QRSS Labs Multi-mode QRSS Beacon Kit

1. Introduction

Thank you for purchasing our Multi-mode QRSS beacon kit. This kit is capable of automated transmission of a range of weak signal modes that are capable of worldwide HF propagation using a fraction of a watt of RF output power.

The kit supports the following modes:

- QRSS mode (plain on/off keyed slow CW)
- FSK/CW mode (frequency shift keyed slow CW)
- DFCW mode (dual frequency CW, dit's and dah's on different frequencies)
- WSPR mode (Weak Signal Propagation Reporter)
- Slow-Hellschreiber (frequency shifted slow Hellschreiber)
- Hellshreiber (full-speed standard Hellschreiber, and half-speed Hellshreiber)
- CW (plain CW)
- Customisable FSK patterns

Other features:

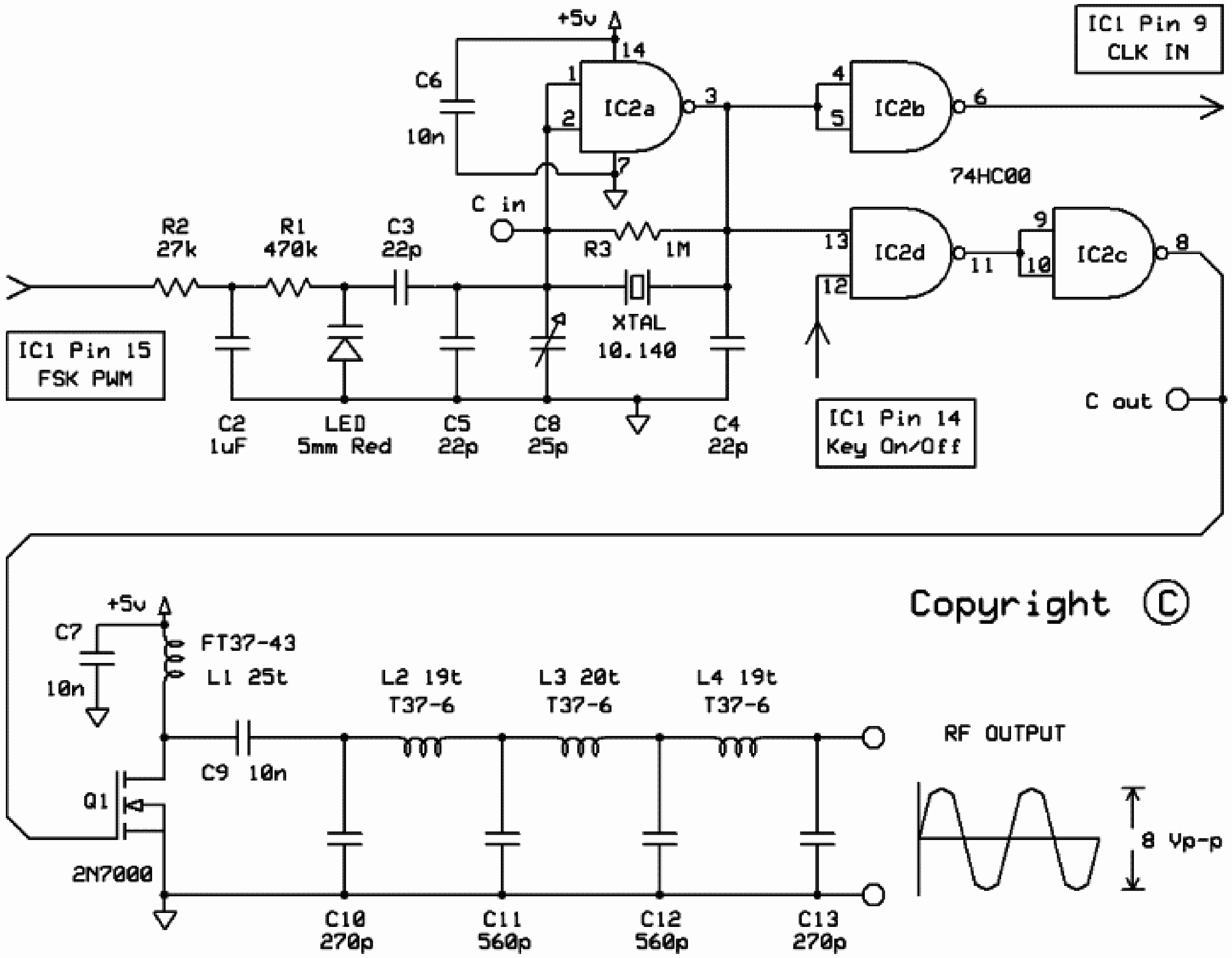
- 24-character LCD + two-button user interface
- User-programmable (callsign, message, speed, FSK, mode, etc.), settings stored in EEPROM
- GPS interface, for locking the frequency in slow-speed modes
- On-chip generation of WSPR encoded message (no PC required)
- WSPR maidenhead locator can be generated from GPS-derived latitude/longitude
- Selectable “frame” size, for stacked QRSS reception
- Plain CW callsign identifier at selectable interval
- Produces 150mW RF output, or AF output for driving an SSB transceiver
- Higher output power by additional PA transistor and/or higher PA supply voltage

The transmitter is designed to be powered with 5-6V DC, which could come from a mobile phone charger, wall wart, or even four 1.5V batteries connected in series. Do not use more than 6V: this may kill the microcontroller. Best results will be obtained with a well-regulated and well-smoothed supply.

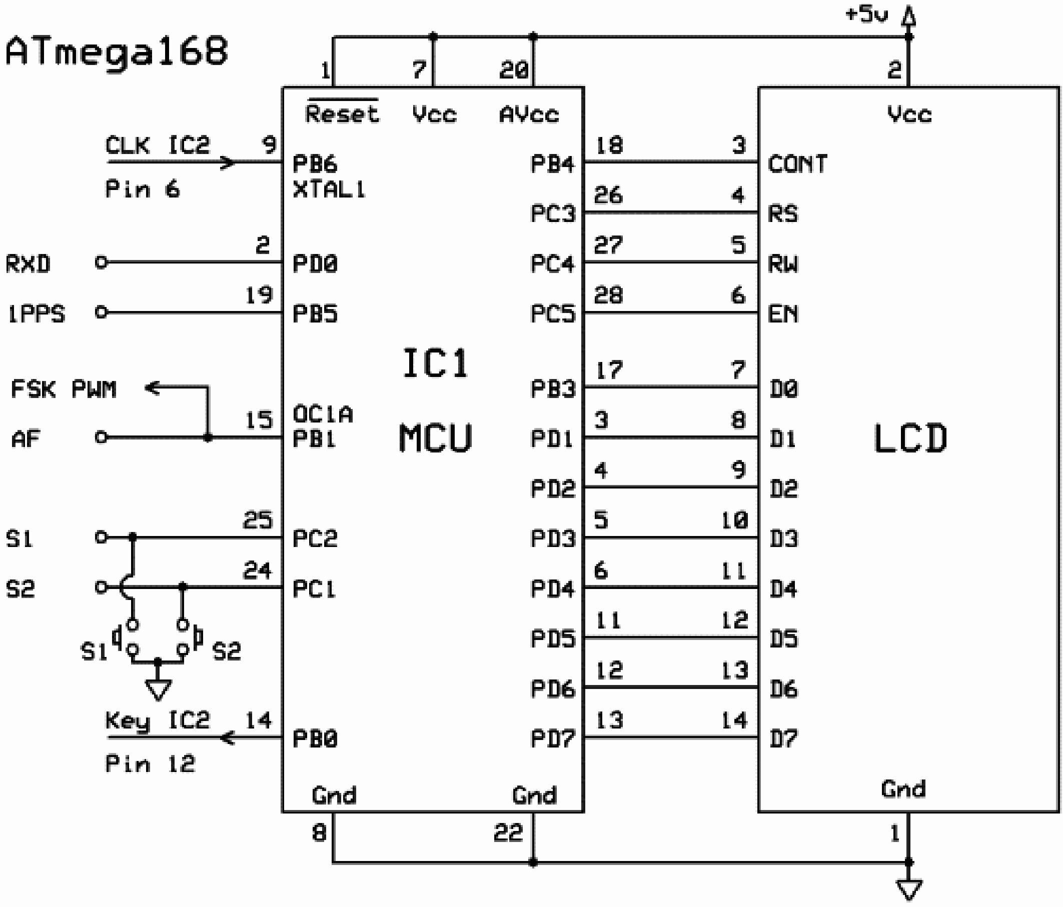
Recommended approach to building the kit: This is a simple design but there are a large number of features which provide a great deal of flexibility. Read this WHOLE manual and understand it! The manual refers to the 30m version, for convenience, so remember this where appropriate. Follow the construction section to build the kit. Use a receiver connected to your PC, with a slow-signal decoding program such as Argo, to monitor your signal, experiment and understand the various features before connecting an antenna! Good performance depends on proper set up: **Pay particular attention to the calibration section, to tune and set up the kit.**

2. Design

The circuit diagram is shown on page 2 (thanks John VK6JY for the computer-drawn diagram). One gate of a 74HC00 quad NAND gate IC (IC2) is used as crystal oscillator. The ATmega168 AVR microcontroller (IC1) is pre-programmed with firmware to control the LCD, buttons, and GPS interface. The microcontroller controls the frequency of the crystal oscillator, shifting it via a reverse-biased 5mm red LED which behaves as a varactor diode. The microcontroller also keys the transmission on/off using one of the NAND gates in IC2. A single 2N7000 transistor (Q1) operates as the power amplifier, producing over 150mW from a 5V supply. Finally a standard 7-element low pass filter removes harmonics of the transmission frequency.



Copyright ©



QRSS-LABS

MULTI-MODE

QRSS KIT

July 2012

3. Parts List

3.1 Resistors

R1	470K (yellow-purple-yellow-gold)
R2	27K (red-purple-orange-gold)
R3	1M (brown-black-green-gold)
R4	No resistor – space is provided on the PCB only for future expansion
R5	No resistor – space is provided on the PCB only for future expansion

3.2 Capacitors

C1	No capacitor – space is provided on the PCB only for future expansion
C2	1uF (blue colour, marking 105)
C3, 4, 5	22pF (ceramic, marking 22)
C6, 7, 9	10nF (ceramic, marking 103)
C8	30pF trimmer capacitor

3.3 Inductors

L1	25 turns, FT37-43 core (black)

3.4 Semiconductors

D1	5mm Red LED
Q1, 2	2N7000 (note, only one is provided in the kit. The second may be added for more power).
IC1	Pre-programmed ATmega168 microcontroller
IC2	74HC00

3.5 Miscellaneous

Push buttons S1 and S2
1 row, 24 character LCD
Printed circuit board
Wire for winding toroids
Socket for IC1
QRSS crystal for chosen band

3.6 Band-dependent components

The following components have a different value depending on which band you chose. Please refer carefully to the following table and ensure the components are correctly chosen:

	160m version	**80m version**	**40m version**	**30m version**
C10, 13	820pF	470pF	270pF	270pF
C11, 12	2200pF (2.2nF)	1200pF (1.2nF)	680pF	560pF
L2, 4	30 turns, T50-2 (red)	25 turns, T37-2 (red)	19 turns, T37-6 (yellow)	19 turns, T37-6 (yellow)
L3	34 turns, T50-2 (red)	27 turns, T37-2 (red)	21 turns, T37-6 (yellow)	20 turns, T37-6 (yellow)

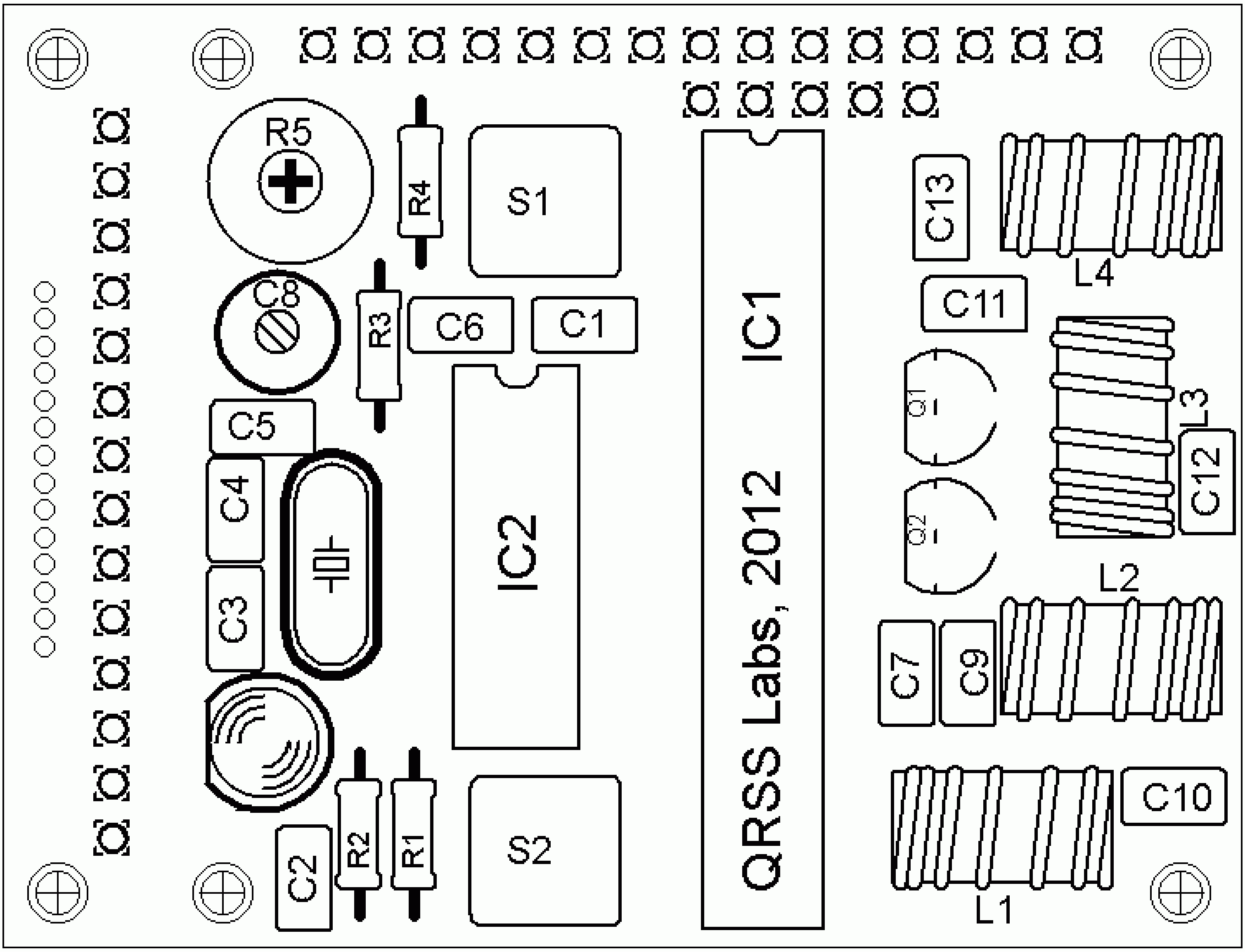
4. Construction

The kit comes in several versions for different bands. 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 your band.

The PCB is quite small and the parts are close together. You are recommended to use a low wattage iron with a fine tip, and fine solder e.g. 1mm diameter or less. Take care not to overheat the PCB and risk damaging it. A well-lit area and magnifying glasses can assist.

Note that components R4, R5 and C1 are not required or supplied in the kit. Q2 is not supplied, an additional 2N7000 may be installed here to increase the output power (see below). IC1 (the microcontroller) has an IC socket, in case in future you wish to change the microcontroller e.g. for a firmware upgrade for new features, etc., or in case you wish to program it yourself. IC2 (the 74HC00) does not have a socket.

Please refer to the parts placement diagram below.



Pay special attention to the orientation of the semiconductors. For IC1 and IC2, the dimple in the PCB silkscreen must be aligned with the dimple at the top of the IC socket and the IC. The LED image on the PCB shows a flat along one side, which should be aligned with the flat on the LED. Similarly, the transistor.

The order of construction is not important but a good principle to follow is to install the smaller components first, so that the larger ones do not prevent easy access. One suggested order of construction is:

1) Install IC2 and the socket for IC1

2) Install all resistors and capacitors, the quartz crystal and LED

3) Install switches S1 and S2, if not using off-board switches

4) Install the transistor Q1

5) Wind and install the toroids

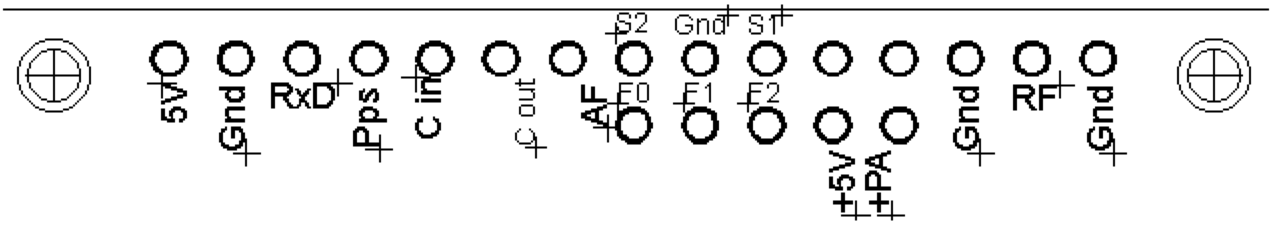
6) Install the LCD – first consider how you want to box the kit (see LCD mounting options below).

Installation of the LCD requires particular care, because the pin-spacing of the LCD is only 0.05-inches. Use of a fine tip soldering iron is advised, and check carefully for shorts between pins, using a magnifying lens and Digital Voltmeter (DVM). Ensure correct orientation of the LCD as per the kit photograph.

When winding the toroids, 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 to aid identification later! Trim the ends of the wire, scrape the enamel off and tin them with solder. As an alternative to scraping the enamel off, the wire ends may be held in a hot blob of solder on the iron tip for a few seconds, and the enamel will bubble away. Check continuity on the board with a DVM.

5. Hardware Options

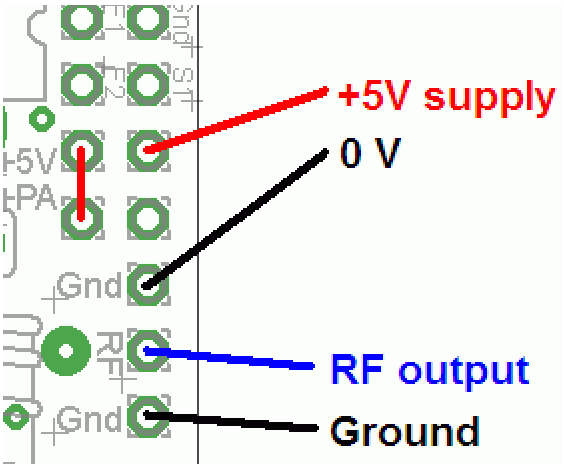
5.1 Explanation of connections



The diagram shows the connection pads along the edge of the board. The pin-spacing is 0.1-inches and a suitable connector could be used if desired. These are further described in the following sections. From left to right:

Label	Group	Description
5V	GPS	+5V supply to GPS module
Gnd	GPS	Ground connection to GPS module
RxD	GPS	Serial data input from GPS module
Pps	GPS	1 pulse per second input from GPS module
C in	Options	External oscillator frequency input
C out	Options	Digitial logic level RF output
AF	Options	Audio output when using the audio output option
S2	Buttons	Optional external button S2, switch to Gnd
Gnd	Buttons	Ground for optional external buttons
S1	Buttons	Optional external button S1, switch to Gnd
F0	Future	GPS Lock – signal high indicates lock
F1	Future	No connection – for future enhancements
F2	Future	Wait – signal is high while transmitting a frame
+5V	Power	+5V power supply
+PA	Power	PA power supply – may be connected to +5V
Gnd	Power	Ground connection to Power supply
RF	RF	RF output
Gnd	RF	Ground for RF output

5.2 Connection for basic operation

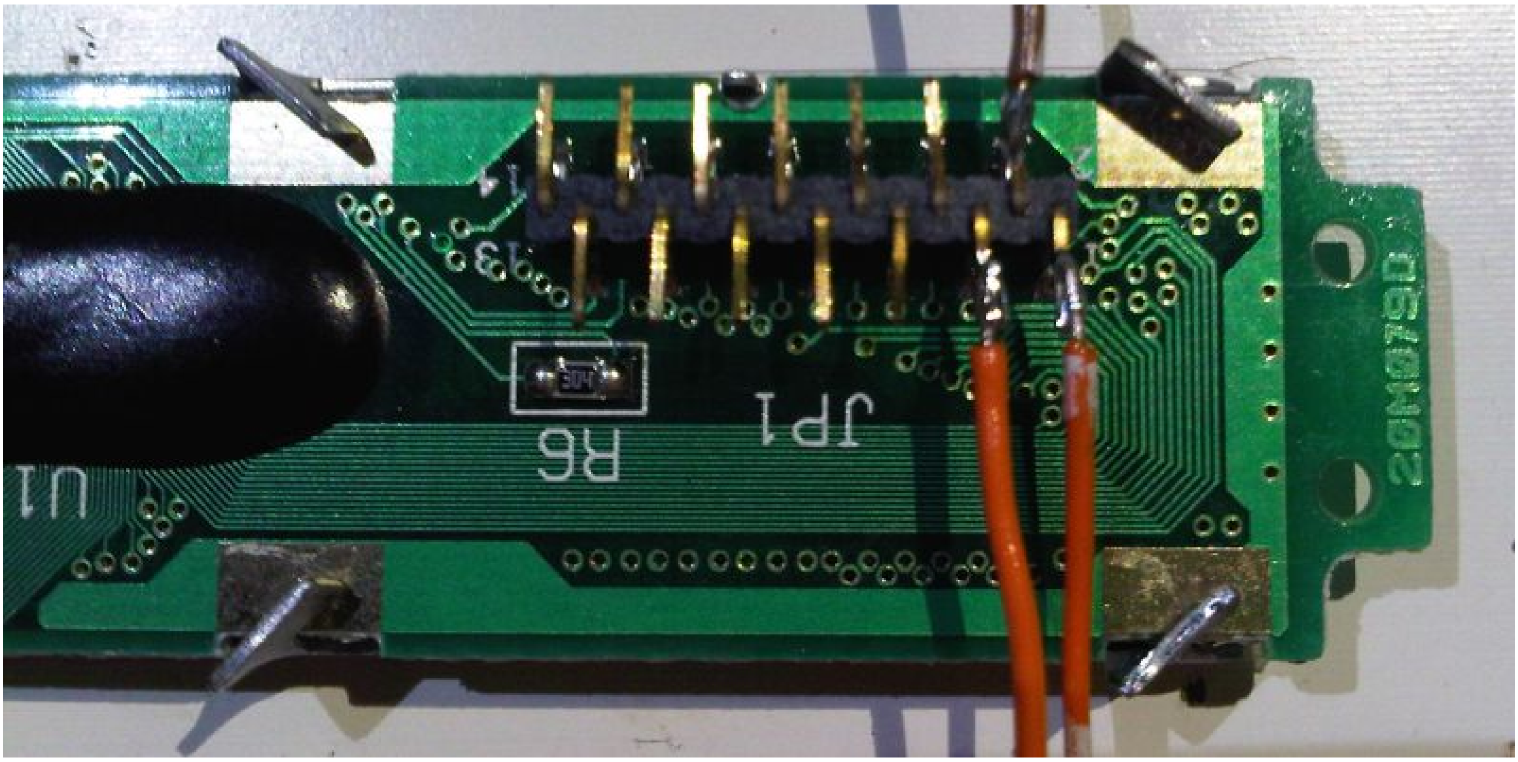


For the most basic operation of this kit as QRSS transmitter using 5V supply, it needs only a power supply and RF output connection.

To allow use of a higher voltage supply for the PA, the PA voltage is not connected to +5V on the PCB. So to run the transmitter using the 5V supply, be sure to connect a wire between the +5V and PA pins, as shown by the short red vertical line in the diagram.

5.3 Alternative LCD mounting options

An alternative method of mounting the LCD is possible. This method removes the requirement to solder the fine-pitch LCD pins. It is also convenient for those of you who wish to install the LCD in an enclosure separately from the board, for example to mount the LCD in a front panel. In this case, the pins of the LCD should be bent alternately in different directions. Short lengths of wire can be used to connect the LCD to the PCB, using the row of holes adjacent to the fine-pitched LCD holes. This row of holes are spaced at 0.1-inch pitch so are much easier to solder.



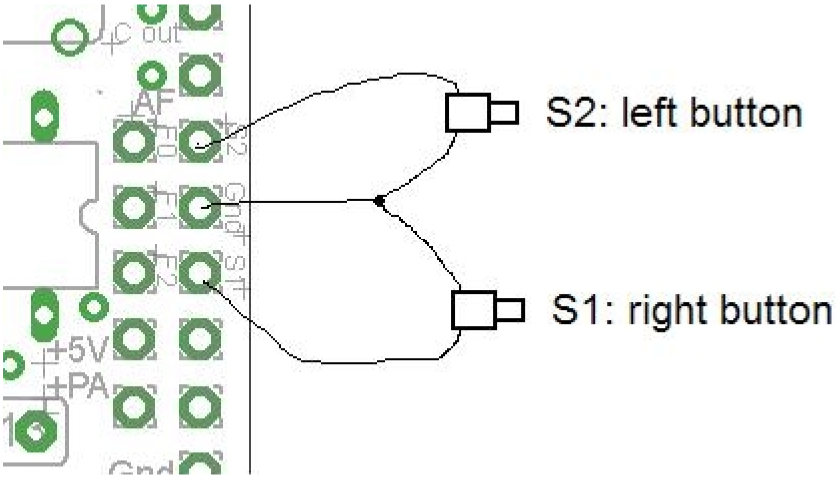
Another alternative LCD mounting option is to insert the LCD into the PCB in the normal way, but before soldering the pins, bend the pins gently 90-degrees, then solder. This method puts the LCD at right-angles to the PCB and is very nice for front-panel mounting; note, however, that the PCB will then be “upside-down” inside the box, when the LCD display is the right way up.

Note 1: this kit is NOT compatible with the common Hitachi HD44780-based LCD display modules.

Note 2: the LCD is quite DELICATE! If you use bolts fix the LCD to a front panel, ensure suitable spacers or washers are used so that stress is not put on the LCD, or it may crack.

5.4 Alternative button mounting option

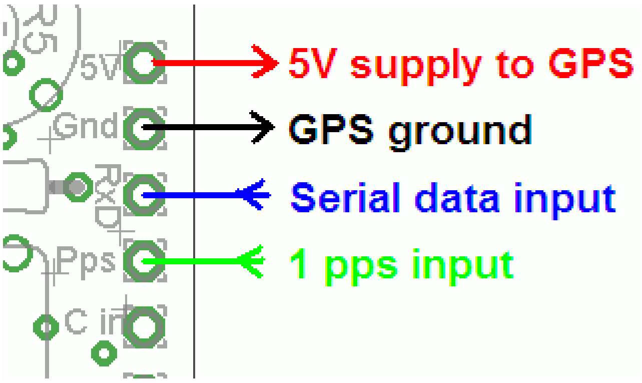
If you wish you may use an external pair of buttons to control the kit. The button input signals are available at the edge of the board, labelled S1 (right button) and S2 (left button). The buttons should be of the push-to-make variety. The right button should be connected between the S1 signal and ground; similarly the left button should be connected between the S2 signal and ground.



5.5 Optional connection of GPS module

A GPS module may be connected to the kit, to provide frequency stability, and latitude and longitude which can be converted to Maidenhead locator format for encoding in the WSPR message transmission.

Check that your module is powered from 5V. Many modules specify a 3.3V supply – in this case you will need to provide an external 3.3V voltage regulator. Where a 3.3V GPS module is used, the serial data and 1 pulse-per-second (pps) inputs are not a problem for the 5V microcontroller on the kit PCB – no voltage level conversion is required.



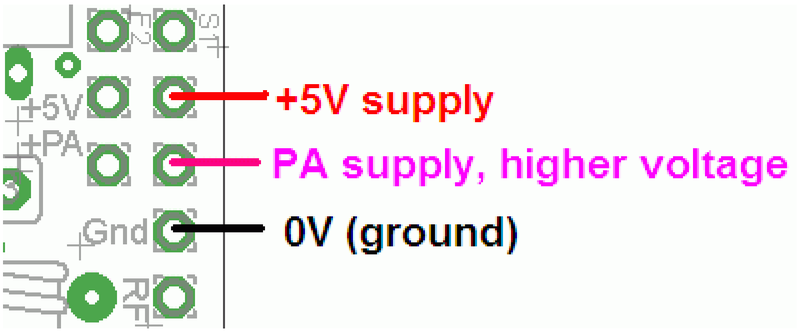
In cases where the location information is not required (modes other than WSPR) or you wish to enter it manually, the serial data input can be left unconnected. The 1 pps input is enough for the frequency lock function. Note that the frequency locking function can only operate in slow-speed modes, or wait periods between messages. GPS lock of the output frequency is indicated by a high signal on connector pin F0.

See operation instructions below for details on how to configure the GPS interface.

5.6 Higher power output

The kit provides around 180mW of output power using a 5V supply and a single 2N7000 transistor. The transistor gets slightly warm to the touch. If higher output powers are desired, this can be achieved by using a higher supply voltage for the Power Amplifier (PA). The microcontroller must still be run at a voltage not exceeding its 5.5V rating (5V is recommended). To facilitate running the PA at higher voltages to provide more power, the PA supply connection is a separate pin.

At higher supply voltages and power outputs, the heat dissipation in the 2N7000 will increase and its temperature will be higher. Provision is made on the PCB for a second transistor in parallel with the first. This will slightly increase the power output even at 5V supply, but more importantly at higher voltages it should share the heat dissipation between the two devices.



It could also be worth experimenting with other transistors such as the BS170 which is similar to the 2N7000 but has a higher power dissipation rating. It may be necessary to fix a heat sink to the transistors if the temperature rise is excessive. Experiment is needed in this area, with PA voltages up to perhaps 12V.

5.7 External Power Amplifier

The connections provide a pad labelled “C out” which is wired to the gate of the 2N7000 FET. This can be used to drive an external power amplifier if desired. The signal at “C out” is a 5V digital squarewave. For example, it could drive the gate of a more powerful FET for example an IRF510.

5.8 Audio frequency output

There may be many reasons why you want to operate on a frequency other than that of the quartz crystal supplied with the kit. For WSPR modes, 30m operation using the 10.140MHz crystal is fine as the WSPR sub-band is just above the QRSS sub-band. On other bands where the WSPR frequency is not near the QRSS frequency, the crystal won't be suitable. Or, maybe you want to transmit on a Hellschreiber frequency.

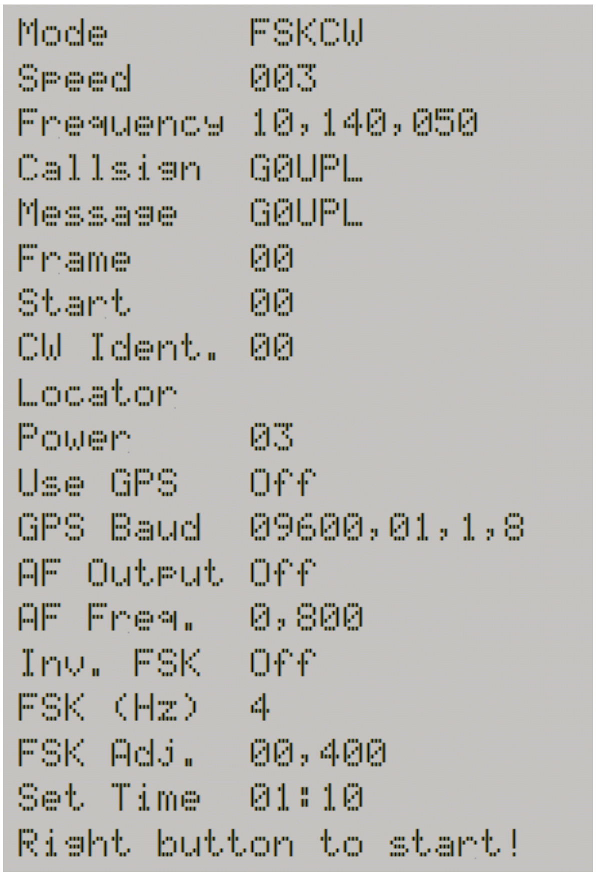
In all these cases (and more), you can use the Audio Frequency output “AF” to feed into an SSB transmitter. It is very important to note that this output is just a 5V peak-to-peak square wave – to connect it to a microphone input (for example) you will need to reduce the signal amplitude considerably. In testing, I used a potential divider made of an 18K resistor and a 10-ohm resistor to ground. The mic input was connected across the 10-ohm resistor and provided a suitable signal amplitude.

Another possible way to use the AF output is to rectify it (diode + capacitor) and use that as a plain on/off keying signal to a CW transmitter. This would be suitable for CW and Hellschreiber modes, but not modes where a frequency shift encoding is involved.

6. Operating instructions

The microcontroller has a 512-byte Electically Erasable Programmable Read Only Memory (EEPROM). This is used for storing your individual settings such as mode, keyer speed, and message. When the kit is first switched on, it automatically enters the “run” mode, in which the message is being transmitted in the selected mode. This is so that in the event of power interruption, the kit recommences the message transmission as soon as power is restored, without user intervention.

When you have just built the kit, the memory will contain only certain default settings, and an empty message. An error will be displayed, saying “Error: Message”, which indicates that the message is empty:



```
Mode       FSKCW
Speed      003
Frequency  10,140,050
Callsign   G0UPL
Message    G0UPL
Frame      00
Start      00
CW Ident.  00
Locator
Power      03
Use GPS    Off
GPS Baud   09600,01,1,8
AF Output  Off
AF Freq.   0,800
Inv. FSK   Off
FSK (Hz)   4
FSK Adj.   00,400
Set Time   01:10
Right button to start!
```

It is now necessary for you to configure the message and settings for your use.

6.1 Menu system

The user interface consists of the 24-character LCD and two push-buttons. There are 18 user-configurable settings available. The menu system allows all of the settings to be edited. The use of only two buttons to edit so many settings necessarily means that you must scroll through various options by many button presses. Clearly, there is no full computer keyboard in the kit with which to edit your alphanumeric message. However, the use of these two buttons to control the kit is fairly intuitive.

The menu is a list of items, and the LEFT button is used to scroll through these items. When you see the item which you want to edit, press the RIGHT button to start editing it. The list of menu items shown has example settings for sending the callsign “G0UPL” in FSKCW with 3-second dit symbols.

After cycling through the list of menu items using the LEFT button, the system is returned to the “Run” mode, transmitting the configured message.

Note the final item “Right button to start!”. When this final item is reached, you must press the RIGHT button, which then resumes the “Run” mode. The reason for this is that if, for example, you only wanted to change the mode, which is the first setting. It would be tiresome to have to scroll carefully though the remaining items until the “Run” mode was resumed. There are many configurations and you could easily miss the run mode, re-entering the menu system again at the first item. With the right button start feature, you can simply keep the LEFT button pressed, relying on the key autorepeat to quickly cycle through all the remaining menu items and stop at the end. It is also useful for when you want to start the system at an exact time.

6.2 Editing a menu item

There are three types of menu item: alphanumeric (e.g. callsign), numeric (e.g. frequency), and list (e.g. mode). Editing a configuration is slightly different depending on the type. When scrolling through the list of menu items using the LEFT button, you enter editing mode by pressing the RIGHT button.

6.2.1 List editing

If the menu item is a list menu, such as the Mode menu, use the RIGHT button to cycle through the list to find the setting you want. An On/Off menu such as Use GPS is also a list, having only two settings (On and Off). When you have chosen the setting you want, press the LEFT button. This saves the setting into memory and returns to the menu list.

6.2.2 Number editing

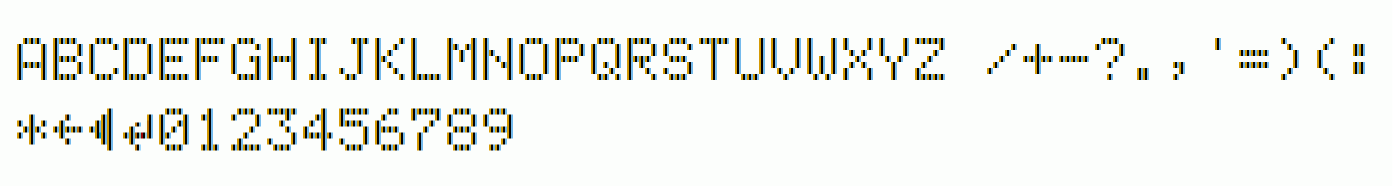
When editing a numeric setting, the RIGHT button is used to cycle through the numbers 0..9 until you find the number you want. The LEFT button moves rightwards one position to the next character to be edited. The current character being edited flashes. When the final (rightmost) digit has been chosen, a LEFT button press saves the setting into memory and returns you to the menu list.

Note that in some cases, there are restrictions on the value which may be chosen. For example, when editing the 10's of minutes digit, the RIGHT button only cycles 0..5 then returns to 0.

6.2.3 Alphanumeric editing

The most complex editing is alphanumeric, which also includes certain punctuation characters supported by the Hellshreiber character set. The principle is the same as for Number editing: the LEFT button moves the cursor one character to the right, and the RIGHT button cycles through the letters, punctuation and digits. However facilities are also provided to delete one character, or the whole message.

The order of the letters, punctuation and numbers is as follows:



ABCDEFGHIJKLMNOPQRSTUVWXYZ /+-?.,'=)(:
*⇠◀↲0123456789

The following characters have special function:

⇠ **Backspace (delete):** If you select this character as the current flashing character using the RIGHT button, then when you press the LEFT button, the current character is deleted and the flashing cursor moves back left one position.

◀ **Delete all:** If selected as the current character, pressing the LEFT button has the effect of deleting the entire message, starting again at the left of the screen. There is no “undo”, so use with caution!

↵ **Enter (finished):** If selected as the current flashing character, pressing the LEFT button is used to finish editing the setting. The setting is saved, and you return to the main menu list. Note that the text that is saved is only the text to the left of the Enter symbol. If you select this symbol and press the LEFT button when you are not at the furthest right position of the message, then everything right of your position is deleted.

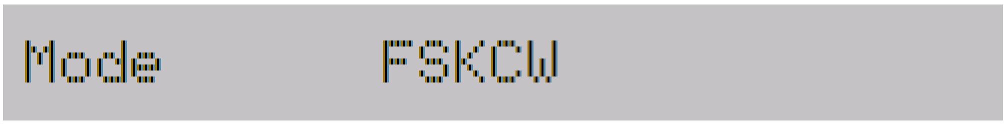
6.2.4 Configuration validation

In some cases, the configuration setting is validated at the time of entry. For example, the system does not allow the entry of an invalid time setting, or entering letters where numbers are required.

Additionally, there are many cases where particular combinations of settings produce an invalid situation for the transmission mode being used. In these cases, the configuration validation is done when the "Run" mode is entered, i.e. transmission started. If there is any reason why the configuration is invalid, an appropriate error message will be displayed, prompting you to return to the configuration setting to correct it.

6.3 Menu configuration settings

There are 18 user-configurable settings available in this kit. Not all of them are going to be relevant to the mode you wish to use, or to your hardware configuration. The following sections explain each setting in detail.

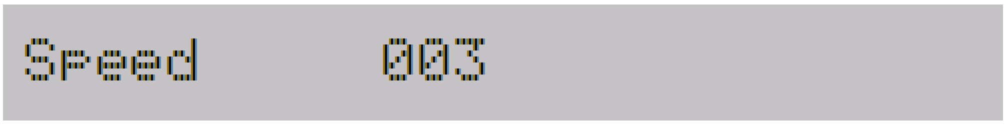


```
Mode        FSKCW
```

The Mode setting selects the transmission mode. The available modes are:
FSKCW, QRSS, DFCW, Slow Hell, Hell, DX Hell, CW, WSPR, TEST

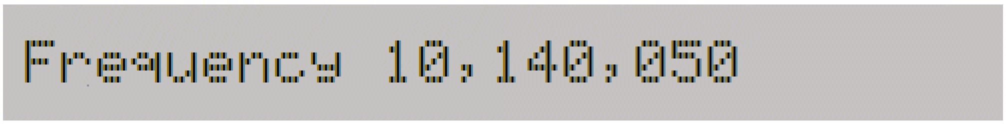
The TEST mode produces a simple squarewave pattern in FSKCW at the chosen dit speed, and with the chosen size of FSK (see below). If you monitor the transmitter output with a receiver and Argo, you can use the TEST mode to check the set-up of your kit – e.g. to check that the FSK size is correct.

Additionally, when a GPS module is connected, and you are in the RF mode (not audio output mode), during TEST mode the output frequency will be measured and displayed on the LCD. Effectively the system is then functioning as a frequency counter, using the GPS 1pps signal as an accurate timebase. This is very useful for setting the frequency using the C8 trimmer capacitor.



```
Speed       003
```

The Speed setting selects the dit speed, in seconds, for the following modes: FSKCW, QRSS, DFCW, TEST. In CW mode, the Speed setting means Words Per Minute. In Slow-Hell mode, the setting is the number of seconds duration of one character (see later section for explanation). The setting has no effect at all, in Hell, DX Hell or WSPR modes, these modes all have a defined timing which cannot be changed.

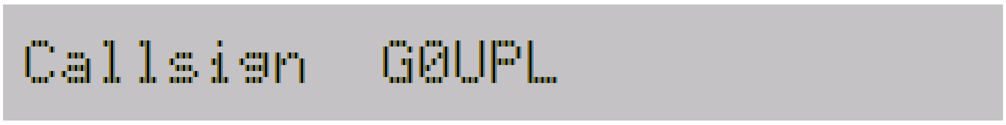


Frequency 10,140,050

This frequency sets the baseline RF output frequency. You must ensure that the frequency selected is the same as the actual operating frequency of your crystal (not just the value printed on the crystal), for the band you have chosen. There are two reasons for this:

1) If you are using a GPS module to lock and stabilise your frequency, the frequency setting in this configuration parameter tells the microcontroller what your target frequency is, i.e. the frequency which it will try to lock on. It is also important that the C8 trimmer capacitor is set so that this set frequency is very close to the measured frequency, because the adjustment control of the microcontroller over the crystal oscillator frequency is quite small.

2) The timing for the message encoding in all modes is derived from the microcontroller system clock, which is the same oscillator as the RF output frequency. The microcontroller therefore has to know what its system clock frequency is, so that it knows how to divide it down for timing purposes.



Callsign G0UPL

The callsign setting is an alphanumeric setting, which may be up to 14 characters long. It can theoretically include punctuation but clearly that isn't normally used in a callsign. The callsign is NOT used for the normal message transmission in most modes. It is used only for TWO purposes:

1) The kit may be configured to automatically send your callsign in standard 12wpm CW at pre-programmed intervals, as a station identifier (read your license conditions)

2) The Callsign is used for generating the WSPR message encoding

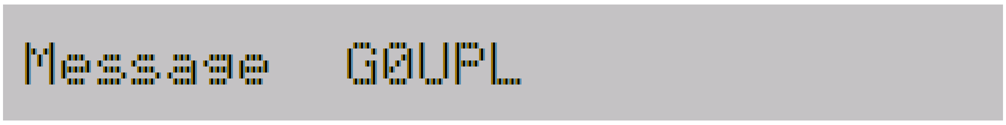
Note that when using the WSPR mode, the callsign must satisfy certain rules. Firstly, it can only be a 5 or 6 characters long. The callsign must consist of the following:

1) One or two characters consisting of A-Z or 0-9

2) One character which must be a number, 0-9

3) Three characters which must be A-Z or space

These restrictions are placed on the callsign in order to satisfy the requirements of the WSPR encoding algorithm. If you are using the WSPR mode, and the callsign violates the above requirements, the program will try to use the contents of your Message setting, to see if that contains a callsign. If the program cannot obtain a valid callsign from either location, then an error will result and be displayed on the LCD (Error: Message).



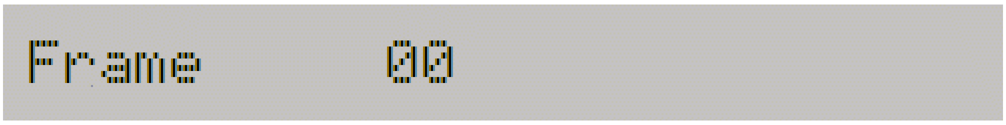
Message G0UPL

The message setting is an alphanumeric parameter which may be up to 390 characters long. For FSKCW, DFCW, CW, QRSS modes, it must be in the range A – Z, 0 – 9, or a space character. Hell modes can use punctuation characters as well, but not the * character.

Any message may be entered, but if the message is invalid for the mode being used, an error will be displayed.

The * character has a special purpose, it is used to signify the start and end of user-customisable pattern generation specification.

IMPORTANT: remember to add a space character at the beginning or end of your message! G0UPLG0UPLG0UPL... etc. is not easy to read, because it has no spaces. G0UPL G0UPL G0UPL is how you want your message sent, with a space between the messages.



Frame 00

This parameter is used to define the message frame length. A technique called “stacking” has been developed by QRSS'ers to improve signal-to-noise ratio even further. The transmitting station transmits messages defined into a frame of a precise length. The receiving station overlays multiple message reception frames on top of each other on the display. Any message which is repeating at that precise frame length gets re-enforced by the averaging of multiple message receptions; any other non-repeating message or QRM gets reduced.

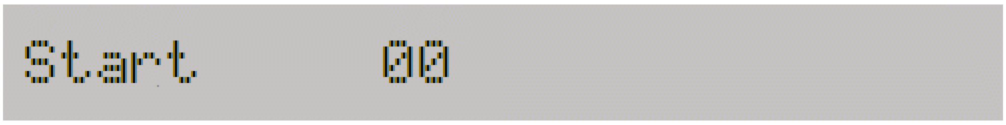
As an example, consider a message which in the chosen QRSS mode and speed happens to take 8 minutes to transmit. You could set the “Frame” setting to 10. This means that after the message transmission has completed, the microcontroller will wait until 10 minutes after the message start, before starting the next transmission. Receiving stations who have are using QRSS decoding software with the stacking feature, and have it set to 10 minutes frame length, will be able to collect six (say) transmissions over one hour and average them together to improve the signal-to-noise ratio.

This parameter is also used for the WSPR mode. In the WSPR mode, the Frame parameter defines how often the WSPR transmission packet is transmitted. WSPR transmissions take just under 2 minutes and are always aligned to the start of an even minute. Therefore in WSPR mode, the Frame parameter must be an even number (a multiple of 2), e.g. 10 minutes (9 minutes is not possible).

Typically, Frame should be set to 10 minutes for WSPR. 10 minutes is also a popular frame length for “stacking” QRSS reception, as long as your message transmission fits within 10 minutes. If it does not, the controller will wait until the next 10 minutes have elapsed.

NOTE: While transmitting a frame, a high signal is output on connector pin F2.

Default: the default value is 00 which means no frame size is defined, and the stacking feature is not used. It must be set, for WSPR.

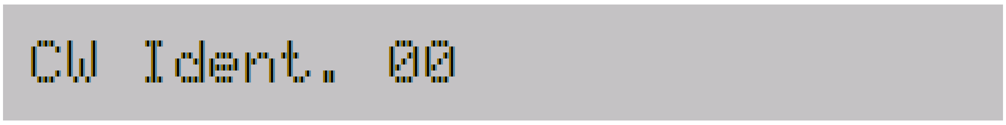


Start 00

This parameter is used in conjunction with the Frame parameter described previously. It defines when the transmission cycle starts. For example if Start is set to 03, and Frame is set to 10, then the message transmission will start on the when time reaches 00:03 (or 3 minutes past any hour), and will repeat at exactly 10 minute intervals.

Again for WSPR, the Start parameter must be even (a multiple of two).

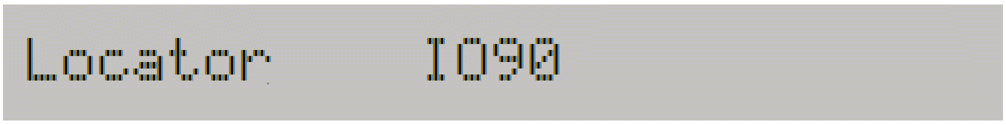
Default: the default value is 00 which means the Start parameter is not used.



CW Ident. 00

If the CW identifier feature is used, then this parameter should be set to a non-zero value indicating the number of minutes interval for sending the CW identifier. The CW identifier sends the callsign in standard 12wpm CW at regular intervals. If the CW Ident parameter is set, then a callsign must be entered in the callsign setting, otherwise an error will be displayed.

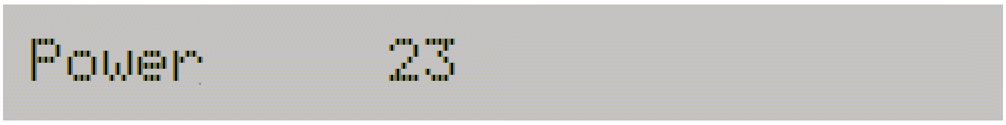
Default: the default value is 00 which means that a CW Identifier is not transmitted.



Locator   IO90

The locator setting is only used in WSPR mode, and is one of the inputs to the WSPR encoding algorithm. It is four characters, the first two are always A – R and the second two are numbers in the range 0 – 9. Editing this parameter automatically prevents incorrect entries. The locator is not needed for any other mode than WSPR and may be left blank.

If a GPS module is connected with a correctly configured serial data stream, then the locator is set automatically from the GPS data.



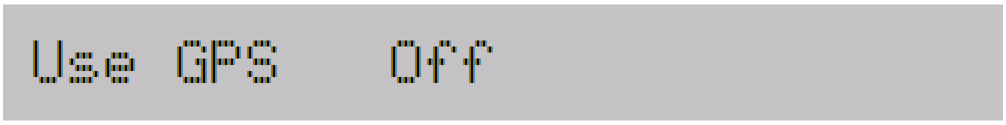
Power   23

The power setting is another input to the WSPR message. This setting is not needed for any other mode than WSPR. Although the parameter will accept any 2-digit value, only certain values are valid for the WSPR encoding scheme. The power is the transmitter output in dBm. It must be in the range 0 – 60, and the least significant digit must be 0, 3 or 7. The allowable values, with their corresponding power in watts, are listed here:

00 dBm	=	1mW
03 dBm	=	2mW
07 dBm	=	5mW
10 dBm	=	10mW
13 dBm	=	20mW
17 dBm	=	50mW
20 dBm	=	100mW
23 dBm	=	200mW
27 dBm	=	500mW
30 dBm	=	1W
33 dBm	=	2W
37 dBm	=	5W
40 dBm	=	10W
43 dBm	=	20W
47 dBm	=	50W
50 dBm	=	100W
53 dBm	=	200W
57 dBm	=	500W
60 dBm	=	1000W

The default value in this kit is 23dBm (200mW) which is closest to the measured prototype output power of 180mW.

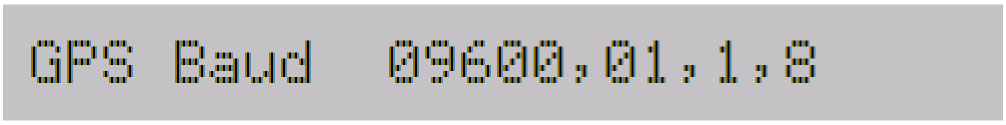
An illegal value will result in a runtime error display.



```
Use GPS   Off
```

This controls whether GPS is used or not. If set to On, and a GPS module is connected with 1pps signal, then the accurate time pulses from the GPS unit will be used to lock the frequency, and to keep precise time.

If additionally the serial data connection from the GPS unit is valid, the Maidenhead locator will be computed from the latitude and longitude received from the GPS module, and the internal realtime clock will be set using the time from the GPS module.



```
GPS Baud  09600,01,1,8
```

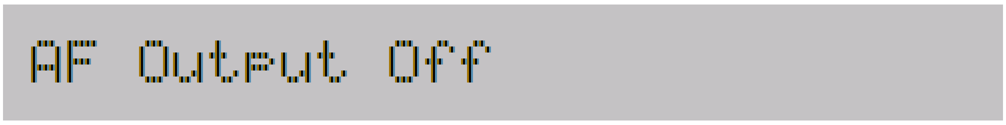
These settings determine the communication parameters for receiving a data input from the GPS module. They must be set to match the data output from the GPS module otherwise no data can be decoded. You will need to refer to your GPS module documentation to determine the correct speed and other settings.

There are four settings in the configuration string:

1) Baud rate: any baud rate can be entered here. 9,600 is common
2) Parity: the first digit is 1 to enable parity checking, 0 to disable it. The second digit is 0 for even parity, 1 for odd parity. Therefore the 01 setting shown above means disabled parity, odd parity.
3) Number of stop bits: this must be 1 or 2. In the example shown above, it is 1.
4) Character size, which technically may be in the range 5 – 9 though anything other than 8 is very unlikely where GPS is concerned.

The default setting shown above, is for 9,600 baud no parity, 1 stop bit and 8-bit data. This is a common setting.

Note that timing for the serial interface is derived from the system clock, which is the crystal oscillator, using a divisor calculated from the known frequency of the oscillator. The frequency is “known” via the Frequency parameter setting described above. If the crystal frequency is not reasonably closely matched by the set frequency parameter, then the timing for the serial interface will be inaccurate and data may not be decoded successfully.



```
AF Output Off
```

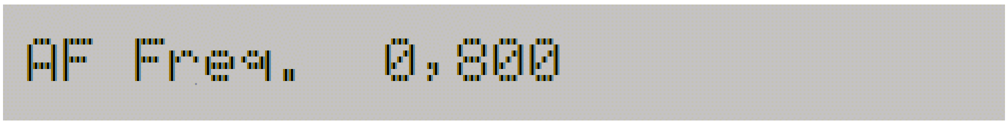
If the AF Output option is used, then the hardware output pin “AF” can be used as an audio signal to transmit using an SSB transmitter. If this is the case, then the kit can be used purely as a keyer, rather than as a complete transmitter. This can be useful if you want higher output power using your transceiver, or if you want to operate on a frequency for which you do not have a crystal. Please see the notes in the hardware options section.

In this mode, the microcontroller does not try to adjust the crystal frequency to produce FSK etc., nor does it try to lock the crystal frequency if a GPS module is present. Instead, it outputs an audio tone which is on/off keyed and shifted appropriate to the mode and message being transmitted.

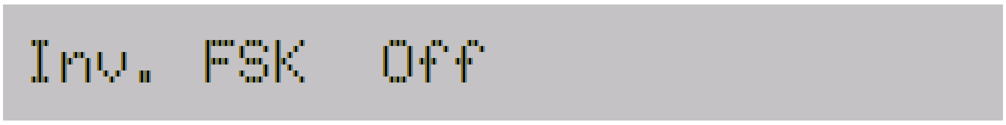
In this case, the actual frequency of the crystal in the kit does not matter, as long as the frequency parameter setting matches the crystal frequency. However for best results, at least 10MHz is recommended because higher oscillator frequencies improve the resolution of the frequency steps obtained at audio.

In this mode there is no RF output.

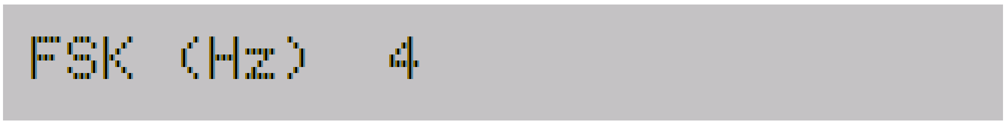
Note that even when AF Output is set to off, there will still be a signal at the “AF” connector – this is because it is connected to the microcontroller's Pulse Width Modulation output, which is integrated and used to adjust the oscillator frequency.


AF Freq. 0,800

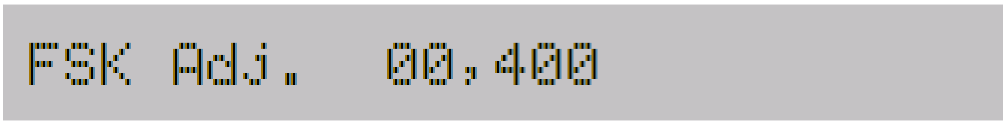
This parameter sets the audio frequency, when the AF Output mode is used. The value must be at least 400Hz.


Inv. FSK Off

This setting allows you to invert the direction of the frequency shift. It isn't normally used and should be left Off ordinarily.

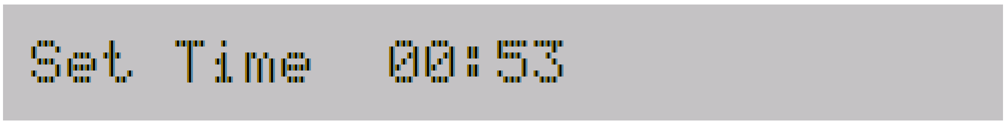

FSK (Hz) 4

FSK size is determined by this parameter. For FSKCW and DFCW modes, the specified FSK is the height of the symbol shift. For Slow-Hell mode, it is the height of the text. The parameter does not apply for the other modes.


FSK Adj. 00,400

The microcontroller adjusts the oscillator frequency by means of a 16-bit Pulse Width Modulation signal which is integrated by R2/C2 and applied to a reverse-biased diode, whose varied capacitance changes the frequency. This parameter is the amount of change in the 16-bit PWM value which is required to cause a 1Hz shift in frequency.

It is crucial to correctly set this parameter. Without it, the FSK height will be incorrect, and probably WSPR modes will not work at all. Please refer to the tuning and calibration section below, for details of how to adjust the kit to get this parameter correct.



Set Time 00:53

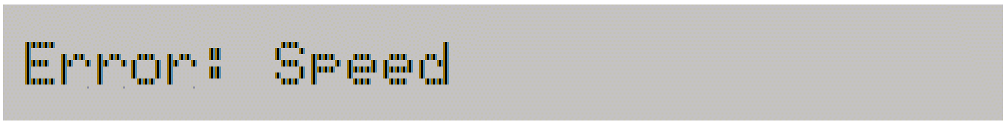
The kit has an internal real time clock which is used for the Frame (QRSS stacking), CW Ident, and for timing WSPR messages. This configuration allows you to set the real time clock. The clock setting takes effect, and the seconds (internally counted but not displayed) are set to zero, when the LEFT button is pressed after setting the minutes.

Note that if using a GPS module with “Use GPS” switched on, and if there is a valid serial data stream properly decoded (by correct choice of “GPS Baud” parameters), then there is no need to set the time, because it is automatically decoded from the GPS data.

An accurate real time clock is not required other than for the WSPR mode. Even if the real time clock is not set, it will start from 00:00 when the system is powered up. This is sufficient for the production of CW identification (if enabled) and stacking if enabled (i.e. if Frame is non-zero).

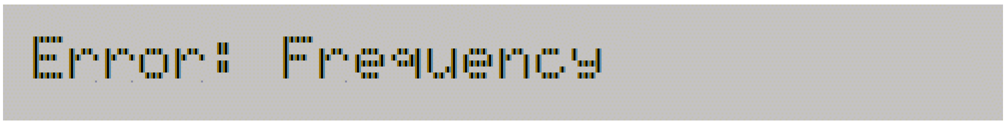
6.4 Understanding error messages, and how to fix them

In the event of inconsistent parameter setting, an error message may be displayed on the LCD. The cause of these messages is well-defined. The following lists the possible error messages and their causes.



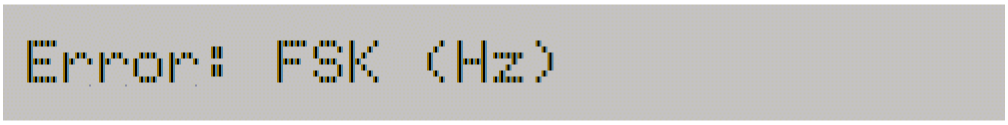
Error: Speed

This error results from having a zero Speed parameter, in any mode other than Hell, DX Hell and WSPR. These three modes have specific defined speeds which are not adjustable, and so the Speed setting is ignored for them.



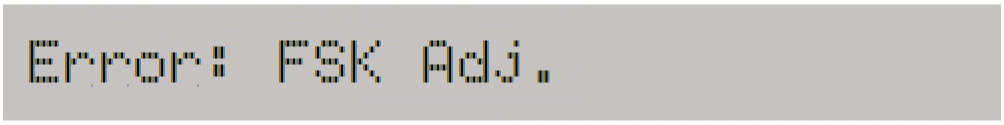
Error: Frequency

The Frequency parameter is set to a value less than 450,000. You should enter a valid value, the parameter setting must match the crystal frequency being used, and if the RF Output is being used (AF Output is Off), the frequency setting must also be set to your desired baseline output frequency.



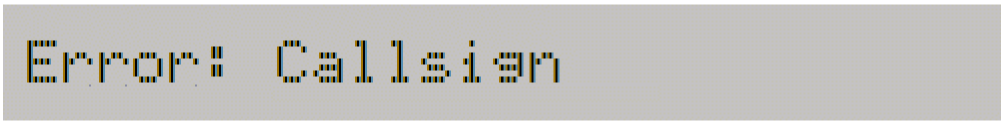
Error: FSK (Hz)

In modes FSKCW, DFCW, Slow Hell and TEST, the FSK size must be set. This error message indicates zero FSK (Hz) parameter setting.



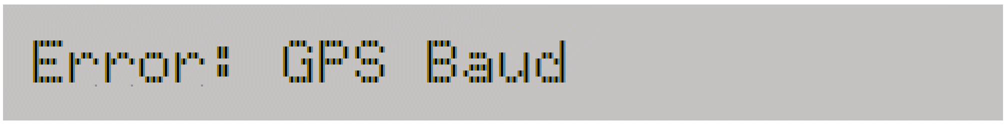
Error: FSK Adj.

The FSK Adj. parameter is zero. This parameter must be non-zero in all modes.



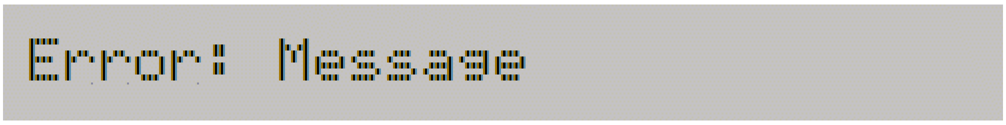
Error: Callsign

This error arises when the CW Ident parameter is non-zero, which means you have requested a CW Identifier transmission in plain 12wpm CW at a periodic interval, but the Callsign has not been set. The resolution to this error is to enter your callsign in the Callsign parameter.



Error: GPS Baud

This error can only occur if the “Use GPS” configuration is On and if the configured Baud rate is less than 1,000. The solution is to correctly configure the Baud rate setting.



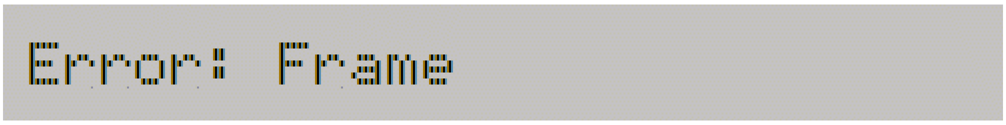
Error: Message

This error concerning the transmission message can occur for three reasons:

1) The message is empty, and the mode is set to anything except WSPR. In WSPR mode, the message encoding is generated automatically from the callsign, locator and power settings. In all other cases, the message must be set, otherwise the kit has nothing to transmit.

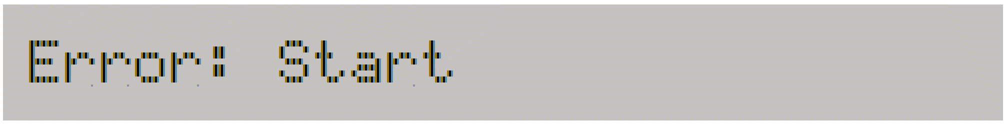
2) In WSPR mode, if the Callsign is empty or invalid, then the WSPR encoding algorithm will try to use the main message. If the Message parameter also does not contain a valid callsign for WSPR, then this error display will occur. To resolve this: study the requirements for a valid callsign in the configuration settings section above, and ensure that the Callsign parameter is set with a valid WSPR callsign.

3) In modes QRSS, CW, FSKCW, DFCW, there is an invalid character in the message. Valid characters are A – Z, 0 – 9, space and /. in the case of FSKCW the message may also contain an asterisk * which enables the user-customised pattern mode.



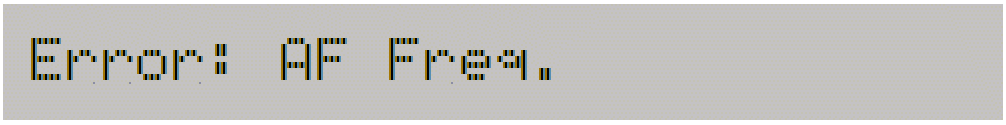
Error: Frame

In WSPR mode, Frame (number of minutes between transmissions) must be even (a multiple of two), and it must be non-zero. 10 [minutes] is the recommended value.

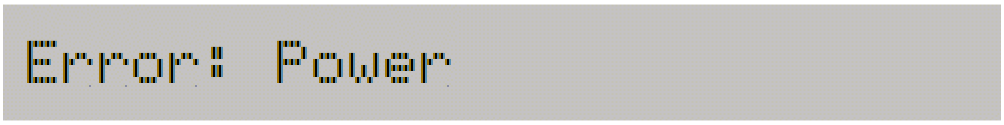


Error: Start

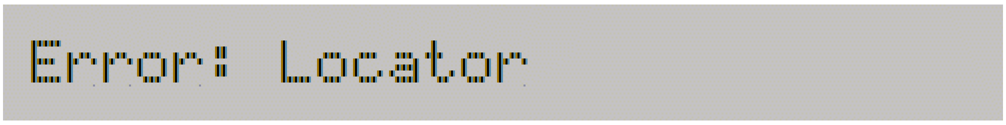
In WSPR mode, the Start parameter must be an even minute (power of two). In all modes, the Start parameter must be less than 60 (as there are only 60 minutes in an hour).


Error: AF Freq.

When in Audio frequency output mode (AF Output On), the AF Freq. Setting must be at least 400Hz.


Error: Power

This error can only arise during WSPR mode. The Power setting must be in the range 00 – 60 and the last digit must end in 0, 3 or 7. i.e. 00, 03, 07. 10, 13, 17 are all Ok.

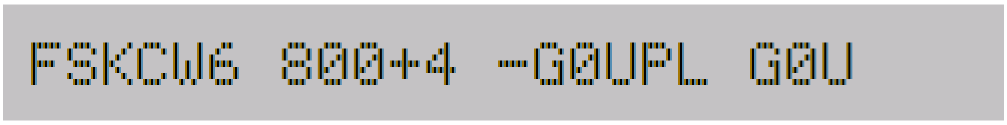

Error: Locator

Again, an error which can only arise in WSPR mode. The locator must be a standard 4-digit Maidenhead locator (e.g. IO90). If you are getting this error, it means you have not properly set the Locator configuration parameter – it should consist of two letters in the range A – R followed by two digits in the range 0 – 9.

7. QRSS Modes

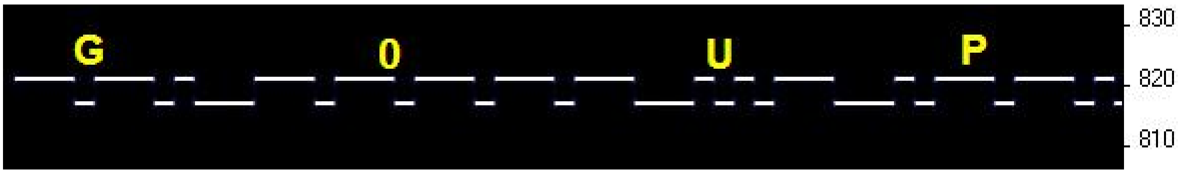
The following is a short description of each kind of QRSS mode, and what to expect.

7.1 FSK/CW


FSKCW6 800+4 -G0UPL G0U

The display will show FSKCW followed by the symbol length – in this example 6 seconds. Next comes the base frequency. In AF Output mode, that is Hz. In RF Output mode, it is the last three digits of the frequency – e.g. if the Frequency parameter is set to 10,140,050 then the display will show 050. The FSK shift in this case, from the “FSK (Hz)” setting, is 4Hz – indicated by the +4. Finally the message will show. The leftmost character is the one currently being sent, and the – before it indicates a dash is being sent. As the message is sent, the message will scroll to the left.

In the following image, the decoded letters have been drawn above the received signal in yellow, to show clearly how the mode is to be decoded.



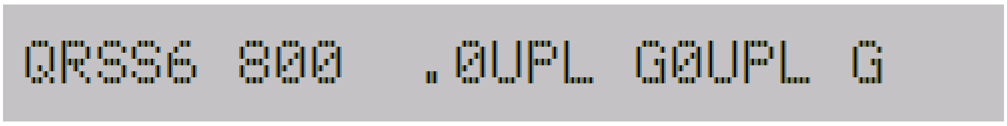
In Frequency Shift Keyed CW (FSK/CW) the dahs and dits have the same timing as traditional (slow) morse i.e. QRSS, but the carrier is always on and “key down” is indicated by a slight upwards shift of a few Hz. FSK/CW is the most popular of the slow-signal modes.

Good for: The transmitter being continuously on, avoids any “chirp” (frequency pulling) of the oscillator, that can be a problem, particularly in more simple circuits. The signal is very readable in the presence of

QRM or in weak conditions, because of the continuously present carrier. Decoding the signal is intuitive, just by reading the top line like ordinary CW.

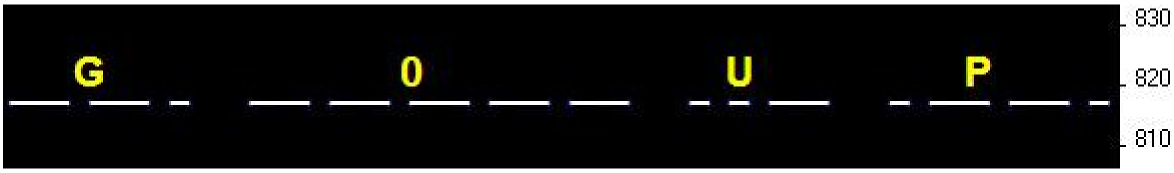
Bad for: Because the carrier is continuously on, it will take more battery power (if that's a concern), than an on/off keyed mode like plain CW. It also takes more bandwidth than the single sub-Hz bandwidth of plain QRSS.

7.2 QRSS



```
QRSS6 800  .0UPL G0UPL G
```

Here the display shows the scrolling message, with a – or a . (for dah and dit), to show which character and symbol is currently being transmitted. The dit speed follows the mode (QRSS), in this example 6 second dits. The frequency is as for FSK/CW (see above) – in this case 800Hz audio output.

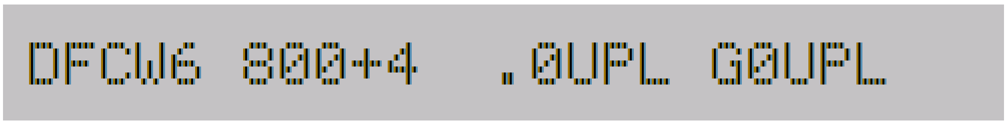


The original plain QRSS mode is simply plain CW, but massively slowed down, typically with dit speeds from 3 seconds or slower, even all the way to 120 seconds in some cases.

Good for: Simple to decode by eye, since it is the same was what we can imagine hearing. Very narrow bandwidth and band occupancy, sub-1Hz.

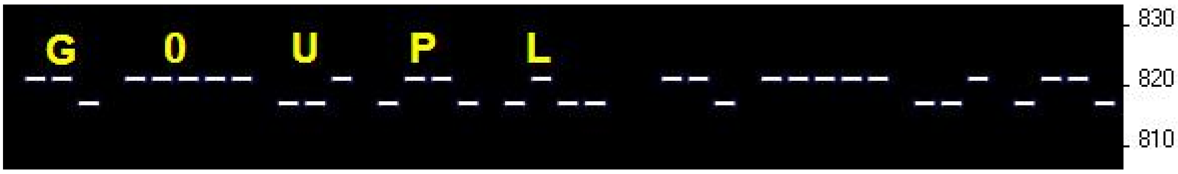
Bad for: In weak signal conditions or where there is a lot of QRM, an interfering weak carrier may also look like an intermittent horizontal line, so the wanted signal can be hard to distinguish from the QRM. Fading (QSB) can cause interfering carriers to look like QRSS too. In simpler circuits, “chirp” (oscillator pulling) can be a problem.

7.3 DFCW



```
DFCW6 800+4  .0UPL G0UPL
```

The display will show DFCW followed by the symbol length – in this example 6 seconds. This is followed by the base frequency (see FSK/CW section above) and the FSK size. The scrolling message is shown as usual, with the – or . to show the current symbol of the current character.

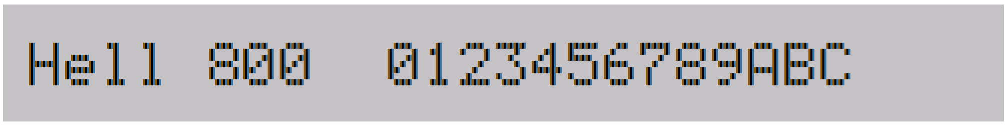


DFCW means dual-frequency CW. The dits and dahs are the same length, but dahs are shifted upwards by the FSK amount. A 1/3 dit-length gap inserted between symbols improves the readability.

Good for: The mode is faster than QRSS, with the same signal-to-noise ratio improvement, because the dah's are the same length as the dits. In QRSS dah's are three times longer. So the message transmission is faster.

Bad for: Not very intuitively easy to read as QRSS is. If the signal is a bit weak and hard to decode, the eye/brain may have more trouble "filling in the gaps", because it is not so intuitive.

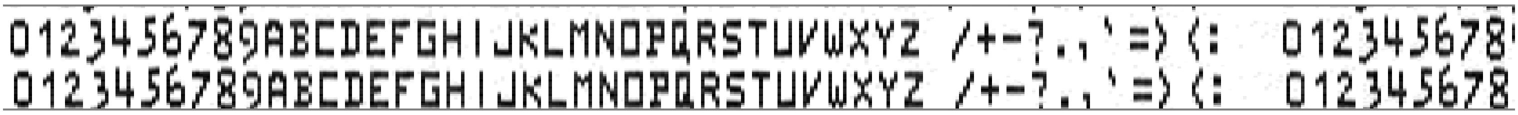
7.4 Hellschreiber (also called FeldHell).



Hell 800 0123456789ABC

As usual, the display shows the mode, frequency and the message scrolls to the left as it is sent.

The screenshot fragment below is from IZ8BLY's excellent Hellschreiber receiving (and sending) software.



0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ /+-?.,'=)(: 012345678

0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ /+-?.,'=)(: 012345678

Hellschreiber is a fax-like mode designed in the 1920's by Rudolf Hell in Germany. It was later used in WWII. The literal translation into English would be "Light Writer" and of course rhymes with its inventor's name. Each character is made up of a 7-row, 5-column grid. Technically there are 14 half-rows, but there are no lonely half-pixels in the characters, half-pixels are only ever seen in groups of at least two. This was an ingenious idea to minimise transmission bandwidth yet improve readability. The baud rate of a standard Hell transmission is 122.5. Most of the characters fit into a 10-half-row, 5-column grid, but the numbers are extended above and below this in some cases.

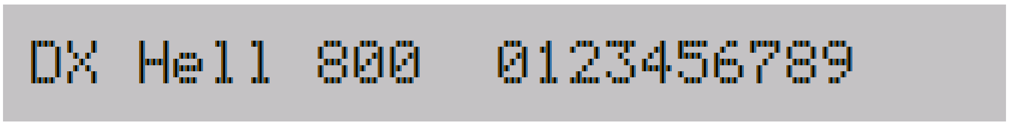
Good for: Fun, history, and easy readability.

Bad for: Bandwidth. Hell is not a slow-signal mode; the bandwidth is said to be around 400Hz.

We are proud to be using the original Hell font in this kit as carefully designed by Rudolf Hell. I know this because Pierre ON5SL, who owns a real antique FeldHell machine, kindly sent me a scan of one page of the operator's manual. This page shows the drum pattern of the machine.

Various software is freely available that can send and receive Hellschreiber, using any Windows font. Specially designed fonts are available. One called "FeldReal.fon" claims to use the original Hellschreiber character design but I am not convinced – if you are interested, look for example, at the K. FeldReal.fon's K actually looks nicer, but it doesn't quite match the one in the machine, look at the top right of the K.

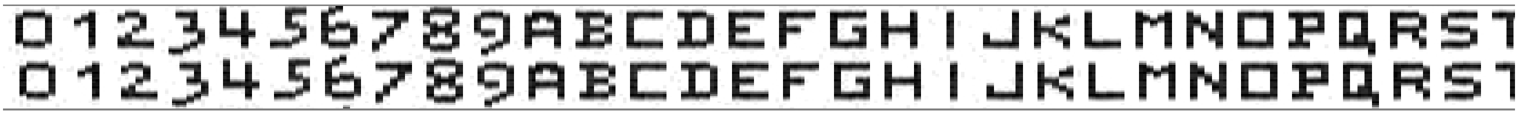
7.5 DX Hellshreiber



DX Hell 800 0123456789

As usual, the display shows the mode, frequency and the message scrolls to the left as it is sent.

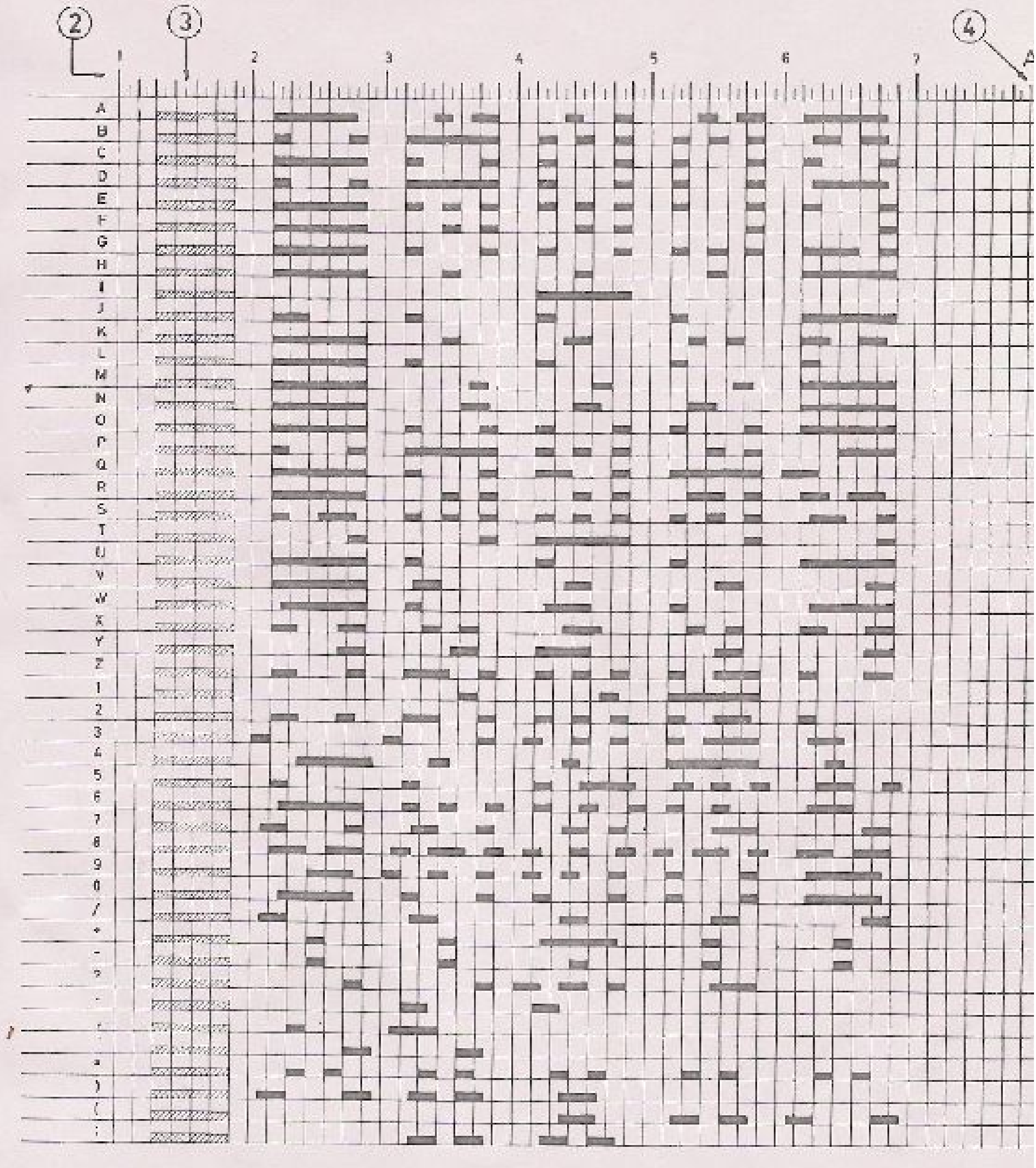
The screenshot fragment below is from IZ8BLY's excellent Hellschreiber receiving (and sending) software.



0123456789ABCDEFGHIJKLMNOPQRST

0123456789ABCDEFGHIJKLMNOPQRST

So-called DX Hellschreiber is the same as ordinary Hellschreiber described above, except that every column is sent twice. The characters are therefore twice as wide, the message takes twice as long to send, but it has the advantage of potentially being more readable in weak conditions.



① Abwicklung der Sendenocken

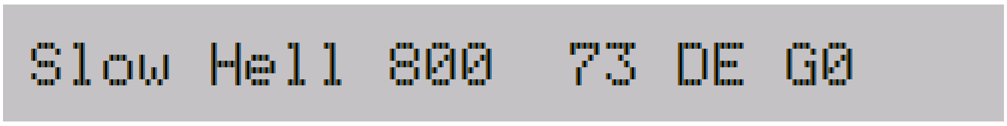
② Einteilung der Meßscheibe

③ Anlaufschritt

④ Nullstellung der Senderachse

Page from a real Hellschreiber machine owned by Pierre ON5SL, showing the character drum.

7.6 Slow Hellshreiber


Slow Hell 800 73 DE G0

As usual, the display shows the mode, frequency and the message scrolls to the left as it is sent.

Slow Hellschreiber is the slow-signal Hellschreiber equivalent. Each Hellschreiber letter row is shifted slightly in frequency. Typically the single pixel rate is very slow, for example 1 second per pixel (0.5 seconds per half-pixel). The character pattern is “scanned” one column at a time from bottom to top, shifting the frequency as the row increases up the character. In this way an image of the character is built up at the receiving station.

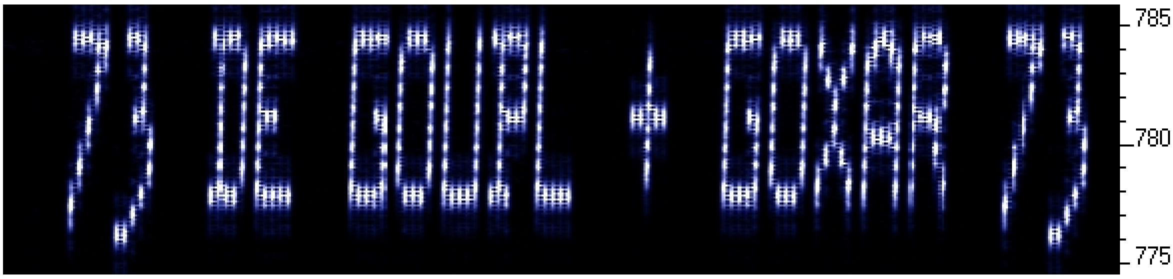
In Slow Hellshreiber mode, certain configuration settings in the kit require special attention:

Speed: The speed setting is the number of seconds required to transmit one whole character. Recall that a Hell character is composed of 7 rows (14 half-rows) and 5 columns; there is also one empty column as an inter-character space. Therefore each character is composed of 42 “pixels”. Therefore if you want the transmission time to be 1 second per pixel, set Speed to 042. If you wanted 1.5 seconds per pixel, you would set Speed to 063, and so on.

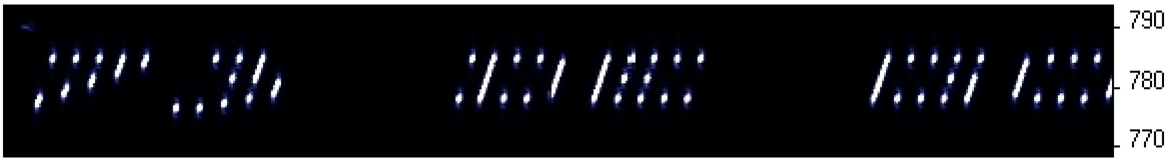
FSK (Hz): The FSK size configuration specifies the height of each character, it is the number of Hz for 5 rows of the character. Recall that most Hell characters, such as the letters, fit on a 7-column, 5-row grid. FSK (Hz) specifies the size of that grid; some of the numbers extend above or below the grid. For example an FSK (Hz) setting of 05 would result in characters 5Hz high, i.e. 1Hz per row (pixel).

Be aware of fundamental limitations of information theory. If the pixel rate is 1 second (Speed = 042), the minimum theoretical bandwidth is 1Hz. Therefore choosing a pixel size of less than 1Hz, i.e. FSK (Hz) = 05, does not make any sense. Faster transmissions require larger bandwidth, and you will have to set the FSK size higher to match. If you do not, the image at the receiving end will be blurred.

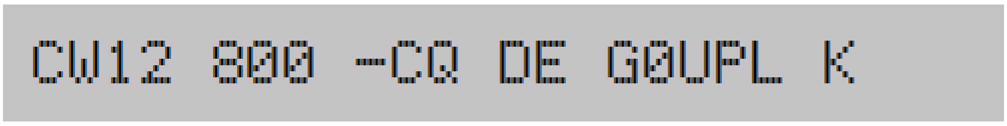
This example image was taken in Argo, in its 10 second “slow” mode. The message is “73 DE G0UPL + G0XAR”, the Speed is 042, and FSK (Hz) is 08. Note that even here, the characters start to look slightly blurred – a consequence of the fourier transform and minimum bandwidth, because the bucket size in Argo's 10 second mode is well under 1Hz so the 1Hz.



The following image shows part of the same thing in Argo's 3 second “slow” mode. The larger fourier transform bucket size means the image is more clear, but the faster scroll rate makes the letter's height to width ratio somewhat less attractive. Other slow-speed fourier transform software present more configuration options and may be more suited for slow-Hellshreiber reception than Argo.



7.7 CW



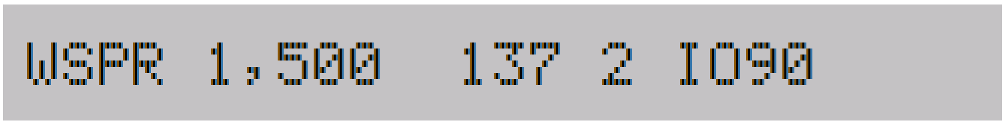
```
CW12 800 -CQ DE G0UPL K
```

CW mode is just plain, ordinary on/off keyed CW (morse code) at ordinary speeds. The display will show CW followed by the speed – in this example, 12 words-per-minute (wpm), followed by the message which scrolls to the left as it is sent. The character position right before the message shows – or . depending on whether a dah or dit is being sent.

In CW mode, the Speed setting is the words-per-minute speed, e.g. 012 for 12 wpm.

Have you ever spent a lazy Sunday afternoon calling CQ on 80m CW with QRP power, with no luck? Why not use the kit as a CQ machine, and take the hard work out of calling CQ! In AF output mode, the AF output could be rectified and used to key a CW transmitter, with the Frame setting used to control how often the CQ message would be transmitted.

7.8 WSPR



```
WSPR 1,500  137 2 IO90
```

The WSPR mode display is composed of several elements. Given the example display above:

WSPR The mode being used!
1,500 Baseline output frequency – in this case, 1,800Hz (Audio output mode)
137 Symbol number currently being sent; WSPR messages are made of 162 symbols
2 Tone currently being sent – WSPR messages are made of four tones, 0 – 3.
IO90 Maidenhead locator setting coded into the WSPR message

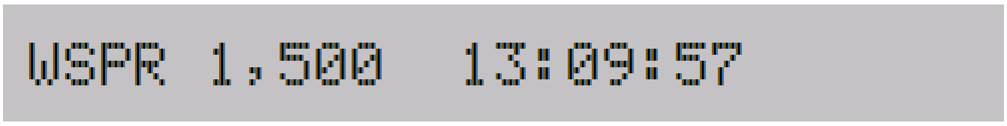
WSPR stands for Weak Signal Propagation Reporter. The message consists of three parts: the operator's callsign, Maidenhead locator (4-character, e.g. IO90) and two digits specifying the power. At the receiving station, messages are decoded and uploaded to a central internet database.

The WSPR message is encoded into a set of 162 symbols, each may be 0, 1, 2 or 3, using a compressed data format with forward error correction. The symbols are transmitted as tones, each tone separated by 12,000 / 8,192 Hz i.e. approximately 1.46Hz. The duration of each symbol is the reciprocal of the tone spacing, which is approximately 0.683 seconds. WSPR messages take about 110.6 seconds to transmit, and always start at even minutes past the hour.

In WSPR, timing is critical, so when using WSPR you must set the time configuration parameter as exactly as possible. Furthermore, make sure that the “Frequency” setting accurately reflects the actual oscillation frequency of the crystal – this is important because the crystal oscillator is the master timebase for generating WSPR tones, durations and deciding when to start the transmission. If careful attention must be given to setting the frequency and the real time clock, then successful WSPR reports will be obtained. Of course these things are easier if you are using a GPS module: the Maidenhead locator will be calculated from the received latitude and longitude, and the time decoded nicely from the GPS serial data stream.

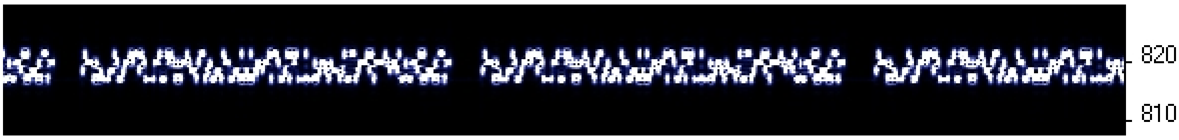
The microcontroller in this kit takes care of the WSPR message encoding algorithm, without any assistance from a PC host computer. It also calculates the tone spacing and symbol duration. When using the Audio output mode, it is recommended to set the Audio frequency to 1,500Hz.

In between message transmissions, the display will show instead just a clock, while we wait patiently for the next WSPR transmission to begin, according to the settings of the configuration parameters Frame and Start:

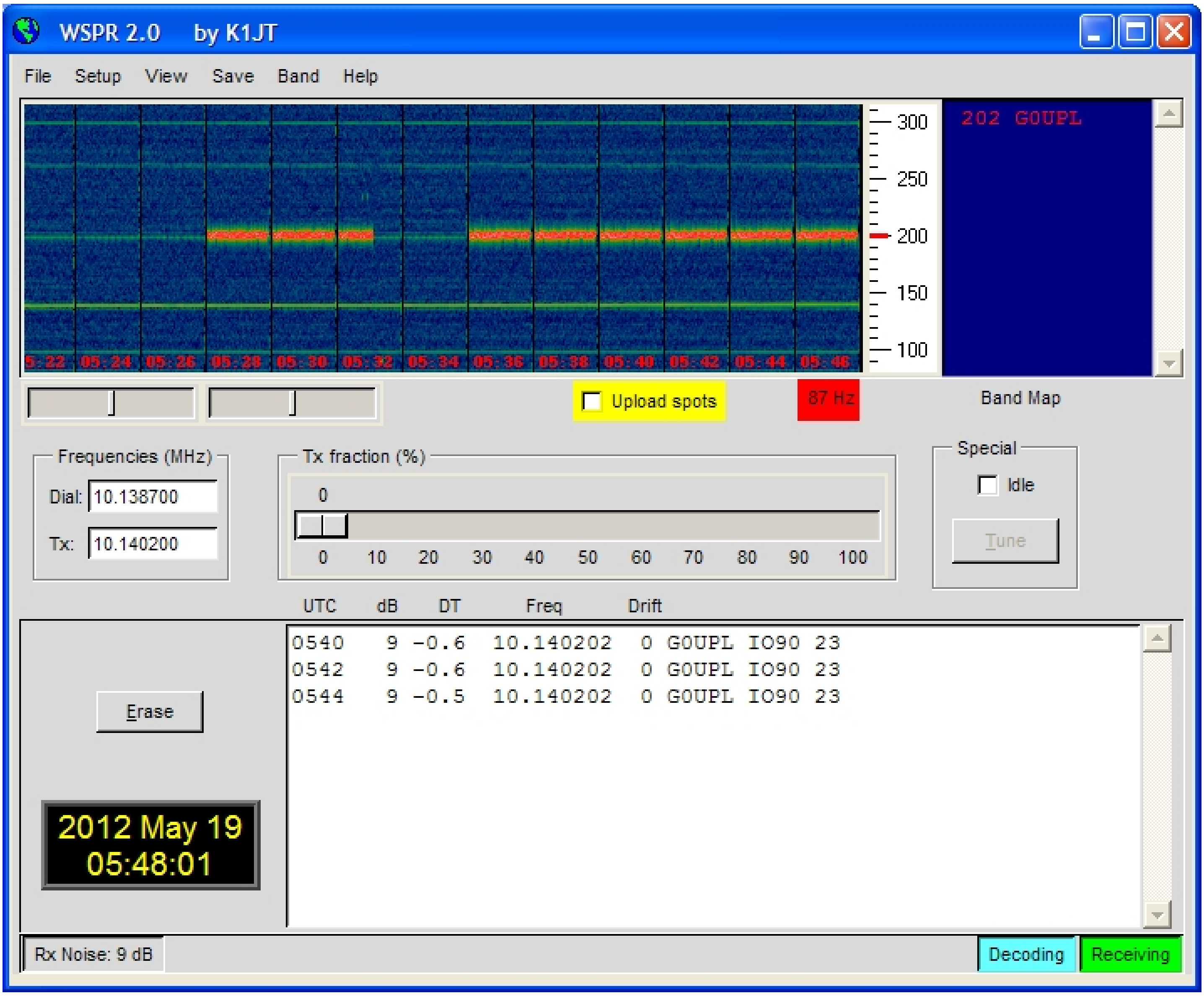


```
WSPR 1,500  13:09:57
```

On Argo, WSPR messages look ugly, like this:



WSPR decoding takes place in the WSPR program by K1JT (see http://physics.princeton.edu/pulsar/K1JT/wspr.html). This is a screenshot showing the WSPR 2.0 screen following reception of a few transmissions (Audio output mode, audio frequency = 1,500Hz, Frame = 02, Start = 00).



202 G0UPL

Frequencies (MHz)

Dial: 10.138700

Tx: 10.140200

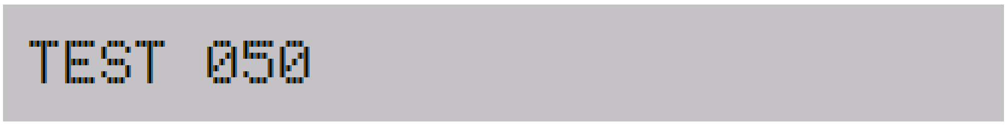
Tx fraction (%): 0

```
UTC   dB   DT      Freq     Drift
0540   9 -0.6  10.140202   0 G0UPL IO90 23
0542   9 -0.6  10.140202   0 G0UPL IO90 23
0544   9 -0.5  10.140202   0 G0UPL IO90 23
```

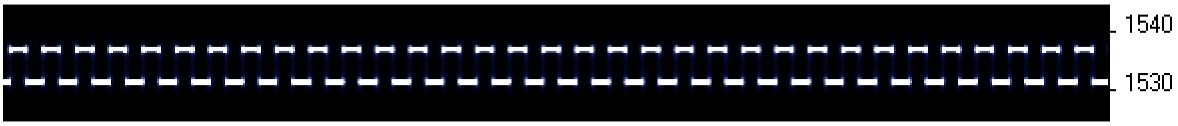
2012 May 19
05:48:01

Rx Noise: 9 dB

7.9 Test


TEST 050

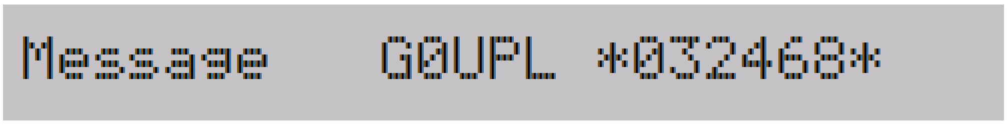
The Test mode is used to calibrate the frequency and FSK setting of the kit. It simply outputs a squarewave FSK pattern, with FSK height determined by the FSK (Hz) setting and period determined by the Speed setting:



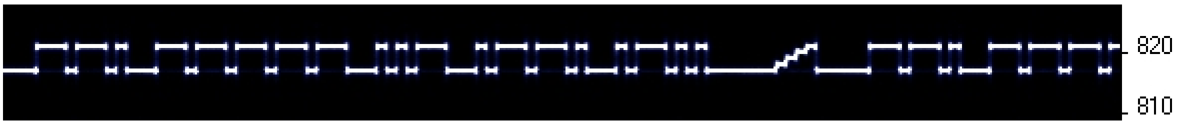
8. Customised message patterns

The kit has the ability to produce customised message patterns in the FSKCW mode. To enter or leave custom mode, a * character is inserted in the message. The first two characters after the * character specify the symbol duration of each symbol, in seconds. Subsequent characters specify the frequency shift of each symbol one by one, in units of 0.5Hz. For example, 5 means a shift of 2.5Hz. After the numbers, you can continue with letters. So A would mean a shift of 5Hz, B a shift of 5.5Hz, C a shift of 6Hz, and so on.

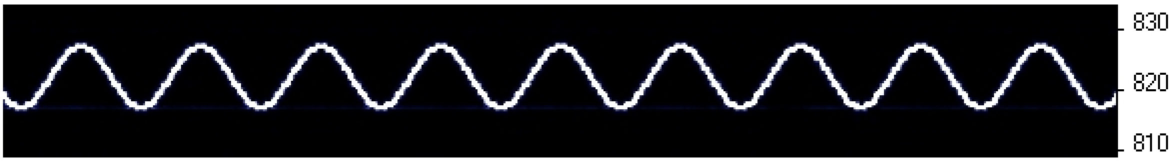
For example consider the following message:


Message G0UPL *032468*

The callsign G0UPL is sent in ordinary FSK/CW with height and speed according to the configuration settings. Next comes a custom message, the duration of each symbol is 03 seconds. The shifts are 1, 2, 3, 4Hz respectively. On Argo, this message produces (with kit configuration FSK (Hz) = 04, Speed = 003):



As an example of what kinds of patterns can be produced, the pattern below is created using message *01ACDFGIJJKKKJJIGFDCA87542110001124578*



9. Calibration, tuning and setting up

This section is REALLY IMPORTANT! 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.

Additionally the FSK size must be carefully set. If it is not, then the WSPR mode may not work at all, and the ability of a GPS receiver module to lock the frequency will be impaired.

The best way to adjust the kit for RF output operation is as follows:

1) Set up an accurate receiver with Argo and monitor the output signal frequency.

2) Use the Test mode (see section above), with Speed set to 005 and FSK (Hz) set to 5. The pattern displayed should be similar to the section shown in the section concerning the “Test” mode, see above.

3) Adjust the trimmer capacitor C8 to get the output frequency to the middle of the band, say 10,140,050.

4) In step 2 above you set the FSK (Hz) to 5Hz. Probably the height of the squarewave on Argo is not 5Hz! The setting “FSK Adj.” must now be adjusted, to produce the correct amount of FSK shift as set in the FSK (Hz) setting, i.e. 5Hz in this case. FSK Adj. tells the microcontroller what value to change the 16-bit PWM register by, in order to get a 1Hz change in RF output frequency. The default value of FSK Adj is 00,400. If the observed squarewave on Argo is higher than 5Hz, you will need to reduce the FSK Adj. value and observe again. For example, if you notice that the squarewave on Argo is 10Hz high, then half the value of FSK Adj.

5) The adjustment of FSK Adj. and C8 trimmer (frequency) may depend on each other, and it may be necessary to repeat steps 4 and 5 a few times, until you have set the frequency to 10,140,050 and the height of the squarewave is 5Hz. You can use Argo's 10-second Slow mode to get a better measurement of the squarewave height.

For Audio output mode only, there is no need to set the FSK Adj. setting. Only the crystal frequency needs to be trimmed accurately using C8.

It is possible to do these adjustments using a frequency counter, but generally an accurate receiver with Argo running on a PC, will lead to an easier and more accurate measurement and set up process.

In the unlikely event that there is no setting of the C8 trimmer that gets the crystal oscillating on 10,140,050, it may be necessary to use different values of capacitors C4 and C5. Lower capacitor values increase the oscillation frequency, higher capacitor values decrease it. C3 sets the amount of effect that the microcontroller can have on the frequency via the varicap (LED). If the frequency range is too large, C3 can be decreased. Conversely if the microcontroller cannot pull the frequency over a large enough range, C3 can be increased. C3, C4 and C5 all affect the frequency and the pulling range so if you change any one of them, you may need to think of changing others too.

10. Factory Reset

This is a new feature available in firmware version v1.06 and above. On powering up the kit, a message is shown for a few seconds on the LCD, which indicates the version number. If the Right button is pressed during the display of this splash screen message, then the chip EEPROM contents are erased and returned to the factory settings.

The effect of this feature is to entirely return the microcontroller chip to its original settings, it is the same as buying a new chip and inserting it. All calibration settings, message, frequency, etc. are erased and returned to the default settings.

11. Resources

Please see the kit page http://www.hanssummers.com/qrsskitmm for any information on latest updates and issues.

Use the QRSS Knights mailing list: http://cnts.be/mailman/listinfo/knightsqrss_cnts.be to announce your operation.

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).

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.

For general information and more QRSS projects, links to other QRSS resources etc., see http://www.hanssummers.com.

Argo QRSS decoding software by Alberto I2PHD: http://www.weaksignals.com/

11. Version History

1 19-May-2012

- Initial draft version

1.0.1 23-Jun-2012

- First official version
- Computer drawn circuit diagram (kindly provided by John VK6JY)
- R2 value changed to 27K, (previously 56K), C2 value changed to 1uF (previously 470nF)
- Added component identification information (capacitor codes e.g. 103 for 10nF)
- Added construction recommendation about a low wattage, fine-tipped iron
- Added photo illustrating the alternate method of LCD connection using wires at 0.1-inch pitch
- Added paragraph about the 3'rd LCD method (90-degree bent pins, upside-down PCB)
- Added warning about not damaging the LCD by stressing it with front-panel bolts

1.01.0 18-Jul-2012

- Document version for v1.01 of firmware
- v1.01: bug fix to the "Frame" parameter (beacon can sometimes stop transmitting after 1 hour)
- v1.01: Lowered the minimum frequency from 1,000,000 Hz to 450,000 Hz

1.01.1 21-Jul-2012

- Corrected pin naming error on circuit diagram (pins 12 and 13 of IC2d were swapped)
- Corrected section heading for test from section 7.7 to 7.9
- Clarification to section 6.3 "Frequency" section, to make it clear that the frequency setting must equal the actual crystal operating frequency.

1.02.0 22-Jul-2012

- Document version for v1.02 of firmware
- v1.02: bug fix correcting the operation of the on/off keying in some modes

1.03.0 31-Jul-2012

- Document version for v1.03 of firmware
- Corrects error on circuit diagram (label in top right corner was to pin 6, should be pin 9)
- v1.03: improved GPS stabilisation algorithm
- v1.03: added ability to use different cursor types on the display
- v1.03: cosmetic bug fix for baud rate setting so it only accepts numeric characters
- v1.03: cosmetic bug fix with the display when waiting for the start of the transmission in "frame" mode – the display was showing the message, overwritten with the real time clock.

1.04.0 07-Aug-2012

- Document version for v1.04 of firmware
- v1.04: bug fix correcting the conversion of GPS latitude and longitude to Maidenhead locator

1.04.1 28-Aug-2012

- Added component values for 160m version on page 2

1.05.0 28-Aug-2012

- Document version for v1.05 of firmware
- v1.05: Bug fix: When “Start” is non-zero, it always transmitted one frame on the hour
- v1.05: Removed the restriction on zero “FSK (Hz)” parameter
- v1.05: Output a high signal on connector F0 when the GPS is enabled and the frequency is locked
- v1.05: Now a space character in a custom message turns off the carrier

1.06.0 22-Sep-2012

- Document version for v1.06 of firmware
- v1.06: Added new “Reset to factory settings” feature
- v1.06: “Waiting” state (in Frame modes) is now output on connector F2 (high when transmitting)

1.07.0 12-Oct-2012

- Document version for v1.07 of firmware
- v1.07: Ignore any invalidly decoded GPS locator (no not write to EEPROM)
- v1.07: Bug fix: rare Latitude/Longitude to Maidenhead locator conversion problem