

fritzing

```
#include "servo_control.h"    // add custom header file to the program
#include "display.h"
#include "bluetooth.h"
#include "RGB.h"
#include "buz.h"
#include "rfid.h"
#include "setup.h"
#include "road.h"
```

```
void setup()
{
    setup1();
}
```

```
void loop() {
    char m = bluetooth(); //read data from hc05
    Serial.println(m);
```

```
    if(m == 'A' || m == 'B') k = 0;
```

```
    → if(m == 'M') k = 1;
```

'M' - manual access

```
    if(m == 'A'){
        road1();
        m = '0';
    }
```

// change the variable so that  
it can't conflict next same  
input value.

```
    if(m == 'B'){
        road2();
        m = '0';
    }
```

```
    if(m == 'M' || k == 1){
        manual();
    }
```

// continuously need Analog input  
from ~~se~~ Joystick

```
}
```

Software serial.h - Allow Atmega chip  
receive serial communication even while  
on other task as long as there  
room for 64 bytes serial buffer.

```

/*****RGB.h*****/
#define bluePin 4
#define greenPin 7
#define redPin 8

void red(){
    digitalWrite (redPin,HIGH);
    delay(1000);
    digitalWrite (redPin,LOW);
}
void green(){
    digitalWrite (greenPin,HIGH);
    delay(1000);
    digitalWrite (greenPin,LOW);
}
void yel(){
    digitalWrite (bluePin,HIGH);
    digitalWrite (greenPin,HIGH);
    delay(1000);
    digitalWrite (bluePin,LOW);
    digitalWrite (greenPin,LOW);
}
void redOn(){
    digitalWrite (redPin,HIGH);
}
void redOff(){
    digitalWrite (redPin,LOW);
}

/*****bluetooth.h*****/
#include<SoftwareSerial.h>
#define TxD 3
#define RxD 2
char c;
SoftwareSerial bluetoothSerial(TxD, RxD);           //declare hc05 object

char bluetooth(){}

```

SPI - serial peripheral Interface  
used to send data between microcontroller  
and small peripheral.

```

if( bluetoothSerial.available() ){
    c = bluetoothSerial.read();
    return c;
}
}

/*****buz.h*****/
#define buzzer 6
int i = 1000;
void alarm (){

    tone(buzzer, i, 500);
    delay(1000);
    digitalWrite(buzzer,0);

}

/*****display.h*****/
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F,12,2);

void display(int x, int y , String s) {
    lcd.setCursor(x,y);
    lcd.print(s);
    delay(100);

}

/*****rfid.h*****/
#include <SPI.h>
#include<MFRC522.h>
#define RST_PIN 9
#define SS_PIN 10
bool id = false;

MFRC522 mfrc522 (SS_PIN, RST_PIN);    //declare rfid objet

void rfid() {

```



```

Serial.println();
if( ! mfrc522.PICC_IsNewCardPresent())
{
    return;
}
if( ! mfrc522.PICC_ReadCardSerial()) // If rfid can not read card info
{
    return;
}
Serial.print("UID tag :");
String content = "";
byte letter;

for(byte i = 0; i < mfrc522.uid.size;i++)
{
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(mfrc522.uid.uidByte[i], HEX));
}
Serial.println();
Serial.println("Message : ");
content.toUpperCase();

if(content.substring(1) == "C5 27 69 67")
{
    Serial.println("Authorized access");
    Serial.println();
    lcd.clear();
    display(0,0,"Authorized access"); // display
    green(); // rgb
    id = true; // confirmation
}

else

```

```

{
  Serial.println(" Access denied\n");
  lcd.clear();
  display(0, 0, "Access denied" );
  red();
  alarm();
  id = false;
}
delay(1000);
}

```

// buzzer

```

void rfidCheck() {
  Serial.println("Put your card to the reader ....");
  display(0,0,"Put your card");
  delay(3000);

```

```

while ( id != true ) {
  rfid();
}

```

// repeat until valid card press

```

delay(1000);
lcd.clear();
display(0,0,"Enter Command");
return;
}

```

```

/*****road.h *****/

```

```

int k;

```

```

void road1(){
  lcd.clear();
  display(0,0,"Road 1 clear");
  alarm();
  redOff();
  green();
  servo2.write(170);
  delay(500);
  servo1.write(20);
}

```



```

    delay(500);
    servo2.write(80);
    delay(500);
    redOn();
}

```

```

void road2() {
    lcd.clear();
    display(0,0,"Road 2 clear");
    alarm();
    redOff();
    green();
    servo2.write(170);
    delay(500);
    servo1.write(100);
    delay(500);
    servo2.write(80);
    delay(500);
    redOn();
}

```

```

void manual(){

```

```

    lcd.clear();
    display(0,0,"Manual Access");
    //alarm();
    servo();
}

```

```

/*****servo_control.h*****/

```

```

#include<Servo.h>

```

```

Servo servo1;

```

```

Servo servo2;

```

```

void servo ()

```

```

{

```

```

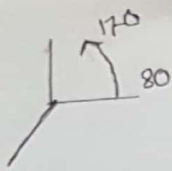
    int Jy = analogRead( A2 );    //read analog input from joystic

```

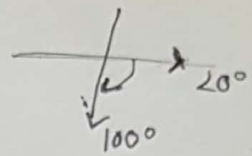
```

    int Jx = analogRead( A3 );

```



Servo 1 -  $20^{\circ}$  -  $100^{\circ}$   
Servo 2 -  $80^{\circ}$  -  $170^{\circ}$



```
int s1 = map (Jx, 1000 , 0, 20, 100 ); //convert the value which degree needed  
int s2 = map (Jy , 1023, 0, 80 , 170);
```

```
Serial.print (s1);  
Serial.print (" ");  
Serial.println(s2);  
servo1.write(s1); // implement the degree to servo  
servo2.write(s2);  
delay(10);
```

```
}  
/*****"setup.h"*****/
```

```
void setup1() {  
    /// RFID initialization  
    SPI.begin();  
    mfrc522.PCD_Init();
```

```
    /// RGB initialization  
    pinMode( redPin , OUTPUT);  
    pinMode( greenPin ,OUTPUT);  
    pinMode( bluePin ,OUTPUT);  
    pinMode( buzzer ,OUTPUT);
```

```
    /// display initialization  
    lcd.init();  
    lcd.clear();  
    lcd.backlight();
```

```
    /// servo motor initialization  
    servo2.attach( A1 );  
    servo1.attach( A0 );
```

```
    /// Bluetooth module initialization  
    bluetoothSerial.begin(9600);  
    Serial.begin(9600);  
    rfidCheck();  
}
```