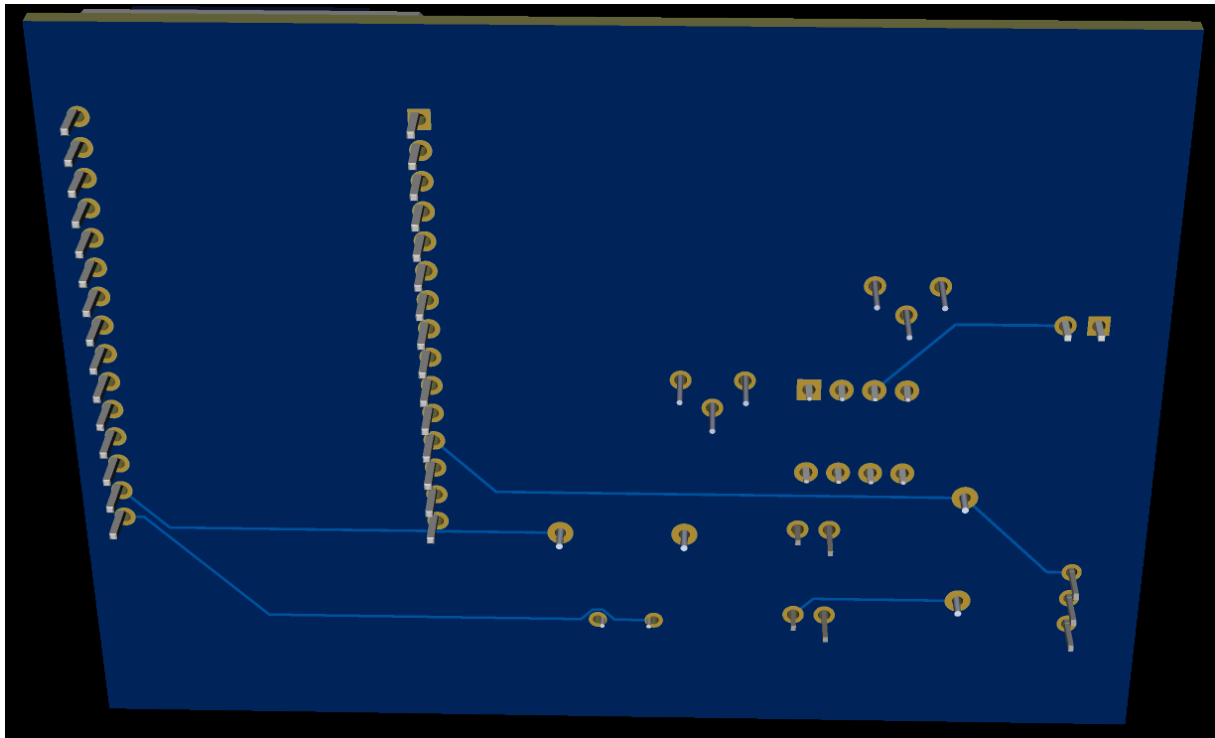


Figura 4: Representação 3D da placa feita em EasyEDA - bottom layer

5 Resultados

As imagens a seguir ilustram o funcionamento do sistema, incluindo a montagem dos componentes, a interface do ThingSpeak com os dados recebidos dos testes realizados.

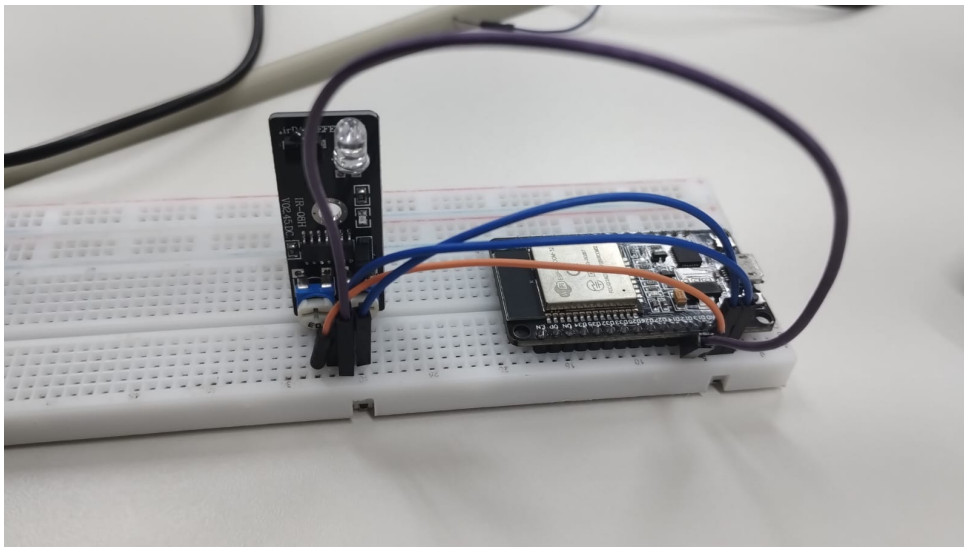


Figura 5: Montagem do circuito com ESP-32 e KY-032

O gráfico do ThingSpeak mostra o dado recebido a cada 15s: o número de contagens de obstáculo feita nesse intervalo de tempo. Conforme exibido no código, o delay com que o sensor registra um valor que pode incrementar ou não o contador de detecções é de 100ms. Por isso, os valores exibidos no gráfico variam entre 0 e 150 conforme Figura 7.

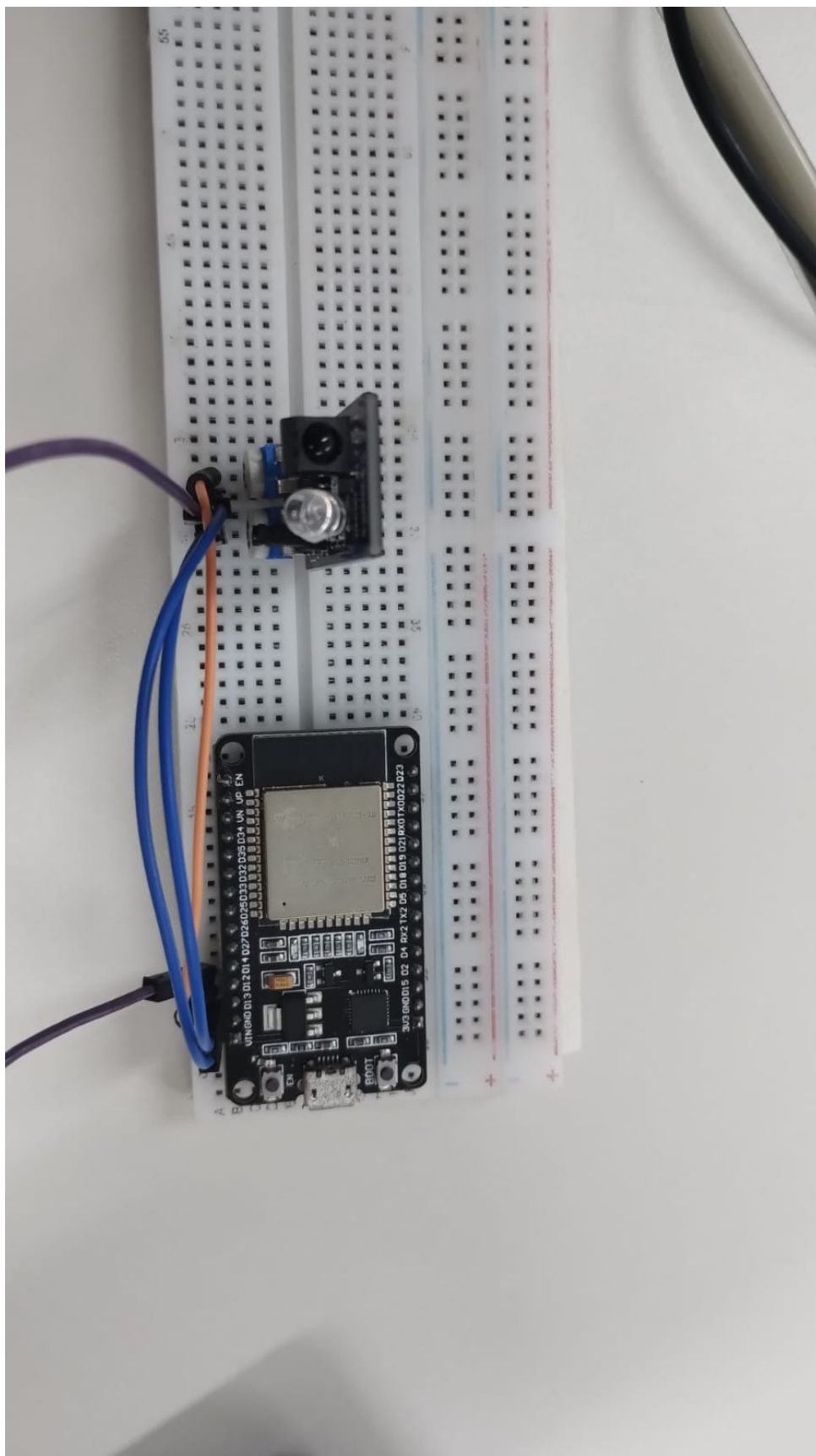


Figura 6: Montagem do circuito com ESP-32 e KY-032

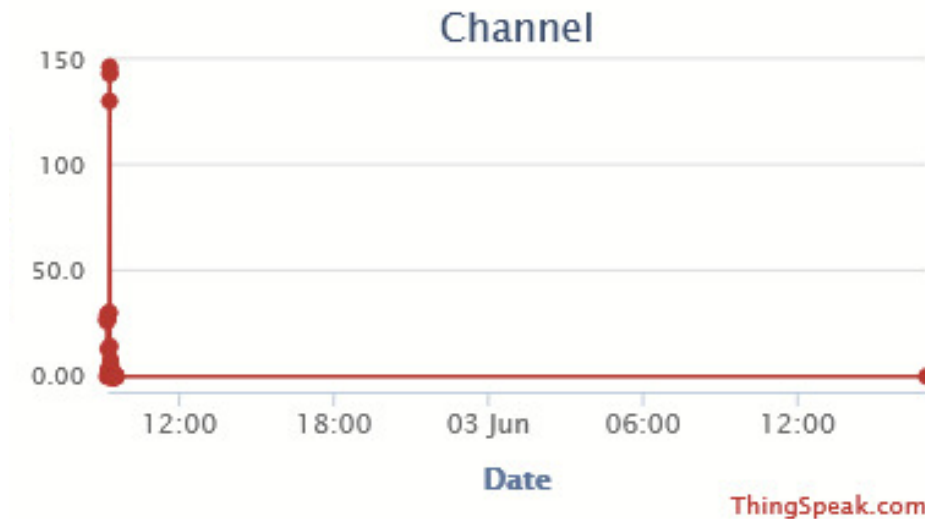


Figura 7: Interface do ThingSpeak exibindo os últimos registros realizados

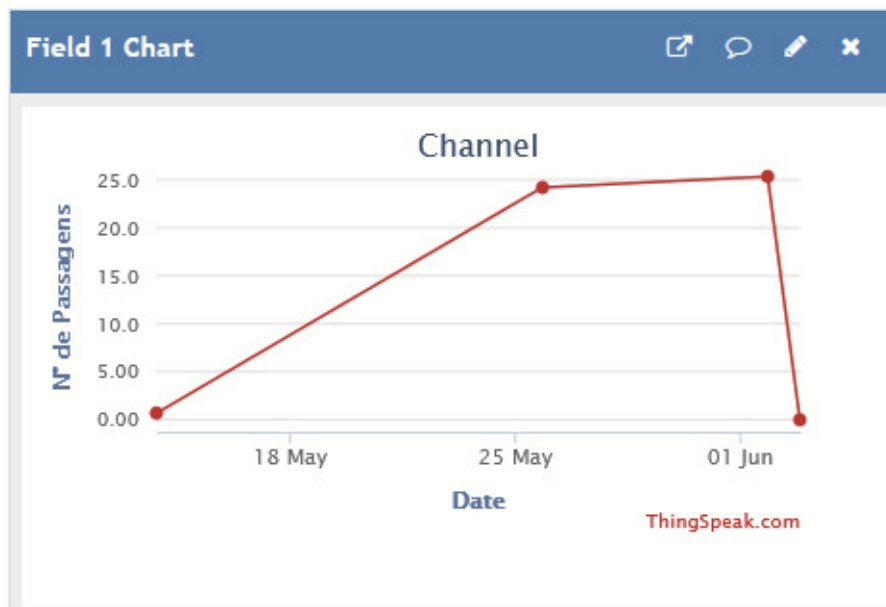
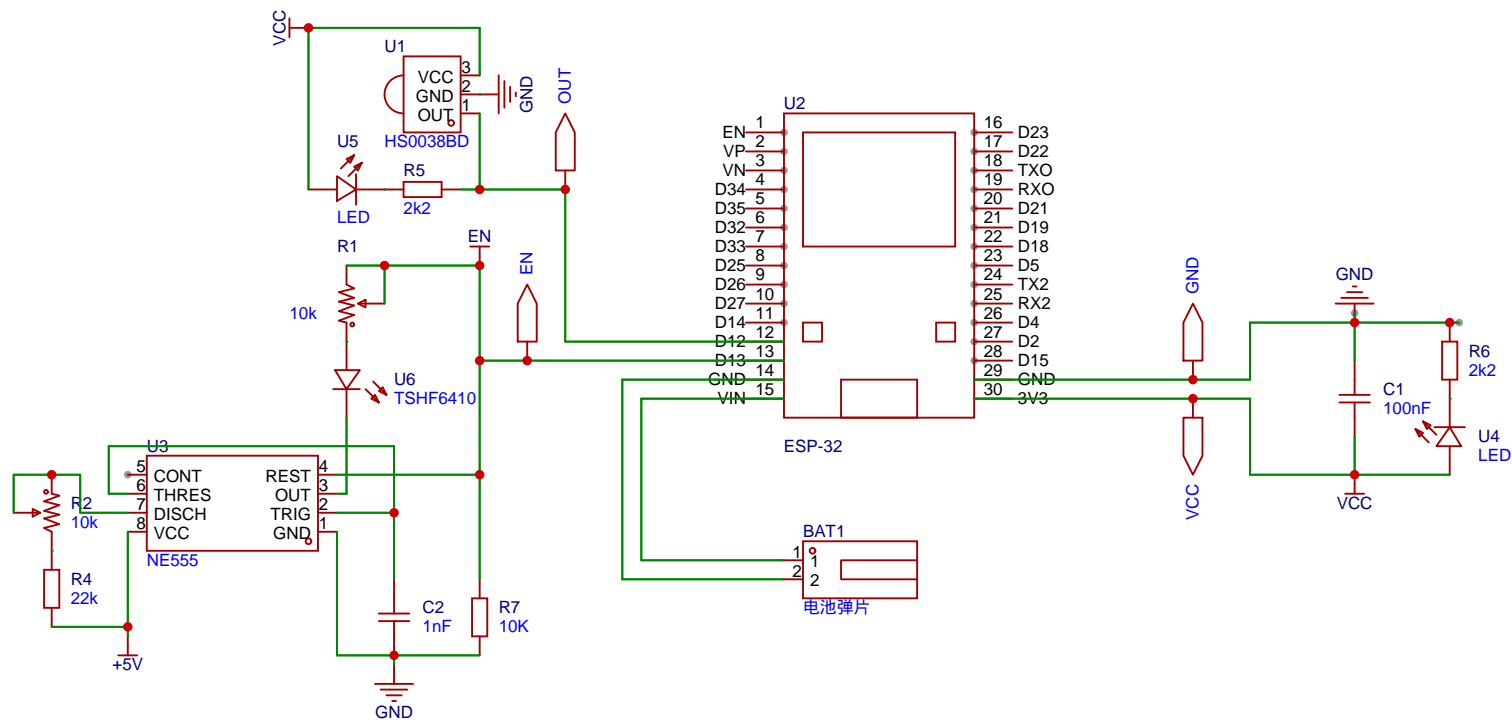


Figura 8: Interface do ThingSpeak exibindo médias dos dados recebidos sintetizados durante todos os testes realizados

6 Conclusão

O projeto apresentado trouxe resultados significativos no processo de aprendizagem em sistemas embarcados. A execução das tarefas, desde a escolha do sensor a ser empregado, montagem do circuito até a transmissão dos dados para a nuvem, representou um grande ganho em know-how para os futuros engenheiros, permitindo-lhes compreender na prática o funcionamento de sistemas modernos de monitoramento, como uma forma primária de *IoT*. Além disso, o trabalho em grupo e a resolução de problemas reais foram aspectos fundamentais para o desenvolvimento de habilidades técnicas e de aspectos atitudinais.



Schematic	Schematic1			Create at	2025-05-28
Board	Board1			Update at	2025-06-04
Drawn		PSEE-ESP32			
Reviewed					
		Version	Size	Page 1 Total 1	
EasyEDA		V1.0	A4	EasyEDA.com	