Safe Crack

Introducere

- Din telefon se seteaza un cod de 3 cifre folosind Blynk Cloud. Acesta trebuie ghicit invartind 3 potentiometre, fiecare apartinand unei cifre. Pentru a putea schimba numerele folosind potentiometrele, trebuie plasata o sursa de lumina asupra unui fotorezistor. Cu cat este mai aproape de a ghici numarul, un led rgb isi va schimba culoarea de la rece la cald. Cand este ghicit numarul, un servomotor se va roti si va deschide cutia. De asemenea led-ul se va schimba in verde. Pe un lcd se va afisa constant numarul actual.
- Inspiratia proiectului a fost o image de genul: https://media.gettyimages.com/id/88295292/photo/burglar-cracking-safe.jpg?s=612x612&w=gi&k=20&c=j6YNp-2L02fGJ1xSpV-2O3-M2Mxs4jXVT1vmUiYcgBE=, plus dorinta de a face un proiect care se conecteaza cu telefonul.

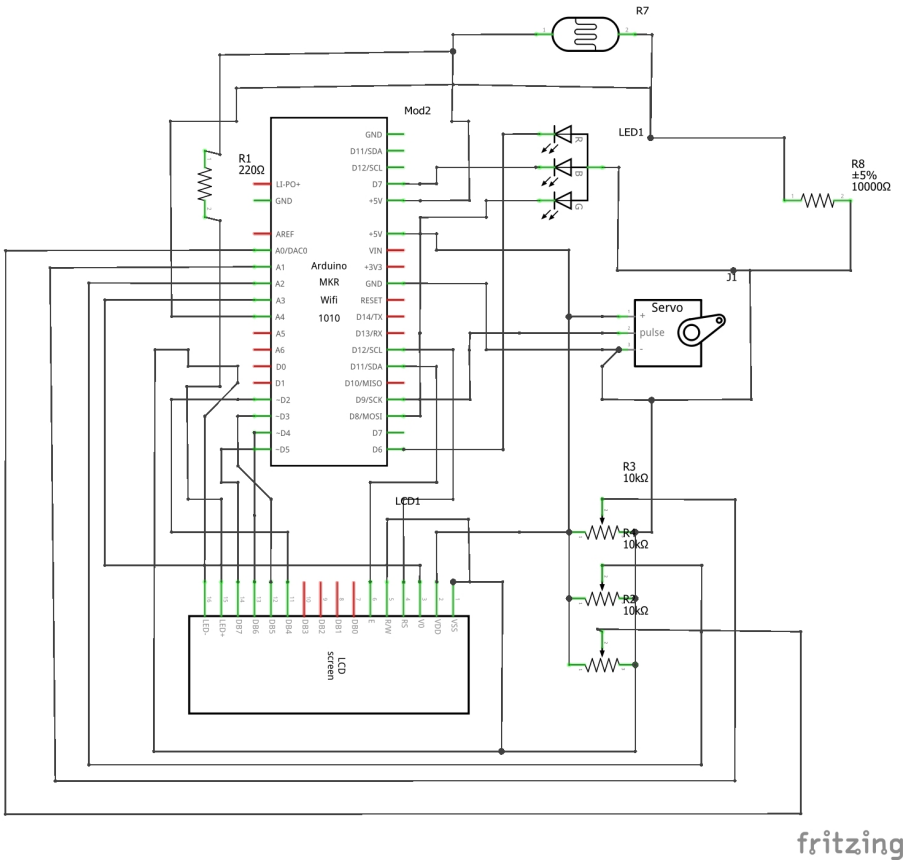
Descriere generală

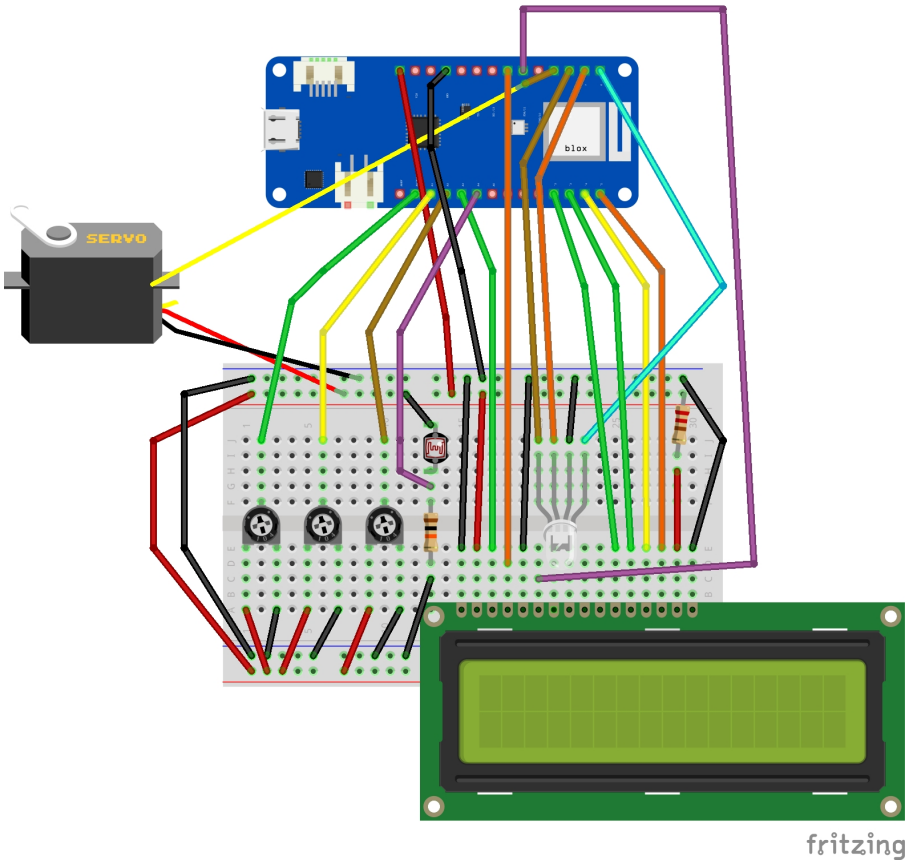
Diagrama proiect:

Hardware Design

- piese: Arduino MKR Wifi 1010, 3 potentiometre 10k, rezistor 220ohm, rezistor 10kohm, lcd 1602, servomotor, fotorezistor
- schema electrica board: https://docs.arduino.cc/resources/schematics/ABX00023-schematics.pdf
- pinout board: https://docs.arduino.cc/resources/pinouts/ABX00053-full-pinout.pdf

Schematic proiect:





Voi atasa si in capitolul de rezultate obtinute o demonstratie initiala a functionalitatii proiectului.

Update 20.05.2024: am adaugat si un modul I2C lcd-ului pentru usurinta in asamblarea finala, dar si un buzzer conectat la pin-ul 1 ce va canta cand este ghicit codul

Software Design

- Am lucrat in Arduino IDE. Pentru surse si librarii am folosit WifiNina (modulul wifi al boardului) si BlynkSimpleWifiNINA (pentru conectarea cu cloud), LiquidCrystal (pentru lcd), Servo si SPI.

Declaram bibliotecile:

```
#include <SPI.h>
#include <WiFiNINA.h>
#include <BlynkSimpleWiFiNINA.h>
#include <Servo.h>
#include <LiquidCrystal_I2C.h>
```

Declaram informatiile de autentificare in Blynk:

```
#define BLYNK_TEMPLATE_ID "<id>"
#define BLYNK_TEMPLATE_NAME "<template_name>"
#define BLYNK_AUTH_TOKEN "<auth_token>"
```

si WiFi credentials pentru conectare la internet si apoi la cloud:

```
char ssid[] = "<ssid Wifi";
char pass[] = "<parola Wifi>";
```

Definim functiile ce primesc valoarea slider-elor pentru setarea codului. V1, V4, V5 sunt pini virtuali setati din Blynk pentru diferentiarea lor

```
// extragerea valorilor
BLYNK_WRITE(V1) {
        SliderValueOne = param.asInt();
}
BLYNK_WRITE(V4) {
        SliderValueTwo = param.asInt();
}
BLYNK_WRITE(V5) {
        SliderValueThree = param.asInt();
}
```

Realizam setup-ul:

```
LiquidCrystal_I2C lcd(0x3f, 16, 2);
void setup()
{

  pinMode(buzzerPin, OUTPUT); //pin buzzer
```

```
  //initalizare pini led rgb
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);

  lcd.init();
  lcd.backlight();

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  myservo.attach(9);  //pin servo
  myservo.write(pos);

}
```

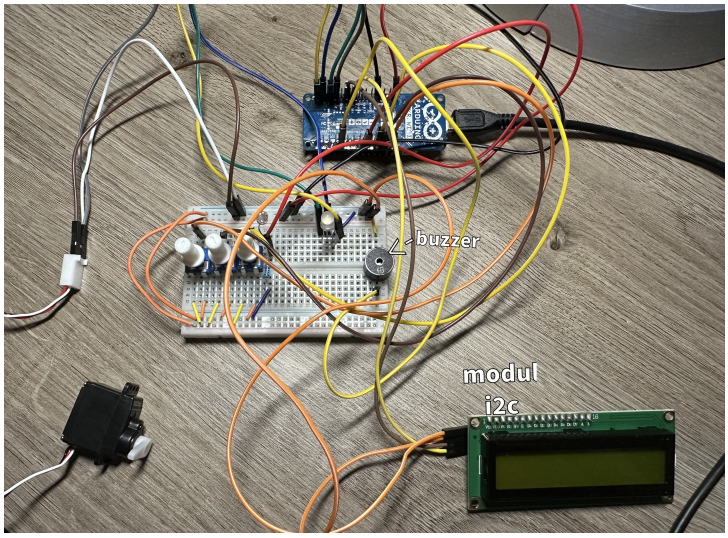
In loop se vor intampla urmatoarele:

- citim constant valoarea intentistitatii luminii → `int lightLevel = analogRead(photoPin)`
- doar daca lightLevel-ul > threshold putem permite codului sa primeasca si urmatoarele valori necesare
- citim acum valorile primite de potentiometre. Ele in mod normal citesc valori pana la 1023, dar noi le vom mapa doar pana la 9: `int PotOne = map(analogRead(A0), 0, 1023, 0, 9)`, urmand apoi sa printam constant pe lcd valorile acestora
- cat timp nu a fost ghicit codul, vom urmari cat de aproape suntem de acesta pentru a afisa o culoare de la rece la cald folosind led-ul rgb
- dupa ce a fost gasit, se vor intampla urmatoarele lucruri:
  - buzzer-ul va canta o melodie
  - servomotorul se va roti pentru a deschide cutia
  - led-ul va schimba si va clipi verde
- pentru a inchide la loc cutia, tot ce trebuie sa facem este sa mutam potentiometrele pe 0 0 0

Rezultate Obţinute

Demonstratie initiala functionalitate proiect - 12.05.2024: Safe Crack

Update: adaugat modul i2c si buzzer:



Download

O arhivă (sau mai multe dacă este cazul) cu fişierele obţinute în urma realizării proiectului: surse, scheme, etc. Un fişier README, un ChangeLog, un script de compilare şi copiere automată pe uC crează întotdeauna o impresie bună 😉.

Fişierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fişierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Bibliografie/Resurse

- https://docs.arduino.cc/resources/pinouts/ABX00023-full-pinout.pdf
- https://docs.arduino.cc/tutorials/iot-bundle/puzzlebox
- https://blynk.io/getting-started

Export to PDF