Research and analysis of the Multi\_Role Application

Due to the constraint as a result of the short amount of time until the end of the project, choosing the best approach is the difference between a successful application and the failure of the project. That is why, while exploring the possibility of using a Java application to connect to the SensorTag, still through the use of the LaunchPad, it would be necessary to continue the research on the Multi\_Role application, as well as understanding how to change the code in order to be able to adapt it to our current needs. Since the documentation on the Multi Role application on the internet is lacking, the best approach would instead be to look at the code and try to understand how it works. To help, we are trying to compare the Multi\_Role application to the Host\_Test application.

Multi\_Role issues that prevent implementation

Firstly, the Multi\_Role application is, as mentioned previously, hardly documented. This made it a challenge at first to understand how the application works. After setting up the environment using PuTTY, we were able to use the application. After running the application, everything seemed to work except for the GATT Read/Write operation. In order to track the execution process, we used the debugger feature on Code Composer Studio to track the path of the code, with the hope that it would eventually lead to the function, or at least the file, that handled the Read/Write operation. However, because the application requires input from the LaunchPad buttons in order to execute its functions, the debugger would stop right after the BIOS was initialized, and even though the code would run, no input would be recognized. The alternative approach meant looking at the code itself, and tracking the definition of the methods to get to the right place. Doing so eventually led to the answer that contained the Read/Write function, which consisted of basically an IF statement, that counted the times the operation was successful. Therefore, the main problem became being able to implement a working Read/Write operation.

Comparing Multi\_Role to Host\_Test

Since the Host\_Test application allows to connect and read and send data from and to a single SensorTag, it makes sense that the code could possibly be adapted to the Multi\_Role in some way. So far, there have been some obstacles along the way. For starters, the file structure for the two applications is different. This means that, without fully understanding the Multi\_Role application, any adaptation of the Host\_Test Read/Write might not work. However, using an online web page to compare the code of the two main files, it shows that the #include statements are mostly, if not all, the same. This was true for other files as well. Therefore, it should be possible to recycle and refactor code from the Host\_Test to the Multi\_Role.

Research on the Host\_Test application

Unlike the Multi\_Role, the Host\_Test application has a higher amount of documentation available. However, the amount is almost infinitely bigger, and finding the small piece of information required to make the implementation of the application is just as difficult as it was for the Multi\_Role. So far, a quick skim of the main documentation given by TI gave us a basic understanding of how the Host\_Test works. However, little of help was given regarding the Read/Write in particular, at least as far as the implementation of it is concerned. More progress will be done until we have a final solution.