

به نام خدا

10/7/2021

گزارش آزمایش شماره 1

آزمایشگاه ریزپردازنده و زبان اسمبلی

محمد جواد زندیه , ابوالفضل بکیاسای کیوی
دانشگاه صنعتی امیرکبیر دانشکده مهندسی کامپیوتر

• تفاوت پردازنده (Processor) و ریزپردازنده (Micro Processor) :

ریزپردازنده ها مدارهایی هستند که واحد پردازنده مرکزی (cpu) را احاطه کرده و شامل پردازنده های دیگری از جمله واحد پردازنده گرافیکی (gpu) ، کارت های صدا (sound card) و کارت های شبکه (network cards) می شوند. بنابراین پردازنده بخشی از ریزپردازنده است.

برخی تفاوت ها :

1. ریزپردازنده علاوه بر انجام انواع توابع منطقی و ریاضی که پردازنده یا cpu قادر به انجام آن است با BIOS و مدارهای حافظه نیز در تعامل است.
2. عملکرد ریزپردازنده گسترده تر از پردازنده است.
3. به دلیل افزودن پردازنده های مختلف بر روی ریزپردازنده ، سرعت آن کندتر از پردازنده است.
4. CPU بخش اصلی کامپیوتر است در حالی که ریزپردازنده یک تراشه ساده در motherboard است.

• انواع ریزپردازنده ها :

سه نوع ریزپردازنده داریم

1. CISC (Complex Instruction Set Computer)

دستورالعمل ها به شکل پیچیده ای هستند. این بدان معناست که یک دستورالعمل واحد می تواند شامل بسیاری از دستورالعمل های سطح پایین باشد. به عنوان مثال بارگذاری داده ها از حافظه ، ذخیره داده ها در حافظه ، انجام عملیات اساسی و غیره. از آنجا که عملیات های زیادی در یک دستورالعمل واحد وجود دارد ، آنها از ثبات های بسیار کمی استفاده می کنند.

2. RISC (Reduced Instruction Set Computer)

دستورالعمل ها بسیار ساده هستند ، بنابراین سریع اجرا می شوند. علاوه بر این، دستورالعمل ها در یک چرخه ساعت کامل می شوند و فقط از چند حالت آدرس دهی استفاده می کنند. همچنین از چندین ثبات استفاده می کند تا تعامل با حافظه کمتر شود

3. EPIC (Explicitly Parallel Instruction Computing)

اجازه می دهد تا دستورالعمل ها به طور موازی با استفاده از کامپایلرها محاسبه شوند و دستورالعمل های پیچیده در فرکانس های ساعت کمتری نیز پردازش می شوند.

• معرفی برد های Arduino :

1. Uno: آردوینو Uno یک برد میکروکنترلر متن باز و مبتنی بر میکروکنترلر Microchip ATmega328P است که توسط Arduino.cc ساخته شده است.
فرکانس: 16Mhz ATmega328
ولتاژ: 5V
ورودی/خروجی دیجیتال: 14
ورودی/خروجی آنالوگ: 6 input, 0 output
2. Nano: برد آردوینو نانو فاقد جک DC است و به جای کابل استاندارد با کابل USB Mini-B کار میکند. این برد برای پروژه هایی که دارای محدودیت فضا هستند بسیار مناسب است.
فرکانس: 20Mhz ATmega4809
ولتاژ: 5V
ورودی/خروجی دیجیتال: 14
ورودی/خروجی آنالوگ: 8 input, 0 output
3. Mega: برد Mega 2560 با اکثر شیلد های طراحی شده برای Uno و برد های قبلی Duemilanove یا Diecimila سازگار است. از برد آردوینو مگا میتوان برای پروژه هایی که نیازمند پین های دیجیتال فراوان هستند استفاده کرد.
فرکانس: 16Mhz ATmega2560
ولتاژ: 5V
ورودی/خروجی دیجیتال: 54
ورودی/خروجی آنالوگ: 16 input, 0 output
4. Due: برد آردوینو دیو را میتوان قویترین برد آردوینو دانست. Arduino Due با استفاده از نرم افزار Arduino IDE برنامه ریزی میشود.
فرکانس: 84Mhz AT91SAM3X8E
ولتاژ: 3.3V
ورودی/خروجی دیجیتال: 54

ورودی/خروجی آنالوگ: 12 input, 2 output

5. Micro: دارای ارتباط USB داخلی است و نیاز به پردازنده ثانویه را از بین می برد. این ویژگی اجازه می دهد تا به عنوان موس یا کیبورد به کامپیوتر متصل شود.

فرکانس: 16Mhz ATmega32U4

ولتاژ: 5V

ورودی/خروجی دیجیتال: 20

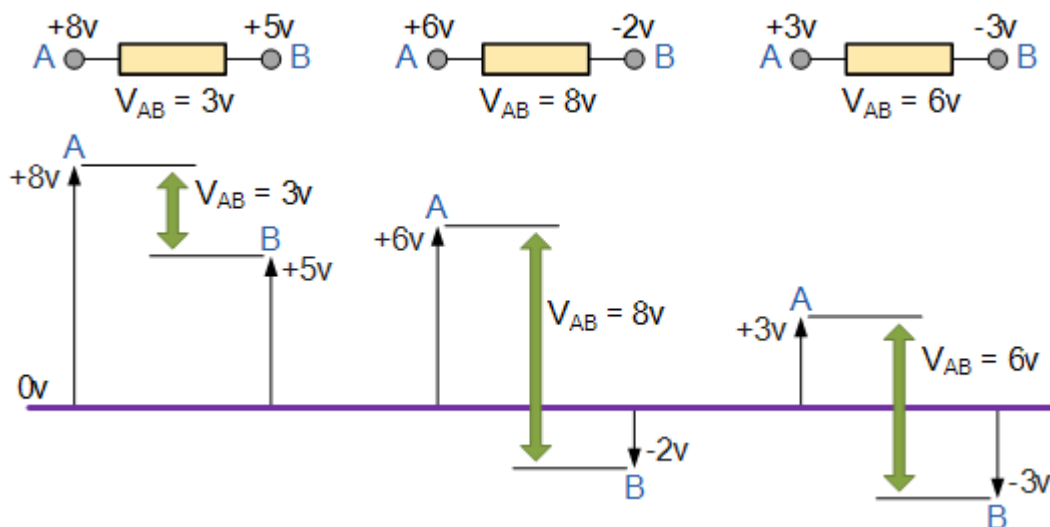
ورودی/خروجی آنالوگ: 12 input, 0 output

• مفاهیم پایه الکترونیک:

1. **اختلاف پتانسیل:** تفاوت ولتاژ بین دو نقطه از یک مدار، «اختلاف پتانسیل» (Potential Difference)

نامیده می شود. عبور جریان، به دلیل اختلاف پتانسیل بین دو نقطه روی می دهد. در یک مدار الکتریکی بسته، جریان به فرم بار الکتریکی حرکت می کند. اما اختلاف پتانسیل مفهومی متفاوت دارد و حرکت نمی کند.

واحد اختلاف پتانسیل ولت نام دارد. $V = RI$



2. **شدت جریان:** جریان الکتریکی به صورت نرخ تغییر بار الکتریکی نسبت به زمان تعریف شده که با نماد I نشان

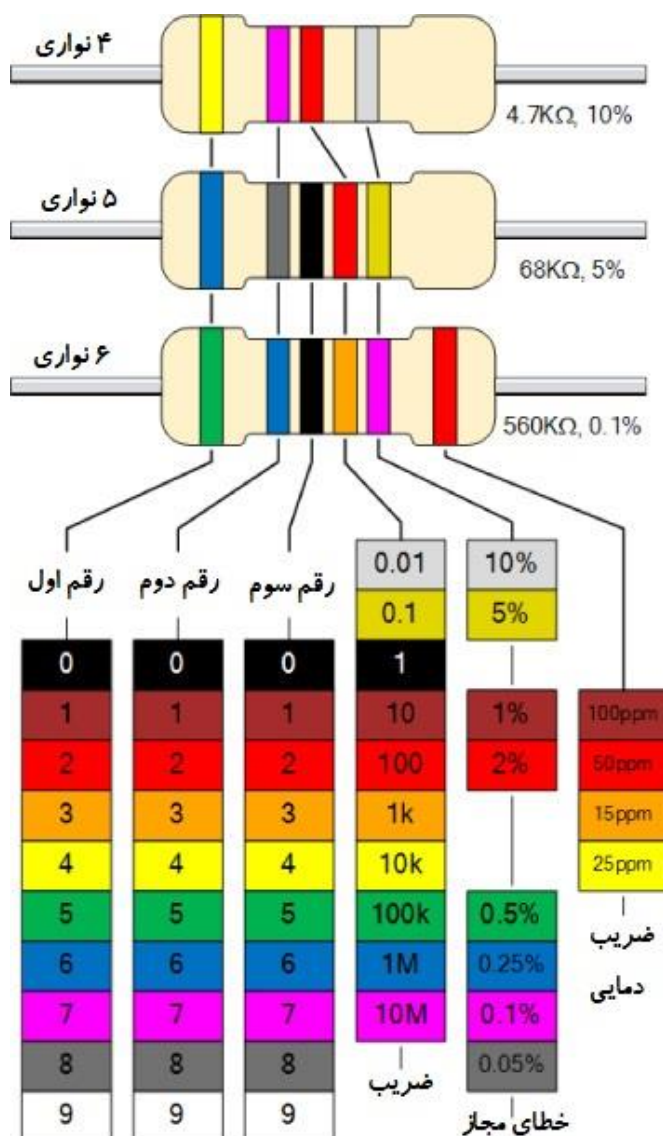
داده می شود و واحد آن آمپر می باشد. $I = \frac{q}{t}$

3. **قانون اهم:** بیان می دارد که نسبت اختلاف پتانسیل (یا افت ولتاژ) بین دو سر یک جسم (مقاومت) به

جریان عبورکننده از آن به شرطی که دما ثابت بماند، مقدار ثابتی است و برابر است با $R = \frac{V}{I}$ که واحد آن

اهم میباشد.

4. نحوه خواندن کد رنگی:



جدول کد رنگی مقاومت

رنگ	عدد مناظر	ضریب	خطای مجاز
مشکی	0	1	
قهوه ای	1	10	± 1%
قرمز	2	100	± 2%
نارنجی	3	1,000	
زرد	4	10,000	
سبز	5	100,000	± 0.5%
آبی	6	1,000,000	± 0.25%
بنفش	7	10,000,000	± 0.1%
خاکستری	8		± 0.05%
سفید	9		
طلایی		0.1	± 5%
نقره ای		0.01	± 10%
خالی			± 20%

1% = قهوه ای، 2% = قرمز، 5% = طلایی، 10% = نقره ای

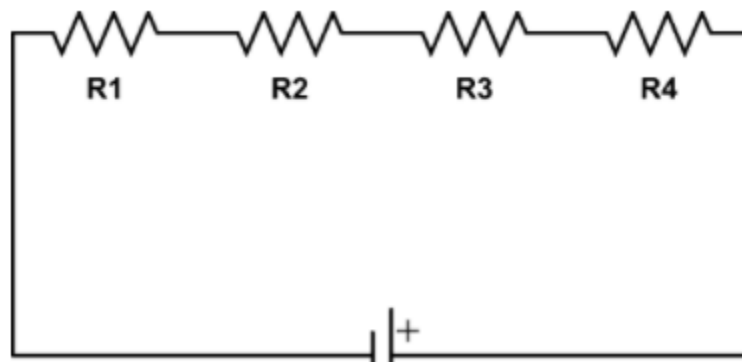
نوارها از چپ به راست خوانده می شوند و نوار اول از نزدیکترین نوار به لبه کناری در طرف چپ مقاومت شروع می شود:

نوار سوم $10 \times$ نوار دوم، نوار اول = ضریب، رقم دوم، رقم اول

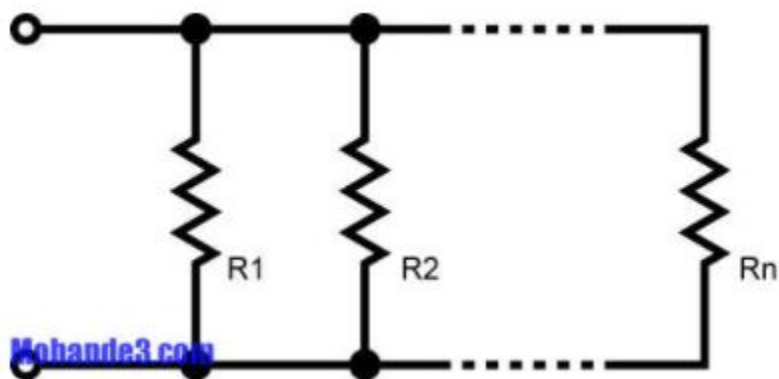
$$4700\Omega = 47 \times 10^2 = 472 \text{ قرمز بنفش زرد}$$

5. نحوه سری و موازی بستن مقاوت ها و فرمول آنها:

اتصال سری: $R_{total} = R_1 + R_2 + R_3 + \dots + R_n$



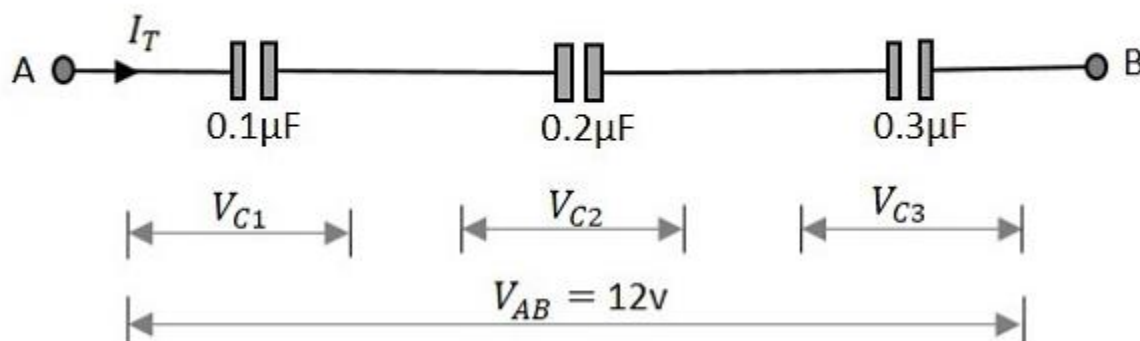
اتصال موازی: $\frac{1}{R_{total}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \dots + \frac{1}{R_n}$



6. نحوه سری و موازی بستن خازن ها و فرمول آنها:

اتصال سری:

Capacitors in Series

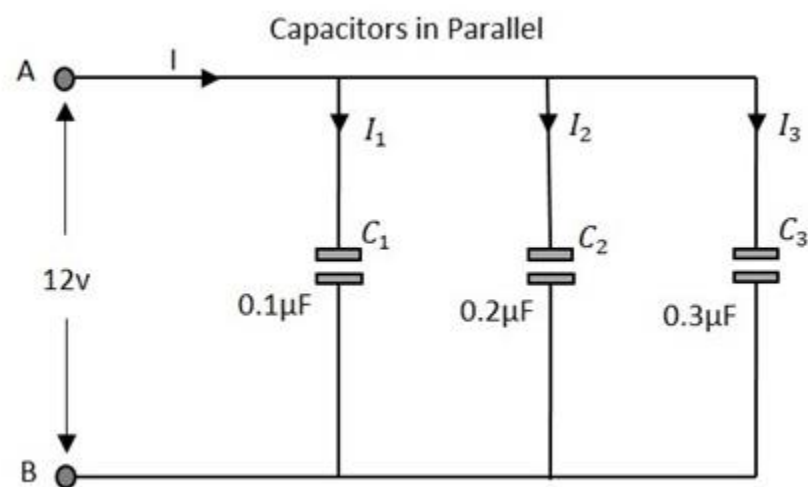


$$\frac{1}{C_T} = \frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3}$$

$$V_{C1} = \frac{Q_T}{C_1} \quad V_{C2} = \frac{Q_T}{C_2} \quad V_{C3} = \frac{Q_T}{C_3}$$

$$V_T = V_{C1} + V_{C2} + V_{C3}$$

اتصال موازی:



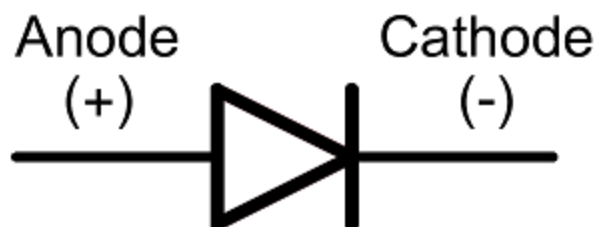
$$C_T = C_1 + C_2 + C_3$$

$$V_T = V_1 = V_2 = V_3$$

$$I_T = I_1 + I_2 + I_3$$

7. توضیح چگونگی عملکرد دیود و دیود نورانی:

دیود ها مدل های متنوعی دارند که ساختار آن براساس اتصالات P-N است. آنها یکی از قطعات نیمه هادی هستند و از دو پایه آند و کاتد تشکیل شده اند که جریان الکتریکی را فقط از یک جهت عبور می دهند (در شکل می توان دید که جریان از آند به کاتد امکان پذیر است اما عکس آن صادق نیست).



بررسی ساختار P-N : ناخالصی نوع n و p در طرفین اتصال قرار دارد و باعث ایجاد حفره ها و الکترون های آزاد شده است. برای ساخته شدن ناحیه P می توان ناخالصی هایی از نوع گالیوم، آلومینیوم و بور به ماده نیمه هادی اضافه کرد. در اثر اضافه شدن این مواد تعداد بسیار زیادی حفره در این ناحیه ایجاد می شود. ناحیه P در یک دیود به عنوان آند شناخته می شود و پلاریته مثبت آن را می سازد. برای ناحیه N از ناخالصی هایی مثل فسفر، آنتیموان و آرسنیک استفاده می شود که باعث ازدیاد الکترون در این ناحیه خواهد شد. به ناحیه N کاتد گفته شده و دارای پلاریته منفی است.

دیود های نوری (LED) معمولاً از نیمه رسانای گالیم-آرسناید ساخته می شوند. حرکت الکترون ها از ناحیه N به سمت حفره های ناحیه P باعث گرفتار شدن آنها در دام اتم های فاقد الکترون آزاد می شود. این الکترون ها چون از لایه آزاد به یک لایه اتم سقوط می کنند مقداری انرژی از دست می دهند و این انرژی به صورت نور مادون قرمز ساطع می شود.

گزارش آزمایش شماره 1 :

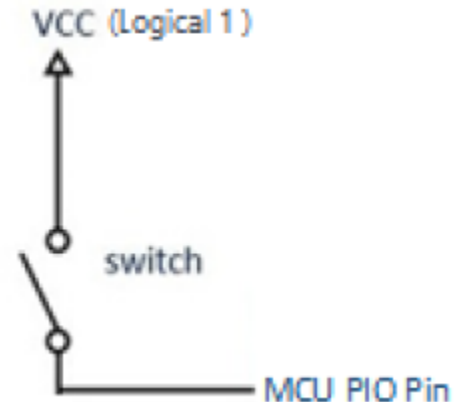
• تفاوت روش های سرکشی (Polling) و وقفه محور (Interrupt Driven):

INTERRUPT DRIVEN	POLLING
دستگاه CPU را متوجه می کند که به آن نیاز دارد	CPU مدام چک می کند که آیا دستگاه به توجه نیاز دارد یا نه
Interrupt یک مکانیزم سخت افزاری است	POLLING یک PROTOCOL است
دستگاه توسط interrupt handler سرویس داده می شود	دستگاه توسط CPU سرویس داده می شود

- چرا روش زیر برای فهمیدن اینکه چه زمانی کلید بسته شده درست نیست؟ در این مدار پایه میکرو در چه حالتی می باشد؟

زیرا در زمانی که switch باز است پایه MCU PIO Pin به حالت float است و مانند آنتی برای نویز عمل می کند.

در درس مدار های منطقی به حالت اصطلاحاً High Impedance هم می گفتند.

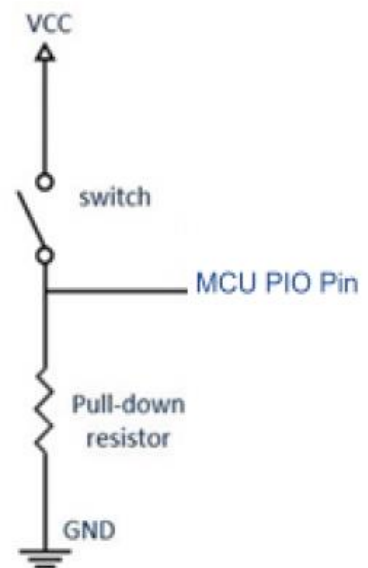
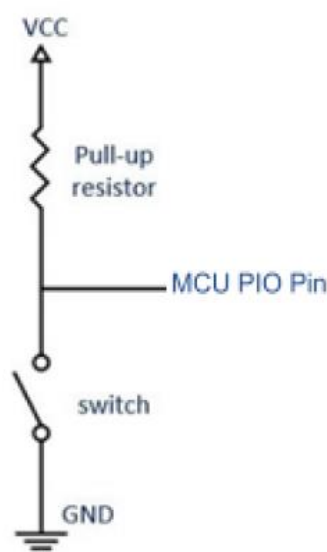


- درباره چگونگی کارکرد مدار های پایین توضیح دهید. به چه دلیل نیاز به مقاومت Pull-up / Pull-Down داریم؟

دقیقا به دلیلی که در سوال قبل توضیح داده شد برای آنکه حالت بی مقداری را نداشته باشیم مدار را توسط یک مقاومت به زمین متصل می کنیم که در حالتی که منطق یک را نداشته باشیم، منطق صفر حاکم باشد.

وقتی switch وصل نیست در مدار چپ منطق یک برقرار است چون به VCC وصل است و در مدار راست منطق صفر چون به زمین وصل است.

وقتی switch متصل می شود این مقدار تغییر می کند و گیت not ما ساخته شد.



- توضیح دستورات به کار رفته در کد Arduino :

ورودی یا خروجی بودن پین خاصی را مشخص می کند: pinMode ()

استیت یک پین را مشخص می کند: digitalRead ()

استیت یک پین را تغییر می دهد: digitalWrite ()

به اندازه مقدار داده شده به آن برحسب میلی ثانیه تاخیر ایجاد می کند: delay ()

برای سرویس دهی به وقفه ها استفاده می شود: attachInterrupt()

- آیا رخ دادن یک اتفاق در صورت اعلام شدن (Assertion) لزوما منجر به اجرای روال سرویس وقفه متناظر با آن می شود؟

خیر ابتدا به وقفه ای که در حالت اجراست رسیدگی می شود و سپس اگر اولویت با این وقفه بود سراغ آن می رویم.

- پایه وقفه در برد ATmega 2560 و شیوه پیاده سازی وقفه ورودی را به دست آورید.

2,3,18,19,20,21 پایه های وقفه در Arduino mega 2560 می باشد.

DIGITAL PINS USABLE FOR INTERRUPTS: 2, 3, 18, 19, 20, 21 (pins 20 & 21 are not available to use for interrupts while they are used for I2C communication)

```
attachInterrupt (digitalPinToInterrupt (pin) , ISR, mode) (recommended)
```

interrupt: the number of the interrupt. Allowed data types: int.

pin: the Arduino pin number.

ISR: the ISR to call when the interrupt occurs; this function must take no parameters and return nothing. This function is sometimes referred to as an interrupt service routine.

mode: defines when the interrupt should be triggered. Four constants are predefined as valid values:

- **LOW** to trigger the interrupt whenever the pin is low,
- **CHANGE** to trigger the interrupt whenever the pin changes value

- **RISING** to trigger when the pin goes from low to high,
- **FALLING** for when the pin goes from high to low.

The Due, Zero and MKR1000 boards allow also:

- **HIGH** to trigger the interrupt whenever the pin is high
- اگر بخواهیم در زمان تغییر مقدار پایه، وقفه فعال شود از چه mode ای درون تابع `attachInterrupt` استفاده می شود؟ **CHANGE**
- انواع اتفاق های ورودی را که واحد GPIO در برد آردوینو ATmega 2560 می تواند رخ دادن آنها را بفهمد و اعلام کند بنویسید.

values:

- **LOW** to trigger the interrupt whenever the pin is low,
- **CHANGE** to trigger the interrupt whenever the pin changes value
- **RISING** to trigger when the pin goes from low to high,
- **FALLING** for when the pin goes from high to low.

The Due, Zero and MKR1000 boards allow also:

- **HIGH** to trigger the interrupt whenever the pin is high

● اگر دکمه را در حالت فشرده برای زمان طولانی نگه داریم چه اتفاقی خواهد افتاد؟ آیا با منطق کارکرد خواسته شده سازگار است؟ چه راه حلی برای این مشکل (در صورت وجود) می توان پیشنهاد کرد؟

در صورت پایان فرایند دوباره فرایند تکرار می شود. می توان به این صورت سرویس دهی کرد که به تغییر سوییچ حساس باشد نه لک لبه خاص مثلا low or high

● فرض کنید می خواهیم برد مورد نظر علاوه بر فراهم کردن کارکرد خواسته شده در بالا، عمل دیگری را نیز به صورت زمان دار انجام دهد. برای نمونه در کنار کارکرد بالا، وضعیت روشن یا خاموش بودن یک LED را نیز هر

5 ثانیه یکبار تغییر دهد. روشی برای افزودن این کارکرد تازه به برنامه پیشنهاد دهید.

می توان در loop از دستور (millis) استفاده کرد و هر پنج ثانیه متوجه شد که باید تغییر اعمال شود ضمن اینکه به صورت موازی با دستورات دیگر اجرا می شود. یعنی نیاز به نگه داشتن برنامه نیست.

● فرض کنید می خواهیم کارکرد دیگری را به دستگاه اضافه کنیم به این صورت که در صورت یک شدن پایه

عملیات مشخصی را به عنوان پاسخ انجام دهد. (محدودیت زمانی برای پاسخ دادن وجود دارد) هیچ یکاز اتفاق های

یک شدن پایه نباید از دست برود (بی پاسخ بماند). و یک شدن پایه نیز در هر زمانی ممکن است رخ دهد. آیا برنامه

شما - که به روش سرکشی واحد های جانبی را بررسی میکند- می تواند در هر شرایطی (مثلا هنگام فشردن کلید)

این کارکرد را فراهم کند؟

خیر ممکن است در زمانی که پایه یک می شود در روش سرکشی ما در حال سرویس دهی به یک سری عملیات دیگر باشیم و چون ممکن است آن عملیات طول بکشد و دستورات هم خط به خط اجرا می شوند پس ممکن است از دست برود.

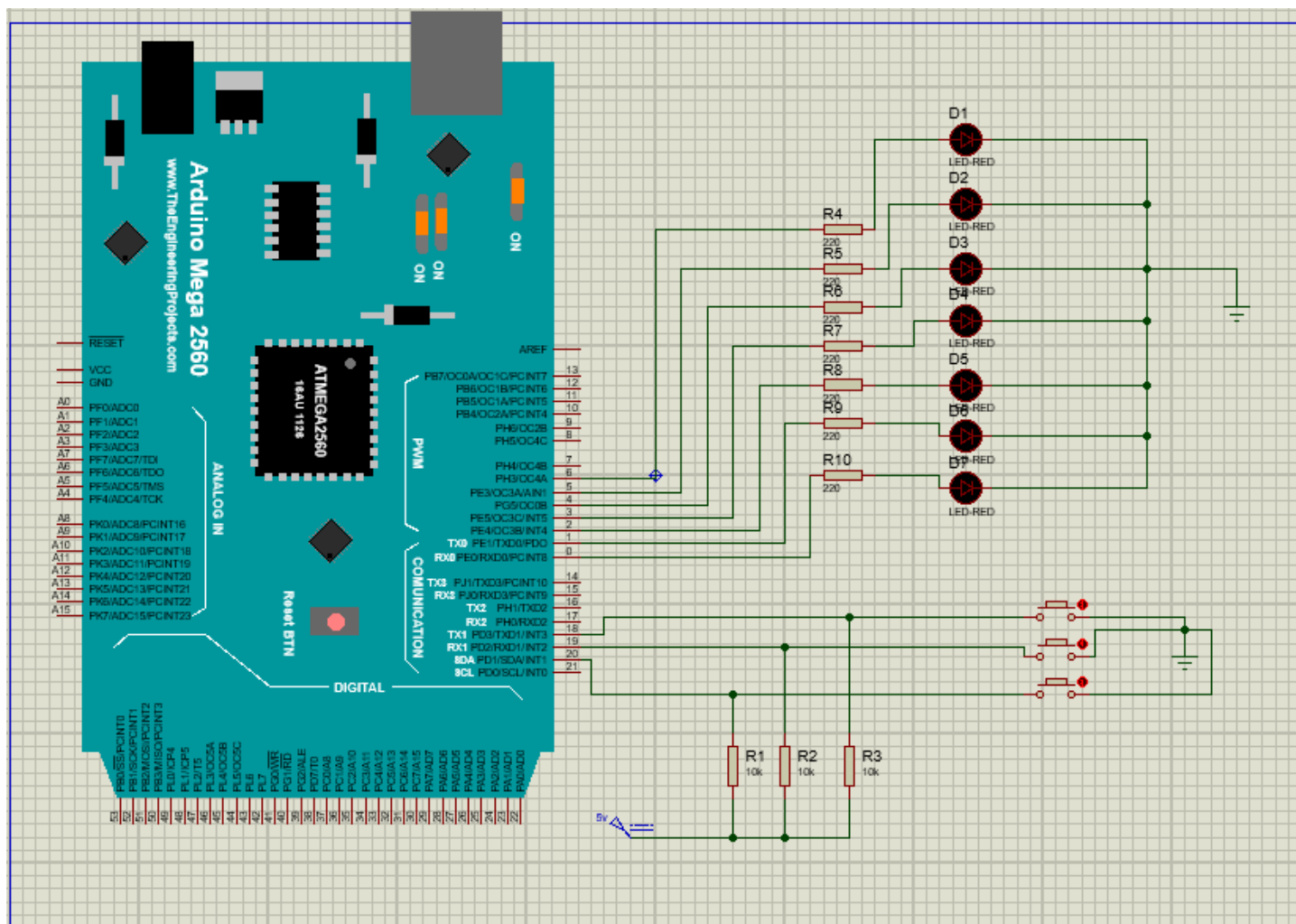
● فرض کنید به دلیل محدودیت در توان مصرفی می خواهیم پردازنده در هنگام بیکاری به خواب برود. در زمان

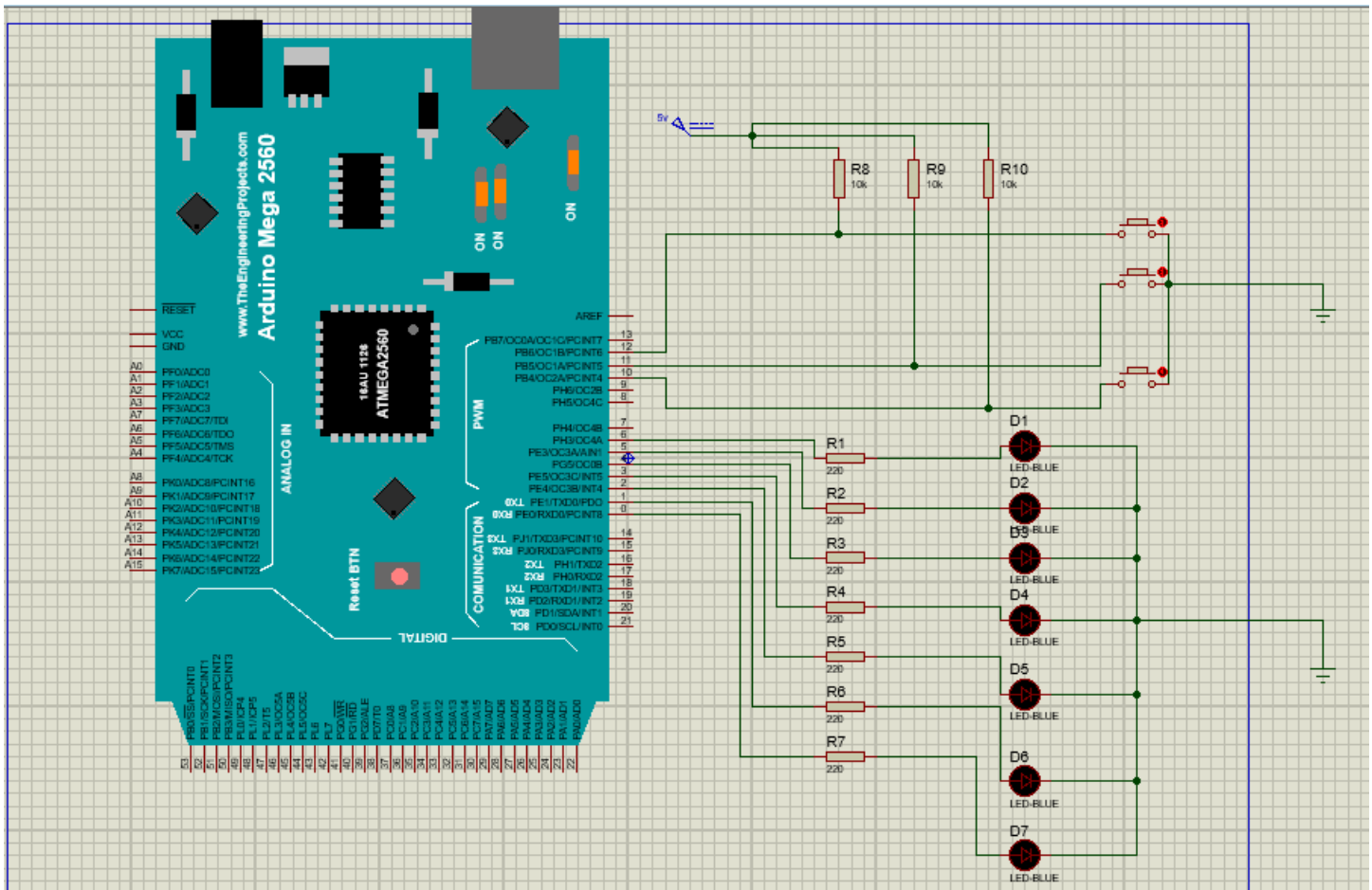
خواب پردازنده هیچ دستوری را اجرا نمی کند. روش سرکشی چه قدر با این نیازمندی سازگاری دارد؟ آیا می توان با

این روش هم به خواب رفت و هم کارکرد درست آزمایش را فراهم کرد؟

خیر نمی شود زیرا پایه روش سرکشی این است که داعما چک کند که آیا باید سرویسی ارایه بدهد یا نه پس نمی تواند به خواب برورد، اما اگر از وقفه استفاده شود می تواند پردازنده را به حالت low power mode برد.

مقایسه سرکشی و وقفه محور در جدول صفحه 7 موجود می باشد.





ویدیو ها اجرای کد ها را نیز مشاهده بفرمایید.