MSP430
Digital IOs
MSP430-2013

• Has an 8-bit digital IO port P1
  - The digital I/O features include:
    • Independently programmable individual I/Os
    • Any combination of input or output
    • Individually configurable P1 and P2 interrupts
    • Independent input and output data registers
    • Individually configurable pullup or pulldown resistors
  - Interrupt support is also available
  - All these port pins are brought out on the development board, and hence to our breadboards
Using A Switch as Input

- Kit should contain a DIP switch block
  - DIP = Dual In-line Package
P1 Digital IO Control

- Digital I/O is configured by user software
  - Input Register P1IN
  - Output Register P1OUT
  - Direction Register P1DIR
  - Pullup/Pulldown Resistor Enable Register P1REN
  - Function Select Registers P1SEL and P1SEL2
    - Most likely will not need to use these
  - Each pin in P1 has interrupt capabilities
    - We will cover this next week
P1 Example

//****************************************************************************
// MSP430F20xx Demo - Poll P1 With Software with Internal Pull-up
// Description: Poll P1.4 in a loop, if hi P1.0 is set, if low, P1.0 reset.
// Internal pullup enabled on P1.4.
// ACLK = n/a, MCLK = SMCLK = default DCO
//
//    MSP430F20xx
//    --------------
//        /|\       XIN|--
//        |      |
//        --|RST    XOUT|--
//        /|\       | R
//        --o--| P1.4-o P1.0|-->LED
//        \|/
//
// M. Buccini / L. Westlund
// Texas Instruments Inc.
// September 2005
// Built with CCE Version: 3.2.0 and IAR Embedded Workbench Version: 3.40A
//****************************************************************************
P1 Example - cont

```c
#include <msp430x20x3.h>

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop watchdog timer
    P1DIR = 0x01;              // P1.0 output, else input
    P1OUT = 0x10;              // P1.4 set, else reset
    P1REN |= 0x10;             // P1.4 pullup

    while (1)
    {
        if (0x10 & P1IN) P1OUT |= 0x01; // if P1.4 set, set P1.0
        else P1OUT &= ~0x01;           // else reset
    }
}
```