

# ENGR 1202

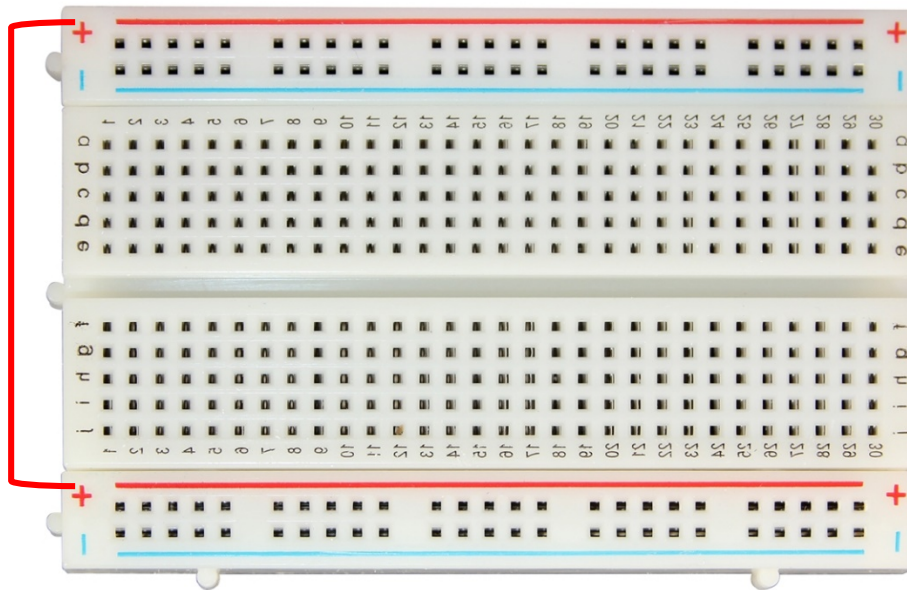
## Lecture 3



# Breadboard configurations

Note: row 1, cols. a-e are connected; row 1, cols. f-j are connected (all of row 1 is not connected)

power rails,  
each rail  
is separate  
from the  
other



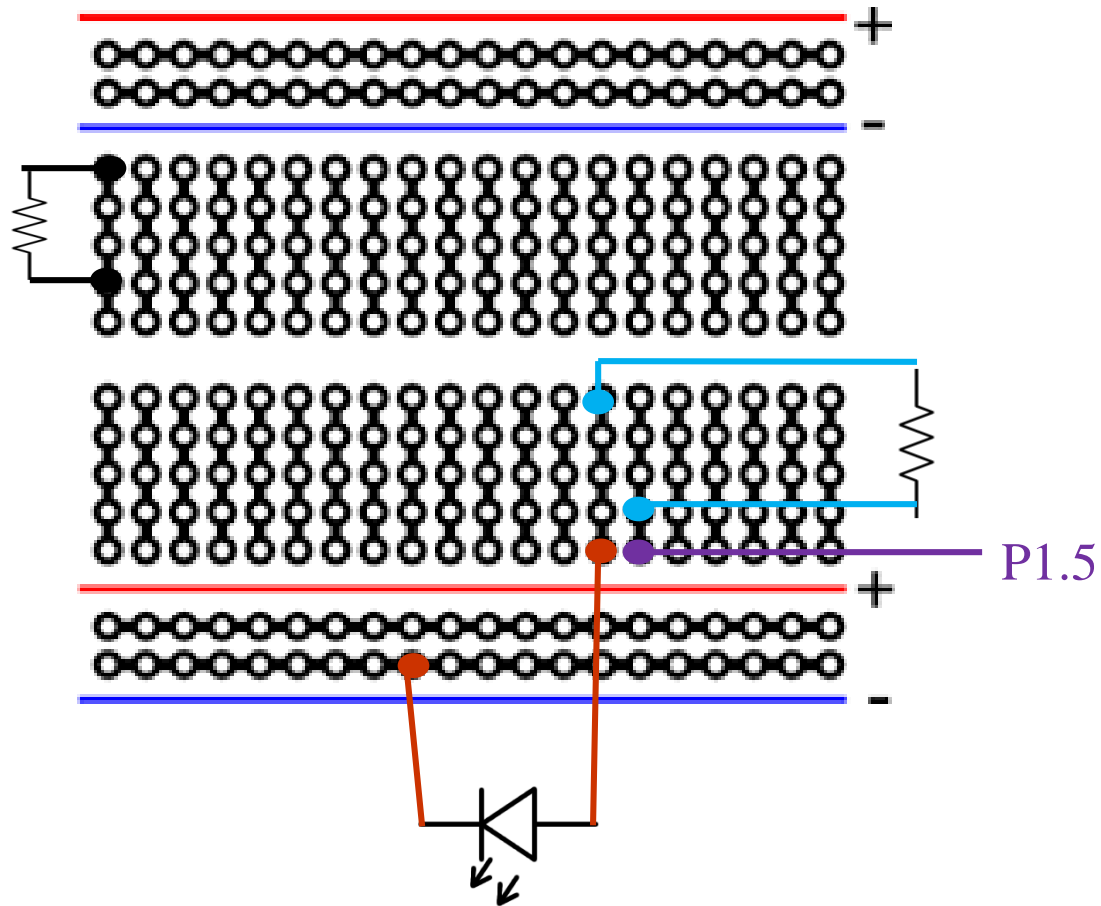
ground rails,  
each rail  
is separate  
from the  
other



# Breadboard configurations

Here is an image of the breadboard “schematic”

Which resistor is wrong/bad?



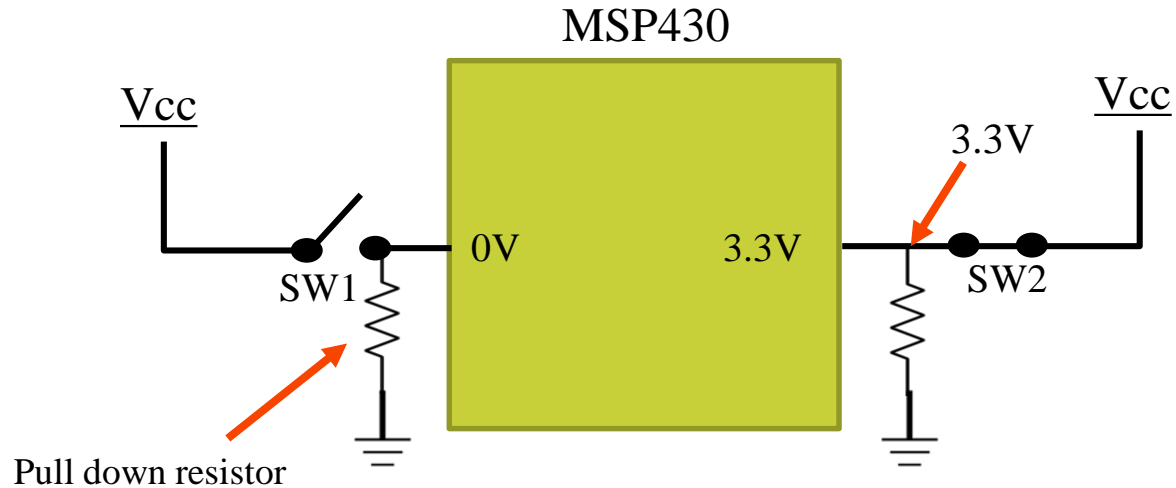
# Refresher - Lab 2 code

```
int main{
    long int i;
    WDTCTL = ...
    P1DIR = ...
    while(1){          //Do this forever
        //Light LED1 only
        P1OUT = ...
        //Delay for 2 seconds
        for(i = 0;...
        //Light LED2 only
        P1OUT = ...
        //Delay for 2 seconds
        for(i = 0;...
    } //end of the while instruction
return 0;
}
```

The // line is a comment;  
Anything after the // is  
ignored



# Refresher - Switch



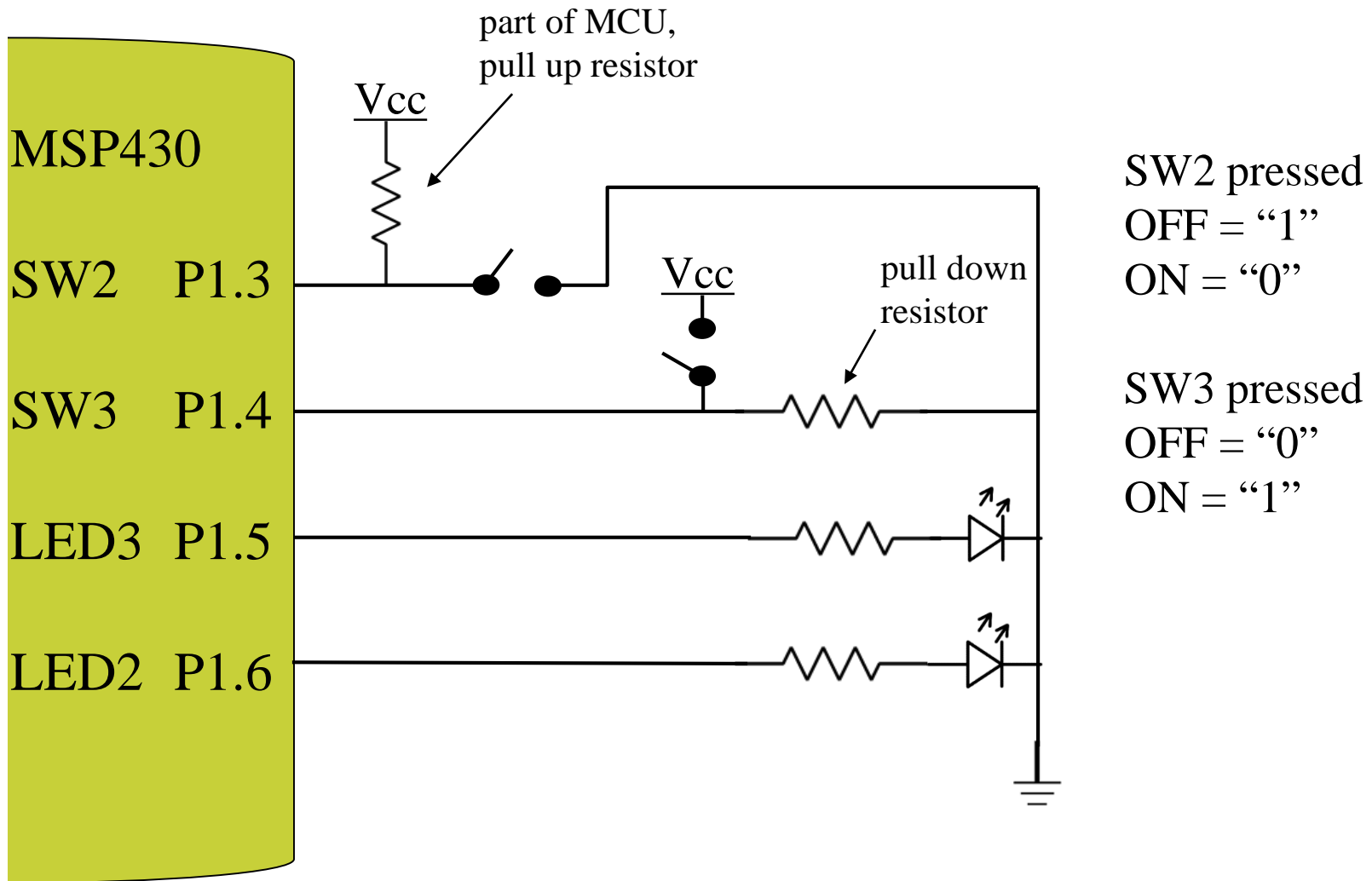
data = P1IN;

switch pressed = port1, bit 3

data contains  $0000\ 1000_2 = 0x08 = 8_{10}$



# Lab 3 Schematic



# What we have

New instruction for “pull up” resistor in the MCU

```
P1REN = 0x08; //more on this later
```

Now our PORT1 looks like:

7	6	5	4	3	2	1	0
Unused	LED2	LED3	SW3	SW2	Unused	Unused	Unused



# How they work

When SW2 is pressed, reading 0

SW2 is not pressed, reading 1

When SW3 is pressed, reading 1

SW3 is not pressed, reading 0

char readit;

readit = P1IN;

readit = readit & 0x08; //SW2 isolated

$$\begin{array}{r} 0001\ 0000 \\ \& \underline{0000\ 1000} \\ 0000\ 0000 \end{array}$$

↙ SW2 pressed

$$\begin{array}{r} 0001\ 1000 \\ \& \underline{0000\ 1000} \\ 0000\ 1000 \end{array}$$

↙ SW2 not pressed





# Build a table for Lighting LED 2

SW3	SW2	P1IN	P1OUT	LED2	LED3
	0 (pressed)	0x00	0x00	OFF	
	1 (not pressed)	0x08	0x40	ON	
	0 (pressed)	0x00	0x00	OFF	
	1 (not pressed)	0x08	0x40	ON	

Note: the table will change when accommodating for P1REN

```
readit = P1IN & 0x08;
```

```
if(readit == 0x00) P1OUT = 0x00; //SW2 pressed, turnoff LED2
```

```
else if(readit == 0x08) P1OUT = 0x40;
```

...

	7	6	5	4	3	2	1	0
	Unused	LED2	LED3	SW3	SW2	Unused	Unused	Unused

# Build a table for Lighting LED 3

SW3	SW2	P1IN	P1OUT	LED2	LED3
0 (not pressed)					ON
0 (not pressed)					ON
1 (pressed)					OFF
1 (pressed)					OFF

Note: the table will change when accommodating for P1REN

```
readit = P1IN & 0x10;
```

```
if(readit == 0x__) P1OUT = 0x__; //SW3 pressed, turnoff LED3
```

```
else if(readit == 0x__) P1OUT = 0x__;
```

...

	7	6	5	4	3	2	1	0
	Unused	LED2	LED3	SW3	SW2	Unused	Unused	Unused

# Build a table for Lab 3

SW3	SW2	P1IN	P1OUT	LED2	LED3
0 (not pressed)	0 (pressed)	0x00	0x20	OFF	ON
0 (not pressed)	1 (not pressed)	0x08	0x60	ON	ON
1 (pressed)	0 (pressed)	0x10	0x00	OFF	OFF
1 (pressed)	1 (not pressed)	0x18	0x40	ON	OFF

Note: the table will change when accommodating for P1REN

```
readit = P1IN & 0x18;
```

```
if(readit == 0x00) P1OUT = 0x20; //SW2 pressed, turnoff LED2
```

```
else if(readit == 0x08) P1OUT = 0x60;
```

...

	7	6	5	4	3	2	1	0
	Unused	LED2	LED3	SW3	SW2	Unused	Unused	Unused

# Isolating inputs using “AND”

(1) Read in P1, (2) and to id if switches are pressed, (3) Light LED

→With both: readit = P1IN & 0x18;

Remember that if statements work like this:

if(expression is true) execute this;

example: if ((P1IN & 0x18) == 0x08) P1OUT = 0x60;

Read port1, then look at only sw2 & sw3 inputs by using an “AND” operation

Equal check (not single =, which is assignment)

From table; neither switch was pressed

0x60;  
both LEDs lit

Warning-to be examined again still

So, lab3 can be done with this simple logic in a while(1) loop

```
if (____) P1OUT = 0x____;  
else if (____) P1OUT = 0x____;  
else if (____) P1OUT = 0x____;  
else P1OUT = 0x____;
```

# Table with P1REN accounted for

To get SW2 to work with the pull up resistor, you must use

`P1REN = 0x08;`

`P1OUT = 0x08;`

every time you output to port 1.

Therefore, the LED output instruction combined with setting the pull up resistor is in the new table below:

SW3	SW2	P1IN	P1OUT	LED2	LED3
0 (not pressed)	0 (pressed)	0x00	0x28	ON	OFF
0 (not pressed)	1 (not pressed)	0x08	0x68	OFF	OFF
1 (pressed)	0 (pressed)	0x10	0x08	ON	ON
1 (pressed)	1 (not pressed)	0x18	0x48	OFF	ON

# All together...

Therefore, the whole assignment for lab3 is:

```
P1DIR = _____;
```

```
P1REN = 0x08;
```

```
P1OUT = 0x68;
```

```
while(1){
```

```
    your logic
```

```
    ...
```

```
}
```

Don't forget about include, main, and disable watchdog timer.