**Assessment type (🗹):**

Questioning (Oral/Written)

Practical Demonstration

3rd Party Report

Other – Project/Portfolio (*please specify)*

Practical Portfolio work with embedded knowledge questions

**Assessment Resources:**

|  |
| --- |
| The base requirements this assessment task include:   * Web server, Python interpreter and database server * IDE or editor for developing Python programs (only PyCharm supported by the college) * Raspberry Pi with SenseHat or other IoT devices, like Arduino Uno or ESP32 * Access to Office 365 & Microsoft Word * Report Template (Portfolio: AT2 POR-Task-4 Template) as supplied   Use of some of these items may not occur in this part of the assessment task. |

**Assessment Instructions:**

|  |
| --- |
| Follow the steps listed in this assessment item.  Submission of the documentation, code, and associated items is at the end of each part of the portfolio.  Each part of the portfolio has a deadline for submission.  It is advantageous to you to attempt to meet the deadline provided.  **GENERAL INSTRUCTIONS**  Use this document for all your answers and evidence except for final code submission.  We provide a document template for your answers.  Download the AT2 POR-Task-4 Template from Blackboard, and then DOUBLE CLICK it to create a new blank document for your answers.  Save the file as:   * XXX-POR-Task-4.docx   Replacing the XXX with your initials.  For example, Adrian Gould would use AG-POR-Task-4.docx for his submitted filename.  **ANSWERING QUESTIONS**  When a step includes a question, you must attempt to answer it.  There is a minimum and maximum number of SENTENCES to use for each answer.  All answers must be in complete sentences unless indicated.  If required, make sure to add any code you’ve written in a separate file to your submission. DO NOT put code in a Word document.  Short snippets of code (up to about 10 lines) are allowed to be used as evidence if required.  **SOURCES OF INFORMATION**  In industry, it is good practice to keep track of where information was obtained. This is especially true if it is a written document, or even code.  If you answer any questions using information from web sites, please include the site name and URL (Web site address) after the answer. Likewise, include the title and author for books and magazine articles. For example:  • RS Electronics Ltd: <https://au.rs-online.com/>  • Slack API Documentation, Users List Method: <https://api.slack.com/methods/users.list>  **CODE STORAGE**  We advise that you create a GIT repository on GitHub and use this to store a copy of your work.  You may also use OneDrive within your college Office365 to store a backup of your code or keep a copy on a USB thumb drive. |

**Assessment Instrument:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | Step | Task to perform | | --- | --- | | 00 | Create Evidence Document Make sure you have followed the instructions on creating the answer document, as given in the General Instructions.  Familiarise yourself with the content and document your progress in this assessment.  Make sure that you complete the title page of the document. IMPORTANT: This assessment should be done on actual ESP32 hardware, which will be provided.  You can access the ESP32 API Reference here:   * <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/gpio.html>   This link specifically points to GPIO & RTC GPIO, but it will give you easy access to all the other areas of the API too.  If the question requires you to submit code with your final submission, you may add it as a zip-file OR in a separate folder. | |  | *This space left intentionally blank.* | | 01 | A Familiar Starting Point For this task, you will be provided (in class) with a simple electronic circuit built around an ESP32 microcontroller. It is the same circuit that we used in the previous portfolio task.  You should also be using the final code of Task 3 from this point onwards.  The schematic for it looks like this (simplified):    No evidence required for this step. | | 02 | Prepare the environment For this task, you will be modifying the program so that it sends updates of the switch’s status to an external service on the internet.  For this purpose, you will be using MQTT.  The external MQTT service that you will be using is Adafruit.io.  It provides a very easy way of accepting data from IoT devices and creating dashboards to visualise the data.  Naturally, you will be using Wi-Fi to connect to the internet.  You should have already created an account with Adafruit.io, but in case you haven’t, refer to the learning materials in Blackboard or ask your lecturer.  Create a new **Feed** in your account.  Choose a name and document it in the A02 space below, as you will need it in the following steps.  In the Arduino IDE (which is used to program the ESP32), check that the library Adafruit\_MQTT is installed. Install the library if it is not.  Provide screenshots of your Adafruit.io account with the new Feed and the Arduino IDE that shows Adafruit\_MQTT has been installed. | | A02 | Adafruit Details  |  |  | | --- | --- | | Account Name | winroof | | Feed Name | intermediate-iot-portfolio |   Screenshots in following space. | | A02 | Screenshot of your Adafruit Account Details | | 03 | Prepare the program You will need to prepare your program to use WiFi and MQTT. For that purpose, it will need to include the relevant header files.  Follow these instructions:   * Add the following lines to your program:   #include <WiFi.h>  #include <Adafruit\_MQTT.h>  #include <Adafruit\_MQTT\_Client.h>   * Make sure you get the capitalisation correct, otherwise the compiler will not be able to find the files. * Hit the ‘Compile’ button (the checkmark). If you see any errors fix them and try again. * Make screenshots of the source file showing the includes and the output of the compilation process. | | A03 | Screenshot(s) of the source file and compilation | | 04 | Create Network Configuration File To successfully connect to the internet and send data to the external MQTT service, you will need to provide the program with credentials.  You will not hardcode those credentials in the source file.  Instead, you will create a separate configuration file (config.h), that should not be added to the repository!  You may even decide to add the .gitignore file in the appropriate folder to ignore the configuration file. That way you won’t even commit it accidentally.  You will need credentials for Wi-Fi (SSID and password) and for MQTT.  The Wi-Fi credentials will be provided to you by your lecturer.  The MQTT credentials can be obtained from your Adafruit.io account.  Follow these instructions:   * Create a new file called Config.h. Please note that AIO is short for Adafruit IO. * Add the following line to main.ino, just below the other includes:   #include “config.h”   * Add the following lines to the file and substitute the strings “<…>” with the appropriate value:   #define WIFI\_SSID “<wifi\_ssid>”  #define WIFI\_PASS “<wifi\_pass”>  #define AIO\_SERVER “<aio\_server>”  #define AIO\_SERVERPORT“<aio\_serverport>”  #define AIO\_USERNAME “<aio\_username>”  #define AIO\_KEY “<aio\_key>”  #define AIO\_FEED “<feed\_name>”   * Create a zip-file of the current state of your project and add this to your final submission. * Clearly indicate that it pertains to question 4 (e.g., add -v4). | | 05a | Connect to the Wi-Fi You will now write the code to connect to the Wi-Fi and test it before you connect to the external MQTT service.  You will create a few different functions to enable the connectivity. Start with connecting to the Wi-Fi access point.  Follow these instructions:   * Create the following function at the end of your source file:   void wifi\_connect() {  **WiFi.begin(WIFI\_SSID, WIFI\_PASS);**  Serial.print("Connecting to ");  Serial.println(WIFI\_SSID);  **while (WiFi.status() != WL\_CONNECTED) {**  delay(500);  Serial.print(".");  }  Serial.println();  Serial.println("WiFi connected");  Serial.print("IP address: ");  Serial.println(WiFi.localIP());  }  The most important parts have been highlighted.  Add the following lines to the setup() function, just after the call to flash\_led().  wifi\_connect();  flash\_led(); | | 05b | Verify Connectivity to Wi-Fi Because connecting to the Wi-Fi can be a (relatively) slow process, we want to have an indication what’s going on.  We can use the Arduino IDE’s Serial Monitor now, but not when the system is operational.  For this reason, we will flash the LEDs twice:   * once after the device has woken up, and * once after it has successfully connected.   (There is another reason we must call this function after connecting to the MQTT service.)  Compile and flash your program.  Solve any errors you may encounter.  Use the Serial Monitor to check that the device connects to the Wi-Fi and observe the LEDs flashing twice. Video Evidence Take a short video of the circuit in action (no more than 15 seconds).  Ensure you speak your name clearly when videoing the circuit.  Save this video as AT2-T4-05-XXX.mp4 (mp4, vog, avi or similar allowed). Replace XXX with your initials. Questions Answer the questions below:   * In your own words, what does the function wifi\_connect do? * What is one obvious disadvantage of the current implementation of the wifi\_connect function?  Create Zip v5 Add the current project state (zip-file) to your submission. Clearly indicate it is for question 5 (e.g., add -v5). | | A05b | Q: What is the purpose of wifi\_connect? In your own words, what does the function wifi\_connect do?  1-5 sentences, code allowed.    The function ‘wifi\_connect()’ establish a wifi connection using the provided credentials, which are ‘WIFI\_SSID’ and ‘WIFI\_PASS’ , and print out the connection status, such as the SSID and IP address. | | A05b | Q: What is one obvious disadvantage of the wifi\_connect method? 1-5 sentences, code allowed.  One obvious disadvantage of the ‘wifi\_connect()’ is that it blocks the execution of other code until the wifi connection is established. This code block enters the ‘while loop’ where it continuously checks the status of the wifi connection and print dots to indicate the ongoing connection process. During this time, it is not able to perform any other code. | | 06a | Connect to MQTT In the previous step, you have successfully connected your ESP32 device to the Wi-Fi.  In this step, you are going to connect the device to an external MQTT service, for which you have provided the credentials in the configuration file Config.h.  You will do this in two steps. This is step 1.  Follow these instructions:  Add the following function to your source file (at the end – you may not need the line breaks shown below).  void mqtt\_update(const int pin\_state) {  WiFiClient client;  // Setup the MQTT client class by passing in the WiFi client and  // MQTT server and login details.  Adafruit\_MQTT\_Client mqtt(&client, AIO\_SERVER, AIO\_SERVERPORT, AIO\_USERNAME, AIO\_KEY);  wifi\_connect();  if (mqtt\_connect(mqtt)) {  Serial.println("Connected!");  } else {  Serial.println("MQTT Not Connected! Going back to sleep.");  }  }  Please note that this function takes the state of the input pin (the switch) as an argument and calls the wifi\_connect function that we defined earlier.  The routine also calls a function mqtt\_connect, which we will create next. | | 06b | MQTT Connection (continued) Add the following code to your source file, above the one you created just now.  bool mqtt\_connect(Adafruit\_MQTT\_Client& mqtt) {  Serial.print("Connecting to MQTT… ");  uint8\_t retries = 3;  int8\_t ret;  while ((ret = mqtt.connect()) != 0) {  Serial.println(mqtt.connectErrorString(ret));  if (retries-- == 0) {  Serial.println("Stopping retries…");  return false;  }  mqtt.disconnect();  Serial.println("Retrying MQTT connection in 5 seconds…");  delay(5000); // wait 5 seconds  };  return true;  }  Finally, in the setup function, change the line that says  wifi\_connect();  to  mqtt\_update(pin\_state);  Compile your program and test it.  Fix any problems you find along the way. | | 06c | MQTT Connection Questions Answer the following questions:   * How does the function mqtt\_connect handle the connection? * How is that an improvement over the wifi\_connect function’s way?   1-5 sentences each, code allowed. | | A06c | Q: MQTT Connection Method: How does the function mqtt\_connect handle the connection? ‘mqtt\_connect’ function handles the connection to the MQTT broker by continuously attempting to connect until successful or until it runs out of retries. Also, it provides feedback on the connection status through serial monitor output and return ‘true’ if the connection is successful and ‘false’ otherwise. | | A06c | Wi-Fi Connection Method: How is that an improvement over the wifi\_connect function’s way? ‘mqtt\_update’ function represents an improvement over the ‘wifi\_connect’ function by encapsulating the MQTT connection logic to make the code easier for better understanding and maintaining and by sharing a role of mqtt connection and wifi connection to enhance code readability and maintainability. | | 06d | MQTT Connection Evidence Add the current project as a zip-file to your submission.  Indicate that this is for question 6 (e.g., add -v6) | | 07a | Update the feed This is the second and final step to connect to MQTT.  Follow these instructions.  To update your MQTT feed, you will need to send relevant data to the MQTT service.  You will do this by updating the function mqtt\_update:  void mqtt\_update(const int pin\_state) {  WiFiClient client;  // Setup the MQTT client class by passing in the WiFi client and  // MQTT server and login details.  Adafruit\_MQTT\_Client mqtt(&client, AIO\_SERVER, AIO\_SERVERPORT, AIO\_USERNAME, AIO\_KEY);  // Setup a feed called‘feed’for publishing  Adafruit\_MQTT\_Publish feed = Adafruit\_MQTT\_Publish(&mqtt, AIO\_USERNAME "/feeds/" AIO\_FEED);  wifi\_connect();  if (mqtt\_connect(mqtt)) {  Serial.println("Connected!");  if (!gdoor.publish((uint32\_t)pin\_state)) {  Serial.println(F("Failed"));  } else {  Serial.println(F("OK!"));  }  } else {  Serial.println("MQTT Not Connected! Going back to sleep.");  }  }  The changes have been marked in red. | | 07b | Update the Feed (Continued) Compile your new program.  Fix any problems you encounter and flash the ESP32 device.  Test that you can connect to MQTT and that your feed updates correctly. Important: MQTT Update Fails? If your program fails to update the MQTT feed and you have checked the credentials and everything related to that, you may need to check that the function flash\_led is called after mqtt\_update.  The former contains a very short delay that is needed for the MQTT (feed) update to succeed.  Without this delay, the feed will not be updated at all even if the connection to the MQTT service succeeded. | | 07c | Update the Feed (Evidence)Video Evidence Take a short video of the circuit and the feed updating in action (no more than 20 seconds).  Ensure you speak your name clearly when videoing the circuit.  Save this video as AT2-T4-05-XXX.mp4 (mp4, vog, avi or similar allowed). Replace XXX with your initials. Code Evidence Add the current state of your project to your submission. Clearly indicate it pertains to Question 7. | | 08a | Improving the solution with Multiple Connection Attempts While you have now created a program that successfully updates the status of the input pin with an external MQTT service, there are a few improvements that you can make.  Follow these instructions:  Compare mqtt\_connect with wifi\_connect.  The former gives up connecting after a few tries.  Look at the implementation and update the latter to provide the same functionality. | | 08b | Update mqtt\_update so it uses the return value of wifi\_connect.  Add a function log\_pin\_state (which takes pin\_state as the only argument), which prints the state of the pin to the Serial Monitor.  Call the method from somewhere in your program, but make sure it’s only called once for a wake-sleep cycle.  (Resist the temptation to call it from within flash\_led.)  (Optional) Feel free to add comments where you seem fit. | | A08 | Final Improvements Evidence Create a zip-file of your current project and upload as part of the final submission to Blackboard. | | END | Submission of Portfolio Work To submit the portfolio, do the following:   * Save the document with your answers as a MS Word file (.docx). * Put all your code inside a zip-file. * Open Blackboard, and locate the AT2 Portfolio Task 1 assessment * Open the assessment and upload the original word-processed document. * Click submit.   Whilst there is no need to use any other word processing software as you have access to Office 365 for free as a student, if you use Apple Pages, or Open Office, we will then require you to upload the original file AND a PDF version. | |

# Appendix A: Code Style Guidelines

Your code will follow the PEP 8 standard.

Readability Counts  
- Zen of Python

Explicit is better than implicit.  
- Zen of Python

Other code standards available in the Presentation, “Python Coding Standards for North Metropolitan TAFE”.