

گزارش کار اول آزمایشگاه ریزپردازنده

تهیه و تنظیم: مبین خیبری

شماره دانشجویی: 994421017

استاد راهنما: دکتر نیکزاد

چکیده:

در جلسه‌ی نخست درس آزمایشگاه ریزپردازنده ابتدا دانشجویان به اختصار با یکی از میکروپروسسورهای شناخته‌شده‌ی صنعت الکترونیک آشنا شدند. این میکروپروسسور، Arduino نام دارد و می‌توان آن را عضوی از خانواده‌ی پردازنده‌های AVR دانست.

انواع مختلف این میکروپروسسور دارای تعدادی پین یا پورت ورودی و خروجی می‌باشند. اجزای این برد علاوه بر این که یک سیستم کامپیوتری مینیمم را نمایندگی می‌کنند، شامل حداقل یک لامپ LED هم هستند.

برای برنامه‌نویسی و کنترل اجزای مختلف این برد می‌توان با استفاده از زبان‌های C و Assembly دستوراتی را به آن ارسال کرده و پاسخ متناظر را دریافت نمود.

در کلاس درس برای ارتباط با اجزای مختلف این برد از برنامه‌ای با عنوان Arduino IDE استفاده شد.

در قسمت‌های پیش‌رو تاریخچه‌ی این برد را مورد بررسی قرار داده و ویژگی‌های آن را توصیف می‌کنیم.

آردوینو

آردوینو یک پلتفرم سخت‌افزاری و نرم‌افزاری متن‌باز است. پلتفرم آردوینو شامل یک میکروکنترلر تک‌بردی متن‌باز است که قسمت سخت‌افزار آردوینو را تشکیل می‌دهد. علاوه بر این، پلتفرم آردوینو یک نرم‌افزار آردوینو IDE که به منظور برنامه‌نویسی برای بردهای آردوینو طراحی شده است و یک بوت لودر نرم‌افزاری که بر روی میکروکنترلر بارگذاری می‌شود را در بر می‌گیرد. پلتفرم آردوینو به منظور تولید سریع و ساده پروژه‌های سخت‌افزاری تعاملی و ساخت وسایلی که با محیط تعامل داشته باشند طراحی شده است، البته بردهای آردوینو اهداف آموزشی را نیز دنبال می‌کنند.

اغلب بردهای آردوینو که تمام آن‌ها سخت‌افزار متن‌باز هستند بر پایه میکروکنترلر ای‌وی‌آر اتمل و تعداد کمی از بردهای آردوینو بر پایه میکروکنترلرهای آرم اتمل طراحی شده‌اند. به عنوان مثال برد آردوینو UNO که پرکاربردترین برد آردوینو و برد پایه آردوینو در اکثر دوره‌های آموزش آردوینو است و بر پایه میکروکنترلر AVR ATmega328 ساخته شده است، دارای رابط یواس‌بی جهت بارگذاری برنامه و ارتباط با کامپیوتر، ۶ پین ورودی آنالوگ و همچنین ۱۴ پین ورودی/خروجی دیجیتال است که شما را قادر می‌سازند تا برد آردوینو را به قطعات، سنسورها، بردها و ماژول‌های دیگری متصل کنید. تعداد ورودی خروجی‌های

آنالوگ و دیجیتال در مدل‌های مختلف بردهای آردوینو با توجه به میکروکنترلر اصلی استفاده شده بر روی برد متفاوت است.

آردوینو می‌تواند جهت طراحی و ساخت سریع و آسان وسایل تعاملی مورد استفاده قرار گیرد. به عنوان مثال فرض کنید شما دوست دارید وسیله‌ای داشته باشید که با استفاده از گوشی موبایل بتوانید چراغ اتاقتان را خاموش و روشن کنید. یا دوست دارید زمانی که اتاق شما گرم می‌شود کولر اتاقتان روشن شود و شب‌ها که دمای اتاق پایین می‌آید کولر اتاق شما به صورت خودکار خاموش شود! این‌ها وسایل تعاملی هستند که آردوینو به ساخت سریع و بی درد سر آن‌ها کمک می‌کند. برد آردوینو می‌تواند مقادیر ورودی را از تعداد زیادی سنسور و کلید و... بخواند و بر اساس برنامه‌ای که درون آن بارگذاری شده است تصمیم بگیرد و خروجی خاصی که می‌تواند کنترل تعدادی لامپ، موتور و... را برای شما انجام دهد. آردوینو در سال ۲۰۰۵ به منظور ایجاد راهی ارزان و ساده برای برنامه‌نویسی اشیای تعاملی ایجاد شد. آردوینو به همراه یک محیط یکپارچه توسعه نرم‌افزار (IDE) ساده ارائه می‌شود که در رایانه‌های عادی قابل اجرا است که اجازه برنامه‌نویسی به کمک سی یا سی++ را برای آردوینو می‌دهد.

آردوینو می‌تواند پارامترهایی مانند نور محیط، کلیدها یا حتی یک ایمیل را به عنوان ورودی دریافت نماید و بعد از پردازش، خروجی‌هایی مانند روشن کردن یک وسیله برقی، تغییر رنگ LEDها یا ارسال یک ایمیل یا نظیر آن را ارائه دهد.

پلتفرم آردوینو شامل نرم‌افزار و سخت‌افزار متن باز می‌شود. سخت‌افزار آردوینو متشکل از بردهای مختلفی بر پایه میکروکنترلرهای مختلف است که پر مصرف‌ترین سخت‌افزار آردوینو، Arduino UNO است که از میکروکنترلر Atmega328 استفاده می‌کند و در بسیاری از پروژه‌های دانشجویی، رباتیک و پروژه‌های تحقیقاتی کاربرد دارد.

تاریخچه

ایده ساخت آردوینو در سال ۲۰۰۳ میلادی در انستیتو طراحی تعاملی ایورثا در کشور ایتالیا شکل گرفت. ایده عبارت بود از ساخت وسیله‌ای ساده و کم‌هزینه برای انجام پروژه‌های دیجیتال دانشجویان، به خصوص آن‌هایی که آشنایی چندانی با اصول مهندسی و برنامه‌نویسی ندارند. سه فرد کلیدی در به ثمر نشاندن این ایده نقش داشتند:

هرناندو باراگان، ماسیمو بانزی، و کیسی ریس.

باراگان یکی از دانشجویان انستیتو ایورثا بود که تصمیم گرفت پایان‌نامه کارشناسی ارشد خود را در این زمینه اجراء نماید. بانزی و ریس نیز استادان راهنمای پایان‌نامه باراگان بودند. تا آن زمان هنوز اسمی از آردوینو در میان نبود. نتیجه پایان‌نامه باراگان بسیار موفقیت‌آمیز بود و منجر به ایجاد سخت‌افزار و نرم‌افزاری شد که وایرینگ نام گرفت. سخت‌افزار وایرینگ ویژگی‌های مورد نظر را نسبت به سایر نمونه‌های موجود در بازار آن زمان داشت یعنی ساده و کم‌هزینه بود. نرم‌افزار وایرینگ نیز بر مبنای یکی از زبان‌های برنامه‌نویسی موجود به نام پراسسینگ تهیه شده بود.

پس از اتمام پایان نامه، بانزی درصدد کاهش هزینه های سخت افزار وایرینگ برآمد و در سال ۲۰۰۵ میلادی با همکاری دیوید کوآرتلس و دیوید ملیس (که به ترتیب کارمند و دانشجوی انستیتو ایورثا بودند)، به توسعه پروژه وایرینگ پرداخت و نام آن را به آردوینو تغییر داد. این نام جدید برگرفته از نام کافه ای به نام آردوین در شهر ایورثا بود که اکثر جلسات گروه در آنجا تشکیل می شد. واژه آردوین، نام یکی از شاهزادگان قدیم ایتالیا است که زمانی حکمران شهر ایورثا بود و در قرن یازدهم میلادی به پادشاهی ایتالیا رسید.

نرم افزار آردوینو

آردوینو، یک نرم افزار متن باز اختصاصی برای برنامه نویسی بردهای خود تهیه کرده است که به نام نرم افزار آردوینو IDE (محیط توسعه یکپارچه آردوینو) شناخته می شود. نرم افزار آردوینو بدون نیاز به پروگرامر می تواند مستقیماً میکرو کنترلر شما را برنامه ریزی نماید. امکاناتی مانند ترمینال سریال نیز در داخل نرم افزار آردوینو قرار داده شده تا بتوانید به وسیله ارتباط سریال و USB برنامه هایی را که برای میکرو کنترلر می نویسید عیب یابی نمایید. می توان مقادیر ADC و وضعیت پین ها یا هر اطلاعات دیگری را به ترمینال سریال آردوینو فرستاد و از طریق کامپیوتر اطلاعات را از میکرو کنترلر دریافت کرد. همان طور که پیش تر اشاره شد، شیوه نگارش دستورها در این نرم افزار، تا حدود زیادی مشابه برنامه نویسی به زبان C و C++ است.



سخت افزار آردوینو

همان طور که در بالا گفته شد، آردوینو از دو بخش سخت افزاری و نرم افزاری تشکیل می شود. مدل های مختلفی از بردهای سخت افزاری آردوینو وجود دارد که یکی از آنها، مدل اونیو است. آردوینو اونیو (Arduino UNO) یکی از پرکاربردترین مدل های سخت افزار آردوینو است. در این برد، از یک میکرو کنترلر ATmega328P به عنوان پردازشگر و حافظه استفاده می شود. آردوینو مدل اونیو، یکی از متداول ترین نمونه ها برای آموزش آردوینو است. سخت افزار آردوینو اونیو از چهار بخش کلی زیر تشکیل می شود:

میکرو کنترلر

بخش تغذیه

بخش ارتباط با رایانه

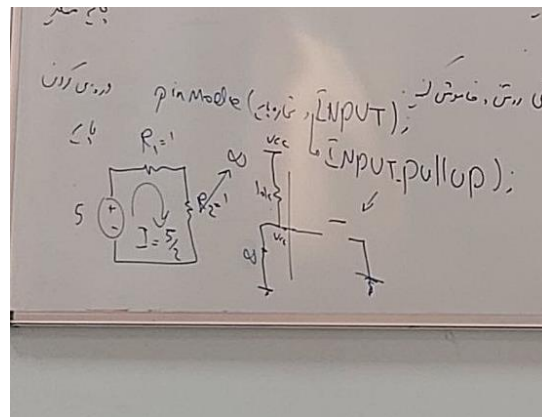
درگاه های گروهی

آزمایش شماره 1:

عنوان: کار با پورت‌های AVR

- الف - توسط یکی از پایه‌های AVR یک LED را یک ثانیه روشن و یک ثانیه خاموش کنید. (به‌طور متناوب)
- ب - توسط یک Push Button یک LED را به‌صورت لحظه‌ای روشن و خاموش کنید.
- ج - توسط یک Push Button یک LED را به‌صورت دائمی روشن و خاموش کنید.
- د - توسط یک Push Button یک LED را از طریق رله به‌صورت دائمی روشن و خاموش کنید.

مدار اولیه‌ی پیاده‌سازی بخش‌های نخست این آزمایش بدین صورت است:



در ادامه تصاویر بخش‌هایی از اقدامات عملی و کدهای نوشته شده در محیط Arduino IDE جهت پیاده‌سازی خواسته‌های آزمایش را با هم مرور می‌کنیم:

```
Ex-1

This example code is in the public domain.

https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(2, INPUT_PULLUP);
}

// the loop function runs over and over again forever
void loop() {
  if(digitalRead(2)==LOW)
    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  else
    // wait for a second
    digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
    // wait for a second
}

Done Saving

Sketch uses 1516 bytes (0%) of program storage space. Maximum is 253952 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 8183 bytes for local v
```

```
Ex-1 $

void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(2, INPUT_PULLUP);
}

// the loop function runs over and over again forever
bool flag=false;
void loop() {
  if(digitalRead(2)==LOW)
  {
    delay(100);
    if (digitalRead(2)==LOW)
    {
      while (digitalRead(2)==LOW);
      flag=!flag;
    }
    digitalWrite(LED_BUILTIN, flag); // turn the LED on (HIGH is the voltage level)
    // turn the LED off by making the voltage LOW
    // wait for a second
  }
}

Done uploading.

Sketch uses 1700 bytes (0%) of program storage space. Maximum is 253952 bytes.
Global variables use 10 bytes (0%) of dynamic memory, leaving 8182 bytes for local va
```