

خواندن از Keypad و نمایش بر روی LCD

هدف: کار با Keypad و LCD

شرح آزمایش:

در این آزمایش می‌خواهیم کلید زده شده بر روی keypad را خوانده و بر روی LCD نمایش دهیم.

پورت‌های ورودی و خروجی در AVR:

پورت‌های ورودی و خروجی در AVR هرکدام دارای سه ثابت هشت بیتی به نام‌های PORT و DDR و PIN می‌باشد. به عنوان مثال سه ثابت مربوط به پورت C عبارتند از: PORTC و DDRC و PINC.

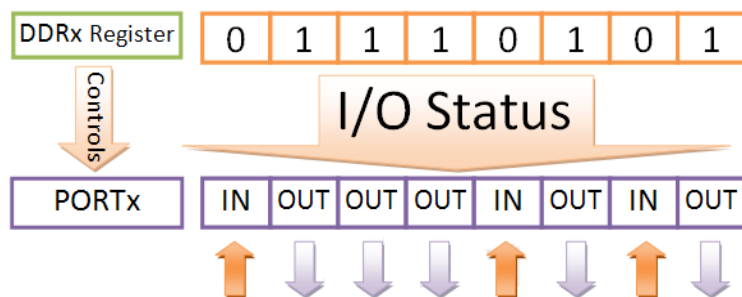
رجیستر $DDRx$:

یک پورت می‌تواند هم نقش ورودی داشته باشد و هم نقش خروجی. این کار با برنامه‌ریزی ثابت DDR (Data Direction Register) انجام می‌شود. برای هر پورت، یک ثابت DDR داریم که هم نام پورت است. به عنوان مثال ثابت DDR مربوط به پورت A به صورت DDRA شناخته می‌شود. هر یک از بیت‌های ثابت $DDRx$ ($x=A,B,C,D$) یکی از پین‌های پورت X را کنترل می‌کند. برای مثال بیت پنجم ثابت، پنجمین پین پورت را کنترل می‌کند.

به منظور تنظیم کردن یک پین به عنوان ورودی، باید بیت مربوط به آن پین در ثابت DDR با مقدار صفر برنامه‌ریزی شود. به طور عکس برای تنظیم کردن یک پین به عنوان خروجی، باید بیت مربوط به آن پین در ثابت DDR با مقدار یک برنامه‌ریزی شود. به عنوان مثال فرض کنید می‌خواهیم پین ۳ و ۷ از پورت B را به عنوان خروجی و بقیه‌ی پین‌ها را به عنوان ورودی تنظیم کنیم، باید بنویسیم:

$DDRB = 0b10001000;$

The DDRx register controls if a pin is in Input mode or Output mode

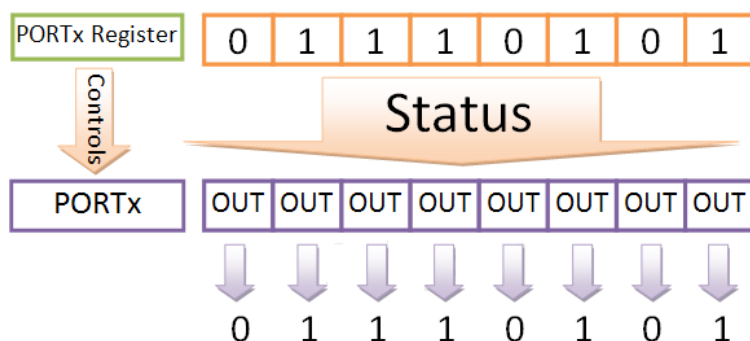


رجیستر $PORTx$:

این رجیستر بر حسب این که پورت به عنوان خروجی و یا ورودی تنظیم شده باشد (توسط تنظیم رجیستر DDRx)، دو وظیفه دارد:

حالت ۱) صفر یا یک کردن پین (در صورتی که پین مورد نظر در نقش خروجی عمل کند):

If all the pins in PORTx are configured as output:



همان طور که در شکل بالا مشاهده می‌شود، در حالتی که پین به عنوان خروجی عمل کند (بیت مربوط به آن در رجیستر DDRx روی یک تنظیم شده باشد) با یک کردن بیت مربوط به آن پین در رجیستر PORTx، مقدار آن پین در خروجی یک (high) خواهد شد. و به طور عکس با صفر کردن بیت مربوط به آن پین در رجیستر PORTx مقدار آن پین در خروجی صفر (low) خواهد شد. به طور مثال در کد زیر بیت‌های صفر، سه و هفت ست (set) و بیت‌های صفر و دو ریست (reset) می‌شوند. دیگر بیت‌ها به عنوان پین‌های ورودی تنظیم شده‌اند:

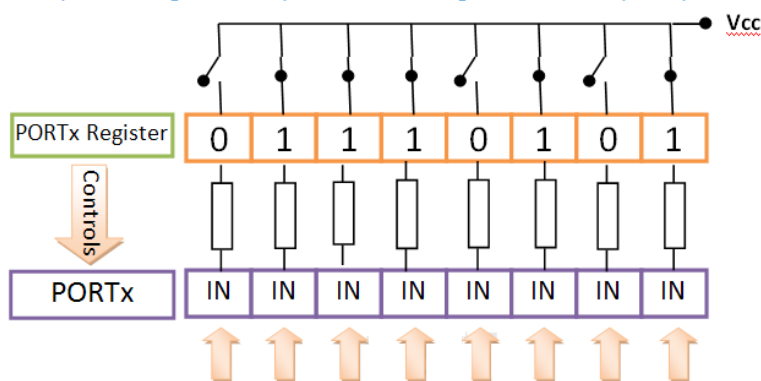
```
DDRB = 0b10001111;
PORTB = 0b10001010;
```

حالت ۲) فعال/غیرفعال کردن مقاومت‌های پول‌آپ داخلی (در صورتی که پین مورد نظر در نقش ورودی عمل کند):

در میکروکنترلر AVR همه‌ی پین‌ها به مقاومت پول‌آپ مجهزند. در حالتی که یک پین در نقش ورودی عمل کند، مقاومت پول‌آپ مربوط به آن را می‌توان با برنامه‌ریزی بیت مربوط به آن در رجیستر PORTx فعال یا غیر فعال کرد. در صورتی که آن را با یک

برنامه‌ریزی کنیم مقاومت پول‌آپ فعال می‌شود و به طور عکس اگر آن را با صفر برنامه‌ریزی کنیم مقاومت پول‌آپ غیر فعال می‌شود. فعال کردن مقاومت‌های پول‌آپ ورودی را به حالت یک می‌برد. با فعال کردن مقاومت‌های پول‌آپ از نویزی شدن ورودی جلوگیری می‌کنیم.

If the pin is configured as input, PORTx manages the internal pull-up



در برنامه‌ی زیر ، پورت B به عنوان ورودی برنامه‌ریزی شده و سپس مقاومت‌های پول‌آپ پین‌های ۱، ۳، ۵، ۶ و ۷ آن فعال شده‌اند:

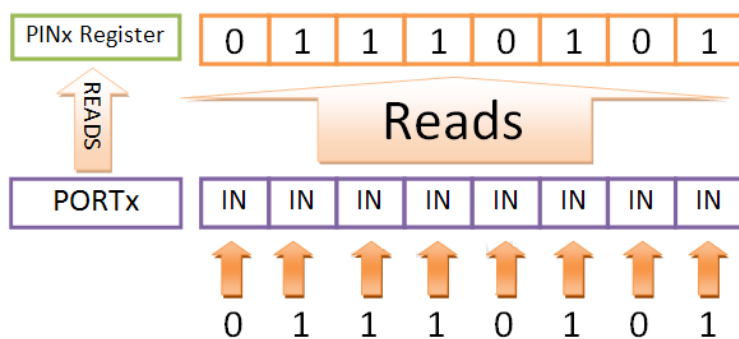
```
DDRB = 0b00000000;
PORTB = 0b01110101;
```

رجیستر $PINx$:

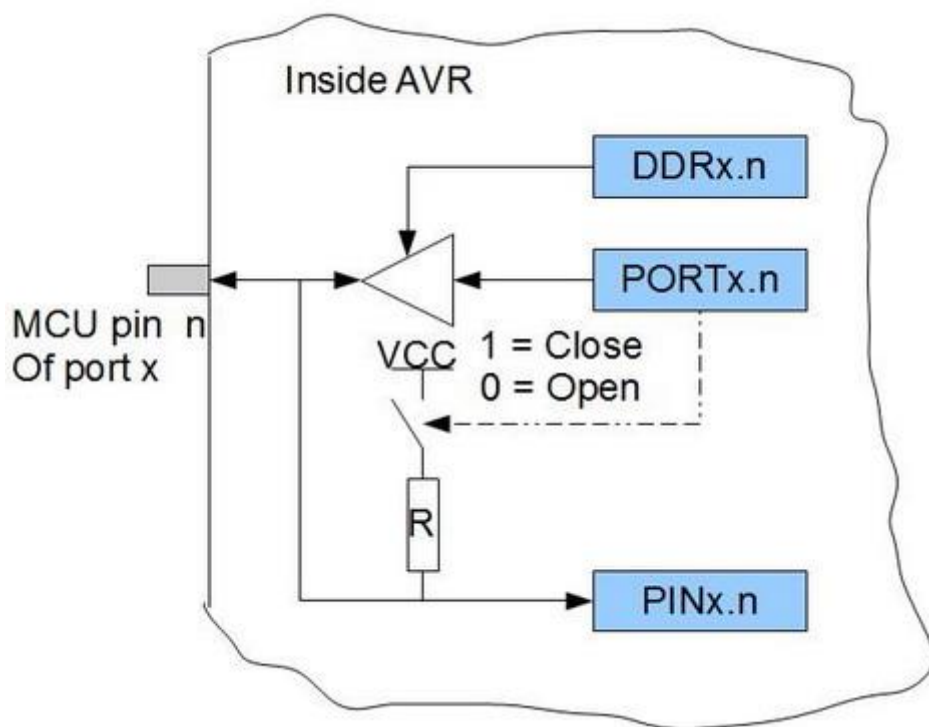
رجیستر $PINx$ شامل وضعیت تمام پین‌های مربوط به آن پورت است. اگر پین مربوطه به عنوان ورودی تنظیم شده باشد، بیت مربوط به آن پین در رجیستر $PINx$ سطح منطقی آن پین را نشان می‌دهد. و اگر پین مربوطه به عنوان خروجی تنظیم شده باشد، بیت مربوط به آن پین در رجیستر $PINx$ شامل آخرین داده‌ی خروجی بر روی آن پین می‌باشد. برنامه‌های زیر وضعیت پورت D را خوانده و ذخیره می‌کنند:

```
DDRD = 0b00000000;
unsigned char status;
status = PIND;
```

$PINx$ register reads the value of input pins



شکل زیر نشان‌دهنده‌ی ارتباط بین سه رجیستر معرفی شده در بالا می‌باشد. لازم به ذکر است که این شکل یک شماتیک ساده می‌باشد و مدار داخلی پورت‌های I/O بسیار مفصل‌تر است.

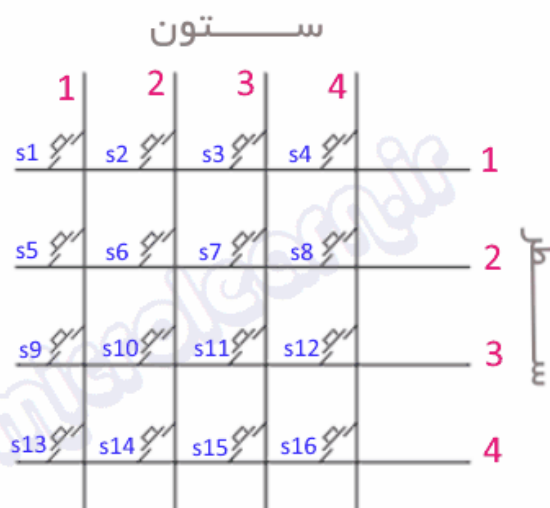


$\left\{ \begin{array}{l} \text{output} \\ \text{input} \end{array} \right.$	$DDRx.n = 1$	$output\ value = PORTx.n$	$pullup\ enable$	$PORTx.n = 1$
	$DDRx.n = 0$	$input\ value = PINx.n$	$pullup\ disable$	$PORTx.n = 0$

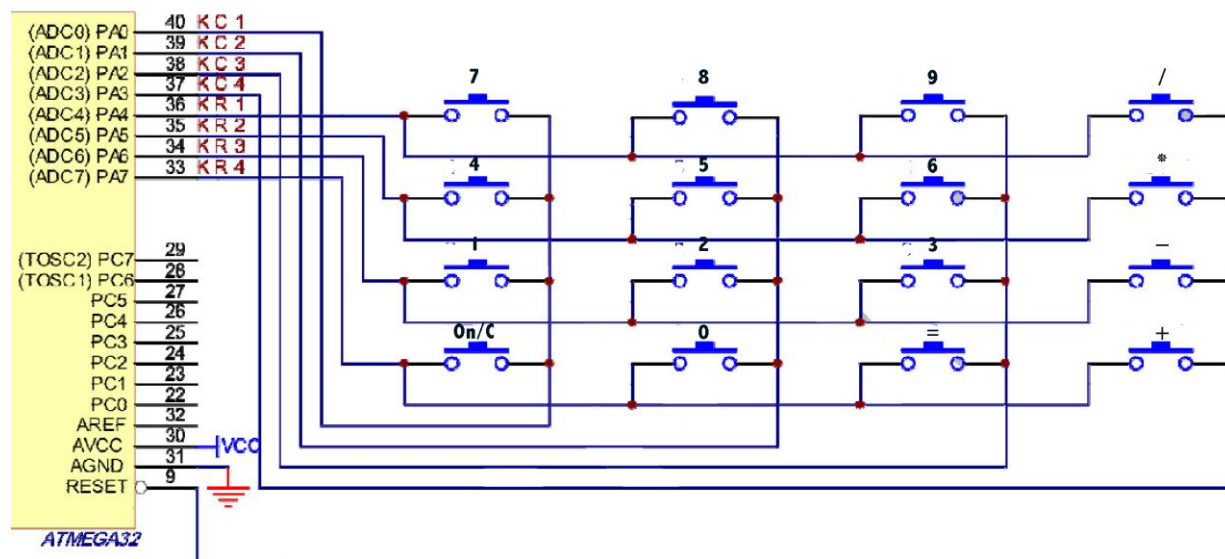
$x = A, B, C, D \quad n = 0, \dots, 7$

صفحه کلید ماتریسی :

ما می‌توانیم با استفاده از یک پورت هشت بیتی یک ماتریس 4×4 از کلیدها را به میکرو متصل کنیم. وقتی یک کلید فشار داده می‌شود یک سطر و ستون به هم متصل می‌شوند. در غیر این صورت هیچ گونه اتصالی بین سطرها و ستون‌ها وجود ندارد. همان طور که در شکل زیر مشاهده می‌کنید، ۴ سطر و ۴ ستون وجود دارد که هیچ گونه اتصالی به هم دیگر ندارند. برای مثال اگر کلید s3 فشرده شود. سطر ۱ و ستون ۳ به هم دیگر وصل می‌شوند.



شکل زیر نحوه اتصال یک کیپد 4×4 به میکرو را نشان می‌دهد:



همانطور که در شکل بالا ملاحظه می‌کنید از پورت A برای پیمایش کلیدها استفاده شده است. ستون‌ها را به ۴ پین کم ارزش پورت A وصل کرده‌ایم. هم چنین سطرها به ۴ پین پر ارزش پورت A وصل شده‌اند. مثلاً ستون صفر به PINA.0 و سطر سه به PINA.7 وصل شده‌اند.

پایه‌هایی که به عنوان ورودی تنظیم شده‌اند باید دارای مقاوت بالاکش باشند چرا که در زمان فشردن نشدن کلید هیچ سطح منطقی رو پورت وجود ندارد و این امری نامطلوب است. (برای خواندن وضعیت یک پورت باید همیشه یک سطح منطقی معین روی آن باشد). برای فعال کردن مقاومت بالاکش پایه‌های ورودی باید در رجیستر PORTA بیت معادل آن پایه‌ها را یک کرد.

برای مشخص شدن کلید فشردن شده باید از روشی استاندارد استفاده کرد که به روش اسکن معروف است. یعنی میکروکنترلر به صورت پیوسته صفحه کلید را اسکن می‌کند تا کلید فشردن شده را تشخیص دهد.

مراحل اسکن 4*4 keypad:

۱. در میکروکنترلر، سطرهای کیپد (DDRA.4-7) را خروجی و ستون‌های کیپد (DDRA.0-3) را ورودی تنظیم کنید.
۲. به سطر اول (PORTA.4) مقدار صفر بدهید و هم زمان پول‌آپ ستون‌ها (PORTA.0-3) را فعال کنید.
۳. مقدار ورودی ستون‌ها (PINA.0-3) را بخوانید. اگر در یکی از آنها صفر ظاهر شود، به این معنا است که یکی از چهار کلید سطر اول زده شده است. اینکه کدام کلید از سطر اول زده شده است را می‌توان با نگاه کردن به ستون آن تشخیص داد. مثلاً اگر PINA.0 برابر با صفر باشد یعنی کلید ۷ زده شده است. یا اگر مثلاً PINA.3 صفر باشد، یعنی کلید تقسیم زده شده است.
۴. حال به سطر دوم (PORTA.5) مقدار صفر بدهید و هم زمان پول‌آپ ستون‌ها (PORTA.0-3) را فعال کنید.
۵. مانند گام ۳، مقدار ورودی ستون‌ها (PINA.0-3) را بخوانید. اگر در یکی صفر ظاهر شد، با نگاه کردن به ستون صفر شده می‌توان فهمید کدامیک از کلیدهای سطر دوم یعنی ۴، ۵، ۶ یا ضرب زده شده اسن.
۶. عملیات گام ۴ و ۵ را برای سطر سوم تکرار کنید.
۷. عملیات گام ۴ و ۵ را برای سطر چهارم تکرار کنید.
۸. در صورتی که هیچ کلیدی زده نشده عدد ۲۵۵ را برگردانید.

نکته: برای کلیدهای غیر عددی باید یک عدد متناظر در نظر گرفت: تقسیم (۱۰)، ضرب (۱۱)، منها (۱۲)، on/c (۱۳)، مساوی (۱۴) و جمع (۱۵).

یک تابع به نام `getkey` بنویسید که کیپد را اسکن کند، اگر کلیدی زده شده اندیس آن را برگرداند، اگر کلیدی زده نشده است، عدد ۲۵۵ را برگرداند.

تنظیمات LCD:

RS PORTC.1
RD PORTC.2
E PORTC.3
D4 PORTC.4
D5 PORTC.5
D6 PORTC.6
D7 PORTC.7

برای نوشتن بر روی LCD می‌توانید از دستورات زیر استفاده کنید:

lcd_clear();	پاک کردن
lcd_puts("Good Day");	نوشتن یک رشته
lcd_putchar('A');	نوشتن یک کاراکتر
lcd_gotoxy(col, row);	تغییر سطر و ستون

مثال: یک عدد سه رقمی مثل 153 را بر روی LCD نمایش دهید.

#include <stdlib.h>	
char num = 153;	متغیر عددی
char numstr[10];	متغیر از جنس آرایه کاراکتری (رشته)
itoa(num, numstr); // integer to ascii	تبدیل عدد به چندین کد اسکی و قرار دادن آن در آرایه کاراکتری
lcd_puts(numstr);	نمایش آرایه کاراکتری

تنظیمات برد:

Chip : Atmega32

Clock : 11.059200 MHz

گزارش کار:

گزارش کار شما باید شامل موارد زیر باشد:

- اسامی اعضای گروه
- کد برنامه‌ی نوشته شده
- پاسخ به سوالاتی که در انتهای هر گزارش کار آمده است.
- گزارش کار باید در قالب **یک فایل pdf** باشد.
- کافی است تنها توسط یکی از اعضای گروه در سامیاد بارگزاری شود.
- برای بارگزاری هر گزارش کار یک هفته و یک روز زمان دارید.
- دقت کنید گزارش کار را در بخش مربوط با گروه کلاسی خود بارگزاری نمایید.

سوالات:

۱. ثبات‌های DDR, PORT و PIN چگونه عمل می‌کنند؟ (۲ نمره)
۲. برای خواندن از صفحه کلید از چه روشی استفاده کردید؟ (۴ نمره)
۳. برای جلوگیری از چند بار زده شدن یک کلید، از چه روشی استفاده کردید؟ (۲ نمره)
۴. کد برنامه. (۲ نمره)