An Introduction to Datamodes

AMTOR

AMTOR is a specialised form of RTTY. The term is an acronym for ‘Amateur Tele-printing Over Radio’ and is derived from the commercial SITOR system (Simplex Telex Over radio) developed primarily for Maritime use in the1970s. In the early 1980's, Peter Martinez, G3PLX, made several minor changes to the SITOR protocol and called it AMTOR.

AMTOR improves on RTTY by incorporating a simple Error Detection technique. The system remains relatively uncomplicated but AMTOR performs well even in poor HF conditions. While there can still be many errors in AMTOR data, the Error Detection helps a lot and the result is quite tolerable for normal text mode conversations because of the high redundancy in plain language text. Certainly much better than RTTY. But for more critical data such as program code, or even some technical information messages, NO errors can be tolerated. There are two modes used in Amtor: ARQ and FEC.

ARQ

This mode is a little different in that it is a Synchronous protocol, which means both stations are synchronised to each other's signals. In ARQ mode (Automatic Repeat Query), sometimes called Mode A, data is sent in groups of 3 characters. Although each character is only 5 bits (same as for RTTY), two additional control bits make it up to 7 bits per "character" and they are set so there are always 4 marks and three spaces in every transmitted character. If the receiving station gets some other combination it knows an error has occurred. The 40 percent overhead is considered worthwhile to get some error detection. This technique can identity a lot of errors that might occur but is not as thorough as the methods used in PACTOR and Packet which we look at later.

The receiver responds to each 3 character group by sending either an ACK (ACKnowledge) code (if OK) or a NAK negative ACKnowledge). Each time the transmitting station gets a NAK, that 3 character group is sent again.

If you listen around on the HF bands in the recognised Data Segments of the bands, you might hear a chirp-chirp sound that identifies an ARQ transmission. Even when there is no data actually being transmitted, the transmitting station continues to send idle "chirps" to maintain the link.

Your AMTOR equipment probably supports a Listen Mode too and that allows you to monitor another ARQ session even though you are not participating in the session with the usual acknowledgements. Of course that means you don't get the opportunity to say "NAK" if you don't copy something properly!

FEC

In FEC mode (Forward Error Correcting), sometimes called Mode B, the sending station sends each character twice so this mode provides a means of transmitting to several stations at once. The receiving station does not acknowledge the data received. If a receiving station matches both instances of a character, that character will be printed, otherwise some error symbol is printed. This mode does not provide for the receiver to ask for the missing data to be retransmitted. An FEC transmission sounds more like a Baudot RTTY signal.

The two stations need to keep in phase with each other so each FEC transmission is started with several sets of "phasing pairs" and these are sent at regular intervals even while there is no data being transmitted. FEC Mode is still better than ordinary RTTY but its error detection is not as reliable as that in the ARQ Mode.

AMTOR systems are still limited to the technology of the 60s with limitations such as the character set and the maximum transmission rate (100 baud) geared to the mechanical teleprinter. The Error Detection technique provides improved accuracy over the "vanilla" RTTY mode, but is still not entirely reliable. It is perhaps better termed Error Reduction than Error Detection and has limited application for critical data.

PACTOR

Amateur Packet Radio has been in use since the early 8O's and is used extensively on both HF and VHF/UHF. But it is better suited to the VHF/UHF bands. It was found AMTOR would often perform better than Packet on the HF bands and so despite the limitations of AMTOR, a large number of HF operators preferred AMTOR to Packet Radio.

As computerised systems became more popular, amateurs began to explore more sophisticated options for AMTOR. Various alternatives were tried but every time, AMTOR's poor error handling prevented any significant breakthrough. And as AMTOR did not handle the standard ASCII code, a new Protocol was called for.

In 1987 two German amateurs, Hans-Peter Helfert (DL6MAA) and Ulrich Strate (KF4KV) began work on a new Protocol which they called PACTOR.

Although this new system incorporated the basic structure of AMTOR with its fixed interval data blocks and corresponding acknowledgements, PACTOR also combines important characteristics of Packet Radio.

PACTOR is designed especially for HF operations. There is no reason why it could not be used on VHF/UHF, but it does not offer significant advantages over Packet Radio on those bands. It is on HF that the new protocol shines.

The block length of PACTOR is fixed at 1.44 seconds with significantly lower overheads than those of AMTOR. This was found to be the most suitable compromise for HF operations with the relatively short block having a better chance of survival through difficult conditions and allowing short break-in times. The protocol provides for automatic adjustments to the baud rate too, so when the going gets really tough it will reduce the speed; and when things improve it will increase the speed to improve the throughput again.

Ruffinan data compression is used in PACTOR and that adds to the effective performance. The Huffinan code is based on the assumption that in plain language text there is a high content of frequently occurring letters such as E, A, etc. It has been found that a block of normal ASCII text can often be compressed to half its original size without any loss of data at all.

PACTOR offers high quality error detection, which overcomes the main weakness of AMTOR. PACTOR uses a longer acknowledgment signal as well as the CRC checksum code used in Packet Radio to detect errors in the transmission.

Other features of PACTOR include an improved error correction system which has become known as "Memory ARQ". It improves the performance of the receiving system in weak or noisy conditions. For example in any digital system, a signal is reduced to a series of bits which are either a binary 1 or 0, but this does not distinguish between an input signal of a millivolt or 10 volts! So long as the analogue input signal is above the threshold, the digital output data is the same!

The Memory ARQ system remembers not only the binary is and Os, but also the incoming analogue values, so if the received block reveals errors, the analogue values from the repeated blocks can be combined into a sum block and subjected to another CRC error check. While this system is not perfect it can provide error free communications in conditions that other modes, including Packet, could not handle.

RTTY

RTTY or RadioTeletype is a direct machine to machine communications mode using the Baudot (or Murray) code. This mode became popular with many amateurs when surplus TTY machines became available at a reasonable cost after World War II. These mechanical monsters provided a keyboard for Input and a paper roll for printed Output. They were also useful to help hold the house down in times of hurricane winds - they must weigh a ton. Video displays were still too exotic and expensive in those days. It was not until the mid-1970s that we began to see the Video Display come into more widespread use. (By the way, have you ever wondered why early Program Languages like BASIC use the command PRINT to display their output?)

When transmitting Morse Code, the transmitter is switched on and off to make the ‘dits’ and ‘dahs’. When sending Teletype however the transmitter runs continuously, sending either of two frequencies conventionally known as Mark and Space (a reference to paper tape reception of telegraphy). The early pioneers found on-off keying was not all that successful for Teletype signals because of interference from static.

They experimented with FSK, or Frequency Shift Keying and found it performed much better. With FSK, the transmitter is shifted up in frequency every time a Mark is to be sent, reverting to the lower frequency for a Space. The amount of the shift is usually 170 Hz for Amateur Radio use although many commercial Teletype signals use other shifts, notably 425 Hz and 850 Hz.

Many systems use AFSK or Audio Frequency Shift Keying. When this is sent, the transmitting station generates the Mark and Space audio tones and feeds them into the transmitters microphone input. The result at the receiving end is that the same audio tones are heard and processed, whether the transmitting station used FSK or AFSK.

When listening to a teletype signal off air, you will soon get to recognise the familiar warble of Mark and Space tones.

In the amateur shack the RTTY machine is usually connected to an HF receiver or transceiver which the operator tunes so that the received audio is just the right pitch or audio frequency to trigger the demodulator's Mark and Space resonators.

If the receiver is slightly off the correct frequency the tones vary and the text becomes garbled or even lost altogether. To help the other station tune the receiver correctly, a RTTY operator can send a string of alternate R and Y characters viz: RYRYRYRYRY. This pattern is chosen as it produces the most frequent and almost symmetrical alternation of Mark and Space tones, giving the receiving operator the best chance to tune the receiver before the "real" message starts. However, even if the signal is accurately tuned, the information can become garbled or completely lost due to interference, fading, or noise. Often, it is possible to make sense of the message even with parts missing, but RTTY is by NO means an error free mode!

I should point out that similar problems exist for other modes including Packet. While information can still fail to get through on the more sophisticated modes the Error Detecting capability of some, especially Packet and PACTOR, ensure that the operator will receive either accurate information or nothing at all. Usually, where "nothing at all" is received, the information will automatically be retransmitted when the radio is retuned, or the interference stops, (etc) and nothing is lost.

The Baudot code is a 5 bit code and those of you who are familiar with Binary Notation will know that the maximum number of values we can have with 5 bits is 32. That means that each unit of transmission, one keystroke if you like, can contain any one of 32 possible values. If you look up a table of Baudot codes you will see there are 32 values listed, one code for each letter of the alphabet plus a few other codes for other things such as a space and a Carriage Return. But, what if we want to send a number such as "9" or a question mark? These are not mentioned in that table because all 32 codes are already used~

The solution is rather similar to the Typewriter or Computer Keyboard where we have the Shift key to get various additional codes from the keyboard. Most keys will produce a different result if we hold down the Shift key as we type. Well, one of those original 32 codes is a special code known as FIGS (for Figures Shift). The convention is that when we want to send a number or some other special character such as a punctuation mark, we can do that by firstly transmitting a FIGS code.

Then instead of using that original table of 32 codes, we have a second table of codes to use, and that second table includes all ten numeric digits and various punctuation marks. Provided both sides of the conversation observe the convention, the sender can send a FIGS and start using the second table; the receiver will see the FIGS code and it too will interpret all data that follows from the second table.

With just 5 bits of data we then have almost 64 different codes we can send and receive. (I say almost because there is some duplication in the two tables, including a space and a Carriage Return but that is not important here). Even that many codes is not enough to handle all 26 letters of the alphabet in both UPPER and lower case, so RTTY systems always operate in upper case only.

If we wanted to type a big number (say "13579") we don't have to send FIGS before every digit. We send that code only once and the receiver then will take EVERYTHING we type from now as if it belongs in the second table. When we want to revert to the normal alphabetic table of codes we can send another special code, this one called LTRS (for Letters Shift). Then everything goes back to normal, using the original alphabetic table of codes.

Normally we don't have to concern ourselves with these FIGS and LTRS codes. Our computing equipment will take care of those things for us. We just type away and rely on the system to generate and send those codes when necessary

Today, RTTY is still a popular mode especially on the HF bands, and the advent of the "Glass Terminal'~, firstly the Dumb Terminal and now the Personal Computer, has brought this mode to even more operators the world over. Many specialised RTTY systems were developed for the Amateur enthusiasts but have been superseded now by the Personal

Computer with one of the Multi-Mode TNCs which handle RTTY and many other modes besides.

The latest Computerised RTTY equipment generally allows us to use the mode better, quieter, more efficiently, using less power and occupying less space than the old TTY machines, but the limitations of the mode remain.