Creative Electronics

Arduino

30-05-2016

TU/e Technische Universiteit
Eindhoven
University of Technology
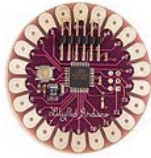
Where innovation starts

Why Arduino?

- **Software vs. Hardware**
  - **Functionality easily changed while HW stays static.**
  - **Some things not possible (e.g. HF)**

- **Large community**
  - **Blog, Forum, Wiki, Website**

Designed Intelligence Group

TU/e Technische Universiteit **Eindhoven** University of Technology

Arduino versions

Arduino Uno

Arduino LilyPad

Arduino Ethernet

Arduino Nano

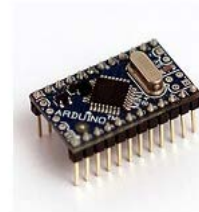Arduino BT

Arduino Mini

Arduino Mega 2560

Arduino Fio

Arduino BT

Arduino Mini

USB/Serial Light Adapter

Arduino Pro Mini

Arduino Mega ADK

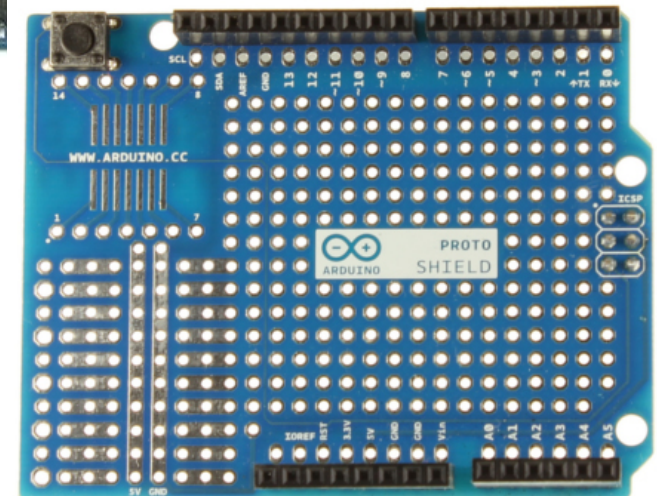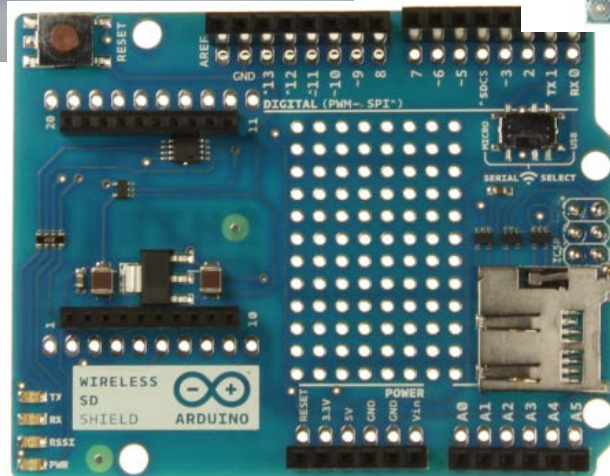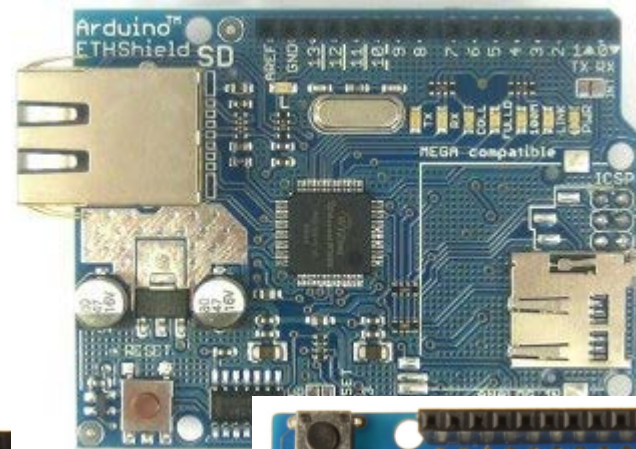Arduino Pro

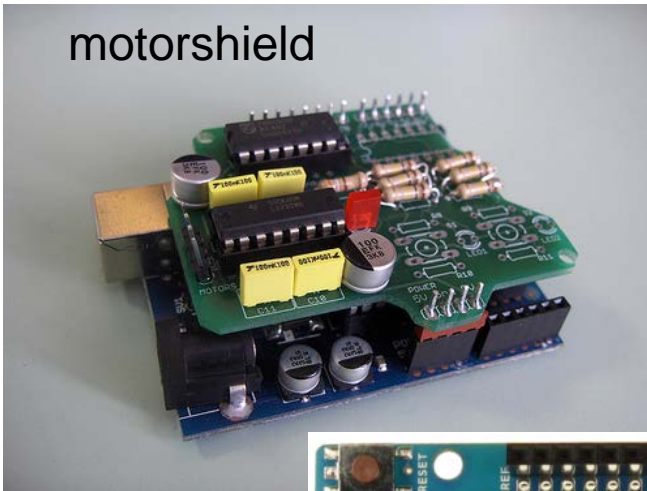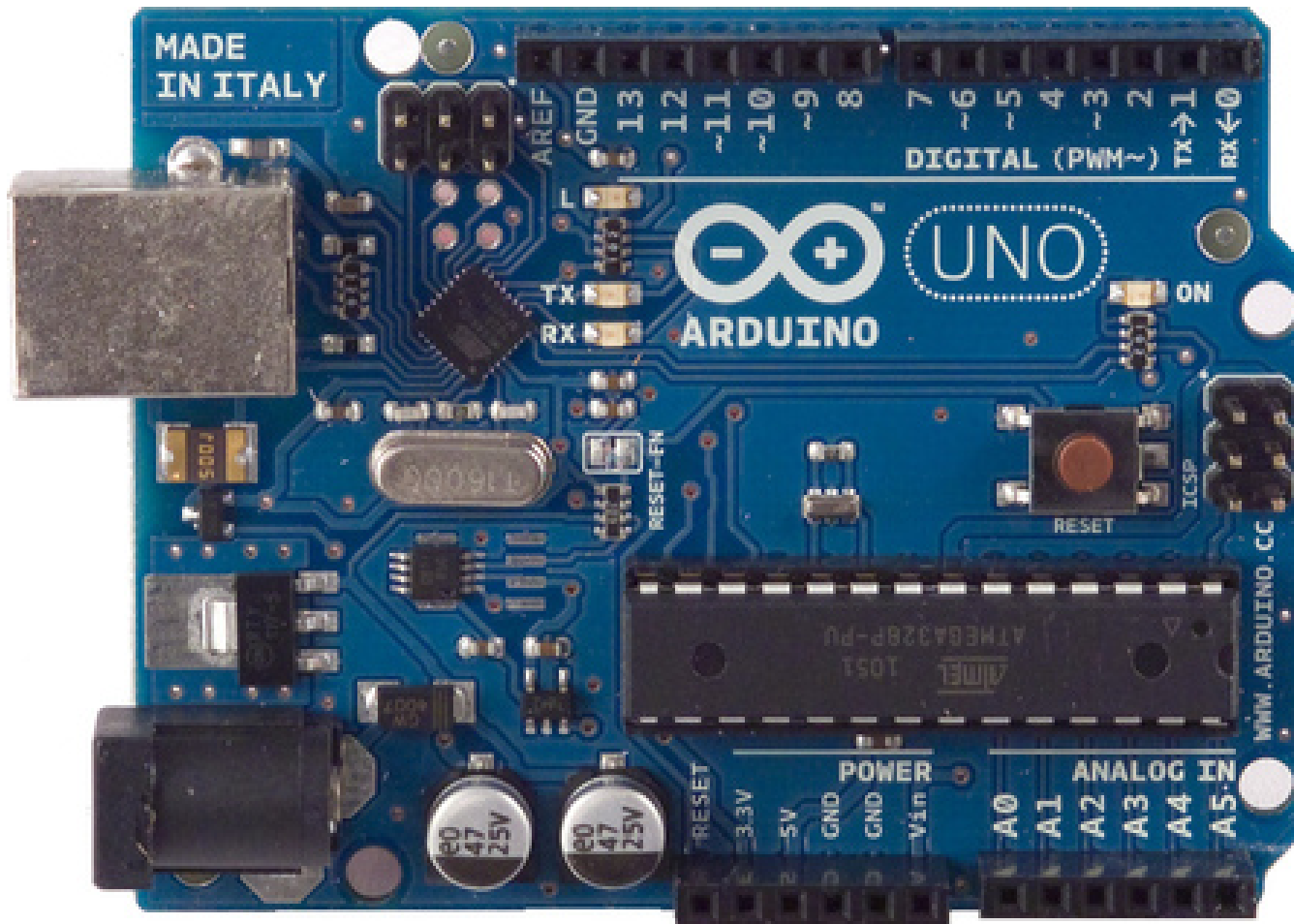USB/Serial Light Adapter

Arduino Pro Mini

Additional hardware: shields

- **Add-on electronics modules for adding functionality**

motorshield

Arduino UNO

Arduino UNO

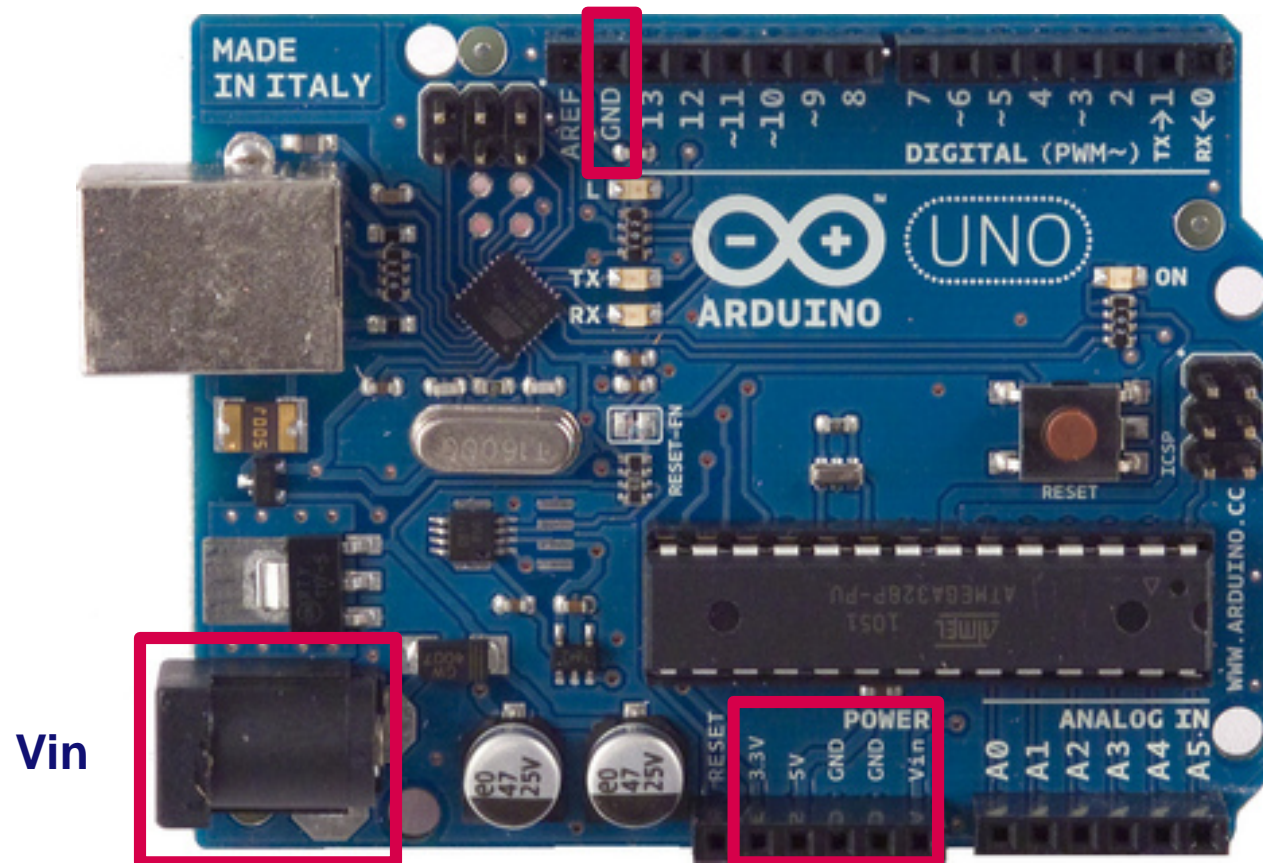|	
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (VIN) (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM (Static RAM)	2 KB (ATmega328)
EEPROM (Electrically erasable programmable ROM)	1 KB (ATmega328)
Clock Speed	16 MHz

TU/e Technische Universiteit Eindhoven University of Technology

Arduino UNO

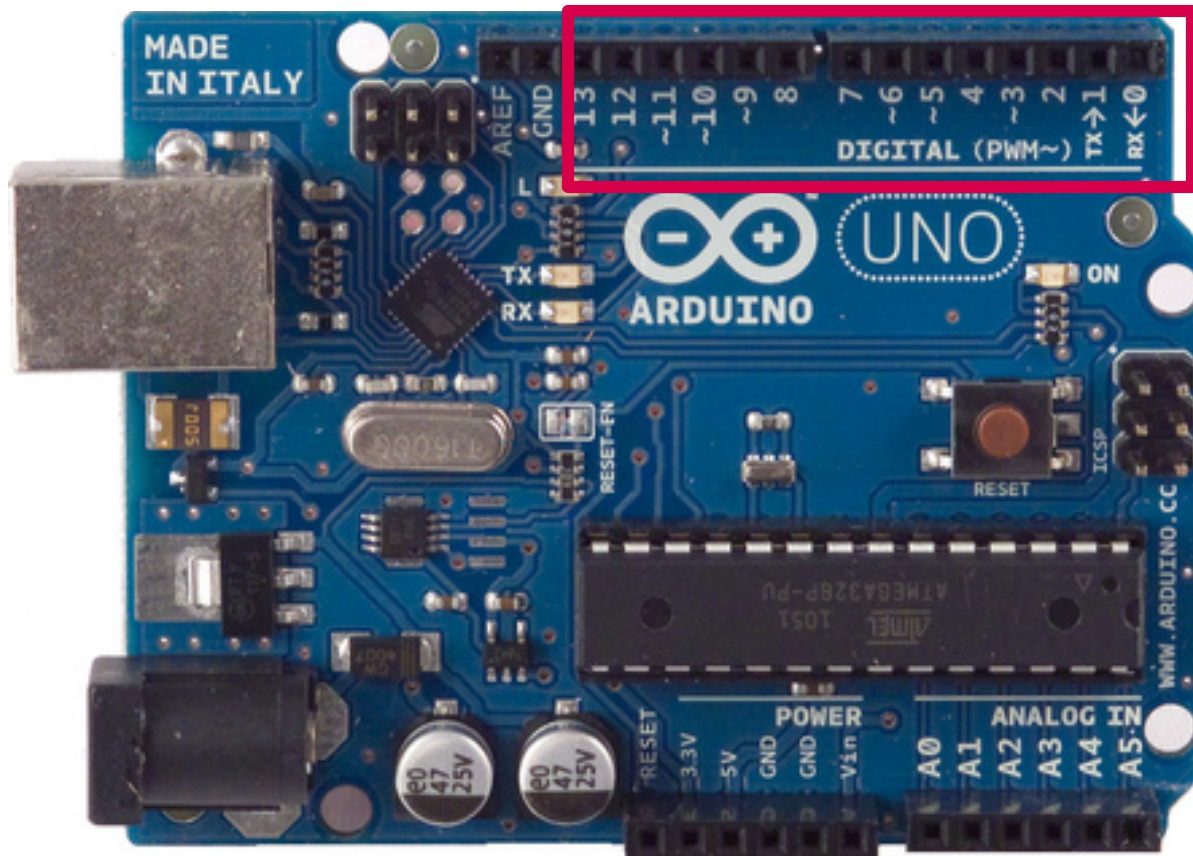- **Power: USB Power supply (5V)**

USB
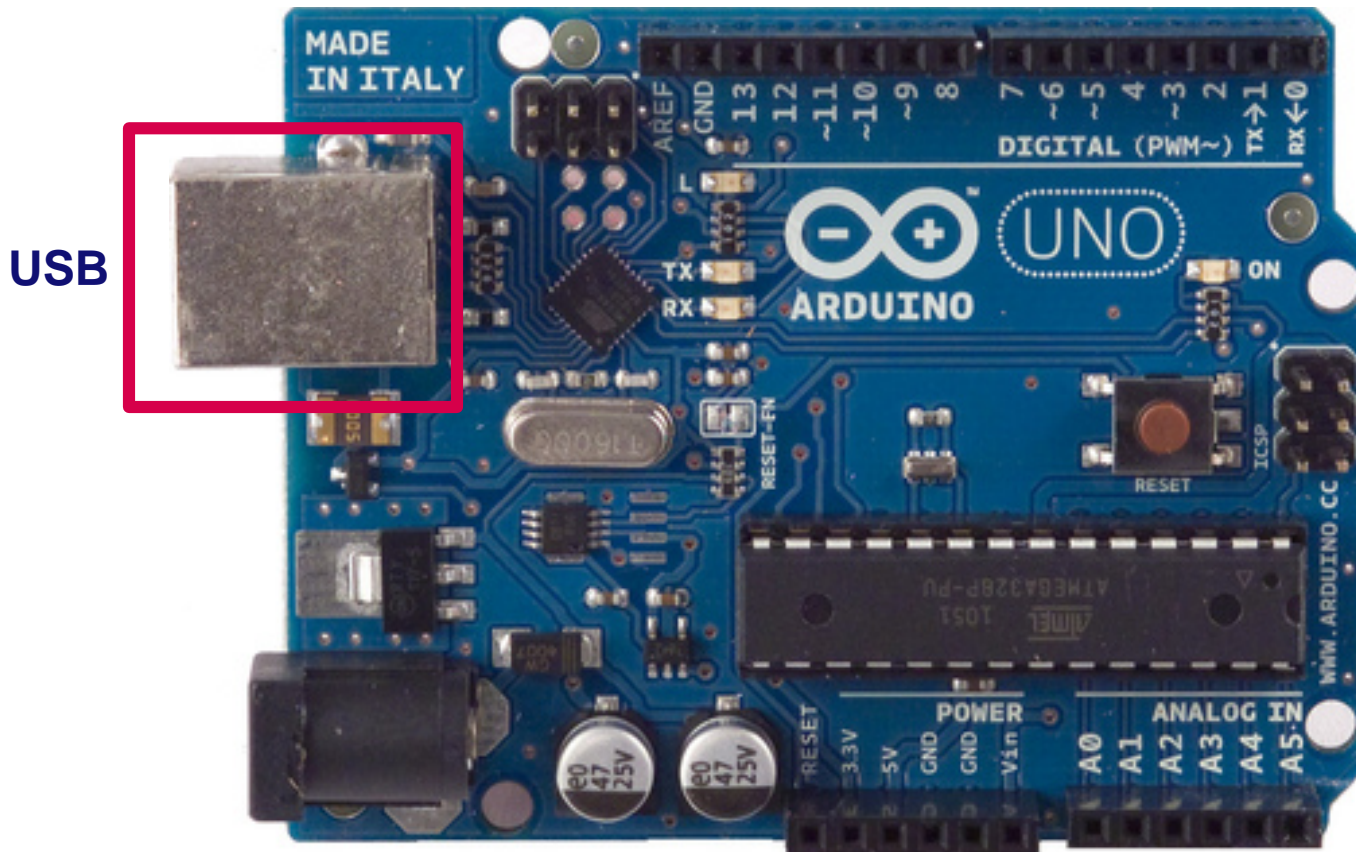
Arduino UNO

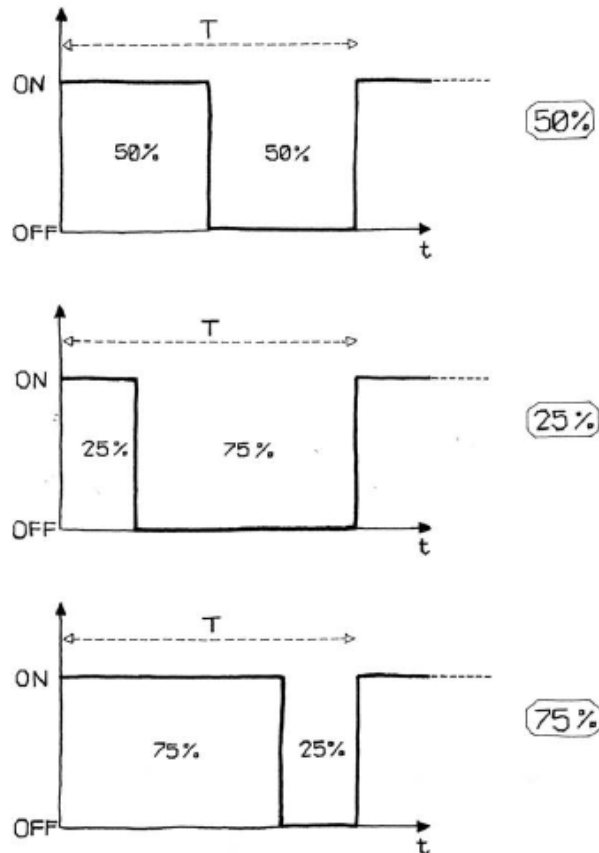- **Power: Vin (7-12V), 5V, 3.3V, GND**

Vin

Arduino UNO

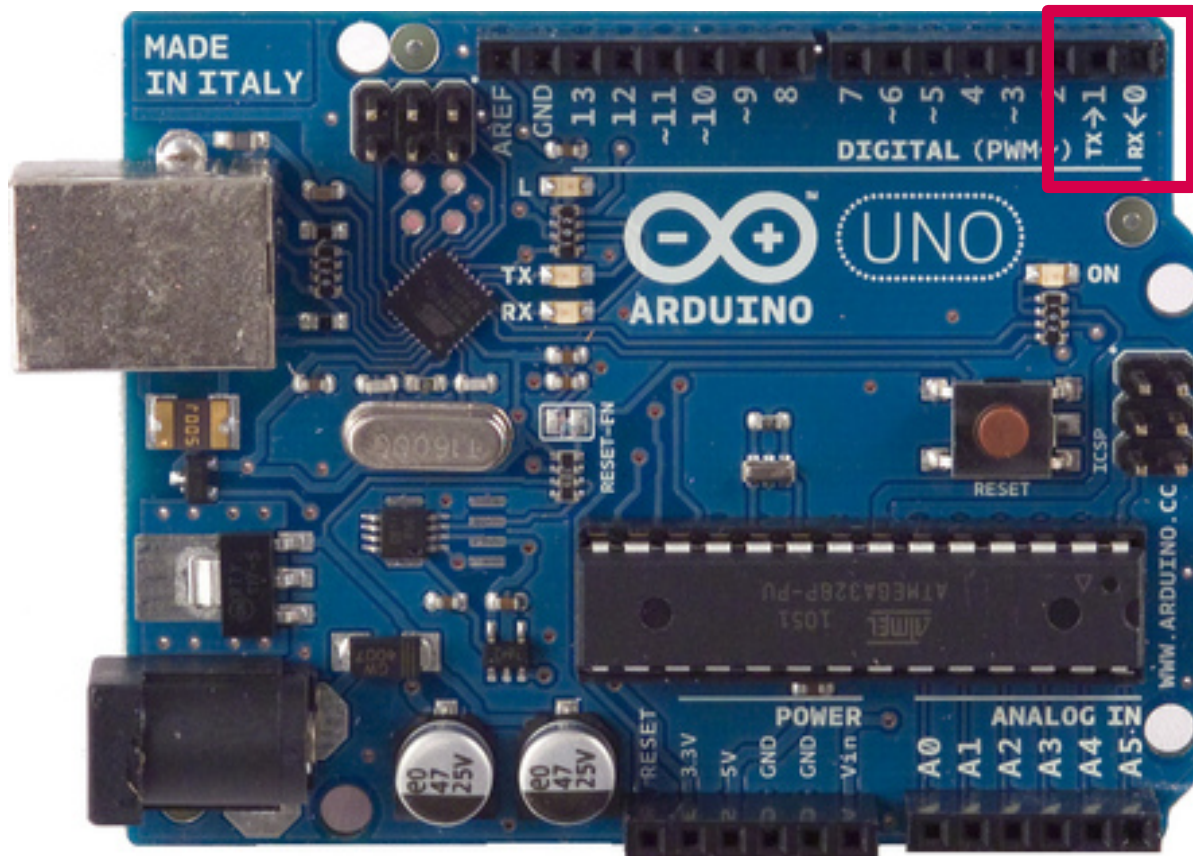- **14 Digital I/O (6 can provide PWM)**

Arduino UNO

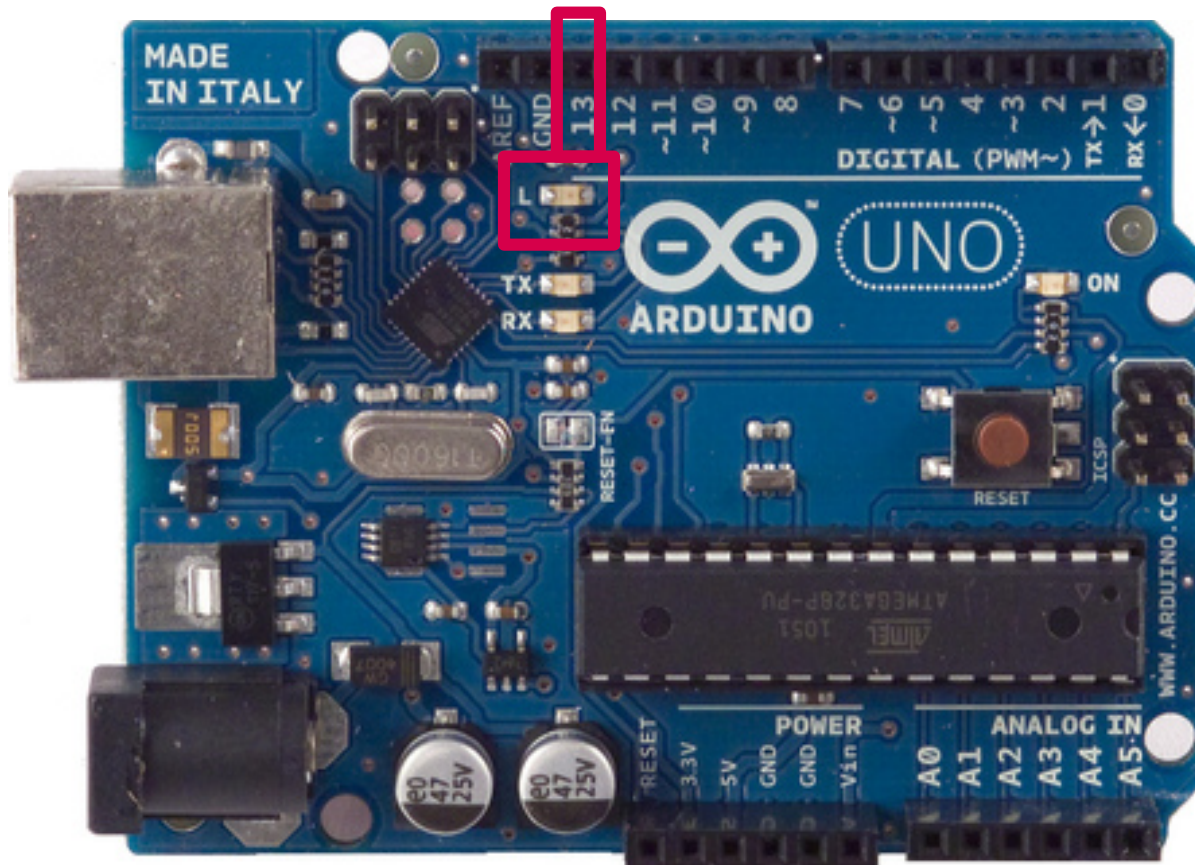- **14 Digital I/O (6 can provide PWM)**

Arduino UNO

- **Serial: receive (RX) and transmit (TX)**

Arduino UNO

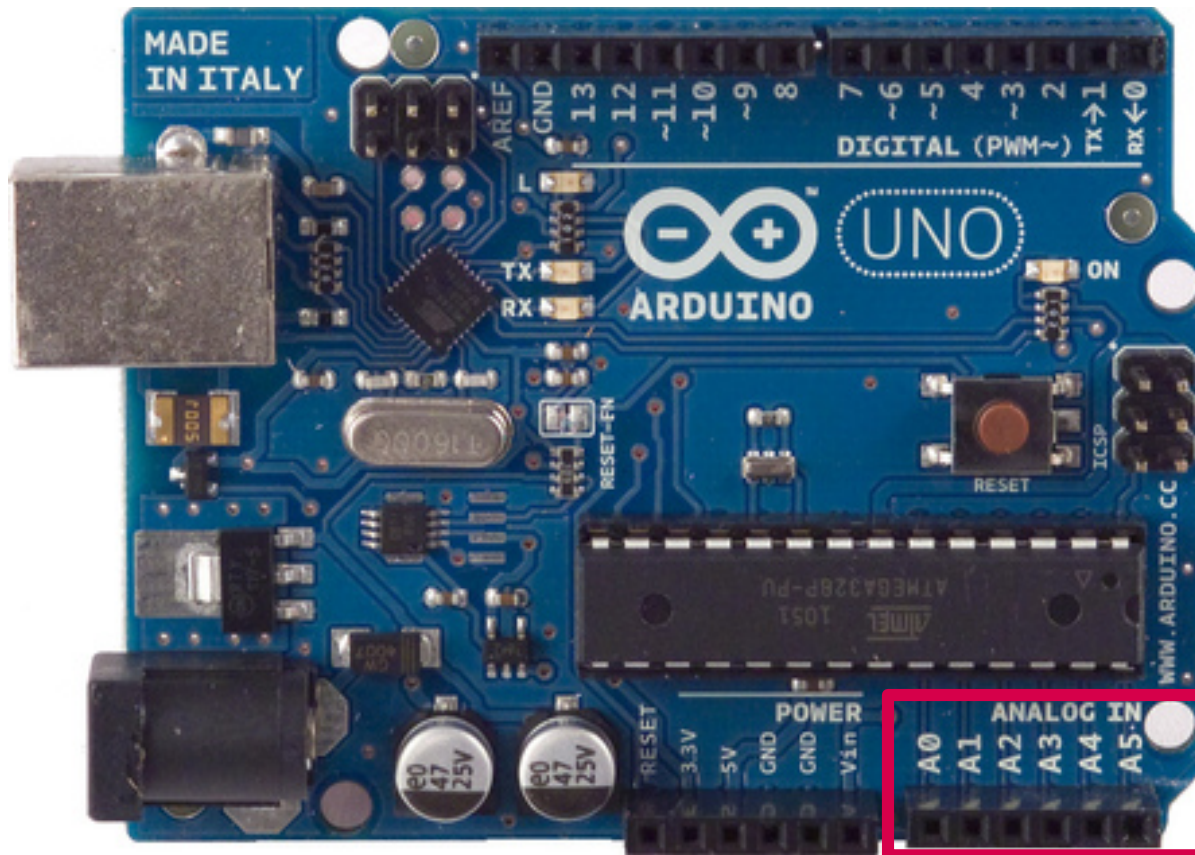- **LED: Pin 13**

Arduino UNO

- **6 analog inputs, 10 bits resolution (1024 values)**

Designed Intelligence Group

TU/e Technische Universiteit Eindhoven University of Technology
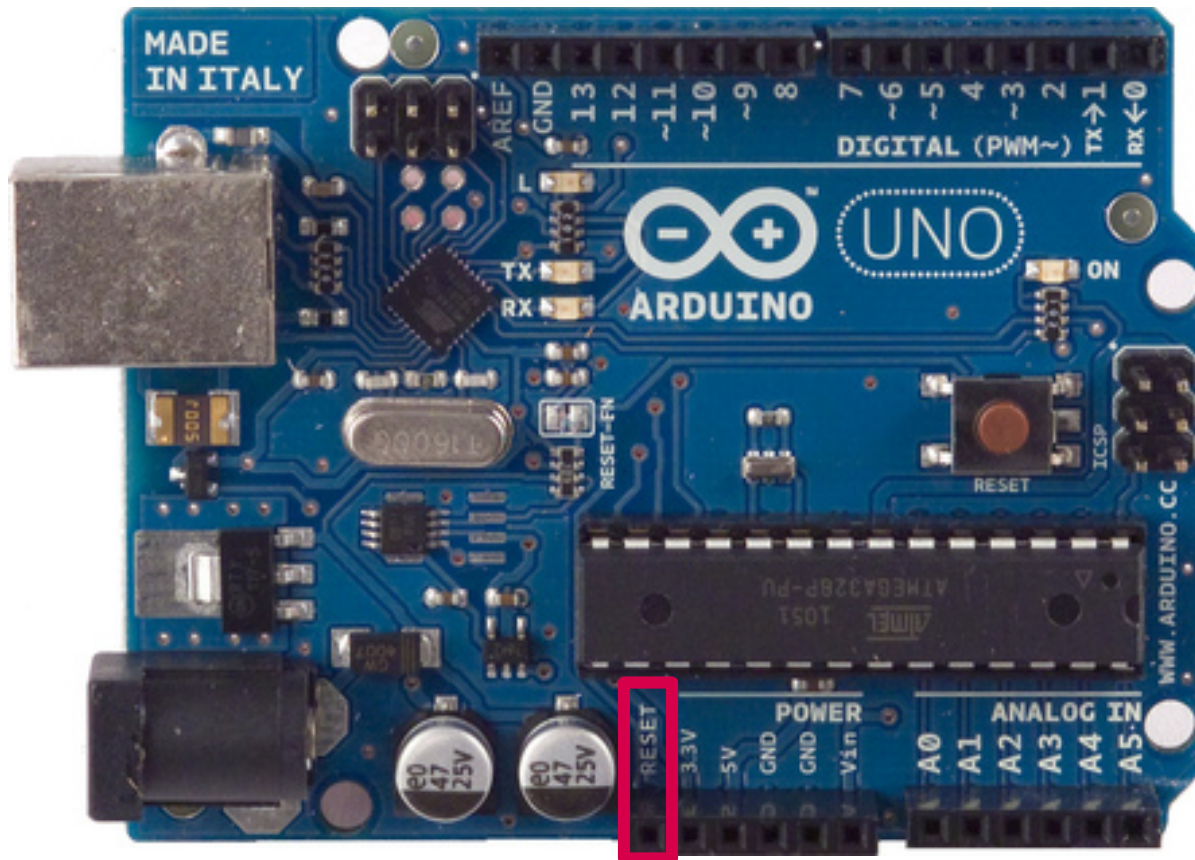
Arduino UNO

- **AREF: Reference voltage for the analog inputs**
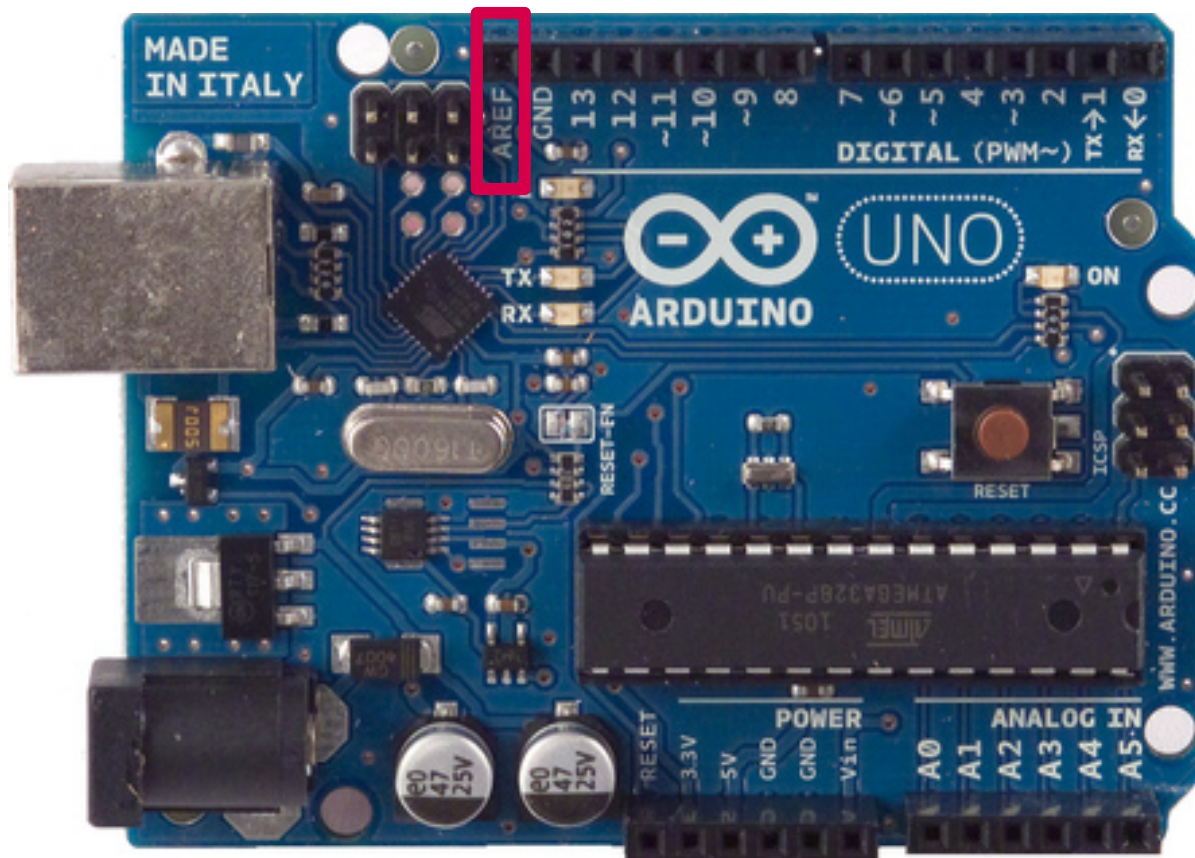
Arduino UNO

- **Reset. LOW to reset the microcontroller**

Software: Arduino IDE

- **http://arduino.cc/en/Main/Software**
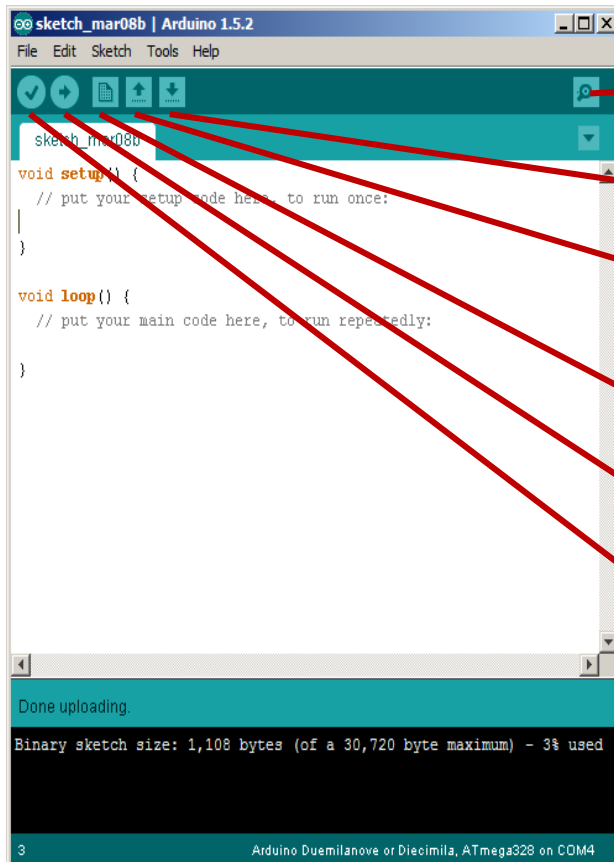
Serial Monitor

Save Sketch

Open Sketch
(old: *.pde, new:
*.ino)

New Sketch

Compile and Upload

Verify (Compile)

Driver Installation and Port Identification

- **Refer to the instructions in**
  - **http://arduino.cc/en/Guide/HomePage**
- **Check the wiki**
  - **http://wiki.id.tue.nl/ce/CreativeElectronicsAssignment201509#Installing_the_Arduino_IDE**

Designed
Intelligence
Group

TU/e Technische Universiteit
Eindhoven
University of Technology

Really getting started

Sense/Perceive

Sensors

Arduino

Actuators

Act/React
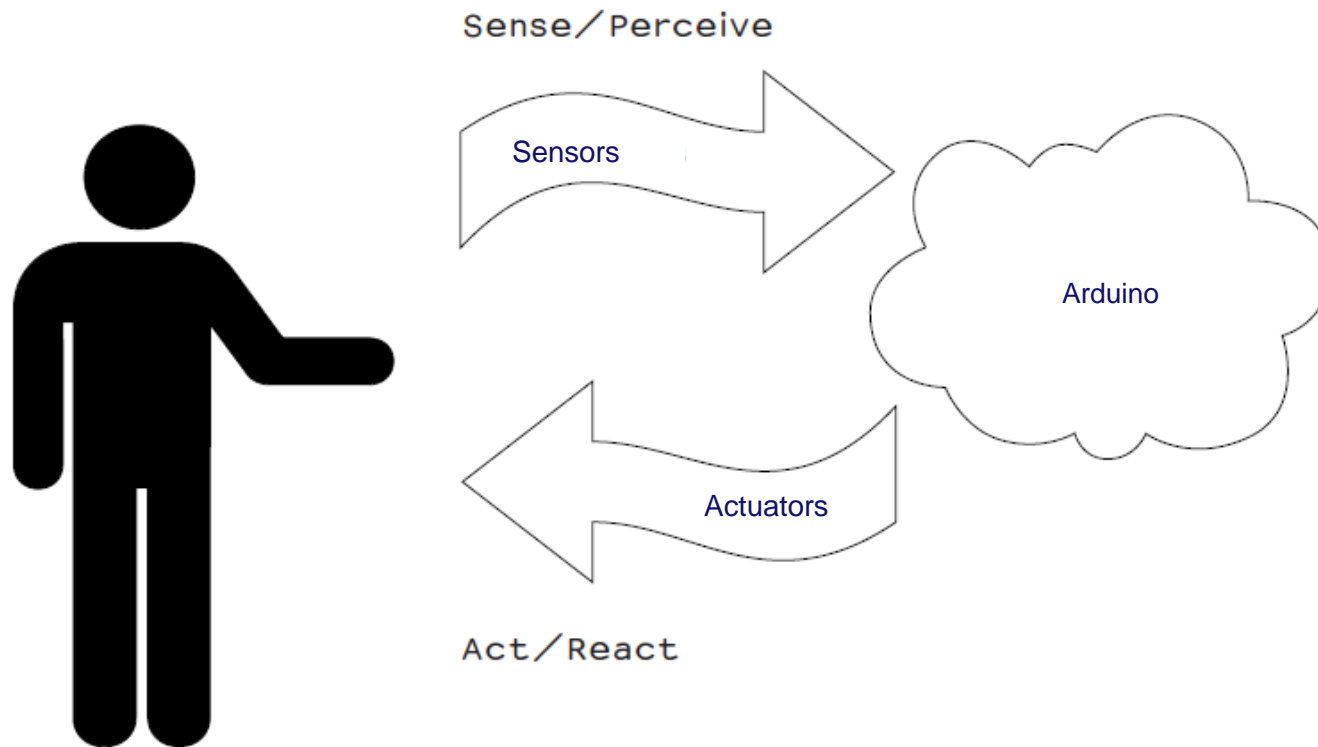
Common sensors
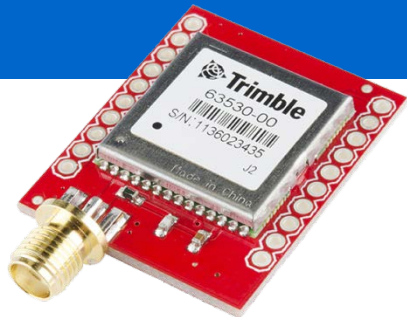
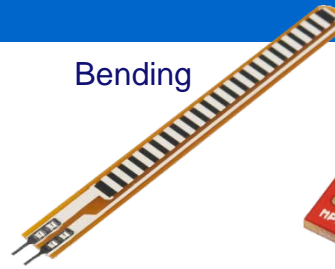Position (GPS)
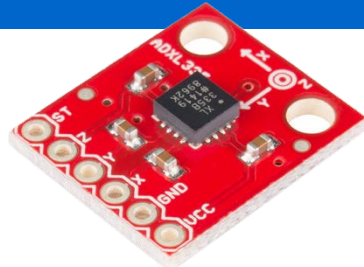
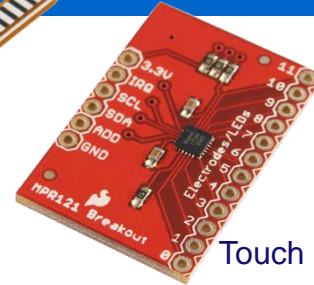Acceleration

Bending

Touch

Distance

Force
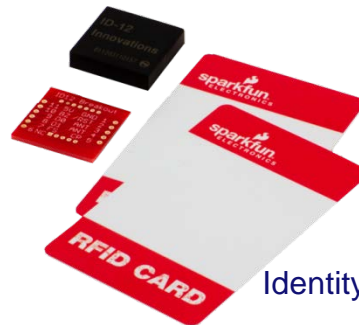
Identity
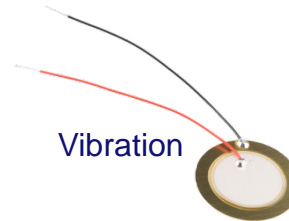
Vibration
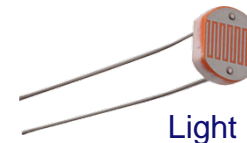
Light

Sound

Movement

Temperature

On-off status

Setting or Orientation

Rotation

Tilt

Designed Intelligence Group

TU/e Technische Universiteit Eindhoven University of Technology
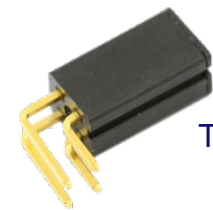
Common actuators

DC motor
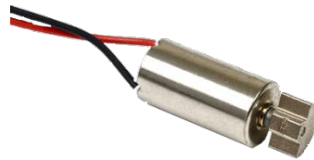
Vibration motor

Display
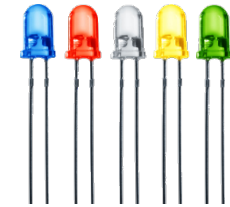
Led
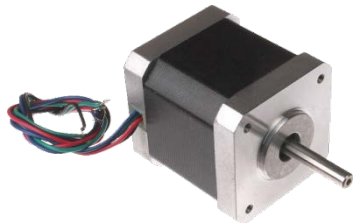
Stepper motor

Solenoid

Buzzer

Servo motor

Fan

Speaker

Heating pad
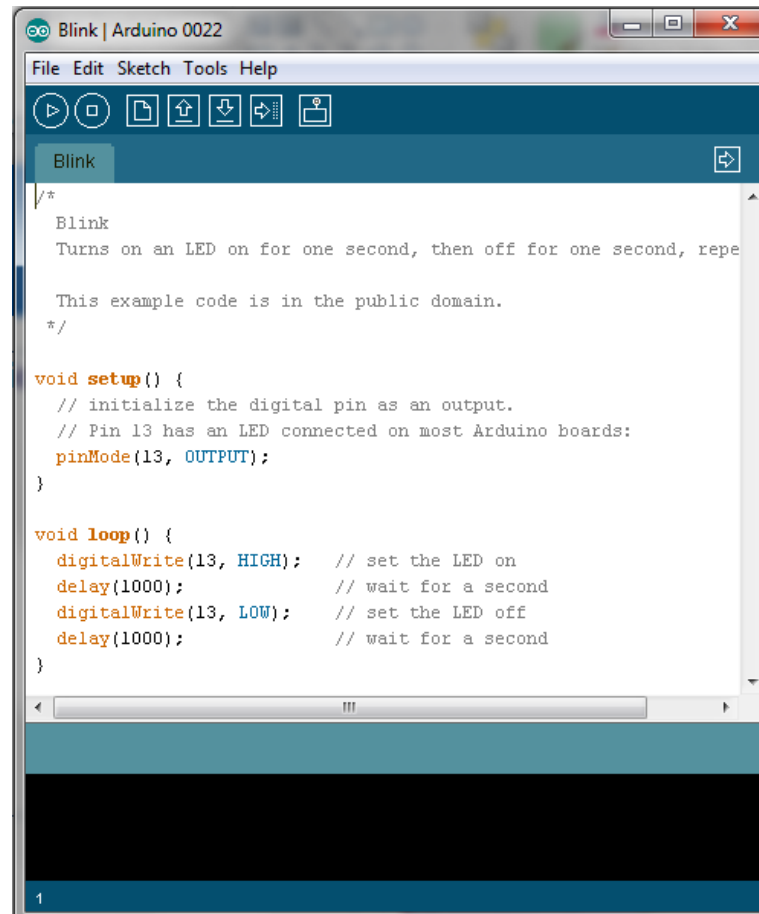
Breadboard
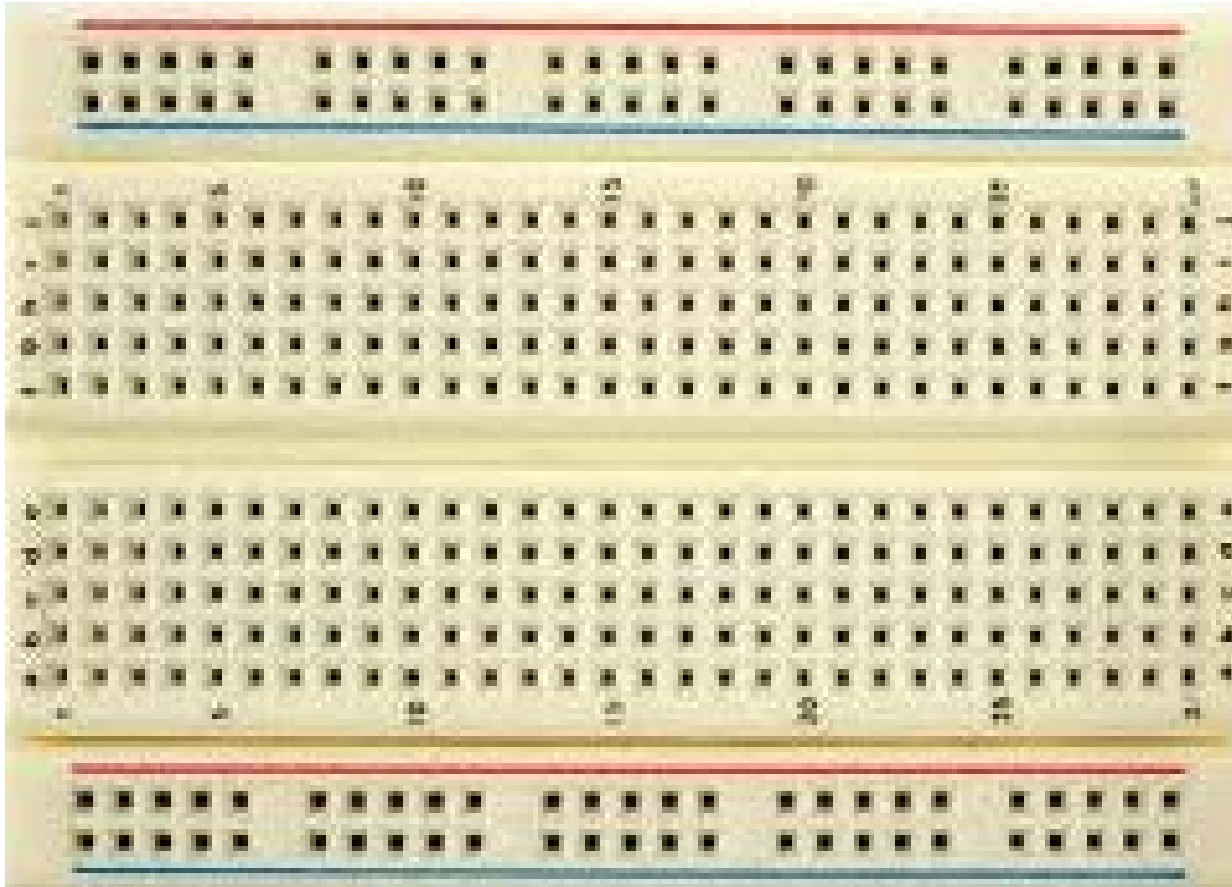
Blinking a LED

- **File>Examples>Basics>Blink**
  - **LED: light-emitting diode**

Blinking a LED

Blinking a LED

```
#define LED 13

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


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

Button to control a LED

- **File>Examples>Digital>Button**

Button to control a LED
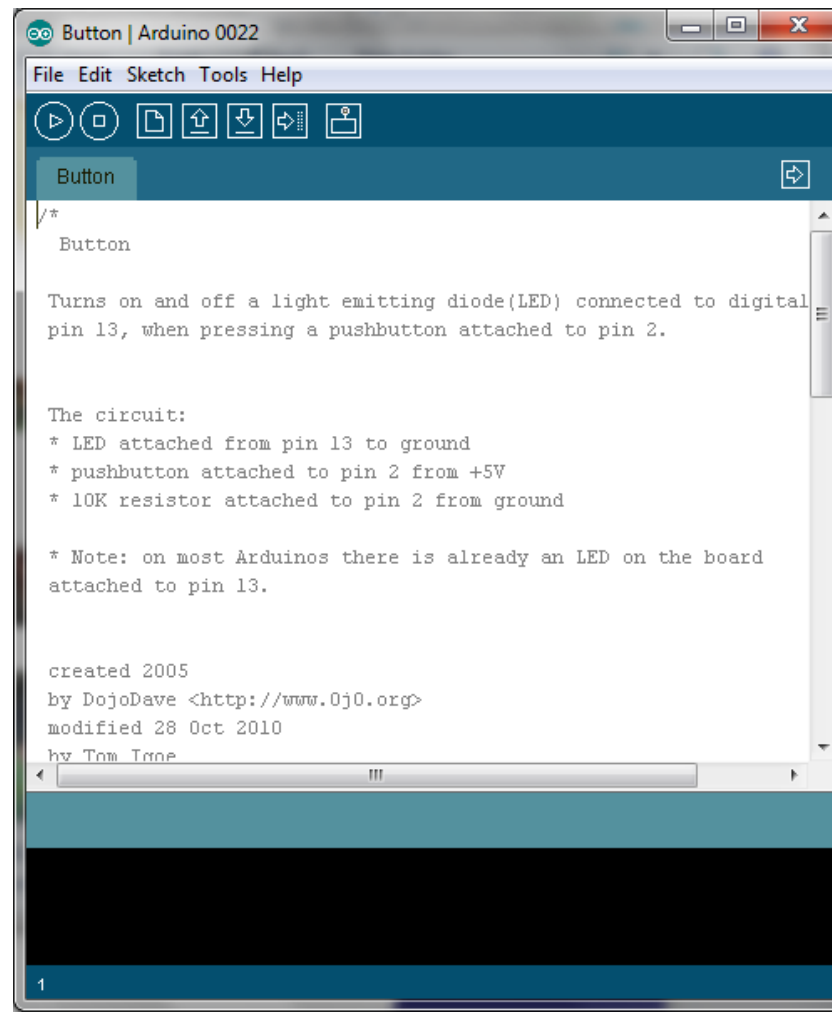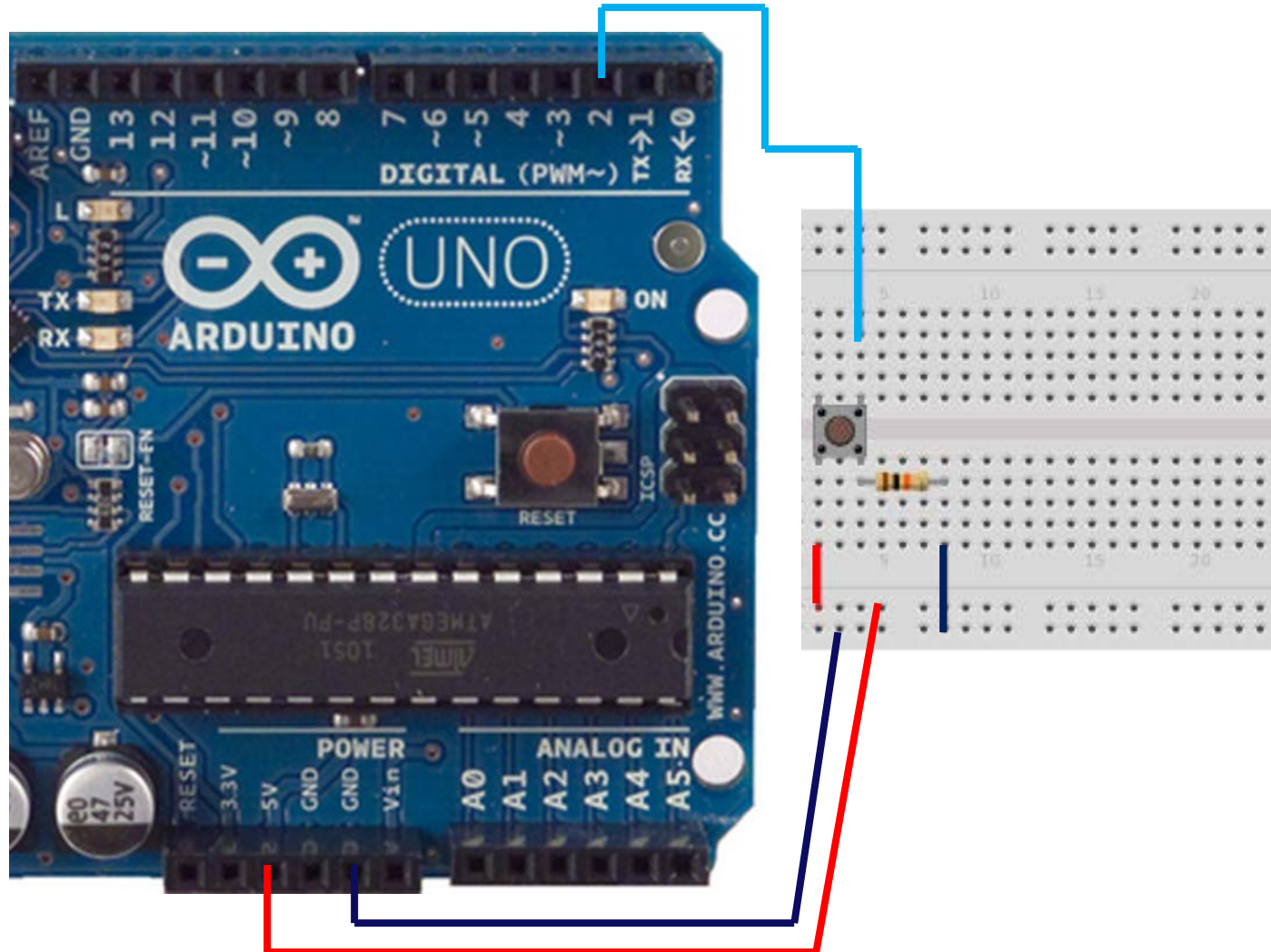
Button to control a LED

```
// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 2;      // the number of the pushbutton pin
const int ledPin =  13;       // the number of the LED pin

// variables will change:
int buttonState = 0;          // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}
```

Designed Intelligence Group

TU/e Technische Universiteit Eindhoven University of Technology

Button to control a LED

```
void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

Analog I/O

- **File>Examples>Analog>AnalogInput**
  - **Instead of a potentiometer, we use a light sensor**
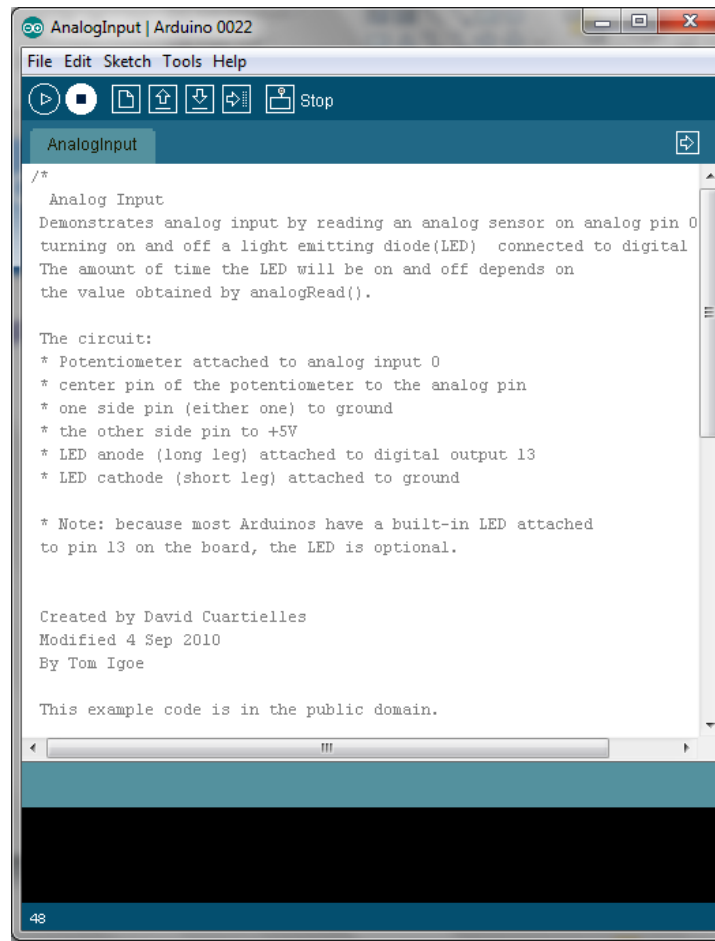
Analog I/O
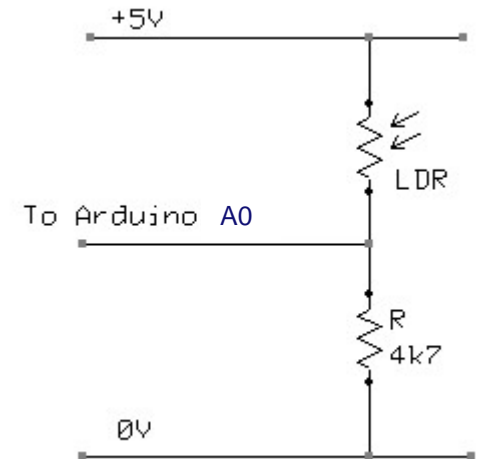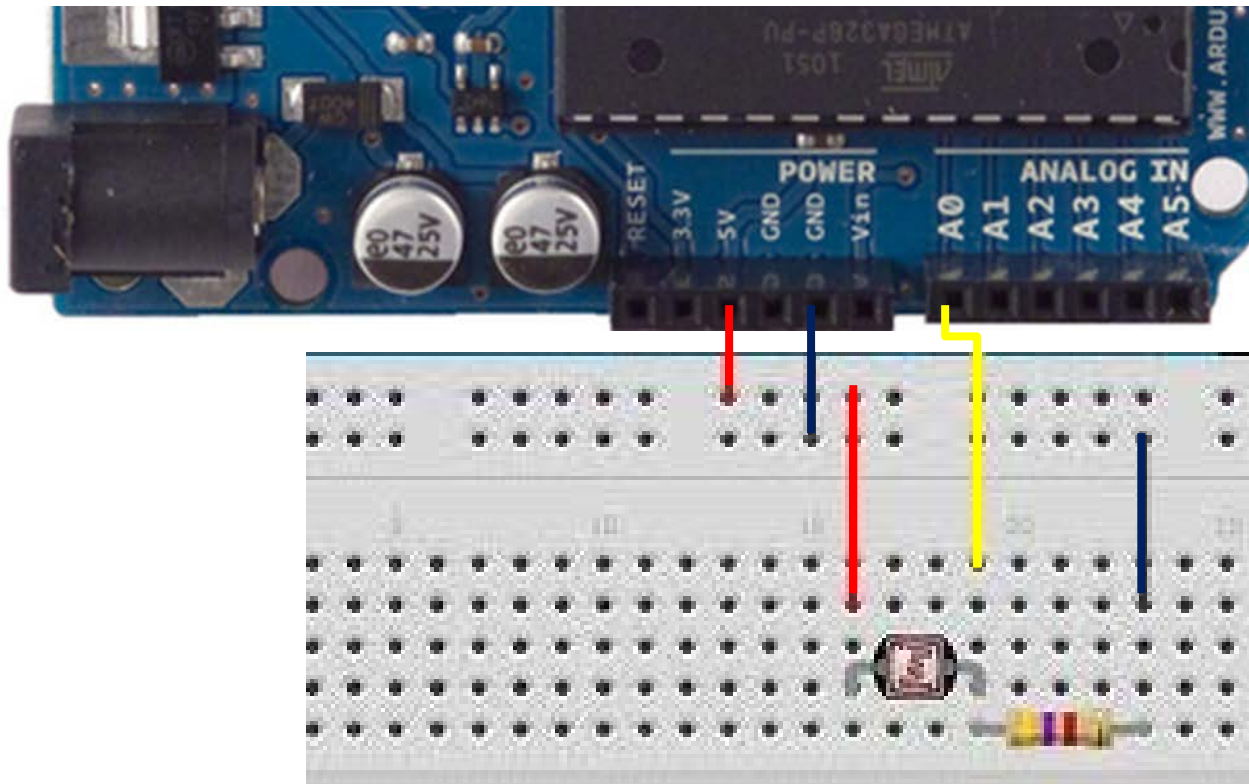
Analog Input

```
int sensorPin = A0;     // select the input pin for the potentiometer
int ledPin = 13;        // select the pin for the LED
int sensorValue = 0;    // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  // turn the ledPin on
  digitalWrite(ledPin, HIGH);
  // stop the program for <sensorValue> milliseconds:
  delay(sensorValue);
  // turn the ledPin off:
  digitalWrite(ledPin, LOW);
  // stop the program for for <sensorValue> milliseconds:
  delay(sensorValue);
}
```

Designed Intelligence Group

TU/e Technische Universiteit Eindhoven University of Technology

Analog I/O

```
int sensorPin = A0;     // select the input pin
int ledPin = 11;        // select the pin for the LED
int sensorValue = 0;    // variable to store the value coming from the sensor

void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  // turn the ledPin on
  analogWrite(ledPin, sensorValue/4);
}
```

PWM pin!!!

Serial Communication

```
// Example 11-06 from "Getting Started with P
// by Reas & Fry. O'Reilly / Make 2010

// Note: This is code for an Arduino board, n

int sensorPin = 0;   // Select input pin
int val = 0;


void setup() {
  Serial.begin(9600);   // Open serial port
}


void loop() {
  val = analogRead(sensorPin) / 4;   // Read value from sensor
  Serial.println(val);   // Print variable to serial port
  delay(100); // Wait 100 milliseconds
}
```

COM22

Send

91
89
88
86
85
86
90
93
96
98
98
98
97
96

☑ Autoscroll   Carriage return ▾   9600 baud ▾

Designed Intelligence Group

TU/e Technische Universiteit Eindhoven University of Technology
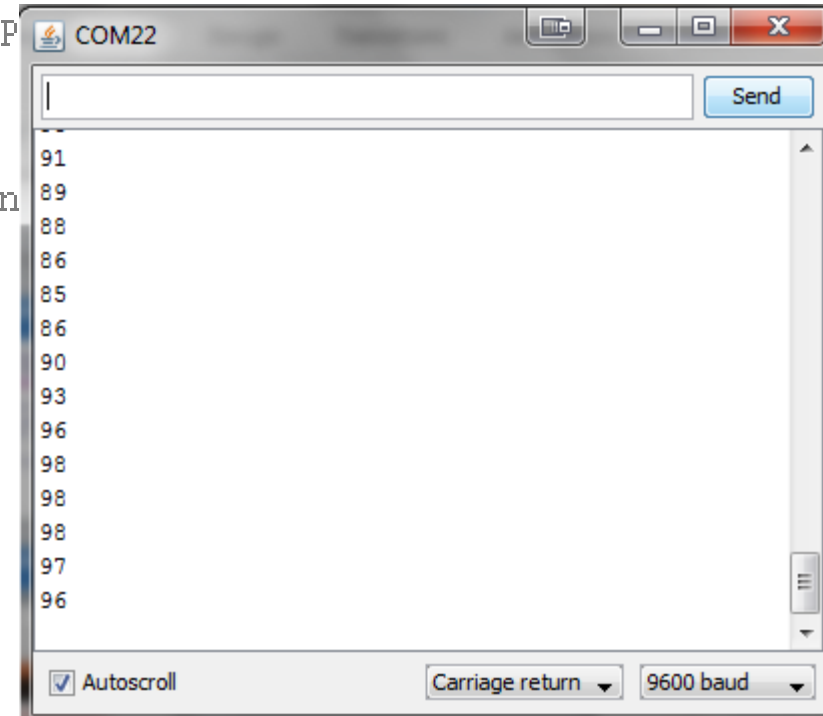
Serial communication

- **Allows Arduino to communicate**
  - **via USB cable / COM port**
  - **via WiFi module**
  - **via Xbee module**
  - **via Bluetooth module**
  - **…**

- **Monitoring**
  - **Monitoring and serial communication can NOT be done at the same time through RX/TX**

Take care

- **Serial.print: translates to *text***
  - **int value 12 is sent as 2 characters '1' and '2'**
  - **float value 1.23 is sent as 4 characters '1', '.', '2' and '3'**
  - **String value "Hi!" is sent as 3 characters 'H', 'i' and '!'**
- **Serial.println: does the same + sends end of line**

- **Serial.write: if you want to senda single**

Designed Intelligence Group

TU/e Technische Universiteit Eindhoven University of Technology

Examples

- **Bad code / Good code**

- **Arduino communicating to Processing**

SerialPrint

SensorGraph

SensorRead

Designed
Intelligence
Group

Technische Universiteit
Eindhoven
University of Technology