



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

دستور کار آزمایشگاه ریزپردازنده

نگارش

امیرحسین پولاد، علی انصاری، ریحانه باقری، محمدحسین صباغی،
امید خباز قانع، امیررضا طربخواه، علیرضا جلوخانی

زمستان ۱۴۰۱

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

فهرست مطالب

۱	مقدمات و نکات امنیتی	۱
۱-۱	مسائل مقدماتی در آزمایشگاه ریزپردازنده	۲
۱-۱-۱	قوانین آزمایشگاه ریزپردازنده	۲
۱-۱-۲	آشنایی با بردبورد	۲
۱-۱-۳	آشنایی با برد Arduino Uno	۳
۱-۱-۴	آشنایی با محیط Arduino IDE	۳
۲-۱	نکات امنیتی کار با قطعات الکترونیکی	۵
۱-۲-۱	دید نور	۵
۲-۲-۱	جریان خروجی از پین های میکروکنترلر	۵
۳-۲-۱	راهنمای پین های آردوینو	۵
۲	دستور کار آزمایش ها	۷
۱-۲	کار با پایه های ورودی خروجی به روش های سرکشی و وقفه محور	۸
۱-۲-۱	اهداف آزمایش	۸
۱-۲-۲	قطعات مورد نیاز	۸
۱-۲-۳	مقدمه	۹
۱-۲-۴	شرح آزمایش	۱۱
۲-۲	صفحه نمایش LCD و صفحه کلید ماتریسی	۱۳
۱-۲-۲	اهداف آزمایش	۱۳
۲-۲-۲	قطعات مورد نیاز	۱۳
۲-۲-۳	مقدمه	۱۴
۲-۲-۴	شرح آزمایش	۱۸
۳-۲	ارتباط بین دو آردوینو با استفاده از پروتکل SPI	۲۰
۱-۳-۲	اهداف آزمایش	۲۰
۲-۳-۲	قطعات مورد نیاز	۲۰
۳-۳-۲	مقدمه	۲۰
۴-۳-۲	شرح آزمایش	۲۱
۴-۲	کتابخوان	۲۲
۱-۴-۲	اهداف آزمایش	۲۲
۲-۴-۲	قطعات مورد نیاز	۲۲
۳-۴-۲	مقدمه	۲۲
۴-۴-۲	شرح آزمایش	۲۵
۵-۲	کار با سنسور LM35 و استفاده از آن در ساخت فن هوشمند بر اساس دما	۲۶

۲۶	۱-۵-۲ اهداف آزمایش
۲۶	۲-۵-۲ قطعات مورد نیاز
۲۶	۳-۵-۲ مقدمه
۲۸	۴-۵-۲ شرح آزمایش
۳۱	۶-۲ سروو موتور
۳۱	۱-۶-۲ اهداف آزمایش
۳۱	۲-۶-۲ قطعات مورد نیاز
۳۱	۳-۶-۲ مقدمه
۳۳	۴-۶-۲ شرح آزمایش

فصل اول

مقدمات و نکات امنیتی

۱-۱ مسائل مقدماتی در آزمایشگاه ریزپردازنده

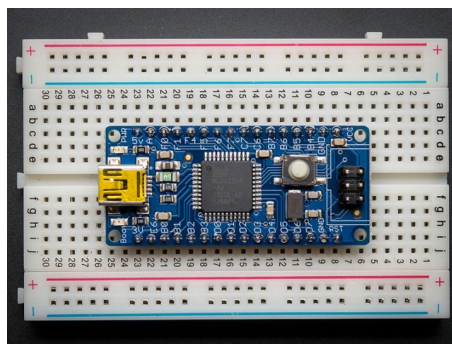
۱-۱-۱ قوانین آزمایشگاه ریزپردازنده

به منظور افزایش کارایی درس آزمایشگاه ریزپردازنده، رعایت عدالت میان همه‌ی گروه‌های آزمایشگاه و آموزش حداکثری مطالب درس به صورت عمل، مدرسی و دانشجویان ملزم به رعایت نکات و قوانین زیر هستند:

۱. مدرسین و دانشجویان موظفند راس ساعت در کلاس حاضر شوند.
۲. غیبت در هیچ یک از جلسات آزمایشگاه مجاز نیست.
۳. آزمایش‌ها در گروه‌های دو نفره انجام می‌شوند.
۴. ارائه‌ی پیش‌گزارش به صورت انفرادی است و هر دانشجو موظف است برای هر آزمایش پیش‌گزارش آن را تهیه نماید. پیش‌گزارش شامل جواب سوال‌هایی که با رنگ قرمز در قسمت مقدمه‌ی هر آزمایش آمده می‌شود.
۵. نیازی به ارائه‌ی گزارش کار برای آزمایش‌ها نیست و تحویل خروجی صحیح به مدرس آزمایشگاه کفایت می‌کند.

۲-۱-۱ آشنایی با بردبورد

در دنیای سیستم‌های نهفته و مدارهای الکترونیکی معمولاً برای سریع‌تر شدن فرایند توسعه، ابتدا یک نمونه‌ی اولیه با بردبورد می‌سازند و بعد از توسعه‌ی محصول نهایی سراغ تهیه‌ی مدار چاپی و یا روش‌های دیگر می‌روند. ما نیز در آزمایشگاه ریزپردازنده آزمایش‌های گفته شده را با استفاده از بردبورد انجام می‌دهیم.



شکل ۱-۱: تصویر یک بردبورد که روی آن یک میکروکنترلر قرار داده شده است.

بردبورد در واقع یک مدار با اتصالات از پیش تعیین شده است، به طوری که سوراخ‌های میانی به صورت عمودی و سوراخ‌های دو ردیف بالا و پایین به صورت افقی به یکدیگر اتصال الکتریکی دارند.

پیشنهاد می‌شود که از این دو ردیف برای اتصال مثبت و منفی منبع ولتاژ استفاده کنید. عمدتاً در این درس مثبت و منفی از پین‌های 5V و GND میکروکنترلر تامین می‌شود اما می‌تواند به هر صورت دیگری نیز باشد.

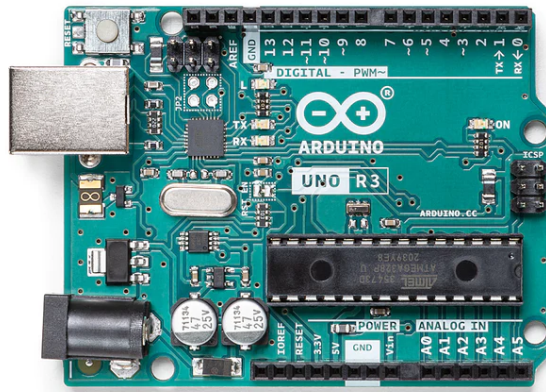
۳-۱-۱ آشنایی با برد Arduino Uno

در این آزمایشگاه از برد Arduino Uno استفاده می‌کنیم. این برد مبتنی بر میکروکنترلر ATmega328P است که شامل ۱۴ پین خروجی و ورودی است که ۶ تا از آنها قابلیت خروجی PWM و ۶ تا از آنها قابلیت خروجی آنالوگ دارند. همچنین این میکروکنترلر ۳۲ کیلوبایت حافظه‌ی فلش (که از آن ۵.۰ کیلوبایت برای برنامه‌ی بوتلودر استفاده شده) و ۲ کیلوبایت SRAM و ۱ کیلوبایت حافظه‌ی EEPROM دارد.

ATmega328P	Microcontroller
5V	Operating Voltage
7-12V	Input Voltage (recommended)
6-20V	Input Voltage (limit)
14 (of which 6 provide PWM output)	Digital I/O Pins
6	PWM Digital I/O Pins
6	Analog Input Pins
20 mA	DC Current per I/O Pin
50mA	DC Current for 3.3V Pin
32 KB (ATmega328P) of which 0.5 KB used by bootloader	Flash Memory
2 KB (ATmega328P)	SRAM
1 KB (ATmega328P)	EEPROM
16 MHz	Clock Speed

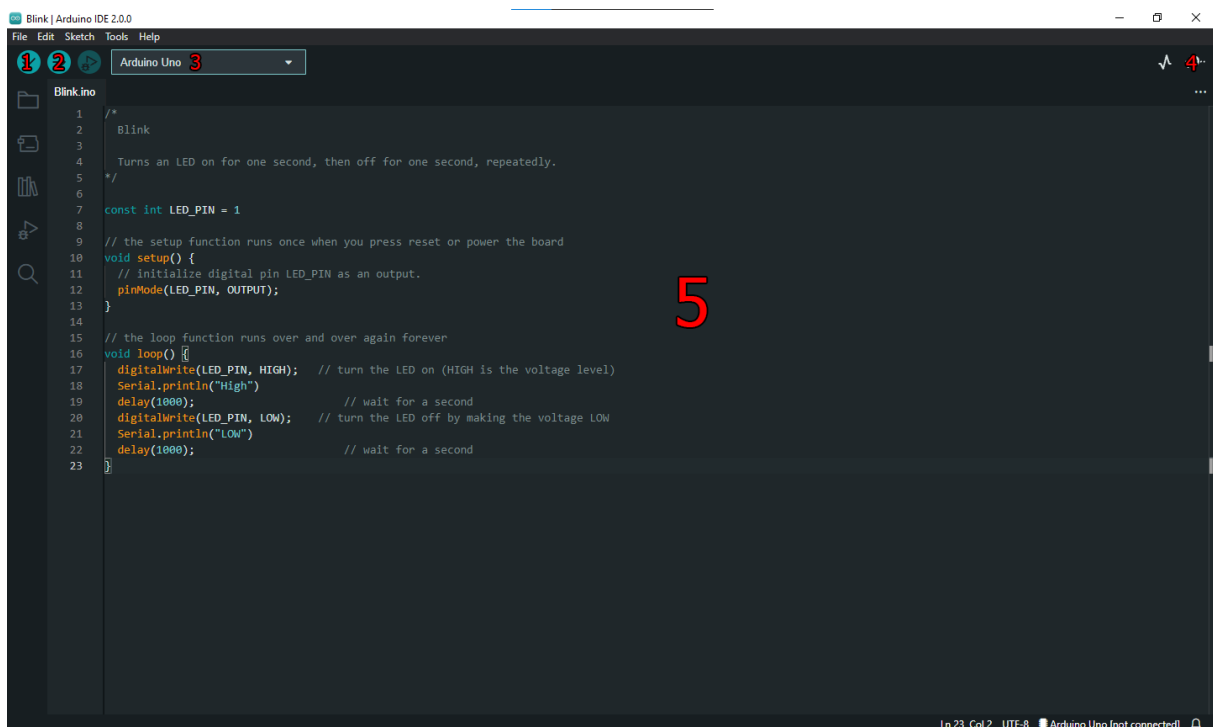
۴-۱-۱ آشنایی با محیط Arduino IDE

برد‌های آردوینو را می‌توان به راحتی با استفاده از برنامه‌ی Arduino IDE برنامه‌ریزی کرد و نقطه‌ی قوت اصلی آن نسبت به پلتفرم‌های دیگر نیز همین سهولت در برنامه‌ریزی آنهاست. در ابتدا شما باید ابتدا



شکل ۱-۲: تصویری از برد Arduino Uno

این برنامه را از [این لینک](#) دریافت کرده و نصب کنید و سپس آن را اجرا کنید. پس از اجرای برنامه با محیط زیر روبرو می‌شوید:



شکل ۱-۳: تصویری از محیط Arduino IDE

۱. کامپایل برنامه

۲. کامپایل برنامه و آپلود آن بر روی برد

۳. انتخاب برد

۴. سریال مانیتور

۵. ادیتور کد

با این محیط در ویدیو های آموزشی که در اختیار شما قرار می گیرد بیشتر آشنا خواهیم شد.

۲-۱ نکات امنیتی کار با قطعات الکترونیکی

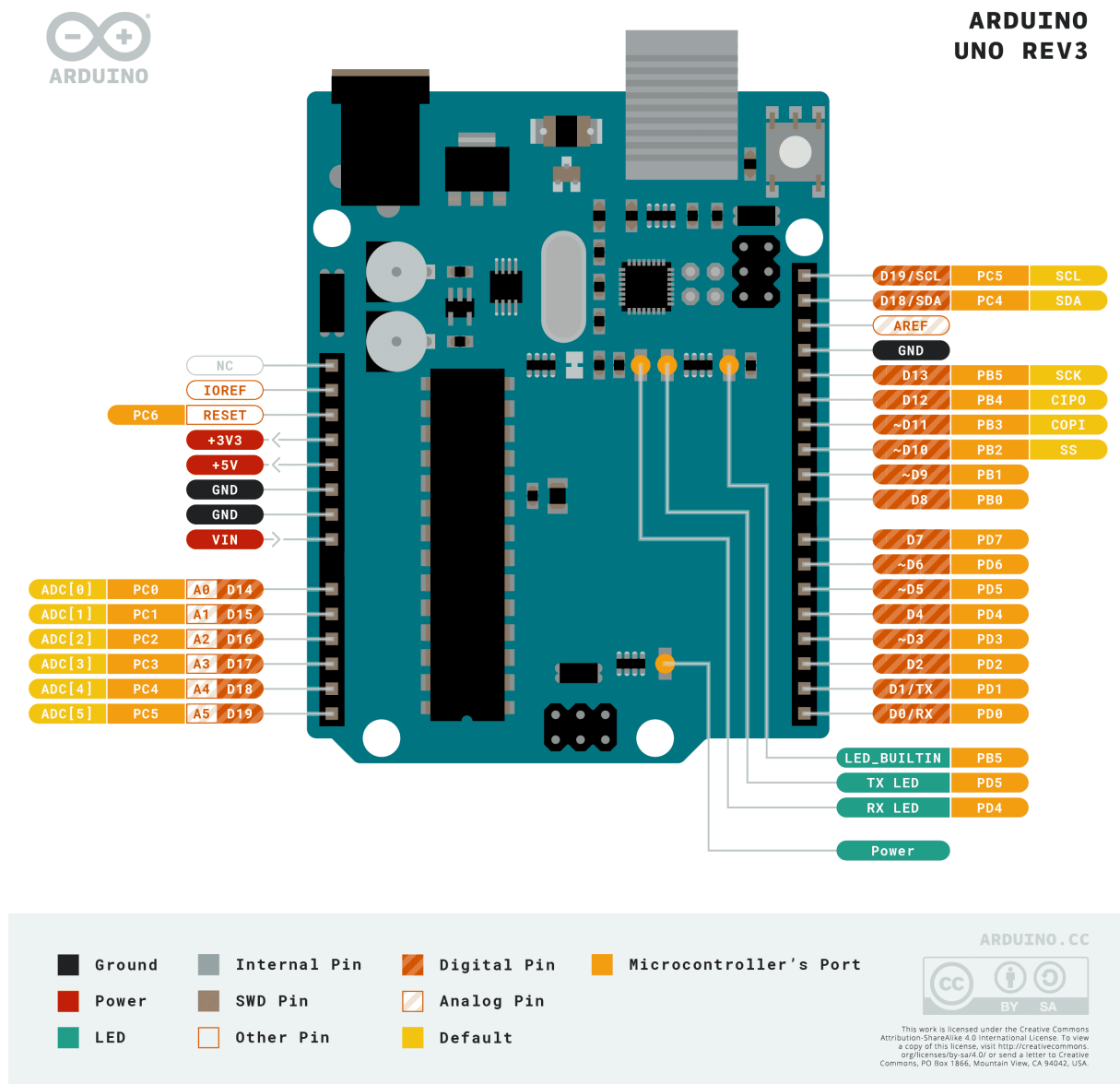
۱-۲-۱ دیود نوری

دیود های نوری یک حداکثر جریان قابل تحمل دارند و اگر جریان عبور از آنها از یک مقدار معین بیشتر شود می سوزند. برای همین همواره از یک مقاومت سری حداقل ۱۰۰ اهمی برای محدود کردن جریان دیود استفاده می کنیم. در آزمایش اول بیشتر با نحوه ی محاسبه ی مقاومت مورد نیاز برای دیود های نوری آشنا می شویم.

۲-۲-۱ جریان خروجی از پین های میکروکنترلر

پین های ورودی و خروجی میکروکنترلر ها محدودیت جریانی ۲۰ میلی آمپری دارند. توجه کنید و به هیچ وجه مداری نبندید که جریان خروجی از پین های میکروکنترلر از ۲۰ میلی آمپر بیشتر شوند. در غیر این صورت امکان خرابی قطعه وجود دارد و عواقب آن بر عهده ی خودتان است. برای اینکه مدار هایی با جریان عبوری بیشتر را توسط میکروکنترلر کنترل کنیم باید از ترانزیستور و یا رله استفاده کنیم که در آزمایش های آینده بیشتر با آن آشنا خواهیم شد.

۳-۲-۱ راهنمای پین های آردوینو



شکل ۴-۱: راهنمای پین های آردوینو

فصل دوم

دستور کار آزمایش ها

۱-۲ کار با پایه های ورودی خروجی به روش های سرکشی و وقفه

محور

۱-۱-۲ اهداف آزمایش

- آشنایی با GPIO
- آشنایی با روش های سرکشی و وقفه محور برای مدیریت Peripheral های جانبی
- مقایسه ی دو روش سرکشی و وقفه محور

۲-۱-۲ قطعات مورد نیاز

- برد Arduino Uno
- دیود نورانی LED
- کلید Switch
- مقاومت ۲۲۰ اهم
- مقاومت ۱۰ هزار اهم

۳-۱-۲ مقدمه

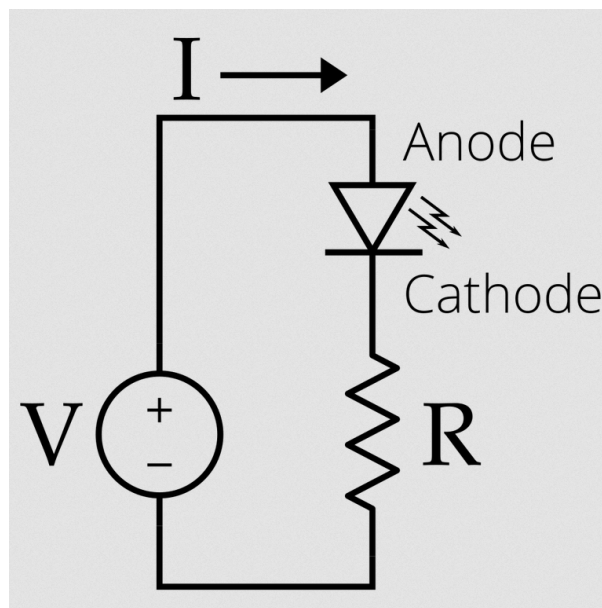
دو تابع اصلی در برنامه‌نویسی آردوینو

محیط آردوینو برای راحت تر شدن برنامه‌نویسی ۲ تابع مهم در اختیار ما می‌گذارد. نام این دو تابع begin و loop است.

سوال: در مورد این دو تابع تحقیق کنید و بنویسید که هر کدام در چه زمانی اجرا می‌شوند.

دیود نوری

دیود های نوری نوعی دیود هستند که هنگام عبور جریان از پایانه‌ی مثبت به پایانه‌ی منفی آنها، از خود نور ساطع می‌کنند. خواص الکتریکی دیود های نوری مشابه با دیود های دیگر است. این یعنی می‌توان یک دیود نوری روشن را به صورت یک منبع ولتاژ مدل سازی کرد. همچنین جریان عبوی از دیود های نوری دارای یک محدودیت است و اگر جریان بیشتر شود دیود نوری می‌سوزد و به همین دلیل باید در مدار شامل دیود نوری یک مقاومت حدوداً ۲۲۰ اهمی قرار دهیم. معمولاً ولتاژ دیود های نوری بین ۲ تا ۳ ولت و یک جریان مناسب و به دور از خطر برای آنها ۲۰ میلی‌آمپر است. برای اینکه بفهمیم مقدار ۲۲۰ اهم از کجا آمده، مدار زیر را در نظر بگیرید.

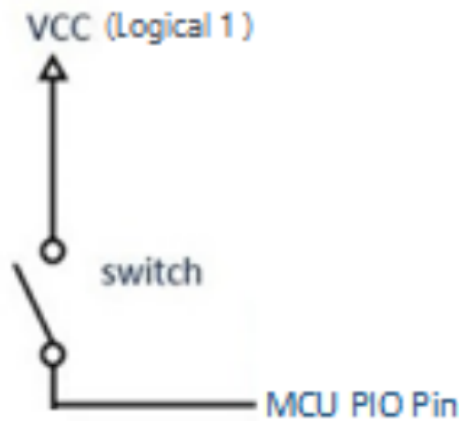


شکل ۱-۲: مدار دیود نوری

سوال: با فرض اینکه ولتاژ دیود ۲ ولت باشد و منبع ولتاژ ۵ ولتی باشد، مقدار مقاومت سری را به گونه‌ای بیابید که جریان مدار ۱۵ میلی‌آمپر شود.

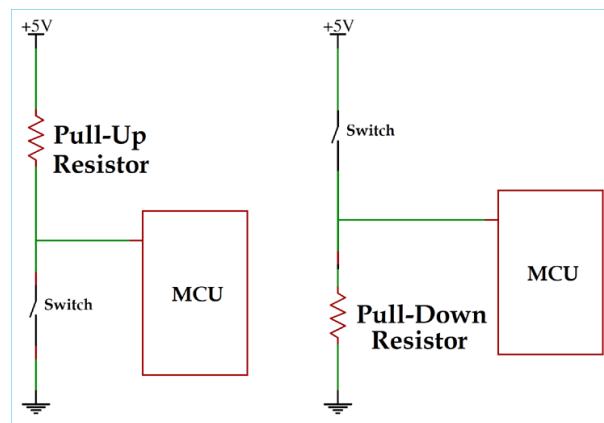
توجه کنید که دیود های نوری دو پایه دارند و پایه‌ای که بلند تر است پایانه‌ی مثبت دیود (آنود) و پایه‌ای که کوتاه تر است پایانه‌ی منفی (کاتود) دیود است. به هیچ عنوان دیود های نوری را به صورت برعکس در مدار قرار ندهید.

اتصال صحیح کلید به برد
مدار زیر را در نظر بگیرید.



شکل ۲-۲: اتصال مستقیم کلید

سوال: اگر کلید باز باشد ولتاژ MCU PIO Pin چیست؟ این اتفاق چه مشکلی را پیش می‌آورد؟
برای حل مشکل مدار قبلی، باید از مقاومت های Pull Up و Pull Down استفاده کنیم. همانطور که از اسامی آنها پیداست، این مقاومت ها در هنگام قطع بودن کلید، ولتاژ را به بالا و یا به پایین می‌کشند. مدار های زیر را در نظر بگیرید:



شکل ۲-۳: اتصال کلید با مقاومت های Pull Up و Pull Down

سوال: ولتاژ ورودی به میکروکنترلر را در هر یک از مدار های بالا در حالت بسته و یا باز بودن میکروکنترلر محاسبه کنید.

سوال: بهتر است مقدار مقاومت استفاده شده مقدار بالایی باشد. دلیل این کار چیست؟

در نتیجه همواره برای اتصال یک کلید به میکروکنترلر از یکی از روش های Pull Up و یا Pull Down استفاده می کنیم.

ورودی و خروجی

استفاده از پین های ورودی و خروجی از محیط آردوینو بسیار ساده است و با صدا زدن چند تابع انجام می شود.

سوال: در مورد توابع زیر تحقیق کنید و ورودی و خروجی هر تابع و کاری که انجام می دهند را بنویسید.

• pinMode

• digitalRead

• digitalWrite

• delay

وقفه

با وقفه ها در کلاس درس آشنا شدید. در سطوح پایین تر و نزدیک به سخت افزار به دلیل متفاوت بودن میکروکنترلر مورد استفاده در آزمایشگاه و میکروکنترلری که در کلاس درس تدریس شد، استفاده از وقفه ها مقداری متفاوت است اما به صورت کلی فرایند همان است.

سوال: انواع رویداد های ورودی که میکروکنترلر موجود در برد Arduino Uno می تواند تشخیص دهد و اعلام وقفه کند بنویسید.

سوال: کدام یک از پین های برد شما قابلیت تشخیص وقفه را دارند؟

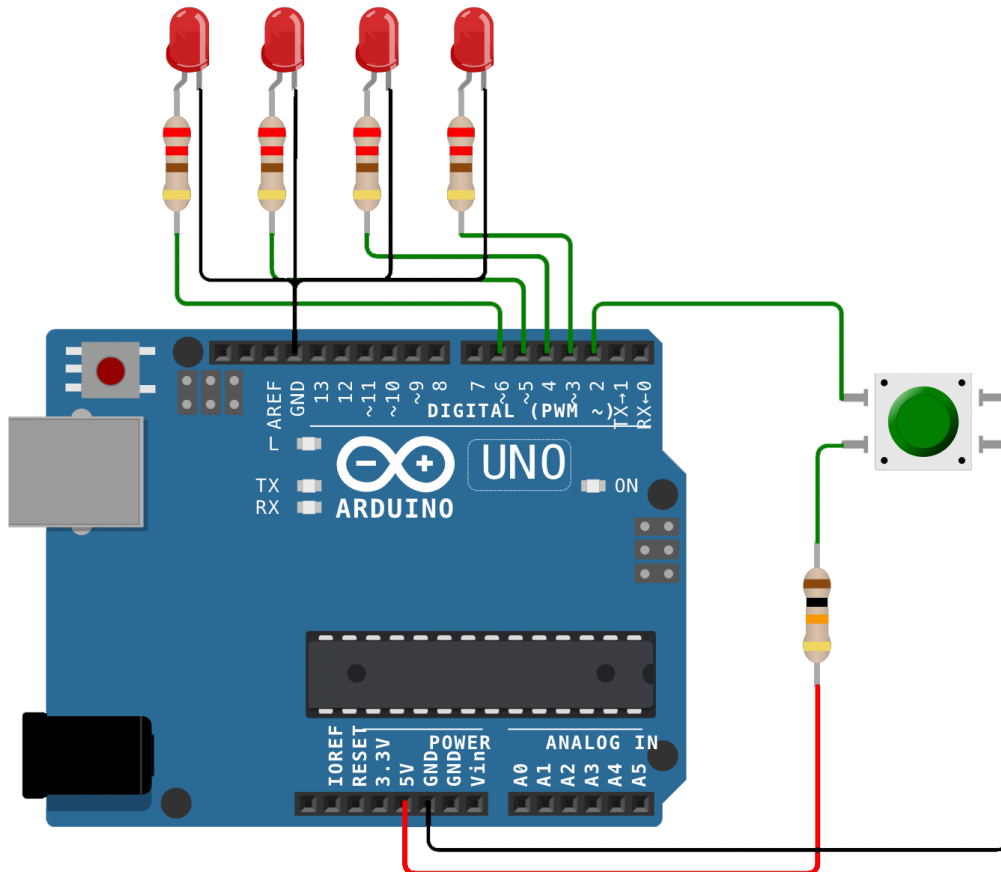
در محیط آردوینو استفاده از وقفه های مربوط به پین ورودی با استفاده از تابع attachInterrupt صورت می گیرد.

سوال: در مورد تابع attachInterrupt تحقیق کنید و ورودی و خروجی تابع و کاری که انجام می دهد را بنویسید.

۴-۱-۲ شرح آزمایش

می خواهیم با استفاده از دیود نوری و کلید، یک شمارنده ی ۴ بیتی بسازیم. برای این کار، یک کلید را به همراه مقاومت Pull Up یا Pull Down به برد متصل کنید و ۴ دیود نوری را نیز به هر یک از پین های برد وصل کنید. فراموش نکنید که مقاومت سری ۲۲۰ اهمی را در مدار دیود های نوری قرار دهید. سپس برنامه ای بنویسید که با استفاده از این ۴ دیود نوری از ۰ تا ۱۵ به صورت باینری شمارش کند و با هر بار فشردن کلید، یکی به عدد نمایش داده شده اضافه شود. برنامه را ابتدا به روش سرکشی و سپس به روش وقفه محور بنویسید.

- به نظر شما اگر برنامه‌ی ما می‌خواست چند کار دیگر را نیز انجام دهد و شمردن با دیود نوری فقط یکی از وظایف آن بود، بهتر بود از کدام روش استفاده می‌کردیم؟
- اگر دکمه را برای مدت طولانی نگه دارید چه اتفاقی می‌افتد؟ دلیل این اتفاق چیست؟ برای حل این مشکل چه راه حلی دارید؟



شکل ۲-۴: مدار آزمایش اول

۲-۲ صفحه نمایش LCD و صفحه کلید ماتریسی

۱-۲-۲ اهداف آزمایش

- آشنایی با صفحه نمایش LCD و صفحه کلید ماتریسی به عنوان رابط های کاربری
- آشنایی با نحوه ی اضافه کردن کتابخانه ها در محیط آردوینو

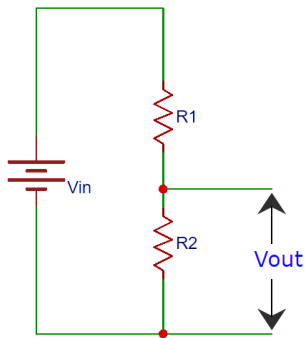
۲-۲-۲ قطعات مورد نیاز

- Arduino Uno
- صفحه نمایش LCD
- صفحه کلید ماتریسی
- پتانسیومتر ۱ کیلو اهم B1K
- مقاومت ۲۲۰ اهم

۳-۲-۲ مقدمه

یاد آوری مقسم ولتاژ و پتانسیومتر

در درس و آزمایشگاه مدار های الکتریکی با مدار مقسم ولتاژ آشنا شدید. این مدار در واقع از ۲ مقاومت سری تشکیل شده که ولتاژ ورودی میان آنها تقسیم می شود.

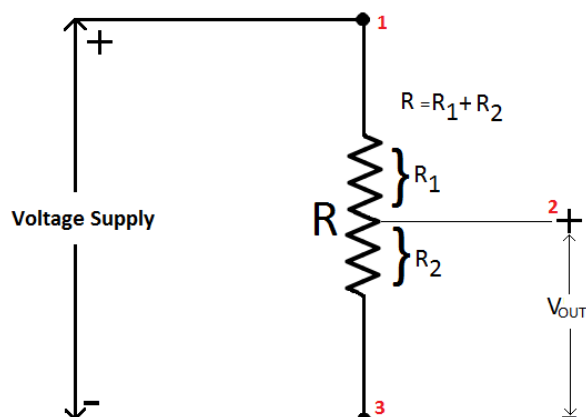


شکل ۲-۵: مدار مقسم ولتاژ

با استفاده از KVL به سادگی به فرمول زیر می رسیم:

$$V_{out} = V_{in} \times \frac{R_2}{R_1 + R_2} \quad (۱-۲)$$

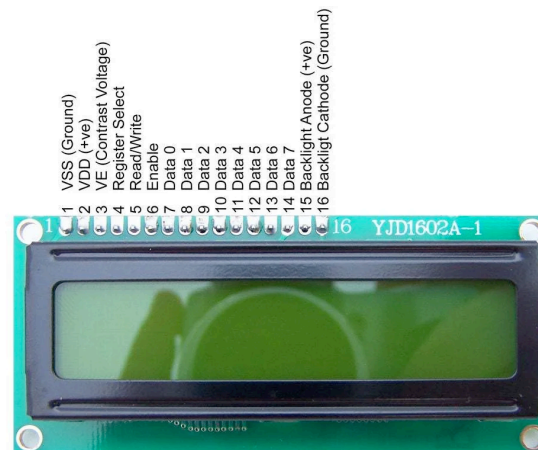
استفاده ای اصلی ما از این مدار، کم کردن ولتاژ ورودی به یک وسیله است. پتانسیومتر ها مقسم های ولتاژی هستند که مقدار مقاومت های R_1 و R_2 در آنها (معمولا با چرخاندن سر پتانسیومتر) قابل تغییر است و در این آزمایش برای کنترل کنتراست صفحه نمایش از آن استفاده می کنیم.



شکل ۲-۶: مدار معادل یک پتانسیومتر. توجه کنید هنگامی که می گوئیم یک پتانسیومتر ۱۰ کیلو اهمی، ۱۰ کیلو اهم بیانگر مقدار R است.

صفحه نمایش LCD

صفحه نمایش کاراکتری 16x2 یکی از روش های متداول ارتباط با کاربر در سیستم های نهفته است. منظور از 16x2 اندازه ی صفحه نمایش است، یعنی این صفحه های نمایش قابلیت نشان دادن ۲ ردیف و ۱۶ ستون از کاراکتر ها را دارند. خود صفحه نمایش توسط یک IC کنترل می شود که میکروکنترلر ما باید با آن ارتباط برقرار کند. ابتدا با پین های ورودی یک ماژول صفحه نمایش کاراکتری آشنا می شویم:



شکل ۲-۷: نمایی از یک صفحه نمایش کاراکتری ۱۶ در ۲

- پین شماره ۱: ولتاژ Ground
- پین شماره ۲: ولتاژ Vcc
- پین شماره ۳: کنترل کنتراست. با استفاده از یک پتانسیومتر می توانیم کنتراست صفحه نمایش را کنترل کنیم
- پین های شماره ۴ تا ۶: پین های کنترلی
- پین های شماره ۷ تا ۱۴: پین های ورودی داده
- پین های شماره ۱۵ و ۱۶: به ترتیب آنود و کاتود Backlight LED. توجه کنید این پین ها مستقیماً به یک LED متصل می شوند و ارتباطی با آی سی کنترلر ندارند. طبیعتاً مانند دیگر LED ها حتماً برای جلوگیری از خرابی قطعه باید یک مقاومت با آن سری قرار دهید.

از آنجایی که ما از کتابخانه های مخصوص ارتباط با LCD استفاده خواهیم کرد، نیازی به دانستن جزئیات این ارتباط نیست و در صورت علاقه می توانید خودتان از دیتاشیت مربوطه مطالعه کنید، اما باید نحوه ی کار با کتابخانه را بلد باشید و بتوانید از آن استفاده کنید.

سوال:

در مورد توابع زیر از کتابخانه ی LiquidCrystal.h تحقیق کنید و ورودی و خروجی توابع و کاری که انجام می دهند را بنویسید.

LiquidCrystal() •

begin() •

clear() •

home() •

setCursor() •

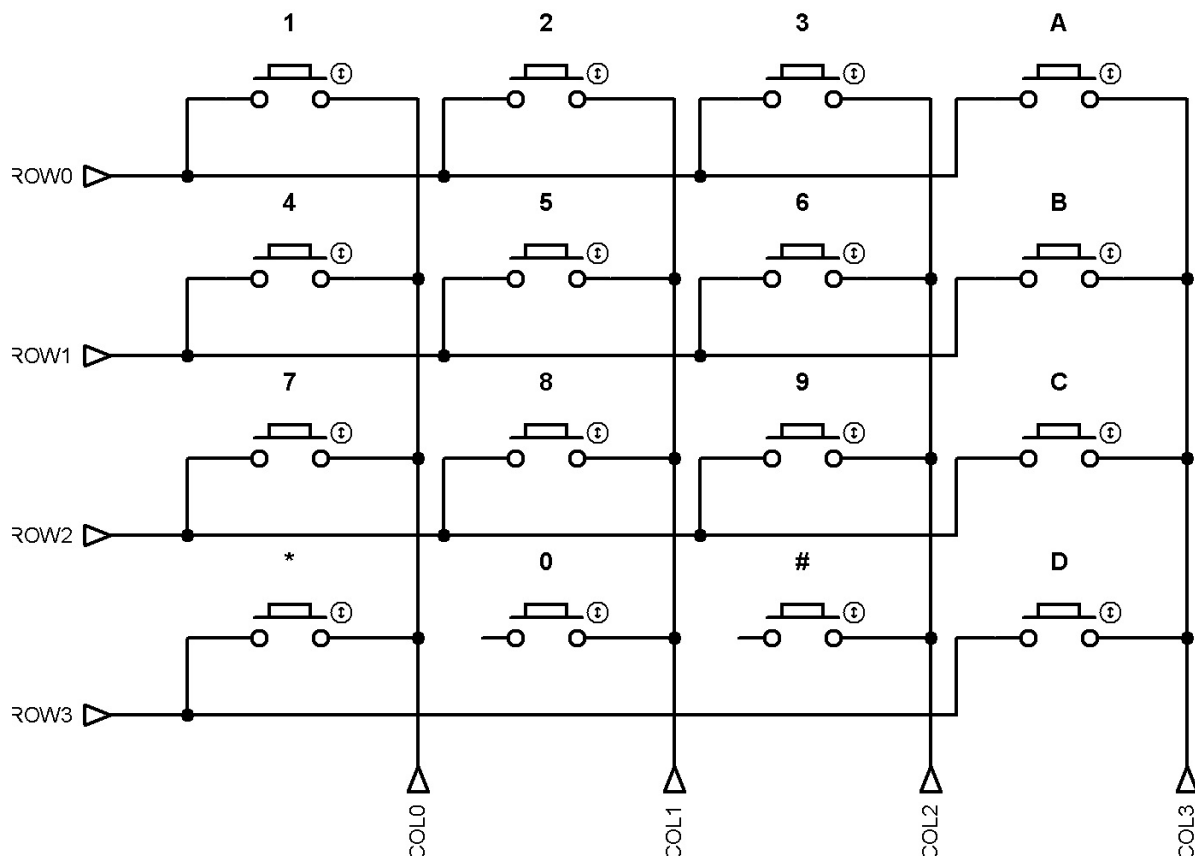
write() •

print() •

صفحه کلید ماتریسی

در آزمایش قبلی از کلید های معمولی برای گرفتن ورودی از کاربر استفاده کردیم. اگر بخواهیم مانند روش آزمایش قبل ۱۶ عدد کلید برای ورودی داشته باشیم به مشکل می خوریم، چون تعداد پین های ورودی میکروکنترلر ما محدود است. به همین دلیل باید از روش ماتریسی برای اتصال کلید ها به میکروکنترلر استفاده کنیم.

در این روش کلید ها به صورت زیر در یک جدول قرار می گیرند:



شکل ۲-۸: مدار یک صفحه کلید ماتریسی

برای استفاده از این مدار، باید به همه ی سطر ها ولتاژ صفر منطقی بدهیم و ستون ها را با استفاده از Pull Up Resistor به ورودی های میکروکنترلر وصل کنیم. هنگامی که یک کلید بسته می شود سطر و ستون متناظر با آن متصل می شوند و ورودی آن ستون به میکروکنترلر از یک به صفر منطقی تغییر پیدا می کند. در این مرحله می دانیم که یک کلید در ستونی که ورودیش تغییر کرده روشن شده است. برای اینکه سطر آن را نیز پیدا کنیم و بفهمیم کدام کلید روشن شده، یک به یک سطر ها را به یک منطقی تغییر می دهیم. اگر هنگامی که یک سطر به یک منطقی تغییر کرد، ستون نیز به یک منطقی تغییر کند سطری که کلید در آن قرار دارد را نیز پیدا کرده ایم و تشخیص دادیم که کدام کلید بسته شده بود.

سوال: شبه کد/فلوچارت استفاده از صفحه کلید ماتریسی را بنویسید.

برای کار با صفحه کلید نیز از کتابخانه های مخصوص به آن استفاده می کنیم که نام آن Keypad.h است. برای دریافت این کتابخانه، از منوی سمت راست Arduino IDE به قسمت Library Manager بروید و کتابخانه ی Keypad by Mark Stanley را دریافت نمایید. مستندات مربوط به این کتابخانه را می توانید در **این لینک** مشاهده کنید.

سوال: در مورد توابع زیر از کتابخانه ی Keypad.h تحقیق کنید و ورودی و خروجی توابع و کاری که انجام می دهند را بنویسید.

• Keypad()

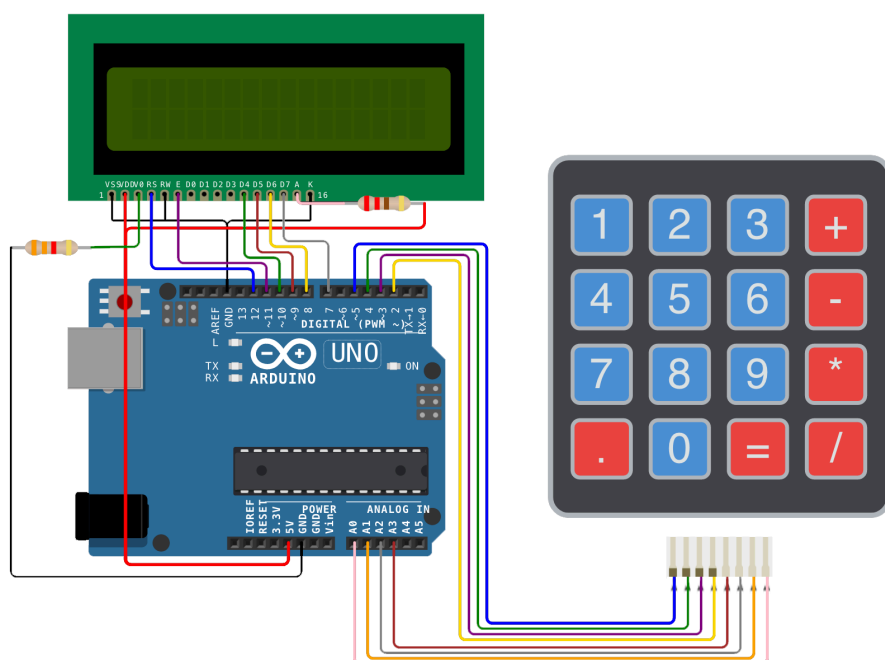
• begin()

• waitForKey()

• getKey()

۴-۲-۲ شرح آزمایش

می خواهیم یک ماشین حساب درست کنیم. ابتدا مانند مداری که در انتهای آزمایش در اختیار شما قرار گرفته، صفحه نمایش و صفحه کلید را به آردوینو متصل کنید. نیازی نیست مدار شما دقیقا همان مدار باشد، اما توجه کنید که به هیچ عنوان قرار دادن مقاومت و پتانسیومتر در مدار را فراموش نکنید. در صورتی که پتانسیومتر ندارید، می توانید به پین Vo یک مقاومت ۳.۳ کیلو اهمی متصل به زمین وصل کنید. همچنین وقتی صفحه کلید ماتریسی رو به روی شما باشد و پین ها رو به پایین باشند، پین ها از چپ به راست ابتدا پین های ردیف ها هستند و سپس پین های ستون ها هستند. برنامه ای بنویسید که کلید ها را از صفحه کلید بخواند و آنها را روی صفحه نمایش نمایش دهد و بعد از اینکه کلید = زده شد، عبارت ریاضی که کاربر وارد کرده را محاسبه کرده و مقدار آن را روی صفحه نمایش نمایش دهد. توجه کنید نیازی به رعایت اولویت ها و پشتیبانی از اعداد اعشاری و اعداد منفی نیست.



شکل ۲-۹: مدار مربوط به آزمایش دوم

۳-۲ ارتباط بین دو آردوینو با استفاده از پروتکل SPI

۱-۳-۲ اهداف آزمایش

- آشنایی با پروتکل ارتباطی SPI
- آشنایی با پروتکل سریال و ارتباط با کامپیوتر از طریق آن

۲-۳-۲ قطعات مورد نیاز

- Arduino Uno

۳-۳-۲ مقدمه

تا اینجا آردوینوی ما ارتباطی با کامپیوتری که از طریق آن برنامه ها را روی آن آپلود می کردیم نداشت. می توانیم با استفاده از ارتباط سریال، این کار را انجام دهیم. برای این کار کفایت از کتابخانه سریال استفاده کنیم.

سوال: در مستندات آردوینو جست و جو کنید و در مورد توابع کتابخانه Serial توضیح دهید. با پروتکل SPI در درس ریزپردازنده آشنا شدید. همانطور که می دانید این نوع ارتباط از نوع ارتباطات Master است. در این نوع ارتباطات یک دستگاه می تواند با چند دستگاه دیگر ارتباط برقرار کند. در ارتباط SPI برد مرکزی یا همان Master بوردی که می خواهد با آن ارتباط برقرار کند را انتخاب کرده و برای آن پیامی ارسال می کند و در صورت نیاز، از آن درخواست پاسخ می کند.

سوال: در مورد پروتکل SPI به سوالات زیر پاسخ دهید:

- این پروتکل از ۴ سیم استفاده می کند. وظیفه ی هر یک از این ۴ سیم را شرح دهید.
- آیا این پروتکل امکان حضور چند Master را به ما می دهد؟
- در برد Arduino Uno کدام پین های به صورت پیش فرض برای پین های SPI تخصیص داده شده اند؟

- در این پروتکل ارتباطی سنکرون، کلاک توسط Master تعیین می شود یا Slave؟

می توانیم برای استفاده از این پروتکل، از کتابخانه SPI.h استفاده کنیم. این کتابخانه یک لایه ی انتزاع برای ارتباط با سخت افزار مخصوص SPI در میکروکنترلر های مختلف است و این اجازه را به ما می دهد که به صورت نسبتاً Portable برنامه ای بنویسیم که از این کتابخانه استفاده کند.

سوال: در مستندات این کتابخانه جست و جو کنید و در مورد توابع زیر توضیح بدهید:

- SPI.begin()

- SPI.beginTransaction()
- SPI.end()
- SPI.transfer()
- SPI.usingInterrupt()

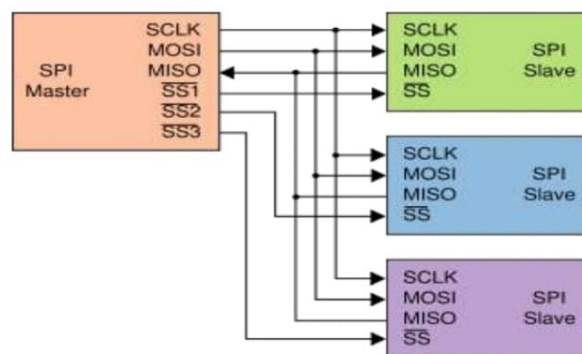
در این آزمایش باید آردوینوی ما هم در حالت Master و هم در حالت Slave کار کند.
سوال: دستوری که با استفاده از آن آردوینو در حالت Slave قرار می‌گیرد را بنویسید و آن را توضیح دهید.

۴-۳-۲ شرح آزمایش

می‌خواهیم یک پیام را از یک آردوینو به آردوینو دیگر انتقال دهیم و برای این کار از پروتکل SPI استفاده می‌کنیم. هر یک از گروه‌ها باید برنامه Master و Slave خود را درست کنند و با برنامه متناظر گروه دیگر آزمایش کنند.

برنامه Master باید با استفاده از پروتکل Serial یک متن را از ورودی بگیرد و با استفاده از SPI آن را به برنامه‌ی Slave بدهد. برنامه‌ی Slave نیز باید پیام را دریافت کرده و از طریق پروتکل Serial نمایش بدهد.

توجه داشته باشید که دستگاه Slave حتماً باید در حالت Slave قرار بگیرند. در غیر این صورت هر دو دستگاه می‌خواهند کلاک را خودشان تامین کنند و احتمال خرابی قطعات وجود دارد. می‌توانید برای اطلاعات بیشتر به **این لینک** مراجعه کنید.



شکل ۲-۱۰: یک نمونه از باس SPI

توجه کنید که در شکل پین‌های برد آردوینو که در بخش مقدمه‌ی دستور کار آمده به جای اسامی MOSI و MISO به ترتیب از CIPO و COPI استفاده شده است.

۴-۲ کتابخوان

۱-۴-۲ اهداف آزمایش

- آشنایی با پروتکل ارتباط سریال I2C
- کار با حافظه‌ی EEPROM بیرونی و نحوه‌ی خواندن از آن

۲-۴-۲ قطعات مورد نیاز

- Arduino Uno
- صفحه نمایش LCD
- پتانسیومتر ۱ کیلو اهم B1K
- مقاومت ۲۲۰ اهم
- EEPROM (AT24C16)
- کلید Switch
- مقاومت ۱۰ هزار اهم

۳-۴-۲ مقدمه

در این آزمایش می‌خواهیم با نحوه کار با حافظه EEPROM آشنا شویم و از آن برای نمایش دادن یک کتاب که متن آن درون این حافظه‌ی بیرونی ذخیره شده استفاده کنیم. برای نمایش دادن خط به خط کتاب بر روی LCD با استفاده از ۲ کلیدی که در مدار قرار می‌دهیم به خط قبلی یا خط بعدی کتاب می‌رویم.

حافظه EEPROM

در بسیاری از برنامه‌ها ما نیاز داریم که اطلاعاتی را ذخیره کنیم تا پس از اجرای مجدد برنامه از آنها استفاده کنیم. همچنین اطلاعاتی که حجم زیادی دارند را نمی‌توان در حافظه‌ی خود برد ذخیره کرد و نیاز داریم که یک حافظه بیرونی برای ذخیره این اطلاعات داشته باشیم که پس از متوقف کردن برنامه نیز اطلاعات در آن ذخیره بمانند. همچنین در دنیای میکروکنترلر ها یکی از اصلی ترین مسائلی که قیمت میکروکنترلر های مختلف را متفاوت می‌کند میزان حافظه‌ی آنهاست و یکی از راه هایی که می‌توان قیمت تمام شده‌ی یک محصول دارای یک سیستم نهفته را کمتر کرد استفاده از میکروکنترلری با حافظه‌ی کمتر و وصل کردن یک حافظه‌ی بیرونی به آن است. به همین دلیل ما به EEPROM ها

نیاز داریم.

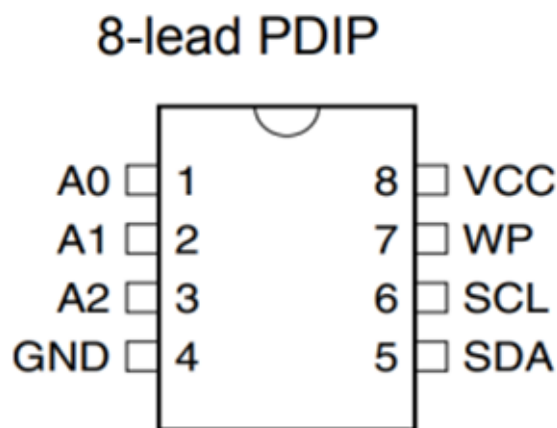
در یک دسته‌بندی می‌توان EEPROM ها را به دو نوع درونی و بیرونی دسته‌بندی کرد. حافظه‌های درونی در کنار پردازنده بر روی همان مدار مجتمع قرار می‌گیرند و نیازی به استفاده از پایه‌های آدرس ندارند. همچنین به دلیل حضور در یک مدار مجتمع اتصال بهتری میان آن‌ها و میکروکنترلر برقرار است و سرعت آن‌ها بیشتر است. برد Arduino Uno نیز یک حافظه ی EEPROM درونی به اندازه ۱ کیلوبایت دارد که در این آزمایش مورد توجه ما نیست.

گونه‌ای دیگر از EEPROM ها بیرونی هستند که بر روی یک IC جداگانه قرار می‌گیرند. برای ارتباط با این حافظه‌ها اکثراً از پروتکل‌های ارتباط سریال استفاده می‌شود. حافظه AT24C16 نیز که در این آزمایش به کار برده می‌شود، پروتکل سریال TWI را به کار می‌گیرد. به این صورت که در این پروتکل حافظه‌ها همان برده‌ها (Slaves) هستند و برد کنترلر نیز سرپرست (Master) است. در این دسته گاهی برای اینکه بتوان چند حافظه را بر روی یک باس کنترل کرد (چند برده داشت) از پایه‌های آدرس استفاده می‌شود. این آدرس به صورت سخت‌افزاری پیکربندی می‌شود (Hard-Wired).

AT24C16 از پروتکل TWI یا همان I2C استفاده می‌کند و مطابق چیزی که از این پروتکل می‌دانیم باید ۷ بیت آدرس داشته باشد. چهار بیت پر ارزش آدرس آن مقدار ثابت 0b1010 است و سه بیت کم ارزش آن را پایه‌های A0 تا A3 مشخص می‌کند. البته این پایه‌ها در واقعیت ممکن است استفاده بشوند یا نشوند. در واقع چیپ حافظه‌ی AT24C16 از این پین‌ها استفاده نمی‌کند و همواره سه بیت کم ارزش آدرس آن صفر است و چیپ AT24C08 نیز فقط از پین A2 استفاده می‌کند و دو پین دیگر همواره صفر هستند. همیشه برای اینگونه فهمیدن جزئیات کار با قطعات الکترونیکی به دیتاشیت قطعه مراجعه کنید. همچنین EEPROM پایه‌ای به نام WP دارد که مخفف Write Protection است که اگر مقدار منطقی آن برابر یک باشد و به عبارتی به آن ولتاژ وارد شود، فعال می‌شود و امکان نوشتن روی EEPROM دیگر وجود ندارد. دو تصویر از پایه‌های AT24C16 و کارکرد هرکدام از آن‌ها نمایش داده شده است.

Pin Name	Function
A0 - A2	Address Inputs
SDA	Serial Data
SCL	Serial Clock Input
WP	Write Protect
GND	Ground
VCC	Power Supply

شکل ۲-۱۱: کارکرد پین‌های حافظه



شکل ۲-۱۲: دیاگرام پین های حافظه

کار با کتابخانه TWI در Arduino

کتابخانه wire.h یکی از کتابخانه‌های استاندارد آردوینو است که نیازی به نصب آن نیست. این کتابخانه توابع لازم برای کار با پروتکل TWI را فراهم می‌کند. برخی از توابع این کتابخانه به شرح زیر است:

- begin()
- setClock()
- beginTransmission()
- write()
- endTransmission()
- requestFrom()
- available()
- read()

سوال: هر یک از توابع بالا را در مستندات کتابخانه بررسی کنید و ورودی ها، خروجی ها و عملکرد آنها را توضیح دهید.

چگونگی خواندن و نوشتن بر روی حافظه‌ی EEPROM

الگوریتم خواندن به صورت زیر است:

۱. شروع ارتباط با دستگاهی که آدرس I2C آن آدرس حافظه‌ی ما است.
 ۲. نوشتن یک بایت آدرس حافظه‌ای که می‌خواهیم بخوانیم.
 ۳. پایان ارتباط.
 ۴. درخواست تعداد بایت دلخواه از دستگاهی که آدرس I2C آن آدرس حافظه‌ی ما است.
 ۵. به اندازه‌ی بایت هایی که درخواست کردیم داده می‌خوانیم.
- الگوریتم نوشتن به صورت زیر است:

۱. شروع ارتباط با دستگاهی که آدرس I2C آن آدرس حافظه‌ی ما است.
۲. نوشتن یک بایت آدرس حافظه‌ای که می‌خواهیم در آن بنویسیم.
۳. نوشتن بایت هایی که می‌خواهیم بنویسیم.
۴. پایان ارتباط.

۴-۴-۲ شرح آزمایش

یک چیپ حافظه به شما داده می‌شود که از قبل روی آن یک متن ذخیره شده. شما باید ۱۶ کاراکتر از این متن را بخوانید و روی صفحه‌نمایش نمایش دهید. همچنین دو کلید داشته باشید که با فشردن این دو کلید دستگاه شما ۱۶ کاراکتر بعدی یا قبلی متن را بخواند و نمایش دهد. توجه کنید نیازی به نوشتن چیزی بر روی حافظه نیست و این کار از قبل برای شما انجام شده است.

۵-۲ کار با سنسور LM35 و استفاده از آن در ساخت فن هوشمند بر

اساس دما

۱-۵-۲ اهداف آزمایش

- آشنایی با سنسور دمای LM35
- آشنایی با نحوه‌ی کار با ترانزیستور برای سویچینگ و استفاده از منبع تغذیه خارجی

۲-۵-۲ قطعات مورد نیاز

- برد Arduino Uno
- سنسور LM35
- موتور DC
- ترانزیستور BC140
- دیود 1N4001
- منبع تغذیه
- مقاومت ۱ کیلو اهم

۳-۵-۲ مقدمه

سنسور LM35

LM35 یک سنسور دمای ولتاژ پایین به درجه سانتی گراد است. خروجی این سنسور یک ولتاژ است که به طور خطی با دما به درجه سانتی گراد متناسب است و کار با آن بسیار آسان است. این سنسور نیازی به کالیبراسیون ندارد و دقت آن ۱ درجه سانتی گراد در محدوده دمایی -۵۵ تا +۱۵۵ است. این سنسور می‌تواند با منبع تغذیه ۴ تا ۳۰ ولت تغذیه شود و مصرف جریانی کمتر از ۶۰ میکروآمپر دارد.

LM35 در سه شکل مختلف عرضه می‌شود اما رایج ترین نوع آن بسته ۳ پین است که شکلی مشابه یک ترانزیستور دارد.

سوال: در رابطه با پایه های سنسور LM35 تحقیق کرده و وظیفه و محدوده ولتاژ ورودی و خروجی قابل قبول هر پایه را بنویسید. برای این کار می‌توانید به دیتاشیت این سنسور مراجعه کنید. محدودی

های ولتاژی و جریانی معمولاً در قسمتی از دیتاشیت به اسم Electrical Characteristics قرار دارند.

پایه خروجی LM35 به یکی از ورودی های آنالوگ آردوینو متصل می شود. مقدار این ورودی را می تواند با تابع analogRead خواند.

سوال: در رابطه با تابع analogRead تحقیق کنید و در مورد ورودی و خروجی این تابع توضیح دهید. خروجی این تابع چیست و در چه محدوده ای قرار دارد؟

برای تبدیل خروجی تابع analogRead به مقدار دمایی که سنسور به ما می دهد باید ابتدا این خروجی را به مقدار بین ۰ تا ۵ ولت تبدیل کرده و سپس آن را ضرب در ۱۰۰ کنید.

سوال: شبه کد تبدیل خروجی analogRead به دما را بنویسید.

ترانزیستور

آردوینو تنها می تواند ۴۰ میلی آمپر در ولتاژ ۵ ولت را روی پین های دیجیتال خود ارائه دهد. اکثر موتورها برای کار کردن به جریان ویا ولتاژ بیشتری نیاز دارند. یک ترانزیستور می تواند به عنوان یک سوئیچ دیجیتال عمل کند و آردوینو را قادر می سازد تا بارهای با نیازهای الکتریکی بالاتر را کنترل کند. به این صورت که ما از پین خروجی آردوینو صرفاً برای خاموش و روشن کردن کلید استفاده می کنیم و جریان تغذیه اصلی را از یک منبع تغذیه خارجی می گیریم.

ترانزیستور ها دارای سه پایه هستند. در ترانزیستور های پیوندی دوقطبی یا Bipolar Junction Transistor این پایه ها بیس، امیتر و کلکتور نام دارند. نحوه کار آنها به زبان ساده به این صورت است که عبور جریانی کوچک در پایه بیس باعث این می شود که جریانی بسیار بزرگ تر از پایه کلکتور به امیتر برود. این ترانزیستور ها در دو نوع npn و pnp وجود دارند که اشاره به ساختار داخلی آن دارند.

توجه کنید که به دلیل ساختار داخلی ترانزیستور های دوقطبی جهت جریان عبوری بین پایه های امیتر و کلکتور مهم است. به اسم پایه ها توجه کنید، پایه امیتر همانطور که از اسمش پیداست وظیفه تزریق حامل بار و پایه کلکتور وظیفه جمع آوری آنها را دارد. در ترانزیستور های npn حامل بار الکترون است پس جهت الکترون ها از امیتر به کلکتور و جهت جریان از کلکتور به امیتر است. در ترانزیستور های pnp اما برعکس این است و جریان از امیتر به کلکتور است. در قرار دادن ترانزیستور در مدار حتماً به این نکته توجه کنید.

سوال: با توجه به نکته ای که گفته شد، در هر کدام از ترانزیستور های npn و pnp مشخص کنید که کدام یک از پایه های امیتر و کلکتور به موتور و منبع تغذیه متصل می شوند.

موتور DC

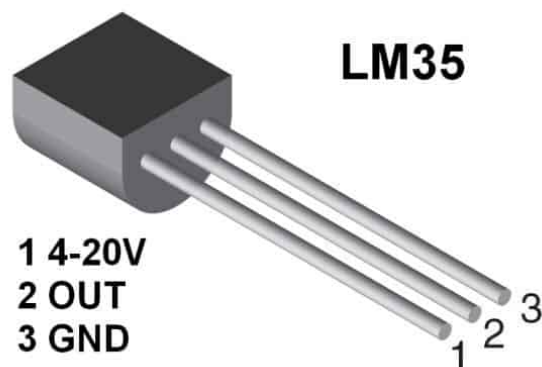
موتورهای DC از طریق القای مغناطیسی کار می کنند. هنگامی که جریان وارد سیم یک میدان مغناطیسی ایجاد می شود. با افزایش جریان میدان مغناطیسی قوی تر شده و ارتباط این میدان مغناطیسی با آهن ربا های داخل موتور سبب چرخیده شدن شفت مرکزی موتور می شود. برعکس این عملیات نیز صادق است، یعنی با چرخاندن موتور می توانیم تولید جریان کنیم. به همین دلیل باید در مدار یک دیود نیز قرار دهیم تا در صورتی که موتور ما به هر دلیلی تولید جریان کرد از عبور این جریان جلوگیری شده تا

باعث آسیب رساندن به قطعات نشود.

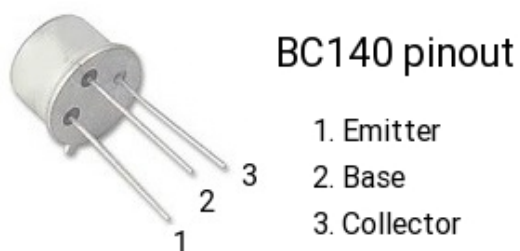
۴-۵-۲ شرح آزمایش

۱. در هیچکدام از مراحل گفته شده قبل از دو مرحله آخر آردوینو متصل به برق و روشن نباشد.
۲. منبع تغذیه را روشن کنید و مقدار خروجی آن را روی ۵ ولت تنظیم کنید.
۳. سنسور را به آردوینو متصل کنید. برای این کار به شکل پایه‌های سنسور نگاه کنید. پین‌های 4-20V و GND به ترتیب به مثبت و منفی منبع تغذیه متصل می‌شود و پایه OUT نیز به یکی از پین‌های آنالوگ آردوینو. پین‌های آنالوگ آردوینو آن‌هایی هستند که شماره آنها با حرف A شروع می‌شود.
۴. ممکن است به دلایل مختلف سیگنال خروجی سنسور مشکل دار باشد. خروجی سنسور را به اسیلوسکوپ وصل کنید. با استفاده از کنترل مربوط به DIV/VOLT اندازه‌ی موج نمایش داده شده را تغییر دهید تا به حد مناسبی برسد.
۵. اگر شکل موج ثابت بود خروجی شما درست است، اما اگر موج دندانه‌اره‌ای داشتید خروجی مشکل دارد. سعی کنید با جابجا کردن سنسور در بردبورد این مشکل را رفع کنید.
۶. ولتاژ خروجی سنسور را از اسیلوسکوپ را مشاهده کنید و آن را ضرب در ۱۰۰ کنید. عددی که دارید چه چیزی را نشان می‌دهد؟
۷. مدار مربوط به موتور را ببندید. ابتدا برای آسیب نرسیدن به برد بر اثر جریان‌های بازگشتی یک دیود در جهت منفی به مثبت موتور قرار دهید. توجه کنید که جهت مثبت و منفی موتور را خودتان تعیین می‌کنید و با تعویض آن صرفاً جهت چرخیدن موتور تغییر می‌کند.
۸. ترانزیستوری که در آزمایشگاه استفاده می‌کنید یک ترانزیستور NPN است. این یعنی باید پایه کلکتور به مثبت منبع تغذیه وصل شود.
۹. پایه‌ی امیتر را به مثبت موتور وصل کنید و منفی موتور را به زمین وصل کنید.
۱۰. پایه بیس را ابتدا به مقاومت ۱ کیلو اهمی و سپس به یکی از پین‌های آردوینو وصل کنید.
۱۱. برنامه شما باید به گونه‌ای باشد که هر ۵۰۰ میلی‌ثانیه اطلاعات را از سنسور بگیرد و اگر دما از حد آستانه‌ی ۳۰ درجه بیشتر بود، موتور را روشن کند.
۱۲. برای برنامه ریزی آردوینو، ابتدا تمام اتصالات به پین‌های آردوینو را جدا کرده و سپس با استفاده از USB آن را پروگرام کنید.

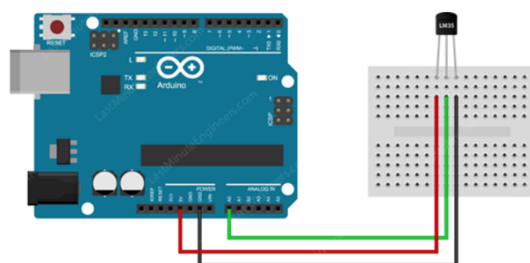
۱۳. برای اجرای برنامه تان، بعد از قطع کردن اتصال USB، مثبت منبع تغذیه را به پین VIN و منفی منبع تغذیه را به پین GND آردوینو متصل کنید.



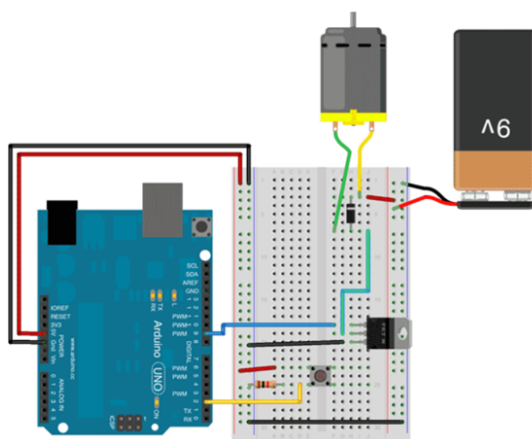
شکل ۲-۱۳: پین های سنسور



شکل ۲-۱۴: پین های ترانزیستور. زائده ای که روی ترانزیستور قرار دارد جهت پایه امیتر را نشان می دهد.



شکل ۲-۱۵: مدار اتصال سنسور به تنهایی



شکل ۲-۱۶: مدار اتصال موتور. توجه داشته باشید که به جای باتری از منبع تغذیه استفاده می کنیم و مقاومت پایه بیس در شکل نیامده. به هیچ عنوان بدون مقاومت متصل به پایه بیس ترانزیستور مدار را روشن نکنید.

۶-۲ سروو موتور

۱-۶-۲ اهداف آزمایش

- آشنایی با سروو موتور
- آشنایی با PWM
- آشنایی با مقاومت های حساس به نور

۲-۶-۲ قطعات مورد نیاز

- Arduino Uno
- سروو موتور
- مقاومت حساس به نور (فوتوسل)
- مقاومت ۱۰ کیلو اهم

۳-۶-۲ مقدمه

سروو موتور

سروو موتور ها یک نوع از موتور های الکتریکی هستند که به جای چرخیدن، خود را در یک زاویه خاص نگه می دارند. این نوع موتور ها در رباتیک کاربرد بسیار زیادی دارند. برای مثال در ساخت دست مصنوعی، ربات های مونتاژ و هر جایی که نیاز داریم چیزی را در یک زاویه خاص قرار دهیم می توانیم از سروو موتور استفاده کنیم.

سروو موتور ها ۳ ورودی دارند، یکی برای VCC که معمولا یک سیم قرمز رنگ است، یکی برای GND که معمولا قهوه ای یا سیاه رنگ است و سیم سوم که ورودی کنترلی را می گیرد. ورودی کنترلی سروو موتور به صورت یک سیگنال PWM است و با هر چه Duty Cycle بیشتر باشد سروو موتور در زاویه بیشتری قرار می گیرد.

سوال: دیتاشیت سروو موتور SG90 را مطالعه کنید و به سوالات زیر پاسخ دهید:

- دوره تناوب سیگنال PWM چند است؟
- دامنه حرکتی این موتور ۱۸۰ درجه است. برای حالتی که سروو موتور وسط، سمت چپ و یا سمت راست است Duty Cycle را محاسبه کنید.

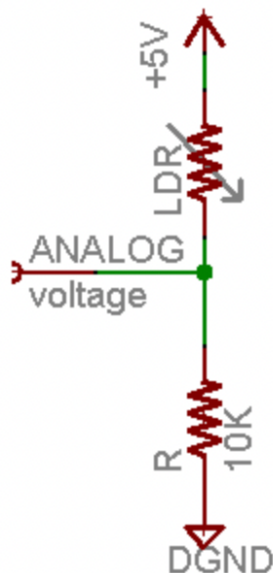
سوال: برای کار با سروو موتور از کتابخانه‌ی Servo.h استفاده می‌کنیم. در مستندات این کتابخانه تحقیق کنید و در مورد ورودی، خروجی و کارکرد توابع زیر توضیح دهید.

• attach()

• write()

مقاومت حساس به نور

مقاومت های حساس به نور یا فوتورزیستور یا فوتوسل نوعی از مقاومت متغیر است که مقدار مقاومت آن نسبت به نوری که به آن تابیده می‌شود تغییر می‌کند. مقاومت یک فوتوسل در تاریکی زیاد و در روشنایی کم می‌شود. برای اتصال آن به آردوینو می‌توانیم از یک مقاومت Pull Down استفاده کنیم. **سوال:** مدار زیر را در نظر بگیرید. اگر مقاومت فوتوسل ۱۰۰ اهم باشد ولتاژ خروجی چند است؟ اگر ۱۰۰ کیلو اهم باشد چه؟



شکل ۲-۱۷: مدار اتصال فوتوسل به آردوینو

۴-۶-۲ شرح آزمایش

۱. فوتوسل را مانند مداری که در مقدمه بررسی کردید به آردوینو متصل کنید. مشخصا باید Analog Voltage را به یکی از پین های ورودی آنالوگ متصل کنید.

۲. برنامه ای بنویسید که با آن بفهمید خروجی analogRead هنگامی که بر روی فوتوسل نور می تابد و هنگامی که دستتان را روی آن گرفتید چند است.

۳. سروو را به آردوینو وصل کنید. سیم های VCC و GND را به ولتاژ ۵ ولت و زمین وصل کنید و سیم کنترلی را به یکی از پین های PWM آردوینو. برای این کار می توانید از پین های ۹ یا ۱۰ استفاده کنید.

۴. برنامه ای بنویسید که ولتاژ فوتوسل را بخواند، اگر مقدار آن ۱۰ درصد کمتر از ولتاژ حالت تاریک باشد زاویه سروو موتور را صفر درجه و اگر مقدار آن ۱۰ درصد بیشتر از ولتاژ حالت روشن باشد زاویه سروو موتور را ۱۸۰ درجه کند. برای این کار می توانید از تابع map استفاده کنید.