CCIR 493-4 HF Selcall

A look at the Selcall protocol concentrating on 4-digit “Barret/Codan” format

Background

To enable selective calling on noisy HF SSB radio channels it isn't practical to use techniques commonly found on VHF FM, such as CTCSS or DTMF. Minor tuning errors with SSB, and noise impulses and other propagation disturbances, can seriously degrade the accurate decoding of DTMF and CTCSS. CTCSS is also impractical in that the tones are all outside the typical passband of normal SSB radio. A better solution is to use a relatively high audio frequency and frequency-shift keying (similar to RTTY) to improve the performance. Other techniques are used to detect and where possible correct errors and these make CCIR 493-4 very sensitive in practical HF usage. Accurate decoding of HF selcall can be achieved where the signal to noise ratio is below that which would allow reasonable SSB voice operation.

Part One : Basic Selcall messages

The information transmitted in a Selcall message includes

- The Sender’s ID
- The Addressee's ID
- The Format of the message – Selective Call, Beacon Call
- The Category of the message – Routine
- Additional message elements, often proprietary
- End of Sequence – usually a request for reply

The general format of a Selcall Message

Dotting Pattern	DX/RX Phasing Sequence	A Format Specifier	B Called Party Address 2 Characters	C Category 1 Character	D Self-Identity 2 Characters	E End of Sequence

Symbols

Each piece of a Selcall message is allocated a three-digit “Symbol” value. This is a number between 000 and 127. These “Selcall Symbols” are the heart of the protocol. The different parts of the message (Addresses, Message Format etc.) are coded into specific symbol values – always between 000 and 127. We’ll look at how each part of a message is coded into symbols next.

Addresses

A Secall ID is usually a 4-digit number, although there are variations that allow 6-digit Ids, but these are beyond the scope of this basic discussion.

Format

The Format defines whether a message is a "Selective Call to an individual station" or a "Beacon Call". The latter is used to trigger an over the air revertive from the called station, for path quality assesment, without alerting the operator at the station being interrogated..

Format Values

Format	Meaning
120	Selective Call to an Individual Station
123	Beacon Call to a specific station

There is another form of "Beacon" which uses format 120 (Selective Call) but a "wildcard" address, ending in **99**. **Beacon-99** calls will be dealt with later.

Category

Messages can have one of four "Category" values which show the importance of the message. Each different Category is allocated a specific number. In general it appears that only "100 / Routine" is used.

Category Values

Category	Meaning
100	Routine
108	Safety
110	Urgency
112	Distress

The End of Sequence symbol

Most, if not all, Selcall signals observed use the **EOS** symbol **"117"** for **Request Repsonse**. Other EOS symbols are possible e.g. 122 (ACK) and 127 (EOS)

End Of Sequence Values

End of Sequence	Meaning
117	Ack RQ (REQ)
122	Ack BQ (ACK)
127	EOS

Building a message – a worked example.

The general format of a Selcall Message

Dotting Pattern	DX/RX Phasing Sequence	A Format Specifier	B Called Party Address 2 Characters	C Category 1 Character	D Self-Identity 2 Characters	E End of Sequence

An "Selective Call to an Individual Station"

A Format : *Individual Stations Call*	**120**
B Called Party Address:	**3602**
C Category : *Routine*	**100**
D Self-Identity Address:	**3701**
E EOS : *REQ* (*RQ*)	**117**

We can now slot these values into the correct places to build the message.

The 4-digit IDs are split across 2 symbols

The "Called Party" of 3602 encodes as: 036 002

The "Self-Identity" of 3701 encodes as: 037 001

The basic message is now:

A	B	B	C	D	D
120	036	002	100	037	001

We add the EOS of 117

120 036 002 100 037 001 117

This is the basic message

The EOS symbol is repeated twice:

120 036 002 100 037 001 117 117 117

Error detection

Symbols and Parity and the “10 to 7-bit Parity Test”

We know that the Selcall message is composed of “symbols”, numbers between 0 and 127 which carry the information and that any symbol, with a value between 0 and 127, can be represented in binary with seven bits:

Decimal **0**   = Binary **0000000**

Decimal **127** = Binary **1111111**

Each symbol is therefore a 7-bit number. To allow for detection of bit errors an extra three bits, “the parity bits”, are added to each 7-bit symbol, prior to transmission. This allows the receiver to perform a “parity check” to determine that the symbol has (probably) been received correctly.

The parity check is a number between 0 and 7, expressed in 3-bit binary, and is a count of the number of “zeros” in the original binary 7-bit symbol.

The Parity Check allows us to determine, to a degree, whether an individual symbol has been received correctly.

Let’s look at how we convert a **7-bit Symbol** into a **10-bit word with parity**.

Using the message that we’re building:

120 036 002 100 037 001 117 117 117

The first symbol **120** in 7-bit binary is:

120 = 1111000

We count the number of “zeros” in the 7-bit symbol. There are **THREE**.

The parity check bits are therefore the binary for **3 = 011**

To make the actual 10-bit word that we want to transmit we *reverse the order of the original 7-bits* and then add the new 3-bit parity bits to the end:

Our 10-bit parity protected word is **0001111011**

How does the parity check help us?

Passing a Parity Check

We’ll “reverse engineer” this 10-bit word back to our original symbol.

0001111011

The last 3 bits represents the number of zeros we expect to find in the first 7-bits (the real data).

011 in binary = **3** in decimal**.**

We expect 3 zeros in the remainder of the 7-bits : **0001111** and there are indeed 3 zeros. Our word has "passed the parity test". We reverse the order **1111000** and convert to decimal = **120**

We now know how to check a 10-bit word for "parity errors" and to re-create the original 7-bit symbol value, if the parity check is "good".

Failing a Parity Check

Suppose we received the 10 bit word **0001011011**

Can we check if it's a valid word?

The parity bits **011** tell us to expect THREE zeros in the main part of the symbol. There are actually FOUR zeros. The word is corrupt and must be ignored.

Suppose we received **0001111010**

Can we check if this one is valid?

The parity bits **010** tell us to expect TWO zeros in the main part of the symbol. There are actually THREE. The word is corrupt, and also must be ignored.

Passing a Parity Check, even when there is an error

Suppose we received this 10-bit word **1001111010**

The parity bits **010** tell us to expect TWO zeros in the main part of the symbol. There ***are*** actually TWO zeros. The word has passed the parity check.

Suppose though that this was originally transmitted as **0001111011 (120)**

Comparing the two copies:

1001111010

0001111011

Two bits are different. The received word **1001111010**, when converted back to 7-bits, and reversed, becomes **1111001**, which is decimal **121**.

This is the incorrect value – although it's "passed the parity test". Two bits being swapped can lead to false positives.

Forward Error Correction

Time Diversity Interleaving : The DX and RX copies

Each 10-bit word is sent twice, to give the receiver two opportunities to get an accurate version of each symbol. The symbols are sent once in what is called the "**DX**" position, and after four other symbols have been transmitted they are sent again, in the "**RX**" position. The bit-rate of CCIR 493-4 (100 baud) is such that the intervening four symbols (of 10 bits each) between the **DX** and **RX** copies of any symbol take 400ms – so each symbol is sent twice spread out by 400ms. A burst of noise, or a fade, of less than this duration won't wipe out both the **DX** and **RX** copies of any symbol. Even if one copy is missing, ***as long as the other copy is intact*** we can still reconstruct the message. The **Parity Check** is thus a valuable tool for deciding whether to discard one or other of the **DX** or **RX** copies.

The Message – almost ready for transmission

To build the message we've taken the following steps

- Create the basic message :   **120 036 002 100 037 001 117**
- Add copies of the EOS characters : **120 036 002 100 037 001 117 117 117**
- Interleave the DX and RX copies – separated by 4 intervening words

This gets us here, with the "DX" copies in the usual colour and "RX" copies in **black**

120 120 036 120 002 120 100 036 037 002 001 100 117 037 117 001 117 117

Dotting and Phasing

The very start of a message is a "dotting pattern" – a series of alternating 1s and 0s to allow the receiver to synchronize with the bit-rate of the message. The length of the dotting period is chosen to ensure a receiver which might be scanning multiple channels will detect a new call during the dotting period – different implementations use dotting periods of up to 20 seconds, possibly more.

After the Dotting Pattern comes the "phasing sequence". To find the symbols from the series of 1s and 0s we need a sequence of "known symbols" which lets the decoder intially find the start of the message, and then to find the boundaries between 10-bit words.

The sequence of DX and RX characters in the Phasing Sequence is

125 109 125 108 125 107 125 106 125 105 125 104

The receiver takes in the bits one at a time, and looks for the pattern "`125 109 125 108…`" etc. by shifting the bits along one at a time until the phasing pattern is found. There are several chances to find the phasing symbols. Once a selection of "known" symbols from the phasing sequence are detected we will be correctly "locked" to the word boundaries, and will be able to count off 10 bits at a time, and treat each 10-bit chunk as a new "parity protected word" for processing.

The full time-interleaved message is now (DX and RX)

```
125 109 125 108 125 107 125 106 125 105 125 104 120 120 036
120 002 120 100 036 037 002 001 100 117 037 117 001 117 117
```

The symbols now need to be converted to 10-bit parity protected words and then we'll have a stream of 30 words x 10 bits = 300 bits. Add on a typical 6 second (600 bit) dotting pattern we have 900 bits. At 100 bits per second, a typical Secall message on MF/HF takes approximately 9 seconds to transmit. Longer dotting sequences, depending on the manufacturer, will extend the overall length, and allow larger groups of channels to be scanned.

Modulation and transmission characteristics

CCIR493-4 is transmitted as a Frequency Shift Keyed (FSK) signal at 100 baud and the frequency shift is 170Hz. The modem centre frequency is 1785Hz. The tone representing a binary 0 being the lower of the two transmitted tones, 1700Hz and the tone representing binary 1 is the higher of the two, 1870Hz. The bit period is 10mS which this means that each bit-period does not contain an integer number of cycles. Switching between tones at each bit-boundary would therefore lead to discontinuities in the waveform and an excessively wide bandwidth. To counteract this a technique known as “Continuous-Phase Frequency-Shift-Keying” is used which prevents discontinuities of the waveform as the tone frequency shifts from “mark” to “space” values.

Recap - Error detection and methods to improve reliability

To improve the successful reception of messages, and to provide a means of detecting errors, CCIR 493-4 uses two methods.

1) Parity checking in each transmitted symbol

2) Repeat transmission of each symbol. They are sent again after four other characters, so each symbol is sent a second time after 400ms have elapsed, which means a burst of noise, or interference, must be longer than 400ms before it can destroy both copies of the same symbol, and we only need one copy to be received correctly to construct the received message.

Part Two :

Types of Messages

Format 120 "Selective Call"

These are the basic message types used to make a call to a remote station, and alerting the distant operator that a call has been received. When a message addressed to the Selcall ID of a station is received the radio will firstly send an over-the-air "revertive" signal, generally a distinctive series of beeps, to confirm to the caller that the call has been received. The radio that received the call will then sound an audible alarm to alert the operator. Some installations may also flash vehicle lights, or sound the vehicle horn.

Format 123 "Beacon Call"

This type of message is used to estimate the path quality between two stations, without alerting the operator at the distant station. A "Beacon Call" is addressed to a specific station, the same as a "Selective Call", but is transmitted with the format symbol value "123" rather than "120". The station being called will interpret this message differently to a "120 Selcall". A different revertive will be sent over-the-air for the benefit of the calling operator, but no local alert will sound and the radio will simply resume it's previous state (usually scanning a selection of channels). The over-the-air revertive is used by the calling operator to audibly judge the signal-to-noise ratio and to make a decision as to whether this particular frequency is suitable for a voice call. Beacon calls can be done on each channel used by the net, and the best channel can be assessed, all without interrupting the operator at the distant station.

Format 120 "Beacon-99 Call"

This is another type of Beacon Call, but is sent with Format symbol 120, like a "Selective Call". The Addressee ID is set to **XX99**, where XX is the first two numbers of an common group of addresses – e.g. imagine a net with members 3601, 3603, 3643 and 3675. Sending a "Beacon-99" call to "3699" would generate revertives from any of the "36" stations who successfully decode the call. No "operator alert" is generated at any station responding. The operator making the Beacon-99 call can now know that one (or more) stations in his net (the net of 36XX stations) are in range.

Collected Selcall information

1. There are protocols for requesting Telephone connections, with the required 'phone number carried in the message symbols

2. Barret Radios (2050, 950 etc) have a “Status” call. This causes the station addressed to send a reply containing details of (e.g) Last Received signal strength, PSU voltage, TX power, VSWR, Firmware version etc. An example of a Status Call can be seen in a You Tube video: https://www.youtube.com/watch?v=r3-NkR9JvYY . A first attempt to decipher the protocol shows that the Outgoing Status call is of the form: `120 033 099 100 088 099` **`103 103`** `117 117 117` and the revertive is of the form: `120 088 099 100 033 099 100 111 001 073 059 053 006 012 002 059 127 088 099 117 117` The Station being called is “3399” and the caller is “8899”. The Status Call request appears to be a “Format 120” selective call, with two extra symbols after the Self-Address symbols, both of these extra symbols are “103”. The reply, containing the status information is from “3399”, addressed to “8899”. Following the Self-Address symbols is “100 111 001 073 059 053 006 012 002 059” then “127” (all ones) and then addressee ID “3399” is sent again before the EOS “117” symbol(s).

3. Wild-card “Group” and “Super Group” calling is found in some implementations.... (more to follow)

4. I have written a Python Selcall generator: https://dl.dropboxusercontent.com/u/3551430/selcal_gui.zip which generates Selcall (120) and Beacon Call (123) as well as a morse code callsign for use on the amateur bands

5. Youtube video of the Icom IC-F8101 sending “Open Selcall” : https://www.youtube.com/watch?v=458fqmL0G-U The calls appear not to decode using Sorcerer, or another simple demodulator based on YaDD (the DSC decoder – DSC shares many of the same features). Further investigation is needed to see why these messages don't decode – are they CCIR493-4 at all??

6. Youtube video of a Barrett 250 on the VKS737 network : https://www.youtube.com/watch?v=YTRDOuxDdIl – no selcall signals are audible, but the basic procedure for Selcall & Beacon-99 are shown.

7. Youtube video of Q-Mac HF90 selcall (haven't yet tried decoding the audible selcall : https://www.youtube.com/watch?v=RC2bUuvWdjE

8. HFLink Yahoo group. Primarily HF ALE, but with some Selcall users, and a “co-ordinated Selcall ID database” and some information on CCIR493-4 on the sister site: http://hflink.com/selcall/

9. VKS737 Australian outback network. Base station map: http://www.vks737.on.net/pdfs/DOC%2019.pdf

10. Manuals for Land Mobile radios, many have Selcall : http://www.hf-radio.com.au/manual-menu/manual%20index.htm

11.