



دانشگاه صنعتی شریف

دانشکده مهندسی برق

درس ساختار کامپیوتر و ریزپردازنده و آزمایشگاه (۲۵۵۳۵)

آزمایش PIC32-1

تهیه کنندگان:

دکتر محمد رضا موحدین

مجتبی پورعلی محمدی

به نام خدا

هدف:

آشنایی با ورودی (سوئیچ) و خروجی (LED) پردازنده PIC32 و روش debouncing ورودی‌ها

متن آزمایش:

بخش اول: کدی که در فایل PIC32_intro بعنوان Sample Program در اختیار شما قرار داده شده است را بر روی میکروکنترلر اجرا کنید. علت استفاده از دستور `sl` را بیان کنید.

بخش دوم: دستور `andi $8, $8, 0x0001` را از کد Sample Program حذف کنید تا حالت تمامی سوئیچ‌ها علاوه بر سوئیچ RB0 روی LEDها نمایش داده شوند. سپس مقدار `immediate` که در دستور `sl $8, $8, 8` است را از ۱ تا ۸ تغییر داده و سوئیچ‌های مختلف را روشن و خاموش کنید. مشاهده‌ی خود را توجیه کنید.

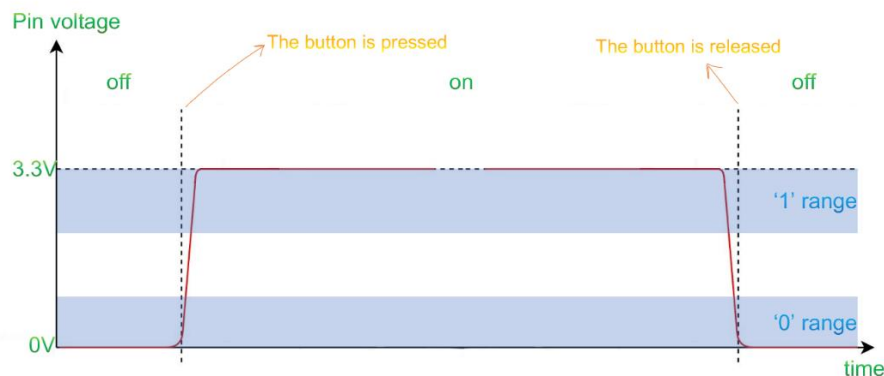
بخش سوم: کدی بنویسید که توسط کلیدها دو عدد تک رقمی را به صورت BCD گرفته و حاصل جمع آنها را در LEDها به صورت BCD نشان دهد به شکلی که با تغییر ورودی، خروجی بلافاصله تغییر حالت دهد و نیاز به اقدام خاصی، مثلاً ریست کردن میکروکنترلر نداشته باشد.

بخش چهارم: کدی بنویسید که توسط کلیدها دو عدد تک رقمی را به صورت BCD گرفته و حاصل ضرب آنها را در LEDها به صورت BCD نشان دهد به شکلی که با تغییر ورودی، خروجی بلافاصله تغییر حالت دهد و نیاز به اقدام خاصی، مثلاً ریست کردن میکروکنترلر نداشته باشد.

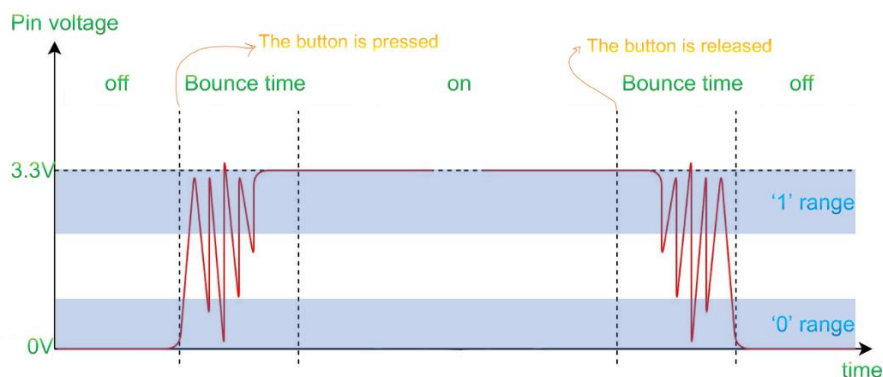
بخش پنجم: حال کدی بنویسید که اگر دکمه‌ی A0 فشرده شده باشد، LED مربوط به پین B8 روشن باشد و با رها شدن دکمه دوباره خاموش شود.

بخش ششم: حال کد قبلی را به گونه‌ای تغییر دهید که با هر بار زدن (tap) دکمه، LED مربوط به پین B8 تغییر وضعیت دهد. (برخلاف بخش الف پس از فشردن کلید و رها کردن آن، LED آنقدر روشن بماند تا باری دیگر دکمه زده شده و خاموش شود.) این کد را بنویسید و ببینید به چه مشکلی برمی‌خورید.

مشکل این است که ما انتظار داریم به محض فشردن شدن کلید، باید مقدار پین متصل به آن یک شود و به محض رها کردن کلید باید مقدار آن صفر شود مانند شکل زیر:



ولی در عمل اتفاق دیگری می افتد و پس از فشرده شدن کلید یا پس از رها کردن آن کمی اعوجاج در سیگنال میبینیم:



پس بعد از فشرده شدن کلید، مقدار پین ورودی در یک بازه‌ی زمانی کوتاه بین صفر و یک تغییر می‌کند (به این فرایند، bouncing گفته می‌شود) و پس از آن به صورت پایدار یک می‌شود و یک باقی می‌ماند تا اینکه دکمه رها شود. پس از رها شدن دکمه باز هم سیگنال متناوباً بین صفر و یک تغییر کرده تا اینکه نهایتاً به صورت مستمر صفر شود. سرعت بالای پردازنده در نمونه‌برداری از ورودی به معنای خواندن تعداد قابل توجهی صفر و یک‌های غیر متوالی در زمان bounce می‌باشد.

راه حل این مشکل را debouncing می‌نامند و یکی از ساده‌ترین روش‌ها آن است که اگر تعداد زیادی (مثلاً ۲۰۰ بار) یک یا صفر کاملاً متوالی توسط پردازنده خوانده شد، آنگاه آن را به عنوان یک یا صفر معتبر قبول نماید. توجه کنید که این فرایند بایستی هم برای زده شدن کلید (یک شدن) و هم برای رها شدن آن (صفر شدن) انجام شود. اکنون با این روش، کد این بخش را اصلاح کنید و عملکرد صحیح آن را امتحان نمایید.