Introduction to Computer Systems
COSC 2473
Assignment 2 - Project

Assessment Type	Individual assignment. Submit online via Canvas→Assignments→Project Assignment. Marks awarded for meeting requirements as closely as possible. Clarifications/updates may be made via announcements/relevant discussion forums.
Due Date	Week 12, Sunday 18th October 2020, 11:59pm
Marks	30

A. Overview

For this assignment you are required to build a system using BBC Micro:bit device. You will use the Micro:bit to gather data automatically using its sensors; and make it available on the internet.

You are to deliver this data in a rigorous fashion to a PC attached via USB using the onboard Python, and then to run a local Python server on the PC with appropriate web pages to serve the result locally. Remote access to the PC is not required but would be simple to add.

This assignment is designed to be as open as possible, and you are encouraged to be creative and innovative. This might be in the data collected, the manner of collection, the use of the data on the PC, etc. Marks will be awarded for interesting solutions; however, most of the marks are for the use of the techniques described above and below.

Where an instruction might not be clear, you are to make a decision of what is reasonable to work, and justify this reasoning in the report, and video.

B. Learning Outcomes

This assessment relates to all the learning outcomes of the course which are:

1. critically review and re-design existing computer system specifications for meeting stated performance criteria.
2. Be able to use small IoT devices to capture sensory information and interact with other hardware.
3. Describe the layered networking model, basic networking protocols and ways in which data is encoded and transmitted
4. Analyze and determine an appropriate enterprise networking system and secure web server configuration based on stated user needs;

C. Assessment details

1. <u>Data Sampling</u>
    1. Take a measurement of at least two sensor variables that are plausibly connected (e.g. Temperature and light level), and regularly collect their measurements, storing them internally until the buffer is more than ¾ full, and then dump the contents to the PC.
       For example…
        - You need to simulate the changes for the sensors including light, temperature, accelerometer, and compass and explicitly describe their experimental setup.
        - Take the micro: bit into warmer and colder places and see how the temperature readings change.
        - Try out the built-in light sensing capability of the Micro:bit,
        - Flip the board;

- Test the kind of values that you get for the compass heading.

2. At the start of collection, the PC is to provide a time/date stamp to all devices, and they are to resynchronize every time a data transfer has successfully completed, and new data collection has started. If there is a significant difference, between internal date and synchronize-date, this is to be reported by the PC to the data files before re-synchronization to all data files prior to dumping the data onto them.
3. The sampling frequency of data is determined by the application (eg. temperature will not change frequently – unless it is a thermostat).

2. Display.
   1. Whenever a sample is taken, the 5x5 display is to increment a single digit.
   2. When the device is downloading, the display is to show a download icon.
   3. When the collected data is confirmed to have been delivered and is cleared, then the display shows a tick mark. If a download error occurred, a cross is shown on the display.
3. Data Transmission
   1. The data is to be labeled, and the time/date of collection is to also be recorded and dumped to the PC. Suggested data format is JSON.
   2. A suitable protocol is to be developed to ensure reliable transmission of the data. For example, a simple stop and wait ARQ.
   3. The data in the onboard buffer of the connected Micro:bit is not to be deleted until the data has been confirmed to have arrived. Matched checksums might be sufficient.
4. PC Data
   1. On the PC, the data can be stored as TAB-or Comma delimited data in a file in a fixed location.
   2. The data is appended to any existing data in the file. This must be done at the PC end, in order to avoid running out of space, and the fact that file open in append mode is not possible on Micro:bit.
   3. The data is to include the time/date stamp, device# doing the collection, the sensor variable name, and value.
   4. Separate files are maintained for each sensor variable. This makes it easy to plot these files using Excel, or some such. As an extension. another Python package could run on the server to provide plot images.
5. Web server
   1. The Python web server is to point to the folder containing the files. A web user could then select the file and see text data. An optional extension might be to store the data in HTML format but beware of table-end delimiters.

There will be lab assistance for the following

- How to gather data into a Python List including time/data stamp
- How to transfer data from device to P using serial connection C
- How to set up a Python web server and have it generating content for certain URLs using handlers, not just delivering files from a predetermined location.
- Some other LinkedIn Learning videos.

D. Submission

You must upload to Canvas all the materials necessary in order to reproduce the content created.

Include a PDF report which details how to reproduce the sampling including some sample data and a corresponding graph in order to demonstrate something about the data you collected. Also include a detailed description of how you arranged the data collection.

Provide a video presentation

- using Microsoft Teams wherein you describe your work. Teams will create a shareable recording in Microsoft Streams

- As a guide to content, you should present evidence in the video of all the dot points achieved.

  - For example, show the data being collected and the data shown on a browser pointing to http://localhost/path-to-folder/.

- A video can be most easily made by creating a team's session and sharing screenshots where needed.
- Note that these videos should include your faces in the corner, and at least one section where you can be fully seen handling your device and show it collecting data,
- Submit the URL of the Microsoft Streams recording, and ensure the permissions allow all the teaching staff access.

After the due date, you will have 5 business days to submit your assignment as a late submission. Late submissions will incur a penalty of 10% per day (pro-rata, so ½ day late = 5%), rounded to nearest percentage below. After these five days, Canvas will be closed, and you will lose ALL the assignment marks.

E. Marking Guidelines

Please check Rubric before submission for detailed marking

- Implementation : 20/30;
- Documentation: 7/30
- Innovation and aesthetic 3/30;
- Report 2/30;Video Presentation: Total scaled to 50% if no video present.

Approximate Indicative Marking Levels

PA – Sampling code on Micro:bit without any time/date stamp. Continuous sampling with output as print statements collectible via USB serial connection. Saving output to a file.

CR – As above, but with PC providing the time/date information via serial connection.

DI . – As above, but with functional web Python server. Collected data to be redirected to a file that the server can serve. Should be CSV file containing [sensor, sample Count, timeDate, value]. File must be opened in append mode.

HD – As above but with each sensor having their own file, or a CSV containing both sensor values on one line. Same format for each. Web server able to serve all these.

One Optional extra mark given for :

- HTML versions of the data files, as an HTML table.
    - A CSV → HTML handler in the server can automate a table output.
- Remote start / restart This can be triggered by a special URL causing the web server handler for that URL to send a serial message to the device with the time/date. The device is always waiting for input if not sending output. Example: http://localhost:port/restart.
    - Other utility commands can be handled the same way, using special URLs and handlers to deal with them.

F. Assessment declaration:

When you submit work electronically, you agree to the assessment declaration:

https://www.rmit.edu.au/students/student-essentials/assessment-and-exams/assessment/assessment-declaration

Academic integrity and plagiarism (standard warning)

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods,
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to https://www.rmit.edu.au/students/student-essentials/rights-and-responsibilities/academic-integrity