Introduction to the Raspberry Pi

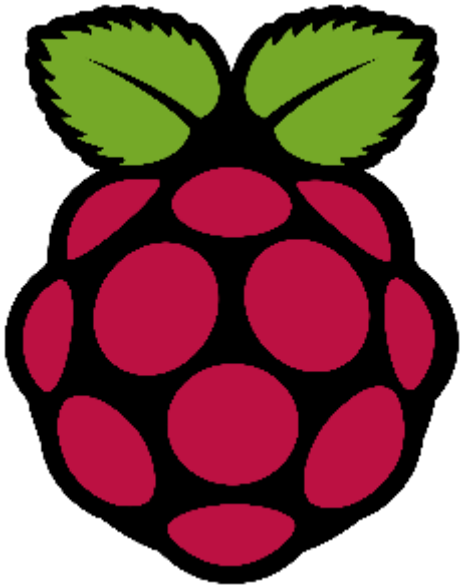
Jani Kalasniemi

CERN

Ideas

Agenda

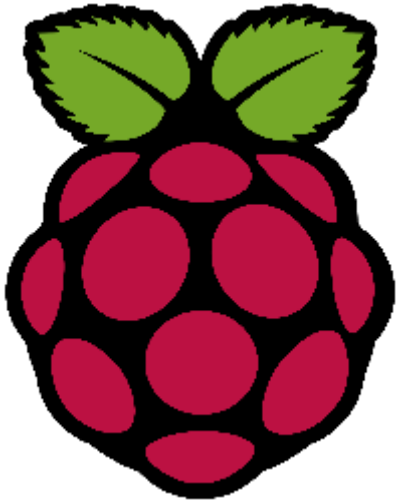
- FIRST STEPS
  - What is a Raspberry Pi?
  - Setting up your Raspberry Pi
  - Using your Raspberry Pi as a Desktop Computer
- PROGRAMMING WITH PYTHON
  - Start Programming in Python
  - Write your own Function
  - Using GPIO to drive outputs and read inputs
- MAKING A DESKTOP APPLICATION
  - Learn about event-driven programming
  - Create a simple Graphical User Interface(GUI)



CERN

Ideas

FIRST STEPS



CERN

Ideas

What is a Raspberry Pi?

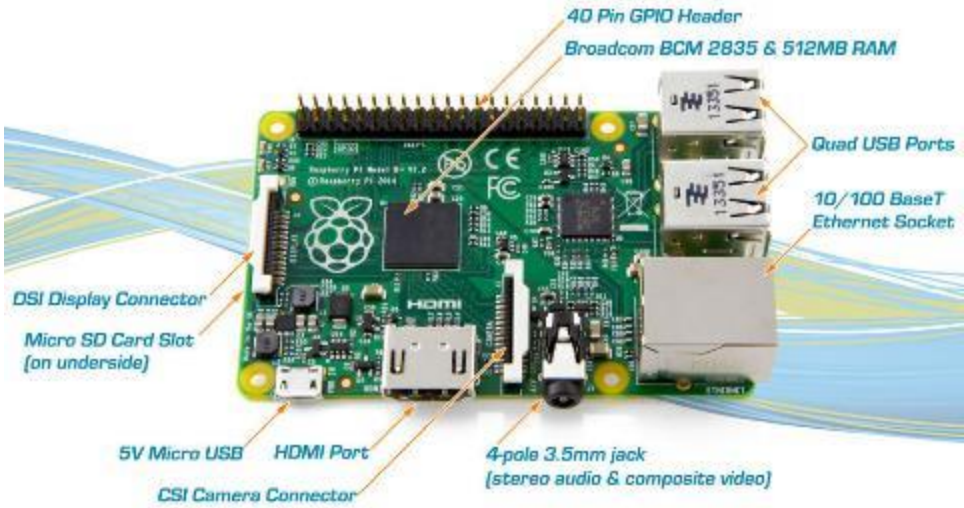
- A credit-card-sized computer
- Runs on several operating systems
  - Raspbian WE WILL USE RASPBIAN
  - Windows 10 IoT Core
  - RetroPie
  - OpenElec
- Possible to connect with a variety of sensors to interact with the physical world.
  - Make decisions based on processing of gathered sensor data

CERN Ideas

Getting started with your Raspberry Pi

Basic 1-2-3

1. Connect the Raspberry Pi to the keyboard, mouse and the screen.
2. Plug in the power and wait for it to boot.
3. **Username** : **pi**
   Password : raspberry
   (should not be needed)



CERN
Ideas

Using you Raspberry Pi as a Desktop Computer

- Try to access the internet through your Raspberry Pi and find a picture that reflects your group. Hands up when you are done, for help if needed!
- Save this picture to the desktop.

CERN

Ideas

Communicating through the terminal

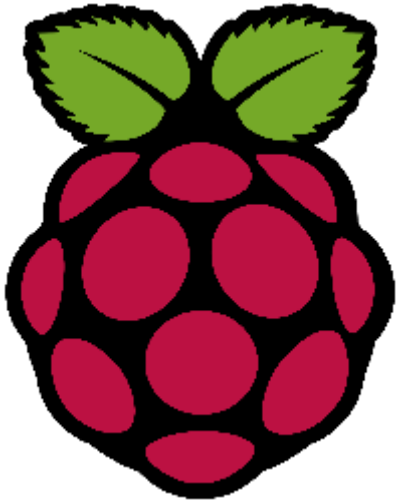
- Start the terminal by going to Accessories and then clicking on the Terminal program.

- Run through the commands and get a feel for how to change directories, and list its contents.

Basic Commands cheat-sheet	
Command	**Description**
`ls`	List contents of directory
`cd <dir>`	Change to directory: <dir>
`mkdir <dir>`	Make a new directory called <dir>
`rmdir <dir>`	Delete the empty directory, <dir>
`rm <file>`	Delete the file, <file>
`nano`	Run the *nano* text editor program

CERN Ideas

PROGRAMMING WITH PYTHON



CERN

Ideas

Programming in Python3

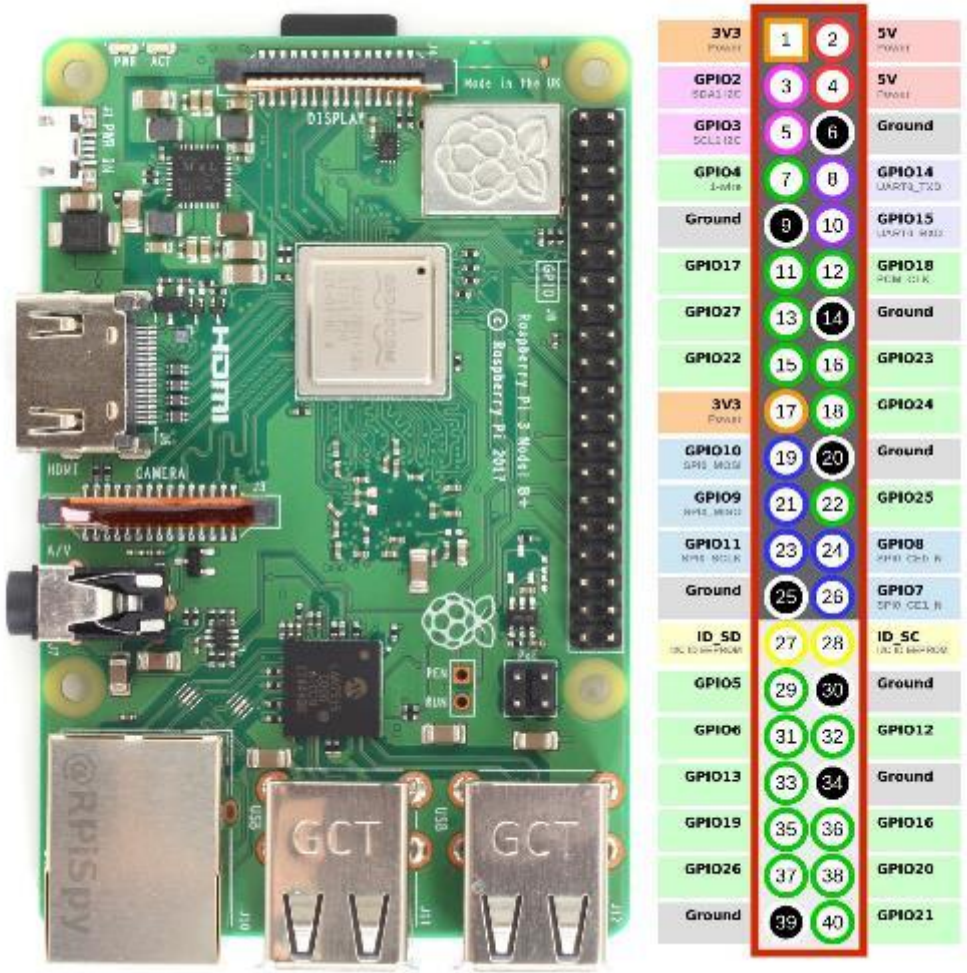
- General-purpose programming language used for scientific and numerical applications as well as desktop and web applications.
- Open source language that has a lot of online resources for problems you might come across.
- We are using Thonny IDE for programming with Python3 in our Raspberry Pi 3 Model B
- **https://stackoverflow.com is your best friend for programming troubles**

CERN

Ideas

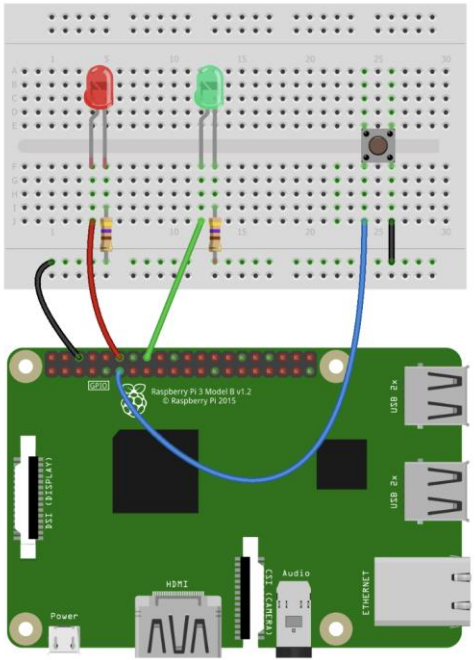
The GPIO layout

General
Purpose
Input /
Output

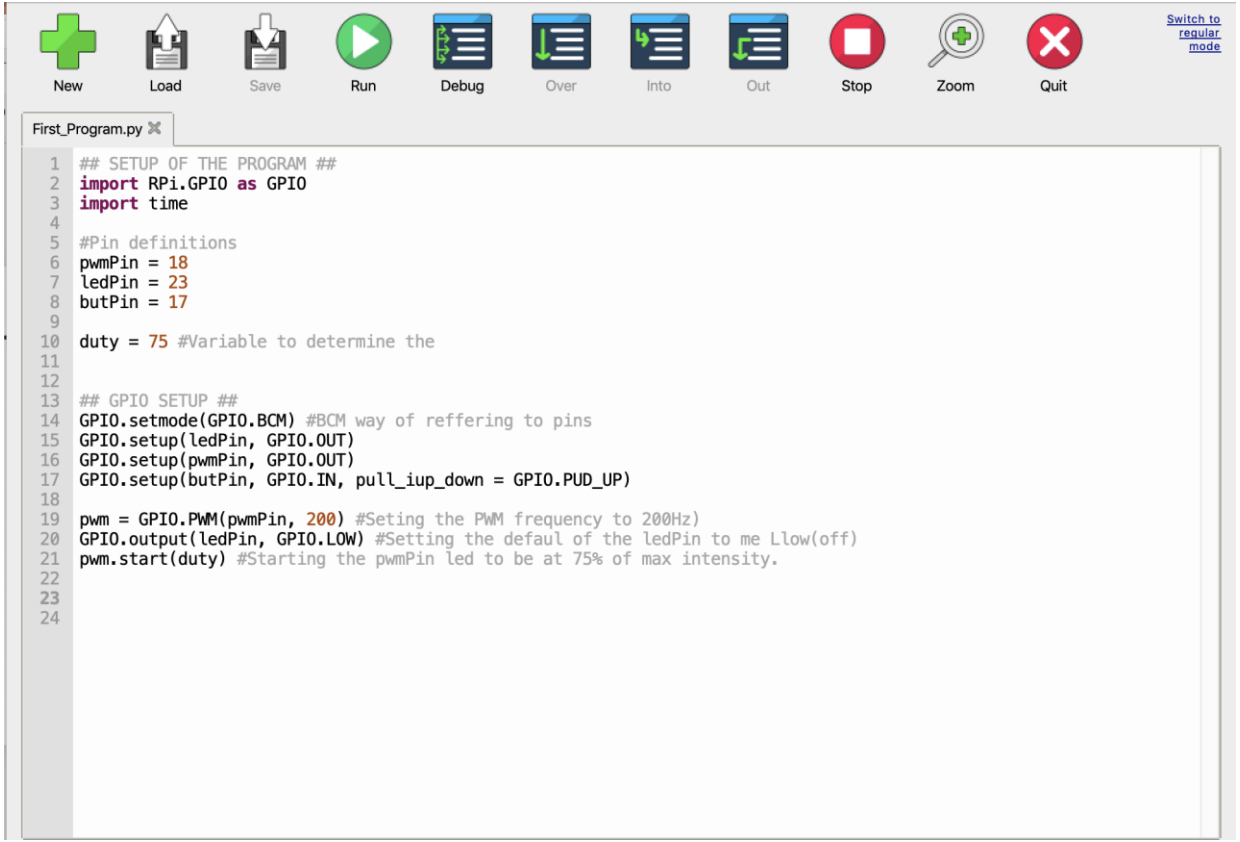


Use the GPIO to drive outputs and read inputs

- Build the circuit in the figure. Use 470 Ohm resistor
- Next we will make a program that makes one LED blink on and off, and the other dim up and down in idensity when the Button is pressed.



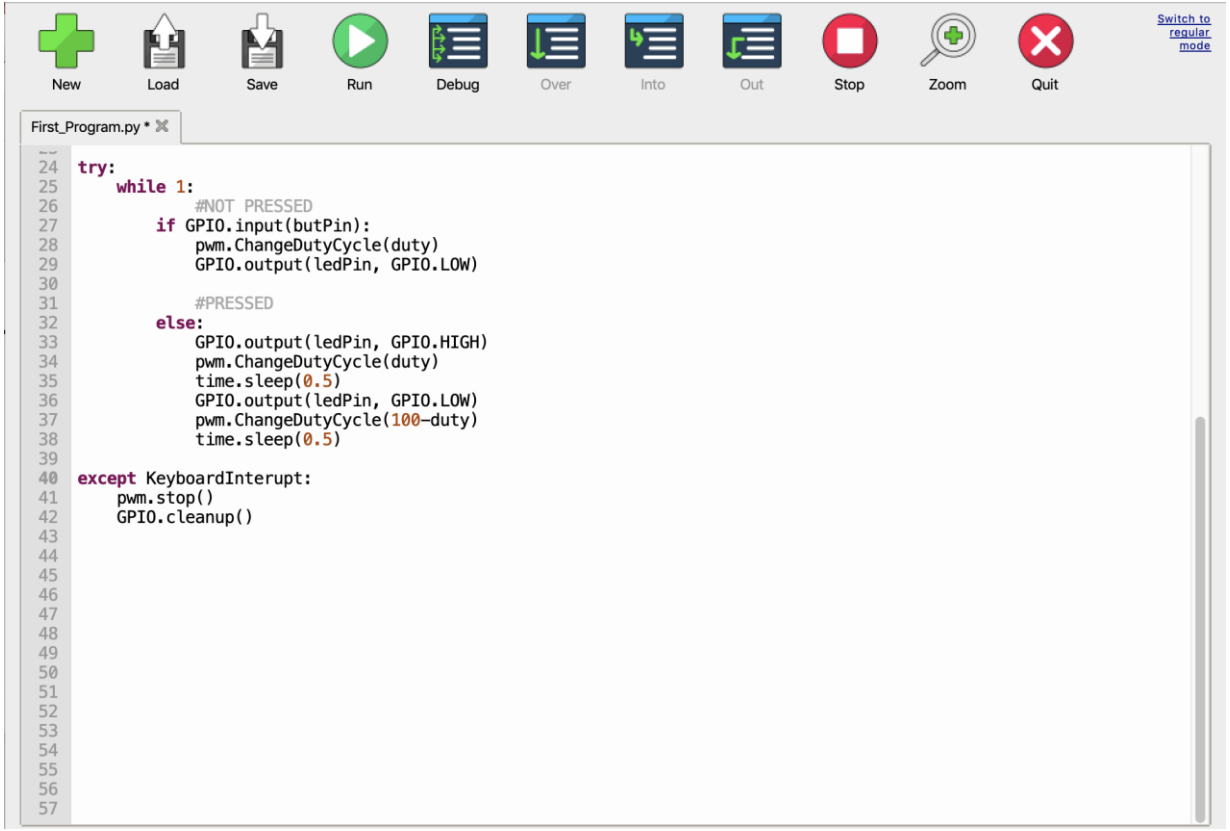
Use the GPIO to drive outputs and read inputs




```
## SETUP OF THE PROGRAM ##
import RPi.GPIO as GPIO
import time

#Pin definitions
pwmPin = 18
ledPin = 23
butPin = 17

duty = 75 #Variable to determine the


## GPIO SETUP ##
GPIO.setmode(GPIO.BCM) #BCM way of reffering to pins
GPIO.setup(ledPin, GPIO.OUT)
GPIO.setup(pwmPin, GPIO.OUT)
GPIO.setup(butPin, GPIO.IN, pull_iup_down = GPIO.PUD_UP)

pwm = GPIO.PWM(pwmPin, 200) #Seting the PWM frequency to 200Hz)
GPIO.output(ledPin, GPIO.LOW) #Setting the defaul of the ledPin to me Llow(off)
pwm.start(duty) #Starting the pwmPin led to be at 75% of max intensity.
```

CERN Ideas

Use the GPIO to drive outputs and read inputs




```
try:
    while 1:
            #NOT PRESSED
        if GPIO.input(butPin):
            pwm.ChangeDutyCycle(duty)
            GPIO.output(ledPin, GPIO.LOW)

            #PRESSED
        else:
            GPIO.output(ledPin, GPIO.HIGH)
            pwm.ChangeDutyCycle(duty)
            time.sleep(0.5)
            GPIO.output(ledPin, GPIO.LOW)
            pwm.ChangeDutyCycle(100-duty)
            time.sleep(0.5)

except KeyboardInterupt:
    pwm.stop()
    GPIO.cleanup()
```

CERN Ideas

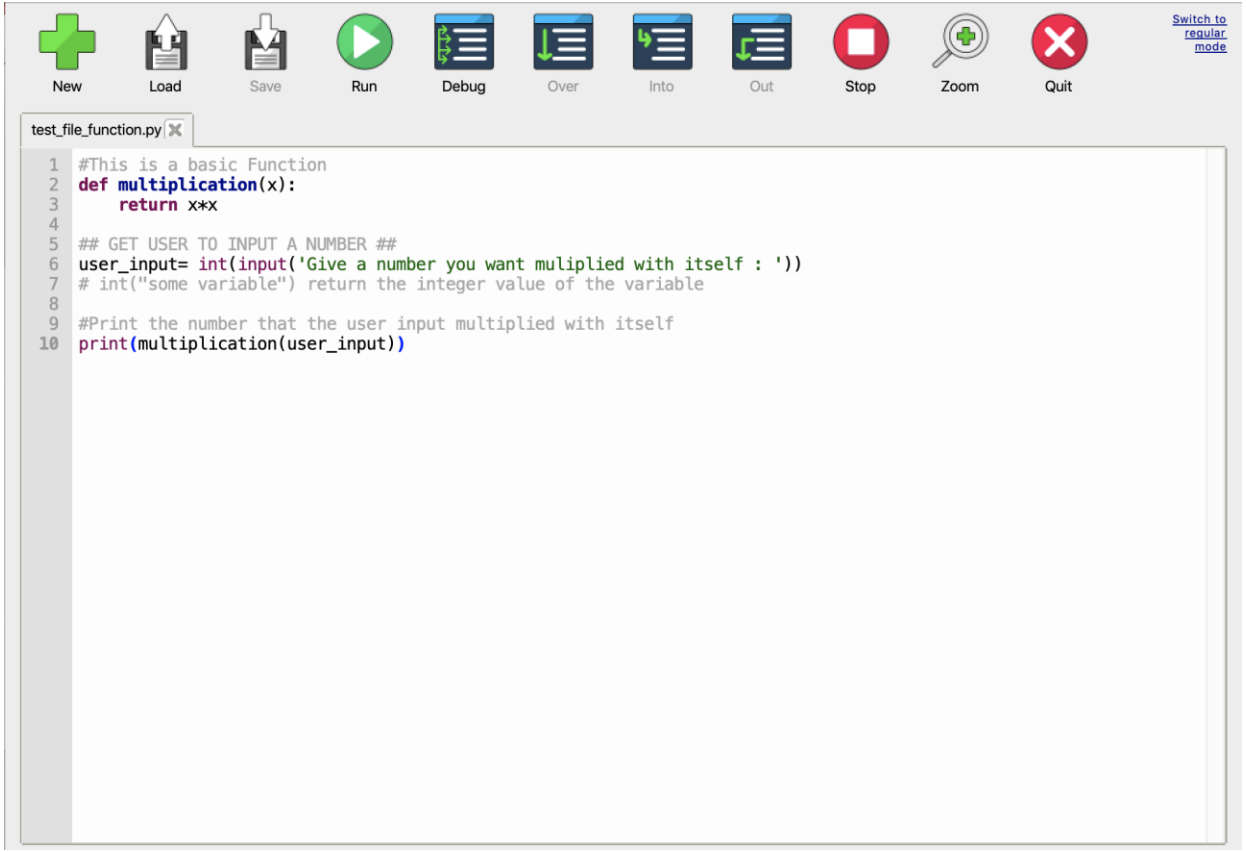
Common errors and debugging

- Python is sensetive to intendation.
- Make sure that the variables and the functions have the same exact writing. Watch out for small and Capital letters!
- Remember that python starts counting from zero.
  i.e when using loops.

Test your program and what you have made!

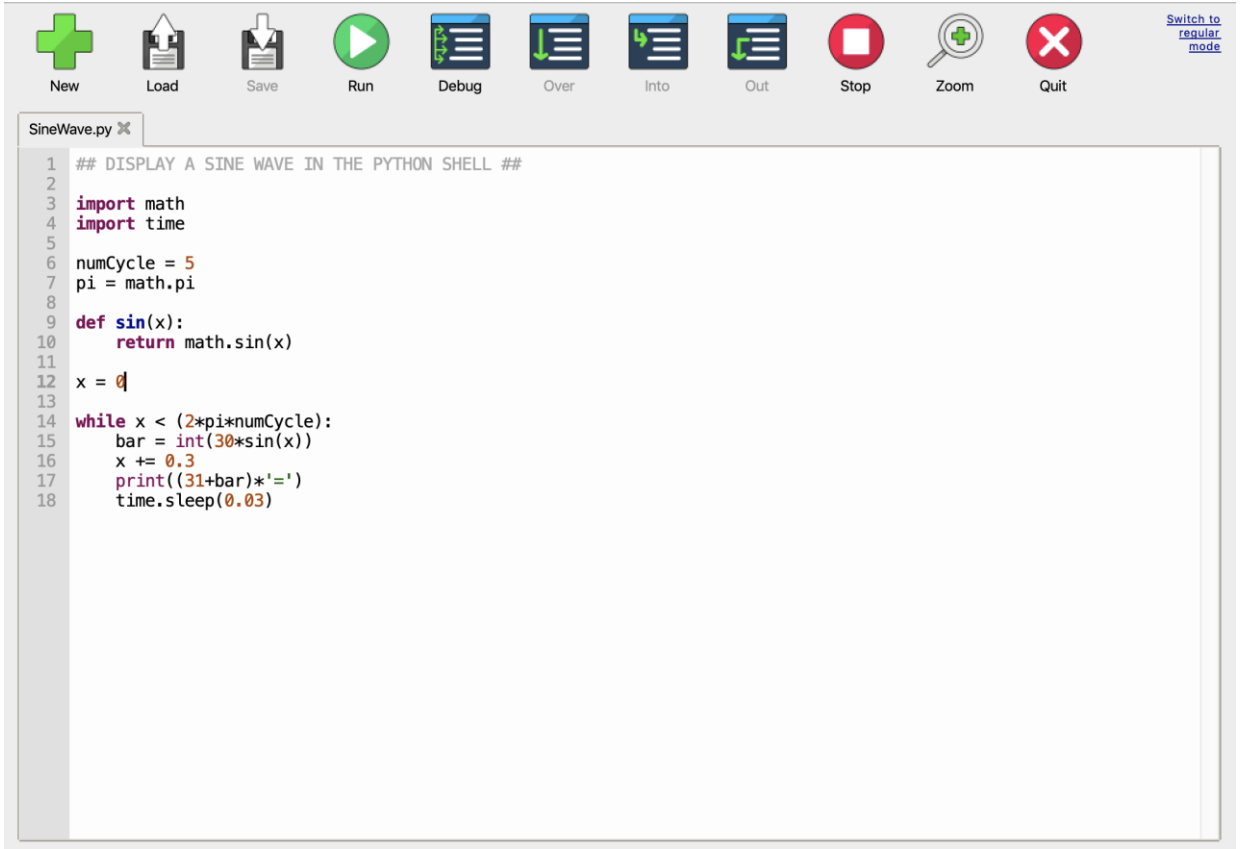
Questions?

Write you own Functions



CERN Ideas

Making a sin(x) function to get some output




```
## DISPLAY A SINE WAVE IN THE PYTHON SHELL ##

import math
import time

numCycle = 5
pi = math.pi

def sin(x):
    return math.sin(x)

x = 0

while x < (2*pi*numCycle):
    bar = int(30*sin(x))
    x += 0.3
    print((31+bar)*'=')
    time.sleep(0.03)
```

Should return something like this

Shell

```
=========
==========
=====================
==============================
======================================
==============================================
==================================================
==========================================================
==========================================================
==========================================================
==========================================================
==================================================
==========================================
==============================
======================
==============
==========
=====
==
==
====
=========
==========
======================
==============================
======================================
==============================================
==================================================
==========================================================
==========================================================
==========================================================
==========================================
==============================
======================
==========
=========
====
==
==
=====
==========
==========
======================
```

CERN

Ideas

Using a sensor with the Raspberry Pi

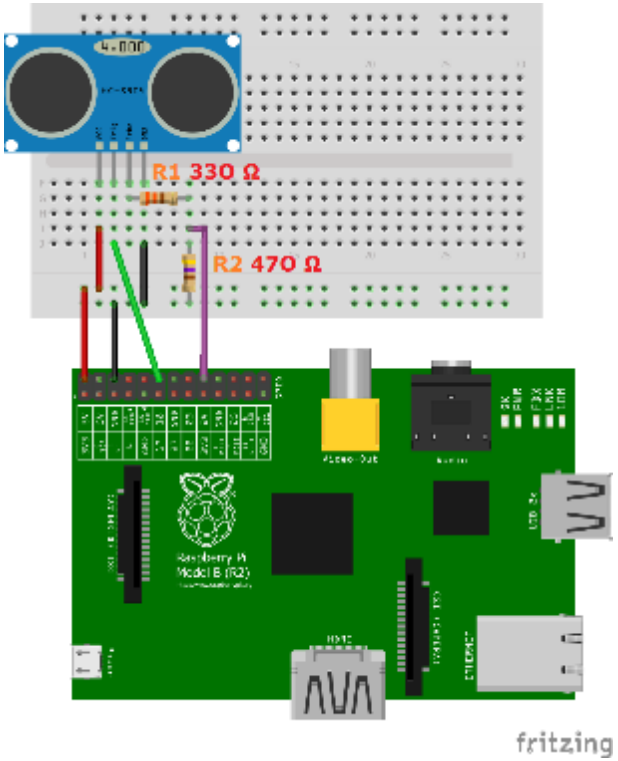
- The GPIO on the Raspberry Pi can only handle a maximum of 3.3V, but many of the available sensors can return a signal of 5V to the GPIO. This will damage the Raspberry Pi.
- How do we deal with this?
  - Resistors connected as voltage dividers.
- Important to be aware of this.

CERN

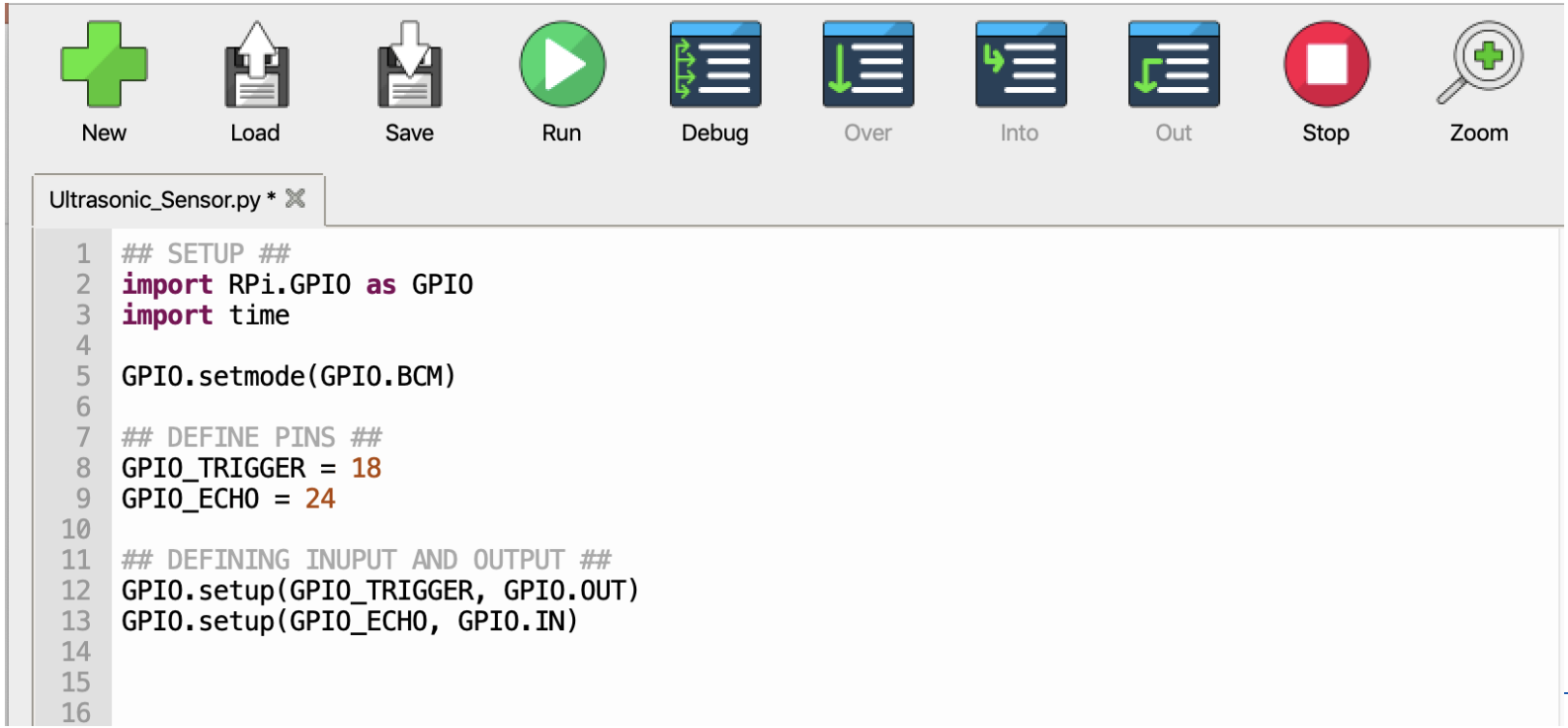
Ideas

Measuring Distance with a Ultrasonic Sensor: HC-SR04

- The Trigger sends a burst of sound that bounces and hits the Echo.
- If we measure the time it takes from sending to receiving we can calculate the distance.
- NOTE: Two different resistors here



Setting up the Raspberry Pi




```
1  ## SETUP ##
2  import RPi.GPIO as GPIO
3  import time
4
5  GPIO.setmode(GPIO.BCM)
6
7  ## DEFINE PINS ##
8  GPIO_TRIGGER = 18
9  GPIO_ECHO = 24
10
11 ## DEFINING INUPUT AND OUTPUT ##
12 GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
13 GPIO.setup(GPIO_ECHO, GPIO.IN)
14
15
16
```

CERN Ideas

Now we need a function to calculate the distance

- With the sensor we can measure the time it takes for a signal to be sent out and reflected back.
- We need to convert the elapsed time to distance by using the speed of sound in air and multiplying it with the time divided by two (Because it travels to the object, and back).

CERN

Ideas

The distance function

```
17  def distance():
18      # set Trigger to HIGH,
19      GPIO.output(GPIO_TRIGGER, True)
20
21      # set Trigger after 0.01ms to LOW
22      # this sends a signal that the ECHO can read in
23      time.sleep(0.00001)
24      GPIO.output(GPIO_TRIGGER, False)
25
26      StartTime = time.time()
27      StopTime = time.time()
28
29      # save StartTime
30      while GPIO.input(GPIO_ECHO) == 0:
31          StartTime = time.time()
32
33      # save time of arrival
34      while GPIO.input(GPIO_ECHO) == 1:
35          StopTime = time.time()
36
37      # time difference between start and arrival
38      TimeElapsed = StopTime - StartTime
39      # multiply with the sonic speed (34300 cm/s)
40      # and divide by 2, because: there and back
41      distance = (TimeElapsed * 34300) / 2
42
43      #Return the Distance that the sensor measured
44      return distance
```

CERN Ideas

Making it loop until we tell it to stop

```
47  try:
48      while True:
49          #Find the measured distance with distance()
50          dist = distance()
51          print ("Measured Distance = %.1f cm" % dist)
52
53          #Wait one second before a new measurement is made
54          time.sleep(1)
55
56      # Reset by pressing CTRL + C
57  except KeyboardInterrupt:
58      print("Measurement stopped by User")
59      GPIO.cleanup()
60
```

CERN

Ideas

If the program is not giving any measurements

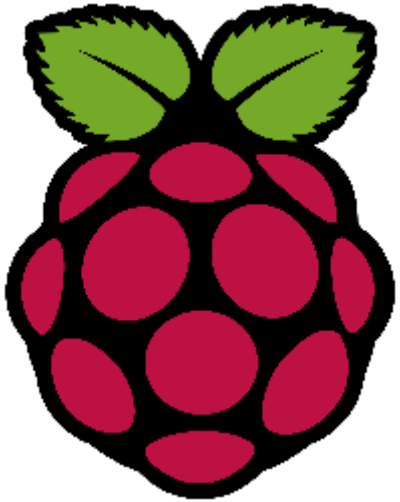
- Check the wiring!
- Check you code one more time.
- Check the resistors
- Check the wiring!

Questions?

CERN

Idea^S

MAKING A DESKTOP APPLICATION



CERN

Idea^{s}

Introducing Tkinter : A GUI Toolkit for Python

- De facto standard GUI for Python
- Toolkit for GUI programming in Python. Not the only one, but the most commonly used one.
- Enables you to build GUI’s with buttons, sliders, drop downs, and other interactions for the user of the program/device you build.

CERN

Ideas

Event driven programming.

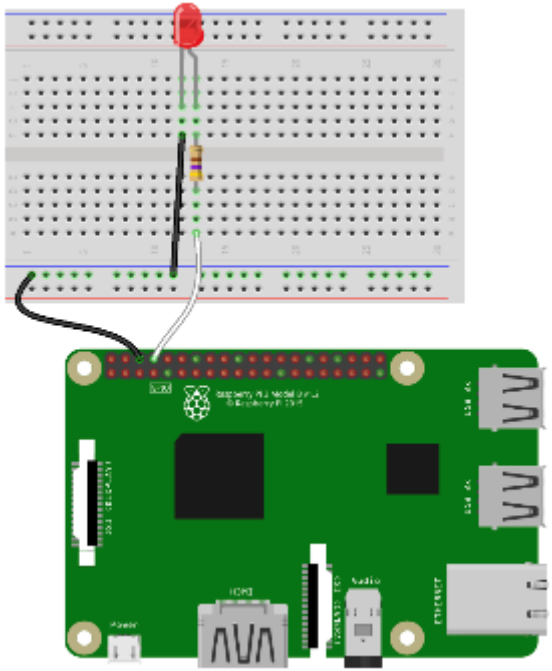
- When creating a GUI you use event driven programming.
- In the GUI all buttons, sliders, etc, are known as **Widgets.**
  - When a widget is used it make an event in you code start.

CERN

Ideas

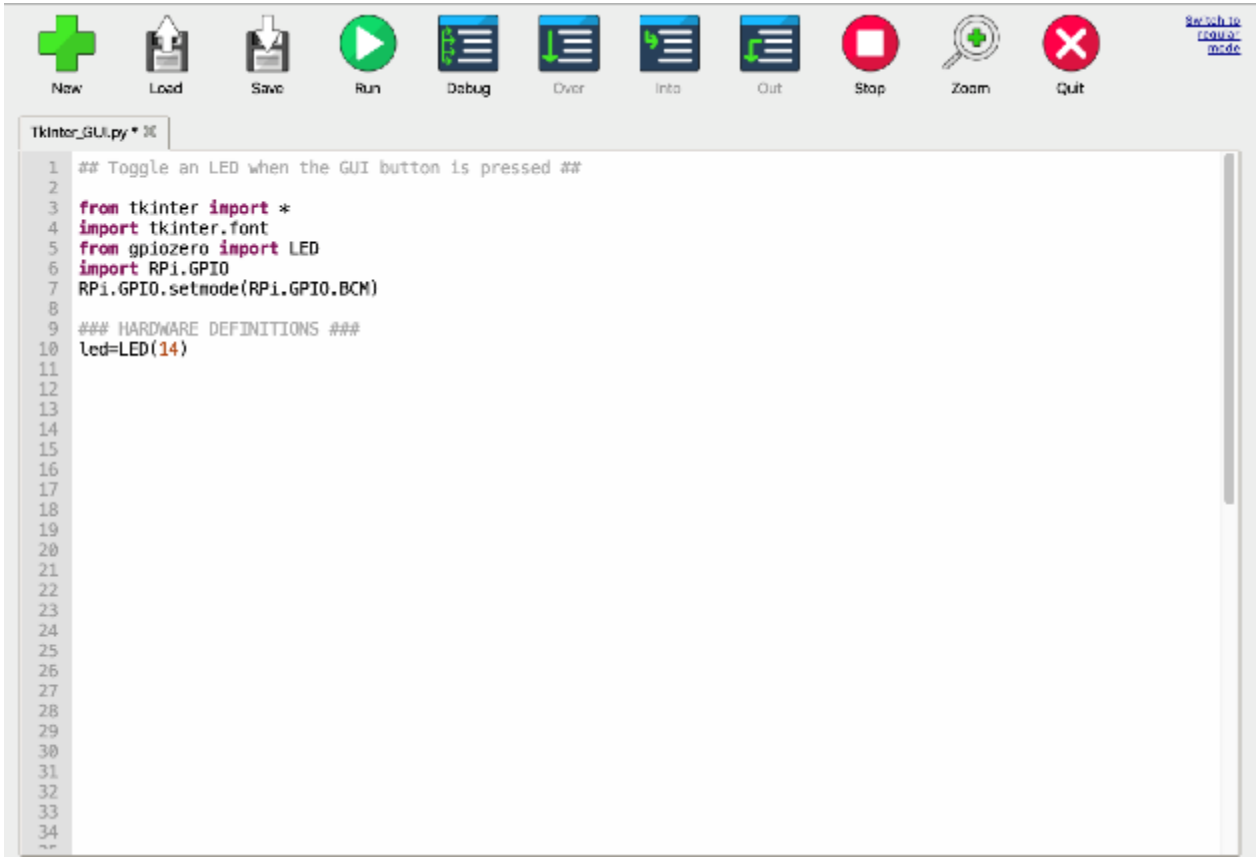
Create a simple Graphical User Interface(GUI)

- We will create a GUI that enables us to turn the LED on and off with a button in a GUI.
- Use the 470 Ohm resistor in the circuit.

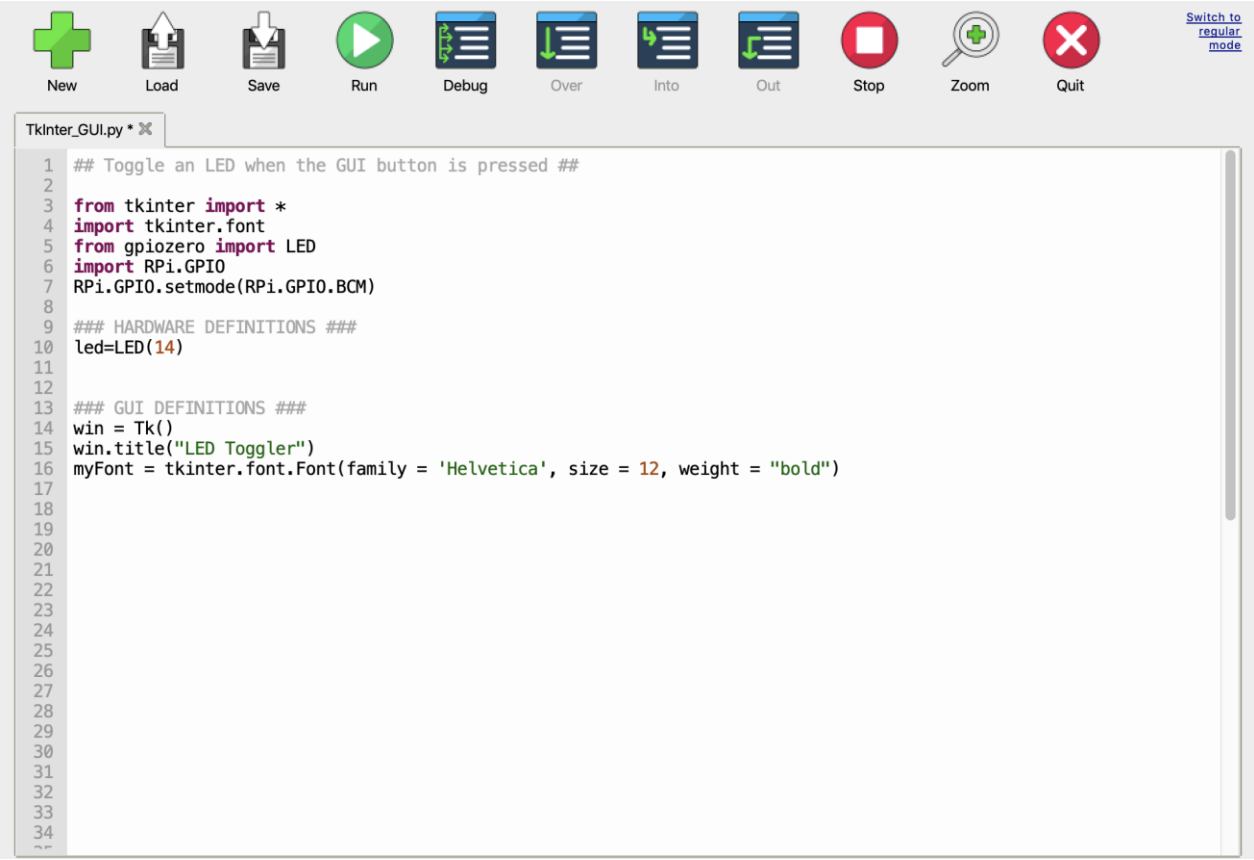


CERN Ideas

Building the code for this GUI



Building the GUI




```
## Toggle an LED when the GUI button is pressed ##

from tkinter import *
import tkinter.font
from gpiozero import LED
import RPi.GPIO
RPi.GPIO.setmode(RPi.GPIO.BCM)

### HARDWARE DEFINITIONS ###
led=LED(14)


### GUI DEFINITIONS ###
win = Tk()
win.title("LED Toggler")
myFont = tkinter.font.Font(family = 'Helvetica', size = 12, weight = "bold")
```

Building the GUI

- Now we need do decide what the GUI will actually do.

1. We will create a button that will switch the LED on and off

2. We also have to make an exit button so that when we exit the program the LED turns off, and the GPIO port is reset to default as input ports.

Creating the widget triggered event : ledToggle()

```
19  ### Event Functions ###
20  def ledToggle():
21      if led.is_lit:
22          led.off()
23          ledButton["text"]="Turn LED on" # Change only the button text property
24      else:
25          led.on()
26          ledButton["text"]="Turn LED off"
```

CERN

Ideas

Creating the widget triggered event : close()

```
29 def close():
30     RPi.GPIO.cleanup()
31     win.destroy()
```

CERN

Ideas

Creting the toggle button, aka a Widget

```
35  ### WIDGETS ###
36
37  # Button, triggers the connected command when it is pressed
38  ledButton = Button(win, text='Turn LED on', font=myFont, command=ledToggle, bg='bisque2', height=1, width=24)
39  ledButton.grid(row=0,column=1)
```

Creating the exit button

```
41 exitButton = Button(win, text='Exit', font=myFont, command=close, bg='red', height=1, width=6)
42 exitButton.grid(row=2, column=1)
```

CERN

Ideas

What happens if we exit the program through the window exit?

Making sure that we have a clean exit

```
44 win.protocol("WM_DELETE_WINDOW", close) # cleanup GPIO when user closes window
45
```

What happens if you run your program and press a GUI button?

Making the program run after we exit it on purpose

```
46 win.mainloop() # Loops forever
47
48 print('will this be printed?')
49
```

Will the last line be printed? If so, when?

CERN

Ideas

Test your program and what you have made!

Questions?

Summary – What have we done?

- Introduced the Raspberry Pi and used it as a computer
- Learned simple commands to use in terminal
- Introduced the GPIO and how to get inputs from the physical world, interpret them, and create output from input
- Used a **XX** sensor to read and evaluate data from your environment
- Introduced the Tkinter package and created a simple, event-driven, GUI

Resources

- **Raspberry Pi GPIO examples:** https://sourceforge.net/p/raspberry-gpio-python/wiki/Examples/
- **Python Documentation:** https://www.python.org/doc/
- **Tkinter Documentation:** https://www.tutorialspoint.com/python/python_gui_programming.htm

CERN
Ideas


CERN
Ideas