## Basics

1. (2pts) What is the name of the Python interpreter that runs on the Raspberry Pi?
   1. python
   2. python3
   3. py
   4. raspi
2. (2pts) What is the difference between a variable and a constant in Python?
   1. A variable can change its value, while a constant cannot.
   2. A variable is a name for a value, while a constant is a value itself.
   3. A variable is declared with var, while a constant is declared with const.
   4. A variable is written in lowercase, while a constant is written in uppercase.
3. (2pts) What will the following Python code print?  
   x = 5  
   y = 10  
   z = x + y  
   print(z)
   1. 5 + 10
   2. x + y
   3. z
   4. 15
4. (2pts) What is the purpose of the for statement in Python?
   1. To define a function
   2. To create a loop
   3. To execute code conditionally
   4. To print a message
5. (2pts) Which data type is used to store text values in a single variable in Python?
   1. int
   2. float
   3. string
   4. list

## Python and Raspberry Pi

Consider the following Python program:

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BCM)

pins = [2, 3, 4, 14, 15]

# Initialize pins as outputs

for pin in pins:

GPIO.setup(pin, GPIO.OUT)

i = 0

try:

while True:

if i >= 32:

i = 0

binary = format(i, '05b') # convert

# output

for i in range(5):

GPIO.output(pins[i], int(binary[i]))

time.sleep(1) # wait

i = i + 1

finally:

GPIO.cleanup()

1. (2pts) What is the name of the Python module that is used to control the GPIO pins on the Raspberry Pi?
   1. RPi.GPIO
   2. GPIOZero
   3. PiGPIO
   4. RaspberryPi.GPIO
2. (2pts) From the above code, what is the meaning of the argument GPIO.BCM in the setmode() function?
   1. It sets the pin numbering scheme to use the Broadcom SOC channel numbers
   2. It sets the pin mode to use binary coded modulation for output signals
   3. It sets the pin configuration to use bidirectional communication mode for input and output
   4. It sets the pin function to use board control mode for special features
3. (4pts) If you wanted to change how fast the programs performs its task, what would you change in the above code?
4. (4pts) Explain how the above program uses a try-finally block and why it is important.

## MQTT

1. (2pts) What is MQTT and what does it stand for?
   1. Machine Query Transfer Technology, a protocol for sending queries to machines
   2. Message Queuing Telemetry Transport, a protocol for lightweight messaging between devices
   3. Multi-Threaded Task Tracker, a protocol for managing concurrent tasks on devices
   4. Mobile Quality Testing Tool, a protocol for testing the quality of mobile applications
2. (2pts) What are the three components of an MQTT network?
   1. Publisher, subscriber, and broker
   2. Client, server, and topic
   3. Sender, receiver, and channel
   4. Device, cloud, and service
3. (2pts) What are the two types of messages in MQTT?
   1. Request and response
   2. Publish and subscribe
   3. Data and control
   4. QoS and retain
4. (2pts) What is the purpose of a topic in MQTT?
   1. To identify the message content and filter the messages for subscribers
   2. To encrypt the message payload and secure the communication between devices
   3. To compress the message size and optimize the bandwidth usage for publishers
   4. To acknowledge the message delivery and confirm the quality of service for brokers

## Python, Raspberry Pi, and I2C

Consider the following Python program:

import smbus

import time

class TemperatureSensor:

def \_\_init\_\_(self, i2c\_bus, i2c\_address):

self.i2c\_bus = i2c\_bus

self.i2c\_address = i2c\_address

def get\_temperature(self):

data = self.\_read\_sensor\_data()

temperature = ((data[0] & 0x0F) << 8) | data[1]

temperature /= 16.0

return temperature

def \_read\_sensor\_data(self):

try:

bus = smbus.SMBus(self.i2c\_bus)

data = bus.read\_i2c\_block\_data(self.i2c\_address, 0x5, 2)

return data

except Exception as e:

print(f"Error reading data from I2C sensor: {e}")

return None

i2c\_bus\_number = 1

i2c\_device\_address = 0x18

sensor = TemperatureSensor(i2c\_bus=i2c\_bus\_number,i2c\_address=i2c\_device\_address)

while True:

print(f"Temperature: {sensor.get\_temperature():.2f}°C")

time.sleep(1)

1. (2pts) What is the purpose of the above program?
   1. To read analog input from a temperature sensor connected to the RPi via I2C and print the value
   2. To write analog output to a LED connected to a DAC module via I2C and control the brightness
   3. To read digital input from a button connected to a GPIO pin and print the state
   4. To write digital output to a buzzer connected to a PWM pin and control the frequency
2. (2pts) From the above code, what is the address of the I2C device?
3. (4pts) If you wanted to use a different I2C device for this program, what would you change and how would you determine the address of the new device?
4. (6pts) Explain how the above program calculates the temperature from the value provided.

## Arduino/ESP32

For the final 2 questions, you may refer to the following code:

#include <WiFi.h> //wifi connection

#include <PubSubClient.h> //MQTT messaging

#include <Arduino.h> //Input and output

// WiFi + MQTT information

const char\* ssid = "iot\_wireless"; // The name of the WiFi network

const char\* psk = "Unsecure!"; // The WiFi network passkey

const char\* mqtt\_server = "your-pi.local";

const int mqtt\_port = 1883;

const char\* mqtt\_topic = "house/greeting";

String mqtt\_client\_name = "ESP32\_"; //MQTT client name prefix

WiFiClient espClient; // Create a WiFiClient object

PubSubClient mqtt\_client(espClient); // Create a PubSubClient object

// Timer for publishing every 5 seconds

unsigned long previousMillis = 0;

const long interval = 5000;

void setup() {

Serial.begin(115200); //Start Serial communication.

uint8\_t mac[6]; char macStr[18]; //set a unique client name

esp\_read\_mac(mac, ESP\_MAC\_WIFI\_STA);

snprintf(macStr, sizeof(macStr), "%02X:%02X:%02X:%02X:%02X:%02X", mac[0], mac[1], mac[2], mac[3], mac[4], mac[5]);

mqtt\_client\_name = mqtt\_client\_name + macStr;

WiFi.begin(ssid, psk);//Connect to WiFi

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

Serial.println("Connecting to WiFi...");

}

Serial.println("Connected to WiFi");

mqtt\_client.setServer(mqtt\_server, mqtt\_port); //set MQTT broker info

}

void loop() {

if (!mqtt\_client.connected()) { mqtt\_connect(); }

// Publish a message every 5 seconds

unsigned long currentMillis = millis(); //current time

if (currentMillis - previousMillis >= interval) {

previousMillis = currentMillis;

String message = "Hello from " + mqtt\_client\_name;

mqtt\_client.publish(mqtt\_topic, message.c\_str());

Serial.println("Message sent: Hello from " + mqtt\_client\_name);

}

mqtt\_client.loop();//for PubSubClient object message processing

}

void mqtt\_connect() {

while (!mqtt\_client.connected()) {// Loop until we are connected

Serial.println("Attempting MQTT connection...");

if (mqtt\_client.connect(mqtt\_client\_name.c\_str())) {

Serial.println("Connected to MQTT broker.");

} else {

Serial.println("Connection failure. Waiting 5 seconds.");

delay(5000);

}

}

}

1. (2pts) What is the purpose of the WiFi library above?
   1. To manage Wi-Fi connections
   2. To handle analog input from sensors
   3. To implement MQTT communication
   4. To control LED lighting
2. (2pts) Why is it logical to use the MAC address to produce a unique client name to connect to an MQTT broker?
   1. MAC addresses are unique.
   2. MQTT Brokers will refuse connecting two devices with the same name.
   3. The MAC address of the device is available without complex operations.
   4. All of the above.