Universitatea Politehnica Timișoara
Facultatea de Automatică și Calculatoare
Calculatoare și Tehnologia Informației

Sisteme Incorporate (S.I.)

Sistem Antifurt

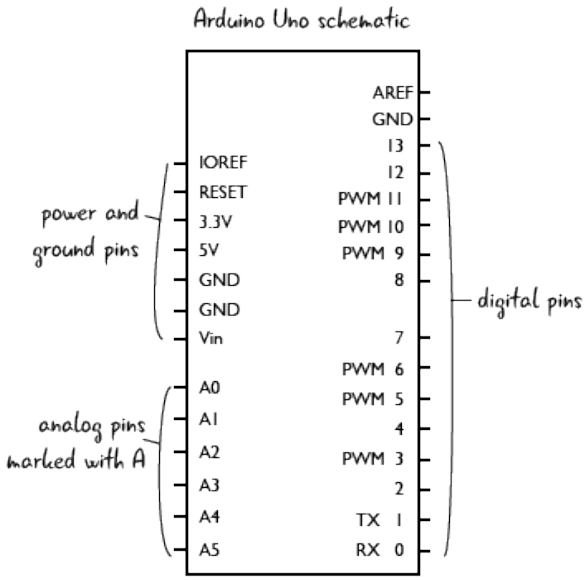
Fulga Fabian-Marian
Velciov Ana-Diana
Secosan Catalin-Luca
Anul III CTI-RO
Anul Universitar 2021-2022

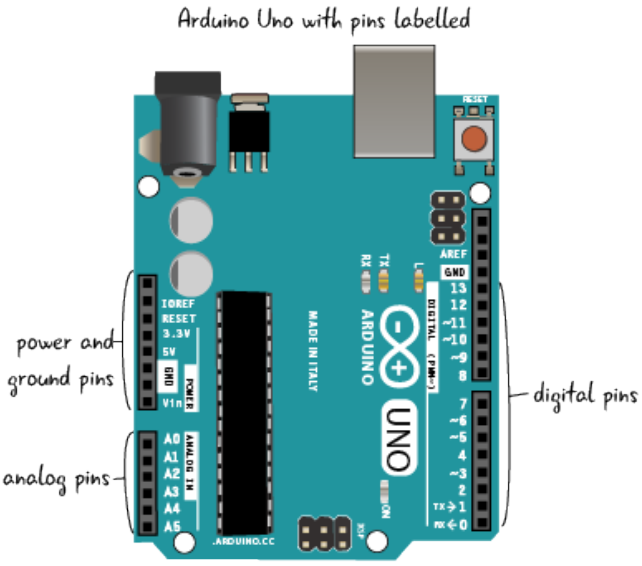
Explicarea funcționalităților sistemului

Scopul principal al sistemului este detectarea automata a proprietarilor si deschiderea automata a usii odata ce le-a fost confirmata identitatea. Pentru acestea, se va folosi un senzor de amprenta si o camera video, pentru detectia datelor biometrice ale persoanei. De asemenea, se va folosi un modul de comunicare wifi pentru a trimite un mesaj pe aplicatie de fiecare data cand cineva incearca sa deschida usa.

Descrierea plăcii Arduino UNO

Arduino este o platformă cu microcontroler foarte simplă de utilizat. Arduino poate fi folosit pentru dezvoltarea de aplicaţii interactive. Practic, informaţia este preluată de la o gamă variată de elemente de intrare (senzori şi comutatoare), se procesează în interiorul microcontrolerului şi este transmisă către o gamă la fel de variată de elemente de ieşire: leduri, motoare, actuatoare, etc.

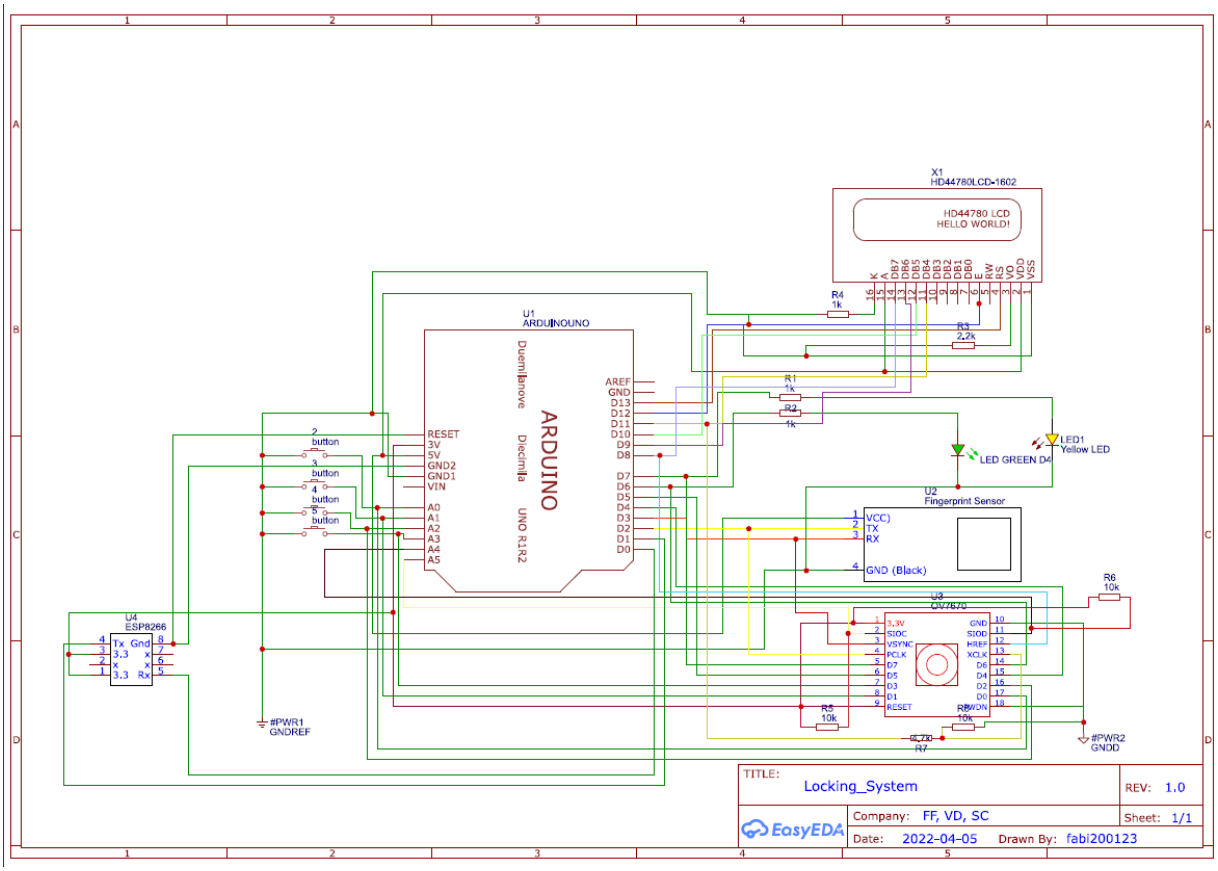




O platformă Arduino este compusă dintr-un microcontroler Atmel AVR de 8, 16 sau 32 biţi în special ATmega8, ATmega168, ATmega328, ATmega1280 şi ATmega2560 precum şi componente complementare care facilitează programarea şi încorporarea în alte circuite. Din 2015 platformele Arduino conţin microcontrolere de la alţi producători. Arduino Uno este o placa de dezvoltare bazată pe microcontrolerul ATmega328P, având 6 intrări analogice, 14 intrări digitale/pini de ieşire (din care 6 pot fi utilizate ca ieşiri PWM), un oscilator cu quart de 20 MHz, o conexiune USB, o mufa de alimentare, şi un buton de resetare.

Conectarea Hardware

Schema realizata in EasyEDA:



Placa Arduino UNO R3 (ATmega328p)

Senzor Amprentă

Ecran LCD 1602 IIC/I2C

Modul camera OV7670

Modul Wifi ESP8266 Transreceiver, ESP-01

Descrierea modulelor microcontrolerului

1. Senzor Amprenta

Senzorul de amprenta functioneaza la un voltaj de 3.6 - 6.0 V, acesta avand nevoie de o durata de timp de sub 1 secunda pentru a citi amprenta. Acesta poate retine in memorie pana la 162 de amprente si are o interfata TTL seriala. Functioneaza la un Baud Rate de 9600, 19200, 28800, 38400, 57600 (default fiind 57600). Conexiunea acestuia la placa se realizeaza prin 4 conectori, comunicand printr-o interfata UART:

- Rosu -> 5V
- Negru -> GND
- Galben -> D2
- Verde -> D3

2. Ecran LCD 1602 IIC/I2C

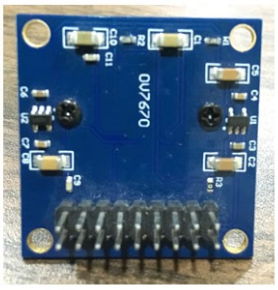
Ecranul LCD IIC/I2C are un contrast ajustabil si reprezinta un mare avantaj in conexiunea sa intr-un proiect unde exista o multime de conexiuni la microcontroller, datorita adaptorului pentru interfata I2C ce are nevoie doar de 4 conexiuni (SDA/SCL, GND, VCC). Acest LCD functioneaza la un voltaj de 4.5 – 5.5 V, iar afisajul este de forma 16 x 2 (16 coloane, 2 linii), unde fiecare element reprezinta un caracter. Culoarea acestora este alba, iar backlightul este albastru.

Conexiunea pinilor LCD-ului la Arduino este urmatoarea:

- SDA -> A4
- SCL -> A5
- GND -> GND
- VCC -> 5V

3. Modul camera OV7670

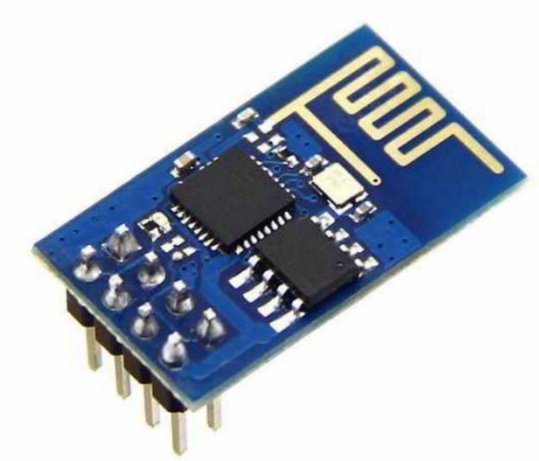
Modulul de camera utilizat este OV7670. Aceasta camera VGA necesita o tensiune de alimentare de 3.3V si o tensiune in pini de 2.5 – 3V. Ea are un numar de 18 pini si are comunicarea de tip I2C. Rezolutia maxima a imaginii transmise este de 640x480, insa in proiectul nostru o sa folosim o rezolutie de 320x240. Camera are un numar de 18 pini, iar conexiunea la microcontroller este urmatoarea:

- VCC -> 3.3V
- GND -> GND
- SIOC -> A5
- SIOD -> A4
- XCLK -> //system clock input
- PCLK -> //Pixel Clock Output -> Data is valid on rising edge
- HREF -> //Horizontal reference -> high during active pixels of row
- VSYN -> //Vertical Sync Output -> low during frame
- PWDN -> GND // Power Down Mode Selection
- RESET -> 3.3V // Semnalul de Reset
- D0 – D3 -> A0 - A3
- D4 – D7 -> D4 - D7

4. Modul Wifi ESP8266 Transreceiver, ESP-01

Modulul Wifi ESP8266 este un SOC (System on a Chip) cu protocol TCP/IP integrat ce ii poate da access la reteaua WiFi oricarui microcontroller. ESP8266 este capabil de hosting pentru o aplicatie sau sa preia toate functiile de retea Wi-Fi de la un alt procesor de aplicatie. Fiecare modul vine pre-programat, cu firmware cu comenzi AT. Acest modul comunica printr-o interfata UART, iar conectarea acestuia s-a realizat astfel:



ESP8266 Wi-Fi Module

- TX -> D8
- GND -> GND
- RX -> D9
- 3V3 -> 3.3V
- EN -> 3.3V

Descrierea codului

Functionarea sistemului are la baza un cod ce este incarcat pe placa Arduino pentru a porni sistemul de deschidere al usii. Acest cod este compus din mai multe functii care verifica amprenta citita si initializeaza modulul wifi ESP8266. De asemenea, pentru testarea si verificarea utilizarii corecte a sistemului pe Serial Monitor o sa fie afisate pe rand initializarea senzorului de amprenta si a modulului wifi odata ce le este cerut sa intre in executie. Atat modulul wifi cat si senzorul de amprenta sunt conectate in mod serial, deci pentru a le putea utiliza corespunzator o sa le activam pe rand la momentul in care le este ceruta executia.

```
#include<SoftwareSerial.h>
#include <LiquidCrystal_I2C.h>
SoftwareSerial comm(8, 9); //setting Tx and Rx pins
```

```
#include <Adafruit_Fingerprint.h>
//fingerprint sensor
SoftwareSerial mySerial(13, 11);

//lcd
LiquidCrystal_I2C lcd(0x27, 16, 2);

const byte green = 10; //btn

int redLed = A0; //led
int greenLed = A1; //led
String stringToSend="";
String nume;

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
//fingerprint sensor

String server = ""; //variable for sending data to webpage
boolean No_IP = false; //variable to check for ip Address
String IP = ""; //variable to store ip Address
void setup()
{
  //lcd init
  lcd.init();
  lcd.backlight();
  lcd.clear();

  //leds + buttons
  pinMode(green, INPUT);
  pinMode(redLed, OUTPUT);
```

```
  pinMode(greenLed, OUTPUT);
  digitalWrite(redLed, LOW);
  digitalWrite(greenLed, LOW);
  Serial.begin(9600);

  //fingerprint sensor
  while (!Serial);  // For Yun/Leo/Micro/Zero/...
  delay(100);
  Serial.println("\n\nAdafruit finger detect test");

  // set the data rate for the sensor serial port
  finger.begin(57600);
  delay(5);
  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
  }

  Serial.println(F("Reading sensor parameters"));
  finger.getParameters();
  Serial.print(F("Status: 0x"));
Serial.println(finger.status_reg, HEX);
  Serial.print(F("Sys ID: 0x"));
Serial.println(finger.system_id, HEX);
  Serial.print(F("Capacity: "));
Serial.println(finger.capacity);
  Serial.print(F("Security level: "));
Serial.println(finger.security_level);
  Serial.print(F("Device address: "));
Serial.println(finger.device_addr, HEX);
```

```
  Serial.print(F("Packet len: "));
Serial.println(finger.packet_len);
  Serial.print(F("Baud rate: "));
Serial.println(finger.baud_rate);


  finger.getTemplateCount();


  if (finger.templateCount == 0) {
    Serial.print("Sensor doesn't contain any fingerprint data.
Please run the 'enroll' example.");
  }
  else {
    Serial.println("Waiting for valid finger...");
      Serial.print("Sensor contains ");
Serial.print(finger.templateCount); Serial.println("
templates");
  }
  comm.begin(115200);
  wifi_init();
  Serial.println("System Ready..");
}


void loop()
{
    while (comm.available())
    {
      if (comm.find("0,CONNECT"))
      {
        Serial.println("Starting");
        sendToServer();
        Serial.println("Finished");
        delay(1000);
```

```
    }
  }

if(digitalRead(green) == LOW){
  finger.begin(57600);
  digitalWrite(greenLed, LOW);
  digitalWrite(redLed, LOW);
  lcd.clear();
  lcd.print("CHECKING");
  lcd.setCursor(4,1);
  lcd.print("FINGERPRINT");
  for(int i=0; i<50; i++){
    int b;
    b = getFingerprintID();
    if(b == 2) nume = "Fabian";
    if(!b){
      digitalWrite(greenLed, HIGH);
      digitalWrite(redLed, LOW);
      lcd.clear();
      lcd.print(nume);
      lcd.setCursor(4,1);
      lcd.print("ENTER!");
      stringToSend += "<h2>" + nume +  " entered!</h2>";
      comm.begin(115200);
      wifi_reinit();
      digitalWrite(greenLed, LOW);
      lcd.clear();
      return;
    }
  }
```

```
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, LOW);
    lcd.clear();
    lcd.print("FINGERPRINT");
    lcd.setCursor(4,1);
    lcd.print("NOT OKAY!");
    stringToSend += "<h2>Attempt failed!</h2>";
    delay(1550);
    digitalWrite(redLed, LOW);
    comm.begin(115200);
    wifi_reinit();
    lcd.clear();
  }
}



void findIp(int time1) //check for the availability of IP
Address
{
  int time2 = millis();
  while (time2 + time1 > millis())
  {
    while (comm.available() > 0)
    {
      if (comm.find("IP has been read"))
      {
        No_IP = true;
      }
    }
  }
```

```
}

void showIP()//Display the IP Address
{
  IP = "";
  char ch = 0;
  while (1)
  {
    comm.println("AT+CIFSR");
    while (comm.available() > 0)
    {
      if (comm.find("STAIP,"))
      {
        delay(1000);
        Serial.print("IP Address:");
        while (comm.available() > 0)
        {
          ch = comm.read();
          if (ch == '+')
            break;
          IP += ch;
        }
      }
      if (ch == '+')
        break;
    }
    if (ch == '+')
      break;
    delay(1000);
  }
```

```
  Serial.print(IP);
  Serial.print("Port:");
  Serial.println(80);
}


void establishConnection(String command, int timeOut) //Define
the process for sending AT commands to module
{
  int q = 0;
  while (1)
  {
    Serial.println(command);
    comm.println(command);
    while (comm.available())
    {
      if (comm.find("OK"))
        q = 8;
    }
    delay(timeOut);
    if (q > 5)
      break;
    q++;
  }
  if (q == 8)
    Serial.println("OK");
  else
    Serial.println("Error");
}


void wifi_reinit(){
```

```
  establishConnection("AT+CWJAP=\"Orange-ad53\",\"AFNHr8uc\"",
7000); //provide your WiFi username and password here


  establishConnection("AT+CIPMUX=1", 100);
  establishConnection("AT+CIPSERVER=1,80", 100);
}


void wifi_init() //send AT commands to module
{
  establishConnection("AT", 100);
  delay(1000);
  establishConnection("AT+CWMODE=3", 100);
  delay(1000);
  establishConnection("AT+CWQAP", 100);
  delay(1000);
  establishConnection("AT+RST", 5000);
  delay(1000);
  findIp(5000);
  if (!No_IP)
  {
    Serial.println("Connecting Wifi....");
    establishConnection("AT+CWJAP=\"Orange-
ad53\",\"AFNHr8uc\"", 7000); //provide your WiFi username and
password here
  }
  else
  {
  }
  Serial.println("Wifi Connected");
  showIP();
  establishConnection("AT+CIPMUX=1", 100);
```

```
  establishConnection("AT+CIPSERVER=1,80", 100);
}

void sendData(String server1)//send data to module
{
  int p = 0;
  while (1)
  {
    unsigned int l = server1.length();
    Serial.print("AT+CIPSEND=0,");
    comm.print("AT+CIPSEND=0,");
    Serial.println(l + 2);
    comm.println(l + 2);
    delay(100);
    Serial.println(server1);
    comm.println(server1);
    while (comm.available())
    {
      //Serial.print(Serial.read());
      if (comm.find("OK"))
      {
        p = 11;
        break;
      }
    }
    if (p == 11)
      break;
    delay(100);
  }
}
```

```
void sendToServer()//send data to webpage
{
  server = "<h1>Lock System!</h1>";
  sendData(server);
  sendData("<h2>Here you can see who entered:</h2>");
  sendData(stringToSend);
  delay(1000);
  comm.println("AT+CIPCLOSE=0");
}

//fingerprint sensor function to check fingerprint
uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("No finger detected");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      return p;
    default:
      Serial.println("Unknown error");
      return p;
```

```
}

// OK success!

p = finger.image2Tz();
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}
// OK converted!
p = finger.fingerSearch();
if (p == FINGERPRINT_OK) {
  Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
```

```
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }
  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of ");
Serial.println(finger.confidence);


  //return finger.fingerID;
  return 0;
}
// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;
  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)  return -1;
  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;
  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of ");
Serial.println(finger.confidence);
  return finger.fingerID;
}
```

Bibliografie

https://create.arduino.cc/projecthub/PatelDarshil/how-to-communicate-with-esp8266-via-arduino-uno-f6e92f

https://www.electronicwings.com/arduino/esp8266-wifi-module-interfacing-with-arduino-uno

https://circuitjournal.com/arduino-OV7670-to-pc

https://www.makerguides.com/character-i2c-lcd-arduino-tutorial/

https://maker.pro/arduino/tutorial/how-to-interface-the-ov7670-camera-module-with-arduino

https://www.teachmemicro.com/detect-objects-camera-arduino/

https://electronicsworkshops.com/2020/09/12/fingerprint-door-based-lock-system-arduino/

https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library

https://circuitdigest.com/microcontroller-projects/how-to-use-ov7670-camera-module-with-arduino

https://www.instructables.com/OV7670-Arduino-Camera-Sensor-Module-Framecapture-T/