School of Science, Computing & Engineering Technologies

# Portfolio-Projects Guidelines (On-Campus Class)

## EEE20003

## Embedded Microcontrollers

Semester 2 2022

## General Information:

This unit, EEE20003 Embedded Microcontrollers of study introduces microcontroller and their application in embedded system. Investigative, design and problem-solving skills are emphasised within the laboratory program.

For the portfolio project, you will work in a group of two to three students. Your team will be formed on the first week in your laboratory class. The project tasks will generally involve planning, design, construction, programming, testing and debugging. When you form the group, discuss allocation of these roles among your team members. Moreover, you should actively involve in the discussion of each task design and fully understand the design details of your entire project although some designs do not belong to your main role. It is advised that each team should choose a team coordinator to lead the team and chair group meetings.

## Online project management tool:

You will be using an online project management tool, Trello for your group work Trello (https://trello.com/), which is quick and easy to use and will help students stay on track and allocate jobs and timelines.

| | | | | |
|---|---|---|---|---|
| Portfolio - Project<br><br>i) Project Progress (10%)<br>ii) Project Demonstration (25%)<br>iii) Reflective report (5%) | Group/ Individual<br>Group<br><br>Group<br><br>Individual | 40% | 1, 2, 3, 4, 5, 6 | i) Sep 9, 2022 by 23:59pm (End of Week 6); submit in Canvas<br>ii) Your on-campus lab class in week 12 & project codes due in your lab class of week 12<br>iii) Nov 4, 2022 by 23:59pm. |

## Portfolio – Project Assessments:

1) Project progress
    - To demonstrate project part 1 according to the design requirement part 1.
    - One student in each group must submit group slides and video demonstration.
    - The slides and video submitted as single compressed ZIP file, which must be named as 'your group ID-PP'. For example: 12345678-PPS.zip.
    - Share the link of the project in Canvas.
2) Project code from project demonstration
    - To demonstrate project part 2 according to the design requirement part 2.
    - One student in each group must submit a copy of project code demonstrated in class.
    - The code must be submitted as a single compressed ZIP file, which must be named as 'your group ID-PC'. For example: 12345678-PC.zip.
3) Reflective report
    - This is individual reflection according with the design requirement part 1 and part 2.

- Each student must submit a copy of individual report.
- The report must be submitted as a single PDF or Word file, which must be named as 'your ID-PR'. For example: 12345678-PR.pdf

## List of Projects:

Project A: Visual IMU with GLCD

Project B: Visual IMU with LED Matrix

Project C: IMU-based Wheelchair commands with GLCD

Project D: IMU-based Wheelchair commands with LED Matrix

Project E: Access control System with GLCD

Project F: Access control System with LED Matrix

Project G: Calculator

# Project A: Visual IMU with GLCD

1. Project Description
   Accidental falls are common, especially among elderly people and result in injuries that involve hospitalisation or death. As a result, immediate fall detection to enable timely rescue is important in reducing the negative effects of a fall on the health and well-being of that particular person. Inertial measurement unit (IMU) sensors are widely used as wearable systems for detecting falls in real-time.

2. **Design Requirement Part 1 (Simulation)**
   - Students will use the simulator to explore components includes IMU using MPU-6050 6-Axis (Gyro + Accelerometer), SSD1306 OLED display and Arduino Mega microcontroller.
   - Program the microcontroller with the routine to read the IMU and displaying symbol/figure in display. The display should show the displacement in 3 axis: inward, outward movements and displacement in the x and y-axis of the IMU.
   - Important: interrupt routine is not needed for this part.
   - Assignment due date: this is part of the assignment for Portfolio-Project Progress and Portfolio-Project Report.

3. **Design Requirement Part 2 (Simulation/Real Hardware)**
   - Students will explore components including the same IMU from part 1 for fall detection.
   - Students are required to design and implement a visual IMU using graphics LCD (GLCD), IMU and Arduino microcontroller.
   - Generate routine for fall detection or sudden change in height to represent fall detection. Welcome to formulate theories for fall detection.
   - Graphical design to show the different parameters are open for all, any logical labelling for dimensions are accepted. Graphical design to show the displacement in 3 axis: going down, upward movement and displacement in x and y-axis.
     - Memory for initial position (read and write from EEPROM/PROGMEM).
   - Important requirements:
     i. The system is protected by a password, which is stored in the EEPROM/PROGMEM. The default password is EEE20003. The password needs to be entered from serial monitor. When the system is started, it will wait for the string to be entered in the serial monitor and compared to the password stored in the EEPROM/PROGMEM. If the password is corrected entered, it will continue further.
     ii. If the password is entered correctly, create menu in the serial monitor to select different options:
       - Option-1 The initial position of the IMU. write/read this data into the EEPROM/PROGMEM.
       - Option-2 The data from the additional sensor. Write and read this data into the EEPROM/PROGMEM (Note: if section iv is done)

- Option-3 interrupt-driven programming.

iii. If option-3 is selected, it will enter the main routine for the main task of the project. Here, the project needs to implement interrupt-driven programming with IMU for fall detection. This can be done by using external and internal interrupts or their combination. There should be no significant delay in the interrupt service routine (ISR), which can degrade the system's overall performance.

iv. For getting High Distinction in the Project Demonstration, you will need to add components from the simulator and write the routine/program to use the additional component of enhancing the product. The additional component should be able to relate back to the original task of the system. Interrupt-driven programming is needed. The components can be used, including photo-resistor/sensor, DHT22 Digital Humidity and Temperature sensor, HC-SR04 Ultrasonic Distance Sensor and RTC (Real Time Clock). Other criteria for the HD will need to be satisfied, according to Rubrik. Other components, please discuss with your tutors.

- Assignment due date: this is part of the assignment for Portfolio-Project Demonstration and Project Report.

4. Project Components

Part 1 & Part 2 (Simulation): Components from Simulator: MPU-6050 6-Axis (Gyro + Accelerometer), SSD1306 OLED display, Arduino Mega microcontroller and additional chosen component.

or Part 2 (Real Hardware): IMU - MPU9250, GLCD, Arduino Due microcontroller and additional chosen component.

## Project B: Visual IMU with LED Matrix

1. Project Description

   Accidental falls are common, especially among elderly people and result in injuries that involve hospitalisation or death. As a result, immediate fall detection to enable timely rescue is important in reducing the negative effects of a fall on the health and well-being of that particular person. Inertial measurement unit (IMU) sensors are widely used as wearable systems for detecting falls in real-time.

2. **Design Requirement Part 1 (Simulation)**
   - Students will use the simulator to explore components includes IMU using MPU-6050 6-Axis (Gyro + Accelerometer), LED Matrix and Arduino Mega microcontroller.
   - Program the microcontroller with the routine to read the IMU and displaying symbol/figure in LED Matrix. The display should show the displacement in 3 axis: inward, outward movements and displacement in x and y-axis of the IMU.
   - Important: interrupt routine is not needed for this part.
   - Assignment due date: this is part of the assignment for Portfolio-Project Progress and Portfolio-Project Report.

3. **Design Requirement Part 2 (Simulation/Real Hardware)**
   - Students will explore using the same IMU in part 1 for fall detection.
   - Students are required to design and implement a visual IMU using LED Matrix, IMU and Arduino Mega.
   - Graphical design to show the displacement in 3 axis: Inward circle shows going down; Outward circles shows upward movement; Arrow to show displacement in x and y axis
   - Memory for initial position (read and write from rom).
   - Generate routine for fall detection or sudden change in height to represent fall detection. Welcome to formulate theories for fall detection.
   - Important requirements:
     i. The system is protected by a password, which is stored in the EEPROM/PROGMEM. The default password is EEE20003. The password needs to be entered from serial monitor. When the system is started, it will wait for the string to be entered in the serial monitor and compared to the password stored in the EEPROM/PROGMEM. If the password is corrected entered, it will continue further.
     ii. If the password is entered correctly, create menu in the serial monitor to select different options:
        - Option-1 The initial position of the IMU. Write/read this data into the EEPROM/PROGMEM.
        - Option-2 The data from the additional sensor. Write and read this data into the EEPROM/PROGMEM (Note: if section iv is done).
        - Option-3 interrupt-driven programming.

     iii. If option-3 is selected, it will enter the main routine for the main task of the project. Here, the project needs to implement interrupt-driven

programming with IMU for fall detection. This can be done by using external and internal interrupts or their combination. There should be no significant delay in the interrupt service routine (ISR), which can degrade the system's overall performance.

iv. For getting High Distinction in the Project Demonstration, you will need to add components from the simulator and write the routine/program to use the additional component of enhancing the product. The additional component should be able to relate back to the original task of the system. Interrupt-driven programming is needed. The components can be used, including photo-resistor/sensor, DHT22 Digital Humidity and Temperature sensor, HC-SR04 Ultrasonic Distance Sensor and RTC (Real Time Clock). Other criteria for the HD will need to be satisfied, according to Rubrik. Other components, please discuss with your tutors.

- Assignment due date: this is part of the assignment for Portfolio-Project Demonstration and Project Report.

4. Project Components

Part 1 and Part 2 (Simulation): Components from Simulator: MPU-6050 6-Axis (Gyro + Accelerometer), LED Matrix and Arduino Mega microcontroller and additional chosen component.

Or Part 2 (Real Hardware): IMU - MPU9250, LED Matrix, Arduino Due microcontroller and additional chosen component.

## Project C: IMU-based Wheelchair commands with GLCD

1. Project Description
   An assistive technology (AT) is a device that bridges the gap between the physical (cognitive) abilities of individuals with disabilities and a function that they want to perform. A wheelchair as AT provides mobility and improves function for people who are otherwise unable to move around by themselves. In the application of a wheelchair control for people with disabilities, various hands-free technologies have been used to replace the joystick. One of the technologies is using Inertia Measurement Unit (IMU) based head movement systems for controlling wheelchair.
2. **Design Requirement Part 1 (Simulation)**
   - Students will use the simulator to explore components includes IMU using MPU-6050 6-Axis (Gyro + Accelerometer), SSD1306 OLED display and Arduino Mega microcontroller.
   - Program the microcontroller with the routine to read the IMU and displaying symbol/figure in display. The display should show 5 wheelchair commands: steady (default), left, right, up and down respectively related to the IMU movements.
   - Important: interrupt routine is not needed for this part.
   - Assignment due date: this is part of the assignment for Portfolio-Project Progress and Portfolio-Project Report.
3. **Design Requirement Part 2 (Simulation/Real Hardware)**
   - Students will to explore to use the same IMU for wheelchair commands.
   - Students are required to design and implement a visual IMU using GLCD, IMU and microcontroller.
   - Graphical design to show 5 wheelchair commands: steady (default), left, right, up and down respectively.
   - The full operation and commands is as follows:
     - if the IMU in steady position, the display should show it.
     - If the IMU tilts to the left, the display should be showing arrow left symbol moving from right to left.
     - If the IMU tilts to the right, the display should be showing arrow right symbol moving from right to left.
     - If the IMU tilt to the front, the display should be showing arrow up system moving from bottom to the top.
     - If the IMU tilt to the back, the display should be showing arrow up system moving from top to the bottom.
   - Standalone embedded system.
   - Important requirements:
     i. The system is protected by a password, which is stored in the EEPROM/PROGMEM. The default password is EEE20003. The password needs to be entered from serial monitor. When the system is started, it will wait for the string to be entered in the serial monitor and compared to the password stored in the EEPROM/PROGMEM. If the password is corrected entered, it will continue further.
     ii. If the password is entered correctly, create menu in the serial monitor to select different options:

- Option-1 The initial position of the IMU. Write/read this data into the EEPROM/PROGMEM.
- Option-2 The data from the additional sensor. Write and read this data into the EEPROM/PROGMEM (Note: if section iv is done)
- Option-3 interrupt-driven programming.

iii. If option-3 is selected, it will enter the main routine for the main task of the project. Here, the project needs to implement interrupt-driven programming with IMU for wheelchair commands. This can be done by using external and internal interrupts or their combination. There should be no significant delay in the interrupt service routine (ISR), which can degrade the system's overall performance.

iv. For getting High Distinction in the Project Demonstration, you will need to add components from the simulator and write the routine/program to use the additional component of enhancing the product. The additional component should be able to relate back to the original task of the system. Interrupt-driven programming is needed. The components can be used, including photo-resistor/sensor, DHT22 Digital Humidity and Temperature sensor, HC-SR04 Ultrasonic Distance Sensor and RTC (Real Time Clock). Other criteria for the HD will need to be satisfied, according to Rubrik. Other components, please discuss with your tutors.

- Assignment due date: this is part of the assignment for Portfolio-Project Demonstration and Project Report.

4. Project Components
   Part 1 & Part 2 (Simulation): Components from Simulator: MPU-6050 6-Axis (Gyro + Accelerometer), SSD1306 OLED display, Arduino Mega microcontroller and additional chosen component.

   Or Part 2 (Real Hardware): IMU - MPU9250, GLCD, Arduino Due microcontroller and additional chosen component.

## Project D: IMU-based Wheelchair commands with LED Matrix

1. Project Description
   An assistive technology (AT) is a device that bridges the gap between the physical (cognitive) abilities of individuals with disabilities and a function that they want to perform. A wheelchair as AT provides mobility and improves function for people who are otherwise unable to move around by themselves. In the application of wheelchair control for people with disabilities, various hands-free technologies have been used to replace the joystick. One of the technologies is using Inertia Measurement Unit (IMU) based head movement systems for controlling the wheelchair.

2. **Design Requirement Part 1 (Simulation)**
   - Students will use the simulator to explore components includes IMU using MPU-6050 6-Axis (Gyro + Accelerometer), LED Matrix and Arduino Mega microcontroller.
   - Program the microcontroller with the routine to read the IMU and displaying symbol/figure in LED Matrix. The display should show 5 wheelchair commands: steady (default), left, right, up and down, respectively related to the IMU movements.
   - Important: interrupt routine is not needed for this part.
   - Assignment due date: this is part of the assignment for Portfolio-Project Progress and Portfolio-Project Report.

3. **Design Requirement Part 2 (Simulation/Real Hardware)**
   - Students will to explore using the IMU for wheelchair commands.
   - Students are required to design and implement a visual IMU using LED Matrix, IMU and microcontroller.
   - Graphical design to show 5 wheelchair commands: steady(default), left, right, up and down, respectively.
   - The full operation and commands is as follows:
     - if the IMU in steady position, the display should showing it.
     - If the IMU tilts to the left, the display should be showing arrow left symbol moving from right to left.
     - If the IMU tilts to the right, the display should be showing arrow right symbol moving from right to left.
     - If the IMU tilt to the front, the display should be showing arrow up system moving from bottom to the top.
     - If the IMU tilt to the back, the display should be showing arrow up system moving from top to the bottom.
   - Important requirements:
     i. The system is protected by a password, which is stored in the EEPROM/PROGMEM. The default password is EEE20003. The password needs to be entered from serial monitor. When the system is started, it will wait for the string to be entered in the serial monitor and compared to the password stored in the

EEPROM/PROGMEM. If the password is corrected entered, it will continue further.

ii. If the password is entered correctly, create menu in the serial monitor to select different options:
   - Option-1 The initial position of the IMU. write/read this data into the EEPROM.
   - Option-2 The data from the additional sensor. Write and read this data into the EEPROM/PROGMEM (Note: if section iv is done).
   - Option-3 interrupt-driven programming.

iii. If option-3 is selected, it will enter the main routine for the main task of the project. Here, the project needs to implement interrupt-driven programming with IMU for wheelchair commands. This can be done by using external and internal interrupts or their combination. There should be no significant delay in the interrupt service routine (ISR), which can degrade the system's overall performance.

iv. For getting High Distinction in the Project Demonstration, you will need to add components from the simulator and write the routine/program to use the additional component of enhancing the product. The additional component should be able to relate back to the original task of the system. Interrupt-driven programming is needed. The components can be used, including photo-resistor/sensor, DHT22 Digital Humidity and Temperature sensor, HC-SR04 Ultrasonic Distance Sensor and RTC (Real Time Clock). Other criteria for the HD will need to be satisfied, according to Rubrik. Other components, please discuss with your tutors.

- Assignment due date: this is part of the assignment for Portfolio-Project Demonstration and Project Report.

4. Project Components
   Part 1 and Part 2: Components from Simulator: MPU-6050 6-Axis (Gyro + Accelerometer), LED Matrix and Arduino Mega microcontroller and additional chosen component.

   Or Part 2 (Real Hardware): IMU - MPU9250, LED Matrix, Arduino Due microcontroller and additional chosen component.

## Project E: Access control System with GLCD

1. Project Description
   Keypad and button can be used in the access control application. It is important to make a clear distinction between authentication and access control. Correctly establishing the identity of the user is the responsibility of the authentication service. Access control assumes that authentication of the user has been successfully verified prior to the enforcement of access control.
2. **Design Requirement Part 1 (Simulation)**
   - Students will use the simulator to explore components includes keypad, SSD1306 OLED display and Arduino Mega microcontroller.
   - Program the microcontroller to read input from keypad and displaying the input in the display. Use the student ID of each member as the passcode. If the passcode is entered correctly, shows different symbol/figure for correct authentication, which can differentiate each member otherwise shows incorrect symbol/figure in the display
   - Important interrupt routine is not needed for this part.
   - Assignment due date: this is part of the assignment for Portfolio-Project Progress and Portfolio-Project Report.
3. **Design Requirement Part 2 (Simulation/Real Hardware)**
   - Students will explore to use a button with the purpose for access control system
   - Students are required to design and implement access control system using, button, GLCD and microcontroller Mega
   - Graphical design to show graphical user interface.
   - The full operation and commands are as follows:
     - Your student ID of each member needs to be stored in the EEPROM/PROGMEM of the embedded system as the correct password/authentication.
     - Use the first button pressing to indicate that the system in the user entering passcodes mode.
     - Use a duration of 10 seconds to press the correct number via the button
     - After 10 seconds, the selected digit will stay on display.
     - Repeat this until the whole digits of the password are entered.
     - Use 15 seconds or more to indicate all digits have been entered.
     - As soon all digits of the passcodes are entered, compare digits with the stored student IDs in the EEPROM/PROGMEM.
     - If the entered password is similar to the digits in the EEPROM/PROGMEM, display a correct symbol in the display to that how the authentication has been successfully verified to provide access control.
     - Otherwise, display symbol or text for incorrect passcodes.

- Important requirements:
    i. The system is protected by a password, which is stored in the EEPROM/PROGMEM. The default password is EEE20003. The password needs to be entered from serial monitor. When the system is started, it will wait for the string to be entered in the serial monitor and compared to the password stored in the EEPROM/PROGMEM. If the password is corrected entered, it will continue further.
    ii. If the password is entered correctly, create menu in the serial monitor to select different options:
        - Option-2 change 3 members passcodes. write/read this data into the EEPROM/PROGMEM.
        - Option-2 The data from the additional sensor. Write and read this data into the EEPROM/PROGMEM (Note: if section iv is done)
        - Option-3 interrupt-driven programming.
    iii. If option-3 is selected, it will enter the main routine for the main task of the project. Here, the project needs to implement interrupt-driven programming for the access control system. This can be done by using external and internal interrupts or their combination. There should be no significant delay in the interrupt service routine (ISR), which can degrade the system's overall performance.
    v. For getting High Distinction in the Project Demonstration, you will need to add components from the simulator and write the routine/program to use the additional sensor for enhancing the product. The additional component should be able to relate back to the original task of the system. Interrupt-driven programming is needed. The components can be used, including photo-resistor/sensor, DHT22 Digital Humidity and Temperature sensor, HC-SR04 Ultrasonic Distance Sensor and RTC (Real Time Clock). Other criteria for the HD will need to be satisfied, according to Rubrik. Other components, please discuss with your tutors.
- Assignment due date: this is part of the assignment for Portfolio-Project Demonstration and Project Report.

4. Project Components

Part 1 and Part 2 (Simulation): Components from Simulator: keypad, SSD1306 OLED display and Arduino Mega microcontroller, button and additional chosen sensor.

Or Part 2 (Real Hardware): GLCD, button, Arduino Due microcontroller and additional chosen component.

## Project F: Access control System with LED Matrix

1. Project Description
   Keypad and button can be used in the access control application. It is important to make a clear distinction between authentication and access control. Correctly establishing the identity of the user is the responsibility of the authentication service. Access control assumes that authentication of the user has been successfully verified prior to enforcement of access control.
2. **Design Requirement Part 1 (Simulation)**
   - Students will use the simulator to explore components includes keypad, LED matrix and Arduino Mega microcontroller.
   - Program the microcontroller to read input from keypad and displaying the input in the display. Use the student ID of each member as the passcode. If the passcode is entered correctly, shows different symbol for correct authentication, which can differentiate each member otherwise shows incorrect symbol in the display
   - Important: interrupt routine is not needed for this part.
   - Assignment due date: this is part of the assignment for Portfolio-Project Progress and Portfolio-Project Report.
3. **Design Requirement Part 2 (Simulation/Real Hardware )**
   - Students will explore to use a button for access control system
   - Students are required to design and implement an access control system using pushbutton, LED Matrix and microcontroller.
   - Graphical design of LED Matrix to show graphical user interface.
   - The full operation and commands are as follows:
     - Your student ID of each member needs to be stored in the EEPROM/PROGMEM of the embedded system as the correct password/authentication.
     - Use the first button pressing to indicate that the system in the user entering passcodes mode.
     - Use a duration of 10 seconds to press the correct number via the button
     - After 10 seconds, the selected digit will stay on display.
     - Repeat this until the whole digits of the password are entered.
     - Use 15 seconds or more to indicate all digits have been entered.
     - As soon all digits of the passcodes are entered, compare digits with the stored student IDs in the EEPROM/PROGMEM.
     - If the entered password is similar to the digits in the EEPROM/PROGMEM, display a correct symbol in the display to that how the authentication has been successfully verified to provide access control.
     - Otherwise, display symbol or text for incorrect passcodes.

- Important requirements:
    i. The system is protected by a password, which is stored in the EEPROM/PROGMEM. The default password is EEE20003. The password needs to be entered from serial monitor. When the system is started, it will wait for the string to be entered in the serial monitor and compared to the password stored in the EEPROM/PROGMEM. If the password is corrected entered, it will continue further.
    ii. If the password is entered correctly, create menu in the serial monitor to select different options:
        - Option-1 change 3 members passcodes. write/read this data into the EEPROM/PROGMEM.
        - Option-2 The data from the additional sensor. Write and read this data into the EEPROM/PROGMEM (Note: if section iv is done).
        - Option-3 interrupt-driven programming.
    iii. If option-3 is selected, it will enter the main routine for the main task of the project. Here, the project needs to implement interrupt-driven programming for the access control system. This can be done by using external and internal interrupts or their combination. There should be no significant delay in the interrupt service routine (ISR), which can degrade the system's overall performance.
    vi. For getting High Distinction in the Project Demonstration, you will need to add components from the simulator and write the routine/program to use the additional sensor for enhancing the product. The additional component should be able to relate back to the original task of the system. Interrupt-driven programming is needed. The components can be used, including photo-resistor/sensor, DHT22 Digital Humidity and Temperature sensor, HC-SR04 Ultrasonic Distance Sensor and RTC (Real Time Clock). Other criteria for the HD will need to be satisfied, according to Rubrik. Other components, please discuss with your tutors.
- Assignment due date: this is part of the assignment for Portfolio-Project Demonstration and Project Report.

4. Project Components
   Part 1 and Part 2 (Simulation): Components from Simulator: keypad, LED matrix and Arduino Mega microcontroller, button and additional sensor.

   Or Part 2 (Real Hardware): LED Matrix, button, Arduino Due microcontroller and additional chosen component.

## Project G: Calculator

1. Project Description
   Example of the basic calculator is shown below. To make it work, the calculator should has the microcontroller for the main central processing unit, keypad as the input and LCD display as an output.

2. **Design requirement Part 1 (Simulation)**
   - Students will use the simulator to explore components includes keypad, LCD and Arduino Mega microcontroller.
   - Program the microcontroller to implement virtual calculator using input from keypad (note: 4x4 keypad is available from the simulator) and displaying the calculation in the LCD display. The 16 keypad functions for virtual calculator including: on/off,0,1,2,3,4,5,6,7,8,9,.,=,x,-,+. The calculation can calculate at least 3 operands (inputs) and above, for example “1+2+3”, “2×3×4×5”. The on/off function is basically to turn on/off the LCD.
   - Important: interrupt routine is not needed for this part.
   - Assignment due date: this is part of the assignment for Portfolio-Project Progress and Portfolio-Project Report.

3. **Design requirement Part 2 (Simulation)**
   - Students are required to design and implement virtual calculator as shown picture above using Arduino Mega simulator
     using keypad, LCD and Arduino microcontroller.
   - The keypad is using 5 x 5 push buttons. The keypad module and its library can used.
   - The 25 pushbuttons will be used for the basic keys for the calculator including: on/off, C-CE, +/-, √, %,MC,MR, M-, M+, ÷, 0,1,2,3,4,5,6,7,8,9,.,=,x,-,+.
   - The full operation and commands is as follows:
     - The on/off button is used to turn on/off the calculator.
     - Each of the 25 pushbuttons should able to provide inputs for calculator function.
     - The calculation can calculate at least 3 operands (inputs) and above, for example “1+2+3”, “2×3×4×5.
     - Memory operation (MC,MR, M-, M+), should performing properly.

- The fraction operation also need to be working.

- Important requirements:
  i. The system is protected by a password, which is stored in the EEPROM. The default password is EEE20003. The password needs to be entered from serial monitor. When the system is started, it will wait for the string to be entered in the serial monitor and compared to the password stored in the EEPROM. If the password is corrected entered, it will continue further.
  ii. If the password is entered correctly, create menu in the serial monitor to select different options:
    - Option-1 change intial text to be display in LCD. Write/read this data into the EEPROM.
    - Option-2 Display initial data from the sensor to be stored in the EEPROM (this option if section iv is done)
    - Option-3 interrupt-driven programming.
  iii. If option-3 is selected, it will enter the main routine for the main task of the project. Here, the project needs to implement interrupt-driven programming for calculator system. This can be done by using external and internal interrupts or their combination. There should be no significant delay in the interrupt service routine (ISR), which can degrade the system's overall performance.
  iv. For getting High Distinction in the Project Demonstration, you will need to add components from the simulator and write the routine/program to use the additional sensor for enhancing the product. The additional component should be able to relate back to the original task of the system. Interrupt-driven programming is needed. The components can be used, including photo-resistor/sensor, DHT22 Digital Humidity and Temperature sensor, HC-SR04 Ultrasonic Distance Sensor and RTC (Real Time Clock). Other criteria for the HD will need to be satisfied, according to Rubrik.
- Assignment due date: this is part of the assignment for Portfolio-Project Demonstration and Project Report.

4. Project Components
   Part 1 and Part 2: Components from Simulator: keypad, LCD and Arduino Mega microcontroller, 5 x 5 keypad and additional sensor.

## Rubric – Portfolio - Project (40%) - Group/Individual

**(i) Project Progress (10%) - Group mark**

Your group needs to record video presentation on your group project progress which is to demonstrate the project part 1 according to the design requirement part 1 (simulation – online). This assessment is undertaken as a group and thus all members in your group get the same mark. The requirements for your group presentation are as follows:

- Convey messages clearly
- Time control (minimum 9 minutes & maximum 10 minutes)
- Submit your video in ZIP file and share the link of the project in Canvas.

| Criteria | Weighting (100%) | Levels of achievement (The standard of performance) | | | | |
|---|---|---|---|---|---|---|
| | | Fail – N<br>0 - 49 | Pass – P<br>50 -59 | Credit – C<br>60 - 69 | Distinction – D<br>70 - 79 | High Distinction – HD<br>80 -100 |
| Ability to design and implement an embedded microcontroller project | 40% | Poor understanding of design requirements and no design solutions proposed or implemented. | Fair understanding of design requirements and explore limited alternative solutions but not implemented. | Clear understanding of design requirements and explore some design solutions and implement a solution that basically meet the design requirements. | Clear understanding of design requirements, explore different design solutions, and implement a solution that well meet the design requirements. Clear evaluation and justification provided for final solution. | Demonstrate an advanced understanding of design requirements; propose alternative design solutions and constraints. Implement a creative solution in the practical design. Displays with high level of techniques by industry. |
| Demonstrate programming skills in project design | 40% | Very limited program presented in the design | Present a working program but may codefor incomplete design requirements | Present a single functional programbut may have somelogical errors in theoutputs | Demonstrate high levelof programming skill incompleting the design efficiently and no errors presented in theoutput | Demonstrate a complex level of programming skills, a functional and efficient program. A sophisticated level of logic behind the design and clear explanations on choices. |
| Ability to make a product with required specifications | 20% | Product fails to demonstrate as a whole | Product works as a whole but may not perform well | Product works as a whole with standard achievement against performance index | Demonstrate high achievement against the performance index | Demonstrate high level achievement against the performance indexand display other achievement not specified |

**(ii) Project Demonstration (25%) - Group mark**

The project demonstrations are scheduled in week 12 in your laboratory class. This is the part 2 project demonstration according to the design requirement part 2. Each group member is mandatory to attend the demonstration. Your group is required to present how your developed project works and demonstrate the tasks and specifications it could achieve. You also need to answer the questions raised by the tutor or the class: example questions include your final design solution and explain the program code and results, elaborate the design specifications you have achieved, summarise the teamwork environment, procedure and effectiveness and tasks completed by each group members.

| Criteria | Weighting (100%) | Levels of achievement (The standard of performance) | | | | |
|---|---|---|---|---|---|---|
| | | Fail – N<br>0 - 49 | Pass – P<br>50 -59 | Credit – C<br>60 - 69 | Distinction – D<br>70 - 79 | High Distinction – HD<br>80 -100 |
| Ability to design and implement an embedded microcontroller project (scaled by individual contribution) | 25% | Poor understanding of design requirements and no design solutions proposed or implemented. | Fair understanding of design requirements and explore limited alternative solutions but not implemented | Clear understanding of design requirements and explore some design solutions and implement a solution that basically meet the design requirements | Clear understanding of design requirements, explore different design solutions, and implement a solution that well meet the design requirements. Clear evaluation and justification provided for final solution. | Demonstrate an advanced understanding of design requirements; propose alternative design solutions and constraints. Implement a creative solution in the practical design. Displays with high level of techniques by industry. |
| Evaluate and apply a range of components/sensors with microcontroller in a project (scaled by individual contribution) | 25% | Little knowledge of the range of components and very limited use of components with microcontroller in the project | Limited knowledge of the range of components but few components are used in the project with microcontroller. | Basic knowledge of the range of components with microcontroller in the project. Basic embedded microcontroller used in the design | Demonstrate advanced understanding of components used. Evaluate and effective embedded microcontroller project design | Demonstrate advanced understanding of components used and specifications. Successfully use the sensors in the project with sophisticate embedded system design. |
| Demonstrate programming skills in project design (scaled by individual contribution) | 20% | Very limited program presented in the design | Present a working program but may code for incomplete design requirements | Present a single functional program but may have some logical errors in the outputs | Demonstrate high level of programming skill in completing the design efficiently and no errors presented in the output | Demonstrate a complex level of programming skills, a functional and efficient program. A sophisticated level of logic behind the design and clear explanations on choices. |
| Ability to make a product with required specifications (scaled by individual contribution) | 20% | Product fails to demonstrate as a whole | Product works as a whole but may not perform well | Product works as a whole with standard achievement against performance index | Demonstrate high achievement against the performance index | Demonstrate high level achievement against the performance index and display other achievement not specified |
| Safely execute experiments (scaled by individual contribution) | 10% | Violate laboratory rules and conduct experiments that may induce high level hazard | Follow the laboratory rules and conduct experiments that may induce low level hazard | Follow the laboratory rules and conduct experiments in improper procedure but will not cause hazard | Follow the laboratory rules and conduct experiments as per the operational procedure | Strictly follow the laboratory rules and conduct experiments as per the operational procedure with the ability for potential risk assessment |

### (iii) Project Report (5%) – Individual

You need to write and submit an individual project report as reflective journal, which is the basis for the assessment of your understanding on the individual tasks inthe project and individual contribution to the project. You cannot include any work in this report that is not your own genuine contribution. It needs to arrange in the order of weeks starting from week 1 to week 12 in the semester. The weekly journal should include the following contents:

- Your group activities, e.g. group meeting minutes, group progress, teamwork issues
- Discuss about your personal working progress and how your efforts contribute to the whole group
- Evidence of your personal progress achievements
- Your intention to work with other group members
- Reflection on your learning experience

| Criteria | Weighting (100%) | Levels of achievement (The standard of performance) | | | | |
|---|---|---|---|---|---|---|
| | | Fail – N<br>0 - 49 | Pass – P<br>50 -59 | Credit – C<br>60 - 69 | Distinction – D<br>70 - 79 | High Distinction – HD<br>80 -100 |
| Ability to work in a team and contribute to the project development | 50% | Lacked commitment to the team project and relied on others to complete the work. Demonstrated no concern for deadlines and little evidence of contribution addressed in the report. Attended very few lab classes for group project. | Completed tasks allocated. Showed no contribution to idea building in the team but contributed to discussions. Limited evidence of contribution addressed in the report. Met deadlines. Attended some lab classes for group project. | A good team member who regularly contributes to team decisions and at times offered new ideas to the team. Written and verbal contribution to the team of a high standard. Able to contribute to the advancement of the project. Attended most lab classes for group project. | Demonstrates very good teamwork skills; constructively contributing to the team, clear contribution addressed in the report. A strong contribution to the task and able to engage positively to the team while able to build on the ideas of the team. An effective team member. Attended most lab classes for group project. | Demonstrated ongoing commitment to the team project, to the needs of all team members, to the recording of all results and findings, and was a major contributor to the final solution presented. Consistently worked to all deadlines and assisted other team members. A major contributor to the advancement of the project.<br>Attended all lab classes for group project. |
| A high quality project report | 50% | Poor communication in the report. Structure, organisation, and meaning in the report was unclear. Report was difficult to follow and main ideas not satisfactorily conveyed. | Communicates mostly in a clear manner in the report. Generally appropriate structure and organisation with basic meanings mostly conveyed. | Applies a range of good communication skills appropriate for the report. Structure conforms to required standards and meanings are generally clear. | Demonstrates a very good understanding of conveying ideas verbally and in writing, with clear structure and well-organised outputs. Mostly applies a range of well-developed language skills to provide clear meaning throughout. | Demonstrates complex communication and language skills in the report with a high degree of clarity and meaning. Conveys clear messages throughout demonstrating depth of understanding and thought within a very clear structure. |