Raspberry PI & Arduino Uno

Chapter Outline

- Raspberry PI
  - Specs
  - Setup & Configuration
  - Python
  - Hello World
- Arduino Uno
  - Specs
  - Programming
  - Hello World
- Comparison

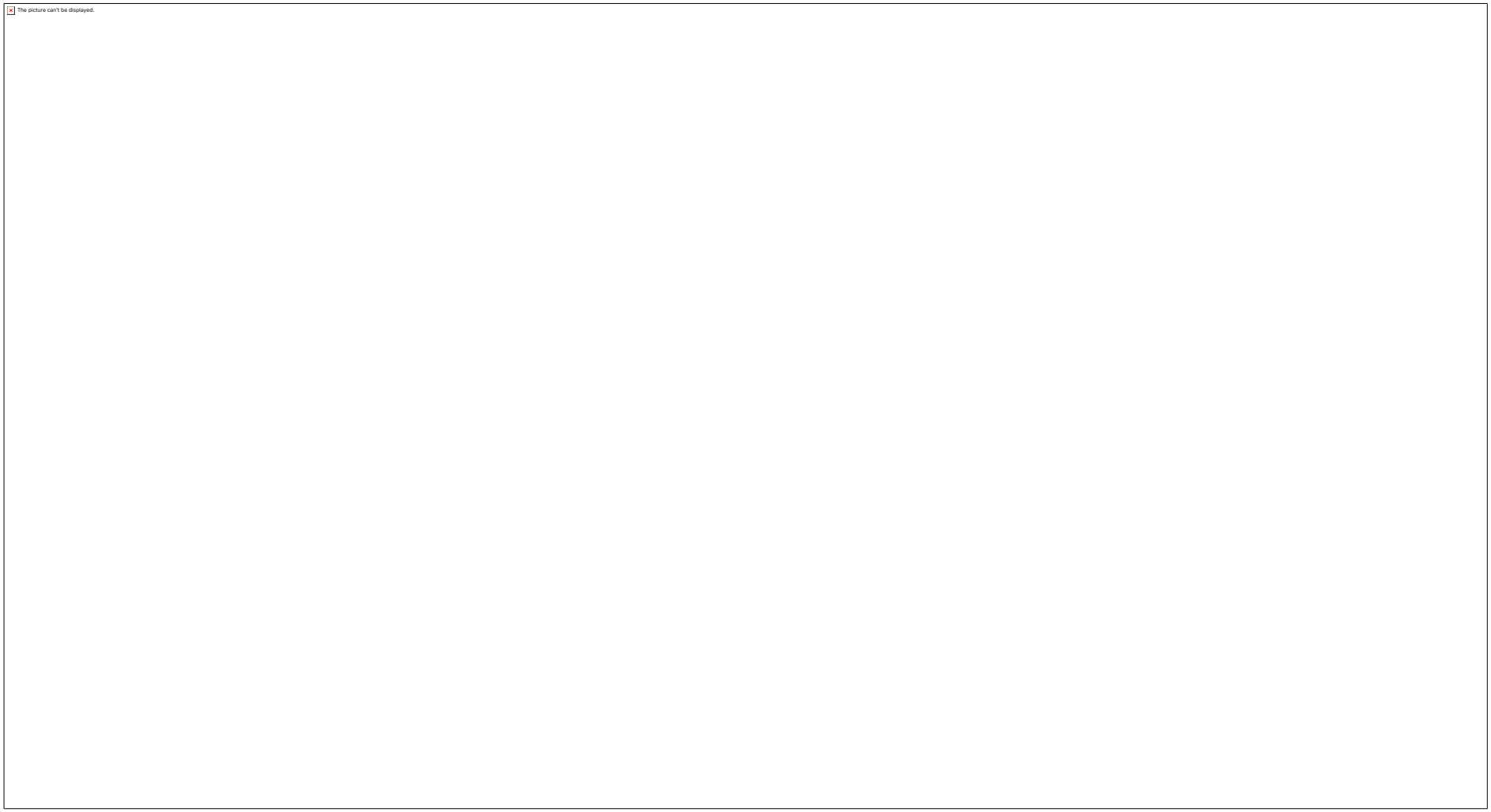
RASPBERRY PI

Raspberry PI 3 & the Rest of the World

	Raspberry PI, B	Beagle Bone Black	Odroid C1+	Banana Pi	Pine 64	pcDuino3	C.H.I.P
OS							
COST							

Src: www.pi-top.com

Raspberry PI is the Most Popular

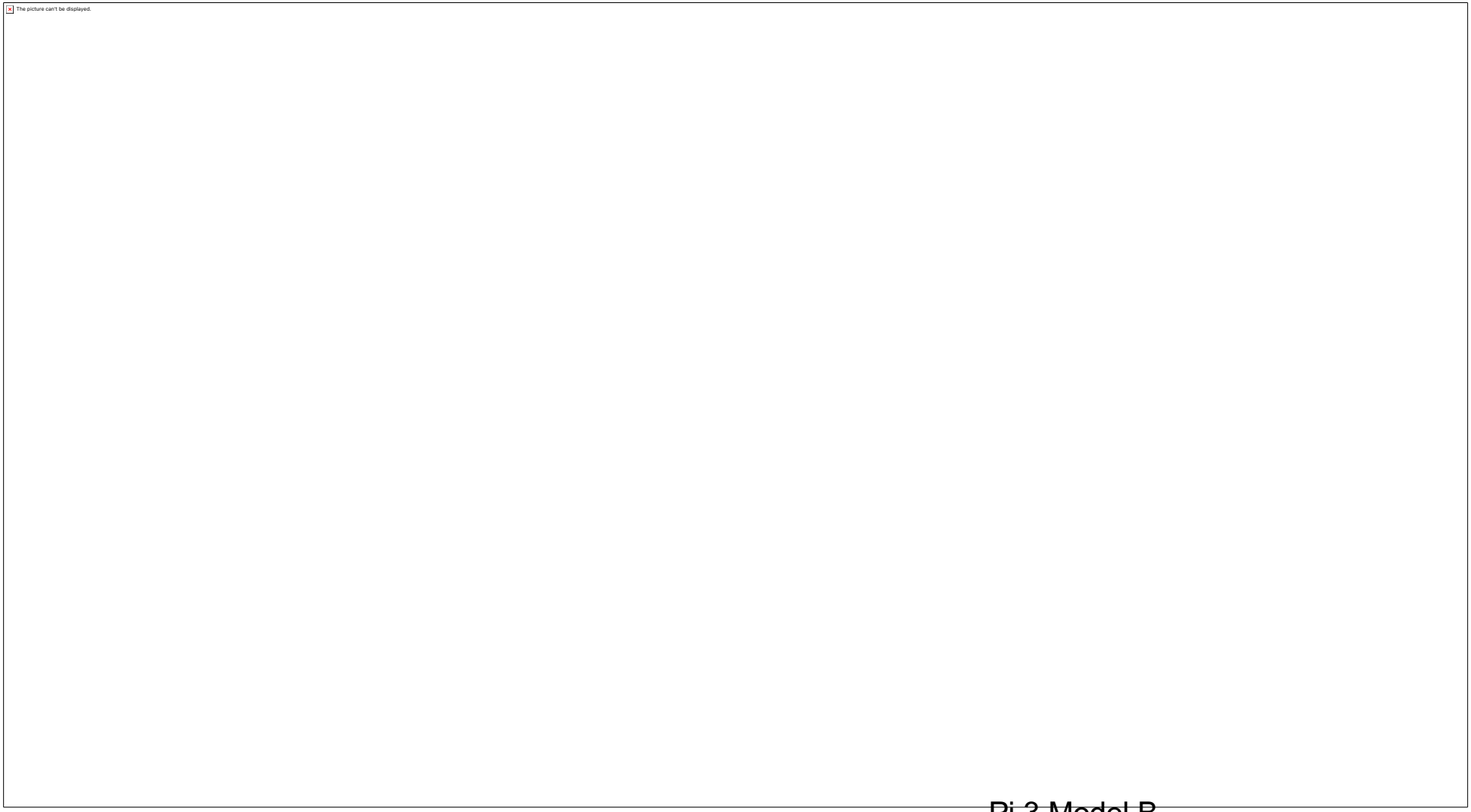


SRC: https://docs.google.com/spreadsheets/d/1zWwpcckDEEVAhNH3y7JQGxxbjP42nUywPOzDWr1fH28/edit#gid=0

Two Models

- Model A
  - Lower-spec variant of the Raspberry Pi (256 MB RAM, 1x USB port, no Ethernet)
  - Lighter and consumes less power
  - Suitable for embedded projects
    - Robotics
    - Projects where weight and low power are paramount
- Model B

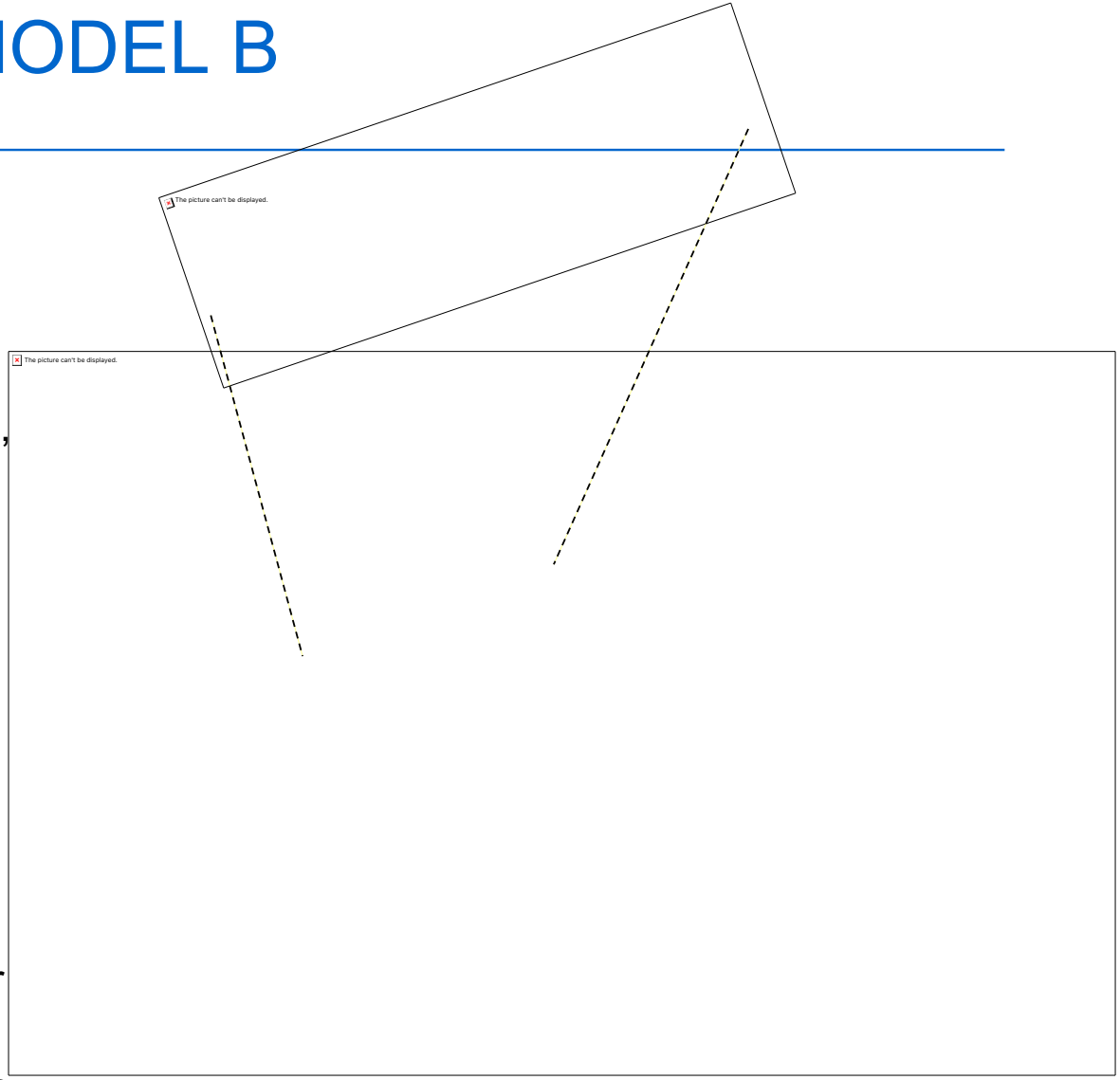
Model B Timeline



RASPBERRY PI 3 MODEL B

Specifications (Specs):

SoC (System on Chip): Broadcom BCM2837
CPU: 4xcore, ARM Cortex-A53, 1.2GHz, 64-bit
GPU: Broadcom VideoCore IV
RAM: 1GB LPDDR2 (900 MHz)
Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless
Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
Storage: microSD
GPIO (General-Purpose Input/Output)**:** 40-pin header (only 26 for Pi A)
Ports: HDMI, 3.5mm analog audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)



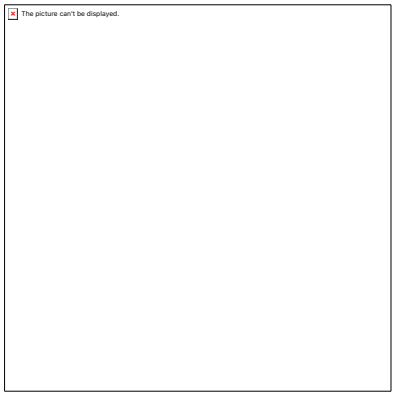
https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks

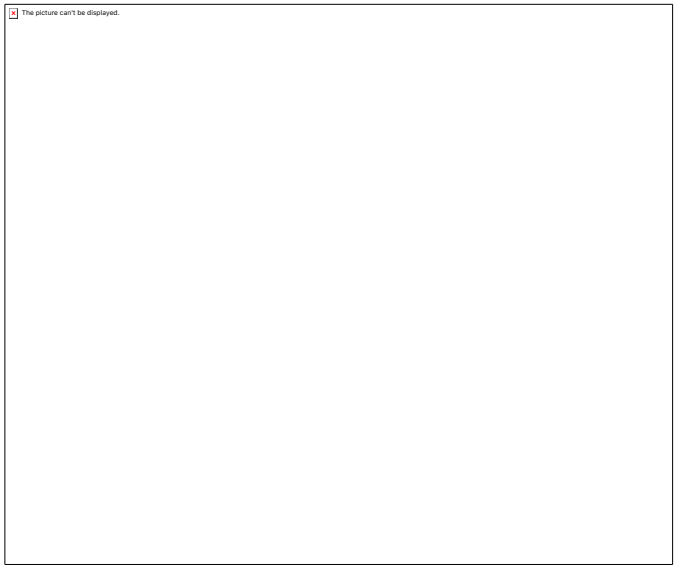
Setup of the Raspberry Pi

- Step 1: Setup an interface to the device
  - Plug in a monitor (via HDMI)
  - Keyboard/mouse via USB
- Step 2: Get an Operating System (OS)
  - Raspberry Pi needs an OS
  - OS image must be present on the micro SD card
- Step 3: Power supply
  - Micro USB power supply (at least 2A at 5V)

Installing an OS

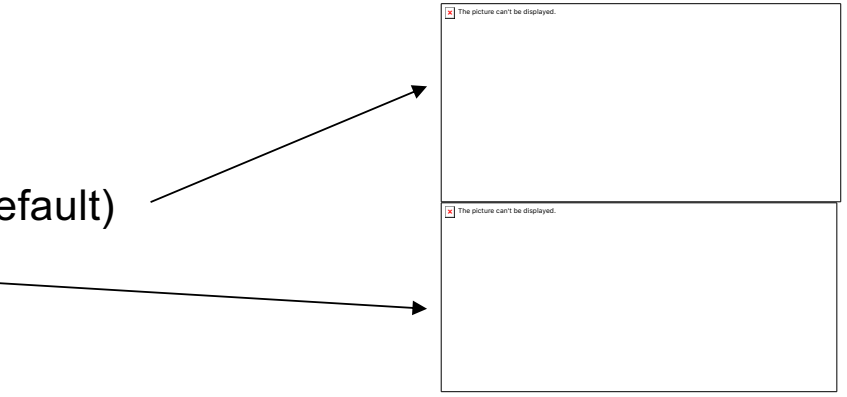
- Use NOOBS (New Out-Of-Box-Software)
  - Comes pre-installed on Micro SD bundled with Raspberry Pi boards
  - Otherwise use a “good quality” 8GB+ micro SD and do:
    - Format the micro SD (need an SD reader)
    - Download NOOBS for free from www.raspberrypi.org/downloads
    - Extract NOOBS download
    - Put it on the micro SD
- NOOBS will install an OS on the SD card
  - You get a choice of OS
    - Longer list if you are connected to Internet
  - Choose RASPBIAN (distribution of Linux/Debian)





Configuration of Raspberry Pi

- **Raspi-Config**
  - is a tool, which provides various setup/boot options for Raspberry Pi
  - will run automatically when you boot with a new SD card for the first time
- Raspi-Config key Options
  - Expand Filesystem: reformats your micro SD card filesystem to allow access to all the memory
  - Change User Password (highly important!)
    - Raspberry Pi starts with one default user account
      - Username: pi
      - Password: raspberry
  - Change Boot options
    - Console (text-based interface, default)
    - Desktop graphic interface

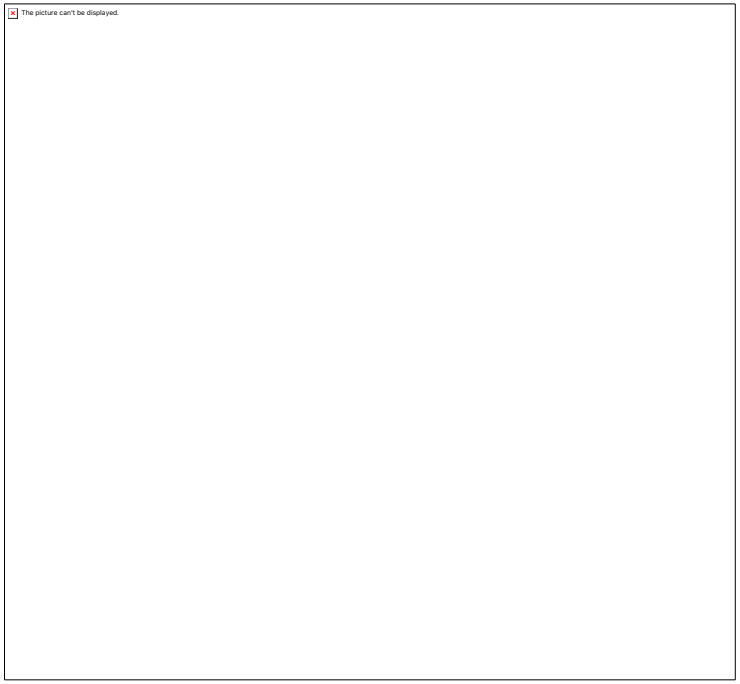


Programming Raspberry Pi

- Many programming languages can be used
  - Need a compiler (C, C++, Java, etc) or an interpreter (Python, Perl, etc)
  - Python is most convenient
    - Good programming environment built-in
    - Good APIs available to access Raspberry Pi hardware
- Python language
  - High-level language, easy to use
    - No need to explicitly declare data types
    - No pointers
    - Object-oriented programming
  - Slow compared to C/C++ (interpreted not compiled)
  - Two versions: 2.x and 3.x (3.x recommended)

Python Programming Environmment

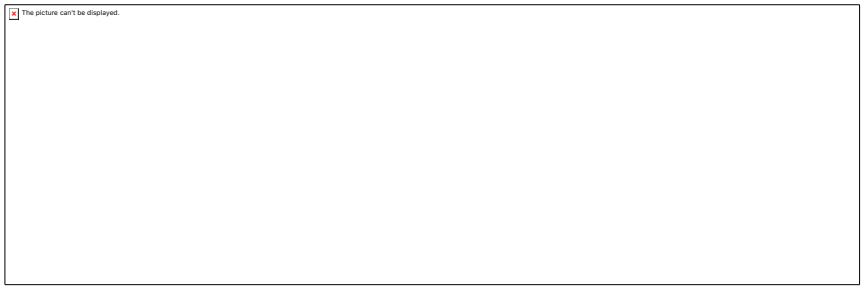
- Two possible environments
  - Integrated Development Environment (IDE)
    - IDLE is the best option
    - Invoke via ***Menu > Programming > Python***
    - Select Python 3
  - Text editor and interpreter separately
    - Use Raspberry Pi text editor (e.g., Pico or Nano) to write a program „test.py“
    - Execute program by typing „python3 test.py“



Executing Python Code

Two ways to do it:

- Interactive
  - Execute lines typed interactively in a Python console/shell
  - Start IDLE, shell is default
  - In terminal type „python3“

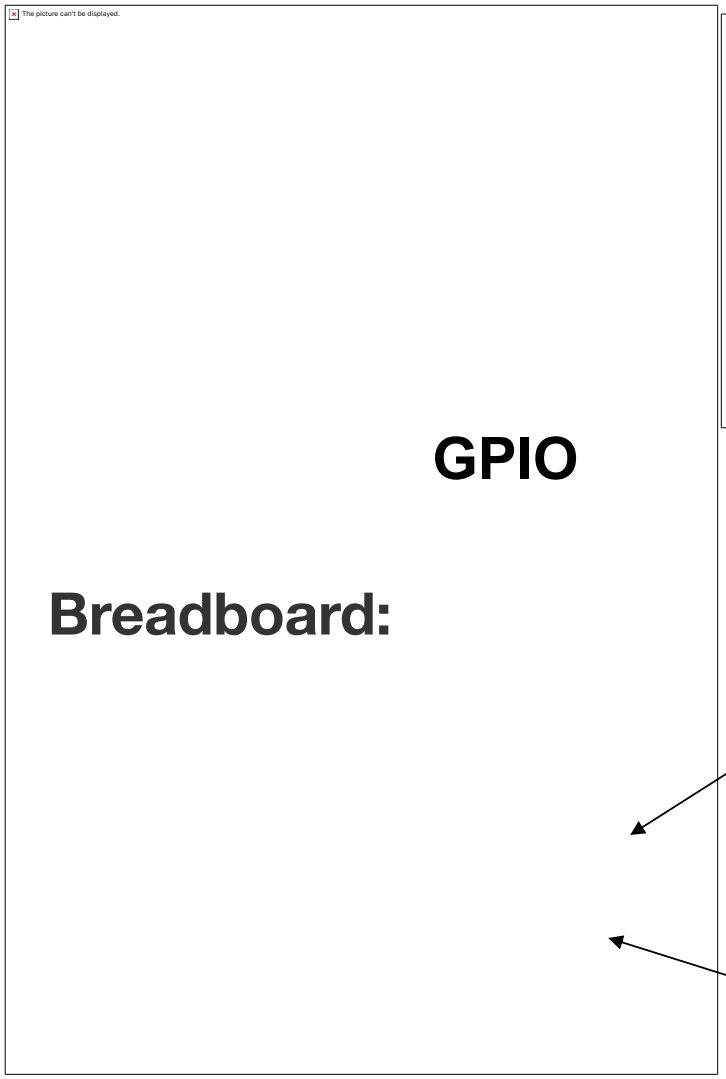


- Batch
  - Execute an entire Python program
  - Start IDLE
  - ***File > New File*** to create a new text editor window

  - Type in code
  - ***Run > Run Module***
  - Python shell will open and code will execute

Setup for Optional LED Lab (1)

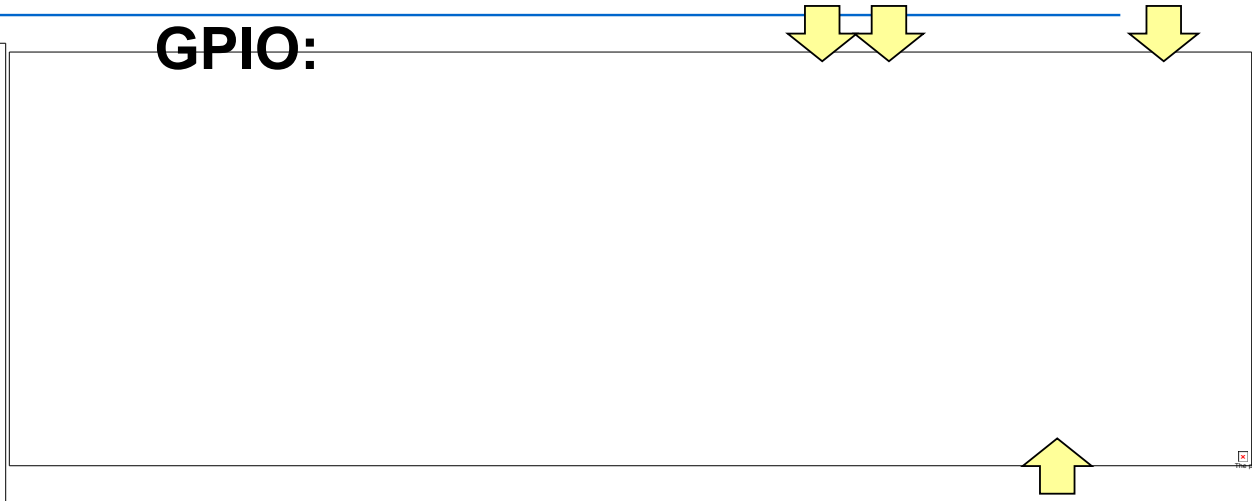
Command: gpio readall



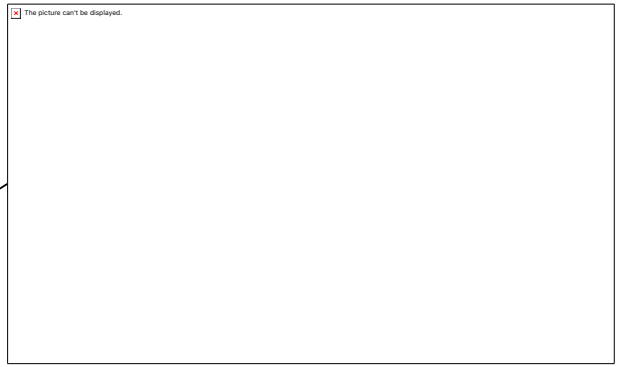
GPIO

Breadboard:

GPIO:



LED:

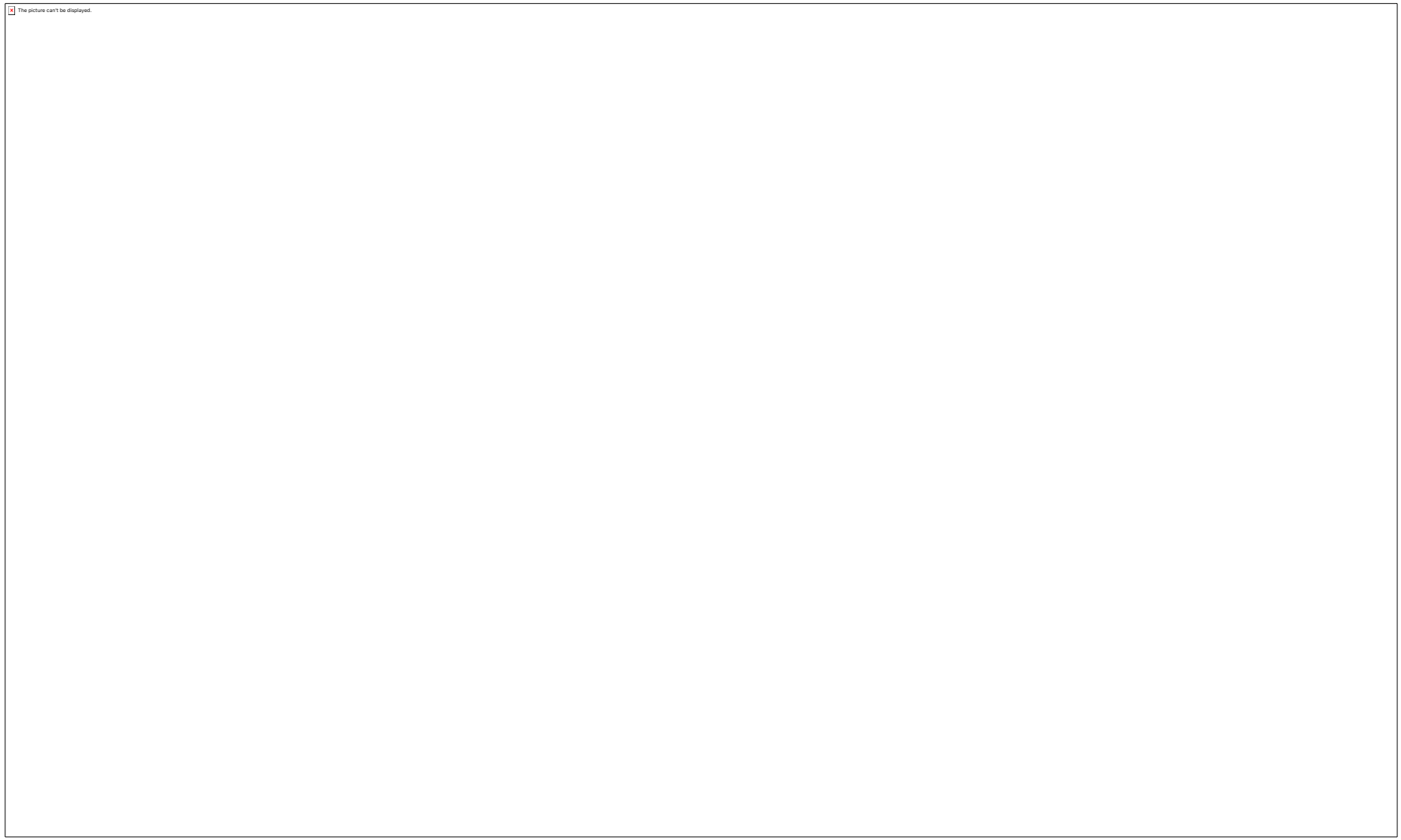


Resistor: Place the LED with a 270 ohm resistor in series with the GND and GPIO pin of Raspberry Pi (Always use a resistor with LEDs to limit the current, else you might end up with a burnt LED!)

Lab Setup (2)

LED on/off

File in Moodle: blink.py

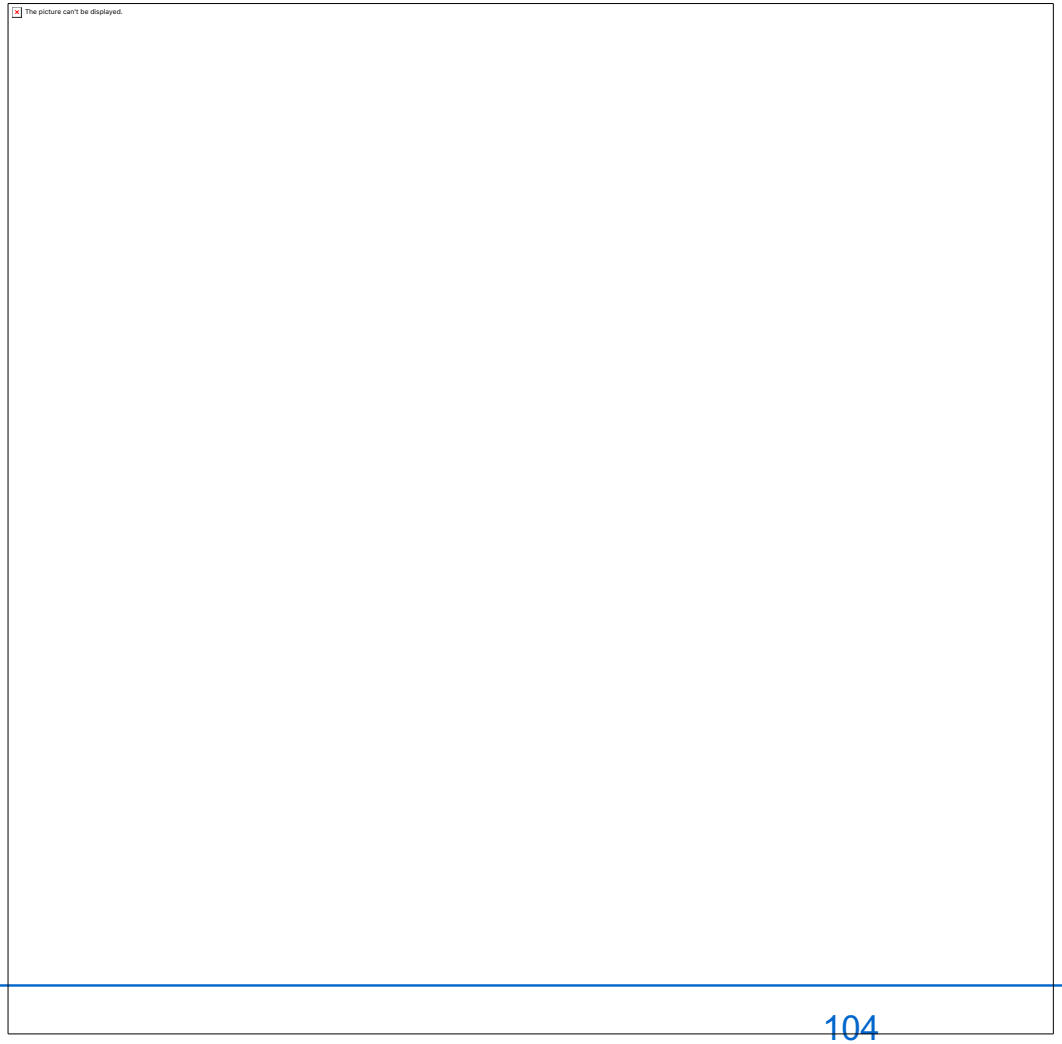


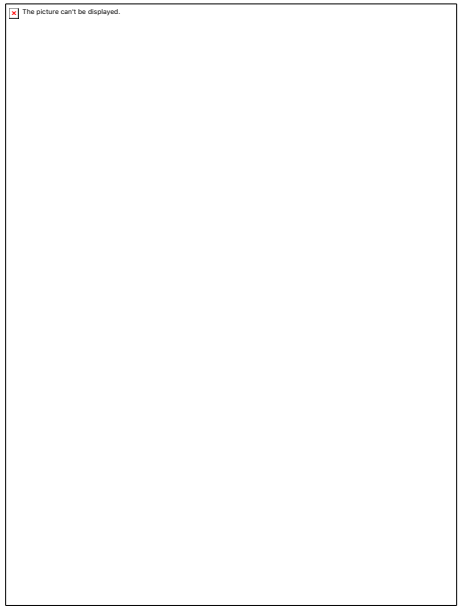
Is Raspberry Pi an IoT Device?

- Maybe – Depends on how it is used!
- Similarities
  - Network connectivity and computational intelligence
  - Small and cheap (relative to a PC)
  - Can interface directly with sensors/actuators via pins
- Differences
  - Interface can be exactly the same as a PC running Linux
    - Complexities of the system can be visible

Raspberry PI - Samples of IoT Projects

http://makezine.com/2013/04/14/47-raspberry-pi-projects-to-inspire-your-next-build/





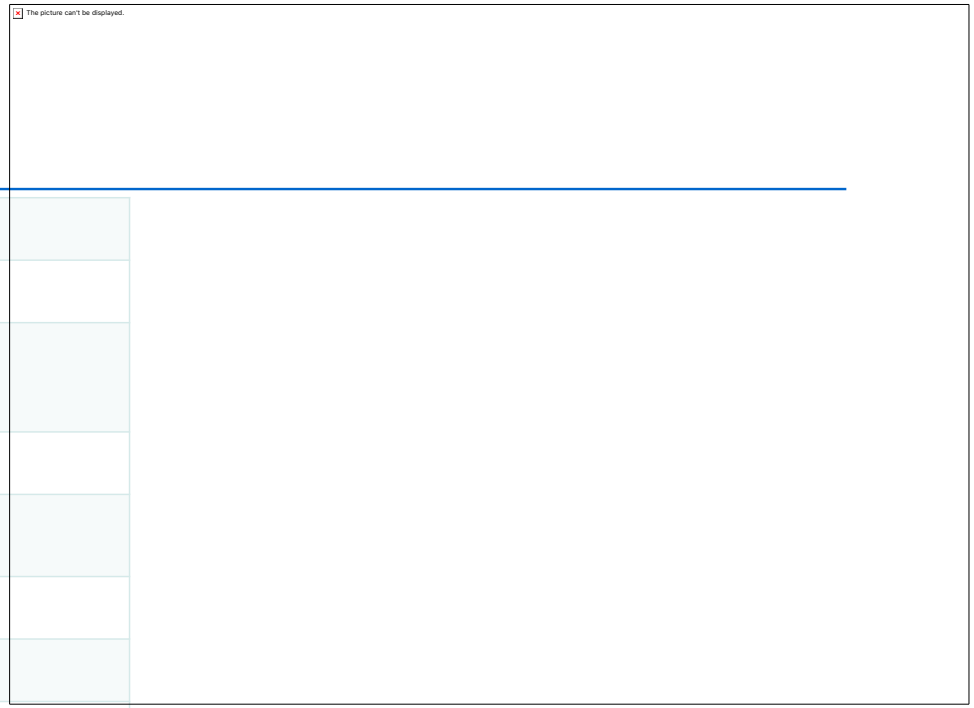
http://drstrangelove.net/2013/12/raspberry-pi-power-cat-feeder-updates/

Lab Hardware: Raspberry Pi

Raspberry PI 3 B All-In-Bundle: **58,78 €**

ARDUINO

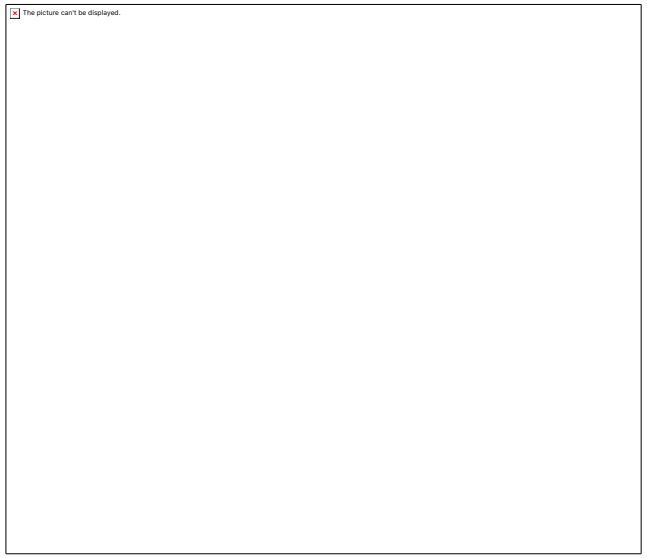
Arduino UNO



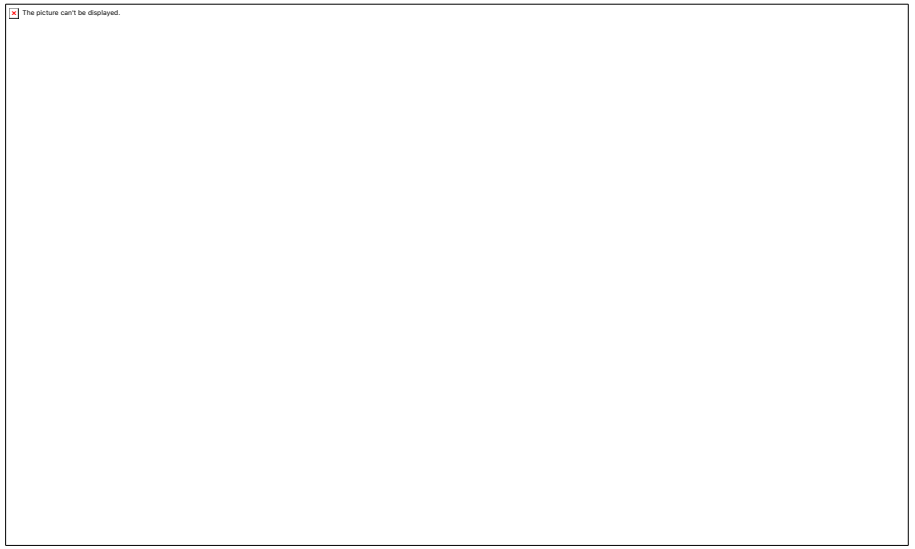
Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14
PWM Digital I/O Pins	6 (out of 14)
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Programming Arduino (1)

- Is designed for turning electronic inputs to outputs
  - Rapidly & Cheaply!



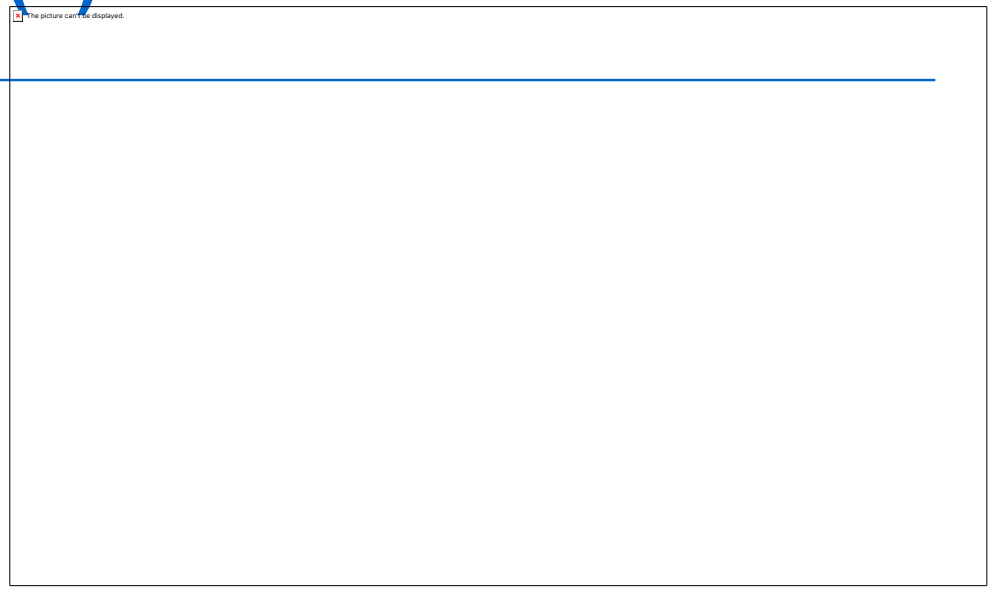
- Writing programs (called *sketches* is done on a separate machine
  - Uploaded to the Arduino for execution



Arduino is based on a microcontroller

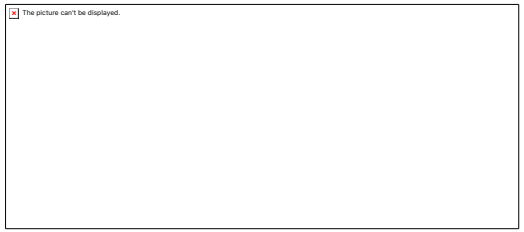
- Microcontroller vs microprocessor
  - Microprocessor = CPU
  - Microcontroller = CPU, RAM, ROM + some peripherals on 1 chip
- How do you run code without an operating system

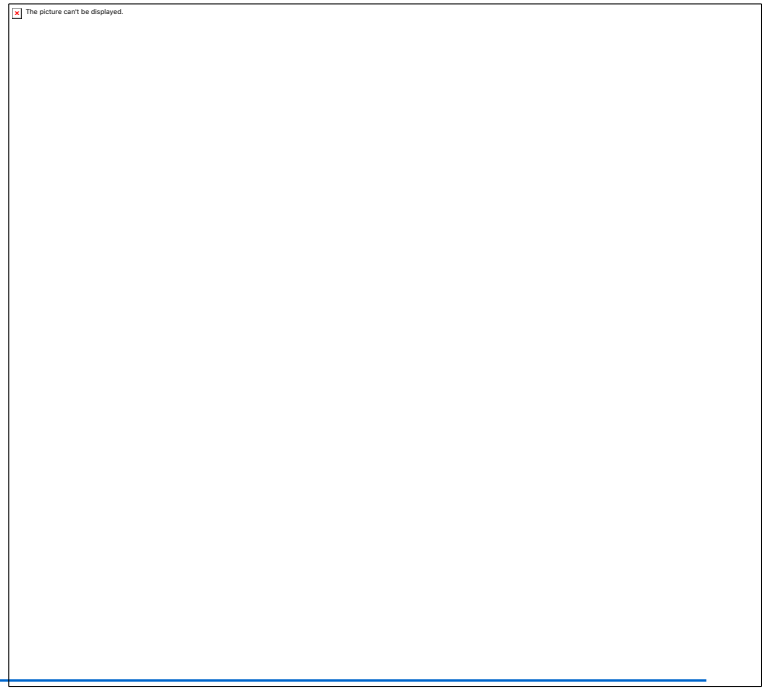
Programming Arduino (2)



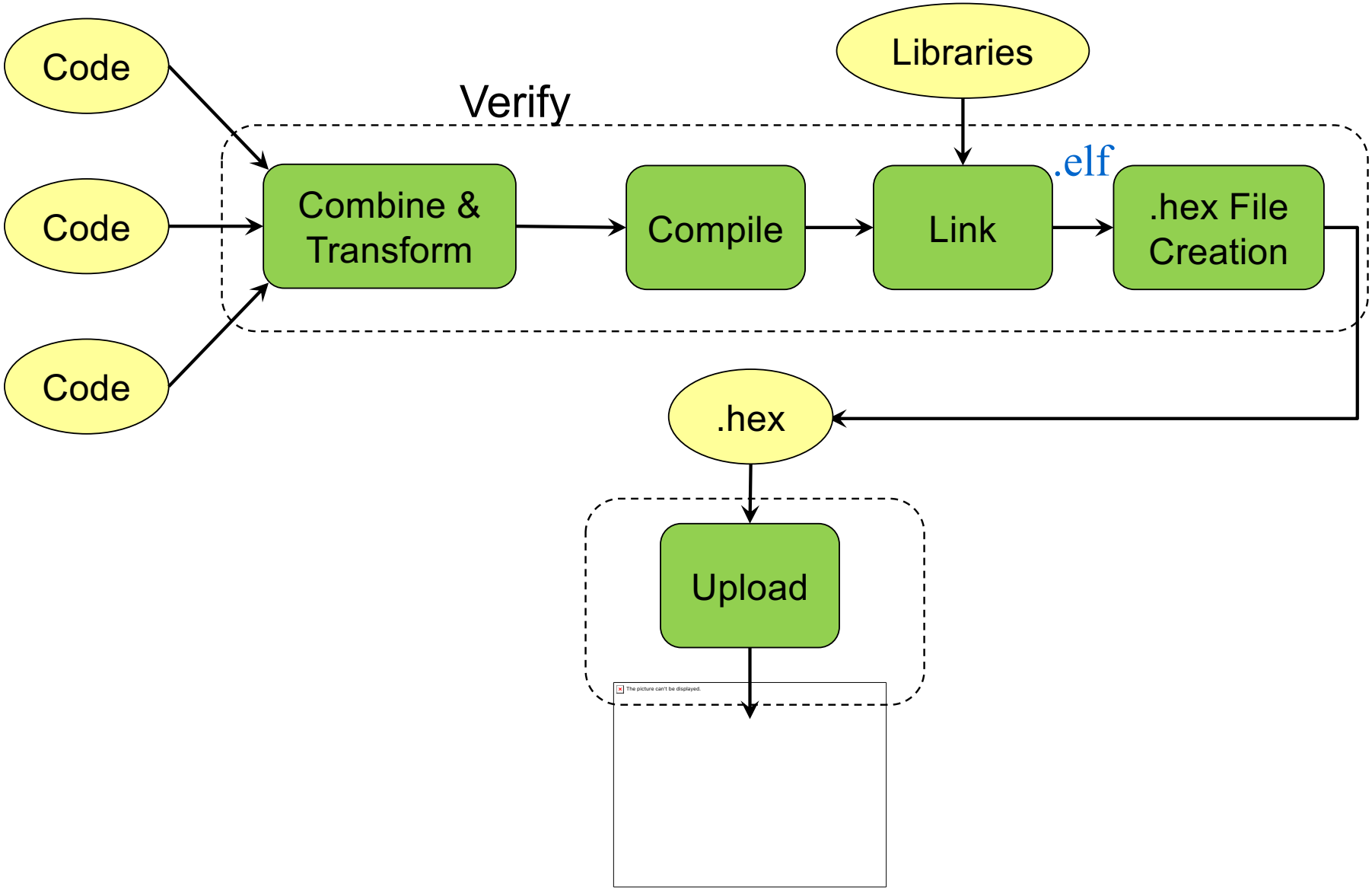
- www.arduino.cc

- IDE
  - Platform agnostic – works on
    - Desktop
    - Laptop
    - Tablets
    - Mobile
  - OS agnostic





Verify and Upload



Verify (1)

Combine & Transform

- All program files are combined into one
- An #include is added to reference basic Arduino libraries
- Function prototypes are added
- A main() function is created

Compile & Link

- avr-gcc is invoked to cross-compile the code
  - Resulting code executes on AVR
- Generates an object file (.o)
- Object file is linked to Arduino library functions
- Result is an .elf file

Verify (2)

Hex File Creation & Programming

- avr-objcopy is invoked to change the format of the executable file
- A .hex file is generated from the .elf file

Arduino Programs

- A program is called sketch
- C++ program using Arduino library functions (C++ is a superset of C)
- Classes defined in libraries
  - Ethernet.begin(mac);
  - Serial.begin(speed);
  - Client.print(“Hello“);
  - Serial.print(“Hello“);

Sketch Structure

Setup() Function

- A sketch does not have a main() function
- Every sketch has a setup() function
  - Executed once Arduino is powered up
  - Used for initialization operations
  - No argument / no return value

```
Void setup(){
        …
}
```

Loop() Function

- Every sketch has a loop() function
  - Executed iteratively as long as Arduino is powered up
  - Loop() starts executing after setup() has finished
  - Loop() is the main program control flow
  - No argument / no return value

```
Void loop(){
        …
}
```

Input/Output (I/O): Functions to Access Pins

Pin Mode

Void pinMode(pin, mode)

- Sets a pin to act as either I/O
- pin is the pin number
  - 0-13 for digital pins
  - A0-A5 for analog pins
- mode is the I/O mode the pin is set to
  - INPUT
  - OUTPUT
  - INPUT_PULLUP: acts as INPUT with reversed polarity

Digital I/O

Int digitalRead(pin)

- Returns state of an input pin
- Returns either LOW (0 volt) or HIGH (5 volts)

Void digitalWrite(pin, value)

- Assigns the state of an output pin
- Assigns either LOW or HIGH

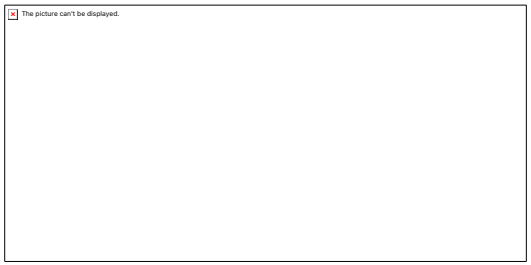
Analog Input

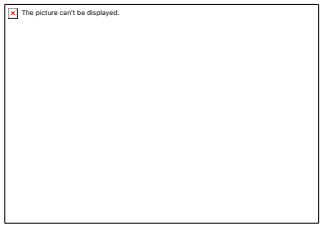
Int analogRead(pin)

- Returns state of an analog input pin
- Returns an integer 0 .. 1023
- 0 (o volt), 1023 (5 volts)

Blink Sketch

Wired LED

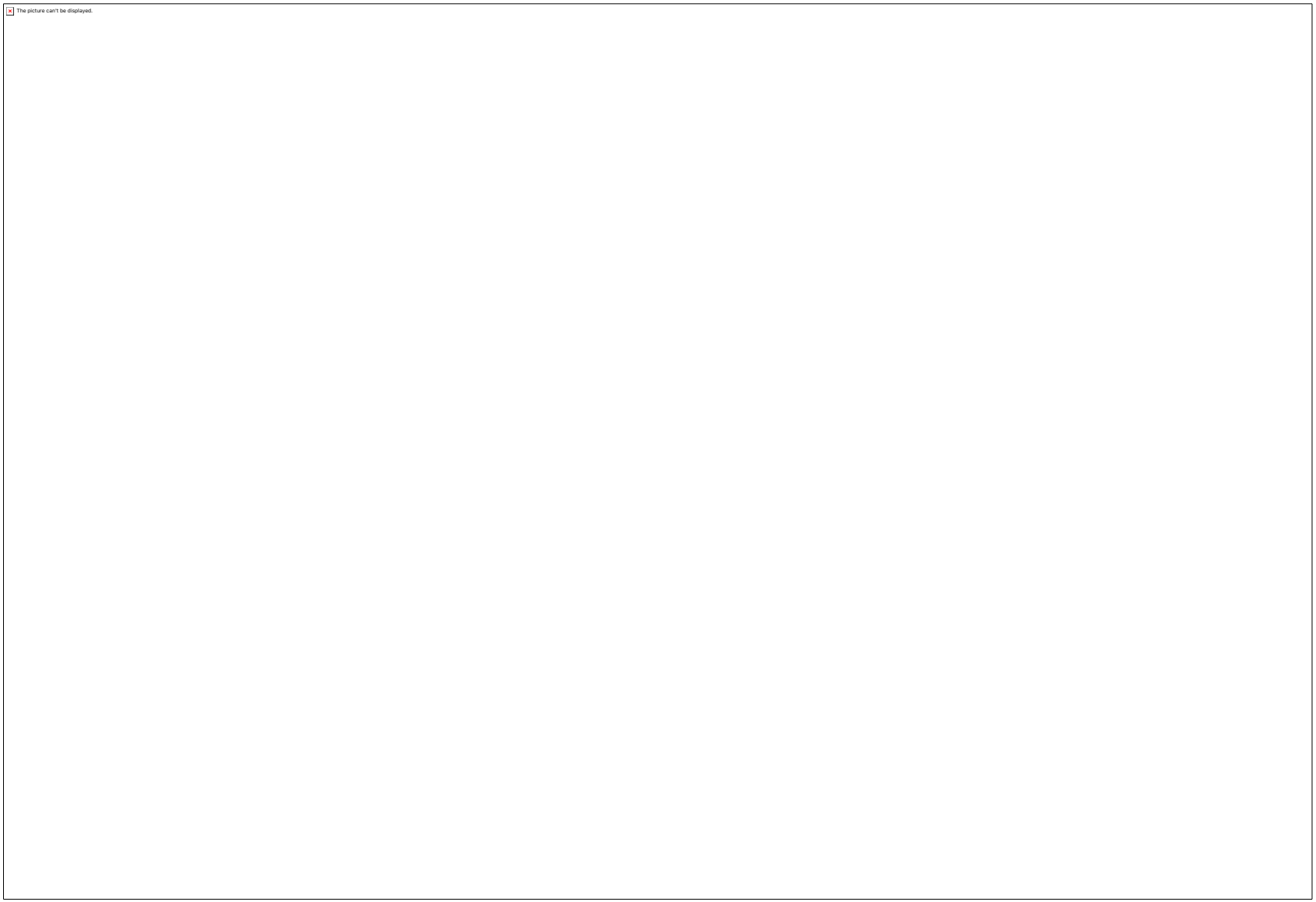




Built-in LED

Lab Hardware: Arduino

Allnet 4duino Starter-Kit **41,98 €**



Raspberry Pi vs. ARDUINO

Raspberry Pi vs. Arduino

	Raspberry Pi	Arduino	
Processor	1200 MHz	16 MHz	PI is faster Larger address space
	64 Bit	8 Bit	
Memory	1024 MB SRAM	2 KB SRAM	Pi has more memory
	4 GB Flash	32 KB Flash	
	-	1 EEPROM	
OS	Full fledged OS	-	
	Processes	-	
IO	3.3 V voltage level	5 V	PI higher energy efficiency
	Ethernet	-	
	SD Card	-	
	-	Analog input	
	Accessing to pins may be time-consuming	Accurate time for writing to pins	Arduino better supports time-sensitive applications