

نیم پروژه درس آزمایشگاه ریزپردازنده و زبان اسمبلی

آرمان حاتمی و امیرحسین باریکلو

شرح آزمایش :

ما در این نیم پروژه تصمیم گرفتیم ماشین کنترلی را انتخاب و پیاده سازی کنیم

برای این آزمایش به ابزار زیر نیاز داشتیم :

1. برد آردوینو 2560
2. led به مقدار لازم
3. سروو موتور
4. آرمیچر
5. رله
6. بازر
7. ترانزیستور
8. دیود
9. ترمینال مجازی

چراغ های ماشین :

برای این قسمت از 2 led استفاده کردیم و آن ها را سری کردیم و به مقاومت و سپس به پین 22 برد وصل کردیم اگر میخواستیم led ها را روشن کنیم پس از وارد کردن حرف "i" در ترمینال این پین در آردوینو به high میرود و روشن میشوند و تا زمانی که ما دوباره این حرف را وارد ترمینال مجازی نکرده ایم خاموش نمیشوند

بوق ماشین :

برای بوق ماشین از یک buzzer استفاده کردیم که یک سر آن به ground و یک سر آن به پین 23 برد وصل است و اگر میخواستیم بوق بزنیم حرف "b" را وارد میکردیم که بعد از یک بوق کوتاه دوباره این پین low میشود

چرخ های جلو ماشین :

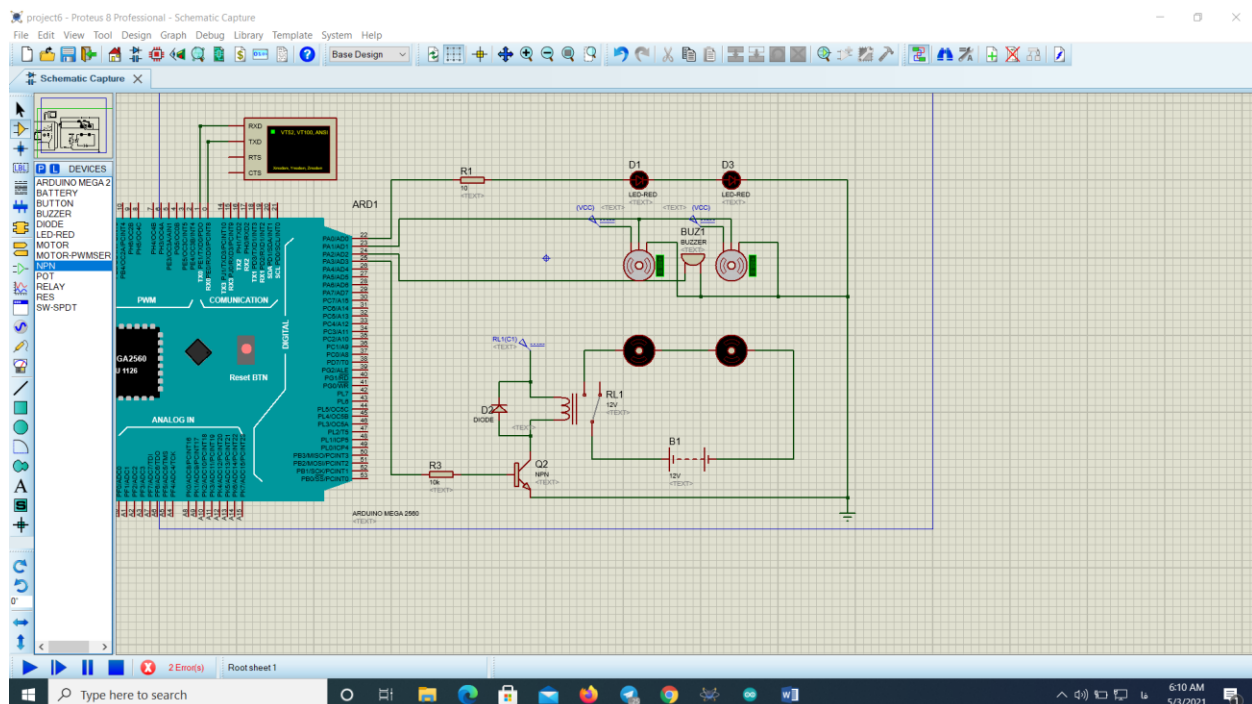
برای این قسمت از 2 عدد سروو موتور استفاده کردیم که سر هردوی آن ها به پین 24 برد وصل است برای چرخاندن آن ها منتظر میمانیم کاربر یک عدد وارد کند و سپس enter بزند تا به زاویه مورد نظر کاربر برویم در ابتدا در زاویه قائم 90 درجه قرار داریم اگر بخواهیم به سمت چپ بپیچیم کاربر باید عدد 0 تا 180 را وارد کند و برای اینکه بخواهیم به سمت راست بپیچیم کاربر باید عدد 180 تا 360 را وارد کند تا نسبت به بزرگی عدد وارد شده توسط کاربر به آن سمت بچرخیم اما امکان ندارد زاویه چرخ ها از 0 کمتر یا از 180 بیشتر شود زیرا از حالت منطقی و واقعی دور میشود

چرخ های عقب ماشین :

برای این قسمت از 2 عدد موتور و یک رله و یک ترانزیستور و یک دیود و یک باتری استفاده میکنیم
مانند آزمایش شماره 5 موتور ها و رله و دیود و ترانزیستور و باتری را به هم وصل میکنیم و در نهایت سیم ورودی به ترانزیستور
را به پین 25 وصل میکنیم برای روشن کردن چرخ ها و به حرکت درآمدن ماشین باید حرف "t" مخفف true را وارد کنیم و
برای متوقف کردن آن ها باید حرف "f" مخفف false را در ترمینال مجازی وارد کنیم

ماشین نهایی :

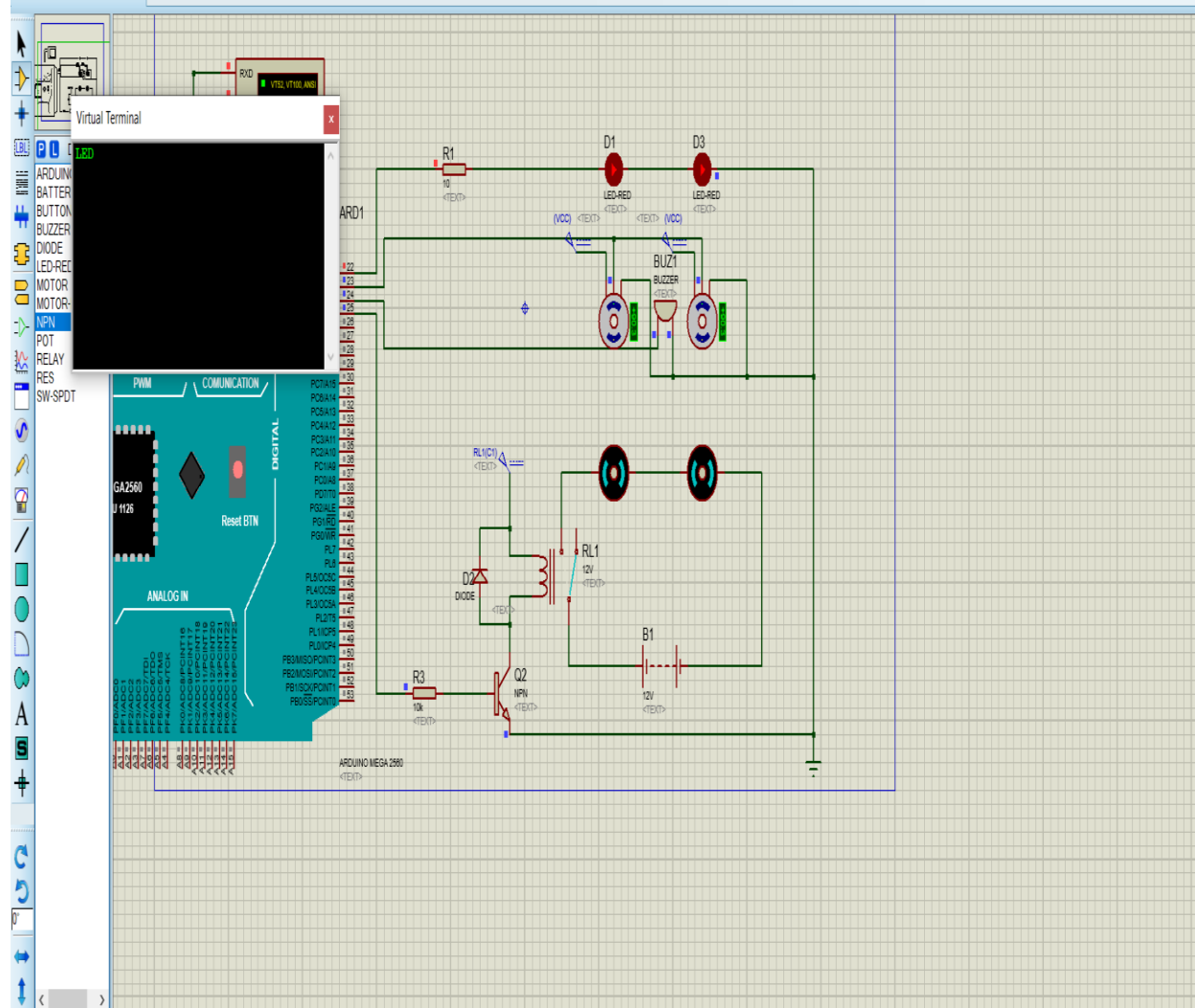
ماشین نهایی پس از اتصال تمامی قطعات و قرار دادن قطعه های مختلف ماشین در کنار هم به صورت زیر در میاید :



عکس هایی از عملکرد برنامه در زیر موجود است :



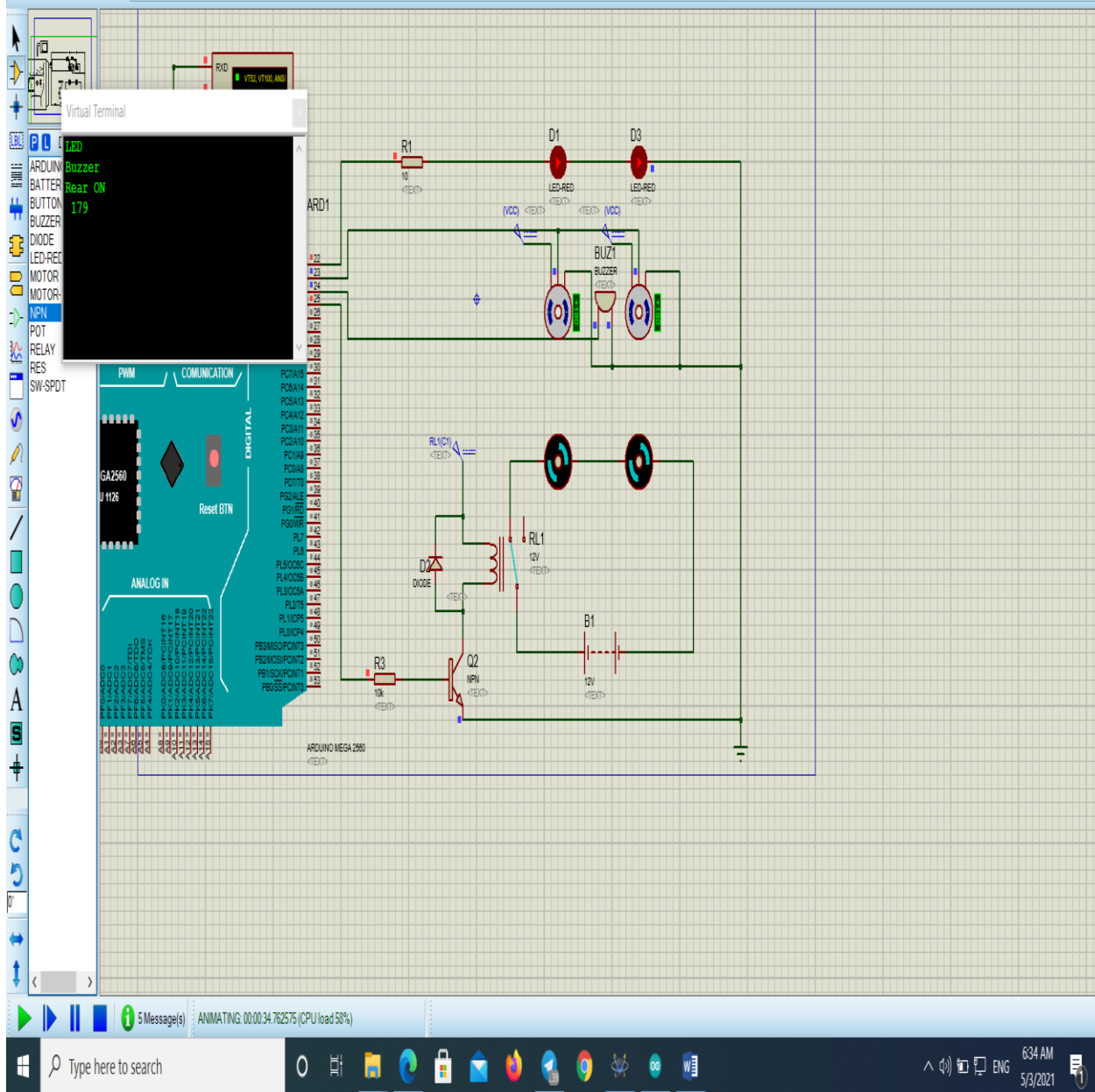
Schematic Capture X



5 Message(s) ANIMATING: 00:00:10.085526 (CPU load 66%) x: -4400.0 y: -700.0

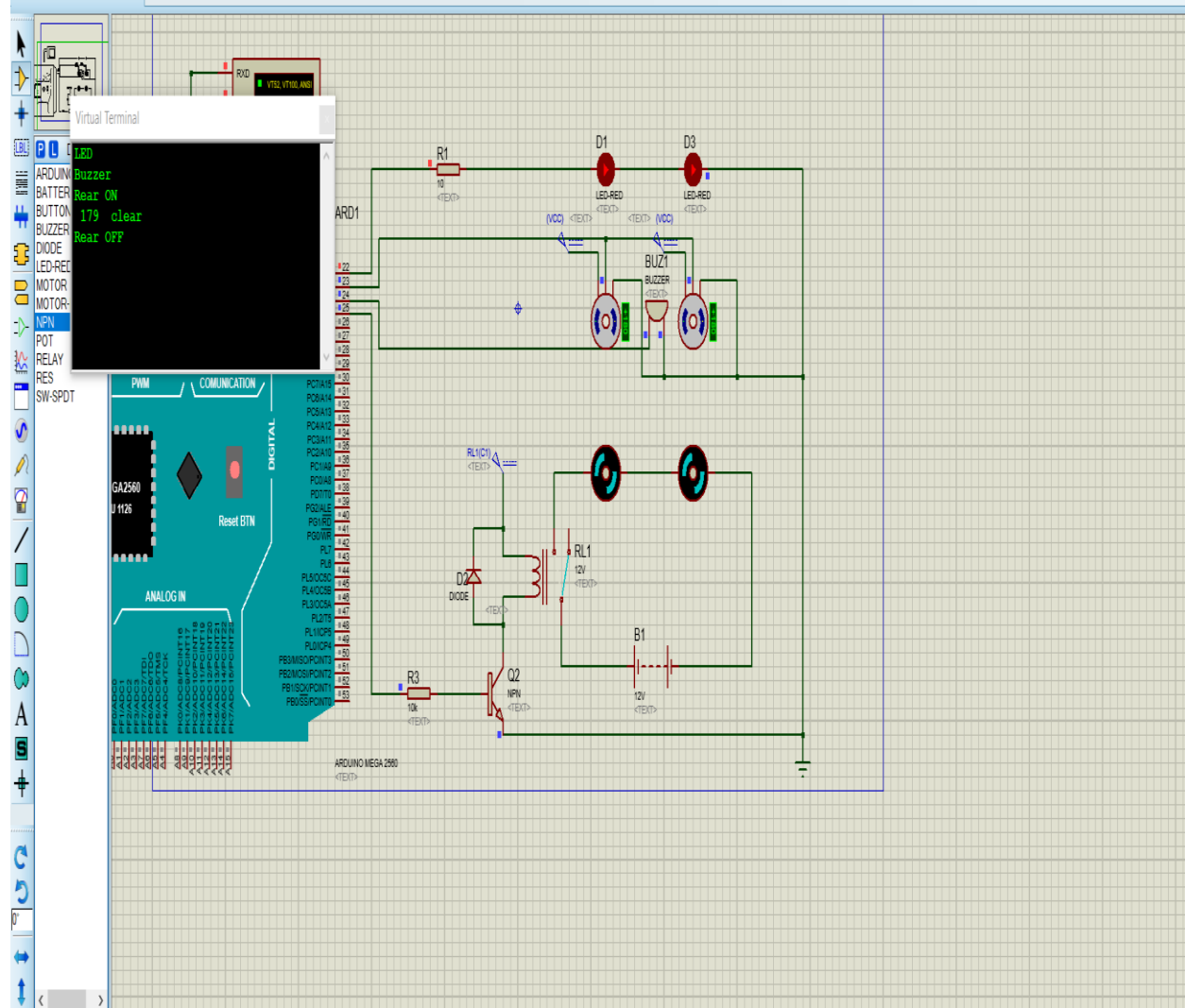


Schematic Capture X





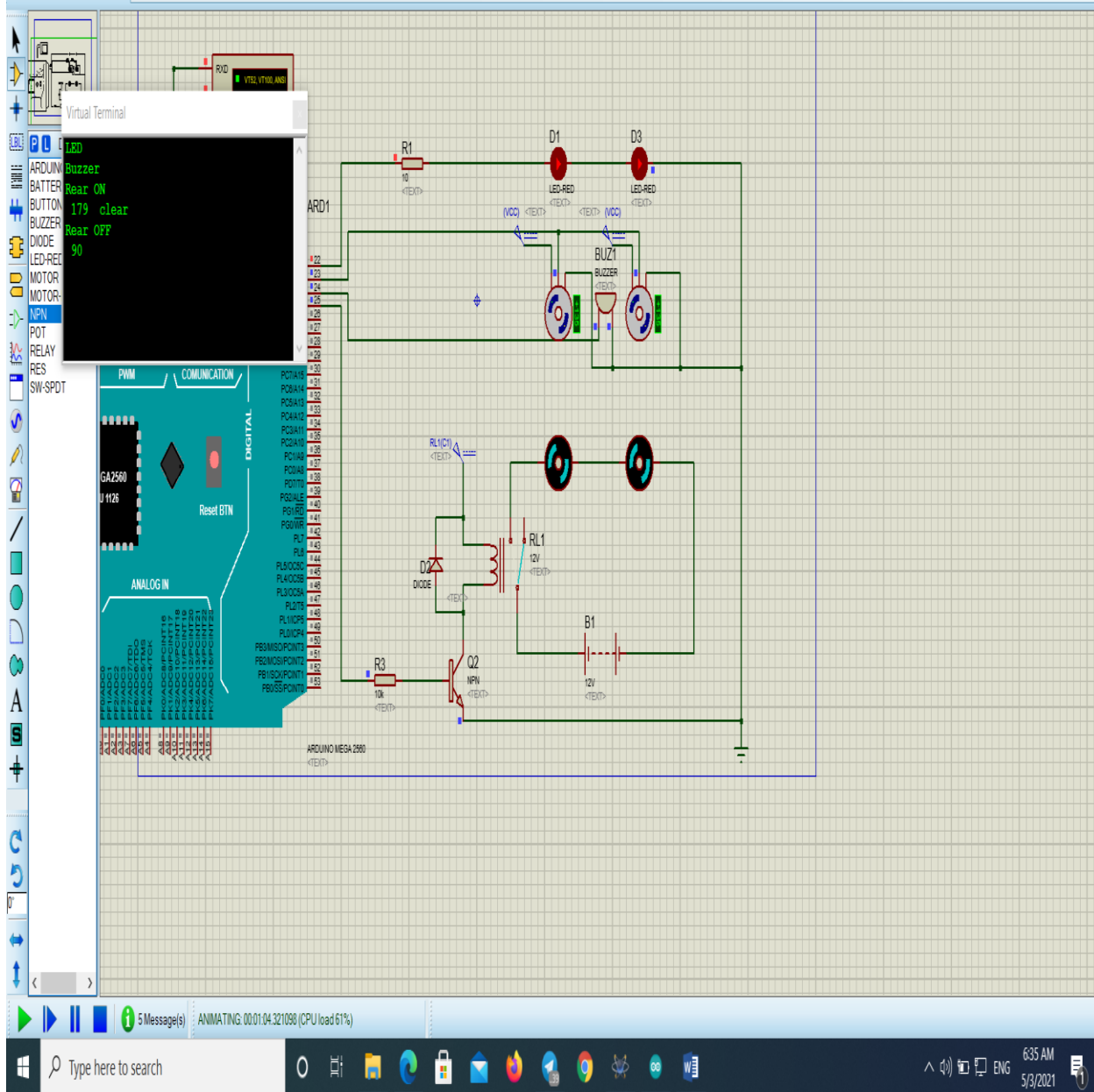
Schematic Capture X



5 Message(s) ANIMATING: 00:00:45:56303 (CPU load 68%)



Schematic Capture X



قسمت کد زدن :

در این قسمت ابتدا مثل همیشه بین ها را مشخص کردیم و سپس در `setup()` سریال و سروو موتور و بین ها را ست کردیم سپس به درون حلقه `loop()` رفتیم در این حلقه هر بار سریال را میخوانیم و با توجه به کد اسکی کاراکتر خوانده شده اعمال مناسب را انجام میدهیم و اگر کاراکتر خوانده شده هیچ یک از کاراکتر های `"l,t,f,b,enter"` نبود میفهمیم این کاراکتر یک عدد است که در واقع برای زاویه چرخ های جلو وارد شده بنابراین آن را در یک متغیر ذخیره میکنیم زمانی که کلید `enter` فشرده شود به معنای تبدیل رشته عدد به عدد و دادن این عدد به سروو موتور برای تغییر زاویه چرخ های جلو ماشین است اگر بخواهیم همه چیز را `reset` کنیم کاراکتر `"."` را وارد میکنیم

عکس های کد برنامه در زیر قابل مشاهده است :

```
sketch_may03a

#define ledPin 22
#define servoPin 23
#define buzzerPin 24
#define relePin 25

#include <Servo.h>

Servo servo;
String enter = "";
int value;
boolean led = false;
int num = 1500;

void setup(){

  Serial.begin(9600);
  servo.attach(servoPin);
  servo.writeMicroseconds(num);

  pinMode(ledPin, OUTPUT);
  pinMode(servoPin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(relePin, OUTPUT);
}

void loop(){

  if (Serial.available() > 0) {

    int key = Serial.read() - 48;
```

Done compiling.

Sketch uses 7340 bytes (2%) of program storage space. Maximum is 253952 bytes.

Global variables use 399 bytes (4%) of dynamic memory, leaving 7793 bytes for local variables. Maximum is 8192 bytes.



sketch_may03a

```
// BUZZER
// ASCII code of 'b' is 98
if( key == 50){
  Serial.println("Buzzer ");
  digitalWrite(buzzerPin, HIGH);
  delay(100);
  digitalWrite(buzzerPin, LOW);

// LED
// ASCII code of 'l' is 108
}else if (key == 60){
  Serial.println("LED ");
  led = !led;
  if(led)
    digitalWrite(ledPin, HIGH);
  else
    digitalWrite(ledPin, LOW);

// RELE
// ASCII code of 't' is 116
}else if (key == 68){
  Serial.println("Rear ON ");
  digitalWrite(relePin, HIGH);

// RELE
// ASCII code of 'f' is 102
}else if (key == 54){
  Serial.println("Rear OFF ");
  digitalWrite(relePin, LOW);
```

Done compiling.

Sketch uses 7340 bytes (2%) of program storage space. Maximum is 253952 bytes.

Global variables use 399 bytes (4%) of dynamic memory, leaving 7793 bytes for local variables. Maximum is 8192 bytes.



sketch_may03a

```
digitalWrite(relePin, LOW);

// ASCII code of "ENTER" is 13
}else if (key == -35){
    value = enter.toInt();

    Serial.write(' ');
    Serial.print(value);
    Serial.write(' ');
    num = 1500 + ((value / 180.0) * 500);
    if(value < 180 )
        servo.writeMicroseconds(num);
    else{
        num = 1500 - (((value - 180) / 180.0) * 500);
        servo.writeMicroseconds(num);
    }

// CLEAR
// ASCII code of . is 50
}else if (key == -2) {
    enter = "";
    Serial.write(' ');
    Serial.println("clear");

// OTHER KEY
}else {
    enter += key;
}

}

}
```

Done compiling.

Sketch uses 7340 bytes (2%) of program storage space. Maximum is 253952 bytes.
Global variables use 399 bytes (4%) of dynamic memory, leaving 7793 bytes for local variables. Maximum is 8192 bytes.

< >

87 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560)