



گزارش آزمایش اول

مریم سعیدمهر

شماره دانشجویی: ۹۶۲۹۳۷۳

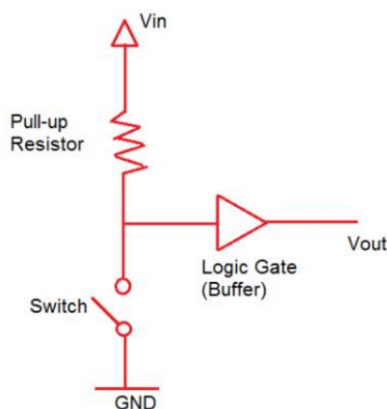
۱

- آ- ساختار مقاومت pull up و کاربرد آن را مختصراً شرح دهید.
ب- انواع روش های اتصال کلید به ریزپردازنده را نام ببرید.
ج- انواع روش های برنامه ریزی ریزپردازنده های AVR را نام ببرید.
د- ریزپردازنده AVR چند نوع حافظه دارد؟ نحوه استفاده از آنها چگونه است؟ تفاوت آنها در چیست؟

پاسخ

۱.۱

آ- اگر یکی از پایه های ریزپردازنده ورودی تعریف شده باشند ولی هیچ مقداری به آنها متصل نشده باشد و بین به Vcc یا GND متصل نباشد، مقدار دیجیتال این بین که توسط میکروکنترلر خوانده میشود تحت تاثیر نویز و وضعیت پایه های کناری و غیره قرار میگیرد و به طور کلی مقدار خوانده شده قابل پیش بینی نیست. اصطلاحاً به این حالت ، حالت شناور میگوییم . در صورتی که بخوایم وضعیت یک کلید را توسط این بین بخوانیم ، مشکل ذکر شده باعث میشود که در زمان باز بودن کلید مقدار دیجیتال خوانده شده قابل پیش بینی نباشد و هر دو مقدار صفر یا یک منطقی ممکن است خوانده شود. بنابراین نمیتوان با خواندن مقدار بین ورودی ، وضعیت کلید را تشخیص داد. برای رفع مشکل بیان شده میتوان از مقاومت pull up استفاده کرد که ساختار آن مطابق شکل ۱ است که اگر کلید بسته باشد مقدار صفر و اگر باز باشد مقدار یک به ورودی اعمال میشود.(البته ریزپردازنده ها اصولاً مقاومت بالاکش درونی دارند.)



شکل ۱: مقاومت بالاکش (Pull up)

- ب- ۱. فقط یک کلید و فعال کردن مقاومت های داخلی ریزپردازنده
 ۲. کلید با مقاومت بالا کش (pull up)
 ۳. کلید با مقاومت پایین کش (pull down)

- ج- ۱. میتوان ریزپردازنده را ابتدا با پروگرامر برنامه ریزی کرد و سپس آن را روی برد قرار داد.
 ۲. میتوان ریزپردازنده را به صورت onboard برنامه ریزی کرد یعنی ریزپردازنده را روی برد قرار داده و سپس به پروگرامر وصل کرده و آن را برنامه ریزی میکنیم.

د- سه نوع :

۱. حافظه برنامه یا program memory : برای قرار دادن برنامه و دستورات آن است و اگر ریزپردازنده خاموش شود اطلاعات آن ثابت خواهد ماند یعنی غیر فرار است.
 ۲. حافظه دیتا یا data memory : شامل چندین نوع رجیستر است :
- SRAM : که در طی برنامه و محاسبات آن و ذخیره مقادیر استفاده میشود و فرار است.
 - رجیسترهای I/O : غیر فرار هستند و به عنوان ورودی و خروجی های ریزپردازنده استفاده میشوند. البته بعضاً کاربردهای ویژه تری نیز دارند.
 - رجیسترهای extended I/O : کاربرد آنها برای زمانی است که رجیسترهای I/O فضای کافی نداشته باشند.
۳. EEPROM : اطلاعاتی که برای configuration ریزپردازنده میباشد در این قسمت ذخیره میشوند و غیر فرار هستند.

۲ نحوه عملکرد رجیسترهای DDRx و PORTx و PINx را توضیح دهید و پیکربندی لازم برای موارد زیر را بنویسید:

- آ- پورت A را طوری تعریف کنید که بیت های آن یک در میان ورودی خروجی باشند.
 ب- پورت B را به صورت خروجی تعریف کنید به طوری که مقاومت Pull up آن حذف نشود.

۱.۲ پاسخ

رجیسترهای گفته شده مربوط به پورت های ریزپردازنده هستند اگر پورت A را در نظر بگیریم ، DDRA رجیستری است که مقدار آن تعیین میکند این پورت ورودی است یا خروجی به شکل که اگر مقدار هر بیت از آن ۱ باشد یعنی خروجی و اگر صفر باشد یعنی ورودی است. اگر رجیستر خروجی شده باشد مقدار دیتای آن روی رجیستر PORTA قرار خواهد گرفت و اگر ورودی شده باشد باید مقدار ورودی آن را از طریق رجیستر PINA بخوانیم که در این حالت مقداری که در PORTA قرار میگیرد مشخص میکند برای این پین ورودی شده ، مقاومت بالا کش داخلی فعال شود یا خیر.





آ- باید به بیت های DDRA یک در میان صفر و یک داد یعنی $DDRA = 0b01010101 = 0x55$ یا به صورت $DDRA = 0b10101010 = 0xAA$

ب- ممکن نیست زیرا همان طور که گفته شده فقط برای پین هایی که ورودی شده اند میتوان مقاومت داخلی را فعال نمود.

۳ برای نمایش کاراکترهای زیر چه مقادیری را باید به ورودی خط دادهی 7-segments اعمال کرد.

d
A
H
F

بسته به اینکه نوع سون سگمنت چگونه باشد و بیت کم ارزش برای سگمنت a باشد یا سگمنت g (یعنی ترتیب پایه ها abcdefg یا gfedcba باشد) مقدار hex مناسب را از شکل ۲ باید انتخاب کرد.

Digit	Display	gfedcba	abcdefg
d		0x5E	0x3D
A		0x77	0x77
H		0x76	0x37
F		0x71	0x47

شکل ۲: جدول معادل hex کاراکترهای خواسته شده