

# Redare muzica pe baza senzorului de mișcare

## Cuprins

- 1. Descriere proiect**
- 2. Componente**
- 3. Partea practică**
- 4. Bibliografie**

### **1. Descriere proiect:**

Scopul proiectului este de a reda muzică pe baza unui senzor de mișcare. Liceul Teoretic „Constantin Șerban” – Aleșd este beneficiarul acestui proiect și va fi pus în holul liceului. Pentru a adăuga un colț de muzică în liceu, conducerea împreună cu prof. Ionomi Flavia au propus această idee de proiect la care am fost bucuroși să participăm.

Am folosit doi senzori ultrasonici HC-SR04 pentru detectarea mișcării, iar redarea muzicii s-a realizat printr-un modul MP3 YX5300. Aceștia au fost conectați prin intermediul Arduino UNO pentru a se crea legătura între aceștia.

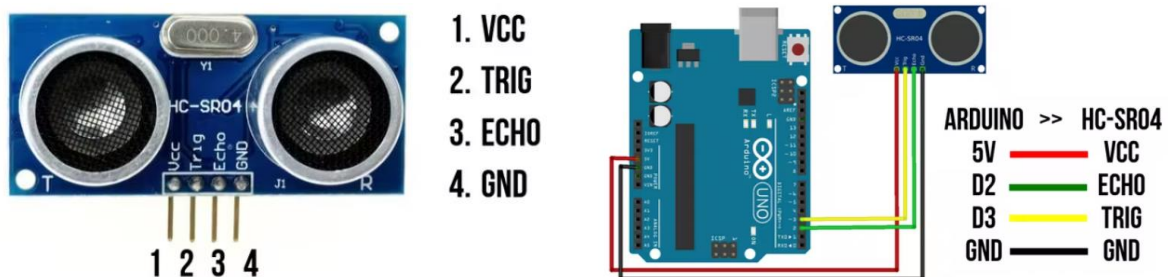
Plan de funcționare: pornirea redării muzicii la detectarea mișcării pe coridor și oprirea acesteia la detectarea altei mișcări. Pentru aceasta s-au folosit doi senzori (Senzor 1 pentru pornire și Senzor 2 pentru oprire), un card SD cu muzică. Acesta conține fișiere cu muzică pe diferite teme, care se accesează în funcție de perioada anului școlar.

### **2. Componente:**

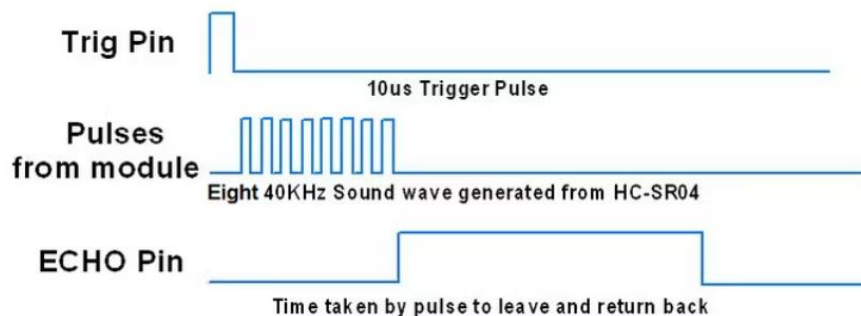
- 2 X Ultrasonic Sensor HC-SR04
- Modul MP3 YX5300
- Breadboard
- Ardiono UNO
- Fire

**Senzorul HC-SR04** este un senzor care poate măsura distanța. Emite ultrasunete la 40.000 Hz care se deplasează prin aer și dacă întâlnește un obiect în fața lui, va întoarce răspuns la modul. Astfel se poate calcula distanța, folosindu-ne de timpul de deplasare și viteza sunetului.

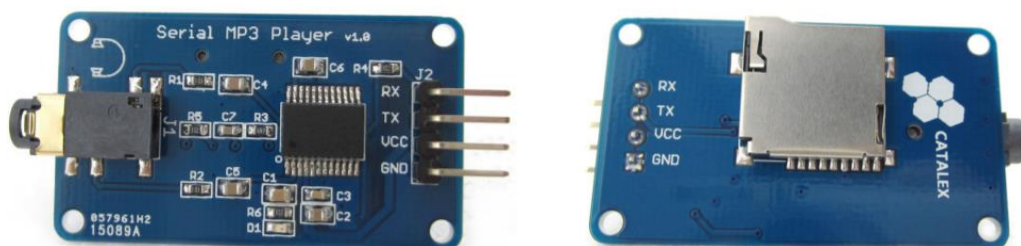
Pinii sunt VCC, TRIG, ECHO și GND. Tensiunea de alimentare a VCC este de +5V și puteți atașa pinul TRIG și ECHO la orice I/O digitală din placa Arduino.

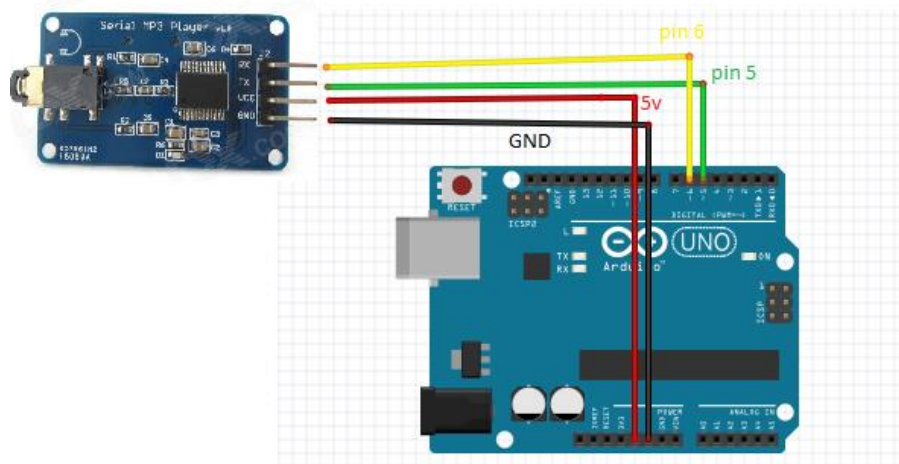


Pentru a genera ultrasunetele, trebuie să setăm pinul de declanșare pe o stare de 10  $\mu$ s. Aceasta va trimite un impuls cu 8 cicluri care va fi recepționat în Echo Pin. Echo Pin-ul va emite timpul în microsecunde pe care unda sonoră a parcurs.



**Modul MP3 YX5300** este un dispozitiv de redare MP3 și suportă formatele MP3 și WAV cu frecvența de eșantionare de 8k Hz ~ 48k Hz.





Pe acest modul se află un slot pentru cardul TF, unde se poate introduce cardul micro SD care stochează fișierele audio. MCU poate controla starea redării MP3 prin trimiterea de comenzi către modul prin portul UART, cum ar fi schimbarea melodiilor, schimbarea volumului și modul de redare.

#### Caracteristici:

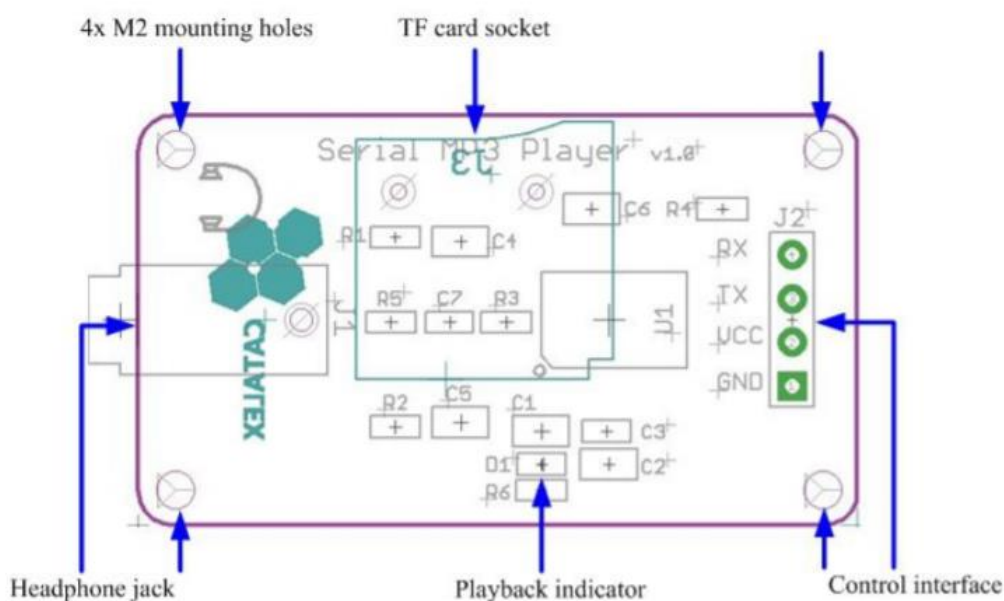
- Acceptă frecvența de eșantionare (kHz): 8 / 11.025 / 12 / 16 / 22.05 / 24 / 32 / 44.1 / 48
- Întăltă calitate
- Suport format de fișier: MP3 / WAV
- Suport card: Micro SD, Card Micro SDHC
- Volum reglabil: clasa 30
- Modul de redare: cu control serial UART TTL, rata de transmisie este de 9600 bps
- Sursa de alimentare: poate fi 3.2 ~ 5.2VDC
- Interfața logică de control: poate fi 3.3V / 5V TTL
- Compatibil cu: Arduino UNO / Leonardo / AVR / ARM / PIC / Mega2560 / DUE.

#### Comenzi pentru modulul MP3:

Comandă	Octeți de comandă (HEX)	Rezultat
Piesa următoare	7E FF 06 01 00 00 00 EF	Pornire piesa următoare
Piesa anterioară	7E FF 06 02 00 00 00 EF	Pornire piesa anterioară
Piesa anume	7E FF 06 03 00 00 01 EF	Pornire prima piesă
	7E FF 06 03 00 00 02 EF	Pornire a doua piesă
Mărire volum	7E FF 06 04 00 00 00 EF	Mărire volum
Micșorare volum	7E FF 06 05 00 00 00 EF	Micșorare volum
Setare volum	7E FF 06 06 00 00 1E EF	Setare volum la 30
Pornire un ciclu	7E FF 06 08 00 00 01 EF	Pornește ciclu la prima melodie
Selectare dispozitiv	7E FF 06 09 00 00 02 EF	Selectează stocarea de pe card
Adormire	7E FF 06 0A 00 00 00 EF	Chipul intră în modul adormire
Trezire	7E FF 06 0B 00 00 00 EF	Chipul se trezește
Resetare	7E FF 06 0C 00 00 00 EF	Chipul se resetează

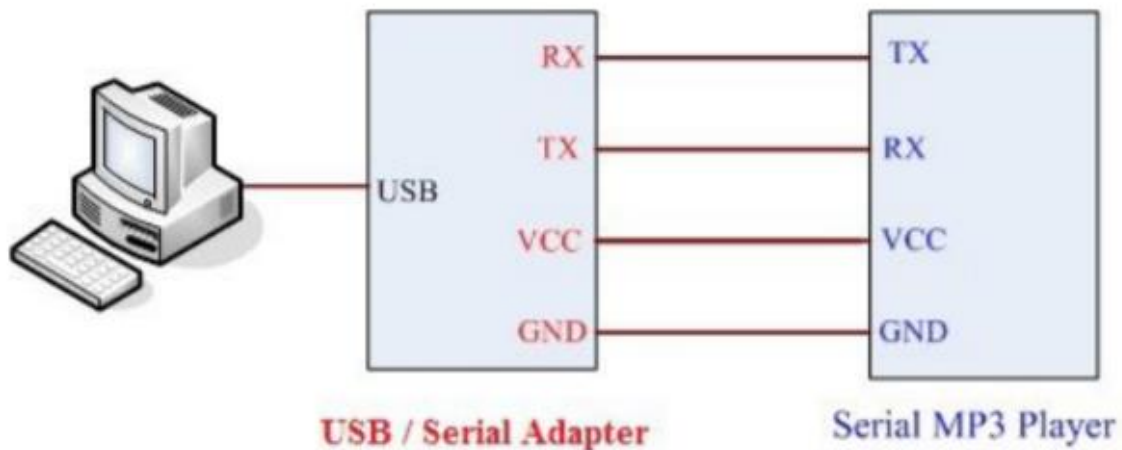
Pornire	7E FF 06 0D 00 00 00 EF	Reia redarea
Pauză	7E FF 06 0E 00 00 00 EF	Redarea intră în pauză
Pornire din fișier	7E FF 06 0F 00 01 01 EF 7E FF 06 0F 00 01 02 EF	Pornește din fișierul 1, piesa 1 Pornește din fișierul 1, piesa 2
Opre	7E FF 06 16 00 00 00 EF	Oprește redarea
Pornire folder în ciclu	7E FF 06 17 00 01 02 EF	Pornește ciclu la primul folder
Setarea ciclului de pornire	7E FF 06 19 00 00 00 EF 7E FF 06 19 00 00 01 EF	Pornește ciclul Oprește ciclul
Setează DAC	7E FF 06 1A 00 00 00 EF 7E FF 06 1A 00 00 01 EF	Pornește ieșirea DAC Fără ieșirea DAC
Setează volumul	7E FF 06 22 00 1E 01 EF 7E FF 06 22 00 0F 02 EF	Setare volum la 30 Setare volum la 15

Interfața:



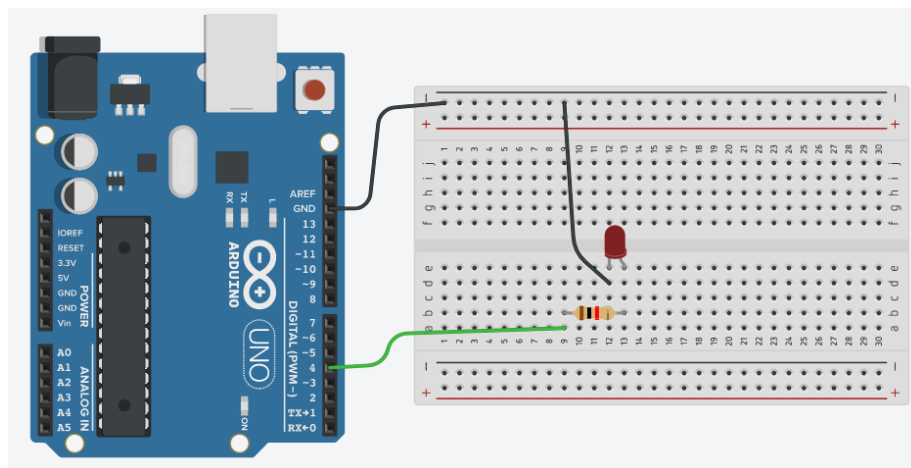
- Interfața de control: este interfața UART TTL. Un total de patru pini (GND, VCC, TX, RX), GND la pământ, VCC este sursa de alimentare.
- Priză card TF: cardul micro SD poate fi conectat la el.
- Indicator de redare: lumină verde. Dacă este gata de redare sau este întreruptă, rămâne aprinsă. Dacă este în redare, clipește.
- Mufă pentru căști: poate fi conectată la căști sau la un amplificator extern.
- Găuri de montare: 4 găuri de montare, astfel încât modulul să fie ușor de instalat sau de combinat cu alte module.

Este nevoie de un modul USB la Uart TTL (cum ar fi un adaptor USB/serial) pentru a vă conecta Player MP3 serial pe PC. Instalarea hardware se face în felul următor:



## Breadboard

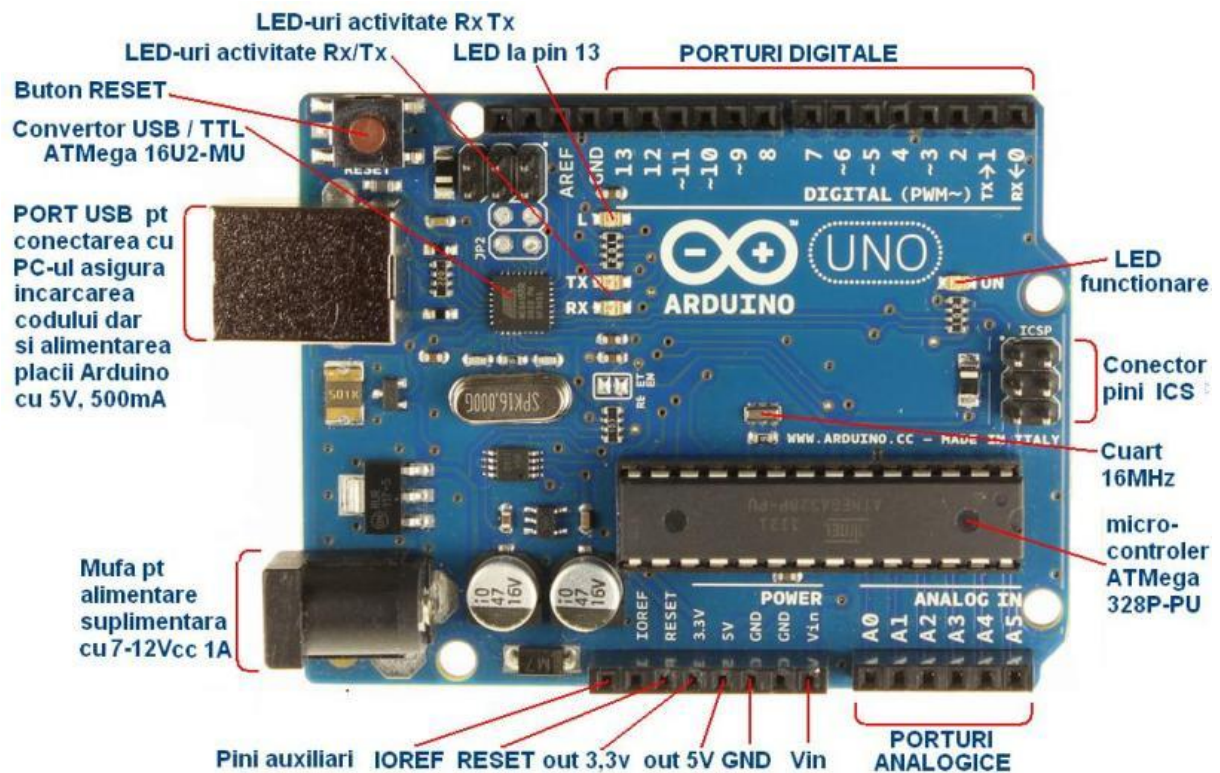
Un breadboard este un dispozitiv simplu conceput pentru a vă permite să creați circuite fără a fi nevoie de lipire.



## Ardiono UNO

Arduino UNO constituie o platforma de procesare tip open-source, bazata pe un software si hardware flexibil construita in jurul unui microcontroler ATMEGA 328P-PU capabila de a prelua date printr-o serie de senzori conectati la pinii placii si de a actiuna asupra altor dispozitive ca LED-uri, motoarelor, servomotoare, sau alte tipuri de dispozitive mecanice pe baza unor comenzi cuprinse in codul scris intr-un limbaj de programare, similar cu limbajul C++ incarcat in memoria microcontrolerului.





### 3. Parte practică:

// includem librăria SoftwareSerial pentru modulul MP3

#include <SoftwareSerial.h>

// conectare la pini serial

#define ARDUINO\_RX 8//should connect to TX of the Serial MP3 Player module

#define ARDUINO\_TX 9//connect to RX of the module

// conectare senzori ultrasonici

int trigPin1=2;

int echoPin1=3;

int trigPin2=4;

int echoPin2=5;

// inițializare contor

int contor=0;

// initializam pinii la care am conectat modulul MP3

SoftwareSerial mySerial(ARDUINO\_RX, ARDUINO\_TX);

//initializare unui vector pentru comunicarea HEXA cu modulul MP3 serial

static int8\_t Send\_buf[8] = {0};

// definirea macro-definițiilor pentru byte-ul de comandă cu modulul MP3

#define NEXT\_SONG 0X01

#define PREV\_SONG 0X02

#define CMD\_PLAY\_W\_INDEX 0X03

#define VOLUME\_UP\_ONE 0X04

#define VOLUME\_DOWN\_ONE 0X05

#define CMD\_SET\_VOLUME 0X06

#define SET\_DAC 0X17

#define CMD\_PLAY\_WITHVOLUME 0X22

#define CMD\_SEL\_DEV 0X09

#define DEV\_TF 0X02

#define SLEEP\_MODE\_START 0X0A

#define SLEEP\_MODE\_WAKEUP 0X0B

#define CMD\_RESET 0X0C

#define CMD\_PLAY 0X0D

#define CMD\_PAUSE 0X0E

#define CMD\_PLAY\_WITHFOLDER 0X0F

#define STOP\_PLAY 0X16

#define PLAY\_FOLDER 0X17

#define SET\_CYCLEPLAY 0X19

#define SET\_DAC 0X17

// setări inițiale pentru a putea comunica cu modulul MP3

```

void setup() {
  Serial.begin (9600);
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);

  mySerial.begin(9600);           //pornirea modulului serial MP3
  delay(500);                     //Introducerea unei întârzieri de 5 secunde
  sendCommand(CMD_SEL_DEV, DEV_TF); //selectarea cardului TF
  delay(200);
}

```

//rularea programului – în această secțiune se va calcula distanța în cm pentru primul senzor

```

void loop() {
  long duration1, distance1;
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distance1 = (duration1/2) / 29.1;
}

```

// Dacă senzorul detectează un obiect la o distanță de 0 – 25 cm în fața primului senzor, iar contorul are valoare pară, atunci modulului de MP3 serial i se va trimite comanda de SLEEP\_MODE\_WAKE\_UP, care va porni chipul de pe modul. Se introduce o întârziere după pornirea modulului. După ce modulul a pornit, se va seta volumul la maxim (valoarea maximă este 30), iar apoi se va trimite o comanda pentru a rula toate melodiile dintr-un fișier specific. Introducem o întârziere de 10 secunde, iar înainte să ieșim din instrucțiunea IF vom crește contorul pentru a fi impar. Acesta ne ajută în cazul în care în fața senzorului va mai exista alt obiect între 0 – 25 cm, atunci instrucțiunea IF nu va mai intra pe prima ramură și nu se va mai repeta comanda de redare a melodiilor din folder.



//Vom observa, comanda de rulare a melodiilor va rula doar dacă modulul primește înainte o comandă de oprire rulare melodii.

```
if (distance1 <= 25 && distance1 >= 0 && contor%2==0){  
    Serial.print ( "Sensor1 ");  
    Serial.print ( distance1);  
    Serial.println("cm");  
    sendCommand(SLEEP_MODE_WAKEUP, 0X0000);  
    delay(1000);  
    sendCommand(CMD_SET_VOLUME, 0X001E);  
    sendCommand(PLAY_FOLDER, 0X0102);  
    delay(10000);  
    contor=contor+1;  
} else {  
    Serial.println("Out of range"); }
```

// se va calcula distanța în cm pentru al doilea senzor.

Primul senzor – pornire. Al doilea – oprire.

```
delay(2000);  
long duration2, distance2;  
digitalWrite(trigPin2, LOW);  
delayMicroseconds(2); // Added this line  
digitalWrite(trigPin2, HIGH);  
delayMicroseconds(10); // Added this line  
digitalWrite(trigPin2, LOW);  
duration2 = pulseIn(echoPin2, HIGH);  
distance2= (duration2/2) / 29.1;
```

// instrucțiunea IF va rula prima ramură dacă în fața senzorului 2 există un obiect în intervalul 0 – 25 cm, iar contorul va fi impar. Dacă e impar, conturul de MP3 rulează melodii. În cazul în care vom intra pe prima ramură, se va trimite o comandă de STOP\_PLAY care va opri rulare melodiilor, contorul va fi crescut și va avea o valoare pară, iar pentru a proteja modulul MP3 i se va trimite o comandă SLEEP\_MODE\_START pentru a intra în stare de adormire.

```
if (distance2 <= 25 && distance2 >= 0 && contor%2!=0){
```

```

Serial.print("Sensor2 ");
Serial.print(distance2);
Serial.println("cm");
sendCommand(STOP_PLAY, 0X0000);
contor=contor+1;
delay(5000);
sendCommand(SLEEP_MODE_START, 0X0000);
delay(1000);
}
else {
    Serial.println("Out of range");
}
delay(2000);
}

```

// Această funcție a fost creată pentru a trimite mai ușor comenzile către modulul MP3 Serial.  
 În această funcție se va inițializa și se va trimite byte cu byte către modulul MP3.

```

void sendCommand(int8_t command, int16_t dat) {
    delay(20);
    Send_buf[0] = 0x7e;
    Send_buf[1] = 0xff;
    Send_buf[2] = 0x06;
    Send_buf[3] = command;
    Send_buf[4] = 0x00;
    Send_buf[5] = (int8_t)(dat >> 8);
    Send_buf[6] = (int8_t)(dat);
    Send_buf[7] = 0xef;
    for(uint8_t i=0; i<8; i++)
    {
        mySerial.write(Send_buf[i]) ;
        Serial.print(Send_buf[i],HEX);
    }
}

```

```
Serial.println();  
}
```

