This document outlines the journey we took to prolong the battery life of the Telus Kits.

Starting point: Running the default Avnet code resulted in a ~25 hour battery life.

Goal: A battery life of 7 days.

The following, in chronological order, describe the hypothesis for each experiment conducted and our findings.

|  |  |
| --- | --- |
| Increasing message interval will increase battery life | |
| Description | We believed that the act of sending a message consumes more battery, so therefore sending less should result in less consumption. |
| Action | While keeping everything else (the code) the same, we increased the message intervals. The default interval was 5 seconds, and tests were done up to 110 seconds. |
| Findings | Doubling the message interval to 10 seconds increased battery life by 5 hours but did not scale. Additional increases to message interval did not result in much more savings |
| Conclusion | The BG96 module takes up a lot of power just by being on and will do little to help us achieve our goal. |

|  |  |
| --- | --- |
| Turning off GPS will save power | |
| Description | We noticed that upon starting the device, it takes some time for the GPS to get the location. The GPS location is also read every time we send out a message. |
| Action | While keeping everything else (the code) the same, we do not turn on the GPS. |
| Findings | Battery life increased by 5 hours, but does not seem to go above that. Does not increase proportionally to message intervals. |
| Conclusion | Turning of the GPS will give us around 2 hours extra but does not scale. |

|  |  |
| --- | --- |
| Building in release mode will increase battery life. | |
| Description | We realized that we were compiling the code in debug mode which is not optimized. |
| Action | While keeping everything else (the code) the same, we compiled the code in release mode. |
| Findings | No noticeable change. |
| Conclusion | While the build might have been more optimized and the program running faster, the BG96 module is taking up so much power that even if release mode made a little difference we weren’t able to notice it. |

|  |  |
| --- | --- |
| Putting the board to deep sleep between sending out messages will increase battery life. | |
| Description | The program uses a thread to connect with Azure and sends out messages. Every time it sends out a message, the thread is put to sleep and that is how the message intervals are controlled. After enabling CPU statistics, we found that although the board is sleeping while the thread is sleeping, it is not going into deep sleep mode. Deep sleep mode is reported to use the least power. |
| Action | Research on how to put the board into deep sleep mode. |
| Findings | It turns out that the operating system we are using, MBED, should automatically put the board to deep sleep whenever it can. The next step was to find out what was blocking this. There are instructions on how to seek out deep sleep locks [here](https://os.mbed.com/docs/mbed-os/v5.14/tutorials/power-optimization.html#power-modes) however we were unsuccessful in implementing it. The article does however, mention that UARTSerial blocks deep sleep. Since the BG96 module communicates with the board through UART we believe that is the reason deep sleep is being blocked.  We created a tiny “blinky” program without connecting the board to the sensor shield and BG96 module and was able to see that the board was going to deep sleep when the thread is sleeping. |
| Conclusion | The board is capable of going into deep sleep and it should do so automatically when we sleep the thread. Since the BG96 communicates with the board via UART it is blocking the board from going into deep sleep. |

|  |  |
| --- | --- |
| Turning off the BG96 module between sending messages will achieve significant battery savings | |
| Description | Since finding out that the module being powered on is consuming a lot of power and that it’s also blocking the board from going to deep sleep, turning it on only when we’re sending a message and turning it off in between should result in significant battery savings. |
| Action | Implemented the code to turn on/off the BG96 module. |
| Findings | We were successful in turning on the board to send a message, and turning it off afterwards. However, we found that this resulted in a hard fault exception almost every time (the program crashes) within 5 minutes to 2 hours. |
| Conclusion | A lot of research was done to see what was the cause of the program crashing but we were not able to pinpoint what was happening. |

|  |  |
| --- | --- |
| Utilizing Power Saving Mode (PSM) will increase battery life | |
| Description | The LTE-M network we are using has a PSM feature that our BG96 module also supports. Getting this PSM to work should actually consume less power than turning the module off/on because PSM is like powering off the device except it keeps the connection to the network alive. |
| Action | Research on how to implement PSM with the BG96 module. |
| Findings | We were not able to put the board to PSM |
| Conclusion | Further research and effort to implement PSM should be the top priority for future work on this project. |