**Universitatea Politehnica Timisoara**

**Facultatea de Automatica si Calculatoare**

**Proiect Sisteme Incorporate**

**Proiect realizat de :**

**Cojocaru Flavius Gr. 2.1**

**Nitu Vladimir Gr. 5.1**

**Tomuta Tudor Gr. 6.1**

**Anul 3 CTI 2018-2019**

**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;

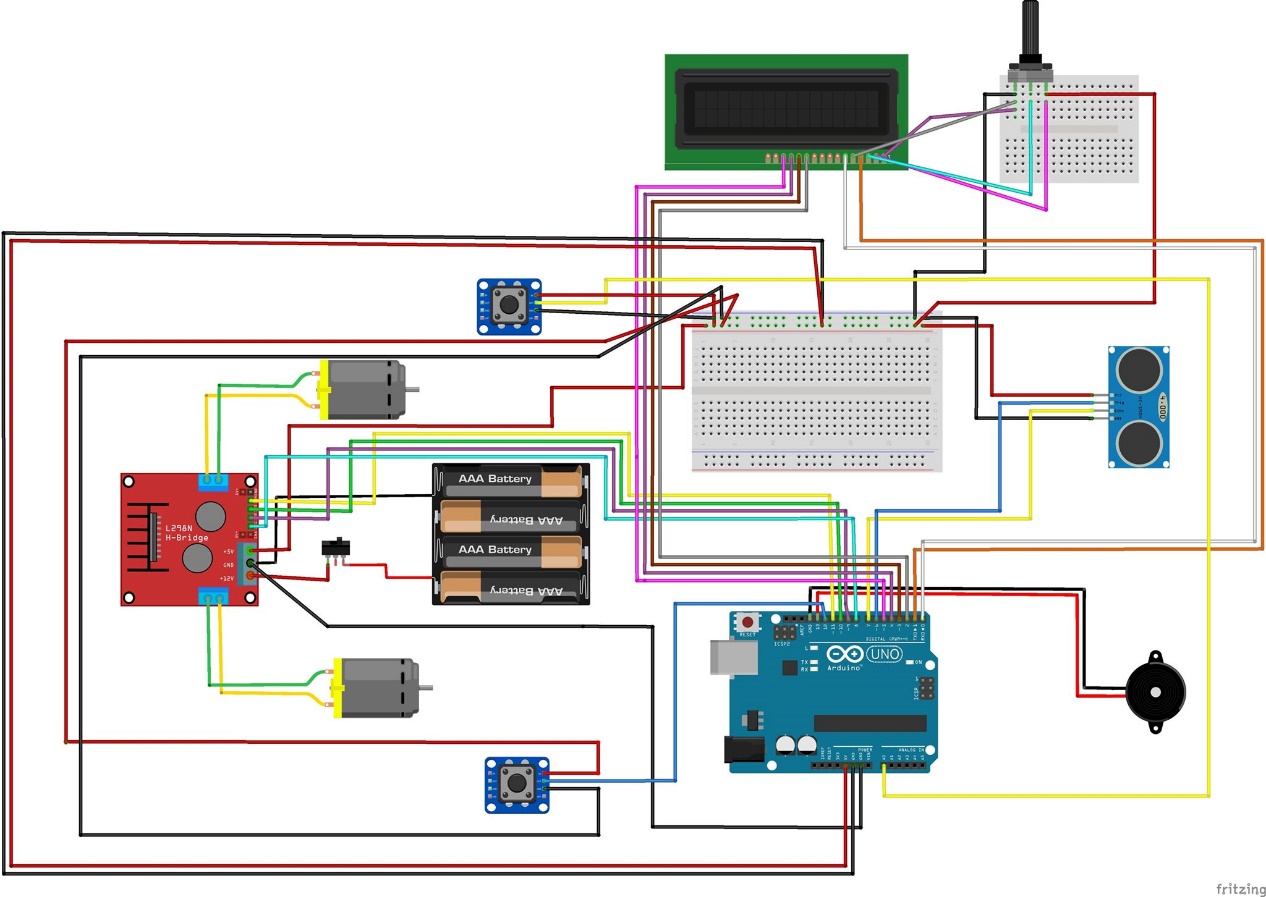
• Se va măsura o distanţă cuprinsă între 10 şi 70 – 100 cm;

• Măsurarea distanţei va fi însoţită de o alarmă sonoră: frecvenţa sunetului va creşte pe masură ce distanţa până la obstacol scade;

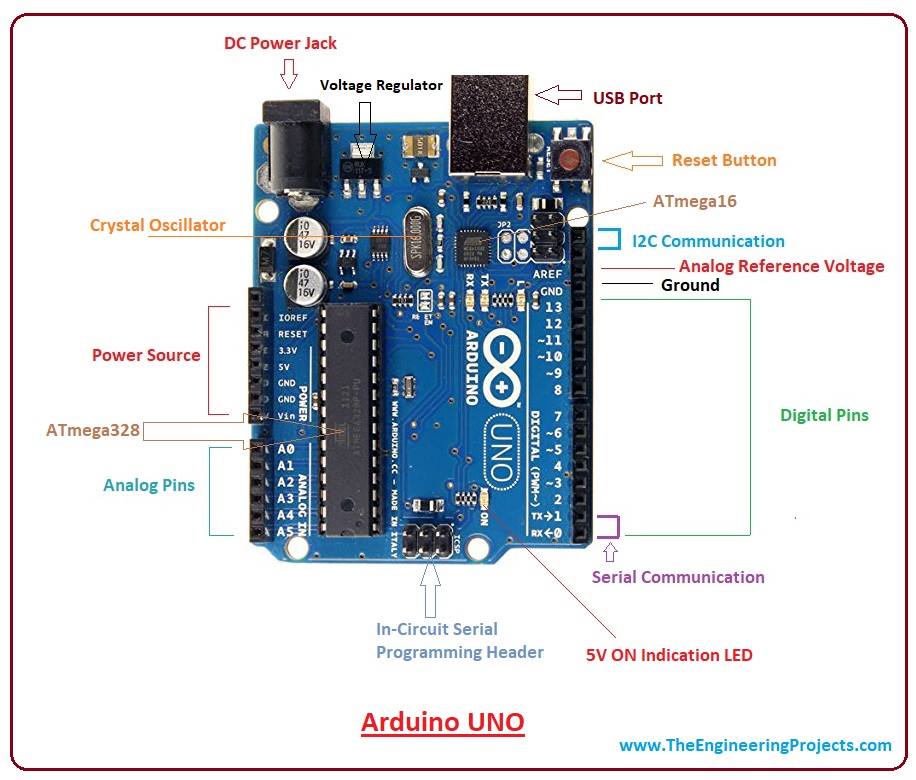
• Valoarea distanţei măsurate va fi afişată, la alegerea studentului, pe afişaje cu 7 segmente sau matrice de LED-uri sau afişaj LCD;

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

**Schema circuitului :**



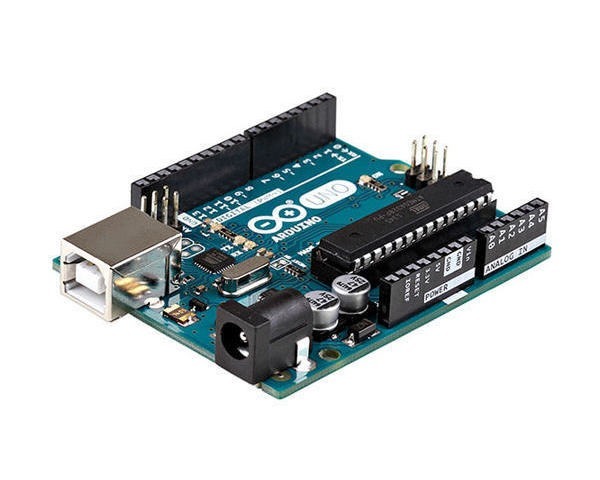
**Placa Arduino Uno R3 :**



Arduino UNO  este o platforma de procesare open-source, bazata pe software si hardware flexibil si simplu de folosit. Consta intr-o platforma de mici dimensiuni (6.8 cm / 5.3 cm – in cea mai des intalnita varianta) construita in jurul unui procesor de semnal si este capabila de a prelua date din mediul inconjurator printr-o serie de senzori si de a efectua actiuni asupra mediului prin intermediul luminilor, motoarelor, servomotoare, si alte tipuri de dispozitive mecanice.

Placa Arduino UNO se conecteaza la portul USB al calculatorului folosind un cablu de tip USB A-B. Poate fi alimentata extern (din priza) folosind un [alimentator extern](https://www.robofun.ro/alimentator-extern-arduino). Alimentarea externa este necesara in situatia in care consumatorii conectati la placa necesita un curent mai mare de cateva sute de miliamperi. In caz contrar, placa se poate alimenta direct din PC, prin cablul USB.

Arduino UNO ne pune la dispozitie 3 timere, acestea incrementandu-se la fiecare ciclu de tact. Prin registrele de comparare integrate atunci cand aceste timere ajung la anumite valori se activeaza intreruperile de tipul Clear Time on Compare, ulterior resetandu-se valoarea pe 0.



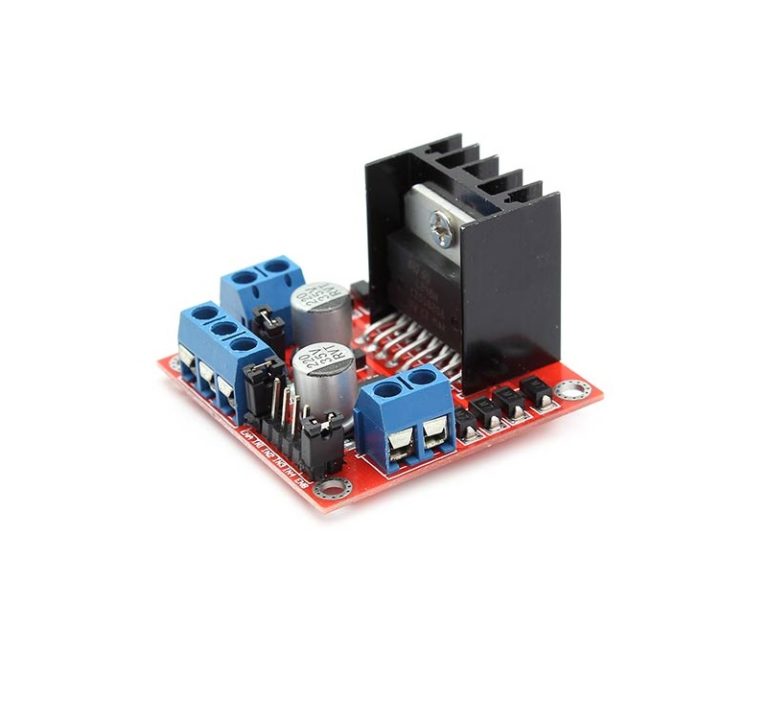
Această placuță de dezvoltare este ideală pentru proiecte creative din domeniul electronicii. Este o placa de dezvoltare compatibila cu Arduino Uno și utilizează microntroller-ul ATmega328p.

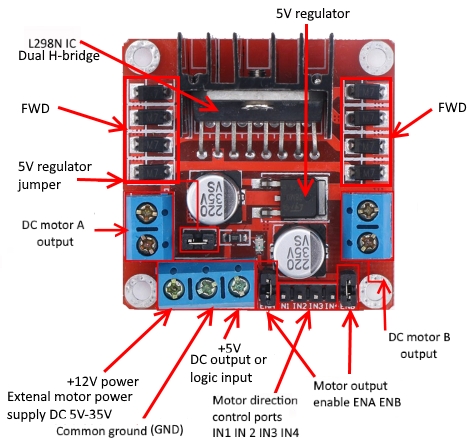
### Caracteristici tehnice :

* Tensiune de funcționare: 5V;
* Tensiune de alimentare Jack: 7V - 12V;
* Pini de I/O: 14;
* Pini PWM: 6 (din cei 14 de I/O);
* Pini ADC: 8;
* Memorie flash: 32kB (8 ocupați de bootloader);
* Comunicație TWI, SPI și UART;
* Frecvență de funcționare: 16 MHz.

# L298N Dual H-Bridge Motor Driver Module

Schema & componente :





Caracteristici:

Accepta tensiune standard TTL pana la 46V si curent continuu de pana la 2 A.

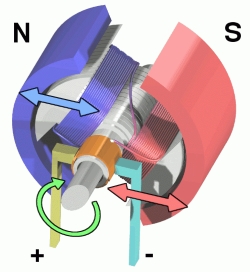
Specificatii:

* Driver: L298 Dual H Bridge DC Motor Driver IC
* Tensiune de folosire : 7-35V
* Curent folosit: 2A
* Putere maxima: 20W (T = 75 °C)
* Dimensiuni: 55mm \* 49mm \* 33mm
* Greutate: 33g

CUM FUNCTIONEAZA MOTORUL DC

Într-un simplu motor DC există două componente principale, "statorul" și "armatura". Statorul este un magnet permanent și oferă un câmp magnetic constant. Armătura, care este partea rotativă, este o simplă bobină. Armătura este conectată la o sursă de curent continuu folosind un inel din 2 piese instalat în jurul arborelui motor, aceste secțiuni de inel sunt denumite "inele de comutator". Cele două bucăți ale inelelor de comutator sunt conectate la fiecare capăt al bobinei de armătură. Curentul direct al unei tensiuni adecvate este aplicat inelelor comutatorului prin două "perii" care se freacă de inele. Când DC este aplicat inelelor comutatorului, acesta curge prin bobina armăturii, producând un câmp magnetic. Acest câmp este atras de magnetul statorului și arborele motorului începe să se rotească. Arborele motor se rotește până ajunge la joncțiunea dintre cele două jumătăți ale comutatorului. În acest punct, periile vin în contact cu cealaltă jumătate a inelelor comutatorului, inversând polaritatea bobinei armăturii.Acest lucru este important deoarece, în acest moment, arborele motorului a fost rotit la 180 de grade și polaritățile câmpului magnetic trebuie inversate pentru ca motorul să continue să se rotească. Acest proces se repetă pe termen nelimitat până când curentul este îndepărtat din bobinele de armatură.

Motoarele fără perii utilizează un aranjament mai complex de bobine și nu necesită un comutator. Partea mobilă a motorului este conectată la magnetul permanent. Deoarece nu conțin perii, aceste motoare fără perii vor dura mai mult și sunt, de asemenea, mult mai silențioase decât motoarele DC periate. Motoarele cu curent continuu sunt specificate de nivelul de tensiune la care funcționează. Motoarele comune de pasionați rulează la 6 V sau 12 V DC. Pentru a inversa direcția în care motorul DC se rotește pur și simplu inversați polaritatea curentului DC pe care îl aplicați. Schimbarea vitezei cu toate acestea este o poveste diferită. O metodă de modificare a vitezei unui motor de curent continuu este de a reduce pur și simplu tensiunea de alimentare. În timp ce acest lucru va funcționa într-o oarecare măsură, nu este de fapt o metodă foarte bună de control al vitezei motorului, deoarece scăderea tensiunii va reduce, de asemenea, cuplul pe care motorul este capabil să îl producă. De asemenea, odată ce tensiunea scade sub un anumit punct, motorul nu se va roti deloc. O metodă mult mai bună de a controla motoarele cu curent continuu este utilizarea modulației lățimii pulsului sau a modului PWM.

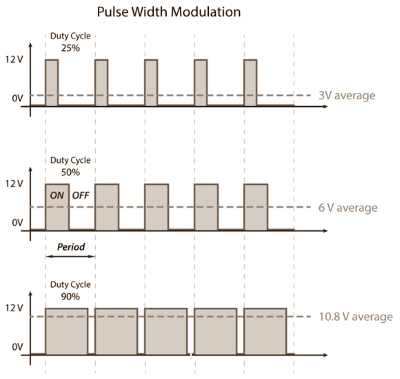


Un motor de curent continuu poate genera electricitate atunci când arborele se rotește în jurul acestuia. Când opriți alimentarea cu energie a unui motor, acesta va continua să se rotească, pentru că este inerție. Când se învârte, va genera o tensiune în direcția opusă față de curentul pe care l-ați dat. Un mod mai versatil de a controla un motor DC este de a folosi un circuit numit "H-Bridge". Un "H-Bridge" este un aranjament de tranzistori care vă permit să controlați atât direcția, cât și viteza motorului. Astăzi vom examina un modul foarte comun H-Bridge bazat pe IC L298N.

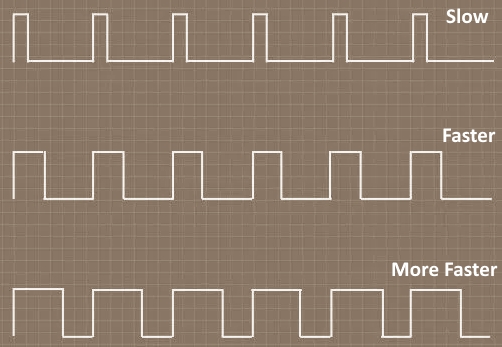
Înțelegerea modulului motorului L298N H-bridge :

H-Bridge sunt utilizate în mod obișnuit pentru controlul vitezei și direcției motoarelor, dar pot fi utilizate și pentru alte proiecte, cum ar fi luminozitatea anumitor proiecte de iluminat, cum ar fi magistralele cu LED-uri de mare putere. Un H-Bridge este un circuit care poate conduce un curent în ambele polarități și poate fi controlat prin **Pulse Width Modulation (PWM**).

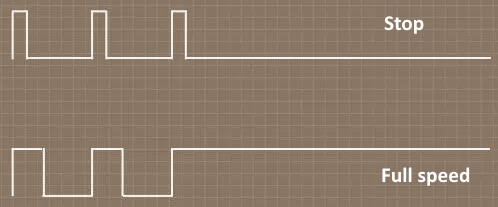
PWM este o tehnică care ne permite să ajustăm valoarea medie a tensiunii care se deplasează la dispozitivul electronic prin pornirea și oprirea alimentării la o viteză rapidă. Tensiunea medie depinde de ciclul de funcționare sau de timpul în care semnalul este pornit față de timpul în care semnalul este oprit într-o singură perioadă de timp.



Cu PWM motorul DC este trimis o serie de impulsuri. Fiecare puls este de tensiune completă pe care motorul o poate manevra, astfel încât un motor de 6 volți va fi trimis impulsuri de 6 volți în timp ce un motor de 12 volți va fi trimis impulsuri de 12 volți. Lățimea impulsurilor este variată pentru a controla viteza motorului, impulsurile cu o lățime îngustă vor determina motorul să se rotească destul de încet. Creșterea lățimii impulsului va crește viteza motorului, după cum se arată mai jos.



Pentru a opri complet motorul, opriți-va să-l pulsați, în esență trimițându-l cu zero volți. Pentru al rula la viteză maximă, trimiteți tensiunea completă, din nou fără să îl pulsați.



PWM, așa cum este descris mai sus.Activarea se face prin două switch-uri speciale, în același timp, putem schimba direcția fluxului de curent, astfel schimba direcția de rotație a motorului.

**L298N H-Bridge**

În timp ce puteți utiliza tranzistori pentru a construi un H-Bridge există o serie de avantaje în utilizarea unui circuit integrat. Un număr de IC-uri pentru motoarele H-Bridge sunt disponibile și toate funcționează în aceeași manieră. Una dintre cele mai populare este L298N. L298N este un membru al unei familii de IC care au toate denumirea "L298". Diferența dintre membrii familiei este în cantitatea actuală pe care o pot gestiona. L298N conține de fapt două circuite complete H-Bridge, astfel încât este capabil să conducă o pereche de motoare de curent continuu. Acest lucru îl face ideal pentru proiectele robotizate, deoarece majoritatea roboților au două sau patru roți motrice. L298N poate fi de asemenea utilizat pentru a conduce un singur motor pas cu pas.

**Funcționarea sistemului cu afișaj LCD (16x2 caractere) :**

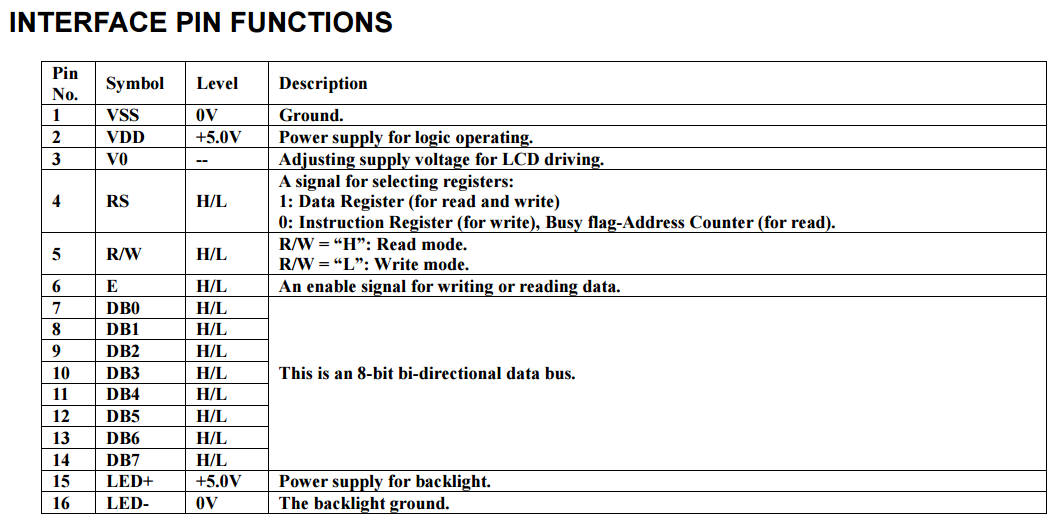
LCD-urile au o interfață paralelă, ceea ce înseamnă că microcontrolerul trebuie să manipuleze mai multe terminale de interfață dintr-o dată pentru a controla afișarea.

Modulele LCD sunt foarte des intalnite in proiectele embedded. Motivul este reprezentant de pretul scazut si de faptul ca acest modul este user-friendly in ceea ce priveste programarea sa. Pinii si terminalele vor fi descrise mai jos, acum vom enumera cateva detalii tehnice despre ecranul LCD.

**16x2 LCD** isi justifica numele prin cele 16 randuri si 2 coloane. Sunt foarte multe combinatii valabile, ca de exemplu 8x1, 8x2, 10x2, 16x1, dar cel mai folosit este 16x2.Deci ,vom avea 16x2 = 32 de caractere in total, fiecare caracter fiind constituit din 5x8 Pixel Dots.

Acum stim ca fiecare caracter are 5x8 =40 pixeli si pentru toate caracterele vom avea 32x40 =1280 pixels. Mai mult,ecranul LCD e capabil sa retina si pozitia fiecarui pixel. Prin urmare, va fi o sarcină grea ca ecranul LCD să se ocupe de tot cu ajutorul MCU, de unde se folosește o interfață IC, precum HD44780, care este montată pe partea din spate a modulului LCD. Funcția acestei IC este de a obține comenzile și datele de la MCU și de a le procesa pentru a afișa informații semnificative pe ecranul LCD.



Pinii ecranului LCD :

**Registru Selector (RS)**: in acest terminal se stocheaza locatia de memorie a ecranului LCD unde se vor scrie datele . Se poate selecta fie registrul de date, care reține ceea ce va apărea pe ecran, sau un registru de instrucțiuni, care este folosit atunci când controlerul ecranului LCD caută urmatoarele instrucțiuni.

**Citire/Scriere (R/W)**: terminal care selectează modul citire sau modul scriere

**Enable**: terminalul care permite scrierea în registre

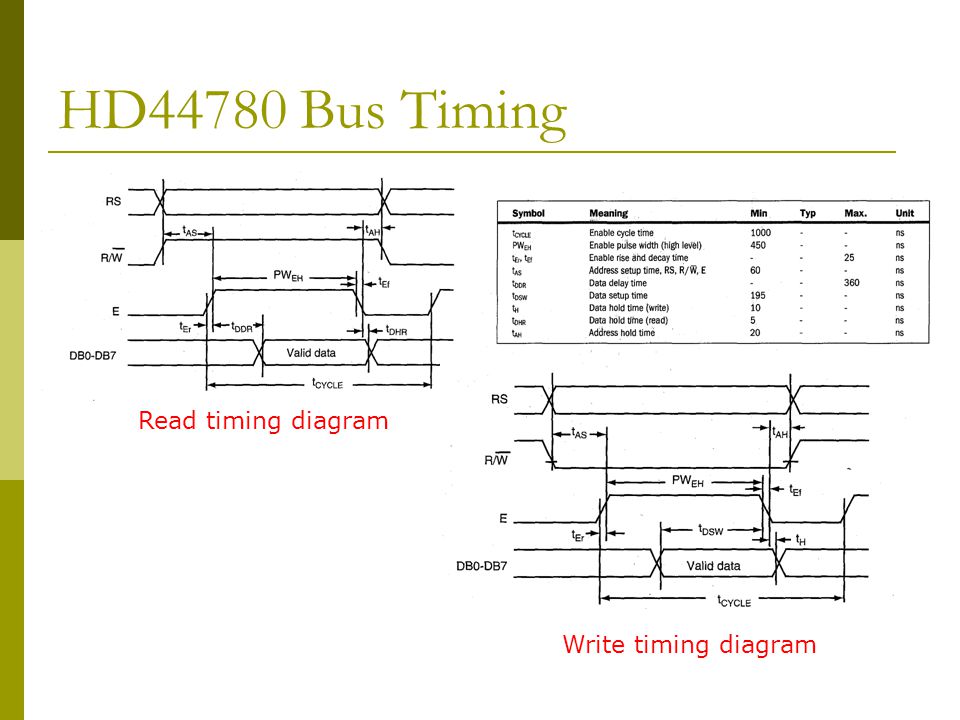
8 terminale de date **(D0-D7)**. Stările acestora (0 sau 1 logic) sunt biții care sunt scriși într-un registru atunci când lucrăm în modul scriere, sau valorile care au fost citite când lucrăm în modul citire.

Există, de asemenea, un terminal pentru constrastul LCD-ului (**Vo/VEE**), terminalul de alimentare (+5V și GND) și terminalele pentru LED-urile de iluminare din spate (Bklt+ și BKlt-).

Aceste terminale sunt utilizate pentru a alimenta LCD-ul, pentru a controla contrastul afișajului și pentru a activa și dezactiva iluminarea din spate a afisajului.

Procesul de control al afișajului implica punerea datelor care formeaza imaginea a ceea ce trebuie să fie afișat în registrele de date, apoi punerea instrucțiunilor în registrul de instrucțiuni.

LCD-urile cu driver Hitachi HD44780, sau LCD-urile compatibile cu acestea, pot fi controlate în două moduri: 4-biți sau 8 biți. Modul pe 4 biți necesită șapte terminale digitale de intrare/iesire din Arduino pentru a fi controlate, în timp ce modul de 8-biți necesită 11 terminale. Pentru afișarea textului pe ecran este suficient controlul în modul 4 biți.



# Senzor ultrasonic HC-SR04 :



Senzorul ultrasonic HC-SR04 este unul dintre cei mai utilizați senzori pentru aflarea distanței. În special folosit pentru proiectele cu plăci de dezvoltare Arduino, are avantaje față de senzorii analogici, necesitând doar pini I/O digitali și are imunitate mai mare la zgomotul din jur.  
  
 Senzorul emite ultrasunete la o frecvență de 40000Hz care circulă prin aer, iar dacă întâlnește un obstacol, acesta se va întoarce înapoi spre modul, astfel, luând in considerare viteza sunetului se poate calcula distanța până la obiect.

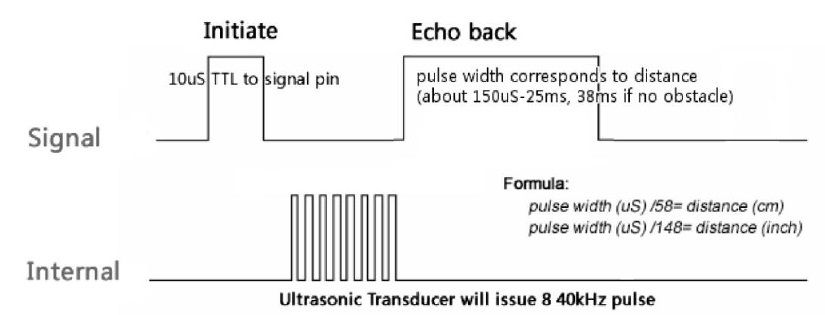
**Caracteristici tehnice:**

* Tensiune de alimentare: 5V
* Curent consumat: 15mA
* Distanță de funcționare: 2cm - 4m
* Unghi de măsurare: 15 grade
* Eroare de doar 3mm
* Durată semnal input: 10us
* Dimensiuni: 45mm x 20mm x 15mm

**Calculul distanței:**

Distanța= (timpul pe nivel înalt \* viteza sunetului (340 m/s) ) / 2

Distanța în centimetri = timpul pe nivel înalt / 58



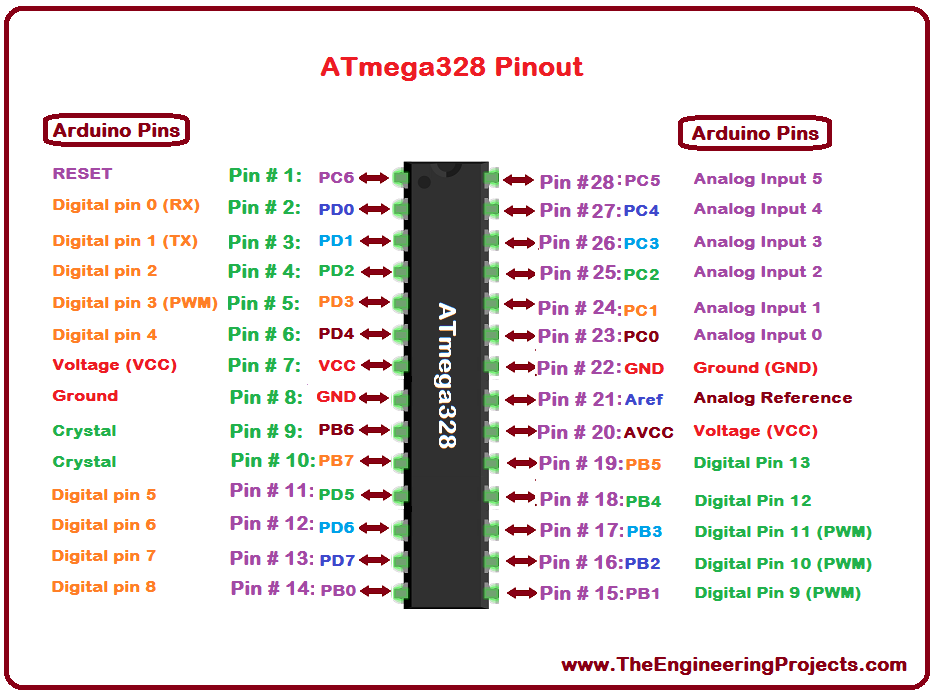
# Mini Difuzor Brick

Mini difuzorul-ul brick este un difuzor simplu pentru Arduino, care poate fi conectat direct pe unul dintre pinii PWM ai placii pentru a crea efecte sonore sau melodii.

**Atmega328:**

Atmega328 [AVR](https://ro.wikipedia.org/w/index.php?title=Atmel_AVR&action=edit&redlink=1) [8-bit](https://ro.wikipedia.org/wiki/8-bit) este un [circuit integrat](https://ro.wikipedia.org/wiki/Circuit_integrat) de înaltă performanță ce se bazează pe un microcontroler [RISC](https://ro.wikipedia.org/w/index.php?title=RISC&action=edit&redlink=1), combinând 32 KB [ISP](https://ro.wikipedia.org/w/index.php?title=In-system_programming&action=edit&redlink=1) [flash](https://ro.wikipedia.org/w/index.php?title=Memoria_Flash&action=edit&redlink=1) o memorie cu capacitatea de a citi-în-timp-ce-scrie, 1 KB de memorie [EEPROM](https://ro.wikipedia.org/wiki/EEPROM), 2 KB de [SRAM](https://ro.wikipedia.org/w/index.php?title=Static_Random_Access_Memory&action=edit&redlink=1), 23 linii E/S de uz general, 32 [Înregistrari](https://ro.wikipedia.org/w/index.php?title=%C3%8Enregistrare_(hardware)&action=edit&redlink=1) procese generale, trei cronometre flexibile/[contoare](https://ro.wikipedia.org/w/index.php?title=Contoare&action=edit&redlink=1) în comparație cu, [întreruperi](https://ro.wikipedia.org/w/index.php?title=%C3%8Entreruperi&action=edit&redlink=1) internă și externă, programator de tip [USART](https://ro.wikipedia.org/w/index.php?title=USART&action=edit&redlink=1), orientate interfață serială byte de 2 cabluri, [SPI](https://ro.wikipedia.org/w/index.php?title=Serial_Peripheral_Interface_Bus&action=edit&redlink=1) port serial, 6-canale 10-bit [Converter A/D](https://ro.wikipedia.org/w/index.php?title=Converter_A/D&action=edit&redlink=1) (8-canale în [TQFP](https://ro.wikipedia.org/w/index.php?title=TQFP&action=edit&redlink=1) și [QFN](https://ro.wikipedia.org/w/index.php?title=QFN&action=edit&redlink=1)/[MLF](https://ro.wikipedia.org/w/index.php?title=Quad-flat_no-leads_package&action=edit&redlink=1) packages), "watchdog timer" programabil cu [oscilator](https://ro.wikipedia.org/w/index.php?title=Oscilator&action=edit&redlink=1) intern, și cinci moduri de software-ul intern de economisire a energiei selectabil. Dispozitivul funcționează 1,8-5,5 volți.

Prin executarea instrucțiuni puternice într-un singur ciclu de ceas, aparatul realizează un răspuns de 1 [MIPS](https://ro.wikipedia.org/w/index.php?title=MIPS&action=edit&redlink=1)

****

|  |  |
| --- | --- |
| **Parametri** | **VALORI** |
| Flash | 32 Kbytes |
| RAM | 2 Kbytes |
| Cantitate Pini | 28 |
| Frecvență maximă de funcționare | 20 MHz |
| CPU | 8-bit AVR |
| Numărul de variabile Canale | 16 |
| Pini maximi de E/S | 26 |
| Întreruperile externe | 24 |

**Secventele de cod:**

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(0,1,2,3,4,5); // creaza o variabila de tip LiquidCrystal, aceasta are rolul de a conecta pinii aceia la pinii de pe ecranul LCD folosim 8 data reg

// Partea dreapta - Motor A

//int ENA =6;

int IN1=8;

int IN2=9;

// partea stanga Motor B

//int ENB = 7;

int IN3 = 10;

int IN4= 11;

//butoaneControl

//int butonFata = 13;

int butonSpate = 12;

const int distantaPericol =7;

//controlSenzor

const int trigPin1 = 6;

const int echoPin1 = 7;

const int trigPin2 = 4;

const int echoPin2 = 5;

long duration1;

int distance1;

long duration2;

int distance2;

void setup() {

// set pin modes

lcd.begin(16, 2);// initializare lcd cu 16 coloane si 2 randuri

lcd.print("hello, world!");

Serial.begin(9600);

// pinMode(butonFata, INPUT);

pinMode(trigPin1, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin1, INPUT); // Sets the echoPin as an Input

pinMode(trigPin2, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin2, INPUT); // Sets the echoPin as an Input

pinMode(butonSpate, OUTPUT);

pinMode(IN1 , OUTPUT);

pinMode(IN2 , OUTPUT);

pinMode(IN3 , OUTPUT);

pinMode(IN4 , OUTPUT);

// pinMode(ENA , OUTPUT);

// pinMode(ENB , OUTPUT);

}

void startCar()

{

digitalWrite( IN1 , HIGH);

digitalWrite( IN2 , LOW);

digitalWrite( IN3 , HIGH);

digitalWrite( IN4 , LOW);

}

void reverseCar()

{

digitalWrite( IN1 , LOW);

digitalWrite( IN2 , HIGH);

digitalWrite( IN3 , LOW);

digitalWrite( IN4 , HIGH);

}

void stopCar()

{

digitalWrite( IN1 , LOW);

digitalWrite( IN2 , LOW);

digitalWrite( IN3 , LOW);

digitalWrite( IN4 , LOW);

}

void stanga(){

digitalWrite( IN1 , HIGH);

digitalWrite( IN2 , LOW);

digitalWrite( IN3 , LOW);

digitalWrite( IN4 , LOW);

}

void dreapta(){

digitalWrite( IN1 , LOW);

digitalWrite( IN2 , LOW);

digitalWrite( IN3 , HIGH);

digitalWrite( IN4 , LOW);

}

void loop() {

Serial.println("fata");

//Serial.print(digitalRead(butonFata));

//delay(500);

Serial.println("spate");

Serial.print(digitalRead(butonSpate));

//delay(500);

if(senzor1() < distantaPericol)// daca senzorul masoare mai putin decat distanta periculoasa atunci sa se opreasca masina{

stopCar();

}

else if(analogRead(0)>1000 && senzor1() > distantaPericol)//daca citirea pinului 0 e mai mare ca 1000 si senzorul e nu e - atunci start decizie de compormis altfel pe ce era mai jos pe butonspate il punea la fel casi pe butonfata si anume pe 1{

startCar();

}

else if(digitalRead(butonSpate) == 1 && senzor1() > distantaPericol){

reverseCar();

}

else if(digitalRead(butonSpate)== 0 && analogRead(0) <900 && senzor1() < distantaPericol){

stopCar();

}

senzor1();

lcd.setCursor(0, 1);

lcd.print(senzor1());

//if(digitalRead(butonSpate) == 1){

// // startCar();

//reverseCar();

// }else{

// stopCar();

// }

}

int senzor1(){

// Clears the trigPin

digitalWrite(trigPin1, LOW);

delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin1, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin1, LOW);

// Reads the echoPin, returns the sound wave travel time in microseconds

duration1 = pulseIn(echoPin1, HIGH);

// Calculating the distance

distance1= duration1\*0.034/2;

// Prints the distance on the Serial Monitor

Serial.print("Distance: ");

Serial.println(distance1);

return distance1;

}

**Bibliografie**

**[Placa Arduinio Uno-R3]**

<https://www.optimusdigital.ro/ro/placi-avr/656-uno-r3-atmega328p-atmega16u2-placa-de-dezvoltare-compatibila-cu-arduino-cablu-30-cm.html?search_query=uno&results=146&fbclid=IwAR2I4Fg3unPDuSeHcMbbBfCJzBUVB7eFKeETaKTE2VS1i9hx5ntV_HAd6v4>

**[Placa Arduinio Uno-R3]**

<https://www.robofun.ro/arduino_uno_v3>

**[Placa Arduinio Uno-R3]**

<https://store.arduino.cc/arduino-uno-rev3>

# [Senzor ultrasonic HC-SR04]

<https://cleste.ro/senzor-ultrasonic-hc-sr04.html>

**[Ecranul LCD]**

<https://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet>

# [16x2 LCD Module Datasheet]

<https://components101.com/16x2-lcd-pinout-datasheet>

**[Atmega328]**

<https://ro.wikipedia.org/wiki/Atmega328>

# [L298N Dual H-Bridge Motor Driver Module]

<http://acoptex.com/project/162/basics-project-033a-l298n-dual-h-bridge-motor-driver-module-one-or-two-dc-motors-6v-at-lex-c/#sthash.BQoLuD82.dpbs>

**[Tool pentru desenarea circutului]**

<http://fritzing.org/home/>

**[Schema pentru Atmega328]**

<https://www.theengineeringprojects.com/>

<https://github.com/impulse1997/Project-Si>