

Arduino



TU / **e**

Technische Universiteit
Eindhoven
University of Technology

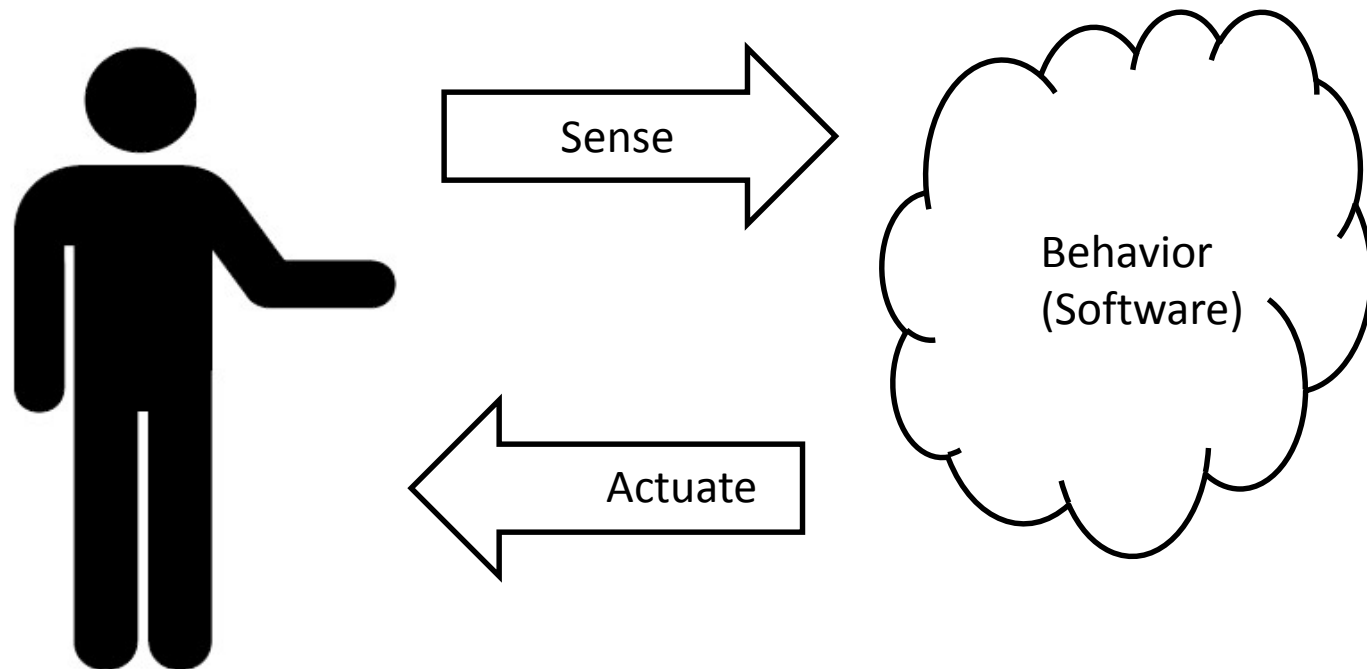
Today

- **Arduino Board**
- **Blink an LED**
- **Digital Input**
- **Analog Input**
- **Analog Output**

Why Arduino?

- **'Physical' Computing**
 - **Fast prototyping**
 - **Easy to use**
 - **Used by designers and artists**
- **Large user community**
 - **Blog, Forum, Playground (wiki)**

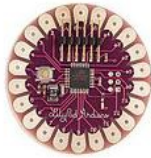
Really getting started



Arduino board types (non exhaustive)



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



Designed
Intelligence
Group

TU/e

Technische Universiteit
Eindhoven
University of Technology

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

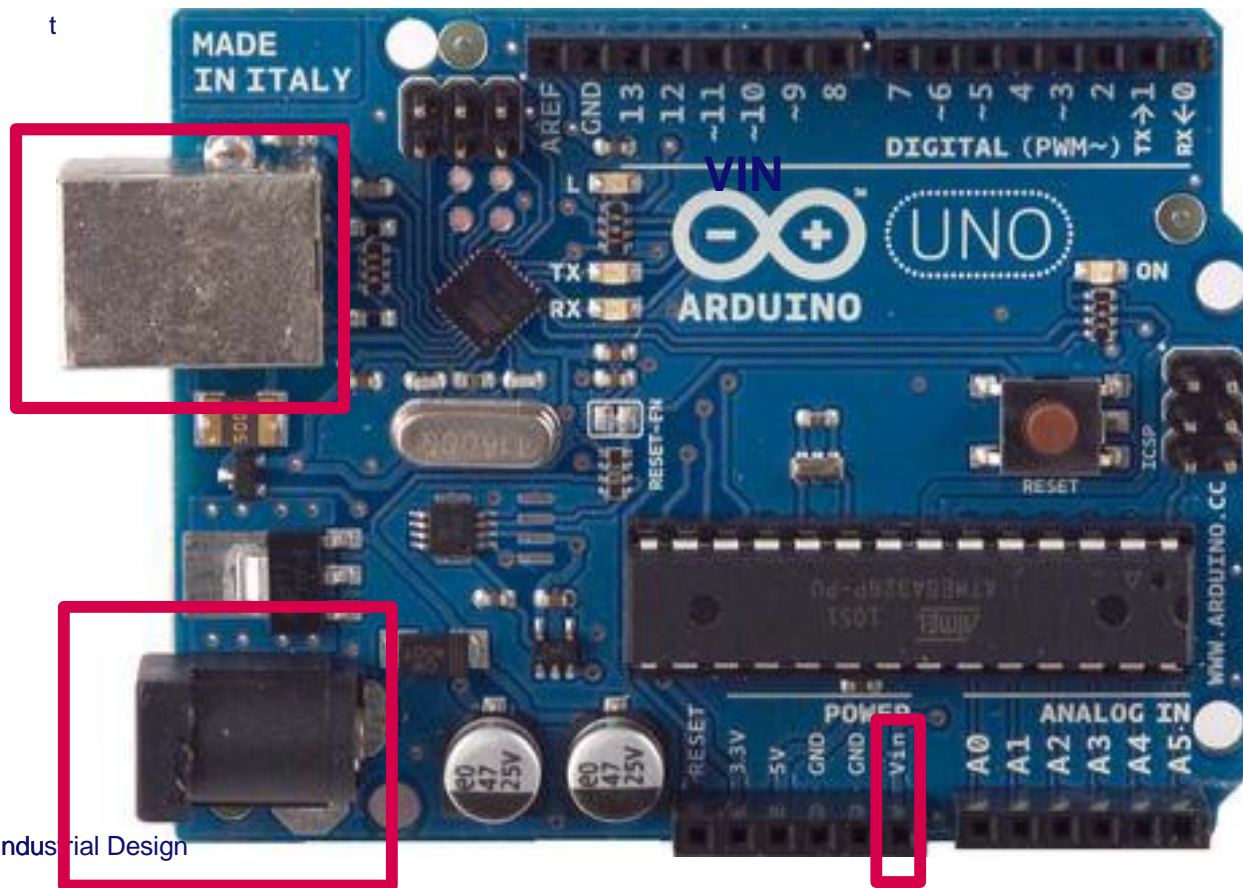


Arduino UNO



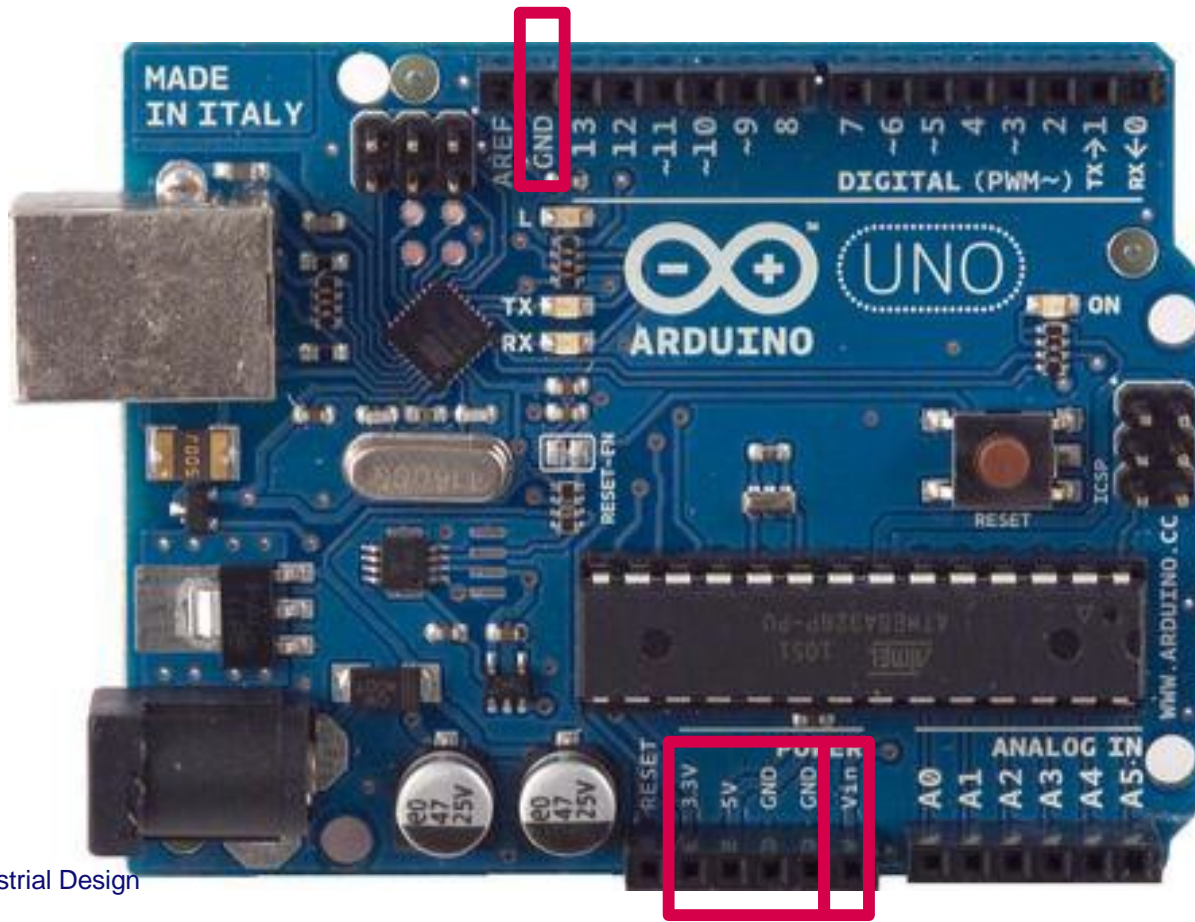
UNO

- USB Power supply (5V)
- External power supply (7V-12V)



UNO

- Power pins: Vin, GND, 5V, 3.3V



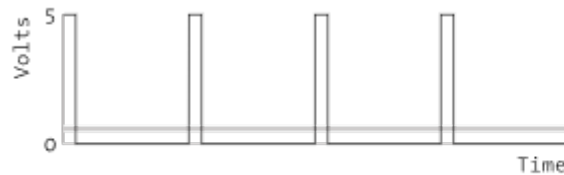
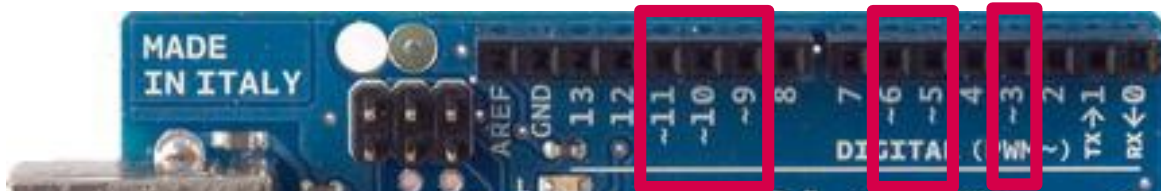
UNO

- 14 digital I/O pins (0 or 5V), Pin 13: LED

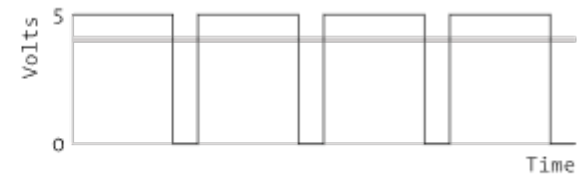


UNO

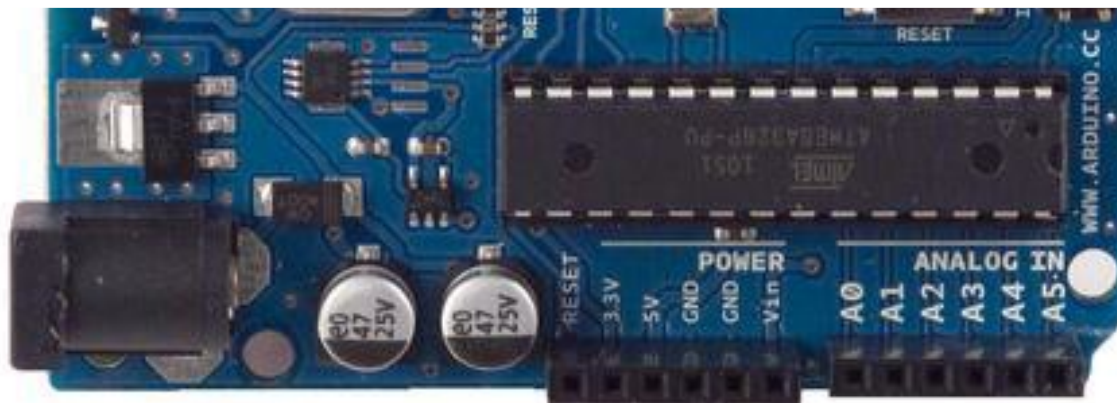
- 6 digital I/O pins provide PWM (~) output



10% duty cycle emulates 0.5 volts



80% duty cycle emulates 4 volts



UNO

- 2 digital I/O pins provide serial communication

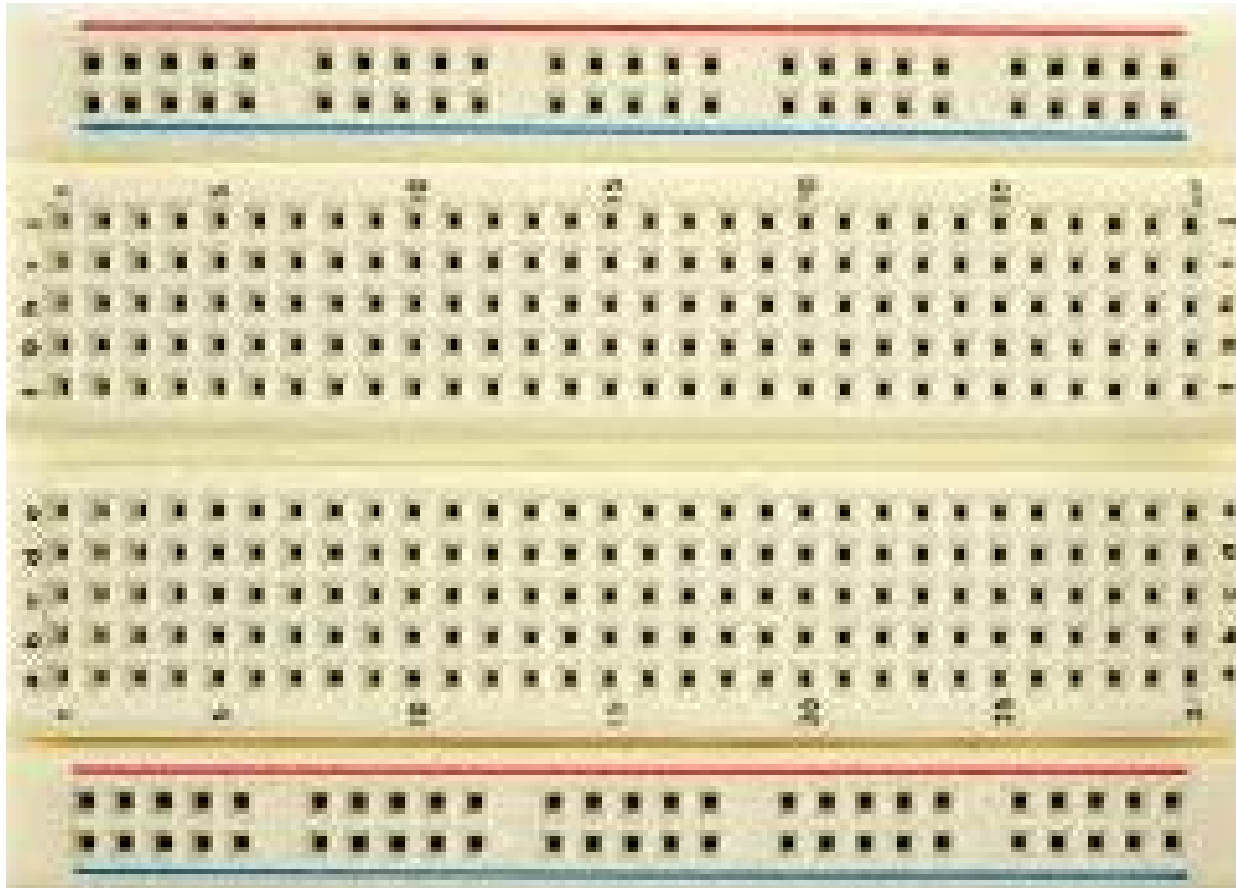


UNO

- 6 analog inputs, 0..5V => 0..1023

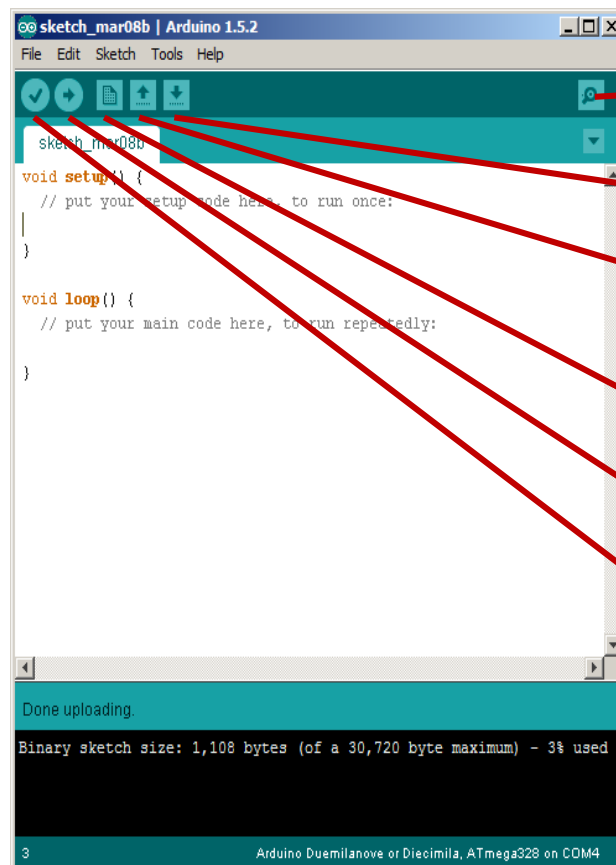


Breadboard



Software: IDE

- <http://arduino.cc/en/Main/Software>
- <https://www.arduino.cc/en/Guide/HomePage>



Serial Monitor

Save Sketch

Open Sketch
(* .ino)

New Sketch

Compile and Upload

Verify (Compile only)

Flashing TX/RX
LEDs



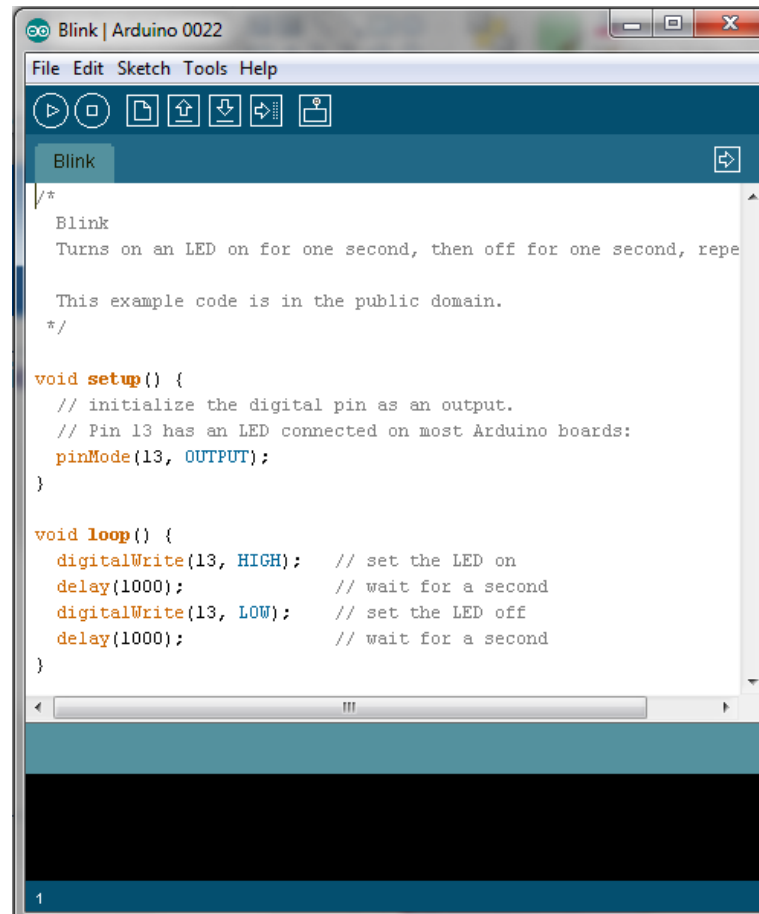
Designed
Intelligence
Group

TU/e

Technische Universiteit
Eindhoven
University of Technology

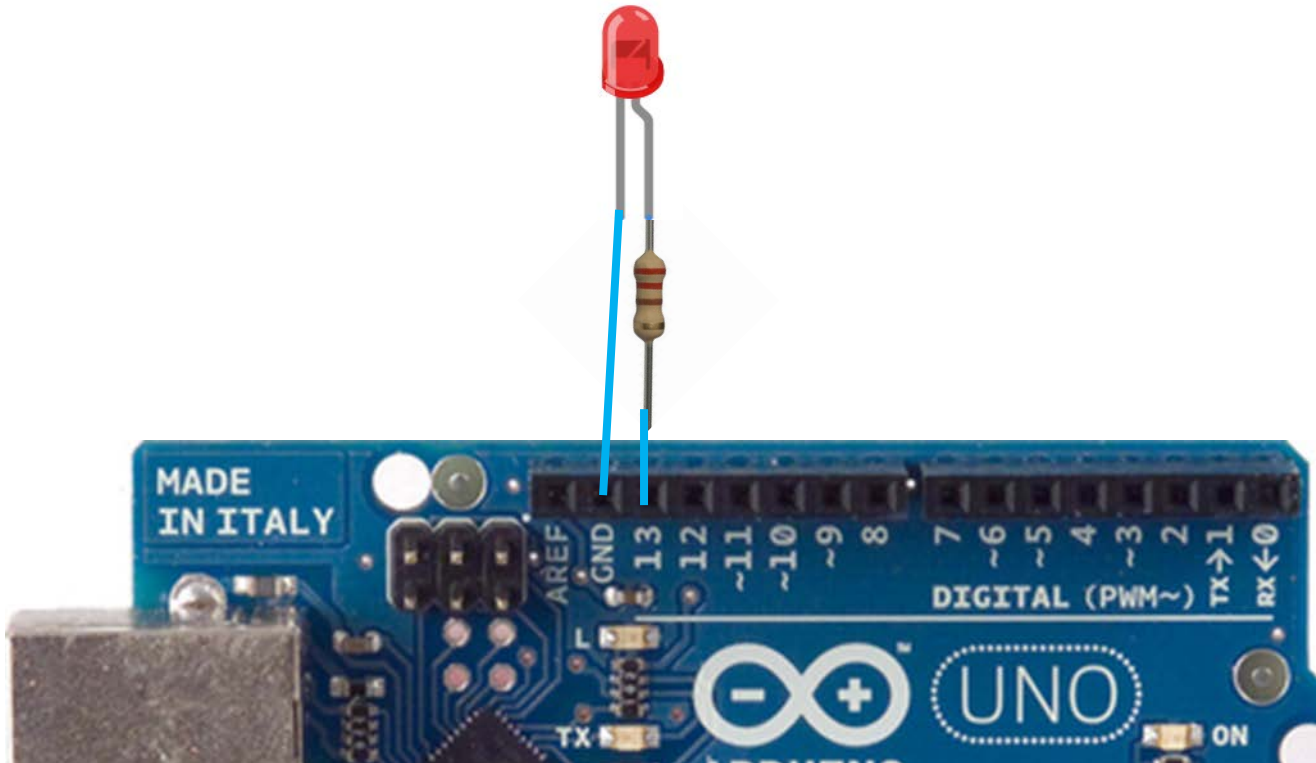
Hello World!

- **Arduino IDE: File>Examples>Basics>Blink**

A screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 0022". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for running, stopping, saving, and other functions. The main text area shows the code for the "Blink" example. The code is as follows:

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeats.  
  
  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  
}
```


Blinking a LED



Blinking a LED

- **#define LED 13**

```
#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);
}

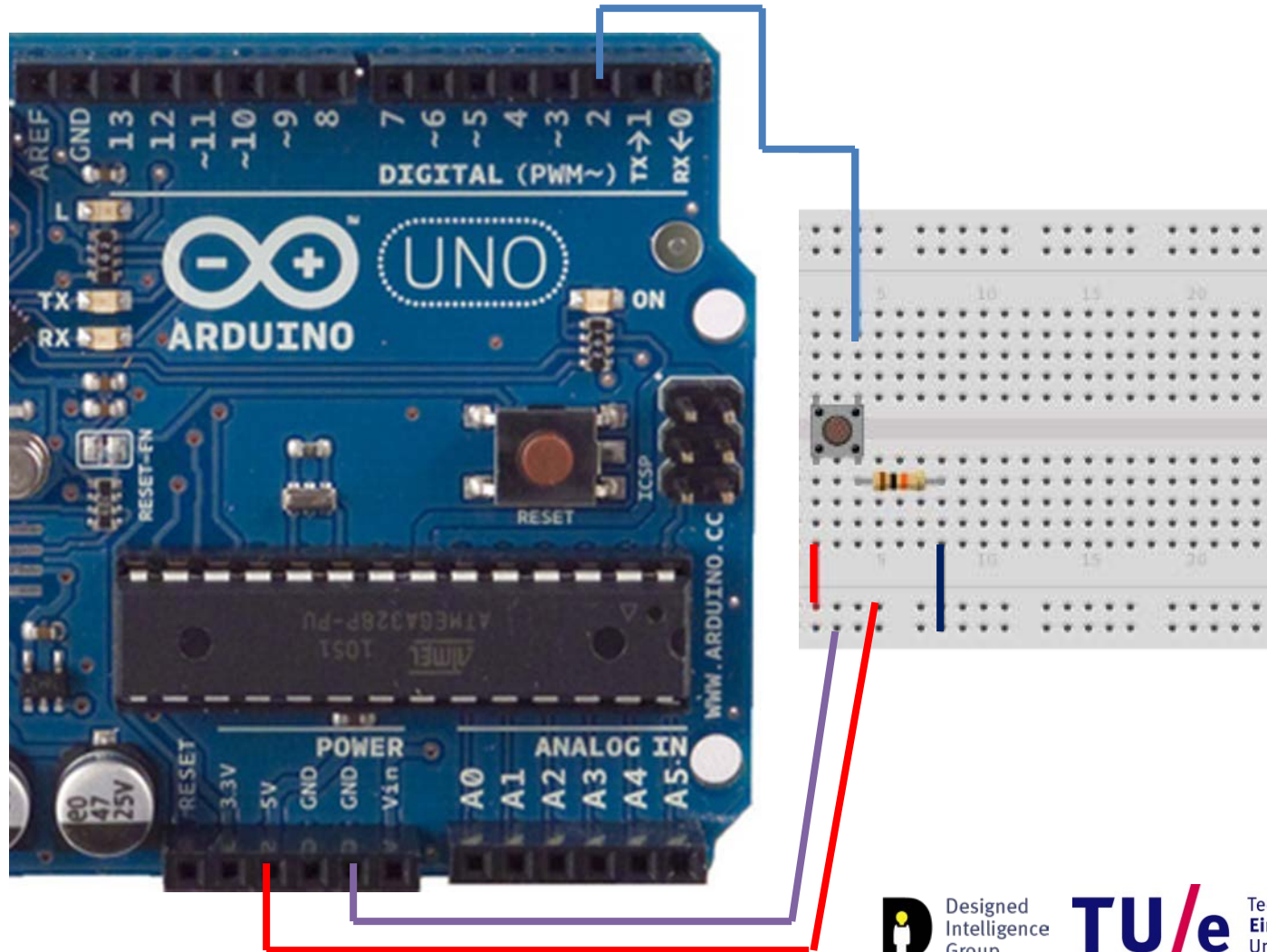
void loop() {
  digitalWrite(LED, HIGH); // set the LED on
  delay(1000);             // wait for a second
  digitalWrite(LED, LOW);  // set the LED off
  delay(1000);             // wait for a second
}
```

Button to control the LED

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

```
/*
 * Button
 *
 * Turns on and off a light emitting diode(LED) connected to digital
 * pin 13, when pressing a pushbutton attached to pin 2.
 *
 * The circuit:
 * * LED attached from pin 13 to ground
 * * pushbutton attached to pin 2 from +5V
 * * 10K resistor attached to pin 2 from ground
 *
 * Note: on most Arduinos there is already an LED on the board
 * attached to pin 13.
 *
 * created 2005
 * by DojoDave <http://www.0j0.org>
 * modified 28 Oct 2010
 * by Tom Igoe
 */
```

Button to control the LED



Button to control the 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);
}
```

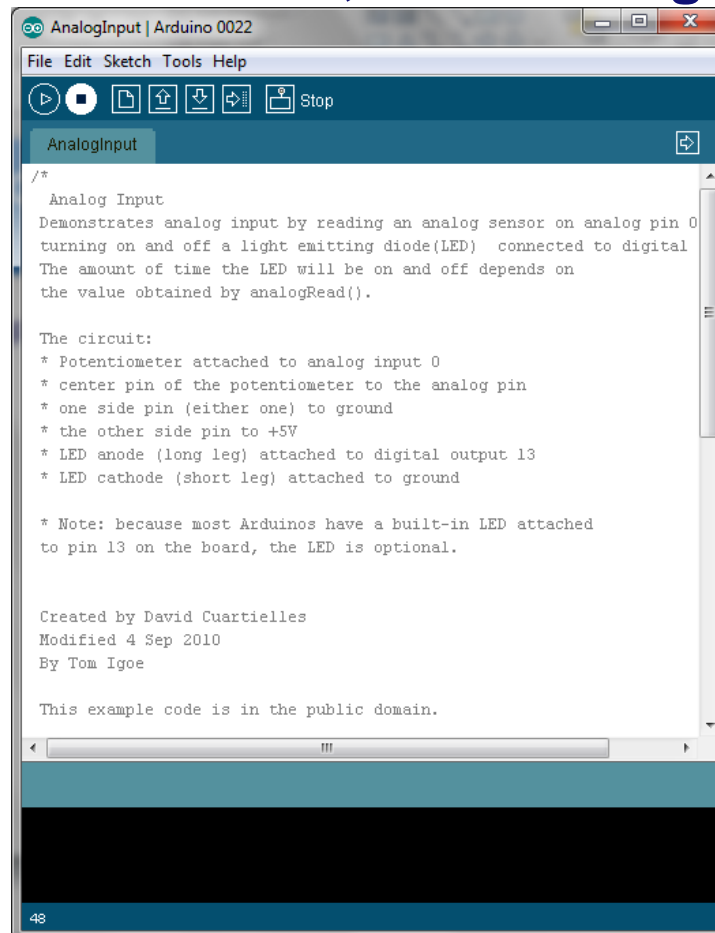
Button to control the LED

```
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 input

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



```
AnalogInput | Arduino 0022
File Edit Sketch Tools Help
[Run] [Stop] [Upload] [Download] [Refresh] [Send]
AnalogInput
/*
 Analog Input
 Demonstrates analog input by reading an analog sensor on analog pin 0
 turning on and off a light emitting diode(LED)  connected to digital
 The amount of time the LED will be on and off depends on
 the value obtained by analogRead().

 The circuit:
 * Potentiometer attached to analog input 0
 * center pin of the potentiometer to the analog pin
 * one side pin (either one) to ground
 * the other side pin to +5V
 * LED anode (long leg) attached to digital output 13
 * LED cathode (short leg) attached to ground

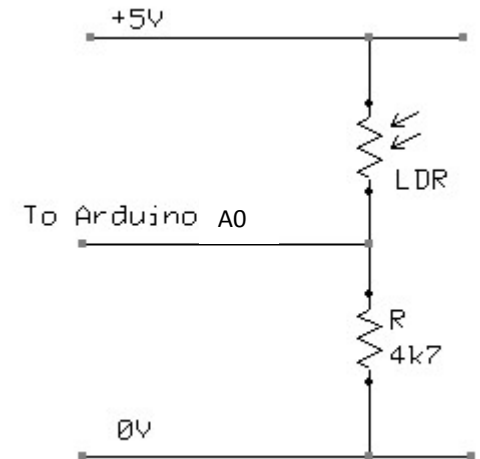
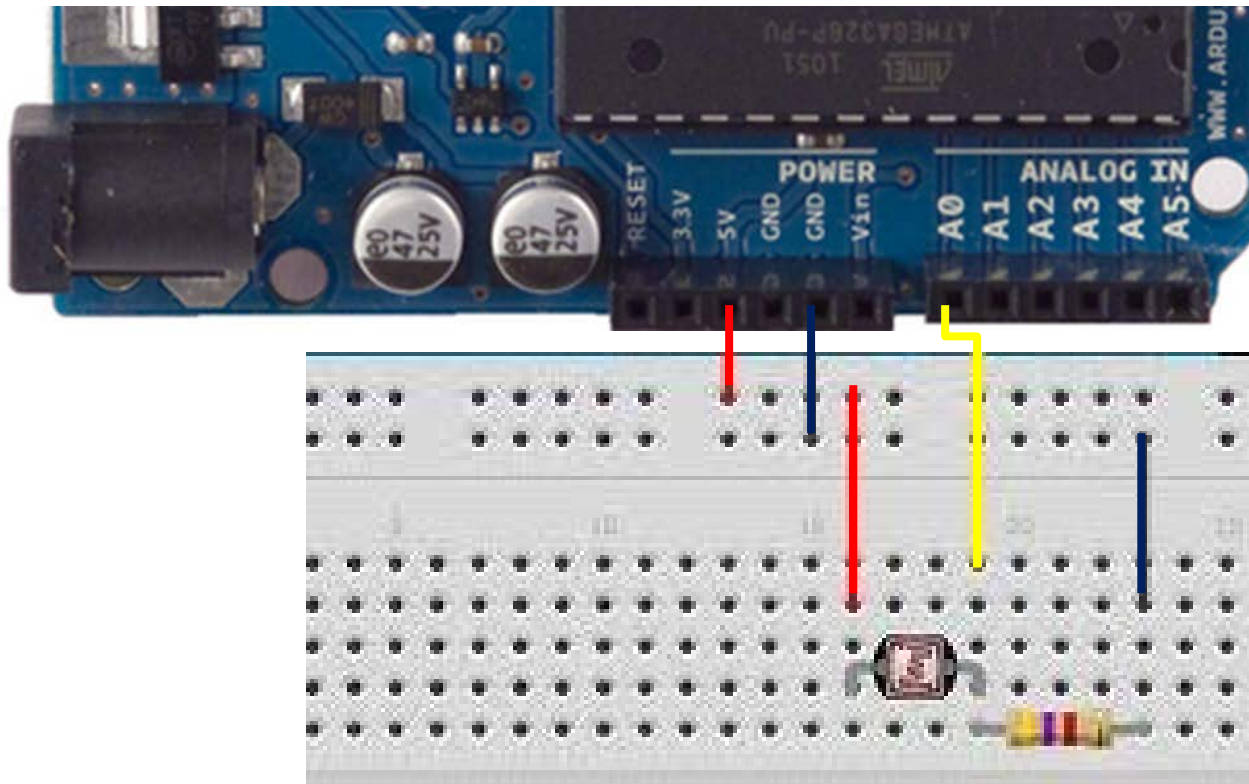
 * Note: because most Arduinos have a built-in LED attached
 to pin 13 on the board, the LED is optional.

 Created by David Cuartielles
 Modified 4 Sep 2010
 By Tom Igoe

 This example code is in the public domain.

```

Analog input



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);
}
```

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);
}
```

Additional Resources

You may look at:

- Processing Electronics tutorial

<https://processing.org/tutorials/electronics/>

And Many many more...

Examples

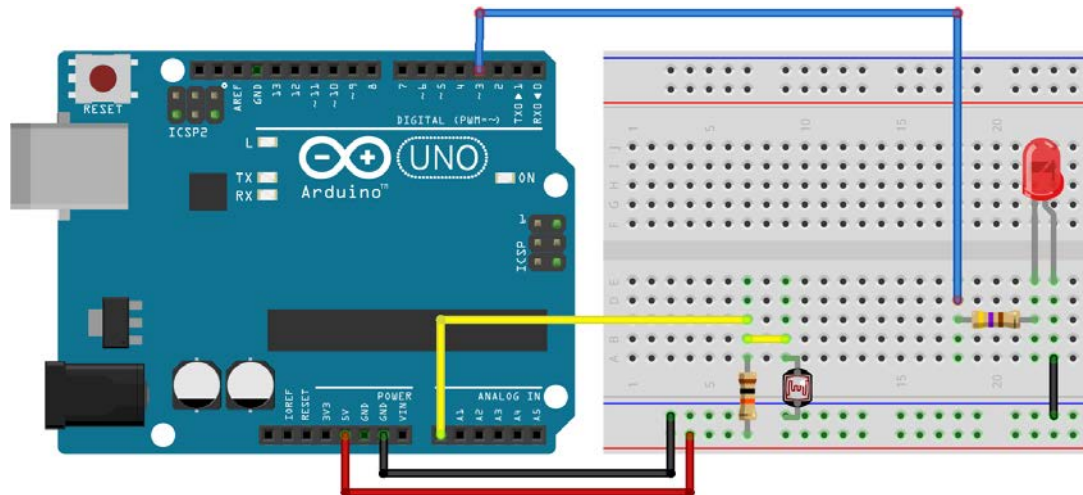
- Bad example
- Better example

Try this (homework-ish)

Connect a light sensor to the Arduino to detect when you wave your hand above it. Upon detection the Arduino activates a (LED) for a certain amount of time.

More complex:

It has to work in different lighting conditions...





That was a short introduction to Arduino

TU/e Technische Universiteit
Eindhoven
University of Technology