IS4151/IS5451 – AIoT Solutions and Development

AY 2024/25 Semester 2

Practical Lab 04 – Single-board Computer (II)

Part 1 – Basic Programming

PE04-1-1 – Advanced Messenger – Raspberry Pi Version

Recall that in PE02-1-2, you have created an advanced messenger that allows a designated micro:bit device nearest to a fog server to relay messages to it. The fog server in PE02-1-2 is implemented using a conventional computer with the micro:bit device communicating with it using serial over USB.

In this exercise, you are required to replace the conventional computer with a Raspberry Pi device as the fog server. The micro:bit device will communicate with it using serial over USB.

When you are done, try changing the serial communication mode to GPIO instead of USB.

PE04-1-2 – Ultimate Messenger

In the previous PE04-1-1, you have created an advanced messenger that allows a designated micro:bit device nearest to a fog server (implemented using a Raspberry Pi) to relay messages to it using serial communication over USB and GPIO.

Earlier in PE02-1-2, you have also created a similar version of an advanced messenger but using a conventional computer as the fog server. For that version of the advanced messenger, the micro:bit device communicates with the conventional computer using serial over USB.

Both serial over USB and serial over GPIO impose a constraint on the maximum distance between the last-mile micro:bit device and the fog server due to the use of a physical, wired connection. Recalled that in Lecture 7, we have discussed the Bluetooth UART service, which emulates the behaviour of a physical UART system using Bluetooth Low Energy (BLE). The maximum wireless transmission range of BLE with line of sight can be in excess of 100 m. In an empirical test conducted by a micro:bit user (http://bluetooth-developer.blogspot.com/2016/08/microbit-and-bluetooth-range-testing.html), the maximum distance with line of sight was at least 225 m.

Suppose that the fog server at the corner of the rectangle is still a Raspberry Pi. Modify the existing messaging protocol to allow the designated last-mile micro:bit device nearest to the fog server to relay the message to the fog server using the Bluetooth UART service. Upon receiving the messages, the Raspberry Pi will display them on the output.

a. Did you have to make any changes to the hardware setup of the last-mile micro:bit device? If yes, describe the changes that you have made.

b. What are the changes that you have made to the existing messaging protocol?

PE04-1-3 – Smart Lighting Switch

Design and implement a simple IoT system to allow a micro:bit device to function as a smart lighting switch. More specifically, the micro:bit device is to interact with a Raspberry Pi device over its Bluetooth UART service to enable the micro:bit device to act as a wireless remote control switch for a hypothetical smart light. In particular, you will use micro:bit onGesture event handler to operate the smart light as follows:

1. Gesture.ScreenDown – Switch Off the Light

2. Gesture.ScreenUp – Switch On the Light

3. Gesture.Shake – If Light is currently switched On, toggle the brightness between full and dim (e.g., 50% brightness), otherwise ignore the event.

4. onButtonPressed(Button.A) – Reset the Light to Off and prompt user to place the micro:bit with the screen facing down.

Create a Python program that would run on the Raspberry Pi device to scan and connect to the micro:bit device on its Bluetooth UART service. Thereafter, create a simple protocol that consists of various commands to be sent by the micro:bit to the Raspberry Pi in response to each of the above events.

When the Raspberry Pi device receives a particular command, it will simply display the command on the output.

Part 2 – Intermediate Programming

PE04-2-1 – Smart Lighting Switch with a “Real Light”

Enhance the smart lighting switch in PE04-1-3 to enable the Raspberry Pi device to control a red colour LED upon receiving commands from the smart lighting switch, i.e., the micro:bit device.

More specifically, set up a PWM LED circuit with the help of a MCP3008 ADC to simulate the smart light using a red colour LED. Upon receiving a particular command from the micro:bit, perform the actual actuation on the LED by modifying the Python program running on the Raspberry Pi device.