Embedded computing

UDOO – Single-board and microcontroller

Master ATIAM - Informatique

Philippe Esling (esling@ircam.fr)

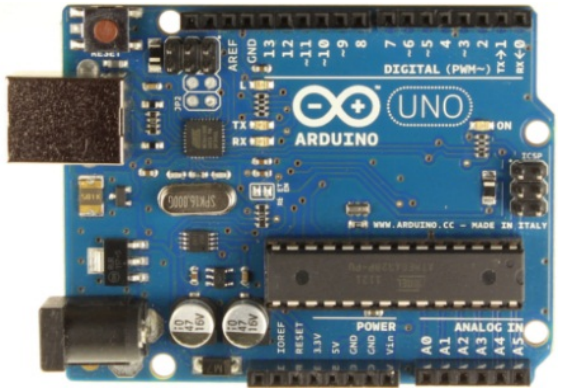
Maître de conférences – UPMC

Equipe représentations musicales (IRCAM, Paris)

UPMC SORBONNE UNIVERSITÉS master ATIAM ircam Centre Pompidou

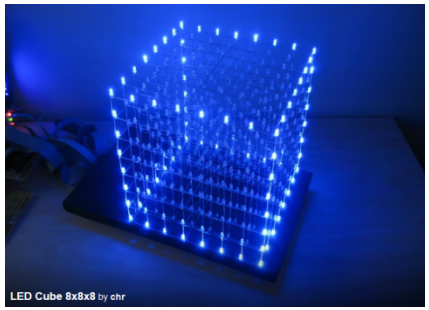
Single-Board Computer vs. Microcontroller

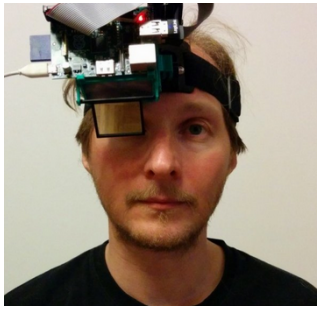
Raspberry Pi
Single-Board Computer

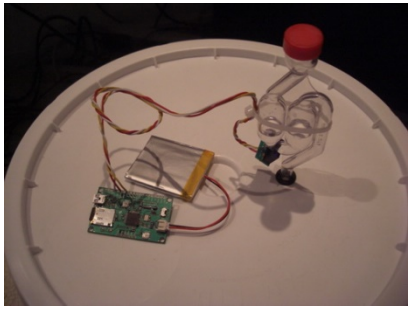


Arduino UNO
Microcontroller (MCU)

What can you do with this ?







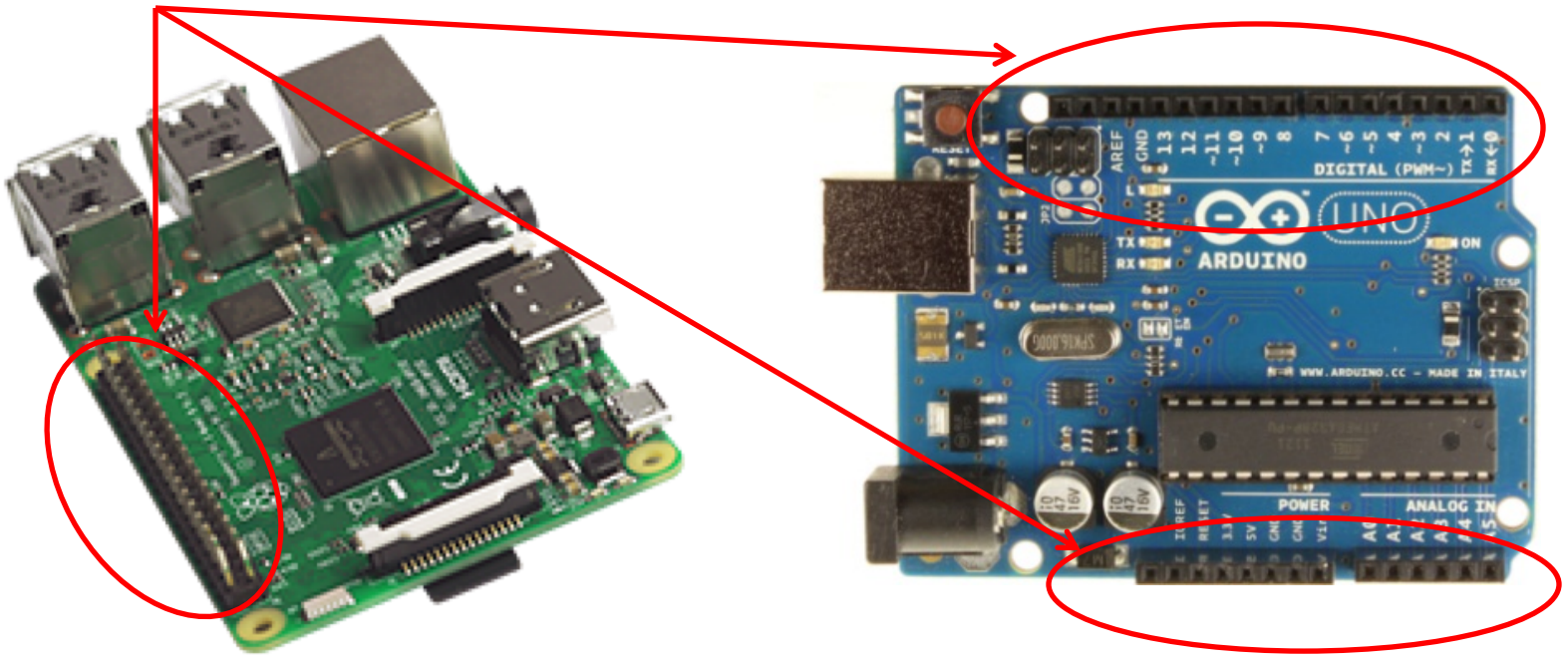
Jasper – Voice Controller powered by Rpi https://jasperproject.github.io
LED Cube - http://www.instructables.com/id/Led-Cube-8x8x8/
Flamethrower Jack-O-Lantern - http://www.instructables.com/id/Flamethrowing-Jack-O-Lantern/
Arduino RC- Lawnmower http://www.instructables.com/id/Arduino-RC-Lawnmower/
Fermentation Monitoring - https://www.sparkfun.com/tutorials/131
Raspberry Eye - https://hackaday.io/project/865-raspberry-eye

Why so popular?

- They are cheap

GPIO = General Purpose Input/Output

- They sense and manipulate the physical world
  - GPIO pins (reading values/sending instructions)

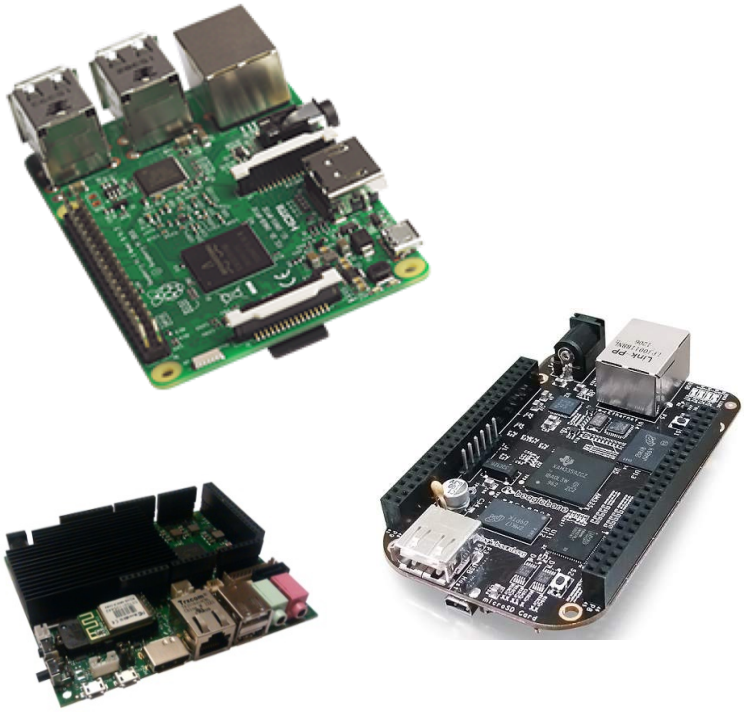


Internet of Things (IoT)

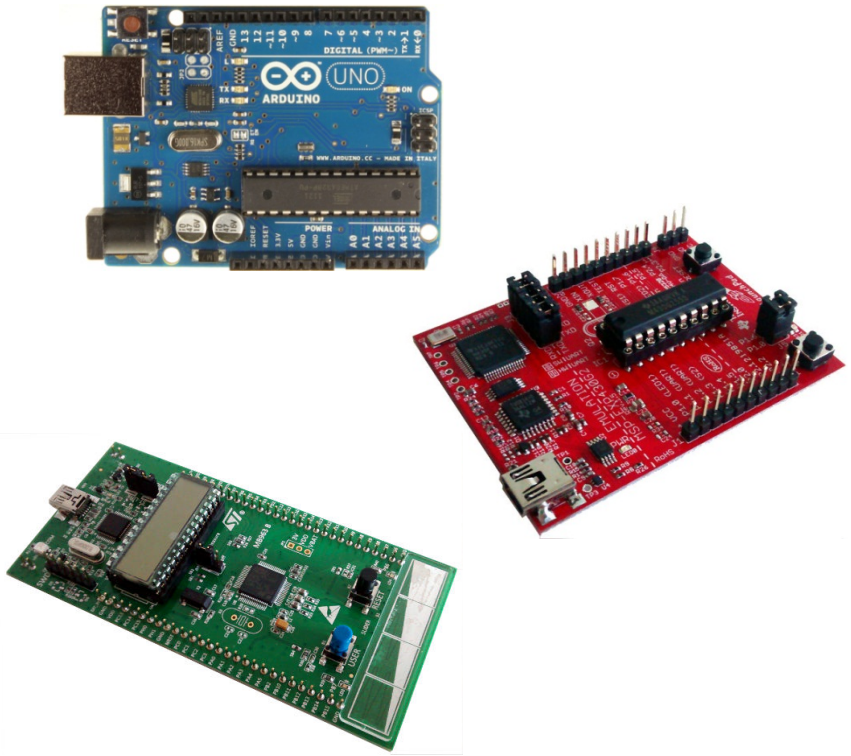
- Connection of embedded computing devices
  - Home Automation
  - Augmented Reality
- Connecting non-computer devices
  - Home Entertainment
  - Aquariums
  - … your fridge ?!
  - Etc.

Single-Board Computer vs. Microcontroller

What is the difference?



Single-board Computers



Microcontrollers

Single-Board Computer vs. Microcontroller

Defining Features

- Single-board computer
  - Lots of processing power
  - Operating system (Linux, Android)
  - Monitor Connection
  - Cheap, “weak” computer

- Microcontroller
  - Devoted to having GPIO pins
  - Low power, quick boot
  - Very cheap

Raspberry Pi

Arduino

Micro-controller ?

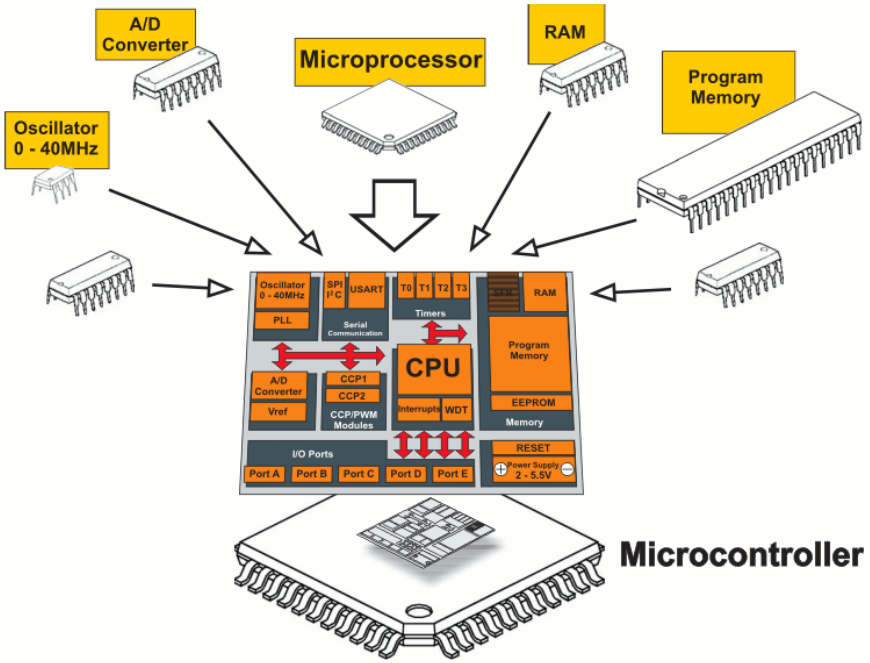


Fig. 0-1 Microcontroller versus Microprocessor

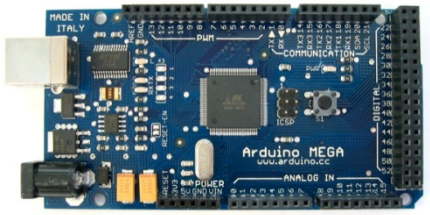
A single chip has to emulate a small computer

- Processor, memory, and input/output

Microcontrollers

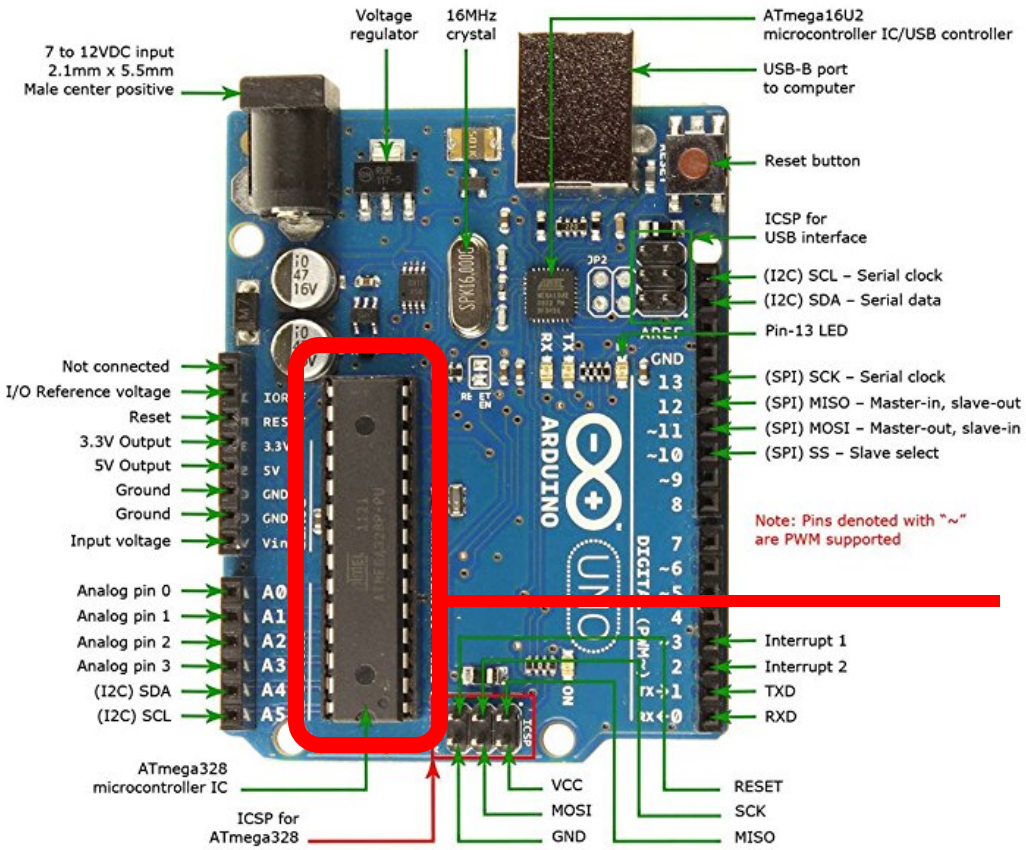
Arduino

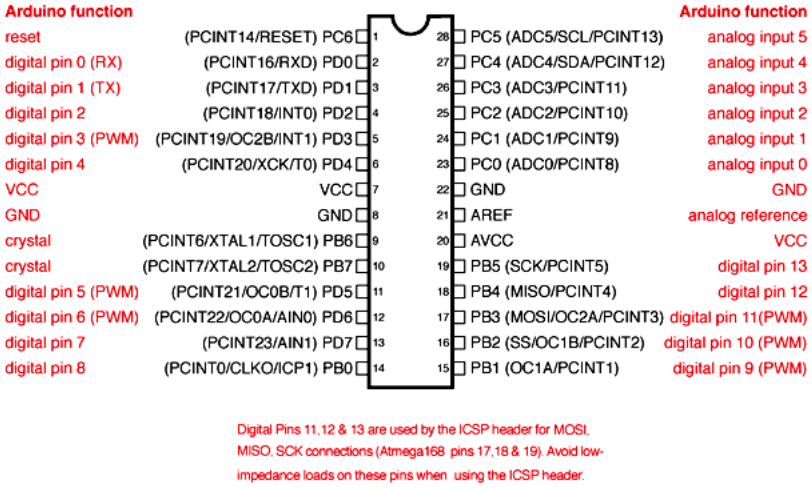
	Arduino Uno	Arduino Mega2560	Arduino Nano (ATmega328)
Microcontroller	ATmega328	ATmega2560	ATmega328
Flash Memory	32KB	256KB	32KB
Clock Speed	16MHz	16MHz	16 MHz
RAM	2KB	8KB	1KB
Price (approx, USD)	$24.95here	$45.95 here	$34.99 here
Digital I/O Pins	14	54	14



http://arduino.cc/

Arduino Microcontroller





GPIO Pins !

How do we use this ?

- Working with these devices requires
  1. **Programming**
  2. **Electronics**

First Challenge

Programming – Simplifying Aspects

- Pre-written Libraries
- Code Examples
  - https://github.com/
- Community help (Forums, StackExchange)
- Free online courses
  - Search for “MIT Arduino Course”
  - Programming courses on:
    - Khan Academy
    - Coursera
    - Udacity

Second Challenge

Electronics – the hard part

- Difficult to get started
- Easy to make mistakes
  - Some mistakes cannot be undone
- Often needs additional equipment:
  - Soldering Iron
  - Voltmeter
  - Oscilloscope

Electronics

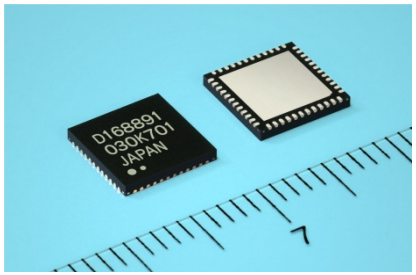
Simplifying Aspects

- Starter Packs and Kits
  - Contain a minimum number of electronic components that you may need
- Microcontroller Attachments
  - Shields (for Arduino)
    - Motor shields, sensor shields
  - Capes (for Beagleboard)
  - BoosterPacks (TI LaunchPad)

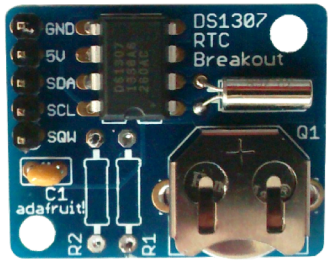
Electronics

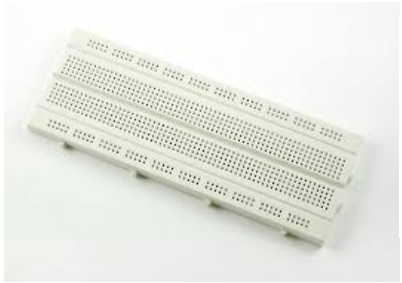
Simplifying Aspects

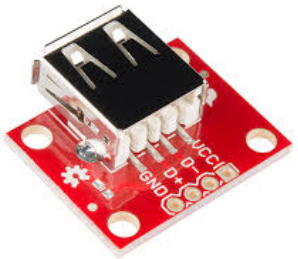
- A lot of electronics come like this:



- Solution: Breakout boards: pre-soldered breadboard-ready electronics:







Source:
http://elecshop.blog.com/files/2014/01/313-44-TQFP.jpg
http://www.renesas.com/media/press/news/2010/20101006.jpg

Microcontroller

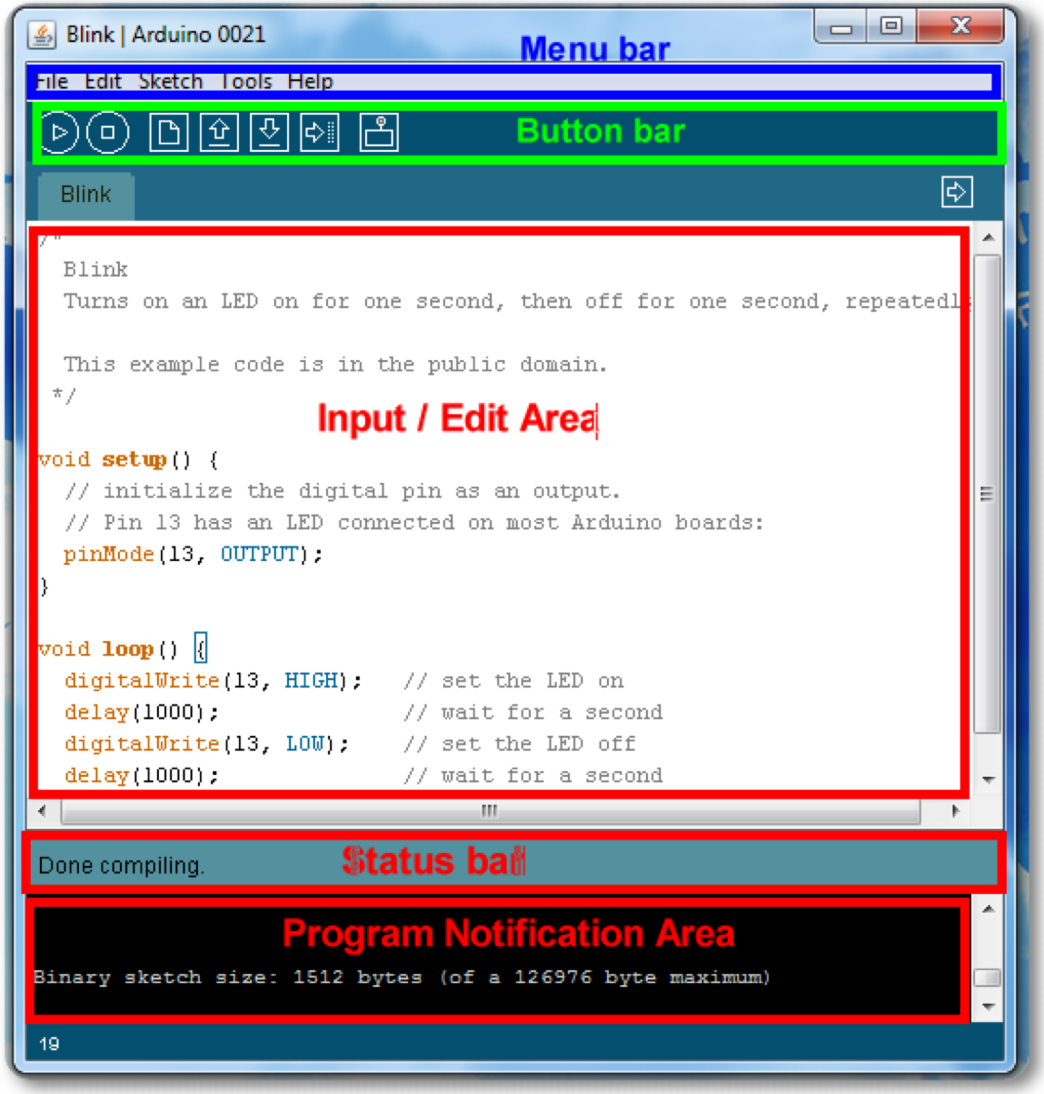
Expansion Boards features

- Communication Cable Attachments
- Programmer
- Buttons
- Power Supply Connectors
- Pin connectors
- On-board Sensors
  - (accelerometers, real-time clocks, temperature sensors)

How to use Arduino ?

1. Download & install the Arduino environment (IDE)
2. Connect the board to your computer via the USB cable
3. Launch the Arduino IDE
4. Select your board
5. Select your serial port
6. Open the blink example
7. Upload the program

Arduino IDE

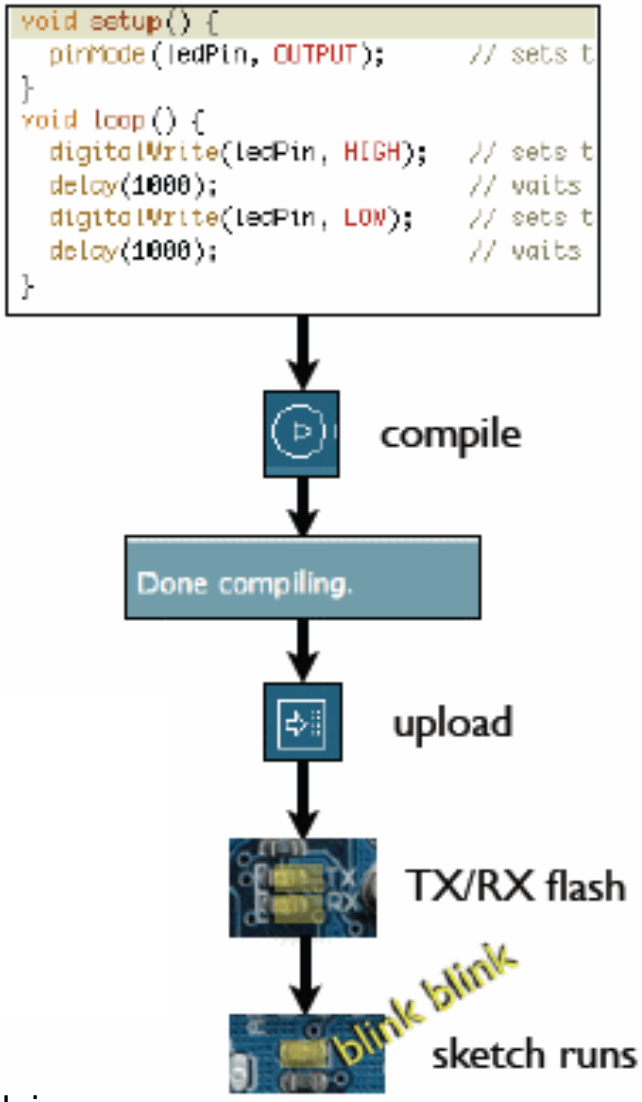

Blink | Arduino 0021
Menu bar
File Edit Sketch Tools Help
Button bar
Blink
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedl
This example code is in the public domain.
*/
Input / Edit Area
void setup() {
// initialize the digital pin as an output.
// Pin 13 has an LED connected 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
Done compiling.
Status bar
Program Notification Area
Binary sketch size: 1512 bytes (of a 126976 byte maximum)
19

Using Arduino

- Write your sketch
- Press Compile button (to check for errors)
- Press Upload button to program Arduino board with your sketch

???

Profit

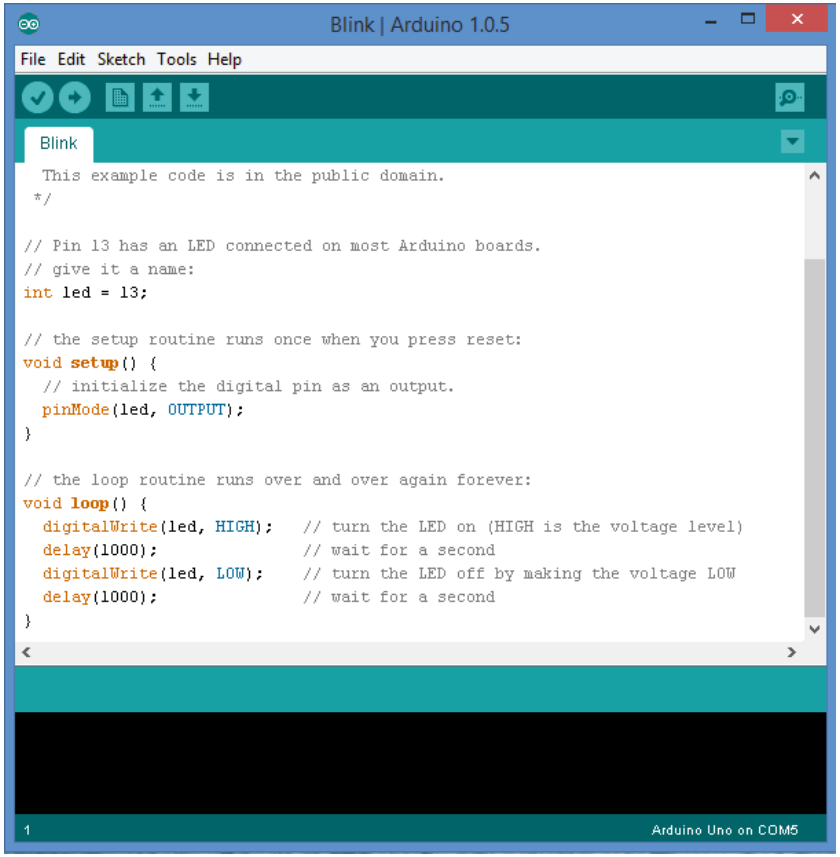


todbot.com/blog/bionicarduino

Microcontrollers

Arduino IDE - Programming

- http://arduino.cc/



Arduino terminology

"*sketch*" – a program you write to run on an Arduino board

"*pin*" – an input or output connected to something.

e.g. output to an LED, input from a knob.

"*digital*" – value is either HIGH or LOW.

(aka on/off, one/zero) e.g. switch state

"*analog*" – value ranges, usually from 0-255.

e.g. LED brightness, motor speed, etc.

Bare minimum code

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

void loop() {
    // put your main code here, to run infinitely:
}
```

Bare minimum code

- **setup** :
  - Called only when Arduino is powered on or reset
  - Used to initialize variables and pin modes

- **loop** :
  - Runs continuously till the device is powered off.
  - Main logic of the code goes here.
  - A form of while(True) for micro-controller

Digital I/0

`pinMode(`*`pin, mode`*`)`

Sets pin to either `INPUT` or `OUTPUT`

`digitalRead(`*`pin`*`)`

Reads `HIGH` or `LOW` from a pin

`digitalWrite(`*`pin, value`*`)`

Writes `HIGH` or `LOW` to a pin

PinMode

- A pin can be set as input or output

(All I/O pins can serve both mode)

pinMode function.

- pinMode(13, OUTPUT);
- // sets pin 13 as output pin

- pinMode(13, INPUT);
- // sets pin 13 as input pin

Reading/writing digital values

- **digitalWrite**(13, LOW);

// Makes the output voltage on pin 13 , 0V

- **digitalWrite**(13, HIGH);

// Makes the output voltage on pin 13 , 5V

- int **buttonState** = digitalRead(2);

// reads the value of pin 2 in buttonState

Analog to Digital (ADC) Conversion

Analog ?

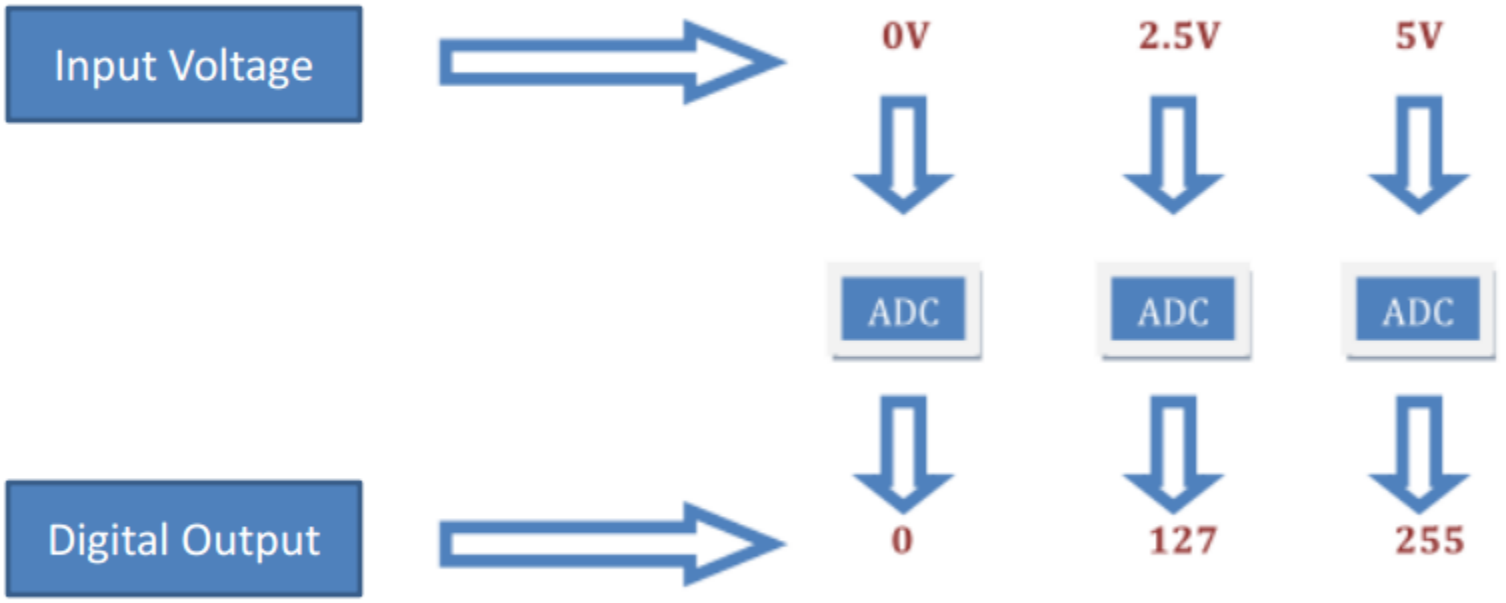
- Continuous range of voltage values
- Based on electrical in/outs

- Why convert to digital ?
- Because our microcontroller only understands digital.

ADC in Arduino

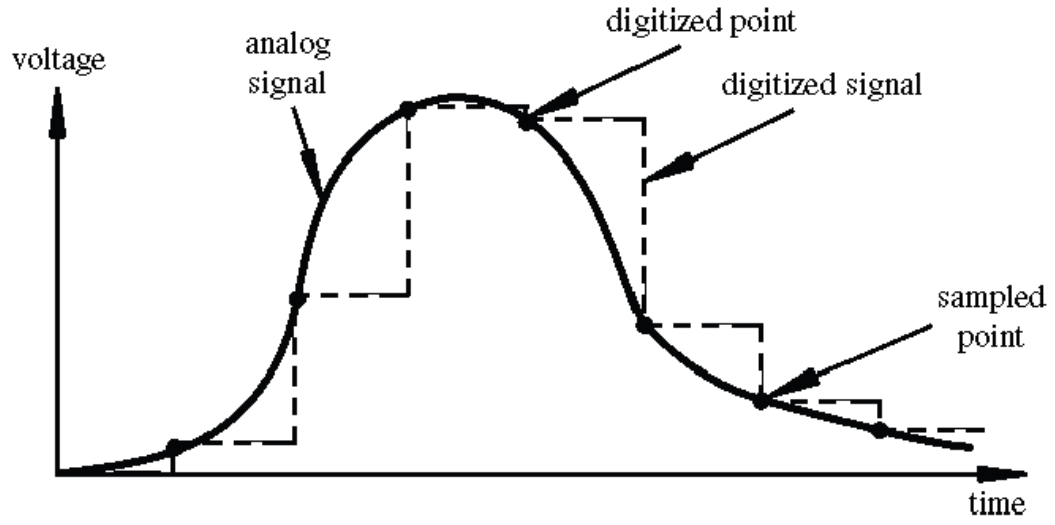
- The Arduino Uno board contains pins for ADC
- 10-bit analog to digital converter
- Map input voltages between 0 and 5 volts into integer values between 0 and 1023

Analog Value to Digital ?



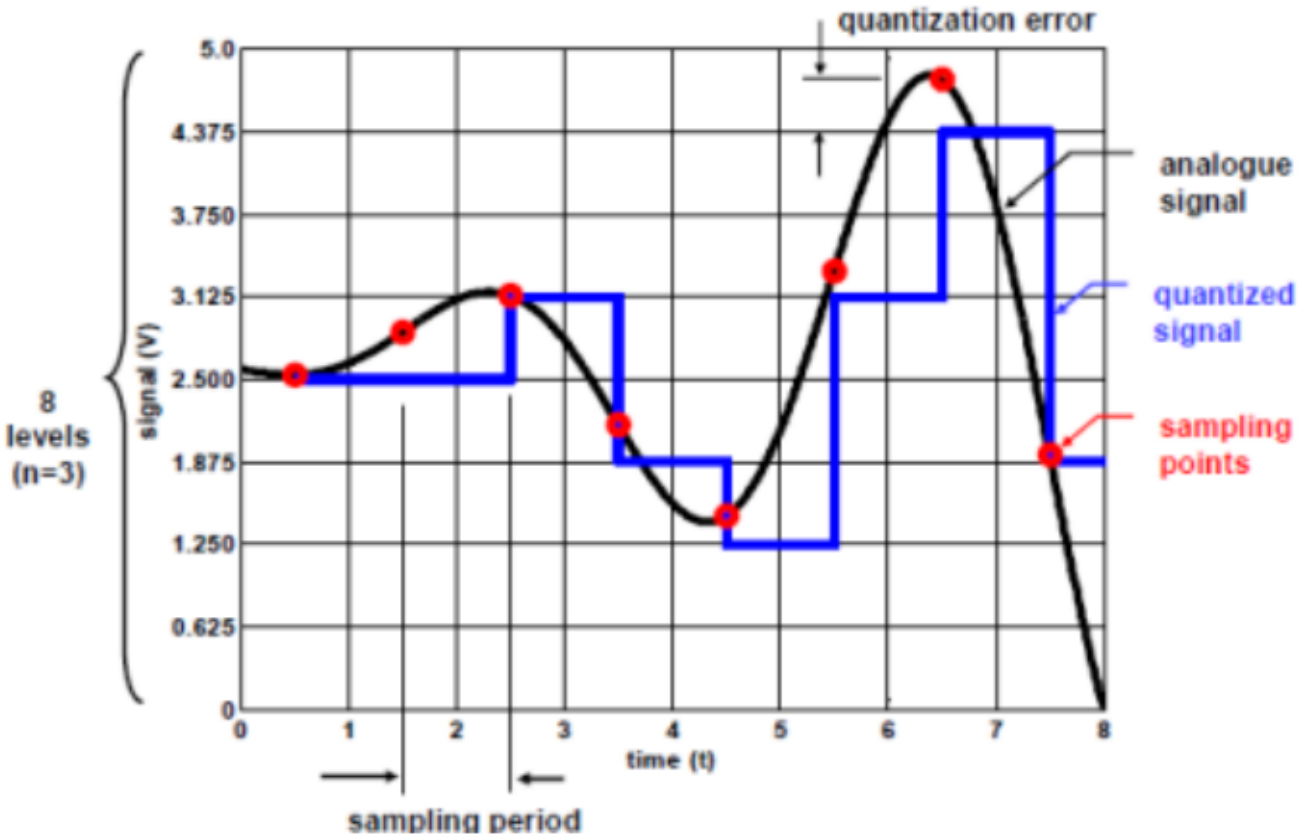
Digital? Analog?

- Digital has two values: **on** and **off**
- Analog has many (infinite) values
- Computers don't really do analog, they ***quantize***
- Remember the analog input pins---here's how they work



todbot.com/blog/bionicarduino

Signal quantization



Reading/Writing Analog Values

- analogRead(A0); // used to read the analog value from the pin A0
- analogWrite(2,128);

ADC Example

```
// These constants won't change.  They're used to give names to the pins used:
const int analogInPin = A0;  // Analog input pin that the potentiometer is attached to
const int analogOutPin = 9; // Analog output pin that the LED is attached to

int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}

void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  // change the analog out value:
  analogWrite(analogOutPin, outputValue);
  // print the results to the serial monitor:
  Serial.print("sensor = " );
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);
  // wait 2 milliseconds before the next loop for the analog-to-digital converter to settle
  delay(2);
}
```

Arduino Timing

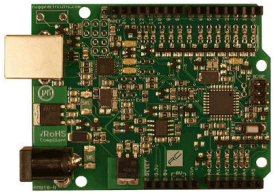
- `delay(`*ms*`)`
  - Pauses for a few milliseconds
- `delayMicroseconds(`*us*`)`
  - Pauses for a few microseconds
- More commands: arduino.cc/en/Reference/HomePage

Microcontrollers

Thousands Others

Some of these had limited production runs or have been discontinued.

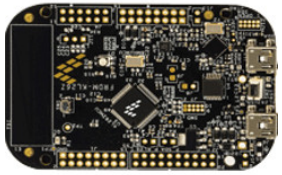
Beetle



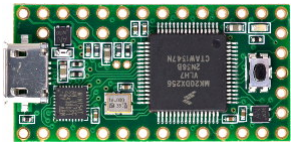
Ruggeduino

Pinguino PIC32

Gamebuino



Freescale Freedom

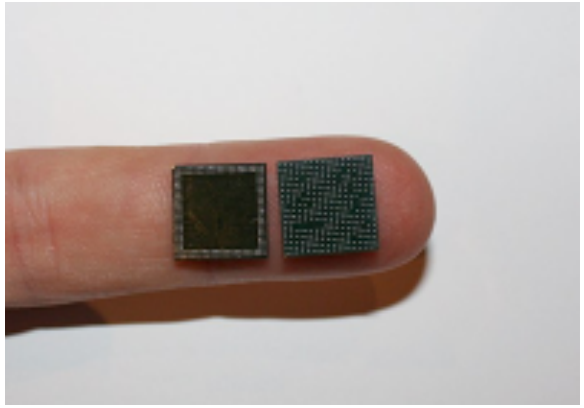


Teensy

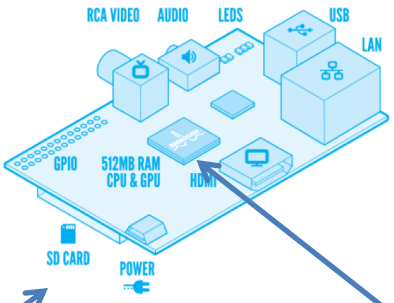
Beetle: http://www.dfrobot.com/wiki/index.php/Beetle_SKU:DFR0282
Nanode: http://www.nanode.eu/
Pinguino PIC32: https://www.olimex.com/Products/Duino/PIC32/PIC32-PINGUINO/open-source-hardware
Ruggeduino: http://www.ruggedcircuits.com/ruggeduino/
Gamebuino: http://gamebuino.com/
Freescale Freedom: http://www.freescale.com/webapp/sps/site/overview.jsp?code=FREDEVPLA
Teensy: https://www.pjrc.com/teensy/

Single-Board Computer vs. Microcontroller

Examples of Embedded Systems

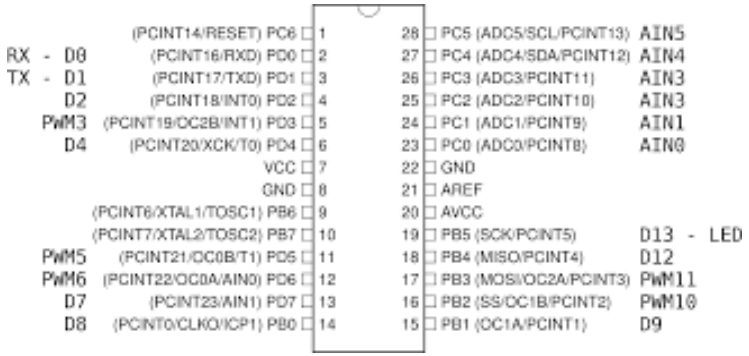


RASPBERRY PI MODEL B



Storage

Broadcom BCM2835
CPU, GPU, RAM



ATMega328
CPU, RAM, Storage

Image Source
http://pixshark.com/raspberry-pi-model-b-schematic.htm
http://images.studica.com/images/product/arduino/9a9b5c9e479e1a5b237b41a173d34dcfimage472x354_72_200x.gif

Single-Board Computers

Raspberry Pi

	Raspberry Pi Model B	Raspberry Pi 2
Processor	ARM1176JZF-S core	Quad-core ARM7
Flash Memory	None	None
Clock Speed	700 MHz	800 MHz
RAM	512MB	1 GB
Price (approx, USD)	$39.95	$35.00
Other Features	2x USB ports 26x GPIO pins	4x USB2.0 40x GPIO pins

http://www.raspberrypi.org/

Image Source: http://www.techradar.com/news/computing-components/peripherals/raspberry-pi-2-outed-with-faster-cpu-double-ram-and-windows-10-support-1283075

Raspberry Pi ?

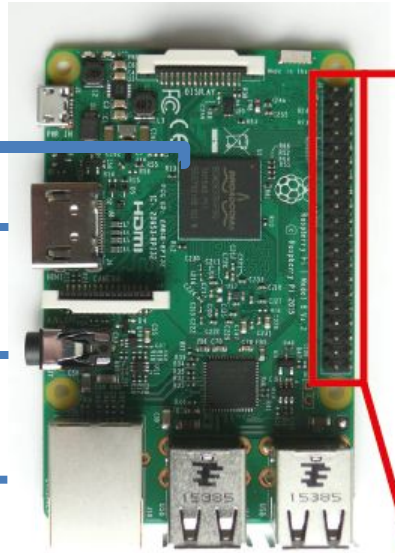
But also …

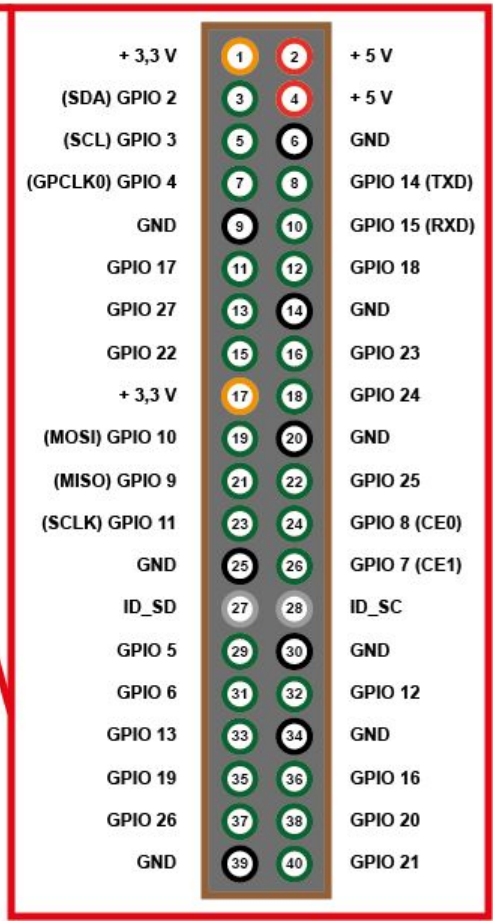
Powerful CPU

Screen display

Audio (sometimes)

Ethernet / WiFi





And most importantly

External storage

Operating system (**Linux**)

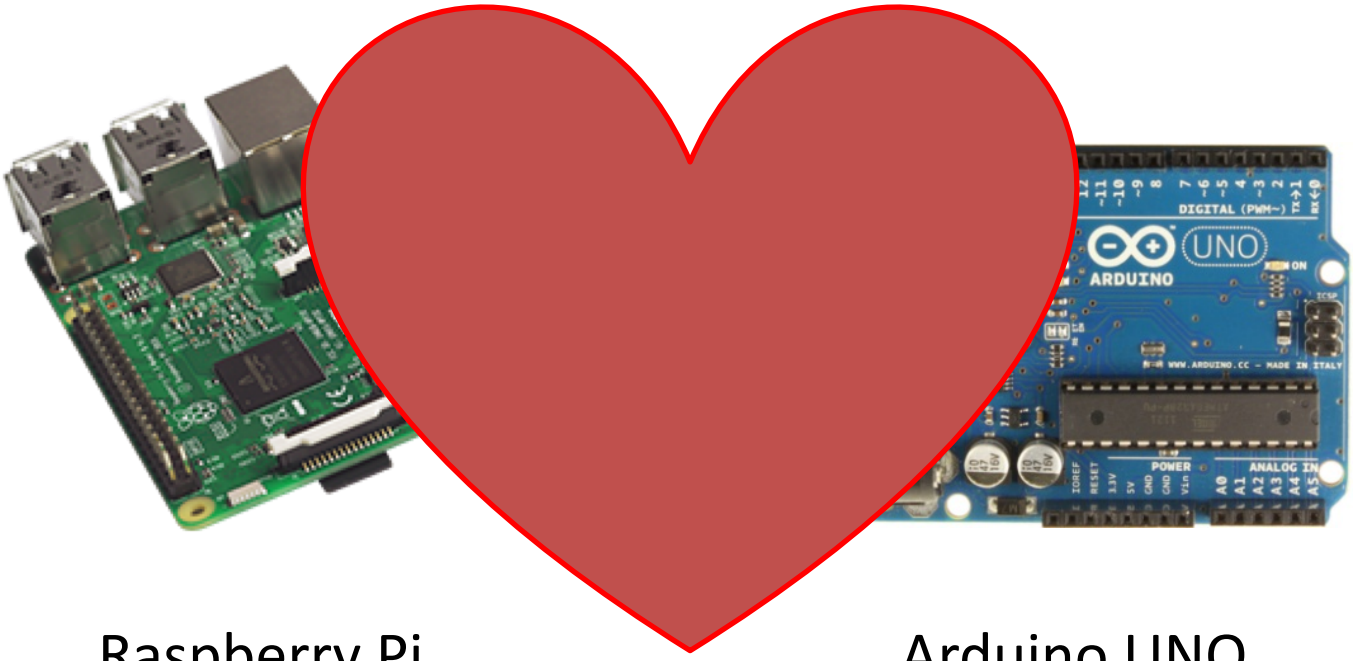
Computing / RAM power

Ease of use/debug

Single-Board Computers

- Properties
  - Processor Speed (Ghz, Multiple core)
  - RAM (speed of RAM)
  - Video Card (Home entertainment)
  - On-board storage (FLASH)
  - Hard-drive storage
  - GPIO pins
  - Features
    - Audio jack
    - IR remote
    - Blue tooth
    - Other connectors

Single-Board Computer vs. Microcontroller



Raspberry Pi
Single-Board Computer

Arduino UNO
Microcontroller (MCU)

Single-Board Computer vs. Microcontroller

UDOO

Single-Board Computer

+ Arduino Microcontroller (MCU) !

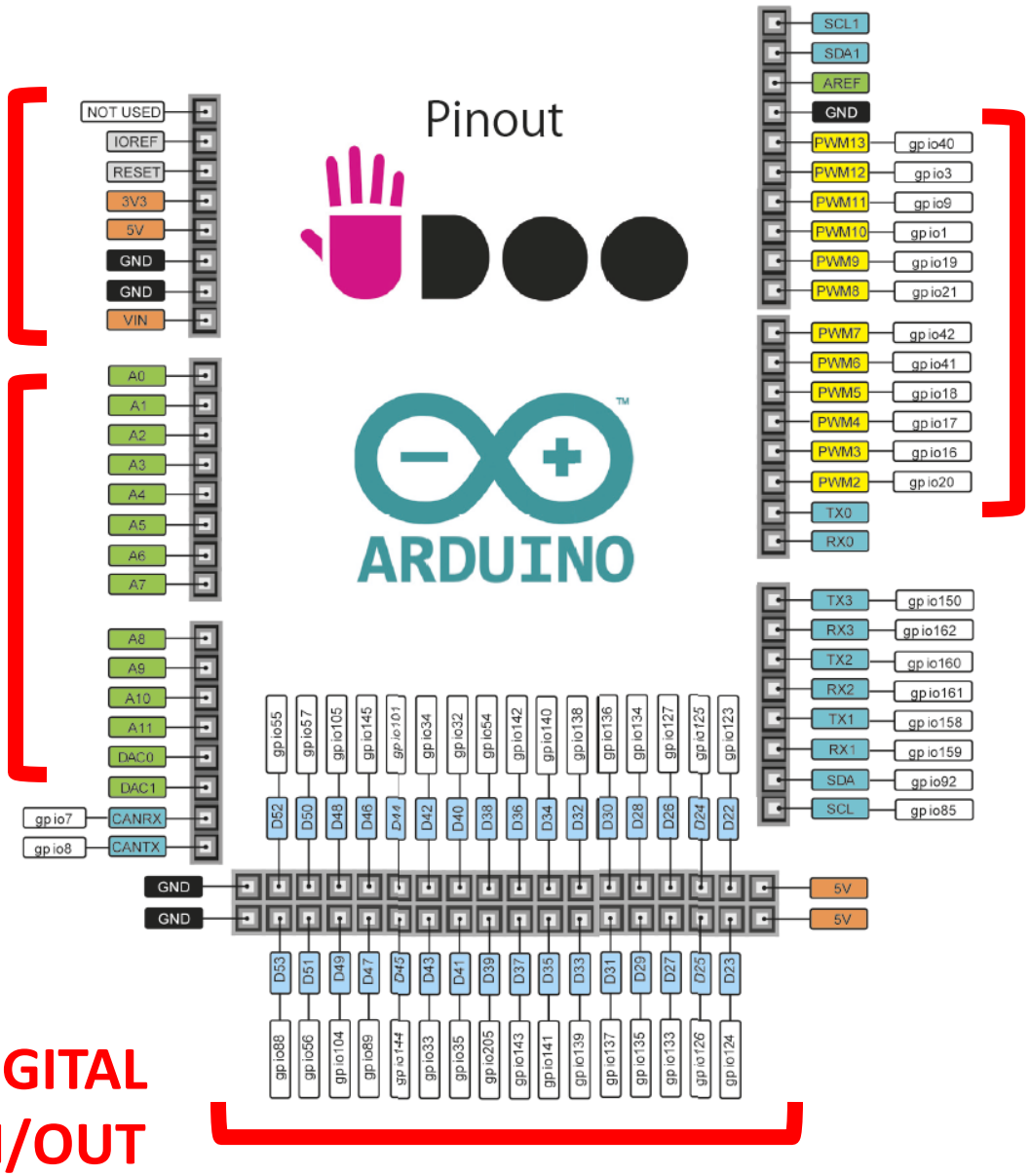
Single-Board Computers

UDOO

	UDOO Quad
Processor	Quad-core 1GHz
Flash Memory	None (SATA connector)
RAM	1 GB DDR3
Other Features	2x USB port WiFi Module 76 x GPIO pins Analog Audio and Mic
Price (approx, USD)	$135.00

http://www.udoo.org/

UDOO


POWER
ANALOG IN/OUT
DIGITAL IN/OUT
PWM
Pulse Width Modulation
Pinout
ARDUINO
NOT USED
IOREF
RESET
3V3
5V
GND
GND
VIN
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
DAC0
DAC1
CANRX
CANTX
gpio7
gpio8
SCL1
SDA1
AREF
GND
PWM13 gpio40
PWM12 gpio3
PWM11 gpio9
PWM10 gpio1
PWM9 gpio19
PWM8 gpio21
PWM7 gpio42
PWM6 gpio41
PWM5 gpio18
PWM4 gpio17
PWM3 gpio16
PWM2 gpio20
TX0
RX0
TX3 gpio150
RX3 gpio162
TX2 gpio160
RX2 gpio161
TX1 gpio158
RX1 gpio159
SDA gpio92
SCL gpio85
gpio55 gpio57 gpio105 gpio145 gpio101 gpio34 gpio32 gpio64 gpio142 gpio140 gpio138 gpio136 gpio134 gpio127 gpio125 gpio123
D52 D50 D48 D46 D44 D42 D40 D38 D36 D34 D32 D30 D28 D26 D24 D22
GND GND
5V 5V
D53 D51 D49 D47 D45 D43 D41 D39 D37 D35 D33 D31 D29 D27 D25 D23
gpio88 gpio56 gpio104 gpio89 gpio144 gpio33 gpio35 gpio205 gpio143 gpio141 gpio139 gpio137 gpio135 gpio133 gpio126 gpio124

UDOO - PureData

- PureData has a special distribution for UDOO
- You can find pre-compiled versions for it or compile it

https://www.udoo.org/forum/threads/guide-compile-pure-data-extended-on-udoo.478/

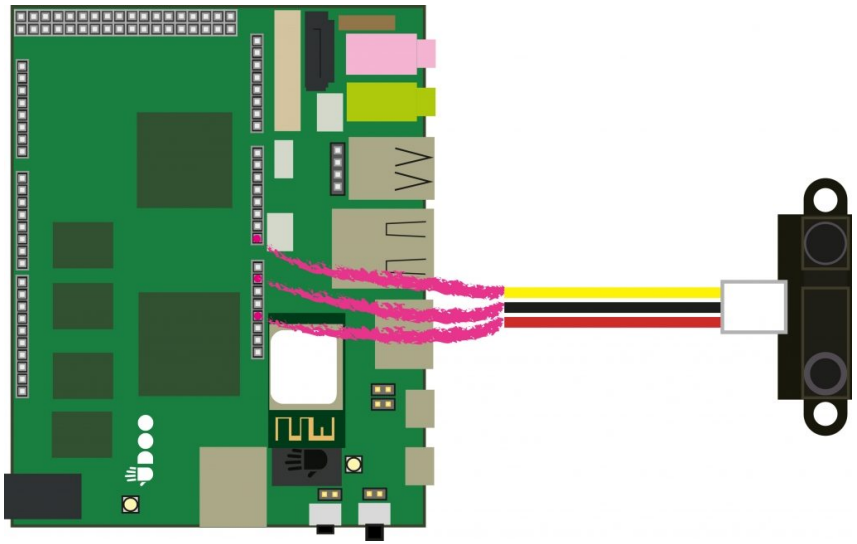
- This PureData version can use two objects
  - **[gpio]** allows a direct interaction with pins
  - **[comport]** is a serial data transfer extern (interact with Arduino code)

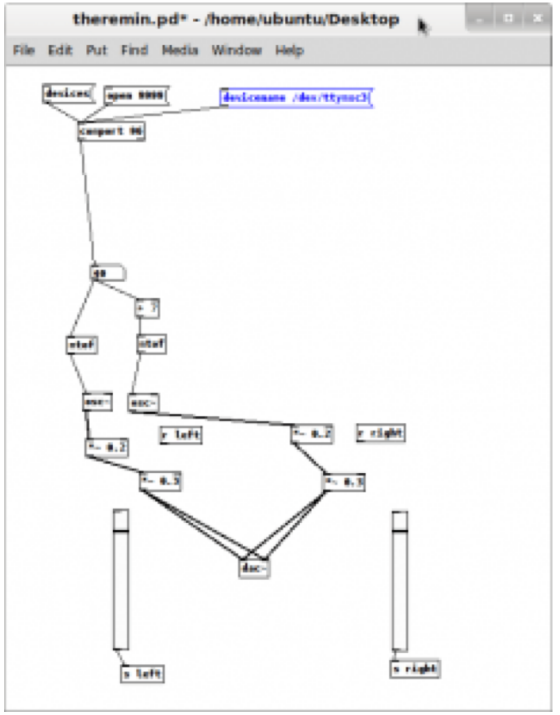
- Find more informations and tutorials at

  https://github.com/irllabs/Udoo

UDOO – PD

Project - Theremin of light





UDOO – PD

Project - Theremin of light

```
int IRpin = A0;
int distance = 0; int msg = 0;
void setup() { Serial.begin(115200); }
void loop()
{
distance = analogRead(IRpin);
if (distance < 100) { distance = 100; }
else if (distance > 900){ distance = 900; }
msg = map(distance, 100, 900, 0, 255);
Serial.write(msg);
}
```