Thymio Scratch interface tutorial

1 Introduction

The Thymio Scratch interface allows scratch to communicate with the Thymio robot so that commands and data can be exchanged. The interface can only be run on a Linux machine, so far the interface has been tested on Ubuntu 14.0.4 and Raspbian on the Raspberry Pi. The interface itself and Scratch do not need to be run on the same machine as long as both computers are on the same network.
This tutorial will cover setting up the interface and how to use Scratch to control the Thymio.

2 Linux notes

If you have never used the Linux terminal before then here are a list of common commands as well as useful commands. Please note that the terminal is case sensitive.
sudo: This commands runs the command as the root user. It is similar to the "run as administrator" option in windows. It is required for most commands which can make some modification to the computer and may require you to enter your password.
apt-get: This is the software manager for Ubuntu and Debian (Raspbian) distributions of Linux. It deals with installing, removing and managing software.
cd: This is used to change the directory that the terminal will use and is the same as in the windows console. This command is used with an address after it.
ls –a/ dir: These commands will list the current contents of the directory. Ls –a will include hidden files as well.

3 Fast Setup

There is a bash script which will install everything for you. If you want to learn about the UNIX terminal then you might want to follow the slow setup, otherwise this will do everything for you.
Firstly you need to download the actual program which will do the work. Go to the following link and click "download zip".
https://github.com/lazerduck/Thymio_python_interface
Now extract the files to somewhere convenient, for this example I will be using the desktop

Open the terminal with "ctrl + alt + T" and in the window you will need to navigate to the folder where you extracted the interface. In this example I use the desktop, refer to the Linux notes section for the commands for navigating using the terminal.

```
cd /home/<your_username>/Desktop/Thymio_python_interface-master
```

Now we need to ensure the file has the permissions to run, enter the following.

```
sudo chmod +x Install.sh
```

With that done you need to run the script, enter the following .

```
./Install.sh
```

And now everything has been install, move on to section 5.

4 Slow Setup

To use the interface you need to install some programs which the interface uses. You will need an internet connection and you will need to open the terminal. In most distributions of Linux this can be done with “ctrl + alt + T” otherwise it will likely be in the start menu.

4.1 Install Asebamedulla

Aseba medulla is the software used to allow the interface to communicate with the Thymio. First, you will need to add the location of the program before you can download it and install it. Type the following in the terminal.

```
sudo add-apt-repository ppa:stephane.magnenat\`lsb_release -c -s`
```

After the repository has been added you will need to update the apt-get so its list include the new location.

```
sudo apt-get update
```

Now you can install the Aseba package which will install Asebamedulla.

```
sudo apt-get install Aseba
```

With this done, you can check that the package is installed correctly by typing the following.

```
sudo asebamedulla
```

If you see no errors then everything has worked and you can press “ctrl + C” to stop Asebamedulla from running so we can continue with installing the interface.

4.2 Install the Python Module

The interface is written in Python and requires a module called “scratchpy” to communicate with scratch. To install this module we will use a python package installer. Enter the following code in the terminal

```
sudo apt-get install python-pip
```

Now we can use Python-pip to install the module. Python-pip is used similar to apt-get. To install the module type the following.

```
sudo pip install scratchpy
```

With this done, the interface is almost ready to work.

4.3 Download the Interface

Now you need to download the actual program which will do the work. Go to the following link and click “download zip”.
https://github.com/lazerduck/Thymio_python_interface
Now extract the files to somewhere convenient, for this example I will be using the desktop.

4.4 Install Scratch

The interface uses scratch 1.4 and uses its remote connections. To install scratch enter the following in the terminal

```
sudo apt-get install scratch
```

5 Running the Interface

You are now ready to set up the interface. This will describe how to setup the interface tethered to the computer. For remote connections you will need to connect the interface to Scratch on the other computer, section 4.x will cover this in greater detail.

5.1 Run the Required Programs

For the interface to communicate with Scratch and the Thymio they will need to be running. Connect the Thymio via USB to the computer and enter the following command to run Asebamedulla and make it connect to the Thymio.

```
sudo asebamedulla "ser:name=Thymio-II"
```

If this doesn't work, try it without the speech marks.

Now the terminal should say that it has connected to the Thymio. If this is not the case, press "ctrl + C" to stop Asebamedulla from running and try again.

Open Scratch, it should appear in the start menu or on the desktop. Use scratch to open the Scratch file in the previously extracted files as mentioned in 3.3. It should be called "Thymio-test.sb". The Scratch code should now load and there should be a box saying "remote sensor connections enabled", press okay. There will be code on the cat sprite and on the stage. The code on the cat is example code and the code in the stage is the back end code that performs some simple actions to ensure sensor data is available.

5.2 Run the Interface

You are now ready to run the interface. Open another terminal window as the current one is still running Asebamedulla. In the new terminal window you will need to navigate to the folder where you extracted the interface. In this example I use the desktop, refer to the Linux notes section for the commands for navigating using the terminal.

```
cd /home/pi/Desktop/Thymio_python_interface-master
```

In this example, pi is the username and the desktop is where the folder is, you may need to change this to match your system. Now that the terminal is looking in the right folder you can run the interface. Enter the following in to the terminal

```
sudo python Thymio_Interface.py
```

It should now ask you what sensors you want to use, just say yes to all 3. Finally it will ask you for the address of the computer running Scratch. As we are running Scratch on the same machine you can just enter "localhost" without the quote marks. Localhost refers to the address of the current machine.

If you come across an error saying
"dbus.exceptions.DBusException: org.freedesktop.DBus.Error.ServiceUnknown: The name ch.epfl.mobots.Aseba was not provided by any .service files"

Then the interface couldn't find Asebamedulla. Switch to the terminal running Asebamedulla and stop the program and start it running again.
If you see a line saying "sensor update" then everything is working and you are ready to move on to the tasks.

5.3 Remote connections

To use another machine to run Scratch you will need to run Scratch on that machine and open the scratch file just like before. When running the interface instead of entering localhost as the address of scratch enter the local IP of the computer running Scratch.

A remote connection is intended so that a Raspberry Pi with a battery pack can be put on the Thymio and use Wi-Fi to allow you to control it from a distance and with no trailing wires.

6 Tasks

For reference, here is a list of all of the broadcasts that the interface can use: forward, backward, left, right, null, direct, command and arc. If a command is not available from the dropdown list then click on new and add it again.

Press "C" or broadcast command to enable the interface to use the commands for the next few tasks.

Key events have already been included. The arrow keys broadcast in their directions and the space bar broadcasts null. Try using the arrow keys to make the Thymio move. The scroll wheel will act as either up or down in Scratch and commands will queue so be careful not to accidentally queue lots of actions.

6.1 basic movement

Inside sprite one, use the broadcast tile to make the robot do the following

- Move forward
- Turn right
- Move forward
- Move backwards
- Turn left
- Move backwards

You will need to use the tile as pictured below

Note how all of commands are sent almost straight away and that they the stack up, being performed one after the other.

6.2 Movement conditions

Repeat the same task as before, this time however, us the set variable to change the duration and the speed. Duration must be positive and duration is in seconds and the speed must be between 500 and -500. After entering the value in the box, make sure to click on the box, a white outline should appear and will indicate it has been run.

Now set the duration to 0 and use the wait function to time broadcasts and the null broadcast to stop the Thymio.
Try to make the Thymio do the following:

- Move forward at speed 500 for 1 seconds
- Rotate left as speed 200 for 0.5 seconds
- Stop for 1 second
- Move backwards at speed -200 for 0.5 seconds
- Stop

Notice how moving backwards at a negative speed causes the Thymio to move forwards.

6.3 Arc Movement

The Thymio can also be programed to move in an arc. The arc command works differently to the others as it doesn't use the duration. Instead it uses a radius and length variable, both in centimetres to define the size of the radius of the arc and to define the distance that should be travelled. Set the radius to 10 and the length to 10 as well and then broadcast arc. If arc is not in the dropdown list on the broadcast tile then you will have to add it as a new message.

Notice how the wheels move at different speeds in order to make the Thymio move in an arc. The radius is from the centre of the pen hole and the distance to the wheels is 5cm, try setting the radius to 4 and -3.

The arc cannot have a radius of 0, if it did then the centre would never travel and so it would try to move forever.

6.4 Direct wheel control

You may have noticed the direct and command broadcast. So far we have been using the commands but you can control the wheels individually. Instead of broadcasting, set the LeftSpeed and RightSpeed variable to control the wheels. To enable this mode you will need to press "D" or send the command "direct".

Using these variables make the Thymio do the following.

- Left speed set to 0, right speed set to 400 for 0.7 seconds
- Left speed set to 500, right speed set to -500 for 1 second
- Left speed set to 100, right speed set to 100 for 2 seconds
- Left speed set to -500, right speed set to 0 for 1 second

- Stop

This mode has no built in stop command so you will have to stop it manually

6.5 Sensors

When you started the interface you were asked what sensors you wanted to use, that was because they are broadcast to Scratch every 10th of a second. In the top right window in Scratch there should be 3 lists. The “prox sensor” list stores all the values from the infra-red sensors along the front and back of the Thymio. The ground list stores the values of the 2 sensor one the underside of the Thymio. These don’t record the distance however but instead use ambient light and reflected light to get a more reliable reading. The larger the value the more likely it is that it has gone off the end. The final list is the accelerometer list. This stores the values of the 3 axis for the accelerometer.

Program the Thymio to move forward when there is an object close behind it.

Program the Thymio to turn when the sensors to the side detect an object.

If you are finding this hard, remember that Scratch is very good at parallel processes. One loop could look for objects and then broadcast a message that would start another process. You can broadcast any message and the “when I receive …” tile to react to that broadcast

6.6 Avoid objects

Build on the previous task to make the robot move around an area while avoiding objects and determining the best direction to move in when it does encounter an obstacle.