



UGANDA INSTITUTE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY

END OF SEMESTER EXAMINATIONS

ACADEMIC YEAR 2025/2026

PROGRAMME(S): DIPLOMA IN ELECTRICAL AND ELECTRONICS ENGINEERING

YEAR OF STUDY: YEAR TWO SEMESTER TWO

COURSE: EMBEDDED SYSTEMS

COURSE CODE : EEE2201

DATE: SUNDAY 27TH, JUNE 2026

TIME: 2:00 PM – 5:00 PM

DURATION: 3 HOURS

VENUE: RM 2.5, RM 2.6, RM 2.8, RM 2.9, RM 2.10

INSTRUCTIONS:

- This examination consists of TWO sections: Section A and Section B.
- Section A is **COMPULSORY**
- Section B consists of FIVE (5) questions worth 20 marks each. Question 1 in Section B is **COMPULSORY**. Attempt Question 1, PLUS any other THREE (3) other questions of your choice from Questions 2-5 (4 questions attempted in total, 80 marks).
- All hardware in Section B is limited to: pushbuttons (digital input), a passive buzzer, LEDs, a digital proximity sensor (HIGH/LOW output only), and a PIR motion sensor (HIGH/LOW output only).
- Diagrams, code, and flowcharts must be clear and fully labelled. Marks for each part are indicated in brackets [].
- Non-programmable calculators are permitted.

- All questions and scenarios refer to the ATtiny85 microcontroller unless otherwise stated.

SECTION A:

Answer ALL questions [@1mark] .

1. Which data type would be most appropriate for storing the state of a single pushbutton (pressed/not pressed) in Arduino C/C++?

- A. float
- B. bool
- C. String
- D. double

2. What is the typical range of values that can be stored in a uint8_t variable?

- A. -128 to 127
- B. 0 to 255
- C. 0 to 65535
- D. -32768 to 32767

3. In Arduino C/C++, a function declared as 'void buzzAlert()' means:

- A. It must return an integer
- B. It returns no value
- C. It cannot accept any code inside it
- D. It runs automatically on reset only

4. What does the keyword 'return' do inside a non-void function?

- A. Ends the program permanently
- B. Sends a value back to the point where the function was called
- C. Restarts the function
- D. Declares a new variable

5. Which function is used to configure a pin on the ATtiny85 as either INPUT or OUTPUT?

- A. digitalWrite()
- B. pinMode()
- C. digitalWrite()
- D. setup()

6. Which function would you use to switch an LED ON connected to a digital output pin?

- A. digitalWrite(pin, HIGH)
- B. digitalWrite(pin, HIGH)
- C. pinMode(pin, HIGH)
- D. analogWrite(pin, ON)

7. Which function is used to check the current state (HIGH or LOW) of a pushbutton connected to a digital input pin?

- A. digitalWrite()
- B. digitalWrite()

- C. pinMode()
- D. delay()

8. A PIR motion sensor and a digital proximity sensor used in this exam both output:

- A. A continuous range of values from 0 to 1023
- B. Only HIGH or LOW signals
- C. Only PWM signals
- D. Serial text data

9. When the ATtiny85 is powered at 5V, a digital input pin reading approximately 5V is interpreted as:

- A. LOW
- B. HIGH
- C. Undefined
- D. An analog value

10. What is the term used for the voltage range between a valid LOW and a valid HIGH that a digital input should never settle in?

- A. Saturation zone
- B. Logic gap / undefined region
- C. Ground zone
- D. Reference zone

11. Why is it important to know a microcontroller's logic level (e.g., 5V vs 3.3V) before connecting a sensor such as a PIR or proximity sensor?

- A. To choose the correct programming language
- B. To avoid damaging the microcontroller or sensor with mismatched voltages
- C. To increase the clock speed
- D. To change the data type used

12. On the standard 8-pin DIP ATtiny85, which pin is the RESET pin by default?

- A. Pin 1 (PB5)
- B. Pin 4 (GND)
- C. Pin 8 (VCC)
- D. Pin 5 (PB0)

13. On the ATtiny85 pinout, which pin number supplies VCC (positive supply voltage) to the chip?

- A. Pin 4
- B. Pin 8
- C. Pin 1
- D. Pin 6

14. Which of the following is an on-chip peripheral found on the ATtiny85?

- A. Ethernet controller
- B. 8-bit Timer/Counter
- C. SD card interface
- D. USB host controller

15. Which function runs exactly once when the ATtiny85 is powered on or reset?

- A. loop()
- B. main()
- C. setup()
- D. init()

16. A nested if statement is best described as:

- A. An if statement placed inside the body of another if statement
- B. Two unrelated if statements in different functions
- C. An if statement without a condition
- D. A loop disguised as a conditional

17. Which loop structure is most appropriate for blinking an LED a fixed, known number of times (e.g., exactly 10 times)?

- A. for loop
- B. if statement
- C. switch statement
- D. goto statement

18. Which of the following is a realistic application of embedded systems in addressing Kampala's 2026 traffic congestion challenges?

- A. A standalone desktop spreadsheet application
- B. An embedded traffic light controller with sensors for adaptive signal timing
- C. A printed paper map
- D. A manual traffic police whistle

19. How many general-purpose input/output (GPIO) pins are available for use on the ATtiny85 in an 8-pin DIP package (excluding VCC and GND)?

- A. 3
- B. 5
- C. 6
- D. 8

20. What is the typical recommended operating voltage range for powering the ATtiny85?

- A. 0.5V - 1V
- B. 1.8V - 5.5V
- C. 12V - 24V
- D. 100V - 240V

SECTION B:

INSTRUCTIONS:

This section has FIVE (5) questions.

Question 1 is **COMPULSORY**.

Attempt **Question 1** plus any **THREE (3)** other questions from **Questions 2-5** (4 questions in total). Each question carries 20 marks.

In all scenarios use only the following hardware: pushbuttons,

a passive buzzer,

LEDs,

a digital proximity sensor (**HIGH/LOW**), and a PIR motion sensor (**HIGH/LOW**).

Question 1 (COMPULSORY)

A community neighbourhood-watch unit in Kampala wants a low-cost ATtiny85 security device. A PIR motion sensor (PB2) detects movement at night (**goes LOW**) and immediately sounds a passive buzzer (PB1 **goes HIGH**) and lights a red LED (PB0). A pushbutton (PB3, internal pull-up) is mounted at the gate for the homeowner to silently arm/disarm the system: a properly debounced single press toggles the system between **ARMED** and **DISARMED**, and every press is added to a running count of arm/disarm events stored for the night.

- (a) Draw and label a fully-Labelled ATtiny85 pinout diagram, clearly marking PB0, PB1, PB2, PB3, VCC, GND and RESET, and indicating which pin connects to which device in this scenario. (7 marks)
- (b) Draw a circuit diagram showing the PIR sensor, buzzer, LED and pushbutton correctly wired to the ATtiny85. (5 marks)
- (c) Draw a flowchart for the arm/disarm and motion-alarm logic, including debouncing of the pushbutton. (5 marks)
- (d) Write a complete Arduino C/C++ sketch for the ATtiny85 implementing the scenario above, including a debounced button press counter and the PIR-triggered alarm. (3 marks)

Question 2

A fan-zone turnstile gate (**Entry control gate**) for the 2026 FIFA World Cup uses a digital proximity sensor (PB2) to detect each fan walking through (HIGH when clear, LOW when a fan is present), incrementing an entry counter. A staff override pushbutton (PB3, debounced) opens the gate LED (PB0) green for VIP entry without counting. If the proximity sensor is triggered without the staff override being pressed first, a passive buzzer (PB1) sounds as a tailgating alert.

- (a) Draw and label the ATtiny85 pinout diagram for this scenario, showing all pin connections. (5 marks)
- (b) Draw the circuit diagram connecting the proximity sensor, pushbutton, LED and buzzer to the ATtiny85. (5 marks)
- (c) Draw a flowchart for the entry-counting, staff-override and tailgating-alert logic. (5 marks)

(d) Write a complete Arduino C/C++ sketch implementing the turnstile logic described above, including a debounced staff override button. (5 marks)

Question 3

A factory bell station uses an ATtiny85 with a pushbutton (PB3, debounced) for workers to log a shift-start tap, an LED (PB0) that flashes once per valid tap, and a passive buzzer (PB1) that sounds once the tap-count reaches 50 workers for the shift. The program must use clearly separated custom functions rather than one long loop().

(a) Draw and label the ATtiny85 pinout diagram showing the pins used in this scenario. (4 marks)

(b) Draw the circuit diagram for the pushbutton, LED and buzzer connections. (5 marks)

(c) Draw a flowchart showing how setup(), loop() and your custom functions interact. (4 marks)

(d) Write a complete Arduino C/C++ sketch that defines and uses at least THREE custom functions: readButtonDebounce(), flashLED(), and checkBatchComplete(), implementing the full scenario above. (7 marks)

Question 4

A motorcycle ignition system uses a single pushbutton (PB1, debounced, internal pull-up) connected to the ATtiny85. A short press does nothing. A LONG press (held at least 2 seconds) lights an ignition-status LED (PB0) to represent the engine starting. While running, another LONG press switches the LED OFF to represent a safe engine shutdown. THREE rapid short presses within 1 second instead trigger a passive buzzer (PB2) as an anti-theft alarm.

(a) Draw and label the ATtiny85 pinout diagram showing the pins used. (4 marks)

(b) Draw the circuit diagram for the pushbutton, ignition-status LED and anti-theft buzzer. (5 marks)

(c) Draw a flowchart distinguishing a short press, a long press, and multiple rapid presses, leading to the correct ignition, shutdown, or alarm action. (5 marks)

(d) Write a complete Arduino C/C++ sketch implementing the long-press ignition/shutdown toggle and the multiple-press anti-theft alarm. (6 marks)

Question 5

A small soap-making factory automates quality control using an ATtiny85. A digital proximity sensor (PB2) detects each soap bar passing on the conveyor (LOW when a bar is present, HIGH otherwise), incrementing a count and flashing an LED (PB0) for each bar detected. A debounced pushbutton (PB3) lets an operator manually reset the count to zero at the start of a shift. After every 50 bars counted, a passive buzzer (PB1) sounds for 3 seconds to signal a full batch tray.

(a) Draw and label the ATtiny85 pinout diagram showing the pins used. (4 marks)

(b) Draw the circuit diagram for the proximity sensor, LED, buzzer and reset pushbutton. (5 marks)

(c) Draw a flowchart for the soap-counting, batch-of-50 alert, and manual reset logic. (5 marks)

(d) Write a complete Arduino C/C++ sketch implementing the full soap-counting and batch-alert scenario, including the debounced manual reset button. (6 marks)

END