Amtor, an Improved Error-Free RTTY System†

Sitor, Spector, Microtor and now Amtor — what do they have in common? For one thing, the underlying principles are similar. More importantly, they just might usher in a new era for RTTY communication.

By J. P. Martinez,* G3PLX

Since getting a microprocessor-based home computer working at my station, I've spent some time using it to perform many of the functions of conventional RTTY equipment. The flexibility of the microprocessor (μP) also made possible the experimentation with techniques using other than the well-known start-stop RTTY code. In the UK, we are permitted to carry on experiments of this sort on 2 meters and above, so no time was lost in trying out synchronous systems where no start-and-stop bits are sent and the clocking of data is done by accurate frequency standards at both stations. Some forward-error correction codes were tried in which additional check-bits sent with the data enable the receiver to correctly reconstruct the original data in the event of the presence of some erroneous bits. This proved promising, being about 6 dB better than conventional RTTY.

Another area explored is the ARQ (Automatic ReQuest) whereby any errors at the receiving end are detected by the use of extra parity bits, and a request is made for the repeat of the bad character by the receiving station. One such system, requiring both stations to operate in the duplex mode, gave spectacular results via OSCAR satellites. Under this circumstance, there was complete immunity to fading, interference and errors associated with keeping the receiver on tune. Loss of signal merely caused temporary pauses in the traffic.

Adapting this system to everyday amateur operating practice proved difficult until the "discovery" of an ARQ system already in use in the maritime service for Telex traffic. This system can be used by two stations in simplex communication on the same frequency by working in a synchronized quick-break fashion. On-the-air results from this system were similar to those of the duplex technique. It became clear, therefore, that it would be very useful to amateurs, not only on vhf, but also on the hf bands.

Since this system is already an international standard (CCIR Recommendation 476) and is in worldwide use, we had no difficulty in gaining permission for its use on hf by UK amateurs. Commercially, this system is known by various trade names such as Sitor, Spector and Microtor. To avoid confusion with the commercial equipment, the name "Amtor" has been devised to refer to any amateur use of the system described in CCIR 476. What follows is a description of how Amtor works. I hope to show that this ingenious system could have a lot to offer and that it can be readily implemented by modern μP techniques using either a home computer or a dedicated unit.

First Principles

Imagine two stations, A and B, in simplex communication on ssb with the operators desiring to exchange messages reliably under poor conditions. If A sends three words, for example, B replies with "Roger" or "Say again." A then goes on to the next three words or repeats the last three. If A cannot tell, however, whether B said "Roger or "Say again," then he will have to say instead of three words, something like "Please repeat." To make matters worse, if B cannot tell whether A gave three words or said "Please repeat," leading B to transmit "Say again," then A gets completely confused and doesn't know what to say. This may seem trivial, but if we are to automate this verbal ARQ system, a better method must be found.

In Amtor, A sends three characters in a burst of synchronous frequency-shift data. B, in response, sends the acknowledgement signals in the reverse direction as a single character. How, then, can Amtor provide a solution for the communication problem mentioned above? A practical approach is to encode the acknowledgement signals differentially using two control characters we may call C1 and C2. When B is copying correctly, he replies with C1 and C2 alternately after each block. If a bad block is received, he repeats the same control code as the last time. If A sends "Please repeat," then B repeats the same control code as the last time. Thus B's reply is the same for a "Please repeat" block as for an error. It doesn't matter, therefore, if the bad block was a "Please repeat" block.

In the ssb example, B knows when errors have occurred because he cannot recognize a word. This works because the number of recognizable words is much smaller than the number of different sounds. To put it another way, language contains redundancy. The only errors that will pass undetected are those which transform one word into another. This can be minimized by careful choice of words. There are 32 recognizable characters in the teleprinter system. These

---

STA Granted for Experimental Error-Free Amateur RTTY

The Federal Communications Commission in 1980 granted Special Temporary Authority to four licensed radio amateurs for the purpose of conducting tests with an error-free mode of amateur teleprinter communication (Amtor). This authorization permits the use of digital teleprinter code as described in CCIR Recommendation 476 (Rev. 74). It expires November 30, 1981.

Participants in this STA are William C. Meyn, K4PA, of Reston, Virginia; Charles A. Roettcher, K3FLS, of Washington, DC; Melvin Leibowitz, W3KET, of Wilmington, Delaware; and Walter E. Kaelin, KB6BT, of Saratoga, California.

---

*11 Marchwood Ct., Broadsands Dr., Fort Gomer, Gosport, Hants, England

†Translated and reprinted from *OZ* (Denmark), November 1980.

Table 1

Conversion Between Amtor Code and Murray Code

The codes are transmitted left to right. The higher frequency of the fsk signal is represented by "1."

Murray Code	*Ltrs*	*Figs*	*Amtor Code*
11000	A	—	1110001
10011	B	?	0100111
01110	C	:	1011100
10010	D		1100101
10000	E	3	0110101
10110	F		1101100
01011	G		1010110
00101	H		1001011
01100	I	8	1011001
11010	J	bell	1110100
11110	K	(	0111100
01001	L	)	1010011
00111	M	.	1001110
00110	N	,	1001101
00011	O	9	1000111
01101	P	0	1011010
11101	Q	1	0111010
01010	R	4	1010101
10100	S	'	1101001
00001	T	5	0010111
11100	U	7	0111001
01111	V	=	0011110
11001	W	2	1110010
10111	X	/	0101110
10101	Y	6	1101010
10001	Z	+	1100011
00010	carriage return		0001111
01000	line feed		0011011
11111	letters		0101101
11011	figures		0110110
00100	space		0011101
00000			0101011
	RQ		0110011
	beta		1100110
	alpha		1111000
	control 1		1010011
	control 2		0101011
	control 3		1001101

normally are transmitted by 32 combinations of 5 data bits. If 5 data bits are used, any bit error will transform one character into another and the error will pass undetected. Amtor has the advantage of using 7 data bits giving 128 combinations of which only 32 are recognizable. Careful choice of which 32 are used minimizes the possibility of an undetected error. One would not, for example, have chosen two codes that only differed by one bit. In fact, only those codes with three zeros and four ones are used, making it easy to check for errors at the receiving end. There are 35 such codes, and so the three spares are available for control purposes. Among these is the RQ character used by the transmitting station to signal "Please repeat." There is also the idle character known as beta. A third character, alpha, is explained later in this article. The C1 and C2 codes and C3, to be explained shortly, are also 7-bit characters from the same set. Since these are always sent only in the reverse direction, they are never confused with the others. The conversion from Amtor code to standard teleprinter characters is shown in Table 1. Note that this code is designed to translate easily to and from the Murray code.

The changeover in direction of transmission is not left to the operators, for there could be a misunderstanding if the link fades out just before the expected end of an "over." There are two ways to signal for the changeover. The sending station may end an "over" with the two-character sequence +? or the receiving station operator may press the TRANSMIT button. By either method the receiving station stops replying with C1 or C2 and instead acknowledges with C3. Upon receiving this information, the sending station transmits the block "beta-alpha-beta." In response, the receiving station transmits an RQ character, whereupon the transmitting station goes to the receive mode. Bursts of data from each station are so timed that even if both are transmitting blocks momentarily, each one can still receive one character of the other's block in the position expected to be a control code. This seemingly complicated process does ensure that the changeover proceeds in an orderly manner and cannot go awry, no matter what.

Timing of the various signals is shown in Fig. 1, with some of the possibilities for errors. Note that the two stations do not behave identically in respect to timing. One is called the master station and the other the slave for reasons which will become apparent shortly.

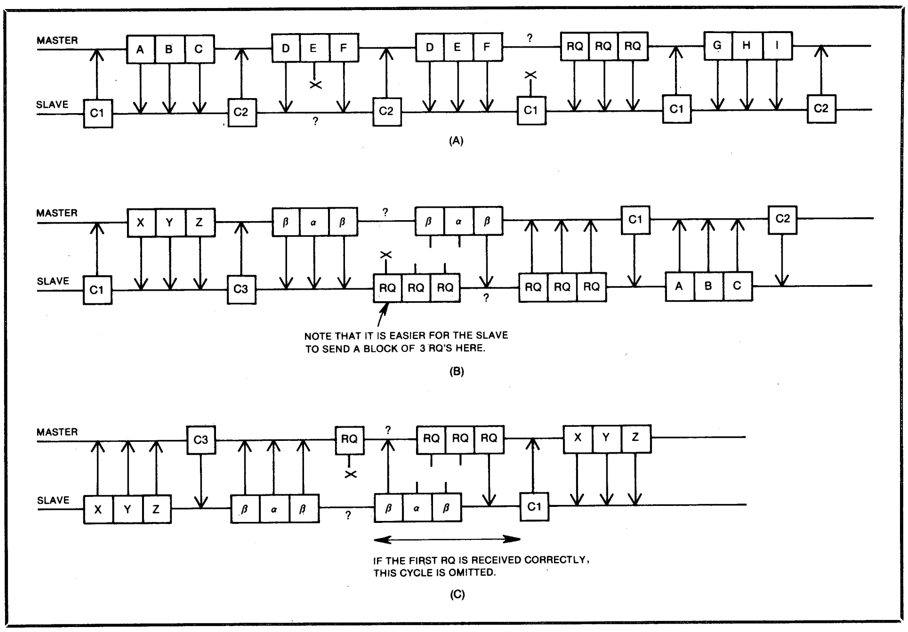

Fig. 1 — Timing of various Amtor signals. Included are some possibilities for errors. As you study this illustration, note that the two stations do not behave identically in respect to timing. One station is designated as the master, the other as the slave. Shown at A is a representation of the master sending to the slave, with errors. At B is a situation where there is a changeover from master to slave sending. The master is once again sending to the slave in the situation at C.

Performance

Although Amtor, in common with any ARQ system, eliminates virtually all errors resulting from the radio link, it is worthwhile pausing to see exactly how good it is. A simple analysis can be made by supposing that the radio link alternates between perfect copy and perfect random noise. With only noise in the receiver, all 128 7-bit patterns are likely to be received with 34 of these being acceptable (the RQ character is treated the same as an error). Thus the chances of a whole block of three being accepted by mistake is $(34/128)^3$ or about 1.9%. Therefore, with no signal, the receiving printer will be idle for 98.1% of the time while the system is asking for repeats and will be printing garble for 1.9% of the time. This compares with 100% correct copy when the signal is good.

By using the foregoing information, we can calculate the proportion of garble to good copy for various proportions of good signal to bad. A similar analysis for the reverse path shows that when there is no signal in this direction, 0.8% (1/128) of the message is unwittingly lost into thin air. The combined effect of these factors is shown in Table 2.

Synchronization

Since Amtor is a synchronous system with no start and stop bits, the timing at both ends must be stable. Some means must be found to get the two stations in step and to keep them that way over a period of time, even if the two clocks are only slightly different in speed. The synchronization procedure starts with the first station (the master) sending a special sync block repeatedly. The slave station continuously shifts in received bits until 21 consecutive bits correspond exactly with the expected sync pattern. The slave then starts to reply in the gaps, sending back

Table 2

Amtor Performance

The assumption is made that the signals in both directions alternate between perfectly good and perfectly bad.

Percentage of Time Signal is Usable	*Percentage of Transmitted Message Received Correctly*	*Number of Spurious Characters Printed, as Percentage of Transmitted Message Length*	*Time Taken as Multiple of 100% Signal Case*
100	100.0	0	1.00
90	99.9	0.2	1.11
80	99.8	0.5	1.25
70	99.7	0.8	1.42
60	99.5	1.2	1.66
50	99.2	1.9	2.00
40	98.8	2.8	2.50
30	98.2	4.4	3.30
20	96.8	7.5	5.00
10	93.0	16.9	10.00
5	85.2	35.6	20.00
2	61.7	91.8	50.00
1	22.7	185.5	100.00

one of the control codes. The master station, meanwhile, has been shifting in received data bits during the gaps in its transmissions. When it recognizes two consecutive control codes, it stops sending sync blocks and changes to sending traffic. In fact, to guard against the possibility of the slave getting the sync pattern right, just by chance, the master sends two different sync blocks alternately, and the slave must get them both in order correctly to lock in. The first of these blocks has an RQ in the second character, with two alphabetic characters in the other two positions. At the same time, the second block has two more alphabetic characters in the first two positions with an RQ in the third. The RQ characters prevent the four alphabetic characters from printing out at the slave station. These four characters can be chosen by the users, but must be agreed upon beforehand by the operators at the two stations concerned.

In commercial maritime service, these characters form a selective-calling code; but for amateur use, the four-character group suggested for all random QSOs is, perhaps not surprisingly, CQCQ, so that the two sync blocks are C,RQ,Q and C,Q,RQ. Alternatively for "sked" QSOs where a random reply might be unwelcome, the letters can be made up from the last four letters in the station call sign.

To accommodate any slow drift in timing between the two stations after initial contact, the slave station monitors the timing of the data transitions received from the master. If these tend to drift away from the optimum point, i.e., half way between the adjacent sampling instants, then the local clock is shifted to correct this. Thus, the slave timing follows exactly that of the master. The master uses the same technique to make sure it is sampling the signal from the slave at the optimum instants.

Resynchronization

The drift correction is very slow in action. As a result, it is not easily disturbed by short periods of interference. However, if contact is lost completely for some time, then both stations must reestablish the correct timing. This is done by operator intervention and restarting the contact as if commencing a new QSO. When both stations have been receiving errors or requests for repeat for 32 blocks, then they both will automatically drop back to the synchronization procedure, with the sending station retaining any unsent message in a buffer. A remarkable feature of the system is that it remembers which station was sending before the interruption, and when back in sync again, a change of direction is made automatically if required. The remainder of the interrupted message is then sent without gaps or errors.

Timing Considerations

CCIR Recommendation 476 specifies the block repetition rate at 2.222 per second and the data rate within bursts at 100 bits per second. Thus, a block of three characters takes 210 ms and a control code 70 ms, leaving 170 ms in which neither station is transmitting. At first it might seem like a good idea to allow the biggest margin of time for delays in antenna changeover relays, and to arrange the slave station to reply 85 ms after the end of the master's transmission. The effects of distance between the two stations, however, cannot be ignored. This is particularly so for intercontinental QSOs. The velocity of radio waves is 186.4 miles/ms (300 km/ms). As a result, the slave station will receive a delayed signal from the master, and the resultant reply will be received late at the master station by 2 ms for every 186.4 miles (300 km) separating the two stations. Thus, to make sure that this slave reply is not obliterated by the next master transmission on long-distance QSOs, the slave must reply as soon as possible after receiving the signals from the master. With practical equipment, and taking into account delays through various filters in the equipment, it looks as though 12,400 miles (20,000 km) is about the maximum range for Amtor to function successfully. In other words, it will just about cover the world on hf, at least by short path, but rules out some satellite possibilities and moonbounce.

Amtor in Practice

Is Amtor really practicable for radio amateurs? From our experience in the UK, the answer is a definite "yes." Many stations in the UK have Amtor in operation using a program written for 6800-based μP machines. A special-purpose unit has also been designed that is essentially a small μP system which will allow any station furnished with conventional RTTY equipment to extend operating capabilities to Amtor. No specialized μP know-how is needed to construct this unit. It is available in the UK in kit form for £76 (about $170). Most stations have found that their existing equipment will change over from transmit to receive and vice versa in less than 10 ms. Only minor modifications have been needed in other equipment. If anything, performance has been better than Table 2 suggests. In one recent QSO where a comparison was made between conventional RTTY and Amtor, with hard copy from both ends to check the errors, G3PLX and G3RSP/MM, working with 50 W erp over a 6200-mile (10,000-km) path on 20 meters, conventional RTTY was producing barely 20% copy while Amtor showed an impressive 99.3% copy, although slowed down by QRM to 25 wpm. Amtor has also been used on nonoptical vhf paths to send such sensitive data as μP machine-code instructions for updating the Amtor program itself as the project developed.

Conclusion

I believe that the Amtor system described in this article is ideally suited to Amateur Radio operation. My hope is that radio amateurs in other countries will join those of us in the UK who have been using this mode. In spite of its complexity, Amtor can be implemented using modern microprocessor techniques which have become available recently. Readers interested in further information on the μP program flow chart are directed to Ref. 1, while further information on the special-purpose unit mentioned will be made available in Ref. 2. Microprocessor enthusiasts with 6800-based machines are invited to contact the writer for further details of the software that is available. QST

References

[1]"Amtor, An Improved RTTY System Using a Microprocessor," *Radio Communication* (RSGB), August 1979.

[2]"Amtor, The Easy Way," *Radio Communication*, (to be published).