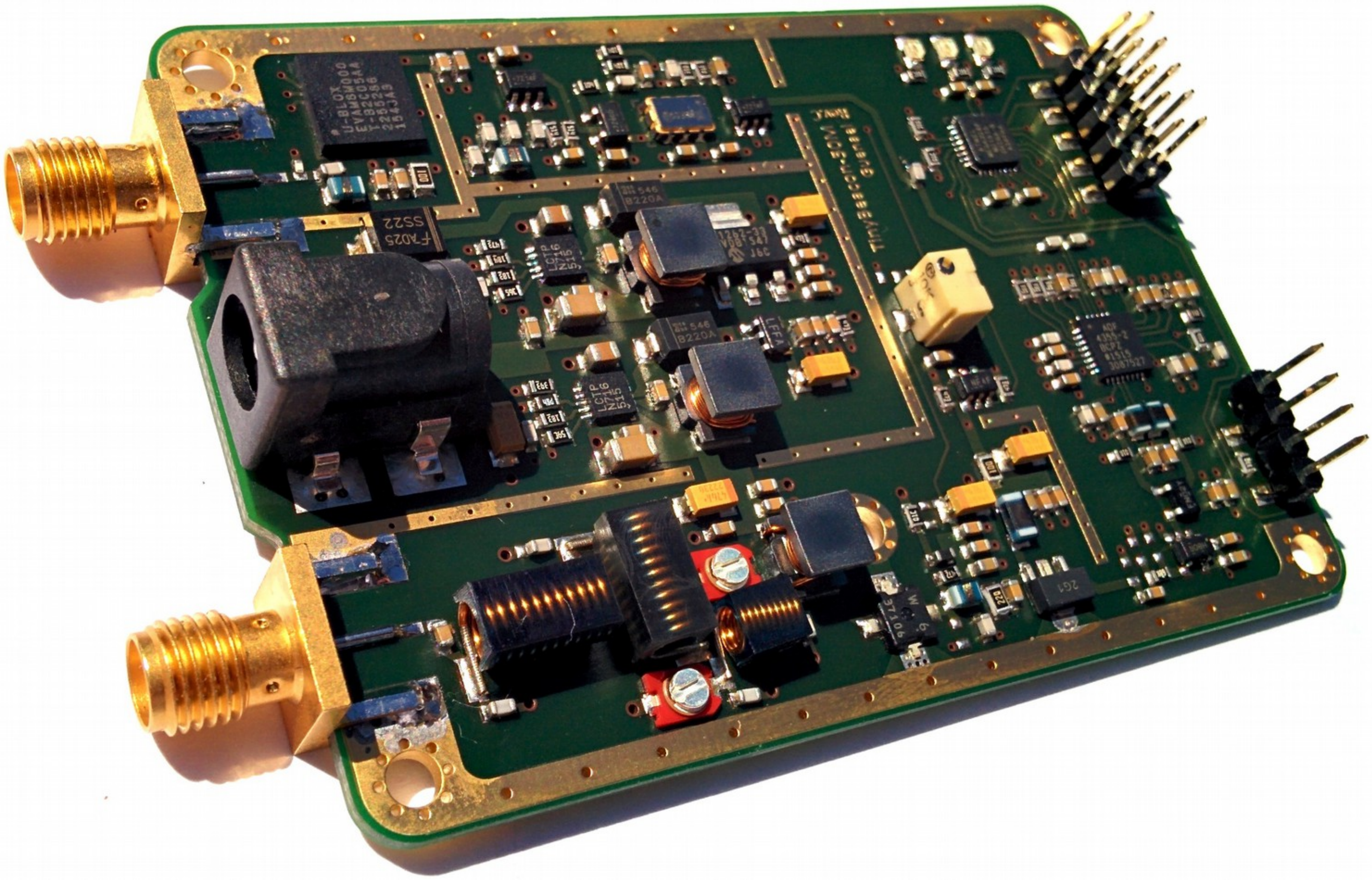
TinyBeacon Project

GPS disciplined VHF/UHF compact beacons

Guenael, VA2GKA

August 8th, 2016

Rev. 1



TinyBeacon project

GPS disciplined
VHF/UHF compact beacons

Guenael Jouchet, VA2GKA

$\{guenael\}@jouchet \cdot ca$

August 8th, 2016

Abstract: *TinyBeacon enables an easy setup of VHF and UHF beacons, through a simple and compact design, using a credit card-size PCB, at a low cost, and with easy installation close to the antenna. The PLL system is disciplined by GPS, allowing digital modes like WSPR and PI4, which require good clock stability and time synchronization. The system integrates a DC-DC regulator that accepts flexible power between 10 and 15 volts, with a common QRP output power of 5W. The reference design is provided for 50, 144, 220 and 432 MHz, but the circuit could be adapted to other bands.*

Keywords : *beacon, vhf, uhf, wspr, pi4, compact, design, qrp*

1 Goals & requirements

Over the past 8 years, the WSPR protocol[1] and programs[2] published by Joe Taylor (K1JT) have experienced growing success in radioamateur community and WSPR is now widely used on the HF bands. A lot of designs were published for transmitters or receivers, and the WSPRnet[3] website now provides extensive statistical data. The availability of this data in real time via the Internet allows discovery of propagation phenomena while they are occurring.

While the HF bands are very well covered[4], there are fewer contributors to the upper bands like VHF/UHF/SHF. However, radio beacons are very important resources for anyone enjoying these bands to identify seasonal openings. Since we cannot continuously listen ourselves, systems such as WSPR allow a highly appreciatied constant monitoring, and this new generation of beacons could help a lot.

In this document, I present a design of a compact beacon, the size of a credit card, including a power regulator with an input voltage between 10 to 15 volts, and a QRP output power of 5 watts. Furthermore, a beacon is expected to maintain an exact frequency, and to achieve this point the local oscillator is disciplined by GPS, which is also used to synchronize the start of each WSPR transmission. The whole system is self-contained and does not need extra maintenance. It can also be installed outside close to the antenna, in isolated places.

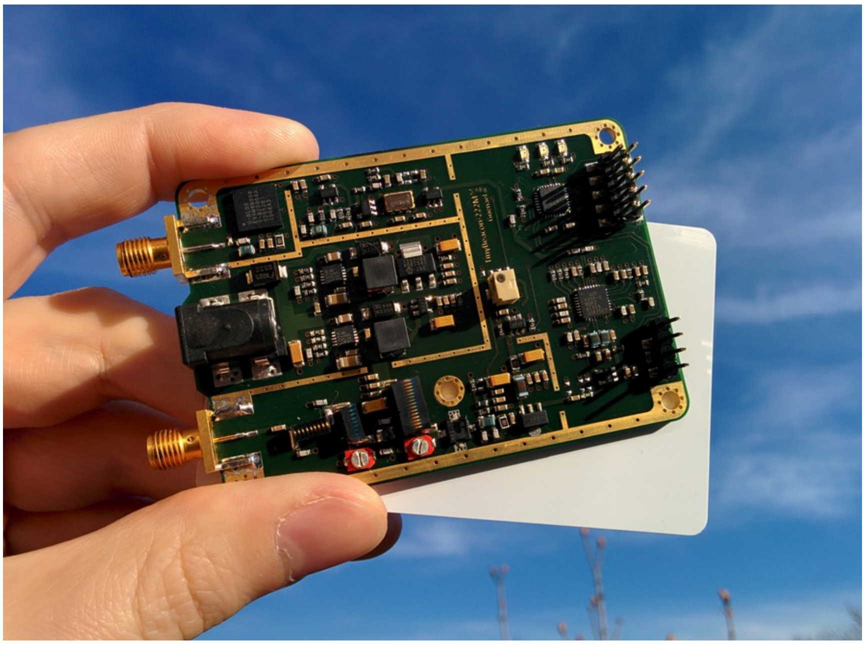


Figure 1: TinyBeacon PCB rev.C, credit card size

A second part of the project is focused on autonomous receivers and reporters (VHF / UHF bands), based on embedded computers like a Raspberry PI[5] or a BeagleBone[6], coupled with a low-cost SDR receiver (ex. RTL-SDR USB dongle[7]). Having a low-power system, independent of the station, might encourage more radio amateurs to setup this kind of automatic reporters. Furthermore, this system can be achieved at a low cost around $100.

Through this project, I hope to see more beacons appear in North America, and have better coverage of these great bands.

While there are many designs and kits for WSPR on HF bands, building transmitters for highest bands requires some precautions. These higher frequencies require a better control and quality of the reference oscillator, often achieved with a Rubidium clock, or a GPS disciplined oscillator (GPSDO). The good news is with miniaturization of components, it is possible to integrate a GPS on a circuit as small as a credit card.

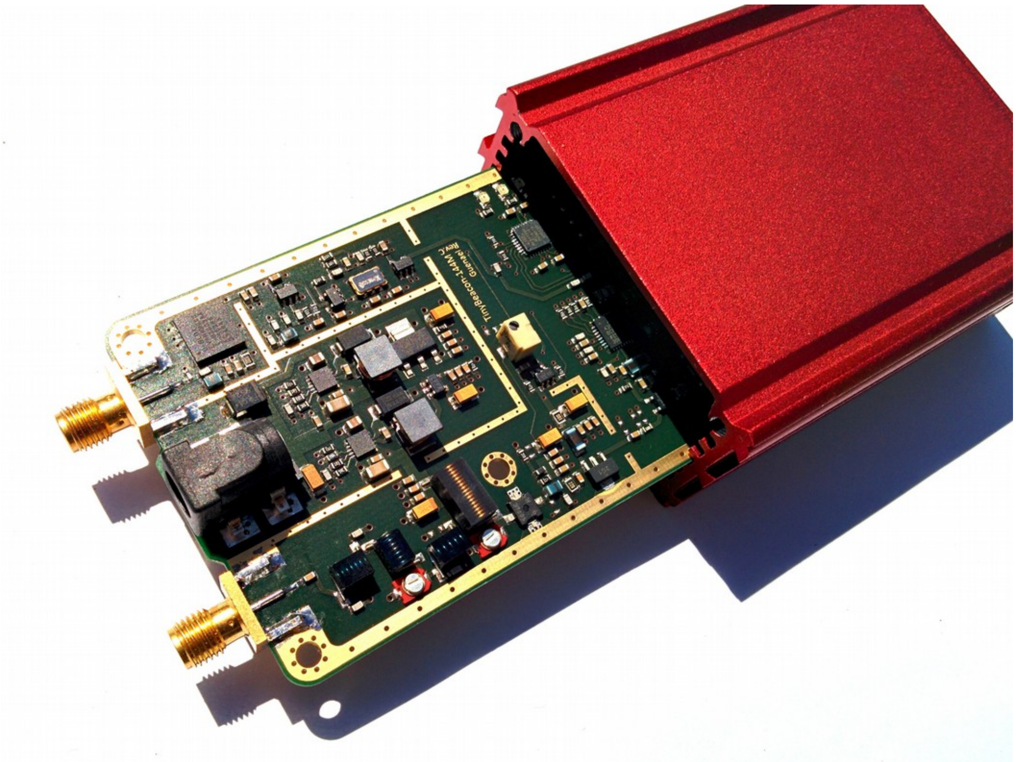


Figure 2: Easy integration into extruded aluminum enclosures

However, there are still other issues related to propagation phenomena. Indeed, some propagation phenomena (troposheric, rain scatter, aurora...) can alter the signal itself, including in frequency, which is a problem for WSPR modes because their tones are separated by only 1.4 Hertz. Proper decoding may be impossible despite the fact the propagation conditions are present.

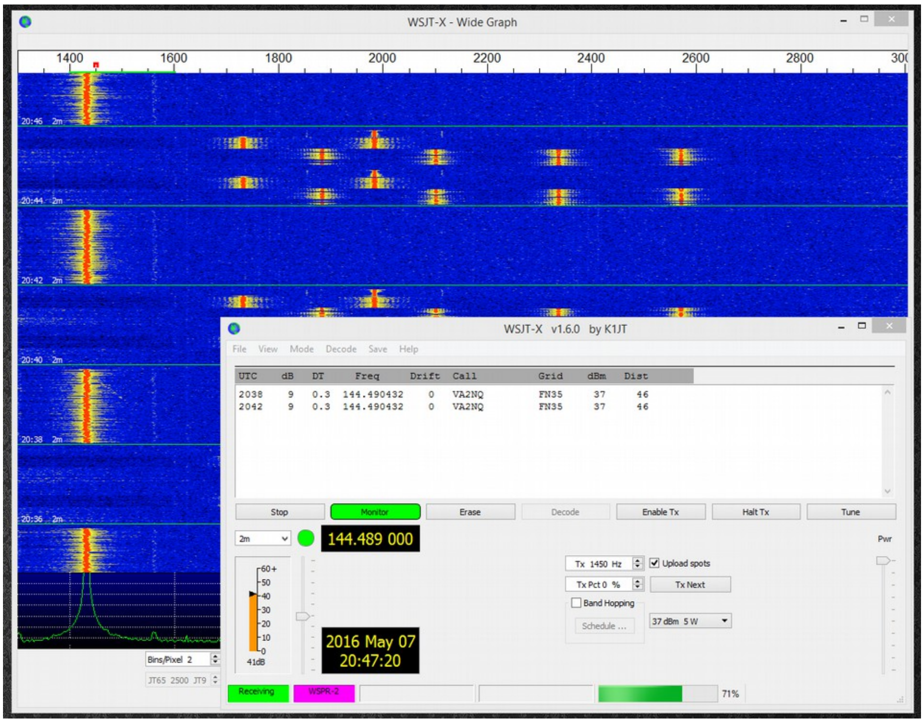


Figure 3: WSJT-X used to decode WSPR part

There is another mode named PI4 (PharusIgnis4) [8], which is very similar to WSPR and based on JT-4. This mode is popular in Europe and uses larger spaces between tones. Consequently, it is less sensitive than WSPR due to the fact it uses a larger bandwidth (but also a shorter duration), but on the other hand, it becomes more robust against propagation modes having frequency disturbances. In addition, it can be used without a rubidium clock or GPSDO, which can be interesting for some stations. That is why I included this interesting alternative to the sequencer. WSPR is transmitted for 2 minutes, and then, two sequences of one minute are sent with PI4. The PI4 protocol transmits the callsign on 4 tones for 25 seconds, then a Morse code message is sent, and finally the transmission is ended with a carrier for 10 seconds. In this way, this alternation of WSPR / PI4 can obtain the best of both worlds. A rate of 50% is commonly considered to be too high on the HF bands because of the number of users, but in VHF/UHF, this is quite acceptable due to the sporadic propagation phenomena, and the limitations on propagation distances.

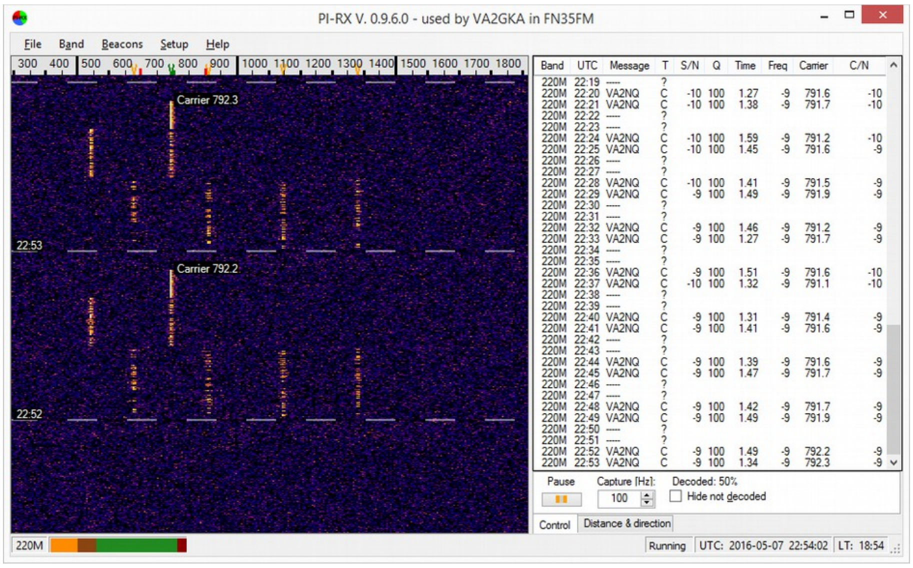


Figure 4: PI-RX used to decode the PI4 part of the transmission

Regarding the power, by convention, 5W or less is usually used on VHF/UHF bands for a beacon, and the design has been dimensioned to operate at this power with low heat dissipation needs. As all transmission modes use only a simple carrier, without any requirement on linearity, the design is simplified by the use of class C. Also, the input stage is over driven to reach class C and limit the lost energy converted into heat by the MOSFET.

The entire project is provided as open source, both the firmware code and the hardware. It should be relatively easy to adapt this design according to your needs. The code is available on GitHub, and the schematic/PCB was shared with CircuitMaker, so that it is possible to view it and fork it using this free software.

2 Design choices and component selection

2.1 Microcontroller

WSPR, PI4 and Morse encoding don't require huge processing power. I chose a simple Atmel 328P, the same that is used in the popular Arduino platforms. Despite the fact the encoding calculations require 32-bits counter, the use of a C compiler (GCC in this case) allows us to not worry about the 8-bit limitation of this Atmel microcontroller and still keep the source code readable.

An interesting point is that this device already includes a management system for I2C and SPI buses. Furthermore, the code remains very simple and requires no advanced libraries.

But first of all, if a simple Atmel 328P is used here that is because there is no need for more, and many radioamateurs will find something they know.

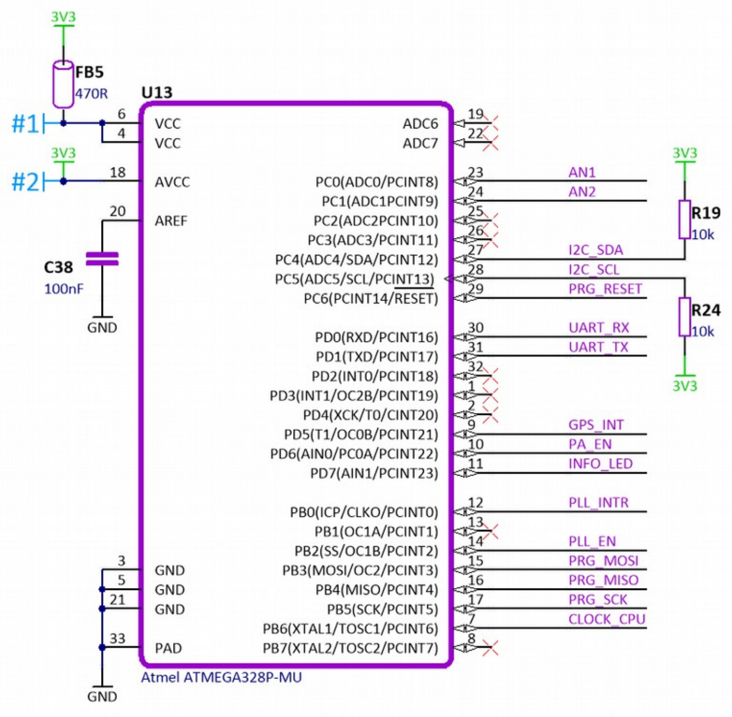


Figure 5: Input and output mapping for the At328P

2.2 PLL

One central component in this design is the PLL used to generate the signal. Initially, I wanted to have a basic circuit able to deliver a signal between 100 MHz and 6 GHz, and basically only tweak the output stage to adjust the impedance matching part. For these requirements, I selected an Analog Devices (ADI) ADF4355, with an integrated VCO. The choice was refined to a cheaper ADF4355-2, which is capable of delivering a signal up to 4.4GHz, keeping in mind these two chips are exchangeable pin to pin.

ADI provides nice software tools to calculate a low-pass filter for these PLL, and to define the programming registers for the ADF4355 series. By the way, a shout out to them for helping designers in their work with high quality tools and documents.

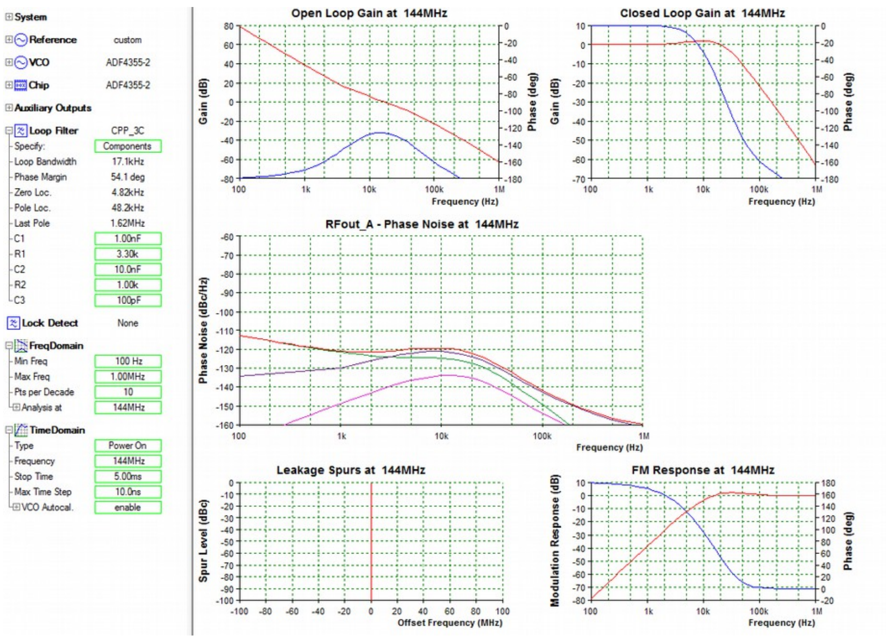


Figure 6: Low pass filter design tool for ADI chipsets

With regard to the noise figure of the generated signal, this PLL provides very good performance that would be very useful for a receiving circuit This is overkill for this beacon, but more is better than less and the output is clean.

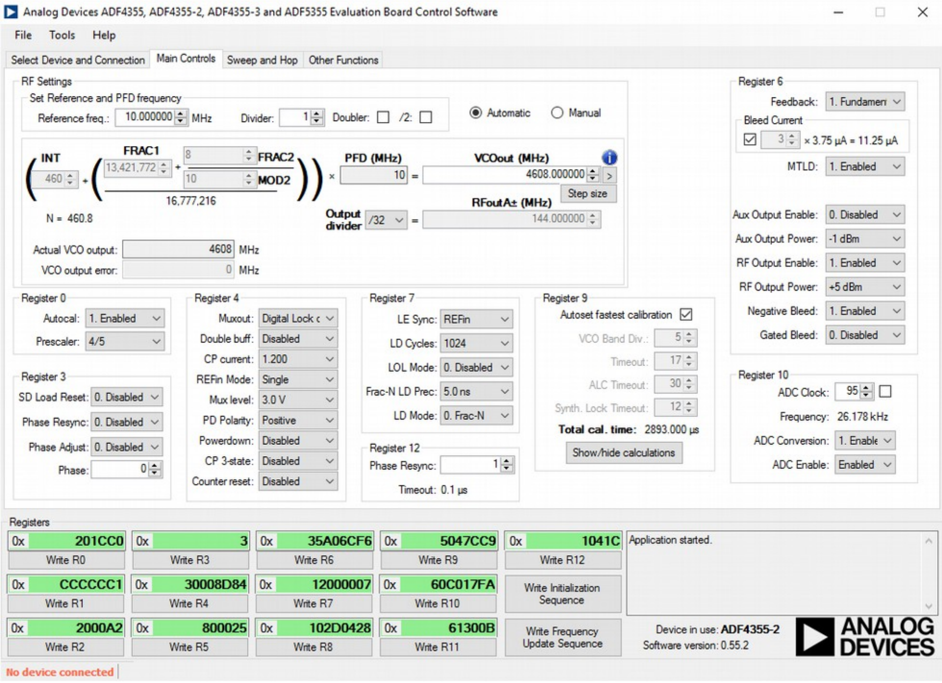


Figure 7: Useful tool to check the ADF4355 configuration

The major downside of the ADF4344 is when the PLL is reprogrammed to a new frequency, a short wideband burst is emitted and you can hear a short sound like –tick– on your radio receiver. This sound is related to the re-synchronization time (approximately 3 milliseconds) and cannot be fixed with this chip.

At the same time, an alternative has been tested and used for bands below 150 MHz, the Silicon Labs Si5351. Silicon Labs clock generators are well known to radioamateurs and already integrated in many designs. Performance is more than acceptable, the unwanted sound is no longer present, and this module requires a smaller and cheaper BOM.

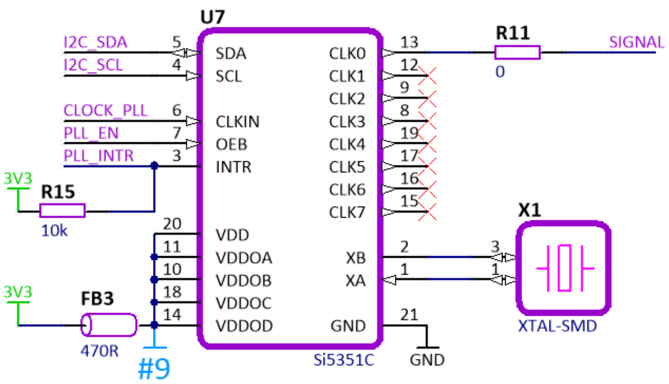


Figure 8: Detail of the Si5351C PLL (rev.D 50 & 144 MHz only)

2.3 GPS & OXCO

To work properly with WSPR at high frequencies, the system requires a highly stable clock. Moreover, frequency accuracy is an important point for a beacon.

Usually, when using a clock stabilized by GPS (GPSDO), a temperature-stabilized crystal is used (TCVXO). It is very stable, but not accurate in frequency. Also, it is gently aligned to an exact frequency with a strong low pass filter. The oscillator will drift to the correct frequency within minutes or hours, depending on the filter. Without temperature stabilization, the crystal could be subject to environmental variation and drift faster than the filter allows the fine tuning.

But there is a problem due to the need for miniaturization. When you replace those bulky oscillators by simple VCXO, you need to adjust the filter circuit to make track faster and aim for a restoration of the right frequency. This is easy to achieve, but on the other hand, you have a weaker filtering, and thus lower phase purity and some possible spurs (which may be similar to a LF modulation). With modes like WSPR or PI4, this will result in interference emissions close to the main carrier, which might interfere with other transmissions.

To reduce this problem, I chose to use a temperature-compensated oscillator (VCTCXO) T604-010.0M from Connor-Winfield. This SMD oscillator does not use an oven, but temperature compensation. Without reaching the same level of accuracy as a traditional TCVXO, performance remains good. However, the cost of this kind of oscillator is much higher than a classic oscillator.

Moreover, depending on the GPS modules used, you can have better or poorer phase purity. GPS modules sold as "precise time pulse" are, of course, very expensive and already have temperature compensated oscillator features and a clean output.

In my design, I made a compromise with a GPS module around $70 and an oscillator around $40 (unit price). Anyway, it is possible to change the design for some more demanding requirements. By using the same GPS manufacturer (uBlox), the code will be very easy to adjust, and to change the oscillator itself, it's only a question of changing the footprint.

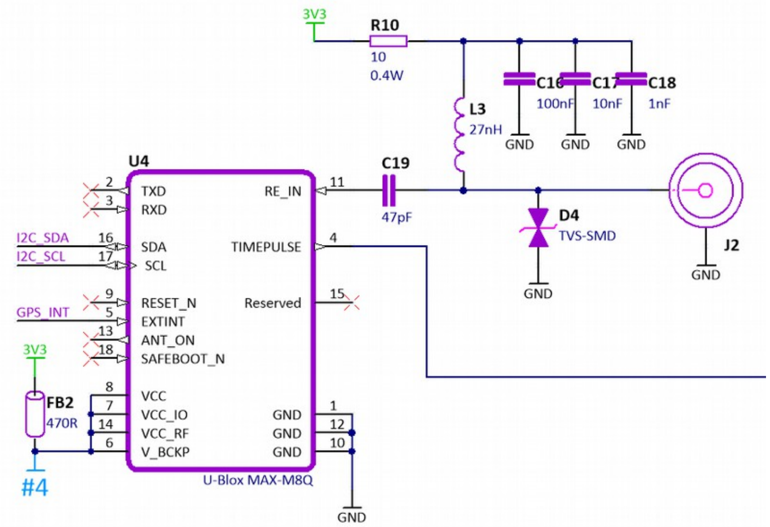


Figure 9: uBlox MAX8 GPS module used in a very basic way

In closing, I would like to remind the reader that this design is not the best or ultimate solution, but only the compromise I chose. Not perfect to me, but it does the job !

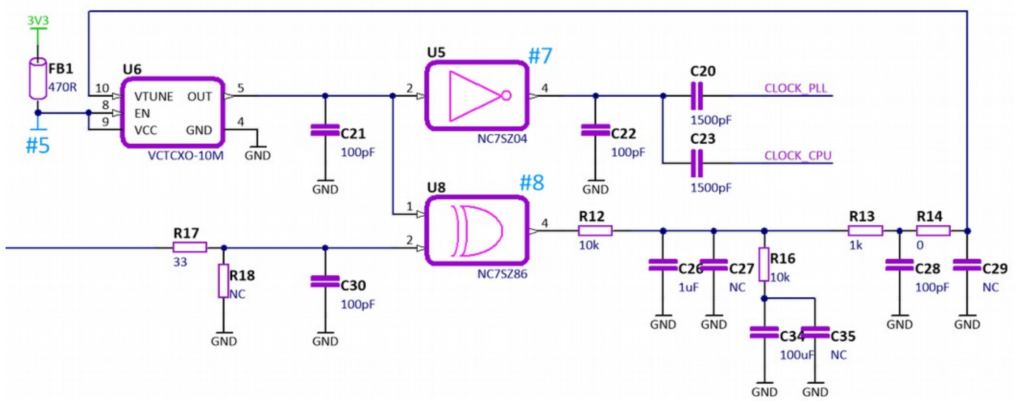


Figure 10: Simple 10 MHz PLL disciplined by GPS (GPSDO)

2.4 DC/DC converter

One of the initial goals was to be able to supply the device with a standard lead-acid battery. Thus, the beacon includes a DC-DC switching regulator supporting an input voltage between 10 and 15 volts. Technically, two DC-DC are used, one for delivering 8.5V and another for 3.8V. 8.5V is used directly used to feed the power FET, and an LDOs is used on the 3.8V output, to provide a final output voltage of 3.3V, used in sub-circuits like the GPS, the PLL, the microcontroller and the preamplifier.

The DC-DC chip is a Linear LT3480, supporting up to 2 amps, with efficiency around 85%. There is no need for extra calculation, the manufacturer gives lots of application examples, very close to this application case.

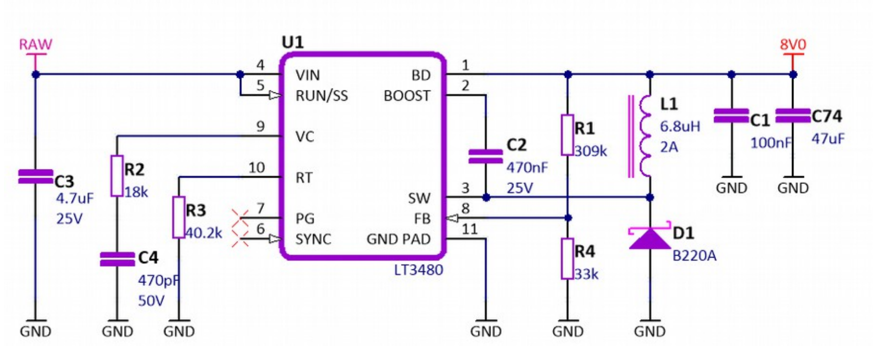


Figure 11: Detail of one of the two DC-DC power supplies

2.5 RF power stage

A CEL/Renesas NE5550979a has been selected to cover a 5 watts output power on 4 bands (50-144-222-432 MHz). The manufacturer's documentation is exceptionally good, and several reference designs are provided for some professional bands, but close to our frequencies.

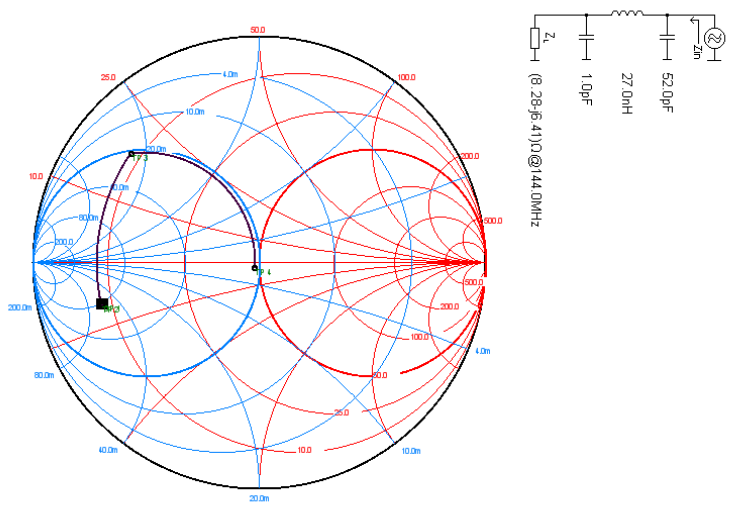


Figure 12: Simple impedance matching, using a pi cap-ind-cap topology

This PA is intended to operate in class C to maximize its efficiency so that there is no need of linearity. Also, the PA is intentionally over-driven and baising current is only slightly used.

Some interesting calculations should be achieved on this part, especially on the impedance matching circuit. However, the S2P files are provided by the manufacturer and the tuning circuit can be easily deduced with a simple tool.

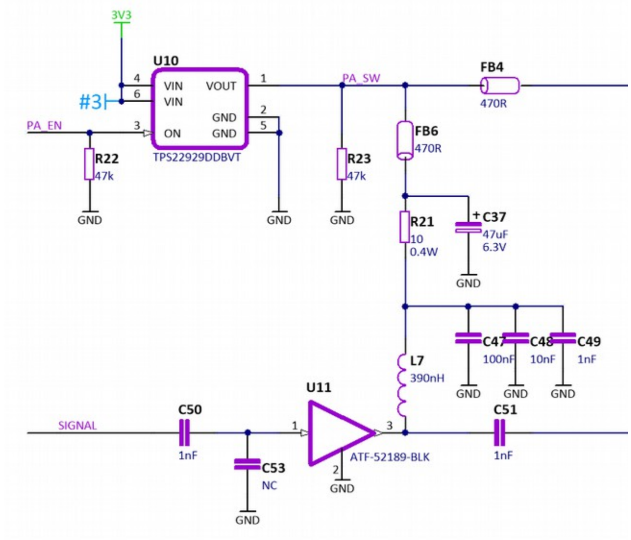


Figure 13: Driver using an ATF-52189

The PA is followed by a 5 poles filter, primarily used to filter harmonic frequencies due to the fact that the PA has a saturated input. In addition, this filter provides a minimal protection against other beacons that may be working in a close range.

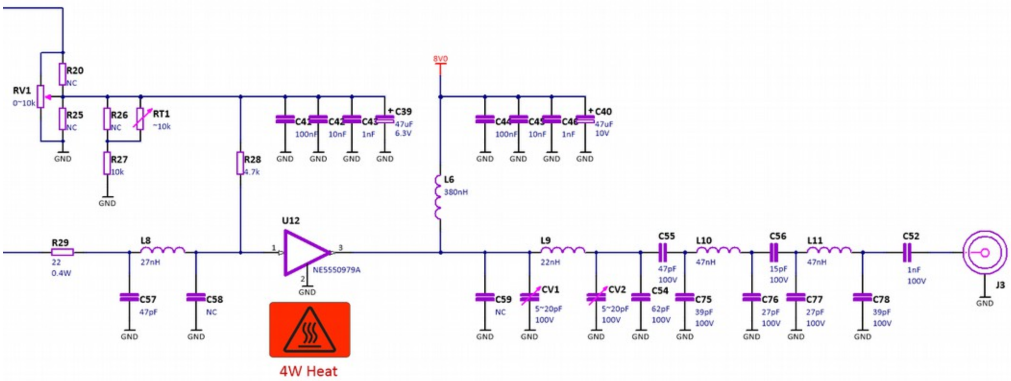


Figure 14: PA part, with a bandpass filter (version 144 MHz)

2.6 Mechanical design & heat dissipation

The entire circuit uses a credit card format, but at the same time, it follows the dimensions used for small aluminum enclosures. When the circuit is tuned at a good efficiency (> 70%), and when the output power does not exceed 5W, the heat can be dissipated by a simple aluminum enclosure (in a ventilated environment).

Note that circuit was also designed to fit into a machined aluminum housing. Rounded edges and holes are provided to attach it on a support with a good thermal contact on the PA area.

A thermistor was also used to limit the voltage on the biasing circuit and avoid a thermal runaway.

Figure 15: Example of the final product, using a small anodized aluminum housing

3 Manufacturing

The PCB itself was routed in 4 layers, with two inner layers reserved for the ground plane and the power plane. Manufacturing using professional services is required, and also allows via plating. Nowadays, companies specialized in prototyping can provide this kind of PCB at low and affordable cost to an amateur.

All the components use SMC type, placed only on the top face, with an exception for the SMA connectors. The smallest size is 1206 (0603), and a simple pair of tweezers is needed for component placement.

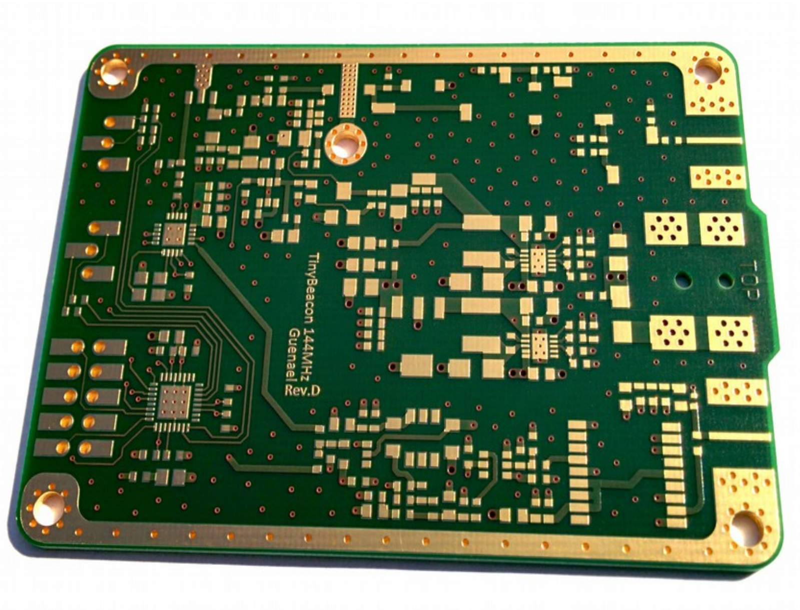


Figure 16: Fresh PCB, using professional manufacturing processes.

4 Measurement, calibration & performance

The calibration of this circuit is done through the polarization potentiometer close to the PA, and two variable capacitors which allow tuning the impedance matching circuit, to achieve maximum efficiency.

To control the output power and efficiency, a simple power meter and ammeter is required.

Figure 17: Tuning the circuit by adjusting 2 caps and 1 trimmer

As of the writing of this article, the PCB has been in operation for a period of 4 months, and the design was later refined in two versions (rev.B and rev.C). This document presents the rev.D version. The only point to watch is good heat dissipation. Regarding the antenna matching, the SWR of the antenna must remain below 1.5.

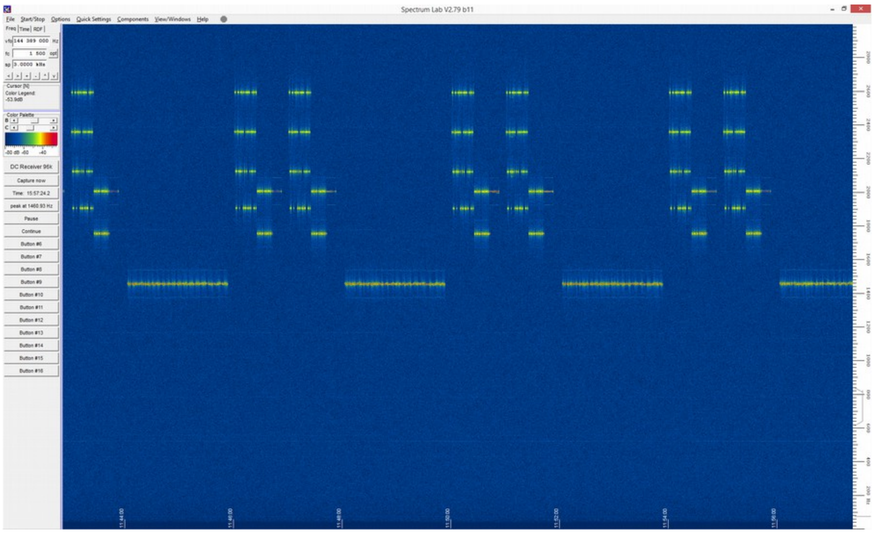


Figure 18: Pattern of alternation of WSPR and PI4

Observing the transmission with a spectrum analyzer shows some imperfections. Some spurs around the main carrier can be measured around -45 dB from the signal. These spurs can be decoded by stations close to the beacon, which is not desirable, but at the same time, this factor of -45 dB is quite good for a transmitter.

5 Microcontroller code

The source code of the project was shared on GitHub. The encoding parts of WSPR and PI4 were taken from their respective projects.

In terms of actions performed, the program first sets up the GPS, defining a 10 MHz output frequency used to discipline the VCTCXO. At the same time, GPS coordinates are extracted, and the locator is calculated. Information about time is also recovered, to start WSPR and PI4 at the right time.

Then the programming registers of the PLL are stored in memory, and will be injected directly in the sequence to set the tone frequencies.

The sequencer then ensures to send PI4 protocol twice (MGM + CW + Tone), and send WSPR on time. Once done, it loops and restarts the sequence.

Further work could be done to write and provide a better library for PLL and GPS modules. For now, the key was to share a functional beacon, easy to understand and modify.

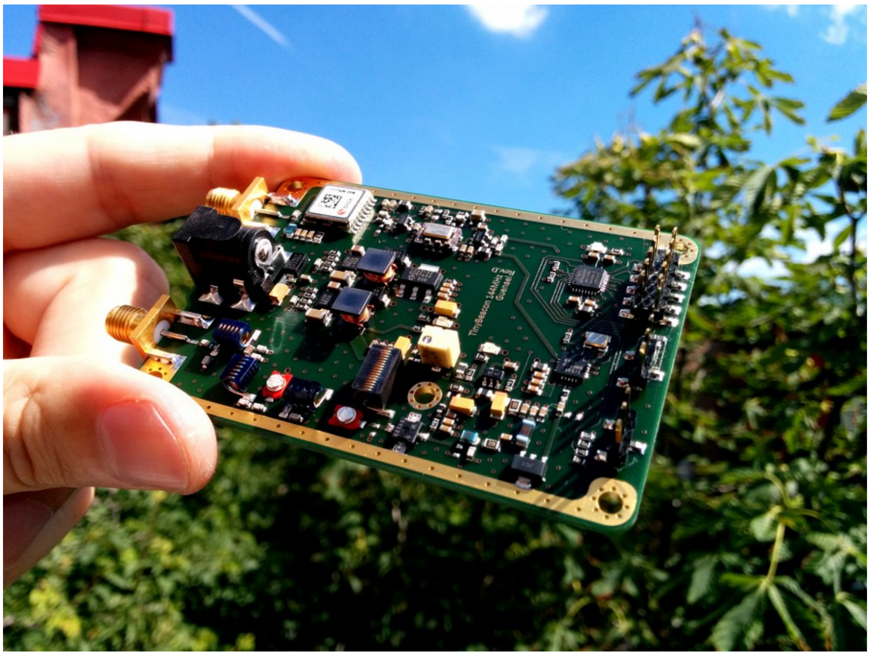


Figure 19: rev.D, definitely the best version

6 Setup & operation

All prototypes have been put in operation with the help of Norman, VE2VAX on his estate at Mount Yamaska (FN35). A total of 4 beacons were set up using these frequencies : 50.295, 144.491, 222.295, 432.302 MHz, and using the callsign VA2NQ.

Different versions have been put in operation from the rev.B. An interesting point : During the design of the rev.C, less expensive components were used for the oscillator and the GPS, but these choices have been revisited on the rev.D and were rolled back. For an additional cost of one hundred dollars, the quality is greatly improved, and this choice was then retained, because we usually expect great quality from a beacon.

Aside from this project, two PCBs have also been developed for a reduced output power of 100 mW. These can produce signals between 100 MHz and 4.4GHz for the first model (ADF4355-2) and between 100 kHz and 150 MHz for the second (Si5351). These frequency generators stabilized by GPS could be useful, for example to drive a receiver, to feed a transverter, to be used as instrumentation or to bench test an antenna.

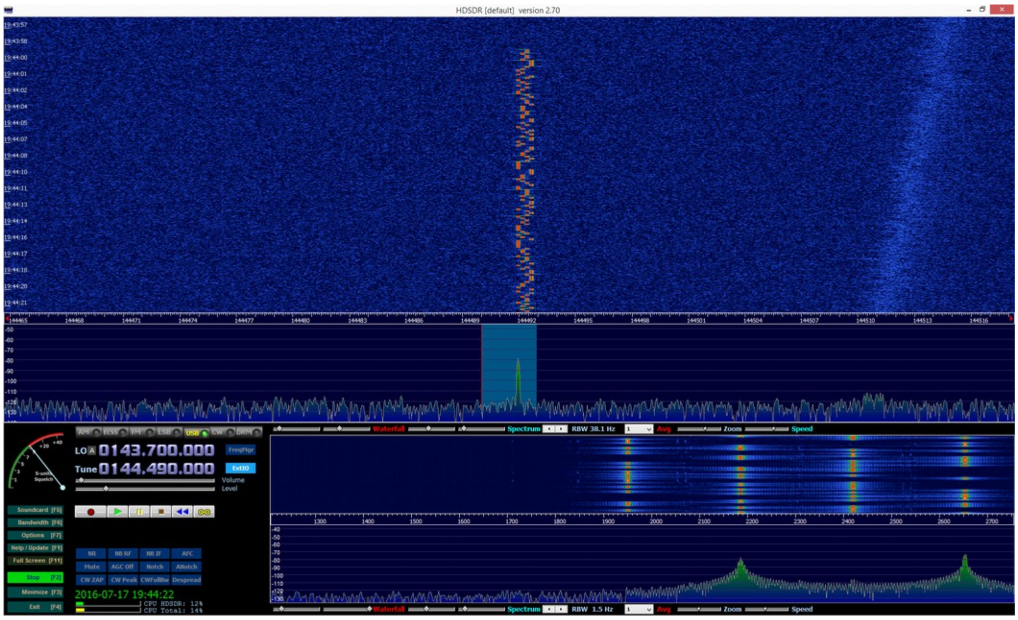


Figure 20: HDSDR the king of SDR receivers, used with a RTL dongle to receive the beacon signals

7 Source code & Open design

Hardware [9] : http://circuitmaker.com/Projects/Details/Guenael-VA2GKA/TinyBeacon-144-RevD

Others versions (50, 222, 432MHz) :
http://circuitmaker.com/User/Details/Guenael-VA2GKA

Software [10] : https://github.com/Guenael/tinybeacon

8 Reception

Related to this project, I wrote two WSPR receiver/decoder running on Linux/FreeBSD directly as a daemon, and usable with Raspberry PI (version 1, 2 or 3). Signal conversion is done using a SDR receiver like the famous RTL-SDR.

The goal was to have a separate setup from the station, inexpensive and which can run non-stop. This daemon decodes the signals, reports it on the Internet and operates on a low-power (fanless) device like a Raspberry PI.

Code source for this project:

https://github.com/Guenael/rtlsdr-wsprd

9 Conclusion

The primary goal of this project is to allow an affordable installation of VHF & UHF radio beacons, to provide useful tools allowing better detection and better mapping of propagation phenomena. I sincerely hope this work can be reused by the radioamateur community to setup new beacons throughout the world.

I also plan to adapt my design for the hyper-bands and cover 1.3 GHz to 10 GHz. These bands, used by too few radioamateurs, are technically difficult to master, but are also very interesting to explore.

Finally, this project will also continue with a side project of a low cost automated reception system for VHF/UHF bands. Always with the same idea to provide some easy to use tools to put in place automatic reporting systems, and to not miss the next opening that amazes us so much. :)

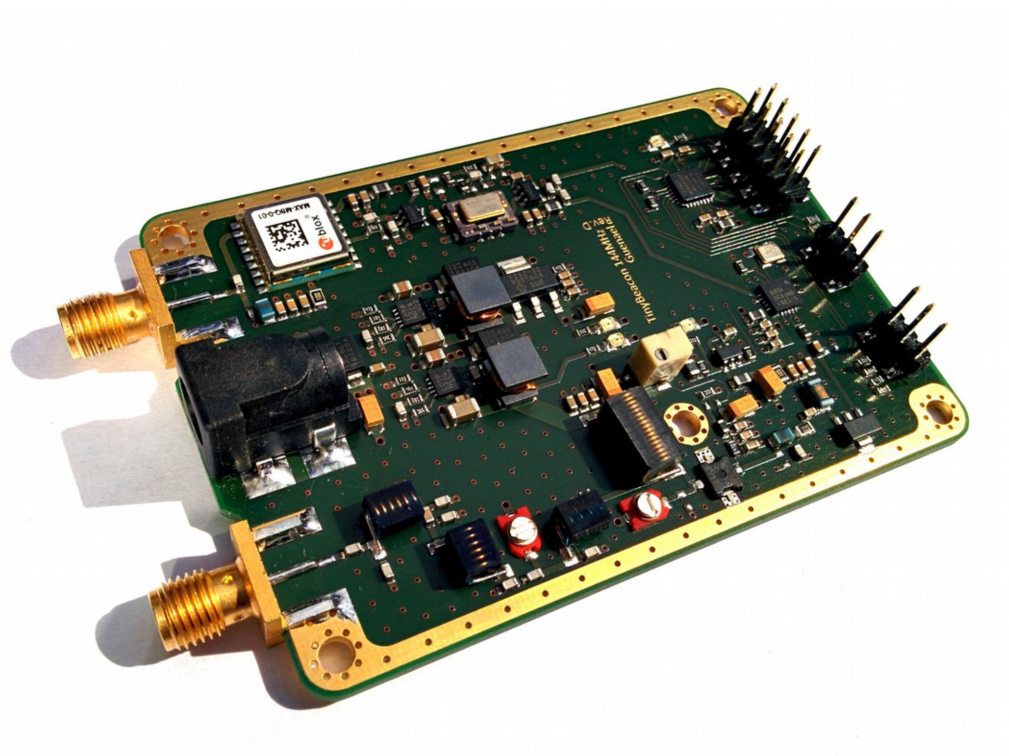


Figure 21: TinyBeacon PCB rev.D, 144 MHz version

Greetings

VE2VAX Normand Labossière, for helping me to set up a new set of beacons in the Montreal area, using a great site and a clean setup using autonomous power.

VA2WHY Rupert Brooks, for his help & feedback on this article

K1JT Joseph, for sharing his knowledge and providing us these nice protocols

OZ2M Bo, for his work on Pi4 and feedback on protocol implementation

References

[1] https://en.wikipedia.org/wiki/WSPR_(amateur_radio_software)

[2] http://physics.princeton.edu/pulsar/k1jt/

[3] http://wsprnet.org/drupal/wsprnet/map

[4] https://pavel-demin.github.io/red-pitaya-notes/sdr-transceiver-wspr/

[5] https://www.raspberrypi.org/

[6] http://beagleboard.org/

[7] http://sdr.osmocom.org/trac/wiki/rtl-sdr

[8] http://rudius.net/oz2m/ngnb/pi4.htm

[9] http://circuitmaker.com/User/Details/Guenael-VA2GKA

[10] https://github.com/Guenael/tinybeacon

Revision

1 Initial publication