Why Wait?

Build a *FAST* Cassette Interface

Dr Robert Suding
Research Director for Digital Group Inc
PO Box 6528
Denver CO 80206

This cassette interface does not have a ±30% speed tolerance. The design requires ±12 V and +5 V to run. A good quality recorder must be used, along with excellent quality tapes. Careful adjustments are required.

So why use it? Well, it works! It's dependable. And it's fast. In contrast, the proposed BYTE standard cassette interface runs at 300 Baud. A Teletype paper tape reads @ 110 Baud. I have 24 K on my system. How long would it take me to completely load my system (not including any Bootstrap Loader operations)?

Teletype @ 110 Baud – 40 minutes 58 seconds
Proposed BYTE standard @ 300 Baud – 15 minutes 1 second

The system to be shown in this article has been running for almost a year at 1100 Baud (with an upper limit of 1750 Baud with critical tuning).

Suding system @ 1100 Baud – 4 minutes 6 seconds

Past issues of BYTE have included several articles on cassette interface proposals and circuits. I would suggest re-reading these articles. You will find one common element. Slow. If you get the impression that I'm impatient, you're right. I'll bet you are too. Imagine reading 300 Baud for 15 minutes to discover a noise pulse had destroyed data, requiring re-reading. Ugh!

Thus the proposed standard of the BYTE Kansas City conference in 1975 has a major disadvantage: The use of a redundant Manchester format with a 1200 Hz low frequency critically restricts the user to slower data rates. A related disadvantage for those who use filters or phase lock loops as an input detection method is the fact that the Manchester code employs harmonically related frequencies; this leads to design problems in detectors based upon frequency discrimination techniques.

The system shown in this article avoids the above pitfalls. It uses the non-harmonically related tones of 2125 Hz – Mark and 2975 Hz – Space. The second harmonic of 2125 Hz occurs at 4250 Hz, well down on the passband of a 2975 Hz detector. Sufficient space exists between the two frequencies to allow for reasonable recorder speed discrepancies. The higher frequencies involved permit increasing the data rate.

Several approaches are possible in cassette interfacing, as seen in past BYTE articles. However, their emphasis on wide cassette speed tolerance made them slower. My

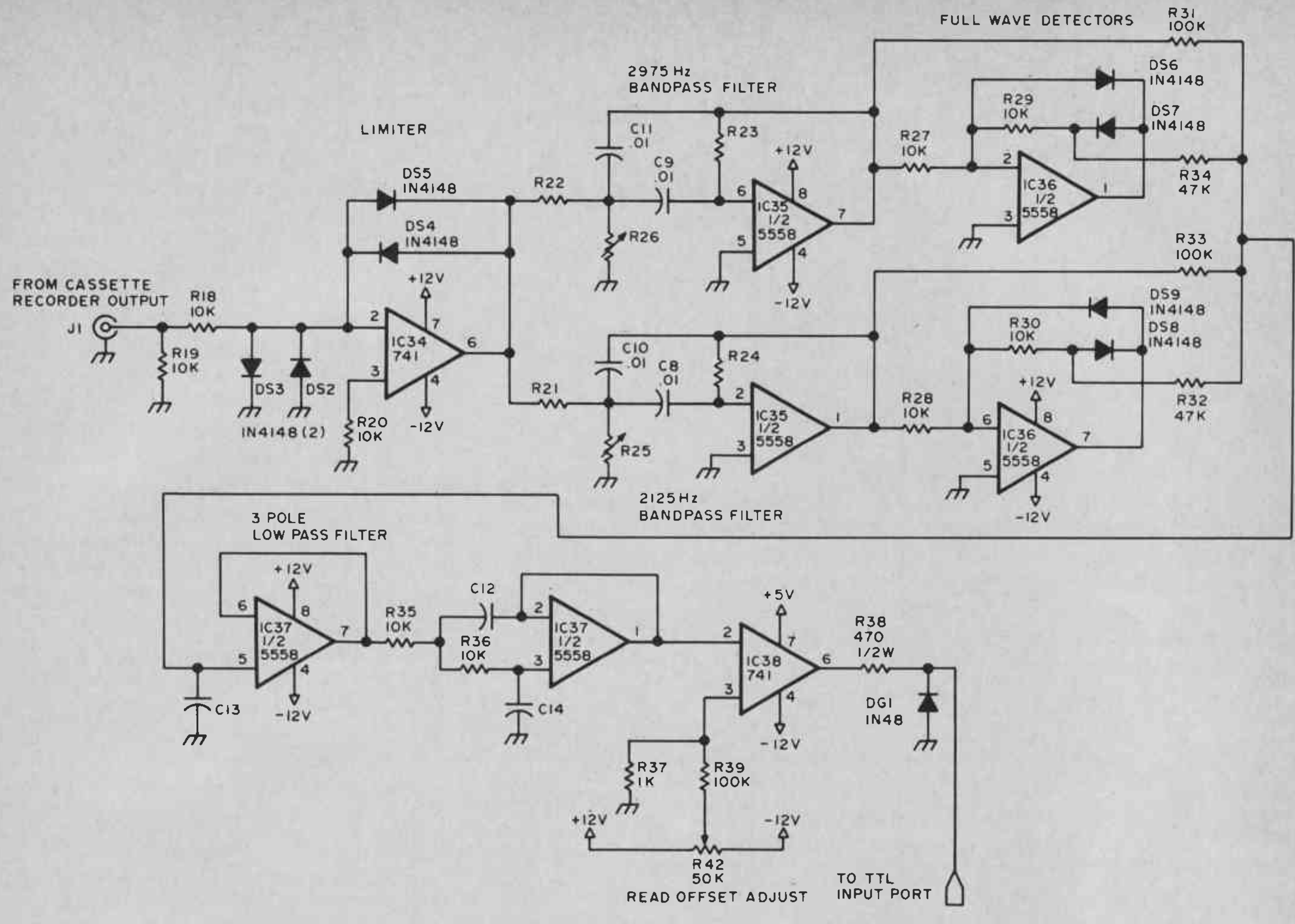

Figure 1: The schematic of the Suding cassette input interface as found in the Digital Group systems. This interface amplifies and clips the cassette output with limiting amplifier IC34, discriminates the two data frequencies (see table 1) with bandpass filters followed by full wave detectors, passes the detected signal through a 3 pole active low pass filter, then converts the result to a TTL level which is read by a single bit input port. One example of software (see listing 1) to drive this input interface uses a programmed simulation of UART input algorithm; an actual UART or ACIA device could be substituted if desired.

approach to "out of specification cassette speed" is — "put it in the specification, or get a good recorder." More of that later.

Theory of Operation

The 1100 Baud Digital Group system uses the circuits of figures 1 and 2. The cassette receive circuitry detects the prerecorded frequency shift keying and produces a "1" or a "0" output as a result of a detected 2125 Hz or 2975 Hz tone at the input. A 741 operational amplifier, IC34, is used as a clamped limiter which prevents variations in cassette amplitude from affecting the detection process. The output of the limiter should be about .6 V peak to peak, roughly a square wave with rounded edges of the incoming frequency, constant in amplitude regardless of tape volume setting or minor tape "dropout" problems.

Two bandpass active filters (IC35) then amplify a tone five times when actually tuned to their respective frequencies of 2975 Hz for the top filter, and 2125 Hz for the lower filter. The further off the tuned frequency the tone is, the less amplification the filter will produce. The gain, bandwidth, and tuned frequency are set by the three resistors and two condensers in each filter. Each filter may be exactly tuned to frequency by carefully setting the variable resistance value (which may be either a potentiometer or selected fixed values).

Full wave active detectors produce rectified full wave pulses at the summing junction, pin 5 of IC37. The 2975 Hz tones are rectified to a positive voltage, and the 2125 Hz tones are rectified to a negative voltage. As received tones depart from either exact frequency, a value less positive or

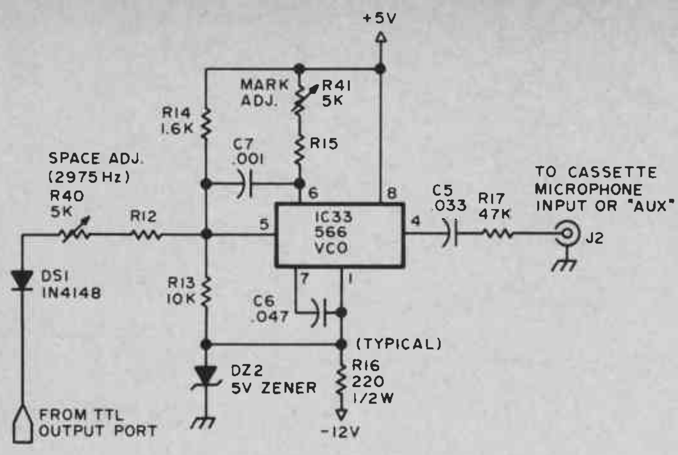

Figure 2: The schematic of the Suding cassette output interface as found in the Digital Group systems. The output interface is a simple audio frequency shift keyer made up of a 566 voltage controlled oscillator with two frequency states controlled by a single TTL data line. The TTL level which drives the output modulator is a single bit derived from an output port. The software (see listing 2) to drive this output interface is shown as a programmed simulation of a UART output algorithm; an actual UART or ACIA device could be substituted if desired.

negative is produced until approximately midway (2550 Hz) a summed voltage of 0 results.

A three pole lowpass active filter then removes the remaining traces of pulsating DC from the summed signal with almost no effect on the data pulses up to a speed of 1000 bits per second. If lower data rates were to be utilized, an improved signal to noise ratio could be obtained by multiplying the values of C12, C13, and C11 by the reciprocal of the data rate ratio. Table 1 shows some component values for alternative frequency designs.

The final receiver section is a 741 operational amplifier, IC38, connected as a slicer. This operational amplifier detects whether the voltage at its pin 2 is positive or negative with respect to the constant voltage at its pin 3. The output voltage will then swing either to nearly −12 V or to nearly +5 V. Notice that this operational amplifier has +5 as its positive supply voltage, pin 7. A forward biased germanium diode prevents the actual output voltage from going less

Tune Up Notes

The cassette interface must be carefully tuned to achieve proper performance. Careless tuning has been the most frequent cause of cassette system failure.

1. Plug in the six integrated circuits of the cassette interface.
2. Connect a calibrated audio oscillator between the limiter input and ground. A digital frequency counter driven by the audio oscillator is highly recommended. The oscillator should cover the desired range of 2 – 3 kHz, with a sine wave output of .5 or so, although the precise level is not at all critical.
3. Apply +5 and ±12 voltages to the circuit. Measure the output at pin 6 of the 741 limiter (IC34) with an oscilloscope. The wave shape should be a rounded square wave of about .6 V peak to peak.
4. Set the audio oscillator to 2125 Hz. Measure the output at pin 1 of the 5558 active bandpass filter. Slowly turn R25 until the signal peaks. Be sure that you are peaking at 2125 Hz, not a harmonic. Vary the oscillator frequency a few decades to insure 2125 Hz is the tuned frequency.
5. Similarly, set the oscillator to 2975 Hz and measure the output at pin 7 of the 5558 (IC35). Slowly turn R26 until the signal peaks. Vary the oscillator to insure a 2975 Hz peak.
6. Measure the detected voltages at pin 5 of IC37. When the oscillator approaches 2125, the voltage should go negative. When approaching 2975, the voltage should go positive. Trouble in this area would most likely be caused by reversed or defective diodes, or shorts between adjacent lines.
7. Measure the voltage at the cathode (bar) end of the output clamping germanium diode (G1). Sweeping the frequency between 2125 and 2975 Hz should result in a clean voltage jump somewhere between 2125 and 2975 Hz. Measure the output swing to insure that it does not exceed +5, –.3 V.
8. Remove the audio oscillator and short input connector J1 temporarily to ground. Measure the output at pin 6 of IC34. A stable condition (no oscillation) should be seen. Connect the oscilloscope to the cathode of G1 again. Adjust the balance potentiometer (R42) so that the output voltage is a negative level. Slowly turn the potentiometer until the output voltage jumps to a positive level and leave the setting at this point.
9. Disconnect the temporary jumper from the input connector and reconnect the audio oscillator. Perform step 7 again. The crossover threshold should be close to 2550 now. If all proceeds well at this point, the cassette interface is ready to receive data.
10. Connect the oscilloscope to pin 4 of the 566 voltage controlled oscillator (IC33). A triangular wave output should be seen.
11. Connect a temporary jumper between the TTL input going to DS1 and +5 V. Connect a frequency counter to pin 3 of the VCO (IC33). Adjust potentiometer R41 for a resultant output frequency of 2125 Hz.
12. Remove the jumper from +5 V and connect the jumper from DS1's input to ground. This time adjust R40 for 2975 Hz output.
13. Remove the jumpers, and you are ready for final tune in the driving circuit. Connect the cassette interface to the driving output port, and program the driving processor to send a TTL high level ("1") output to the cassette interface. Adjust R41 to 2125 Hz. Then have the processor send a "0" level. This time adjust R40 for 2975 Hz output. The cassette interface is now ready for use.

than ≃ −.2 V, so that valid TTL levels are not exceeded. An offset adjusting potentiometer allows the output to be placed in a "Mark Hold" condition when no tone input is being detected.

The cassette recording section (figure 2) uses a single integrated circuit, a 566 voltage controlled oscillator, IC33. A logic level from the computer's output port controls the resultant audio frequency output to the cassette recorder microphone input. A high input ("1") produces a 2125 Hz output, and a low input ("0") results in 2975 Hz. The output wave shape is a symmetrical triangular wave. Should the user object to using a triangular wave, a more nearly sine wave can be obtained by connecting a pair of back to back 1N914 diodes between ground and the output side of the coupling capacitor C5.

Exact values and high quality components will result in a trouble-free voltage controlled oscillator. The 47 K (R17) resistor in series with the output is a typical value to be used when coupling to the low level, low impedence external microphone inputs of most cassette recorders. Using the "AUX" input of your cassette recorder generally gives better results.

Construction

The cassette interface is available as a part of a printed circuit board kit from the Digital Group. The printed circuit board is shared by a television display circuit to be described in the next article in this series. A kit of the cassette interface only is also available from the Digital Group for $30, which includes all parts and the printed circuit board. The experienced builder can build the circuit in an evening or two by hand wiring components on standard .1 inch grid Vectorboard. All the circuitry can be contained in an area of approximately 3 inch by 5 inch (about 8 cm by 13 cm).

Be sure to use only high quality components, particularly in the active bandpass filters and voltage controlled oscillator. Some strange "frequency jump" problems have been traced to surplus 566s which were temperature sensitive. Lay out the receive circuit to avoid feedback paths from output to input, particularly in the limiter, active bandpass filters, and slicer areas. Different op amps could be used, but may result in instability or degradation of final performance due to suboptimization.

Modifying Your Cassette Recorder

It is very helpful to listen to the data from the cassette so that the beginning of the data burst may be detected, as well as

	Low Filter			High Filter			Low Pass Filter			VCO	
	R21	R24	R25 *	R22	R23	R26 *	C13	C12	C14	R12	R15
2125-2975 Hz 1100 Baud	6.8 k	68 k	938	4.7 k	47 k	697	.0056 µF	.01	.015	2.7 k	1.3 k
1200-2400 Hz 300 Baud (Simple)	6.8 k	68 k	4173	4.7 k	47 k	1162	.0056 µF	.01	.015	470 k	2.7 k
1200-2400 Hz 300 Baud (Correct)	12 k	120 k	1668	5.6 k	56 k	906	.015 µF	.033	.047	470 k	2.7 k
2125-2295 100 Baud (Simple)	6.8 k	68 k	938	4.7 k	47 k	1301	.0056 µF	.01	.015	47 k	2.7 k
2125-2295 100 Baud (Correct)	36 k	360 k	156	27 k	270 k	179	.056 µF	.1	.15	47 k	2.7 k

*** means that the value so indicated is the typical calculated value. The precise value is dependent on component tolerance.**

Table 1: Theoretical values of components for alternate frequencies. This table gives values of components to be used with the circuits of figures 1 and 2 in order to make this cassette interface work with several alternate specifications. See the text for a definition of the various comments at the left of the table.

Potential Troubles

Knowing about potential problem areas is a first step to minimization of their effects. Troubles seem to break down into *six* classes.

- Cassette recorders and the cassettes used: A marriage between your $1000 microprocessor and junior's $20 cassette recorder, which has been using 30¢ cassettes for the last five years, will not produce happy offspring! I have been using a Superscope C-104 for the past year, and can report no failures except for defective cassette tapes. The C-104 has several attractive features. Besides the usual conveniences such as index counter, cuing, etc, it has a variable readback speed control, dandy for out of spec cassettes from friends. Inside, another special motor speed control potentiometer is located near the speaker which allows precisely setting the record/write speed. Quality control seems good overall, and the list price of $120 (cheaper at discount stores) is worth the investment. Don't waste your money on cheap cassettes. Sony Low Noise C-45s have been generally good. Some $2 – $4 Data Certified Cassettes are superior, but not needed.
- Microprocessor caused problems: Some microprocessor designs will not work directly with this interface system. This interface was designed to be connected directly to a single bit IO port, with the processor handling all of the bit timings through timing loops. If your processor must periodically catch its breath for such things as dynamic memory refreshing, you may be unable to directly use the "Software UART" system. What a shame! However, a hardware UART will permit using the system even with a system of this nature.
- Cabling problems: It is possible to connect your cassette recorders with the read and write cables reversed. Enough crosstalk from the write line to the read limiter existed to give the appearance of data being read, but so many errors resulted that the programming would not run.
- Tuning problems: Circuit tuning is the most common problem. *Carefully* tune the active filters!
- Cassette Crashes: Cassette damage is frequent on tapes which have always worked before, but now mysteriously fail. The most common cause of this is removing a cassette from the recorder without completely rewinding. The exposed oxide then gets damaged, and is no longer usable.
- Miscellaneous circuit problems:

Defective level output from cassette read limiter.
1. None at all: Check for ±12 V to IC34, and IC34.
2. Too high output level: Diodes (DS4 and DS5) open, or one is reversed.

Bandpass active filters don't filter.
1. Off frequency
2. Bad 5558
3. Check for shorts or out of tolerance condensers C8, C9, C10, or C11. Disk ceramics are a "no-no" in tuned circuits.
4. Resistors improperly wired or inserted.

Full wave detector does not work as described:
1. Diodes open, reversed or shorted.
2. Defective IC36.

Low pass active filter fails to work:
1. Shorted or out of tolerance condensers.
2. Defective IC37.

Output slicer (IC38) fails to produce TTL levels:
1. Reversed, open or not Germanium diode at DG1.
2. Too heavily loaded output. This circuit should drive no more than one TTL load (standard for most IO ports).

VCO won't oscillate.
1. Defective 566 (IC33).
2. Shorted condenser C6.

VCO has parasitic oscillation (high frequency):
1. C7 not connected.
2. Defective 566.
3. C6 is open, producing a very high frequency.

VCO won't tune to frequency or stay there:
1. Out of tolerance or defective C6. You really didn't use a disk ceramic here, did you?
2. Defective 566.
3. Non-TTL levels used to drive VCO.
4. Defective potentiometers R40 or R41.
5. DS1 or DZ2 reversed or defective.

hearing the end of the data. When the cassette read cable is plugged into most cassette recorders' earphone output jack, the speaker output is usually cut off. However, since a closed circuit jack is all that is involved, a quick solution is to connect a jumper on the jack so that the speaker is not disconnected. Even better, use a 100 ohm ¼ watt resistor instead of the jumper, and the data howl won't be so loud. A 10 ohm, ¼ watt resistor from the amplifier lead to jack, to the jack frame will prevent potential damage to the output driving transistor(s).

Alternative Frequencies and Applications

The cassette interface design may be used with the proposed BYTE standard should you so desire. Table 1 has appropriate component values calculated for two alternative possibilities: the simple way (less desirable) and the "right way". The simple way permits using a switch on the bandpass active filters to select the frequency pairs. The right way involves setting the circuit to the optimal values, and using separate interfaces for each frequency pair.

Amateur radio (ham) radioteletype (RTTY) generally uses 2125 – 2295 Hz frequency shift keying for 170 Hz shift. The existing cassette interface can be used by "straddle tuning," but improved performance may be obtained by selecting a second R26 which will tune the high filter to 2295. The cassette read cable may then be attached to the short wave receiver and the microprocessor, programmed as a radioteletype video terminal, which can replace the noisy Teletype machine. Of course, a cassette interface specifically designed for this 170 Hz shift at 100 WPM will give superior performance under marginal conditions.

The cassette interface may be used as a stand alone radioteletype terminal unit and audio frequency shift keying if desired, and works quite nicely in this application.

Software

I would suggest using software for your cassette read and write timings. Sample 8080 software is included as listing 1. Timings at locations <0>/116, <0>/133, <0>/241, and <0>/260 are based on an 8080 system with a 500 ns T time and no wait states. Slower systems will require proportionately decreased loop timings.

A UART could be used instead of the "software UART" system shown. However, several disadvantages arise. First, a slightly greater cost and complexity. More important, however, is a degradation in total

Listing 1: Stand Alone Suding Cassette Input Program. This program is a self contained data transfer routine which will transfer a block of data from cassette to split octal memory locations xxx/xxx through yyy/000. This program assumes that MEMTOCAS (see listing 2) was used to create the tape being read. A more generally useful input facility would be modelled on this program and linked to a system monitor as a subroutine.

Split Octal Address	Octal Code	Label	Op.	Operand	Commentary
<0>/100	041 xxx xxx	CASTOMEM	LXI	H,xxx/xxx	Load starting address in HL pair;
<0>/103	021 010 000	STARTBYT	LXI	D,000/000	Load E, clear D;
<0>/106	333 001	SYNCHLOO	IN	1	Port 1 bit 0 read for input;
<0>/110	346 001		ANI	1	Mask all but bit 0;
<0>/112	302 106 <0>		JNZ	SYNCHLOO	If not start bit then reiterate loop;
*<0>/115	006 300		MVI	B,300	Time delay to middle of first data bit*;
<0>/117	005	WSYNCH	DCR	B	Decrement synch wait count;
<0>/120	302 117 <0>		JNZ	WSYNCH	If not done then keep waiting;
<0>/123	333 001	GETDATA	IN	1	Read port 1 bit 0 again;
<0>/125	346 001		ANI	1	Mask all but bit 0 again;
<0>/127	202		ADD	D	Sum old bits with new bit;
<0>/130	017		RRC		Rotate new and old into next position;
<0>/131	127		MOV	D,A	Save result back in D;
*<0>/132	006 200		MVI	B,200	Time delay between bits;
<0>/134	005	WDATA	DCR	B	Decrement data wait count;
<0>/135	302 134 <0>		JNZ	WDATA	If not done then keep waiting;
<0>/140	035		DCR	E	Decrement data count loaded at 0/103;
<0>/141	302 123 <0>		JNZ	GETDATA	If not done then repeat for next bit;
<0>/144	162		MOV	M,D	Save received data in memory;
<0>/145	043		INX	H	Point to next available location;
<0>/146	174		MOV	A,H	Move high order address to A for end check;
✓<0>/147	376 yyy		CPI	yyy	Has high order address reached end?
<0>/151	302 103 <0>		JNZ	STARTBYT	If not then reiterate for next byte;
<0>/154	166		HLT		End input;

Notes:

- Input is assumed to be wired to bit 0 of port 1, from output of IC38 pin 6 via resistor R38 and shunted by diode DG1.
- Loading proceeds from split octal address xxx/xxx to address yyy/000. Enter this program by jumping to location <0>/100 after setting up constants of address.
- "*" indicates a timing constant for the "software UART" inputs.
- "✓" indicates the end of transfer comparison mentioned in text.
- <0> indicates an arbitrary page location for this program, to be replaced by a real memory page number when actually loading the program at byte 100 of some page.

Listing 2: Stand Alone Suding Cassette Output Program. This program is a self contained data transfer routine which will transfer a block of data from split octal memory locations xxx/xxx through yyy/000 onto cassette tape after a five second leader output delay. This program assumes that CASTOMEM (see listing 1) will be used to read the tape being created. A more generally useful output facility would be modelled on this program and linked to a system monitor as a subroutine.

Split Octal Address	Octal Code	Label	Op.	Operand	Commentary
<0>/200	041 xxx xxx	MEMTOCAS	LXI	H,xxx/xxx	Load starting address in HL pair;
<0>/203	076 001		MVI	A,1	Start port output in high state;
<0>/205	323 001		OUT	1	Send initial state out;
<0>/207	026 012		MVI	D,012	Outer leader delay count;
<0>/211	006 377	LEADER5S	MVI	B,377	Outer leader delay loop return;
<0>/213	016 377	LEADER5X	MVI	C,377	Middle leader delay loop return;
<0>/215	015	LEADER5Y	DCR	C	Inner leader delay loop return;
<0>/216	302 215 <0>		JNZ	LEADER5Y	If inner loop not done then reiterate;
<0>/221	005		DCR	B	Middle leader delay count;
<0>/222	302 213 <0>		JNZ	LEADER5X	If middle loop not done then reiterate;
<0>/225	025		DCR	D	Outer leader delay count;
<0>/226	302 211 <0>		JNZ	LEADER5S	If outer loop not done then reiterate;
		*			
		* Upon reaching this point, 5 seconds of mark (high) state have			
		* been output to the cassette interface.			
		*			
<0>/231	016 011	BYTEOUT	MVI	C,011	Define output bit count (decimal 9);
<0>/233	257		XRA	A	Clear carry (start bit level is 0);
<0>/234	176		MOV	A,M	Move current byte to A;
<0>/235	027		RAL		Rotate bit into position (carry=0 first);
<0>/236	323 001	WNEXBIT	OUT	1	Send current LSB to output port;
*<0>/240	006 200		MVI	B,200	Time delay between bits;
<0>/242	005	WOUTLOOP	DCR	B	Decrement delay count;
<0>/243	302 242 <0>		JNZ	WOUTLOOP	If time left then reiterate;
<0>/246	037		RAR		Rotate new bit into position;
<0>/247	015		DCR	C	Decrement output bit count;
<0>/250	302 236 <0>		JNZ	WNEXBIT	If data left then reiterate;
<0>/253	076 001		MVI	A,001	Stop bit state defined
<0>/255	323 001		OUT	1	then sent out to port;
*<0>/257	006 377		MVI	B,377	Stop bit value set;
<0>/261	005	WIBDELAY	DCR	B	Decrement stop bit counter;
<0>/262	302 261 <0>		JNZ	WIBDELAY	If time left then reiterate;
<0>/265	043		INX	H	Increment memory address;
<0>/266	174		MOV	A,H	Move high order address to A for end check;
✓<0>/267	376 yyy		CPI	yyy	Has high order address reached end?
<0>/271	302 231 <0>		JNZ	BYTEOUT	If not then continue output process;
<0>/274	166		HLT		End output;

Note:

- Output is assumed to be wired from bit 0 of port 1 to DS1 in figure 2.
- See notes to listing 1 for listing conventions.

system flexibility. The "software UART" allows the timing constants to be dynamically modified (if desired) by detecting the variations in the stop bit timing, thereby compensating for wow and flutter. Digital integration of the incoming data bits is possible by setting a register to octal 200 at the beginning of each bit time. During the bit time, repeated sampling either adds or subtracts from the register (depending on whether 1 or 0) and a "branch minus" instruction system effectively eliminates receive problems. This digital integration detection is utilized by the Digital Group Z-80 cassette read software.

Versions of this "software UART" system have been written for 8008, 8080, Z-80, 6502, and 6800. All work satisfactorily.

Operation

This cassette system is utilized by first turning on the cassette recorder and waiting until the lower tone 5 second leader tone is heard. At this point, restart the system to the beginning address of the "Cassette to Memory" software.

Cassette writing is accomplished by restarting the system to the beginning of the "Memory to Cassette" programming. Be sure to set the appropriate start and stop addresses prior to beginning the read or write operations. The monitor programs in the various Digital Group systems automatically set the start and stop addresses. The check marks in the listing (√) indicate the points where start and stop addresses may have to be modified.

The software may be adjusted to run at different data rates by changing the values at the addresses mark with an asterisk (*). Note that the constants at <0>/133 and <0>/241 are the same. The constant at <0>/116 is 50% greater and the constant at <0>/260 is twice the value of the constant at <0>/241.

Summing It Up

This cassette interface represents a simple but fast and dependable way to store programs and data for the serious hobbyist. It does not seek to be all things to all users, but a number of applications can be run using the same basic design. The detail interface design has independence from other components in the system, allowing various processors to use the same cassette system (with appropriate software).■