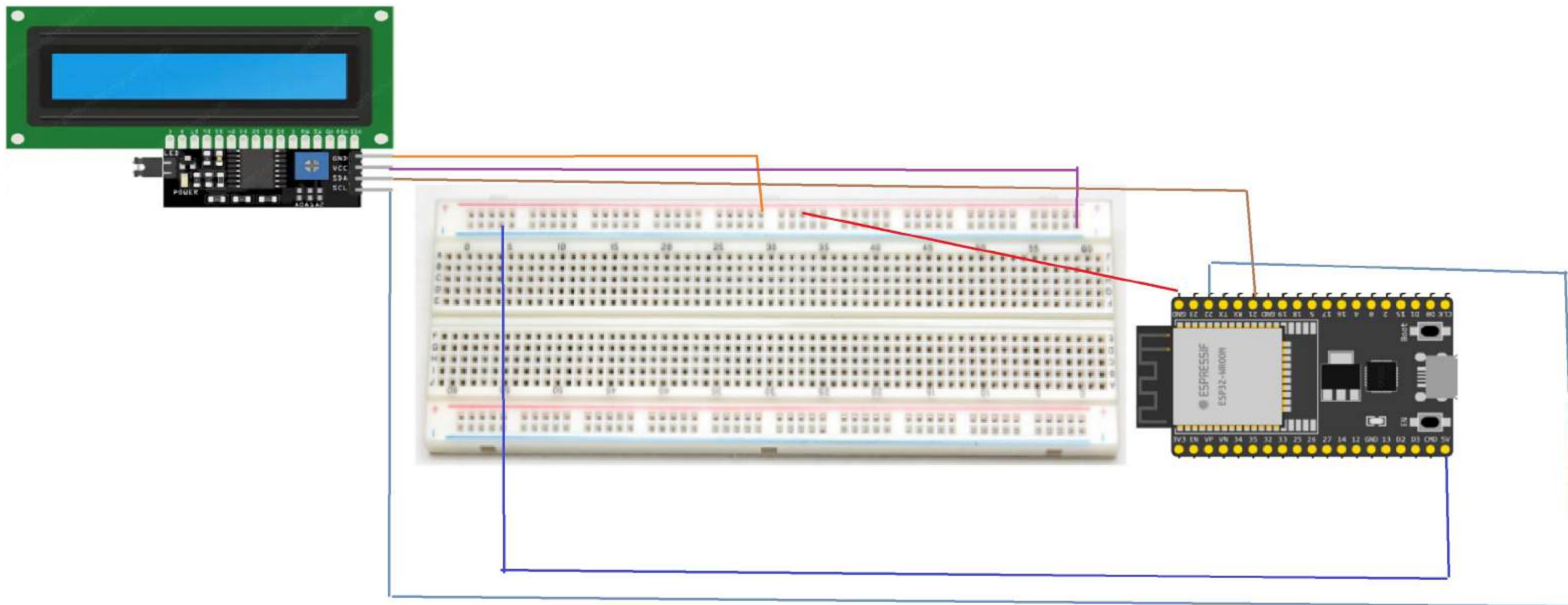


CONTROL LCD BLUETOOTH LOW ENERGY

Matei Radu-Mihai

Mumoiu Robert-Mihai

SCHEMA PROIECTULUI



ELEMENTE DE CIRCUIT – LCD 16X2

- Un LCD 16x2 este un tip de afișaj cu cristale lichide (Liquid Crystal Display) care are capacitatea de a afișa două linii de text, fiecare cu 16 caractere.
- LCD-urile folosesc cristale lichide ca element principal de afișare. Aceste cristale lichide sunt dispuse între două substraturi de sticlă sau plastic și reacționează la câmpurile electrice pentru a schimba orientarea luminii și a crea imagini sau text.
- Multe modele de LCD 16x2 vin cu o iluminare din spate (backlight), care permite afișarea informațiilor și în condiții de lumină redusă sau întuneric. Aceasta face afișajul mai ușor de citit în diverse medii de lucru.
- LCD-urile 16x2 sunt proiectate să fie ușor de utilizat cu microcontrolere sau alte dispozitive electronice. Ele sunt adesea controlate prin intermediul unor linii de date și semnale de control. ceea ce le face potrivite pentru o varietate de aplicații.



```
String input=data.c_str();

DynamicJsonDocument request(15000);

DeserializationError error = deserializeJson(request, input);

if (error) {
  Serial.print("deserializeJson() failed: ");
  Serial.println(error.c_str());
  return;
}
else{
  String action = request["action"].as<String>();
  String teamId= request["teamId"].as<String>();
  Serial.println(action);
}
```

prin variabila input se
inregistreaza datele
primate de la aplicatie

se analizeaza datele
JSON de intrare stocate
în intrarea șirului și să le
deserializeze într-un
obiect JSON

Dacă deserializarea are
succes ,codul preia
câmpuri specifice din
obiectul JSON

```

if(action=="getLCDs")
{
    StaticJsonDocument<128> doc;

    doc["type"] = "16x2";
    doc["interface"] = "Parallel 4-bit";
    doc["resolution"] = "16x2";
    doc["id"] = 1;
    doc["teamId"] = "A30";
    String output1;
    serializeJson(doc,output1);

    characteristic->setValue(output1.c_str());
    characteristic->notify();
}

if(action=="setText"){
    String text = request["text"].as<String>();
    String output2;

    StaticJsonDocument<96> doc;

    doc["id"] = 1;
    doc["text"] = "string";
    doc["teamId"] = "A30";

    serializeJson(doc, output2);
    characteristic->setValue(output2.c_str());
    characteristic->notify();
    lcd.setCursor(0, 0);
    lcd.print(text);
}
}

```

Codul verifică dacă valoarea variabilei de acțiune este egală cu „getLCDs”. Dacă condiția este îndeplinită codul generează un răspuns în format JSON. Acesta creează un StaticJsonDocument pentru a păstra datele JSON. În interiorul documentului, setează diverse attribute.



După popularea documentului JSON cu attributele necesare, codul serializează datele JSON într-un șir numit output1 .
Datele JSON sunt convertite într-un format String și stocate în variabila output1.



Se repeta procesul pentru actiunea “setText” si se afiseaza la final (pe LCD) textul transmis prin aplicatie , prin intermediul Bluetooth Low Energy

```

    if(action=="scroll"){
        String direction = request["direction"].as<String>();

        if(direction=="Left")
        {
            String output3;
            StaticJsonDocument<96> doc;

doc["id"] = 1;
doc["scrolling"] = "Left";
doc["teamId"] = "A30";

serializeJson(doc, output3);
characteristic->setValue(output3.c_str());
            characteristic->notify();
            for (int position = 0; position < 16; position++) {
                lcd.scrollDisplayLeft();
                delay(100);
            }

        }

        if(direction=="Right")
        {
            String output4;
            StaticJsonDocument<96> doc;

doc["id"] = 1;
doc["scrolling"] = "Right";
doc["teamId"] = "A30";

serializeJson(doc, output4);
characteristic->setValue(output4.c_str());
            characteristic->notify();
            for (int position = 0; position < 16; position++) {
                lcd.scrollDisplayRight();
                delay(100);
            }
        }
    }
}

```

Pentru acțiunea „scroll” codul extrage valoarea variabilei de direcție din datele JSON.



Dacă direcția este „Stânga”, codul creează un răspuns JSON care indică derularea la stânga. Datele JSON serializate sunt apoi setate ca valoare a unei caracteristici BLE și notificate dispozitivului conectat. După trimiterea notificării, codul execută o buclă pentru a derula afișajul LCD pentru 16 iterații.



Dacă direcția este „Dreapta”, codul urmează un proces similar pentru a genera un răspuns JSON care indică derularea la dreapta. Codul execută o buclă pentru a derula afișajul LCD la dreapta cu o poziție pentru 16 iterații, cu o întârziere de 100 de milisecunde între fiecare derulare.

```

    }

    if(direction=="Off")
    {
        String output5;
        StaticJsonDocument<96> doc;

        doc["id"] = 1;
        doc["scrolling"] = "Off";
        doc["teamId"] = "A30";

        serializeJson(doc, output5);
        characteristic->setValue(output5.c_str());
        characteristic->notify();
        lcd.setCursor(5, 0);
        lcd.clear();
    }
}

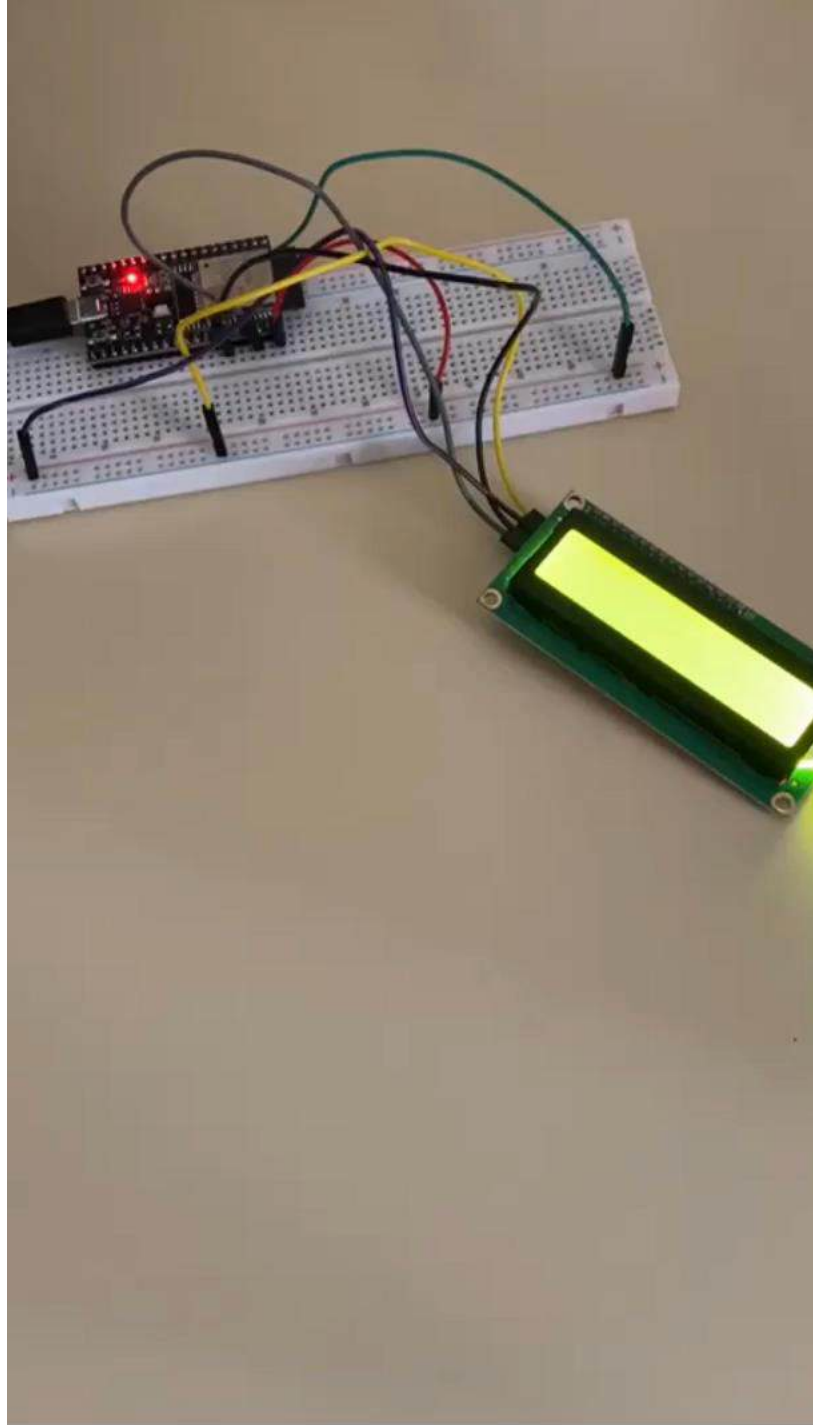
```

Dacă direcția este „Oprit”, codul creează un răspuns JSON care indică faptul că textul este dezactivat.

În interiorul documentului, setează attribute precum „id”, „scrolling” și „teamId” la valori specifice.

Datele JSON serializate sunt apoi setate ca valoare a unei caracteristici BLE (“characteristic”) și notificate dispozitivului conectat.

Apoi șterge afișajul LCD folosind `lcd.clear()`, dezactivând efectiv derularea.



BLE Device

radumatei

E0:5A:1B:5F:71:26



LCD project

Get LCDs

ca73b3ba-39f6-4ab3-91ae-186dc9577d99



MULTUMIM PENTRU
ATENTIA ACORDATA!