

Event Logger for CC2650

Tomer Ben Moshe

A course project in Advanced Computer Systems, Tel-Aviv University, Fall 2016

PRIMARY TARGET:

The main objective of the project was to get messages via UART and save the messages on the SD CARD. And when the user presses the button at the side of the chip the project is supposed to send a message via BLE if there is someone connected.

Implementing the SD CARD:

The SD CARD is applied as a buffer.

Every block of 512 bytes at the buffer has one of three roles:

1. It is the start of a message: If it has the first byte a '\0' and the second byte something other than a '\0'. This means that from the second byte up to the 512th byte is the beginning of the message. (If the message is too short than the project fills the rest of the bytes with '\0').
2. It is a follow-up of the message: If the first byte is something other than '\0', then this block is a continuation of a message that started in the previous block.
3. It is the end of the buffer: If all 512 of the bytes are '\0', then this block indicates that this is the end of the buffer.

So on the initialization of the project, there is a task that checks where is the end of the buffer at that moment.

That task also listens to the UART channel (with 115200-bit rate). When the user enters the first 511 bytes of the message, the task saves the 511 on the SD-CARD (preceded by '\0', to indicate that this is the beginning of the message).

The task continues to listen to the UART channel, saving the next 512 bytes on the SD-CARD when they arrive. And so on until the user hits the 'Enter' button.

Then it saves the end of the message (padding the message with '\0' up to 512 block bytes).

Then wait for another 511 bytes to start a new message.

There is also another task that waits for the user to press the side button on the chip, and when it does, the project gets the last message on the SD-CARD and sends it via BLE to whoever is connected to the chip.

The task then makes the block with the start of that message which was sent – the new end of the buffer.

Difficulties:

The main difficulty was to handle the lack of space in the chip.

In the beginning, I got two problems at the same time –

- The debugger could not perform the SPI_Transaction command
- There was not enough space in my task.

And because I thought there was only one problem I had a hard time understanding what's going on.

And another bug issue I had at the end is that whenever I added static arrays of 512 chars, the program didn't work because of lack in space (and I had to handle the whole algorithm with only one array of these chars).