

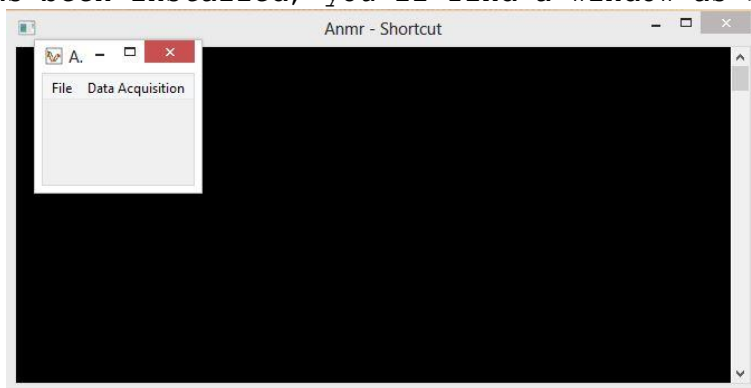
EFNMR User Interface installation:

Use the following link to download the UI files.

<ftp://bloch.phas.ubc.ca/Earthsfield>

Choose anmr-1.0 file for downloading and follow the instructions provided in the "windows install" text file.

Once the UI has been installed, you'll find a window as shown below,



Definition of parameters in UI:

The following are the common parameters in most of the pulse programs,

- **Frequency** - This can be set to the self-resonant frequency of the T-R coil at earth's field.
- **Polarization time** - The time for which the polarizer turns on and off to produce the required polarizing (B_p) field.
- **90 half cycles** - The number of 90 half cycles of RF pulses that should sent.
- **Echo delay** - The time after which the echoes from the sample should be collected.
- **Receiver delay** - The time at which the receiver should be open for collecting the signal after pulsing is done.
- **Repetition delay** - The time between the starting of a new cycle and the ending of the previous one.
- **Num_points** - This represents the number of points to be collected and averaged by the Arduino UNO during reception. This also indicates the duration for which the receiver is turned on.
- **180 half cycles** - This is especially used in spin echo and cpmg sequences where 180 pulses are sent after an echo delay following the 90 pulses.
- **Num_echoes** - The number of echoes to be collected is determined by this parameter.

Apart from One pulse (Pulse and collect), spin echo and cpmg, there are two other sequences that can be played from the UI,

- **Pulser** - This program transmits the RF pulses one after the other with a delay between the two pulses along with a repetition delay. This is used only for transmission purposes.

- **No pulse** - This program is used only for reception with the only parameter being num_points.

Operating the EFNMR system using the UI:

- The first and foremost step is to map the pins given in the "defaults" text file in "Pulse Programs" folder to the pins of the Arduino going to the relays for switching.
- The above step is an important one as the mismatch of pins can result in no FID being displayed in the UI because timings for switching goes wrong.
- Following this, a good measurement of field in and around your setup is required for field homogeneity and the average field value should be input in the frequency text box.
- If field around the coil is inhomogeneous then shimming must be done.
- The higher the number of transients (No. of scans in UI) better the noise averaging shall be.
- Make sure that Arduino is connected to the port properly and that port is visible in "System Devices". If this does not happen, then the program will not be written on to the Arduino.
- When all the sub-system interconnections are done and pins are checked, then click on acquire in the UI to visualize the FID and its spectrum.