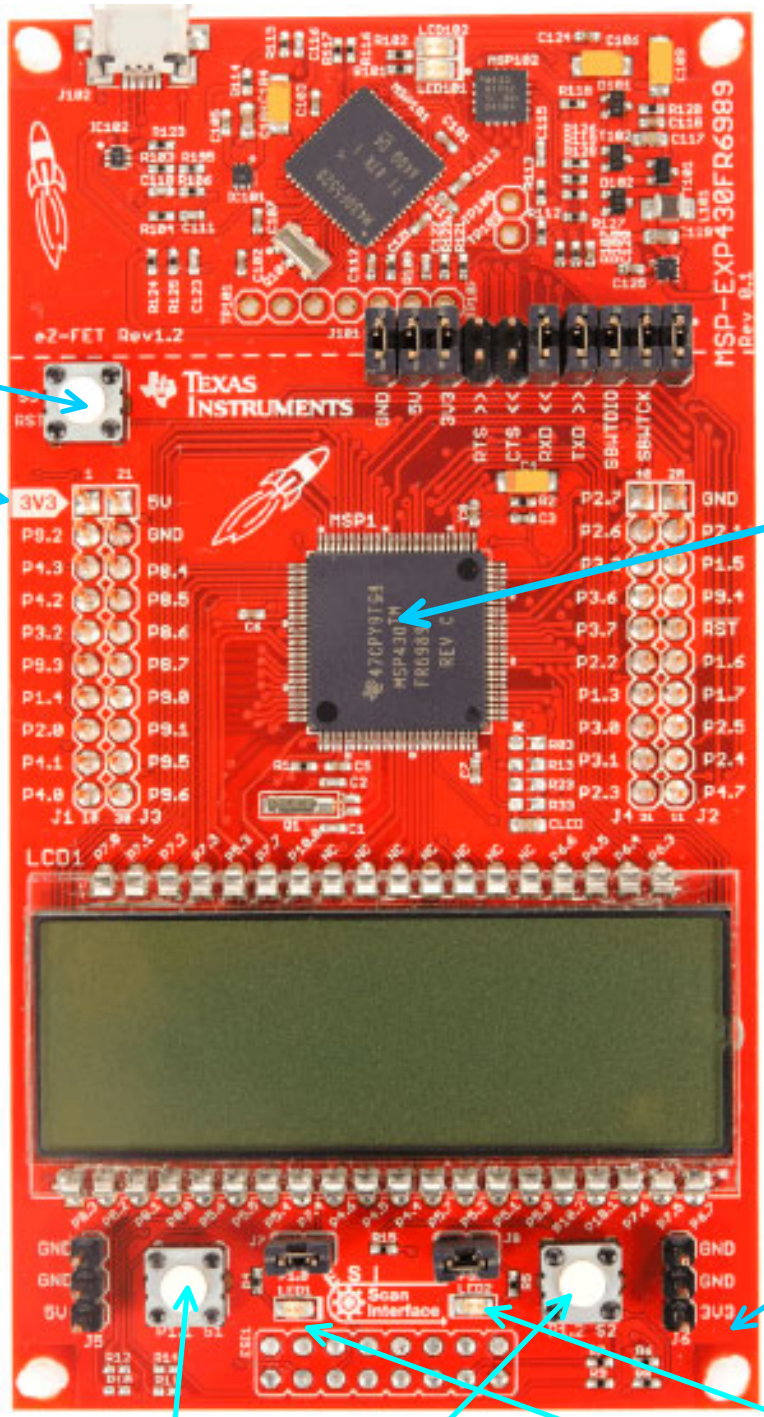


Reset Button S_3

V_{CC} 3.3V

MSP430FR6989IPZ

Ground

MSP-EXP430FR6989 LaunchPad Development Kit

Pushbuttons S1 & S2

Active Low

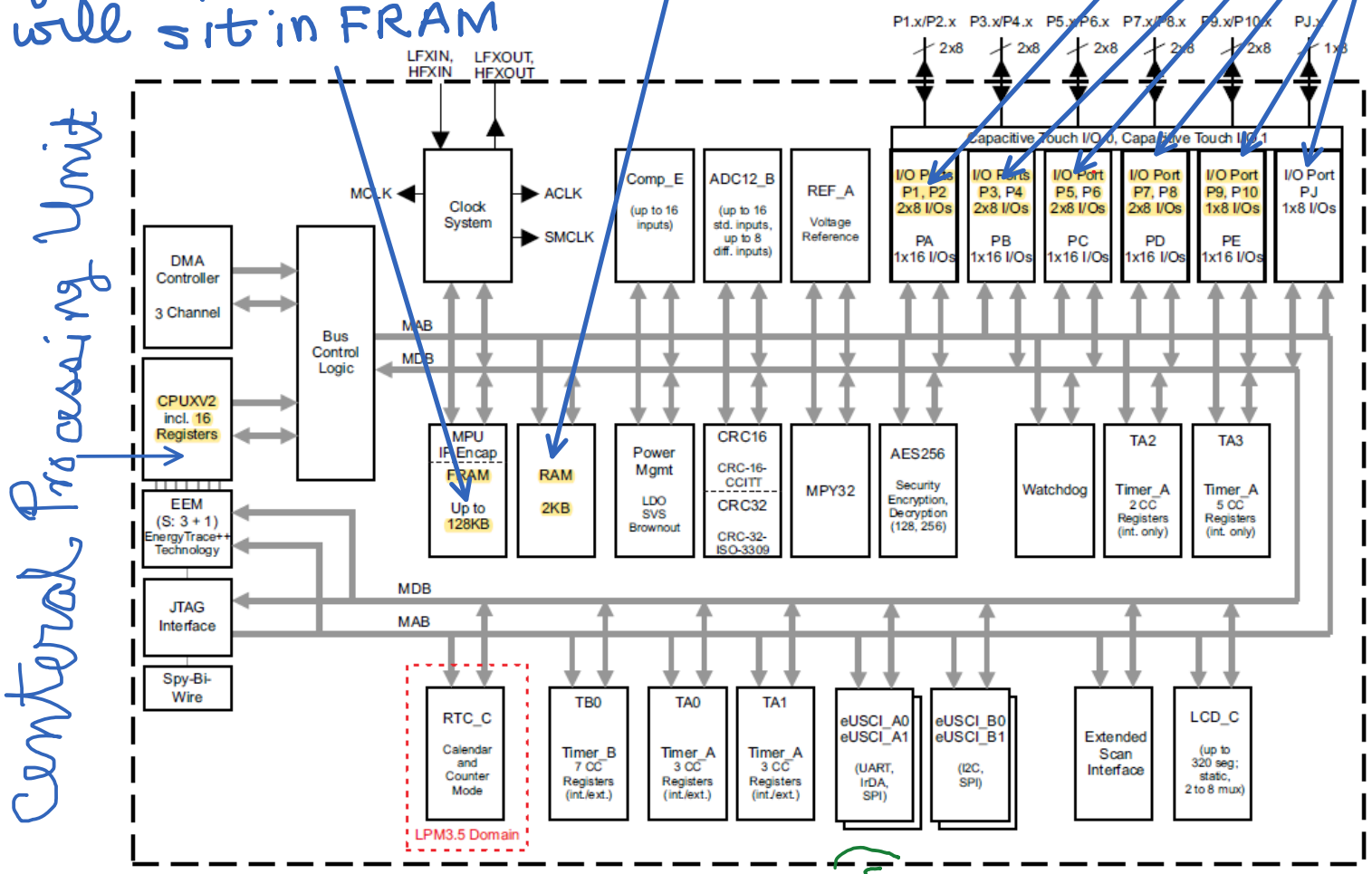
LEDs Red & Green

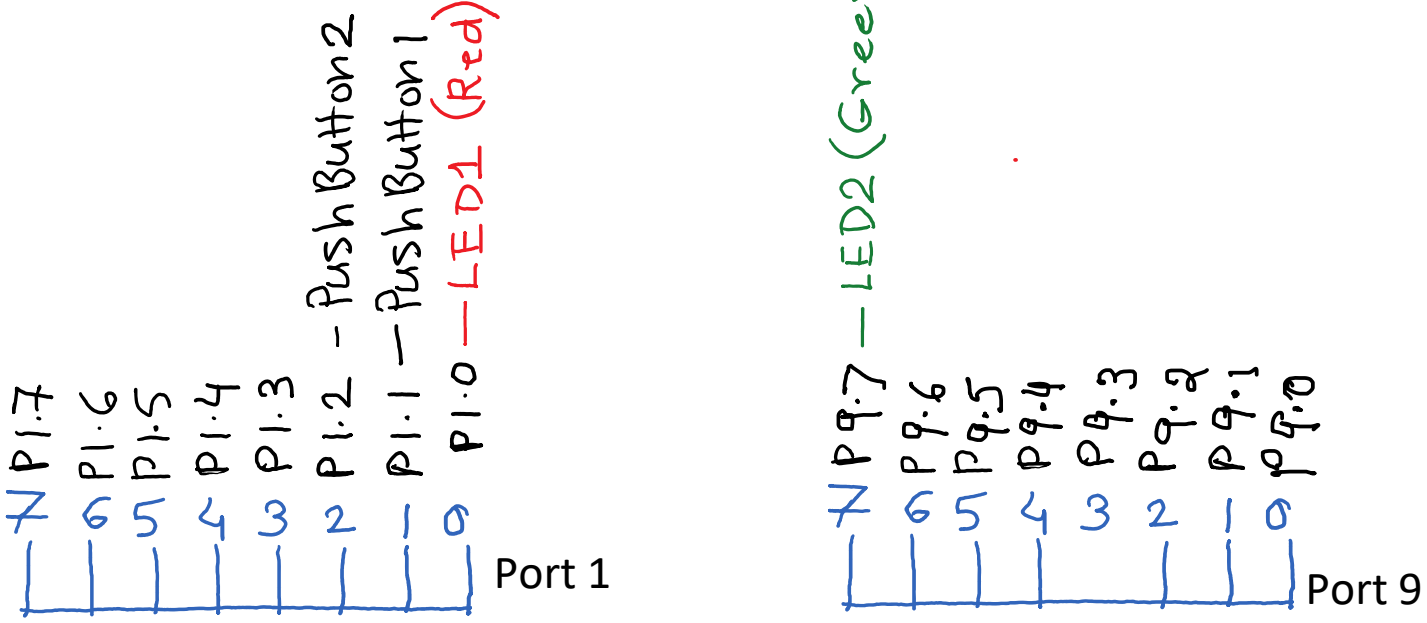
From Device User Guide
SLAS789D

Variables and Stack will be in RAM

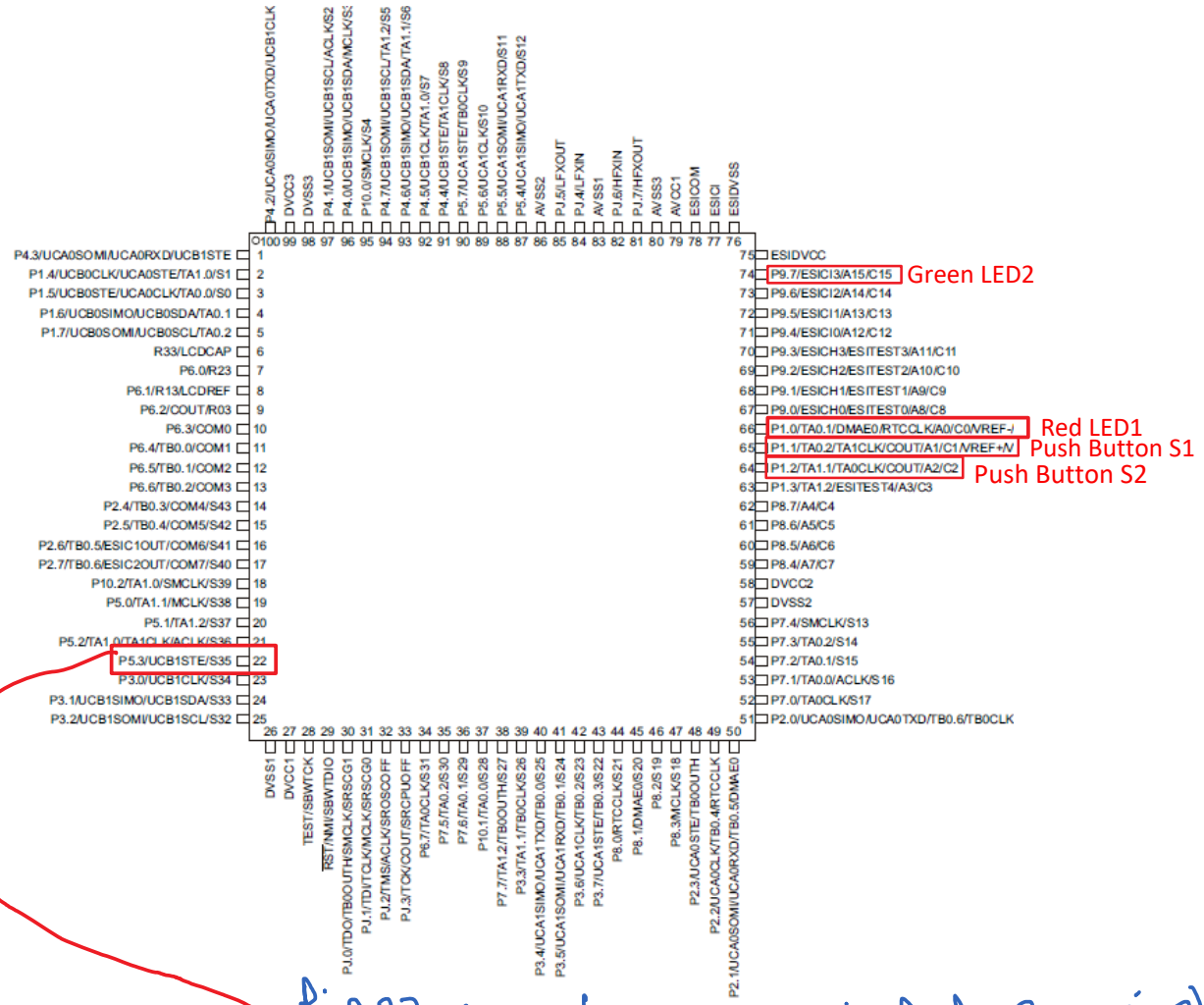
Ports for Digital GPIO (General Purpose Input Output)

Your program will sit in FRAM



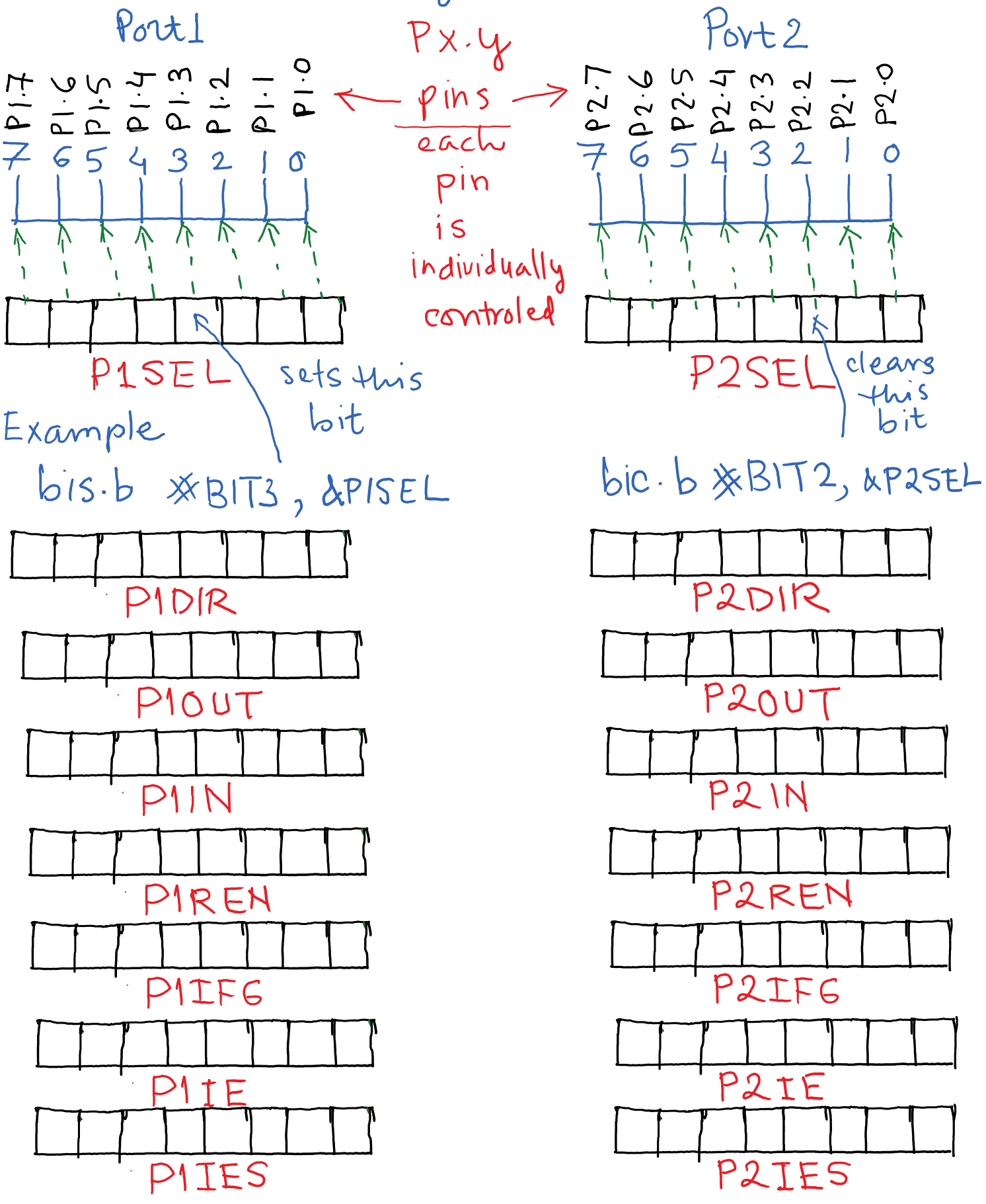


PART NUMBER	PACKAGE	BODY SIZE(2)
MSP430FR6989IPZ	LQFP (100)	14 mm × 14 mm
MSP430FR6989IPN	LQFP (80)	12 mm × 12 mm
MSP430FR5989IPM	LQFP (64)	10 mm × 10 mm
MSP430FR5989IRGC	VQFN (64)	9 mm × 9 mm



P5.3/UCB1STE/Sx	22	S35			General-purpose digital I/O USCI_B1: Slave transmit enable (SPI mode) LCD segment output (segment number is package specific)

GPIO Registers



Configuring Px.y

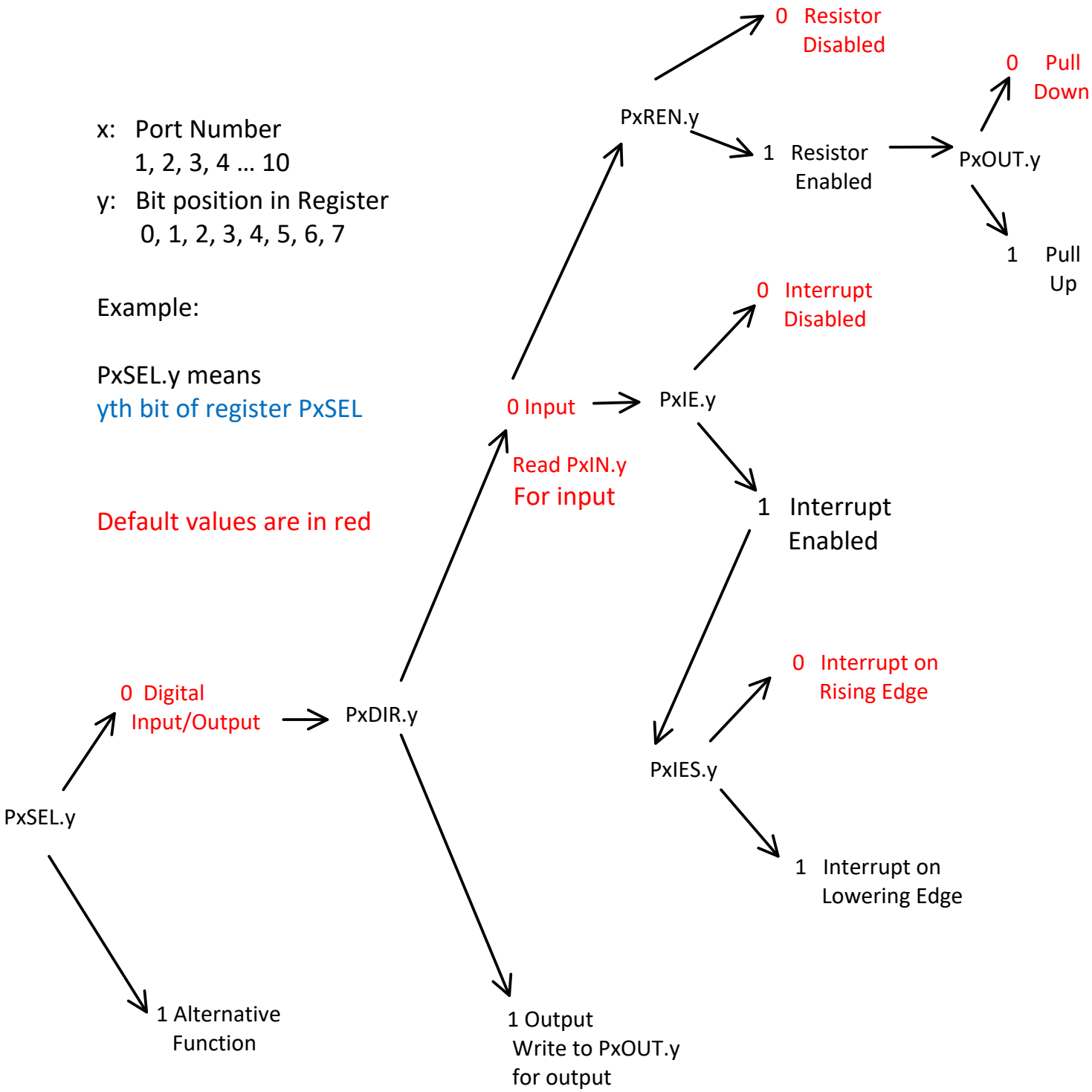
x: Port Number
1, 2, 3, 4 … 10

y: Bit position in Register
0, 1, 2, 3, 4, 5, 6, 7

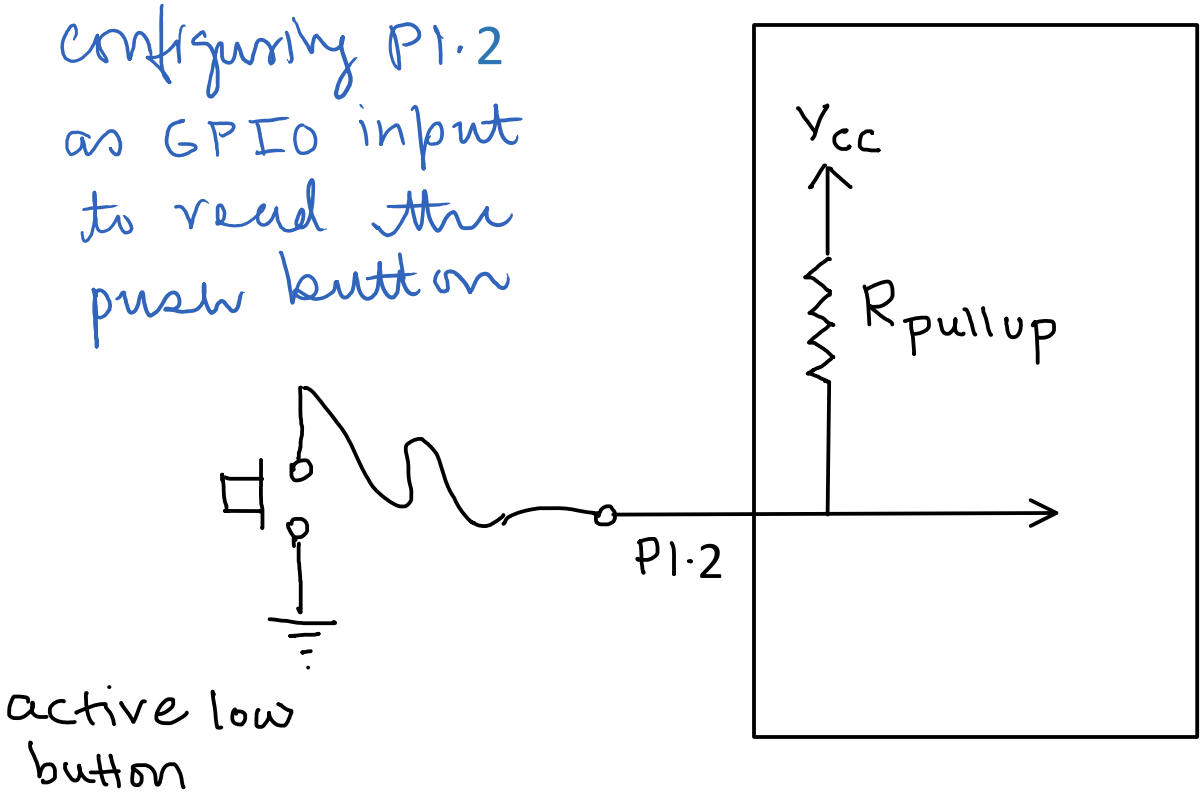
Example:

PxSEL.y means
yth bit of register PxSEL

Default values are in red



configuring P1.2 as GPIO input to read the push button




```
bic.b  #BIT2,&P1SEL   default (digital GPIO)
bic.b  #BIT2,&P1DIR   default (input)
bis.b  #BIT2,&P1REN   (resistor enable)
bis.b  #BIT2,&P1OUT   (pull up resistor)
bic.b  #BIT2,&P1IE    default (disable interrupt)
```

read input value

```
bit.b  #BIT2,&P1IN  ⟶ use in a conditional statement
```

Notes

By default P x.y is configured for:

digital general purpose input
no resistor enable
no interrupt enable

Good practice for writing to output

or bic.b

```
bis.b  #BIT3,&P1OUT   → write output to P1.3
bis.b  #BIT3,&P1DIR   → set P1.3 as output pin
```

Working with LEDs

Example Red LED (P1.0)

```
bis.b  #BIT0,&P1OUT
bis.b  #BIT0,&P1DIR   ← set P1.0 for output
                      → LED turns on
xor.b  #BIT0,&P1OUT   ← toggle P1.0
```

a	b	XOR
0	0	0
0	1	1
1	0	1
1	1	0

1 XOR 0 = 1 toggle

1 XOR 1 = 0

Program: Blinky

```
                        ; Main loop here

            bis.b   #BIT0, &P1OUT
            bis.b   #BIT0, &P1DIR          ; Set P1.0 as output pin
                                           ; Note P1.0 is connected to the red LED
            bic.w   #LOCKLPM5,&PM5CTL0 ; Disable the GPIO power-on default
                                           ; high-impedance mode
loop:       xor.b   #BIT0, &P1OUT          ; Toggle P1.0

            mov.w   #65535, R15            ; Put delay value in R15
sub_loop:   dec.w   R15                    ; Decrease R15
            jnz     sub_loop               ; Delay over?
            jmp     loop                   ; Repeat
```

} delay by wasting time