ECSE426 Microprocessor Systems
Winter 2017
Final Project
IoT – Sensor Data Management from Hardware to Cloud

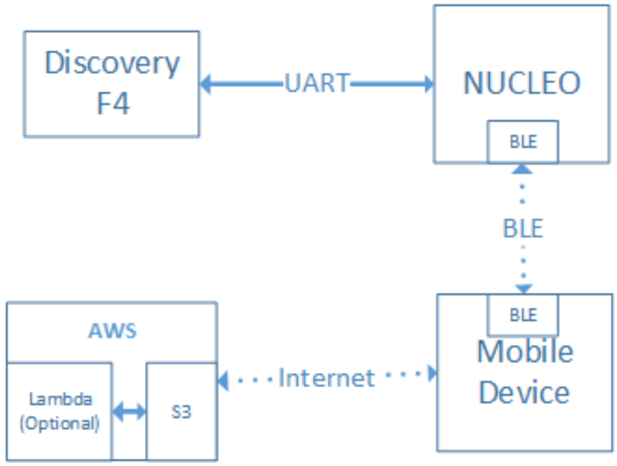
Introduction

For the final project, we will develop a system to explore the interaction of embedded peripherals and sensors with cloud-enabled services, which is one of the main hallmarks of Internet of Things (IoT) designs. The F4-Discovery board will be augmented by a Bluetooth Low Energy (BLE) connection to connect to Internet via a smartphone. The BLE interface will be realized through a STM32F401RE Nucleo board, along with an IDB04A1 BLE daughter board, which will be connected to the Discovery board by a serial link. The BLE module should transmit and receive at minimum the accelerometer sensor data between the F4 Discovery and the smartphone. The smartphone will employ the cloud services for upload, download and processing of the files.

The system will allow the board to send data such as the accelerometer measurement data and a push button status, over the BLE connection to the smartphone device. This data will be saved in a file and uploaded to the Amazon Web Services (AWS) cloud, where it will be stored in an S3 (storage service) bucket. The processing such as filtering or tilt angle detection will take place on the data uploaded, and the result will be saved in a different file. One optional function that is strongly encouraged is to use AWS Lambda, a computation service in the cloud. The result file will be downloaded via the smartphone device and sent to the board, which will output it as an analog signal, using the DAC. The display on the lab oscilloscope will allow for effective visualization of the data.

Project Details and Design

The project is composed of four units. Each of the units needs to accomplish specific function. The system diagram can be seen below.



Functionality Outline by Hardware Parts:

1. STM Discovery board
   - Submits accelerometer readings to the AWS cloud, through the Nucleo board and smartphone.
   - Reads data obtained from AWS cloud through Nucleo board and outputs it on the DAC.
2. STM Nucleo board with BLE board
   - Interconnects Discovery board with smartphone. Provides BLE functionality to the Discovery board.
3. Smartphone
   - Interconnects Nucleo and Discovery board with AWS cloud. Accesses cloud services, including the authentication. Sends processed data back to Nucleo board
4. AWS cloud
   - Stores Discovery board data, manipulated files, makes data visible to clients on any platform. Basic processing includes operations such as mirror-imaging of the data.
   - Bonus: Performs advanced processing (such as filtering, computing pitch and roll) of the data using Lambda functions.

STM Discovery Board

The purpose of the Discovery board is to read the accelerometer raw data (X,Y,Z), and transmit the reading to the Nucleo board, while also reading a value from the Nucleo board (whenever ready) and outputting it on the DAC. This serial communication should be done via a **UART** connection.

A tutorial on DAC and UART connection, as well as reference material will be provided. For the accelerometer, you are free to use the code from the labs to calibrate it and take measurements. Filtering will not be necessary this time, as the data should be processed (filtered) on the cloud or phone.

By pressing (not holding) the **blue button on the discovery board**, you should read accelerometer sensor values (at **25 Hz** data rate) for **10 second** (use a timer to make a 10-sec interrupt). Once the reading is finished, the data should be transmitted to the Nucleo board through UART in the asynchronous mode. Then the board should wait for the processed data from Nucleo board. Obviously, the data size is different as it comes as Pitch and Roll. Finally, you should use Pitch or Roll vector as digital values (normalized from 0 to 2^12) and convert it to an analog voltage by configuring a DAC in 12-bit mode. The DAC input should be updated every 10 ms.

BLE Transceiver (STM32F401RE Nucleo board + IDB04A1 BLE board)

The purpose of the Nucleo board is to provide the Bluetooth Low Energy functionality to F4-Discovery board. Using a BLE daughter board, the Nucleo board will connect the Discovery board to the outside world where it can eventually reach the AWS cloud. The Nucleo board needs to do two things; first, to obtain the accelerometer readings from the Discovery board and transmit them to the phone over BLE, and, secondly, to receive the data from BLE and transmit it to the Discovery board.

The drivers and API for using BLE are provided, and the documentation can be found in (DOC_20 to DOC_23).

The board must be configured to operate as a BLE Peripheral server. There will be at minimum an accelerometer BLE service, and a button service. The accelerometer service will contain three characteristics, each containing the acceleration on one axis (X, Y, Z). The characteristics will have properties 'READ' and 'NOTIFY'. The button service will have the property 'NOTIFY'. The third service should contain the result of processed data needed to be applied to DAC and have the property of 'WRITE'.

Android application

BLE

You will have to connect your phone to the board using Bluetooth Low Energy and obtain the accelerometer readings from it. Additionally, you will need to send to the board the accelerometer readings obtained by the cloud.

It is recommended to use an Android smartphone, and a good IDE to use for this is Android Studio. If you wish to proceed with iOS devices, or if you need assistance acquiring an Android device, do not hesitate to contact the TAs.

For this section, reference material will be provided, and it will be covered during the tutorial. For a good example of code, the Android developer website contains a sample application of BLE use. This application searches for all BLE devices, and after connecting to a device provides all services and characteristics provided by the device. This application can only receive the data, and you have to edit code for writing to a characteristic.

AWS

Amazon Web Services provide sufficient "free-tier" functionality for you to complete the project, but you have to be cautious not to generate excessive data transfers and processing.

In AWS, the S3 service (stands for Simple Storage Service) will be used to store the data sent by the phone application by means of *buckets* of storage. You are required to upload a file to an S3 bucket from the phone, and later download a related file from the S3 bucket to the phone. The derived data may contain filtered data, calculated tilt angles (yaw, roll), the data that constitutes a mirror image of sensor data (for visually identifiable processing) or any other functions that you might find useful. Then, the derived data will need to be downloaded to the phone.

The AWS website provides documentation and code that can be used to access the S3 service and upload/download files to it. Reference material on this and a tutorial will also be provided.

Bonus part – AWS processing

Recent updates to AWS include so-called Lambda functions for processing data in the cloud. Use of Lambda functions for advanced processing will earn you a bonus. In that case, you will develop Lambda functions that manipulate nontrivially the data uploaded to S3 and process it in Lambda. As mentioned above, filtering is one type of processing you can perform. You can implement the accelerometer filter you found that works best in the labs and use the same mapping technique of pitch and roll. Reference material on this and a tutorial will also be provided.

Being a bonus part, using the Lambda service for processing is recommended, but not mandatory. You can use a different processing service if you would prefer. If you would prefer to not do cloud data processing at all, this implies that you can process the data off-cloud, on smartphones or other clients attached to your data in the cloud. Hence, the alternative places for data processing could be the phone itself, the Nucleo board, or even software running on your computer. Important to note is that if you select to apply the data processing off-cloud, you still have to demonstrate that the data is held in cloud and accessible from multiple clients.

Regarding the processing itself, filtering and mapping to pitch and roll is recommended, but you are free to implement more than that. Original design and novelty is encouraged, and bonuses will be awarded on a case-by-case basis, depending on originality, difficulty and usefulness of the options.

Demonstration

There will be two stages of demonstration

- Progress Demo will take place during the week of March 20 to April 7. Every group is obliged to provide the specification of all functionality that will be provided during the final demo.
- Final demo will take place on April 10.

Report

The final project report is supposed to be more formal than the lab reports which you wrote during this semester. You should naturally consider all feedback you've received this semester while preparing your report. In particular, we would like to stress the need for the extra following points in your report:

- The components in your system.
- A timeline of work and a breakdown between team members.
- A block diagram of your firmware, showing roughly how modules interact.
- Proper screenshots of your application.

As always, you should explain the reasoning behind your design choices and provide sufficient information on how the details are realized. The contributions of each member should be clearly delineated in one place in a concise manner. The working code on embedded, phone, and cloud sides will need to be included in the project submission. The code by itself should be clean and well-documented.

Important! Obtaining and returning your Kits

Final project groups can borrow Nucleo boards as soon as the final project groups are formed. All groups should return all components to ECE labs within one week of the final project demo. The parts returned should exactly match the ones given to you (specs, models, and part numbers). They should all be in a fully working condition. This include all boxes, kits (discovery and wireless), tools (screwdrivers and wire wrappers), breadboards, peripheral components (LCD, 7-segment displays, keypads, and motor kits (with all brushes and horns intact and complete)) and the 80 wire connectors. The breadboards returned should have no components mounted on them, ECE labs always tracks and checks your kits and keep us informed. On the occasion of your failure to return the kit on time, or having any missing components, you will be penalized 40% of the total project grade.