

Russell Silva
12/20/2017
AMRUPT

I/Q Extraction Update

Problems

1. The I/Q measurements only work for the 2nd iteration of the main function. This is a weird error that I have only been able to work around (e.g. by setting a high number of samples). The only differences between the code listed (<http://www.ti.com/lit/an/swra571/swra571.pdf>) and my code are:

- a) the printf functionality
- b) the workaround for the iteration problem (if/else statement within the main function), the previous rfPacketRx example code that is not mentioned in the document, and the callback function. Originally, the code for handling the samples was meant to be in the callback function. However, I was not able to print out this data within the callback because I could not find a way to call this callback function within main() because I cannot use callback without the function parameters:

```
RF_Handle h  
RF_CmdHandle ch  
RF_EventMask e
```

2. Verifying the IQ measurements: All of the measurements are the same at each runtime() which makes sense because I do not believe the CC1310 is receiving any packets (regardless of another CC1310 transmitting) while debugging the code in Code Composer Studio. My previous idea to work around this would be implementing a file write while using the CC1310 in SmartRF studio. I am aware that recording this data is probably within the scope of Peidong's side of the project.

Planned Course of Action

There is slow improvement on finding specific coding information (e.g. the functionality of currentReadEntry) necessary for adjusting the code if the I/Q measurements are found to be incorrect. This is because of the lack of information available on the subject, the lack of specifics on all the relevant forum posts (listed below), and my relative unfamiliarity with TI RTOS. Forum Post 1. (see below) has a promising solution that could focus programming on an Arduino Uno (which I have with me) through a GPIO connection. MCE_GPO1 is listed as the GPIO pin for the binary data signal coming from the demodulator when receiving (http://dev.ti.com/tirex/content/simplelink_cc13x0_sdk_1_30_00_06/docs/proprietary-rf/html/rf-core/signal-routing.html#id1), and the signal can be rerouted to a GPIO pin. The MCE core signal is familiar to me as the I/Q document (<http://www.ti.com/lit/an/swra571/swra571.pdf>) sets an MCE_RFE_OVERRIDE, and uses the override to sample a certain binary signal from the MCE into 8-bit buffers. If I am correct, I would be implementing the same I/Q extraction procedure that the document entails, but in a much less complicated manner.

Relevant Forum Posts

1. https://e2e.ti.com/support/wireless_connectivity/proprietary_sub_1_ghz_simpliciti/f/156/t/590495

Note: Forum post 1 is interesting, as the TI employee pointed out the same synchronization issue that we came across in obtaining reliable phase information. However, the employee does not consider the RF switching method that we discussed in our last meeting.

2. https://e2e.ti.com/support/wireless_connectivity/proprietary_sub_1_ghz_simpliciti/f/156/t/528401
3. https://e2e.ti.com/support/wireless_connectivity/proprietary_sub_1_ghz_simpliciti/f/156/p/528097/1920625
4. https://e2e.ti.com/support/wireless_connectivity/proprietary_sub_1_ghz_simpliciti/f/156/p/619108/2281646
5. http://e2e.ti.com/support/wireless_connectivity/proprietary_sub_1_ghz_simpliciti/f/156/p/620543/2292997
6. http://e2e.ti.com/support/wireless_connectivity/proprietary_sub_1_ghz_simpliciti/f/156/p/579320/2127500
7. https://e2e.ti.com/support/wireless_connectivity/proprietary_sub_1_ghz_simpliciti/f/156/p/492932/1780750