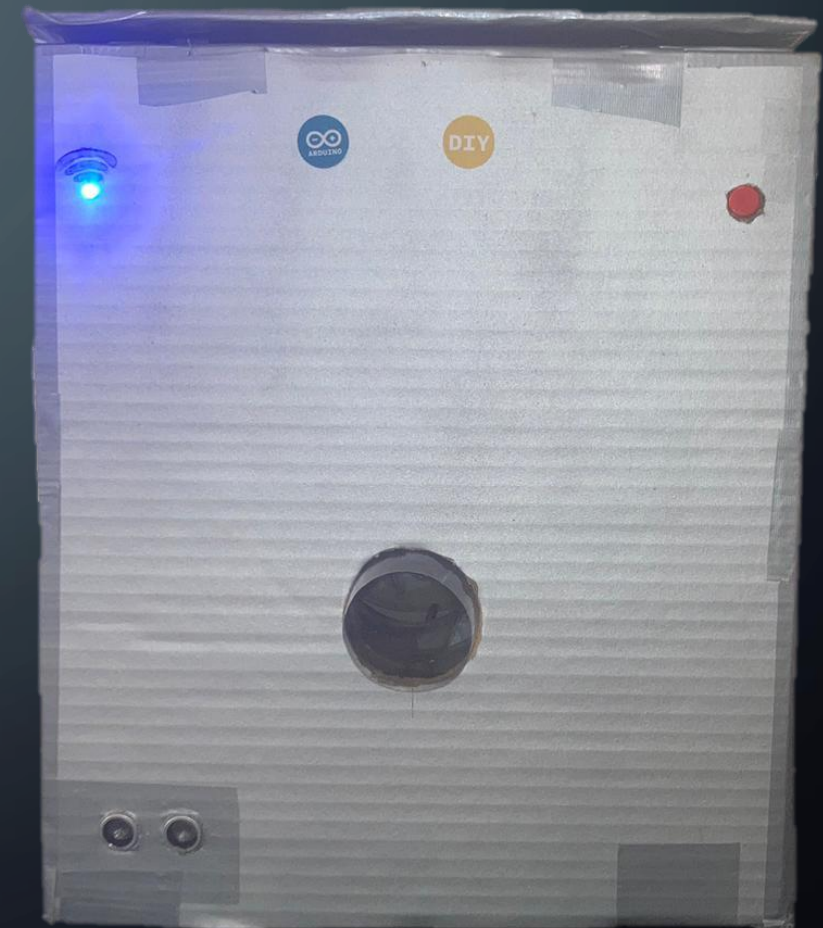


Progetto realizzato da Schito Christian, Ruggeri Emanuele & Dimitri Luca  
Classe 3° A informatica Enrico Mattei, Maglie (LE)

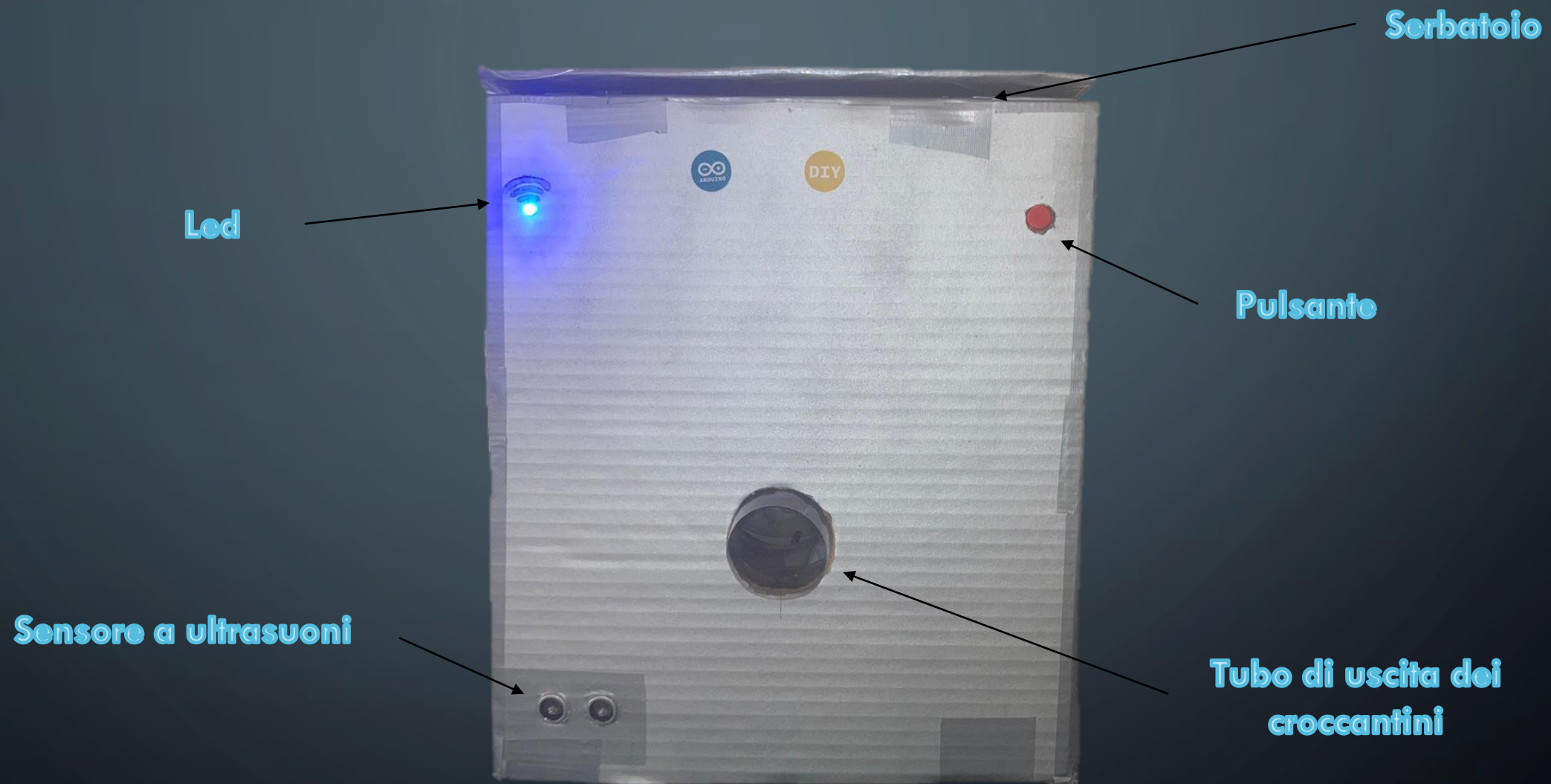
**Unica**  
Capolavoro

# Smart Dispenser for Pet

made with Arduino UNO R4 WIFI



# Struttura



# Led

Il led si accenderà quando  
l'Arduino si collegherà alla  
rete wi-fi prestabilita da  
codice





# Sensore a ultrasuoni

Rileva la presenza del nostro animale domestico, se esso verrà rilevato, il sensore invierà un segnale alla scheda Arduino che farà ruotare un servomotore facendo cadere una porzione di croccantini



# Serbatoio

Al di sopra del nostro distributore smart troviamo un serbatoio chiudibile per depositare i croccantini



# Pulsante



Il pulsante, quando premuto, manderà un segnale alla scheda Arduino che farà ruotare il servomotore, facendo cadere una porzione di croccantini

## Tubo di uscita dei croccantini

Quando il servomotore viene azionato i croccantini scendono da questo tubo.

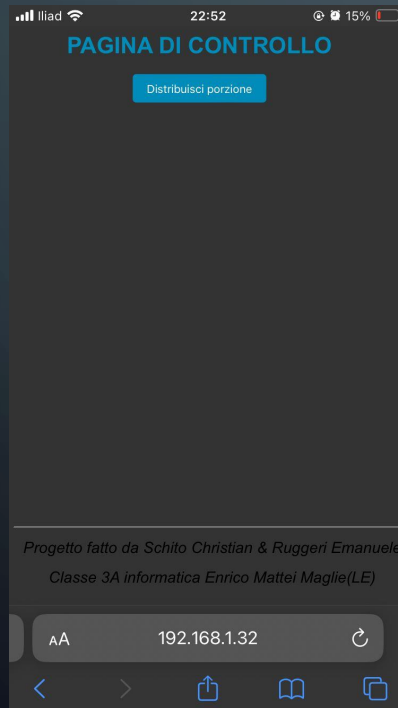




# Metodi di Attivazione

Ci sono 3 diversi modi per controllare il distributore:

## Il web server



## Il pulsante



## Il sensore a ultrasuoni





# Il web server



# Il web server



Quando il pulsante sul web server viene premuto, manderà una http request alla scheda esp32 dell'Arduino che farà ruotare il servomotore, facendo cadere una porzione di croccantini

# Il web server



Con il pulsante 'Torna indietro' è possibile tornare alla schermata precedente per distribuire ancora.

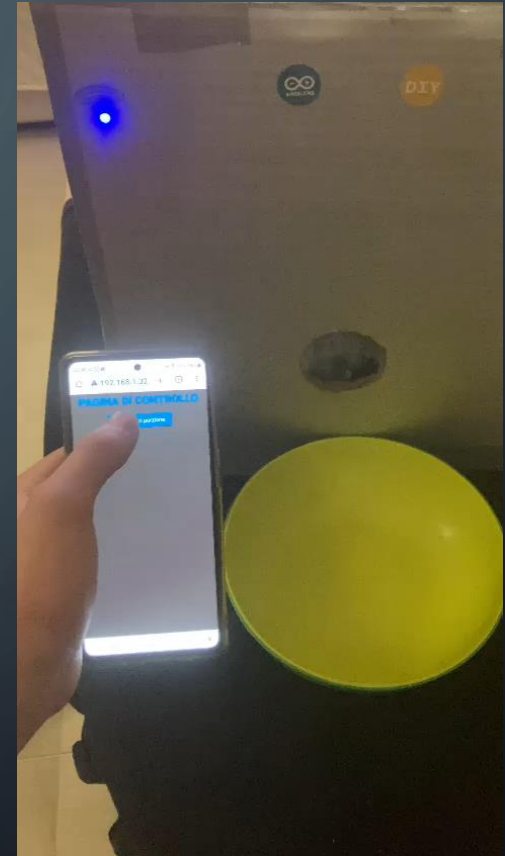


# Il Codice

```
1 #include "Arduino_LED_Matrix.h"
2 #include <Arduino.h>
3 #include <WiFi.h>
4 #include <Servo.h>
5
6 ArduinoLEDMatrix matrix;
7
8 const char *ssid = "Muttia";
9 const char *password = "12345678";
10
11 const int triggerPin = 2; // Pin di trigger del sensore ad ultrasuoni
12 const int echoPin = 3; // Pin di echo del sensore ad ultrasuoni
13 const int servoPin = 9; // Pin del servomotore
14 const int buttonPin = 7; // Pin del bottone
15 const int pinLed = 5; // Pin del led
16
17 Servo myServo;
18 WiFiServer server(80);
19
20 void setup() {
21     myServo.write(0);
22     Serial.begin(9600);
23     pinMode(triggerPin, OUTPUT);
24     pinMode(echoPin, INPUT);
25     pinMode(buttonPin, INPUT_PULLUP); // Imposta il bottone come input con pull-up interno
26     pinMode(pinLed, OUTPUT);
27     myServo.attach(servoPin);
28
29     WiFi.begin(ssid, password);
30     while (WiFi.status() != WL_CONNECTED) {
31         delay(1000);
32         Serial.println("Connecting to WiFi...");
33         digitalWrite(pinLed, HIGH);
34         delay(100);
35         digitalWrite(pinLed, LOW);
36     }
37     Serial.println("Connected to WiFi!");
38     Serial.println(WiFi.localIP());
39     server.begin();
40     analogWrite(pinLed, 150);
41 }
42
43 void loop() {
44     // Gestione delle richieste HTTP
45     WiFiClient client = server.available();
46     if (client) {
47         Serial.println("New Client.");
48         String currentLine = "";
49         while (client.connected()) {
50             if (client.available()) {
51                 char c = client.read();
52                 Serial.write(c);
53                 if (c == '\n') {
54                     if (currentLine.length() == 0) {
55                         client.println("HTTP/1.1 200 OK");
56                         client.println("Content-type:text/html");
57                         client.println();
58                         client.println("<html><head><title>Distributore Smart</title>");
59                         client.println("<style>");
60                         client.println("body { font-family: Arial, sans-serif; background-color: #333; }");
61                         client.println("h1 { color: #008CBA; text-align: center; }");
62                         client.println("button { padding: 10px 20px; font-size: 16px; background-color: #008CBA; color: white; border: none; border-radius: 5px; cursor: pointer; }");
63                         client.println("button:hover { background-color: #005F6B; }");
64                         client.println("input[type='button'] { padding: 10px 20px; font-size: 16px; background-color: #4CAF50; color: white; border: none; border-radius: 5px; cursor: pointer; }");
65                         client.println("input[type='button']:hover { background-color: #45a049; }");
66                         client.println("</style>");
67                         client.println("</head><body>");
68                         client.println("<div style='text-align:center;'>");
69                         client.println("<form method='get'>");
70                         client.println("<button name='button' value='1'>Distribuisci porzione/button");
71                         client.println("</form>");
72                         client.println("<div id='bottomDiv' style='position: absolute; bottom: 0; width: 100%; background-color: #333; padding: 10px; text-align: center;'>");
73                         client.println("<div style='text-align:center;'>");
74                         client.println("<p>Progetto fatto da Schito Christian & Ruggeri Emanuele</p>");
75                         client.println("<p>Classe 3A Informatica Enrico Mattei Maglie(LE)</p>");
76                         client.println("</div>");
77                         client.println("</div></body></html>");
78                         break;
79                     } else {
80                         currentLine += c;
81                     }
82                 }
83             }
84         }
85     }
86 }
```

```
88 }
89 } else if (c != '\r') {
90     currentLine += c;
91 }
92 if (currentLine.endsWith("GET /?button=1")) {
93     // Simulate button press
94     digitalWrite(buttonPin, LOW); // Assuming active LOW button
95     delay(100); // Debounce delay
96     digitalWrite(buttonPin, HIGH);
97     delay(500); // Extra delay to mimic button press
98     client.println("<html><head><title>Distributore Smart</title>");
99     client.println("<style>");
100     client.println("body { font-family: Arial, sans-serif; background-color: #333; }");
101     client.println("h1 { color: #4CAF50; text-align: center; }");
102     client.println("button { padding: 10px 20px; font-size: 16px; background-color: #008CBA; color: white; border: none; border-radius: 5px; cursor: pointer; }");
103     client.println("button:hover { background-color: #005F6B; }");
104     client.println("input[type='button'] { padding: 10px 20px; font-size: 16px; background-color: #4CAF50; color: white; border: none; border-radius: 5px; cursor: pointer; }");
105     client.println("input[type='button']:hover { background-color: #45a049; }");
106     client.println("</style>");
107     client.println("</head><body>");
108     client.println("<div style='text-align:center;'>");
109     client.println("<input type='button' value='Torna indietro' onClick='history.go(-1);return true;' name='button'>");
110     client.println("<div id='bottomDiv' style='position: absolute; bottom: 0; width: 100%; background-color: #333; padding: 10px; text-align: center;'>");
111     client.println("<div style='text-align:center;'>");
112     client.println("<p>Progetto fatto da Schito Christian & Ruggeri Emanuele</p>");
113     client.println("<p>Classe 3A Informatica Enrico Mattei Maglie(LE)</p>");
114     client.println("</div>");
115     client.println("</div></body></html>");
116     // Girare il servomotore di 90 gradi
117     myServo.write(90);
118     delay(500); // Aspetta un secondo
119     // Riportare il servomotore alla posizione iniziale
120     myServo.write(0);
121     break;
122 }
123 }
124 }
125 }
126 delay(10);
127 client.stop();
128 Serial.println("Client disconnected.");
129 }
130
131 // Gestione del servomotore
132 if (digitalRead(buttonPin) == LOW) {
133     // Girare il servomotore di 90 gradi
134     myServo.write(90);
135     delay(800); // Aspetta un secondo
136
137     // Riportare il servomotore alla posizione iniziale
138     myServo.write(0);
139 }
140
141 //Continua...
142 }
143 }
144 }
```

# Video



# Conclusione

Il distributore di croccantini smart con web server e rilevatore a ultrasuoni offre una soluzione pratica e innovativa per la gestione dell'alimentazione degli animali domestici. Permette ai proprietari di programmare e controllare la distribuzione del cibo da remoto, assicurando che gli animali ricevano il giusto apporto nutrizionale anche in assenza dei padroni. Il rilevatore a ultrasuoni garantisce un'erogazione efficiente, evitando sprechi e malfunzionamenti. Questo sistema offre comodità e tranquillità, migliorando la qualità della vita degli animali e dei loro proprietari. In futuro, il dispositivo potrebbe essere ulteriormente sviluppato con nuove funzionalità per una gestione ancora più efficiente e personalizzata.