

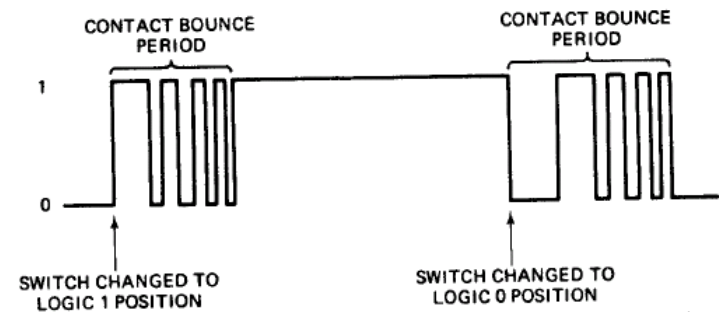
Switch Debounce

Switch Debouncing

- Mechanical switches play an important and extensive role in practically every computer, microprocessor and microcontroller application.
- Mechanical switches are inexpensive, simple and reliable. However, switches can be very noisy electrically.
- The apparent noise is caused by the closing and opening action that seldom results in a clean electrical transition. The connection makes and breaks several, perhaps even hundreds, of times before the final switch state settles.
- The problem is known as switch bounce. Some of the intermittent activity is due to the switch contacts actually bouncing off each other. Imagine slapping two billiard balls together. The hard non-resilient material doesn't absorb the kinetic energy of motion.

Switch Debounce

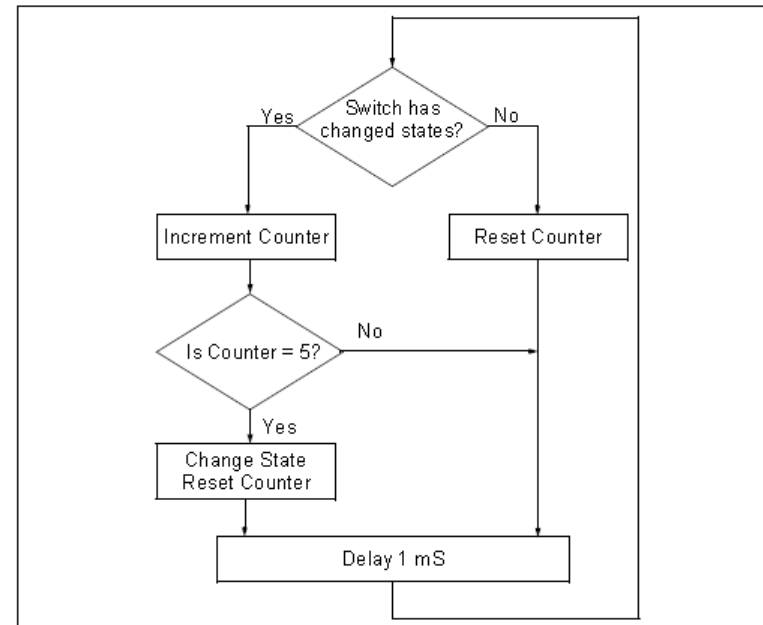
- The consequences of uncorrected switch bounce can range from being just annoying to catastrophic. For example, imagine advancing the TV channel, but instead of getting the next channel, the selection skips one or two. This is a situation a designer should strive to avoid.
- Switch bounce has been a problem even before the earliest computers.
- Debouncing in software eliminates this problem.



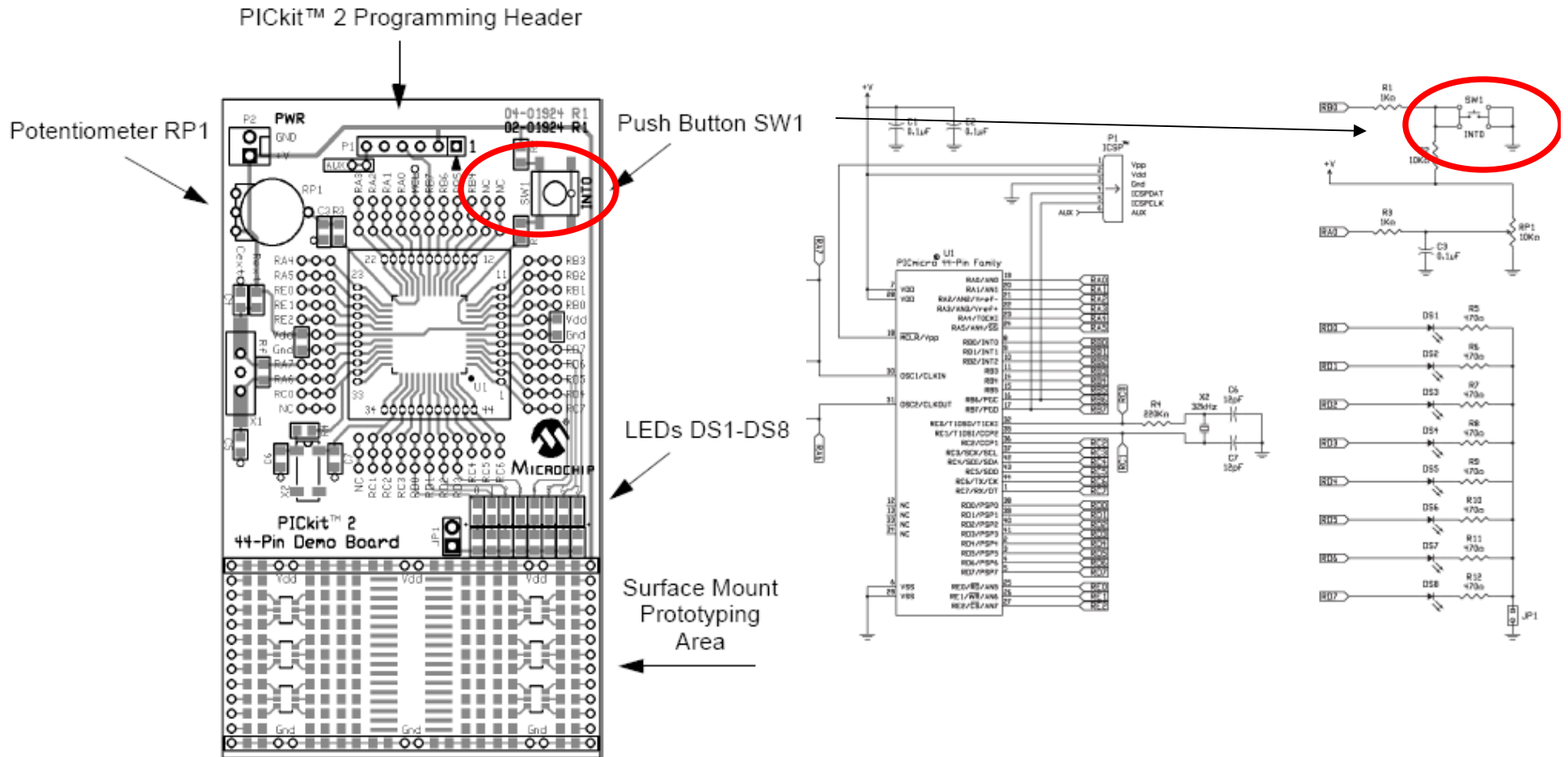
Notional Solution

- One of the simplest ways to switch Debounce is to sample the switch until the signal is stable or continue to sample the signal until no more bounces are detected.
- How long to continue sampling requires some investigation. However, 5 mS-20mS is usually plenty long, while still reacting fast enough that the user won't notice it.
- A notional solution is to sample the line at a 1 mS rate waiting for a number of sequential state changes, which is a simple matter of counting to 5, then resetting the counter every time it's still in the original unchanged state.

FIGURE 3-6: SIMPLE SWITCH DEBOUNCE PROGRAM FLOW



44 Pin Demo Schematic and Layout



Exercise

- Navigate to: C:\EET250\16F887\Lesson 11 Debounce with Demo Board\debounce
- Invoke debounce.mcp
- View source code. We are using the Pushbutton on Demo board—hooked to RB0. The SPR settings shut down all analog contention for shared pins. The only issue is RB0 that has a shared usage with an ADC input

```
ANSEL =0x00;  
ANSELH =0x00;  
CM1CON0 =0;  
CM2CON0 =0;
```

- Select as debugger the PICKIT2
- Build/program part
- Run –operation should show a clean toggling of led upon each switch bounce
- Change constant of Twentym to smaller value of 20.
Rebuild/Reprogram/rerun
- DO you notice a difference in the debounce function when pressing the switch?