
ENGR 1202

Lecture 2



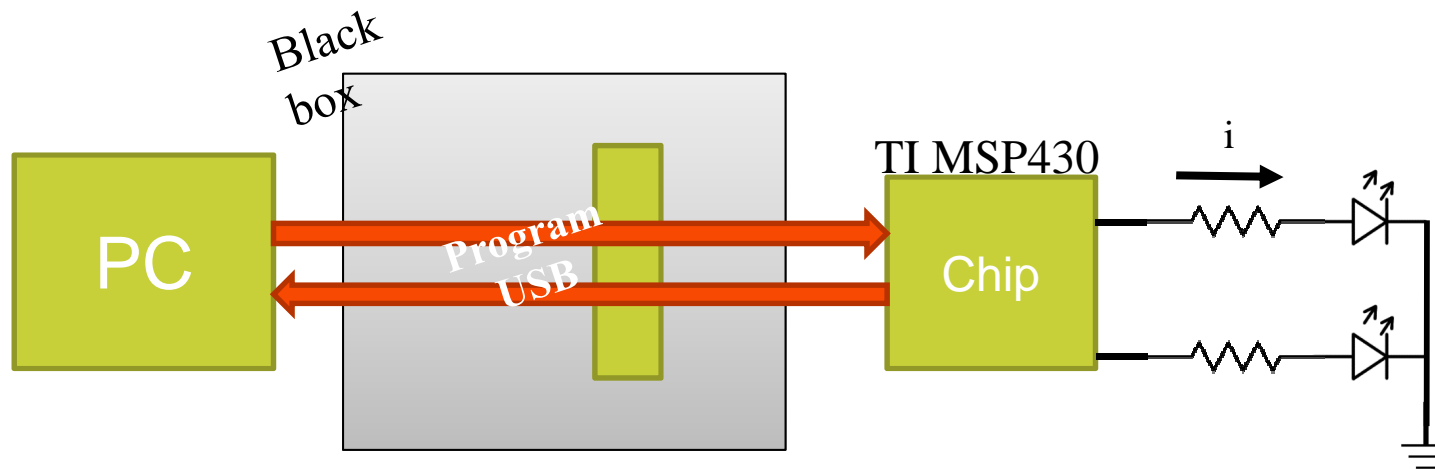
Review of binary → hex

Nibble (4 digits, binary)	Decimal	Hex digit
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

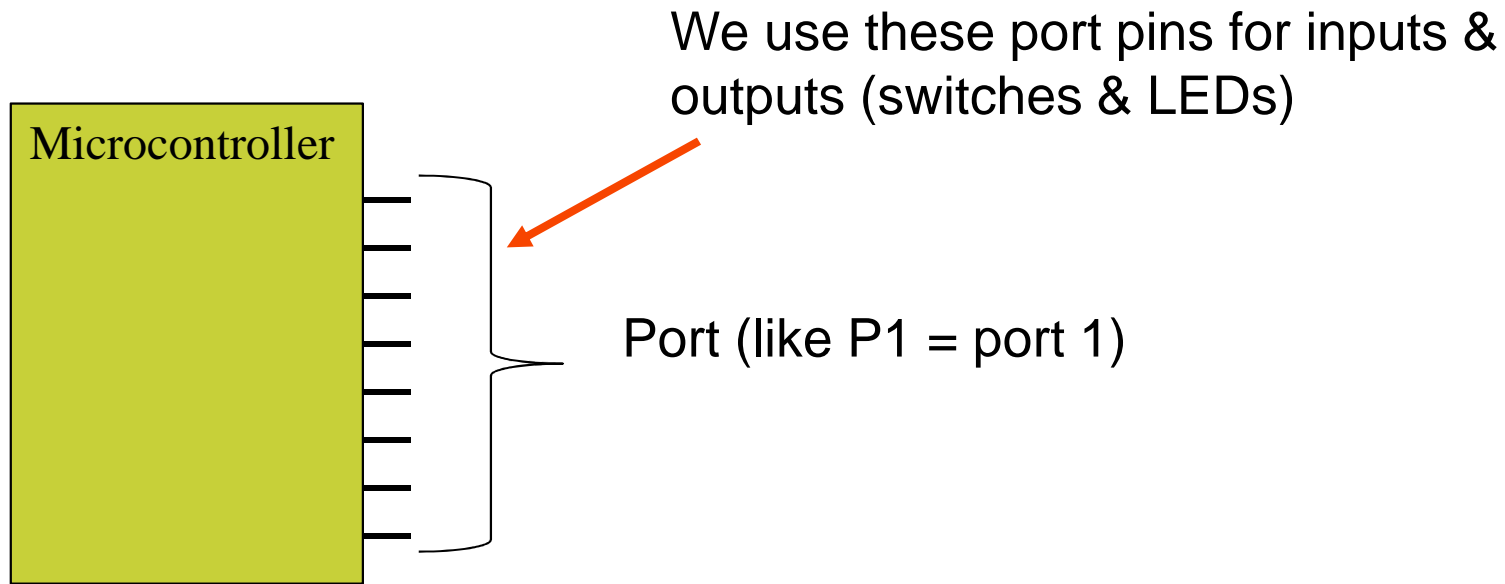


General Setup with Microcontrollers

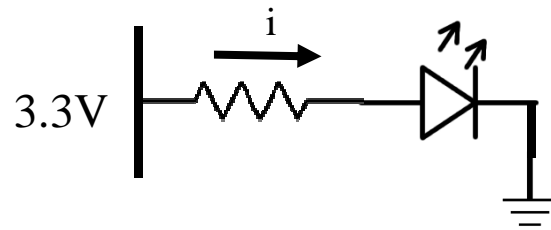
- Manipulate bytes → Organize
→ AND/OR
→ Microcontroller port



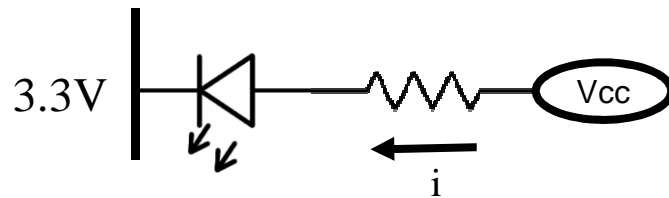
Class Concepts



Wiring Configurations



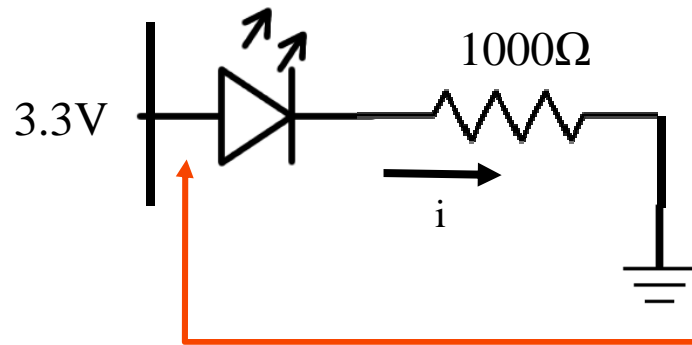
3.3V = ON! → Logic 1
0.0V = OFF → Logic 0



3.3V = OFF → Logic 1
0.0V = ON! → Logic 0



Think of an LED (and Resistor)

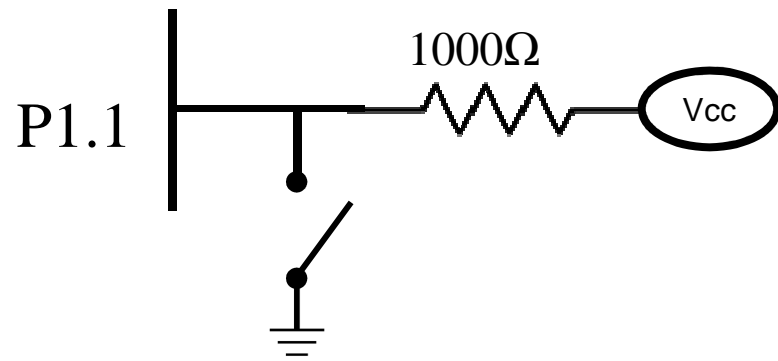


To make it light up, give this wire 3.3V (“1”).

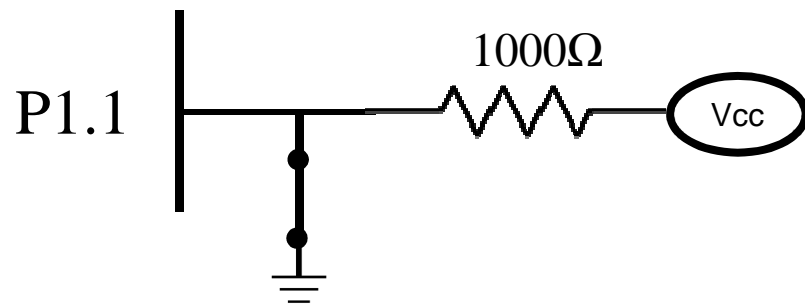
$$V=IR, \quad 3.3V = i \cdot 1000\Omega$$
$$i(A) = 3.3/1000 = 3.3\text{mA}$$

3.3mA will flow through the LED, lighting it up

Think of a switch



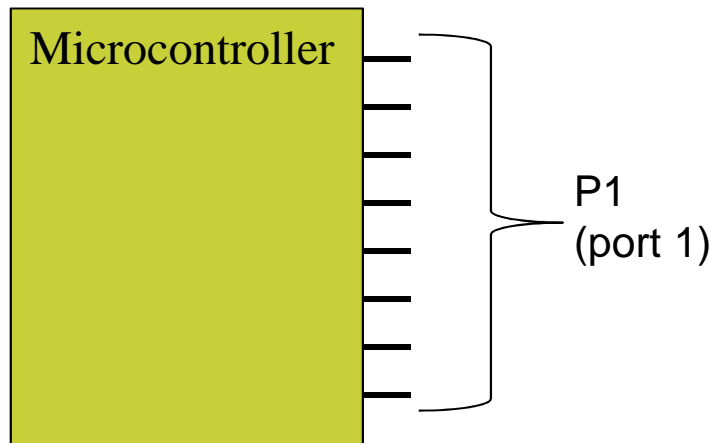
When you do not press the switch, this wire will be 3.3V (the micro-controller will read this; micro-controller ports don't "consume" voltage like a resistor)



When you press the switch, current flows from Vcc to Ground. The wire will be 0V (the micro-controller will read this)

Ports and micro-controllers

The previous two examples are ways we work with input (switches) and output (LEDs) for our micro-controller.



With the MSP430, we can read port 1 by using the code:

```
data = P1IN;
```

Annotations for the code line:

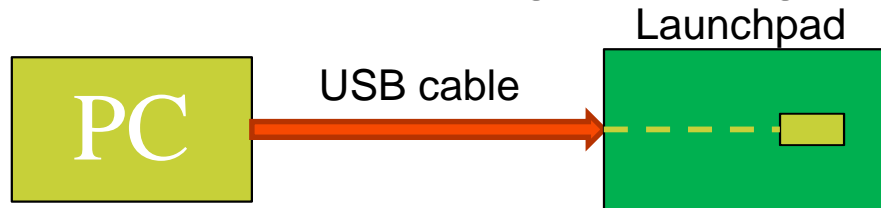
- Arrow from "data" to "a variable"
- Arrow from "=" to "assignment"
- Arrow from "P1" to "port # & what to do"
- Arrow from "IN" to "need for each C line"

A port is 8 bits, so the result of this could be that data holds 0xFF (binary 1111 1111)

Assignment 2

- ❖ See the Moodle site for assignment
- ❖ Download software, run your board

General concept of programming:

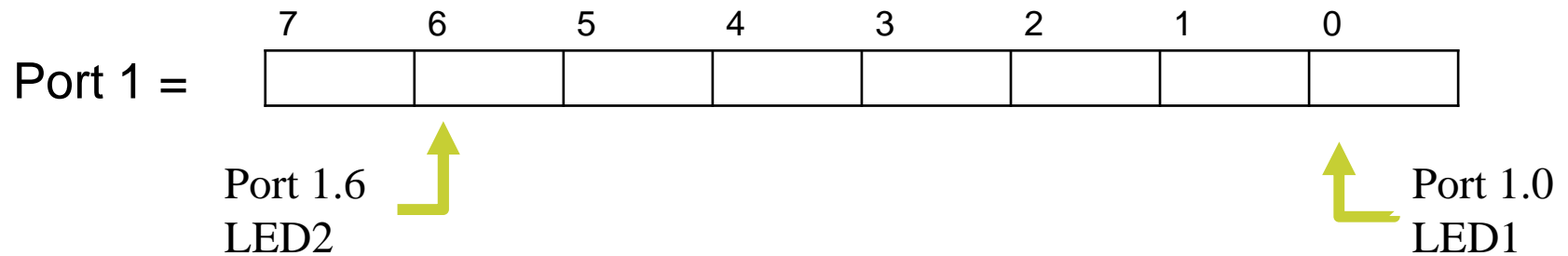


- ❖ Write code
- ❖ Compile code
- ❖ Download to target →
- ❖ Run code on target

See Moodle for the due date, and make sure to demo to TA.

Assignment 2 - “wiggling” a port between 1 & 0

A port is 8 bits organized together

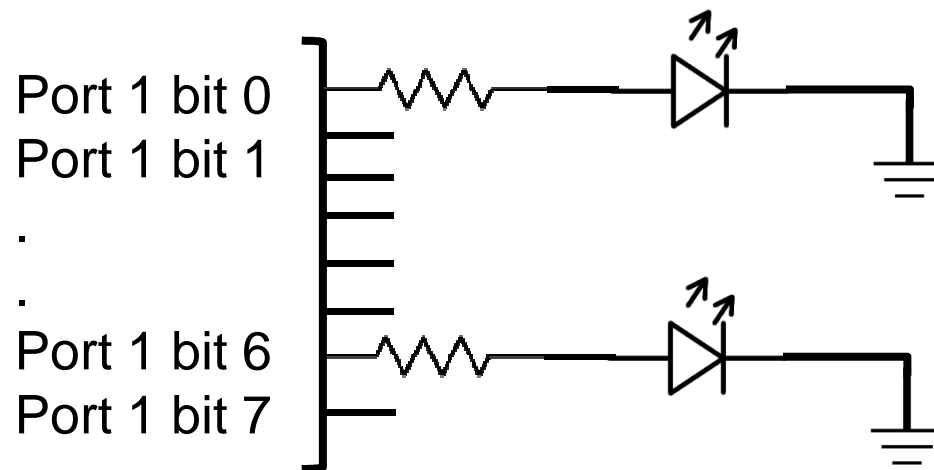


- P1.0 is LED1
- P1.6 is LED2



MSP430 Ports & LEDs

Assuming the diagram below, what value must Port 1 be to light up both LEDs?



Answer: 0100 0001

What is this in hex?

0100 0001 → 0x41

MSP430 Ports & LEDs

Port 1 is P1.7 – P1.0 which is (1000 0000 to 0000 0001)

Light up LED1 (P1.0) only

→ P1OUT = 0x01;

Light up LED2 (P1.6) only

→ P1OUT = 0x40;

Light up both LED1 and LED2

→ P1OUT = 0x41;

However, the direction of the port needs to be set first:

output is “1”

→ P1DIR = 0x41;

[only P1.0 & P1.6 become outputs (1)]



Delays


Lastly, we must introduce a delay. One easy method to implement is using a C “for” loop:

```
long int i;
```

```
...
```

```
for(i = 0; i < 100000; i++);
```

Change this
variable until
you get a 1.5
second delay



Now we pull all these concepts together to complete assignment 2.



Example of 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

