

به نام خدا

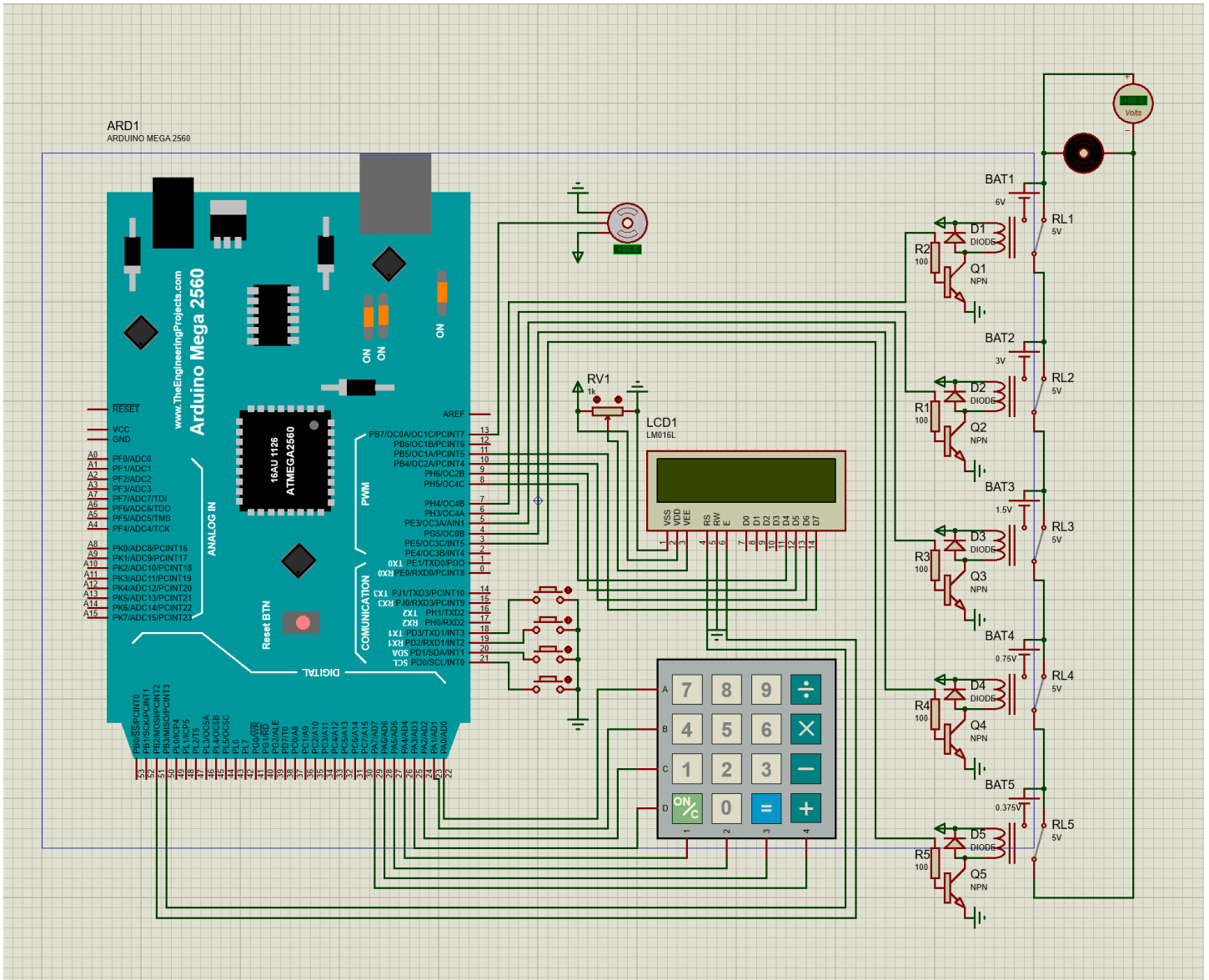
11/26/2021

گزارش آزمایش شماره 6 (تردمیل)

آزمایشگاه ریزپردازنده و زبان اسمبلی

محمد جواد زندیه , ابوالفضل بکیاسای کیوی
دانشگاه صنعتی امیرکبیر دانشکده مهندسی کامپیوتر

شمای کلی مدار:



Code:

```
#include <Keypad.h>
#include <LiquidCrystal.h>
#include <Servo.h>

//keypad object initializing
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
    {'7','8','9','/'},
    {'4','5','6','*'},
    {'1','2','3','-'},
    {'C','0','=','+'}
};
byte rowPins[ROWS] = {22, 23, 24, 25};
byte colPins[COLS] = {26, 27, 28, 29};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

//defining pins
#define RS_PIN 50
#define EN_PIN 51
#define D4_PIN 8
#define D5_PIN 9
#define D6_PIN 10
#define D7_PIN 11

//lcd object initializing
LiquidCrystal lcd(RS_PIN, EN_PIN, D4_PIN, D5_PIN, D6_PIN, D7_PIN);

#define SERVO_PIN 13

Servo servo;

// Motor Speed Control Pins
#define MSC0_PIN 3
#define MSC1_PIN 4
#define MSC2_PIN 5
#define MSC3_PIN 6
#define MSC4_PIN 7

const byte SpeedInPin = 18, SpeedDePin = 19, AngleInPin = 20, AngleDePin = 21;

int speed = 0;
int angle = 0;
int state = 0;
String pad_input = "";
```

```

void setup() {
  pinMode(SpeedInPin, INPUT_PULLUP);
  pinMode(SpeedDePin, INPUT_PULLUP);
  pinMode(AngleInPin, INPUT_PULLUP);
  pinMode(AngleDePin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(SpeedInPin), speedIn, FALLING);
  attachInterrupt(digitalPinToInterrupt(SpeedDePin), speedDe, FALLING);
  attachInterrupt(digitalPinToInterrupt(AngleInPin), angleIn, FALLING);
  attachInterrupt(digitalPinToInterrupt(AngleDePin), angleDe, FALLING);

  lcd.begin(16, 2);
  lcd.clear();

  servo.attach(SERVO_PIN, 1000, 2000);

  pinMode(MSC0_PIN, OUTPUT);
  pinMode(MSC1_PIN, OUTPUT);
  pinMode(MSC2_PIN, OUTPUT);
  pinMode(MSC3_PIN, OUTPUT);
  pinMode(MSC4_PIN, OUTPUT);

  keypad.addEventListener(keypadEvent);
  setMotorSpeed(speed);
  setAngle(angle);
  update_screen();
}

void speedIn() {
  speed = setMotorSpeed(speed+1);
  update_screen();
}
void speedDe() {
  speed = setMotorSpeed(speed-1);
  update_screen();
}
void angleIn() {
  angle = setAngle(angle+10);
  update_screen();
}
void angleDe() {
  angle = setAngle(angle-10);
  update_screen();
}

```

```

int setMotorSpeed(int s) {
    if(s > 31) s = 31;
    if(s < 0) s = 0;
    int res = s;
    digitalWrite(MSC0_PIN, (s&1) ? HIGH : LOW); s >>= 1;
    digitalWrite(MSC1_PIN, (s&1) ? HIGH : LOW); s >>= 1;
    digitalWrite(MSC2_PIN, (s&1) ? HIGH : LOW); s >>= 1;
    digitalWrite(MSC3_PIN, (s&1) ? HIGH : LOW); s >>= 1;
    digitalWrite(MSC4_PIN, (s&1) ? HIGH : LOW); s >>= 1;
    return res;
}

int setAngle(int a) {
    float b = round(a / 10.0f);
    if(b < -1) b = -1;
    if(b > 4) b = 4;
    a = round(b * 10);
    int d = map(a, -10, 40, 0, 180);
    servo.write(d);
    return a;
}

void update_screen() {
    if(state == 0) {
        lcd.clear();
        lcd.print("Speed: "+String(speed));
        lcd.setCursor(11, 0);
        lcd.print("set x");
        lcd.setCursor(0, 1);
        lcd.print("Angle: "+String(angle));
        lcd.setCursor(11, 1);
        lcd.print("set +");
    } else if(state == 1) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Enter Speed:");
        lcd.setCursor(0, 1);
        lcd.print(pad_input);
        lcd.setCursor(7, 1);
        lcd.print("(0 to 31)");
    } else if(state == 2) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Enter Angle:");
        lcd.setCursor(0, 1);
        lcd.print(pad_input);
        lcd.setCursor(5, 1);
        lcd.print("(-10 to 40)");
    }
}

```

```

}

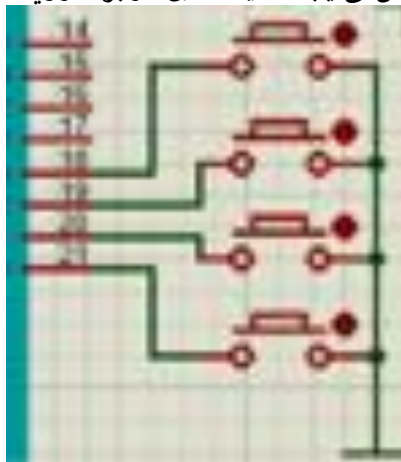
void loop() {
  char c = keypad.getKey();
}

void keypadEvent(KeypadEvent key) {
  if(keypad.getState() == PRESSED){
    if(state) {
      if(key == 'C'){
        pad_input = "";
      } else if(((key >= '0') && (key <= '9')) ||
        (key == '-' && pad_input.length() == 0)) {
        // 0 < key < 9
        if(pad_input.length() < 3)
          pad_input += key;
      } else if(key == '='){
        int v = pad_input.toInt();
        if(state == 1)
          speed = setMotorSpeed(v);
        else if(state == 2)
          angle = setAngle(v);
        state = 0;
      }
    }
    if(key == '*') {
      state = 1;
      pad_input = "";
    }
    if(key == '+') {
      state = 2;
      pad_input = "";
    }
    if(key == '/') {
      state = 0;
    }
    update_screen();
  }
}

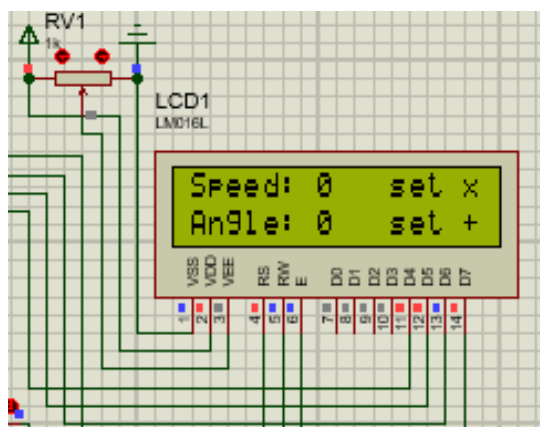
```

در این آزمایش از ما خواسته شده است تا مدار یک تردمیل را طراحی کنیم که بتوان سرعت آن را با موتور (Motor) و زاویه صفحه تردمیل را با یک سرو موتور (Servo Motor) نمایش داد. برای این کار به صورت می توان عمل کرد.

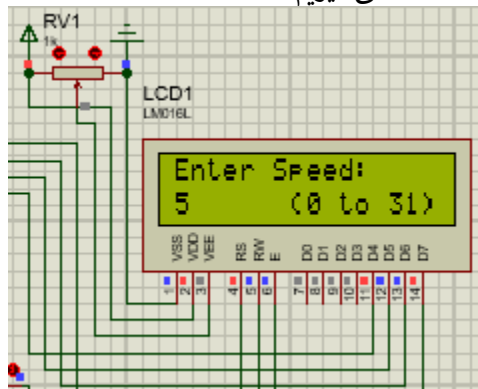
1. یا می توان سرعت موتور را یک واحدی افزایش یا کاهش داد که این کار را با دو push button انجام داده ایم به این صورت که اگر button مربوط به افزایش سرعت فشرده شده باشد آنگاه سرعت یک واحد افزایش می یابد و اگر button مربوط به کاهش سرعت فشرده شده باشد سرعت یک واحد کاهش می یابد. دقیقاً همین کار برای زاویه هم شده است که 10 تا 10 تا کم و زیاد میشود

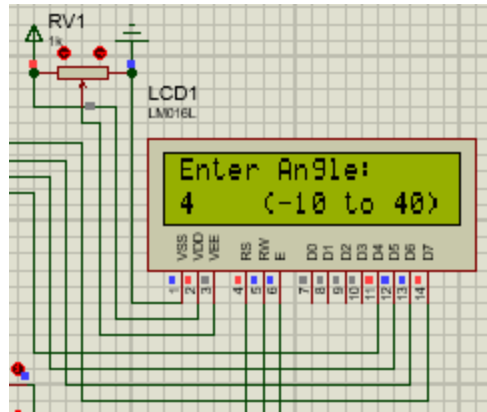


- دقت شود که بازه سرعت مجاز 0 تا 31 و بازه زاویه مجاز -10 تا 40 می باشد در نتیجه زوایا و سرعت های خارج از این بازه ها به ماکس یا مین بازه تبدیل می شود مثلاً سرعت 54 همان 31 در نظر گرفته می شود و ...
2. یا می توان از طریق یک keypad مستقیماً مقدار زاویه و یا سرعت را از کاربر گرفت و اعمال کرد.
3. در keypad سه عدد state تعریف کرده ایم. 0 State برای حالت شروع اولیه و انتخاب بین اینکه میخواهیم سرعت را تغییر دهیم یا زاویه را.

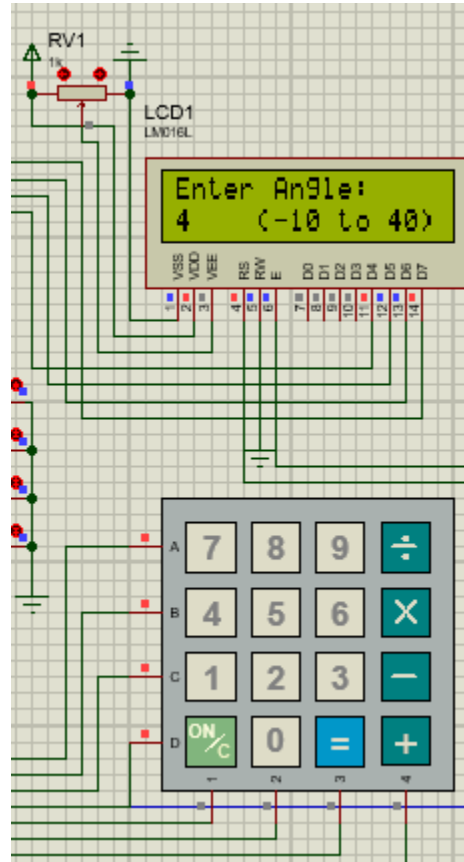


در 1 state بعد از اینکه کاربر با زدن x تغییر سرعت موتور و یا با زدن + تغییر زاویه سرو موتور را انتخاب کرده بود از کاربر میزان عددی که می خواهد برای سرعت یا زاویه را ست کند می گیریم

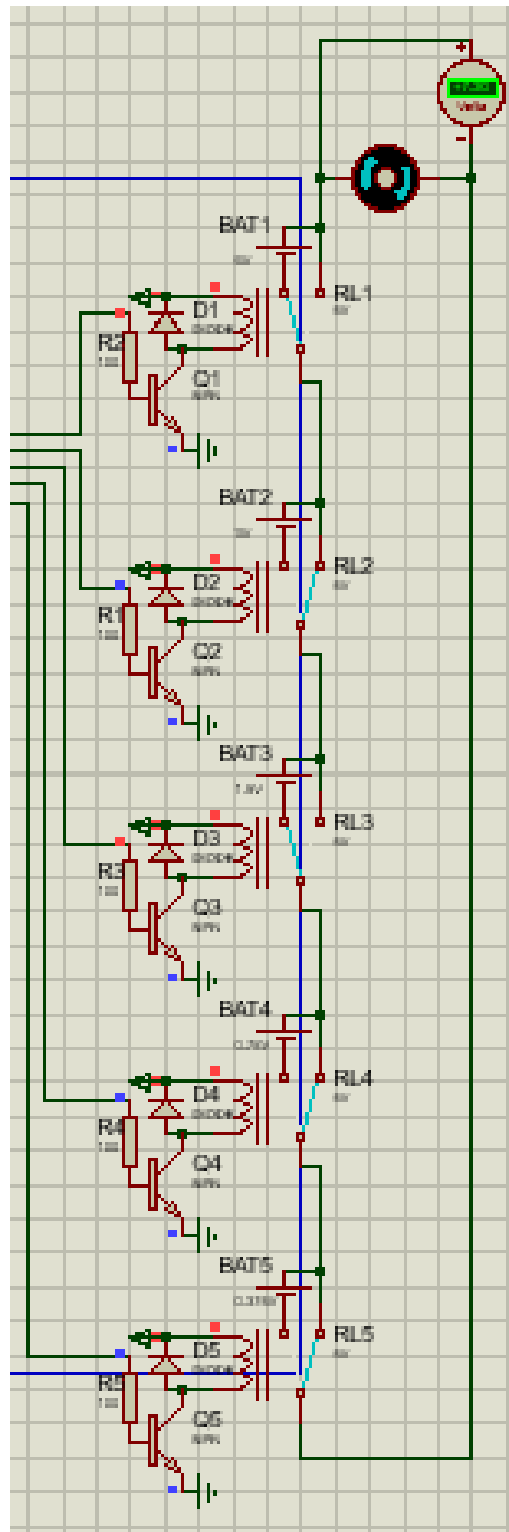




مقادیر lcd هم از یک keypad گرفته می شود.

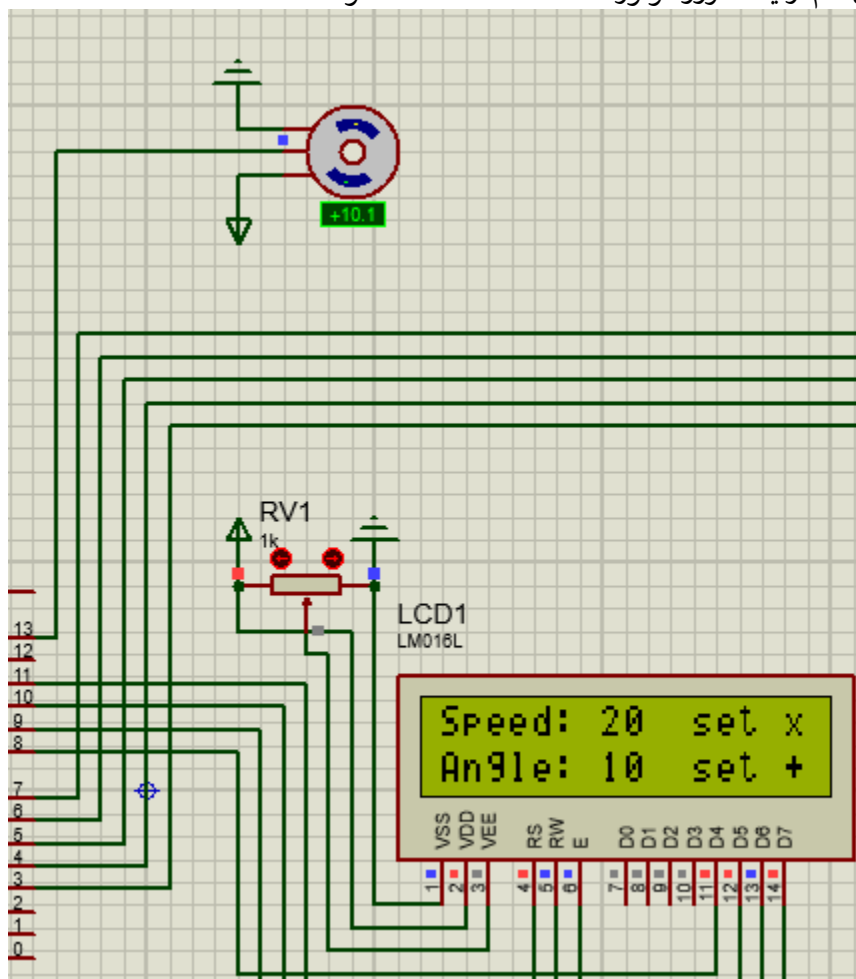


با زدن دکمه = تغییرات لازم در سرعت موتور و یا زاویه servo داده می شود و قابل نمایش است.

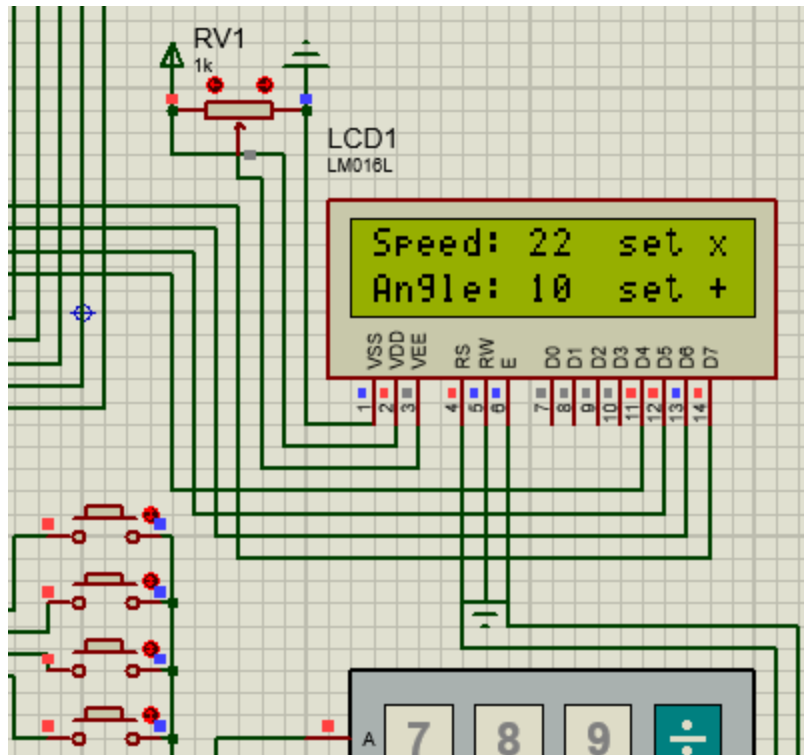


مکانیزم تغییر سرعت موتور به این صورت است که عدد داده شده توسط کاربر برای سرعت را به یک عدد باینری تبدیل می‌کنیم و رله‌ها را با توجه به مقدار تبدیل شده باینری یا به باتری وصل می‌کنیم و یا سیم قرار می‌گیرد به جای آن. مثلاً در شکل بالا سرعت 4 در باینری به صورت 00100 است که می‌توان دید فقط رله وسط به منبع وصل است.

برای زاویه هم طبق شکل دستور کار فقط می توان زوایا را به صورت 10 تا 10 تا تغییر داد به همین منظور زوایای دیگر به این مقدار رند می شوند و البته زوایای مجاز هم در بازه -10 تا +40 می باشند که در خارج از این بازه به مکس و مین بازه مپ می شوند. برای نمایش زاویه تردمیل هم از یک سروو موتور گفته شده که استفاده شود.



به عنوان مثال زاویه 10 درجه برای این سروو تعیین شده است
تغییرات قابل مشاهده در صفحه 0 state ال سی دی هم میباشد همانند شکل بالا.



در این شکل با 2 مرتبه فشردن push button مربوط به افزایش سرعت مشاهده می شود که سرعت در آل سی دی نشان داده شده است و قابل نمایش است.

مواردی که در آزمایش استفاده شده است:

4 عدد push button که 2 تای آن برای افزایش و کاهش یک واحدی سرعت است و 2 تای آن برای افزایش و کاهش یک واحدی زاویه است.

1 عدد lcd و 1 عدد keypad برای نمایش و گرفتن ورودی کاربر برای سرعت و زاویه یک عدد سروو موتور

5 عدد رله که با مکانیزم تبدیل باینری عدد داده شده سرعت را تنظیم می کند.

یک عدد موتور (Motor)