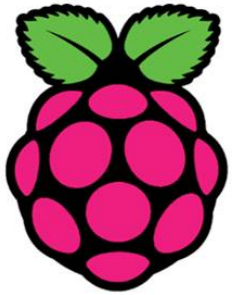
CGS

STEM for All Seasons

Athens Training

8th – 12th May 2017

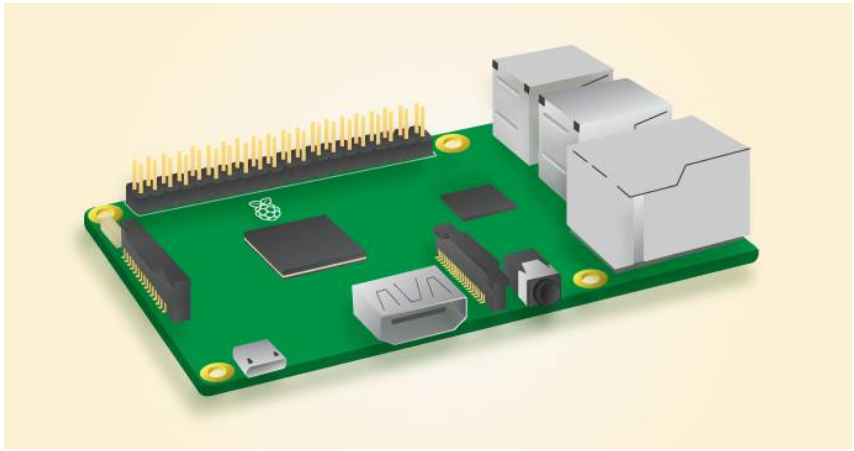


RaspberryPi

Erasmus+

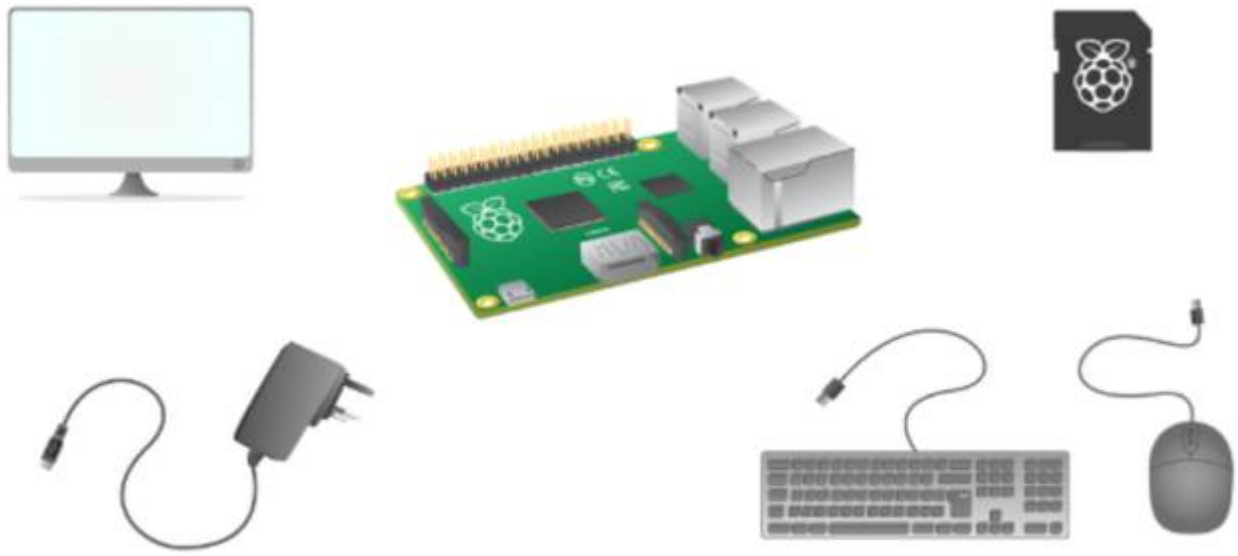
What is the Raspberry Pi?

- It may not look much like the computers we are used to, but this is because we are accustomed to seeing a computer in a case, with a monitor, keyboard and mouse attached.
- It is known as a **single board computer**.



CGS

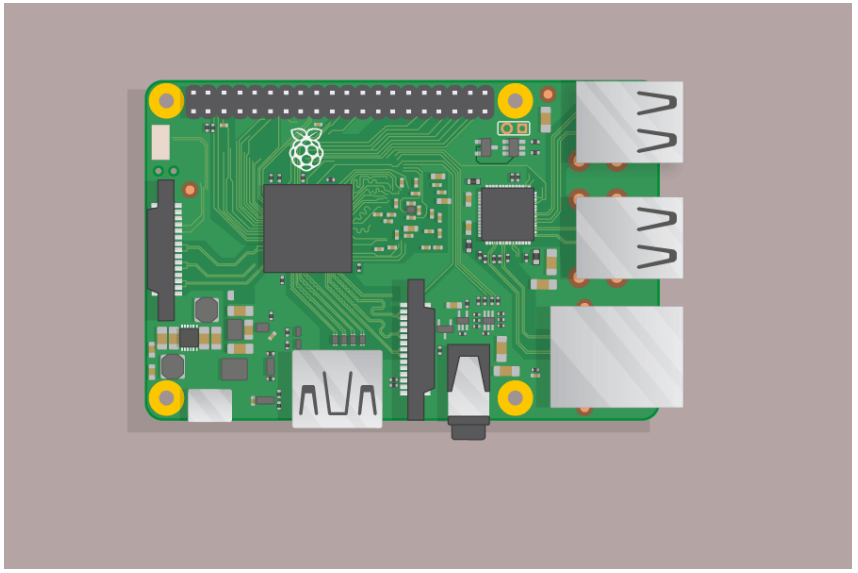
Connecting all the things



CGS

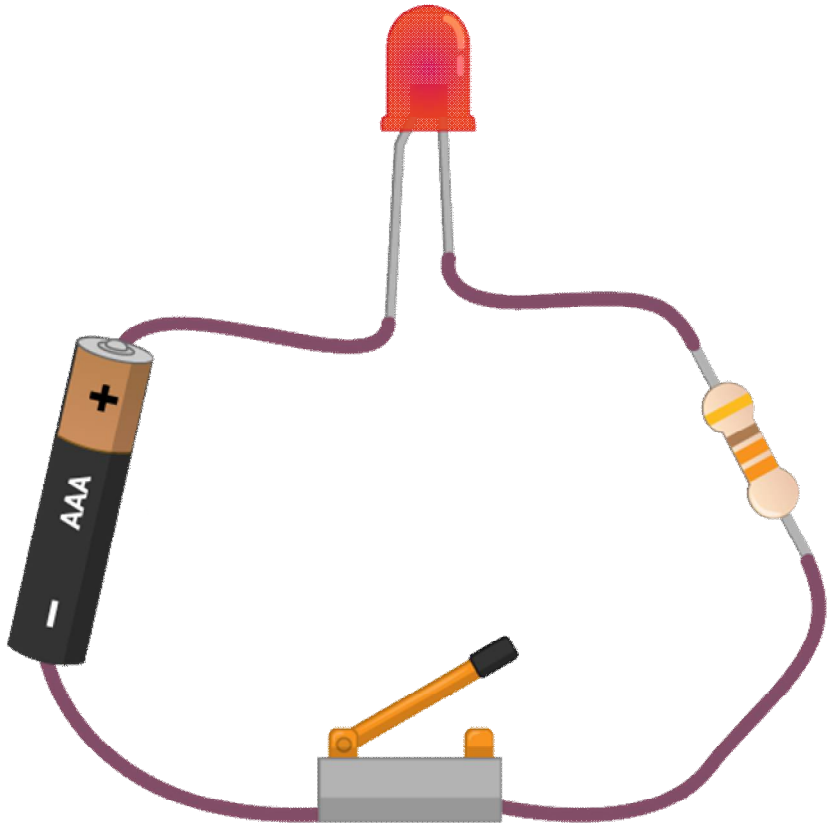
Plugging in your Raspberry Pi

- Begin by placing your **SD card** into the SD card slot on the Raspberry Pi. It will only fit one way.
- Next, plug your **keyboard** and **mouse** into the USB ports on the Raspberry Pi.
- Connect your **HDMI cable** from your Raspberry Pi to your monitor or TV.
- When you're happy that you have plugged all the cables and SD card in correctly, connect the **micro USB power supply**. This action will turn on and boot your Raspberry Pi.



CGS

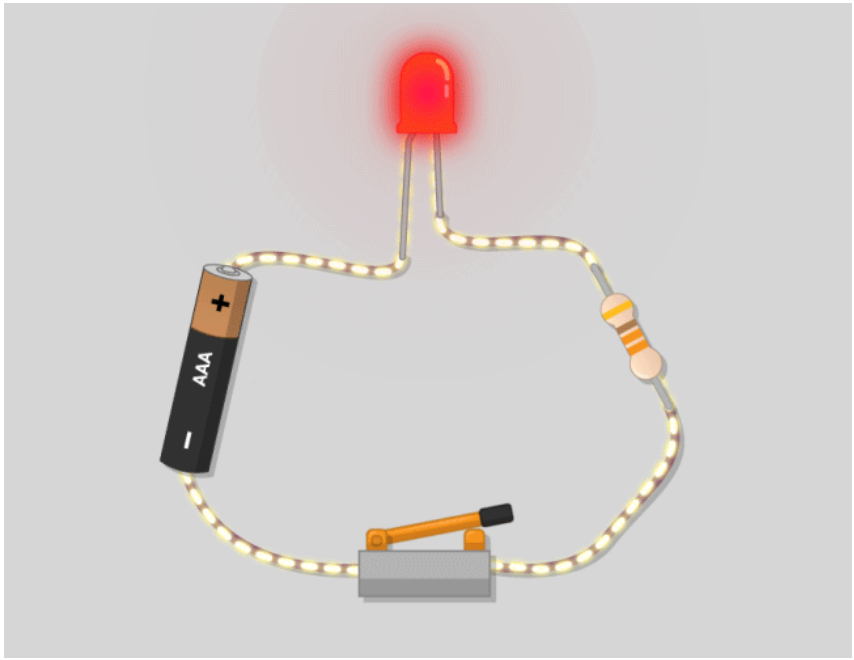
What is a simple circuit



CGS

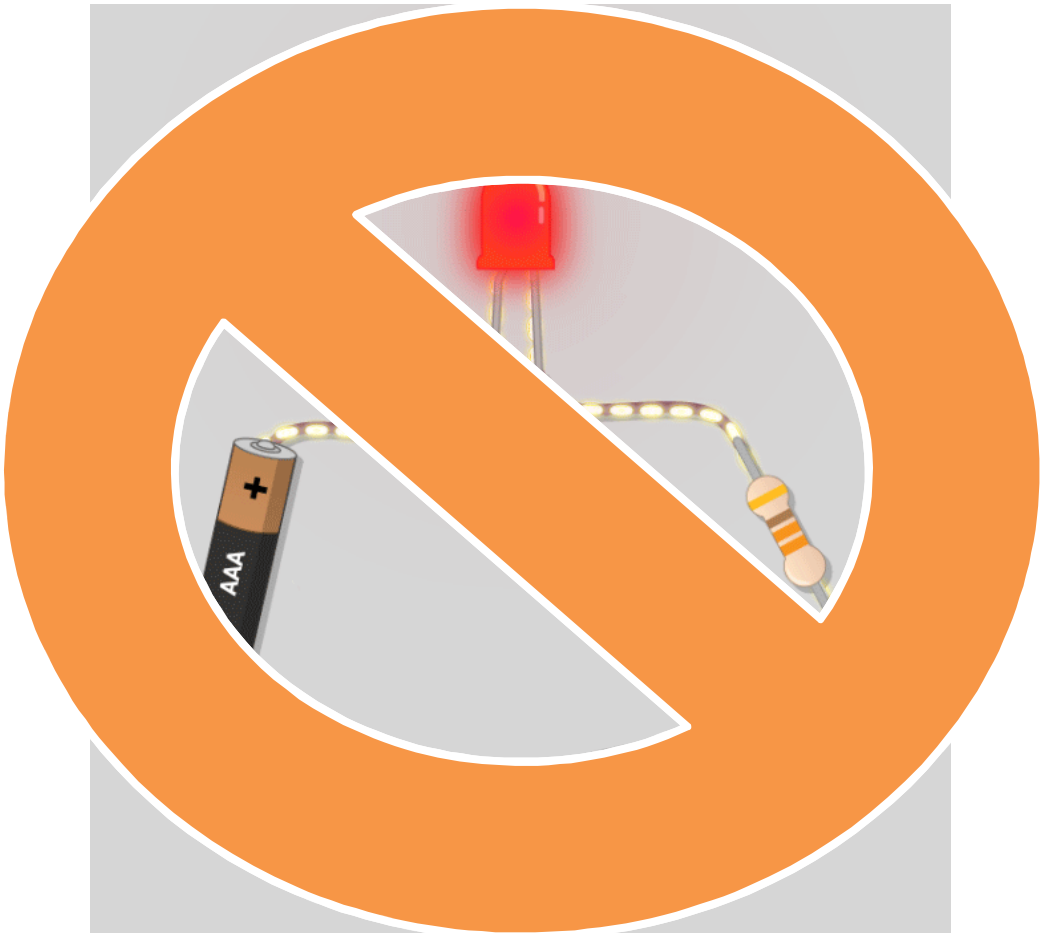
Some circuit theory

- The **cell** provides energy to the circuit in the form of electricity.
- The Light Emitting Diode (**LED**) is a type of **Output Component**. When current flows through the LED, it emits light.
- The **resistor** helps protect the **LED**. Resistors use up some of the energy from the cell, and therefore reduce the amount of energy that reaches the LED. Without the resistor, the LED could **burn out**.
- The **switch** acts as a break in the circuit.



CGS

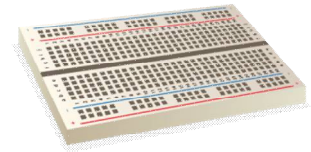
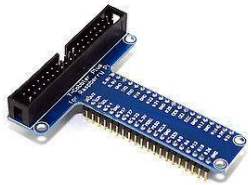
Be Careful!

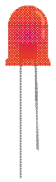
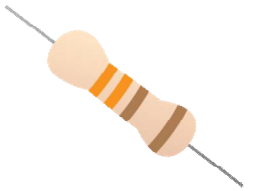
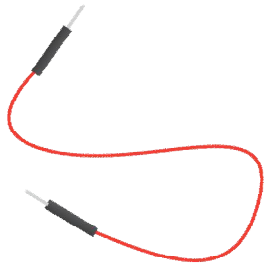


CGS

Building a simple circuit

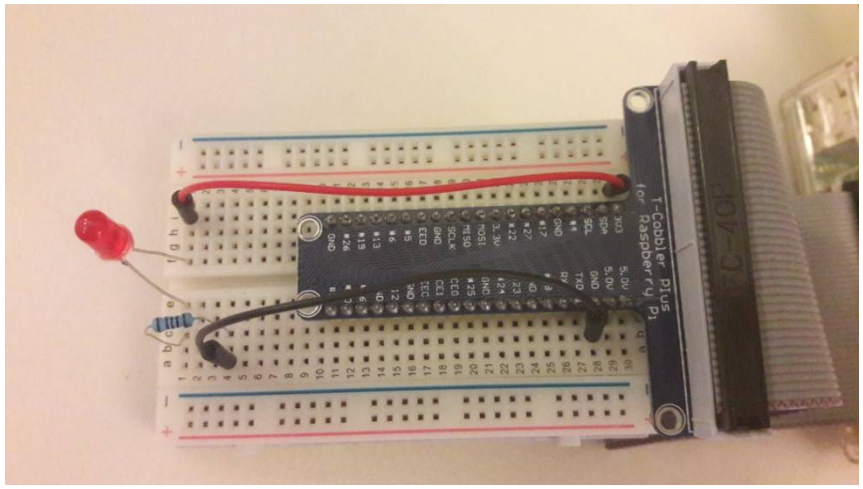
- To build the circuit you're going to need:
  - a breadboard
  - a T-Cobbler Plus
  - a GPIO Ribbon Cable
  - an LED
  - a 330 ohm resistor
  - two male-to-male jumper leads

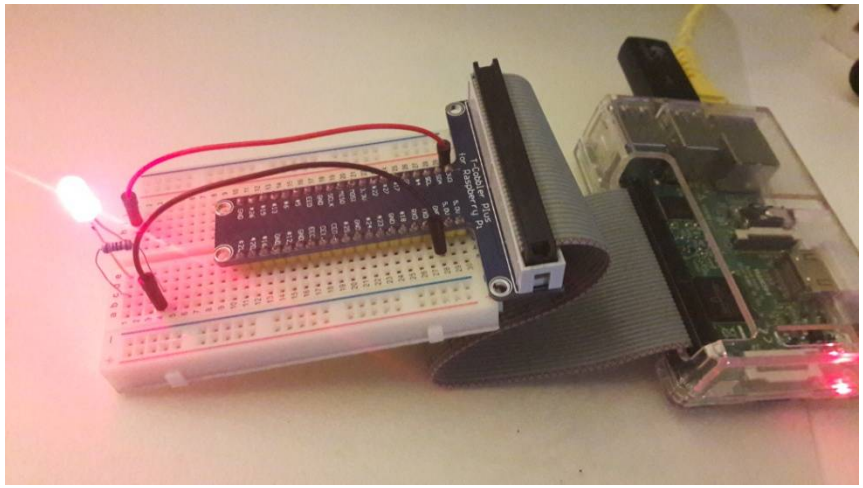



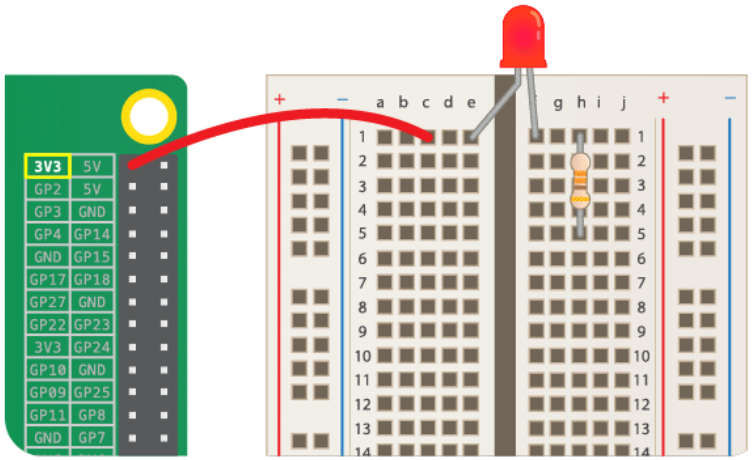





CGS

Your first simple circuit (1/2)

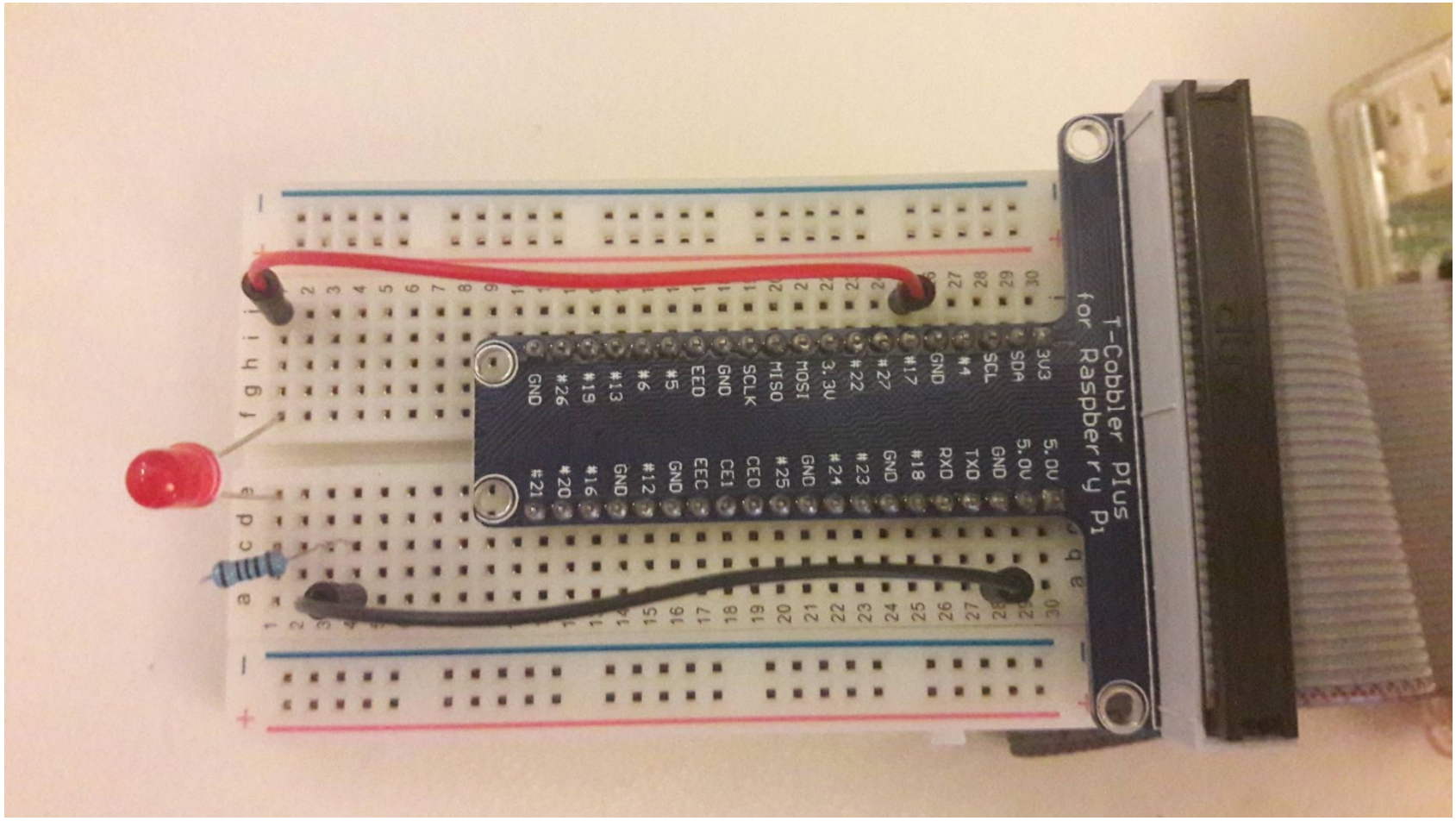






CGS

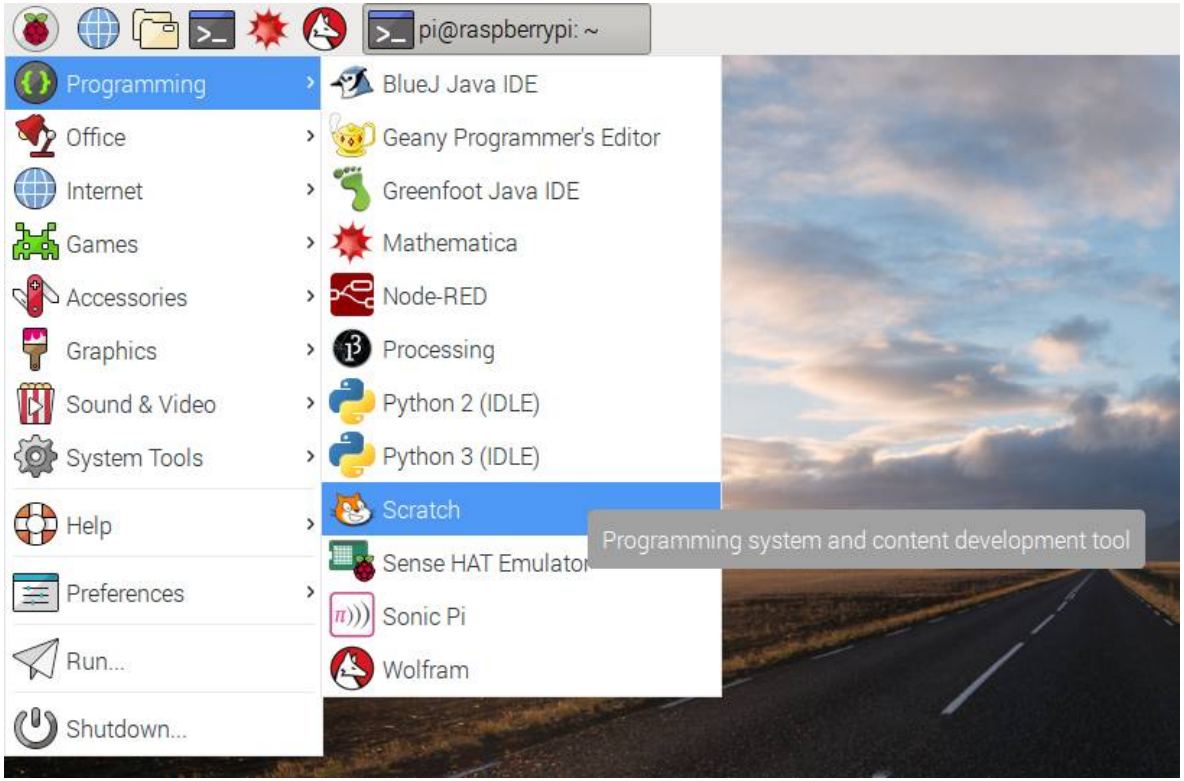
Your first simple circuit (2/2)



CGS

Constructing a Scratch program (1/5)

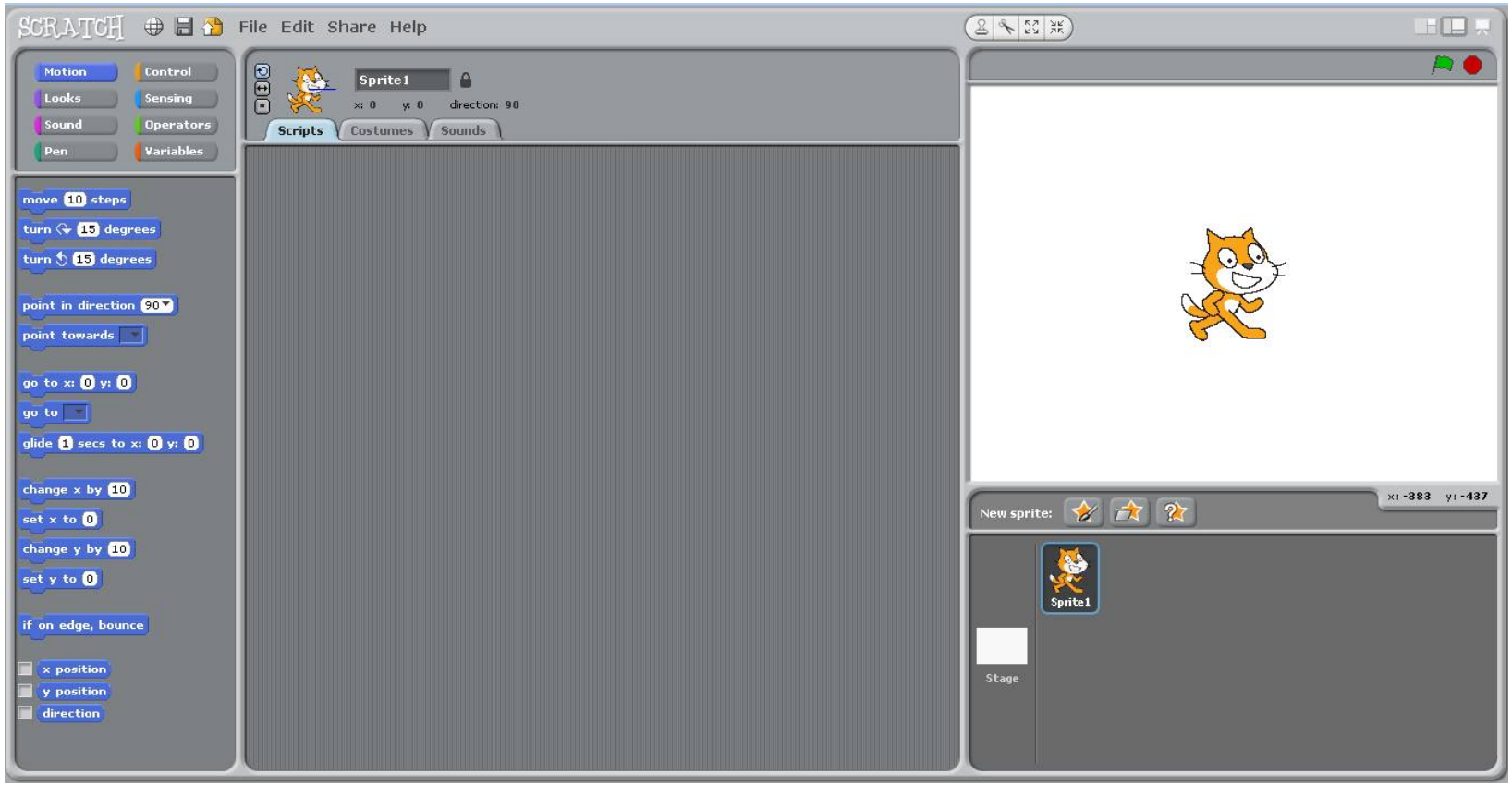
- Locate the Scratch program by clicking on Menu followed by Programming, and selecting Scratch.



CGS

Constructing a Scratch program
(2/5)

- The familiar Scratch interface will then load:



CGS

Constructing a Scratch program

(3/5)

- Click on Control in the top-left display. Drag the "**WHEN GREENFLAG CLICKED**" block onto the scripts area:

- Scratch uses broadcast blocks to communicate with the GPIO pins.
- The first broadcast you need is "**GPIOSERVERON**" which activates the GPIO functionality:

CGS

Constructing a Scratch program

(4/5)

- As your GPIO pin can be used as either input or output, you'll need to specify in which mode your pin is being used with the "**CONFIG17OUT**" broadcast:

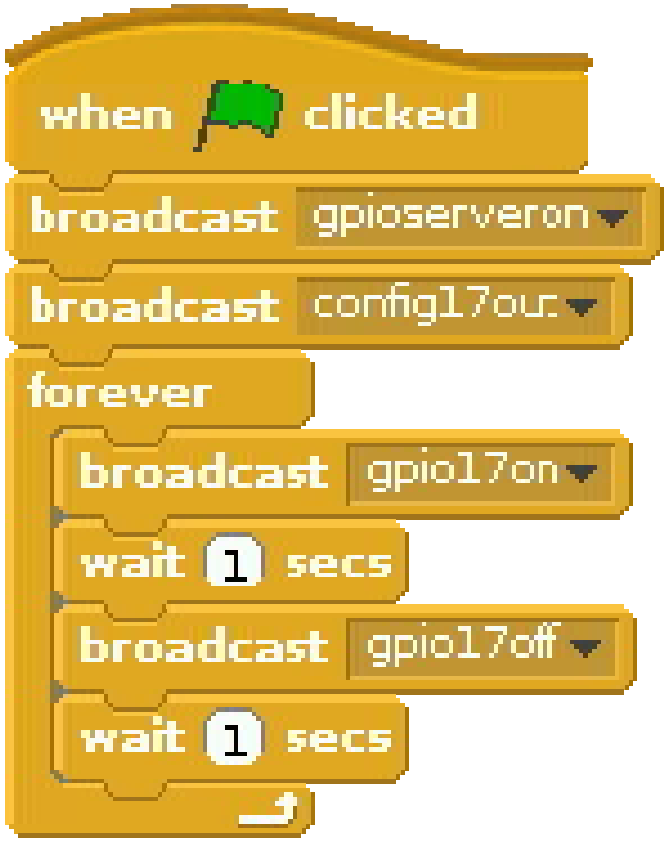
- From this point on, you can control your LED using two broadcasts: "**GPIO17ON**" to turn it on and "**GPIO17OFF**" to turn it off. Using these two messages and some pauses, you can make an LED flash continuously.

CGS

Constructing a Scratch program

(5/5)

- Finally, your Scratch code could be like this:

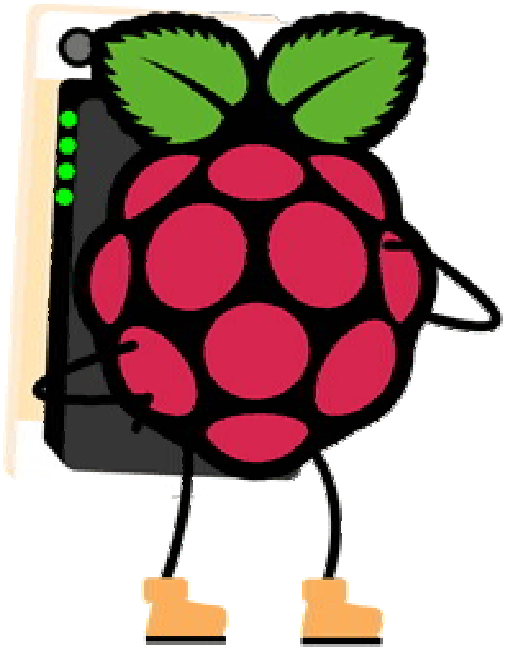


CGS

Task 1

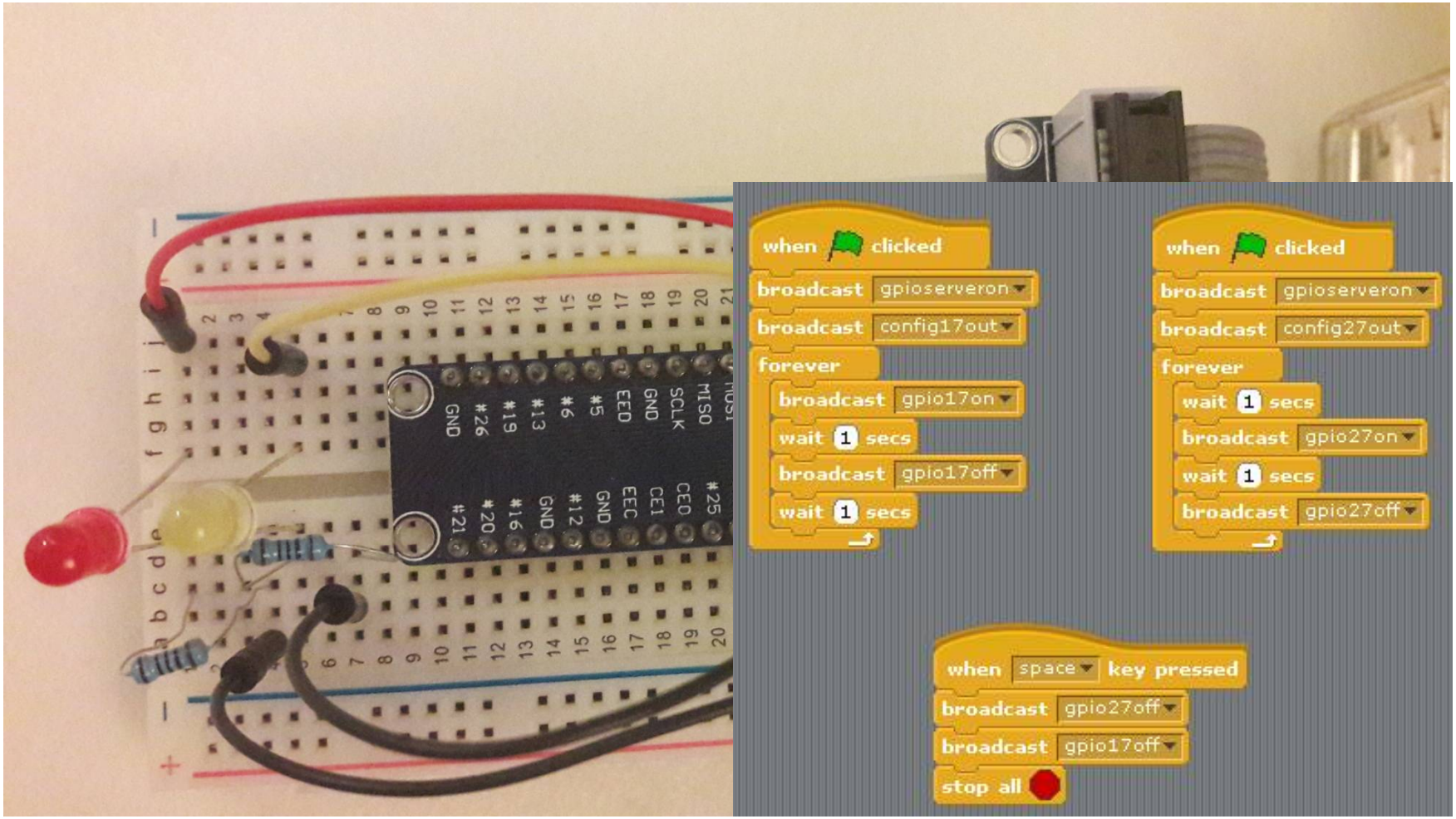
- Now, try to create a new Scratch code, which will allow the LED of your circuit to light up and down with a random pace.

- **TIP:** there could be a block on the Operators menu in the top-left display to help you with the random selection.



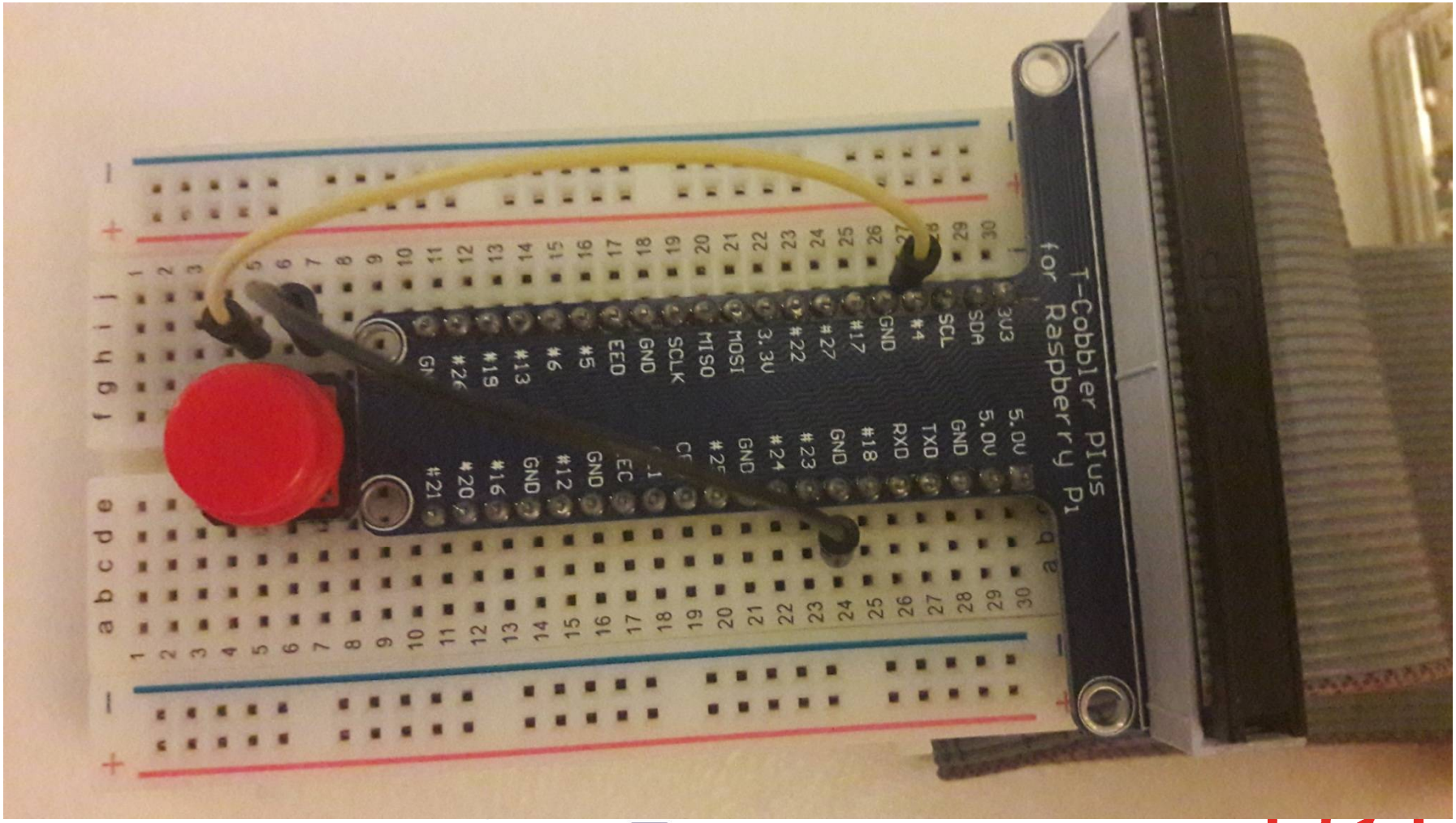
CGS

Task 2


when clicked
broadcast gpioserveron
broadcast config17out
forever
broadcast gpio17on
wait 1 secs
broadcast gpio17off
wait 1 secs
when clicked
broadcast gpioserveron
broadcast config27out
forever
wait 1 secs
broadcast gpio27on
wait 1 secs
broadcast gpio27off
when space key pressed
broadcast gpio27off
broadcast gpio17off
stop all

CGS

Wiring up a button



Configuring your button
(1/2)

- Before Scratch can react to your button, it needs to be told which pin is configured as an input pin.
- Assuming you have started a new Scratch file, you'll also need to start the GPIO server. The following code will configure pin 4 as an input:

- Once you have built the code above, you need to click the **green flag** in order for it to run and for your pin to be set up.

CGS

Configuring your button
(2/2)

- Next, you need to go to the **Sensing menu** in Scratch. From here you need to find the

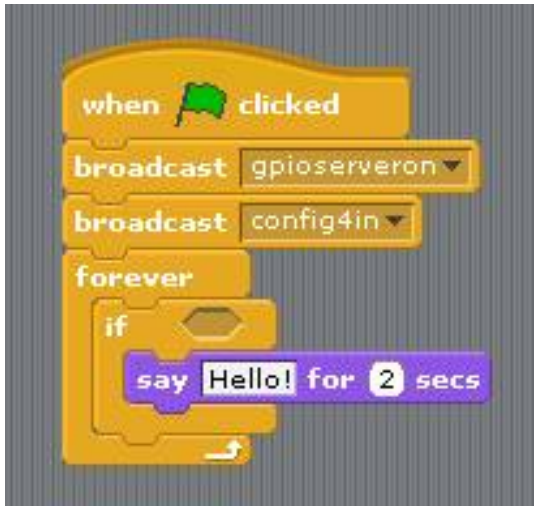
block and click the triangle to reveal a menu. Select **GPIO4** from the menu and click the tickbox to the left. You should now see the current state of the pin in the stage area:

- Now when you press your button, the state should change from 1 to 0

CGS

Responding to a button press
(1/2)

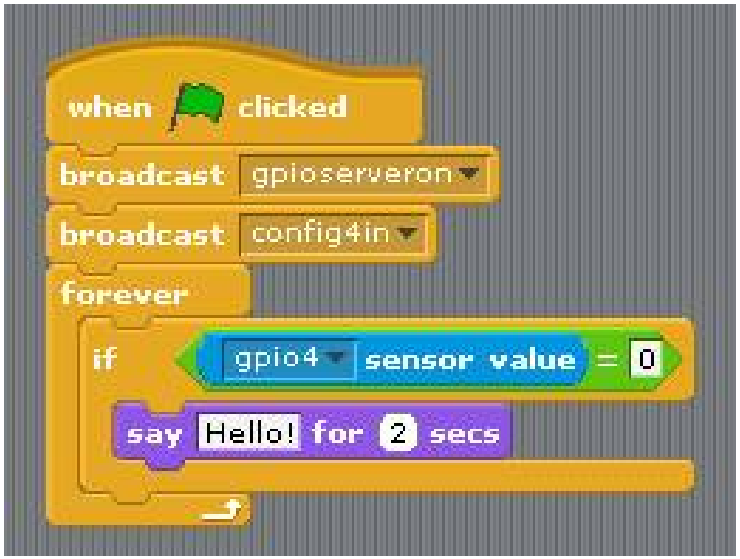
- Now that your button is all set up and working, you can make it do something. You can start off by making it control a sprite.
- Begin with a **FOREVER LOOP** with an **IF BLOCK** inside it. This will continually check the if condition and perform some action if the condition is met. The action in the example below will make the current sprite say "Hello!“:



CGS

Responding to a button press
(2/2)

- Finally, to make this work you need to add the condition, which is that we want the sprite to speak when the **button value = 0**:



- If everything is correct, your button should make the sprite speak.

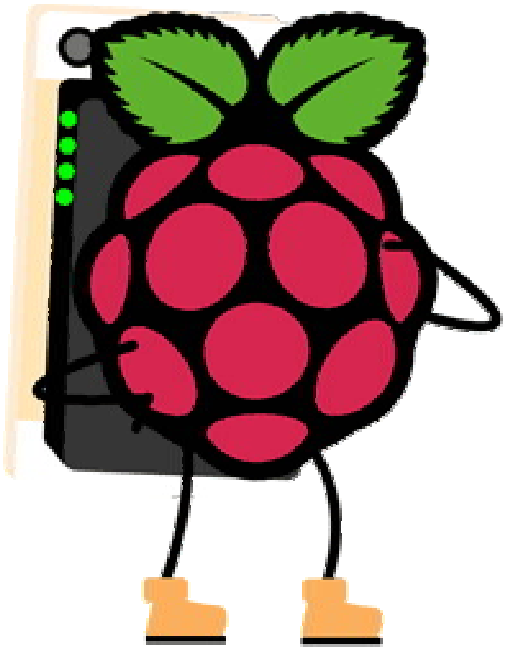
CGS

Task 3

- Now, try to create a new Scratch code, which will allow the sprite say:
  - You don't press the button!

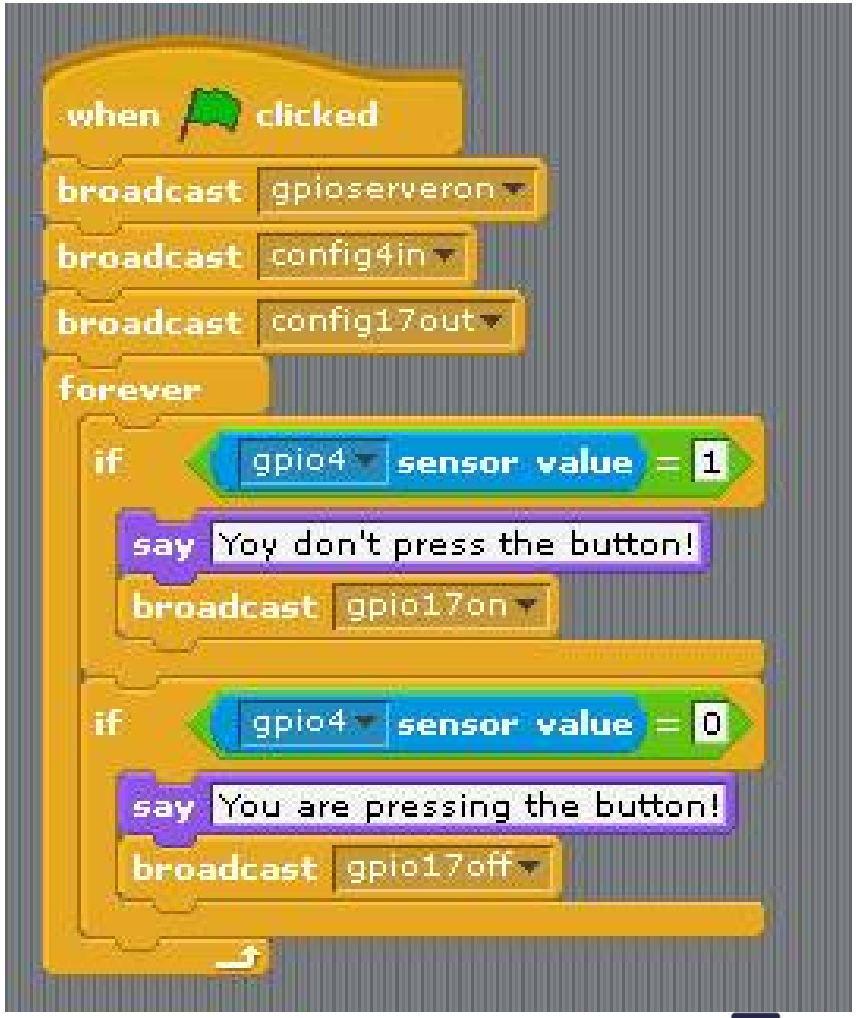
  when the button in not pressed and
  - You are pressing the button!

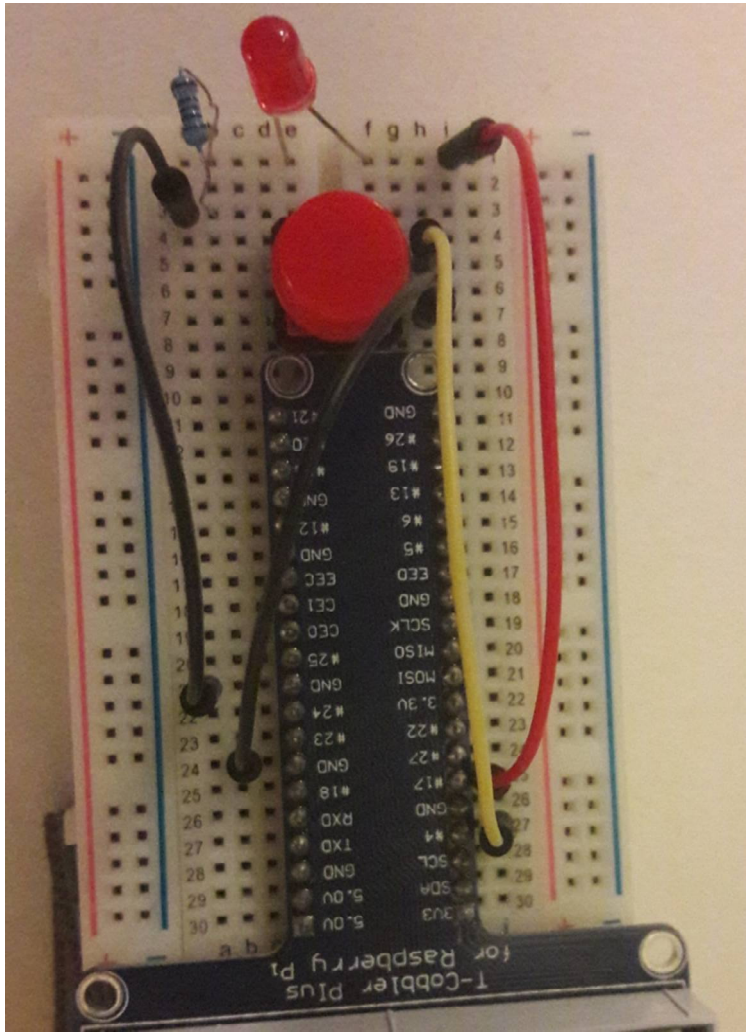
  when you press the button



CGS

Task 4


when clicked
broadcast gpioserveron
broadcast config4in
broadcast config17out
forever
if gpio4 sensor value = 1
say Yoy don't press the button!
broadcast gpio17on
if gpio4 sensor value = 0
say You are pressing the button!
broadcast gpio17off


T-Cobbler Plus
for Raspberry Pi

CGS