CSC270—Circuits & Systems
Arduino Unit #1

Dominique Thiébaut
dthiebaut@smith.edu
Introduction to The Arduino
Outline

- Introduction
- Physical Layout
- Specifications
- Power
- Memory
- Programming
  - Digital I/O
  - Analog I/O
History

The Arduino project was started at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. In 2003 Hernando Barragán created the development platform Wiring as a Master's thesis project at IDII, under the supervision of Massimo Banzi and Casey Reas.

https://en.wikipedia.org/wiki/Arduino
In 2001, together with MIT PhD candidate Ben Fry, Reas created the **Processing programming language**. Processing is widely used by thousands of artists and designers worldwide, and by educators teaching the fundamentals of programing in art and design schools.
```cpp
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
}
```

```java
PImage img;
int smallPoint, largePoint;

void setup() {
  size(640, 360);
  img = loadImage("moonwalk.jpg");
  smallPoint = 4;
  largePoint = 40;
  imageMode(CENTER);
  noStroke();
  background(255);
}

void draw() {
  float pointillize = map(mouseX, 8, width, smallPoint, largePoint);
  int x = int(random(img.width));
  int y = int(random(img.height));
  color pix = img.get(x, y);
  fill(pix, 128);
  ellipse(x, y, pointillize, pointillize);
}
```

You are running Processing revision 0246, the latest build is 0269.
The Mega2650 Arduino (Made by ELEGOO)

"Established in March 2015, ELEGOO Inc. is a thriving technology company dedicated to open-source hardware research & development, production, and marketing. Located in Shenzhen, the Silicon Valley of China, we have grown to over 100+ employees with a 10,763+ square ft. factory."

https://www.elegoo.com/
References

- https://www.arduino.cc/
- https://www.lynda.com/search?q=arduino
Language Reference

Arduino programming language can be divided in three main parts: functions, values (variables), and structure.

FUNCTIONS

For controlling the Arduino board and performing computations.

Digital I/O
- digitalRead()
- digitalWrite()
- pinMode()

Characters
- isAlpha()
- isAlphaNumeric()
- isAscii()
- isControl()
4-min Intro

https://www.youtube.com/watch?v=3Ek7UEPbkqq

time_continue=188
Mega: Specs

- Controller based on **Atmega2560**
- **8-bit** microcontroller. RISC architecture
- **54 digital** IO pins (15 can be used in PWM)
- **16 analog** pins
- **16MHz** crystal
- Programmed with custom **IDE**
- Serial communication ports
Mega: Specs (cont'd)

- Operating voltage: 5V
- Power supply: 7-12V
- Flash memory: 256KB
- RAM 8KB
- EEPROM 4KB
USB. Connect to a laptop via a USB Cable.

VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND. Ground pins.
Power

- +5V
- +7/+12V
- 3.3V 50mA
- 5V
- 6-12V
How Long can a 9V Battery Power the Arduino?

- Hard to tell: depends on what's connected

- **Solution**: make Arduino monitor itself (with analog input)!

Memory

- **256 KB of flash memory** where the sketch is stored (of which 8 KB is used for the bootloader).

- **8 KB of SRAM** where the sketch creates and manipulates variables when it runs (strings of chars are stored there).

- **4 KB of EEPROM** which is memory space that programmers can use to store long-term information.

- Flash memory and EEPROM memory are **non-volatile** (the information persists after the power is turned off). SRAM is volatile and will be lost when the power is cycled.
• If your sketch communicates with a program running on a (desktop/laptop) computer, you can try shifting data or calculations to the computer, reducing the load on the Arduino.

• If you have lookup tables or other large arrays, use the smallest data type necessary to store the values you need.

• If you don't need to modify the strings or data while your sketch is running, you can store them in flash (program) memory instead of SRAM; to do this, use the PROGMEM keyword.

https://www.arduino.cc/en/tutorial/memory
```c
/*
* Blink
* Turns on an LED on for one second, then off for one second, repeatedly.
* This example code is in the public domain.
*/

void setup() {
    // initialize the digital pin as an output.
    // Pin 13 has an LED connected on most Arduino boards:
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH);    // set the LED on
    delay(1000);               // wait for a second
    digitalWrite(13, LOW);     // set the LED off
    delay(1000);               // wait for a second
}
```
Initialization  
(Set pins as Input, output, or other)  

Check sensors, control devices

```cpp
void setup() {
    // initialize the digital pin as an output.
    // Pin 13 has an LED connected on most Arduino boards:
    pinMode(13, OUTPUT);
}

void loop() {
    digitalWrite(13, HIGH);  // set the LED on
    delay(1000);            // wait for a second
    digitalWrite(13, LOW);  // set the LED off
    delay(1000);            // wait for a second
}
```
Programming: Two Options

- Setup
- Loop
- Loop
- Loop
- Loop
- Loop

Time
Programming: Two Options

```c
void loop() {
    // --- compute ---
    count++;
    x = func( count );
    // --- wait 1 sec ---
    delay( 1000 );
}
```

Done uploading.

Sketch uses 1950 bytes (0%) of program storage space. Maximum is 253952 bytes.
Global variables use 194 bytes (2%) of dynamic memory, leaving 7998 bytes free.
Important

• When the **arduino** is powered back up, it automatically restarts the last sketch that was uploaded to it:
  
  • **setup()** is executed again
  
  • **loop()** is repeated forever
Printing is different...
Serial.print()
Serial.print()
Challenge: Explain This!
Challenge: Explain This!

Some questions about previous slide

• Does the timing make sense?

• What does 9600 mean?

• Do we see the 5-second delay?

• What is your best guess as to "who" prints the time we see in the Serial Monitor window, your laptop or the Arduino?
Blink! Exercise

- Go to https://www.arduino.cc/en/Guide/ArduinoMega2560

- Setup your Arduino, and run the **Blink sketch** on your AtMega2560
• Review the previous slides

• Change the delay for the blink sketch. Can you still see the LED blink with a delay of 1?

• Change the blink sketch so that it blinks twice rapidly, then wait a long period of time before repeating to blink again