**Debugging Tasks with Thymio**

**Robotic Semester Project**Ludovic Pujol – EPFL Master

***LESSON (20 min)***

*This section presents a brief lesson on Thymio and the programming software, Thonny, that will be used in the next part. The goal of this lesson is for you to be able to understand the main concepts of programming Thymio. You will not be asked to code an entire algorithm, but rather to understand how the lines of codes are executed with Thonny. Feel free to ask questions if you have any doubt.*

***What is Thymio?***

Thymio is a two-wheeled robot, used for educational purposes. Programs can be sent to the robot using a computer. It can move, display colors, sense the walls around him, sense the darkness of the floor and play sounds.

Buttons

Front proximity sensors

Ground sensors

Back proximity sensors

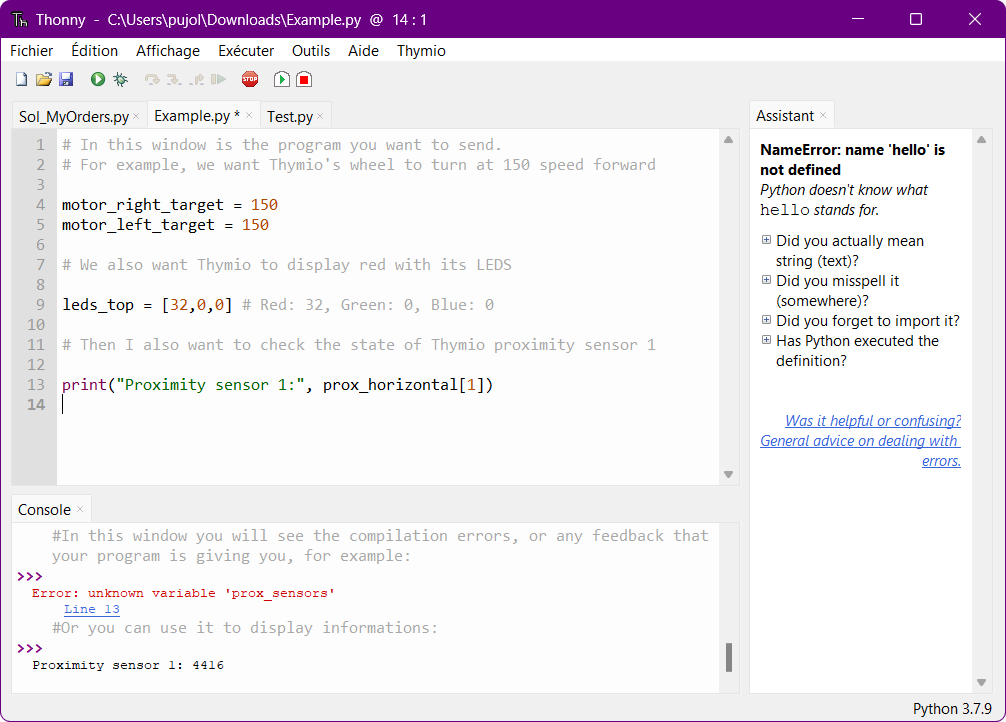
LEDS

***How do we communicate with Thymio?***

Using the USB key, the pre-installed Thymio Suite program can detect the connected Thymio. We can then use different coding interfaces such as VPL3 or Scratch to write programs and send them to Thymio. In our case, we will use Thonny, which is the python interface for Thymio.

***How to use Thonny?***

«

You will use Thonny, a coding software, to write your programs, and to send them to Thymio. The main windows are the Editor (1), where you can modify the code to be sent to Thymio, and the Console (2), where feedbacks are given, from the compiler or from your program directly. You can directly send the program to Thymio and execute it, pressing the green arrow (3), or stop the program pressing the red square (4). You can open files (5), switch (6) between them, or close them.

③

④

⑤

⑥

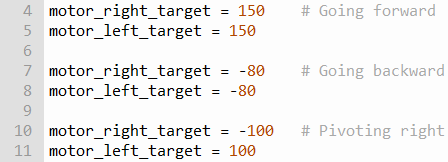
① EDITOR

② CONSOLE

***How to send commands?***

Thonny has some pre-defined variables to help you refer to Thymio’s components.

**Wheels:** to make Thymio move, you should set the rotation speed of both wheels (right and left) to a value from -500 to 500. Thymio will move to a constant speed until another target command is sent. *Open the file* motors.py *to test the wheels.*



**Know a variable state:** you can use the print() function to display the value of a certain variable. For example, you can print the last known value of motor\_right\_target to make sure that it is -100:



**LEDS**: to set the LEDS’ color, you should choose a value between 0 and 32 for red, green and blue. *Open the file* leds.py *to test them.* For example, if I want Thymio to appear orange:

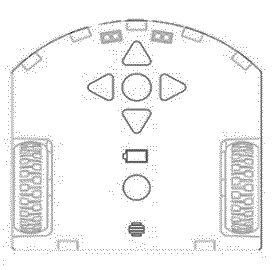


**Buttons**: while a button is pressed, the following variables will be automatically set to True. If they are not pressed, they will be False by default. For example, if I press the forward and right arrows:



**Proximity sensors:** if one of Thymio’s seven proximity sensors detect an obstacle, the corresponding values (*see Figure 1*) will increase from 0 (no obstacle) to 4000 (really close obstacle). For example, to know how far are the walls at the front and at the back, you can print the values of the second and sixth prox. *Open the file* prox.py *to play with the sensors*.





⑤

⑥

④

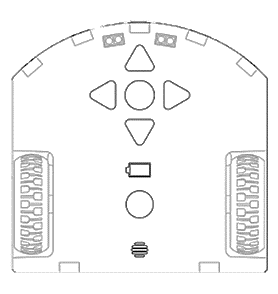
③

②

①

⓪

**Fig. 1: Proximity sensors**



⓪

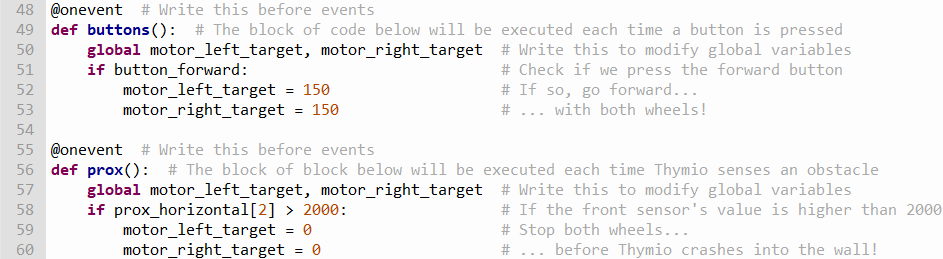
①

**Fig. 2: Ground sensors**

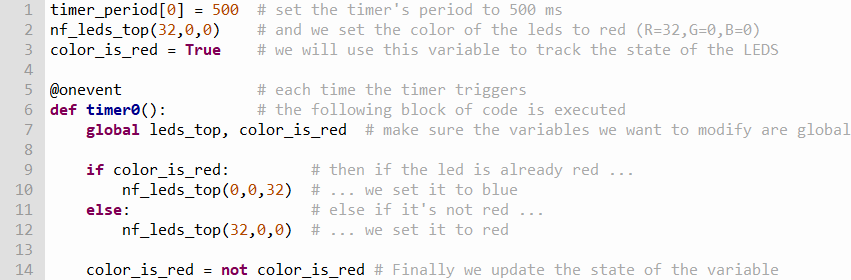
**Ground sensors:** Thymio has two ground sensors at the front (*see Figure 2*). Their values range from 0 (full black floor) to 1000 (fully white/reflective floor).



**Events:** last but not least, you can send commands, only when certain events occur. For example, in the following program, Thymio goes forward only if the forward button is pressed, and will stop if there’s an obstacle in front. *Open the file* events.py *to play with Thymio*.



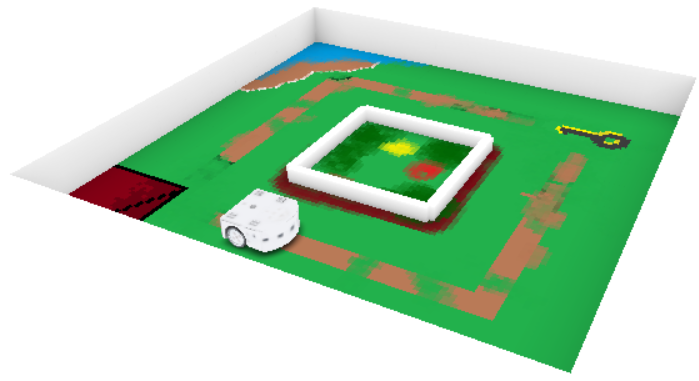
**Timers:** timers can also be used with Thymio, and each time a timer ends, it triggers its corresponding event. We need to set its period (in millisecond) every time it ends. The following code makes Thymio switch between red and blue every second.



Once you finished writing your code, you can send the program to Thymio using the green arrow. Thymio will then execute every command line of your code, so if its behavior does not match the expected one, the error comes from the program. If the error is syntactic, the console will surely let you know. If the error is logical, you will have to review your code, and think twice about the algorithm you are asking Thymio to follow.

***EXERCISES (1 hour)***

*In this section you will be asked to find, and correct bugs in a total of 5 small programs for Thymio. The purpose of these exercises is not to see if you can find the errors, but rather to study how you find them. The first two exercises are for you to familiarize with Thymio, Python and Thonny. Exercise 3 and 4 are a bit more challenging. The last task is considered difficult, don’t worry if you cannot solve it. Expected time ~ 1h*

Exercise 1 – Around the garden (10 min)

* *Open the folder* Exercise 1 – Around the Garden

Thymio is locked outside his house. Make him follow a perfect square around the garden so he can pick up the keys and get back to the house.

Step 1 (8 min)

* *Open the* Bug1\_Garden.py *file*

It seems that Thymio is not able to reach the keys. Can you guess what is wrong in the instructions we give him? There is only one error.

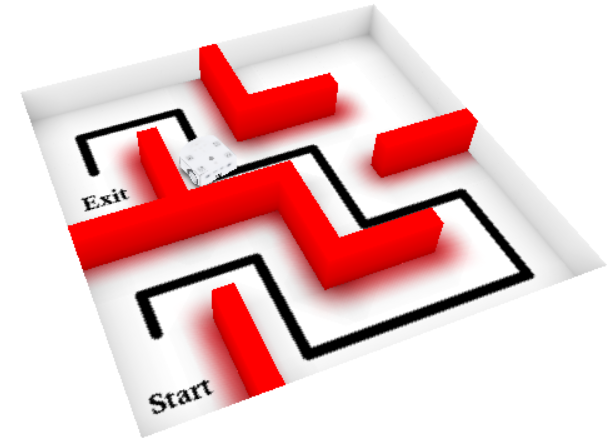
Step 2 (2min)

* *Open the* Bug2\_Garden.py

It seems that Thymio has difficulties orienting himself. Can you guess where we made the mistake? There is only one error.

Exercise 2 – The labyrinth (5 min)

* *Open the folder* Exercise 2 – The labyrinth *and the file* Bug\_TheLabyrinth.py

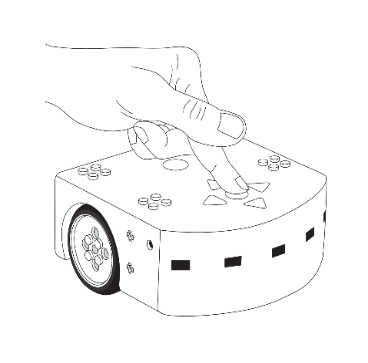
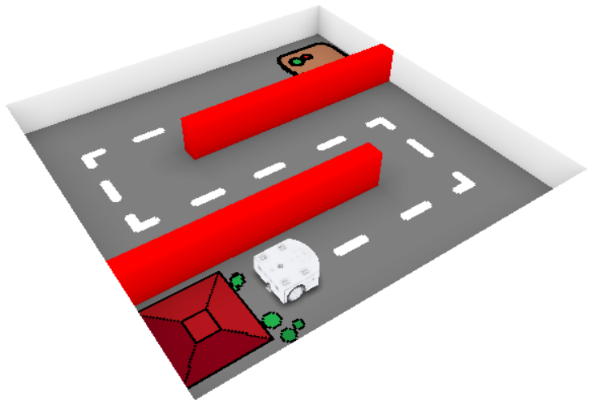
Thymio is lost in this labyrinth. You know where is the exit, and you’re giving him instructions on where to go. However, it seems that there are some mistakes in the code. Can you spot them? There are two errors.

Exercise 3 – Follow my orders (6 min)

* *Open the folder* Exercise 3 – Follow my orders *and the file* Bug\_MyOrders.py

Thymio is late for its first day at work and he does not know the location of the laboratory. You need to indicate the path using the arrows on its head.

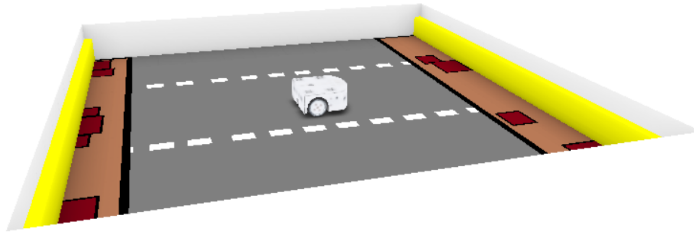
He is finally ready to go to work but it seems that you cannot communicate properly with him and Thymio cannot understand your instructions. What is wrong? There are two errors.



Exercise 4 – The transporter (9 min)

* *Open the folder* Exercise 4 – The transporter *and the file* Bug\_TheTransporter.py

You need to get some piles of wood from a side to another, and Thymio is here to help you. Thymio goes back and forth continuously, and uses its sensors to detect the walls. Once it’s close enough to the wall, it goes back. Well… this would be the case if the code was written correctly. Can find out what’s wrong? There are two errors.



****Exercise 5 – The ultimate test (30 min)****

* *Open the folder* Exercise 5 – The ultimate test *and the file* Bug\_FinalTest.py

Thymio registered for the most prestigious robot evaluation, and needs your help. For this ultimate task, Thymio will have to use its proximity sensors, its ground sensors and the buttons on his head. In the first part of this track, Thymio will orient itself only using the walls. In the second part, he will follow the dark path on the ground. Finally, you will have to push the central button on its head so Thymio stops in the green slot.

However, multiple bugs can be found in the code. It seems Thymio’s behavior does not follow the expected one. Thymio needs to be ready in 30 minutes or he will fail the test. Can you spot the mistakes? There are three of them.

