**Documentazione Progetto E-Library IoT**

Obiettivo del progetto:

Realizzare un sistema di monitoraggio ambientale all’interno della biblioteca scolastica, con l’obiettivo di passare la lettura dei dati su una applicazione Android.

Gestire un sistema per il controllo degli accessi tramite card RFID.

I dati, oltre a essere visualizzabili tramite app, verranno mostrati in tempo reale su un display LCD.

Specifiche del progetto:

Il progetto si divide in in diverse sezioni:

1. Installare luci controllabili tramite arduino e app;
2. Installare sensore di umidità e di temperatura, con dati visualizzabili tramite app;
3. Inserire una piantina, che verrà monitorata tramite sensore di umidità collegato ad Arduino ed app;
4. Inserire sistema per la lettura di una carta magnetica, per il riconoscimento dell’insegnante che sta utilizzando la biblioteca con la propria classe;
5. Controllo della luminosità.

Descrizione delle varie sezioni:

# Sezione 1

L’obiettivo e` installare una lampadina controllabile tramite l’app.

In caso di realizzazione del progetto su una scala ridotta, per simulare l’utilizzo di una lampadina, utilizzeremo un led.

Per la realizzazione di questa sezione abbiamo bisogno di:

*Arduino Uno R3*

*Modulo Bluetooth HC-06*

*LiquidCrystal\_I2C - LCD 1602 con interfaccia I2C*

*LED*

*Resistenza 220Ω*

*Cavi di collegamento - Jumper wires*

*Breadboard - Per il collegamento dei componenti*

# Sezione 2

L’obiettivo è monitorare la temperatura e l'umidità all’interno della biblioteca attraverso l’utilizzo di un sensore DHT 11, i dati letti da Arduino verranno mostrati in tempo reale su un display LCD e su l’app android.

Per la realizzazione di questa sezione abbiamo bisogno di:

*Arduino Uno R3*

*DHT11 - Sensore di temperatura e umidità*

*Modulo Bluetooth HC-06*

*LiquidCrystal\_I2C - LCD 1602 con interfaccia I2C*

*Cavi di collegamento - Jumper wires*

*Breadboard - Per il collegamento dei componenti*

# Sezione 3

L’obiettivo è monitorare l'umidità del terreno di una piantina posta all’interno della biblioteca attraverso l’utilizzo di un sensore di umidità del terreno, i dati letti da Arduino verranno mostrati in tempo reale su un display LCD e sull'app android.

Per la realizzazione di questa sezione abbiamo bisogno di:

*Arduino Uno R3*

*Modulo Bluetooth HC-06*

*Moisture Sensor v1.2 - Sensore di umidità del terreno*

*LiquidCrystal\_I2C - LCD 1602 con interfaccia I2C*

*Cavi di collegamento - Jumper wires*

*Breadboard - Per il collegamento dei componenti*

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Sezione 4

L’obiettivo di questa sezione e` monitorare gli accessi dei professori.

Ogni professore dovrà avere con sé una card RFID con un identificatore univoco, quando il professore entrerà e uscirà all'interno della biblioteca dovrà avvicinare la propria card RFID al lettore RFID.

I dati di ingresso e uscita verranno letti da Arduino, al momento dell’accesso e della lettura della carta sul display LCD verrà visualizzato il messaggio “Benvenuto/a Nome Cognome ”, al momento dell’uscita della biblioteca avvicinando la carta al lettore sul display verrà visualizzato il messaggio “Arrivederci Nome Cognome ”.

Sull’app sara` possibile visualizzare quale professore e` presente in quel momento nella biblioteca.

Per la realizzazione di questa sezione abbiamo bisogno di:

*Arduino Uno R3*

*MFRC522 - Modulo RFID*

*Alcune card RFID a 13.56MHz*

*Modulo Bluetooth HC-06*

*LiquidCrystal\_I2C - LCD 1602 con interfaccia I2C*

*Cavi di collegamento - Jumper wires*

*Breadboard - Per il collegamento dei componenti*

# Sezione 5

L’obiettivo di questa sezione e` misurare la luminosità della stanza attraverso una fotoresistenza.

I dati relativi alla luminosità saranno mostrati sul display LCD e sull’app.

Per la realizzazione di questa sezione abbiamo bisogno di:

*Arduino Uno R3*

*Modulo Bluetooth HC-06*

*Fotoresistenza (LDR) - Sensore di luminosità*

*Resistenze - Per la LDR (generalmente 10kΩ)*

*Cavi di collegamento - Jumper wires*

*Breadboard - Per il collegamento dei componenti*

Componenti necessari per il progetto:

Per svolgere ognuno di queste sezioni abbiamo bisogno dei seguenti componenti:

* Arduino Uno R3;
* DHT11 - Sensore di temperatura e umidità;
* MFRC522 - Modulo RFID;
* Alcune card RFID a 13.56MHz;
* Modulo Bluetooth HC-06;
* Moisture Sensor v1.2 - Sensore di umidità del terreno;
* Fotoresistenza (LDR) - Sensore di luminosità;
* LiquidCrystal\_I2C - LCD 1602 con interfaccia I2C;
* LED;
* Resistenza 220Ω;
* Resistenza - Per la LDR (generalmente 10kΩ);
* Cavi di collegamento - Jumper wires;
* Breadboard - Per il collegamento dei componenti.

Schema associativo componenti-sezioni:

| **Componente** | **Sezione 1** | **Sezione 2** | **Sezione 3** | **Sezione 4** | **Sezione 5** |
| --- | --- | --- | --- | --- | --- |
| Arduino Uno R3 | Necessario | Necessario | Necessario | Necessario | Necessario |
| Modulo Bluetooth HC-06 | Necessario | Necessario | Necessario | Necessario | Necessario |
| Fotoresistenza |  |  |  |  | Necessario |
| Moisture Sensor v1.2 - Sensore di umidità del terreno |  |  | Necessario |  |  |
| Jumper | Necessario | Necessario | Necessario | Necessario | Necessario |
| LiquidCrystal\_I2C - LCD 1602 con interfaccia I2C |  | Necessario | Necessario | Necessario | Necessario |
| MFRC522 - Modulo RFID |  |  |  | Necessario |  |
| Alcune card RFID a 13.56MHz |  |  |  | Necessario |  |
| Breadboard | Necessario | Necessario | Necessario | Necessario | Necessario |
| Sensore DHT 11 |  | Necessario |  |  |  |
| Resistenza 220Ω | Necessario |  |  |  |  |
| Led | Necessario |  |  |  |  |
| Resistenza 10kΩ |  |  |  |  | Necessario |

**Schema elettrico dei componenti con Arduino:**

Di seguito è descritto come collegare i vari componenti all'Arduino Uno.

DHT11:

VCC -> 5V

GND -> GND

DATA -> Pin 2

MFRC522:

SDA -> Pin 10

SCK -> Pin 13

MOSI -> Pin 11

MISO -> Pin 12

IRQ -> Non collegato

GND -> GND

RST -> Pin 9

3.3V -> 3.3V

Modulo Bluetooth (HC-06):

VCC -> 5V

GND -> GND

TXD -> Pin 11 (RX)

RXD -> Pin 12 (TX)

Moisture Sensor v1.2:

VCC -> 5V

GND -> GND

A0 -> Pin A0

Fotoresistenza (LDR):

Un lato del LDR -> 5V

Altro lato del LDR -> Pin A1 e un'estremità della resistenza da 10kΩ

Altra estremità della resistenza da 10kΩ -> GND

LCD 1602 con I2C:

SDA -> A4

SCL -> A5

VCC -> 5V

GND -> GN

LED:

Anodo (lato lungo) -> Pin 13 (attraverso una resistenza da 220Ω)

Catodo (lato corto) -> GND

**Codice Arduino in C++:**

#include <DHT.h>

#include <SoftwareSerial.h>

#include <SPI.h>

#include <MFRC522.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#define DHTPIN 2

#define DHTTYPE DHT11

#define LEDPIN 13

#define SOIL\_MOISTURE\_PIN A0

#define LDR\_PIN A1

#define SS\_PIN 10

#define RST\_PIN 9

DHT dht(DHTPIN, DHTTYPE);

MFRC522 mfrc522(SS\_PIN, RST\_PIN);

SoftwareSerial BTSerial(10, 11);

LiquidCrystal\_I2C lcd(0x27, 16, 2);

String lastCard = "";

bool isProfessorPresent = false;

unsigned long previousMillis = 0;

const long interval = 300;

int position = 0;

unsigned long currentMillis;

float hum;

float temp;

int hum\_ter\_raw;

int lumino\_raw;

int hum\_ter;

int lumino;

void setup() {

Serial.begin(9600);

BTSerial.begin(9600);

pinMode(LEDPIN, OUTPUT);

digitalWrite(LEDPIN, LOW);

dht.begin();

SPI.begin();

mfrc522.PCD\_Init();

lcd.init();

lcd.backlight();

Serial.println("Perfavore appoggia la carta al lettore...");

}

void loop() {

currentMillis = millis();

hum = dht.readHumidity();

temp = dht.readTemperature();

hum\_ter\_raw = analogRead(SOIL\_MOISTURE\_PIN);

lumino\_raw = analogRead(LDR\_PIN);

if (isnan(hum) || isnan(temp)) {

Serial.println("Lettura DHT fallita!");

return;

}

hum\_ter = map(hum\_ter\_raw, 0, 1023, 0, 100); // Converti in percentuale

lumino = map(lumino\_raw, 0, 1023, 0, 100); // Converti in percentuale

hum = map(hum, 0, 100, 0, 100); // Converti in percentuale

String line1 = "Temperatura:" + String(temp) + "C"+ "Luminosita`:" + String(lumino) + "% "; //Scritta che scorre sulla prima riga

String line2 = "Umidita` terreno:" + "% "+ String(hum\_ter) + "Umidita`:" + String(hum) + "% "; //Scritta che scorre sulla seconda riga

//Passaggio dati all'app

BTSerial.print("Temperatura: ");

BTSerial.print(temp);

BTSerial.print("Umidita`: ");

BTSerial.print(hum);

BTSerial.print("Umidita` del terreno: ");

BTSerial.print(hum\_ter);

BTSerial.print("Luminosita`: ");

BTSerial.println(lumino);

BTSerial.println("%");

if (BTSerial.available()) {

char command = BTSerial.read();

if (command == '1') {

digitalWrite(LEDPIN, HIGH);

} else if (command == '0') {

digitalWrite(LEDPIN, LOW);

}

}

if (mfrc522.PICC\_IsNewCardPresent() && mfrc522.PICC\_ReadCardSerial()) {

String cardID = "";

for (byte i = 0; i < mfrc522.uid.size; i++) {

cardID += String(mfrc522.uid.uidByte[i], HEX);

}

cardID.toUpperCase();

String professor = getProfessorName(cardID);

if (professor != "") {

lcd.clear();

if (isProfessorPresent) {

lcd.print("Arrivederci: ");

lcd.setCursor(0, 1);

lcd.print(professor);

delay(8000);

isProfessorPresent = false;

} else {

lcd.print("Benvenuto: ");

lcd.setCursor(0, 1);

lcd.print(professor);

delay(8000);

isProfessorPresent = true;

}

BTSerial.print("Professore:"); //Passaggio dati all'app

BTSerial.println(professor);

}

delay(1000); // Delay per evitare letture multiple rapide

} else {

if (currentMillis - previousMillis >= interval) {

previousMillis = currentMillis;

lcd.clear();

lcd.setCursor(0, 0);

for (int i = 0; i < 16; i++) {

int charIndex = (position + i) % line1.length();

lcd.print(line1[charIndex]);

}

lcd.setCursor(0, 1);

for (int i = 0; i < 16; i++) {

int charIndex = (position + i) % line2.length();

lcd.print(line2[charIndex]);

}

position = (position + 1) % line1.length();

}

}

}

String getProfessorName(String cardID) {

if (cardID == "YOUR\_CARD\_ID\_1") {

return "Vincenzo Rossi";

} else if (cardID == "YOUR\_CARD\_ID\_2") {

return "Paolo Bianchi";

} else if (cardID == "YOUR\_CARD\_ID\_3") {

return "Francesca Verdi";

}

return "";

}

Link per scaricare il codice: [Clicca qui](https://drive.google.com/file/d/1udDG-4bdL7RBGLl1q1FHmZiB0xYjVPNM/view?usp=sharing)

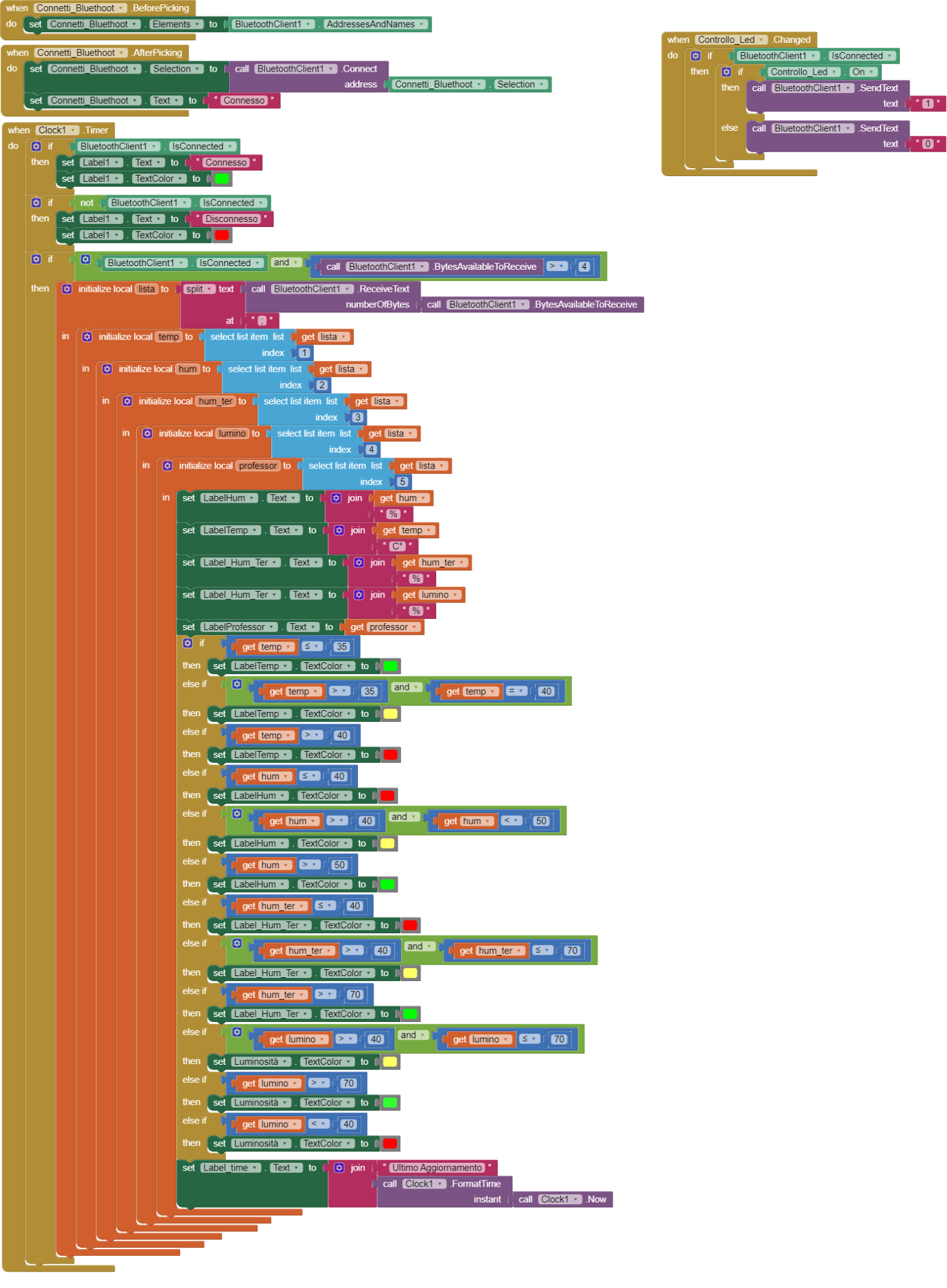
**Specifiche app:**

Per lo sviluppo dell’app abbiamo utilizzato App Inventor, che permette di creare in modo semplice, app per dispositivi Android con una semplice interfaccia.

Le funzionalità dell’app saranno:

* Connessione ad Arduino;
* Visualizzazione in tempo reale della temperatura;
* Visualizzazione in tempo reale della luminosità;
* Visualizzazione in tempo reale della umidita`;
* Visualizzazione in tempo reale della umidita` del terreno di una piantina;
* Visualizzazione in tempo reale del professore presente in biblioteca;
* Accensione e spegnimento di un led.

**Codice a blocchi dell’app da App Inventor:**



**Interfaccia App:**

