ASCII, BAUDOT

AND THE

RADIO AMATEUR

George W. Henry, Jr.
K9GWT

HAL COMMUNICATIONS CORP.

BOX 365

URBANA, ILLINOIS 61801

ASCII, BAUDOT, AND THE RADIO AMATEUR

The 1970's have brought a revolution to amateur radio RTTY equipment and techniques, the latest being the addition of the ASCII computer code. Effective March 17, 1980, radio amateurs in the United States have been authorized by the FCC to use the American Standard Code for Information Interchange (ASCII) as well as the older "Baudot" code for RTTY communications. This paper discusses the differences between the two codes, provides some definitions for RTTY terms, and examines the various interfacing standards used with ASCII and Baudot terminals.

Construction of RTTY Codes

Mark and Space:

Newcomers to amateur radio RTTY soon discover a whole new set of terms, unique to RTTY equipment. Chief amoung these are the words *MARK* and *SPACE*. To be fair, these terms really are not unique to RTTY since they originated with land-line telegraph service before 1900. However, current usage associates *mark* and *space* with something RTTY'ers do to or with machines.

The terms *mark* and *space* date from early pen and moving paper strip recording of telegraph signals. The pen was solenoid operated so that the pen was lowered when the sending key was down, *mark*ing the moving strip; key-up time was represented by the blank *space* between *marks*. Of course, an operator familiar with the Morse code would then have to read the tape. One "tale" has it that the strip recording came first in telegraph operations and was used until the operators discovered that they could mentally decipher the code by listening to the rhythm of the mechanical sounder. Some strip-pen telegraph recorders are still to be found in use to this day.

Teleprinter machines also use solenoids (called *Selector Magnets*) that open and close in response to a signal current (*Loop Current*). Following the convention of telegraph recording, the current-on condition of the signal circuit, or *loop*, is called the *mark state* of the TTY signal and the current-off condition is called *space*. However, the TTY codes differ from telegraph codes in that variable length mark or space conditions (dot, dash, and spaces between them in telegraph codes) are not used to form the characters. Rather, the TTY codes use equal *time length* pulses which can be set to either mark or space. Letters, numbers, and symbols are encoded by different combinations of mark or space pulses.

Start, Data, Stop Pulses:

In a teleprinter machine, the normal "rest" condition of the selector magnet solenoids is with loop current *on*. Interruption of the loop current releases the selector magnet, allowing rotation of a cam in the machine. Transmission of a TTY character begins with a space pulse (current off), called the *start* pulse. The start pulse signals to the machine that reception of a character has begun. Immediately after the start pulse, a series of *data* pulses are transmitted with mark or space condition as indicated by the encoding for the desired character. The number of data pulses used to represent the letters, numbers, and symbols varies with the TTY code being used; "Baudot" code uses five data pulses, ASCII uses eight. Immediately after the last data pulse, a *stop* pulse is included which is always a mark pulse. The stop pulse therefore always occurs a fixed time after the start pulse (after 5 data pulses in Baudot and 8 in ASCII) and gives the machine a "rest time" to prepare for the beginning of the next character, maintaining receive machine synchronization with the transmitted signal. The time length of the start and each data pulse are the same and are often called the *unit pulse* or *select pulse* time. The stop pulse length varies from code to code and even with speeds within a code as will be explained later. In general, the minimum stop pulse length can be one to two times as long as the unit pulse time; stop pulses may be as long as desired since the machine is "at rest" until the next start pulse is received. This type of TTY code that uses start, data, and stop pulses in the construction of each character is called an *asynchronous* or *start-stop serial code*. Other codes also in commercial use include *synchronous serial codes*, in which start and stop pulses are not attached to the data pulses for each character and *parallel data codes* in which each data pulse is assigned a separate wire to and from the terminal device. Such codes are found in common use with computer and line printer devices. Radio amateurs in the United States are currently authorized to use either the Baudot or ASCII serial asynchronous TTY codes.

The Baudot TTY Code

One of the first data codes used with mechanical printing machines uses a total of five data pulses to represent the alphabet, numerals, and symbols. This code is commonly called the *Baudot* or *Murray* telegraph code after the work done by these two pioneers. Although commonly called the *Baudot* code in the United States, a similar code is usually called the *Murray* code in other parts of the world and is formally defined as the *International Telegraphic Alphabet No. 2 Baudot Code* in part 97.69 of the FCC Rules and Regulations. This standard defines the codes for **letters, numerals, and the** slant or fraction bar but allows variations in the choice of code combinations for punctuation. U.S. amateurs have generally adopted a version of the so-called "Military Standard" code arrangement for punctuation, due largely to the ready availability of military surplus machines in the post 1945 years. Amateurs in other countries (particularly in Europe) have standardized on the *CCITT No. 2* code arrangement which is similar to the U.S. standard, but has minor symbol and code arrangement differences.

Since each of the five data pulses can be in either a mark or space condition (two possible states per pulse), a total of $2x2x2x2x2 = 2^5 = 32$ different code combinations are possible. Since it is necessary to provide transmission of all 26 letters, 10 numerals, plus punctuation, the 32 code combinations are not sufficient. This problem is solved by using the codes twice; once in the *Letters (LTRS)* case and again in the *Figures (FIGS)* case. Two special characters, LTRS and FIGS, are used to indicate to the printer whether the following characters will be of the letters or figures case. The printer has a latching mechanism that "remembers" or stores the last received LTRS or FIGS character so that it remains in the last received case until changed. Control operations such as LTRS, FIGS, Carriage Return (CR), Line Feed (LF), Space Bar (SP), and Blank (BLNK = no print or carriage movement) are assigned to both the LTRS and FIGS case so that they can be sent in either case. The remaining 26 code combinations have different letter or numeral/symbol meanings depending upon whether preceded by a LTRS or FIGS character.

Keyboards on Baudot machines such as the Teletype Corporation Models 15 and 28 differ from standard typewriter keyboards, having only three rows of keys with the related letter and number/symbol on each keytop (Q and 1, K and (, etc). The typist soon discovers this difference! Newer electronic terminals such as the HAL DS2000 and DS3100 have standard keyboard arrangements and automatically insert LTRS or FIGS characters as they are needed. The Baudot code itself is restricted to upper case letters only since insufficient codes are available to represent lower case letters.

The Baudot code has seen extensive commercial use throughout the world and is still actively utilized for international wire communications, press, and weather. Due to the ready availability of Baudot mechanical equipment, this code will continue to be quite popular amoung radio amateurs. However, the lack of code space for control, extended punctuation, or lower case letters is a severe limitation of the 5-unit Baudot code. These limitations are particularly inconvenient in computer terminal applications. Various serial and parallel data coding schemes have been used with computers, but the ASCII data code is by far the most common. Figure 1 shows a time diagram of typical Baudot characters and the Baudot data code is shown for both the U.S. and CCITT #2 in Table 1. Notice that the waveform drawing of Figure 1 shows the *current* waveform with mark represented by the upper deflection: If the same data were observed on a RS232 data line, mark would be represented by a downward or *negative* deflection. Also, the bits in Figure 1 are arranged in a left-to-right order as would be observed on an oscilloscope. The bits in Table 1, however, are arranged in *descending* order (b_5 to b_1), conforming to the standard binary representation. Thus, the letter "D" shown in Figure 1 would be written as the binary character "01001".

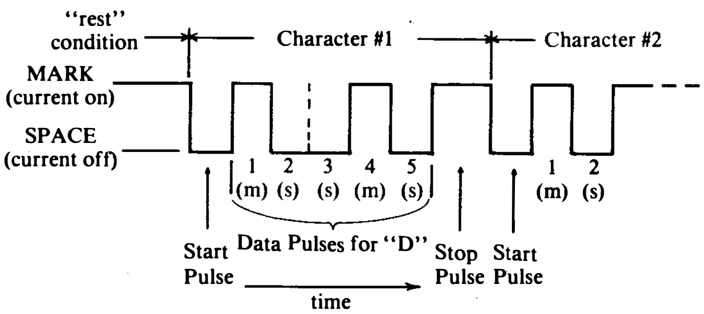


Figure 1. Time Sequence of Typical Baudot Character

Table 1, The Baudot Data Code

Bit Number 54321	Letters	U.S. Figures	CCITT#2 Figures
00000	BLANK	BLANK	BLANK
00001	E	3	3
00010	LF	LF	LF
00011	A	-	-
00100	SPACE	SPACE	SPACE
00101	S	BELL	'
00110	I	8	8
00111	U	7	7
01000	CR	CR	CR
01001	D	$	WRU
01010	R	4	4
01011	J	'	BELL
01100	N	,	,
01101	F	!	!
01110	C	:	:
01111	K	(	(
10000	T	5	5
10001	Z	"	+
10010	L	)	)
10011	W	2	2
10100	H	#	£
10101	Y	6	6
10110	P	Ø	Ø
10111	Q	1	1
11000	O	9	9
11001	B	?	?
11010	G	&	&
11011	FIGS	FIGS	FIGS
11100	M	.	.
11101	X	/	/
11110	V	;	=
11111	LTRS	LTRS	LTRS

Note: FIGS-H (10100) may also be used for MOTOR STOP function.

"1" = Mark = Hole in punched tape

The ASCII Code

In 1968, the American National Standards, Inc. (ANSI) adopted the *American National Standard Code for Information Interchange* (ASCII - commonly pronounced "as - key"), ANSI Standard X3.4-1968. This code uses seven data pulses to specify the letter, number, symbol, or *control operation* desired. An eighth data pulse is provided for optional error checking, called the *parity* bit. As with the Baudot code, the standard ASCII code approved for U.S. amateur use is asynchronous and serial with both start and stop pulses.

Whereas the five-unit "Baudot" code was arranged by Murray so that the most frequently used letters are represented by the least number of mark holes punched in paper tape, the ASCII code has been arranged to optimize computer applications. The code has been particularly designed for rapid collation of alphanumeric lists, one data bit difference between upper and lower case letters, and isolation of all control operations from printing operations. A time diagram of a typical ASCII character is shown in Figure 2. Table 2 shows the ASCII Data Code. As noted for the Baudot waveform drawing, Figure 2 shows the loop current with mark represented by the upward deflection; an oscilloscope trace of a RS232 data circuit would be inverted with mark as the more negative deflection. Also, the bits in Table 2 are arranged in binary number order (b7 - b1). Thus, the letter "S" in Figure 2 would be written as the binary number "0101 0011", with the eighth, parity bit, set to space (0).

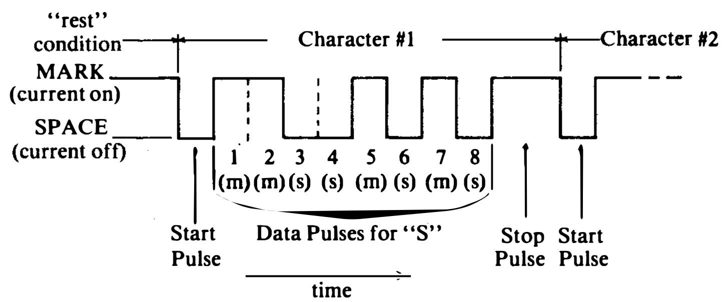


Figure 2. Time Sequence of Typical ASCII Character

Table 2, The ASCII Data Code

7	0	0	0	0	1	1	1	1
6	0	0	1	1	0	0	1	1
5	0	1	0	1	0	1	0	1
4321								
0000	NUL	DLE	SPC	Ø	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

ACK = acknowledge
BEL = signal bell
BS = backspace (←)
CAN = cancel
CR = carriage return
DC1 = device control 1
DC2 = device control 2
DC3 = device control 3
DC4 = device control 4
DEL = (delete)
DLE = data link escape
ENQ = enquiry (WRU)
EM = end of medium
EOT = end of trans.
ESC = escape
ETB = end of block
ETX = end of text

FF = form feed (home)
FS = file separator
GS = group separator
HT = horizontal tab (→)
LF = line feed (↓)
NAK = not acknowledge
NUL = null
RS = record separator
SI = shift in
SO = shift out
SOH = start of heading
SPC = space
STX = start of text
SUB = substitute
SYN = synchronous idle
US = unit separator
VT = vertical tab (↑)

Note: "1" = Mark = Hole in punched tape

As can be seen from the code table, many more punctuation symbols are included in the ASCII code (compared to Baudot). Also, the ASCII code includes a large number of control characters designed for print control of the terminal itself, formatting of data to the computer, and control of other hardware devices by the terminal. Although these control functions are defined by the ANSI definition, variations in the use of the control characters abound in the differing commercial applications.

The keyboards of both mechanical and electronic ASCII terminals are arranged similar to the "standard" typewriter keyboard, thus minimizing any operator re-training required when transitioning from a typewriter to a terminal. The "extra" ASCII keys are arranged around the periphery of the standard keyset if they are provided by the terminal.

A common abbreviation of the full 128 character ASCII code restricts the alphabetic letters to upper case only, often called *CAPS-LOCK* or *CAPLK*. In general, these terminals transmit the upper case ASCII code for a letter whether the SHIFT key is used or not; they may or may not be capable of transmitting all of the control codes. These terminals usually print (or display) the upper case letter when either the upper or lower case letter ASCII code is received. The Teletype Model 33 is an example of a popular upper case only ASCII terminal. Other terminals, such the Teletype Model 43 or the HAL DS3100 ASR, have user selectable upper/lower case or upper case only (CAPSLK) transmit/receive features.

The optional eighth data bit may be set to four conditions: (1) always mark, (2) always space, (3) odd parity, or (4) even parity. All four choices are in common usage. Simple, non-error detecting terminals usually set the eighth bit to be always a mark or space (usually space). Parity is sometimes used with computer and data interconnections where error detection is desired. When used, the parity bit is controlled so as to set the total number of mark data bits in the ASCII character to be always even or odd (even or odd parity). For example, if odd parity is used with the ASCII character "C" (first seven bits = 100 0011), the eight parity bit will be set to space to give an odd number (3) of data bits (0100 0011). Conversely, the odd parity eight bit code for the letter "B" would be 1100 0010. (Logic convention has it that lowest order bits are placed to the right; thus the bit order in the binary representation is "8765 4321".) Upon reception, the receiving terminal simply counts the number of mark pulses in each ASCII 8 bit character. If an odd number is counted, it is ***assumed*** that no errors occurred. Notice, however, that even if a bit error is detected, there is insufficient data to determine which bit was wrong and therefore no error ***correction*** is provided by the parity check itself. Also, if there are two bit errors in the same ASCII character, the parity count will still be odd and no error indication is given even though two occurred. Thus, parity checking will not give complete error detection and does not provide for error correction. Some applications require more sophisticated error detection and correction schemes. Even parity works in a similar manner except that the eighth bit is chosen to make the total number of mark pulses even rather than odd. The U.S. amateur regulations do not specify a requirement for use of the eighth data bit; it may be set to mark, space, odd, or even parity, depending upon the preference of the operator and the capability of his equipment. Relatively simple terminals do not provide parity options; more sophisticated equipments such as the DS3100 ASR do.

Speeds and Baud Rates

The transmission rate of Baudot TTY signals is usually specified in words-per-minute, much like that used for telegraph codes. Actually, the speed is given in the ***approximate*** number of five letter plus space combinations transmitted in a ***continuous sequence*** of start-stop characters in a one minute interval. Convenient choices of gear ratios and motor shaft rpm resulted in the use of non-integer wpm rates. Common usage, however has rounded the exact speeds to easily remembered numbers. Thus, "60 Speed" Baudot is actually sent at 61.33 wpm and "75 Speed" is really 76.67 wpm.

A major problem occurs with the use of words-per-minute (wpm) as a TTY speed specification because of the varying length of stop pulses in use. For example, "60 Speed" Baudot TTY has 22 ms (millisecond) long start and data pulses and a 31 ms stop pulse; the Western Union "65 Speed" also has 22 ms start and data pulse, but the stop pulse is also 22 ms long; electronic terminals commonly use 22 ms start and data pulses and 33 ms stop pulses (1.5 times the data pulse width). ALL of these three codes are compatible and can be received on the same printer or terminal since the stop pulse length is a ***minimum*** time. The common factor between these codes is the 22 ms length of the data, or ***unit*** pulse. Therefore, a new data rate specification has been adopted, the ***Baud*** rate; in this case, the reciprocal of the data or unit or select pulse width:

$$\text{Baud rate} = \frac{1}{t};\ t = \text{length of unit pulse.}$$

Using this definition, ALL three of the above codes have a data rate of 45.45 baud, commonly abbreviated to "45 baud".

As noted above, the length of the stop pulse varies between codes, being from 1.0 to 2.0 times as long as the unit (or data) pulse; multipliers of 1.0, 1.42, and 1.5 are commonly used with the Baudot codes. Standard Baudot data rates and speeds are shown below in Table 3.

Table 3. Baudot Data Rates and Speeds

Baud Rate	Data Pulse (ms)	Stop Pulse (ms)	WPM	Common Name
45.45	22.0	22.0	65.00	Western Union
	22.0	31.0	61.33	"60 Speed"
	22.0	33.0	60.61	45 baud
50.00	20.0	30.0	66.67	European; 50 baud
56.92	17.57	25.00	76.68	"75 Speed"
	17.57	26.36	75.89	57 baud
74.20	13.47	19.18	100.00	"100 Speed"
	13.47	20.21	98.98	74 baud
100.0	10.00	15.00	133.33	100 baud

U.S. Amateurs are authorized to use all of the above Baudot data rates with the exception of 100 baud. This rate has seen limited commercial use in Europe. The 45 baud data rate is by far the most popular world-wide amateur data rate. A limited amount of amateur use of 74 baud ("100 Speed") has been noted on the high frequency bands. Most commercial RTTY transmissions on high frequencies use 50, 57, and 74 baud, with little 45 baud activity.

ASCII data rates are commonly specified as a baud rate, although a character-per-second (cps) or words-per-minute (wpm) may also be sometimes given. The lowest standard ASCII data rate in common usage is 110 baud. ASCII characters sent at 110 baud are usually sent with a 2 unit wide stop pulse, although the one unit stop pulse may also be found in some applications. Above 110 baud, it is common to make the stop pulse one unit pulse in length. The standard ASCII data rates commonly used with asychronous serial transmission are shown in Table 4 below:

Table 4. ASCII Data Rates

Baud Rate	Data Pulse (ms)	Stop Pulse (ms)	CPS (1)	WPM (2)
110	9.091	9.091	11.0	110
	9.091	18.182	10.0	100
150	6.667	6.667	15.0	150
300	3.333	3.333	30.0	300
600	1.667	1.667	60.0	600
1200	0.8333	0.8333	120	1200
1800	0.5556	0.5556	180	1800
2400	0.4167	0.4167	240	2400
4800	0.2083	0.2083	480	4800
9600	0.1041	0.1041	960	9600
19200	0.0520	0.0520	1920	19200

(1): $\text{CPS} = \text{Characters-per-second} = \dfrac{1}{\text{START} + 8(\text{DATA}) + \text{STOP}}$

(2): $\text{WPM} = \text{Words-per-minute} = \dfrac{\text{CPS}}{6} \times 60$

= number of 5 letter plus space groups per minute.

The ASCII data rates up to 300 baud are authorized for U.S. amateur use on frequencies between 3.500 and 21.250 MHz. Data rates up to 1200 baud are permitted between 28 and 225 Mhz; up to 19,600 baud may be used above 420 MHz. The 110 baud rate is by far the most practical for 3.5 to 21.5 MHz use, again due to the ready availability of equipment as well as to the increased susceptibility of the higher data rates to noise, static, interference, etc. VHF-FM amateur activity finds 110 and 300 baud useful for terminal-to-terminal communications and 300 and 1200 baud for computer related activities such as exchanging programs, etc. The very high data rates (1800, 2400, 4800, and 9600 baud) really find their best application in high speed computer-terminal data links. The 150 and 600 baud rates are recognized ANSI and EIA standards, but have seen limited use to date. Some "home-computer" systems are also using 250, 500, and 1100 baud for cassette interfaces, not necessarily with ASCII encoding of the data. The FCC regulations (Part 97.69) specify *maximum* baud rates for each frequency range, but do not require use of the standard rates. Therefore, "non-standard" baud rates may be used with ASCII encoding, but this may not be practical due to lack of compatibility with existing terminal equipment.

Terminal Interfacing Standards

Loop Circuit:

The first commercial applications of teleprinter machines used direct telegraph wire connections between machines. A simple series circuit was used to connect the two machines and a dc power supply. This simple *loop* circuit is shown in Figure 3 below. This connection is also called a *neutral loop* circuit.

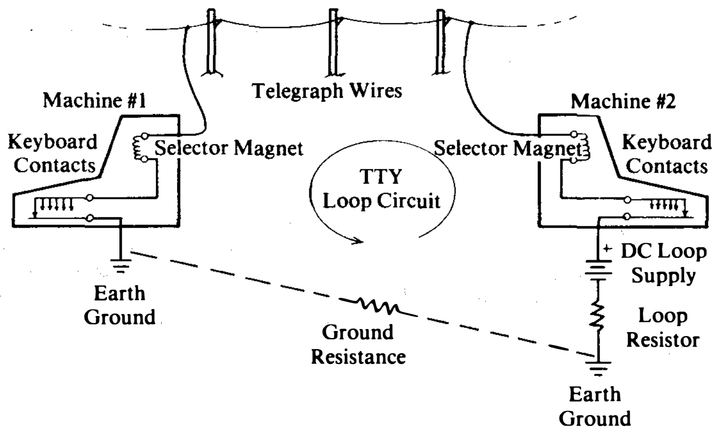


Figure 3. Simple TTY Loop Circuit

As discussed earlier, the printing mechanisms use solenoids or selector magnets to sense the presence (mark) or absence (space) of the loop current. The letters typed on the sending keyboard are encoded with proper mark and space pulses by mechanically driven keyboard contacts. Since the keyboards and selector magnets of both machines are series connected, text typed on one keyboard is reproduced on *both* printers. Connection of the keyboard directly to its associated printer is called a *local loop* and results in what is called *Half Duplex (HDX)*, giving local copy of transmitted test, termed *local echo*. The two machines could be connected with two signal circuits so that keyboard #1 is connected through a loop supply and wire to the selector magnets of printer #2; keyboard #2 is then connected to printer #1 with a second loop circuit. This type of connection is called *Full Duplex (FDX)*. Note that the full duplex connection uses two signal, or loop circuits and does not provide local reproduction or echo of transmitted text. Most radio amateur connection of teleprinters is in the half duplex configuration. The full duplex connection is commonly used with computer terminals. The computer itself may then supply a reproduction of the transmitted text to the printer, but in this case, the *remote* or *computer generated echo* provides confirmation that typed text has been properly processed in the computer.

The battery or dc loop supply in Figure 3 is used to maintain the current in the selector magnets during mark pulses. The actual current in the loop is adjusted with the series *loop resistor*. Selector magnets have been designed for mark loop currents of 60 or 20 mA dc with 60 mA being by far the most common for older machines such as the Model 15 or 28. Newer Baudot machines and most ASCII machines and terminals use electronic interface circuits that accept a wide range of loop currents (10 to 120 mA for the HAL DS3100, for example); a 20 mA loop current is quite commonly used with ASCII terminals.

Since the dc resistance of the machine selector magnets is rather low (100 to 300 ohms, typically), it would at first seem that a low voltage loop supply could be used. However, the *inductance* of the magnet is usually quite high (of the order of 4 Hy for a Model 15), causing a delay in the current rise time. This in turn delays the selector magnet's response to a mark pulse, distorting the signal. This distortion can be severe enough to cause misprinting of received text, particularly if other forms of distortion are present (such as caused by variations in the radio signal). The effect of this inductive distortion is reduced considerably if the L/R ratio (L is solenoid inductance and R is total loop resistance) is reduced by increasing R. Increasing R requires that the dc voltage be increased to maintain the required 60 mA loop current. In general, the higher the loop voltage and loop resistance used, the lower the distortion. In practice, loop power supply voltages between 100 and 300 vdc are common. 130 and 260 volt supplies were often used with Model 15, 19, and 28 Teletype machines. Modern TTY systems use a 150 to 200 volt loop power supply and a 2000 to 3000 ohm loop resistance to set the 60 mA loop current. Because of the related keying circuitry, the *demodulator* unit of a good RTTY system usually includes the loop power supply and current limiting resistor.

The simple circuit of Figure 3 has proven impractical for long distance wire use because of the inclusion of ground resistance in the loop circuit; variations in ground resistance due to rain, terrain, etc. cause variations in loop current and therefore in machine performance. Two telegraph wires could be used to provide a totally wire circuit, but at the expense of twice as many telegraph wires per signal circuit. Another approach, commonly used in telegraph circuits involves reversing the current flow in the loop between mark and space pulses. Thus, in an East-West circuit, current flowing to the West might represent mark and current flow to the East represent space. Current polarity sensing devices, called *polar relays* are then used to sense the direction of current flow. The polar relays then key local loops to operate the machines at each end. This considerably complicates the connections at both ends and requires local loop supplies at each station as well as supplies to generate the two polarities of signal current. Properly adjusted polar relays are very sensitive and can give excellent low distortion operation. Fortunately, amateur RTTY operations do not need the polar relay and such devices are best removed from surplus printers when they are encountered; the *RF hash* generated in a polar relay can be considerable!

Virtually all of the commonly available Baudot machines can be used with 60 to 20 mA high voltage loop circuits. Therefore, to maintain compatibility with this existing equipment, use of the high voltage current loop interface is strongly recommended, even if electronic terminals are also used. Well designed Baudot electronic terminals will include a high voltage loop interface circuit.

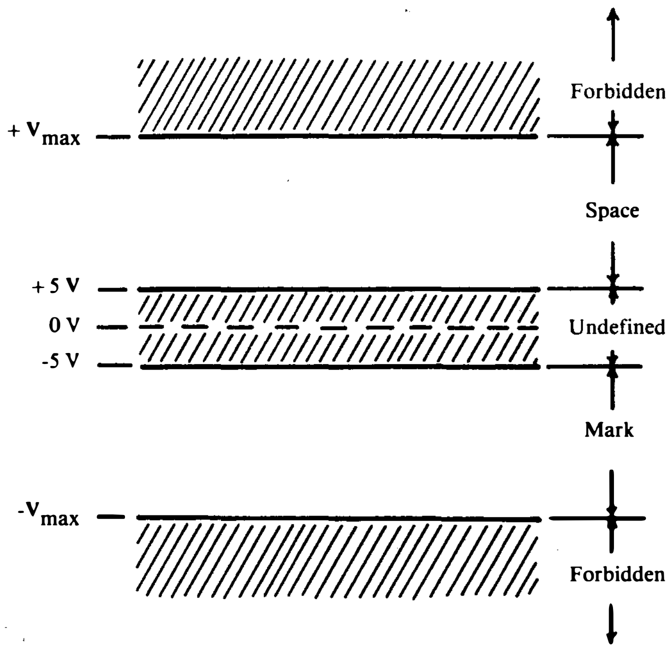
On the other hand, the newer ASCII machines (such as the Models 33, 35, and 43 Teletypes) are available with a wide variety of input/output (I/O) interfaces. These devices usually include a high current, low voltage selector magnet assembly (500 mA, 10-30 volts is typical), an internal *magnet driver* transistor and power supply, and an electronic interface to the data connections. These machines may be supplied with a 20 mA loop interface circuit (or RS232, TTL, or other interface standards).

RS232 Data Interface:

The Electronic Industries Association (EIA) has defined a new standard for interconnection of terminals, modems, and computers, the EIA Standard RS-232-C (August, 1969). This interface standard specifies *voltage* levels for mark and space, rather than the current levels used in a loop circuit. The basic RS232-C voltage ranges are shown in Figure 4.

Note that mark, normally considered to be a logical "1" is represented by a *negative* voltage and the logical "0" by a *positive* voltage. Thus, the RS232 interface standard can be thought of as a *polar voltage* standard. Also, note that RS232 voltage levels really are NOT TTL logic compatible! (In spite of some claims you may read to the contrary!) Users should carefully examine "psuedo RS232 compatible" equipment to make sure no damage will be caused to his RS232 interfaced equipment.

The RS232-C standard also includes definition of control signals to pass between originating and receiving devices and even defines the connector pin assignment. The RS232 connector signal and pin assignment is shown in Table 5.



$R_{source} < 300$ ohms

3000 ohms $< R_{load} <$ 7000 ohms

$C_{load} <$ 2500 pF

$[V_{max}] <$ 25 volts, source open circuited

$<$ 15 volts, source loaded with 3000 ohms

Figure 4. RS-232-C Voltage Standards

Table 5. RS232-C Interface Connector

Pin Number	Circuit	Description	Abbreviation
1	AA	Protective Ground	PG
2	BA	Transmitted Data	TXD
3	BB	Received Data	RXD
4	CA	Request to Send	RTS
5	CB	Clear to Send	CTS
6	CC	Data Set Ready	DSR
7	AB	Signal Ground (Common Return)	SG
8	CF	Received Line Signal Detector	CD
9	--	Reserved for data set testing	
10	--	Reserved for data set testing	
11	--	Unassigned	
12	SCF	Secondary Received Line Sig. Detc.	
13	SCB	Secondary Clear to Send	
14	SBA	Secondary Transmit Data	
15	DB	Transmit Signal Clock	TXC
16	SBB	Secondary Received Data	
17	DD	Receive Signal Clock	RXC
18	--	Unassigned	
19	SCA	Secondary Request to Send	
20	CD	Data Terminal Ready	DTR
21	CG	Signal Quality Indicator	
22	CE	Ring Indicator	RI
23	CH/CI	Data Rate Selector (DTE/DCE Source)	
24	DA	Transmit Signal Clock to Modem	
25	--	Unassigned	

Use of a 25 pin connector is assumed but not defined in the RS232-C standard. Commercial usage has seen the adoption of the TRW-Cinch D-Subminiature connector series (DB25P and DB25S), also manufactured by many other firms (Amphenol 17-10250 and 17-20250, for example). In general, the male *pin* connector (DB25P or 17-20250) is installed on the data terminal, but either polarity may be found on some equipment.

The control signals such as Data Terminal Ready (DTR), Request To Send (RTS), etc. are often called *handshaking* signals; they provide status indicators between data devices (terminal to and from a modem, for example). All control signal voltage and impedance levels also conform to the RS232-C standard shown in Figure 4. Control signals are considered to be *active* or *on* when the voltage is *positive*; a negative voltage control signal is off or inactive. The terms "mark" and "space" are usually not used when describing RS232 control signals. Thus a positive voltage on pin 5 (CB = Clear To Send) is the terminal's signal to the data circuit that the terminal is ready to receive data. Also, an open circuit or no voltage condition on a RS232 signal is interpreted as a "mark" or "active" condition.

Other Data Connection Standards:

A number of other data connection standards may be found in use in commercial equipment. Chief amoung these would be an interface that is compatible with the popular Transistor-Transistor-Logic integrated circuits, a *TTL Compatible* interface. TTL interconnections are particularly useful when directly interfacing computers or other digital devices. Although TTL and RS232 interface standards are not directly compatible, a number of line receiver and line driver integrated circuits are available.

A common data standard used in U.S. military applications is the MIL188 Data Standard. MIL188 is very similar to the RS232 standard with the exception that logic voltages are inverted - mark is represented by positive voltages and space by negative voltages.

Other data interfaces are to be found for CMOS logic integrated circuit connections; interfaces for both +5 volt and +12 volt CMOS operation are in use.

Summary

This paper has discussed the construction of the two common amateur radio teleprinter codes, Baudot and ASCII. The similarities and differences of the two codes have been discussed and a number of terms unique to RTTY have been explained. Data rates and code variations have been presented; standard loop and RS232 data interfacing standards have been compared and discussed. The discussions of this paper have been necessarily brief and the reader is referred to the attached bibliography for additional information on these topics.

Bibliography

American National Standards Institute, Inc.
1430 Broadway
New York, New York 10018

ANSI X3.4 -1977 American National Standard Code for Information Interchange
ANSI X3.15-1976 American National Standard for Bit Sequencing of the American National Standard Code for Information Interchange in Serial-By-Bit Data Transmission
ANSI X3.16-1976 American National Standard Character Structure and Character Parity Sense for Serial-By-Bit Data Communication in the American National Standard Code for Information Interchange
ANSI X3.28-1976 American National Standard Procedures for the Use of Control Characters of American National Standard Code for Information Interchange in Specified Data Communications Links

Electronic Industries Association
EIA Engineering Department
Standards Orders
2001 Eye Street
Washington, D.C. 20006

EIA Standard RS-232-C Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange
EIA Standard RS-269-B Synchronous Signaling Rates for Data Transmission
EIA Standard RS-363 Standard for Specifying Signal Quality for Transmitting and Receiving Data Processing Terminal Equipments Using Serial Data Transmission at the Interface with Non-Synchronous Data Communication Equipment
EIA Standard RS-404 Standard for Start-Stop Signal Quality Between Data Terminal Equipment and Non-Synchronous Data Communication Equipment
EIA Industrial Electronics Bulletin No. 9 Application Notes for EIA Standard RS-232-C

Goacher, D.J., and J.G. Denny; The Teleprinter Handbook, Chapters 1, 2, & 3; Radio Society of Great Britain, 35 Doughty Street, London WC1N 2AE, 1973.

Guentzler, R.G.; RTTY for the Beginner, pp 3-24, RTTY Journal, P.O. Box RY, Cardiff by the Sea, California 92007, 1975.

Kretzman, Byron H.; The New RTTY Handbook, Chapters 1 and 2, Cowan Publishing Corp., 14 Vanderventer Ave., Port Washington, NY, 1962.

Schwartz, Irwin; An RTTY Primer, August, 1977; November, 1977; February, 1978; May, 1978; July, 1978; and August, 1978 issues of CQ Magazine, Cowan Publishing Corp., 14 Vanderventer Ave., Port Washington, NY, 1978.

Tucker, Durward; RTTY from A to Z, Chapters 1, 2, 3, and 11, Cowan Publishing Corp., 14 Vanderventer Ave., Port Washington, NY, 1970.

Reference Data for Radio Engineers, Fifth Edition, Chapter 30, Howard W. Sams & Co., Inc., a subsidiary of International Telephone and Telegraph Corporation, Indianapolis, Kansas City, New York, 1968.

HAL COMMUNICATIONS CORP.
BOX 365
URBANA, ILLINOIS 61801