**آزمایش شماره ی ۶** **آزمایشگاه ریز پردازنده**

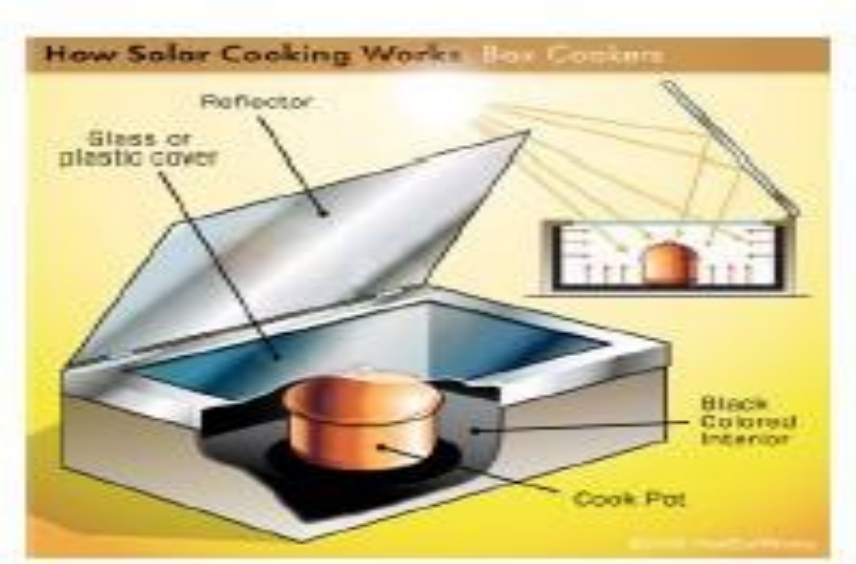
**نام و نام خانوادگی:** هلیا سادات هاشمی پور **شماره دانشجویی:**9831106

**نام استاد:**استاد علیزاده **تاریخ:**۲۷آبان 1400

# اجاق خورشیدی

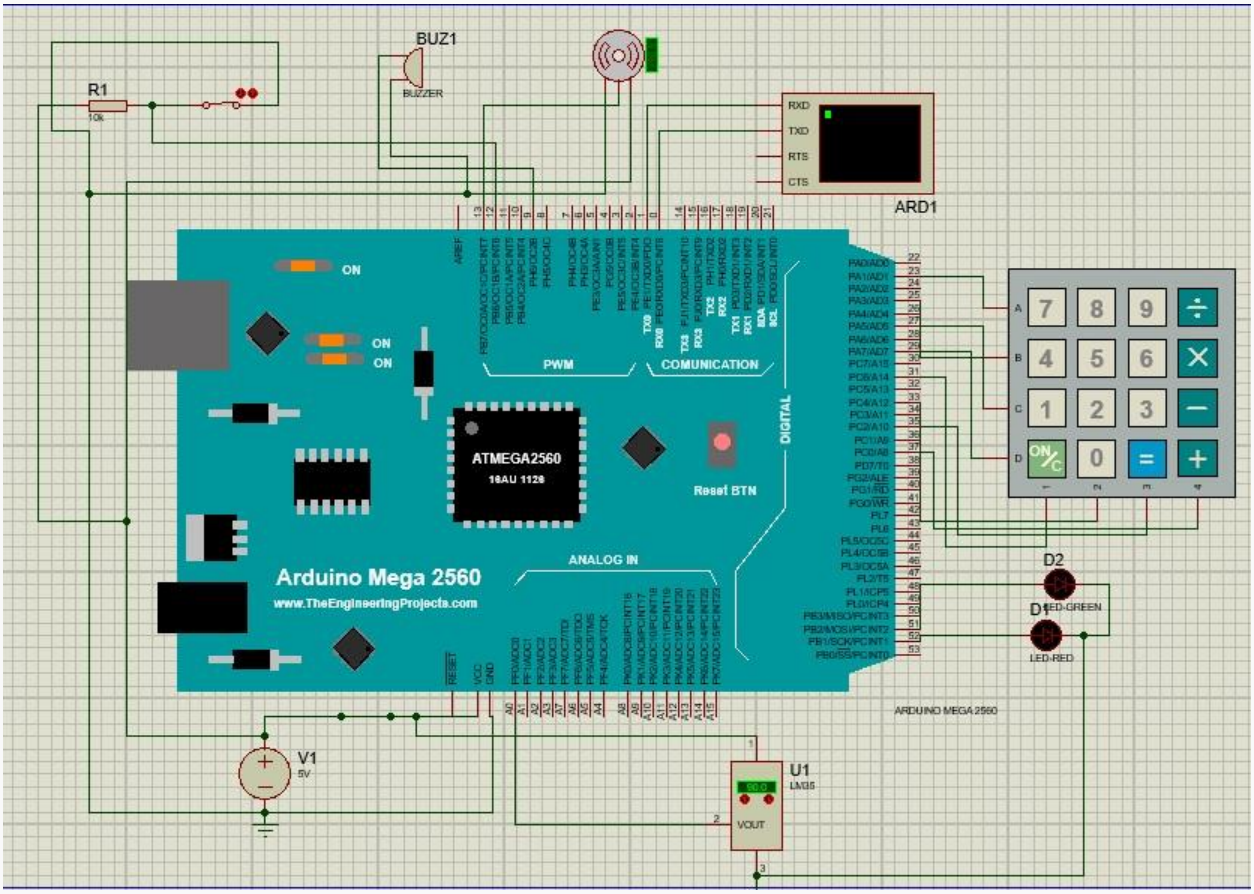
## هدف پروژه

اجاق خورشیدی دستگاه نور خورشید را از شیشه می گذراند و توسط دیواره های سیاه درونی،گرما را به دام می اندازد. میکروکنترلر ما باید شیشه را باز و بسته کند (سروو موتور) یک دما و یک زمان می گیرد (کیبورد) که هنگامی که حرارت درونی به دمای گرفته شده رسید یک تایمر را شروع کند و پس از گذشتن زمان به همان اندازه ای که گرفته شده است، با بازر کاربر را از پخت غذا آگاه کند.حال چیزی که علاوه بر دستور کار من یک ترمینال مجازی و دو LED در نظر گرفتم که در ترمینال مجازی زمان و دمایی که به آن دادیم را نمایش می دهد و اطلاعات مربوطه را نمایش می دهد. در اصل زمان را به شکل معکوس تا تمام شدن پخت نمایش می دهد و مقدار آن کم می شود. دو LED هم در برای این است که وقتی پختن تمام نشده رنگ آن قرمز هست(رنگ قرمز روشن هست) و بعد از تمام شدن فرایند پخت LED سبز روشن می شود.



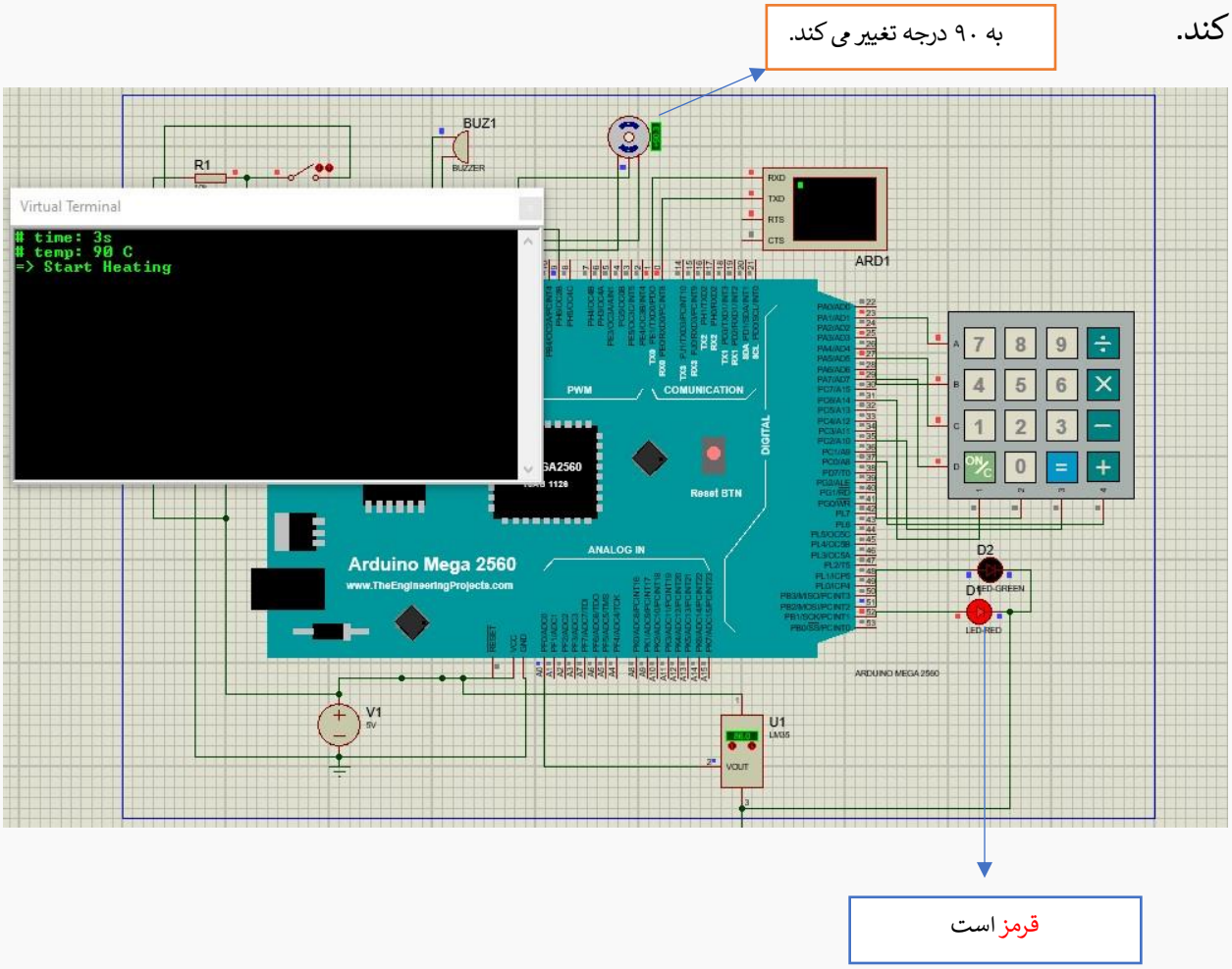
## مدار

سروو موتور و کیبورد و ترمینال مجازی و LEDها به برد وصل کردیم. بازر دو پایه دارد که یکی از آن‌ها باید به زمین و دیگری به برد وصل می شود.حرارت سنج سه پایه دارد: GND, VCC, Vout، Vout به میکرو وصل شده و در اصل خوانده می‌شود. به این گونه عمل می کند به ازای هر یک درجه سانتی‌گراد که دما زیاد می‌شود 10 میلی‌ولت، ولتاژ ورودی به میکرو را زیاد می‌کند.

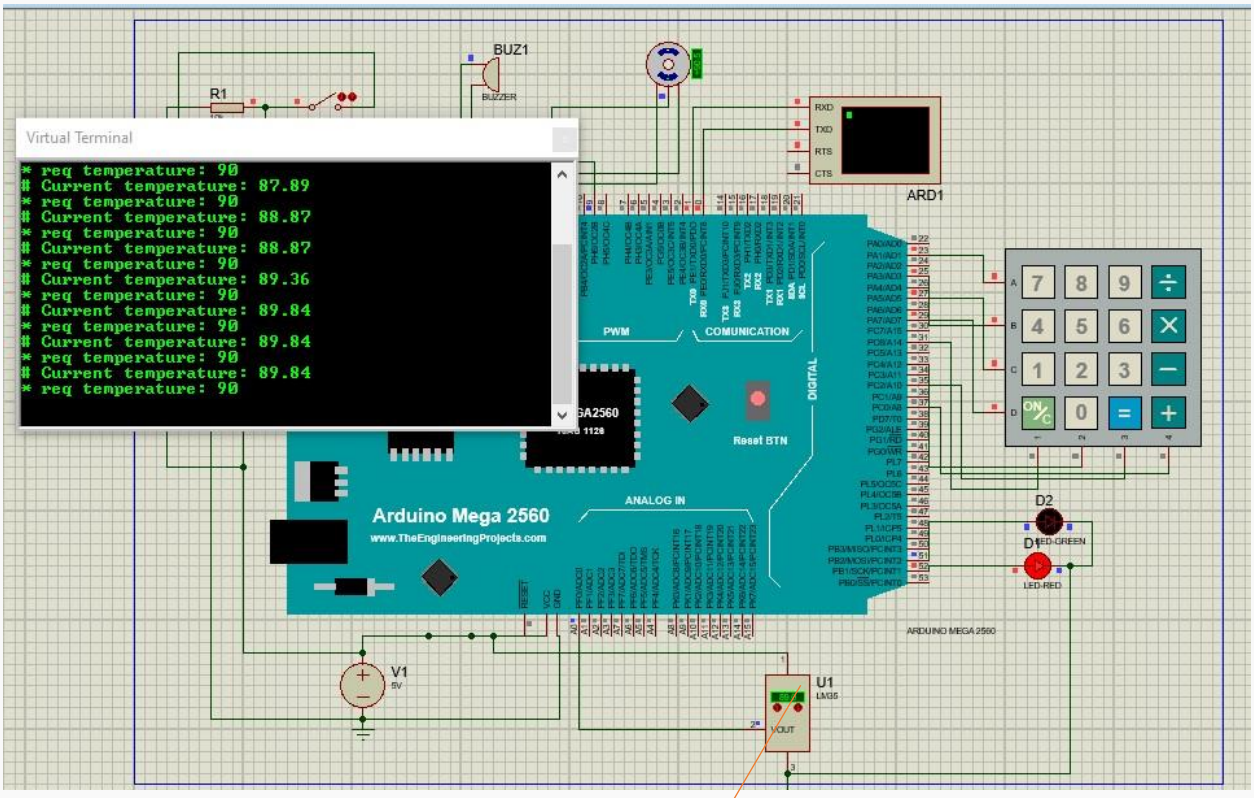


## عملکرد

برنامه را اجرا می کنیم. سپس از طریق کیپد زمان مورد نظر را وارد کرده تا زمانی که مساوی(=) وارد کنیم سپس دما را وارد می کنیم. سپس "=" را می زنیم که مقدار آن ثبت شده و نمایش داده شود. درجه ی سروموتور به 90 درجه تغییر پیدا می کند تا در اجاق اشود تا حرارت وارد آن بشود.اجاق شروع به کار می کند.

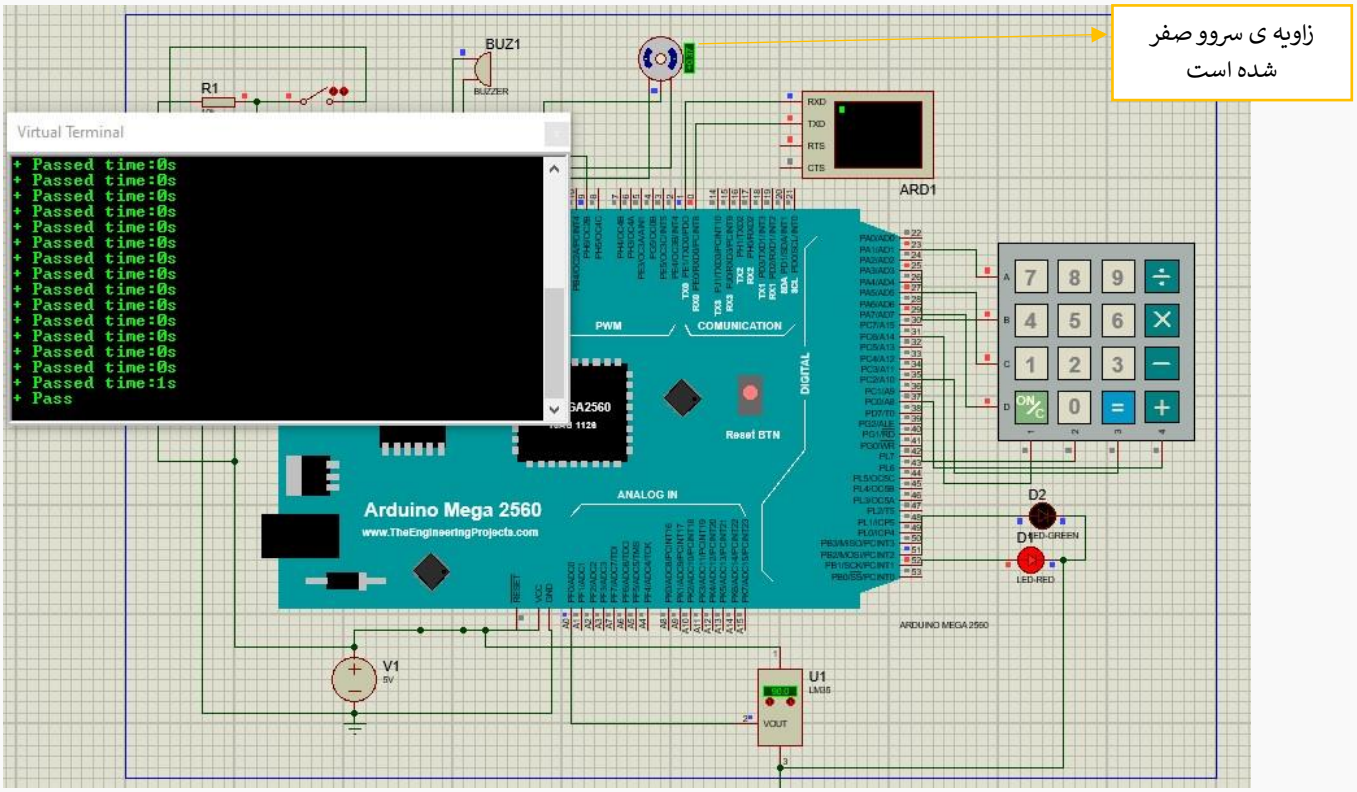


- به صورت دستی دمای حرارت سنج را افزایش می دهیم. این همان کاری هست که خورشید برای ما انجام می دهد. که در اصل دمای حرارت سنج و و اجاق بالا می رود.

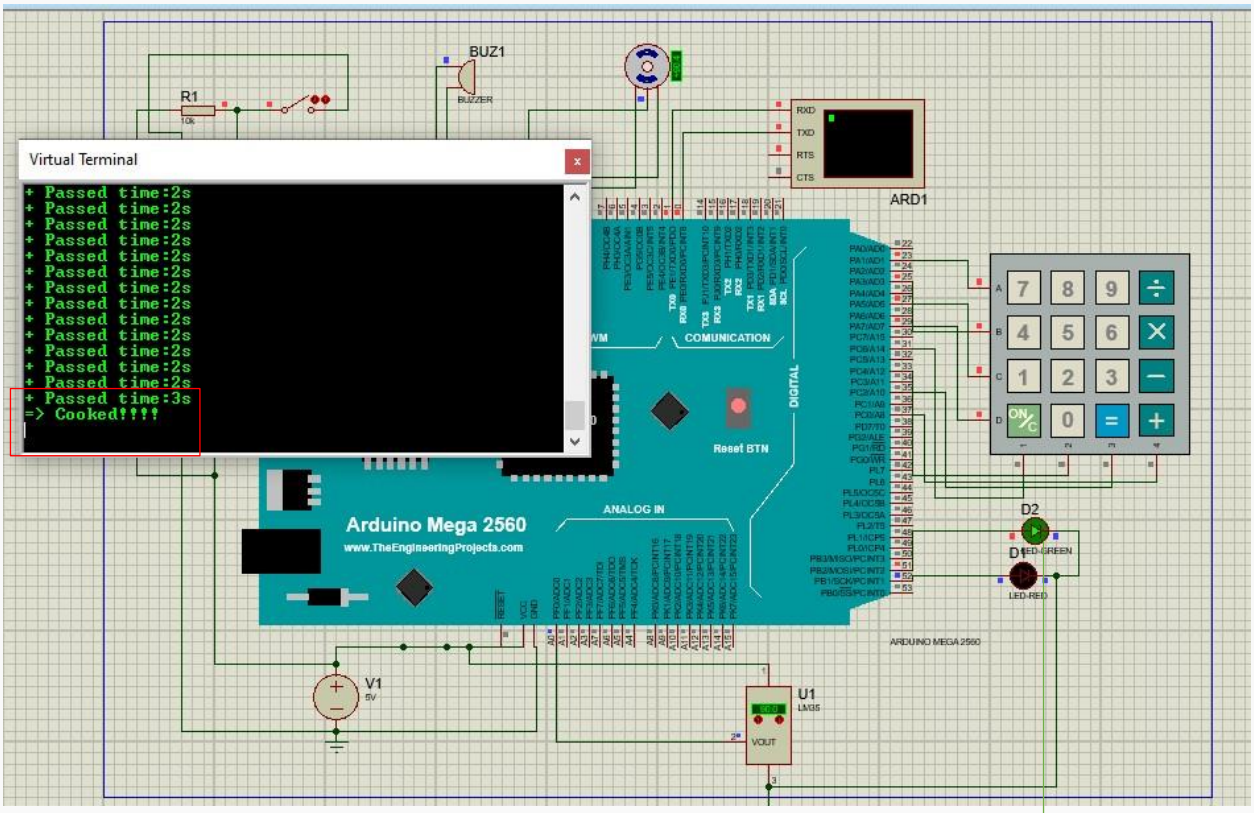



مقدار را در اینجا به شکل دستی تغییر می دهیم.

- بعد از اینکه دما را افزایش دادیم و به دمای مورد نظر رسیدیم، زاویه ی سروو موتور 0 می شود که به این معنی هست که درا اجاق بسته شده است. حال تایمر، شروع به کار می کند.

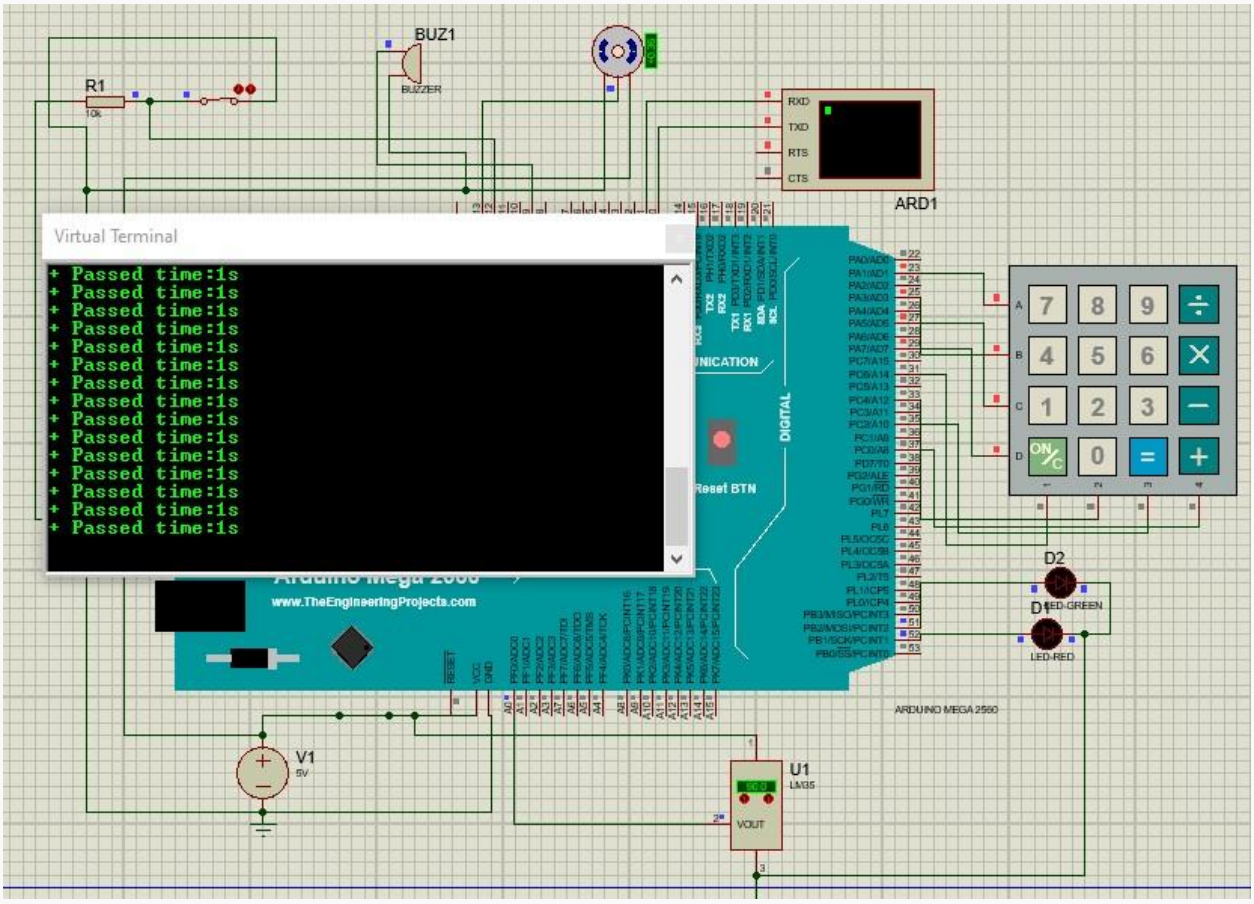


- زمان پخت که تمام شد درجه ی سروو موتور به 90 درجه تغییر پیدا می کندو بازر شروع به کار می کند و رنگLED به رنگ سبز تغییر می کند.




LED سبز روشن می شود.

- در این مرحله هم اگر سوییچ را ببندیم حکم ریست را دارد . در صورت تیاز کاربر با بستن سوییچ فرآیند پخت را متوقف می کند.



## کد

متتغیرهایی که داریم:

- **startTime** برای تعیین شروع زمان است .(مقدار اولیه آن 0 هست)
- **flag** با مقدار اولیهfalse تعریف می‌کنیم که کاربرد آن در خواندن ورودی از کیپد است.
- **timeLimit** زمانی که قرار هست که سپری می شود.
- **temperatureLimitation** دمای موردنظر تا شروع به پختن غذا کند.
- **cooked** یک متغیر بولین برای تشخیص غذا پخته شده .
- **intEntered** یک متغیر بولین برای اینکه چک کند ورودی های مورد نظر وارد شده اند.
- **isHeating** یک متغیر بولین برای اینکه چک کند اجاق در حال گرم شدن است.
- **startCooking** یک متغیر بولین برای اینکه چک کند اجاق شروع به پختن غذا کرده است.

```
#include <Servo.h>
#include <Keypad.h>
//values
int startTime = 0;
bool flag = false;
int timeLimit;
int temperatureLimitation;

bool cooked = false;
bool inEntered = false;
bool isHeating = false;
bool startCooking = false;

//Setting Keypad configs
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char keys[ROWS][COLS] = {
  {'7','8','9','/'},
  {'4','5','6','*'},
  {'1','2','3','-'},
  {'O','0','=','+'}
};
byte rowPins[ROWS] = {23, 25, 27, 29}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {31,33, 35,37}; //connect to the column pinouts of the keypad
//keypad
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

//Servo
Servo servo;  // create servo object to control a servo
```

تابع analogTemperature دما را از نوع float برمی‌گرداند. در اصل ابتدا از پایه آنالوگ A0 که به حرارت‌سنج متصل است را می‌خواند و در متغیر temp ذخیره می‌کند. چون باید خروجی مطابق انتظار ما مقیاس بندی شود، باید ولتاژی که خوانده شده را در 100/1024*5ضرب کنیم که مقدار به درجه سانتی‌گراد به دست آید. سپس شرط را چک می‌کنیم که پختن غذا شروع شده یا خیر، اگر شروع نشده بود یعنی باید باز دما را افزایش دهیم. حال دمای فعلی و دمایی که می خواهیم به آن برسیم را در هر مرحله چاپ می‌کنیم. در آخر هم تابع برای ما temp را بر میگرداند.

```
//return temperature
float analogTemperature(){//float
  float temp = analogRead(A0);
  //convert to celsius
  temp *= 0.48828125;
  //check if cooking is started or not
  if(!startCooking){
  Serial.print("$ Current temperature: ");
  Serial.println(temp);
  Serial.print("* req temperature: ");
  Serial.println(temperatureLimitation);
  }
  return temp;
}
```

عملکرد تابع checkTemperatureبه این صورت است که دمای خوانده شده از حرارت سنج را با دمایی که قرار است به آن برسیم(وارد کردیم) مقایسه می کند سپس یک متغیر bool برمی‌گرداند. در اصل اگر دمای حرارت سنج از دمای مورد انتظار بیشتر یا مساوی باشد true را بر گردانده و اگر کم تر باشد false را بر می گرداند.

```
//compare the read temperature from input
bool checkTemperature(){
  delay(300);
  bool check=false;
  if(analogTemperature() >= temperatureLimitation)
      check=true;
  return check;
}
```

تابع checkTime مقدار زمان سپری شده به میلی‌ثانیه گرفته و با تقسیم بر 1000 به ثانیه تبدیل شده و در متغیر currentTime ذخیره می‌شود و حال در هر مرحله میزان زمان سپری شده چاپ میکند ( اختلاف میان زمان فعلی و زمان شروع به پختن است). در شرط چک می‌کنیم که اگر اختلاف زمانی ما، از زمانی که قرار بوده برای پختن غذا سپری شود بیشتر یا مساوی شد، مقدار true برگردانده می‌شود و در غیر این صورت false بر گردانده می شود.

```
bool checkTime(){
  //convert it to mili second
  int currentTime = millis()/1000;
  //time distinction
  int value=(currentTime - startTime);
  Serial.print("+ Passed time:" );
  Serial.print(value);
  Serial.println("s" );
  //compare the time distinction & time limit
  if(value >= timeLimit){
    return true;
  }
  return false;
}
```

تابع getKeypad را برای خواندن از کیپد تعریف می‌کنیم. اگر false flag باشد و timer برای نگه داشتن زمان است خالی باشد، key را در[0]timer میریزیم و تا زمانی که کلید= فشرده نشده باشد ادامه ورودی را به این رشته اضافه می‌کنیم و زمانی که این کلید فشرده شد آن را چاپ می‌کنیم. رشته را با استفاده از atoi به عدد تبدیل می‌کنیم. مقدار flag را trueمی‌کنیم و temperature را نیز آزاد می‌کنیم. در شرط دوم هم اگر flag یک باشد پس از خواندن مقدار آن را در ترمینال چاپ می‌کنیم، مقدار را به عدد تبدیل می‌کنیم، flag را false می‌کنیم و timer را آزاد می‌کنیم. همچنین متغیر inEntered را trueمی‌کنیم .(ورود اعداد تمام شده است)

```
void getKeypad(){ //Getting  input from keypad
  char timer[20];
  char temperature[20];
  char key = keypad.getKey();
  if (flag){
    if(strcmp(temperature, NULL) == 0){
      temperature[0] = key;
    }
    else if(key != '='){
      strncat(temperature, &key, 1);
    } else {
      temperatureLimitation = atoi(temperature);
      Serial.print("\n");
      Serial.print("# temp: ");
      Serial.print(temperatureLimitation);
      Serial.println(" C");
      inEntered = true;
      free(timer);
      flag = false;
    }
  } else if(!flag){
    if(strcmp(timer, NULL) == 0)
      timer[0] = key;
      //if press "="
    else if(key != '=')
      strncat(timer, &key, 1);
    else{
      timeLimit = atoi(timer);
      flag = true;
      Serial.print("# time: ");
      Serial.print(timeLimit);
      Serial.println("s");
      free(temperature);
    }
  }
}
```

در setup بازر را برای پین 9 از نوع خروجی تعریف می‌کنیم. پین ها 51 و 52 را نیز برای LED ها مشخص کردیم.

```
void setup() {
  pinMode(9, OUTPUT);
  pinMode(12,INPUT);
  Serial.begin(9600);
  servo.attach(13,1000,2000);
  pinMode(51, OUTPUT);
  pinMode(52, OUTPUT);

}
```

```
void loop() {
  //if the input is not entered
  if(digitalRead(12)==HIGH){
  if(!inEntered){
    getKeypad();
    digitalWrite(51, LOW);
  }
   else{
    //if the input is entered and the starttime is zer0
  if(!isHeating && startTime == 0){
    //cooker open, so it can start heating
    servo.write(90);
    isHeating = true;
    Serial.println("=> Start Heating");
    //RED LED
    digitalWrite(52, HIGH);
  }
  else if(!cooked && startCooking){//if the food is not cooked
     //check if cooking time is finished
    if(checkTime()){
      cooked = true;
      servo.write(90);//cooker open
      Serial.println("=> Cooked!!!!");//print cooked
      digitalWrite(52, LOW);//RED LED
      digitalWrite(51, HIGH);
      tone(9, 1000);
      delay(4000);
      noTone(9);
      inEntered = false;
      startCooking = false;
      timeLimit = NULL;
      temperatureLimitation = NULL;
      startTime = 0;
      delay(100);
    }
  }
```

سوییچ ما در حالت pullup هست پس وقتی که HIGH باشد یعنی اینکه سوییچ بسته نشده است.

اگر ورودی وارد نشده، تابع geKeypad صدا زده شود و LED سبز خاموش باشد.

اگر ورودی وارد شده بود ولی هنوز گرم شدن اجاق شروع نشده بود و startTime صفر بود، سروو در اجاق را باز می‌کند تا شروع به گرم شدن کند .حال متغیر isHeating را true می‌کنیم. LED قرمز هم روشن می‌کنیم.

اگر پختن شروع شده باشد و غذا نپخته بود ، با صدا زدن تابع checkTime چک می کنیم که اگر زمان پختن سپری شده باشد متغیر cooked به true تغییر دهد و در اجاق توسط سروو باز شود. LED قرمز خاموش و LED سبز روشن می‌شود .(پخته شدن غذاست) سپس بازر با tone فعال می‌شود و به اندازه ۴ ثانیه در این حالت می ماند بعد غیرفعال می‌شود. در آخر تمام متغیرها را به مقادیر اولیه‌شان برمی‌گردانیم .


تابع checkTemperature را فراخوانی کرده و اگر true باشد یعنی حرارت کافی دریافت شده است. بعد startTime را با زمان فعلی، مقداردهی می کنیم و سروو سر اجاق را می‌بندد.(غذا شروع به پختن کند) متغیر isHeating هم false می‌شود زیرا عملیات گرفتن گرفتن به تمام شده است. مقدار startCooking هم true می شود.

این قسمت هم حکم ریست را دارد اگر کاربر سوییچ را بفشارد، اول LEDها خاموش شده سپس با استفاده از دستور Serial.end() به کار ترمینال پایان می دهیم.