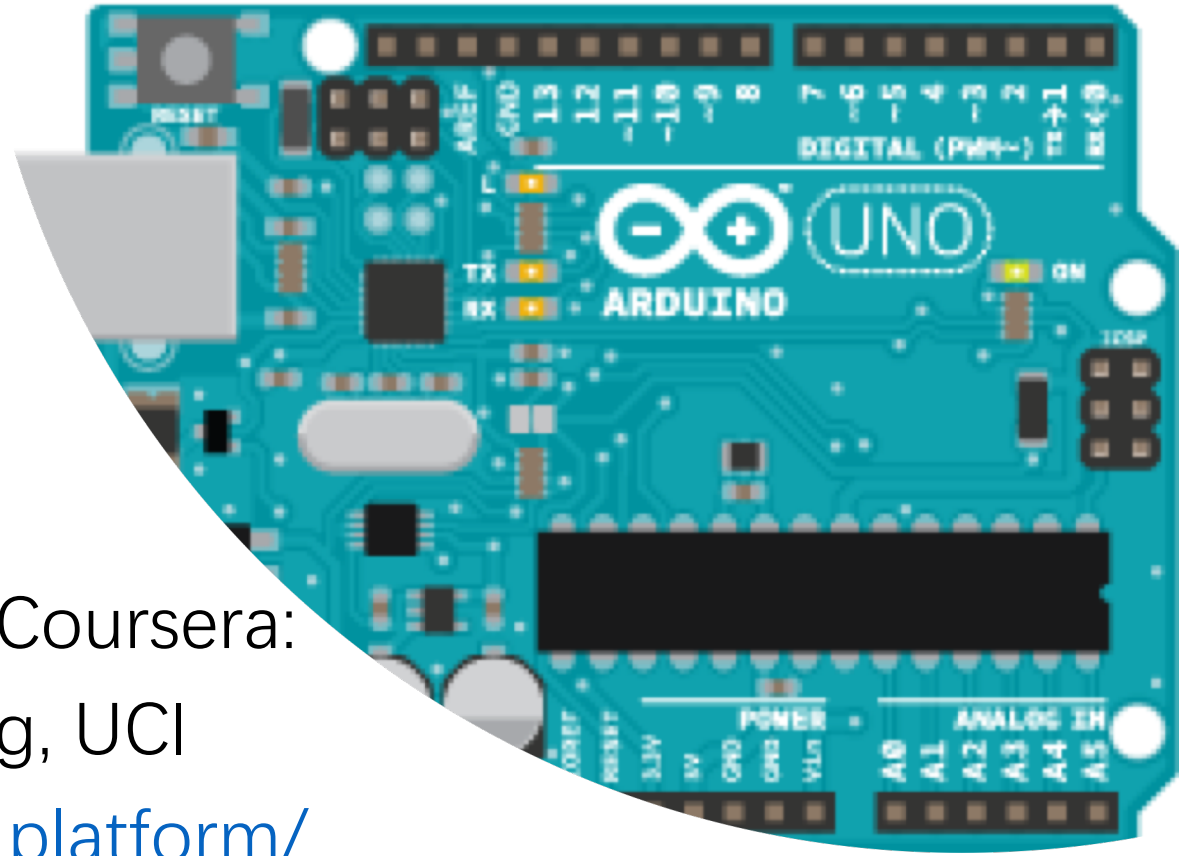


Introduction to Arduino

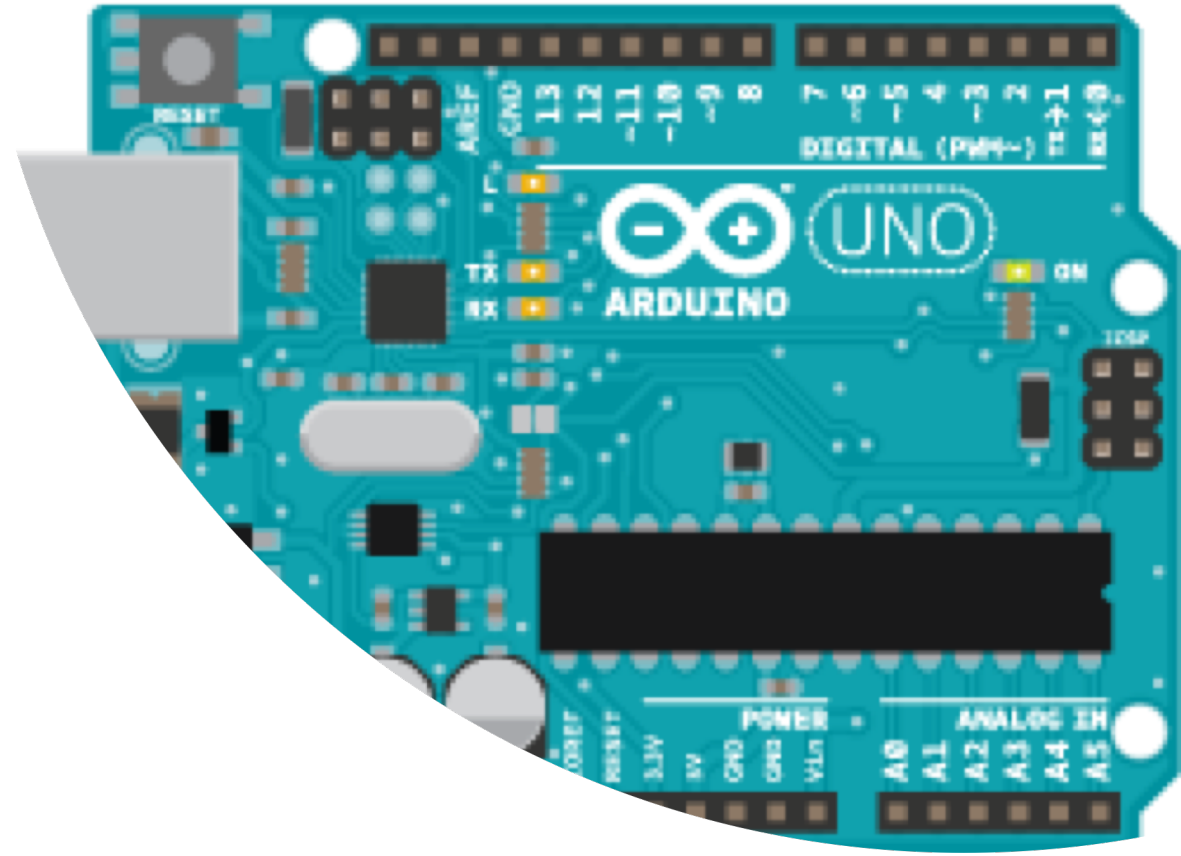
CHEN Tinghuan
GENG Hao

Reference

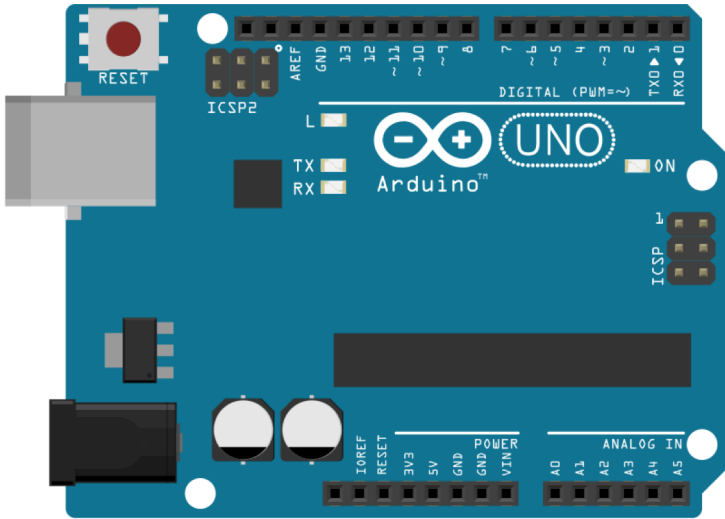
- Credit to Module 1 and Module 3 from Coursera:
The Arduino Platform and C programming, UCI
<https://www.coursera.org/learn/arduino-platform/>
- You can access many tutorials and examples from:
<https://www.arduino.cc/>



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



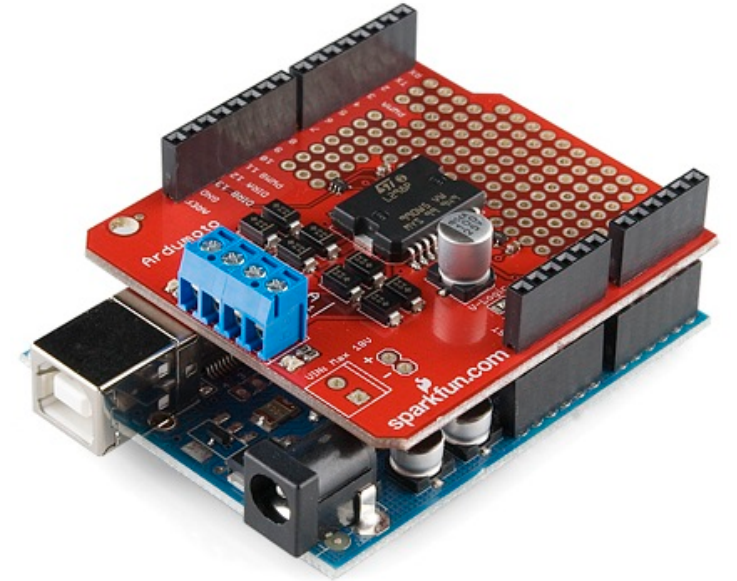
Outline



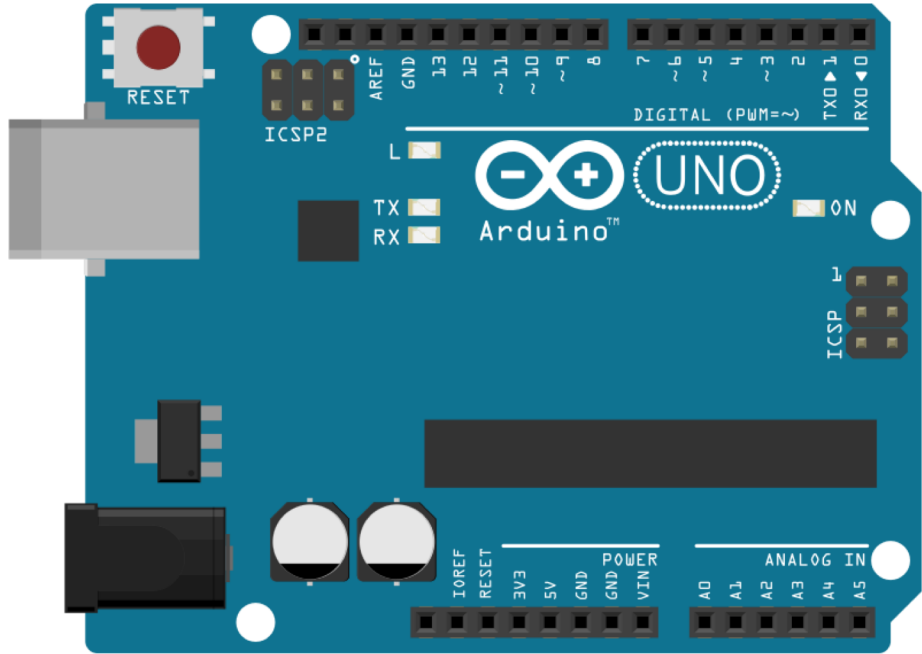
```
Arduino IDE window: Blink | Arduino 1.8.7
File Edit Sketch Tools Help
Blink
/*
 * Blink
 *
 * Turns an LED on for one second, then off for one second, repeatedly.
 *
 * Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
 * it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to
 * the correct LED pin independent of which board is used.
 * If you want to know what pin the on-board LED is connected to on your Arduino
 * model, check the Technical Specs of your board at:
 * https://www.arduino.cc/en/Main/Products
 *
 * modified 8 May 2014
 * by Scott Fitzgerald
 * modified 2 Sep 2016
 * by Arturo Guadalupi
 * modified 8 Sep 2016
 * by Dan Falsella
 */
#define LED_BUILTIN 13

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the positive lead)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the pin LOW
  delay(1000);                       // wait for a second
}
```

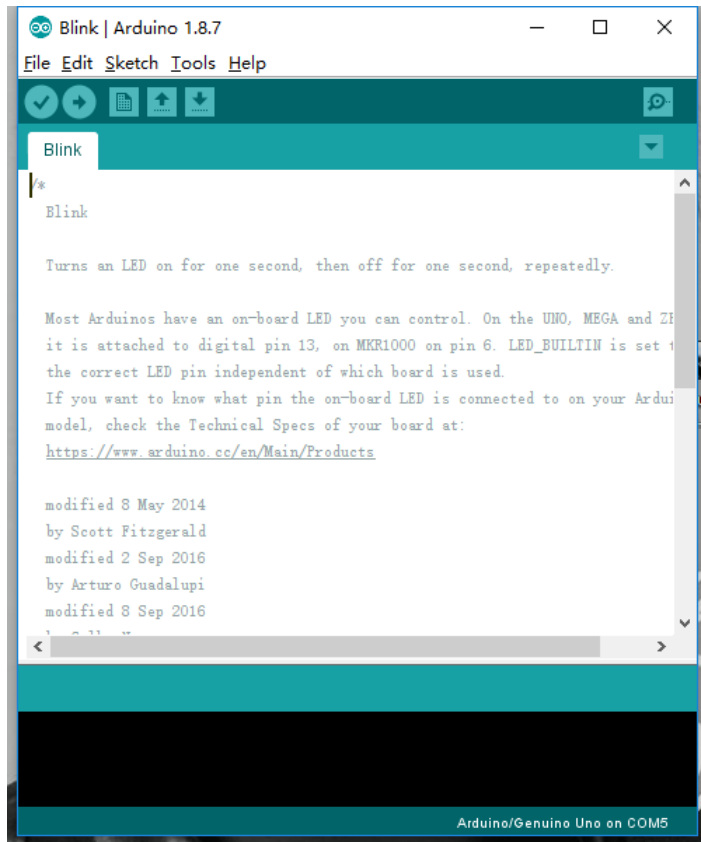


3 components



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

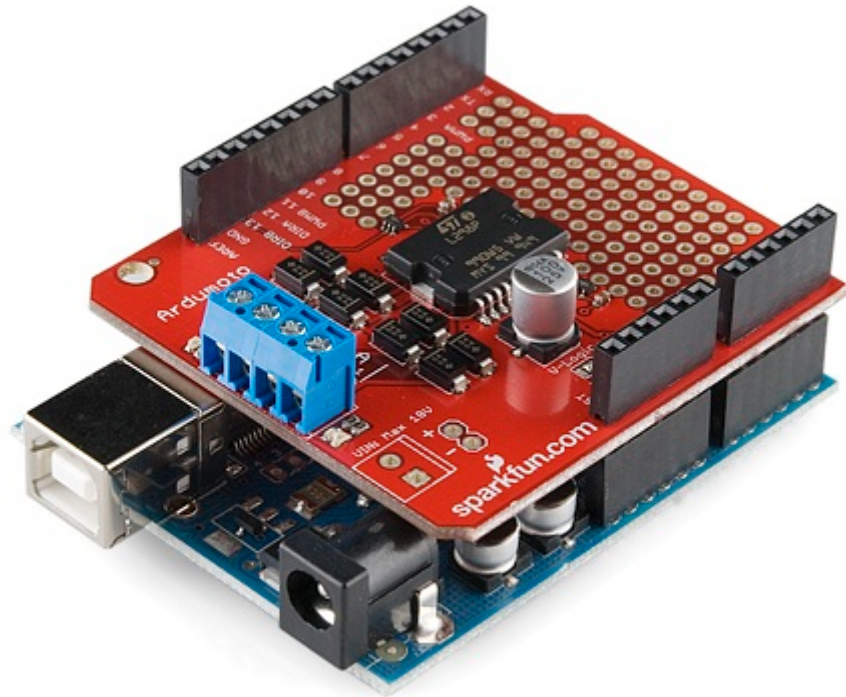
The Arduino Development Board



The Arduino IDE

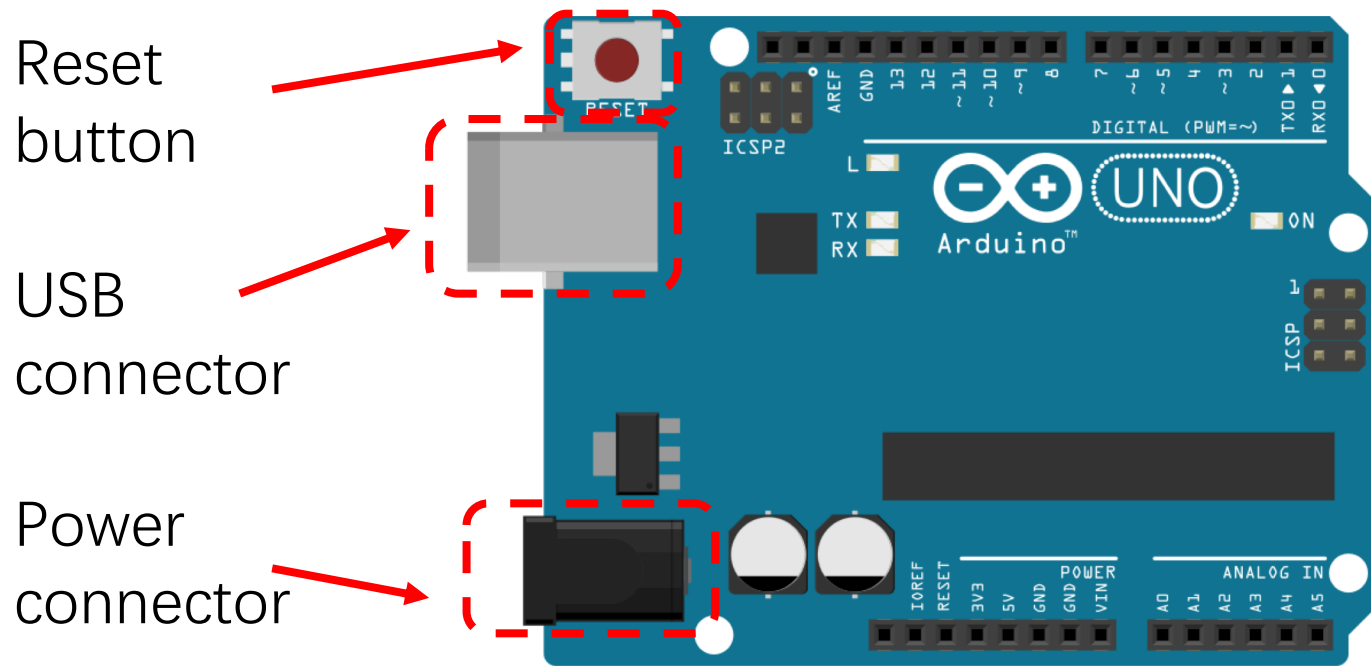
A software environment

- Cross-compiler
- Debugger
- Simulator
- ...



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

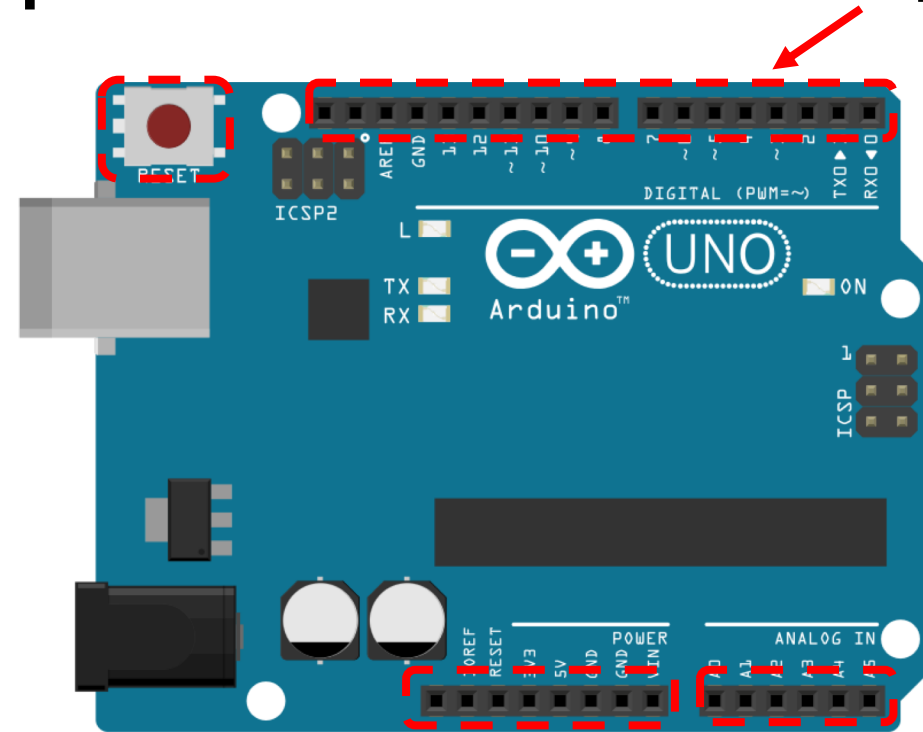
The Arduino Shields



The Arduino Development Board

Input / Output Pins

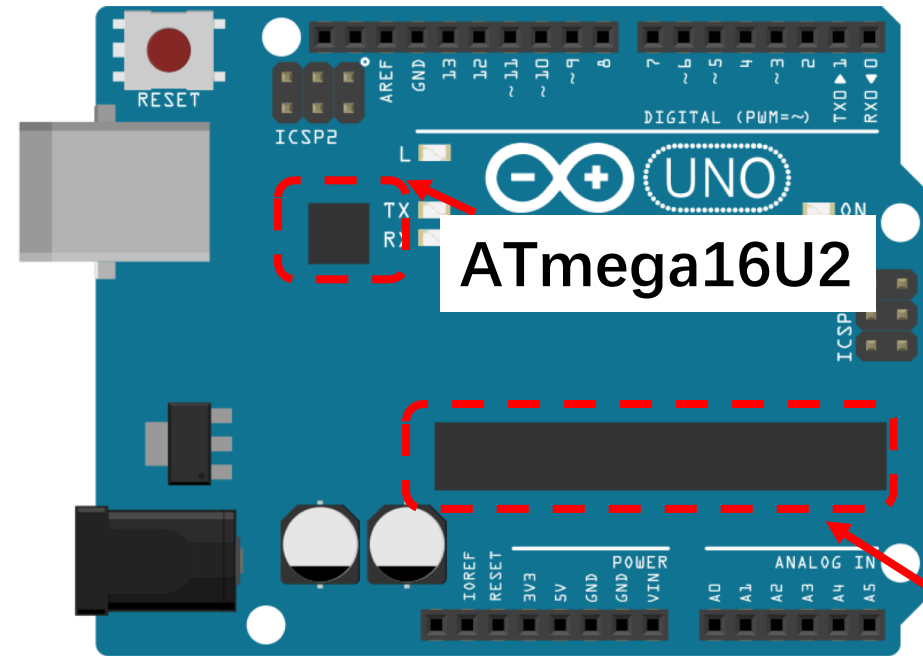
Digital I/O



Analog inputs

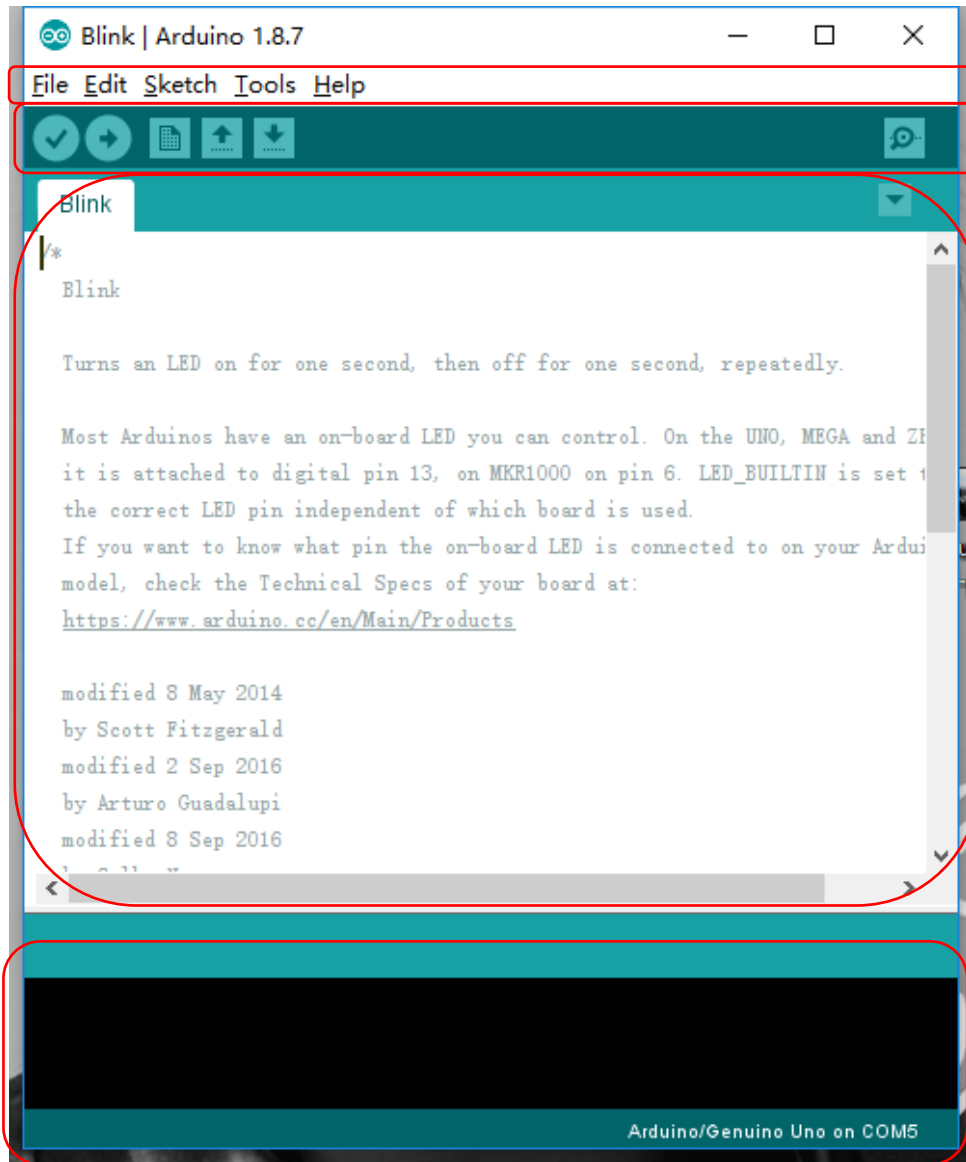
Power / Reset Pins

Microcontrollers



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

ATmega328



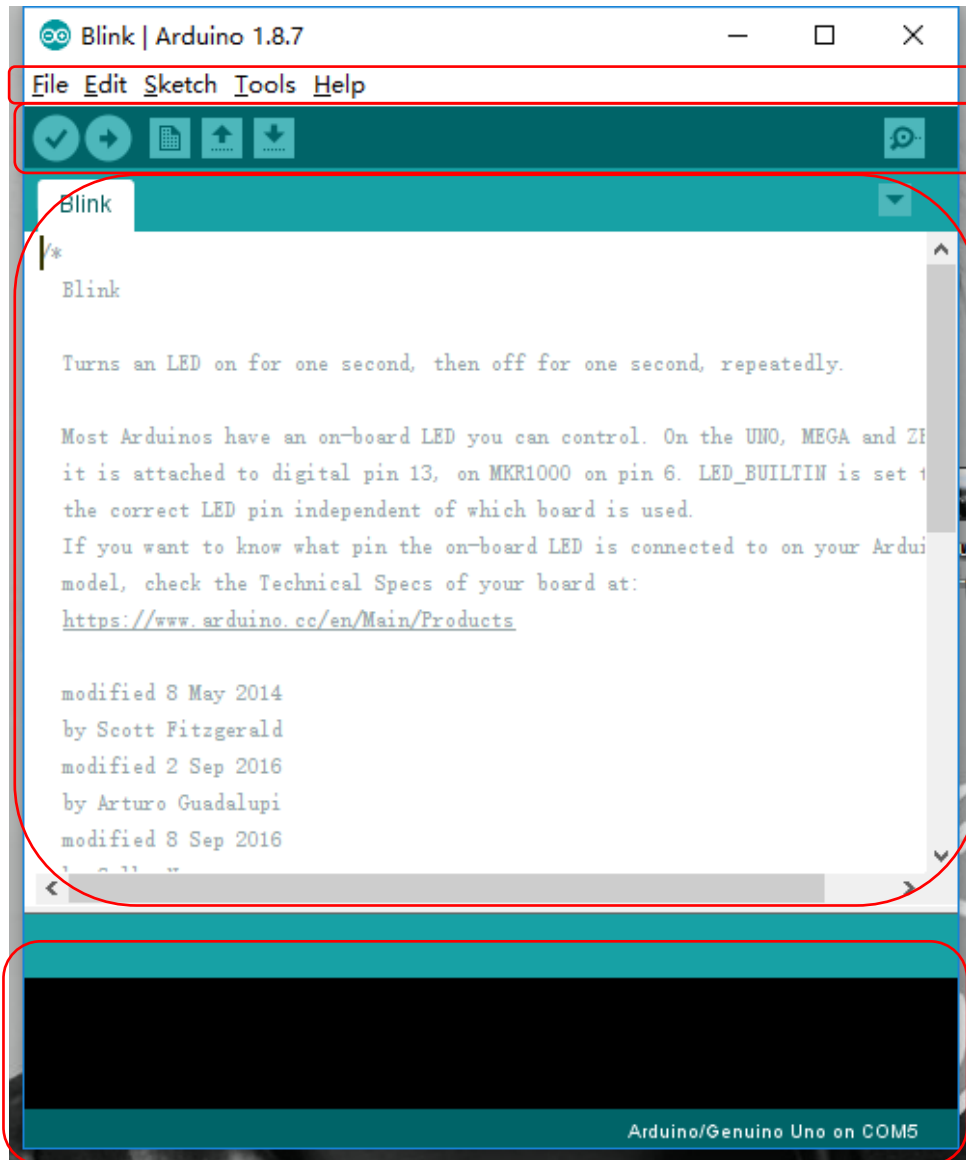
Menus with all commands

Buttons for common commands







Text editor for writing code

Message area

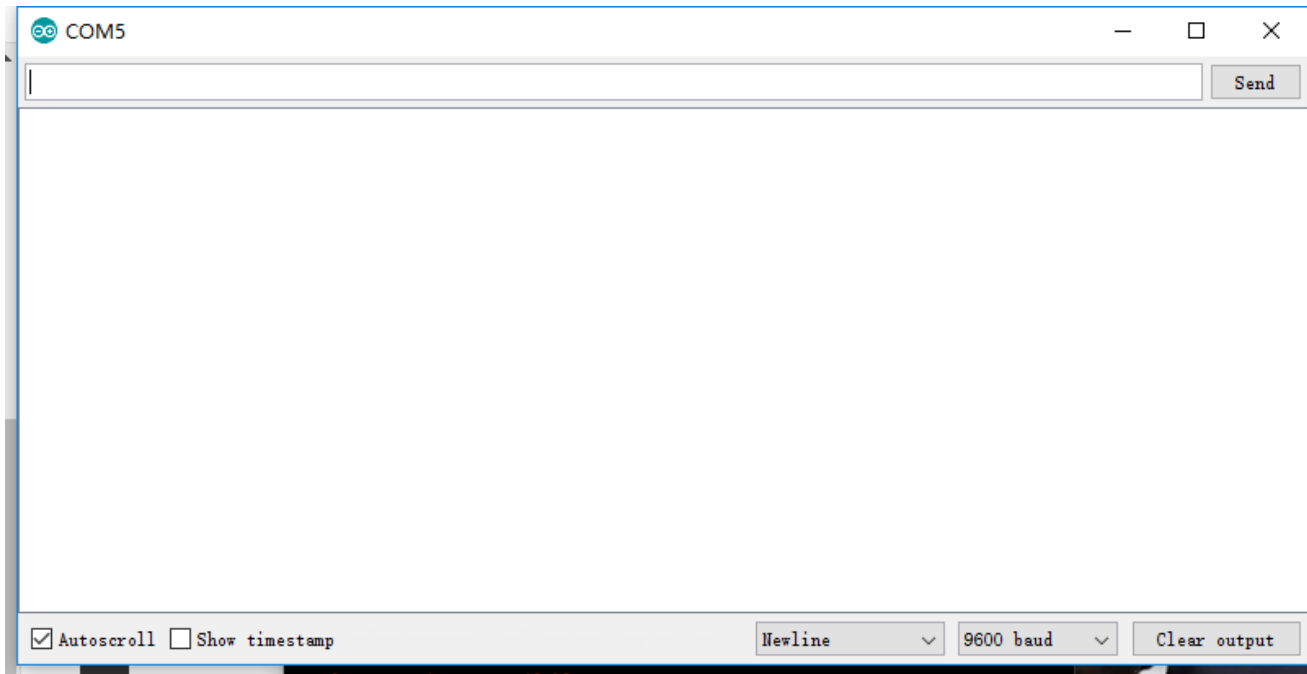
The Arduino IDE



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

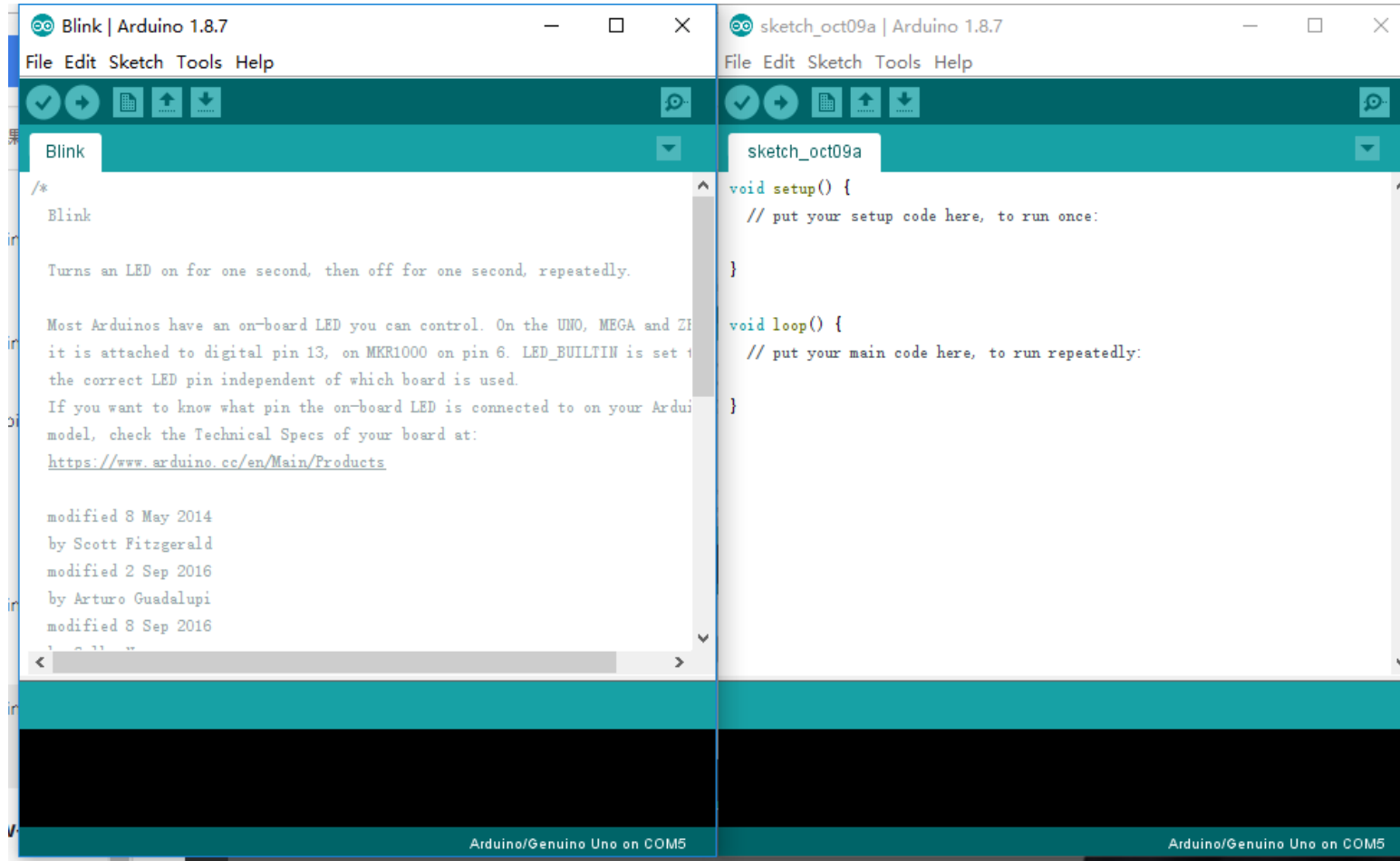


- Displays serial data sent from the Arduino
- Allows serial data to be sent to the Arduino from the keyboard
- Library functions in the serial library

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() {  
    ...  
}
```

```
sketch_oct09a  
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

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() {  
    ...  
}
```

```
sketch_oct09a  
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

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 reserved 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

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

```
// 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
}
```


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 4 times in 1 second and 2 times in 1 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 have to show the LED blink-blink during at least 5 seconds in the video.
- In the 2nd experiment, you have to sequentially 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.readString();
```