C. RODRIGUES / D. VIEIRA	TP Internet of Things

TP 2 : Questions

Goals:

1. **Learn to manipulate a Raspberry Pi**
2. **Learn to assemble and use different devices with a Raspberry Pi**
3. **Implement simple applications in Python, using sensors and actuators connected to the Raspberry**

II – Exercises:

1. **Before proceeding to the exercises, you MUST complete the tutorial examples to understand how to assemble and manipulate the devices we are going to use during this course.**
2. **You will deliver a report that will contain the python code and a picture or sketch (preferable) of the model you assembled for each question.**

Exercise 01: Rotate the servo shaft as follows: the shaft starts at 0º degrees, the application waits 1s and turns it 90º degrees. Then, the application waits another 1s and rotates the shaft 180º degrees from the initial position. After another 1s, the shaft returns to the initial position and the procedure starts again.

Exercise 02: Create an application that requests an angle (between 0º and 180º degrees) to the user and rotates the shaft at this angle.

Exercise 03: Create a circuit with a button and a servo. Each time the button is pressed, the servo rotates the shaft 60º degrees until it reaches 180º degrees from the initial 0º degree. When 180º degrees is reached and the button is pressed again, the shaft will return to 0º degree.

Exercise 04: Turn on a red LED when no movement is detected by the PIR motion sensor. A green LED must be turned on when movement is detected.

Exercise 05: Blink an LED while the PIR motion sensor detects movement.

Exercise 06: Get and show (in Celsius and Fahrenheit) the current ambient temperature every 3s.

Exercise 07: Create a program that requests a temperature in Celsius to the user and turns on an LED only when the temperature detected by the sensor exceeds the reported value. If the temperature goes down the light will be turned off.

Exercise 08: Take last question system and add one more LED. This time, the user enters 2 values, one for temperature and one for humidity. The new LED will turn on if the humidity is above the threshold defined by the user.

Exercise 09: Make a program that reads the temperature and humidity of the environment and writes them on a file.

Exercise 10: Use HC-SR04 ultrasonic sensor to calculate the distance between this sensor and an object. The user can choose the between meters and feet .

Exercise 11: Create a circuit with an ultrasonic sensor and a LED. The LED blinks faster and faster as an object approaches the sensor. When the object is less than 5 cm from the sensor, the led will blink for 50 ms. When the object is 1 m from the sensor, the blinking period is 1s. If the distance is between 5 cm and 1 m, the blinking period will be proportional to the values of these extreme distances. If the distance to the object is greater than 1 m, then the led will not turn on.

Exercise 12: Simulate a protected area and use the sonar to detect when a person overpasses an imaginary fence. You must use a sonar, a red and green LED and a buzzer. The sonar is at the center of the protected area, e.g., a precious work of art (book, jewelry, picture, etc.) and when something comes close to it (less than 1 meter), the buzzer (bell) must be activated for 5 seconds, the red LED must be turned on and the green LED must be off. Otherwise, the buzzer must be off, the green LED must be turned on and red LED off. Finally, change this program so the user can modify the distance to activate the alarm. The value must respect the detection limits.

Exercise 13 This exercise will simulate an automatic door like those you can find at the mall. You must make a project composed of one servo and one motion sensor. First, the system will detect when a person is in front of the door (near the motion sensor). If so, then the system changes the shaft servo to 90º degrees (representing the opening of the door). If the sensor does not detect anyone else, the shaft servo must go back 90º degrees (representing the closing of the door). Second, add two LEDs (red and green). The manager can close the door definitely through the software and, by using the python script, the manager can set the door system to 'enabled' or 'disabled'. When 'enabled', the system will work as the first step of this exercise. However, when the manager chooses 'disabled', the door will stay closed even if someone is detected. The python script will wait for two options: 1 = Enabled Door, or 2 = Disabled Door.

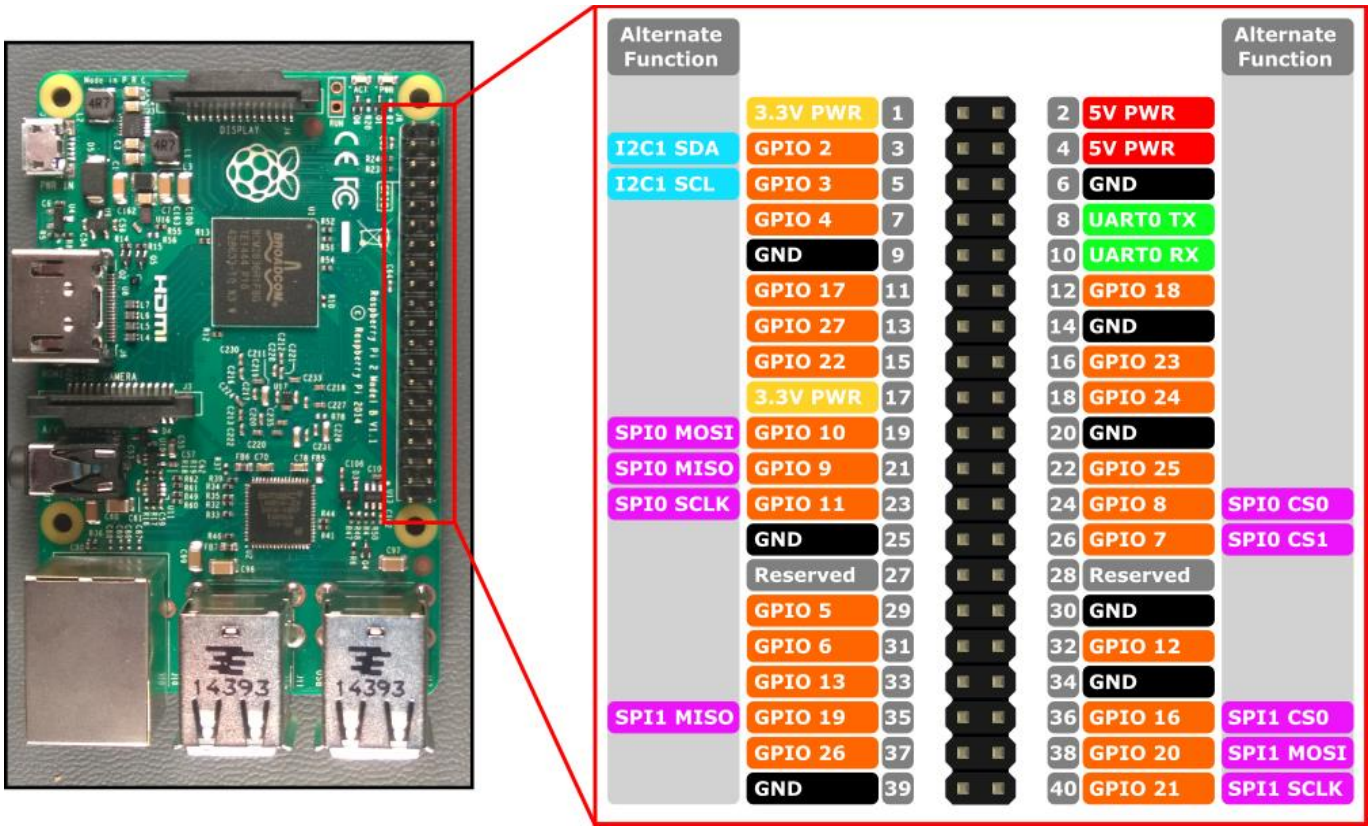
Exercise 14 - Simple smart home:

This exercise represents the idea of a very simple smart home. It illustrates a house with sensors and actuators being controlled by a user and automatic services. The first column of Table 1 exposes all available services such as the alarm system, which could be formed by many motion sensor around the house to amplify the vigilance however (we will only use one); the Lights, which illustrate the lights of a house and we will use different colors to represent different rooms; the temperature, which can be deployed using many sensors spread inside the house to detect a fire. All these services are represented in this project by the components in the third column of Table 1. The system must manage all services using a menu with the options described in the second column of Table 1.

You can use timeouts or threads to implement this simple smart home.

Services	Actions	Component
Alarm system	1 - Enable 2 - Disable	Buzzer and Motion
Lights	1 - Enable [Room] 2 - Disable [Room] 3 - Turn all on 4 - Turn all off	LED Red - Living Room LED Green - Bedroom Led Yellow - Kitchen
Temperature	1 - Set value to trigger 2 - View temperature	Temperature and Buzzer

Table 1: Smart Home



For more information about Raspberry Pi GPIO ports, check:

https://pinout.xyz