

به نام خدا

## آموزش آردوینو جلسه اول

بصورت پروژه محور با ربات NK-100



آردوینو یک کامپایلر رایگان همراه با یک برد آماده با میکروکنترلر AVR است که اخیراً با میکروکنترلرهای دیگر نیز به بازار آمده و تمام قطعات لازم برای راه اندازی یک میکروکنترلر را دار می باشد. بخش هایی مثل تغذیه، اسیلاتور، پرورگرمر (بصورت بوت لودر) ، ورودی و خروجی، مدار ریست و غیره.

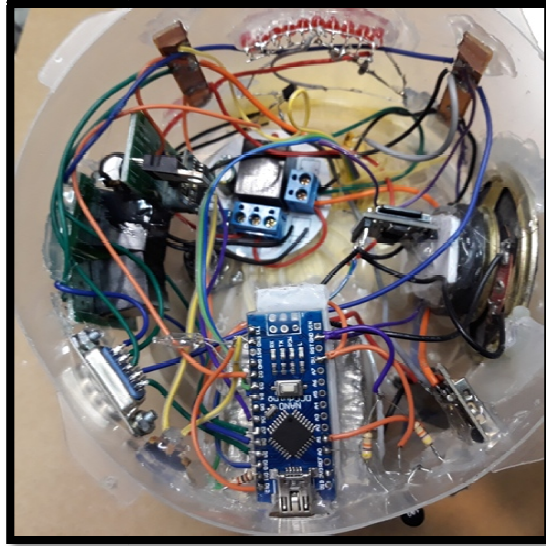
از آردوینو برای مقاصد آموزشی، نمونه سازی و تست و حتی در مواردی در دستگاه نهایی نیز استفاده می شود. آردوینو یک پلتفرم اپن سورس یا متن باز شامل یک کامپایلر رایگان به زبان

C++ و یک سخت افزار با برنامه Boot Loader است که در مدل‌های ساده عموماً یک میکروکنترلر ATmega328 حاوی یک قطعه کد بوت در انتهای حافظه فلش می باشد. برنامه بوت لودر به گونه ای نوشته شده که به کاربر اجازه می دهد از طریق پورت سریال میکرو آن را پروگرام کند و نیازی به پروگرامر جدا نداشته باشد.

بیشتر بردهای Arduino دارای یک مبدل USB به سریال روی برد می باشند که به ما این امکان را می دهد تا برد را با کمک یک کابل USB به کامپیوتر متصل کنیم و براحتی برنامه را روی برد پروگرام کنیم.

ما در این سری جلسات تصمیم داریم به امید خدا به صورت پروژه محور به آموزش کار با این برد بپردازیم. برای جذاب بودن این کار یک ربات به نام NK-100 ساخته ایم که محصول تجاری نوین کیت بوده و مخصوصاً با قطعات در دسترس و ساده ساخته شده است تا نوآموزان براحتی بتوانند این روبات را برای خود بسازند و آموزشها را روی آن انجام دهند.

روبات NK-100 از برد Arduino Nano بعنوان هسته اصلی استفاده می کند و ماژولهای زیادی روی آن نصب شده است. تغذیه این برد از یک پک باتری لیتیوم یون 3.7 ولت و یک ماژول پاوربانک تامین می شود که براحتی میتوان توسط یک آداپتور، روبات را شارژ کرد و از خروجی ۵ ولت ماژول پاور بانک برای راه اندازی برد آردوینو نانو و سایر ماژول ها استفاده نمود.



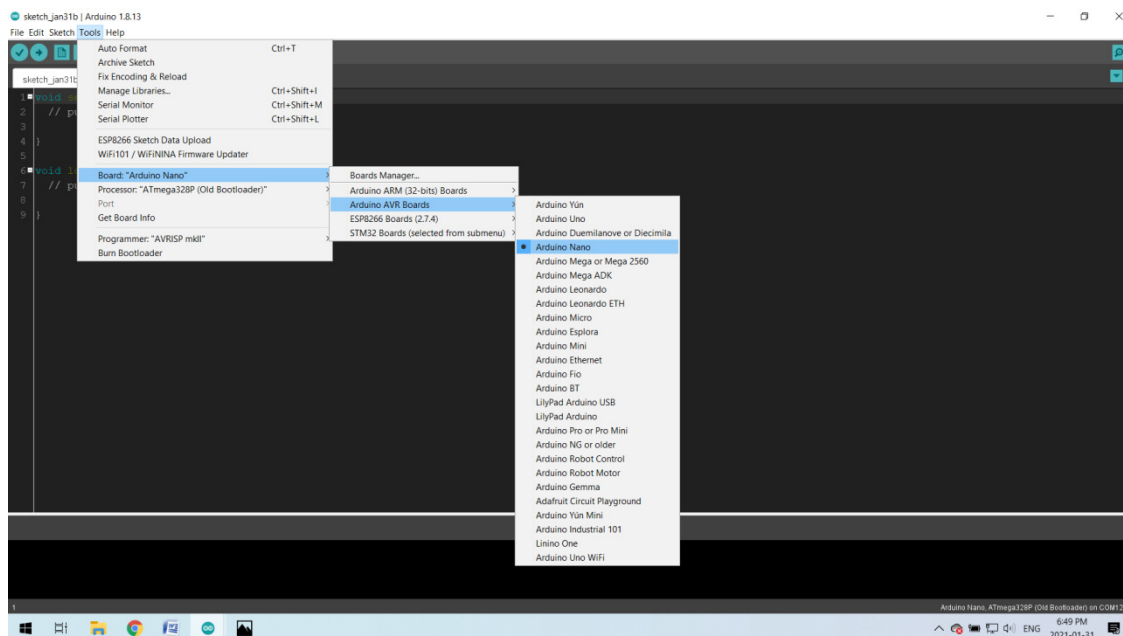
در قسمت دست روبات از يك گريپر با موتور سروو (Servo) به شماره SG90 استفاده شده كه در آموزش هاي بعدي با نحوه راه اندازي آن آشنا مي شويم. اين روبات سخنگو بوده و مجهز به ماژول پخش صداي DF Player مي باشد، همچنين براي تشخيص فاصله از ماژول آلتراسونيك SR05 استفاده کرده و نيز از يك فتوسل براي بررسي شب و روز و يك NTC براي چك كردن دماي محيط بهره مي برد.

روبات NK-100 روي چهار چرخ جلو و عقب مي رود و براي اين كار از دو عدد رله استفاده مي كند. چشم ها و دهان روبات روشن و خاموش شده و براي كنترل از خارج از يك ماژول واي فاي خارجي به نام ESP8266 استفاده مي كند كه با رگولاتور LF33 تغذيه شده و به پورت سريال ميكرو متصل شده است. ماژولهاي متصل به روبات ممكن است در طول آموزش بيشتر شود و بسته به شرايط از ماژولهاي ديگري نيز استفاده شود. همانطوريكه مي دانيد هدف از اين سري آموزش ها يادگيري كار با ماژولها و قطعات مختلف بصورت کاربردي و پروژه محور مي باشد تا بتوان کاربرد قطعات را در عمل مشاهده كرد و براي موارد ديگر براحتي استفاده نمود.

اولین قدم برای کار با برد آردوینو دانلود کامپایلر Arduino می باشد. برای دانلود این کامپایلر کافیست با جستجوی عبارت **download arduino** در گوگل و یا ورود به سایت رسمی این کامپایلر به نشانی <https://www.arduino.cc> آن را دانلود و نصب نمود.

بعد از دانلود کامپایلر آردوینو آن را نصب کنید. نصب این کامپایلر بسیار ساده بوده و با چند کلیک نصب می شود. بعد از نصب و اجرا مطابق شکل زیر از منوی Tools روی Board کلیک کرده و از زیر منوی Arduino AVR Boards روی Arduino Nano کلیک نمایید. بعد از انتخاب برد اگر ماژول آردوینوی شما قدیمی است از منوی Tools و گزینه Processor روی ATmega328P (Old Bootloader) و در غیر این صورت روی ATmega328P کلیک کنید.

البته در صورتی که از برد دیگری مثل Uno استفاده می کنید می توانید برد مورد نظر خود را انتخاب نمایید. توجه داشته باشید گزینه های کامپایلر آردوینوی نصب شده ممکن است اندکی با این تصویر تفاوت داشته باشید چرا که این کامپایلر مدام در روز بروز شدن است.



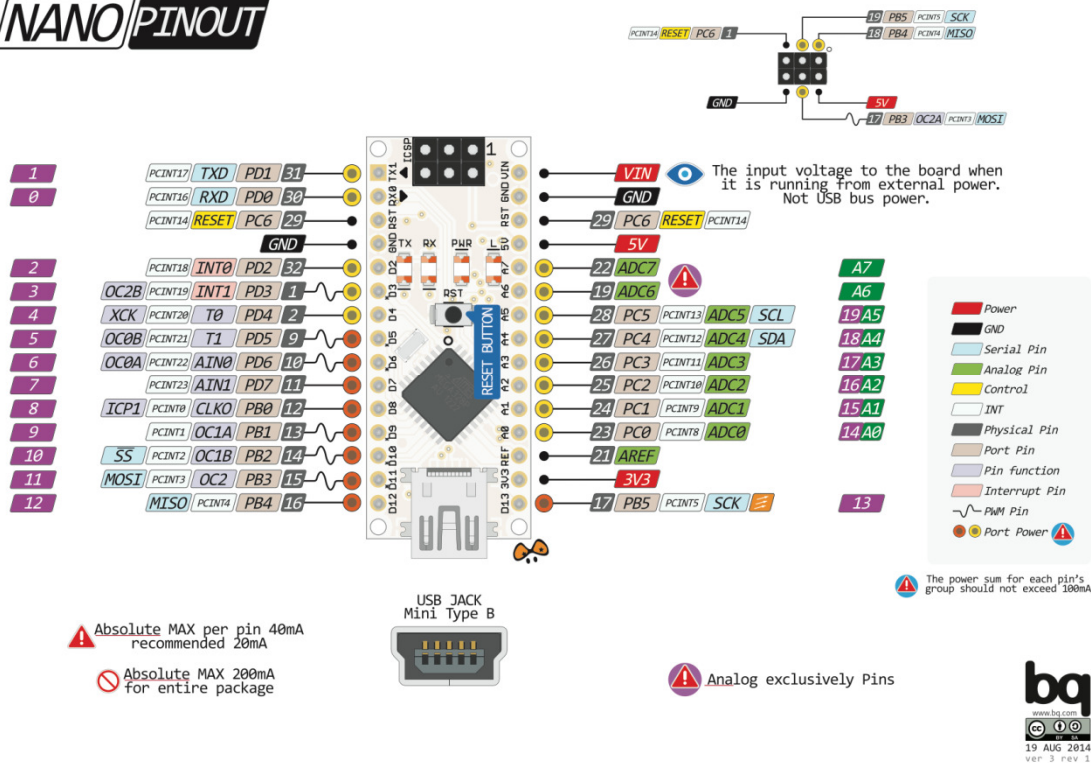
پس از انتخاب برد می توانید آردوینوی خود را به پورت USB کامپیوتر متصل کنید. بسته به سیستم عامل شما و بسته به چیپ مبدل USB به سریال روی برد آردوینو ممکن است نیاز به نصب درایور چیپ باشد و یا بصورت خودکار نصب شود. معمولاً در سیستم عامل های مدرن مثل ویندوز ۱۰ در صورتی که آنلاین باشید نصب درایورهای متداول به سرعت و بصورت خودکار انجام می شود. در صورت نصب شدن درایور می توانید از منوی Tools روی Port کلیک کرده و شماره پورتی که برد به آن متصل شده را انتخاب نمایید.

## ۲- سخت افزار جلسه اول:

ما در جلسه اول قصد داریم به روش تعریف پایه های آردوینو بعنوان ورودی یا خروجی پرداخته و نحوه نوشتن در پورت یا خواندن از آن را آموزش دهیم.

برد آردوینو نانو چیزی جز یک میکروکنترلر Atmega328 نیست که تمام پایه های آن از برد خارج شده و برای راحتی نوآموزان با اعداد نام گذاری می شود. مثلاً D0 تا D13 برای پایه های ورودی و خروجی دیجیتال و A0 تا A7 برای ورودی های آنالوگ. در شکل زیر می توانید پورت های متناظر با هر نام جدید را مشاهده نمایید.

# NANO PINOUT

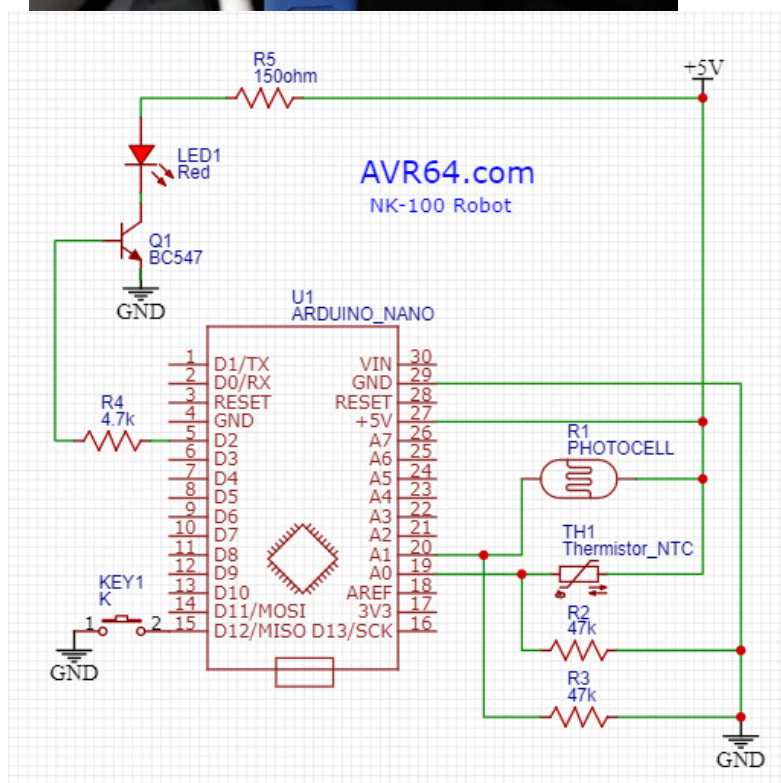
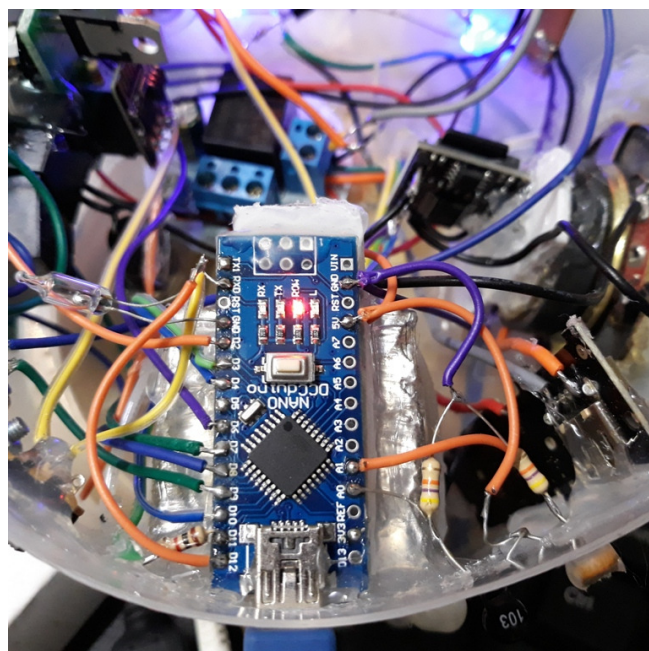


ما در جلسه اول چشم‌ها و دهان ربات NK-100 را با کمک یک ترانزیستور منفی BC547 و یک مقاومت 4.7 کیلو اهم به پایه D2 متصل کرده ایم و قصد داریم این پایه را بعنوان خروجی تعریف کنیم و آن را چشمکزن کنیم.

همچنین یک کلید جیوه ای را به پایه D12 و منفی متصل کرده ایم که با ورودی تعریف کردن این پایه و خواندن وضعیت کلید متوجه تکان خوردن روبات و ضربه شدید به سر روبات خواهیم شد. در جلسات آتی بعد از راه اندازی ماژول سخنگو به روبات می آموزیم که در صورت ضربه خوردن بتواند پاسخ مناسبی دهد.

همانطوریکه در بالا ذکر شد تعدادی از پایه ها به عنوان A0 تا A7 نام گذاری شده اند. این پایه ها معمولا برای ورودی آنالوگ استفاده می شوند و برای تست این پایه های نیز به پایه A0 یک مقاومت حرارتی NTC برای سنجش دما و به پایه A1 یک عدد فتوسل برای تشخیص نور محیط متصل کرده ایم. روبات می تواند با سنجش دما فیدبک های صوتی مناسبی بصورت تصادفی

بیان کند و نیز در صورت روشن بودن چراغ ها و یا رها شدن در یک محیط تاریک عکس العمل تعریف شده را انجام دهد و حس هوشمند بودن را را به مخاطب القا کند. هر کدام از این پایه های ورودی آنالوگ با یک مقاومت ۴۷ کیلو اهم به زمین متصل شده اند تا تا میکرو بتواند تغییرات ولتاژ بین زمین و مثبت را که به وسیله فتوسل و NTC تغییر می کند بخواند.



۳- آشنایی با دستورات و توابع آردوینو:

قبل از شروع کد نویسی به یک سری مفاهیم نیاز داریم. آردوینو از یک میکروکنترلر استفاده می کند و پایه های آردوینو به پورت های میکرو متصل شده است. هر پورت میکرو قبل از



استفاده باید بصورت ورودی یا خروجی تعریف شود. در کامپایلر آردوینو برای این کار یک تابع به نام `pinMode` وجود دارد که دقیقا باید به همین شکل و با رعایت حروف کوچک و بزرگ نوشته شود. دقت داشته باشید که آردوینو از زبان C++ استفاده می کند و در این زبان حروف کوچک و بزرگ باهم تفاوت دارند.

تابع `pinMode()` دو ورودی می گیرد. ورودی اول شماره پایه است که باید بصورت یک عدد صحیح وارد شود و آرگومان دوم حالت ورودی (INPUT) یا خروجی (OUTPUT). بعنوان مثال دستور زیر پایه متصل به LED های چشم و دهان ربات NK-100 را بعنوان خروجی تعریف می کند:

```
pinMode(2,OUTPUT);
```

همچنین دستور زیر پایه متصل به کلید جیوه ای را تبدیل به ورودی می کند:

```
pinMode(12,INPUT);
```

توجه داشته باشید که بیشتر میکروکنترلرهای به کار رفته در بردهای آردوینو از جمله میکروکنترلرهای AVR روی پایه های خود مقاومت Pullup داخلی دارند که در حالت ورودی می توان آن را فعال کرد و با این کار ۱ بودن پایه در حالت عادی تضمین می شود و از دریافت نویزهای محیط جلوگیری به عمل می آید. برای فعال کردن این مقاومت باید با دستور زیر پایه شماره ۱۲ را ورودی تعریف کنیم:

```
pinMode(12,INPUT_PULLUP);
```

در این صورت پایه به طور عادی مثبت، ۱ یا High است و با محض وصل شدن کلید، صفر یا Low می شود. بعنوان یک نکته دیگر فراموش نکنید که تمام پایه های میکرو به صورت پیشفرض ورودی و با پول آپ غیرفعال هستند.

تا اینجا کار تعریف ورودی و خروجی پایه‌های دیجیتال به پایان رسید، قدم بعدی نحوه نوشتن یا خواندن از پایه‌هاست. برای نوشتن در یک پایه از تابع `digitalWrite` استفاده می‌شود. این تابع دو آرگومان می‌گیرد که آرگومان اول شماره پایه و آرگومان دوم وضعیت پایه می‌باشد. به طور مثال با دستور زیر می‌توان LED های چشم و دهان ربات را روشن کرد:

```
digitalWrite(2,HIGH);
```

و با دستور زیر LED ها خاموش می‌شوند:

```
digitalWrite(2,LOW);
```

برای خواندن از یک پایه که بعنوان ورودی تعریف شده باید از تابع `digitalRead` استفاده کنیم. این تابع تنها یک آرگومان می‌گیرد و آن شماره پایه مورد نظر است و خروجی تابع نیز وضعیت پایه را نشان می‌دهد.

به طور مثال در رشته دستورات زیر ابتدا یک متغیر به نام `key` از نوع عدد صحیح تعریف شده و سپس وضعیت کلید خوانده شده و داخل متغیر قرار گرفته است. در قسمت بعد نیز با یک دستور شرطی `if` وضعیت کلید بررسی می‌شود.

```
int key;
```

```
key = digitalRead(12);
```

```
if(key == LOW){
```

```
}else{
```

```
}
```

در متن برنامه و سناریوی اصلی جلسه اول بیشتر با دستورات خواندن و نوشتن آشنا می‌شویم. اما قسمت بعدی ماجرا پورت آنالوگ `Arduino` می‌باشد. این پایه ها که با `A0` الی `A7`

نام‌گذاری شده‌اند بسته به نوع میکرو و نوع تنظیم ولتاژ مرجع قادر به تشخیص ولتاژ ورودی صفر تا پنج ولت بوده و برای تشخیص قطعاتی که در خروجی ولتاژ متغیر می‌دهند استفاده می‌شود.

قطعاتی مثل فتوسل (مقاومت متغیر با نور)، NTC یا مقاومت با ضریب حرارتی منفی (مقاومتی که با افزایش دما مقدار آن کم می‌شود)، ولوم یا پتانسیومتر (مقاومت متغیر)، سنسور دمای LM35، سنسور وزن، سنسور دود و گاز، ولتاژ (برای ساخت ولت‌متر) سنسور جریان و هزاران هزار قطعه و ماژول دیگر براحتی از طریق این پایه‌ها قابل خواندن و اتصال به برد آردوینو خواهند بود.

برای خواندن مقدار ولتاژ فتوسل یا مقدار نور محیط از تابع `analogRead()` استفاده می‌شود، این تابع یک آرگومان بعنوان ورودی گرفته که شماره پایه آنالوگ مدنظر می‌باشد و در خروجی مقدار ولتاژ را که بسته به نوع میکرو و تنظیمات ولتاژ مرجع می‌تواند بین صفر تا پنج ولت باشد بصورت عددی و بین عدد صفر تا ۱۰۲۳ نمایش می‌دهد.

در رشته دستورات زیر ابتدا دو متغیر به نام `temp` و `light` تعریف شده و سپس مقدار دما و نور محیط از پایه‌های مربوطه خوانده شده و در متغیرها ذخیره می‌شود. توجه داشته باشید که برای محاسبه دقیق نور و دما باید بصورت تجربی محاسبات ریاضی بر روی اعداد بدست آمده صورت پذیرد.

```
int temp,light;
```

```
temp = analogRead(A0);
```

```
light = analogRead(A1);
```

۴- کدنویسی در محیط آردوینو:

برای جلسه اول آموزش آردوینو تصمیم داریم برنامه‌ای ساده بنویسیم که به محض اجرا LED های چشم و دهان ربات NK-100 را چند بار روشن کرده و خاموش کرده و در صورت تکان خوردن سر روبات و وصل شدن کلید جیوه‌ای، LED چشم و دهان روبات یک لحظه روشن شده و مقدار ولتاژ فتوسل و سنسور دما به پورت سریال کامپیوتر ارسال شود.

```
const int LED = 2;
const int key = 12;
const int NTC = A0;
const int Photocell = A1;

void setup() {
  // put your setup code here, to run once:
  pinMode(LED,OUTPUT);
  pinMode(key,INPUT_PULLUP);
  Serial.begin(9600);
  for(int i = 0; i<3; i++){
    digitalWrite(LED,HIGH);
    delay(1000);
    digitalWrite(LED,LOW);
    delay(1000);
  }
}

void loop() {
  // put your main code here, to run repeatedly:
  if( digitalRead(key)==LOW ){
    digitalWrite(LED,HIGH);
    delay(500);
    digitalWrite(LED,LOW);
    delay(500);
    Serial.print("Temp:");
    Serial.println(analogRead(temp));
    Serial.print("Light:");
    Serial.println(analogRead(light));
  }
}
```

برنامه یک آردوینو از دو تابع setup و loop تشکیل شده است. تابع setup تنها یک بار در ابتدای برنامه اجرا می‌شود و معمولا برای پیکره‌بندی پورتهای ورودی و خروجی، تنظیمات و کدهایی که فقط یکبار در ابتدای برنامه باید اجرا شوند استفاده می‌شود.

تابع `loop` یک حلقه تکرار بوده و کدهای درون آن دائما تکرار می‌شوند و برای نوشتن کدهای اصلی برنامه که نیاز به تکرار همیشگی دارند استفاده می‌شود. در بالای برنامه و قبل از `setup` و `loop` می‌توان متغیرها، ثابت‌ها، توابع کتابخانه‌ای و سایر توابع را تعریف کرد.

ما در این برنامه برای خوانایی بیشتر برای هر پایه یک ثابت با نام دلخواه تعریف کرده‌ایم. مثلا به جای پایه شماره دو از عبارت `LED` استفاده کرده‌ایم تا موقع کدنویسی و بررسی کدها براحتی بتوانیم منظور برنامه را متوجه شویم و درواقع برنامه را مستندسازی کرده‌ایم.

در قسمت تابع `setup` ابتدا با دستور `pinMode(LED,OUTPUT)` پایه متصل به `LED` را خروجی تعریف کرده و با دستور `pinMode(key,INPUT_PULLUP)` پایه متصل به کلید جیوه‌ای را ورودی و مقاومت پول‌آپ داخلی آن را فعال کرده‌ایم.

دستور بعدی یعنی `Serial.begin(9600)` باعث فعال سازی و پیکربندی پورت سریال میکرو با سرعت ۹۶۰۰ بیت برثانیه می‌شود. پورت سریال میکرو که از طریق مبدل `USB` به سریال به کامپیوتر متصل شده هم برای پروگرم کردن میکرو کاربرد دارد و هم در حالت اجرای برنامه می‌توان برای ارسال و دریافت رشته‌ها و اعداد به محیط ترمینال آردوینو از آن استفاده کرد. با استفاده از این قابلیت براحتی می‌توانیم با برد آردوینو تعامل داشته باشیم و محتوای متغیرها را مشاهده کنیم یا توسط کامپیوتر به برد آردوینو فرمان دهیم.

در انتهای تابع `setup` به کمک یک حلقه `for` دستورات داخل حلقه را سه مرتبه تکرار کرده‌ایم. دستورات داخل حلقه شامل `digitalWrite(LED,HIGH)` برای روشن کردن `LED` متصل به چشم‌ها و دهان روبات، `digitalWrite(LED,LOW)` برای خاموش کردن `LED` و دستور تاخیر یا `delay(1000)` برای ایجاد یک مکث زمانی ۱۰۰۰ میلی‌ثانیه‌ای یا یک ثانیه‌ای برای مشاهده روشن و خاموش شدن `LED`‌ها است. با کمک این رشته دستورات چشم‌ها و دهان ربات در ابتدای برنامه سه بار چشمک می‌زند.

تابع بعدی تابع loop است. در این تابع کد اصلی برنامه نوشته شده که به طور دائم تکرار می‌شود. در این حلقه با دستور شرطی if خروجی تابع `digitalRead(key)==LOW` بررسی شده تا اگر برابر با LOW بود یعنی کلید متصل شده و دستورات درون بدنه if اجرا شود.

در داخل بدنه دستور if در ابتدا با دستور `digitalWrite(LED,HIGH)` چشم‌ها و دهان روبات NK-100 نوین‌کیت روشن شده و بعد از تاخیر نیم ثانیه‌ای که به کمک دستور `delay(500)` ایجاد شده، با دستور `digitalWrite(LED,LOW)` چشم‌ها و دهان روبات خاموش می‌شود. این علامت نشانه متصل شدن کلید در اثر ضربه است. سپس با کمک دستور `Serial.print("Temp:")` عبارت Temp: به سمت پورت سریال برد ارسال می‌شود که بر روی ترمینال Arduino قابل مشاهده خواهد بود. بلافاصله دستور `Serial.println(analogRead(NTC))` اجرا می‌شود که مقدار آنالوگ پایه متصل به مقاومت حرارتی یا همان سنسور دما را خوانده و به پورت سریال ارسال می‌کند.

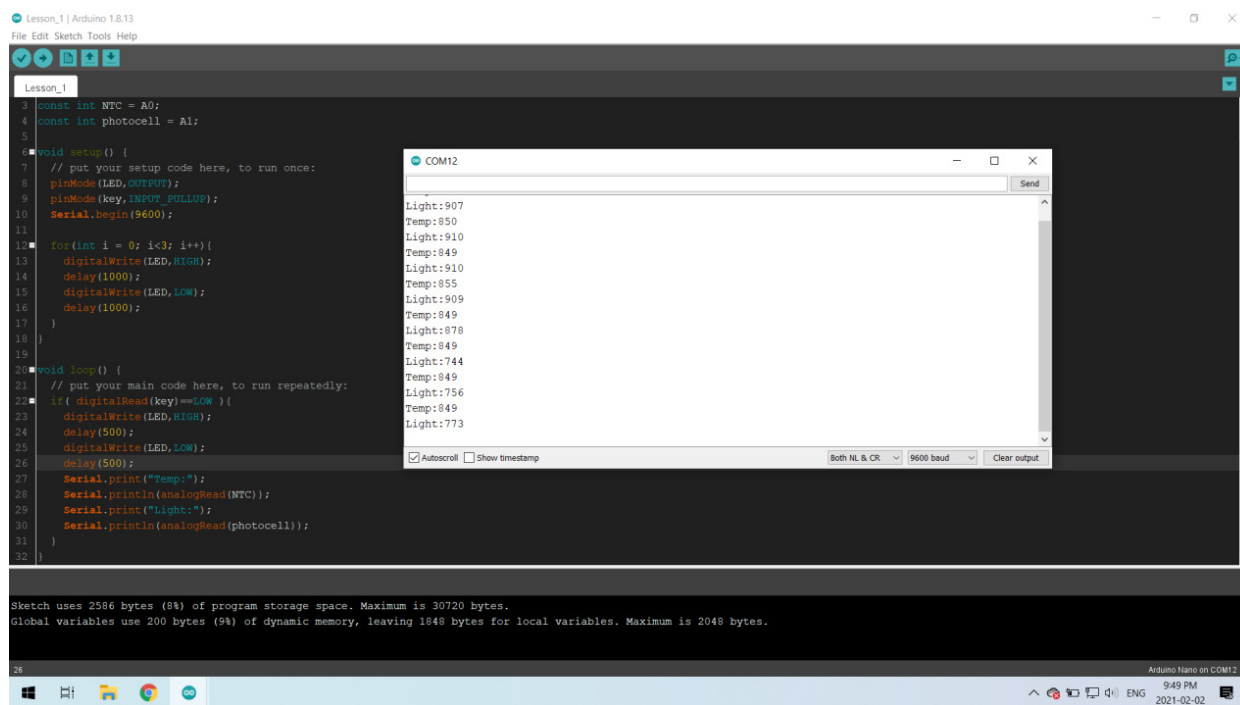
توجه داشته باشید که دستور `Serial.print` عبارت را در همان خط جاری ارسال می‌کند و در همانجا باقی می‌ماند ولی دستور `Serial.println` پس از چاپ عبارت به خط بعدی می‌رود. به همین ترتیب با دستور `Serial.print("Light: ")` عبارت Light: بر روی سریال مانیتور آردوینو چاپ شده و سپس با دستور `Serial.println(analogRead(photocell))` مقدار ولتاژ فتوسل یا همان شدت نور بر روی خروجی ارسال می‌شود.

همانطوریکه متوجه شده‌اید همیشه نیاز به تعریف متغیر و قرار دادن مقدار دیجیتالی یا آنالوگ پورت ورودی در متغیر نیست و می‌توان بدون تعریف متغیر به طور مستقیم تابع خواندن پورت را بعنوان آرگومان در دستور یا تابع دیگری مثل if یا print قرار داد.

پس از باز کردن آردوینو و تنظیم نوع برد و شماره پورت طبق آموزش ابتدای این مقاله، کدهای این جلسه را در محیط کامپایلر وارد کرده و کلید فلش گرد را در گوشه بالا و سمت چپ

برنامه فشار دهید تا برنامه کامپایل شده و داخل برد آپلود شود. همچنین می توانید از منوی Sketch روی Upload کلیک نمایید.

بعد از آپلود برنامه روی برد، روی کلید ذره بین در گوشه بالا و سمت راست کامپایلر کلیک کنید تا ترمینال باز شود. در پایین سمت راست ترمینال Baud را روی ۹۶۰۰ قرار دهید. بعد از روشن شدن برد سه مرتبه LED چشمها و دهان ربات NK-100 چشمک می زند سپس منتظر وصل شدن کلید جیوه ای می ماند. پس از ضربه زدن به سر روبات کلید جیوه ای وصل شده و LED چشمها و دهان ربات نیم ثانیه روشن شده و بلافاصله مقادیر آنالوگ مقاومت حرارتی و مقاومت نوری (فتوسل) بر روی ترمینال ارسال می شود.



The screenshot shows the Arduino IDE interface with the code for Lesson 1 and the serial monitor output. The code is as follows:

```
1 // Lesson 1
2
3 const int NTC = A0;
4 const int photocell = A1;
5
6 void setup() {
7   // put your setup code here, to run once:
8   pinMode(LED, OUTPUT);
9   pinMode(key, INPUT_PULLUP);
10  Serial.begin(9600);
11
12  for (int i = 0; i < 3; i++) {
13    digitalWrite(LED, HIGH);
14    delay(1000);
15    digitalWrite(LED, LOW);
16    delay(1000);
17  }
18 }
19
20 void loop() {
21   // put your main code here, to run repeatedly:
22   if (digitalRead(key) == LOW) {
23     digitalWrite(LED, HIGH);
24     delay(500);
25     digitalWrite(LED, LOW);
26     delay(500);
27     Serial.print("Temp:");
28     Serial.println(analogRead(NTC));
29     Serial.print("Light:");
30     Serial.println(analogRead(photocell));
31   }
32 }
```

The serial monitor output shows the following data:

```
Light:907
Temp:850
Light:910
Temp:849
Light:910
Temp:855
Light:909
Temp:849
Light:878
Temp:849
Light:744
Temp:849
Light:756
Temp:849
Light:773
```

At the bottom of the IDE, the status bar indicates: "Sketch uses 2586 bytes (8%) of program storage space. Maximum is 30720 bytes. Global variables use 200 bytes (9%) of dynamic memory, leaving 1848 bytes for local variables. Maximum is 2048 bytes."

در این جلسه با برد آردوینو و نحوه دانلود و نصب کامپایلر رایگان Arduino آشنا شدیم و ضمن نحوه معرفی پورت‌های ورودی و خروجی یک برنامه ساده با آردوینو نوشتیم. در جلسه بعد به امید خدا ماژول DF Player را به پورت سریال نرم افزاری آردوینو متصل کرده و یک روبات سخنگو می‌سازیم تا بجای چاپ پیام‌ها در ترمینال، با کاربر صحبت کند.

مولف بهنام زکی‌زاده – ۱۳۹۹/۱۱/۱۹

AVR64.com

\*\*\*