

# **Smart Liquid Dispenser**

**Autori: Nemeth Edgar  
Schipala Ianko**

**UNIVERSITATEA POLITEHNICA TIMIȘOARA**

**Facultatea de Automatică și Calculatoare**

## 2. Enunț

Sistem de dozare a unui lichid ”**Smart Liquid Dispenser**”, implementat cu 2 plăcuțe Arduino

- Caracteristici:
  - Dozează lichidul (apă, cafea, ceai etc.), folosind o pompă sumersibilă
  - Utilizatorul trebuie să se autentifice la cititorul de card RFID
  - Utilizatorul introduce volumul de lichid dorit cu ajutorul unei tastaturi 4x4
  - Prind intermediul unui senzor de proximitate (cu infraroșu) se verifică dacă există un recipient (ex. pahar) în care să fie introdus lichidul, în caz contrar, apare pe display (LCD 2004 + interfața I2C) un mesaj de eroare
  - Măsoara lichidul din tanc cu ajutorul unui senzor de ultrasunete
  - Folosirea unor LED-uri pentru a anunța autentificarea admin-ului
  - Prin intermediul unui buzzer, apăsarea butoanelor din tastatură emit un anumit sunet
  - Folosind un modul tranzistor de putere IRF520, controlăm pompa, facilitând conectarea la Arduino

Posibil scenariu de utilizare:

Studentului Tedi i se face sete, așa că se apropie de Dozatorul de apă. Pe display îi apare mesajul ”Scan a card, please!” și un text care îi indică procentajul apei din rezervor, așa că scanează cardul de student UPT. Spre surprinderea lui, îi apare mesajul ”Access denied!”, prin urmare cheama Admin-ul să îl înregistreze.

Admin-ului i se afișează un meniu, având posibilitatea următoarelor acțiuni: ștergerea tuturor utilizatorilor înregistrați, adăugarea/ștergerea unui utilizator, ieșirea din modul de Admin. Tedi apropie cardul de cititor și este înregistrat, introducând acum cu ajutorul tastaturii volumul de apă dorit (ex. 250 ml).

În ziua următoare, Tedi este din nou însetat. Merge lângă dozator și scanează cardul de student UPT. De aceasta dată, nu mai trebuie să contacteze Admin-ul pentru că este deja înregistrat. Astfel se poate bucura de câtă apă dorește!

### 3. Descrierea plăcii de dezvoltare

Am ales 2 plăci Arduino Uno pentru realizarea proiectului.

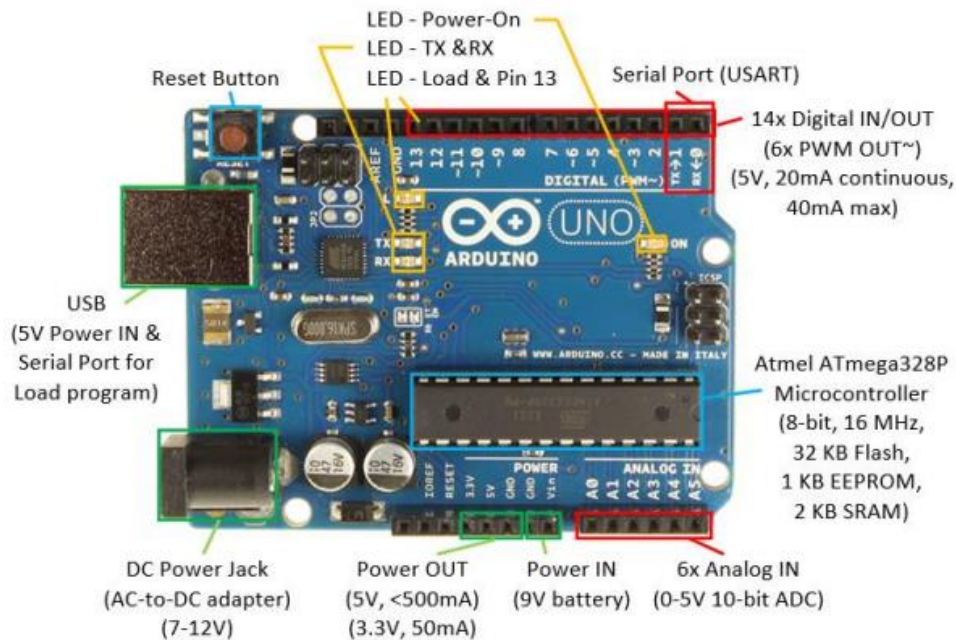
- Caracteristici Arduino Uno:

- Tip open-source.
- Interfață ușoară USB. Cipul de pe placă se conectează direct în portul USB și se înregistrează pe computer ca un port serial virtual.
- Comandă foarte convenabilă a alimentării și reglarea tensiunii încorporată. Se poate conecta o sursă externă de alimentare de până la 12V și va regla atât la 5V cât și la 3,3V. Poate fi alimentată direct de pe un port USB fără nici o putere externă.
- Un creier microcontroler ATmega328
- Un ceas de 16 MHz. Acest lucru nu-l face cel mai rapid microcontroler din jur, dar suficient de rapid pentru majoritatea aplicațiilor.
- 32 KB de memorie flash pentru stocarea codului.
- 13 pini digitali și 6 pini analogi
- Un LED pe placă atașat la pinul digital 13 pentru o depanare rapidă și ușoară a codului.
- Un buton pentru resetarea programului pe cip.

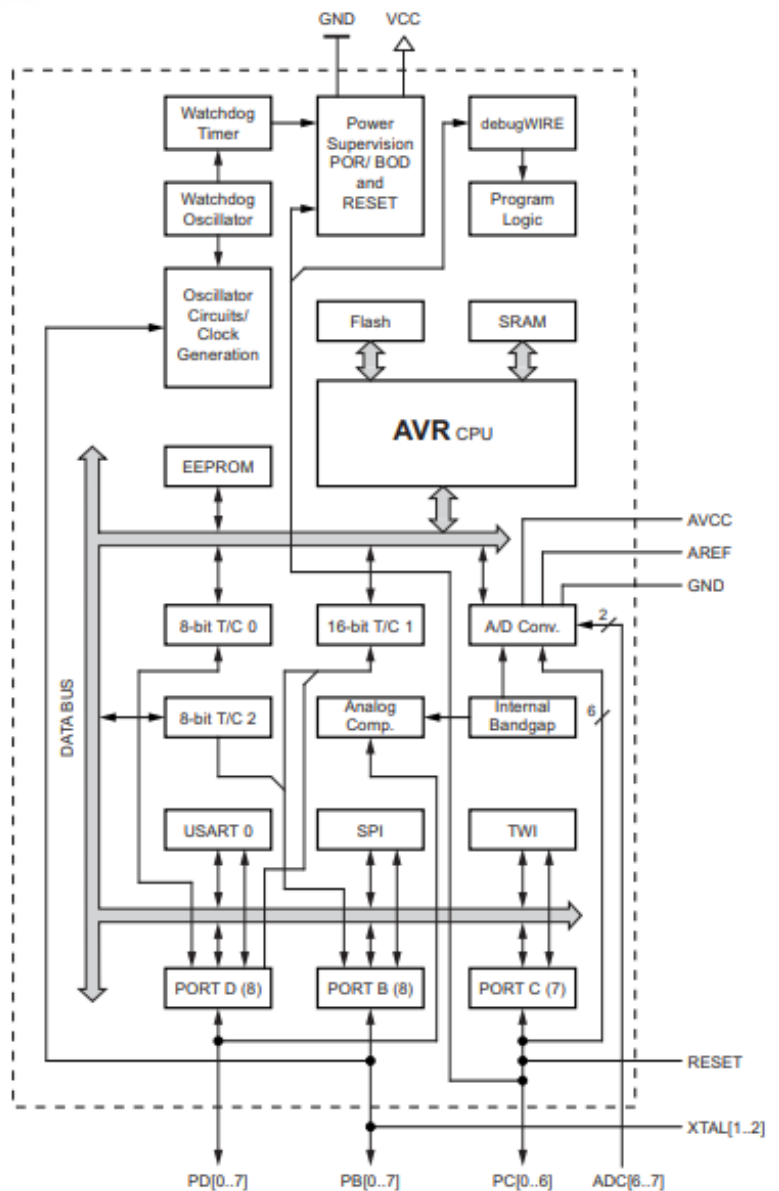
- Caracteristici ATmega328:

- Tip memorie: Flash
- Capacitate memorie EEPROM: 1kB
- Capacitate memorie SRAM: 2kB
- Capacitate memorie Flash: 32kB
- Canale PWM: 6
- Număr timere 8bit: 2
- Număr timere 16bit: 1
- Montare: THT
- Temperatura de lucru: -40°C - 85°C
- Frecvență funcționare: 20MHz
- Tensiune alimentare: 1.8V - 5.5V
- Număr de întreruperi externe: 24
- Numărul canalelor de ieșire: 6
- Număr comparatoare: 1
- Interață: I2C, SPI x2, UART, debugWIRE
- Greutate: 2.16 g

- Schema Arduino Uno



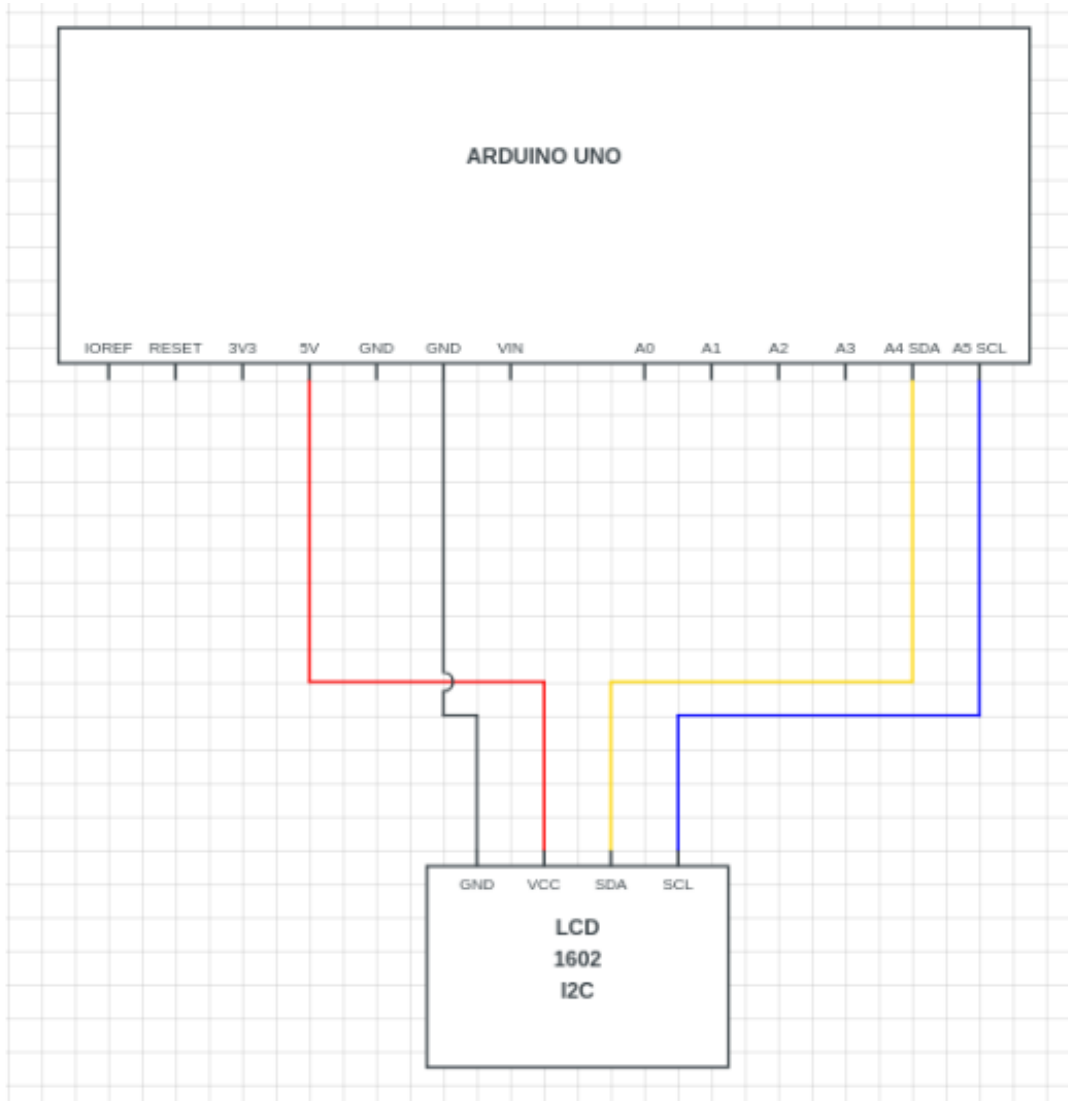
- Schema microcontroller ATmega328:



#### 4. Descrierea detaliată a modulelor microcontrolerului folosite

##### ➤ Modul I<sup>2</sup>C pentru LCD 2004

Modulul reprezintă un adaptor ce se montează direct pe ecranul LCD. Comunicația I2C reprezintă un avantaj deoarece avem nevoie de doar două fire pentru a comunica cu plăcuța de dezvoltare Arduino. Cele două fire sunt necesare pentru clock și pentru date. Modulul conține potențiometrul pentru a regla contrastul.



SDA reprezintă linia de date iar SCL cea de clock.

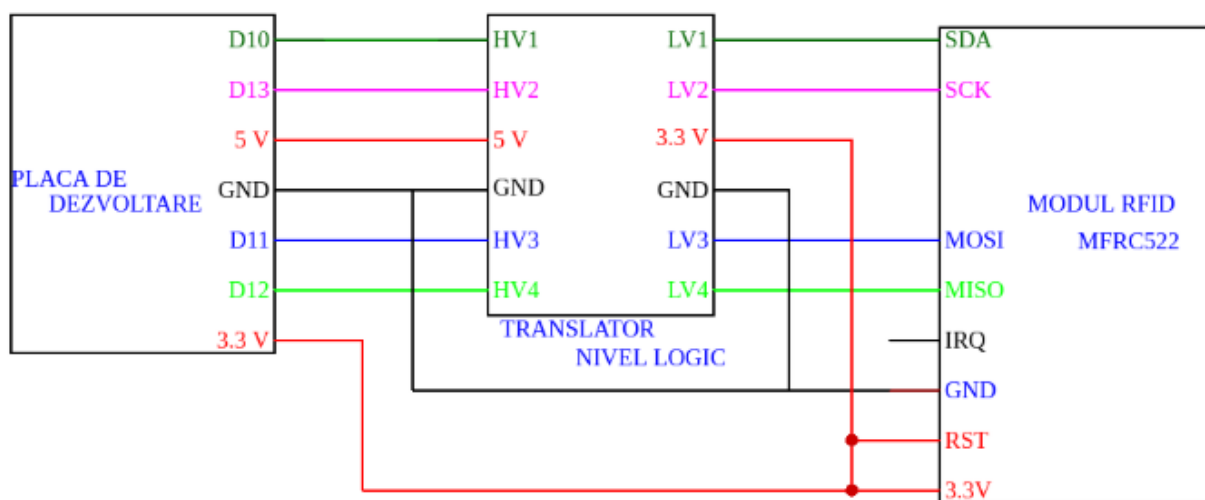
1602 I2C module	Arduino Uno
VCC	5V
GND	GND
SDA	A4
SCL	A5

Modulul poate deci transforma un afișor 16x2 caractere, bazat pe o comunicație nativă paralelă 8-bit, într-un afișor serial controlabil I<sup>2</sup>C prin intermediul a doar 2 fire de semnal/date. Adaptorul utilizează integratul PCF8574 care servește ca expander I/O în comunicarea cu Arduino sau orice alt controler ce utilizează protocolul I<sup>2</sup>C.

#### ➤ Modul RFID RC522

Modul RFID bazat pe circuitul integrat RC522, un circuit specializat pentru scrierea și citirea cardurilor de 13.56 MHz.

Modulul este utilizat pentru a scrie sau a citi date de pe tag-uri suportate ( S50, S70, UltraLight, Pro și Desfire ) prin intermediul comunicației radio (RFID). Acest lucru se face cu ajutorul circuitului integrat RC522. Proiectele de electronică în care poate fi inclus, de obicei, sunt cele care necesită identificarea persoanelor de acces, cum ar fi un interfon. Gestionarea datelor colectate de modul este realizată de către o placă de dezvoltare, cea mai populară fiind Arduino Uno.



- Caracteristici tehnice:
  - Tensiune de alimentare: 3.3 V;
  - Curent idle: 10 - 13 mA;
  - Curent de sleep: 80 uA;
  - Curent maxim: 30 mA;
  - Frecvență de funcționare: 13.56MHz.
  - Carduri suportate: S50, S70, UltraLight, Pro și Desfire.
  - Versiune firmware: 0x12;
  - Dimensiune circuit: 40 x 60 mm;
  - Protocole de comunicare:
    - RS232 Serial UART : până la 1228.8 k Bd;

- SPI: până la 10 MBit/s;
- I2C: până la 400 kBd în Fast Mode și până la 3400 kBd în High-Speed Mode.
- Buffer FIFO;
- Moduri flexibile de întrerupere;
- Temperatură optimă de funcționare: -25 °C - +85 °C;

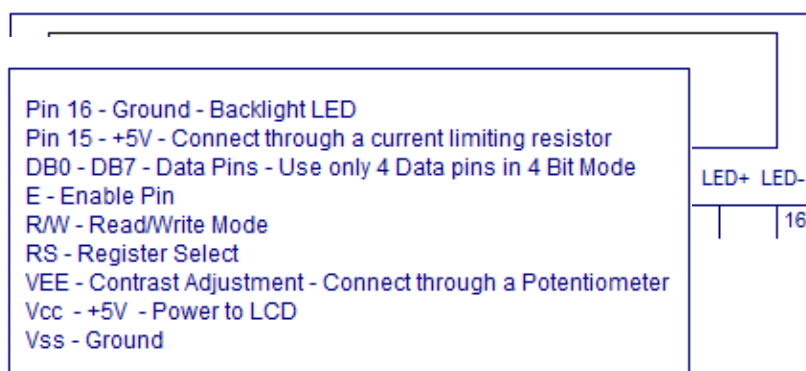
## 5. Sistemului de afișaj

Am ales un display LCD 2004 pentru a afișa diferitele mesaje.

- Caracteristici:

Tip afișaj	alfanumeric
Tensiune de lucru	5 V
Numar pini	16
Dimensiune semne [mm]	2,95 x 4,75
Dimensiune fereastră [mm]	84x31
Dimensiuni modul [mm]	98x60x13
Interfata I2C	Da
Culoare lumina fundal	Albastru
Numar semne	4x20

- Diagrama și descrierea pinilor



## 6. Senzori

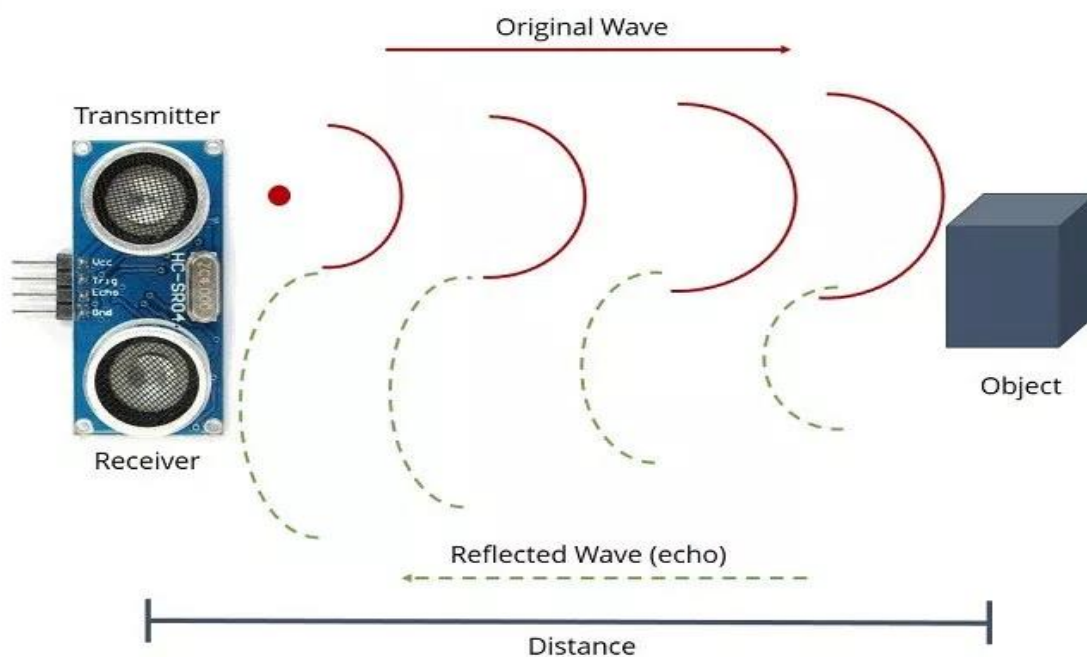
### ➤ Senzor ultrasonic HC-SR04

Senzorul HC-SR04 utilizează un sonar pentru a determina distanța față de un obiect, ca și liliicii. Oferă o detectare excelentă a gamei fără contact, de la 2 cm la 400 cm, cu o acuratețe ridicată. Funcționarea acestuia nu este afectată de lumina soarelui sau de materialul negru, cum ar fi telemetrele (deși material acustic moale, cum ar fi stofa, poate fi dificil de detectat).

- Caracteristici:
  - Alimentare: + 5V DC
  - Curent static: < 2 mA
  - Curent de lucru: 15 mA
  - Unghiul efectiv: <15°
  - Distanță: 2 cm - 400 cm
  - Rezoluție: 0,3 cm
  - Unghiul de măsurare: 30 de grade
  - Trigger Input Lățimea impulsului de intrare declanșare: 10uS
  - Dimensiuni: 45 mm x 20 mm x 15 mm
- Funcționare:

Senzorul ultrasonic utilizează sonarul pentru a determina distanța față de un obiect.

1. Transmițătorul (pinul trig) trimite un semnal: un sunet de înaltă frecvență.
2. Când semnalul găsește un obiect, acesta se reflectă și ...
3. ... receptorul (pinul echo) îl primește.





Timpul dintre transmisia și recepția semnalului ne permite să cunoaștem distanța față de un obiect. Acest lucru este posibil deoarece știm viteza sunetului în aer.

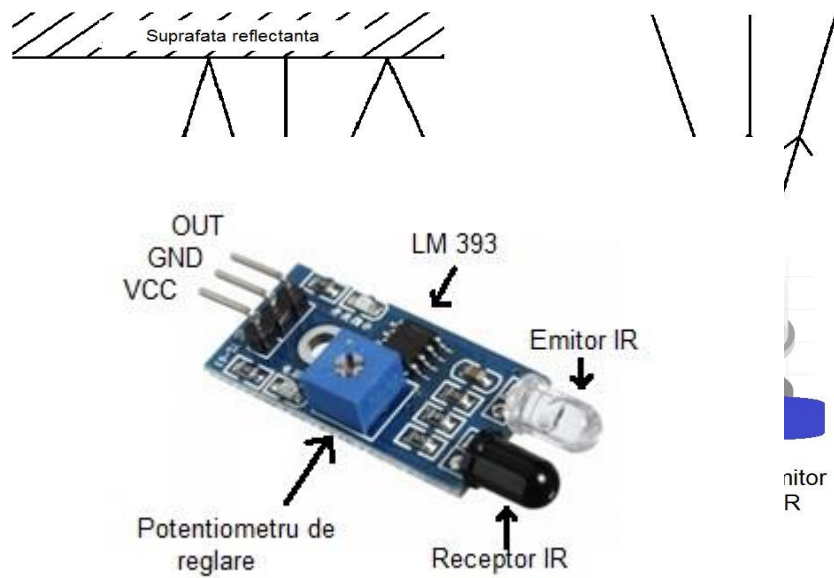


- Pini
  - VCC: + 5 VDC
  - Trig: Trigger (INPUT)
  - Echo: Echo (OUTPUT)
  - GND: GND

#### ➤ Senzor cu infraroșu

Senzorul de obstacole se bazează pe reflexia radiației IR de către obstacol. Radiația IR este emisă de către un LED și este recepționată de către un fototranzistor. Modulul conține două led-uri indicatoare, unul pentru alimentare și celălalt pentru detectarea obstacolului.

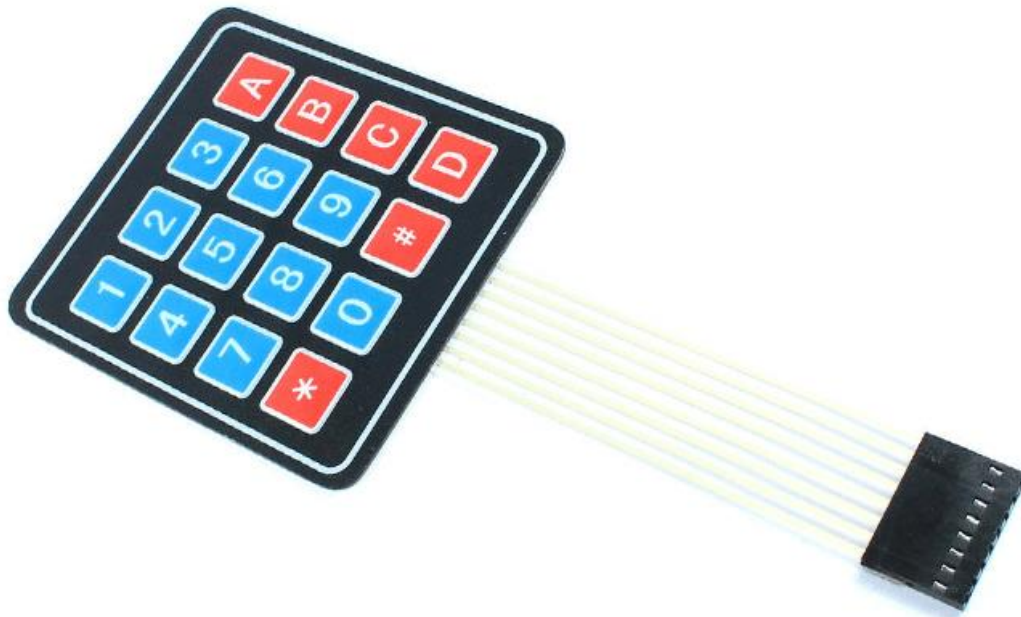
- Caracteristici:
  - Tensiune de alimentare: 3V - 5V
  - Distanță sesizare obstacol: 2cm - 30cm
  - Unghi observare obstacol: 35°
  - Output digital
  - Comparator LM393
  - Tensiune de referință reglabilă



Cu ajutorul potențiometrului putem modifica nivelul de referință, astfel încât să ajustăm sensibilitatea, deci, distanța la care modulul detectează obstacole. Comparatorul furnizează la ieșire 1 logic atunci când nu detectează obstacole și 0 logic atunci când întâlnește un obstacol.

## 7. Periferice

- Tastatură matricială 4x4

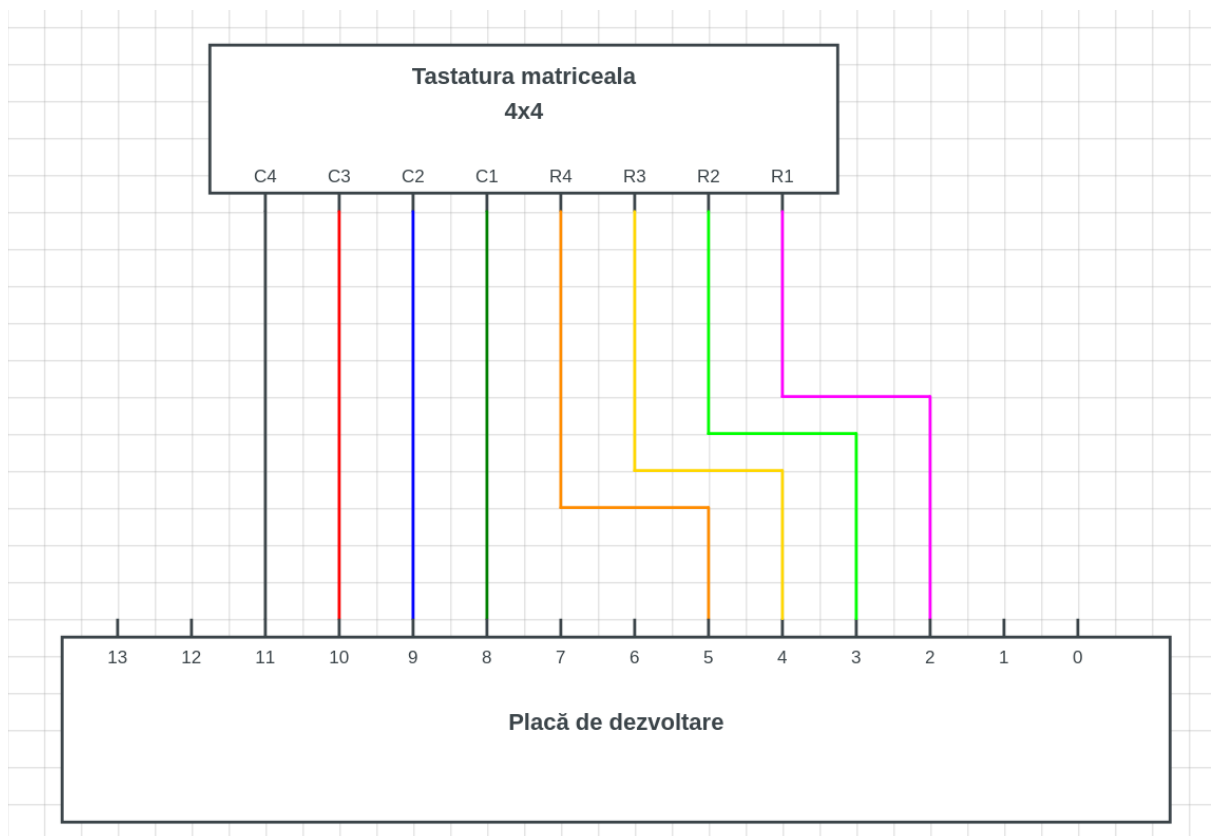
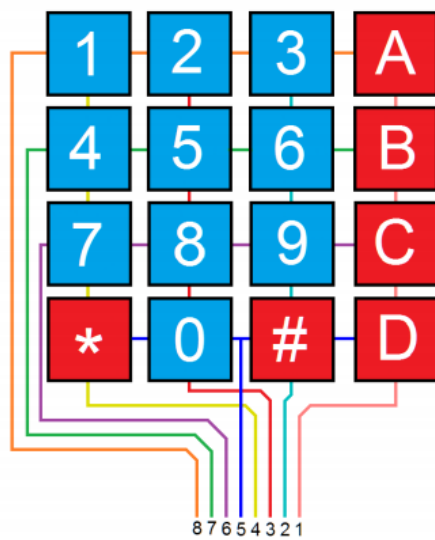


Tastatura cu 16 taste (cifrele de la 0 la 9, A, B, C, D, steluta si diez). Tastele sunt conectate in forma matriceala 4x4. Tastatura este echipata cu un cablu de tip DIP mama cu pas de 2.54mm cu lungimea de 5cm, iar pe spate este cu autoadeziv, ceea ce implica o usoara fixare pe carcase sau panouri.

- Caracteristici:

- |                   |                        |
|-------------------|------------------------|
| ○ Taste           | 16                     |
| ○ Dispunere taste | 4x4                    |
| ○ Conector        | DIP mama cu pas 2.54mm |
| ○ Lungime cablu   | 5cm                    |

- Fixare autoadeziv



- Funcționare:

Tastaturile matriciale utilizează o combinație de patru rânduri și patru coloane. Sub fiecare tastă se află un pushbutton, cu un capăt conectat la un rândul și celălalt capăt conectat la o coloană.

### ➤ Buzzer

Buzzer este foarte util în proiectele de electronică pentru avertizări sonore. Acest dispozitiv poate reda o gamă largă de frecvențe.

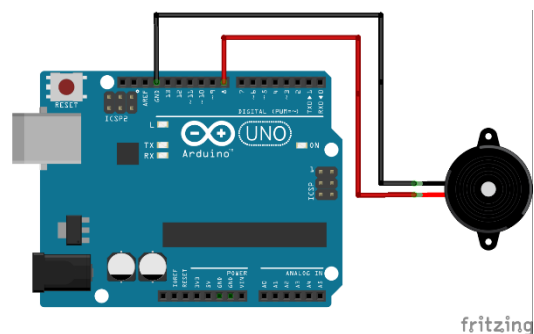
- Caracteristici:
  - Tensiune recomandată: 3V
  - Tensiune de funcționare: 2V - 5V
  - Curent: 30mA (MAX)
  - Minim 80dB, la 10cm distanță
  - Frecvență de rezonanță: 2300  $\pm$  300 Hz

Dispozitive ce pot produce un sunet atunci când se află sub influența unei tensiuni variabile. Acestea au o construcție mult mai simplă decât difuzoarele din boxele audio.

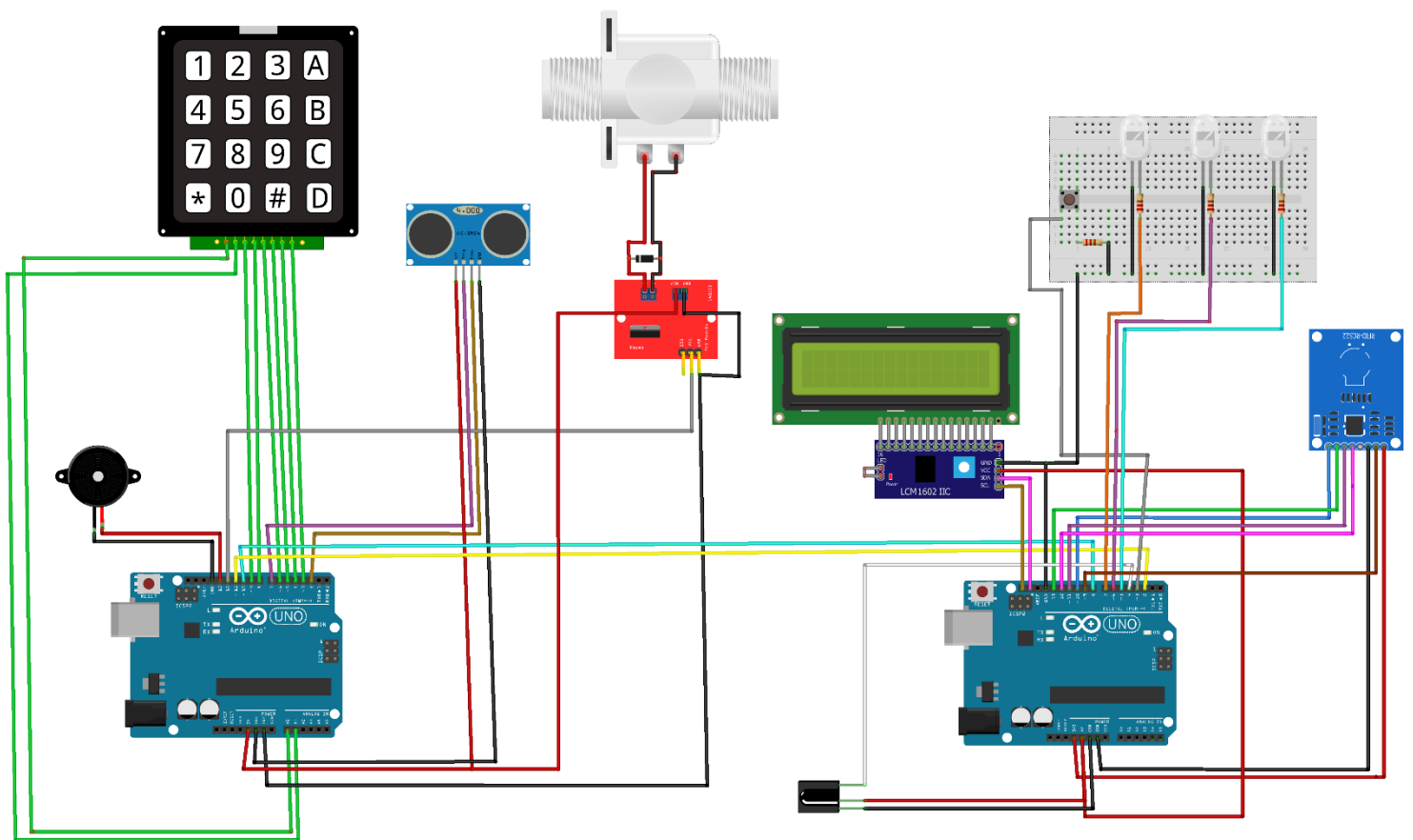
Buzzerele active preiau energia de la o sursă (precum pinul de 5 V de pe placuța Arduino, care poate genera o putere mai mare decât un pin obișnuit) și sunt acționate prin intermediul unui element comutor.

Există 3 pini de conexiune: VCC, GND și I/O. Pinul I/O se conectează la microcontroller sau la sursa de semnal dreptunghiular pentru a genera sunetul.

 Gorham Digital



## 8. Conectarea hardware



fritzing

Schema este alcătuită din două plăci Arduino Uno.

La prima am conectat:

- un ecran LCD prin intermediul unui modul I<sup>2</sup>C,
- un modul RFID,
- un senzor de proximitate cu infraroșu și
- un breadboard pe care e atașat un push button și trei leduri + rezistențe.

La a doua placă am legat:

- tastatură,
- un buzzer,
- un modul tranzistor de putere (diodă pentru flyback) și
- pompă de apă de 5V.

Cele două plăci comunică între ele.













## 9. Programele

```
//////////////////// Setup //////////////////////////////////////
void setup() {
  myserial.begin(9600);
  Serial.begin(9600);
  //Arduino Pin Configuration
  pinMode(ir, INPUT); //sensor IR
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(yellowLed, OUTPUT);
  pinMode(wipeB, INPUT); // Enable pin's pull up resistor
  // Make sure led's are off
  digitalWrite(ir, LOW);
  digitalWrite(redLed, LOW);
  digitalWrite(greenLed, LOW);
  digitalWrite(yellowLed, LOW);
  digitalWrite(wipeB, HIGH);
  //Protocol Configuration
  lcd.init(); // initialize the LCD
  lcd.backlight();
  SPI.begin(); // MFRC522 Hardware uses SPI protocol
  mfrc522.PCD_Init(); // Initialize MFRC522 Hardware
  ShowReaderDetails(); // Show details of PCD - MFRC522 Card Reader details
  //Wipe Code - If the Button (wipeB) Pressed while setup run (powered on) it wipes EEPRO
  // Check if admin card defined, if not let user choose a admin card
  // This also useful to just redefine the Admin Card
  // I write 143 to EEPROM address 1 to know when we have Admin
  if (EEPROM.read(1) != 143) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("No Admin Card Set");
    delay(2000);
    lcd.setCursor(0, 0);
    lcd.print("Scan A Tag to ");
    lcd.setCursor(0, 1);
    lcd.print("Set as Admin");
    do {
      successRead = getID(); // sets successRead to 1 when we get read from reader otherwise 0
      // Visualize Admin Card need to be defined
      digitalWrite(yellowLed, HIGH);
      delay(200);
      digitalWrite(yellowLed, LOW);
      delay(200);
    }
    while (!successRead); // Program will not go further while you not get a successful read
    for ( uint8_t j = 0; j < 4; j++ ) { // Loop 4 times
      EEPROM.write( 2 + j, readCard[j] ); // Write scanned Tag's UID to EEPROM, start from address 3
    }
    EEPROM.write(1, 143); // Write to EEPROM we defined Admin Card.
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Admin set");
    delay(2000);
  }
  for ( uint8_t i = 0; i < 4; i++ ) { // Read Admin Card's UID from EEPROM
    masterCard[i] = EEPROM.read(2 + i); // Write it to adminCard
  }
  ShowOnLCD(); // Print data on LCD
  cycleLeds(); // Everything ready lets give user some feedback by cycling leds
}
```

```

//////////////////// Main Loop //////////////////////
void loop () {

do {
  successRead = getID(); // sets successRead to 1 when we get read from reader otherwise 0
  if (programMode) {
    cycleLeds();          // Program Mode cycles through Red Green Yellow waiting to read a new card
  }
  else {
    normalModeOn();       // Normal mode, yellow Power LED is on, all others are off
  }
}
while (!successRead); //the program will not go further while you are not getting a successful read
if (programMode) {
  if ( isMaster(readCard) ) { //When in program mode check First If master card scanned again to exit program
mode
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Exit programming");
    lcd.setCursor(0, 1);
    lcd.print("mode!");
    lcd.setCursor(0, 2);
    lcd.print("Bye, ADMIN");
    delay(3000);
    ShowOnLCD();
    programMode = false;
    return;
  }
  else {
    if ( findID(readCard) ) { // If scanned card is known delete it
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Already registered,");
      lcd.setCursor(0, 1);
      lcd.print("removal...");
      deleteID(readCard);
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Add/Remove another");
      lcd.setCursor(9,1);
      lcd.print("or");
      lcd.setCursor(0, 2);
      lcd.print("ADMIN card to exit");
    }
    else { // If scanned card is not known add it
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("New Tag,");
      lcd.setCursor(0, 1);
      lcd.print("adding...");
      writeID(readCard);
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("Add/Remove another");
      lcd.setCursor(9,1);
      lcd.print("or");
      lcd.setCursor(0, 2);
      lcd.print("ADMIN card to exit");
    }
  }
}
}

```

```

}
else {
  if ( isMaster(readCard)) { // If scanned card's ID matches Master Card's ID - enter program mode
    programMode = true;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Hello, ADMIN!");
    lcd.setCursor(0, 1);
    lcd.print("You are in");
    lcd.setCursor(0, 2);
    lcd.print("programming mode!");
    uint8_t count = EEPROM.read(0); // Read the first Byte of EEPROM that stores the number of ID's in
EEPROM
    lcd.setCursor(0, 3);
    lcd.print("No. of USERS: ");
    lcd.print(count);
    delay(4000);

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Hold reset button");
    lcd.setCursor(0,1);
    lcd.print("to delete ALL data");
    delay(3000);

    lcd.clear();
    if (digitalRead(wipeB) == LOW) { // when button pressed pin should get low, button connected to ground
      digitalWrite(redLed, HIGH); // Red Led stays on to inform user we are going to wipe
      lcd.setCursor(0, 0);
      lcd.print("Button Pressed");
      delay(1000);
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("This will remove");
      lcd.setCursor(0, 1);
      lcd.print("all records");
      delay(2000);
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("You have 10 ");
      lcd.setCursor(0, 1);
      lcd.print("secs to Cancel");
      delay(2000);
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Unpres to cancel");
      lcd.setCursor(0, 1);
      lcd.print("Counting: ");
      bool buttonState = monitorWipeButton(10000); // Give user enough time to cancel operation
      if (buttonState == true && digitalRead(wipeB) == LOW) { // If button still be pressed, wipe EEPROM
        lcd.print("Wiping EEPROM...");
        for (uint16_t x = 0; x < EEPROM.length(); x = x + 1) { //Loop end of EEPROM address
          if (EEPROM.read(x) == 0) { //If EEPROM address 0
            // do nothing, already clear, go to the next address in order to save time and reduce writes to EEPROM
          }
          else {
            EEPROM.write(x, 0); // if not write 0 to clear, it takes 3.3mS
          }
        }
        lcd.clear();
      }
    }
  }
}

```

```

        lcd.setCursor(0, 0);
        lcd.print("Wiping Done");
        // visualize a successful wipe
        digitalWrite(redLed, LOW);
        delay(200);
        digitalWrite(redLed, HIGH);
        delay(200);
        digitalWrite(redLed, LOW);
        delay(200);
        digitalWrite(redLed, HIGH);
        delay(200);
        digitalWrite(redLed, LOW);
    }
    else {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Wiping Cancelled"); // Show some feedback that the wipe button did not pressed for 10 seconds
        digitalWrite(redLed, LOW);
    }
}

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("ADD/REMOVE a card");
lcd.setCursor(9, 1);
lcd.print("or");
lcd.setCursor(0, 2);
lcd.print("ADMIN card to exit");
}
else {
    if ( findID(readCard) ) { // If not, see if the card is in the EEPROM
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Access Granted");
        granted();
        ShowOnLCD();
    }
    else { // If not, show that the Access is denied
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Access Denied");
        denied();
        ShowOnLCD();
    }
}
}
}

void loop() {
    //ultrasunete
    double val;
    double ddd;

    digitalWrite(trig,LOW);
    delayMicroseconds(2);
    digitalWrite(trig,HIGH);
    delayMicroseconds(10);
    digitalWrite(trig,LOW);
    //FUNCTIE PT MASURAREA DURATEI IMPULSULUI
    val=pulseIn(echo,HIGH);
    ddd=(val*0.034)/2;

```

```

Serial.println();Serial.println(ddd);Serial.println();

if(ddd>15)
    mySerial.write("a");

if(ddd>13.5 && ddd<15)
    mySerial.write("b");

if(ddd>12 && ddd<13.5)
    mySerial.write("c");

if(ddd>10.5 && ddd<12)
    mySerial.write("d");

if(ddd>9 && ddd<10.5)
    mySerial.write("e");

if(ddd>7.5 && ddd<9)
    mySerial.write("f");

if(ddd>4.5 && ddd<7.5)
    mySerial.write("g");

if(ddd>3 && ddd<4.5)
    mySerial.write("h");

if(ddd<3)
    mySerial.write("i");

while(!mySerial.available())
    return;

key = mySerial.read();
Serial.print(key);

if(key == 'l')
{
    if(ddd>15)
    {
        Serial.println();Serial.println("0  ");Serial.println();
        citireaSiDozare(23);
    }
    if(ddd>13.5 && ddd<15)
    {
        Serial.println();Serial.print("15");Serial.print("  ");Serial.println();
        citireaSiDozare(23);
    }
    if(ddd>12 && ddd<13.5)
    {
        Serial.println();Serial.print("25");Serial.print("  ");Serial.println();
        citireaSiDozare(23);
    }
    if(ddd>10.5 && ddd<12)
    {
        Serial.println();Serial.print("40");Serial.print("  ");Serial.println();
        citireaSiDozare(21);
    }
    if(ddd>9 && ddd<10.5)
    {

```



```
    Serial.println();Serial.print("50");Serial.print(" %");Serial.println();
    citireaSiDozare(20);
}
if(ddd>7.5 && ddd<9)
{
    Serial.println();Serial.print("65");Serial.print(" %");Serial.println();
    citireaSiDozare(19);
}
if(ddd>4.5 && ddd<7.5)
{
    Serial.println();Serial.print("75");Serial.print(" %");Serial.println();
    citireaSiDozare(18);
}
if(ddd>3 && ddd<4.5)
{
    Serial.println();Serial.print("90");Serial.print(" %");Serial.println();
    citireaSiDozare(18);
}
if(ddd<3)
{
    Serial.println();Serial.print("100");Serial.print(" %");Serial.println();
    citireaSiDozare(18);
}
}
```

## 10. Bibliografie

- [https://www.google.com/search?q=schema+bloc+Arduino+Microcontroller+ATMEGA328&sxsrf=ALeKk01fPBtakL6DGen8XjQoc8ICQckS1A:1586531419629&source=lnms&tbm=isch&sa=X&ved=2ahUKEwi5msSNkt7oAhWKvosKHdA6DnsQ\\_AUoAXoECAwQAw&biw=1366&bih=625#imgsrc=Aj35s5qe5QiFGM](https://www.google.com/search?q=schema+bloc+Arduino+Microcontroller+ATMEGA328&sxsrf=ALeKk01fPBtakL6DGen8XjQoc8ICQckS1A:1586531419629&source=lnms&tbm=isch&sa=X&ved=2ahUKEwi5msSNkt7oAhWKvosKHdA6DnsQ_AUoAXoECAwQAw&biw=1366&bih=625#imgsrc=Aj35s5qe5QiFGM)
- <https://cleste.ro/arduino-microcontroller-atmega328p-pu.html>
- <https://sites.google.com/site/arduinoelectronicasiprogramare/prima-lectie/5-introducere-in-arduino>
- [http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf)
- [https://www.beta-estore.com/download/rk/RK-10290\\_410.pdf?fbclid=IwAR2k8op6Luqzrl9ScKMxFE6JwjaF21dm-afRfFQkQHZX8lFCnB3dGV-24o](https://www.beta-estore.com/download/rk/RK-10290_410.pdf?fbclid=IwAR2k8op6Luqzrl9ScKMxFE6JwjaF21dm-afRfFQkQHZX8lFCnB3dGV-24o)
- <https://sites.google.com/site/arduinoelectronicasiprogramare/arduino-si-senzori/1>
- <https://www.adelaida.ro/serial-lcd-adapter-2x16-4x20-mcp23s17-adaptor-i2c-lcd.html>
- <https://www.optimusdigital.ro/ro/senzori-senzori-optici/4514-senzor-infrarosu-de-obstacole.html>
- <https://www.xab3.ro/produse/tastatura-4x4>