Internet of Things
Programming

IoT Hardware and Software

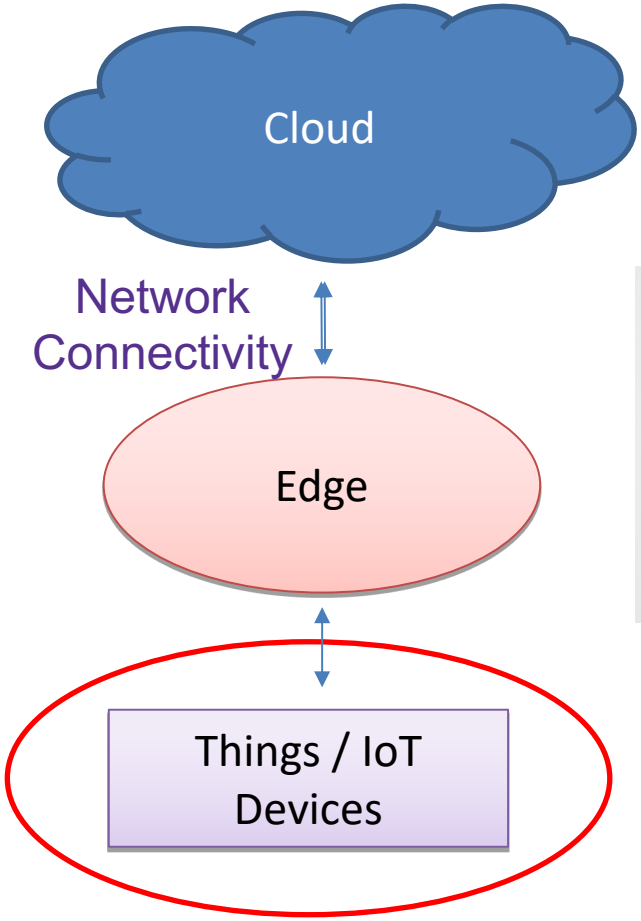
Anas Dawod
adawod@swin.edu.au

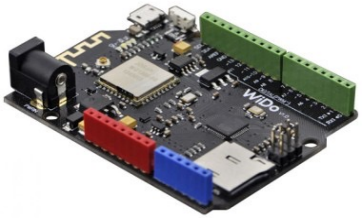
Swinburne University of Technology
March 2023

First Assignment

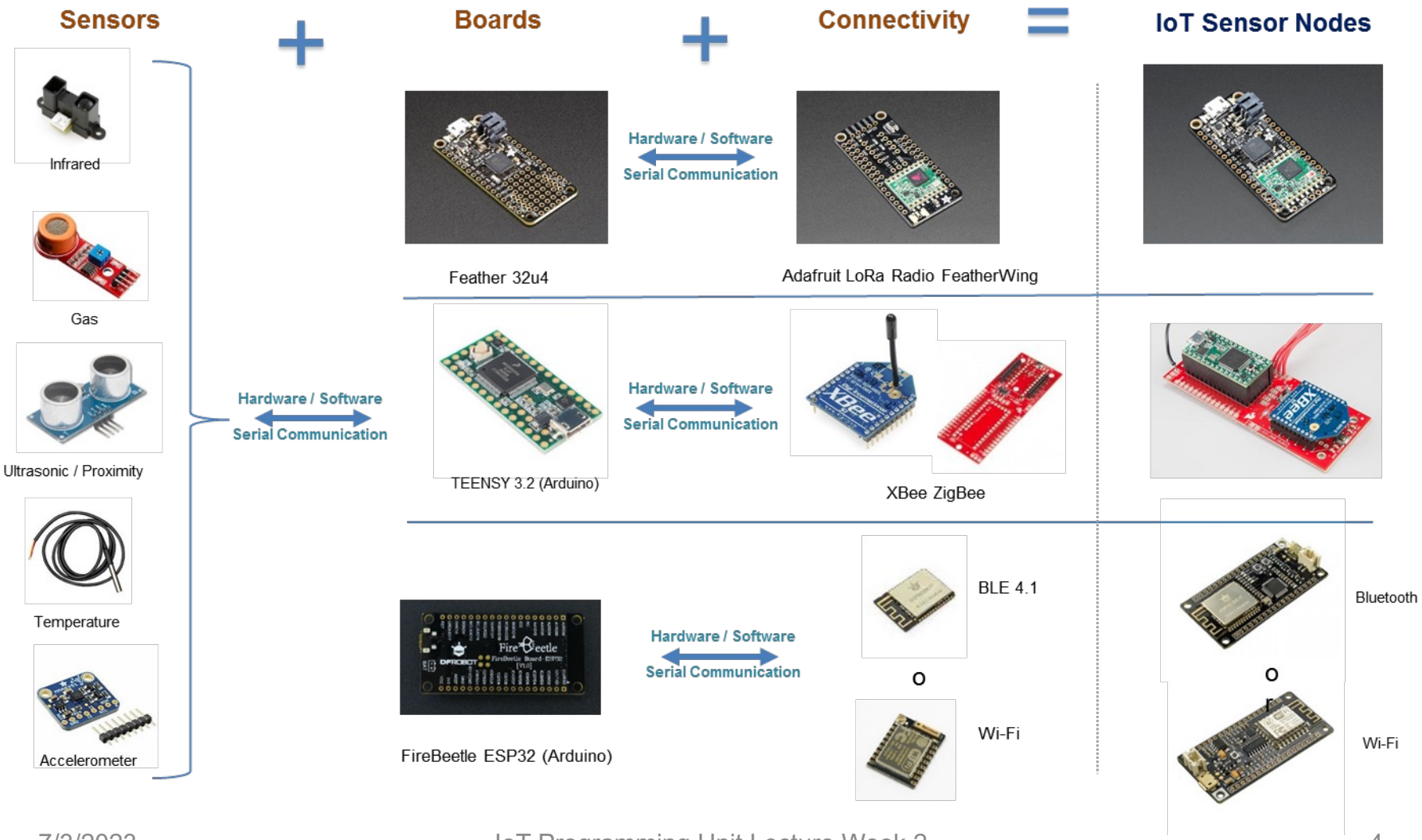
- Survey paper assignment is now released
- The due date is on ….
- Any question?

IoT Hardware and Software





IoT Hardware and Software



IoT Hardware and Software

Sensors

Temperature Sensors

Analog Soil Moisture

Humidity

Infrared Thermometer

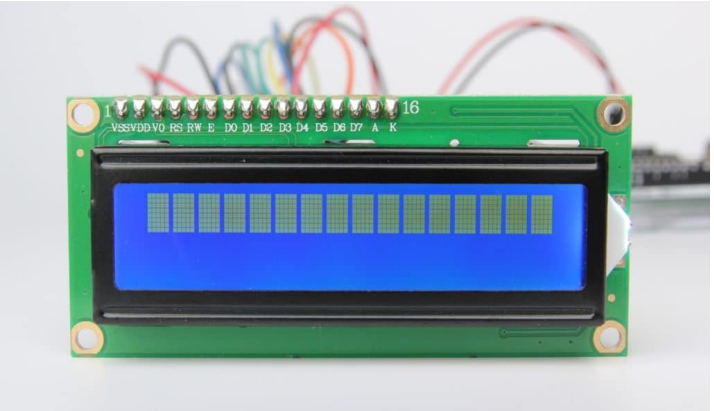
Gas Sensor

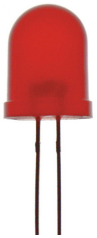
CO2 Sensor

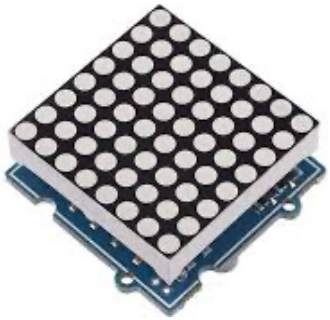
Digital Push Button

IoT Hardware and Software

Actuators

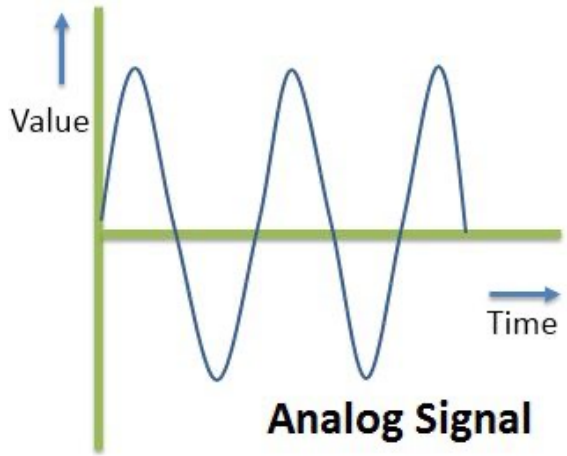


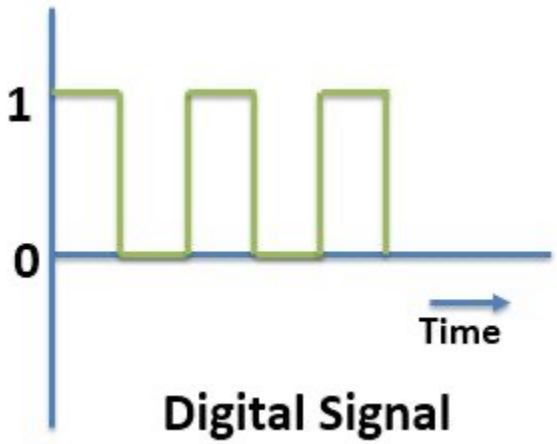




Analog VS Digital Signals

- An analogue signal can take on any number of values, unlike a digital signal which has a limited number and its usually two values: HIGH and LOW.





An analogue signal is a continuous wave that changes over a time period.

A digital signal is a discrete wave that carries information in binary form.

Microcontrollers

- Is an integrated circuit (IC)
- Tiny computer (processor, memory, and input/output)
- It is programmable
- Designed for embedded applications

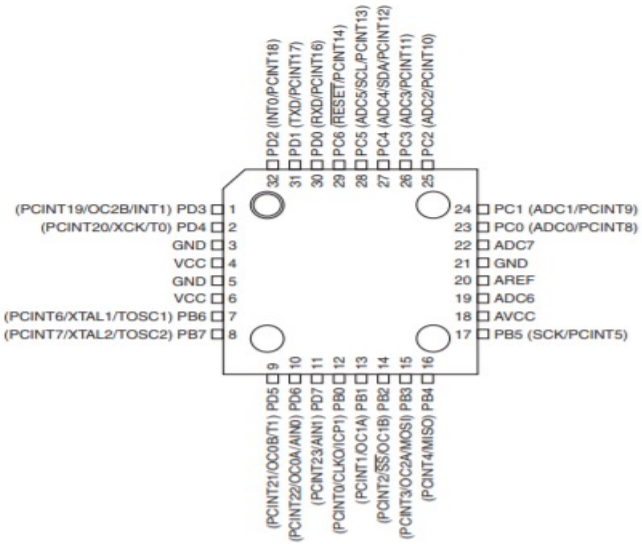
Intel P8051

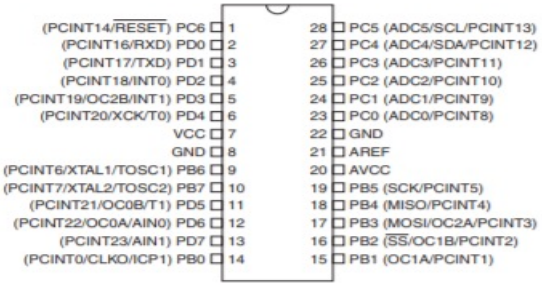
Hitachi HD68000

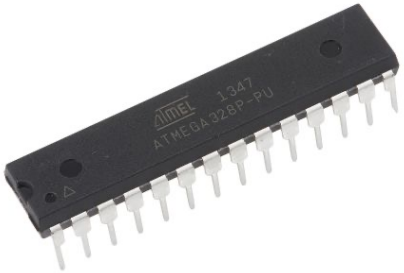
Motorola MC68HC000LC8

Microcontrollers

ATmega 328P





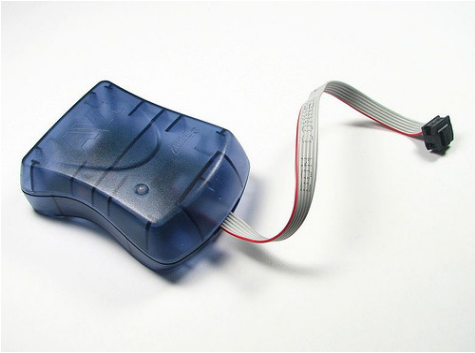


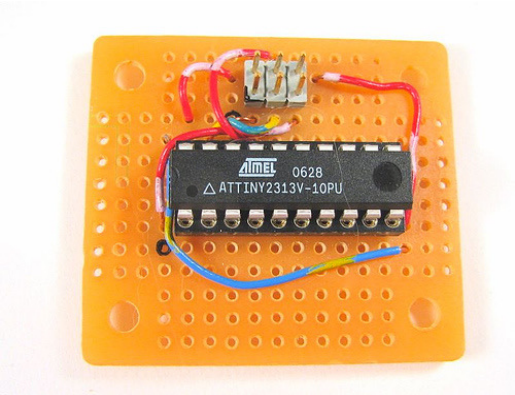
Microcontrollers

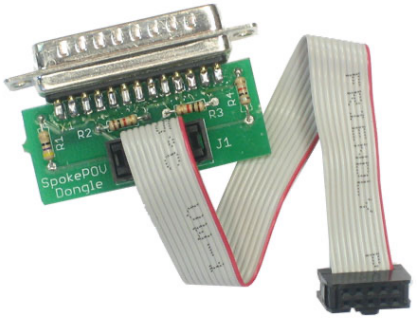
- High Performance, Low Power AVR® 8-Bit Microcontroller
- It has an EEPROM memory of 1KB and its SRAM memory is of 2KB.
- Program Memory 32KB
- 23 Programmable I/O Lines
- Operating Voltage: – 1.8 - 5.5V
- Temperature Range: – -40°C to 85°C
- Power Consumption
  - Active Mode: 0.2 mA
  - Power-down Mode: 0.1 μA
  - Power-save Mode: 0.75 μA

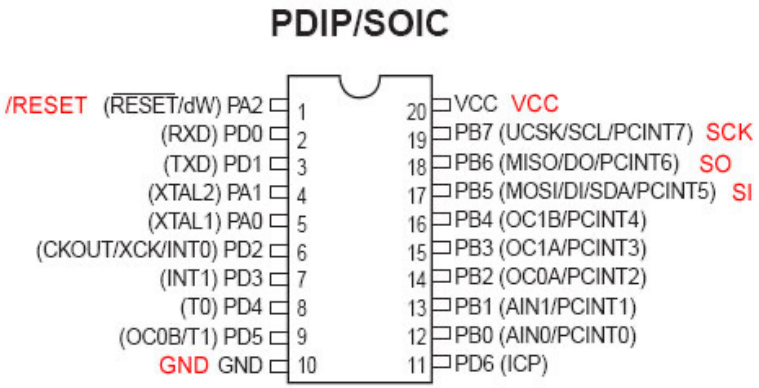
Microcontrollers

ATmega Programming









Pin #	Pin Name	Pin Function
1	#CS	Chip Select (Activate)
2	SI	Serial In
3	GND	Ground
4	VCC	Power
5	SCK	Data Clock
6	Not connected	Not connected
7	SO	Serial Out

Microcontrollers

Arduino

- Arduino is an open-source electronics platform based on easy-to-use hardware and software.
- The Arduino project started in 2003 as a program for students at the Interaction Design Institute Ivrea in Italy
- The name Arduino comes from a bar where some of the founders of the project used to meet. The bar was named after Arduin of Ivrea, who was the margrave of the March of Ivrea and King of Italy from 1002 to 1014.

Arduino d'Ivrea
(Civica raccolta stampe - Milano)

Microcontrollers

Arduino

- The original Arduino hardware was produced by the Italian company Smart Projects.
- Some Arduino-branded boards have been designed by other companies such as SparkFun Electronics and Adafruit Industries.

Microcontrollers

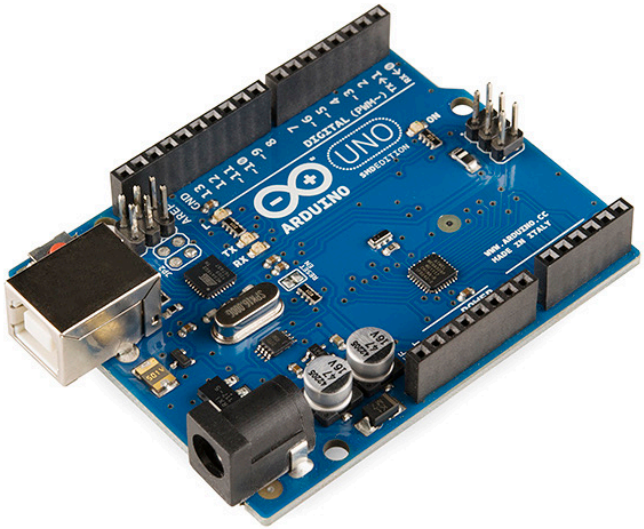
Arduino

- Reasonably Cheap
- Cross-platform
- Simple programming environment
- Open-Source Software
- Open-Source Hardware

Microcontrollers

Arduino UNO

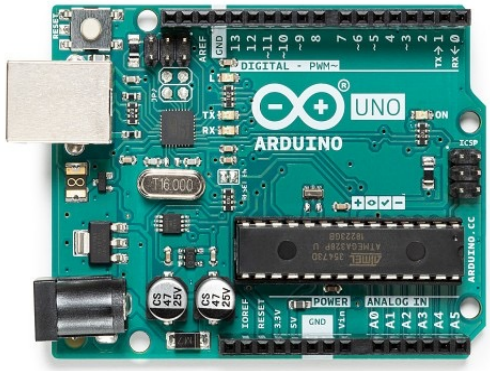
- The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller
- Developed by Arduino.cc



Microcontrollers

Arduino UNO

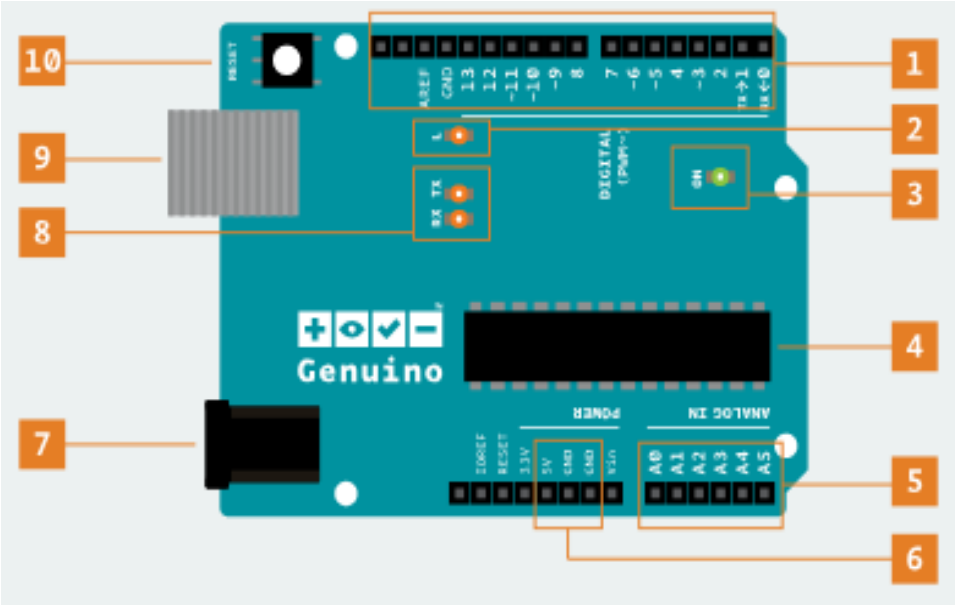
- The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable
- The ATmega328 on the Arduino Uno comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.



Microcontrollers

Arduino Anatomy

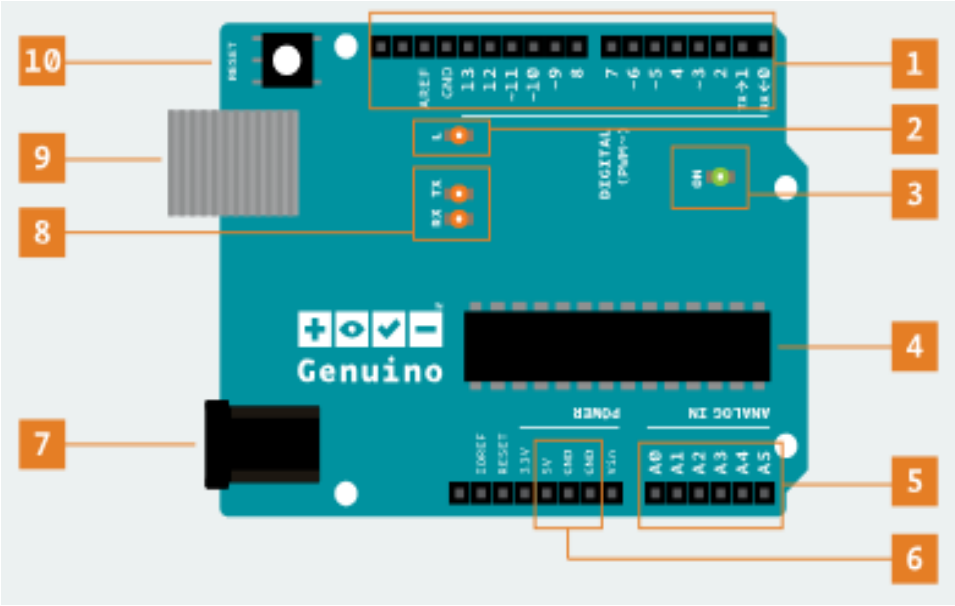
1. Digital pins
2. Pin 13 LED
3. Power LED
4. ATmega microcontroller
5. Analog pins



Microcontrollers

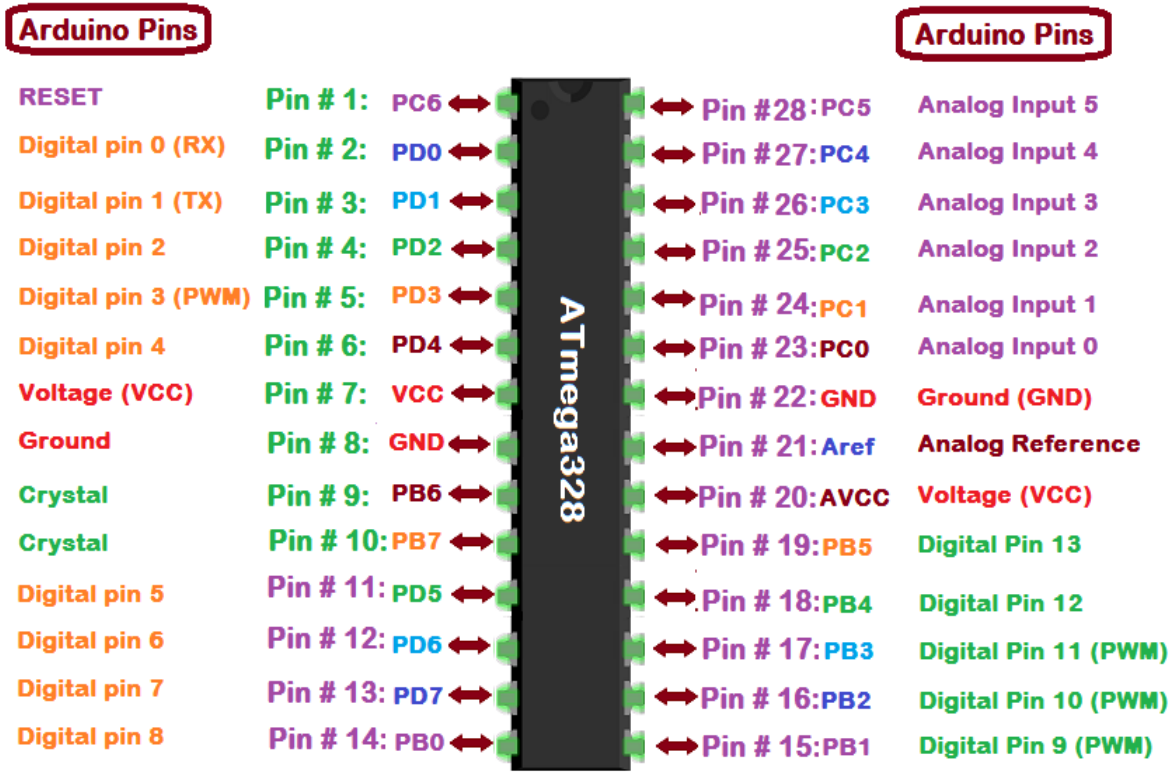
Arduino Anatomy

6. GND and 5V pins
7. Power connector
8. TX and RX LEDs
9. USB port
10. Reset button



Microcontrollers

Arduino's Atmega328 pinout

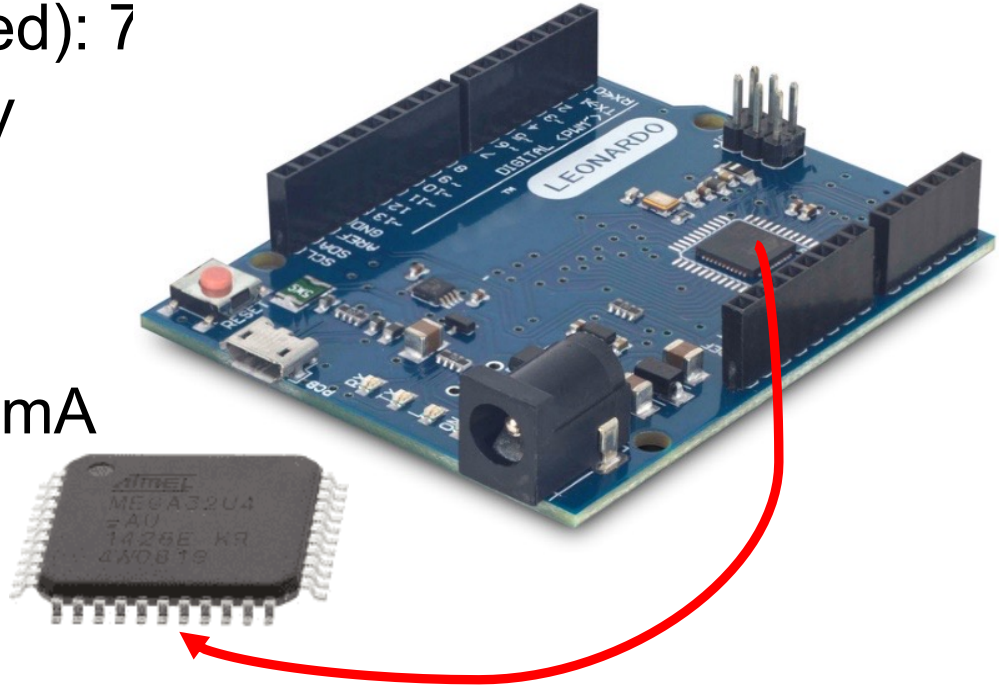


www.TheEngineeringProjects.com

Microcontrollers

Arduino LEONARDO

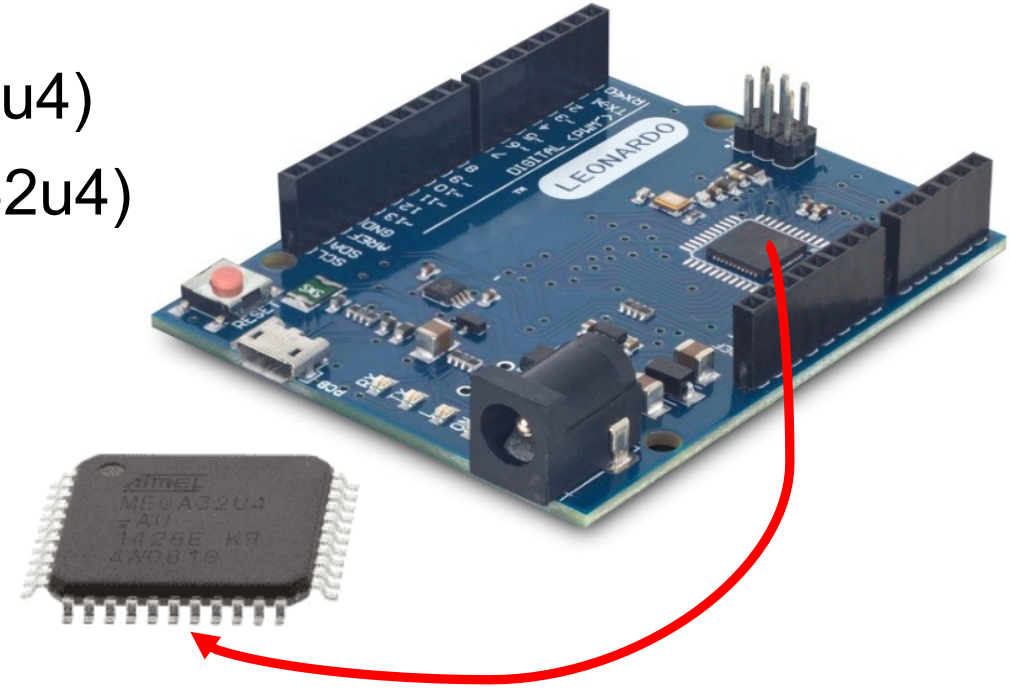
- Microcontroller: ATmega32u4
- Operating Voltage: 5V
- Input Voltage (recommended): 7
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 20
- Analog Input Channels: 12
- DC Current per I/O Pin: 40 mA



Microcontrollers

Arduino LEONARDO

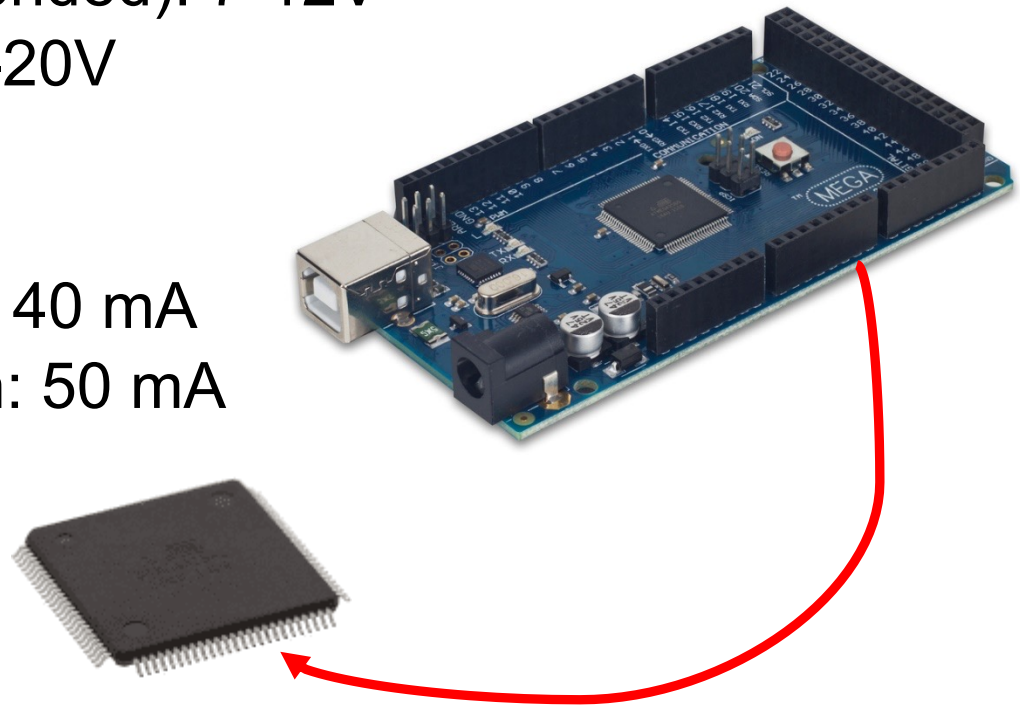
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB (ATmega32u4) of which 4 KB used by bootloader
- SRAM: 2.5 KB (ATmega32u4)
- EEPROM: 1 KB (ATmega32u4)
- Length: 68.6 mm
- Width: 53.3 mm
- Weight: 20g



Microcontrollers

Arduino Mega

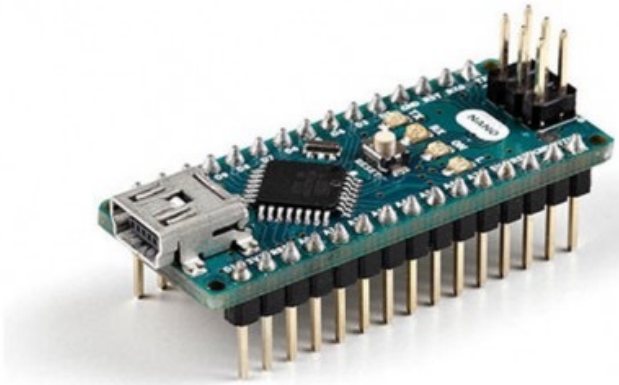
- Microcontroller: ATmega2560
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 54
- Analog Input Pins: 16
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 256 KB
- SRAM: 8 KB
- EEPROM: 4 KB



Microcontrollers

Arduino Nano

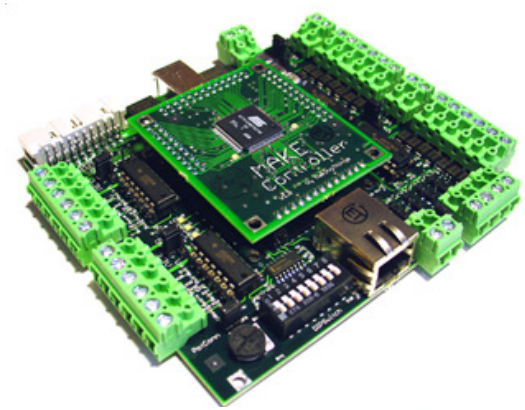
- Microcontroller: ATmega328
- Architecture: AVR
- Operating Voltage: 5 V
- Flash Memory: 32 KB
- SRAM: 2 KB
- Analog Pins: 8
- EEPROM: 1 KB
- Input Voltage: 7-12 V
- Digital I/O Pins: 22 (6 of which are PWM)
- Power Consumption: 19 mA
- PCB Size: 18 x 45 mm
- Weight: 7 g

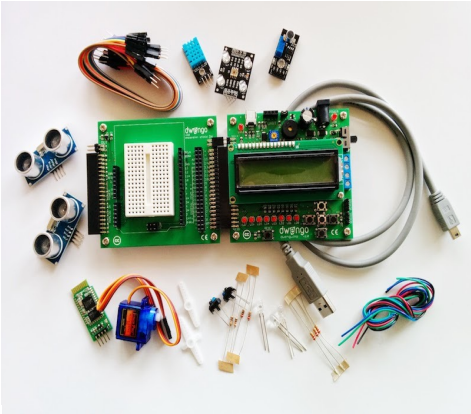


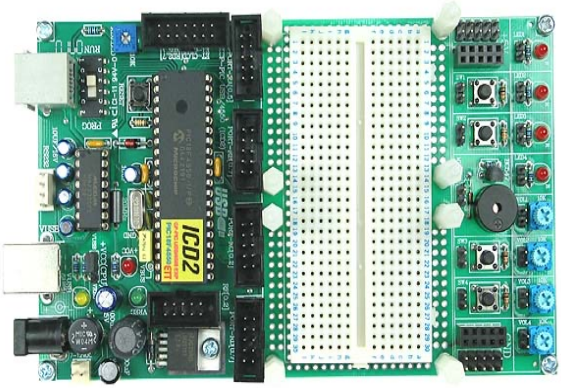
Microcontrollers

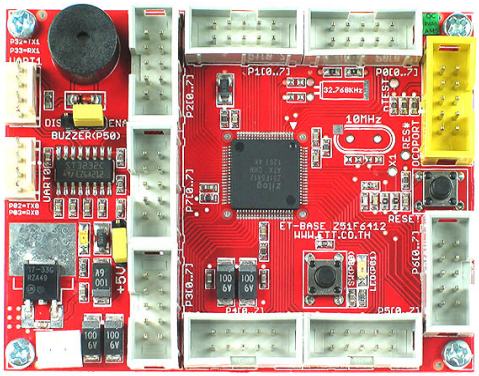
Alternatives board for Arduino

- Single boards microcontrollers





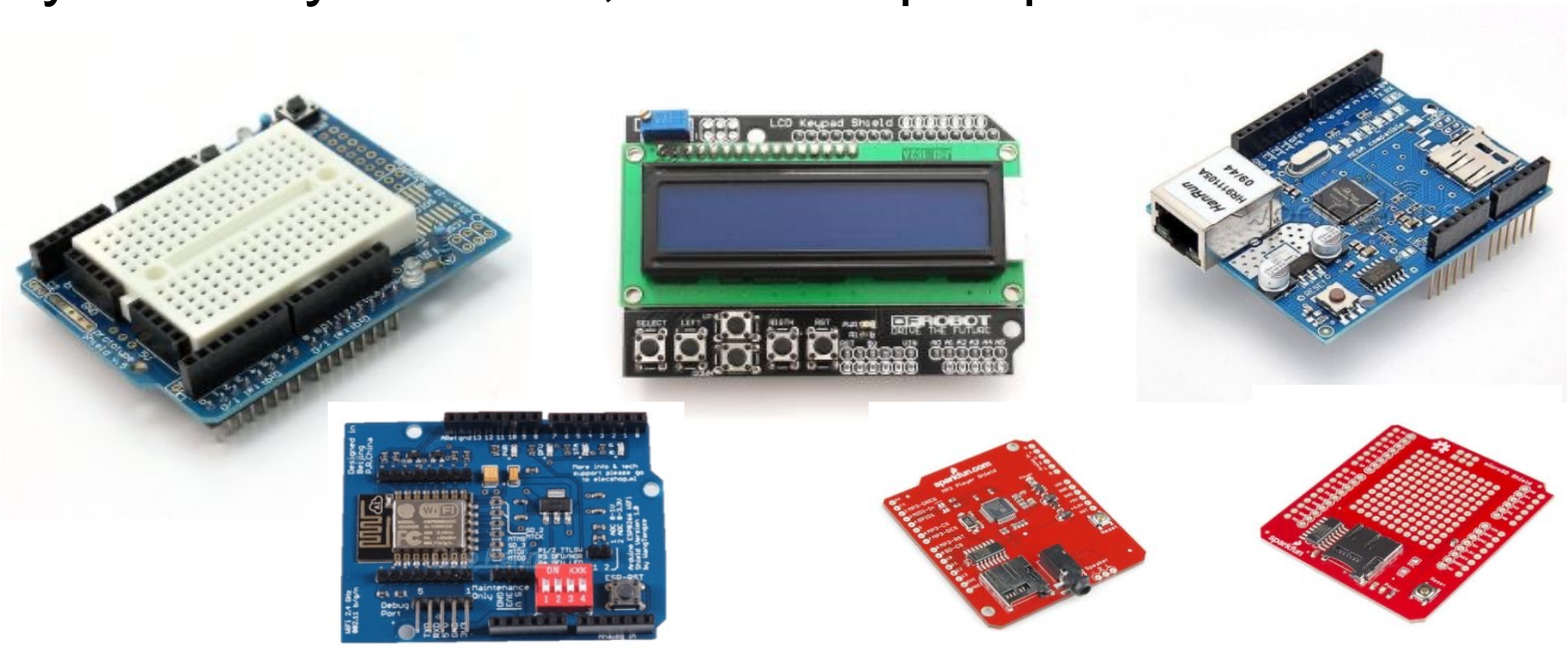




Microcontrollers

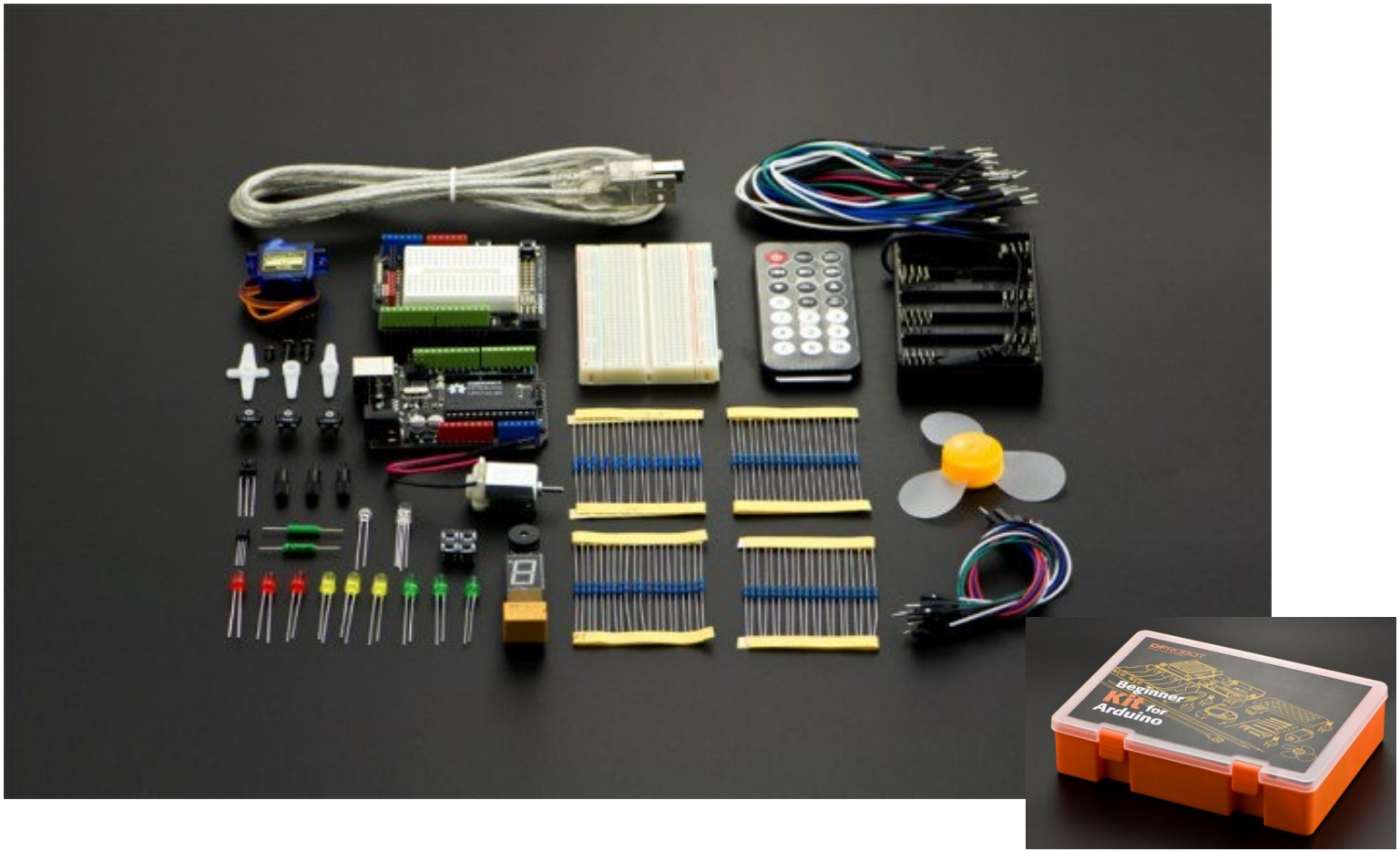
Arduino Shields

- Shields are boards that can be plugged on top of the Arduino PCB extending its capabilities. The different shields follow the same philosophy as the original toolkit: they are easy to mount, and cheap to produce.



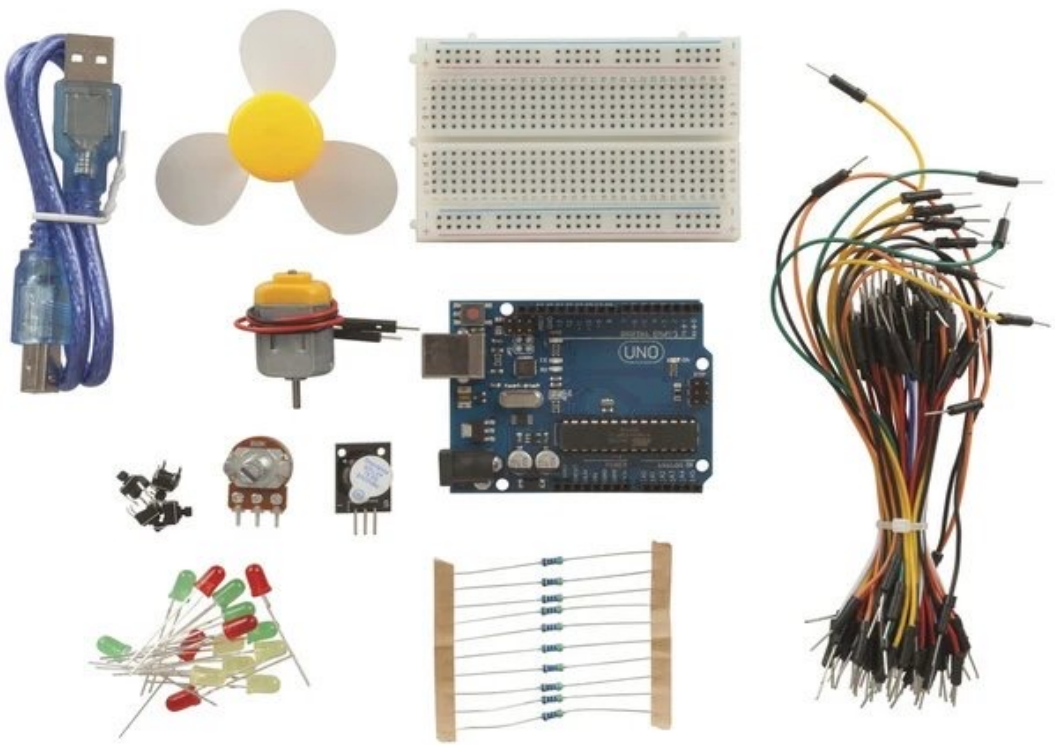
Tutorial Kit

Arduino Kit DFR0100



Tutorial Kit

Arduino Kit



Tutorial Kit

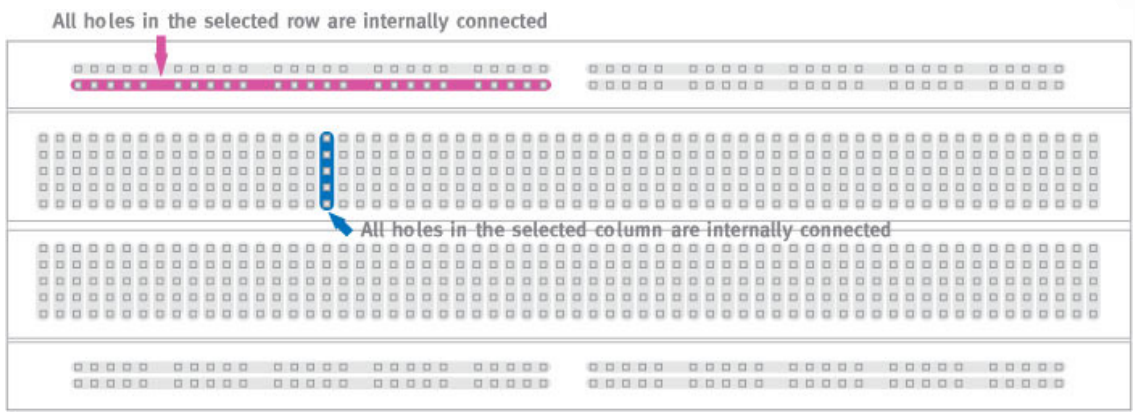
DFRduino UNO R3

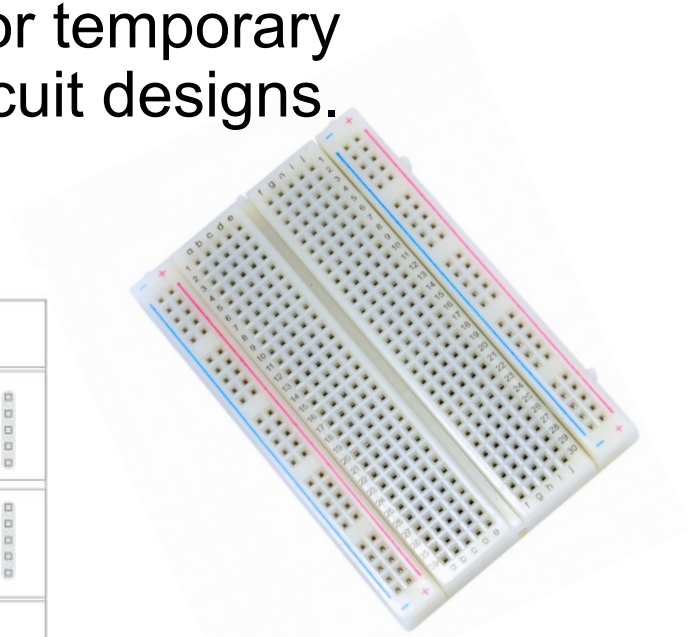
- Microcontroller: ATmega328 (DIP Package)
- Operating Voltage: 5V
- Input Voltage (recommended): 7 ~ 12V
- Input Voltage (limits): 6 ~ 20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB of which 2KB used by bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Size: 75 x54 x15 mm (2.95 x2.13 x0.59")

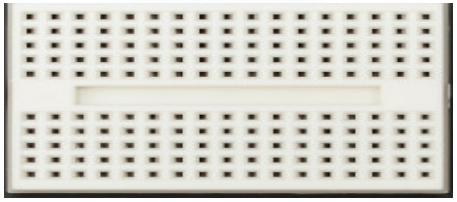
Tutorial Kit

Breadboard

- A breadboard is a solderless device for temporary prototype with electronics and test circuit designs.







http://wiring.org.co/learning/tutorials/breadboard/

IoT Software

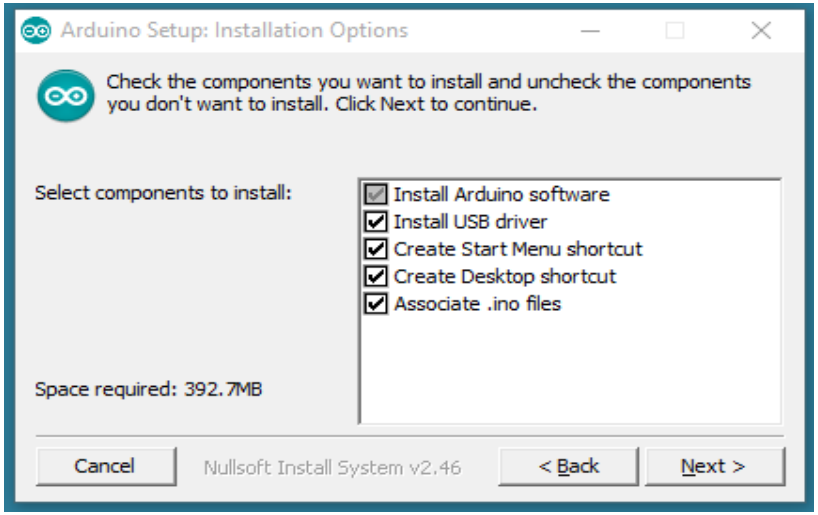
Arduino Sketch

- A sketch is the name that Arduino uses for a program. It's the unit of code that is uploaded to and run on an Arduino board.
- Arduino language is a set of C/C++ functions

IoT Software

Arduino Sketch

- Windows https://www.arduino.cc/en/Guide/Windows
- Mac https://www.arduino.cc/en/Guide/MacOSX
- Linux http://playground.arduino.cc/Learning/Linux
- sudo apt-get install arduino - is not recommended



IoT Software

Arduino Sketch

Blink | Arduino 1.8.5

File Edit Sketch Tools Help

Blink §

```
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

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

1 Arduino/Genuino Uno on COM9

IoT Software

Arduino Programming

- Comments

```
/*
 * This is a comment
 *
 */
```

OR

```
// this is another comment
```

IoT Software

Arduino Programming

- For
  - The *for* statement is used to repeat a block of statements enclosed in curly braces.
  - An increment counter is usually used to increment and terminate the loop.
  - The for statement is useful for any repetitive operation, and is often used in combination with arrays to operate on collections of data/pins.
  - Syntax:

```
for (initialization; condition; increment) {
  // statement(s);
}
```

IoT Software

Arduino Programming

- For
  - Example:

```
void loop() {
  for (int i = 0; i <= 10; i++) {
            Serial.write(7, i);
  }
}
```

Comparison Operators

!= (not equal to)
< (less than)
<= (less than or equal to)
== (equal to)
> (greater than)
>= (greater than or equal to)

https://www.arduino.cc

IoT Software

Arduino Programming

- If
  - The if statement checks for a condition and executes the proceeding statement or set of statements if the condition is 'true.
  - Syntax:

```
if (condition) {
  // statement(s);
} else if (condition 2){
 // statement(s);
} else{
 // statement(s);
}
```

IoT Software

Arduino Programming

- If
  - Example:

```
if (temperature >= 70) {
        //Danger! Shut down the system
} else if (temperature >= 60 && temperature < 70){
        //Warning! User attention required
} else {
        //Safe! Continue usual tasks
}
```

IoT Software

Arduino Programming

- While
  - A while loop will loop continuously, and infinitely, until the expression inside the parenthesis, () becomes false.
  - Something must change the tested variable, or the while loop will never exit. This could be in your code, such as an incremented variable, or an external condition, such as testing a sensor.
  - Example:

```
var = 0;
while (var < 200) {
        //Do something repetitive 200 times
 var++;
}
```

https://www.arduino.cc

IoT Software

Arduino Programming

- Do While
  - The do…while loop works in the same manner as the while loop, with the exception that the condition is tested at the end of the loop, so the do loop will always run at least once.
  - Example:

```
var = 0;
do {
  //Do something repetitive 200 times
 var++;
} while (var < 200)
```

https://www.arduino.cc

IoT Software

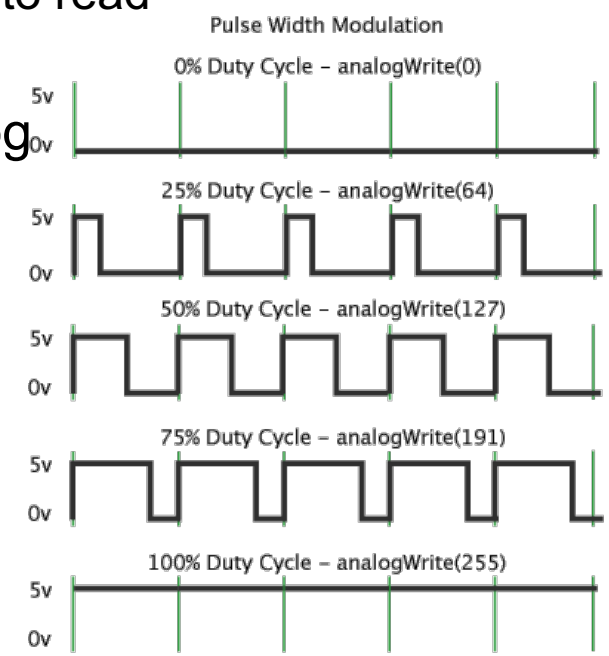
Arduino Programming

- Functions (Digital Input and Output)
  - **digitalRead(pin),** reads the value from a specified digital pin, either HIGH or LOW.
    - pin: the number of the digital pin you want to read
  - **digitalWrite (pin, value),** writes a HIGH or a LOW value to a digital pin.
    - pin: the pin number
    - value: HIGH or LOW
  - **pinMode(pin, mode),** configures the specified pin to behave either as an input or an output.
    - pin: the number of the pin whose mode you wish to set
    - mode: INPUT, OUTPUT

IoT Software

Arduino Programming

- Functions (Analogue)
  - **analogRead(pin),** reads the value from a specified analog pin. 10-bit analog to digital converter.
    - pin: the number of the digital pin you want to read
  - **analogWrite (pin, value),** Writes an analog value (PWM wave) to a pin
    - pin: the pin number
    - value: the duty cycle:
      between 0 (always off)
      and 255 (always on).



IoT Software

Arduino Programming

- Functions (Math)
  - **abs() -** Calculates the absolute value of a number.
  - **constrain() -** Constrains a number to be within a range.
  - **constrain(x, a, b),** x: if x is between a and b, a: if x is less than a, b: if x is greater than b
  - **max(x, y) -** Returns maximum between x and y
  - **min(x, y) -** Returns minimum between x and y
  - **pow(base, exponent)-** Calculates the value of a number raised to a power.
  - **sq()-** Calculates the square of a number: the number multiplied by itself.
  - **sqrt(x) -** Calculates the square root of a number.

IoT Software

Arduino Programming

- Functions (Delay)
  - delay(milliseconds), Pauses the program for the amount of time (in milliseconds) specified as parameter.
    - There are 1000 milliseconds in a second.

IoT Software

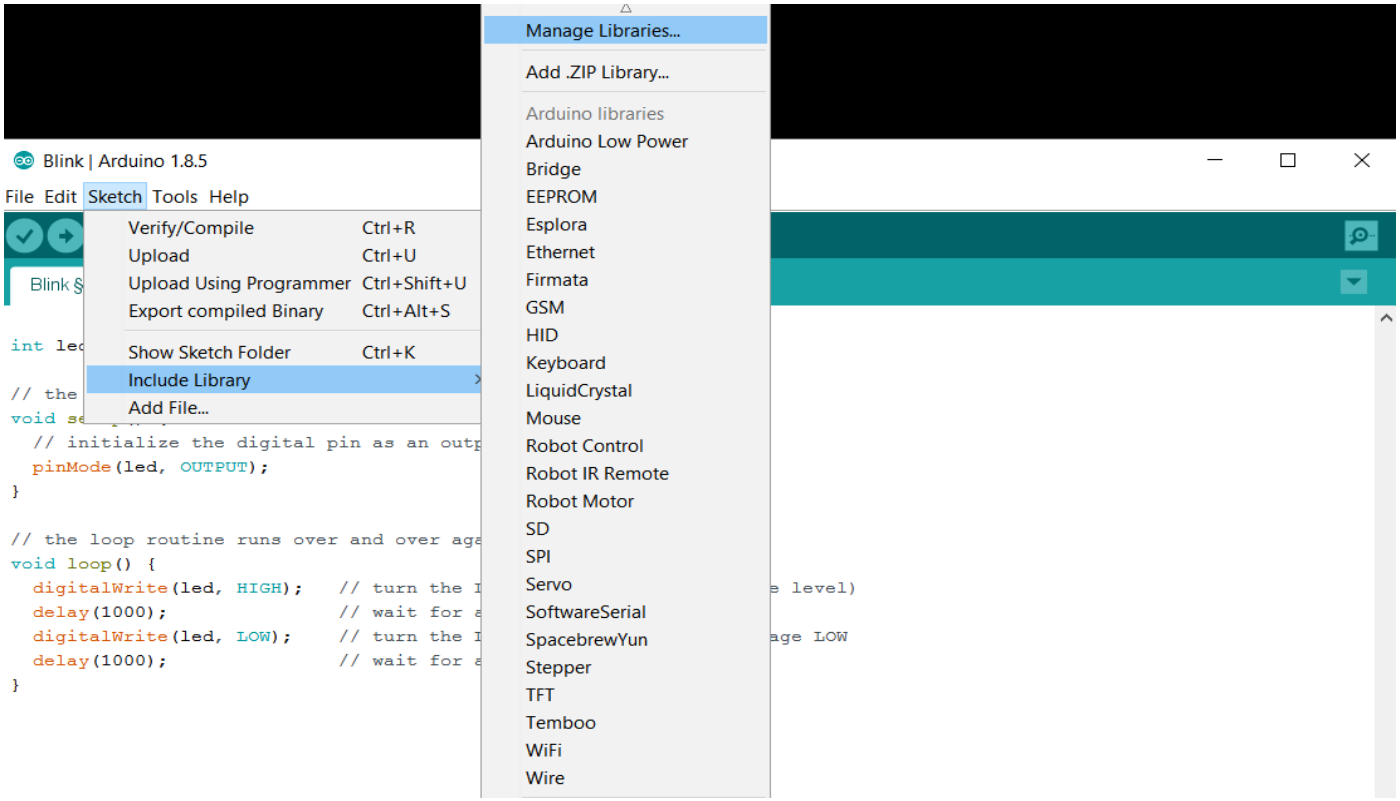
Arduino Programming

- Functions (Structure)
  - #define is a useful C++ component that allows the programmer to give a name to a constant value before the program is compiled.
    - #define ledPin 3
  - #include is used to include outside libraries in your sketch. This gives the programmer access to a large group of standard C libraries (groups of pre-made functions), and also libraries written especially for Arduino.
    - #include <SPI.h>

IoT Software

Arduino Programming

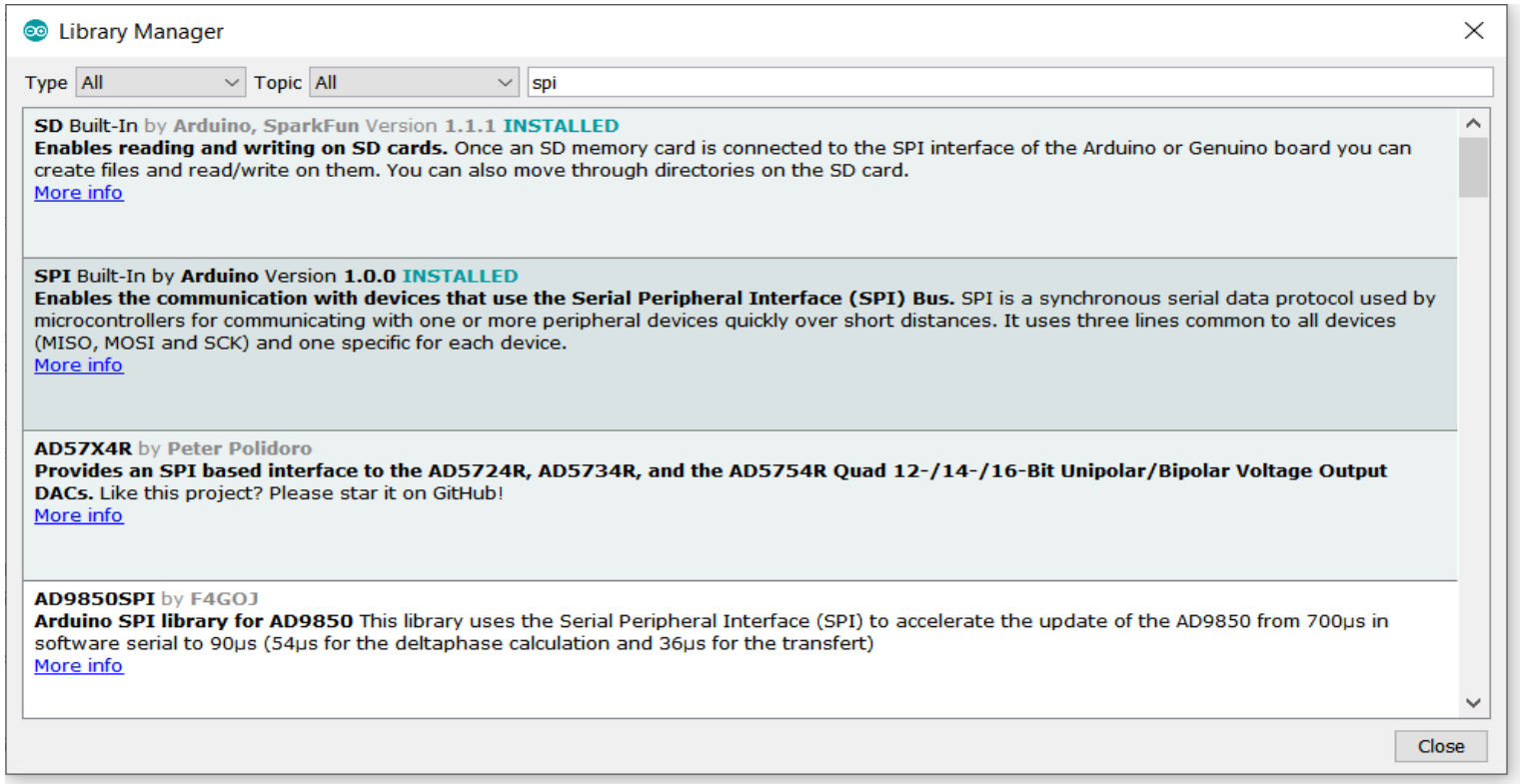
- Adding a library



IoT Software

Arduino Programming

- Adding SPI library (#include <SPI.h>)



IoT Software

Arduino Programming

- Setup
  - The setup() function is called when a sketch starts.
  - Use it to initialise variables, pin modes, start using libraries, etc.
  - The setup() function will only run once, after each power up or reset of the board.
  - Example:

```
void setup() {

        pinMode(8, INPUT);

}
```

IoT Software

Arduino Programming

- Loop
  - After creating a setup() function, which initialises and sets the initial values, the loop() function loops consecutively, allowing your program to change and respond.
  - Use it to actively control the Arduino board.
  - Example:

```
void loop() {
        if (digitalRead(8) == HIGH) {
        Serial.write('High');
        }
        else { Serial.write('LOW'); }
}
```

IoT Software

Serial Communication Port

- COM (Communication port) is the original, yet still common, name of the serial port interface on IBM PC-compatible computers.
- It might refer not only to physical ports, but also to virtual ports, such as ports created by Bluetooth or USB-to-serial adapters.

IoT Software

Serial Communication Port

- Used for communication between the Arduino board and a computer or other devices. All Arduino boards have at least one serial port and some have several.
- Arduino Uno Pin 0 is RX , Pin 1 is TX
- Example:

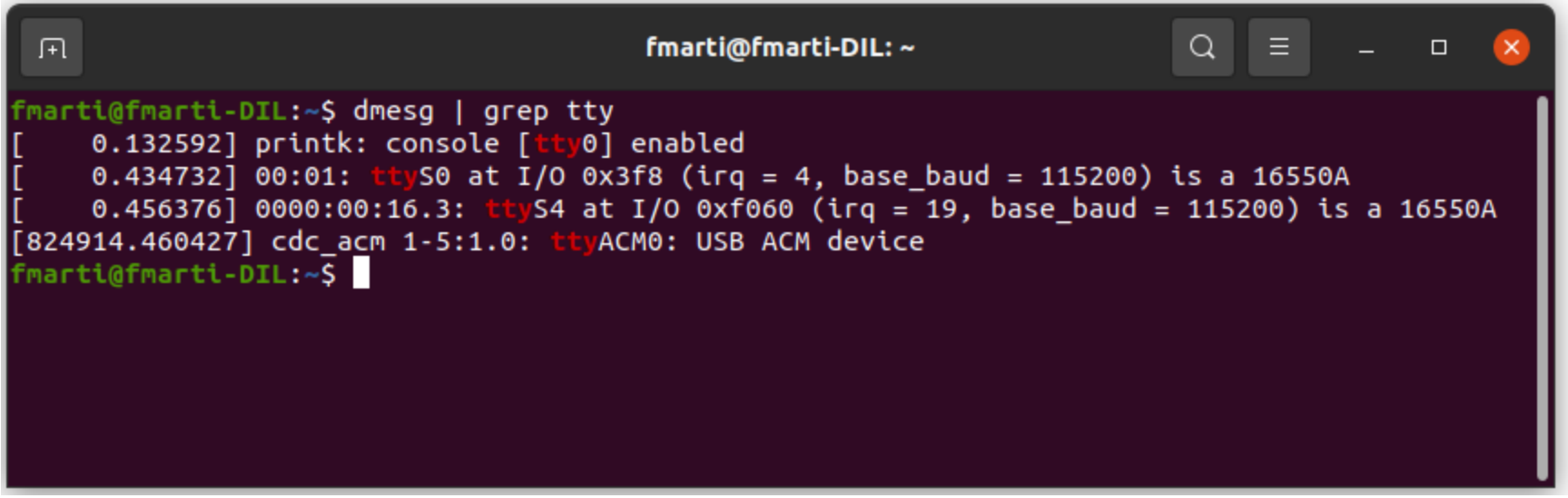
  - Serial.begin(speed), start the serial communication
  - Serial.write(value), writes in the serial port

  speed in baud, which is bits per second rate

IoT Software

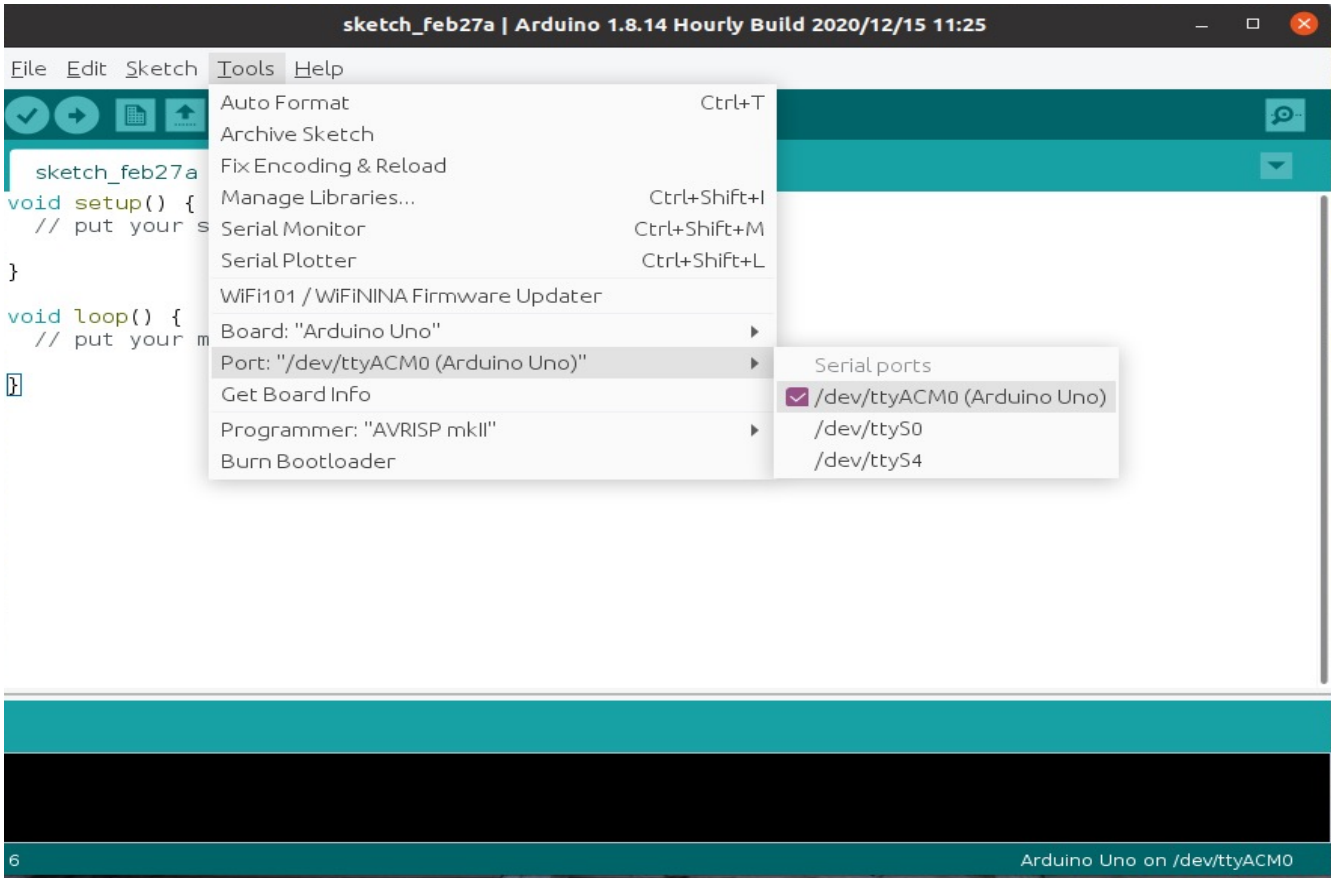
Serial Communication Port

```
Linux: dmesg | grep tty
  Mac:  ls /dev/tty.*
```



IoT Software

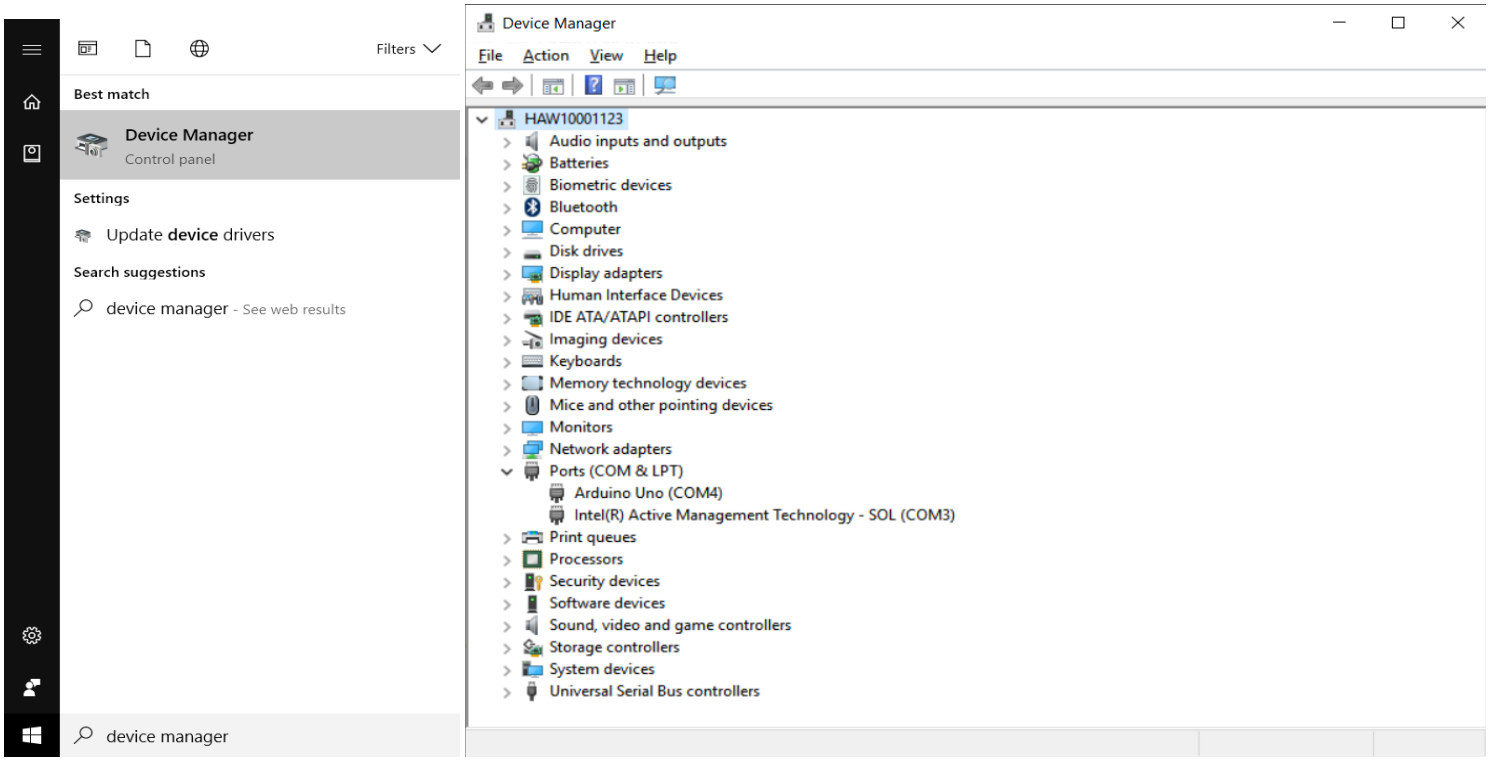
Serial Communication Port



IoT Software

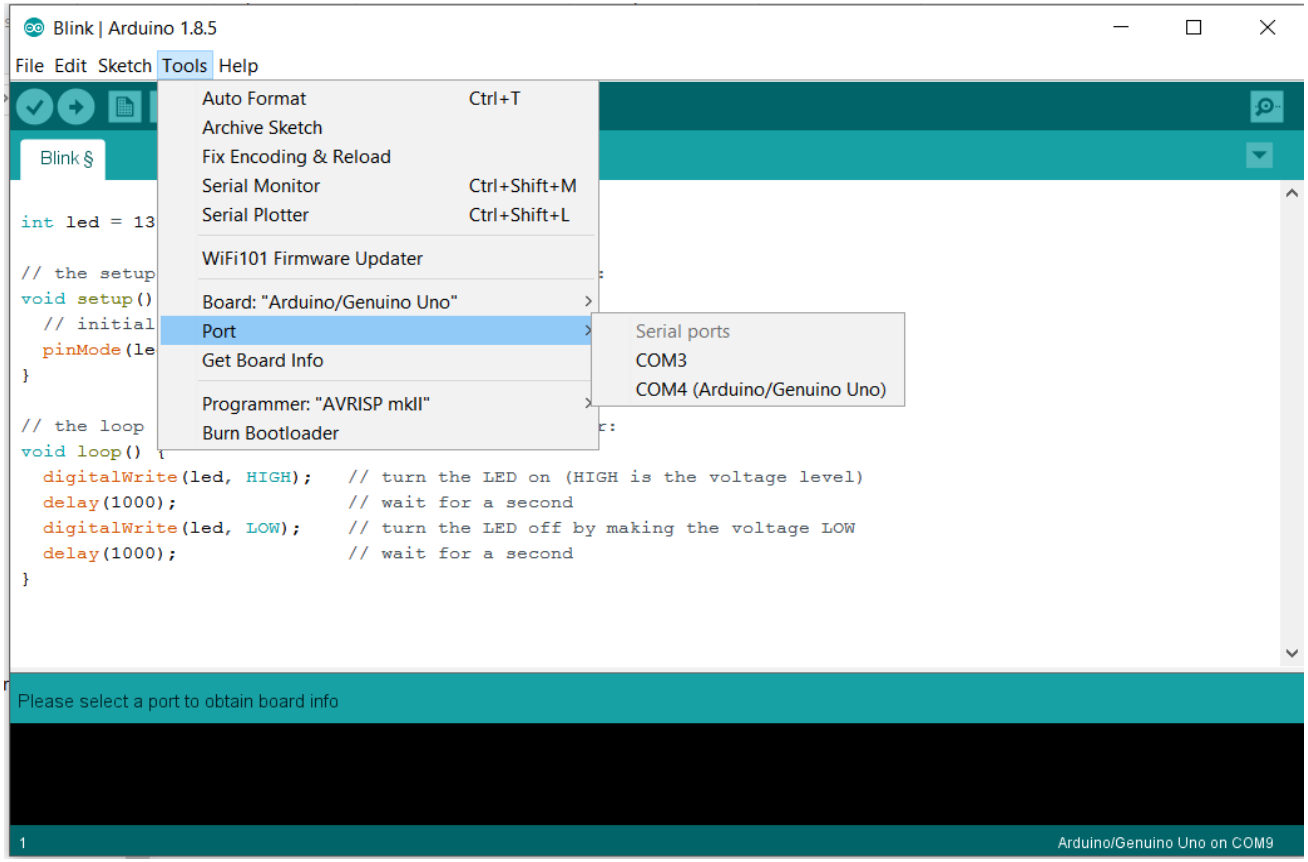
Serial Communication Port

- Windows



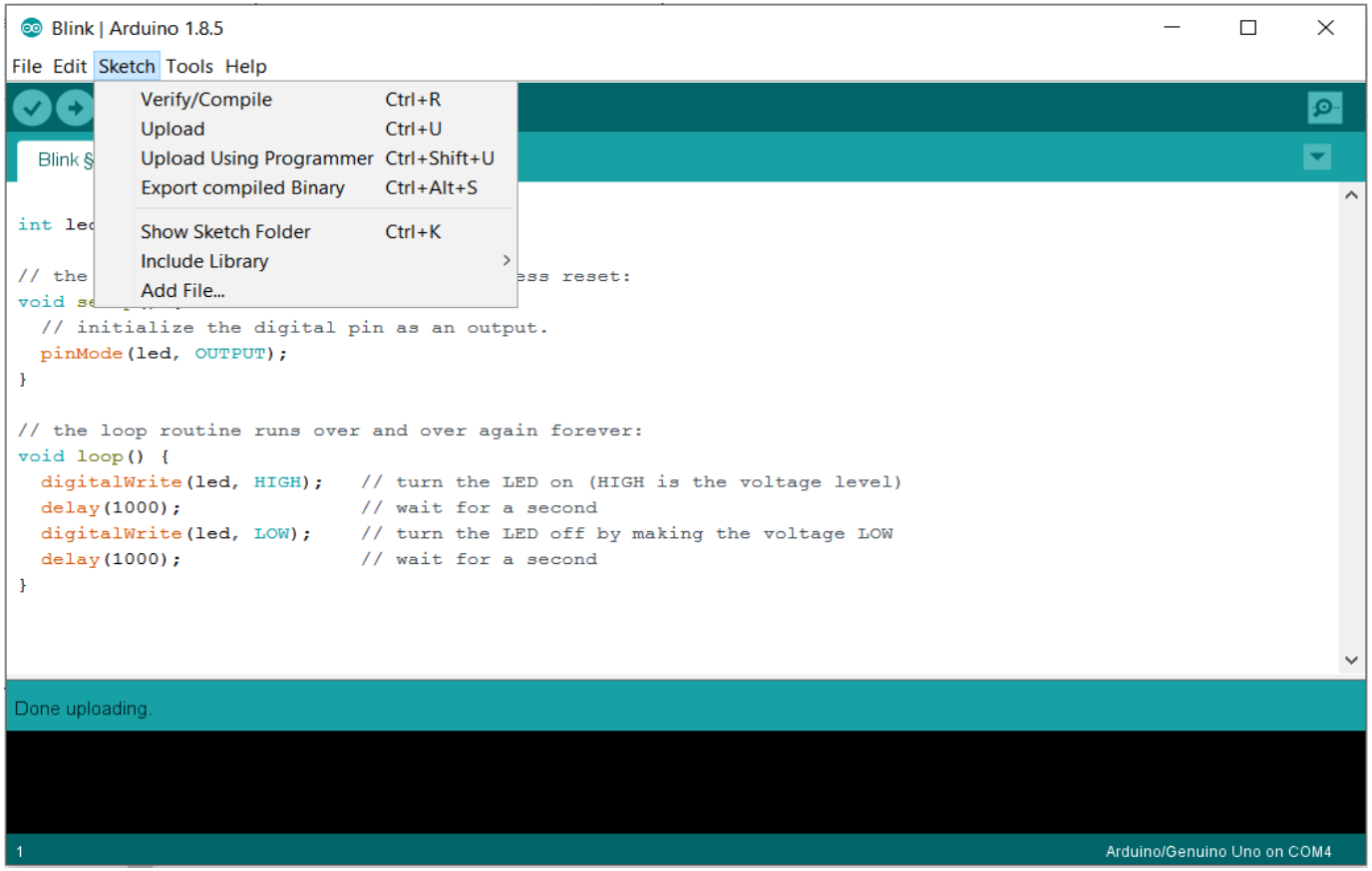
IoT Software

Serial Communication Port



IoT Software

Sketch



IoT Software

Sketch Compiling

Blink | Arduino 1.8.5

File Edit Sketch Tools Help

Blink §

```
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

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

Done compiling.

```
Sketch uses 942 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

1 Arduino/Genuino Uno on COM4

IoT Software

Sketch Uploading

Blink | Arduino 1.8.5

File Edit Sketch Tools Help

Blink §

```
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

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

Done uploading.

```
Sketch uses 942 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

1 Arduino/Genuino Uno on COM4

IoT Software

Sketch Structure

Blink | Arduino 1.8.5

File Edit Sketch Tools Help

Blink §

```
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

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

Setup

Loop

Done uploading.

```
Sketch uses 942 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
```

1 Arduino/Genuino Uno on COM4

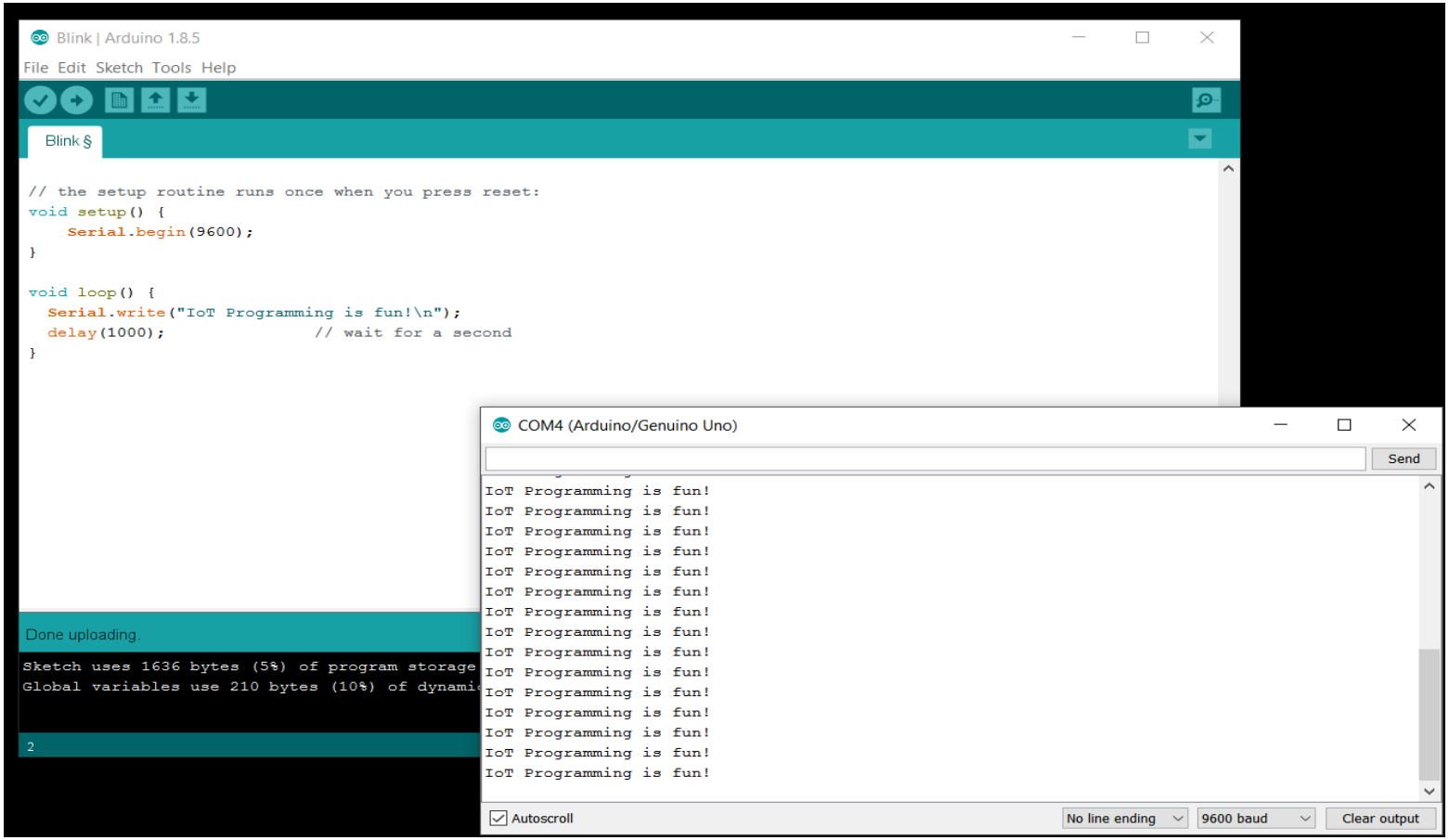
IoT Software

Example 1 (Text)

- Programing the Arduino to send “IoT Programming is fun!” as a text message through serial port every one second.

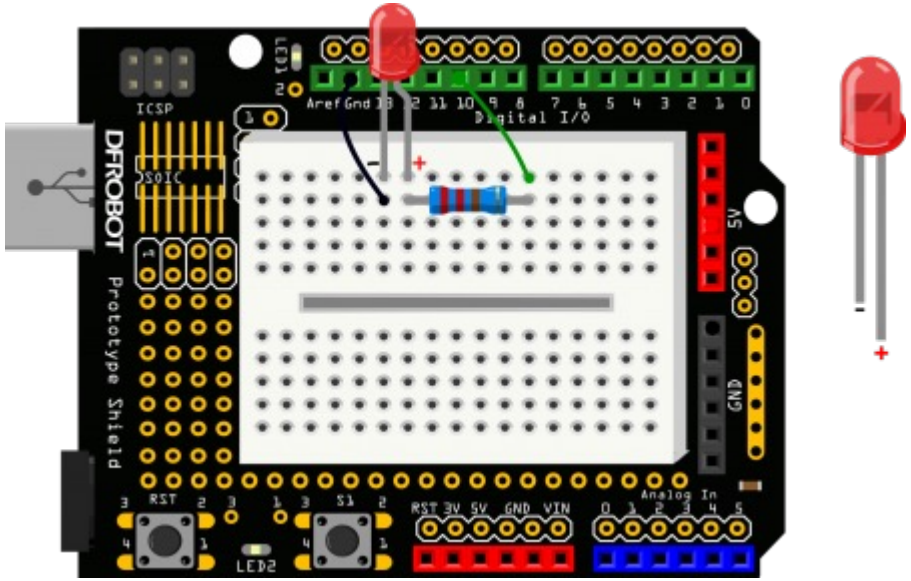
IoT Software

Example 1 (Text)



IoT Software

Example 2 (Blinking LED)



IoT Software

Example 2 (Blinking LED)

Blink | Arduino 1.8.5

File Edit Sketch Tools Help

Blink §

```
/*
 # Description:
 # Turns on an LED on for one second, then off for one second, repeatedly.
*/
int ledPin = 10;
void setup() {
        pinMode(ledPin, OUTPUT);
}
void loop() {
        digitalWrite(ledPin,HIGH);
        delay(1000);
        digitalWrite(ledPin,LOW);
        delay(1000);
}
```

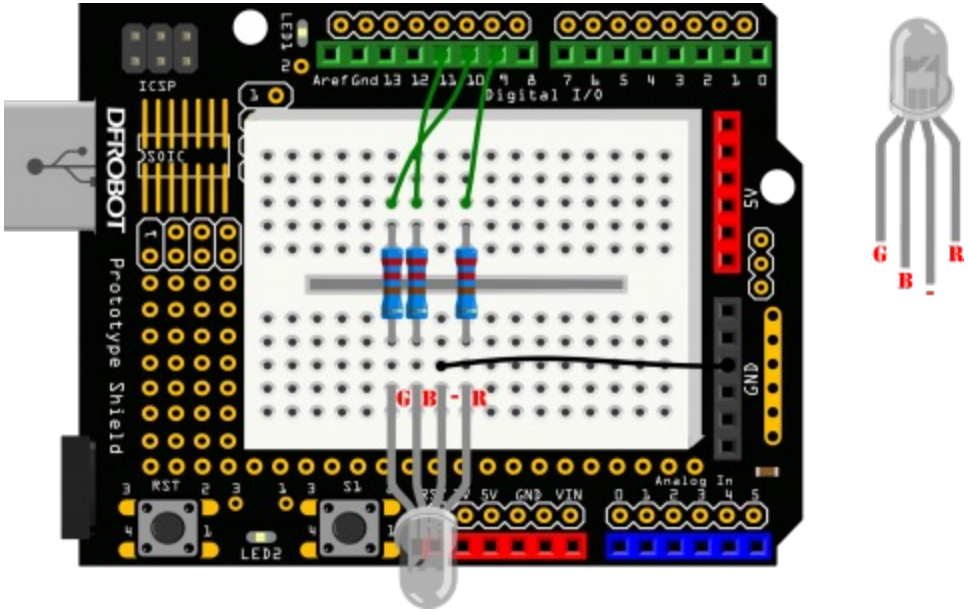
Done uploading.

```
Sketch uses 1636 bytes (5%) of program storage space. Maximum is 32256 bytes.
Global variables use 210 bytes (10%) of dynamic memory, leaving 1838 bytes for local variables. Maximum is 2048 bytes.
```

2 Arduino/Genuino Uno on COM4

IoT Software

Example 3 (Color RGB LED LED)



IoT Software

Example 3 (Color RGB LED LED)

Blink | Arduino 1.8.5

File Edit Sketch Tools Help

Blink §

```
/*
   RGB LED
*/
int redPin = 9;     // the pin that the red LED is attached to
int greenPin = 10; // the pin that the green LED is attached to
int bluePin = 11;  // the pin that the blue LED is attached to

void setup(){
     pinMode(redPin, OUTPUT);
     pinMode(greenPin, OUTPUT);
     pinMode(bluePin, OUTPUT);

}
void loop(){
    // call the function to change the colors of LED randomly.
    colorRGB(random(0,255),random(0,255),random(0,255)); //R:0-255 G:0-255 B:0-255
    delay(1000);
}

void colorRGB(int red, int green, int blue){
     analogWrite(redPin,constrain(red,0,255));
     analogWrite(greenPin,constrain(green,0,255));
     analogWrite(bluePin,constrain(blue,0,255));
}
```

Done compiling.

```
Sketch uses 1786 bytes (5%) of program storage space. Maximum is 32256 bytes.
Global variables use 13 bytes (0%) of dynamic memory, leaving 2035 bytes for local variables. Maximum is 2048 bytes.
```

24 Arduino/Genuino Uno on COM4

Questions?