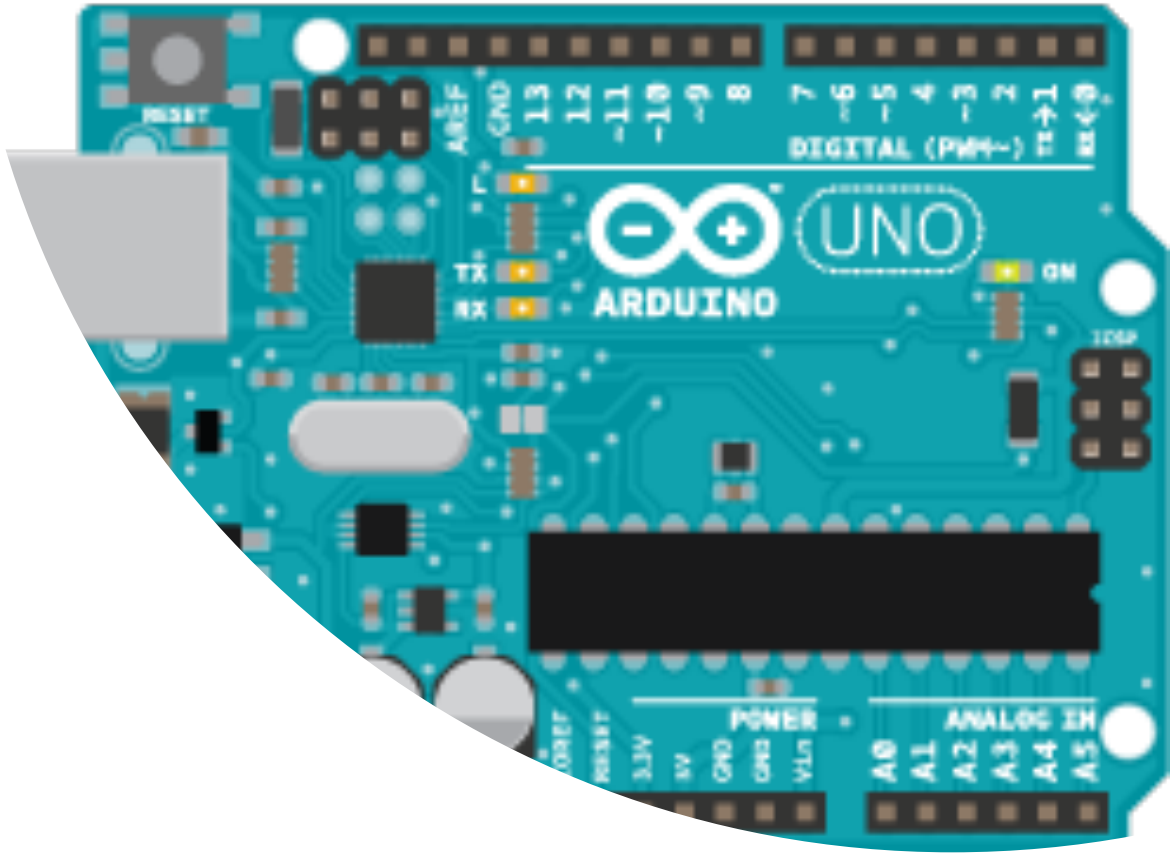
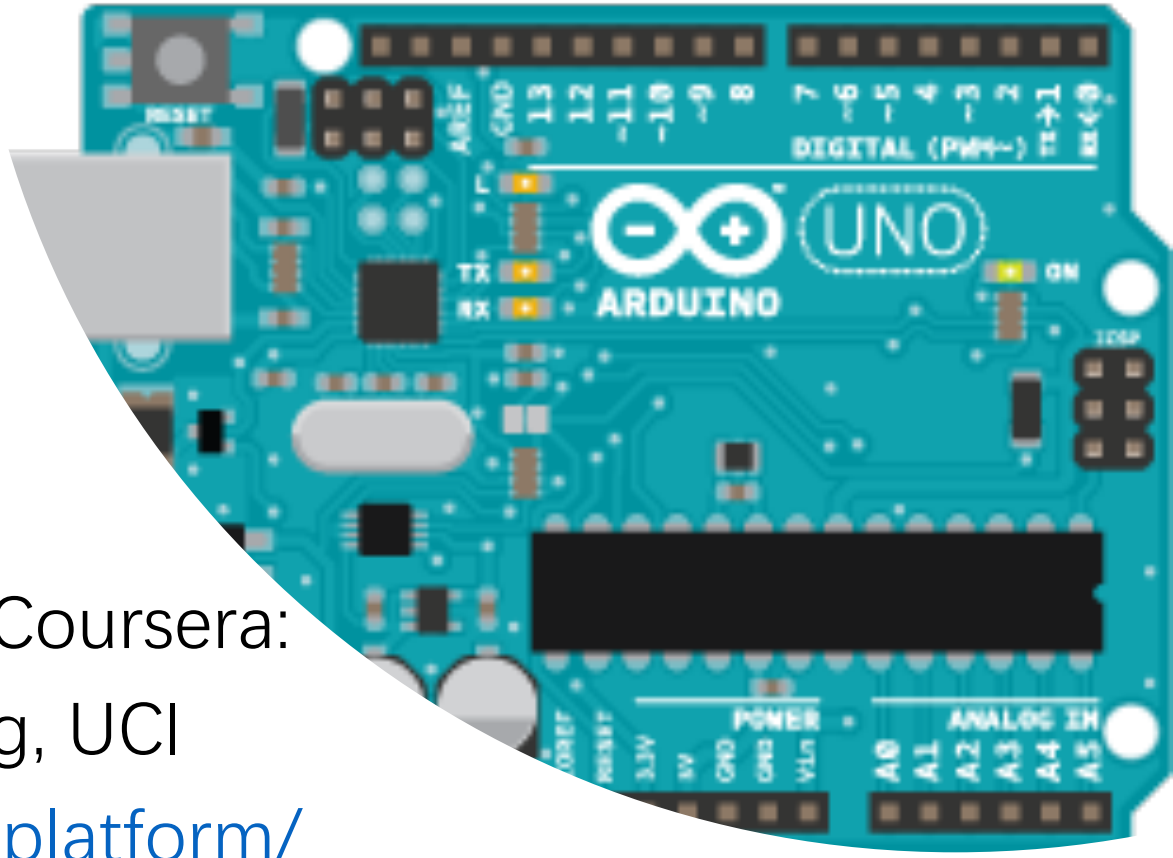


Introduction to Arduino

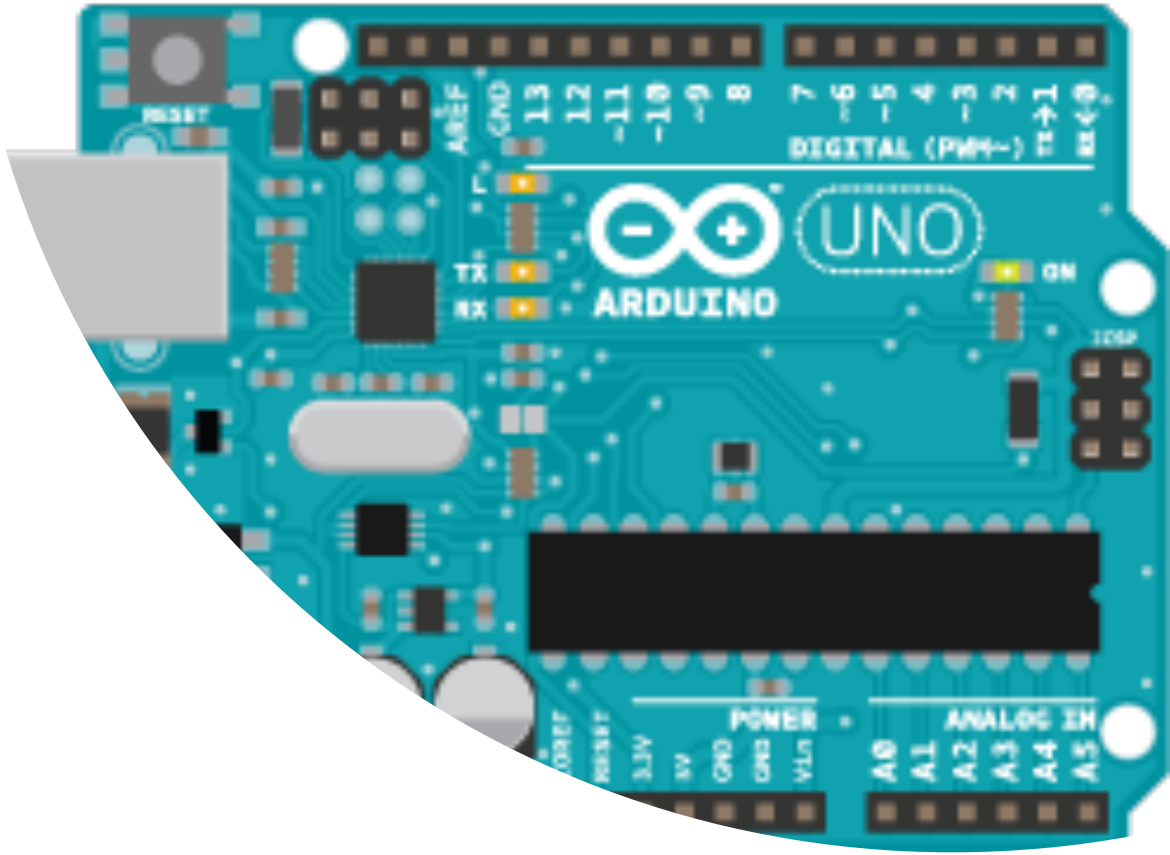
ZHU Binwu
LIU Hongduo

Reference

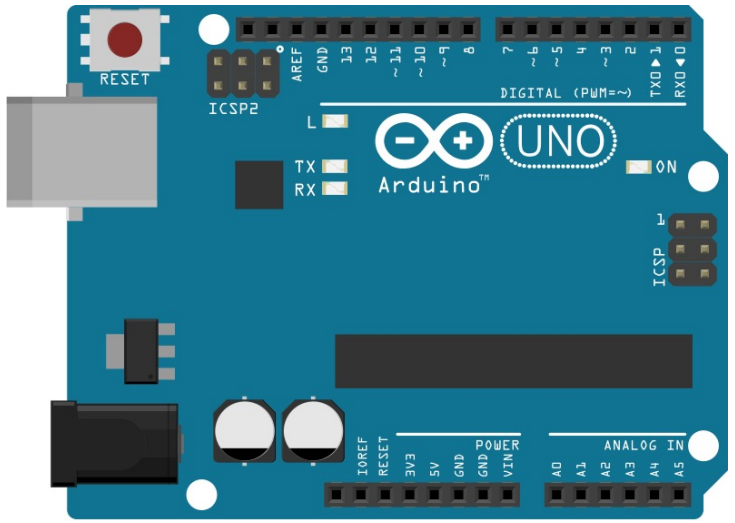


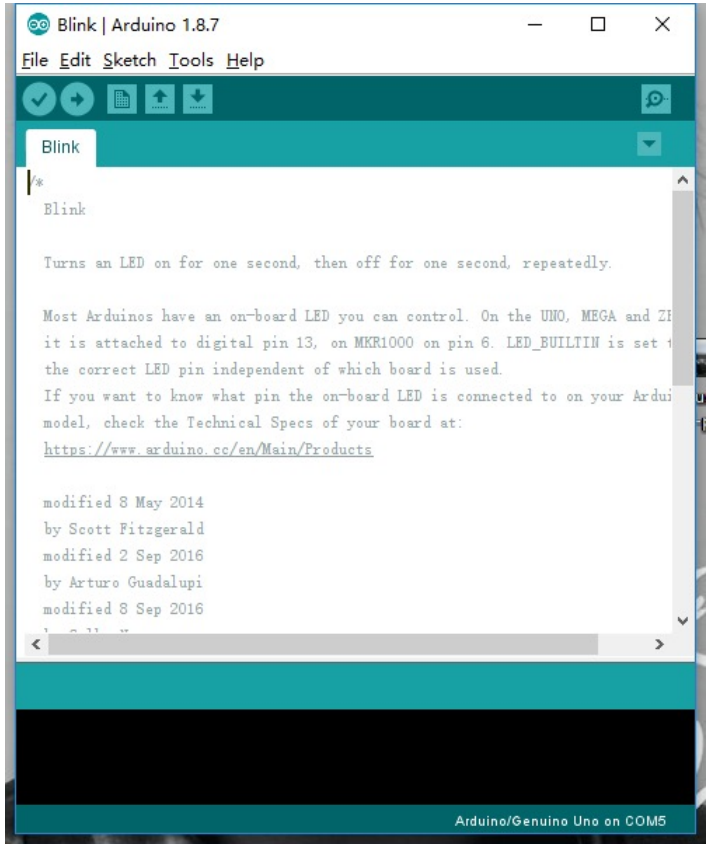
- Credit to Module 1 and Module 3 from Coursera: The Arduino Platform and C programming, UCI https://www.coursera.org/learn/arduino-platform/
- You can access many tutorials and examples from: https://www.arduino.cc/

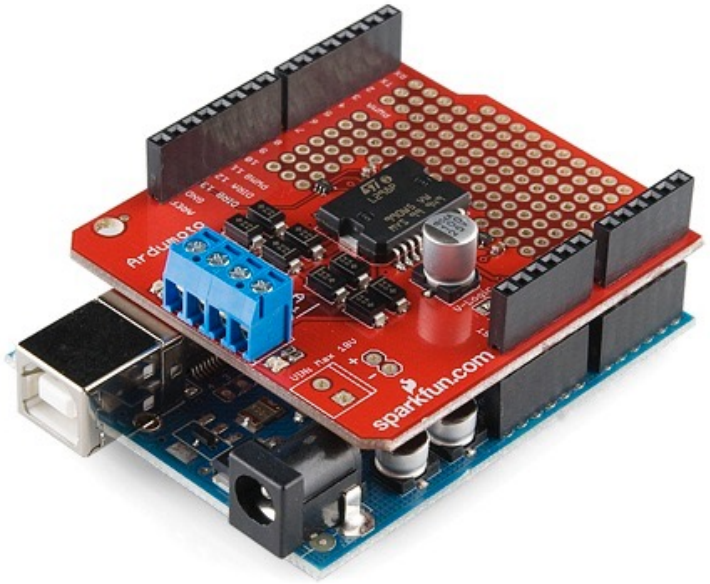
- 3 Components
- Workflow
- Arduino Programming
- I/O Pins
- Examples



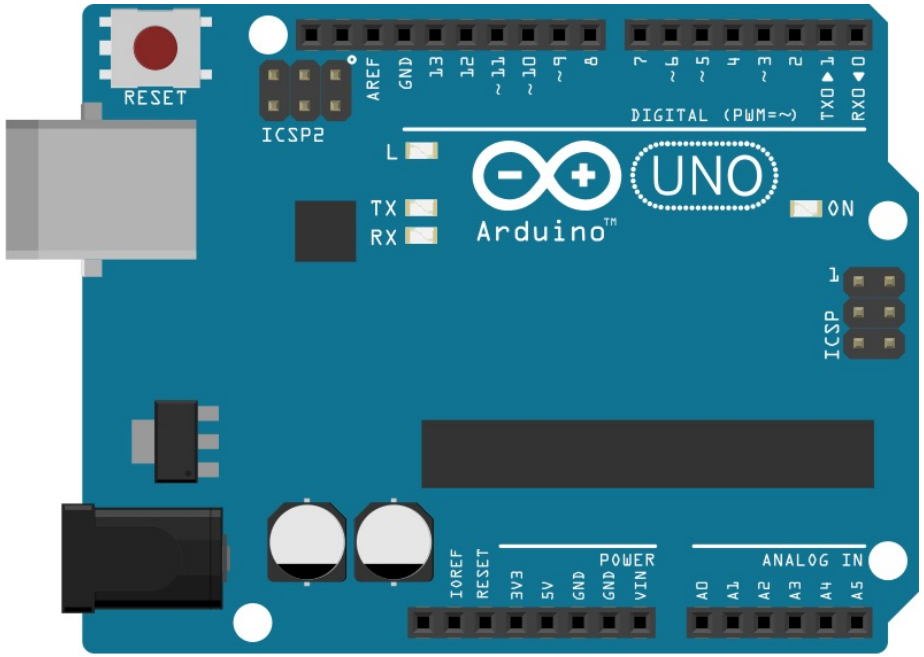
Outline







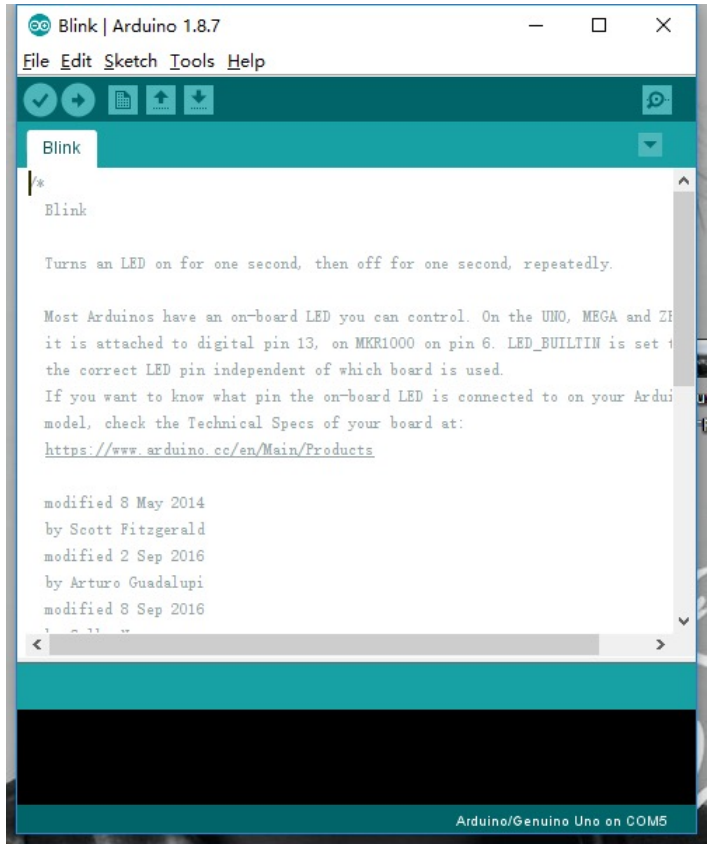
3 components



The Arduino Development Board

A development board

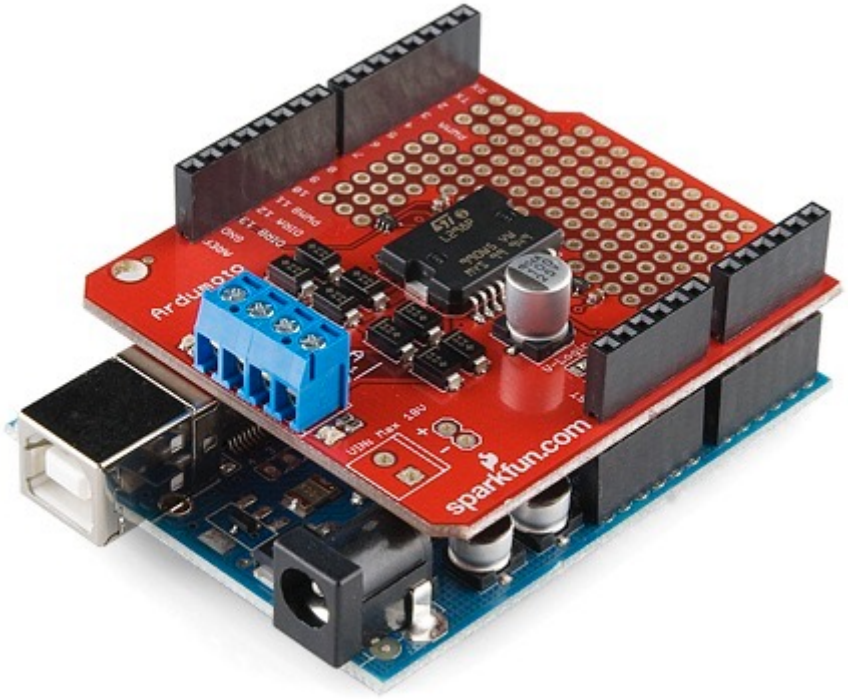
- 8-bit microcontroller
- Programming hardware
- USB programming interface
- I/O pins



The Arduino IDE

A software environment

- Cross-complier
- Debugger
- Simulator
- …



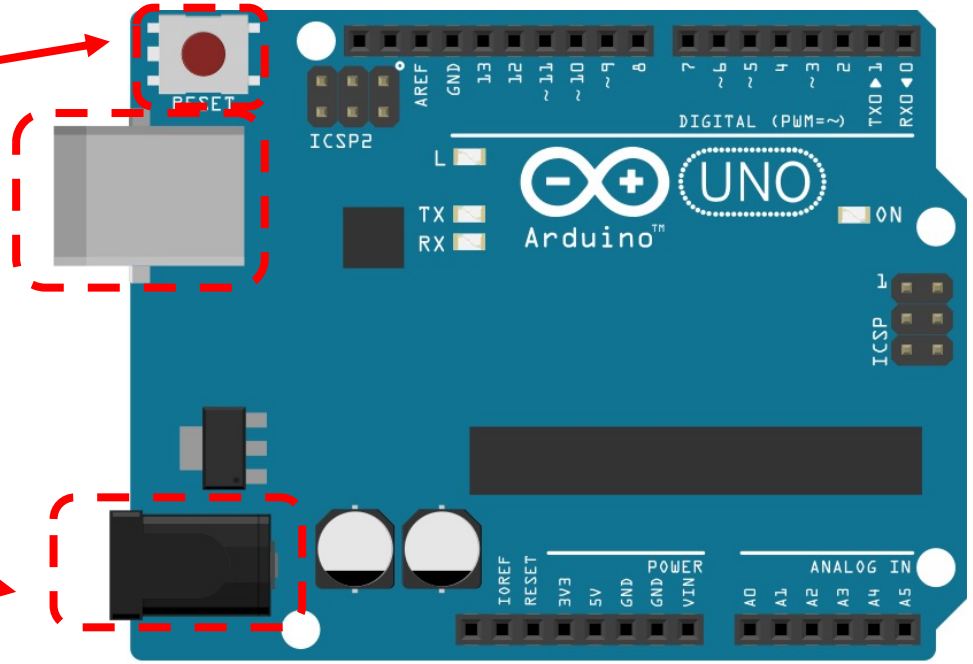
The Arduino Shields

Special-purpose “Shields”
- Daughter boards
- Unique functionalities
- Easy to attach
- Good libraries provided

Reset button

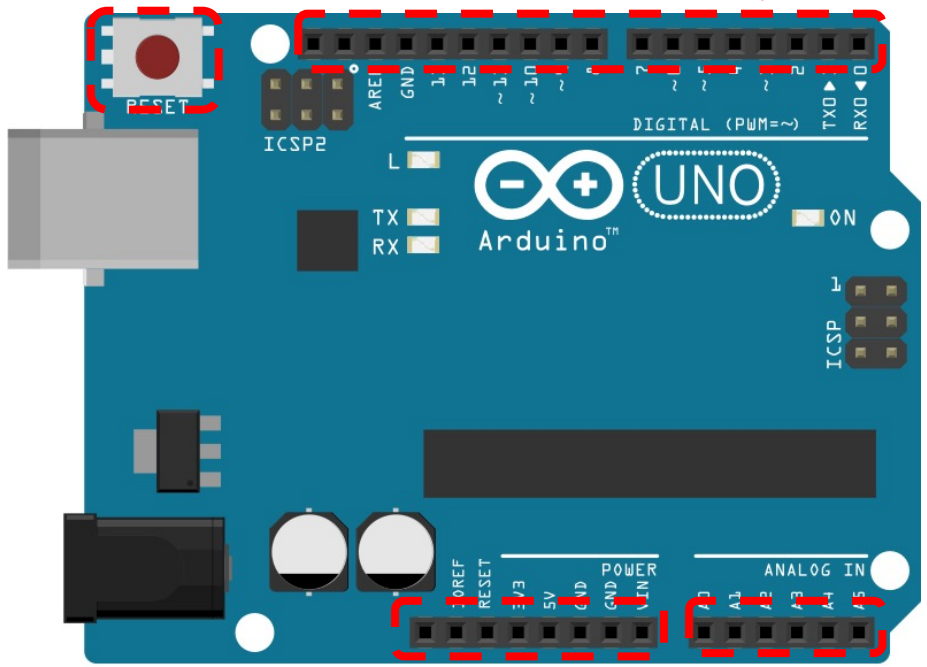
USB connector

Power connector

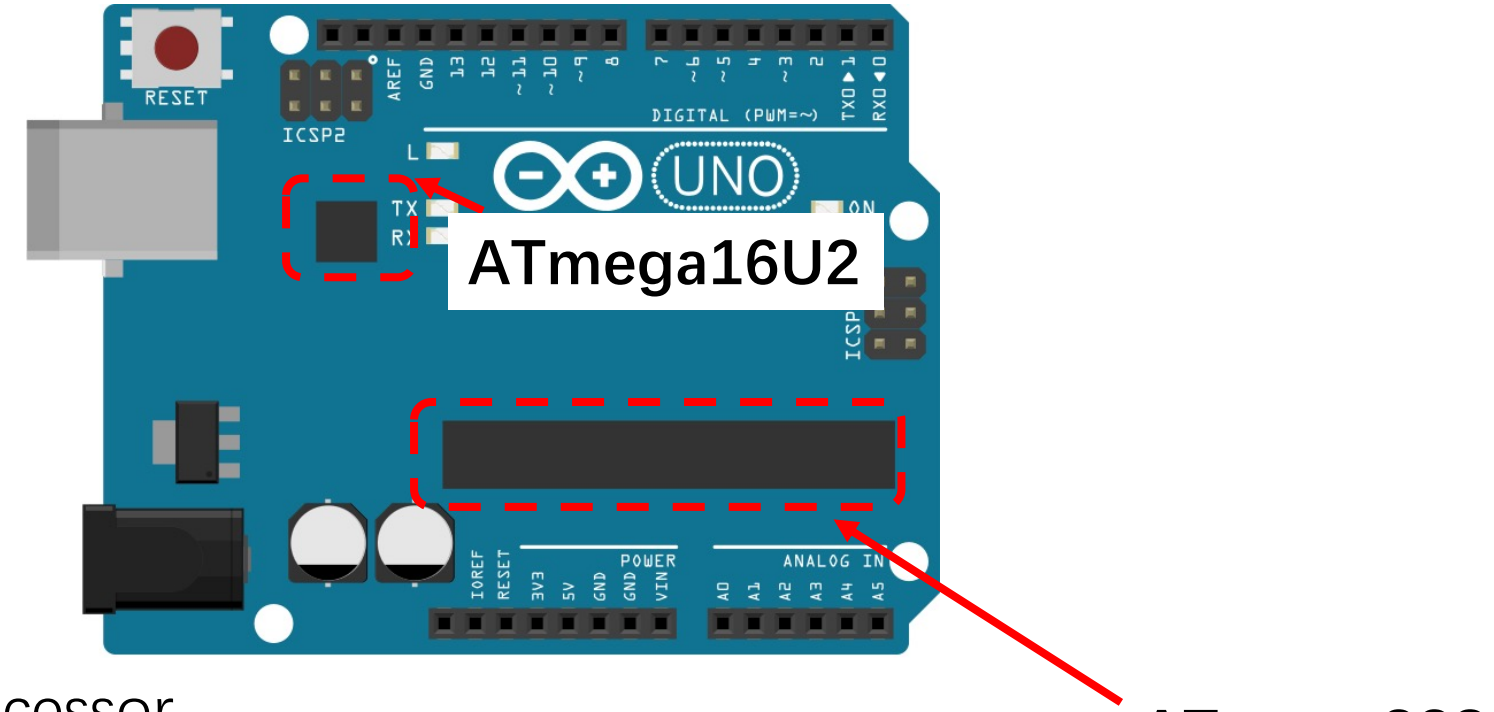


The Arduino Development Board

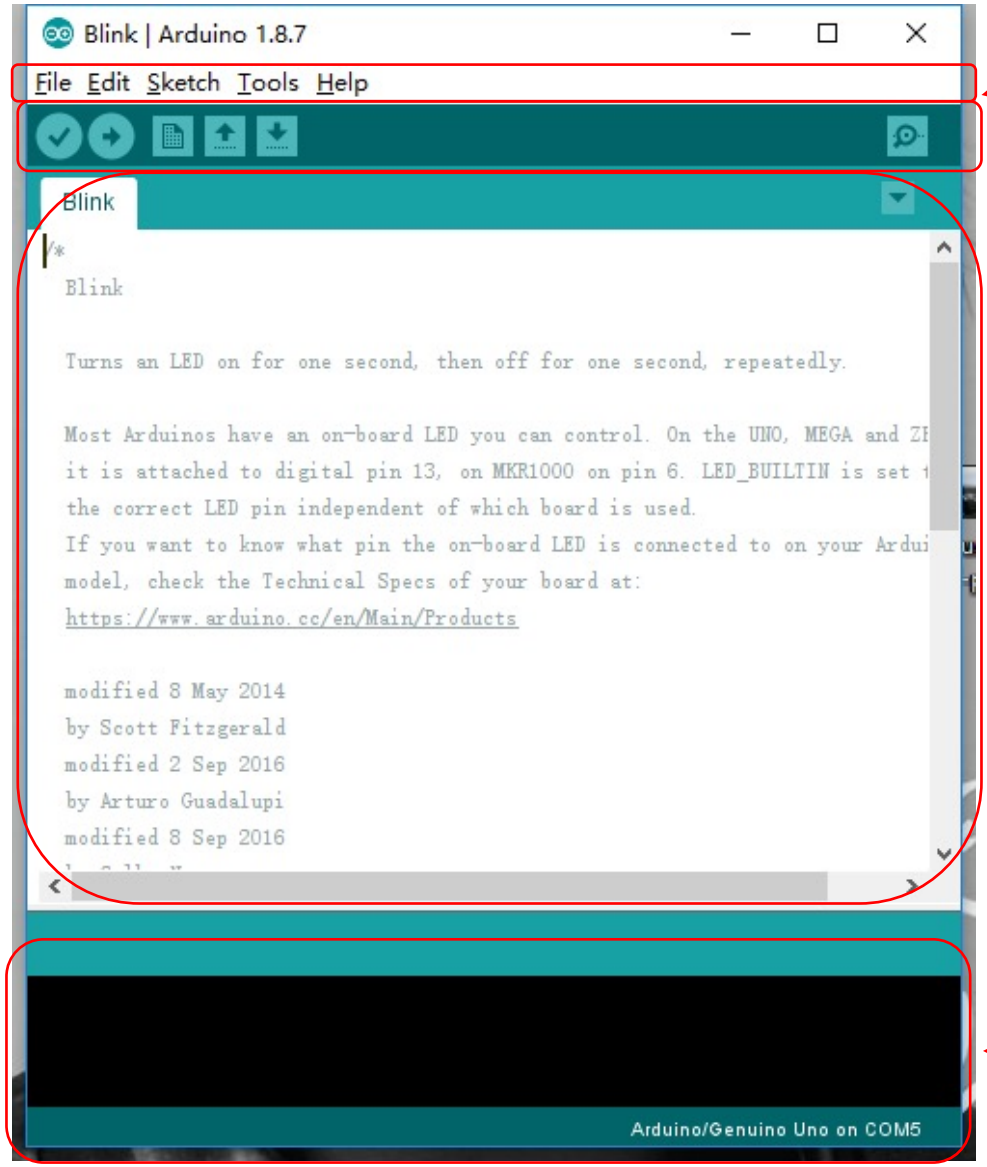
Input / Output Pins



Microcontrollers



- ATmega328 is the processor programmed by user
- ATmega16U2 handles USB communication



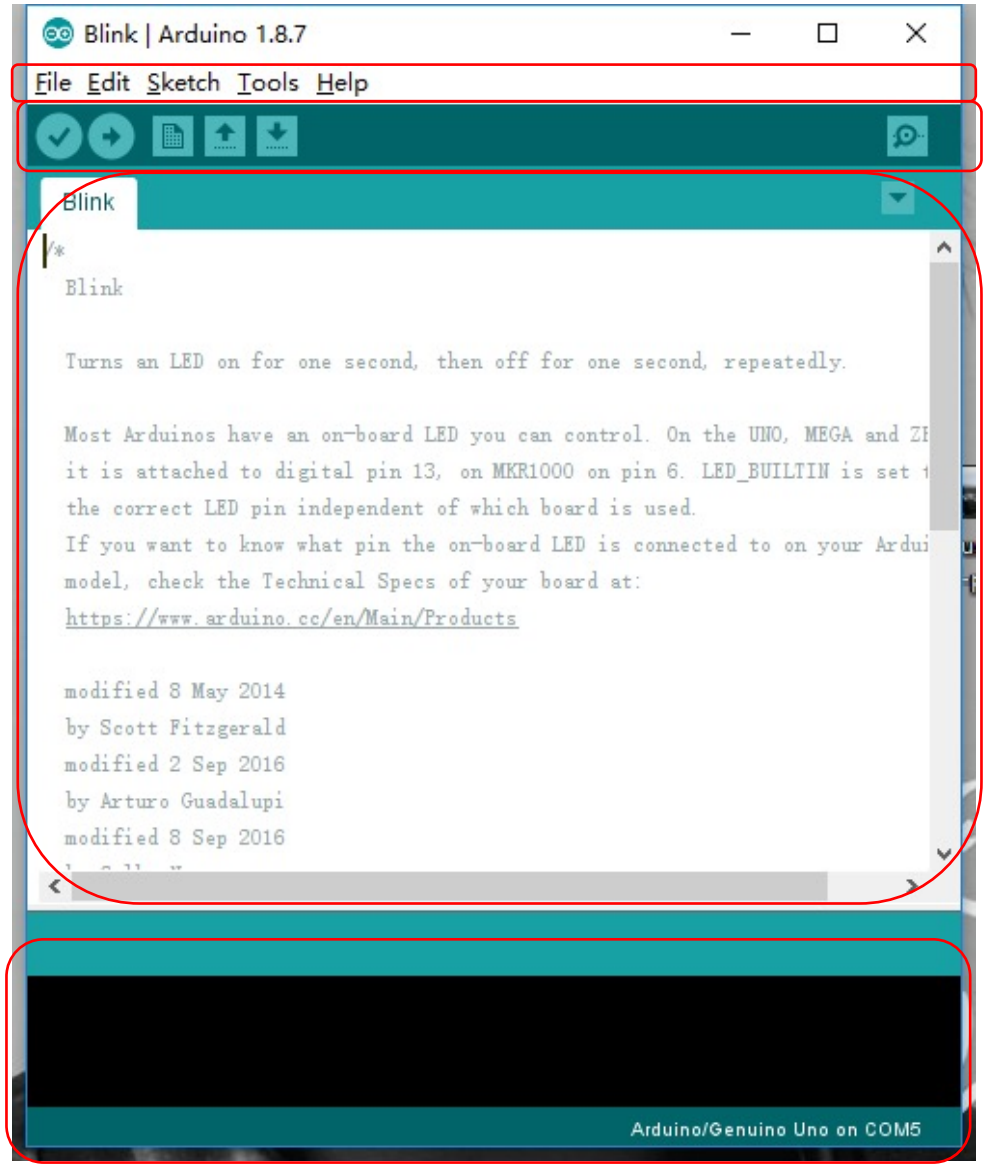
Menus with all commands

Buttons for common commands

Text editor for writing code

Message area

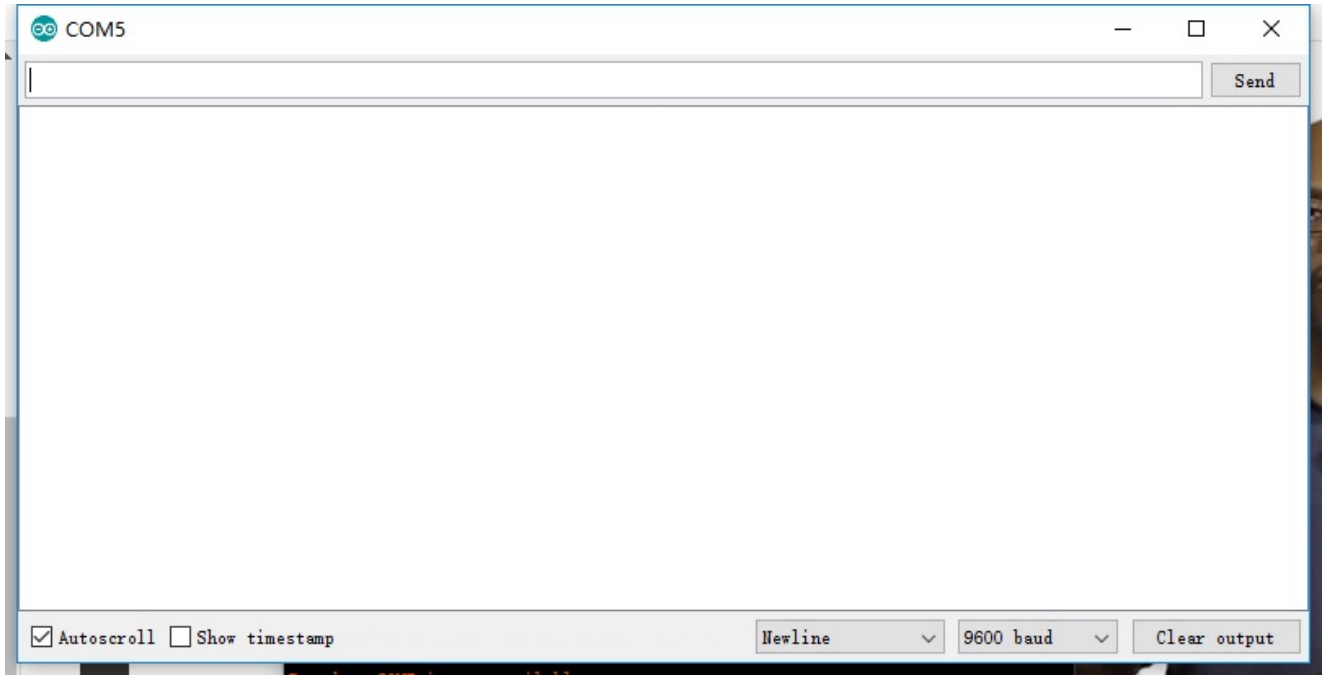
The Arduino IDE



The Arduino IDE

- Verify: Compile codes, checks for errors
- Upload: Compile codes, checks for errors, uploads to board
- New: Creates a new sketch
- Open: Opens an existing sketch
- Save: Saves your sketch to a file
- Serial Monitor: Opens a windows to communicate with the board

Serial Monitor

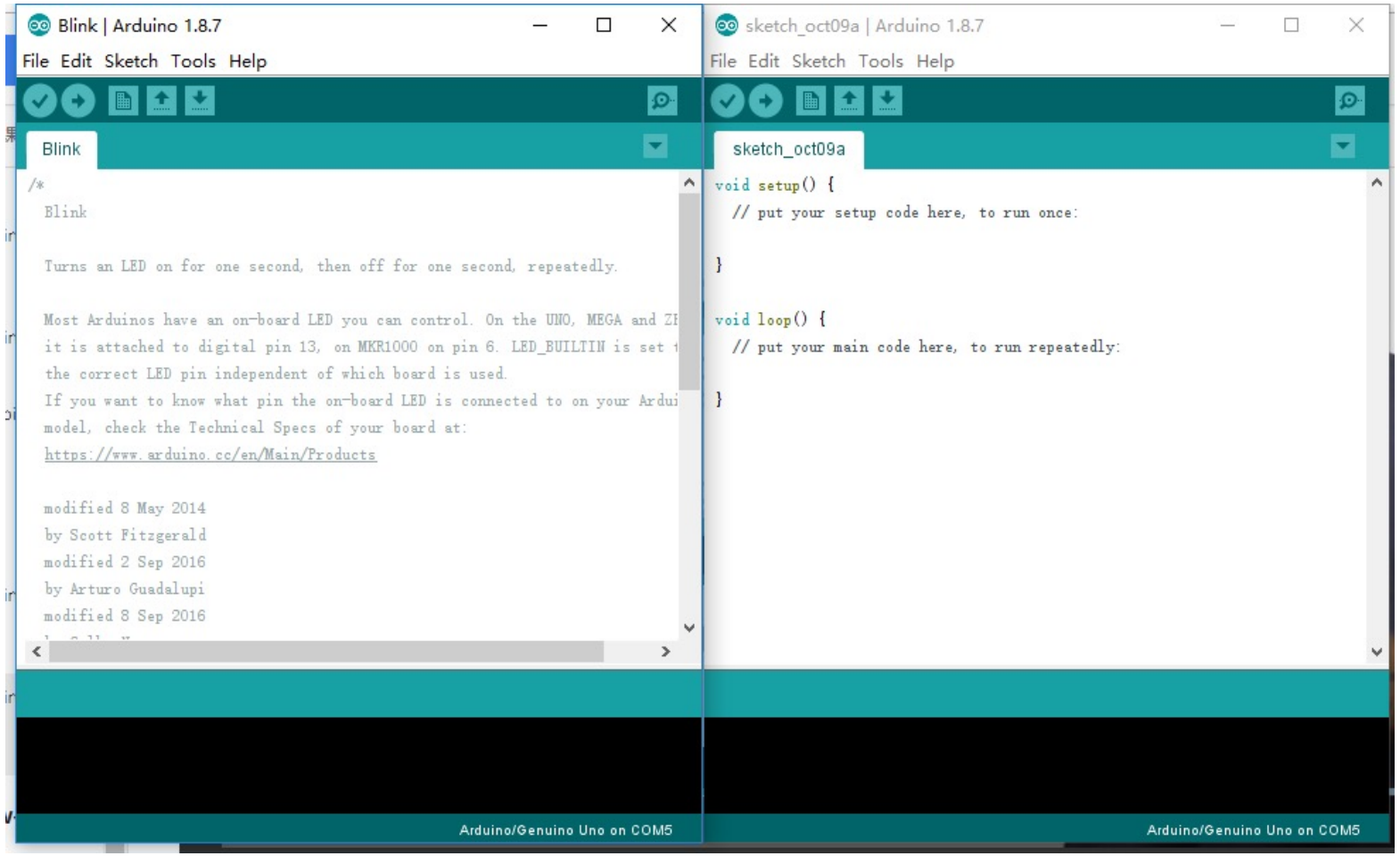


- Displays serial data sent from the Arduino
- Allows serial data to be sent to the Arduino from the keyboard
- Library functions in the serial library

Basic Setup

1. Download the IDE www.arduino.cc/en/Main/Software
   - Easiest to run Windows Installer
   - Also installs USB and other drivers
2. Connect the board to your computer
   - Use USB cable
3. Launch the Arduino application
   - Start the IDE

Launch the Arduino IDE



4. Open the Blink example: File > Example > Basic > Blink

Run a Program

5. Select your Arduino in the tools > Board menu
6. Select your serial port in the Tools > Port menu
   - There should be only one selection (COM3, etc)
7. Upload the program with the upload button
   - This writes the program onto the Flash of the Arduino
8. The LED with sign “L” should blink

Arduino Programs

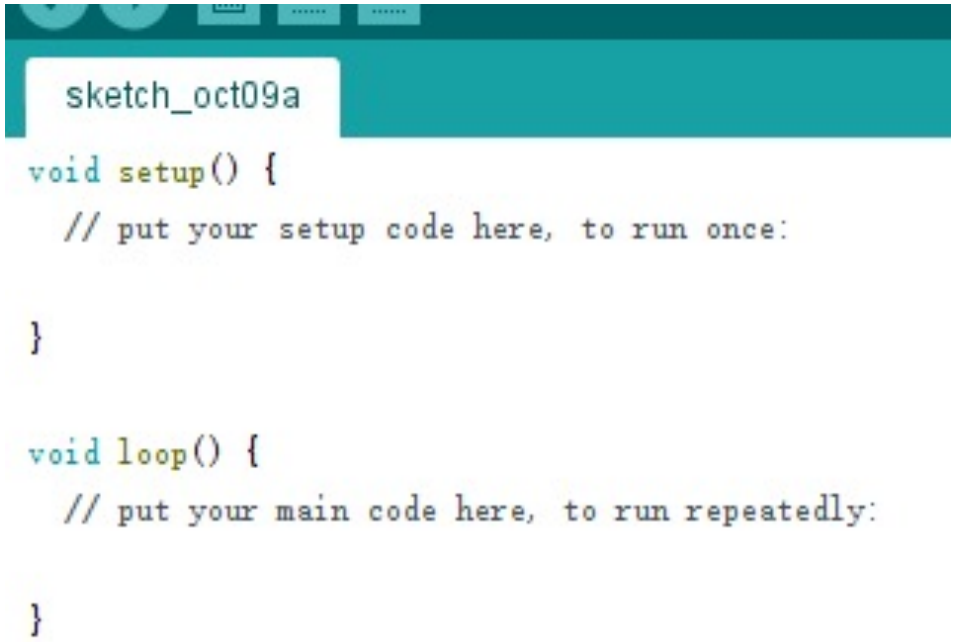
- A program is called a Sketch
- C++ program using Arduino library functions
  - Actually almost C
  - You should be familiar with Classes in libraries

```
Ethernet.begin(mac);
Serial.begin(speed);
client.print(  “Hello”  );
Serial.print(  “Hello”  );
```

Setup() Function

- A sketch does not have a main() func
- Every sketch has a setup() function
  - Executed once when Arduino is powered up
  - Used for initialization operations
  - Return no value, takes no arguments

```
Void setup() {
    …
}
```



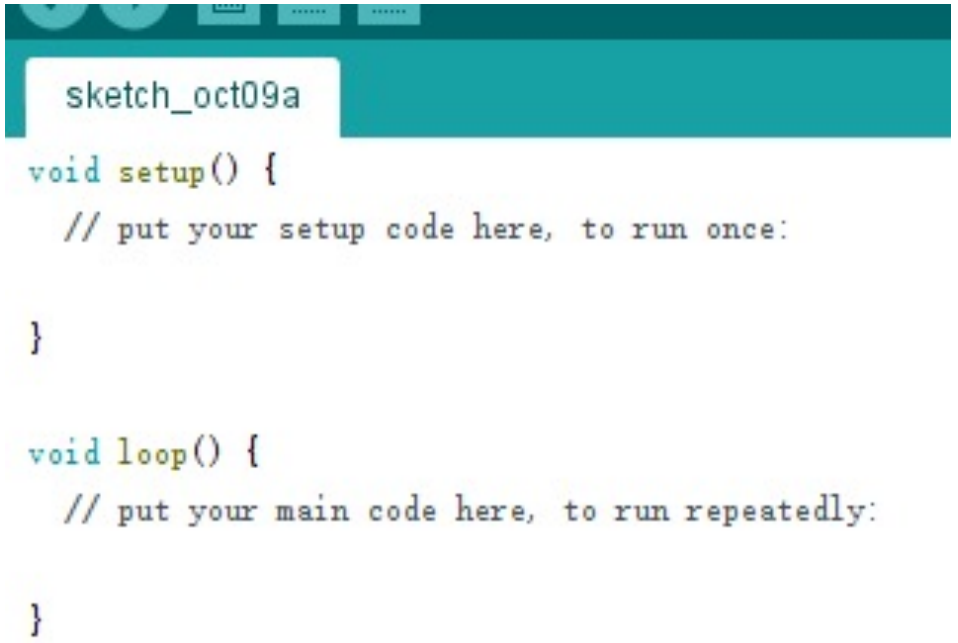
Loop() Function

- Every sketch has a loop() function
  - Executed iteratively as long as the Arduino is powered up
  - loop() starts executing after setup() has finished
  - loop() is the main program control flow
  - Return no value, takes no arguments

Void loop() {

    …

}



Input / Output (I/O)

- These functions allow access to the pins

Void pinMode(pin, mode)

- Set a pin to act as either an input or an output
- pin is the number of pin
  - 0 – 13 for the digital pins
  - A0-A5 for the analog pins
- mode is the I/O mode the pin is to set
  - INPUT, OUTPUT, or INPUT_PULLUP
  - INPUT_PULLUP acts as input with reversed polarity

Digital Input

int digitalRead(pin)

- Returns the state of an input pin
- Returns either LOW (0 volts) or HIGH (5 volts)

Example:

```
int pinval;
pinval = digitalRead(3);
```

- pinval is set to the state of digital pin 3

Digital Output

int digitalWrite(pin, value)

- Assigns the state of an output pin
- Assigns either LOW (0 volts) or HIGH (5 volts)

Example:

```
digitalWrite(3, HIGH);
```

- Digital pin3 is set HIGH (5 volts)

Analog Input

int analogRead(pin)

- Returns the state of an analog input pin
- Returns the integer from 0 to 1023
- 0 for 0 volts, 1023 for 5 volts

Example:

```
int pinval;
pinval = analogRead(A3);
```

- Pin must be an analog pin

Example

- Blink example

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

Delay

void delay(msec)

- Pauses the program for msec milliseconds
- Useful for human interaction
- Example:

digitalWrite(3, HIGH);

delay(1000);

digitalWrite(3, LOW);

- Pin 3 is HIGH for 1 second

Example

- CharacterAnlysis example

setup() function:
Serial.begin(9600);
//set the bit rate for serial port.

loop() function:

Serial.available(): is True if some inputs.
Serial.read(): read the data byte.

```
void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // send an intro:
  Serial.println("send any byte and I'll tell you everything I can about it");
  Serial.println();
}

void loop() {
  // get any incoming bytes:
  if (Serial.available() > 0) {
    int thisChar = Serial.read();

    // say what was sent:
    Serial.print("You sent me: \'");
    Serial.write(thisChar);
    Serial.print("\'  ASCII Value: ");
    Serial.println(thisChar);

    // analyze what was sent:
```

Lab to do today

- 1st: let LED blink 2 times in per second.
- 2nd: write your name to Arduino through serial port, if the name is strictly correct, then the Arduino will return your Student ID.

Requirement

- No lab report.
- You can take a short video to demo your experiments.
- You are required to upload source code AND video to Blackboard.
- In the 1st experiment, you should show the LED blink-blink during at least 5 seconds in the video.
- In the 2nd experiment, you should show the name you input and your student ID returned by Arduino in the video.

Tips

- Use Tool -> Serial Monitor to input your name and display your student ID.
- Use the syntax to read your name:

String thisString = Serial.readStringUntil(‘\n’);