**Using the Platform with MATLAB**

**Notes on connecting the platform to the PC host**

The USB interface does not provide power to any part of the platform. To interface the platform to the host PC, the platform must be powered by a separate source (such as its battery pack).

When communicating with the platform via MATLAB, many of the MATLAB scripts will require that the user select the correct COM port when opening the virtual serial port. The name of the correct port will depend on the computer being used, and is often selected arbitrarily. Therefore the user will likely need to determine the correct COM port name by trial and error.

The COM port name is defined in several MATLAB scripts. The code which defines the COM port has the following form:

s = serial('COM13', 'BaudRate', 3000000)

Where in this case COM13 is the selected port name. For more info on virtual com ports, look at help documentations on serial().

**Host mode experiments**

The simplest experiment to perform is the LO calibration. This performs a simple CPMG (with phase cycling), measures the phase of the resulting spin echoes, and uses that to recommend an adjusted phase offset for the LO.

The main script which performs this experiment is 'LOcal\_phasecycle.m'. However, this script file alone is not sufficient to perform an experiment. You first need a way of defining the experiment structure, which is often done in a separate script file. Such a file is 'declare\_experiment.m,' located in the 'Useful functions' folder. This script creates the experiment structure with initialized parameters. For different experiment types, you will probably want to make different separate copies of this file with different parameter values. Please respect the data types in this structure, and the order of variable definition. Any modifications to the experiment structure format will likely result in the platform not recognizing it as a valid experiment.

Before executing LOcal\_phasecycle.m (or any other experiment script) you should declare the following global variables in the command window:

global s;

global experiment;

global true\_experiment;

This will allow you to access these important objects when not executing the script. Upon executing LOcal\_phasecycle.m, the declare\_experiment.m script will be called, initializing experiment to the values you've defined within. Thereafter, the main script will modify experiment with appropriate parameters for the specific experiment. For example, for a LO phase calibration experiment, the RF pulse phases will be set to 0 (or 180 for the phase cycled portion).

The actual experiment execution occurs when the function run\_host\_mode (located in the Useful function folder) is executed. This uses the contents of experiment to run a full host mode experiment. When attempting to run the experiment, the platform will first verify that it has a valid SD card, and that the experiment structure is valid. If either of these requirements are not met, the platform will reply with an error message and the MATLAB script will abort.

While the experiment is running, the sampled data will be displayed after each sequence execution. At the end of the experiment execution, the raw complex spin echoes will be saved. If the experiment uses phase cycling, then the experiment will be performed a second time with an inverted RF pulse phase (resulting in another set of raw data).

After the experiment has been performed, the main script will take the sampled data and attempt to analyze it. For example, in LOcal\_phasecycle.m, the script will use exponential fitting to find the amplitudes of the in phase and quadrature components of the echo train, and compute the phase of the echoes. It will then subtract this measured phase from the LOphase used in the previous experiment, resulting in a new LO phase which should shift the NMR signal to be real in value.

**Standalone mode experiments**

As with host mode operation, start by defining the three main global variables s, experiment, and true\_experiment. Then define experiment using a declare\_experiment.m script. The host mode experiment scripts can be used in order to define the details of the experiment parameters.

Once experiment is fully defined, it must be sent to the platform and stored on the SD card. This is done with the 'program\_experiment\_to\_SD.m' script (in the Useful functions folder). If the platform detects a valid SD card, it will acknowledge to experiment and save it to memory. Remember that the uSD card is nonvolatile memory, and thus this data is kept until manually overwritten.

Once experiment is saved to the uSD card, the experiment can be executed from the platform's touchscreen GUI. When the platform is powered on, the GUI will give the user the option to verify the SD card, verify the experiment stored on the SD card, and execute the stored experiment (if the former requirements are met). Once the experiment has started, the GUI will display the progress of the experiment by counting off the number of sequence executions. The experiment can be prematurely terminated with the 'Cancel' icon. After being terminated, or after finishing the specified number of experiment repetitions, the platform firmware will return to its idle state.

The raw data sampled during standalone mode is stored on the nonvolatile SD card memory. In order to store the data on the SD card as fast possible, the platform does not use a standard file system on the SD card. Therefore, reading the SD card cannot be done simply by inserting it into a PC and opening a file (the PC will most likely detect the card as corrupt and prompt a reformat). The intended method of extracting the data is through the platform itself. After performing a standalone experiment, connect the platform to the host PC via USB, and run the MATLAB script 'get\_echoes\_from\_SD.m'. MATLAB will request the platform to upload the saved data, and from there the transfer should happen automatically. The result will be that the structure true\_experiment will be updated to the experiment parameters used in the stored experiment, and the variable CPXechoes will contain the raw sampled data.

Note that for certain experiments, the transfer of this data may appear to be quite slow; sometimes at the same rate of the original experiment's execution. Some overnight experiments we performed resulted in raw datasets of several GB, which took several hours to transfer.

Michael Twieg

mdt24@case.edu

March 2013