

آموزش اردوینو - فصل دوم (اولین برنامه شما، کار با LCD، کار با پورت ها)

موضوعات فصل ۲

۸ - دستورات قواعدی زبان آردوینو (سی/سی پلاس پلاس)

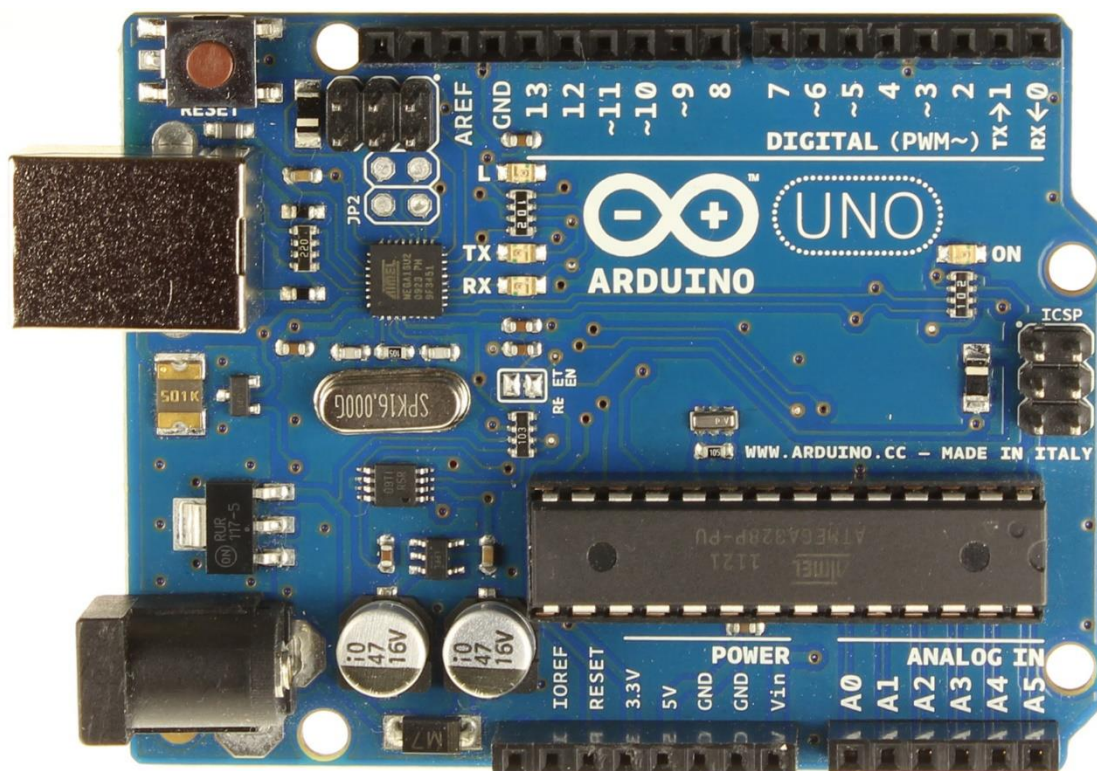
۹ - مراحل نوشتن یک برنامه جدید در آردوینو (بدنه یک برنامه)

۱۰ - دستورات مربوط به پورت ها در آردوینو (کار با پورت ها)

۱۱ - دستورات تاخیر در آردوینو

۱۲ - کار با LCD کارکتری در آردوینو (دستورات مربوط به راه اندازی و ...)

تصویر زیر یک برد Arduino uno است، پایه های این برد به سه قسمت تقسیم شده اند



DIGITAL – 1 : پایه های دیجیتال 14 تا هستند و از 0 شروع می شوند و با 13 به پایان می رسد در متن برنامه ها این پایه ها با اعداد 0 تا 13 فراخوانی و پیکربندی می شوند . در بین این 14 پایه بعضی از پایه ها PWM و یا خروجی ، ورودی سریال هستند . این پایه ها را می توان به صورت خروجی و ورودی تعریف کرد . پایه های 3 ، 5 ، 6 ، 9 ، 10 ، 11 را می توان بعنوان PWM تعریف کرد . پایه 0 بعنوان ورودی سریال و پایه 1 بعنوان خروجی می تواند تعریف شود .

ANALOG IN – 2 : این پایه ها ورودی مبدل آنالوگ به دیجیتال هستند . 6 عدد ورودی آنالود داریم که از A0 شروع می شود و با A5 خاتمه می یابند ، از بین این 6 پایه ، پایه A5 بعنوان SCL و پایه A4 بعنوان SDA تعریف می شوند که در ارتباط I2C از آن استفاده می شود ، بعضی از اردوینو ها این دو پایه را جدا کرده و دو پین مستقل به آن اختصاص داده اند

POWER – 3 : این پایه ها برای تامین ولتاژ قطعات و ماژول های جانبی مورد استفاده است در تمام اردوینو ها ارایش پایه های به شکل بالا است تنها ممکن است شکل فیزیکی برد ها با هم تفاوت داشته باشد **توجه :** برای شبیه سازی اردوینو با نرم افزار پروتیوس از این آموزش استفاده کنید .

۸ – دستورات قواعدی زبان آردوینو (سی/سی پلاس پلاس):

در هر کامپایلری که برای برنامه نویسی تعریف می شود مجموعه قواعدی وجود دارد که باید آن ها را رعایت کرد و گرنا کامپایلر از شما ایراد خواهد گرفت به قواعد زیر توجه کند .

قانون ۱ : بزرگی و کوچکی : کامپایلر اردوینو نسبت به کوچکی و بزرگی متغیر ها حساس است و باید کوچکی و بزرگی متغیر ها و دستورات را رعایت کنیم

قانون ۲ : سمیکالون : باید در انتهای هر خط از برنامه نویسی سمیکالون (;) قرار دهیم **مثال :**

```
int a = 13;
```

قانون ۳ : آکولاد {} : در درون اکولاد {} دستورات اصلی نوشته می شود **مثال :**

```
void setup() {
  pinMode(13, OUTPUT);
}
```

قانون ۴ : تک خط کامنت // : برای اینکه در یک خط بعد از دستور مد نظر یا در خط جدید توضیحاتی را ذکر کنیم ابتدا // قرار می دهیم سپس توضیحات را بعد از آن می نویسیم و مجاز هستیم تنها یک خط توضیحات بنویسیم و اگر رفتیم خط بعدی مجدد // را در ابتدا بنویسیم **مثال :**

```
x = 5; // This is a single line comment. Anything after the slashes is a comment
      // to the end of the line
```

قانون ۵: چند خط کامنت /* */: برای توضیحات طولانی که چندین سطر است از دستور چند خط کامنت استفاده می کنیم مثال:

```
x = 3; /* but not another multiline comment - this is invalid */
```

۹ – مراحل نوشتن یک برنامه جدید در آردوینو (بدنه یک برنامه):

پیکره اصلی یک برنامه در کامپایلر آردوینو به شکل زیر است

```
void setup() {
    // تعریف متغیر (حلقه اول)
}

void loop() {
    // حلقه اصلی
}
```

تعریف متغیر (حلقه اول): در این قسمت اطلاعاتی که یک بار مورد بررسی قرار میگیرد جای دارد. بعنوان مثال، تعریف متغیر، وضعیت پایه ها، تعریف کتابخانه و ... در این بخش قرار می گیرند، این بخش فقط یک بار و هنگام روشن کردن برد یا ریستار کردن برد اجرا می شود. **مثال:**

```
void setup()
{
    pinMode(1, INPUT);
}
```

در مثال بالا پایه 1 بعنوان ورودی تعریف شده است،

حلقه اصلی: در این حلقه برنامه اصلی که مرتباً تکرار می شود، قرار می گیرد این حلقه تا بی نهایت ادامه دارد و همیشه تکرار می شود. **مثال:**

```
void loop()
{
    if (digitalRead(1) == HIGH)
        Serial.write('H');
    else
        Serial.write('L');

    delay(1000);
}
```

در مثال بالا اگر کلید متصل به پایه 1 برابر یک شد در خروجی سریال حرف H ارسال می شود و اگر مساوی یک نباشد حرف L ارسال می شود.

۱۰ – دستورات مربوط به پورت‌ها در آردوینو (کار با پورت‌ها) :

برای استفاده از پایه‌ها در آردوینو باید آن‌ها را به صورت خروجی و ورودی پیکربندی کرد .

یک پایه هنگامی به عنوان خروجی تعریف می‌شود که بخواهیم از آن ولتاژ بگیریم و یک پایه هنگامی به عنوان ورودی قرار می‌گیرد که بخواهیم به آن ولتاژ بدهیم ، (مثلاً هنگامی که قصد داریم کلیدی را به میکرو متصل کنیم ، باید یک سر کلید را به ۵ ولت و سر دیگر آن را به پایه‌های میکرو متصل نماییم ، هنگامی که کلیدی فشرده می‌شود ولتاژ ۵ ولت به پایه‌ی میکرو اعمال شده و آردوینو متوجه تغییر وضعیت می‌شود ، گرفتن ولتاژ نیز همچون یک LED (چراغ) است که بین گراند (صفر ولت) و یکی از پایه‌های آردوینو متصل شده و می‌تواند یک وضعیت خروجی را نمایش دهد)

برای خروجی یا ورودی قرار دادن یکی از پایه‌ها از دستور زیر استفاده می‌شود :

`pinMode (pin, mode)`

این دستور در حلقه اول قرار می‌گیرد و متغیرهای آن به شکل زیر هستند

Pin : یکی از پایه‌های آردوینو است و می‌تواند عدد ۰ تا ۱۳ باشد . (البته در صورت نیاز می‌تواند از ۰ تا ۲۴ باشد)

Mode : وضعیت پایه آردوینو می‌باشد که می‌خواهیم بعنوان ورودی یا خروجی تعریف کنیم و می‌تواند به شکل‌های زیر تعریف شود

INPUT : هنگامی که بخواهیم پایه‌ای را بعنوان ورودی تعریف کنیم

OUTPUT : هنگامی که بخواهیم پایه‌ای را بعنوان خروجی تعریف کنیم

INPUT_PULLUP : هنگامی که به پایه‌ای بعنوان ورودی و برای اتصال کلید به کار برود

مثال :

```
pinMode (0, INPUT_PULLUP);
pinMode (13, INPUT);
pinMode (1, OUTPUT);
```

دستورات مربوط به پورت‌ها :

دستور digitalWrite

`digitalWrite()` : از این دستور هنگامی استفاده می‌شود که یک پایه بعنوان خروجی تعریف شده باشد ، به وسیله این دستور می‌توان

توان یک پایه خاص (پایه ۰ تا ۱۳) را ۰ ولت یا ۵ ولت کرد (HIGH or LOW) ، هنگامی از این دستور استفاده می‌شود که

اگر یک مصرف کننده مانند LED به پایه مد نظر وصل شده باشد

فرم کلی دستور به شکل زیر است :

`digitalWrite(pin, value)`

pin : شماره پایه اردوینو (از 0 تا 13)

Value : وضعیت پایه که می تواند HIGH یا LOW باشد (HIGH= 5V and LOW=0V)

دستور digitalWrite

`digitalRead()` : این دستور وقتی استفاده می شود که یک پایه بعنوان ورودی تعریف شده باشد و وظیفه آن دریافت مقدار دیجیتال (0 یا 1) از بیرون است ، بعنوان مثال وقتی کلیدی به پایه ای از اردوینو وصل می شود و می خواهیم از وضعیت فشردن کلید مطلع شویم از این دستور استفاده می کنیم

فرم کلی دستور به شکل زیر است :

`digitalRead(pin)`

pin : پایه ای از اردوینو (0 تا 13) که به آن کلید (یا وسیله ای دیگر) وصل است

مثال های مربوط به پورت ها

مثال ۱: استفاده از دستور OUTPUT و digitalWrite()

در مثال زیر نحوه استفاده از دستور OUTPUT و `digitalWrite()` آموزش داده شده است ، در این مثال پایه 0 اردوینو را بعنوان خروجی معرفی می کنیم سپس یک led را به پایه 0 اردوینو وصل می کنیم و آن را روشن و خاموش می کنیم . (چشمک زن)

```
void setup() {
  pinMode(0, OUTPUT);
}
void loop() {
  digitalWrite(0, HIGH);
  delay(1000);
  digitalWrite(0, LOW);
  delay(1000);
}
```

