



# IS4151/IS5451 – Pervasive Technology Solutions and Development

AY 2024/25 Semester 2

Individual Assignment 1

## Objectives

At the completion of this individual assignment, you should:

1. Develop a better understanding on how to employ single-board microcontrollers in various roles to create useful real-world solutions.
2. Acquire a deeper technical skillset on working with the wireless data communication and computational capabilities of general purpose microcontrollers such as micro:bit.
3. Acquire a deeper technical skillset on using microcontrollers to control external sensors and devices.
4. Implement a simple edge processing architecture with a hub involving a single-board computer collating and processing data from multiple microcontrollers.

## General Requirements

You are required to develop a hand gesturing game system known as **GesGesWin+** using micro:bit as the primary microcontroller in conjunction with the Grove shield and various Grove modules. This is an individual assignment, and you are required to complete all project tasks on your own. The submission deadline is the end of Week 9, i.e., **Sunday, 23 March, 11:59 pm**. You will do a simple demonstration during one of the lab classes starting from Week 10 onwards.

The GesGesWin+ system consists of two main components:

- The hand gesturing game device, which is further divided into a primary game device and a secondary wireless remote controller.
- The score display device, which is further divided into a primary display program and a secondary wireless receiver.

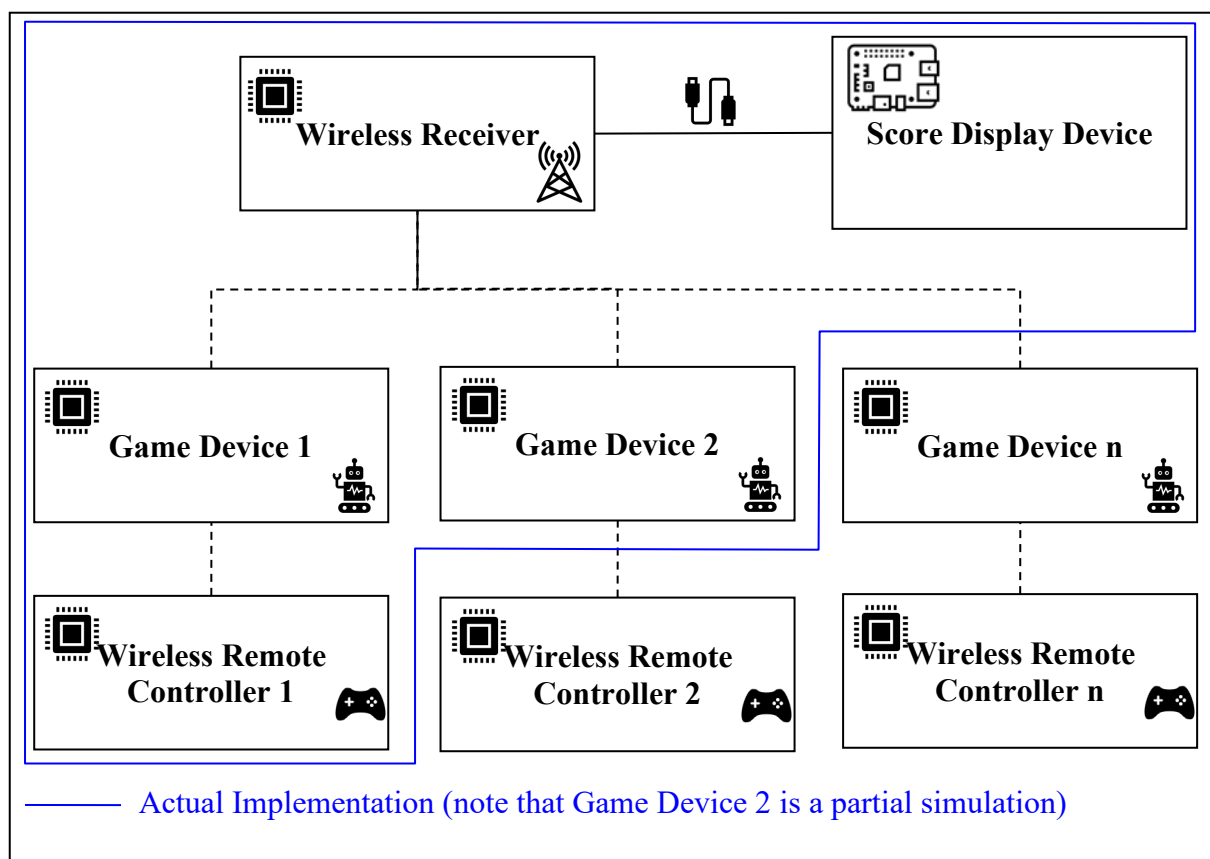
The general game flow begins when a player presses Button A on the wireless remote controller. The player will then be prompted to enter a name consisting of five alphabets through the wireless remote controller. The game will then begin, and player will have three minutes to gesture in the correct directions according to randomly generated arrowheads on the game device. Each correct gesture will score 1 point. The player must stand at least 20 cm

away from the micro:bit at all times during game play. The score will reset to zero each time this rule is violated.

At the end of three minutes, game play will stop, and the player's name and score will be transmitted to the score display program via the wireless receiver. The score display will then show the game device identifier (i.e., device name), player name and player score on the screen in descending order. That is, the highest score will be displayed first. If there are multiple entries with the same score, the latest entry will be displayed first.

Note that the system should be able to accommodate multiple hand gesturing game devices at any one time. However, for the purpose of the assignment, you only need to implement i) one hand gesturing game device fully; and ii) a second game device with partial simulation. This arrangement is mainly due to equipment availability constraint.

A high-level schematic diagram of the GesGesWin+ system is depicted in Figure 1 below.



### **Basic Reference Game Protocol**

The basic reference game protocol is shown in Table 1, and it serves as the main evaluation criteria. However, you are allowed to enhance the protocol in your own implementation as desired.

| S/N | Stage          | Wireless Remote Controller   | Game Device   | Wireless Receiver | Score Display Device   |
|-----|----------------|--|---|-------------------|--|
| 1   | Splash Screen  | <ul style="list-style-type: none"> <li>- Shows a welcome splash screen.</li> <li>- Prompt player to press button A to start game.</li> </ul> | <ul style="list-style-type: none"> <li>- Shows a welcome splash screen.</li> </ul>  |                   | <ul style="list-style-type: none"> <li>- Shows a welcome splash screen.</li> </ul> |
| 2   | Game           | <ul style="list-style-type: none"> <li>- Prompt player to input name</li> </ul>  | <ul style="list-style-type: none"> <li>- Reset the score to 0.</li> </ul>   |                   |  |
| 3   | Initialisation |  | <ul style="list-style-type: none"> <li>- Display player name.</li> <li>- Start game play.</li> </ul>  |                   |  |
| 4   | Game Play      |  | <ul style="list-style-type: none"> <li>- Randomly generate one of six gestures – left, right, up, down, clockwise or anticlockwise.</li> <li>- Display current gesture on the 5x5 LEDs with an appropriate symbol.</li> <li>- The player has 3 seconds to make the correct gesture to score one point.</li> <li>- If after 3 seconds, the user cannot let the gesture sensor detects the correct gesture, the player will not get any point.</li> </ul> |                   |  |

| S/N | Stage | Wireless Remote Controller | Game Device  | Wireless Receiver | Score Display Device |
|-----|-------|----------------------------|--|-------------------|----------------------|
| 5   |       |                            | <ul style="list-style-type: none"> <li>- The game play continues with the random generation of the next gesture, or as soon as the player has made the correct gesture.</li> <li>- That is, the maximum waiting time for each gesture is 3 seconds if the player cannot make the correct gesture. But if the player makes the correct gesture, game play should continue immediately with the next random gesture.</li> <li>- The generation of each new random gesture should be accompanied by a suitable audio tone or clearing of the 5x5 LEDs such that the player is aware even if the next new random gesture is the same as the previous one.</li> </ul> |                   |                      |
| 6   |       |                            | <ul style="list-style-type: none"> <li>- The current score must always be displayed to the player throughout the game play.</li> </ul>   |                   |                      |

| S/N | Stage    | Wireless Remote Controller | Game Device   | Wireless Receiver  | Score Display Device  |
|-----|----------|----------------------------|---|--|---|
| 7   |          |                            | <ul style="list-style-type: none"> <li>- The player must always stand at least 20 cm away from the device during game play.</li> <li>- This refers to the player's body, but the hand may be closer.</li> <li>- If the device detects the player to be less than 20 cm away, the device should sound a warning tone and reset the player's score to 0.</li> </ul> |  |   |
| 8   | Game End |                            | <ul style="list-style-type: none"> <li>- At the end of 3 minutes or 180 seconds, the game will end, and the device will display the player's final score.</li> <li>- Transmit the device identifier (five alphabets), player name and player score to the wireless receiver.</li> </ul>   |  |   |
| 9   |          |                            |   | <ul style="list-style-type: none"> <li>- Receive the device identifier, player name and player score from the game device.</li> <li>- Relay the information to the display program.</li> </ul> |   |
| 10  |          |                            |   |  | <ul style="list-style-type: none"> <li>- Display the latest player score table, including device identifier, player name and score sorted in the required order.</li> </ul> |

Table 1 – Basic reference game protocol.

## **Partial Simulation of Game Device 2**

Game Device 2 is a partial simulation that does not have any attached sensor nor is connected to a Wireless Remote Controller.

Each time user presses Button A, Game Device 2 should simulate a completed game play by randomly generating a player name consisting of five alphabets and a numerical score between 0 and 100. The device identifier, random name and random score should then be relayed to the display program immediately without the need to wait for 3 minutes. The display program must update the score sent by Game Device 2 instantaneously without delay even if there is a game play ongoing on Game Device 1.

## **Score Display Device**

Score Display Device is to be implemented using a Raspberry Pi single-board computer attached with a micro:bit to act as its radio antenna. This allows the relaying of data from the primary game device to the score display device using a combination of wireless radio communication and wired serial communication. You are only required to implement a character-based or console-based user interface with Python. Web interface is not required.

## **Overall Assessment Criteria**

The overall assessment criteria are listed below.

| <b>Criterion</b>                                    | <b>Maximum Possible Marks</b> |
|---|-------------------------------|
| Implementation of the Game Device                   | 5                             |
| Implementation of the Wireless Remote Controller    | 2                             |
| Implementation of the Wireless Receiver             | 1                             |
| Implementation of the Display Program               | 1                             |
| General Coding Techniques and Design Considerations | 1                             |
| <b>Total</b>  | <b>10</b>                     |

## **Assignment Deliverables Submission**

The deliverables to be submitted to Canvas at the end of Week 11 are to be placed in a single zip archive file with the following folders structure:

- **docs** subfolder containing:
  - Softcopy of a Microsoft Word document in DOCX format briefing explaining the design of your system.
  - Special instructions for setting up and running your system.
- **source** subfolder containing:
  - All project folders/source files that constitute your system.

- **readme.txt** containing:
  - Full name
  - Student number beginning with “A0”.
  - Email.
  - Contact number.

Upload this zip archive file to the designated Canvas Assignment: “Assignments > Individual Assignment 1”.

Your assignment deliverables must be submitted latest by **Sunday, 23 March, 11:59 pm**. No assignment will be accepted for assessment after this date/time. For the demonstration, you will be given the source files from your Canvas submission.

***-- End of Individual Assignment 1 Specification --***