Universitatea Politehnica Timișoara

Facultatea de Automatică și Calculatoare

***Sisteme Încorporate***

**Implementarea aplicației Automotive 2**

Haidu Diana-Maria

Rusovan Paul – Daniel

Anul III, CTI

**Tema proiectului:**

**Aplicaţia „Automotive 2”:** Facilitarea manevrelor de parcare a unui automobil:

măsurarea distanţei rămase până la un obstacol folosind un microcontroler şi

senzori de distanţă.

Caracteristici:

• Senzorii de distanţă se pot lega la o interfaţă serială, precum CAN, I2C sau SPI, la

alegerea proiectantului;

• Se va măsura o distanţă cuprinsă între 10 şi 70 – 100 cm;

• Măsurarea distanţei va fi însoţită de o alarmă sonoră: frecvenţa sunetului va creşte pe

masură ce distanţa până la obstacol scade;

• Valoarea distanţei măsurate va fi afişată, la alegerea studentului, pe afişaje cu 7 segmente

sau matrice de LED-uri sau afişaj LCD;

• Se vor utiliza cel puțin 2 senzori de distanță, iar aplicația va avea un prototip practic.

1. **Caracteristici generale - Arduino Uno**



Principalele caracteristici ale plăcii Arduino Uno:

* Microcontroler: ATmega328P Microchip (Atmel)
* Tensiune de operare: 5 V
* Tensiune de intrare: 7 - 20 V
* Pinii I/O digitali: 14 (dintre care 6 au PWM output)
* Pinii input analogici: 6
* Curent DC per I/O Pin: 20 mA
* Curent DC pentru 3.3V Pin: 50 mA
* Flash Memory: 32 KB
* SRAM: 2 KB
* EEPROM: 1 KB
* Viteză Clock: 16 MHz
* Lungime: 68.6 mm
* Lățime: 53.4 mm
* Greutate: 25 g

**Funcțiile generale ale pinilor**

* VIN: Tensiunea de intrare pentru placa Arduino când folosește sursă de putere externă. Se poate furniza tensiune prin acest pin sau dacă se alimentează prin intermediul mufei accesul se face cu ajutorul acestiu pin

5V: Pinul aceste scoate o tensiune de 5 V de la regulatorul de pe placa.Placa poate sa fie pornita de la mufa DC (7-20V),mufa USB(5V) sau de la pin-ul VIN de pe placa(7-20V). Tensiunea de alimentare prin pinii 5V sau 3,3V ocolește regulatorul și poate deteriora placa.

* 3V3: o sursă de 3.3 volți generată de regulatorul de bord. Rezistența curentului maxim este de 50 mA.
* GND: pinii de masa.
* IOREF: Acest pin pe placa Arduino / Genuino furnizează referința de tensiune cu care microcontrolerul funcționează. Un scut configurat corespunzător poate citi tensiunea de pin IOREF și poate selecta sursa de alimentare adecvată sau poate activa traducătorii de tensiune pe ieșiri pentru a funcționa cu 5V sau 3,3V.
* Resetare: Se utilizează în mod obișnuit pentru a adăuga un buton de resetare la scuturile care blochează cel de pe placă.

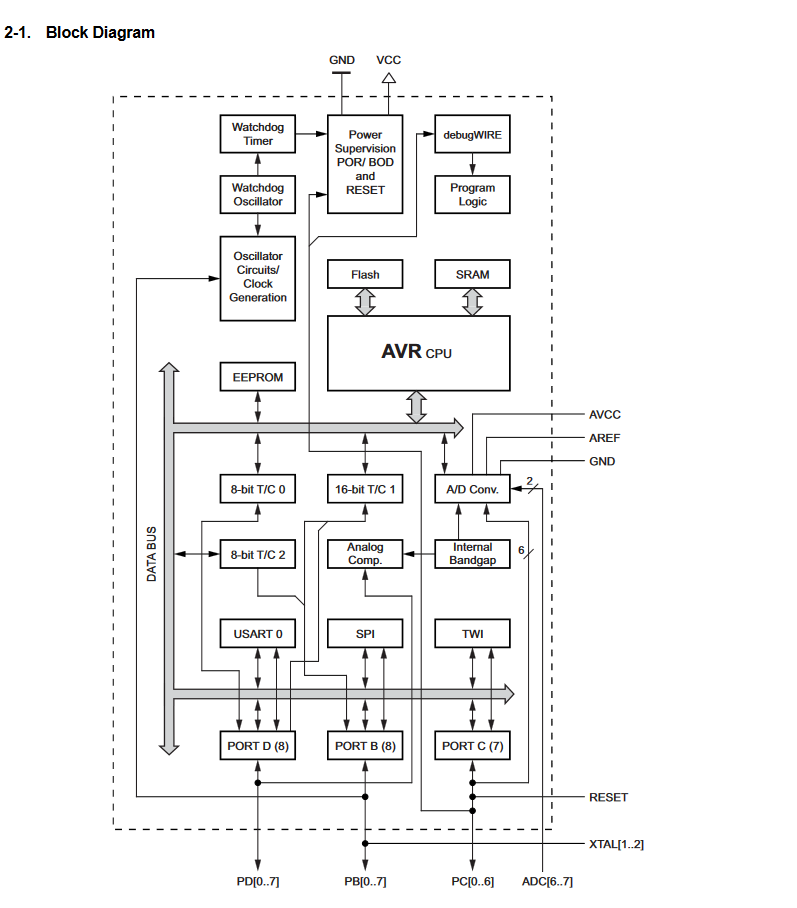
**Functii speciale ale piniilor**

Fiecare dintre cei 14 pini digitali și 6 pini analogi ai Uno pot fi utilizați ca intrări sau ieșiri, utilizând funcțiile pinMode (), digitalWrite () și digitalRead (). Acestea funcționează la 5 volți. Fiecare pin poate furniza sau recepționa 20 mA ca condiție de funcționare recomandată și are o rezistență internă de tracțiune (deconectată în mod implicit) de 20-50k ohm. Un maxim de 40 mA este valoarea care nu trebuie depășită pe nici un pin de intrare / ieșire pentru a evita deteriorarea permanentă a microcontrolerului. Uno are 6 intrări analogice, numite A0 până la A5, fiecare furnizând 10 biți de rezoluție (adică 1024 diferite valori). În mod implicit, acestea măsoară de la sol la 5 volți, deși este posibil să se schimbe capătul superior al domeniului lor folosind pinul AREF și funcția analogReference ().

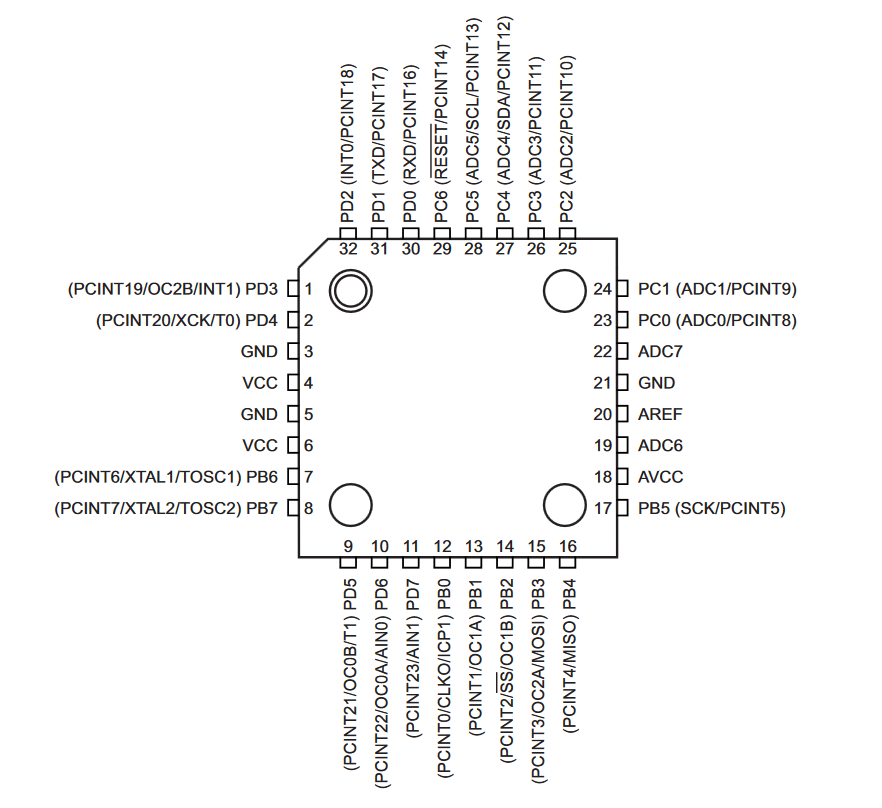
**Pini cu funcții specializate:**

* Serial / UART: pinii 0 (RX) și 1 (TX). Folosit pentru a recepționa (RX) și a transmite (TX) date seriale TTL. Acești ace sunt conectați la pinii corespunzători ai cipului serial ATmega8U2 USB-to-TTL
* Intreruperi externe: pinii 2 și 3. Aceștia pot fi configurați pentru a declanșa o întrerupere la o valoare scăzută, o margine în creștere sau în scădere sau o schimbare a valorii.
* PWM (pulse-width modulation): 3, 5, 6, 9, 10, and 11 Can provide 8-bit PWM output with the analogWrite() function.
* PWM (modulare puls-lățime): 3, 5, 6, 9, 10 și 11 Poate furniza ieșire PWM pe 8 biți cu funcția analogWrite ().
* TWI (interfață cu două fire) / I²C: pin A4 sau SDA și PIN A5 sau SCL. Suportă comunicarea TWI folosind biblioteca Wire.
* AREF (referință analogică): Tensiunea de referință pentru intrările analogice

1. **Microcontrolerul și senzorii folosiți**

**B.1.** **Microcontrolerul ATmega328P**

*Diagrama bloc a unității microcontrolerului*



*Pinii unității microcontrolerului (120 pini)*

Descrierea Piniilor

1.VCC

Alimentare.

2. GND

Împământare.

3. Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Portul B este un port I / O bi-direcțional pe 8 biți cu rezistențe interne de tracțiune (selectate pentru fiecare bit). Tampoanele de ieșire Port B au caracteristici simetrice de acționare, atât cu capacitate ridicată, cât și cu sursă. Ca intrări, pinii portului B extrași din exterior vor fi sursă de curent dacă sunt activate rezistențele de tracțiune. Pinii Port B sunt trimiși când o condiție de resetare devine activă, chiar dacă ceasul nu funcționează. În funcție de setările siguranței de selectare a ceasului, PB6 poate fi folosit ca intrare pentru amplificatorul de oscilator invers și pentru intrarea în circuitul de funcționare a ceasului intern. În funcție de setările siguranței de selectare a ceasului, PB7 poate fi utilizat ca ieșire de la amplificatorul oscilator inversor. Dacă oscilatorul RC calibrat intern este folosit ca sursă de ceas cip, PB7..6 este utilizat ca intrare TOSC2..1 pentru asincron Timer / Counter2 dacă este setat bitul AS2 în ASSR.

4. Portul C (PC5: 0) Portul C este un port I / O bidirecțional pe 7 biți cu rezistențe interne de tracțiune (selectate pentru fiecare bit). Tampoanele de ieșire PC5..0 au caracteristici simetrice de antrenare, atât cu capacitate ridicată, cât și cu sursă. Ca intrări, pinii Port C care sunt extrași extern vor fi sursă de curent dacă sunt activate rezistențele de tracțiune. Pinii portului C sunt trimiși când o condiție de resetare devine activă, chiar dacă ceasul nu funcționează.

5. PC6 / RESET Dacă este programată siguranța RSTDISBL, PC6 este folosit ca un pin de intrare. Dacă siguranța RSTDISBL este neprogramată, PC6 este utilizată ca intrare de resetare. Un nivel scăzut al acestui pin mai mult decât lungimea minimă a impulsului va genera o resetare, chiar dacă ceasul nu este pornit. Lungimea minimă a impulsurilor este dată în tabelul 28-4 de la pagina 261. Nu este garantat faptul că impulsurile mai scurte generează o resetare.

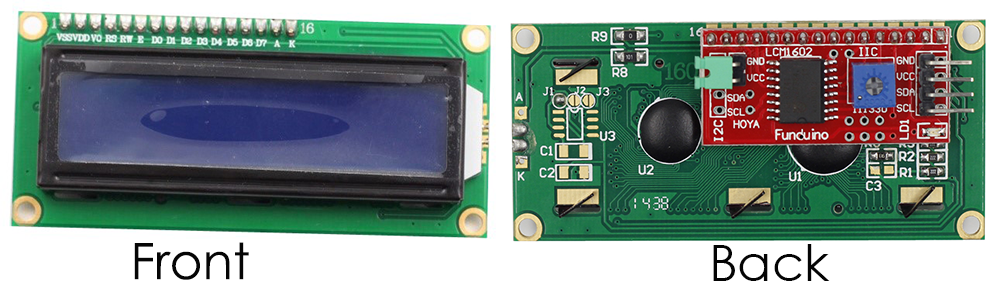
6. Portul D (PD7: 0) Portul D este un port I / O bidirecțional pe 8 biți cu rezistențe interne de tracțiune (selectate pentru fiecare bit). Tampoanele de ieșire ale portului D au caracteristici simetrice de acționare, atât cu capacitate ridicată, cât și cu sursă. Ca intrări, pinii portului D care sunt extrași extern vor fi sursă de curent dacă sunt activate rezistențele de tracțiune. Pinii portului D sunt trimiși când o condiție de resetare devine activă, chiar dacă ceasul nu funcționează.

7. AVCC AVCC este pinul de tensiune de alimentare pentru convertorul A / D, PC3: 0 și ADC7: 6. Ar trebui să fie conectat extern la VCC, chiar dacă ADC nu este utilizat. Dacă se utilizează ADC, acesta trebuie conectat la VCC printr-un filtru cu trecere joasă. Rețineți că PC6..4 utilizează tensiunea de alimentare digitală, VCC.

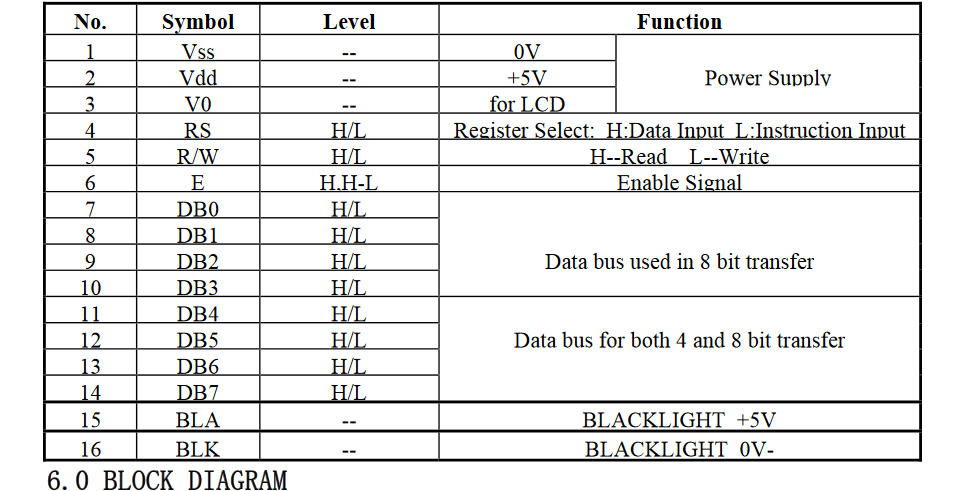
8. AREF AREF este pinul de referință analogic pentru convertorul A / D.

9. ADC7: 6 (doar pachetul TQFP și QFN / MLF) În pachetul TQFP și QFN / MLF, ADC7: 6 servesc drept intrări analogice la convertorul A / D. Acestea sunt alimentați de la sursa analogică și servesc ca canale ADC pe 10 biți

**B.2. Modulul LCD 1602 cu interfața serial I2C**



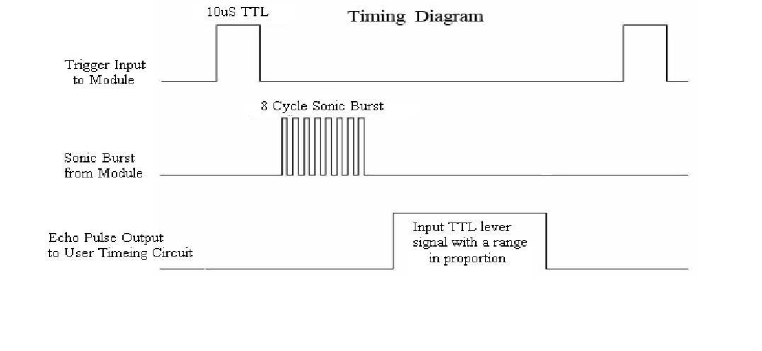
Modulul LCD 1602 cu interfață I2C este un afișaj de 2 linii câte 16 caractere. I2C necesită doar 2 conexiuni, +5V si GND să funcționeze.



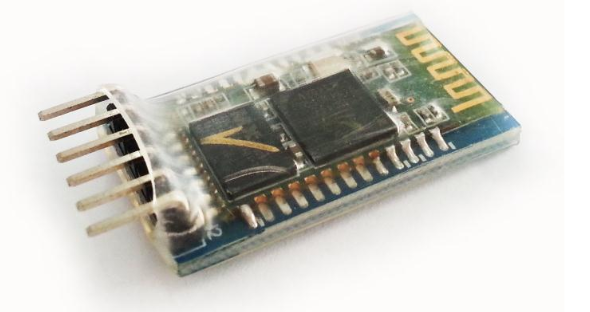
**B.3. Senzorul ultrasonic HC-SR04**

****

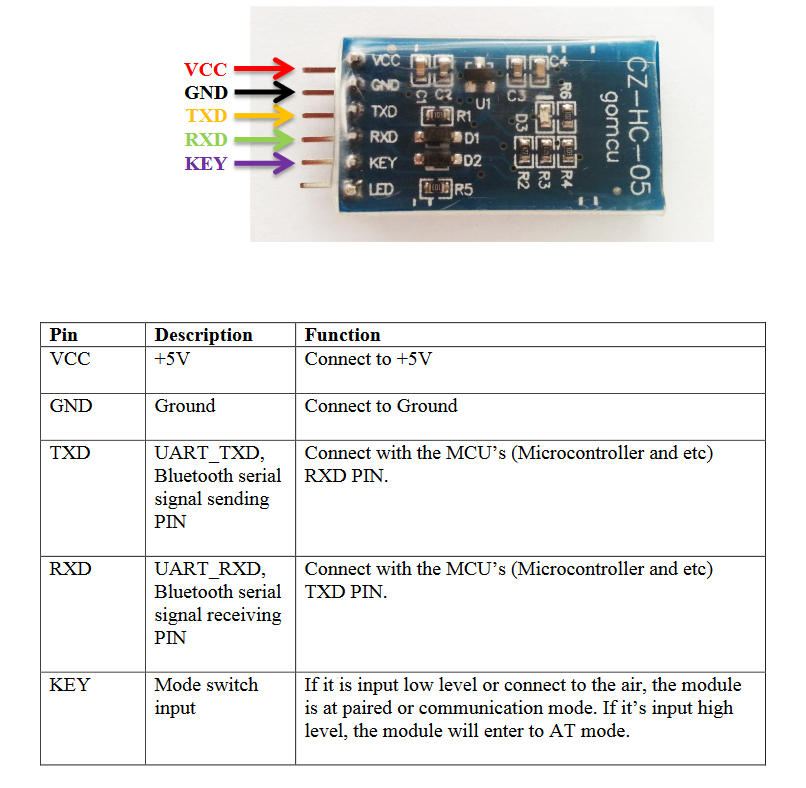
Modulul are nevoie de un puls de 10us spre pinul de input Trig ca să se trimită un semnal de 40kHz. Înapoi vine un semnal de echo care variază. Astfel se poate calcula distanța prin timpul care a trecut între trimiterea semnalului de trigger si recepționarea semnalului de echo . Formula: uS / 58 = centimeters sau range = high level time \* velocity (340M/S) / 2

****

**B.4. Modulul Bluetooth Master-Slave HC05**

****

Modulul BlueTOOTH HC-05 este un modul serial (SPP – Serial Port Protocol) proiectat pt comunicare serial transparentă wireless. Comunicarea dintre module se face serial deci este ușor de interfațat cu un PC sau microcontroller. Modulul poate juca rol de Master sau Slave.



**B.5. Modulul Joystick PS2 Biaxial cu 5 pini**



Modulul dispune de două potentiometer, câte unul pentru fiecare axă. Este compatibil cu interfața Arduino și are 5 pini.

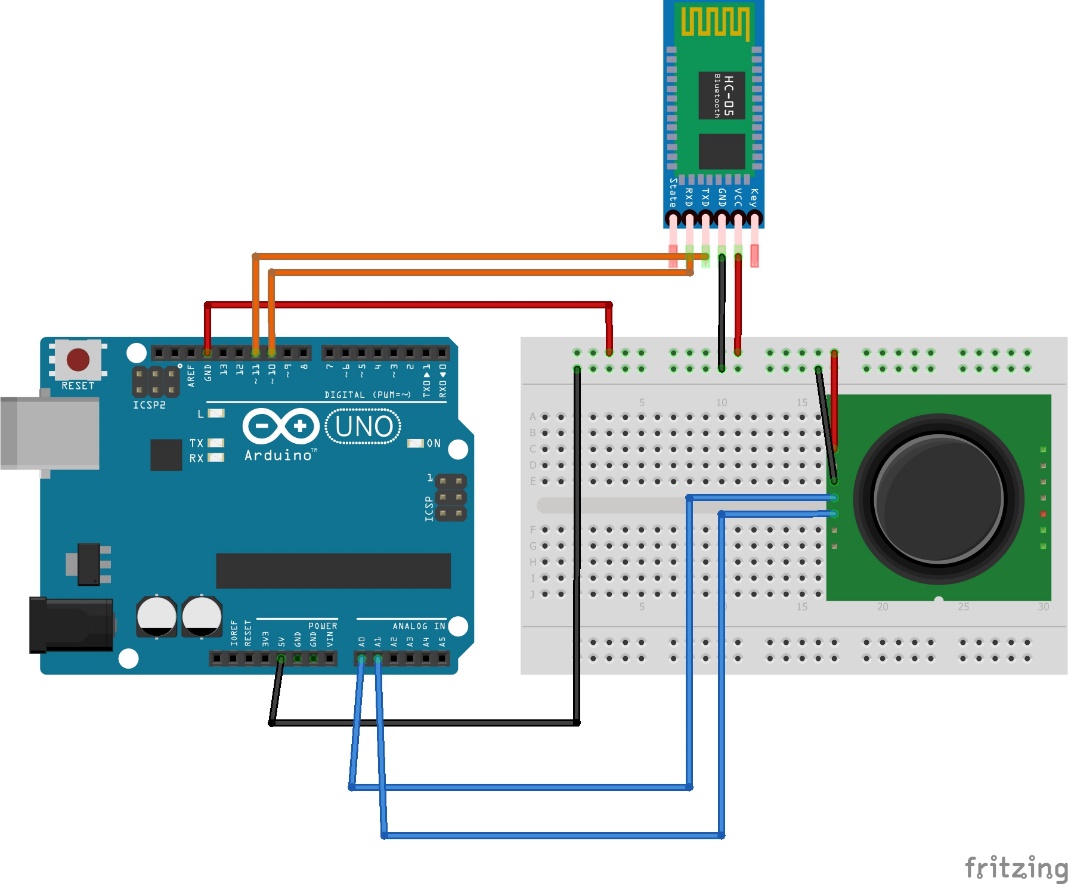
Configurația pinilor: GND – ground, +5V – 5V, VRx – voltaj proportional cu poziția X, VRy – voltaj proportional cu poziția Y, SW – switch pushbutton

1. **Schemă**

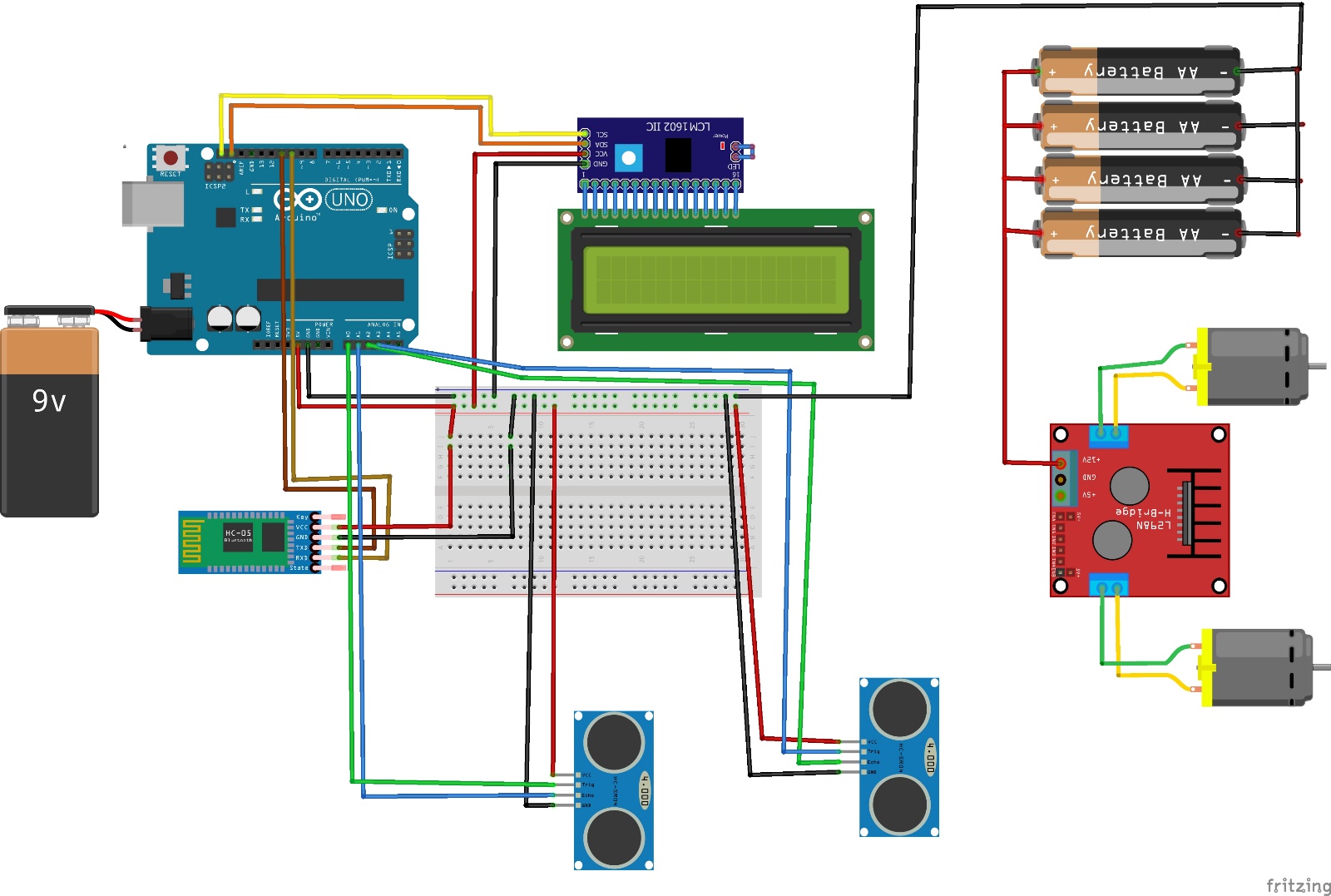
Cei doi senzori de distanță HC-SR04 permit citirea distanței măsurate prin transmiterea unui semnal digital de o anumită durată. Senzorul transmite plăcuței Arduino durata de timp între trimiterea semnalului și recepționarea lui la întoarcere.

Pentru afișaj se va folosește un display LCD 1602 cu interfața serial I2C.

Cele două motoare sunt controlate de către un driver L298N. Modulul oferă facilități de modulare a semnalului transmis către motoare, dar acestea nu se folosesc.

**Telecomandă**

**Mașină**



1. **Programul**

**Joystick:**

#include <SoftwareSerial.h> //bibliotecă pentru comunicarea modulului bluetooth

SoftwareSerial EBlue(10,11); //definirea modulului bluetooth

/\*

MASTER DEVICE - joystick

\*/

int xAxis, yAxis; //axele X si Y ale joystick-ului

void setup() {

EBlue.begin(9600); // canalul de comunicare a modulului bluetooth

}

void loop() {

xAxis = analogRead(A0); // Citește axa X

yAxis = analogRead(A1); // Citește axa Y

// Trimite prin portul serial spre modulul bluetooth slave

if (EBlue.available()){

EBlue.write(xAxis);// Trimite valoarea axei X

delay(10);

EBlue.write(yAxis); //Trimite valoarea axei Y

}

delay(20);

}

**Mașină:**

#define BUZZER\_PIN 3

#include <LiquidCrystal\_I2C.h> //blibliotecă pentru ecranul LCD care comunică prin interfața serială

//I2C

LiquidCrystal\_I2C lcd(0x27, 16, 2); //instanțiere obiect LCD

//pinii de comandă a driverului

int PIN\_DRIVER\_IN1 = 2;

int PIN\_DRIVER\_IN2 = 4;

int PIN\_DRIVER\_IN3 = 7;

int PIN\_DRIVER\_IN4 = 8;

//pinii senzorului de distanță

const int trigPin1 = A0;

const int echoPin1 = A1 ;

const int trigPin2 = A3;

const int echoPin2 = A2;

long duration1, duration2;

int distance1, distance2;

int go\_forward=1;

//buffere

char msg1[21];

char msg2[21];

void setup() {

//setup pentru pinii de control driver

pinMode(PIN\_DRIVER\_IN1, OUTPUT);

pinMode(PIN\_DRIVER\_IN2, OUTPUT);

pinMode(PIN\_DRIVER\_IN3, OUTPUT);

pinMode(PIN\_DRIVER\_IN4, OUTPUT);

//instanțiere obiect lcd

lcd.begin();

lcd.backlight();

//setup pini sensor de distanță

pinMode(BUZZER\_PIN, OUTPUT);

pinMode(trigPin1, OUTPUT);

pinMode(trigPin2, OUTPUT);

pinMode(echoPin1, INPUT);

pinMode(echoPin2, INPUT);

Serial.begin(9600);

}

void senzor1(){

// Sterge trigPin

digitalWrite(trigPin1, LOW);

delayMicroseconds(2);

digitalWrite(trigPin1, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin1, LOW);

duration1 = pulseIn(echoPin1, HIGH);

distance1 = duration1\*0.034/2;

}

void motorStop() {

digitalWrite(PIN\_DRIVER\_IN1, LOW);

digitalWrite(PIN\_DRIVER\_IN2, LOW);

digitalWrite(PIN\_DRIVER\_IN3, LOW);

digitalWrite(PIN\_DRIVER\_IN4, LOW);

}

void senzor2(){

// Sterge trigpin

digitalWrite(trigPin2, LOW);

delayMicroseconds(2);

digitalWrite(trigPin2, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin2, LOW);

duration2 = pulseIn(echoPin2, HIGH);

distance2 = duration2\*0.034/2; //masoara in cm

}

void buzzer(){

if((distance1 < 20) || (distance2 < 20)&&!((distance1 < 10) || (distance2 < 10))){

digitalWrite(BUZZER\_PIN, HIGH);

delay(200);

digitalWrite(BUZZER\_PIN, LOW);

//delay(100);

}else if((distance1 < 10) || (distance2 < 10)){

digitalWrite(BUZZER\_PIN, HIGH);

delay(50);

digitalWrite(BUZZER\_PIN, LOW);

}

else {

digitalWrite(BUZZER\_PIN, LOW);

}

}

void ecran(){

sprintf(msg1, "Distance1: %-7d", distance1);

lcd.setCursor(0, 0);

lcd.print(msg1);

sprintf(msg2, "Distance2: %-7d", distance2);

lcd.setCursor(0, 1);

lcd.print(msg2);

delay(50);

}

void forward()

{

digitalWrite(PIN\_DRIVER\_IN1, HIGH);

digitalWrite(PIN\_DRIVER\_IN2, LOW);

digitalWrite(PIN\_DRIVER\_IN3, HIGH);

digitalWrite(PIN\_DRIVER\_IN4, LOW);

}

//transmiterea semnalelor corespunzătoare către motoare pentru mers înapoi

void backward() {

digitalWrite(PIN\_DRIVER\_IN1, LOW);

digitalWrite(PIN\_DRIVER\_IN2, HIGH);

digitalWrite(PIN\_DRIVER\_IN3, LOW);

digitalWrite(PIN\_DRIVER\_IN4, HIGH);

}

void loop() {

senzor1();

senzor2();

buzzer();

ecran();

if(distance1<10)

{

go\_forward=0;

motorStop();

delay(500);

}

else if(distance2<10)

{

go\_forward=1;

motorStop();

delay(500);

}

if(go\_forward)

{

forward();

}

else

{

backward();

}

}

**E. Bibliografie:**

<https://en.wikipedia.org/wiki/Arduino_Uno>

<http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf>

<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

<https://opencircuit.nl/ProductInfo/1000061/I2C-LCD-interface.pdf>

<http://www.energiazero.org/arduino_sensori/joystick_module.pdf>

<https://www.gme.cz/data/attachments/dsh.772-148.1.pdf>

<https://cdn-learn.adafruit.com/downloads/pdf/adafruit-motor-shield.pdf>

Giuthub: