

A10 Assignment – ESE516-SPRING 2022

DUE DATE: Wednesday April 20 2022 before 11:59pm EST (By almost midnight). To be submitted on Google Drive of Team Folder on a folder called A11. GROUP PROJECT

Remember: Please submit your complete Atmel Project on Google Drive. The project must be complete (must have everything so we can compile it!).

1. Starter Slides [0 points]

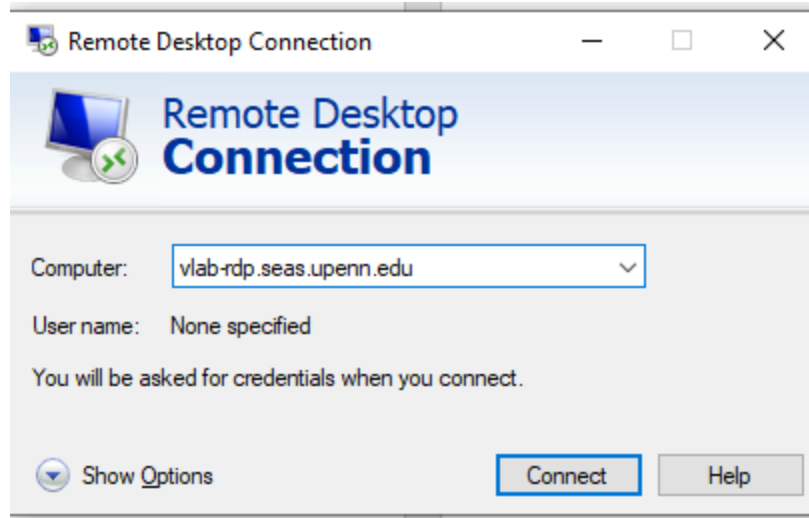
Follow the powerpoint “Cloud Starter.pptx” and “Cloud Starter Annex.pptx” (in that order) to set up your system. This will guide you on how to set up your IBM cloud. You can find it on the A10 assignment folder.

Also, download the A10 started project, which will be an useful point to start out project. A10 contains an unfinished bootloader – please replace the folder “SD_MMC_EXAMPLE_Bootloader_ESE516_SPRING2019” with your A9 bootloader.

You will also need a place to host files online. You could find a free host, or use the school server to host your files. Since most of your implementations involve hosting multiple files on the http server, I would highly recommend you all to use the html folder that is provided to you along with your SEAS account. You should be able to access it at <https://www.seas.upenn.edu/~username/>.

If you do not have this folder setup in your account, follow the instructions at <https://cets.seas.upenn.edu/answers/html-permissions.html> to do it yourself. Make sure that you run all the commands listed under "Using UNIX" section to grant the necessary file permissions.

To have access to your folder, you can do remote desktop to the Penn virtual lab, and place the files you want to host on the HTML folder on your server.



2. OTAFU [100 points]

The starter project has a CLI command, “fw”, that downloads a file from a server. Modify the code on this CLI command to download a file set in your server that is a firmware update. Modify the code in a way that you can make a complete OTAFU: The device **downloads an image from the internet, saves it into SD card, adds any flags that your bootloader needs**, and then **resets** the MCU to perform a FW update.

To deliver: A short video showing that your device can update its firmware via the internet.

Note: You can reset the MCU with the command “`system_reset()`”.

3. Node Red Design [100 points]

Before doing any coding, design the MQTT topics and program flow your system will use. Present a document explaining the following:

- General program flow of your system. What communication does the device need to send to the cloud? What does the cloud do with this information? What does it send back?

- List all the topics your system will use. Describe for each one the information that is sent on each topic (is it an integer? A Boolean? An array of integers? Is it a JSON string?)
- Describe for each topic in your system: Who generates the data? Who subscribes to each topic? Etc.
- Present how you would divide your application into threads, how the threads would communicate, etc.

To turn in: A small document that explains all of the above. A reader should be able to understand how your system will work in a high level with this document.

4.) Implementing Node Red Design [100 points]

We can implement what happens on the Node Red part of the system without programming our devices first. We can leverage the use of a web-browser based MQTT server to simulate input and output to/from the Node Red Design. This will allow us to “divide and conquer” – divide the design into smaller, more achievable tasks.

In this part of the assignment, implement your complete Node Red system and test how it works using a web based browser such as <http://www.hivemq.com/demos/websocket-client/> as seen in class.

Unit test your complete Node Red system – **implement the complete workflow, and use MQTT messages, sent by a client broker** such as <http://www.hivemq.com/demos/websocket-client/>, to test that the inputs work. Be sure to test everything!

Make the frontend to show to the user all the important information. This includes graphs, dials, buttons, etc. In short – all that you need to put in the frontend to allow a user to see your system in function!

What you implement should be complete – that is, it should have all the logic that your application needs as well as all the user interface required, and everything should be tested to work.

To turn in: Turn in a video of your Node Red working, explaining how it works. Turn in pictures of your backend and frontend of Node Red. Also turn in a document with the source code for your node red design (Menu -> Export -> Copy to clipboard).

A10 RUBRIC

Points	Rubric
100 points	Video shows clearly that a OTAFU was done in the system. A video of the update happening on the CLI is fine.
100 points	A document explaining the topics of the system and the general program flow is provided. Each topic proposed mentions what data is transmitted in it and how it is used by the system
100 points	A node red system is provided (Small explanatory video + pictures + code).