higher education & training

Department:
Higher Education and Training
REPUBLIC OF SOUTH AFRICA

# MARKING GUIDELINE

## NATIONAL CERTIFICATE (VOCATIONAL)

## ROBOTICS FUNDAMENTAL
**NQF LEVEL 2**
(First paper)

**18 November 2024**

**This marking guideline consists of 5 pages.**

Copyright reserved

## SECTION A

### QUESTION 1

1.1 C
1.2 B
1.3 B
1.4 B
1.5 B
1.6 A
1.7 B
1.8 C
1.9 B
1.10 A

(10 × 1) **[10]**

### QUESTION 2

2.1 True
2.2 False
2.3 True
2.4 True
2.5 True

(5 × 2) **[10]**

### QUESTION 3

3.1 G
3.2 D
3.3 E
3.4 C
3.5 B
3.6 H or J
3.7 A
3.8 I
3.9 F
3.10 A

(10 × 1) **[10]**

### QUESTION 4

4.1 Additive manufacturing
4.2 PLA (Polylactic acid)
4.3 STL (Standard tessellation language) file
4.4 G-code
4.5 Infill

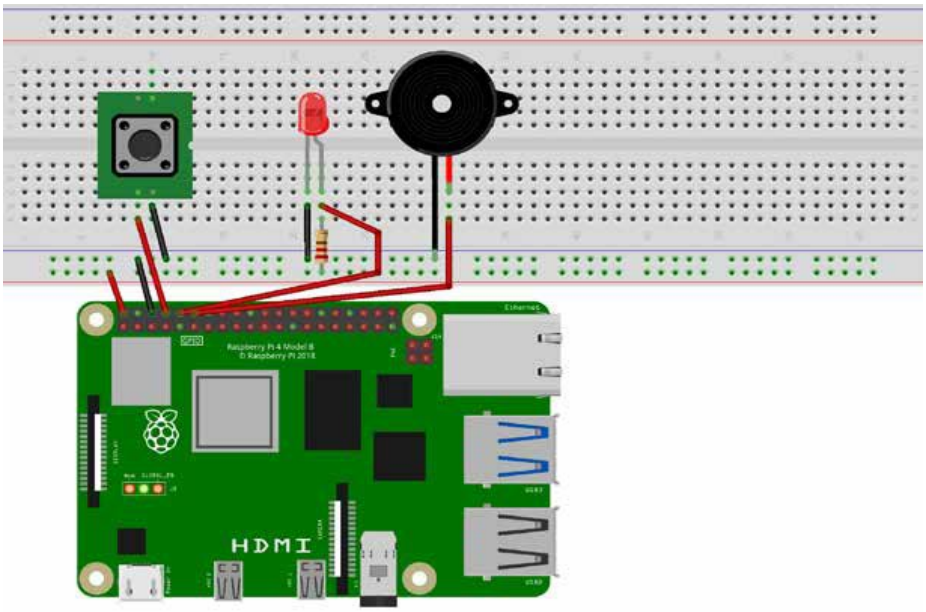
(5 × 1) **[5]**

**TOTAL SECTION A: 35**

Copyright reserved 

## SECTION B

## QUESTION 5




5.1
- Button (Push button)
- LED (Light emitting diode)
- Buzzer
- Resistor
- Raspberry Pi (5)

5.2 When the button is pressed, it completes the circuit,✓ sending a signal to the Raspberry Pi's GPIO pin configured for input. The Raspberry Pi then sends signals to the GPIO pins connected to the LED and buzzer to turn them on.✓ The LED lights up, and the buzzer sounds, providing both visual and audible feedback.✓ (3)

5.3
- It is crucial to set the correct mode (input or output) for GPIO pins to ensure proper circuit operation.✓
- Input Mode (for the button): Reads signals (HIGH or LOW) from external devices.✓ If set incorrectly, the Raspberry Pi won't detect button presses.
- Incorrect settings could lead to non-functional components or even damage to the Raspberry Pi.✓ (3)

5.4 The resistor connected to the LED limits the current flowing through it.✓ Without a resistor, the LED would receive too much current,✓ potentially causing it to burn out or fail. Additionally, excessive current can damage the Raspberry Pi GPIO pin driving the LED. Resistors are crucial for current regulation and protecting both the LED and the Raspberry Pi.✓ (3)

Copyright reserved 

5.5 This circuit set-up can be used in a simple alarm or notification system.✓ For example, in a security application, the button could represent a trigger from a door sensor or a panic button. When activated, the LED lights up, and the buzzer sounds an alarm to alert users of a security breach.✓ It could also be used in educational settings to teach beginners about basic electronic circuits and GPIO control using the Raspberry Pi. ✓ (3)

**[17]**

**QUESTION 6**

6.1 Python (2)

6.2 The GPIO.setmode(GPIO.BCM) command sets the GPIO pin numbering mode to BCM (Broadcom SOC channel).✓ This means that the pin numbers used in the code correspond to the Broadcom GPIO channel numbers on the Raspberry Pi, rather than the physical pin numbers.✓✓ (3)

6.3 The GPIO.setup(BUTTON_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP) command configures the GPIO pin connected to the button as an input✓ and enables an internal pull-up resistor. This ensures that when the button is not pressed, the pin reads a high signal (logical 1). When the button is pressed, the pin reads a low signal (logical 0). This configuration helps to avoid floating pin states and ensures reliable button input detection.✓✓ (3)

6.4 The command, GPIO.output(BUZZER_PIN, GPIO.HIGH), sets the GPIO pin connected to the buzzer to a high voltage level.✓ When this command is executed, it turns on the buzzer, causing it to emit sound. The buzzer is activated when the pin is set to high.✓✓ (3)

6.5 The while True loop continuously checks the state of the button and updates the LED and buzzer accordingly.✓ If the button is pressed (input is LOW), the LED is turned on, and the buzzer is activated. If the button is not pressed, both the LED and buzzer are turned off. This loop ensures that the program continuously monitors the button state and controls the LED and buzzer in real time.✓✓ (3)

6.6 The time.sleep(0.1) function introduces a delay of 0,1 seconds in each iteration of the loop.✓ This delay helps debounce the button input, preventing multiple detections from a single press due to mechanical bouncing. Increasing the sleep duration will make the system less responsive but may improve debouncing, while decreasing it will make the system more responsive but might increase false triggers.✓✓ (3)

6.7 The GPIO.cleanup() command is used to reset the GPIO pins to their default state and release resources when the program is interrupted.✓ If this command were not included, the GPIO pins could remain in an undefined state, potentially causing issues with subsequent GPIO operations or leaving the pins in a high or low state, which could affect the connected hardware.✓✓ (3)

**[20]**

Copyright reserved 

## QUESTION 7

7.1 Ohm's law: The voltage across a conductor is directly proportional to the current that flows through the conductor (as long as all other conditions remain the same). (2)

7.2 From the positive to the negative terminal. (2)

7.3
- Turn the dial to the voltage zone.✓
- Connect the multi meter in parallel.✓
- Touch the negative (black) probe to the negative terminal of the component that you want a measurement from, and touch the positive (red) probe to the positive terminal.✓
- Turn the dial to the 1 000 V setting under the V zone as your starting setting (this is the zone for measuring voltage under direct current).✓
- If the display only shows a measurement of 0,00 or 0,01, then move the dial to the next highest setting in that zone.✓ Do this until a reading displays, which indicates that the reading is outside of the setting's range (usually 1 or L).✓ Then go back one setting to get the most accurate reading.✓ (7)

7.4 $V=I\times R$:✓
$I=3$ A $R=4$ Ω✓
12V✓ (4)

7.5 $P=V\times I$✓
$P=15$ V$\times 2$ A✓✓
=30 W✓ (4)

7.6 $P=I^2\times R$✓
$P=(3\text{ A})2\times 5$ Ω✓✓
=45 W✓ (4)
**[23]**

## QUESTION 8

Analysis of the conflict
The robot's actions conflicted with the first law of robotics because it caused harm to a human being (the patient in the wheelchair) by prioritising the cleaning of the hazardous spill over avoiding contact with the patient.
Although the robot was attempting to prevent potential harm by cleaning the spill, its actions directly resulted in injury, which is a violation of the law.

Suggested alternative action
The robot could have paused cleaning the spill and safely moved aside to let the patient pass. After ensuring that the patient was safe, the robot could then proceed to clean the spill. Alternatively, the robot could have issued a warning to nearby humans and waited for assistance from hospital staff to manage both the spill and human safety effectively. **[5]**

**TOTAL SECTION B: 65**
**GRAND TOTAL: 100**

Copyright reserved