Raspberry Pi for Beginners

LESSON 2

MAKERHOUSE

EMPOWERING MAKERS

TABLE OF CONTENTS

LEARN TO CODE WITH RASPBERRY PI AND SCRATCH

1 INTRODUCTION

In the previous lesson, you have successfully set up and configured the Raspberry Pi. Congratulation! It is working like a charm with the Raspbian OS running on it.

So, what is next?

Let us explore more about the Raspberry Pi. Today we will learn the basics programming and electronics.

1.1 SCRATCH

Created by the scholar in Massachusetts Institute of Technology (MIT) targeting to enables beginners to start programming within minutes, without any prior knowledge. Scratch designed with graphical programming approach; drag and drop the various instructions and joining them like a jigsaw puzzle. Interesting right? Scratch included as standard software installed in the Raspbian OS, so no scratch head for it. Just open and start to explore. Moreover, Raspberry Pi GPIO can be used with Scratch to interact with electronics components, sensors, etc.

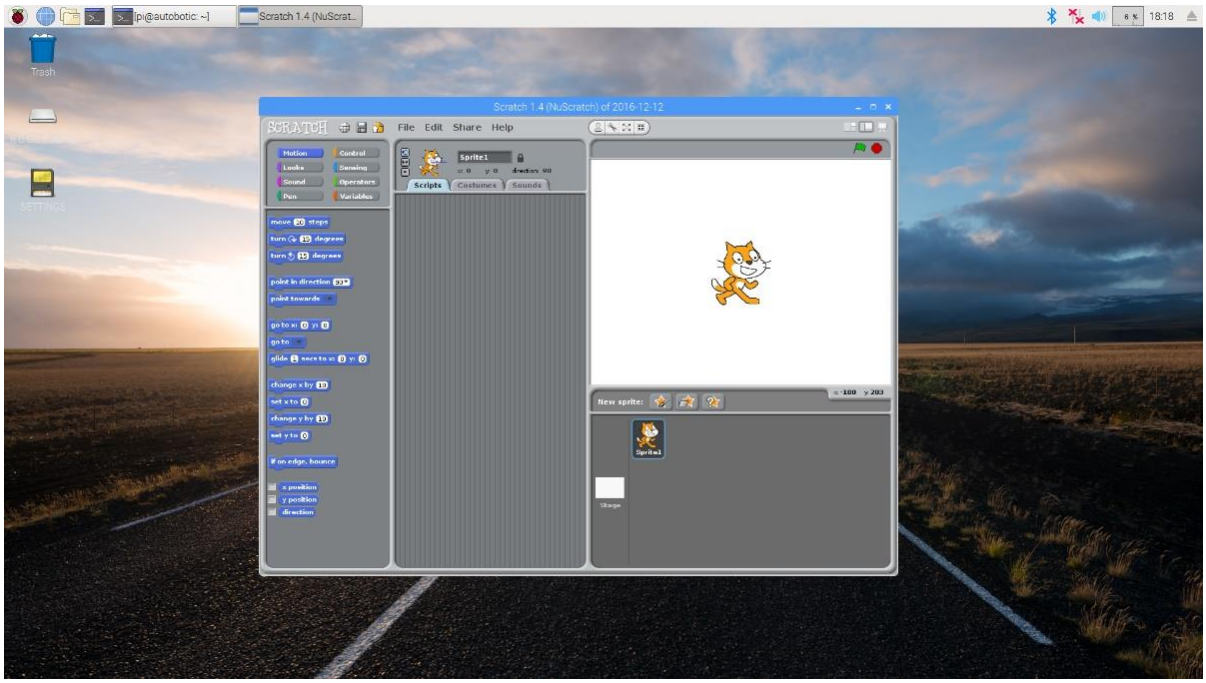


Figure 1: Scratch (1.4) Software on Raspbian OS.

SS

1.1.1 GETTING STARTED

There are two ways we can open the Scratch software:

1. Graphical User Interface (GUI)
2. Terminal

Open Scratch using GUI:

1. Click on the Main Menu – **Raspberry Pi icon** > **Programming > Scratch**.

Figure 2: Scratch open using GUI (Raspberry Pi icon – main menu)

2. Open **Scratch using Terminal**:
   1. Hit **ctrl+alt+t**. New Terminal will appear.
   2. On Terminal, type: **scratch**.

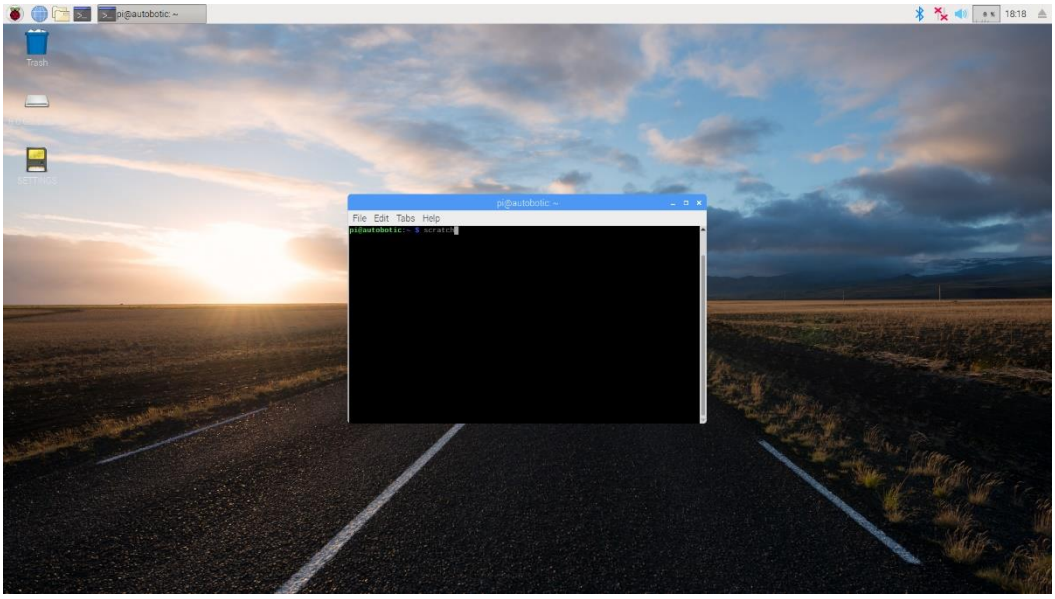


Figure 2: Scratch open using Terminal -- **scratch**

1.2 THE BASICS

The Scratch software interface consists of four basics pane as in

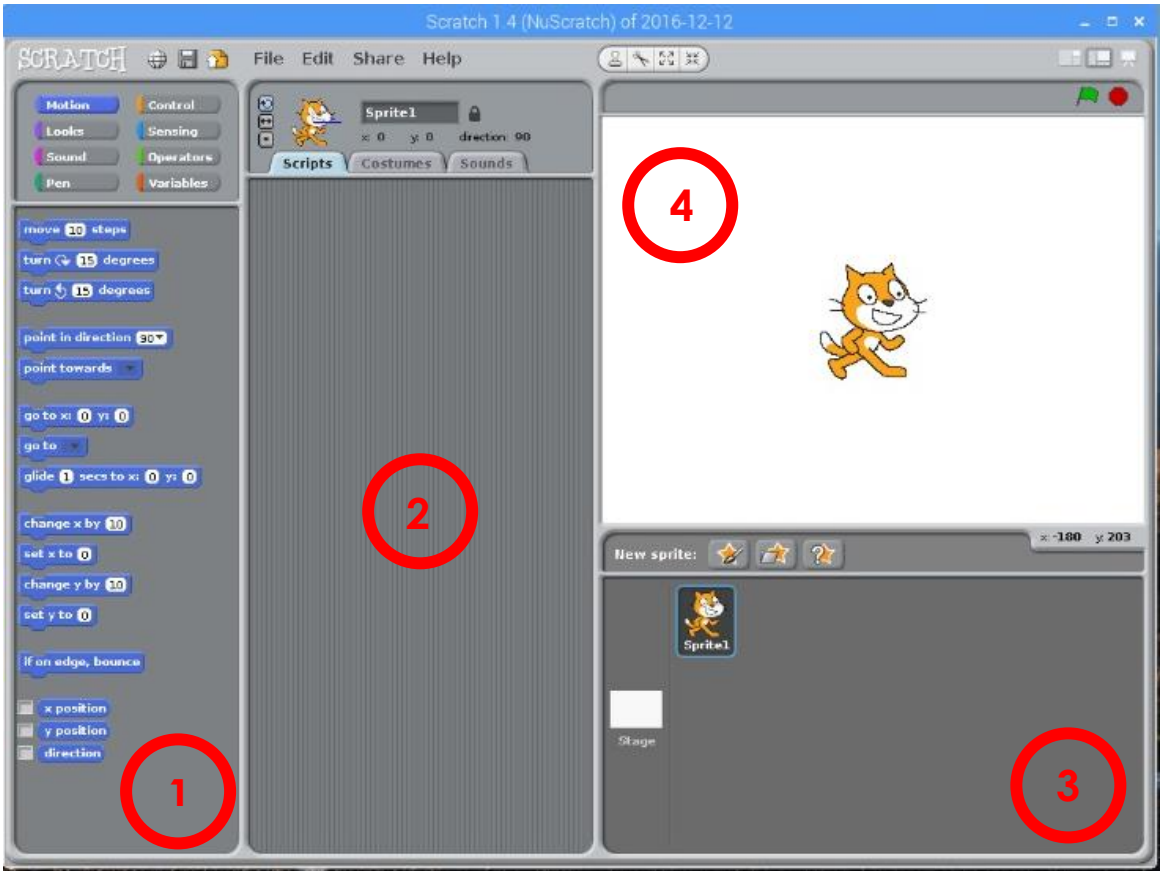


Figure 2: Scratch Software interfaces.

These four panes play a role in a different function whereas as

No	Name	Description
1	Block Palette	Set of instructions
2	Scripts/Costumes/Sound Palette	Programming area where instructions from the block palette are drag in and joining.
3	Sprite List	Controllable images
4	The Stage	Area for the sprites move and interact -- animate

Table 1: Palettes type and description

1.2.1 BLOCK PALETTE

Consist of eight different categories -- motion, look, sound, pen, control, sensing, operators, and variables. Each of the categories is color-coded representation; ease to identify – it will help children or beginners for sure. These categorizes of function are describes as

Block Type	Description
Motions	Used for moving sprites around the Stage
Looks	Used for animating sprites, giving them speech bubbles, and changing their size and appearance
Sound	Used for playing recordings or musical notes
Pen	Used to draw as a sprite moves around the Stage
Control	Used to describe what happens when, and for making bits of your program repeat.
Sensing	Used to test whether your sprite is touching another sprite or another color, or to get information about other sprites. You can also use the sensor value blocks in your own electronics projects on the Raspberry Pi.
Operators	Used for maths, random numbers, and doing things to text. There are also blocks here for combining the blocks used in decision making.
Variables	Used to remember information, such as scores, timer values, or player names.

Table 2: Command instruction of each type block descriptions

1.2.2 GET THINGS MOVING WITH SCRATCH

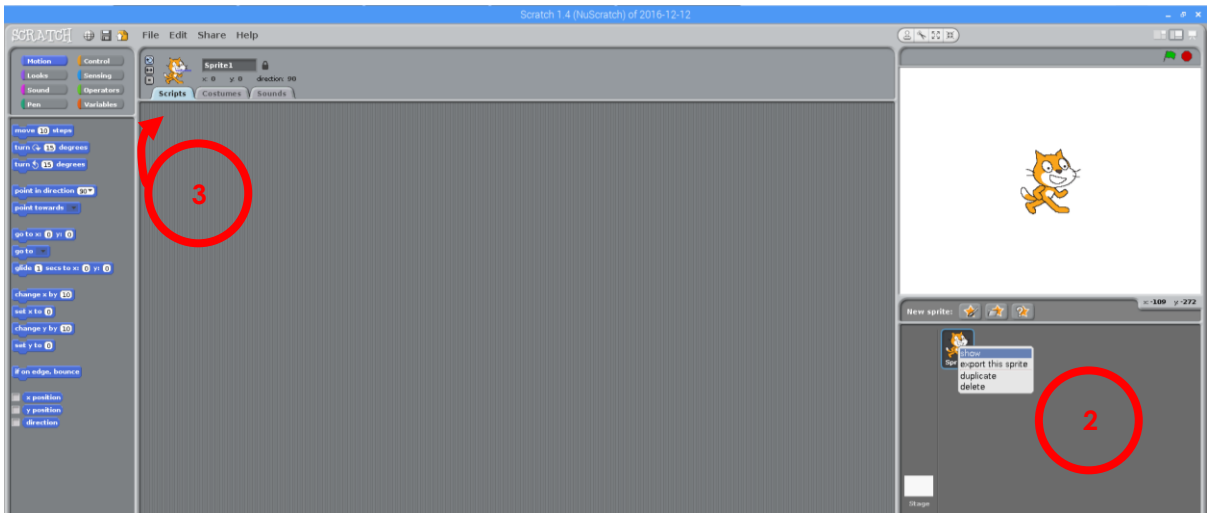
Too many words, and a lot of information.

Never mind, we will start now by practically apply the information given so that you can understand it more. We will start the basics; program to move the Scratch cat around the screen using by detecting the key pressed. Here he is, the Scratch cat

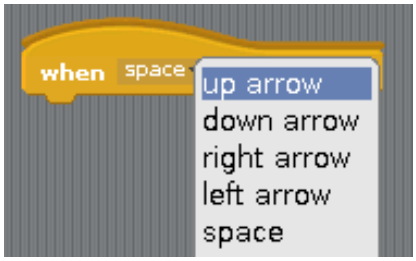
Figure 3: ***Who is Scratch cat?*** One of sprite type available on the Scratch which every new Scratch project will automatically add him.

Ok let us start.

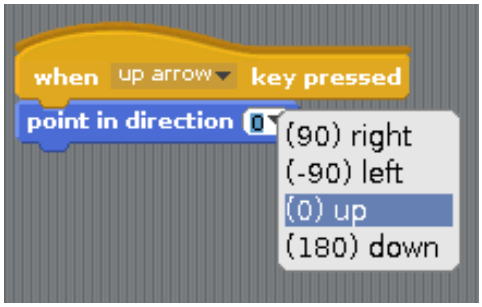
1. **Open Scratch** software.
2. Right-click on Scratch cat sprite select show. It will direct us to the scripts palette where we will program the scratch cat according to key pressed.



3. Select Scripts.
4. Go to **block palette > control**. Drag in **when space key pressed** under control block palette onto the script's palette. Click on the arrow key and change **space to up-arrow.**



5. Go to **block palette > motion**. Drag in **point in direction** block. Change the **direction to 0**.

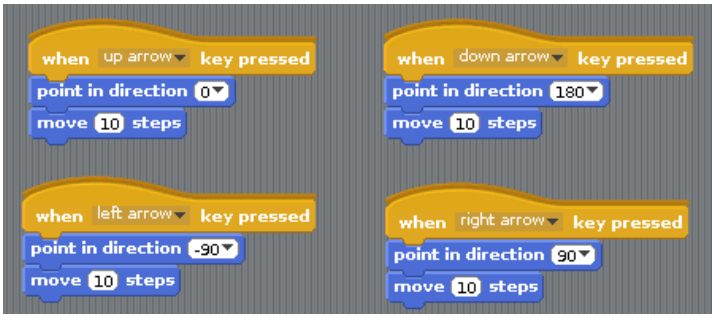


6. Go to **block palette > motion** again. Drag in **move 10 steps**.
7. Try hit up-arrow key. What you observed?

Repeat the step 4 to 6 to achieve the program objective which is to move the Scratch cat around the screen according to the key pressed. Will you succeed?

Hint: Repeat step 3 to 5 for down, left, and right-arrow key pressed.

Final Scratch cat program will be look like this



2 INTERACT WITH RASPBERRY PI GPIO

Raspberry Pi have 40 pins of GPIO. Can we use it with Scratch?

Yes. Scratch can even be used with the Pi's GPIO (general purpose input output) pins to interact with electronic components and sensors -- physical computing. Remember Raspberry Pi GPIO? Here it is

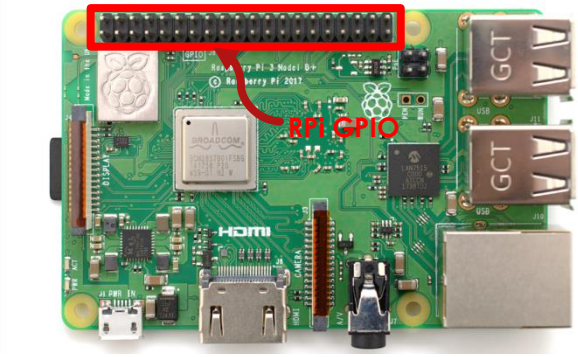


Figure 4: 40 pins of GPIO available on Raspberry Pi 3.

To understand how Scratch can be used with the Raspberry Pi GPIO pins for physical computing, we will conduct three basics project which is:

1. LED blinking (OUTPUT)
2. Button-pressed detection (INPUT)
3. LED controlled Button (INPUT-OUTPUT)

Recommended to use T-Cobbler Breakout + 40 Pin GPIO Cable for Raspberry Pi. It is helping a lot; since Raspberry Pi GPIO on board does not have printed labelling.

Figure 5: T-Cobbler breakout board

2.1 LED CONTROL WITH SCRATCH

We will use several parts for these projects. The part list is

No	Item	Unit
1	Breadboard	1
2	LED	1
3	Tactile-pushed button	1
4	330 ohm resistor	1
5	Male-to-male jumper wires	

Table 3: Required parts.

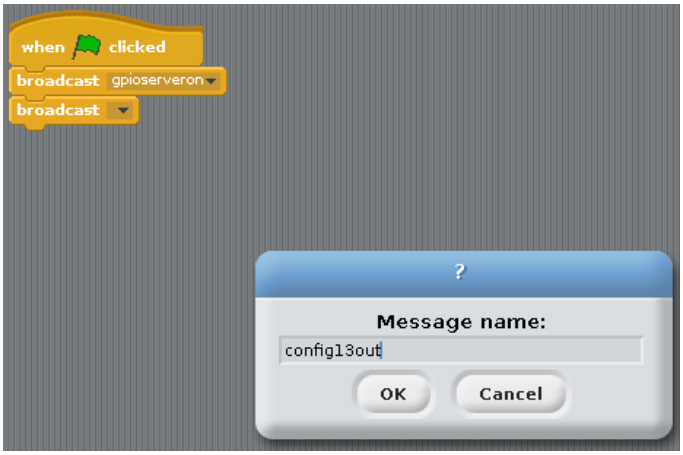
2.1.1 LED BLINKING (OUTPUT)

Software section:

1. **Open Scratch** software.
2. Go to **block palette > control**. Drag in **when green flag clicked** onto scripts palette.

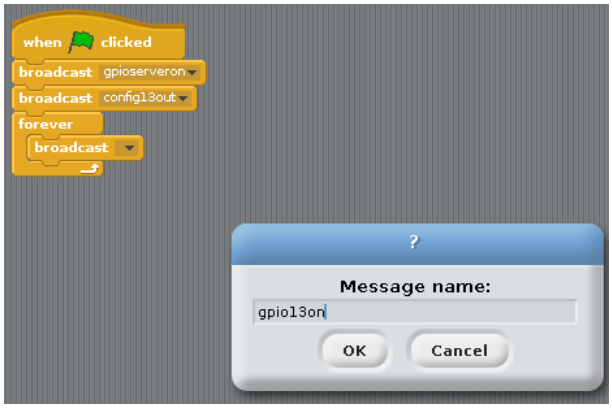
3. Go to **block palette > control.** Drag in **broadcast**. Click on dropdown selection select new/edit. Type **gpioserveron** – in order to use Raspberry Pi GPIO.

4. Repeat step 3, but this time type **config13out** – configure GPIO13 as an output.



5. Go to **block palette > control.** Drag in **forever** command block (repeat again and again -- loops).

6. Repeat step 4, and type **gpio13on** – GPIO logic **HIGH**. Put it under forever command block.



7. Go to **block palette > control.** Drag in **wait 1 secs** – delay for a one seconds. You are free to set how long the delay.

8. Repeat step 6 to 7. Change to **gpio13off** – GPIO logic **LOW**.
9. Done.

Before making the connection, recommended to turn of the Raspberry Pi to avoid mis-connection that probably will later be damaging our Raspberry Pi.

For hardware and electronics connection as

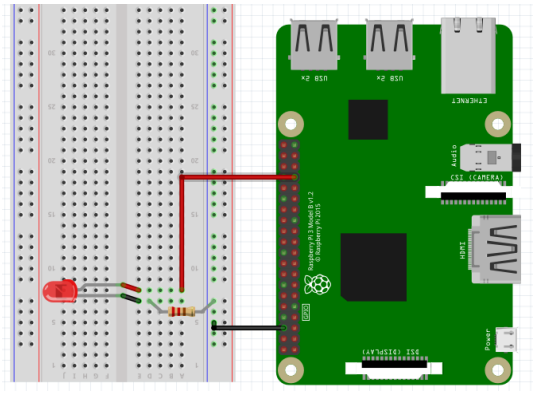
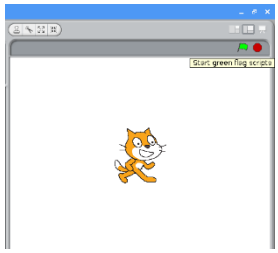


Figure 6: LED connection with GPIO 13.

There are more GPIO can freely be use – labelled as GPIOxx (refer to the T-Cobbler breakout board) – xx is the number. Feel free to try it yourself.

Save your project now by go to **File > Save**. Next, run the code by click the green flag on top of Stage area. What will you observe?



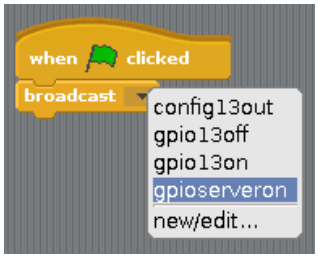
SSSS

2.1.2 BUTTON READING (INPUT)

Software section:

1. **Open Scratch** software – you can continue with previous opened project.
2. Go to **block palette > control**. Drag in **when green flag clicked** onto scripts palette.

3. Go to **block palette > control.** Drag in **broadcast**. Click on dropdown selection select new/edit. Type **gpioserveron** – in order to use Raspberry Pi GPIO.



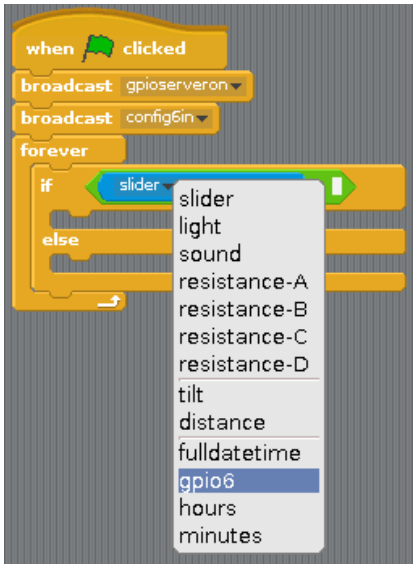
4. Repeat step 3, but this time type **config6** – configure GPIO6 as an input.

5. Go to **block palette > control.** Drag in **forever** command block (repeat again and again -- loops).

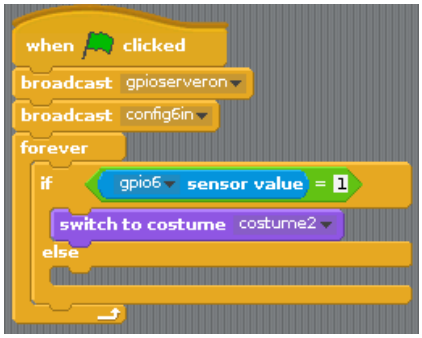
6. Go to **block palette > control.** Drag in **if-else** command block.

7. Go to **block palette > operators.** Drag in **equal** command block.

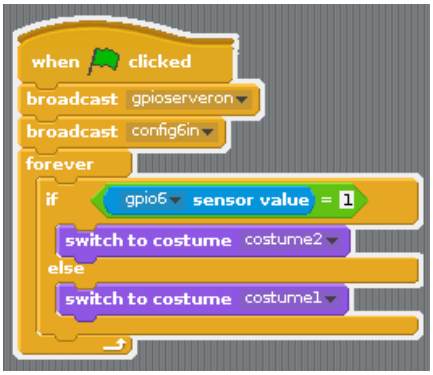
8. Go to **block palette > sensing.** Drag in **slider sensor value** command block. Change the slider to gpio6 and for the **equal** set to **1**.



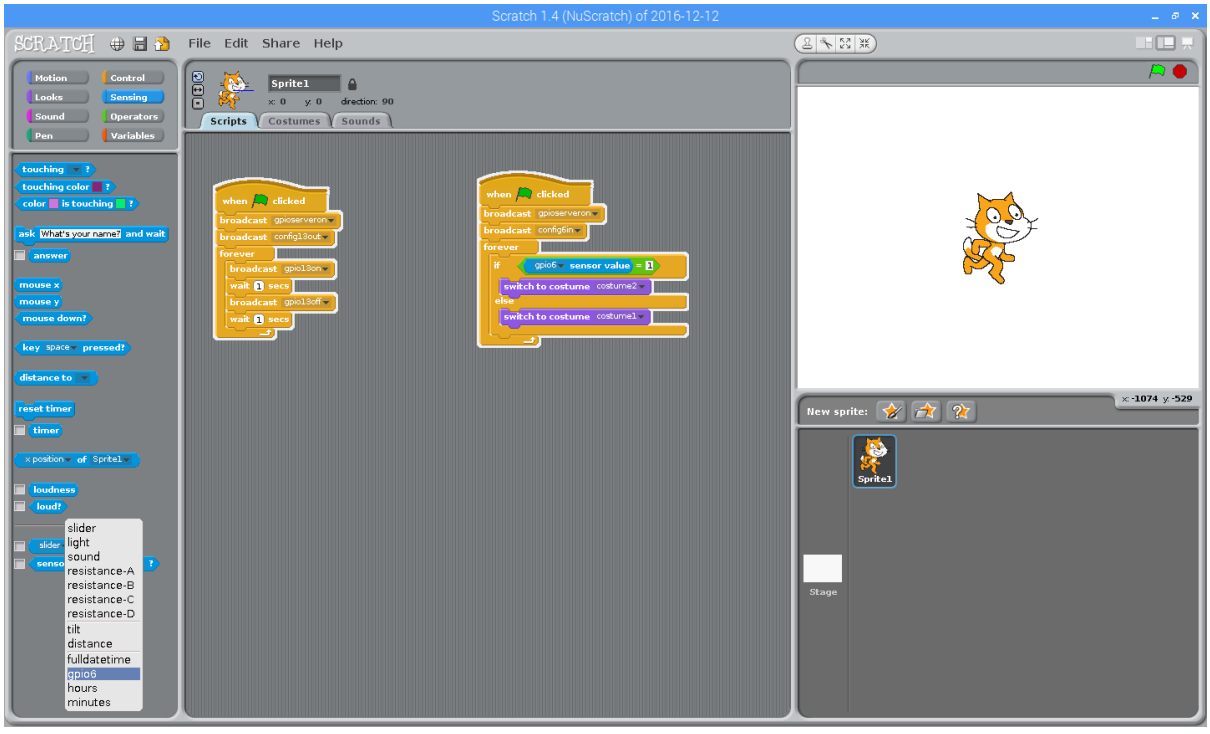
9. Go to **block palette > looks.** Drag in **switch to costume** command block. Set it **costume2**.



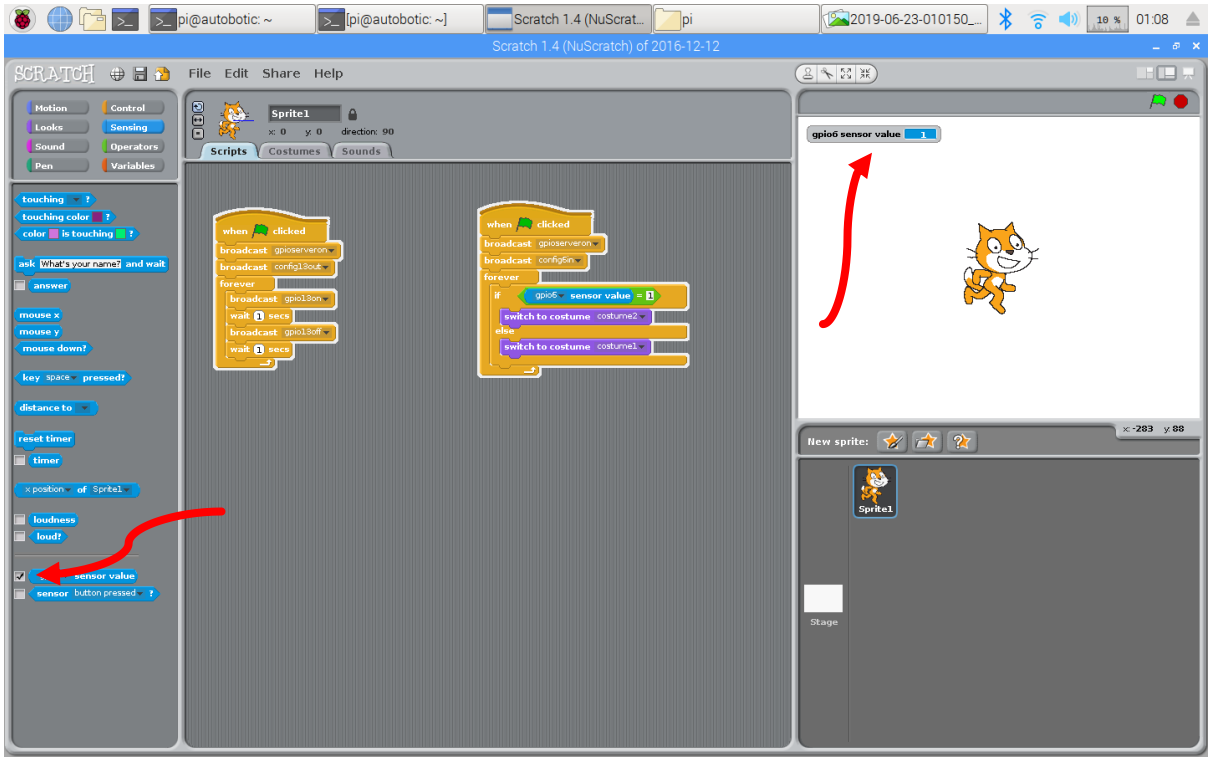
10. Repeat step 9. Change to **costume1**.



11. Go to **block palette > sensing.** Change the **slider** sensor value to **gpio6** sensor value.



12. Tick the **gpio6 sensor value**. There will gpio6 sensor value display on stage.



13. Done.

Before making the connection, recommended to turn of the Raspberry Pi to avoid mis-connection that probably will later be damaging our Raspberry Pi.

For hardware and electronics connection as

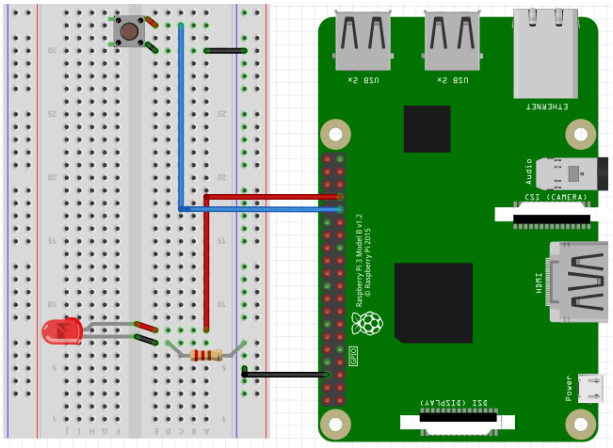


Figure 7: An addition of button to GPIO 6.

There are more GPIO can freely be use – labelled as GPIOxx (refer to the T-Cobbler breakout board) – xx is the number. Feel free to try it yourself.

Save your project now by go to **File > Save**. Next, run the code by click the green flag on top of Stage area and then push the button. What will you observe on the reading and also the Scratch cat?

2.1.3 LED CONTROLLED BUTTON

Congratulation my fellow friends. You managed to use Scratch to interact with Raspberry Pi GPIO – input and output -- separately. Now try to combine it. Once the button pressed, the LED will light up (ON/HIGH) and vice versa.