TI-88 Calculator Peripheral Bus Specification

Consumer Products Group
Calculator Division

April 1981

Revision 1.0

TABLE of CONTENTS

Paragraph Title

SECTION 1

Introduction

1.1 Purpose of This Document

This document describes the interface between the TI-88 programmable calculator and peripheral devices. This document is meant to be used by persons interested in the operation of the peripheral bus for purposes of functional evaluation, peripheral design, product testing, or development of related products.

This document is kept on the Personal Computer development system II as SD1.TOMF.DOC.TI88BUS.

1.2 Scope of This Document

This document only includes information on the functional and software aspects of the interface. Descriptions of electrical parameters such as signal loading and voltage levels are not included.

1.3 Related Documents

The following documents are referenced in the text or may contain other material of interest to the reader of this document. Copies of any of these may be obtained by contacting Tom Ferrio at TI Lubbock; phone 741-3362; MSG PLB.

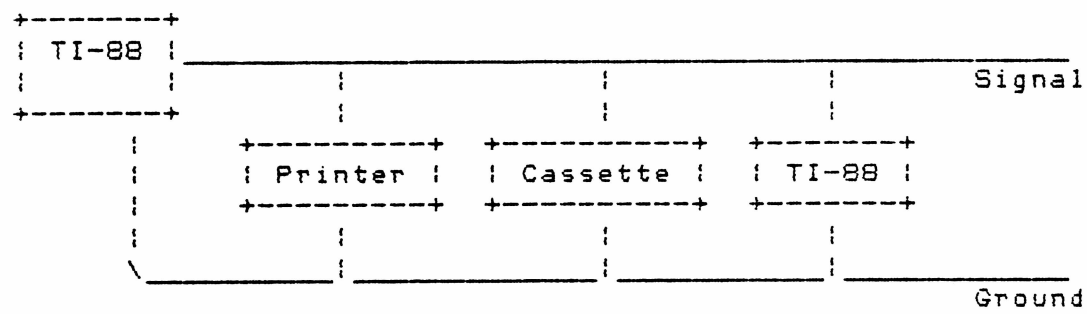
SLIP "Tinman" Specification

SECTION 2

Functional Aspects

2.1 Bus Organization and Connection

The calculator bus is implemented with a two-wire interface. The two wires consist as one bi-directional signal line and one ground line. All devices are attached to these two wires. The following figure illutrates this for some appropriate devices.



As illustrated two TI-88s may be connected to communicate with each other under certain conditions.

2.2 Bus Control

This interface is designed around a master-slave concept. The interface bus will always have one master device which will be a TI-88. If two TI-88s are attached then software must be provided such that one will be the master and the other will be a slave device. Slave devices (peripherals) do not initiate communication except under the idle state as described below. All messages sent by peripherals are processed by the master device and cannot be sent to another peripheral except through the use of special software in the master unit. Normal communications consist of the master requesting an action from a slave device with the slave responding. Deviations from the protocol by two TI-88s acting in the master mode or the

attachment of two identical peripherals could cause bus communications to break down.

2.3 Frames

Data on the bus is transmitted in groups of bits called frames. To initiate an action the master controller will send a frame onto the data bus. From a logical viewpoint the frame will consist of the following three components.

1. Device code.
2. Command code.
3. Data.

2.3.1 Device Code. The device code has values from 0 to 255 (8-bit code) which identify a unique device on the bus. The following table lists device codes in hexidecimal which have been assigned.

90 - Audio cassette interface.
A0 - Thermal printer.

The least significant 4 bits will always be zero for TI-88 peripherals. This provides for future expansion of the communication protocol. The master unit receives and processes all messages sent by peripherals regardless of the device code used in a peripheral transmission.

2.3.2 Command Codes. The command codes have values from 0 to 255 which specify the action desired of the slave device or information returned to the master device. The meanings of the command code values are device dependent; each device has a separate set of command codes as described in section 4.

When the TI-88 receives data it recognizes command codes hexidecimal D0 through DF to indicate that the following data consists of keypushes. The range of command codes is because the least significant four bits are ignored. Frames with command codes outside of this range will be sent to keystroke software for processing if requested.

2.3.3 Data. The data field is always present although sometimes the content is meaningless. At other times it may contain data to be transfered to or from a peripheral or control information.

2.4 Peripheral Response

Certain commands require a response from the selected peripheral. For instance the command to the audio cassette interface to read data requires that the data be returned to the TI-88. The following is a simplified list of the activity on the data bus to handle a request to a peripheral for data.

1. Master device sends a frame commanding data to be read. The data portion of this frame is meaningless but is present.
2. The slave device holds the interface busy while executing the command. This state is optional depending on the speed of executing the command and the software in the master device.
3. The slave device releases the communication line.
4. The master sends an idle pulse.
5. The slave device sends a frame with response data.

SECTION 3

Frame Transmission

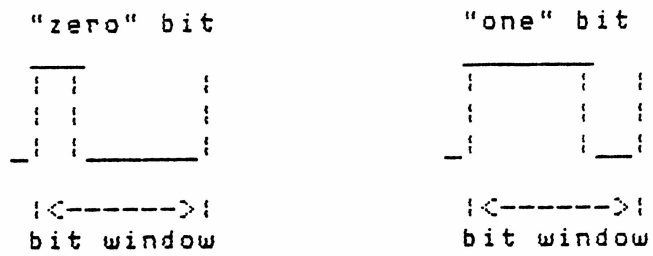
This section describes the transmission of data on the bus in terms of electrical signals. The format of frames and the encoding of data on the signal line are described.

3.1 Signal Timing

Signals on the data bus are broken into a number of time frames called bit windows. The data bus will naturally drop to a low level unless driven high by a device. By controlling the voltage on the signal line with certain timing five different signals are sent on the data bus.

1. Logic high signal.
2. Logic low signal.
3. Zero bit signal.
4. One bit signal.
5. Idle pulse.

Logic high and logic low signals are merely signals which are at the high or low voltage levels respectively for an entire bit window. These are used for synchronization purposes or to indicate a "busy" condition as described below. The zero and one bit signals each have two voltage transitions within the bit window. The first transition will be from low to high voltage and signifies the beginning of the bit window. The second transition will be from high to low voltage. The positon of the second transition within the bit window will indicate whether the bit signal is a zero or a one.



If the downward transition is in the first half of the bit window (nominally one-fourth) then the bit is a zero. If the downward transition is in the second half of the bit window (nominally three-fourths) then the bit is a one.

The idle pulse is used for the master device to control transmissions from slave devices as described below. It consists of at least 4 bit windows of logic low signal followed a narrow logic high pulse.

3.2 Data Rate

The nominal data rate for the interface is 1667 bits (bit windows) per second. This provides a nominal bit window width of 600 micorseconds. The data encoding scheme is self-clocking however, so that accurate data transmission can occur at rates from approximately 1200 to 2000 bits per second. This corresponds to bit window times of 500 to 833 microseconds. The data reception algorithm must allow for this magnitude of variation.

3.3 Frame Format

This section discusses the general format of frames. Following sections discuss how this format relates to transmissions from the master or a slave device.

When discussed above the frame was shown to consist of the device and command codes followed by data. In terms of actual signals on the bus, synchronization control must be added as follows.

1. Frame synchronization.
2. Device code.
3. Command code.
4. Data.
5. Termination.

3.3.1 Synchronization. A frame will not begin unless the signal line is in a logic low state. The synchronization data consists of the following sequence which follows the logic low state.

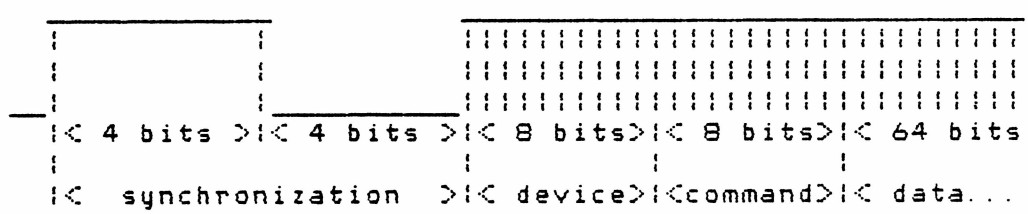
4 bit windows of logic high signal.
4 bit windows of logic low signal.

The status of the signal line preceeding the snychronization data depends on whether the transmission is from the master or a slave as described below.

3.3.2 Device and Command Codes. The 8-bit device and command codes are each sent out with the least significant bit transmitted first.

3.3.3 Data. The data will consist of 64 bits of information and is always present, even if it is meaningless. The 64 data bits will be transmitted in order from the least significant bit to the most significant bit.

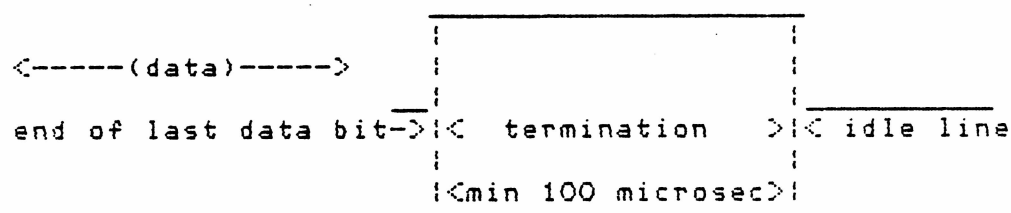
General frame format.



3.3.4 Termination. When the last bit has been transmitted the signal line will be left at a high level (the bit definition

includes a positive-going edge at the end). The block transmission is terminated by returning the signal line to the logic low level. This transition should occur at least 100 microseconds (one-fourth bit window) past the last bit in order to guarantee detection by the receiving device.

Termination.

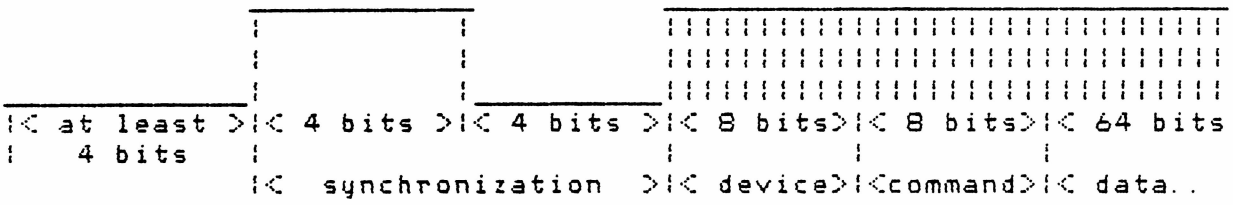


(scale differs from previous figure)

3.4 Transmission from the Master

If the signal line is in a logic-high state then a slave device is indicating a busy status. A logic-low state indicates that the master device may initiate a frame transfer. The master device will not transmit a frame until the signal line is in logic low state for at least 4 bit windows.

Transmission from the master device.



3.5 Transmission from a Slave

A slave device will not transmit a frame until an idle pulse has been received from the master. In this way the master device

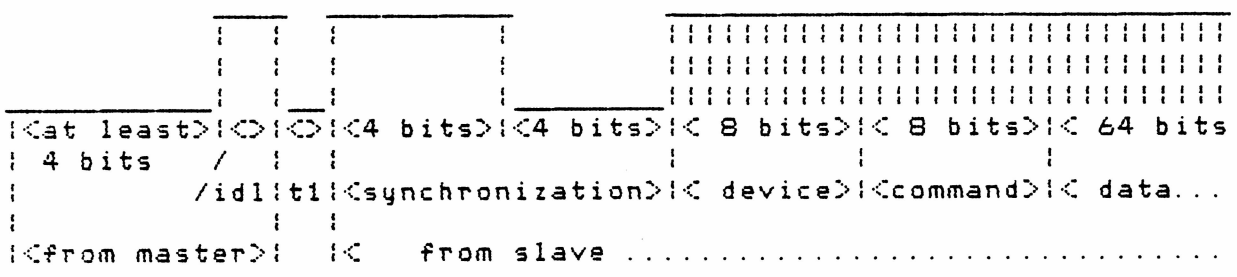
controls transmissions from peripherals. The master device sends an idle pulse when a response is expected or when the master is idle and can accept requests from peripherals. When a peripheral sends a frame with a command code between >D0 and >DF then it is interpreted as keypush input as described in section 4.4. When a peripheral sends a frame when software in the master does not anticipate then the data is placed in the display register with no action taken.

The idle pulse is only sent when the signal line has been at the logic low level for at least 4 bit windows. A peripheral may hold the signal line high after it has been given a command in order to signal to the master that it is busy. This would be used when the master is expecting a data frame in response to a command frame. The idle pulse is a logic high level which lasts for only 200 to 400 microseconds and is shorter than a bit window.

After the idle pulse has been sent peripherals have a response window in which to send a frame to the master. Peripherals for which data was requested by the master must respond in 80 to 200 microseconds. Peripherals which are sending unsolicited data as described in section 4.4 must respond within 220 to 1200 microseconds. With this priority scheme, transactions requested by the master will be completed before unsoliced input can occur. A peripheral wishing to transmit after the idle pulse, must examine the signal line during the response window to detect a possible synchronization signal from another peripheral. A response which begins in less than 80 microseconds is ignored. If no response is received within 1200 microseconds then the operation will abort.

If two peripherals are designed to respond at the same time after an idle pulse then a bus conflict may occur which would inhibit proper communication.

Transmission from a slave device.



idl is the idle pulse
t1 is the response window described above

SECTION 4

Implementation

4.1 TI-88 Driver Software

The software to drive a peripheral from the TI-88 can often be written in keypush language. OP 83 is used from the keypush language to perform the communication protocol to write a frame. OP 84 is used to read a frame from a slave device.

When using OP 83 the device and command codes are stored in the auxilary registers and the data is stored in the display register. The command, device and data must be entered in unformatted mode set up by OP 14. For instance to send data "123456789ABCDEF" using command 20 and device 30 the following sequence is used.

```
2030 (swap register)
123456789ABCDEF
OP 83
```

When the device code has a trailing zero (e.g. 30) then the trailing zero may be omitted. For instance to send command code 20 to device 30 "203" may be entered and would save one program step.

As described above the device and command codes and the data are sent out least significant bit first. This if "20A" were entered the binary sequence sent on the signal line would be "0000010100000100" for the 16 bits of device and command code. The above data would produce a bit stream of "111101111011...01001000."

When using OP 84, the device and command codes are returned in the auxillary register and the data is returned in the display register. This op-code produces an idle pulse on the signal line and waits for a response for a peripheral. Any peripheral may respond but a response is usually expected from one specific peripheral. This op-code does not check that a particular peripheral has responded. If no response is received in 1200 microseconds then the operation times out. If the operation times out then the registers are left unchanged. There is no

other indication that the operation failed.

An operation which requires a response from a peripheral will generally use both op-codes 83 and 84. The 83 op-code will give the peripheral commands and the 84 op-code will receive the peripheral's response. The following examples of peripherals illustrate this.

4.2 Audio Cassette Interface

The audio cassette interface recognizes device code hexidecimal 90.

4.2.1 Master to Slave Command Codes. The following command codes (in hexidecimal) are recognized when recieved by the cassette interface.

10 - disable cassette motor
20 - write ID number
30 - write register or program data
40 - end of block to be written
60 - find ID number
70 - write CRAM data
80 - write main memory data
90 - enable cassette motor

Codes 10 (disable motor) and 90 (enable motor) are used to control whether the cassette motor will run. These aide in positioning the tape. The data fields of these commands are meaningless.

Code 20 (write ID number) is used in a write data sequence as described below. The data field contains the ID number which is an decimal integer from 0 to 999 inclusive. Code 70 or 80 is used in a write data sequence to indicate the type of data being written. The data field for code 70 is meaningless. The data field for code 80 contains the length of the data in program steps. This length will be a multiple of 8. Code 30 (write data) may be used in place of code 70 or 80 as the first frame of data to indicate register or program memory. The data field of this frame will be the number of registers (8 steps each) of data minus one which will be transferred.

Code 30 is also used for the actual data blocks which follow the code 30, 70, or 80 frame. The data field contains 1 register

(8 bytes) of data. Code 40 (end of block) is used as the last frame in a write data sequence as described below. The data field in a code 40 frame is meaningless.

Code 60 (find ID number) is used to start a read data frame sequence as described below. The data field contains the 0 to 3 digit ID number.

4.2.2 Slave to Master Command Codes. The following command codes are sent from the cassette interface to the master device.

10 - ID number found
20 - wrong ID number found
30 - data block follows
40 - end of data block
60 - error found in data
70 - cassette interface is attached and on
80 - main memory data follows
90 - CRAM data follows

Code 10 (ID number found) is used as part of the read data sequence to indicate that the expected ID number was found. The data field for this code is meaningless. Code 20 (wrong ID number found) indicates that an unexpected ID number was found. The data field contains the ID number which was found.

Code 70 (interface is present) is used to indicate to the master device that the cassette interface is attached and powered up. This code is used as part of the read and write data sequences.

Code 80 (main memory data) or 90 (CRAM data) indicate whether the data which follows is for CRAM or main memory. The data field of code 80 contains the number of program steps of data which follow. The data field of code 90 is meaningless.

Code 30 (data block follows) may be used in place of code 80 or 90 as the first block of data transfered and indicates that register or program step data follows. The data field contains the number of registers of data minus 1 which follow. After the first code (30, 80, or 90) subsequent code 30 frames will contain data to be placed in the selected memory. The data field of each frame contains one register of data.

Code 40 (end of data block) is used to terminate the read data sequence normally. The data field is meaningless. Code 60 (error found in data) is used in place of code 40 when one or more frames have been read in error. When errors are detected

the entire file is still read but the frame(s) which were read in error contain (hexidecimal) "E8E8E8E8E8E8E8E8."

4.2.3 Read Data Sequence. The sequence of frames and actions to read data from the cassette consists of the following.

1. A code 90 (enable motor) frame sent from the master.
2. The master prompts the user to position the tape.
3. A code 10 (disable motor) frame sent from the master.
4. The master prompts the user to press playback.
5. A code 60 (find ID) frame sent from the master.
6. A code 70 frame is sent from the peripheral to indicate that it is there. If a code 70 frame is not received by the TI-88 then the "install cassette" message is displayed and the operation aborts.
7. A code 10 or 20 frame is sent from the peripheral If a code 10 is sent, the proper ID has been found. If a code 20 has been sent, the master displays the ID found to the user and asks whether to try again. If so go to step 5 else the operation aborts.
8. After the code 10 frame the interface sends a code 30, 80, or 90 frame to indicate the type of data and the length. If the data type is not as expected then go to step 11.
9. The peripheral sends one or more code 30 frames with data.
10. The peripheral sends a code 40 (end of data) or code 60 (error in data) frame.
11. The master sends a code 10 (disable motor) frame.

The peripheral holds the signal line busy (high level) between each frame to provide processing and tape I/O time. The device code in step 6 is the cassette device code (hexidecimal 90). The device code sent by the peripheral in other steps is zero.

4.2.4 Write Data Sequence. The sequence of frames to write data to the cassette consists of the following.

1. A code 90 (enable motor) frame sent from the master.
2. The master prompts the user to position the tape.
3. A code 10 (disable motor) frame sent from the master.
4. The master prompts the user to press record.
5. A code 20 (write ID) frame is sent from the master.
6. A code 70 frame is sent from the peripheral to indicate that it is there. If a code 70 frame is not received by the TI-88 then the "install cassette" message is displayed and the operation aborts.
7. The master sends a code 30, 80, or 90 frame to indicate the type of data and the length.
8. The master sends one or more code 30 frames with data.
9. The master sends a code 40 (end of data) frame.
10. The master sends a code 10 (disable motor) frame.

The interface will hold the signal line busy between frames to allow processing and tape I/O time. The device code in step 6 from the interface is contains the cassette peripheral device code (hexidecimal 90).

4.3 Thermal Printer

The thermal printer peripheral recognizes device code hexidecimal A0. The following command codes are recognized when received.

10 - Data is least significant bits of a line.
20 - Data is most significant bits of a line.
30 - Start self-test.
40 - Advance paper.
50 - Are you there?

A code 10 frame contains the 16 least significant digits of the characters for a line. A code 20 frame contains the 16 most significant digits of the characters for a line. This consists of 3 bits for each character which are right justified (to the least significant bit) for each digit. For both codes 10 and 20, the first (leftmost) digit in the display register will correspond to the leftmost printed character. The thermal printer character codes are listed in the table below.

Code 30 is intended to be used during production testing and initiates a self-test sequence. The data field is unused.

Code 40 causes the paper to advance one print line. The data field is unused.

Code 50 is sent to the printer when the master unit powers up. The data field is unused. The printer responds to this code with a frame using the printer device code and command code D0. The master interprets the data field as TI-88 keypushes.

When the printer powers up it sends a frame to the master unit at the first available idle pulse. That frame uses device code A0 and command D0. The master interprets the data as TI-88 keypushes. This data field is different from that caused by command code 50.

Thermal Printer Character Codes

blank	>00	]	>01
[	>02	"al"	>03
%	>04	plus-minus	>05
small block	>06	epsilon	>07
large block	>08	right arrow	>09
left arrow	>0A	swap arrows	>0B
big sigma	>0C	little sigma	>0D
EE	>0E	.	>0F
=	>10	)	>11
(	>12	super "n"	>13
+	>14	-	>15
times ("X")	>16	divide	>17
square root	>18	super "1"	>19
!	>1A	theta	>1B
little minus	>1C	little pi	>1D
down arrow	>1E	up arrow	>1F
0 - 9	>20 - >29		
A - F	>2A - >2F		
superscript 0 - 9		>30 - >39	
a - f	>3A - >3F		
G - V	>40 - >4F		
g - v	>50 - >5F		
W - Z	>60 - >63		
<	>64	three dots	>65
<=	>66	@	>67
#	>68	,	>69
little plus	>6A	:	>6B
?	>6C	degrees	>6D
minutes	>6E	seconds	>6F
w - z	>70 - >73		
>	>74	not equals	>75
>=	>76	x-bar	>77
&	>78	delta	>79
big pi	>7A	/	>7B
$	>7C	mu	>7D
"ph"	>7E	*	>7F

4.4 TI-88 to TI-88 Communication

Device code 00 is used to transmit from one TI-88 to another. When this transmission is used the command code must be in the range D0 through DF. Any value in this range acts identically. Values outside of this range cause the receiving TI-88 to store the data in the display register but not use it a keypushes. This may be useful for sending a combination of data and keypushes or to send data to a keypush program running in another TI-88.

The data field represents 8 keypushes. The keypushes are interpreted by the slave TI-88 just as if they were input from the keyboard. The keypushes are transmitted in the reverse order that they are to be interpreted. The eighth key code is transmitted first and the first keycode is transmitted last. Within a keypush the most significant digit is transmitted first followed by the least significant digit. This apparent difference from normal transmission is because keypushes are stored with their digits reversed. However within the digits the least significant bit is transmitted first.

The TI-88 will not receive data unless it is in the idle state as described above.

A "key code" of BB may be transmitted first to turn off alpha, equation, and learn modes if the receiving calculator might be in one of those mode. Any other key code from >00 to >7F may be transmitted. Any keycode greater than >7F (other then >BB) should not be transmitted since they will not be properly interpreted. For instance to send instructions to recall R the following sequence should be sent

"RCL" (>7C), "R" (>7C)

NOTE

There is no way to send the functions indicated by keycodes >F6 through >FF (bit, digit, and heirarchy operations).