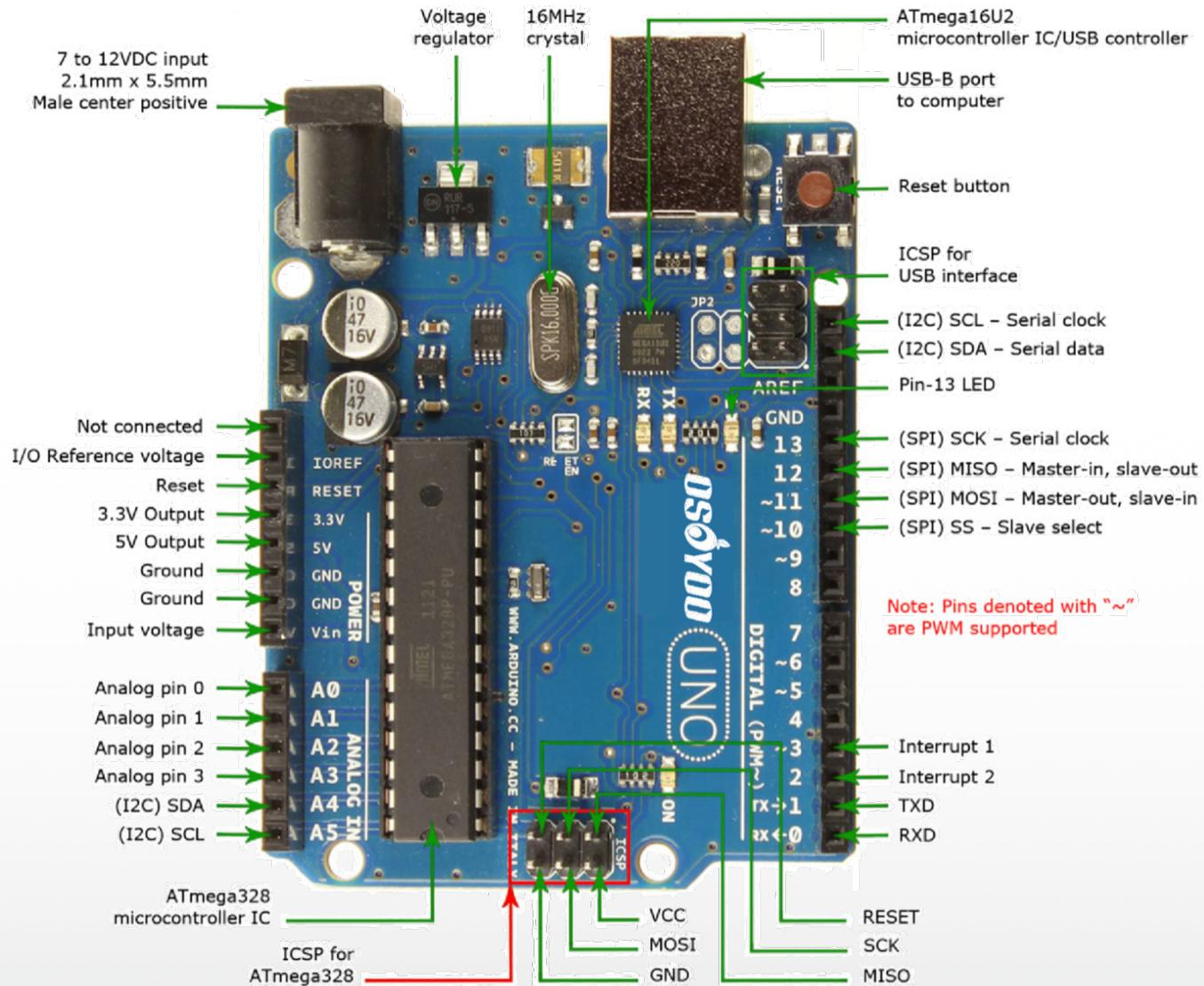


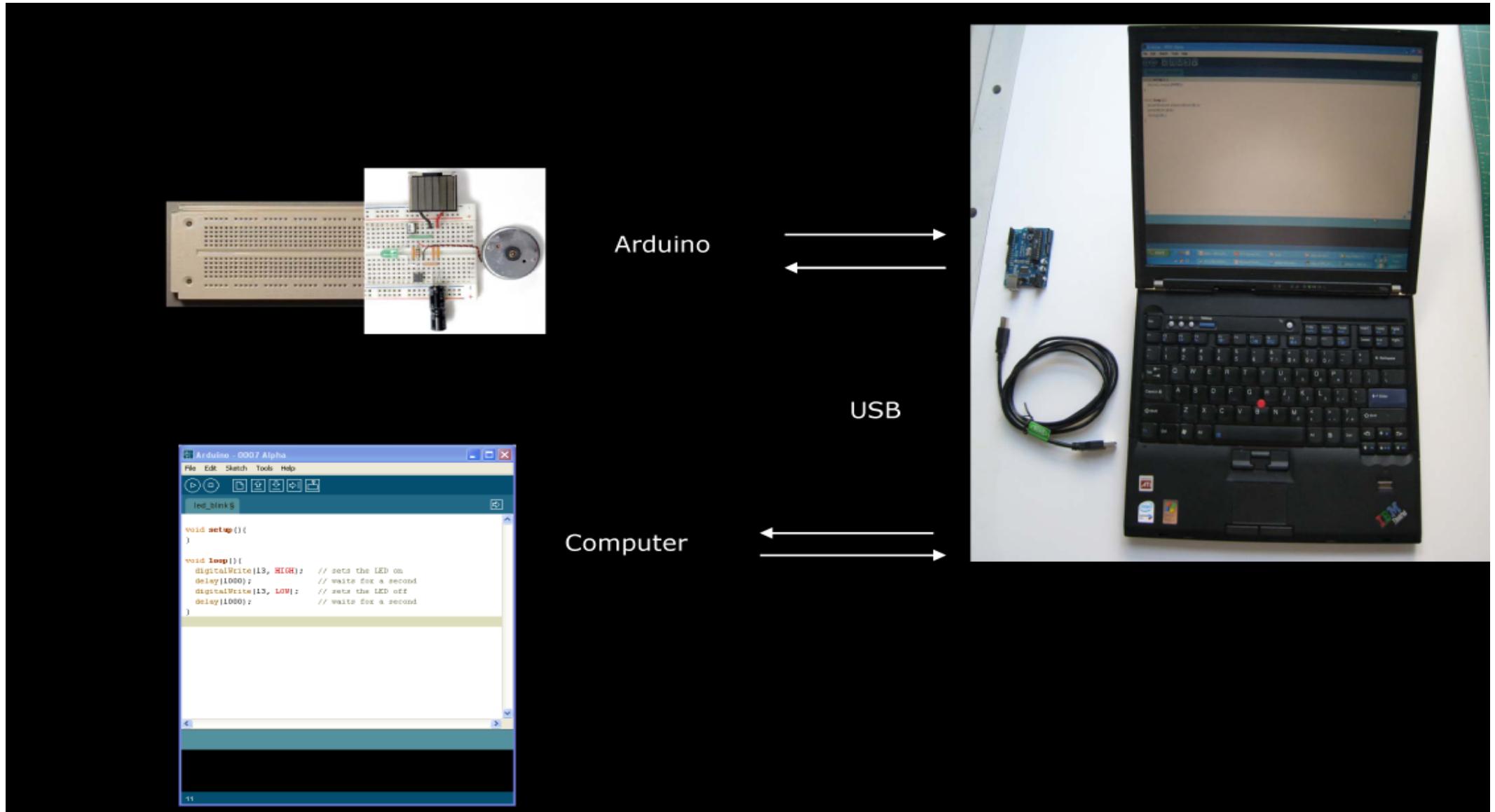
Arduino Programing



Meet Arduino Uno

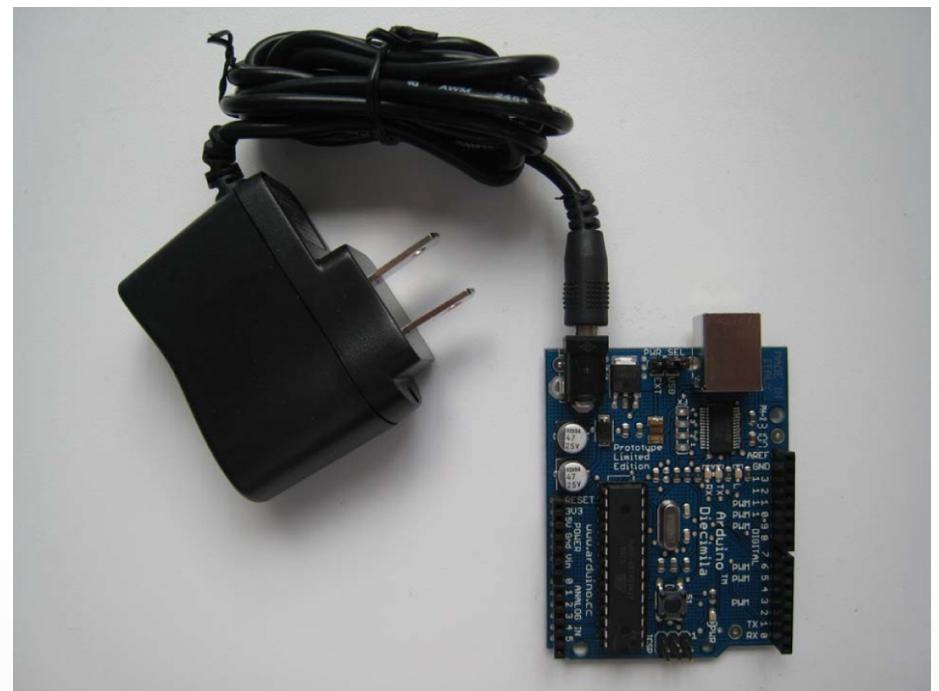
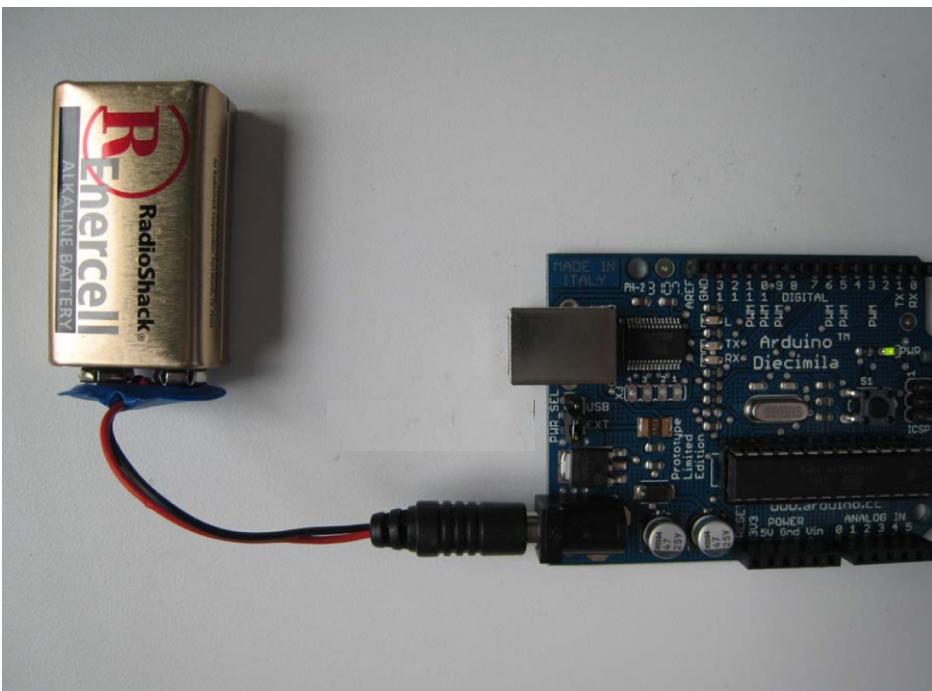


Arduino Working Process





Arduino Power Input



Installation Arduino IDE

Official Hompage : <http://Arduino.cc>

Software -> Download

alternative : <http://iot.kisti.re.kr/Files>

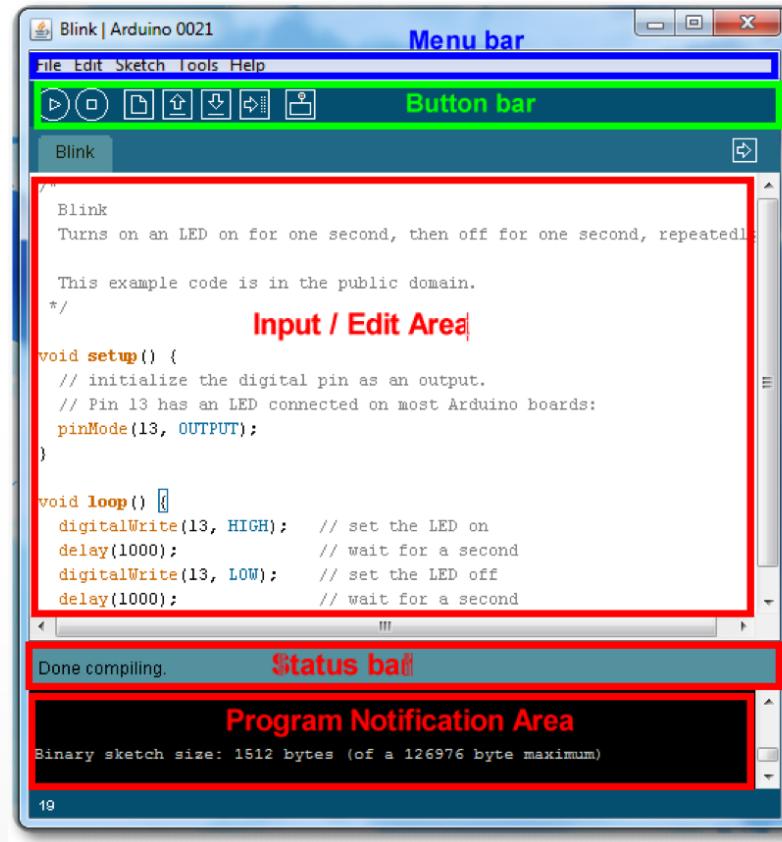


1. Download & install the Arduino environment (IDE)
2. Connect the board to your computer via the USB cable
3. If needed, install the drivers (**not needed in lab**)
4. Launch the Arduino IDE
5. Select your board
6. Select your serial port
7. Open the blink example
8. Upload the program

<https://www.arduino.cc/en/Guide/HomePage>



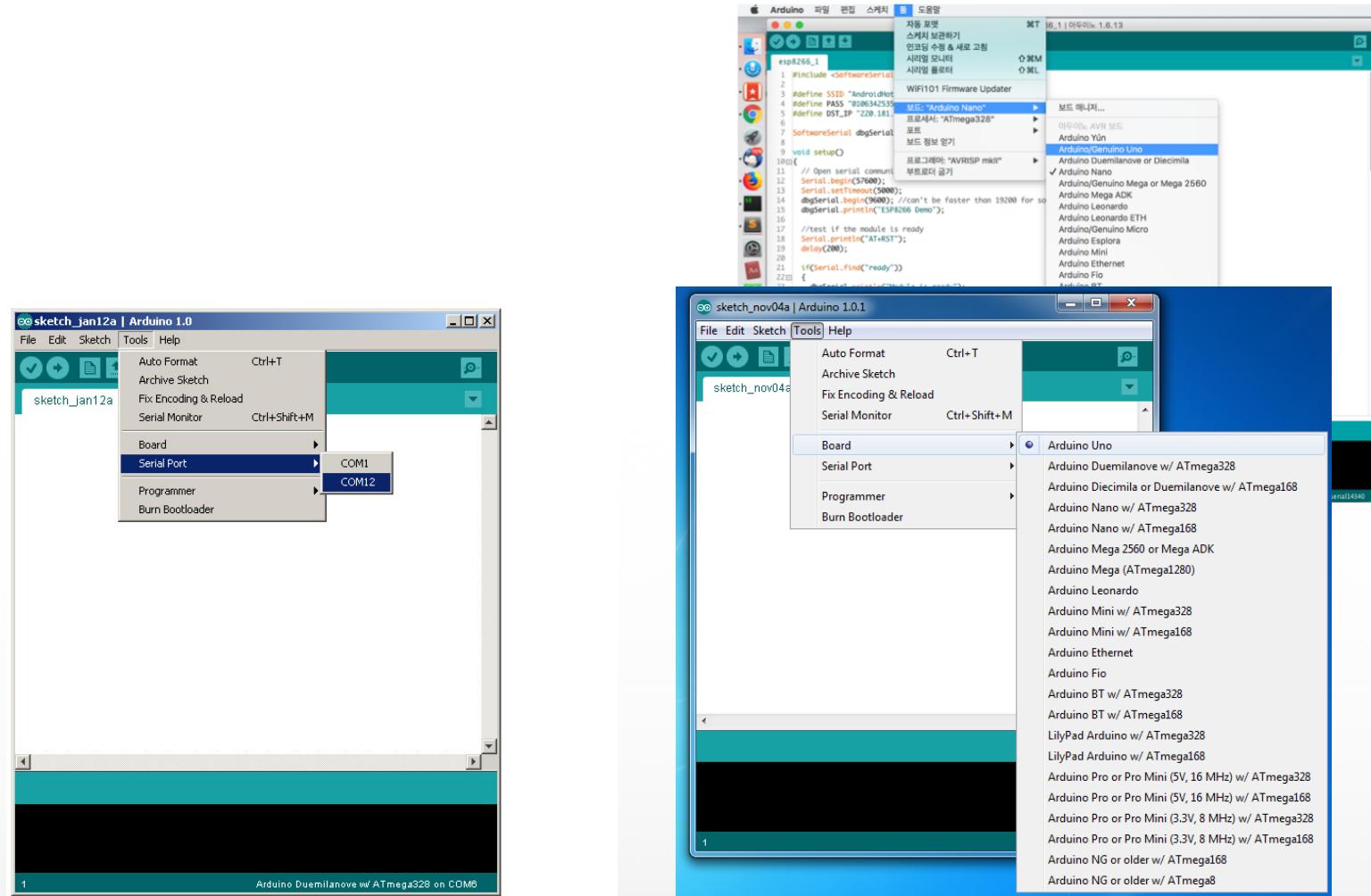
Arduino IDE



See: <http://arduino.cc/en/Guide/Environment> for more information



Select Serial Port and Board



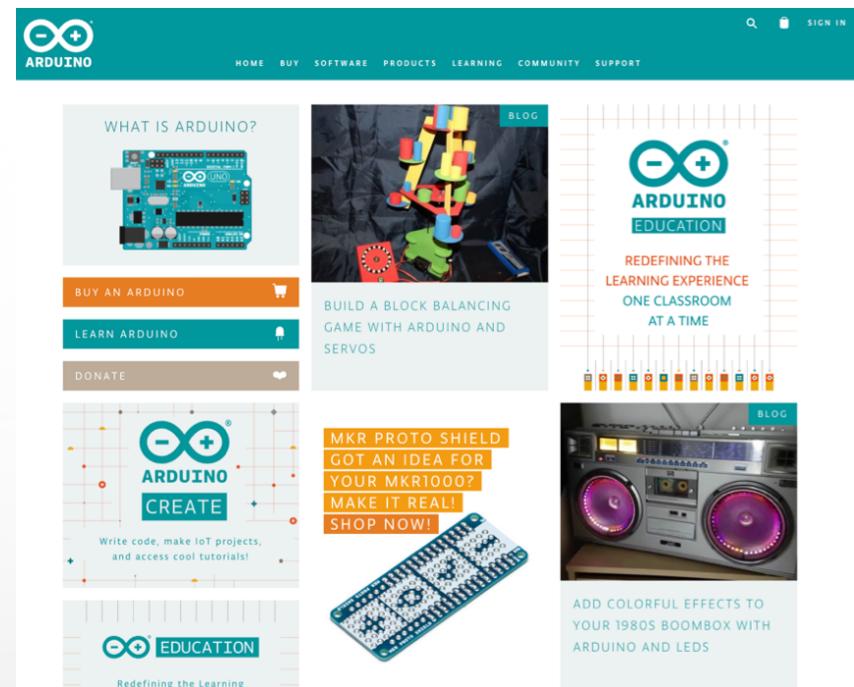
See: <http://arduino.cc/en/Guide/Environment> for more information

Installation Arduino IDE

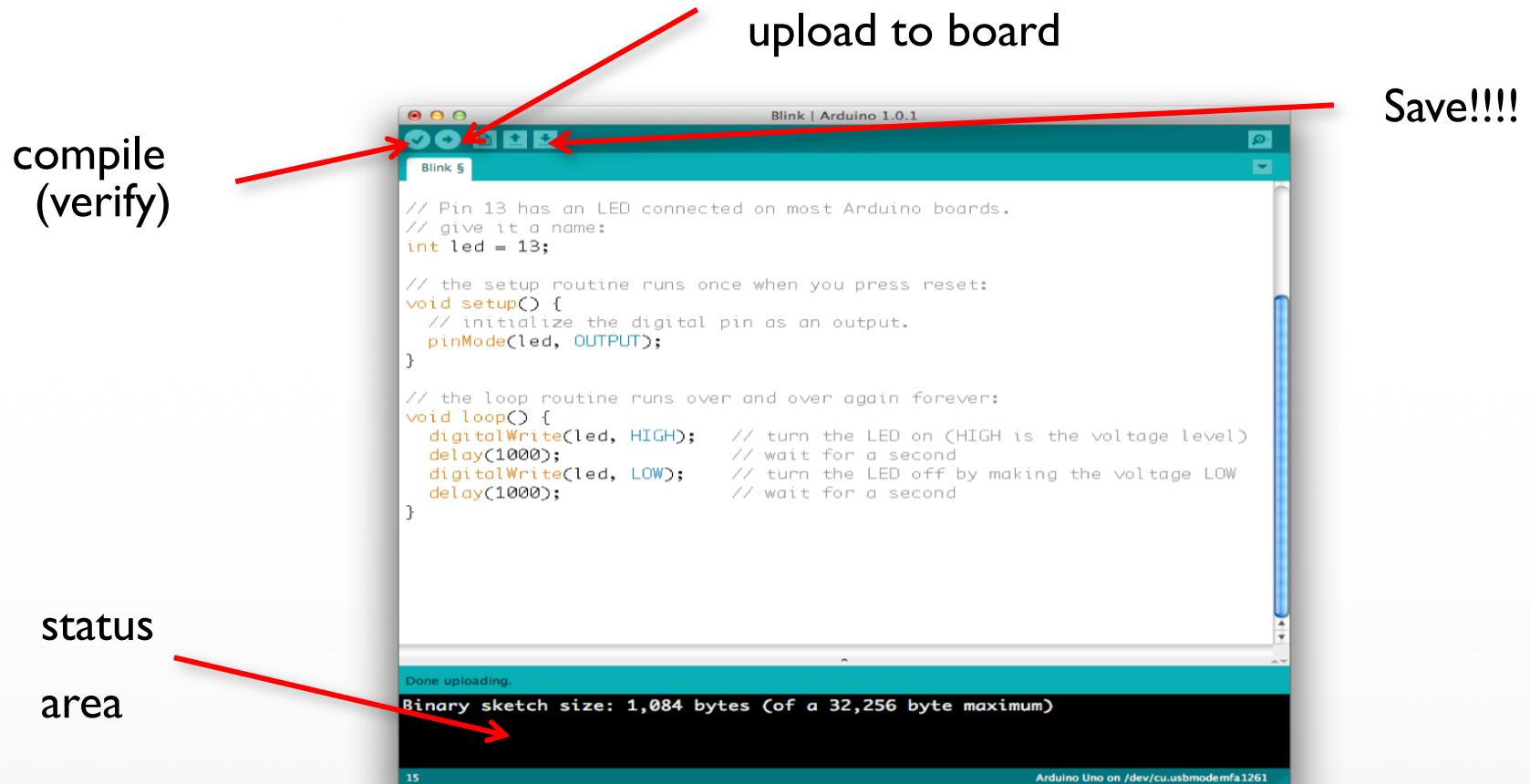
Official Hompage : <http://Arduino.cc>

alternative : <http://iot.kisti.re.kr/Files>

Software -> Download



Installation Arduino IDE





Installation Arduino IDE

Select your Board and Port

- Run your Arduino app and it should display the editor.
Now you need to connect it to the board
- Go to Tools and Select the Board Type – **Arduino/Genuino Uno**
- Go to Tools and Select the Serial Port – **Com##, /dev/cu.usbfmodem##**
should appear when you plug it into your comp.
- Load an example like Blink and make sure it loads
 - When you upload – the LED should start to blink



Installation Arduino IDE

Troubleshooting

- Most common problem is incorrect serial port setting
- If you ever have any “weird” errors from the Arduino environment, just try again.
- The red text at the bottom is debugging output in case there maybe a problem
- Status area shows summary of what’s wrong

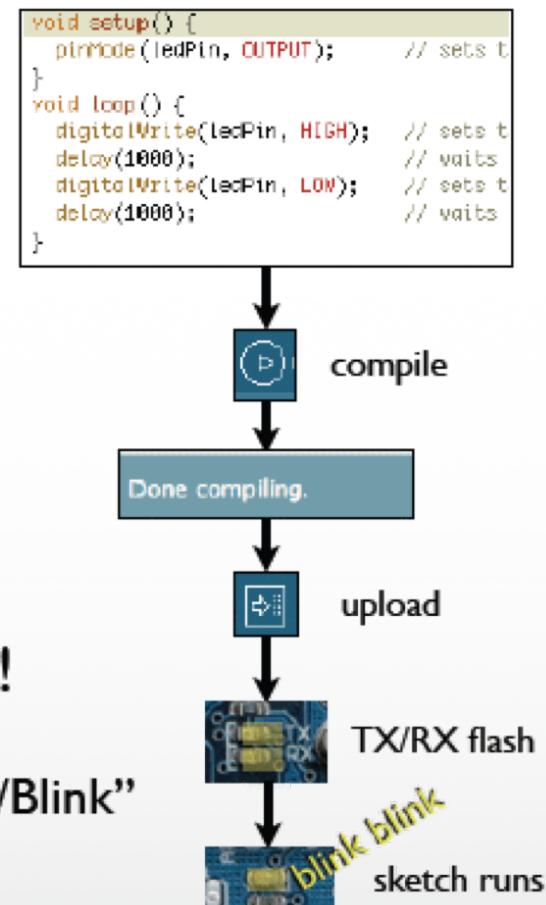
Lets try this

Using Arduino

- Write your sketch
- Press Compile button
(to check for errors)
- Press Upload button to program
Arduino board with your sketch

Try it out with the “Blink” sketch!

Load “File/Sketchbook/Examples/Digital/Blink”

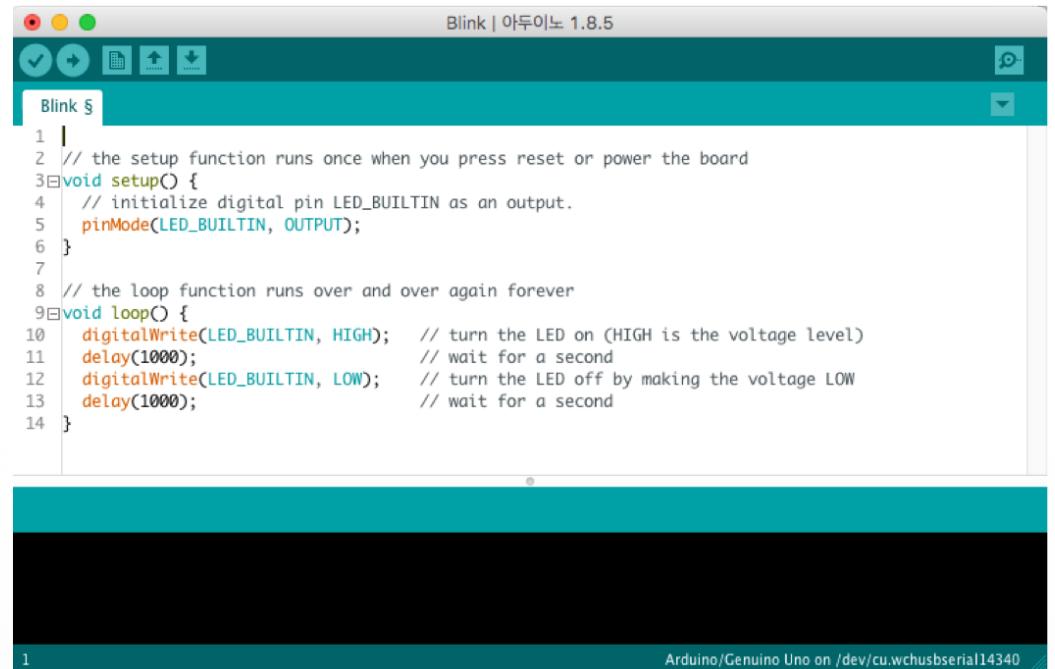




Arduino Code Structure

```
void setup() {  
    // put your setup code here, to run once:  
}
```

```
void loop() {  
    // put your main code here, to run repeatedly:  
}
```



A screenshot of the Arduino IDE interface. The title bar says "Blink | 아두이노 1.8.5". The code editor window shows the "Blink" sketch with the following code:

```
1 //  
2 // the setup function runs once when you press reset or power the board  
3 void setup() {  
4     // initialize digital pin LED_BUILTIN as an output.  
5     pinMode(LED_BUILTIN, OUTPUT);  
6 }  
7  
8 // the loop function runs over and over again forever  
9 void loop() {  
10    digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level)  
11    delay(1000);                      // wait for a second  
12    digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW  
13    delay(1000);                      // wait for a second  
14 }
```

The status bar at the bottom right indicates "Arduino/Genuino Uno on /dev/cu.wchusbserial14340".



Arduino Code Structure

- PinMode
 - A pin on Arduino can be set as input or output by using pinMode function.
 - `pinMode(13, OUTPUT); // sets pin 13 as output pin`
 - `pinMode(13, INPUT); // sets pin 13 as input pin`
- Digital Write
 - `digitalWrite(13, LOW); // Makes the output voltage on pin 13 , 0V`
 - `digitalWrite(13, HIGH); // Makes the output voltage on pin 13 , 5V`
 - `int buttonState = digitalRead(2); // reads the value of pin 2 in buttonState`

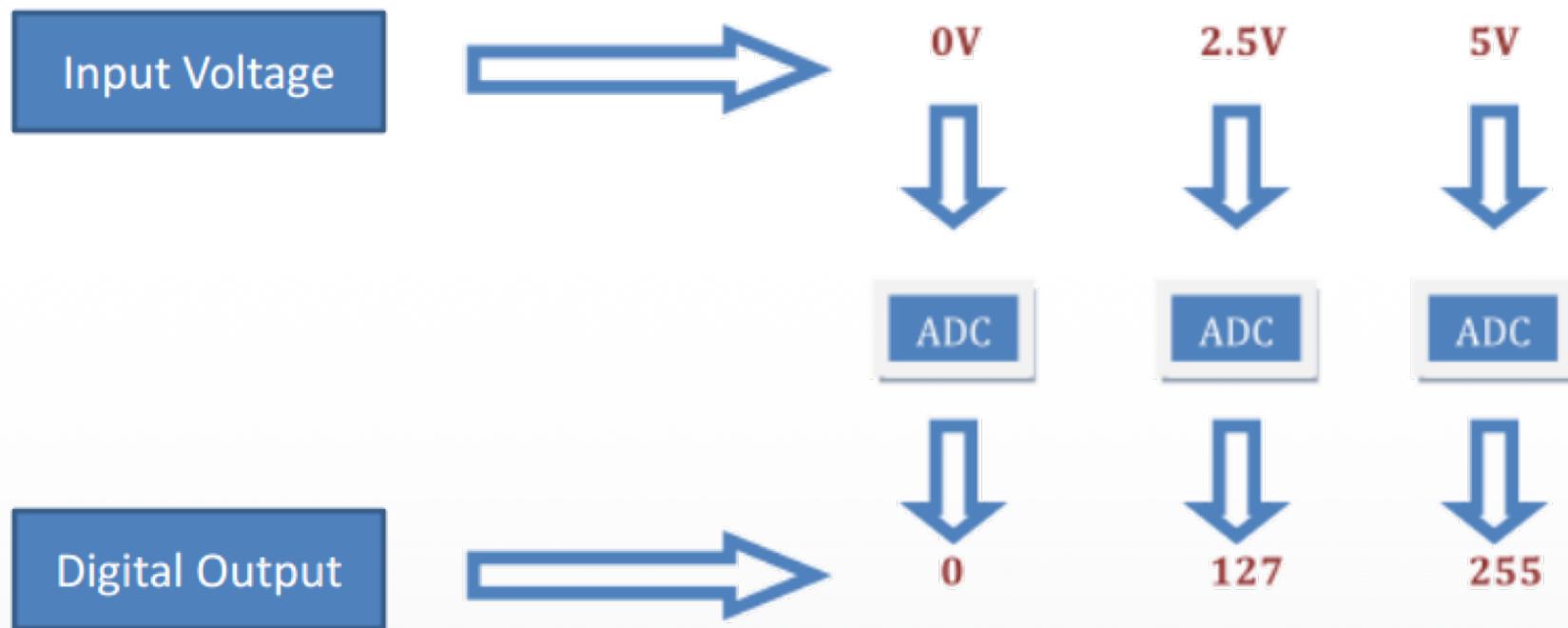


Analog to Digital Conversion

- What is analog ?
 - It is continuous range of voltage values (not just 0 or 5V)
- Why convert to digital ?
 - Because our microcontroller only understands digital.

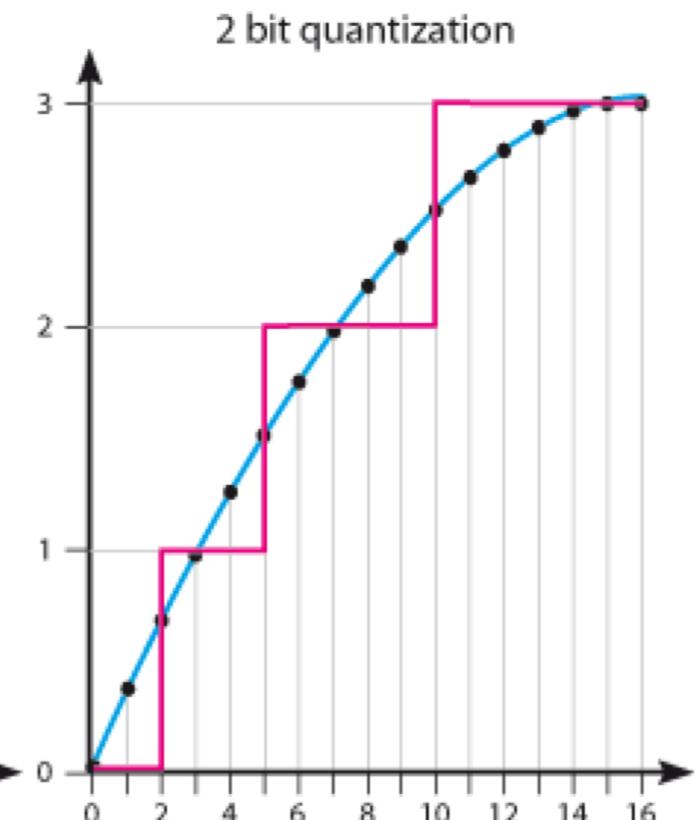
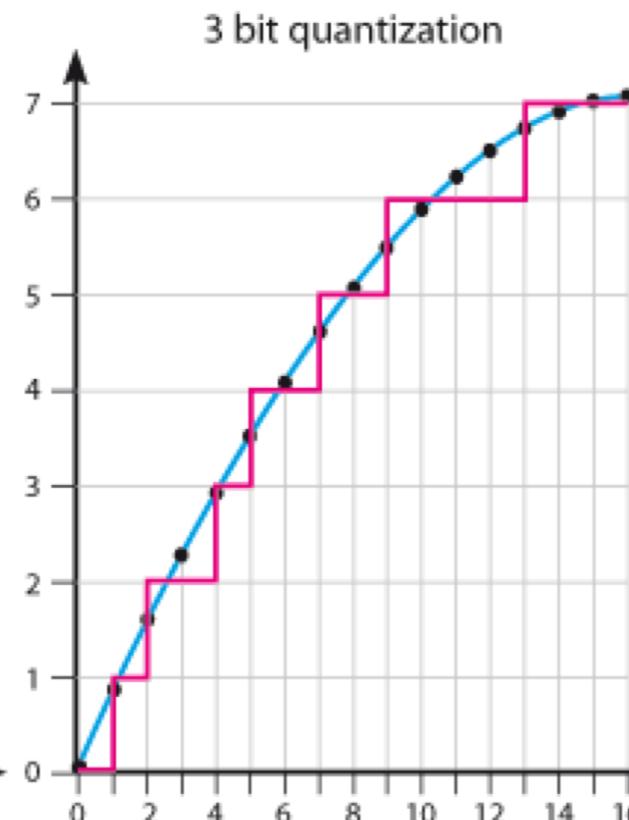
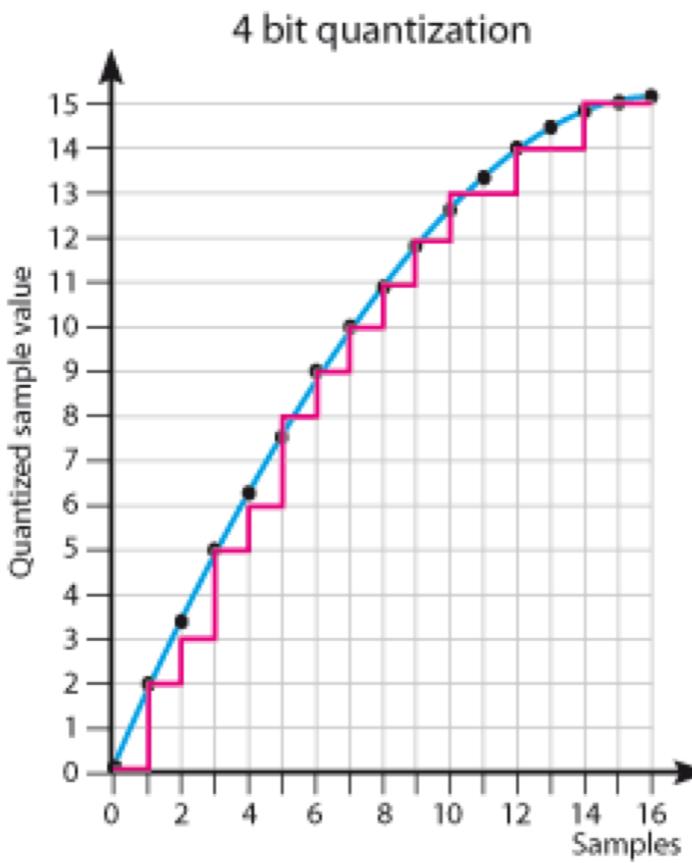


Converting Analog Value to Digital





Quantization the signal





ADC in Arduino

- The Arduino Uno board contains 6 pins for ADC
- 10-bit analog to digital converter
- This means that it will map input voltages between 0 and 5 volts into integer values between **0 and 1023**



Reading/Writing Analog Values

- `analogRead(A0);`
// used to read the analog value from the pin A0
- `analogWrite(2,128);`

Break Time

www.ust.ac.kr

