PROIECT

Concursul de Robotică RoboSmart al Universității din Pitești

PENTRU ELEVII CLASELOR IX – XII

Sponsori

TEXTRON FLEET MANAGEMENT

Facultatea de Electronică, Comunicații și Calculatoare

Concurs de robotică

RoboSmart

Cursul 3

Arduino. Structure. Components. Arduino IDE

Prezinta:

Gabriel Iana

Cosmin Stirbu

Ionescu Valeriu

Sponsori:

TEXTRON FLEET MANAGEMENT

Outline

- Arduino versions
- Communication Protocols: I^2C, SPI
- Interacting with the environment
- Arduino Uno
- IDE programming tools
- Examples

Arduino boards

In 2013: 700,000 official boards!

Arduino Micro

Arduino Uno

Arduino Due

Do It Yourself (DIY) projects!
ARM (**A**dvanced **R**ISC **M**achine) architecture....

License:
GNU Lesser General Public License (LGPL) or
GNU General Public License (GPL)

https://www.arduino.cc
https://ardushop.ro

Arduino boards – types

ENTRY LEVEL	UNO, LEONARDO, 101, ESPLORA, MICRO, NANO, MINI, MKR2UNO ADAPTER, STARTER KIT, LCD SCREEN
ENHANCED FEATURES	MEGA, ZERO, DUE, MEGA ADK, M0, M0 PRO, MKR ZERO, MOTOR SHIELD, USB HOST SHIELD, PROTO SHIELD, MKR PROTO SHIELD, 4 RELAYS SHIELD, MEGA PROTO SHIELD, MKR RELAY PROTO SHIELD, ISP, USB2SERIAL MICRO, USB2SERIAL CONVERTER
INTERNET OF THINGS	YÚN, ETHERNET, TIAN, INDUSTRIAL 101, LEONARDO ETH, MKR FOX 1200, MKR WAN 1300, MKR GSM 1400, MKR1000, YUN MINI, YÚN SHIELD, WIRELESS SD SHIELD, WIRELESS PROTO SHIELD, ETHERNET SHIELD V2, GSM SHIELD V2, MKR IoT BUNDLE
EDUCATION	CTC 101
WEARABLE	GEMMA, LILYPAD ARDUINO USB, LILYPAD ARDUINO MAIN BOARD, LILYPAD ARDUINO SIMPLE, LILYPAD ARDUINO SIMPLE SNAP
3D PRINTING	MATERIA 101

Entry level


ARDUINO UNO


ARDUINO LEONARDO


ARDUINO 101


ARDUINO ESPLORA


ARDUINO MICRO


ARDUINO NANO

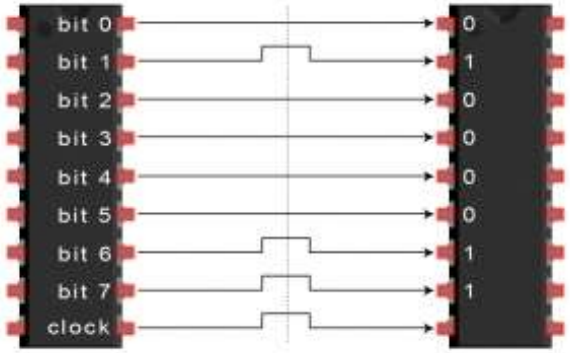

ARDUINO MINI

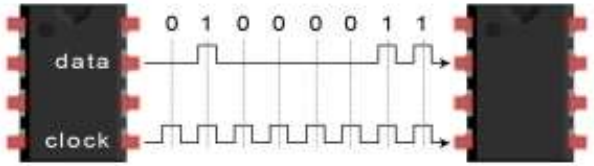

MKR2UNO ADAPTER
MKR2UNO ADAPTER


STARTER KIT
ARDUINO
ARDUINO
STARTER KIT

Communication protocols

- Do you ever think about how the two devices talk to each other?
- Communication between electronic devices is like communication between humans. Both sides need to speak the same language. In electronics, these languages are called *communication protocols*.
- In DIY projects:
  - Parallel communication
    - High speed timing problems
  - Serial communication
    - Inter-Integrated Circuit (I2C)
    - Serial Peripheral Interface (SPI)



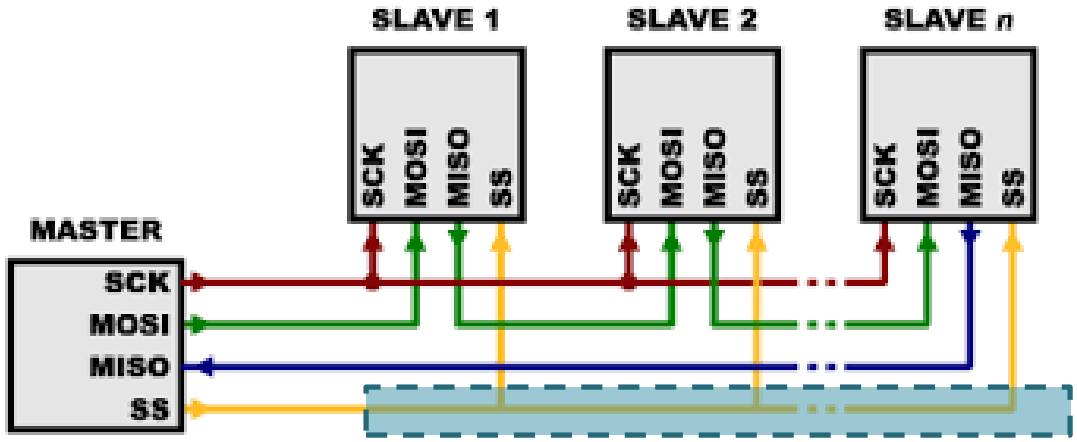


Serial Peripheral Interface

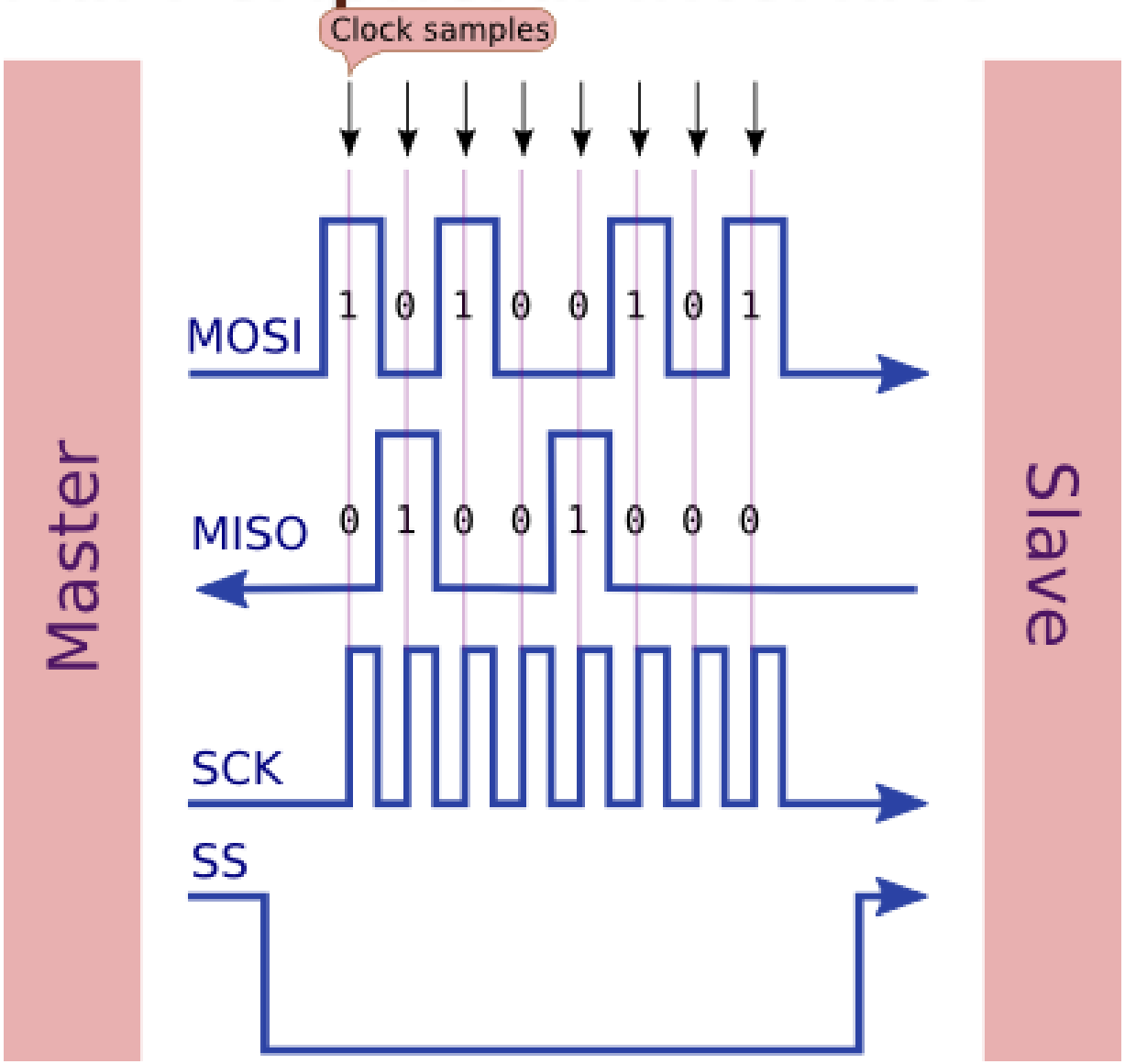
Used in: SD cards, LCD and RFID readers

The SPI bus specifies four logic signals:
•**MISO/SDI** (Master In Slave Out) - The Slave line for sending data to the master,
•**MOSI/SDO** (Master Out Slave In) - The Master line for sending data to the peripherals,
•**SCK** (Serial Clock) - The clock pulses which synchronize data transmission generated by the master
•**SS**: Slave Select - An independent SS signal from master for each slave device!

Single master - multiple slaves!



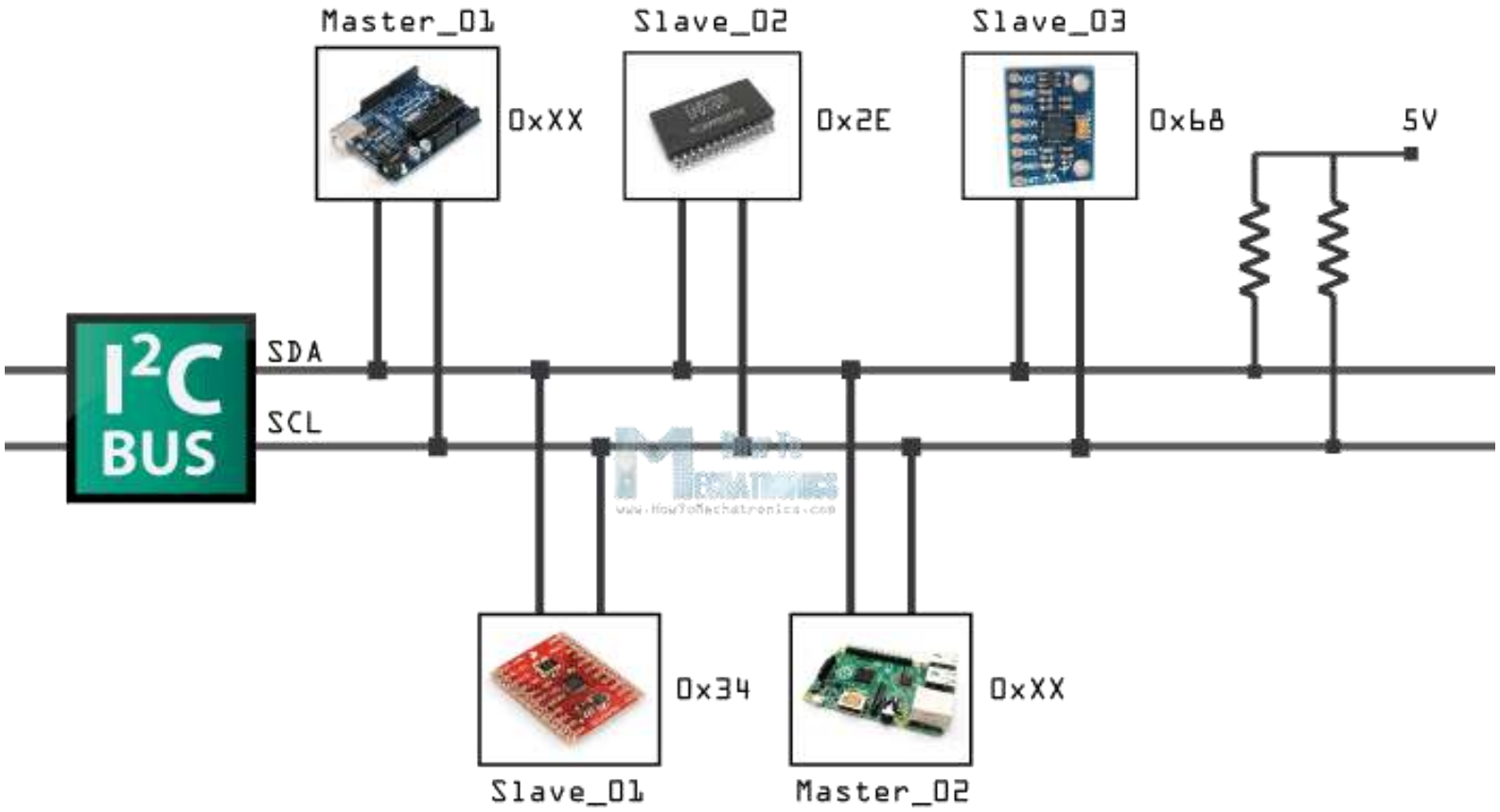
Serial Peripheral Interface



Enhanced features

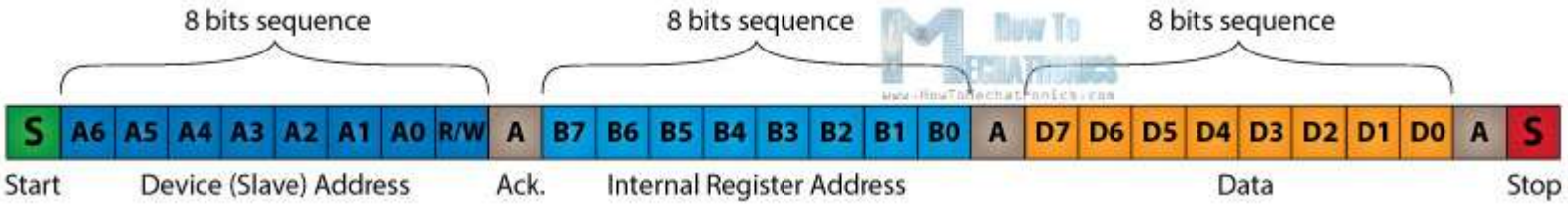
Arduino boards

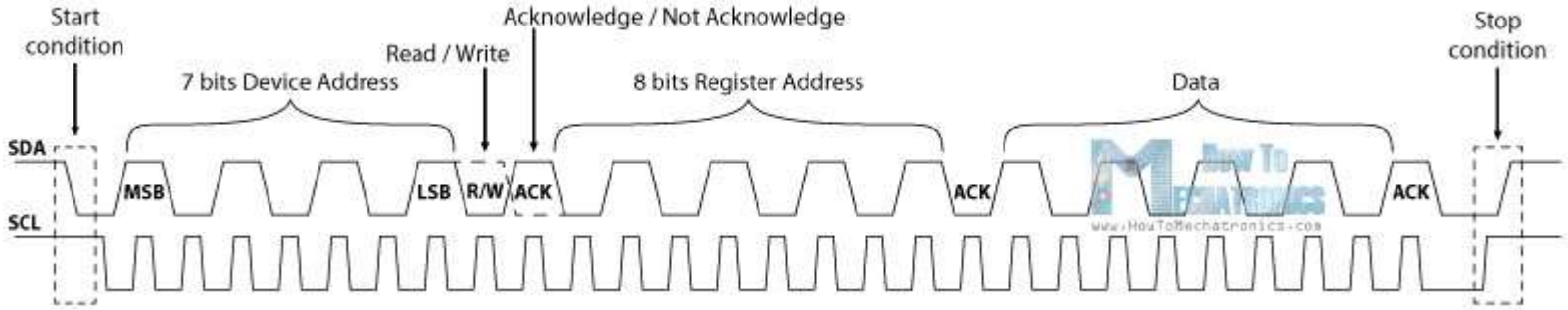
ARDUINO MICRO



Arduino boards

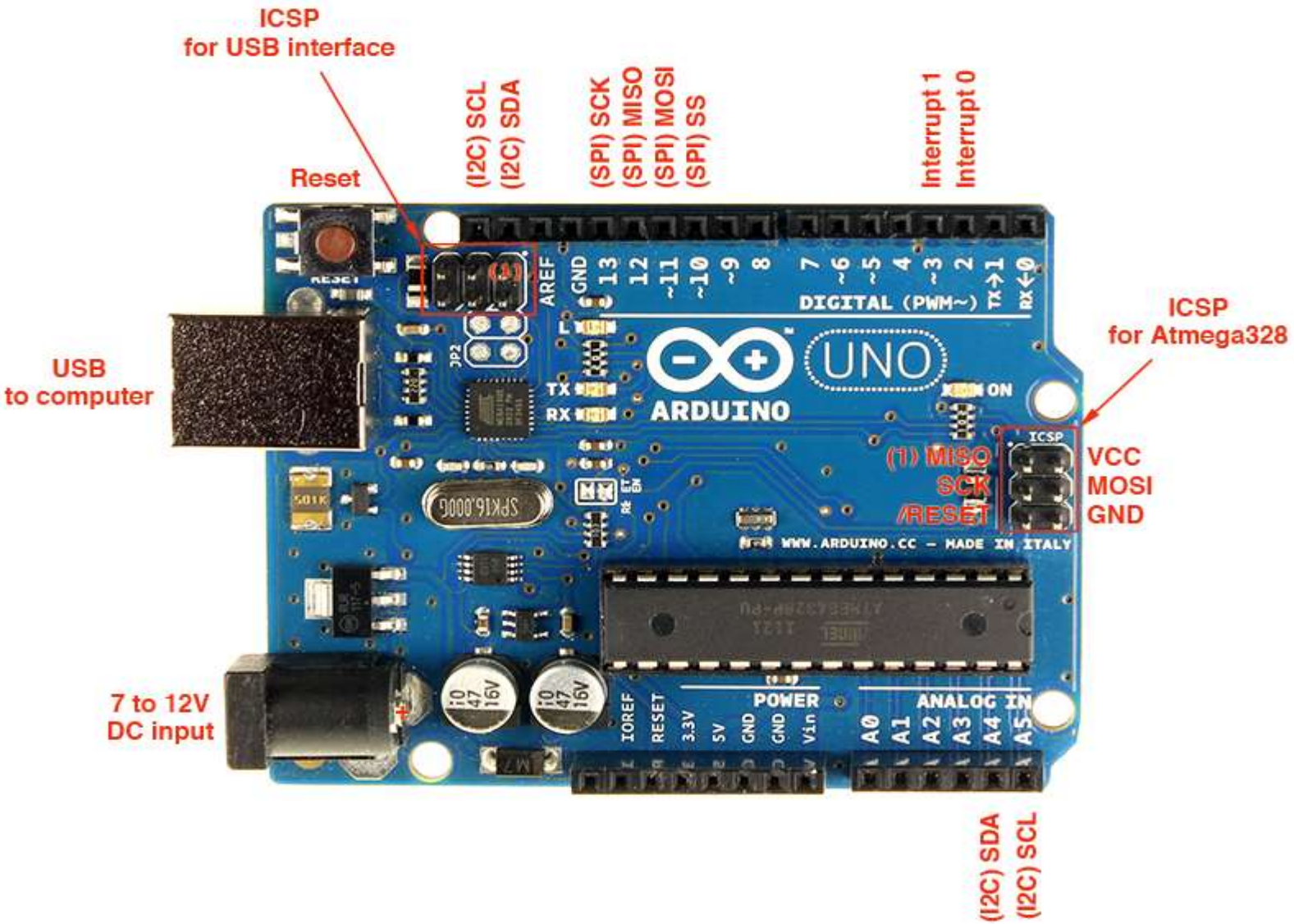
ARDUINO MICRO



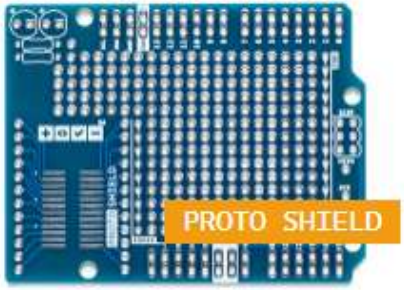


https://www.arduino.cc/en/Reference/Wire

Serial Peripheral Interface



Enhanced features


PROTO SHIELD


MKR PROTO SHIELD


ARDUINO 4
RELAYS SHIELD


ARDUINO MEGA
PROTO SHIELD

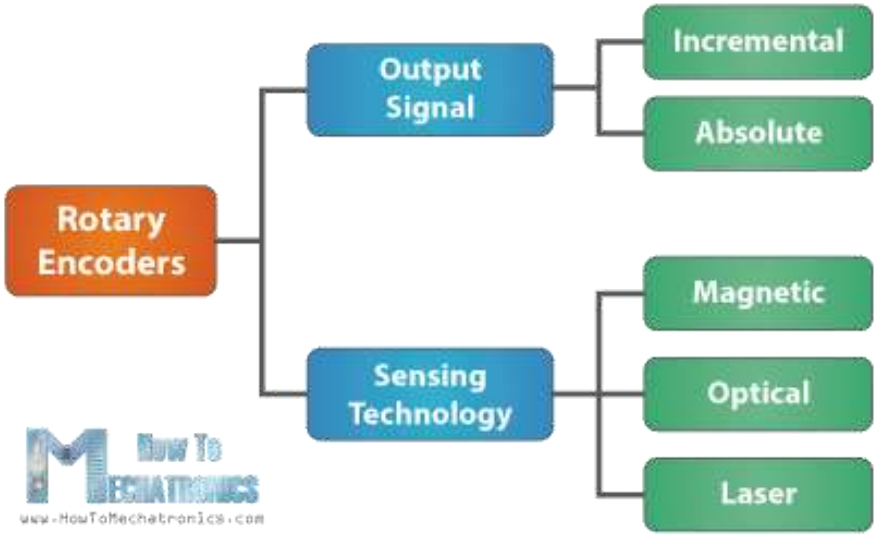

MKR RELAY PROTO SHIELD

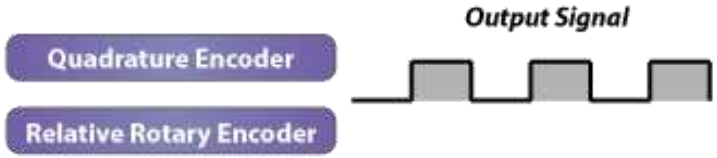

ARDUINO ISP


ARDUINO USB2
SERIAL MICRO

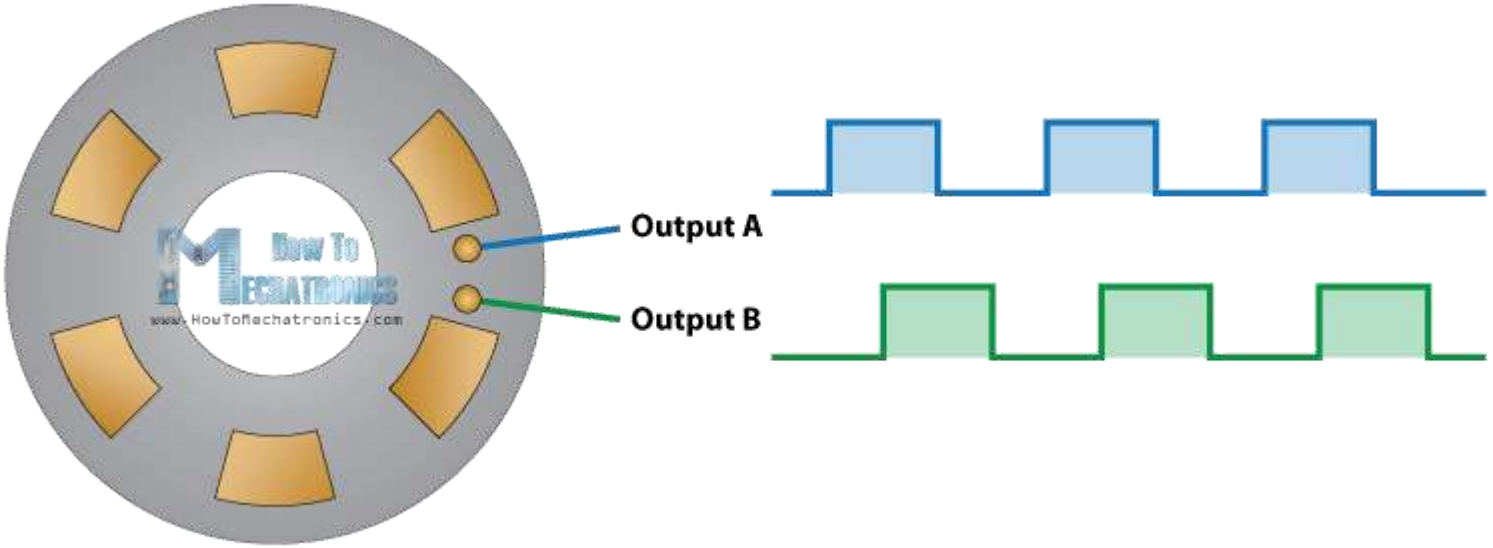

ARDUINO USB2SERIAL
CONVERTER

Rotary encoder





Rotary encoder operation



Internet of Things - IOT

ARDUINO YÚN

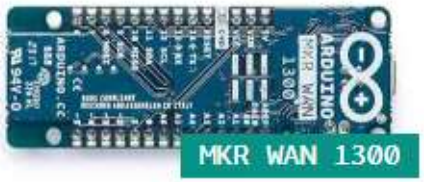
ARDUINO ETHERNET

ARDUINO TIAN


ARDUINO INDUSTRIAL 101


ARDUINO LEONARDO ETH

MKR FOX 1200

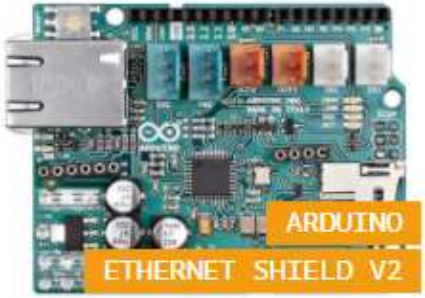


MKR WAN 1300

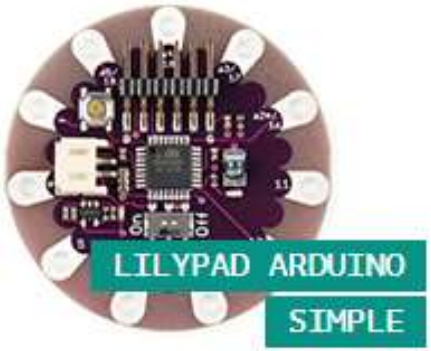
MKR GSM 1400

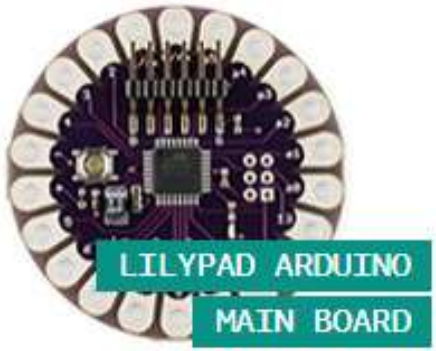
ARDUINO MKR1000

Internet of Things - IOT



Wearable

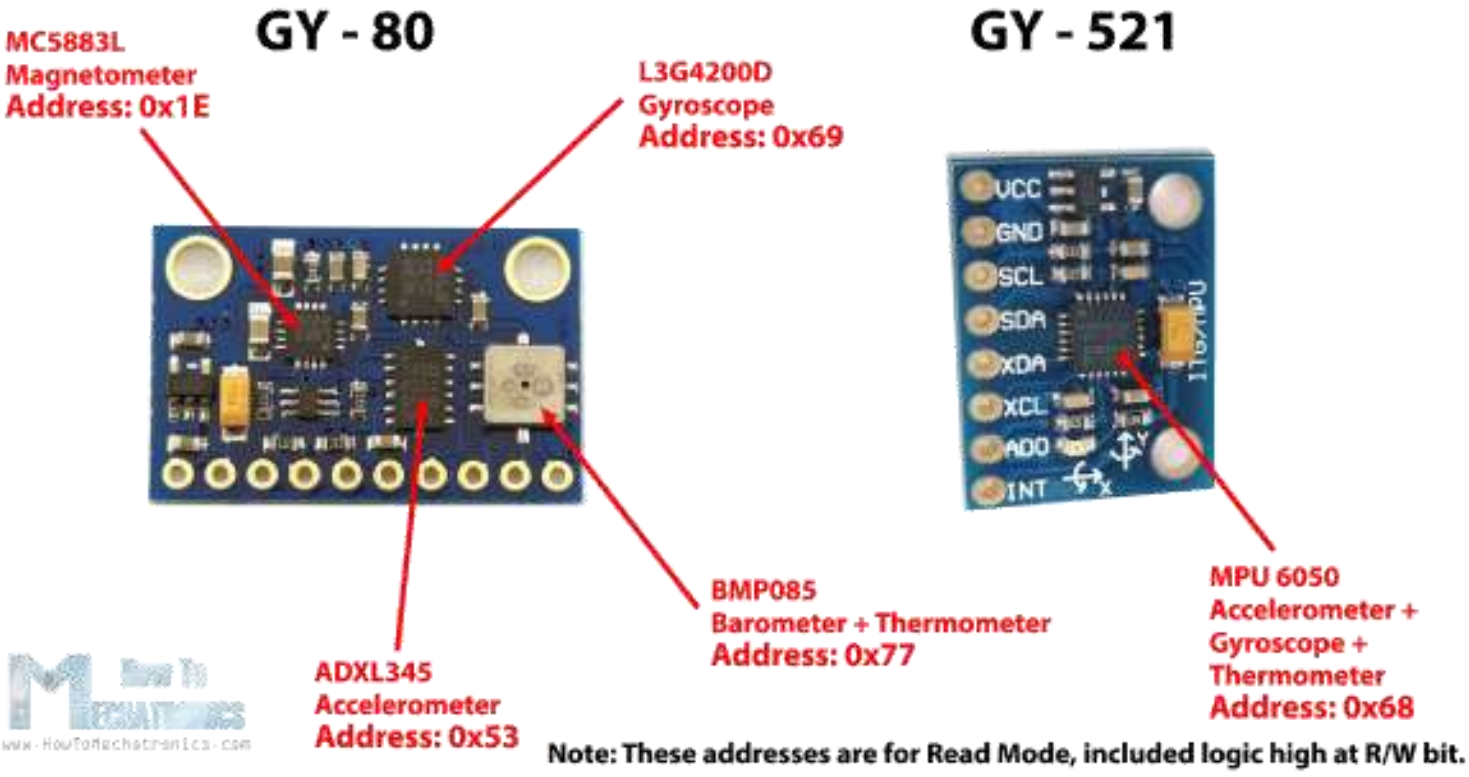




https://www.arduino.cc/en/Main/Products

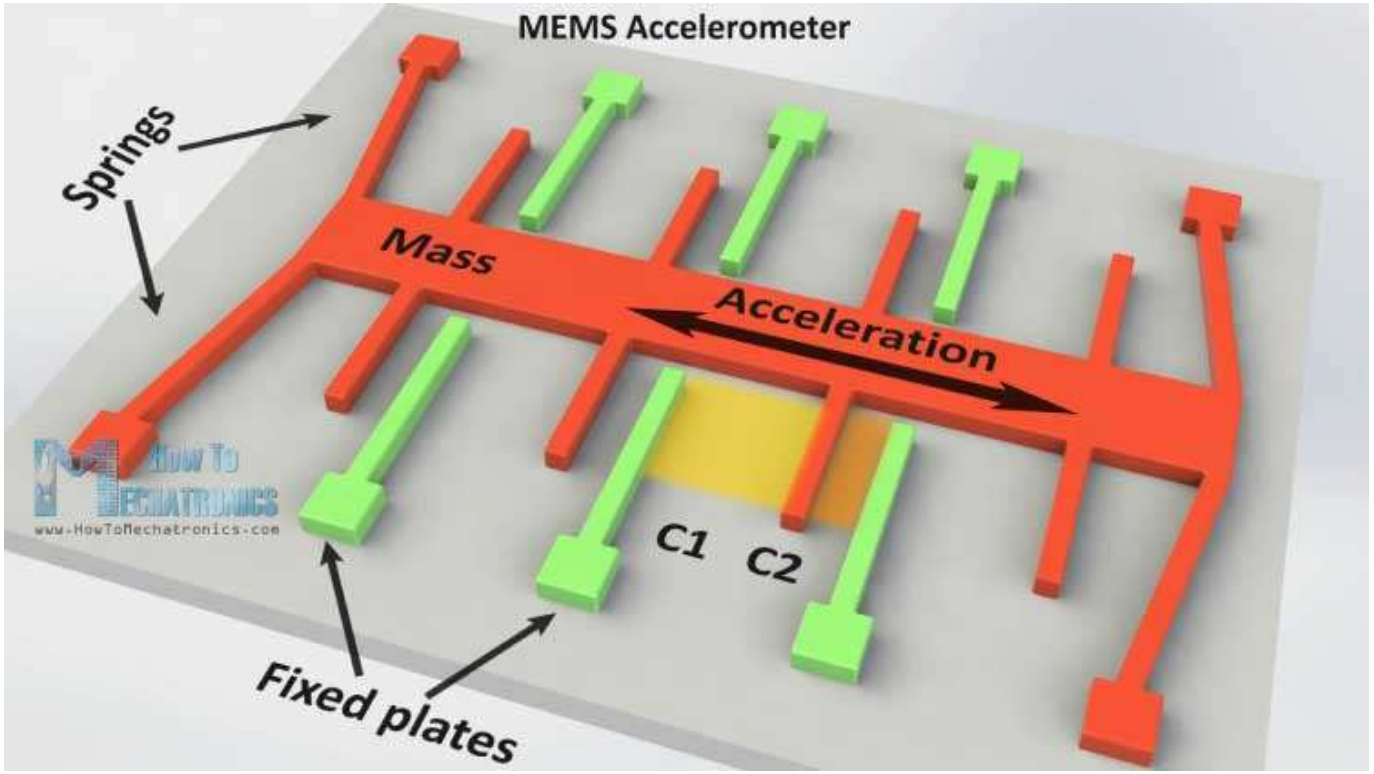
Arduino boards

ARDUINO MICRO



Sensors

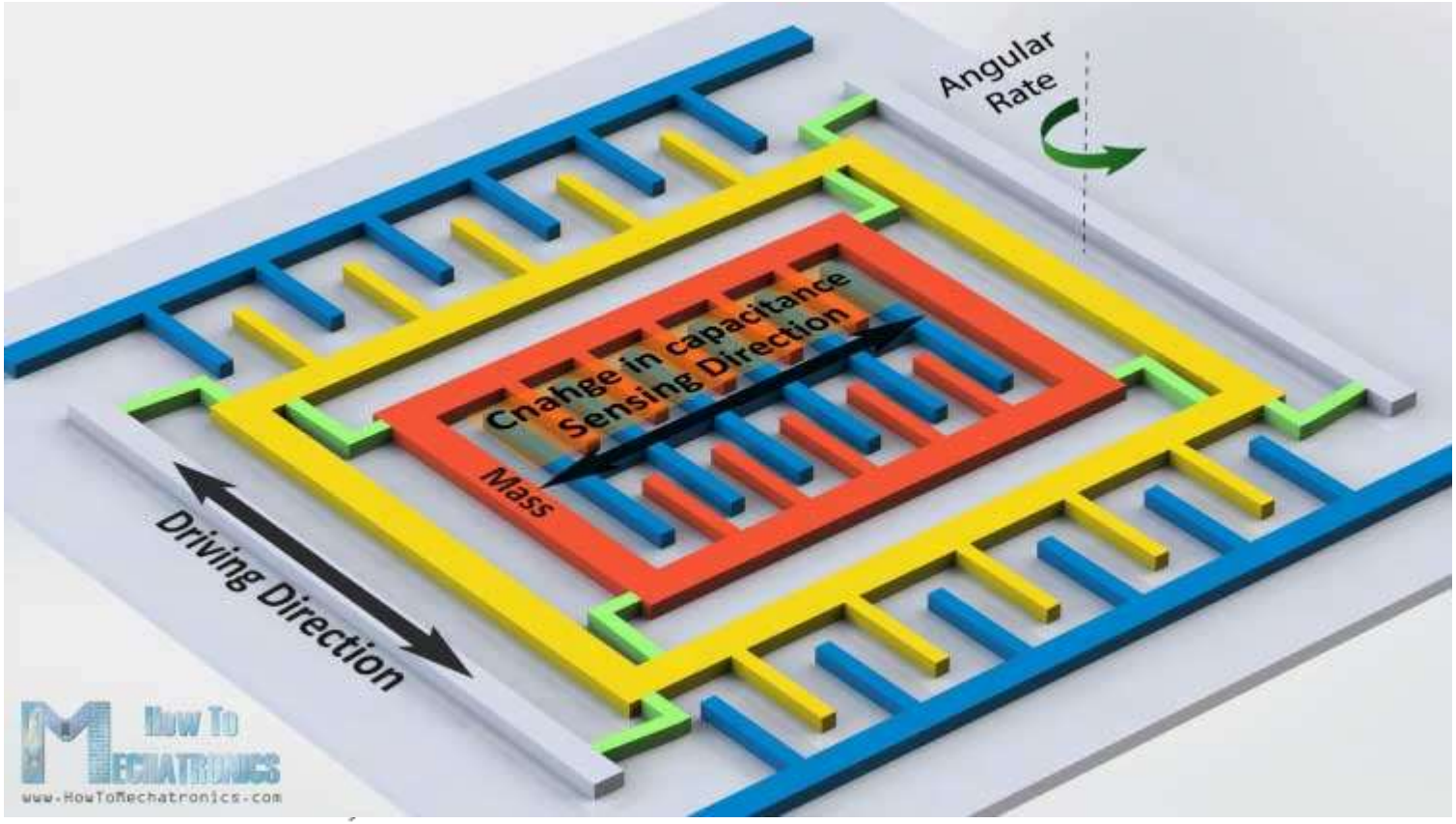
Micro electro-mechanical systems (MEMS) Accelerometer



Sensors

Johann Gottlieb Friedrich von Bohnenberger

MEMS giroscope

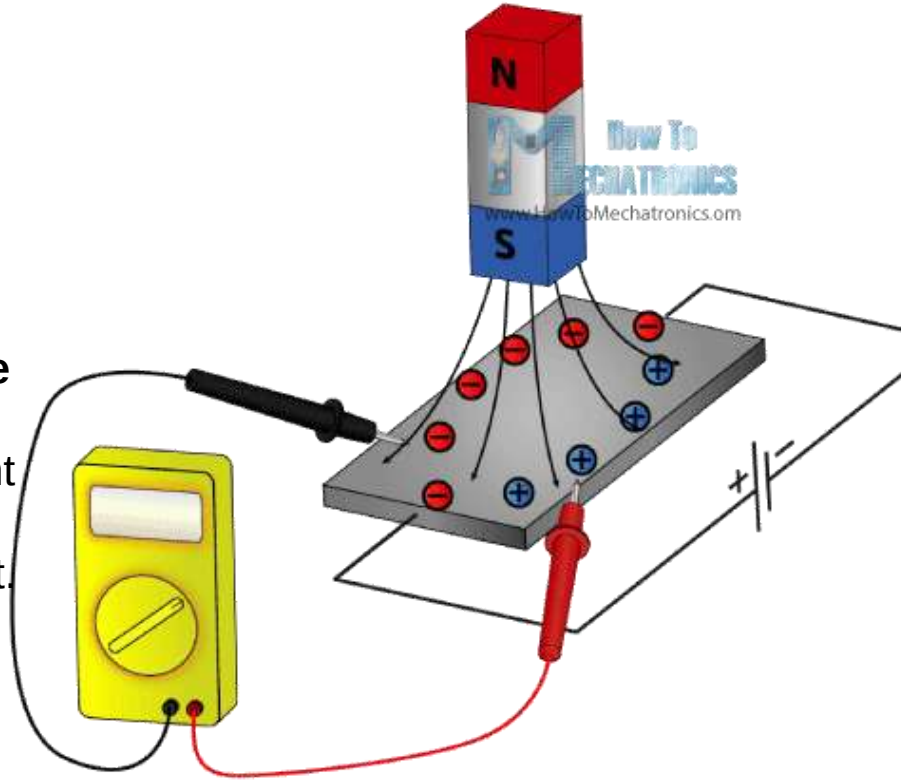


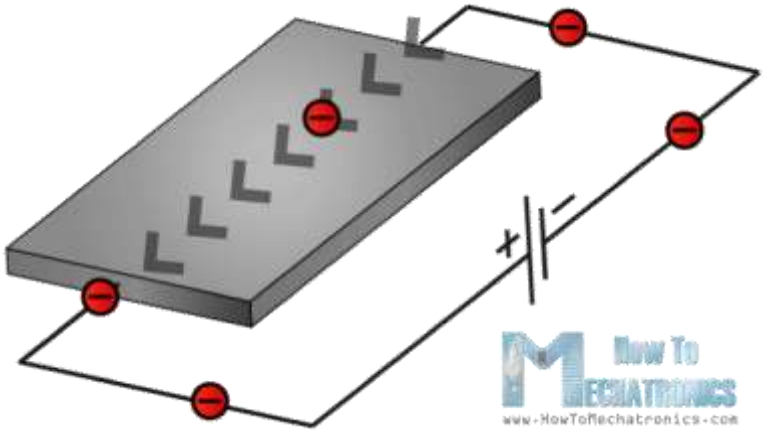
Sensors

Magnetic sensors

Hall Effect

is the **production of a voltage difference** (the Hall voltage) across an electrical conductor, transverse to an electric current in the conductor and to an applied magnetic field perpendicular to the current.





Magneto Resistive Effect

is the tendency of a material to change the value of its **electrical resistance** in an externally-applied magnetic field

Arduino Uno R3

The hardware consists of a simple open source hardware board designed around an 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM.

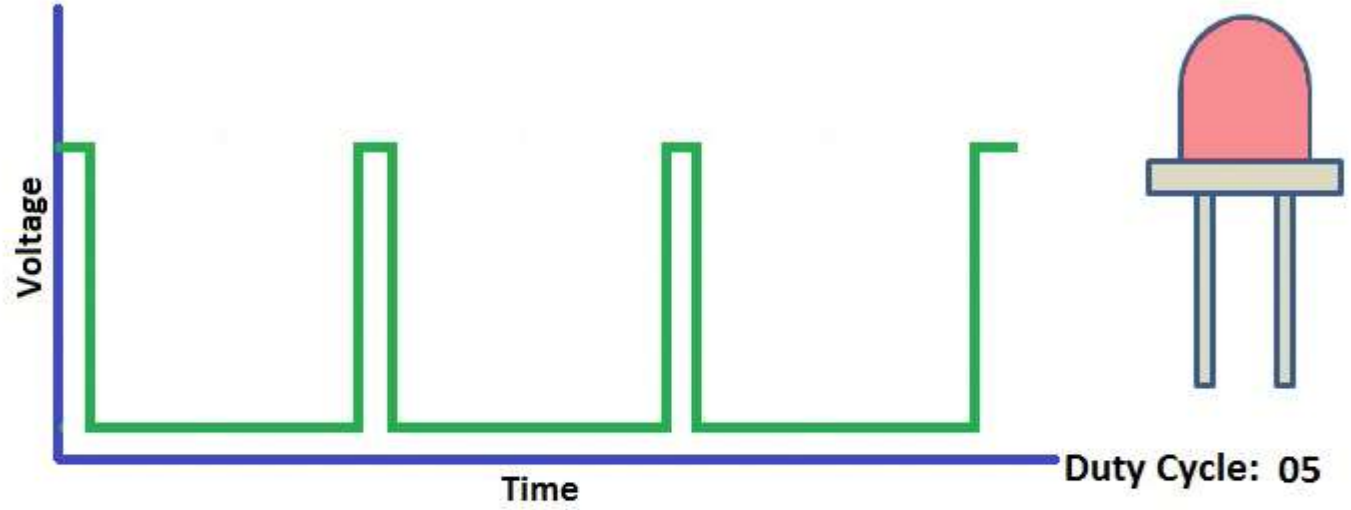
The software consists of a standard programming language compiler and a boot loader that executes on the microcontroller

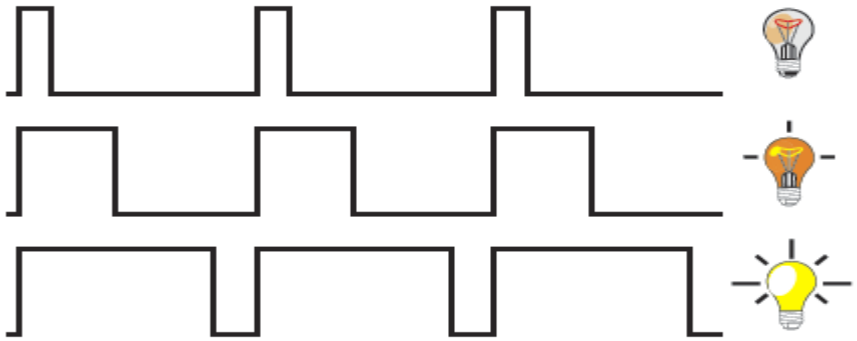
Arduino Uno R3

- Microcontroller: ATmega328
- Operating Voltage: 5V
- Input Voltage (recommended):7-12V
- Digital I/O Pins: 14 (6 provide **PWM** output)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40mA/ 3.3V
- Flash Memory: 32 KB (ATmega328)
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz

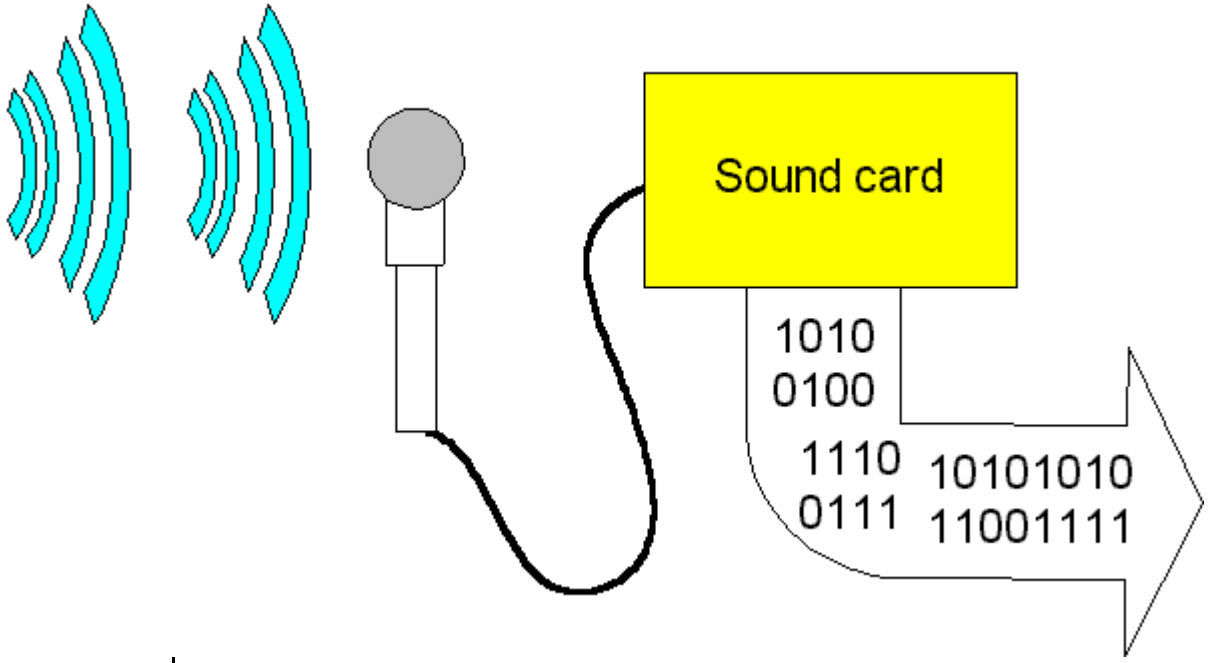
What mean PWM?

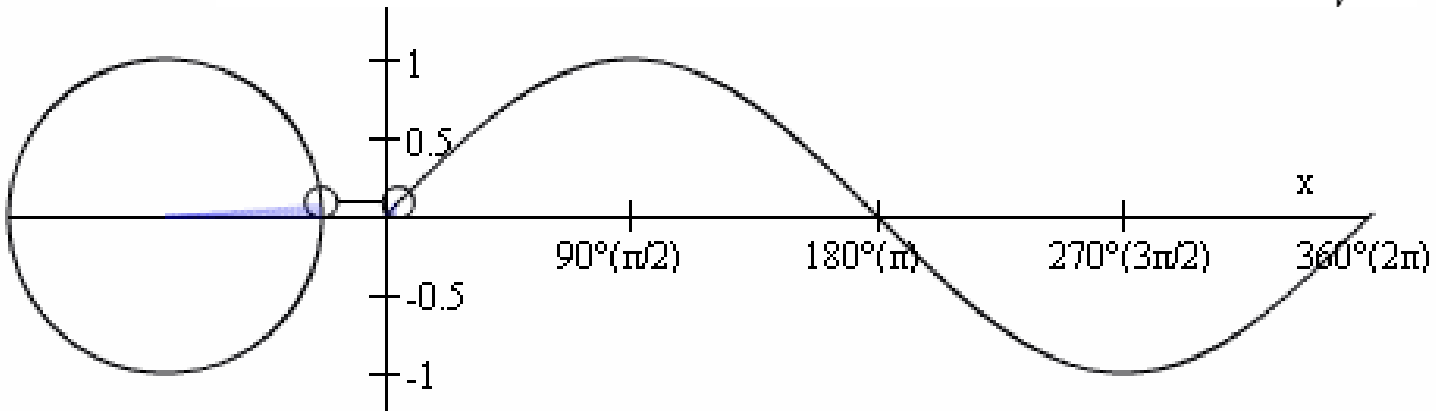
PWM – Pulse With Modulation



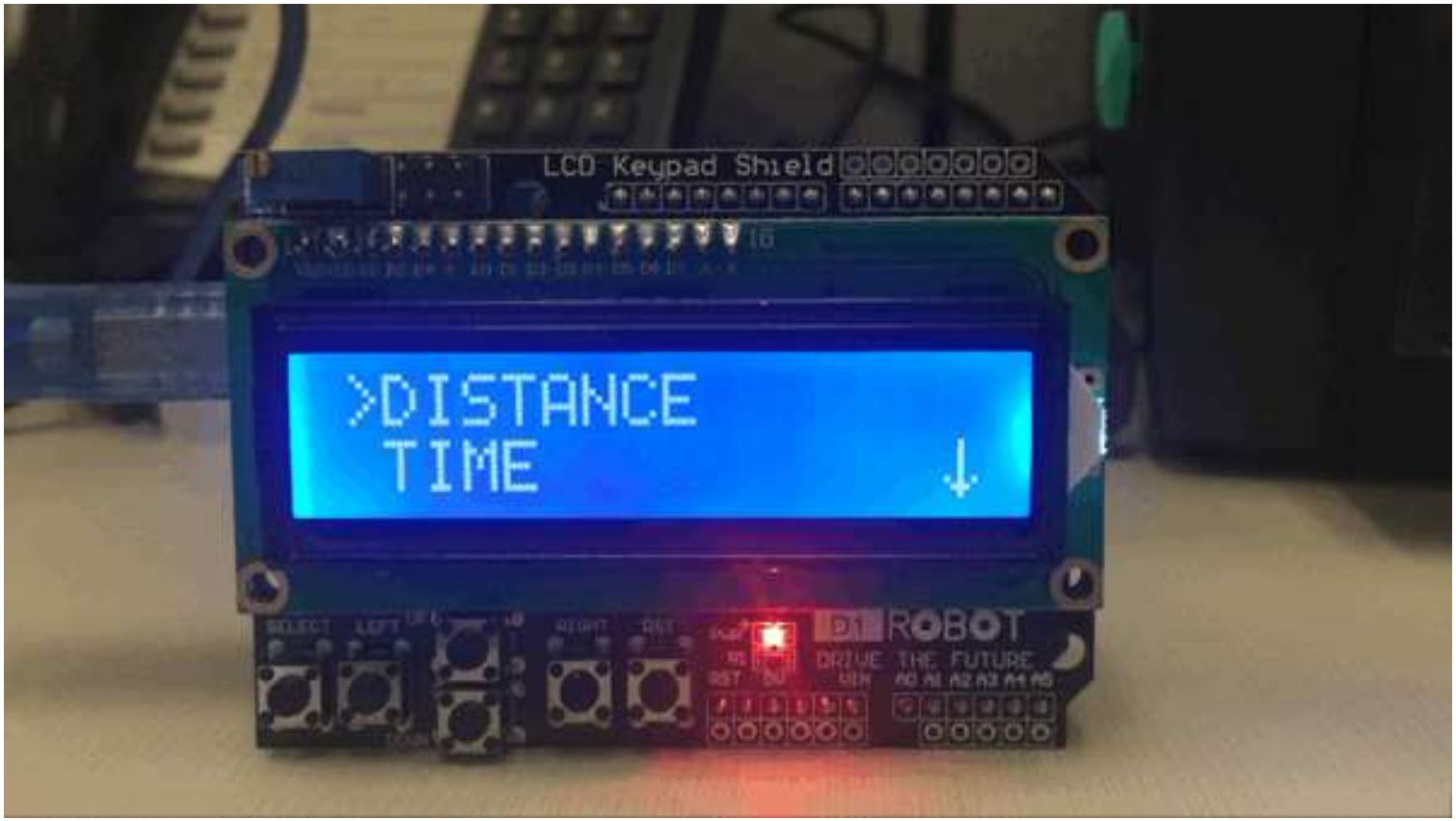


Analog inputs?





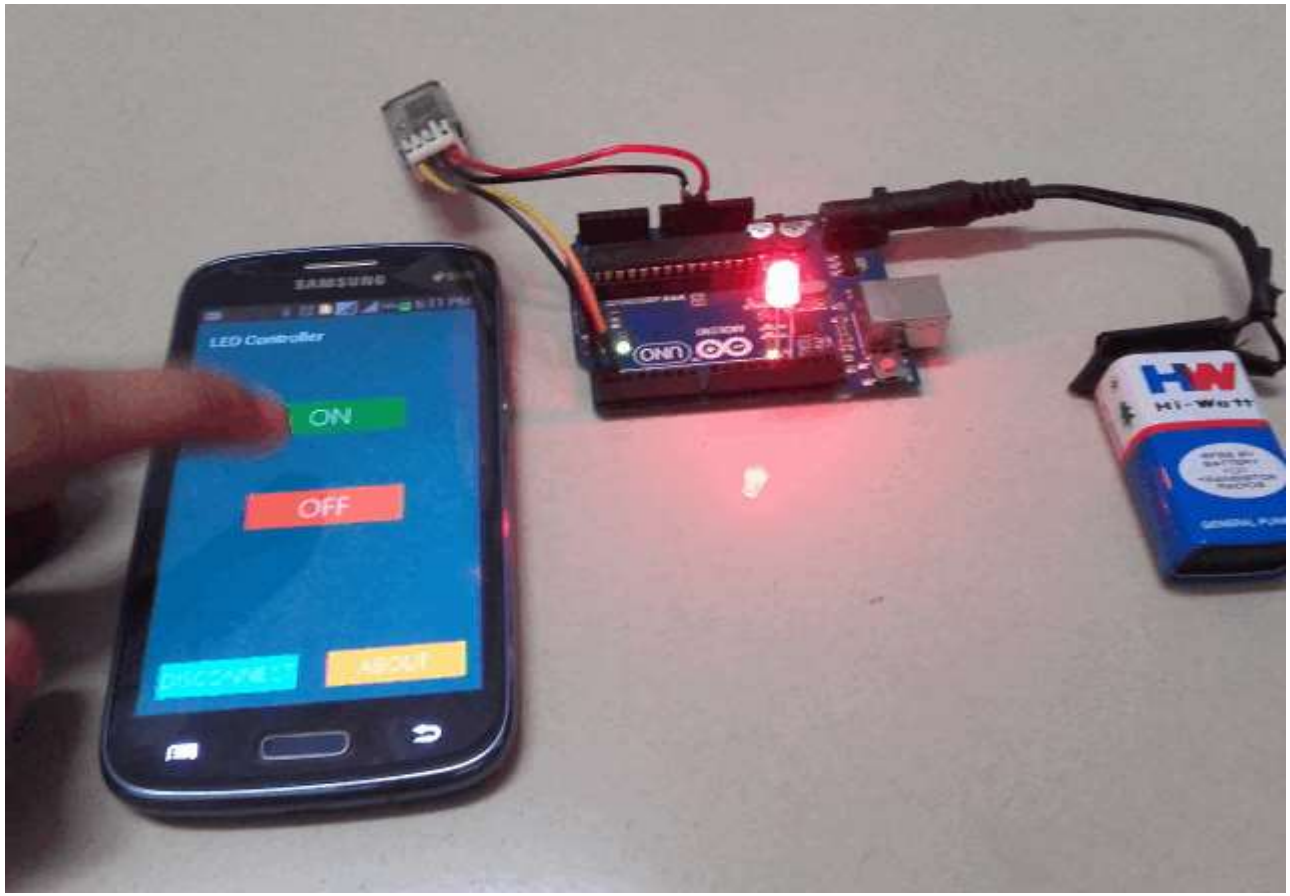
Applications



LCD display

http://www.instructables.com/id/Arduino-Uno-Menu-Template/

Applications

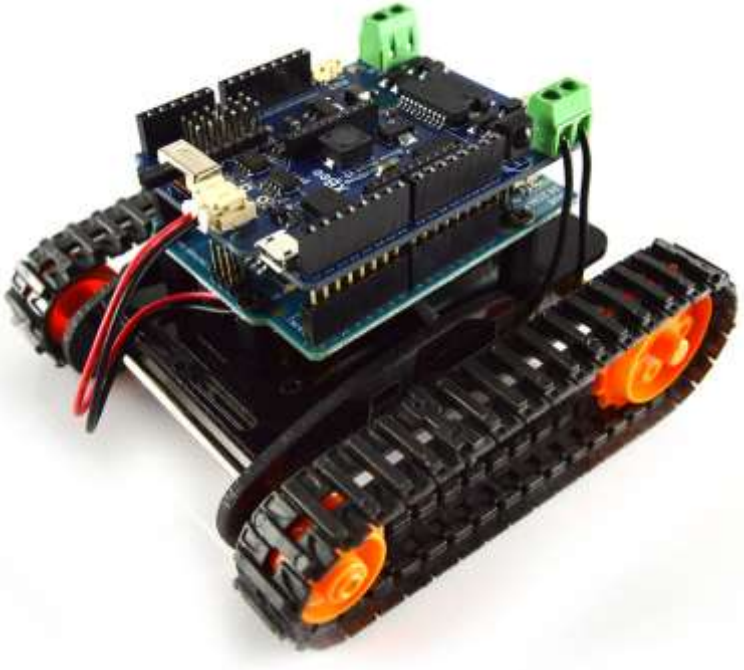


Bluetooth communication

https://diyhacking.com/arduino-bluetooth-basics/

Applications

Arduino uno radar scope



https://thesolaruniverse.wordpress.com/2017/03/24/an-arduino-uno-tft-shield-radar-scope/

Recommended beginner applications

Very simple applications

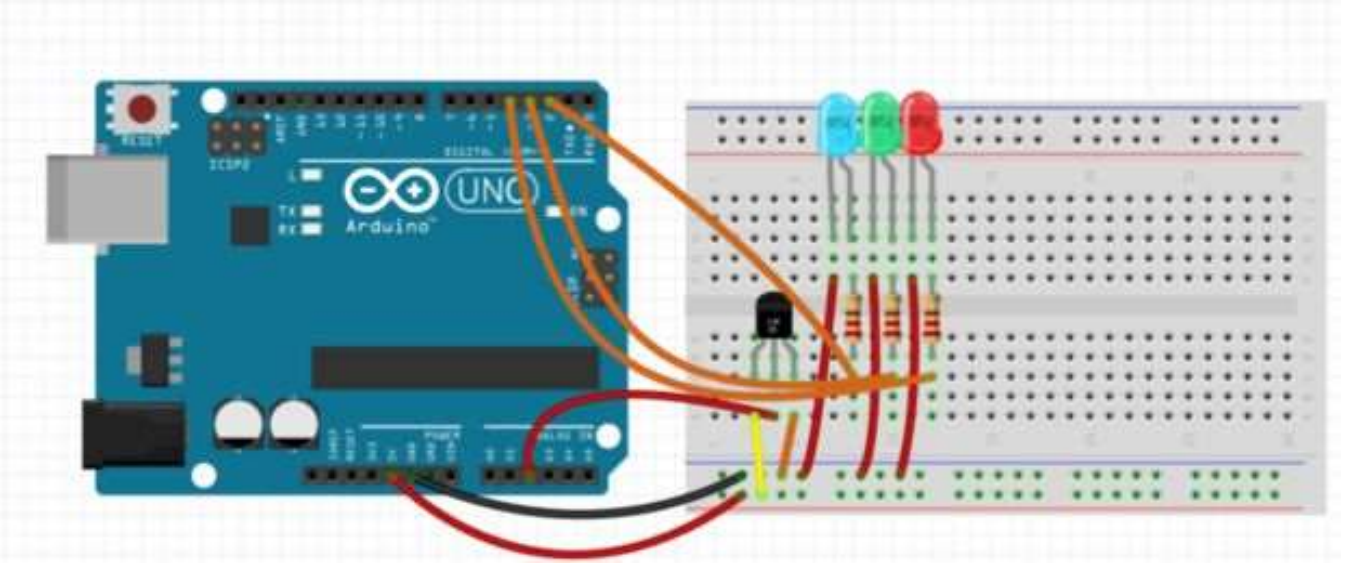
- Turn on of off a LED
- Turn on a LED until a button is pressed
- Cyclically turn on/off a led at each button pressing
- Blink a led with 1Hz frequency
- Blink a LED with 10Hz only when the button is pressed

Medium difficulty applications

- Using the GPIO interrupts turn on the led at button pressing
- Using timer interrupts blink a led with 10Hz frequency
- Using GPIO and timer interrupts blink the led with 10Hz when the button is pressed and 20Hz when the button is released

What You Need for a Working System

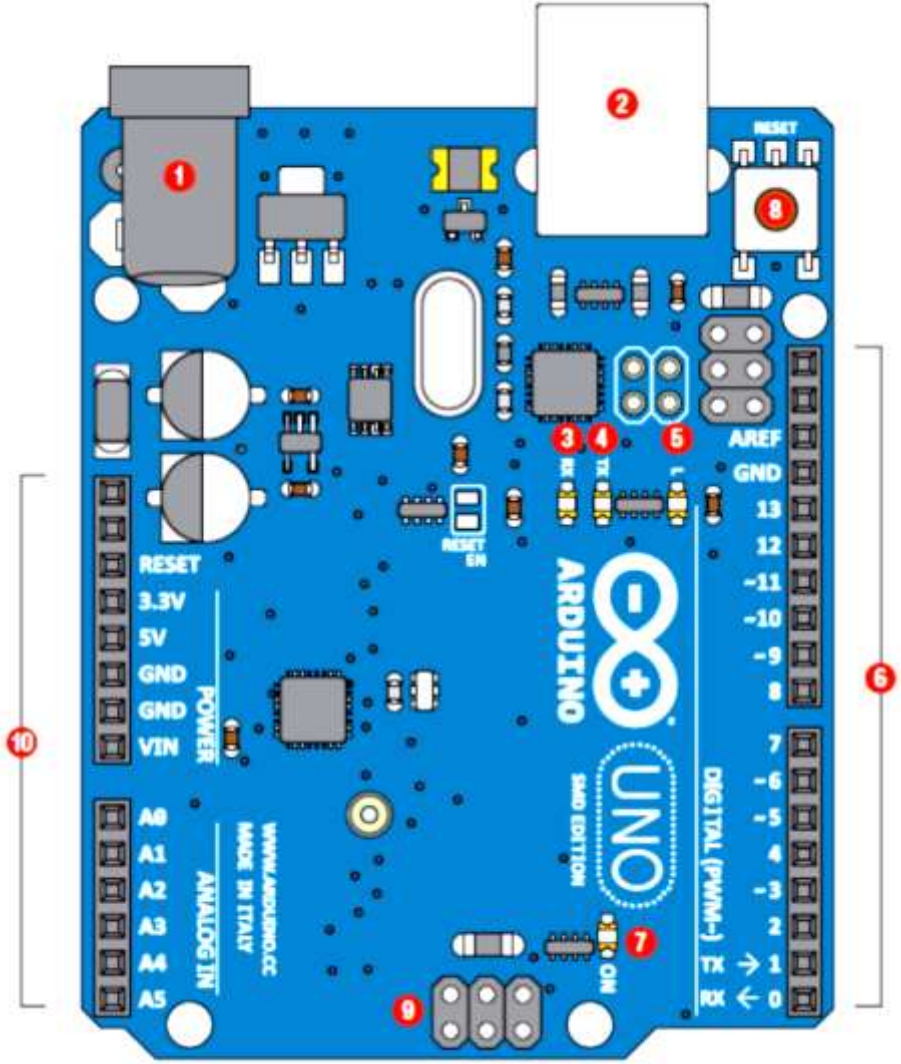
- Arduino UNO development board
- USB programming cable (A to B)
- 3. 9V battery or external power supply (for stand-alone operation)
- Solderless breadboard for external circuits, and 22 g solid wire for connections
- Host PC running



What You Need for a Working System

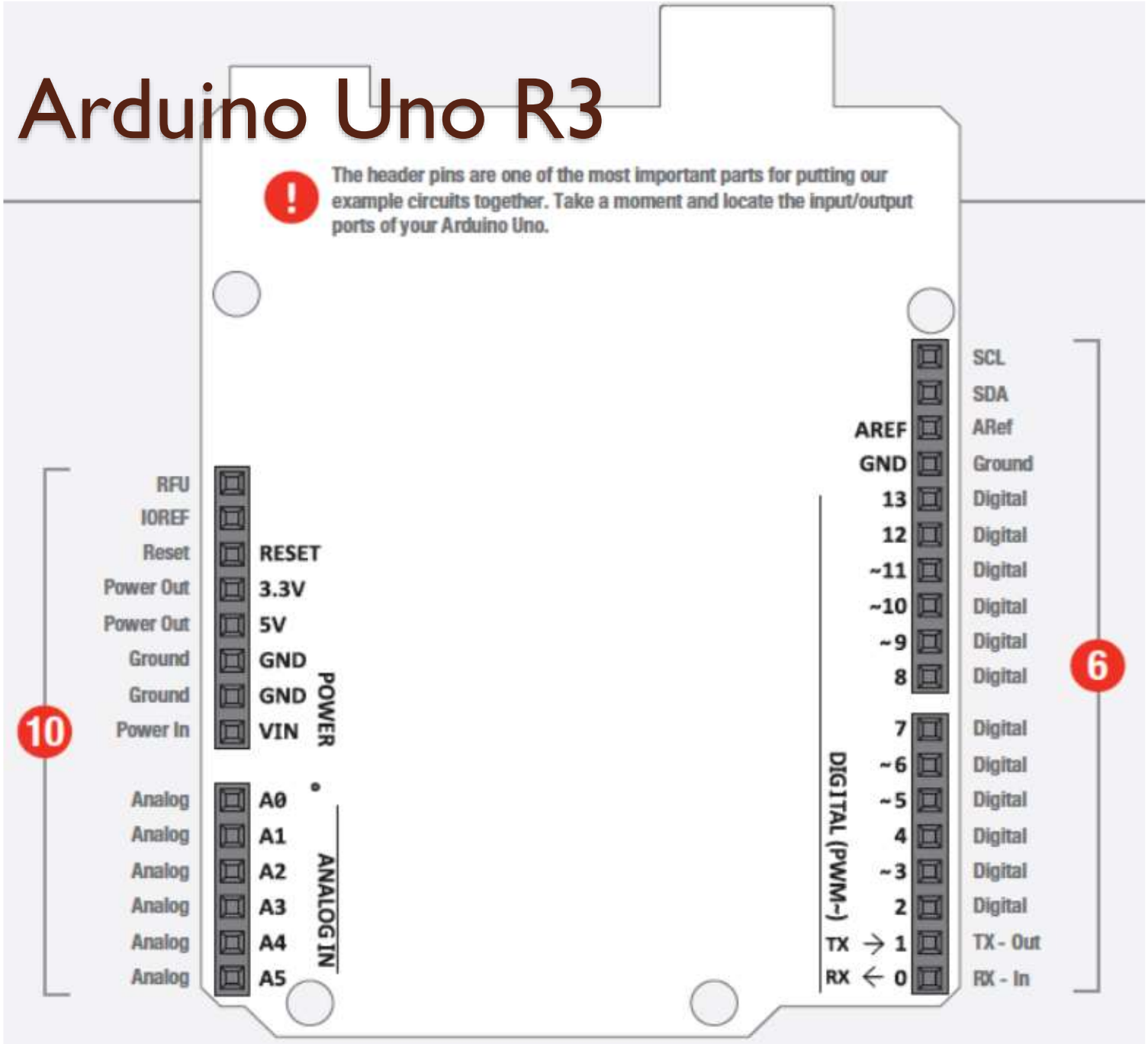
- Battery or external power supply (for stand-alone operation)

Arduino Uno R3



1. Power In
2. Power In USB and communication
3. LED – RX receiving
4. LED – TX transmitting
5. LED – User purpose
6. PINS – User purpose
7. LED – Power indicator
8. Reset Button
9. ICSP connector – upload code
10. 10- User purpose

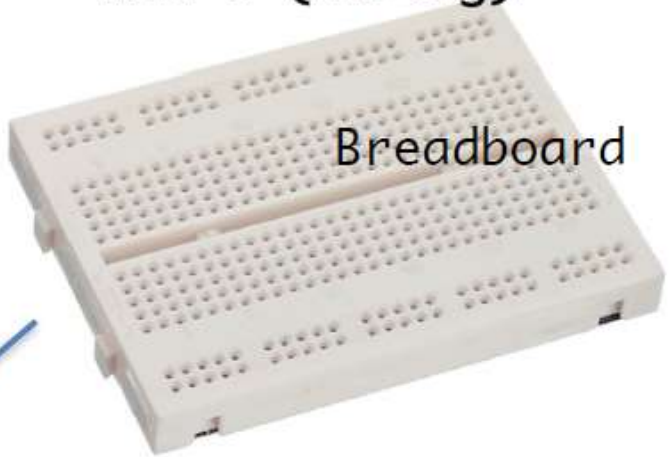
Arduino Uno R3



Alt 1 (no wiring)

TinkerKit
Sensor Shield

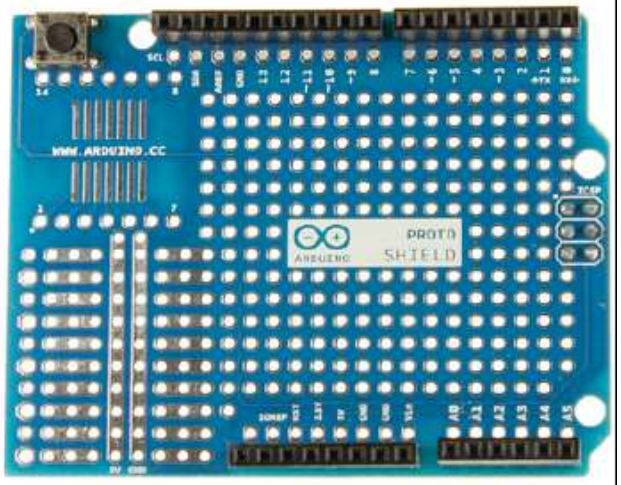
Alt 2 (wiring)



Breadboard

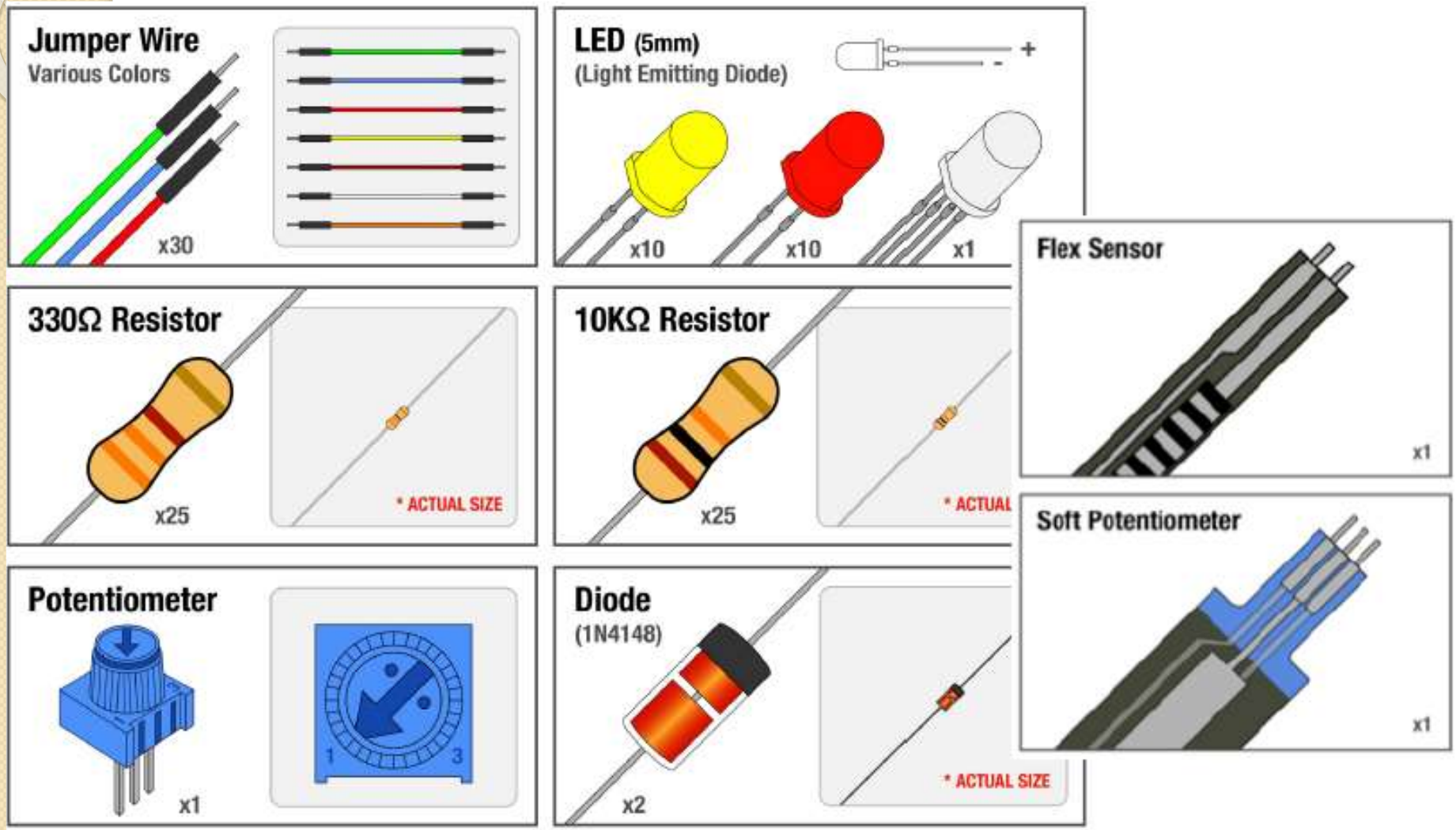
Alt 3 (soldering)

Arduino UNO

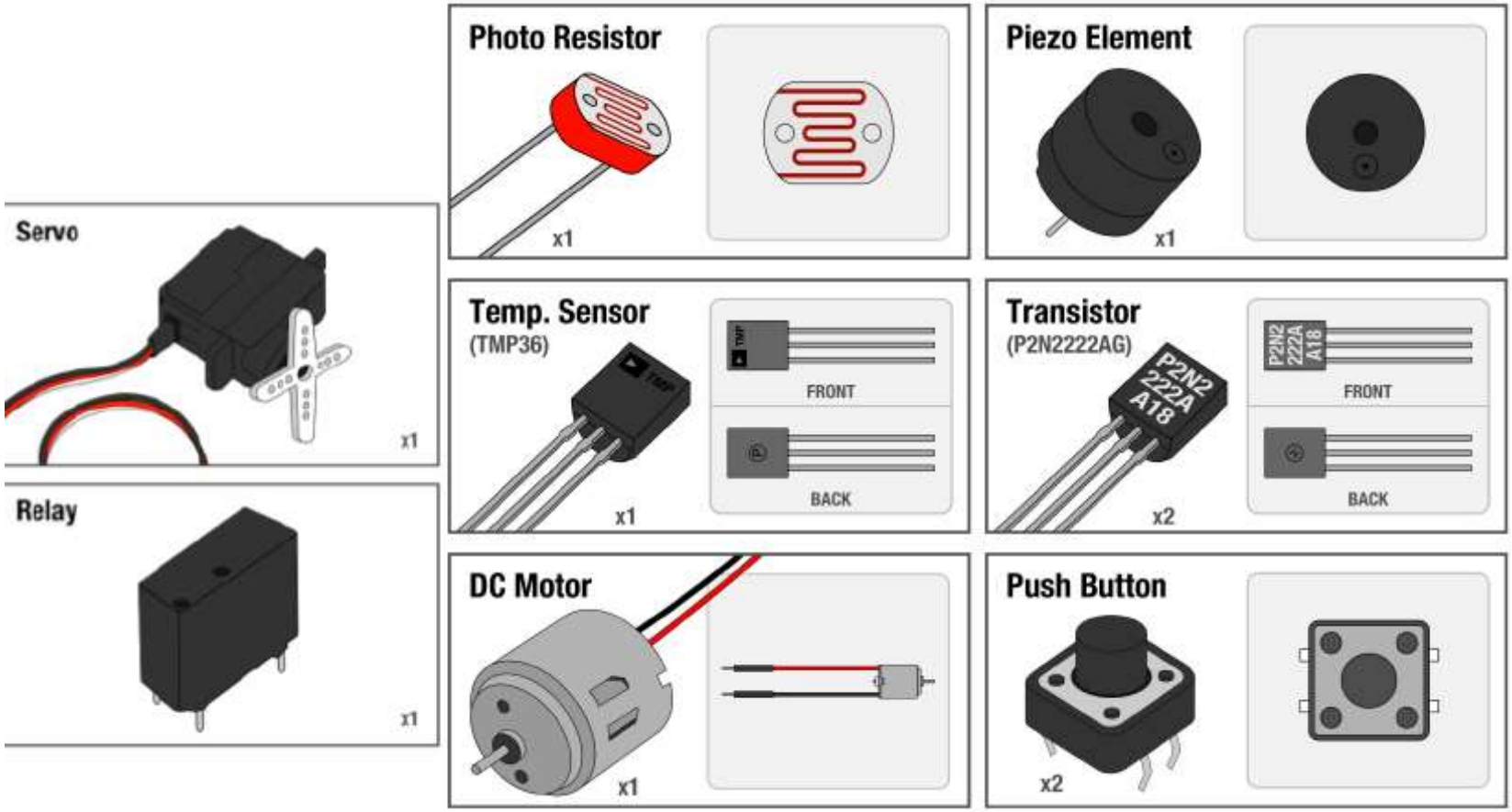


Arduino Proto Shield

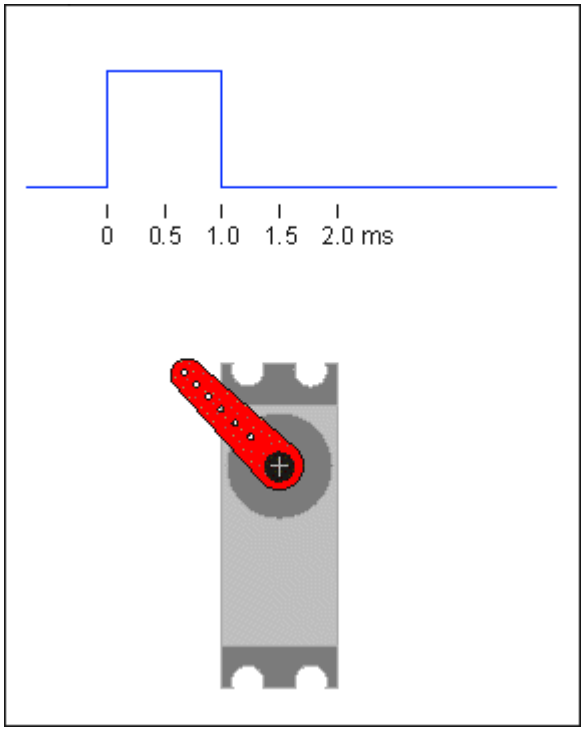
Arduino Uno R3 Connexions

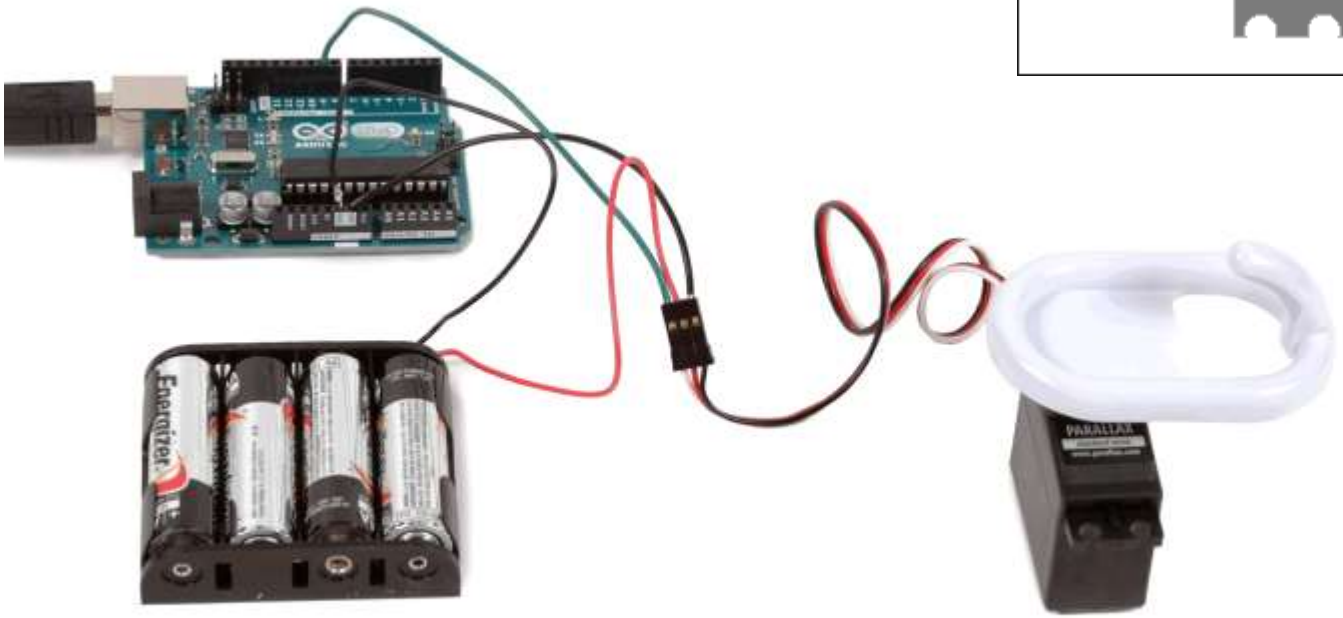


Arduino Uno R3 Connexions

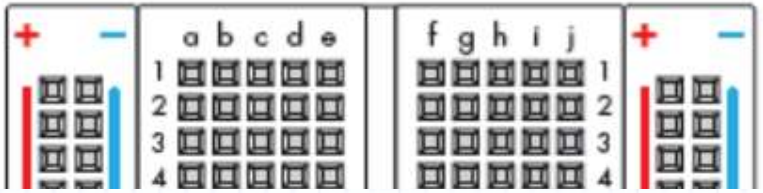


Servo motor

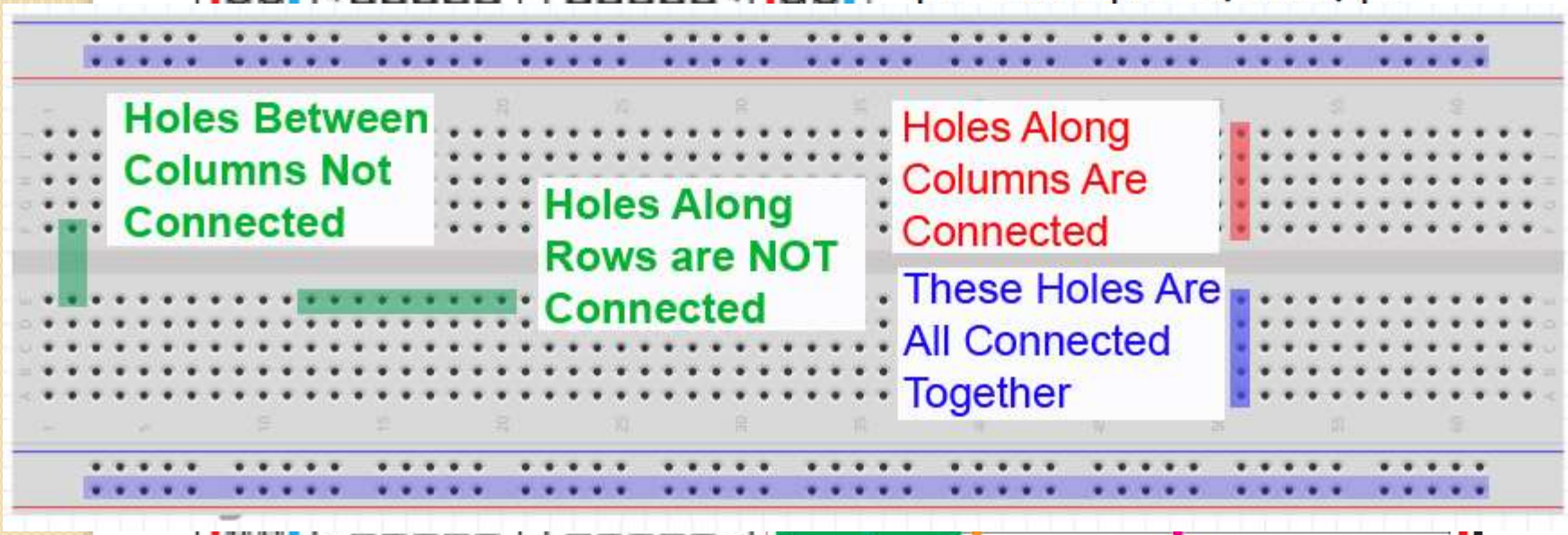


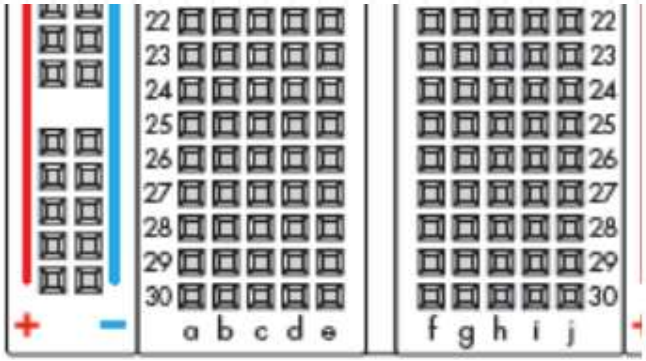


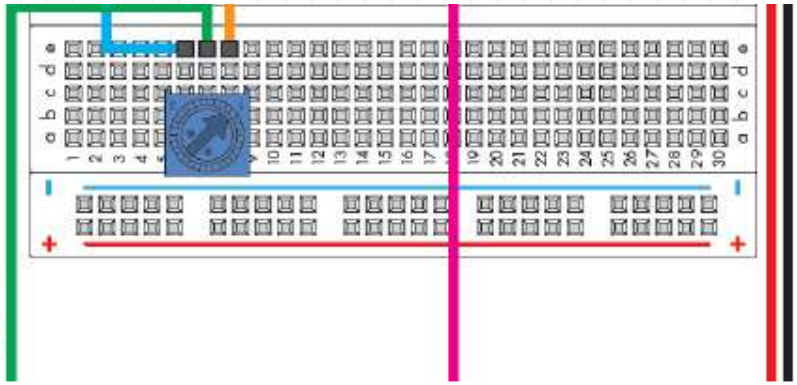
Arduino Uno R3 Breadboard



A breadboard is used





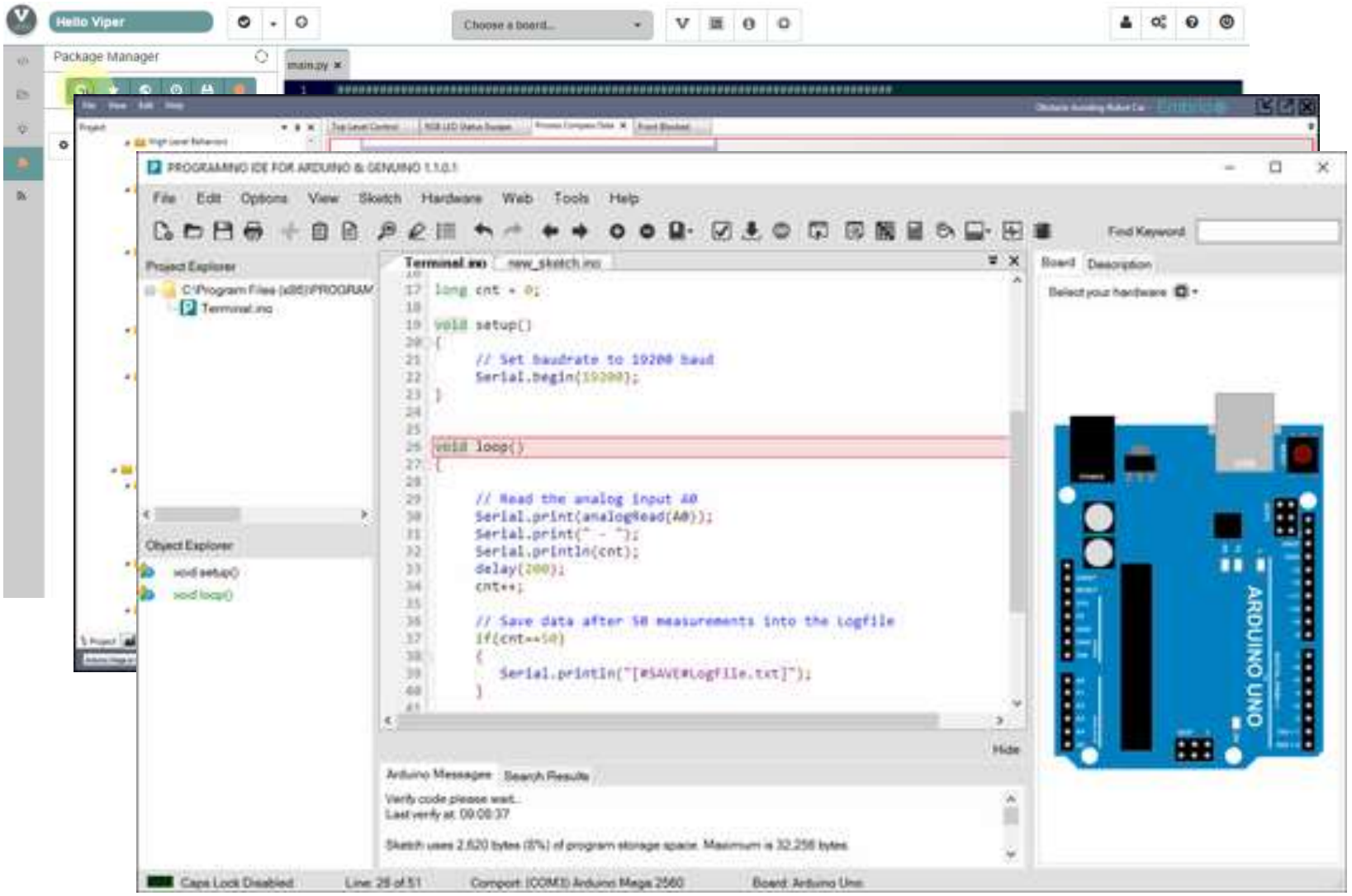


Alternative IDE

PROGRAMINO IDE

Embrio

Zerynth



IDE (Integrated Development Environment)

- Software application that provides comprehensive facilities to computer programmers for software development.
- Advantages:
  - Increased productivity
  - Feedback on syntax errors
  - Hotkeys
- Consists of :
  - source code editor,
  - build automation tools
  - debugger.

Arduino IDE/ Scketch

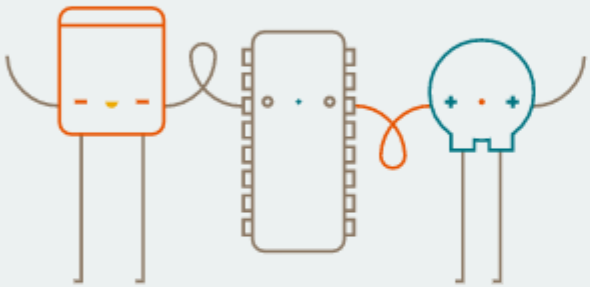
The Arduino Software (IDE) allows you to write programs and upload them to your board.

https://www.arduino.cc/en/Main/Software
http://www.w3ii.com/ro/arduino/arduino_installation.html

Arduino IDE/ Scketch

Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.



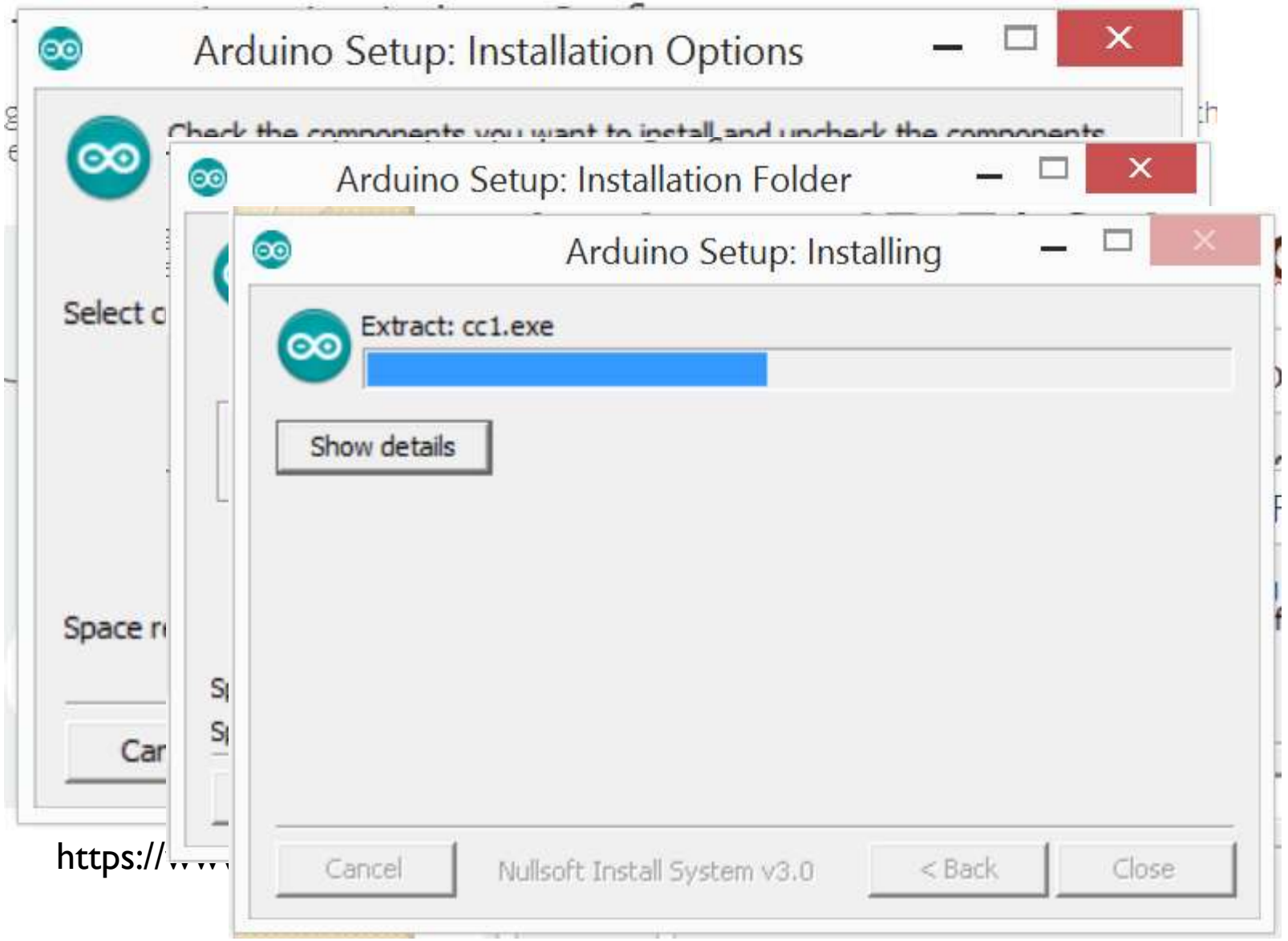
SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED 19,380,191 TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

$3 $5 $10 $25 $50 OTHER

JUST DOWNLOAD CONTRIBUTE & DOWNLOAD

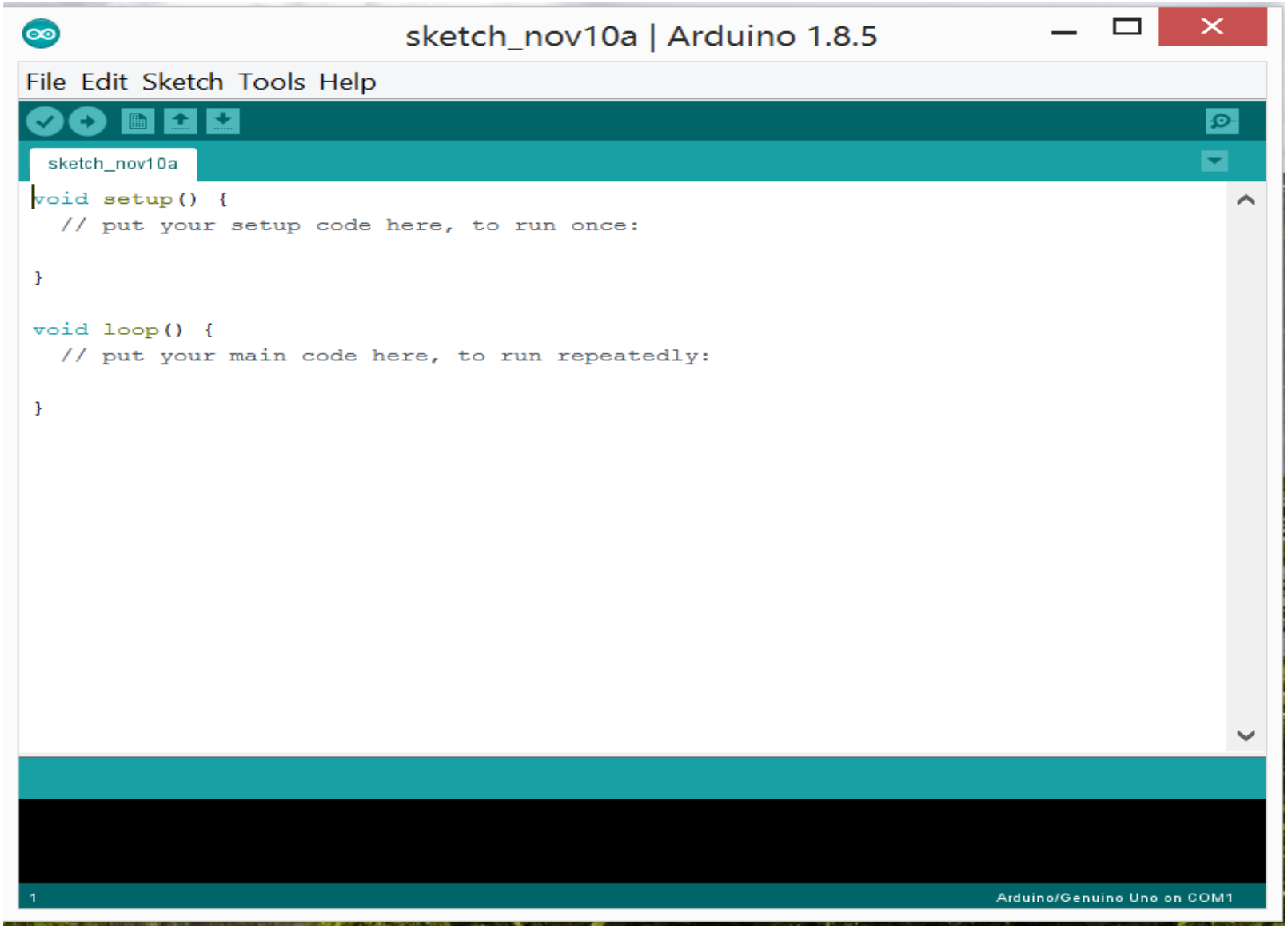
https://www.arduino.cc/en/Main/Software

Arduino IDE/ Scketch



https://

Arduino IDE/ Scketch



File

•***New***

Creates a new instance of the e
structure of a sketch already in

•***Open***

Allows to load a sketch file brov
and folders.

•***Open Recent***

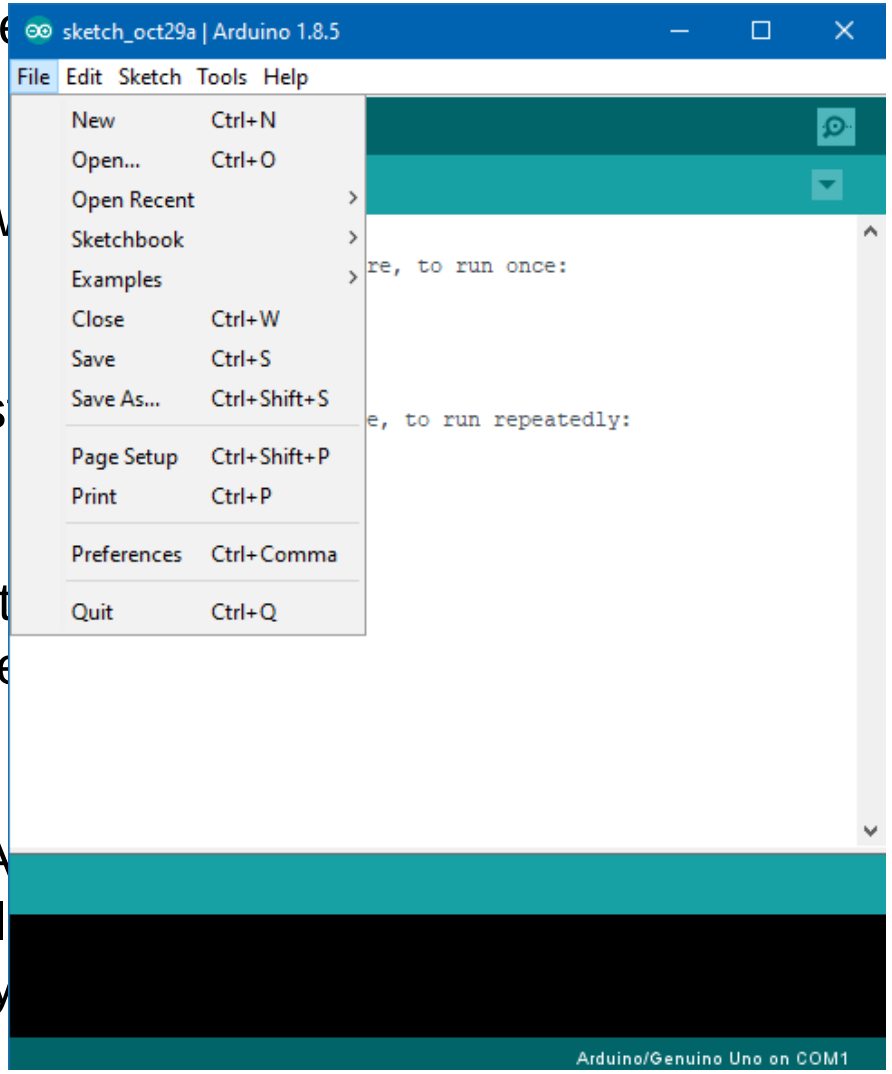
Provides a short list of the mos
opened.

•***Sketchbook***

Shows the current sketches wit
clicking on any name opens the
editor instance.

•***Examples***

Any example provided by the A
shows up in this menu item. All
tree that allows easy access by

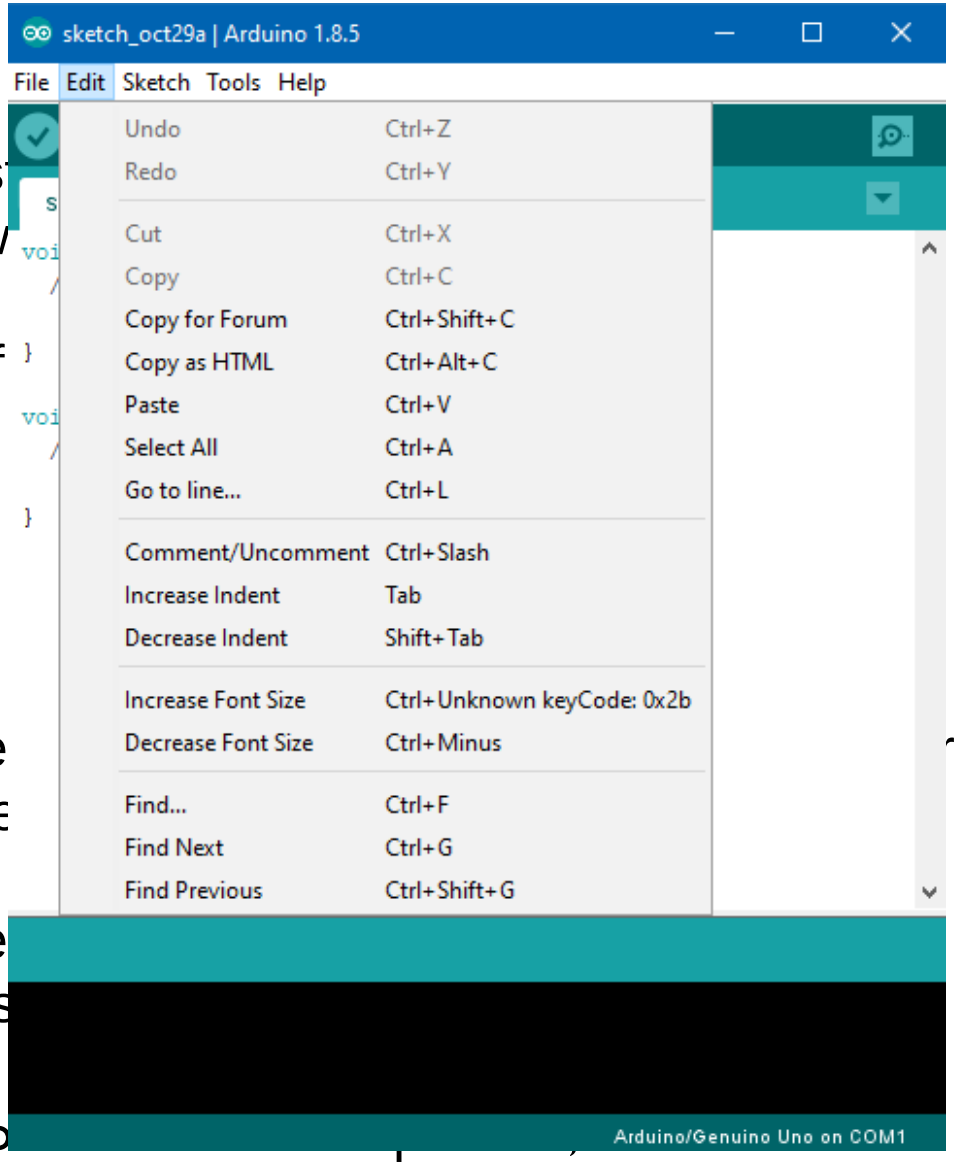


File

•***Close***
Closes the instance of the Arduino Software from which it is clicked.
•***Save***
Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.
•***Save as...***
Allows to save the current sketch with a different name.
•***Print***
Sends the current sketch to the printer according to the settings defined in Page Setup.
•***Preferences***
Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.
•***Quit***
Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

Edit

•***Undo/Redo***
Goes back of one or more s
back, you may go forward w
•***Cut***
Removes the selected text f
clipboard.
•***Copy***
Duplicates the selected text
clipboard.
•***Copy for Forum***
Copies the code of your ske
posting to the forum, comple
•***Copy as HTML***
Copies the code of your ske
for embedding in web pages
•***Paste***
Puts the contents of the clipbo



Edit

•***Select All***

Selects and highlights the whole content of the editor.

•***Comment/Uncomment***

Puts or removes the // comment marker at the beginning of each selected line.

•***Increase/Decrease Indent***

Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.

•***Find***

Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.

•***Find Next***

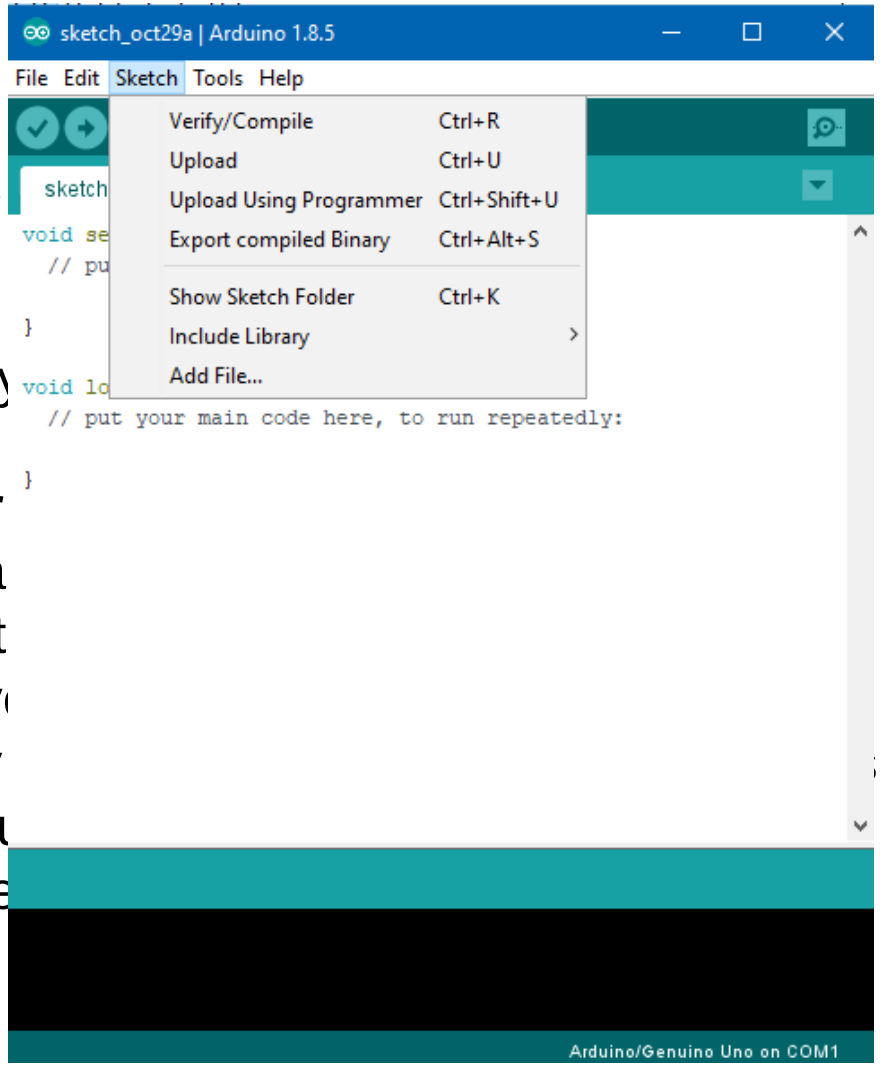
Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.

•***Find Previous***

Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

Sketch

•***Verify/Compile***
Checks your sketch for errors
usage for code and variables
•***Upload***
Compiles and loads the binary
through the configured Port.
•***Upload Using Programmer***
This will overwrite the bootloa
use Tools > Burn Bootloader t
USB serial port again. Howev
capacity of the Flash memory
command will NOT burn the fu
Bootloader command must be



Sketch

•***Export Compiled Binary***
Saves a .hex file that may be kept as archive or sent to the board using other tools.
•***Show Sketch Folder***
Opens the current sketch folder.
•***Include Library***
Adds a library to your sketch by inserting #include statements at the start of your code. For more details, see libraries below. Additionally, from this menu item you can access the Library Manager and import new libraries from .zip files.
•***Add File...***
Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side o the toolbar.

Tools

- ***Auto Format***

This formats your code nice
closing curly braces line up
braces are indented more.

- ***Archive Sketch***

Archives a copy of the curre
placed in the same director

- ***Fix Encoding & Reload***

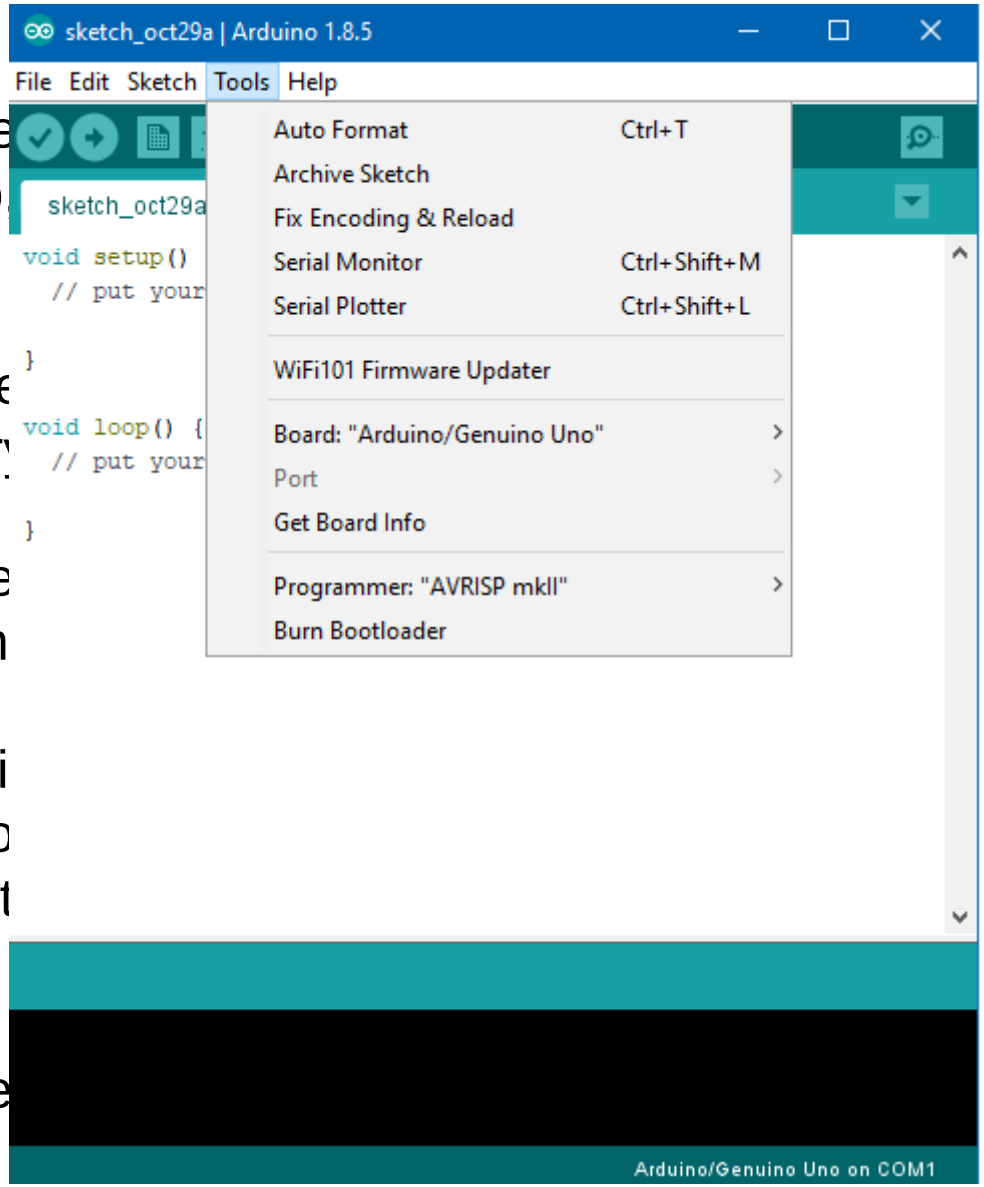
Fixes possible discrepancie
and other operating system

- ***Serial Monitor***

Opens the serial monitor wi
with any connected board o
usually resets the board, if t
opening.

- ***Board***

Select the board that you're
various boards.



Tools

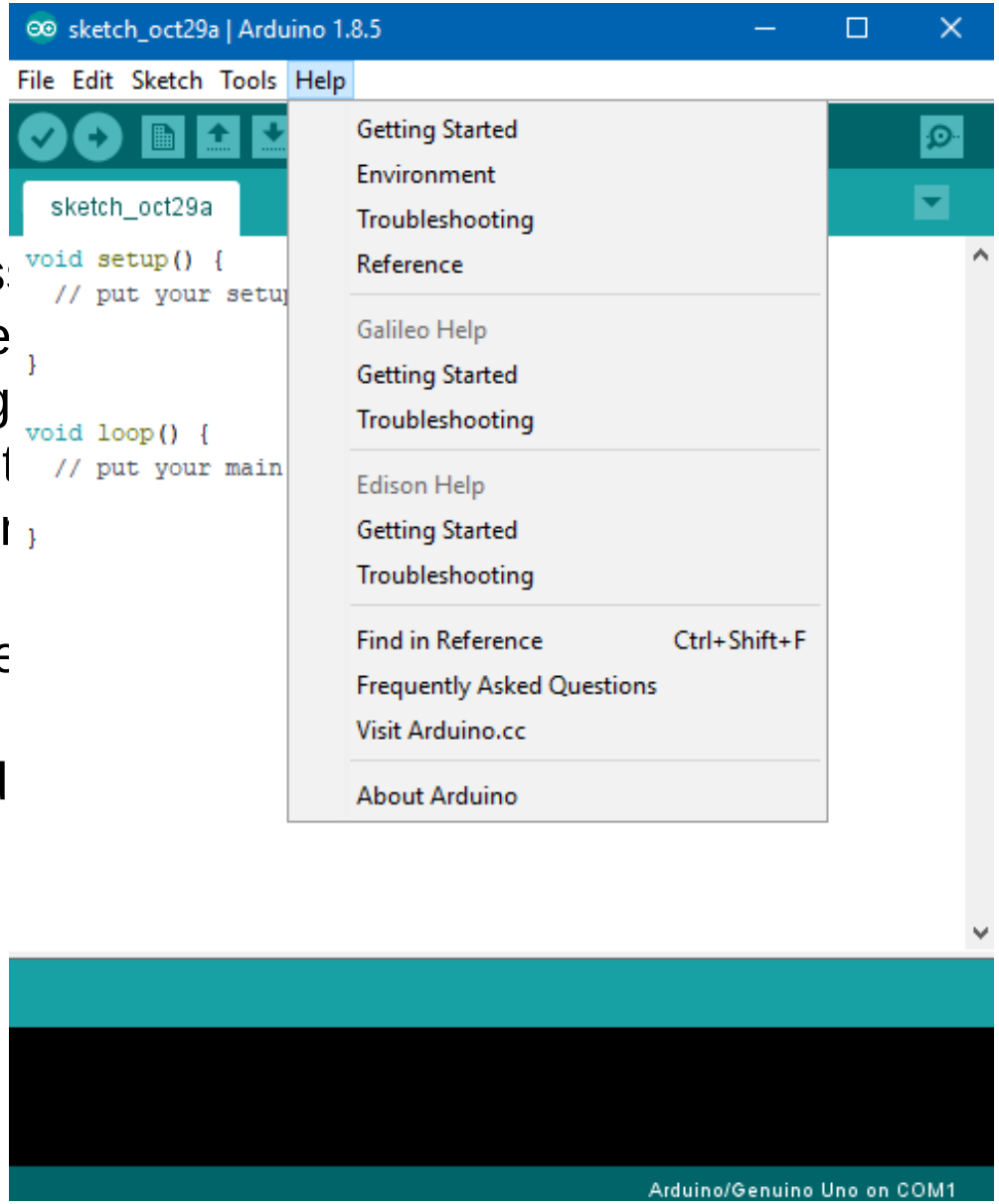
•***Port***
This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.
•***Programmer***
For selecting a harware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.
•***Burn Bootloader***
The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the **Boards** menu before burning the bootloader on the target board. This command also set the right fuses.

Help

Here you find easy acces
with the Arduino Software
Started, Reference, this g
locally, without an internet
copy of the online ones ar

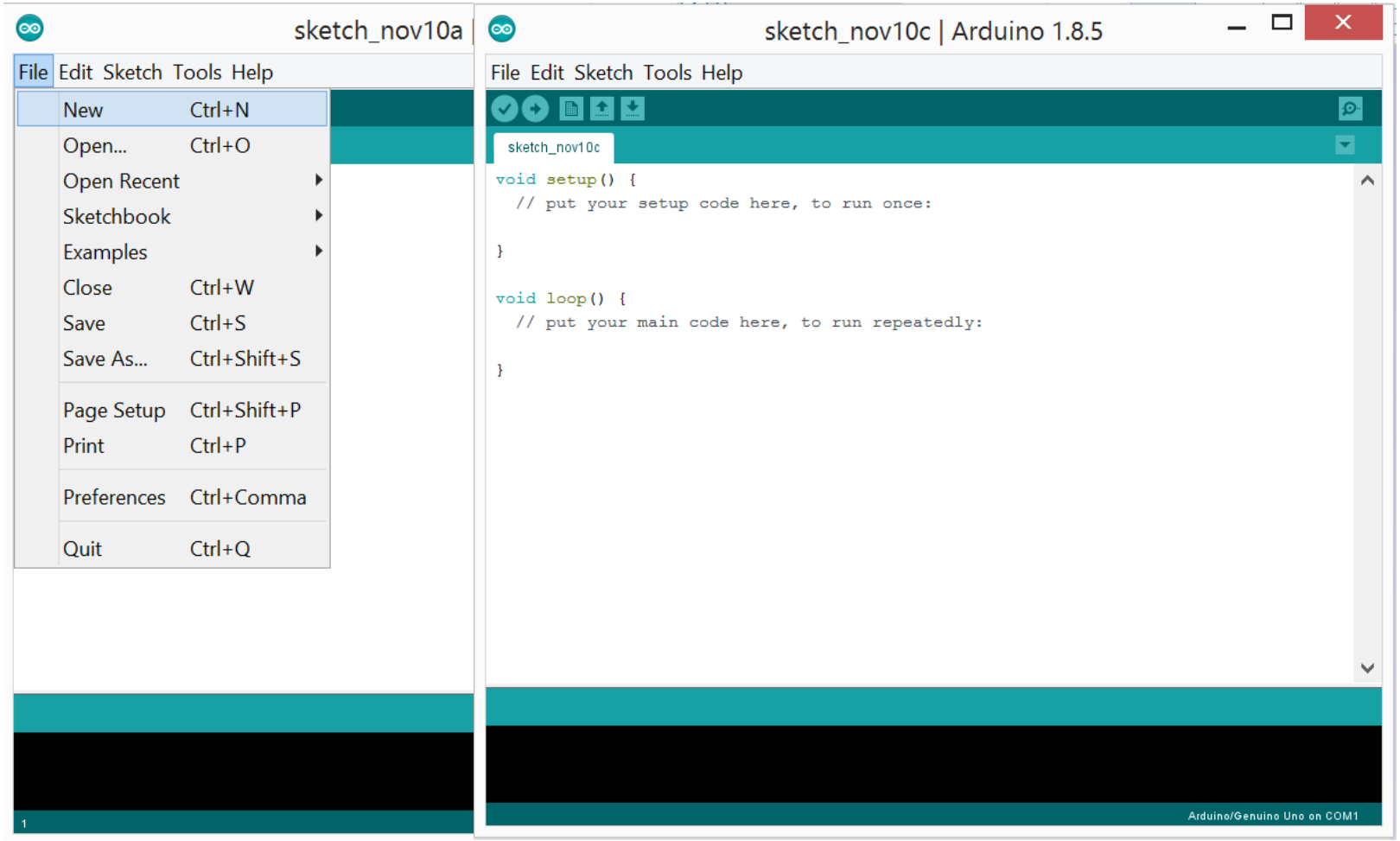
•***Find in Reference***

This is the only interactive
selects the relevant page
function or command und



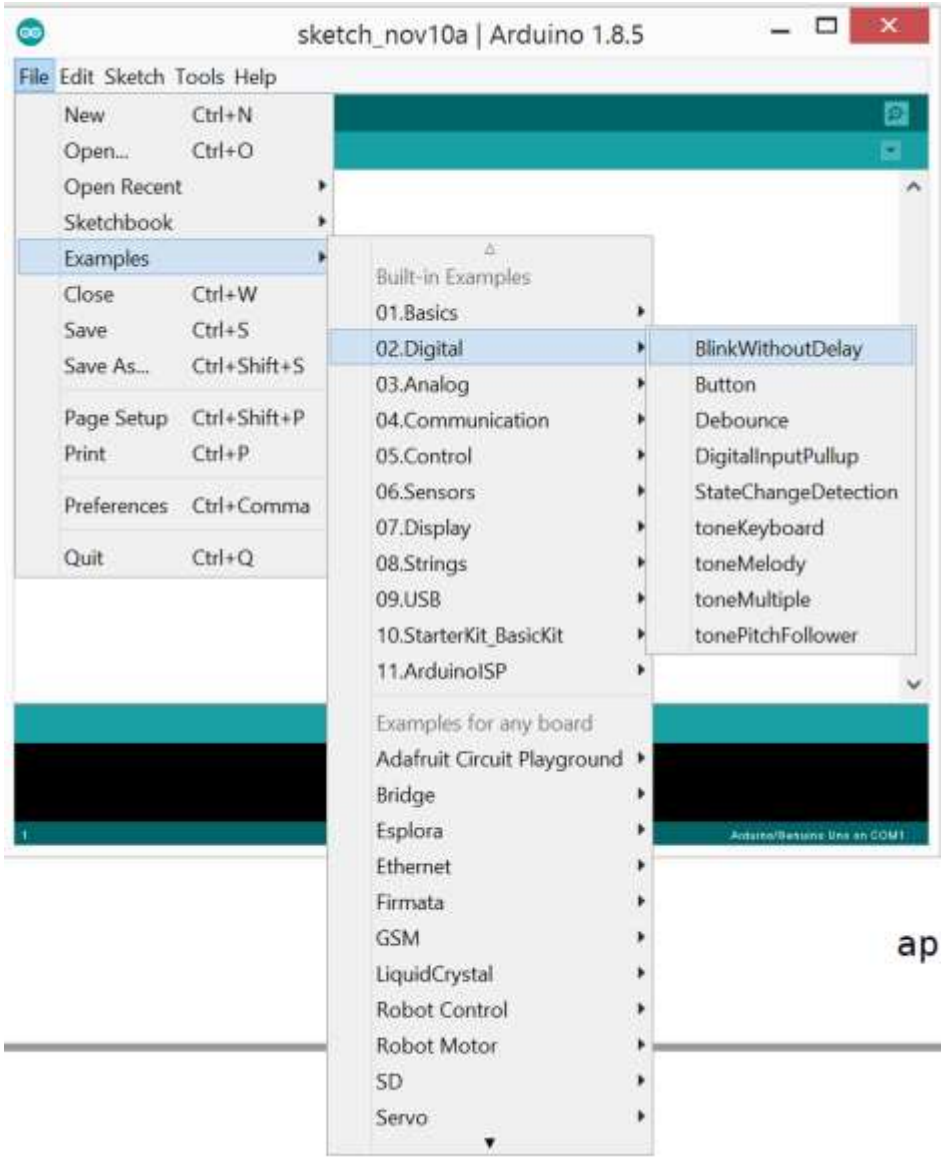
First project

File -> New

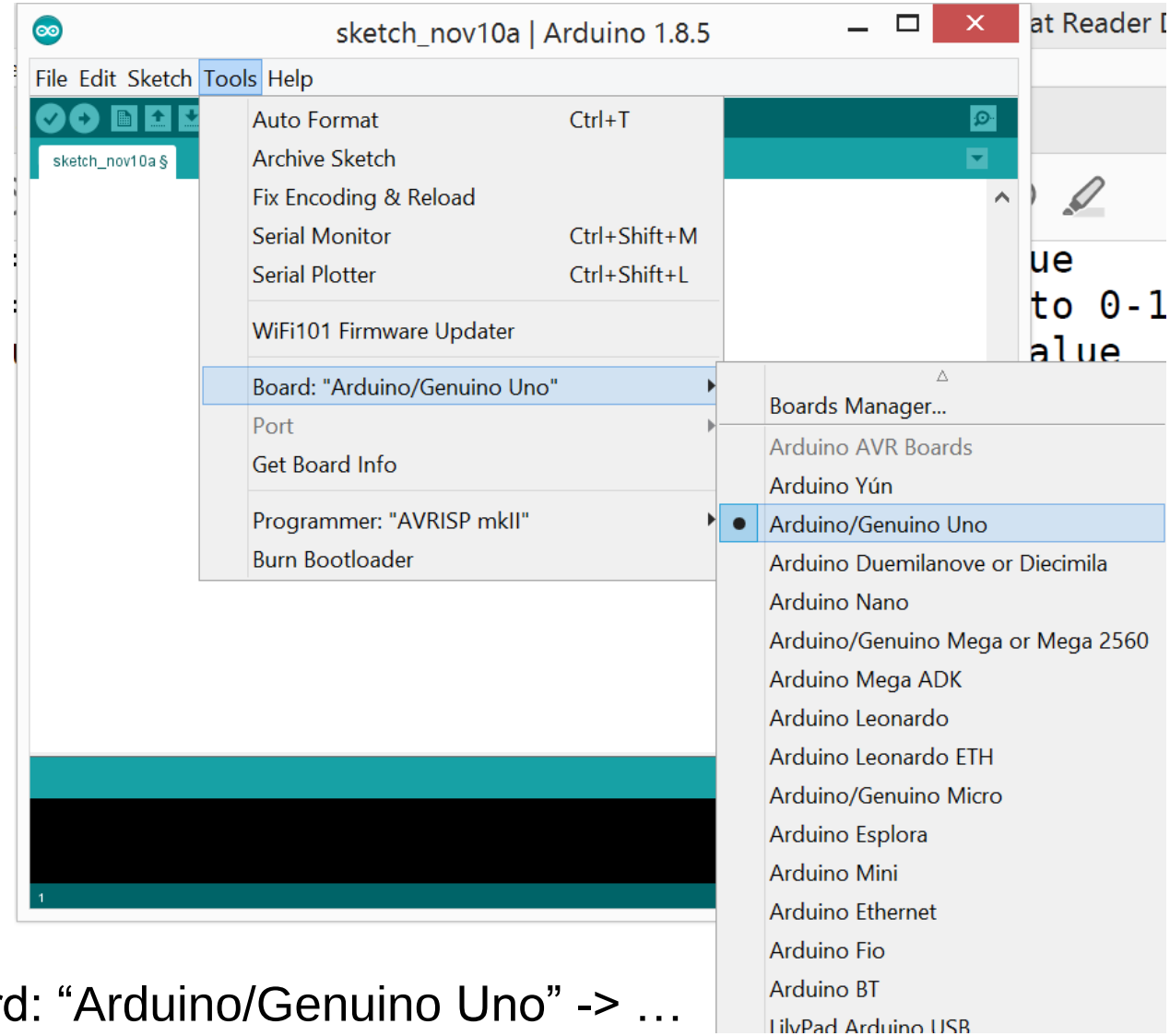


First project

File -> Example -> …

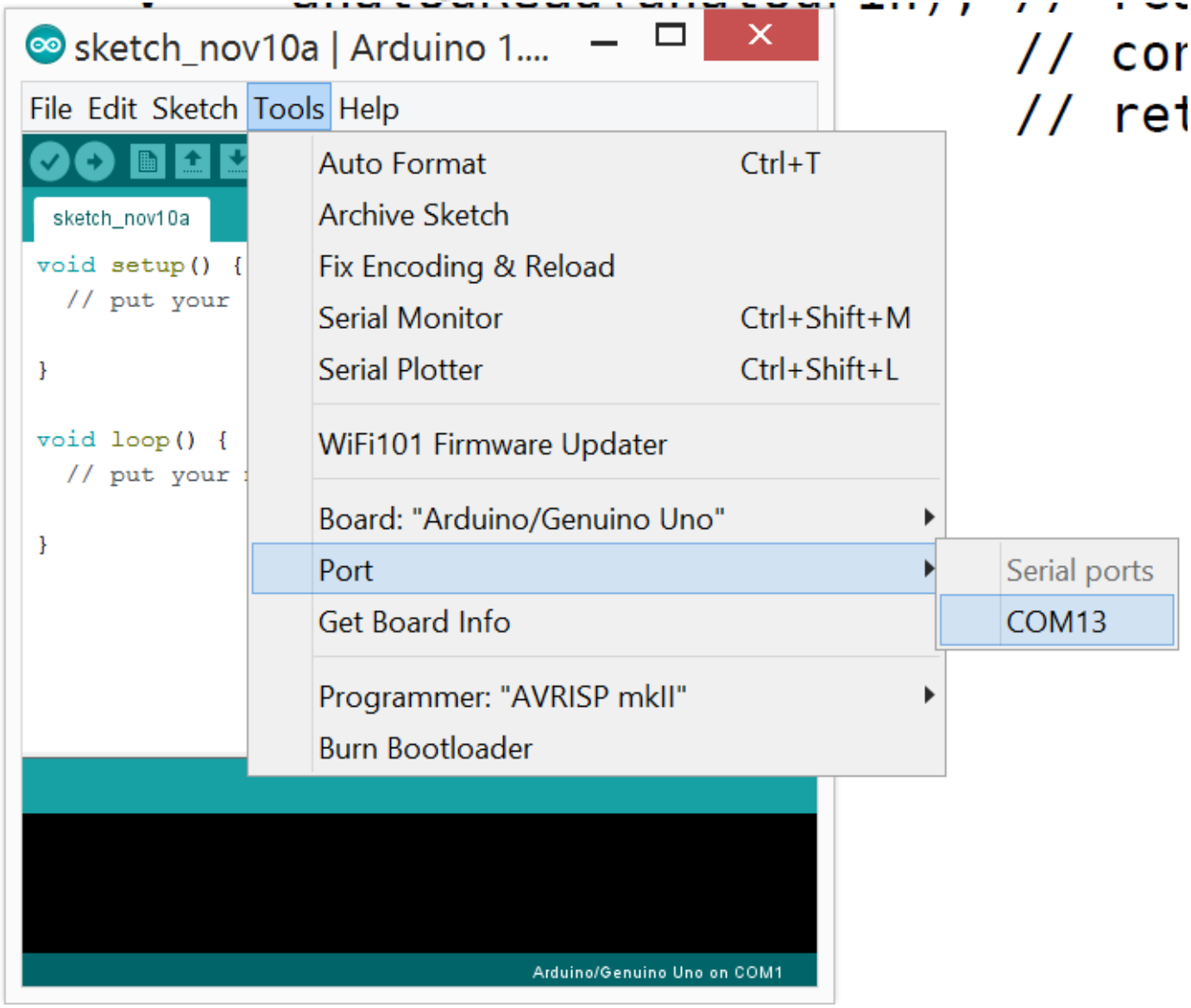


Select Arduino board

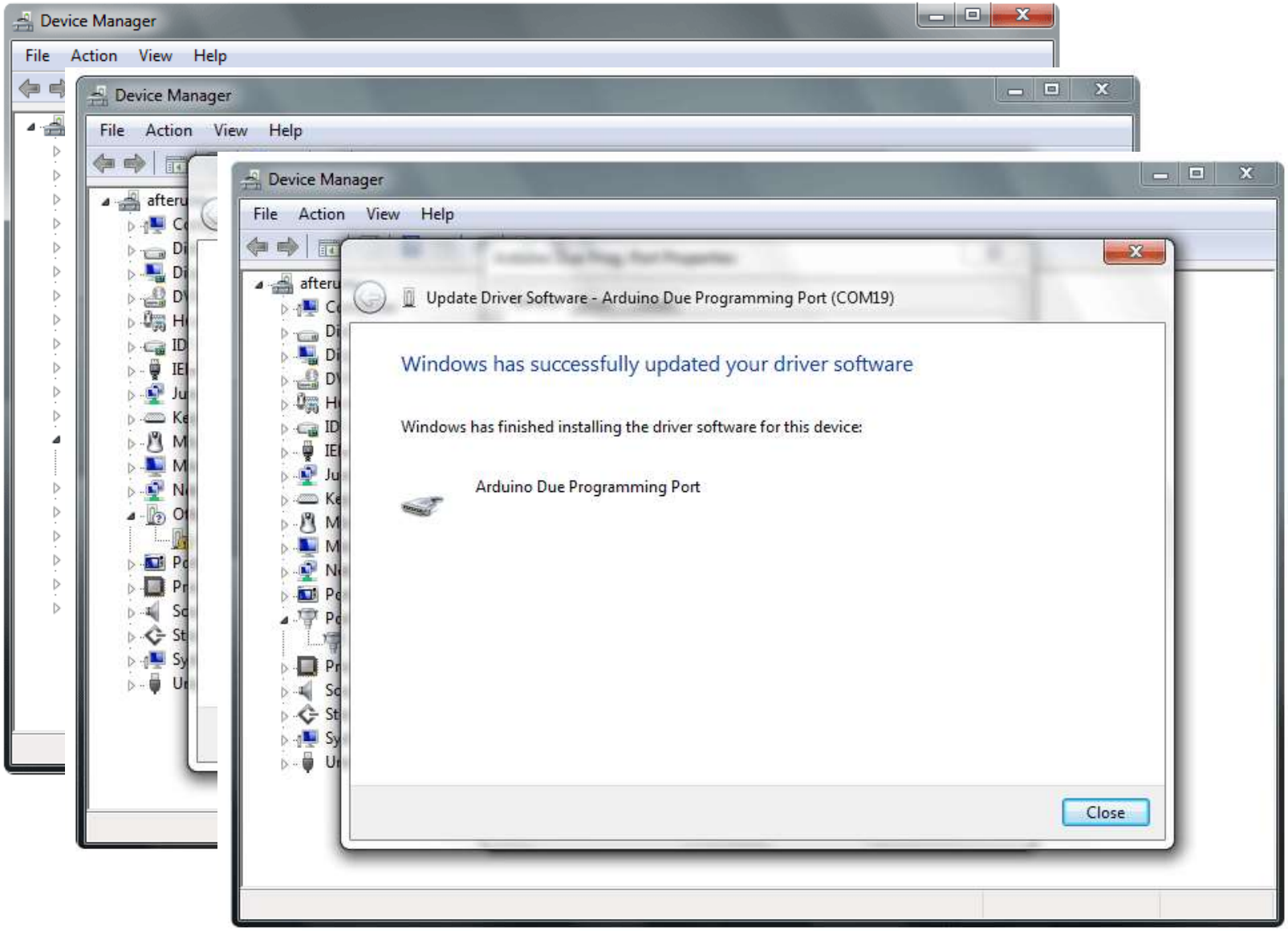


Tools -> Board: “Arduino/Genuino Uno” -> …

Select serial port



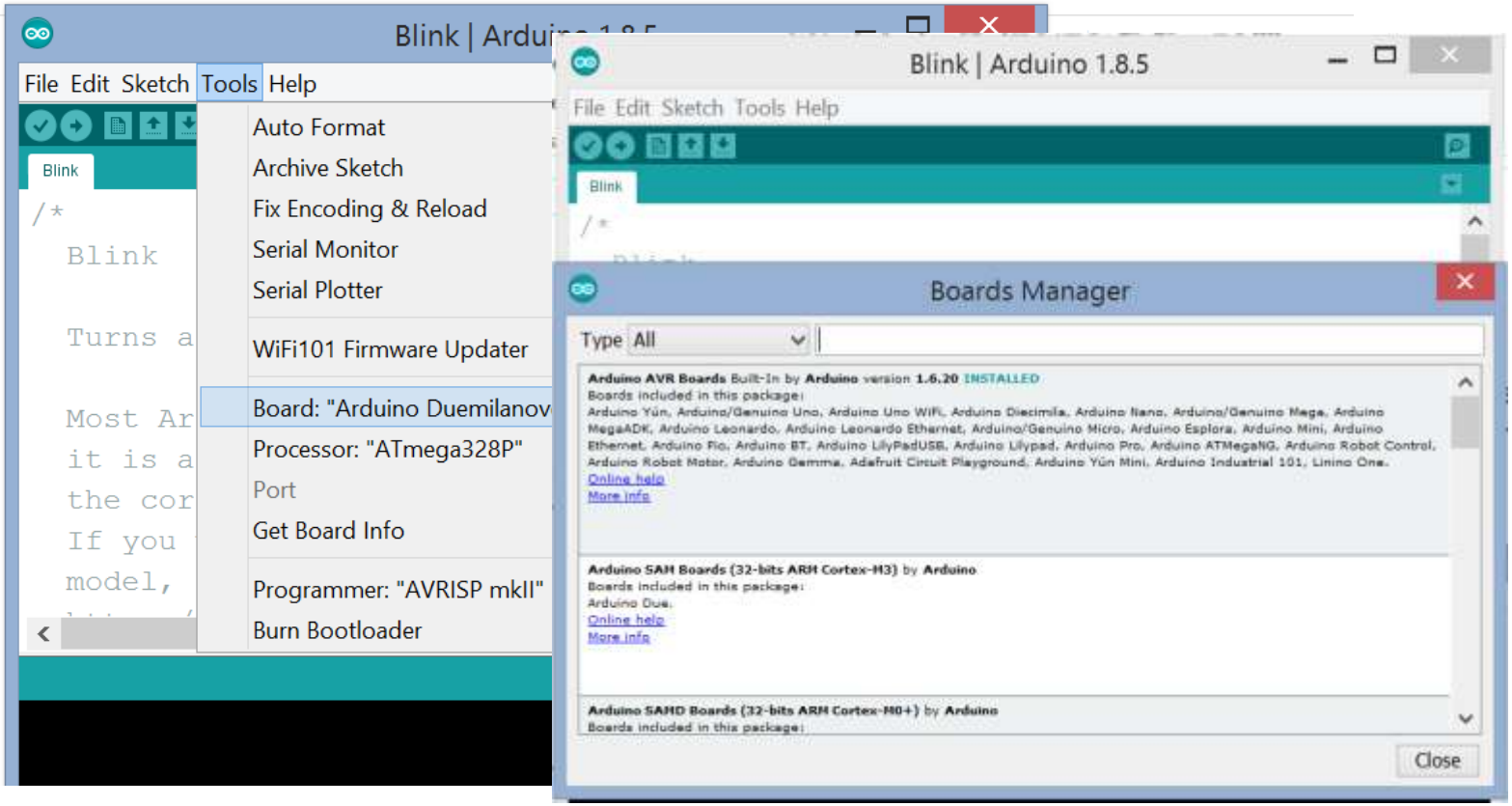
Drivers Install (Win 7)



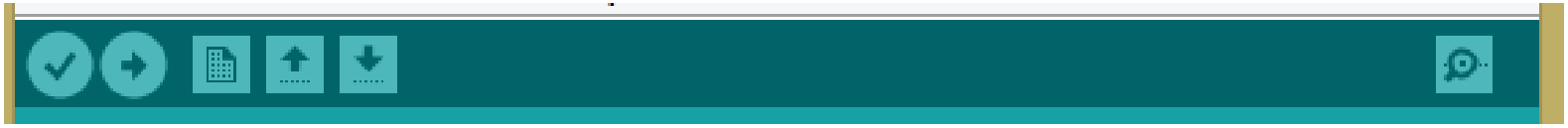
Core install

•Some Arduino boards require an additional Core to be installed. Ex. For Arduino Due an Arduino board based on a 32-bit ARM core microcontroller

• you need to install the SAM Core using the Boards Manager



Scketh toolbar



Verify

Upload

New

Open

Save

Serial Monitor

Typical software code

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

}

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

}
```

Programs written using Arduino Software (IDE) are called sketches.

Language reference

https://www.arduino.cc/reference/en/#functions

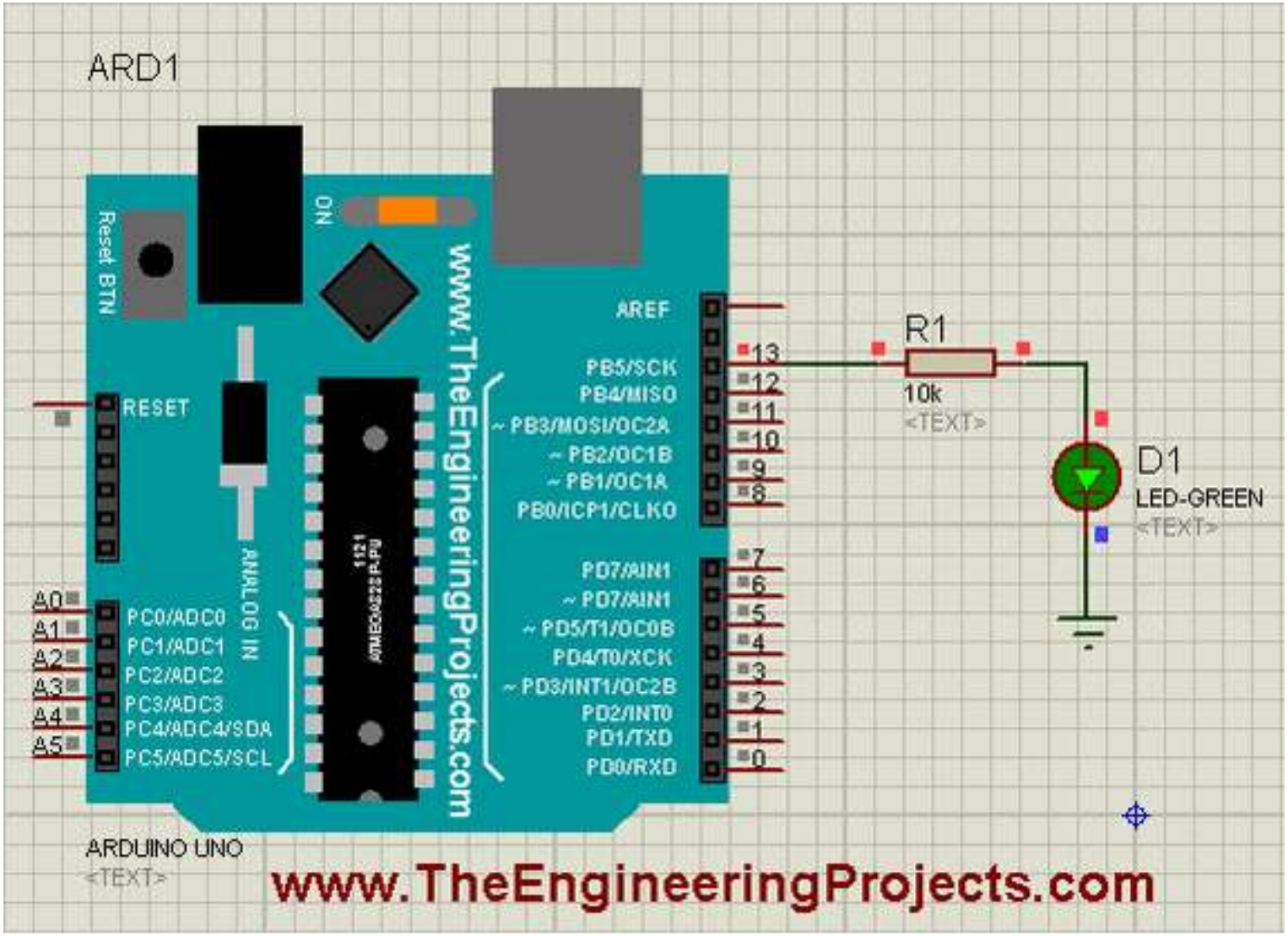
- **LANGUAGE**
- **FUNCTIONS**
- **VARIABLES**
- **STRUCTURE**

IDE and SKETCH overview

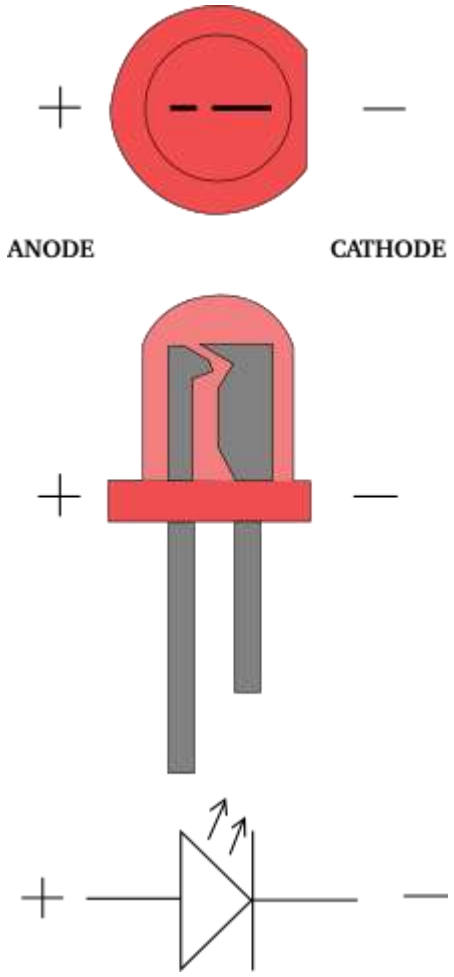
- ; semi-colons
- // single line comments
- /* */ multi-line comments
- { } open and closing curly braces
- ( ) parenthesis
- void setup() – identify the opening and closing curly braces
- void loop() – identify the opening and closing curly braces
- some blue keywords like OUTPUT or INPUT

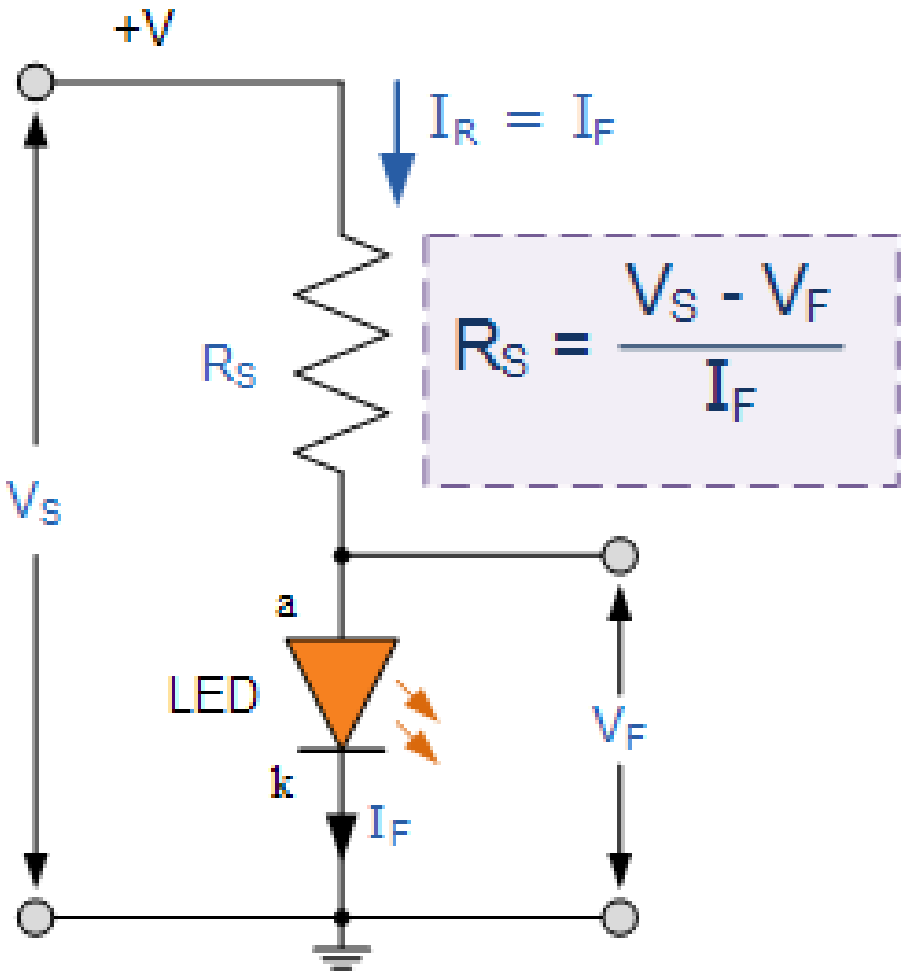
https://programmingelectronics.com/tutorial-3-arduino-ide-and-sketch-overview/

Example: Led Blink



Example: Led Blink





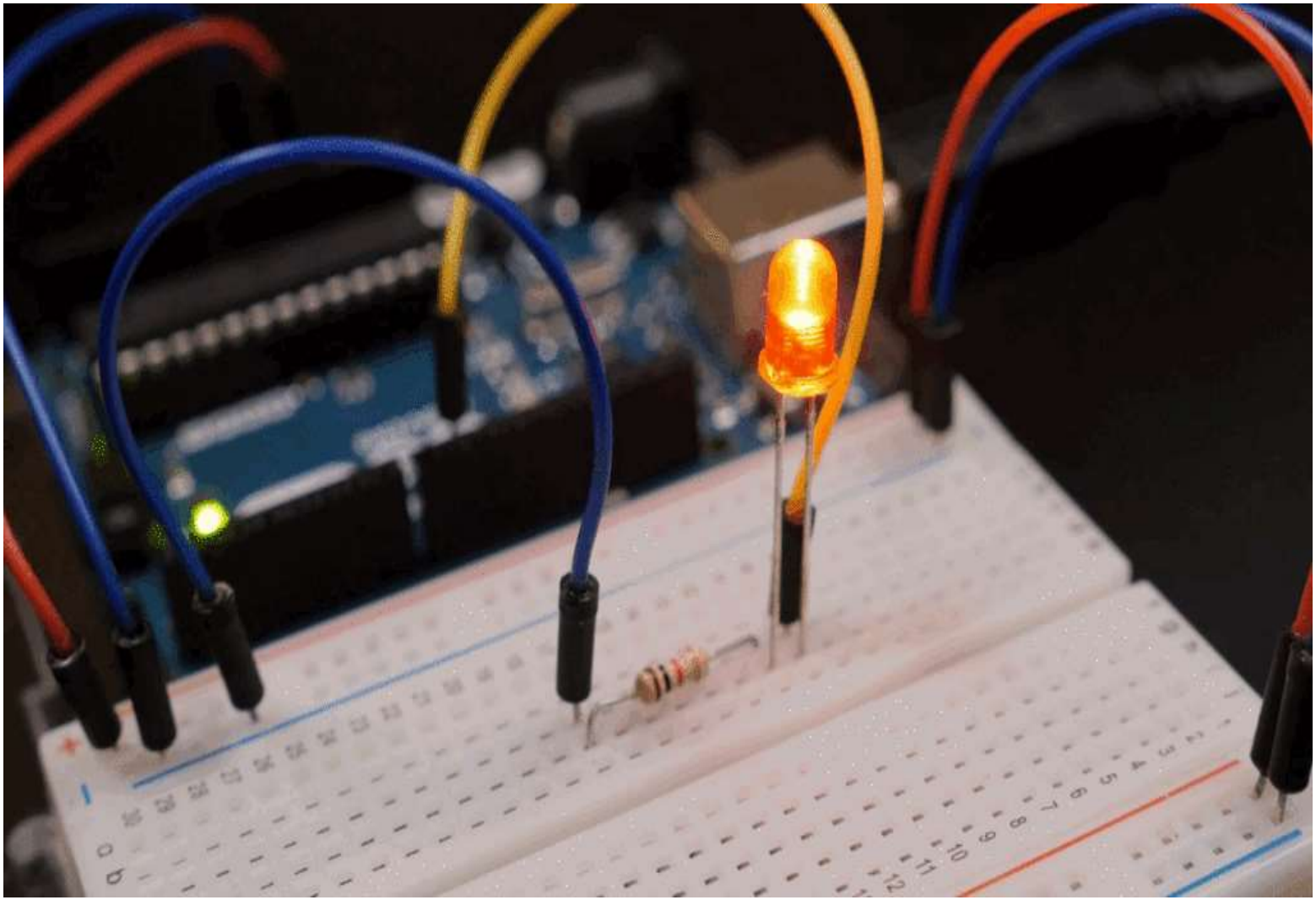
Example: Led Blink

```
void setup()
{
   pinMode(13, OUTPUT);
}
void loop()
{
   digitalWrite(13, HIGH);
   delay(1000); //delay 1000 ms
   digitalWrite(13, LOW);
   delay(1000); //delay 1000 ms
}
```

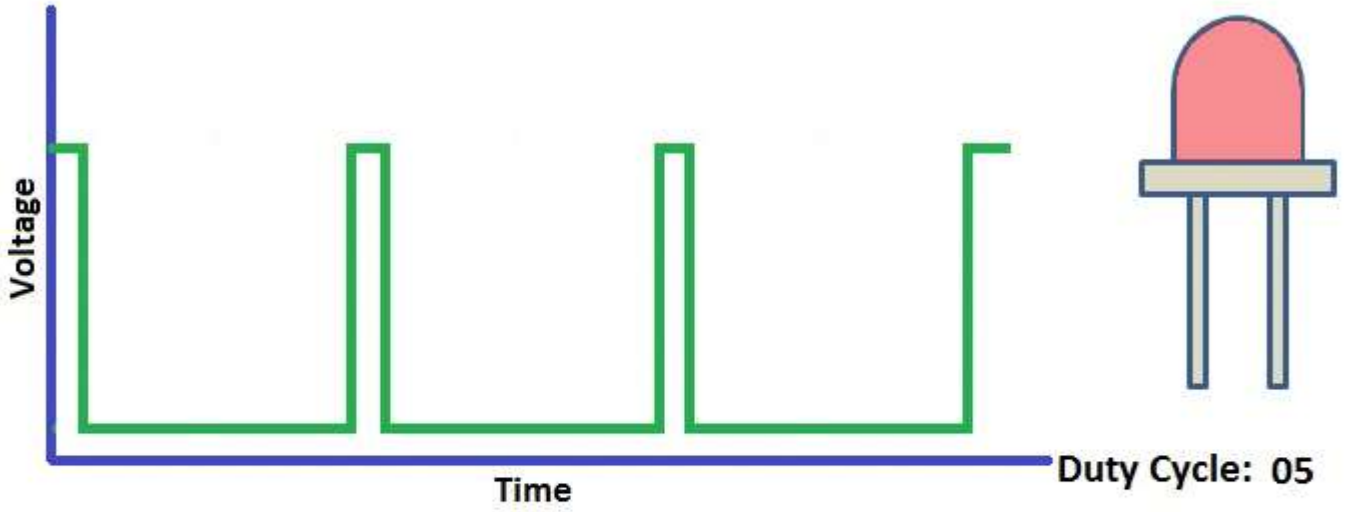
Example: Built-in LED Blink

```
void setup()
{
   pinMode(LED_BUILTIN, OUTPUT);
}
void loop()
{
   digitalWrite(LED_BUILTIN, HIGH);
   delay(500);
   digitalWrite(LED_BUILTIN, LOW);
   delay(500);
}
```

Example: variable intensity Led Brick



Example: variable intensity Led Blink



Frequency > 50Hz

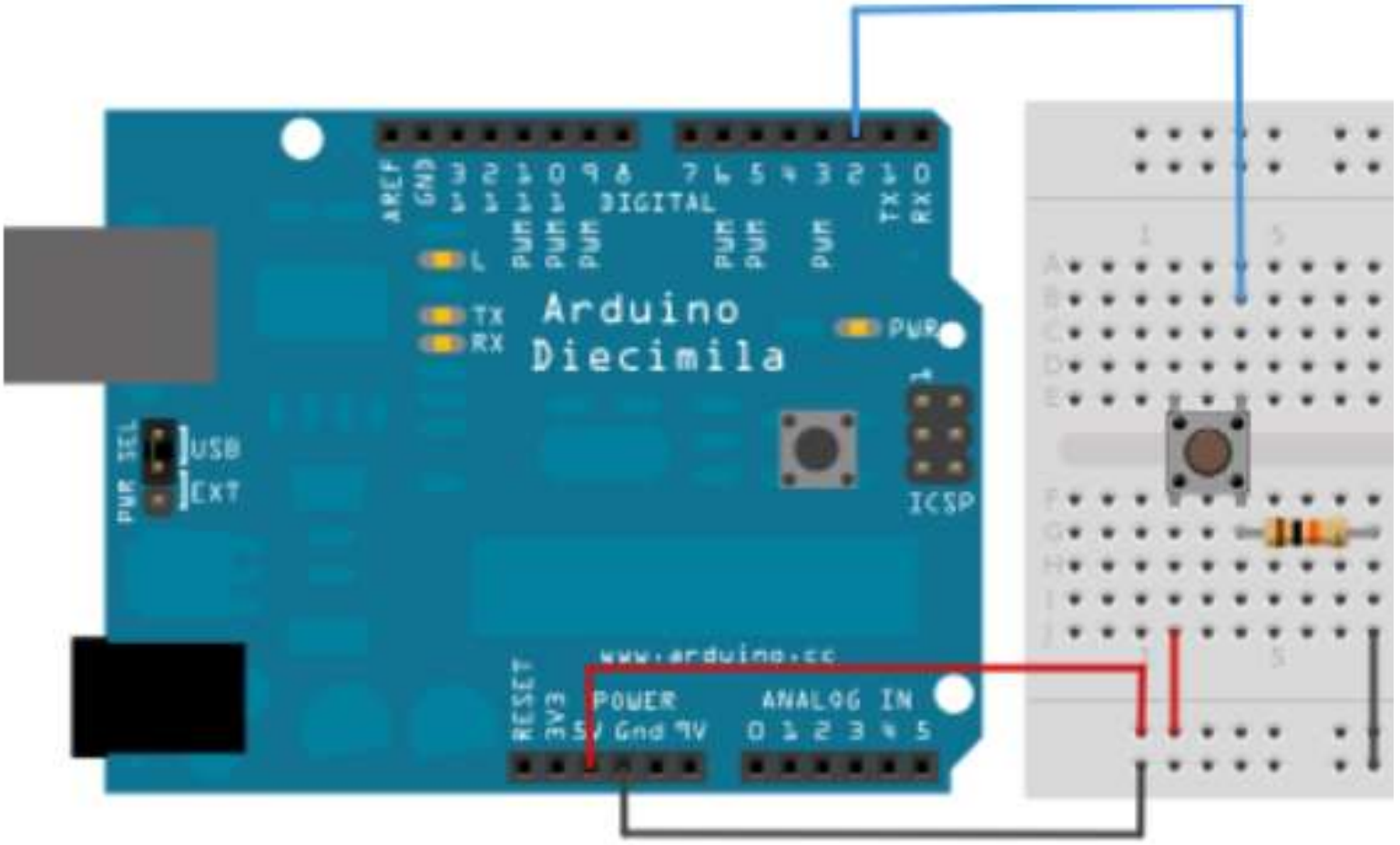
Only 6 pins can generate PWM signal (3, 5, 6, 9, 10 and 11)

Example: variable intensity LED Blink

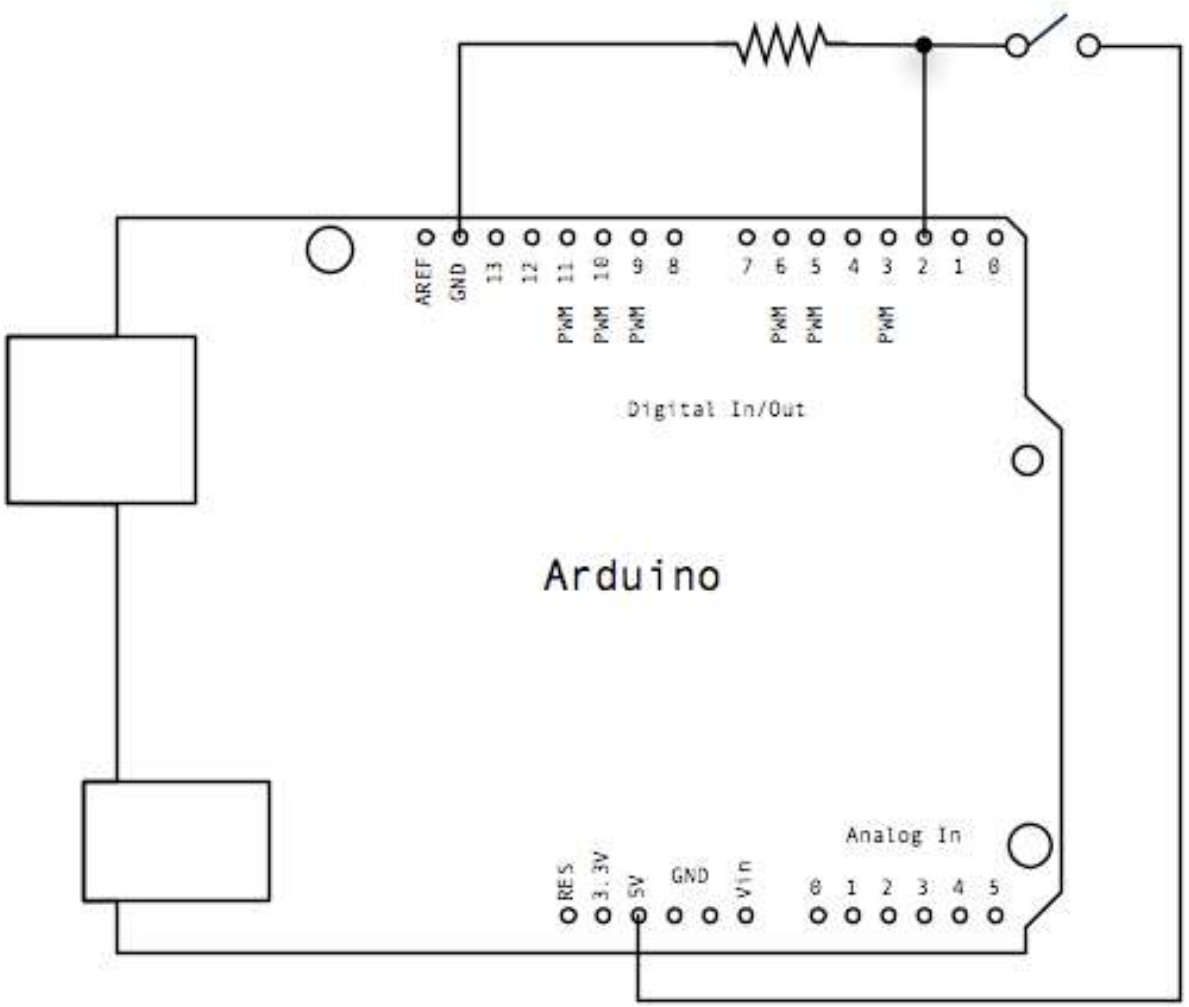
- It is used pin 11
- Duty factor between 0 and 255

```
void setup() {
  pinMode(11, OUTPUT);
}
void loop() {
  for (int i = 0; i < 255; i++)
{
    analogWrite(11, i);
    delay(50);
  }
  for (int i = 255; i > 0; i--) {
    analogWrite(11, i);
    delay(50);
  }
}
```

Example: Button



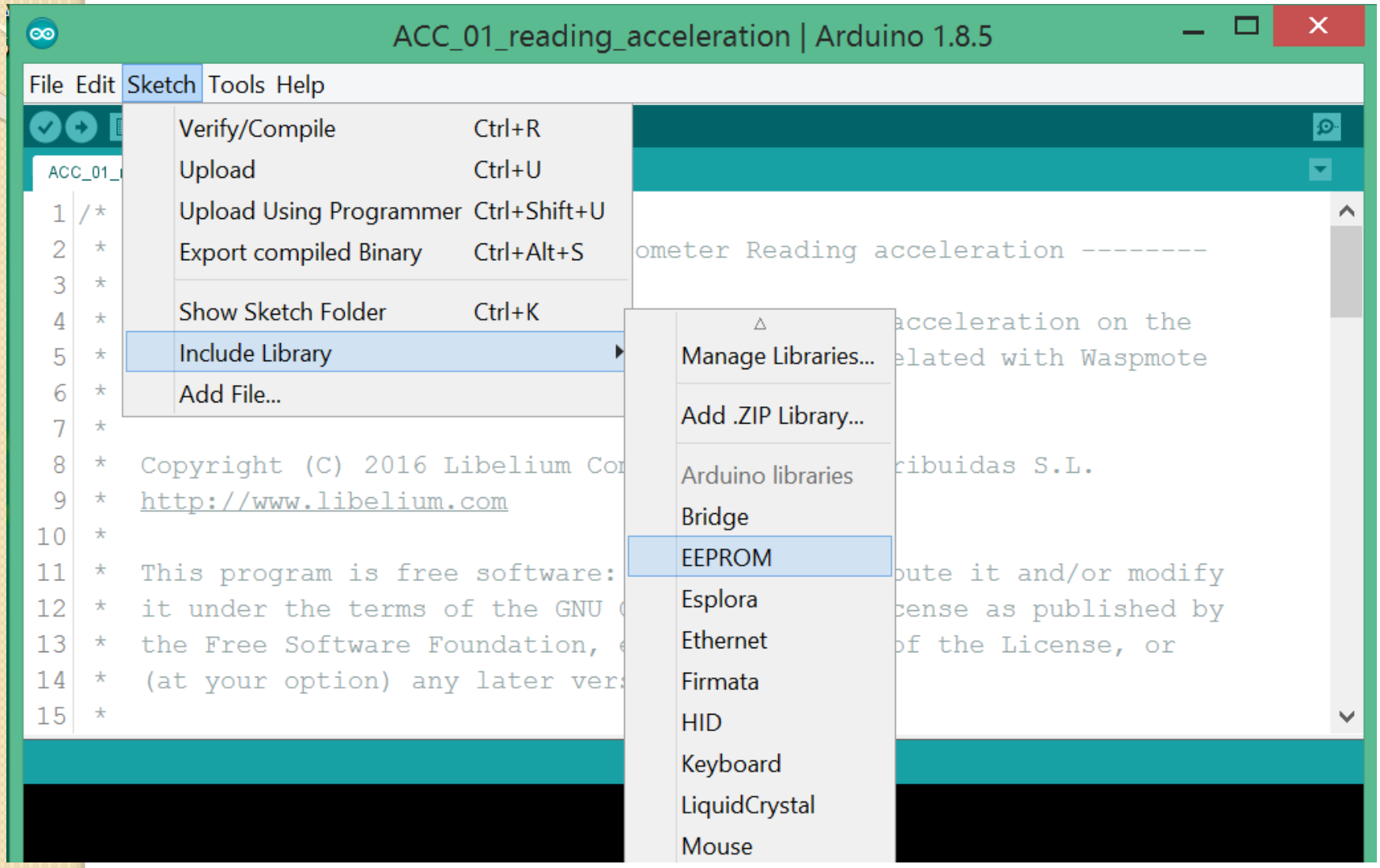
Example: Button



Example: Button

```
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  13;      // the number of the LED pin
int buttonState = 0;         // variable for reading the pushbutton
status
void setup() {
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}
void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

Library Arduino

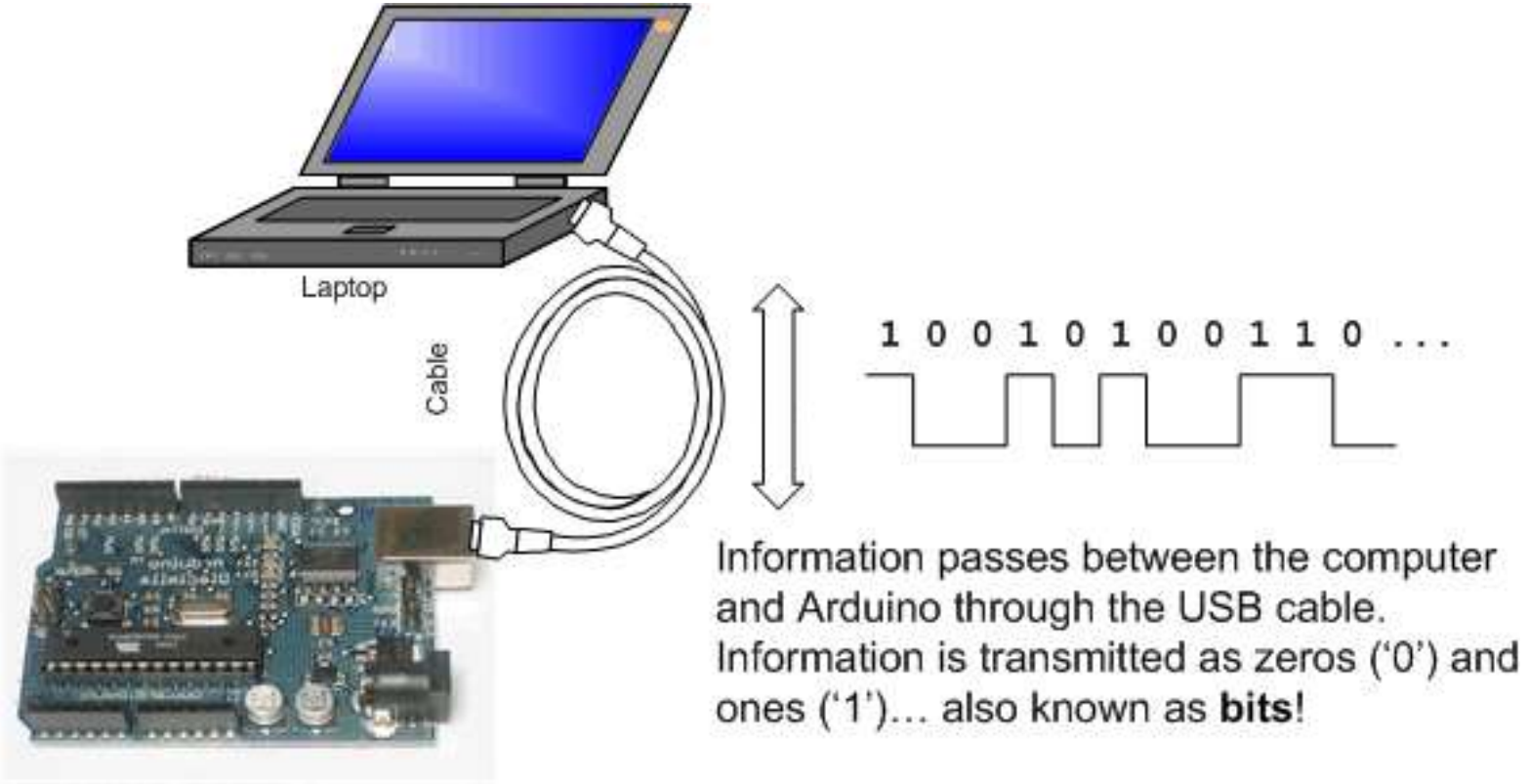


Library Arduino

EEPROM - reading and writing to "permanent" storage
Ethernet / Ethernet 2 - for connecting to the internet using the Arduino Ethernet Shield, Arduino Ethernet Shield 2 and Arduino Leonardo ETH
Firmata - for communicating with applications on the computer using a standard serial protocol.
GSM - for connecting to a GSM/GRPS network with the GSM shield.
LiquidCrystal - for controlling liquid crystal displays (LCDs)
SD - for reading and writing SD cards
Servo - for controlling servo motors
SPI - for communicating with devices using the Serial Peripheral Interface (SPI) Bus
SoftwareSerial - for serial communication on any digital pins. Version 1.0 and later of Arduino incorporate Mikal Hart's NewSoftSerial library as SoftwareSerial.
Stepper - for controlling stepper motors
TFT - for drawing text , images, and shapes on the Arduino TFT screen
WiFi - for connecting to the internet using the Arduino WiFi shield
Wire - Two Wire Interface (TWI/I2C) for sending and receiving data over a net of devices or sensors.

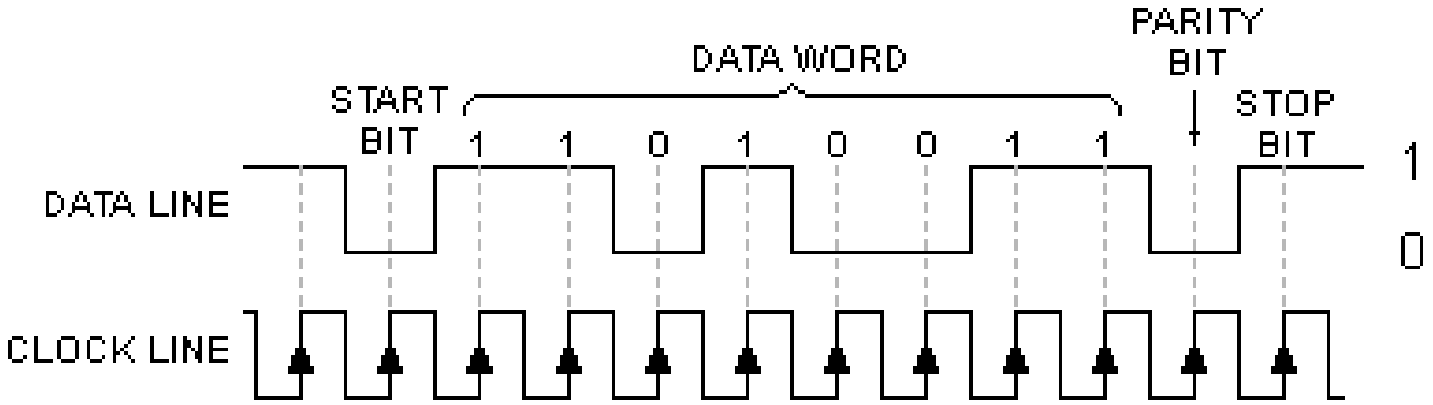
https://www.arduino.cc/en/Reference/Libraries

Example: Serial transmission



Information passes between the computer and Arduino through the USB cable. Information is transmitted as zeros (‘0’) and ones (‘1’)… also known as **bits**!

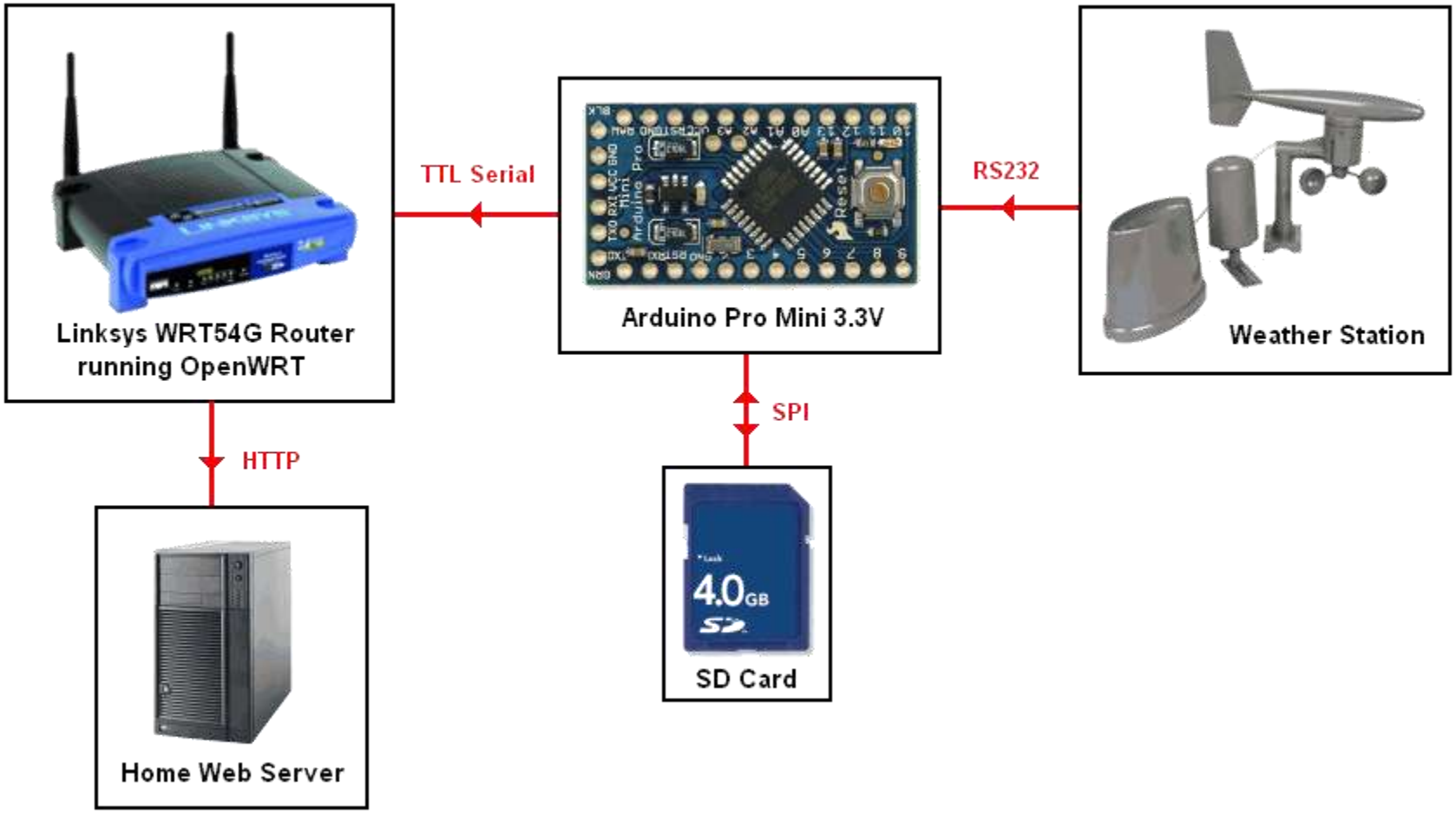
Example: Serial transmission



"standard" baud are 1200, 2400, 4800, 19200, 38400, 57600, and 115200

Eg. 9600 bps

Serial transmission applications

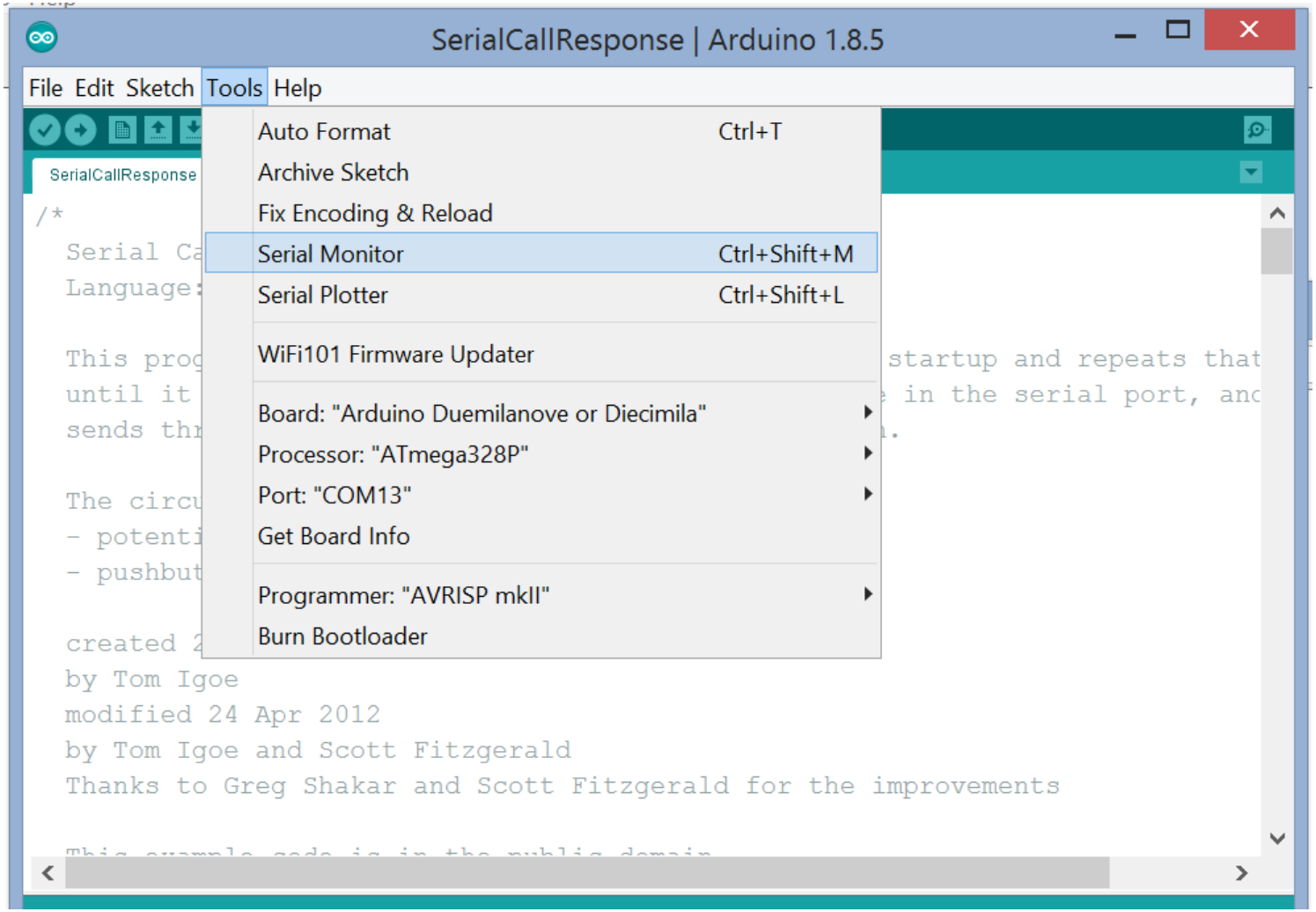


Serial communication

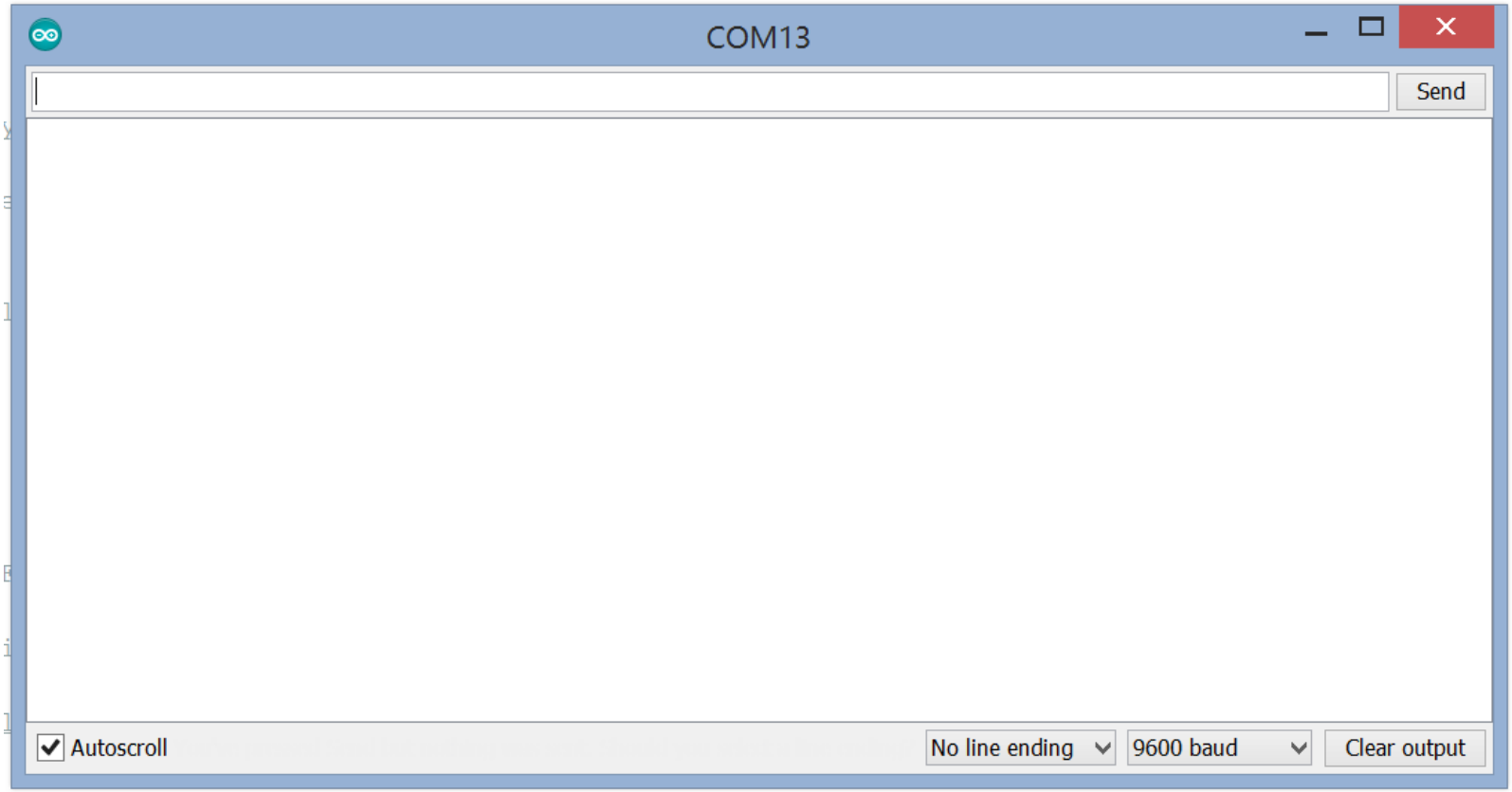
```
void setup() {
  Serial.begin(9600);
  Serial.println("Hello!");
}
void loop()
{

}
```

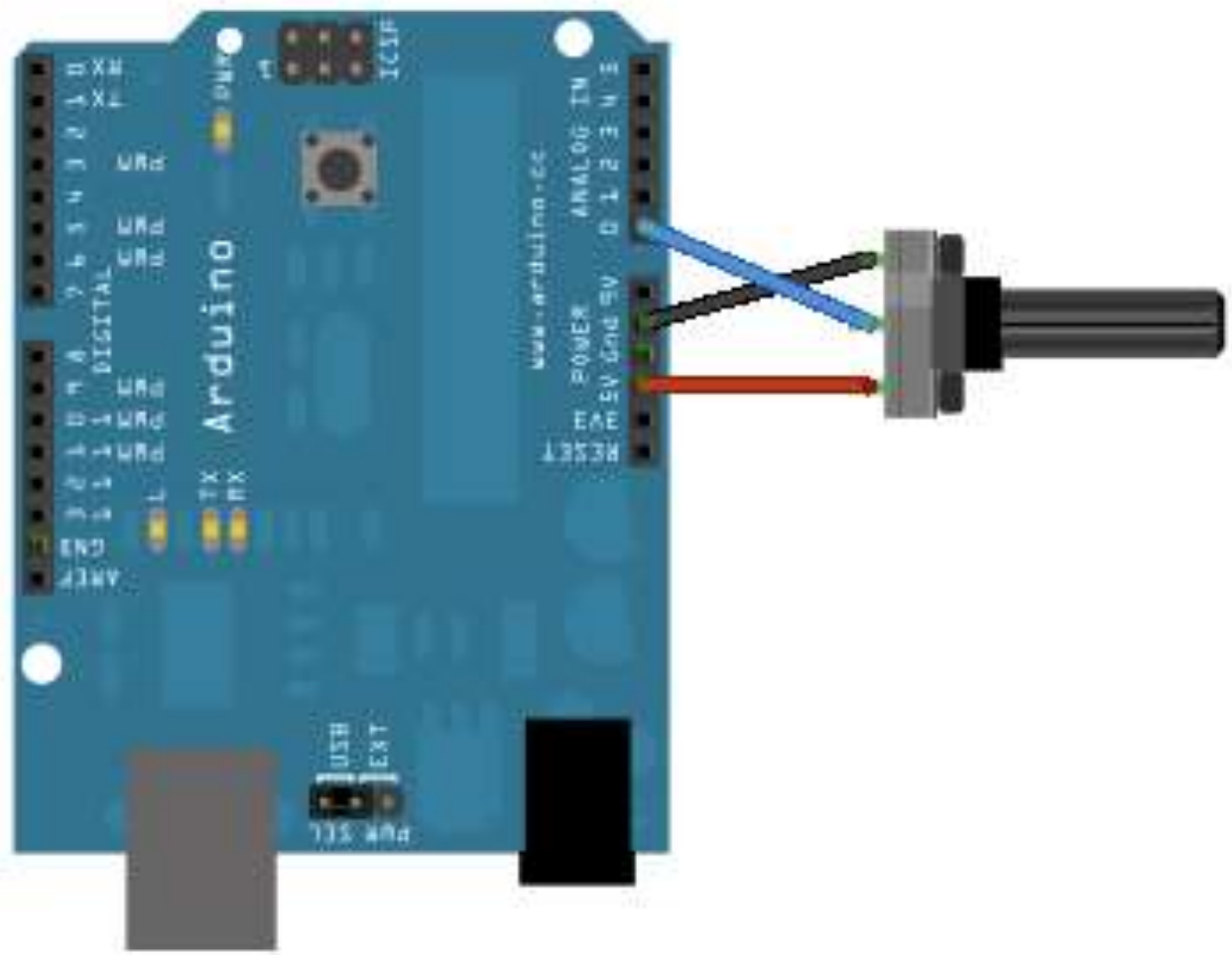
Serial communication



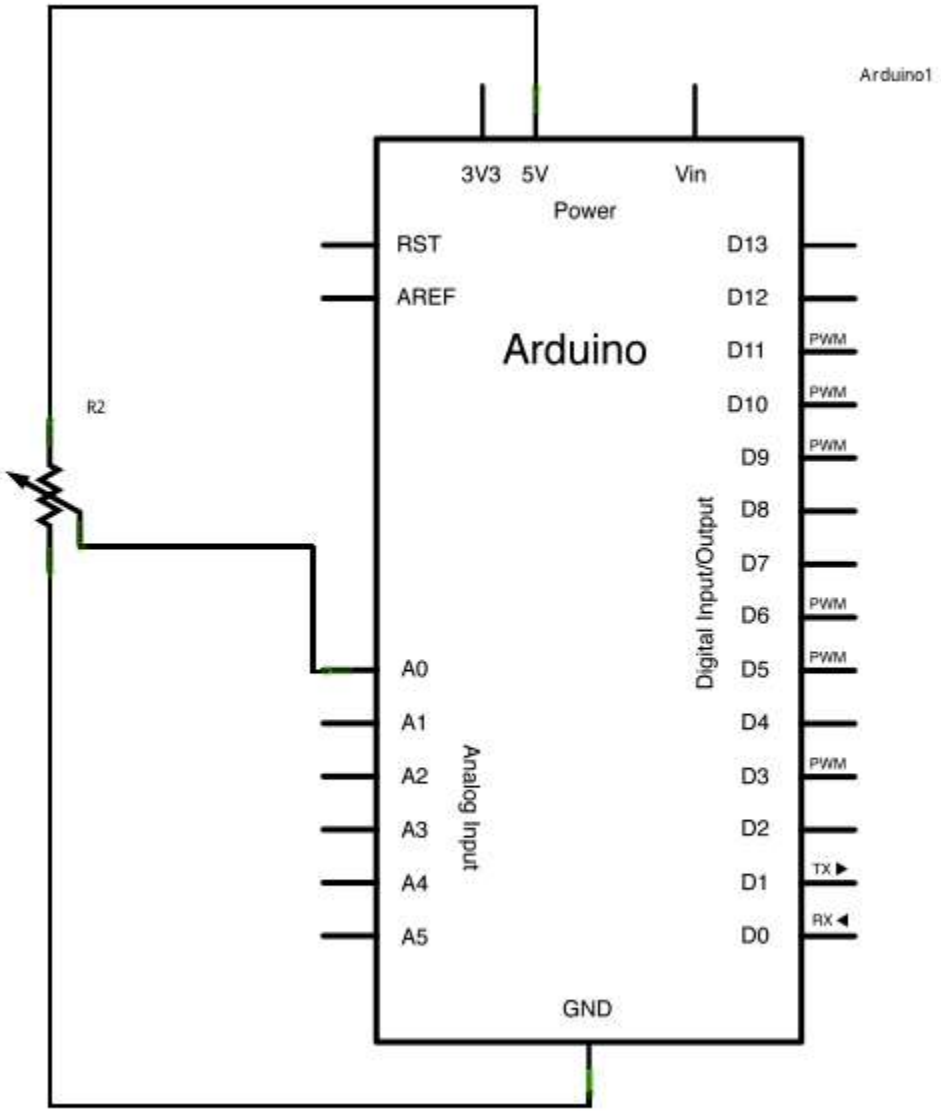
Serial communication



Example: Read analogue value



Example: Read analogue value



Example: Read analogue value

```
void setup() {
  // initialize serial communication
  Serial.begin(9600);
}

void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(100); //delay for stability
}
```

Review

- Types of arduino board
- IDE programming tools
- Basic examples

Questions?