Using the Grove Button

Scratch Version

April 23, 2017

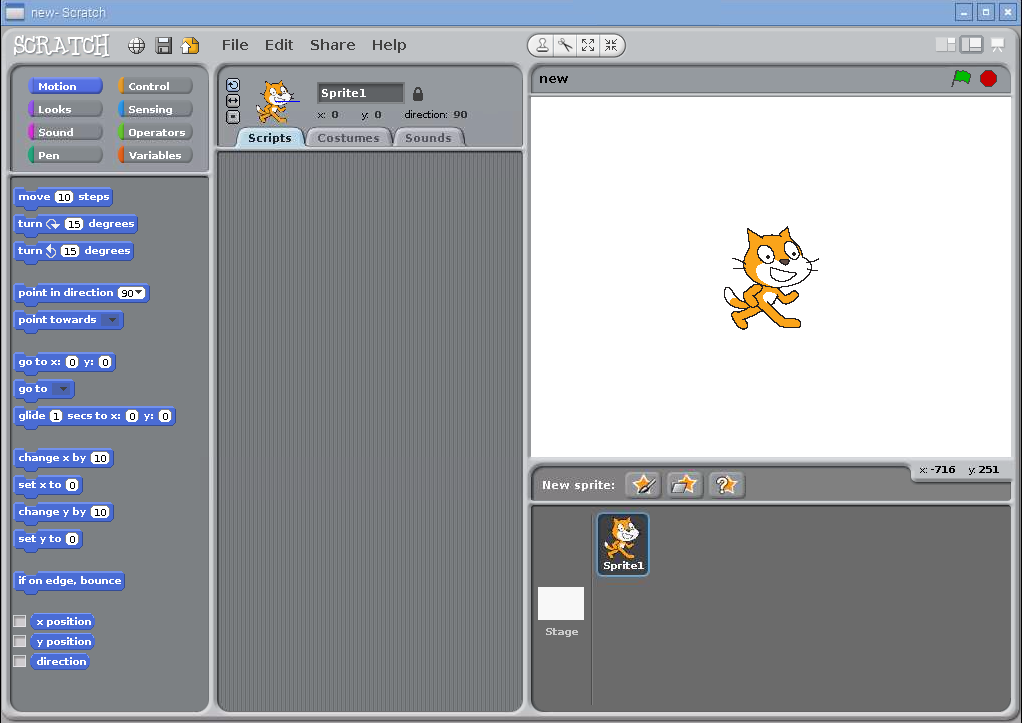


# Overview

The Grove Button is exactly what it sounds like. Use it to trigger actions in your Scratch programs.

# Setting up

Power up your Raspberry Pi/GrovePi and start up Scratch. If you haven’t done this before, work through the “Getting started with Grove Sensors” document. Your screen should look like this:



Now, find the Grove Button and cable in the kit. Connect the button to any of the digital ports on the Grove Pi. In this example, we connected it to port D7. If you connect it to a different port, be sure to replace the number 7 in the scripts below with the number of your port.

# Reading the button

You must ask the button for whether it’s pressed or not. It won’t report it on its own. The way to do this is to broadcast a message with the word “button” and the pin number to it. Try it out by placing a “broadcast” block:

Screen Clipping

Click on the black triangle to change the message and type in “button7”:



Click OK. Double click on the broadcast block so that Scratch knows that you’re interested in button 7.

The next step is to add a sensor block to your script that reads the button’s state. Go to “Sensing” and add the “slider sensor value” block at the bottom. Once you add it, you should be able to click on the black triangle to change what is being sensed. Here’s what you should see:



If you don’t see “button7”, try double clicking on the broadcast block again.

Select “button7” so that your sensor value block looks like this:

Screen Clipping

Now it’s time to try out the button. Build this script:



Run it and watch Scratch Cat tell you whether the button is pressed. The button is what’s called a digital sensor, so it reports in 0s and 1s rather than English words. You should see that the button is normally 0 and that when you press it, it is 1.

# Improving the button

The method used above to get the button’s state uses a programming technique called polling. Polling means that the code asks for the current button state repeatedly in a loop. In Scratch, you’re probably more accustomed to event driven programs. The following block is example of this:

Screen Clipping

The way this block works is when Scratch detects that the “a” key on the keyboard is pressed, it runs all blocks attached to this one. Your program doesn’t have to ask the keyboard repeatedly if keys are pressed down or up.

You can make Grove buttons work in a similar way by using the broadcast block. Here’s an example of what this would look like:

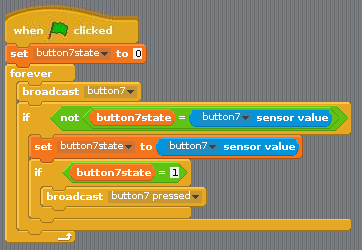


In other words, when the “button7 pressed” message is broadcast, run the following blocks. In this case, the program changes the color effect of the sprite every time someone presses the button.

Think about how you’d write the code that sends the “button7 pressed” message.

**Don’t go to the next page if you want to figure out the answer yourself!**

Here’s one way of writing the code to send events when the button is pressed.



It works by checking whether the sensor value for the button changes. If it changes, it saves the new value to the button7state variable and if the state is now 1, it broadcasts the “button7 pressed” event.