Arduino FSQ Beacon on the Si5351A Breakout Board

A new digital mode by the name of FSQ has arrived on the scene lately, and it looks quite interesting. The mode has received some recent attention as it was featured in an article written by one of its developers, Murray Greenman ZL1BPU, in the Sep. 2015 issue of QST. According to ZL1BPU's FSQ website:

> *FSQ is a Fast Simple QSO mode designed specifically for HF. It works well under NVIS and sunrise/sunset conditions on the lower bands, and also works well for short skip and grey-line on higher bands. It can also be used on VHF FM, and clearly has a much wider useful range of operating conditions that other more conventional digital modes. FSQ transmission is also well within the capability of micro-controller based devices for low-power propagation transmissions (MEPT and telemetry).*

The QST article describes the modulation scheme as 33-FSK, with the inclusion of Incremental Frequency Keying (meaning that an offset of +1 tone is added to each new symbol). The symbol table is varicode and designed to take advantage of the number of tones available so that each lowercase letter is encoded as a single tone, while the rest of the characters are encoded as two consecutive tones. As mentioned in a previous app note, the Si5351 is quite easily used to implement modulation based on FSK at modest rates. Since the author specifically mentioned that FSQ transmission is within the capability of microcontrollers for MEPT and telemetry purposes, it looked like fun to implement a simple Si5351A FSQ transmitter with an Arduino.

The basic hardware configuration is identical to that used in the Si5351 Feld Hell beacon. The Arduino firmware from that

project was also used as a starting point for this firmware, as it already has a lot of the basic program structure (such as a timer for the modulation rate and a multi-symbol lookup table). Not knowing how well the mode would work under QRP or QRPp conditions, for my on-air trials of this transmitter, I decided to use both my 500 mW linear amplifier and my 20 W linear amplifier in order to give myself a good chance of being heard by others. You can see the setup in the photo featured at the top of the post. The Arduino and Si5351A Breakout Board are at the left, followed by the 500 mW linear amp with a low-pass filter on the output, feeding into the 20 W linear amp with another low-pass filter on the output.

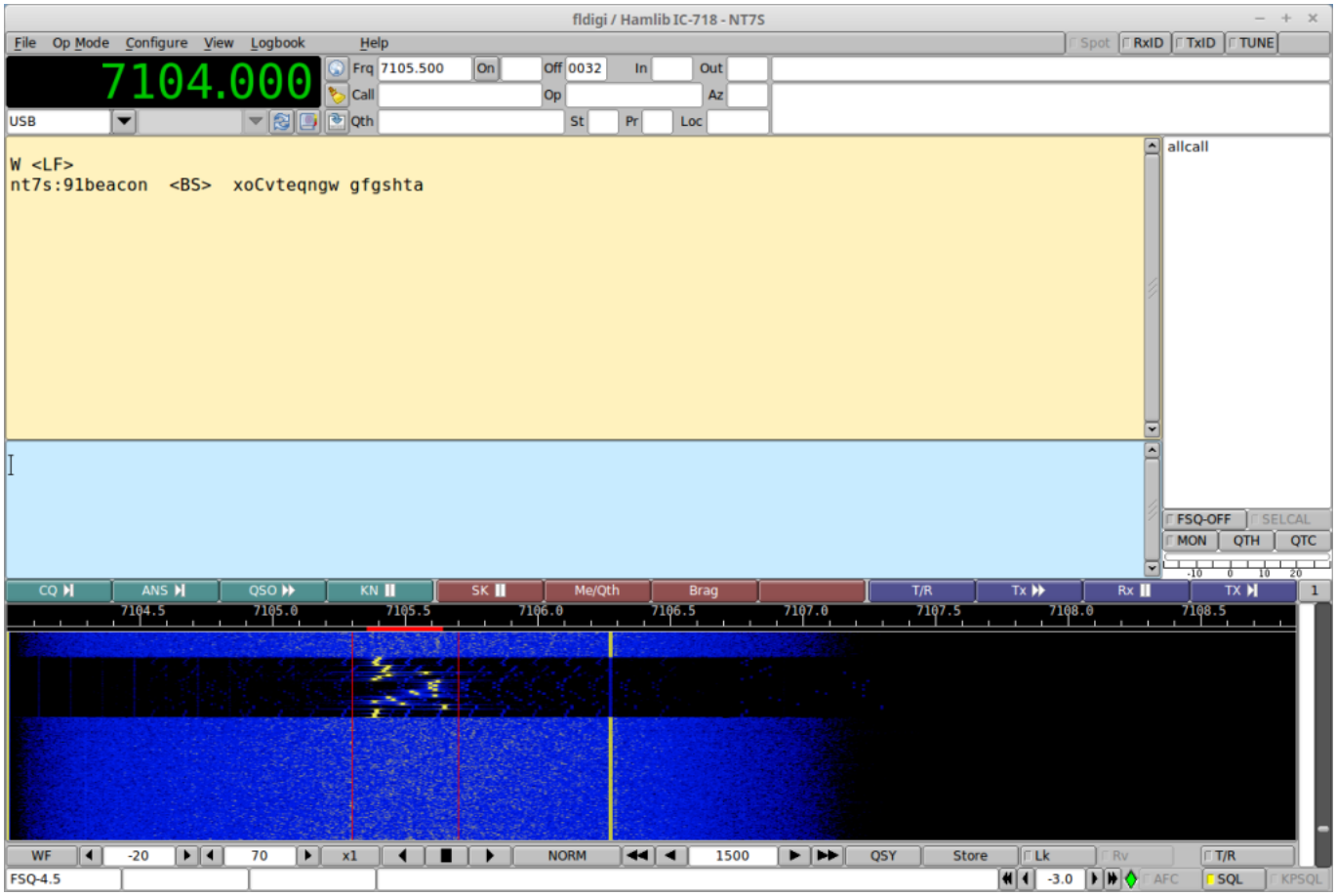
After doing quite a bit of testing on a dummy load and receiving the transmitter signal on my main station rig and fldigi, I put the FSQ transmitter on the air on 23 August 2015 and asked for QSLs from Twitter hams who wouldn't mind monitoring 7.104 MHz for my signal. The beacon was set to send a sounding transmission every 5 minutes on the main designated 40 meter FSQ frequency. It didn't take too long before I received a couple of reception reports for the transmitter from hams at a decent distance from me.

> *#qsl @NT7S Got it, nt7s:91beacon z oajb?ocio8t×b<LF> at about 01:49 and at 01:59 fairly good signal 2388 miles aprox.*
>
> *— Fred Smith (@wd3c) August 24, 2015*

> *@NT7S Also got it in Iowa 7.104MHz*
>
> *21:20*
> *nt7s: MONITOR beacon*
>
> *— wm6h (@wm6h) August 24, 2015*

In fact, it worked a bit too well, as I had someone calling me

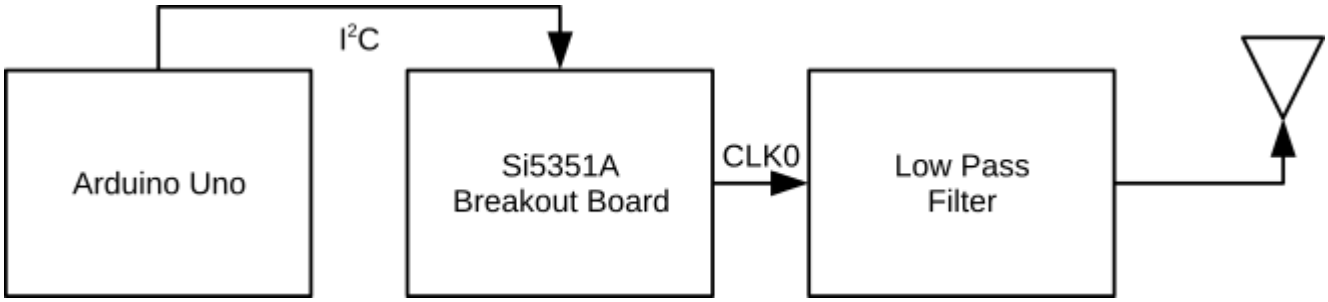
for a QSO and then asking why I wasn't responding! The only reason that I knew this was because I left fldigi running to monitor my own signal and enough of the other ham's signal leaked through my antenna switch that I was able to get a print in fldigi!



Which leads to an important consideration for you if you decide to try this new mode in a transmit-only fashion: as of yet, I have not seen a protocol for MEPT/telemetry uses of FSQ so I cannot advise you of the proper way of conducting such transmissions. My current recommendation is that you make it clear that your transmission is a test or beacon transmission so that others will not try to call you. Better yet, if you have others who are working with you, it would probably be prudent to QSY a bit so as not to be clogging up the main channel with your messages. And of course, never leave your transmitter running unattended if you are in doubt of the legality of such operation in your country.

Since ZL1BPU did mention the use of FSQ in MEPT/telemetry applications, I do hope to get some clarification from him and other developers on the proper protocol soon. I believe that FSQ could have some good potential to be used in such applications due to the message format used and the lack of time synchronization needed (like WSPR and JT65/JT9 have). Based on the small amount of QSO operation of FSQ that I have done so far, I would recommend a 2nd FSQ channel near the main channel that is devoted to MEPT and telemetry. A couple of extensions to the message format specifically for MEPT messages and for telemetry (such as perhaps APRS over FSQ) would also very extremely handy.

Block Diagram



Bill of Materials

Item	Quantity
Arduino Uno (can substitute other variants)	1
Etherkit Si5351A Breakout Board	1
Low Pass Filter	1
Amplifer (optional)	1
RF connectors of choice	

Wiring

Terminal	Arduino Uno Pin
Si5351 SCL	A5
Si5351 SDA	A4
Si5351 5V	5V
Si5351 GND	GND

Usage

Simply load the sketch onto your Arduino Uno, connect the power and I2C lines from the Uno to the Si5351A Breakout Board and then connect an appropriate low-pass filter to the output of the Si5351A Breakout Board CLK0 (or the output of the amplifier, if you are using one). Connect the output of the low-pass filter to a dummy load, then provide power to the Uno, Si5351A Breakout Board, and amplifier (if applicable). Then use your favorite digimode program and a PC-connected receiver to monitor your transmission in order to ensure that your setup is working correctly. Once you are satisfied that is the case, connect the output of the low-pass filter to an antenna in order to transmit your FSQ transmitter on the air. Don’t forget to change your callsign and the message in the Arduino sketch before you put it on the air.

Extending the Transmitter

With only about 10 to 20 mW of output power, it will be tough to hear this transmitter barefoot, so some sort of amplifier would be very useful to bring the output level to a point where you can reasonably expect to be heard. In the near future, we intend to publish an example design for a linear amplifier that will bring the output power up to 500 mW, and will update this post with the link to it when it is ready.

Arduino Sketch

Required Libraries

Si5351Arduino

Links

Si5351Arduino Library on GitHub
Si5351FSQ.ino sketch on Gist