

QRS Labs QRSS Arduino Shield Beacon Kit

1. Introduction

Thank you for purchasing my QRSS Arduino shield beacon kit. QRSS is a slow morse, weak signal mode capable of worldwide HF propagation using a fraction of a watt of RF output power. This kit is a crystal-controlled Arduino-compatible shield capable of transmitting QRSS on one band. It supports on/off keying and variable FSK. The sample sketch provides QRSS, FSK/CW and DFCW modes but any slow-signal modes can be supported with a suitable sketch, feel free to experiment!

The transmitter is designed to be powered with 5V DC supplied by the Arduino board.

2. Design

The circuit diagram is shown on page 2. It consists of a simple Colpitts oscillator (Q1), a buffer stage (Q2), and a power amplifier (Q3) followed by a 7-element low pass filter. The Arduino keyer sketch shifts the oscillator frequency a few Hz via the red LED which behaves as a varactor diode. Q4 and Q5 provide switching of the output, so that on/off keying can be supported as well as simple FSK modes. Power output is approximately 100-150mW but depends on the band.

The kit comes in three versions (80/40/30m), please follow the instructions for your version. Parts placement is defined by the printed legend on the PCB, so please observe it carefully, paying particular attention to the correct orientation of the semiconductors. Please refer to the parts list for the correct capacitor and inductor values to use for the 80m, 40m and 30m versions.

3. Parts list

Resistors

R1 470K (yellow-purple-yellow-gold)
R2 10K (brown-black-orange-gold)
R3 180 ohm (brown-grey-brown-gold)
R4 330 ohm (orange-orange-brown-gold)
R5 150 ohm (brown-green-brown-gold)
R6 6.8K (blue-grey-red-gold)
R7 12K (brown-red-orange-gold)
R8 2.2K trimmer potentiometer
R9 12K (brown-red-orange-gold)

Miscellaneous

Printed Circuit Board
Quartz crystal: 3.500/7.000/10.140MHz
Wire for winding toroids
4-way DIP switch
Arduino Reset button
4x Arduino sockets (2x 6-pin, 2x 8-pin)

Semiconductors

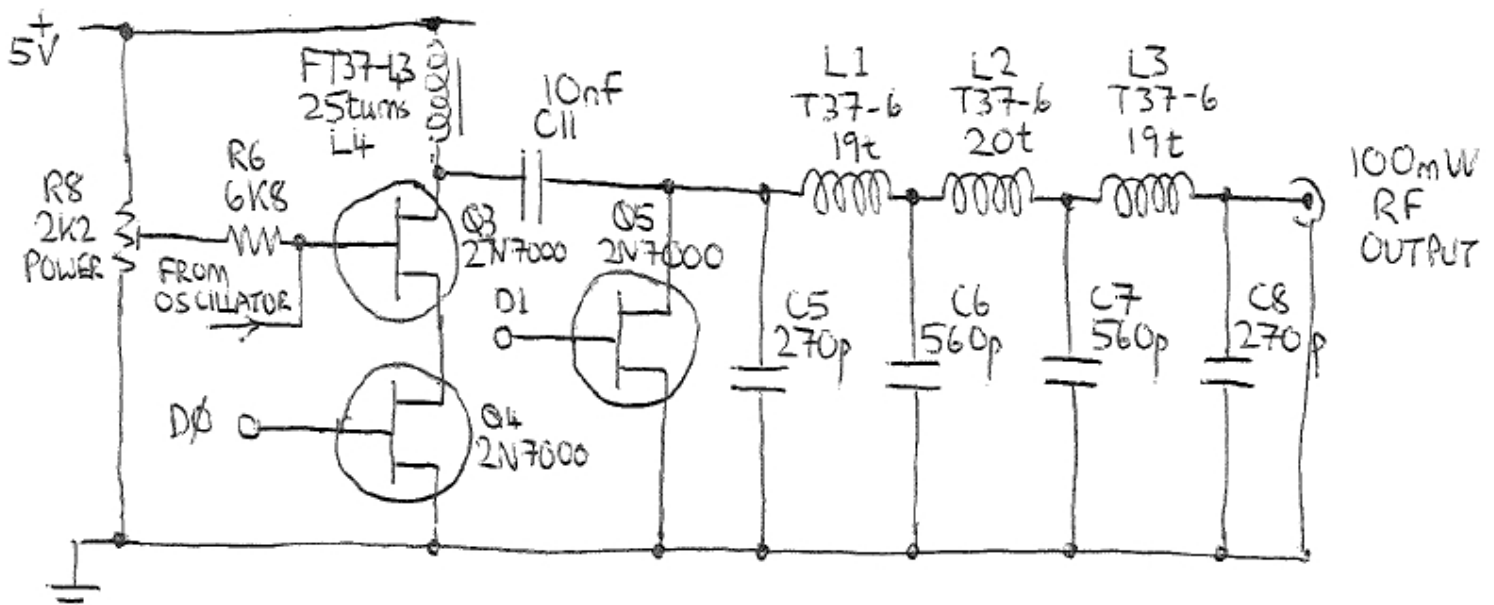
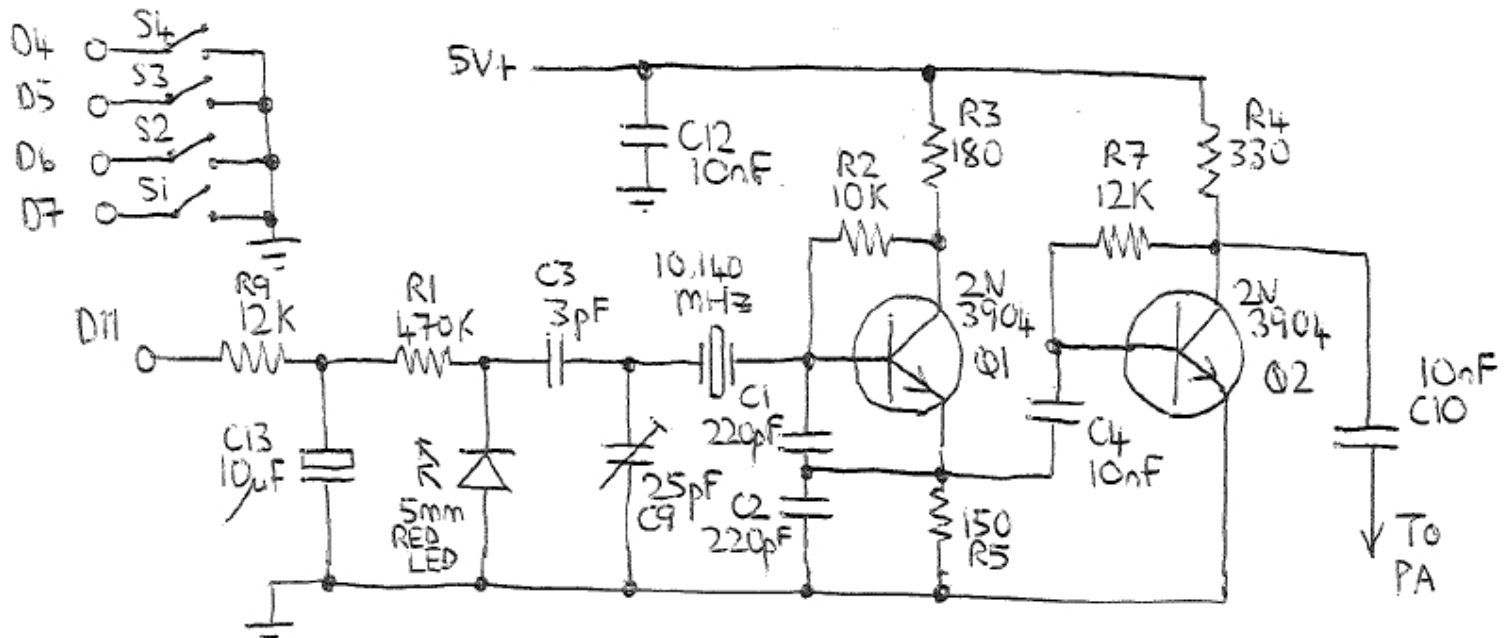
LED 5mm Red LED
Q1,2 2N3904 transistor
Q3,4,5 2N7000 transistor

Inductors

	80m version	40m version	30m version
L1	25 turns, T37-2 (red)	19 turns, T37-6 (yellow)	19 turns, T37-6 (yellow)
L2	27 turns, T37-2 (red)	21 turns, T37-6 (yellow)	20 turns, T37-6 (yellow)
L3	25 turns, T37-2 (red)	19 turns, T37-6 (yellow)	19 turns, T37-6 (yellow)
L4	25 turns, FT37-43 (black)	25 turns, FT37-43 (black)	25 turns, FT37-43 (black)

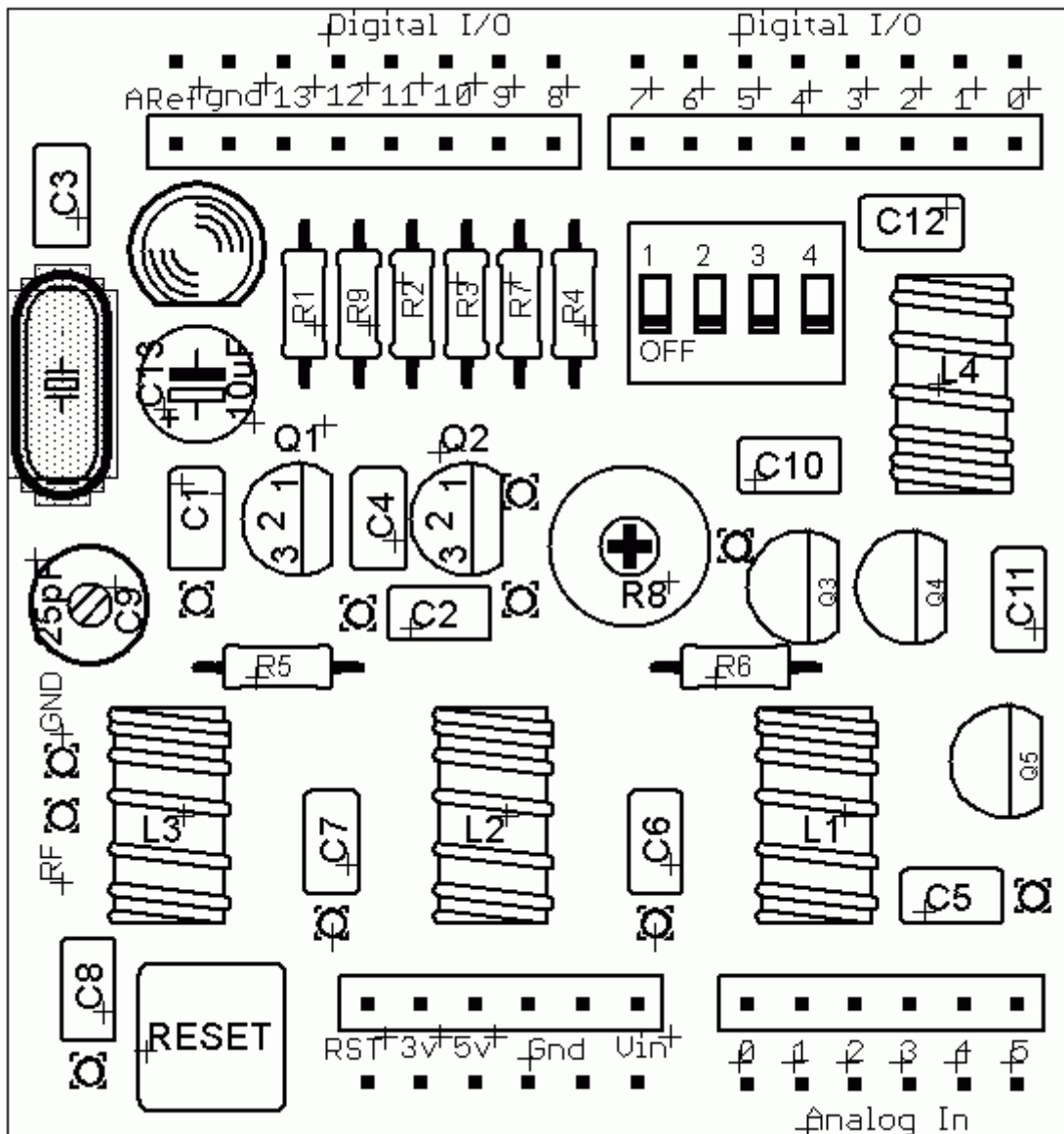
Capacitors

	80m version	40m version	30m version
C1,2	680pF	470pF	220pF
C3	3pF twisted wire	3pF twisted wire	3pF twisted wire
C4, 10, 11, 12	10nF	10nF	10nF
C5, 8	470pF	270pF	270pF
C6, 7	1200pF (1.2nF)	680pF	560pF
C9	25pF trimmer capacitor	25pF trimmer capacitor	25pF trimmer capacitor
C13	10uF	10uF	10uF



4. Construction

Follow the parts layout diagram below carefully, paying particular attention to the correct orientation of the semiconductors (transistors and LED) and the correct orientation of capacitor C13. Read the notes below, before starting.



4.1 Trimmer potentiometer R8

Unfortunately the supplied potentiometer does not fit into the holes in the PCB, which are too small for the connection tabs of the potentiometer. In the interests of providing you with a low cost kit, I did not have another set of PCB's manufactured to correct this design fault. Instead, I recommend the following simple steps to work around this minor inconvenience.

First, you should bend the last 5mm (1/4-inch) or so of the connection pins of the trimmer potentiometer inwards towards the centre of the component, so that they have a 90-degree bend. They will look something like feet now, on the bottom of legs. Now you can apply a generous blob of solder to each of the three holes on the PCB, and "surface mount" the component on top of the PCB, without its pins going through the holes. The holes in the PCB are through-hole-plated so do not worry, there is connection between both sides.

I recommend soldering this component FIRST, so that you have space to work easily.

4.2 Winding the inductors

Remember that each time the wire goes through the centre of the toroid counts as one turn. 35cm of wire should be enough for 25 turns. Label the toroids aids identification later! Wind the coils as per the number of turns in the table in section 3 above.

Trim the ends of the wire and scrape the enamel off and tin them with solder. Alternatively, the ends of the wire can be held in a blob of hot solder on the soldering iron, and the enamel will burn off. You should use a well-ventilated work space to avoid the unpleasant fumes.

4.3 The “gimmick” capacitor

C3 is a small capacitance of a few pF, whose exact value is not very important. To make it, cut about 50mm of the magnet wire and solder the ends to the PCB where C3 is marked. Twist the wire in a spiral and cut the tip so we end up with two wires about 15mm long. Later, the amount of frequency shift is adjustable within the Arduino sketch software. But if it seems still too large, you can cut a little off the two wires, to reduce the capacitance; this will decrease the amount of shift.

4.4 Remaining components

Solder the remaining components on the top half of the board: resistors, capacitors, transistors etc. I recommend soldering the Arduino sockets LAST, because they are bulky and they will get in the way of soldering the other components.

5. Adjustment

TURN R8 FULLY CLOCKWISE. This is **Very Important** - in the fully anti-clockwise position the PA transistor will be biased permanently on and get very hot, probably destroying itself within seconds.

At this stage you need to plug the shield onto an Arduino, load an Arduino sketch for FSK/CW and run it. Connect the output of the LPF to a 50-ohm dummy load (two ¼-Watt 100-ohm resistors in parallel will do). Connect some means of measuring power such as a power meter or an oscilloscope. Slowly turn R8 anti-clockwise and monitor the output power. You should be able to achieve more than 100mW before you notice that power starts to dip; then turn R8 back to the peak power point. With a 50-ohm dummy load, 100mW on an oscilloscope is 6.3V peak-to-peak (Watts = peak-to-peak voltage squared, divided by 400). If you turn R8 too far you may destroy Q3. Q3 should not get noticeably warm. If it does, then R8 is too far anti-clockwise.

6. Setting the keyer speed and QRSS mode

A 4-way DIP switch is provided on the shield board, to provide the user with an easy way to control aspects of the kit operation such as mode (e.g. QRSS, DFCW or FSK/CW), or speed. The DIP switches are read by the Arduino, which interprets them according to the sketch you have loaded. It is up to you, how you want to use the switches! In the sample sketch, 2 are used for the mode (QRSS, DFCW, FSK/CW) and two for the speed.

7. Tuning

It is very important to realise that by far the hardest part of any successful QRSS beacon operation, is tuning the oscillator to the correct frequency. Most 30m QRSS stations monitor a narrow 100Hz-wide band from 10,140,000 to 10,140,100. If you are much outside this, the chances are that nobody will see your signal. It is therefore essential either to adjust your output frequency using an accurately calibrated frequency counter, or an accurately calibrated receiver. Adjust the frequency using trimmer C9, aiming at first for 10,140,050 (for example).

80m and 40m popular QRSS frequencies are the 100Hz-wide band from 800-900Hz above the band edge.

The easiest way to adjust the frequency and frequency shift of your keying, is to install the Argo software (download: <http://www.sdrham.com/argo/index.html>) on your PC, and monitor the output frequency on a 30m receiver. A small length of wire may be necessary at the RF output of the transmitter, to ensure it is picked up by your receiver antenna. Use Argo in the horizontal, 3s dots Slow mode. The image should look something like this (for example, in the FSK/CW mode):



The “height” of the FSK should be around 4-5Hz, and you can adjust this from the Arduino sketch. If the height remains too large, you can cut a piece off the “gimmick” capacitor C3. Do not waste spectrum by using more! Less may be sufficient and preferable!

8. Operation

Connect the antenna and remember, QRSS is all about patience! You should join the QRSS news group http://cnts.be/mailman/listinfo/knightsqrss_cnts.be and announce that your beacon is on the air. You should get reception reports by email and see your signals on the various online “grabbers” (see <http://digilander.libero.it/i2ndt/grabber/grabber-compendium.htm> for a set of links to “grabbers” worldwide). With a reasonable dipole

antenna worldwide reception on 30m (and perhaps 40m and 80m!) should be easily achievable with this kit, when the propagation conditions are on your side.

9. Resources

- 1) Please join the Yahoo group <http://groups.yahoo.com/group/qrplabs/> to for new kit announcements, to discuss any problems with the kit, enhancements you've made, or just to tell everyone how much fun you're having.
- 2) For general information and more QRSS projects, links to other QRSS resources etc., see <http://www.hanssummers.com>.
- 3) The sample Arduino sketch can be downloaded at <http://www.hanssummers.com/qrssarduino> .
- 4) This kit was designed to accompany one of the chapters I wrote for the ARRL book "Ham radio for Arduino and Pixaxe", see <http://hamradioprojects.com/>
- 5) Knights QRSS email forum: http://cnts.be/mailman/listinfo/knightsqrss_cnts.be
- 6) Worldwide grabbers <http://digilander.libero.it/i2ndt/grabber/grabber-compendium.htm>