M.I.D.I.

COMPUTER

INTERFACE

O W N E R ' S M A N U A L

WORD PROCESSED BY IBM S/38

MIDI COMPUTER INTERFACE

INTRODUCTION

In the world of electronic musical instruments the small computer systems are assuming a place of ever increasing importance.
Today almost all the keyboards on the market feature a digital system, thus also the interfacing has to be computerised for the traditional analogue system is not so practical. It is on the basis of this new musical image that a universal standard, the Musical Instrument Digital Interface (MIDI) protocol, has been developed.
MIDI is an interfacing system providing a means to interconnect various instruments through a common interface for a perfect sound synthesis.
The MIDI COMPUTER INTERFACE makes it possible for you to insert into the local MIDI communication net also a computer, thus transforming serial data into parallel data.
The second part of the owner's manual contains an introduction to the MIDI communication protocol and the tables showing SIEL implementations; you have all the elements needed to edit an application or program a completely new one!

MIDI COMPUTER INTERFACE

The SIEL MIDI COMPUTER INTERFACE is a translating hardware interface allowing direct connections between MIDI equipped instruments to a computer. The interface will link to computers based on Central Processing Unit (CPU) Z-80, 6502 and 6510. Examples of computers using these CPU are SINCLAIR ZX SPECTRUM (TM), SINCLAIR ZX 81 (TM), CBM 64 (TM), APPLE II (TM).
This interface makes it possible to interconnect a computer (parallel data) with MIDI equipped instruments (serial digital data) making their data transmission systems perfectly compatible. The MIDI COMPUTER INTERFACE actually translates the digital data transmitted in a non-homogeneous form; it doesn't contain any autonomous operative software.
The interface must always be used with one of the above mentioned computers using specific MIDI compatible software application packages.
The device is supplied with a special connector enabling immediate connection to the SINCLAIR ZX SPECTRUM (TM) and COMMODORE CBM 64 (TM).

Connectors:

- 1 MIDI IN connector – to relay keyboard information back to the computer.
- 3 MIDI OUT connectors – for connection to MIDI IN on keyboards, relaying computer information to them.
- 1 MIDI THRU connector–output from this signal fed into MIDI IN keeps its output unchanged. When information is relayed into MIDI COMPUTER INTERFACE this Thruput will enable same info to be passed.

EXAMPLES OF CONNECTIONS TO COMPUTERS

Let's take for example the connection between a MIDI compatible synthesizer and a computer using the MIDI COMPUTER INTERFACE:

1.connect MIDI OUT on synthesizer to MIDI IN on MIDI COMPUTER INTERFACE;
2.connect MIDI IN on synthesizer to MIDI OUT on MIDI COMPUTER INTERFACE;
3.connect special connector to computer (computer's connector should be the same one that is used for the connection to a printer).

A more sophisticated system could be obtained connecting a MIDI equipped synthesizer to a computer using also one or more musical peripheral units (MIDI synthesizers or expanders). In this case you can have a master instrument (to program sequences or simply as a general monitor of a whole orchestral performance) and additional MIDI instruments playing individual performances with individual pre-programmed timbres, according to the MIDI coding of the various channels to which they have been assigned.

To achieve this kind of system you should interconnect computer, synthesizer and MIDI COMPUTER INTERFACE as per Figure 1, connecting the first peripheral unit (MIDI IN) to one of the MIDI OUTs on the MIDI COMPUTER INTERFACE.

The second peripheral unit should be linked to MIDI THRU on the first peripheral unit by means of MIDI IN; the third peripheral unit should be connected to MIDI THRU of the second peripheral unit via MIDI IN; and so on up to a maximum of 16 peripheral units.

The MIDI COMPUTER INTERFACE is also provided with a special control (IN CONTROL) allowing link-up with external MIDI compatible rhythmic units (synchronized according to the MIDI standard) or with pedals for the external control of specific functions, for which some applicative packages have been developed.

Generally, it is possible to develop a computerised musical system on the basis of Figure 2.

TECHNICAL SPECIFICATIONS

Hereunder are some particular features of the MIDI COMPUTER INTERFACE, which may be of use to those who intend to develop individually MIDI compatible application packages of musical software.

The technical data are subdivided into two parts:

- The first part refers to the logic of communication under MIDI protocol.
- The second part refers to the technical specifications of the interface and to the addresses of the U.A.R.T. appliance.

MIDI PROTOCOL

MIDI is based on the numerical coding of all keyboard operation possibilities. The musical performance can easily be translated into numbers; to do it simply number each key and collect electronically all the parameters. Also the timbre can be expressed in numbers using the value indicated by the potentiometers' settings.

CHANNEL MODES

MIDI utilizes only one couple of cables containing sixteen different 'lines', called channels. When MIDI is implemented, the relation between the sixteen available MIDI channels and the synthesizer's voice assingnment must be defined.Several mode messages are available for this purpose. They are: OMNI, POLY, MONO.

In OMNI mode, the data are transmitted on channel 1 and received in all 16 channels. When OMNI is on, the keyboard transmits only to the instruments receiving on channel 1, but does not distinguish where the received voice messages come from and accepts them without discrimination.

When POLY is on, voice messages are received only in one of the sixteen channels, according to choice. In POLY mode the computer can transmit a 16 voice arrangement on 1 channel.

MONO, when on, restricts the assignment of voices to just one voice per channel controlling each voice separately; this allows you to obtain real 'legato' passages.

In OMNI and POLY modes it will be necessary to set the previous note OFF before setting the following one ON.

BINARY LOGIC AND MIDI PROTOCOL

MIDI transmits the numerical coding by means of a digital connection; it is a system of coding and transmission.

The first step to transmit a number through such a connection is to convert it into binary form. Actually all parameters are collected in binary form and don't need to be converted. Binary numbers are suitable to the digital transmission as a digital electric signal (0 or 1) can express perfectly a binary figure which from now we shall refer to as BIT.

A BIT can represent only two numbers , 0 and 1. One BIT is enough to indicate the position of a switch (0=ON, 1=OFF), but more than one is needed to indicate the potentiometers' settings (0,1,2....,100).

Generally BITs are used in groups of 8. A group of 8 BITs is called BYTE. A BYTE allows the representation of numbers from 0 through 255.

A group of 7 BITs can represent numbers from 0 through 127 and a group of 4 BITS (NIBBLE) represents numbers from 0 to 15.

MIDI always transmits one BYTE at a time of which only 7 BITs are used to communicate the data.

The easiest way of transmitting a BYTE is the parallel transmission, which utilizes eight electric digital signals; every signal represents the value of a BIT.

Another possible system consists in transmitting sequencially the BITS on a single channel.
MIDI uses the latter way (serial transmission) which is not as fast as the former, but enables easier and more economical connections.
Keyboard data are parallel. They have to be converted into serial data to be transmitted.

DATA TRANSMISSION

A MIDI command consists of one, two or three BYTES.
BYTE 1 serves to identify the message type (e.g. note ON); BYTES 2 and 3 carry the content of the message (code of the note, key velocity). We shall refer to the former one as STATUS BYTE and to the latter two as DATA BYTES.
STATUS BYTES are eight-BIT binary numbers in which the Most Significant BIT is set (binary 1).
DATA BYTES are eight-BIT binary numbers in which the Most Significant BIT is reset (binary 0).
Data can vary only in the interval 0-127 (7 bits).
Any messages (e.g. note ON) which are sent successively under the same status, can be sent without a STATUS BYTE until a different STATUS BYTE is needed. For example, if two keys are pressed simultaneously, the two key code numbers, the two velocities and only one STATUS BYTE will be transmitted.
Information can be transmitted in one of five categories. These categories are: Channel, System Common, System Real Time, System Exclusive, System Reset.
There are 16 (1-16) channels and three possible modes: OMNI, POLY, MONO.
In OMNI mode, only channel 1 is used.
In POLY mode, every unit can receive and transmit on one channel, according to choice.
In MONO mode, the voice assignment is restricted to one voice of a polyphonic unit per each voice channel.
The Channel data, which are the most used, are the only ones which can be sent selectively. All the others are received and, if possible, edited by all connected units. For the Channel data, the channel number is specified by the four Least Significant BITS of the STATUS BYTE; the four Most Significant BITS carry one of the following values: 8h, 9h, Ah, Bh, Ch, Dh, Eh, where 'h' means that the number is expressed in exhadecimal notation.
For System Common, System Exclusive and System Real Time data, there is no need to specify the channel number, thus the whole STATUS BYTE can be used to identify the message type. In particular, the first four BITS carry Fh (otherwise they carry a channel data), and the four Least Significant BITS identify the function.
System Exclusive data relate to only one manufacturer's equipment. This kind of data refer to equipments which identify a certain STATUS BYTE and can thus be used to transfer the sound programs or for any other function which is not implemented in MIDI standard (e.g. split keyboard, assignment of potentiometers settings to certain sound parameters).

MIDI COMPUTER INTERFACE

Tables I, II show the STATUS BYTES indicating the effects and the meaning of successive DATA BYTES.
Table III shows System Exclusive relative to SIEL communication, and implementation possibilities relative to OPERA 6/DK 600/ EXPANDER.

CONNECTIONS DIAGRAM

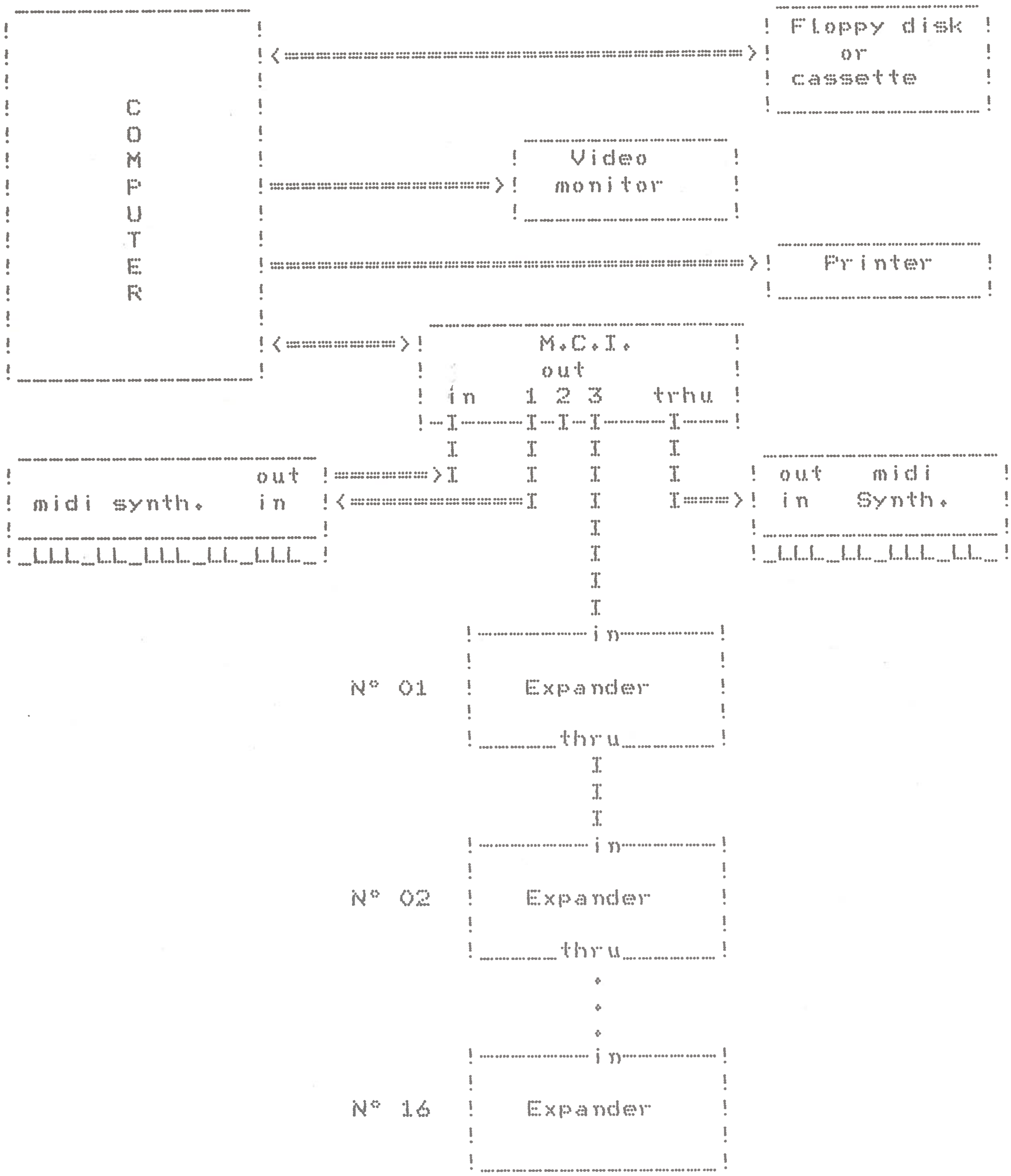

Fig. N° 1

TRANSMITTED DATA

Table I°

STATUS	Data Bytes	Description	Notes
1000NNNN	0KKKKKKK 0VVVVVVV	Note OFF status V = always 40h	1-2
1001NNNN	0KKKKKKK 0VVVVVVV	Note ON status V = dynamic value	1-2 3
10110000	01010111 0RRRRRRR	Control Change Status, 57h Record R = 00h : off R = 7fh : on	
1100NNNN	0PPPPPPP	Program change status	1-4
11110000	0IIIIIII 0DDDDDDD 0PPPPPPP MIDI SIEL Implementation (see Table III°) 11110111	System Exclusive Device Dump program End of System Exclusive	5 6 4

RECOGNIZED RECEIVE DATA

Table II°

STATUS	Data Bytes	Description	Notes
1000NNNN	0KKKKKKK	Note OFF status	1-2
	0VVVVVVV	Velocity ignored	
1001NNNN	0KKKKKKK	Note ON status	1-2
	0VVVVVVV	V = 00h note off	3
		V > 00h dynamic value	3
10110000	01010111	Control Change Status, 57h	
	0RRRRRRR	Record R = 00h : off	
		R = 7fh : on	
	01111011	All Notes Off, 7Bh	
	00000000		
	01111101	Omni Mode Select, All	
	00000000	Notes Off, 7Dh	
	01111111	Poly Mode Select, All	
	00000000	Notes Off, 7Fh	
1100NNNN	0PPPPPPP	Program change status	1-4
11110000	0IIIIIII	System Exclusive	5
	0DDDDDDD	Device	6
	0PPPPPPP	Dump program	4
	MIDI SIEL Implementation (see Table III°)		
	11110111	End of System Exclusive	

MIDI COMPUTER INTERFACE

SYSTEM EXCLUSIVE

Table III° 1/2

Byte	MSB		BIT		POSITIONS			LSB
Num.	7	6	5	4	3	2	1	0
01	0	0	0	0	F1B	F0B	F1A	F0A
02	0	0	0	0	0	0	0	0
03	0	0	0	0	S20	S19	S18	S17
04	0	0	0	0	0	0	0	S21
05	0	0	0	0	S12	S11	S10	S9
06	0	0	0	0	S16	S15	S14	S13
07	0	0	0	0	S4	S3	S2	S1
08	0	0	0	0	S8	S7	S6	S5
09	Resonance VCF				Nibble Right			
10					Nibble Left			
11	Cut off VCF				Nibble Right			
12					Nibble Left			
13	A D S R Amount				Nibble Right			
14					Nibble Left			
15	A D S R Release				Nibble Right			
16					Nibble Left			
17	A D S R Sustain				Nibble Right			
18					Nibble Left			
19	A D S R Decay				Nibble Right			
20					Nibble Left			
21	A D S R Attack				Nibble Right			
22					Nibble Left			
23	Detune Coarse DCO B				Nibble Right			
24					Nibble Left			

see notes

MIDI COMPUTER INTERFACE

SYSTEM EXCLUSIVE

Table III° 2/2

Byte Num.	MSB 7	6	5	4	3	2	1	0 LSB
25	Detune Fine DCO B				Nibble Right			
26					Nibble Left			
27	P.W. DCO B				Nibble Right			
28					Nibble Left			
29	NOISE VOLUME				Nibble Right			
30					Nibble Left			
31	P.W. DCO A				Nibble Right			
32					Nibble Left			
33	L F O III Speed				Nibble Right			
34					Nibble Left			
35	L F O III Depth				Nibble Right			
36					Nibble Left			
37	L F O I/II Speed				Nibble Right			
38					Nibble Left			
39	L F O I/II Depth				Nibble Right			
40					Nibble Left			

see notes

N O T E S

Nº	Code	Description
1	NNNN	Channel Number from 0000 is channel 1 to 1111 is channel 16
2	KKKKKKK	Note Number; range 24h = C2 through 60h = C7
3	VVVVVVV	Dynamic Value ; the musical expression levels are the following PP=10h P=14h MP=16h MF=29h F=45h FF=7Fh
4	PPPPPPP	Program Number (if enabled in "MIDI Ext") From 00h to 5Eh
5	IIIIIII	The ID number can be obtained from the I.M.A. committee, SIELID is 21h
6	DDDDDDD	Device Number for SIEL synthesizers : OPERA 6 , DK 600 , EXPANDER is 01h
7	F1B/F0A	see table below
	S1 / S21	see list below

F1A	F0A	Footage A	F1B	F0B	Footage B
0	0	16'	0	0	16'
0	1	8'	0	1	8'
1	1	4'	1	1	4'

S1 / S21

- S1 =1 on Wheel Pitch DCO A
- S2 =1 on Wheel Pitch DCO B
- S3 =1 on Depth LFO I/II
- S4 =1 on Depth LFO III
- S5 =1 on LFO I/II to DCO A Pitch
- S6 =1 on LFO I/II to DCO B Pitch
- S7 =1 on LFO III Sawtooth Wave
- S8 =1 on LFO III Square Wave
- S9 =1 on LFO III to PWM DCO A
- S10=1 on LFO III to PWM DCO B
- S11=1 on LFO III to VCF
- S12=1 on DCO A Sawtooth Wave
- S13=1 on DCO A Square Wave
- S14=1 on DCO B Sawtooth Wave
- S15=1 on DCO B Square Wave
- S16=1 on DCO B HALF Volume
- S17=1 on Dynamic to ADSR level
- S18=1 on Dynamic to Attack time
- S19=1 on ADSR to VCF
- S20=1 on ADSR to VCA
- S21=1 on Keyboard Tracking

On power up the OPERA 6 (DK 600) is always in OMNI MODE, both in transmission and in reception. On power on the EXPANDER defaults to OMNI MODE. It can be set to POLY MODE.

CONNECTIONS FOR SINCLAIR ZX SPECTRUM (TM)

1. All equipment in 'off' mode.
2. Connect ZX-spectrum (TM) to Monitor and cassette
3. Connect female connector of MCI to user port on rear panel of ZX-spectrum (TM), ensure side marked 'SIDE UP' is in correct state.
4. Connect MIDI Out/s of MCI to MIDI In/s of synthesizer. If using more than three keyboards use MIDI THRU on keyboards to expand system further.
5. If using real time sequencer program which inputs information from synthesizer keyboard rather than computer keyboard then connect MIDI IN or MCI to MIDI OUT of keyboard.
6. Set up In Control/MIDI THRU on MCI if desired.
7. Turn Monitor 'on'.
8. Turn Computer 'on' (Green spy LED on MCI will now light - if no light check connections to computer).
9. Turn cassette 'on'
10. Turn Instruments 'on'
11. Load software program via cassette.
12. RUN

CONNECTIONS FOR COMMODORE CBM-64 (TM) (SX-64)

1. All equipment in 'off' mode.
2. Connect CBM-64 (TM) to monitor and 1541 mini-disk drive or cassette (if using SX-64 connect to monitor if required).
3. Connect male connector of MCI to user port on rear of CBM-64 (TM). (In the case of SX-64, slot male connection into cartridge port on top of computer).
4. Connect MIDI OUT/s on MCI to MIDI in/s on synthesizers. If using more than three keyboards use MIDI THRU to expand system further.
5. Connect MIDI IN/THRU, CONTROL IN if required.
6. Turn monitor 'on'.
7. Turn computer 'on' (Green spy LED will light).
8. Turn 1541 disk-drive/cassette on (in the case of the SX-64, this is on when 6/7 is 'on').
9. Turn instruments 'on'.
10. Load software program via cassette/disk drive (in the case of SX-64 disk only!).

RUN

CONNECTIONS AND INTERFACES'ADDRESSES

- MIDI OUT connector (3 outputs)
- MIDI IN connector
- MIDI THRU connector
- Connector with 4 connections possibilities for external controls (ext.pedal, ext.synchronous clock, etc.)
- 44 way connector for SINCLAIR ZX SPECTRUM
- 44 way connector for COMMODORE CBM 64
- Power spy L.E.D.

CONNECTOR SINCLAIR ZX SPECTRUM
(CPU Z-80 based)

SIDE UP

+ 5V.	D7
NC	NC
SLOT	
GND	D0
NC	D1
NC	D2
A0	D6
A1	D5
A2	D3
A3	D4
NC	NC
GND	NC
NC	NC
NC	NC
NC	IORQ
NC	RD
NC	WR
NC	NC
A7	NC
A6	NC
A5	NC
A4	NC

CONNECTOR COMMODORE CBM 64
(CPU 6510 based)

SIDE UP

GND	NC
NC	+5V
NC	+5V
NC	IRQ
NC	NC
NC	NC
NC	I/O1
NC	NC
NC	NC
NC	I/O2
NC	NC
NC	NC
NC	NC
CA7	CD7
CA6	CD6
CA5	CD5
CA4	CD4
CA3	CD3
CA2	CD2
CA1	CD1
CA0	CD0
NC	NC

NOTE: U.A.R.T. DATA BUS specifications for connections:
SINCLAIR (TM) - FROM D0 TO D7
COMMODORE (TM) - FROM CD0 TO CD7

SA DIN "IN CONTROL"

MIDI DIN CONNNECTOR
(front view)
***

3 ° ° 1

5 ° ° 4

°
2

PIN	CONNECTIONS	
1	D0	CD0
2	GND	GND
3	D4	CD4
4	D1	CD1
5	D2	CD2

U.A.R.T. ADDRESS SYSTEM

Table IV°

REGISTER DESCRIPTION	COMMODORE (TM) CBM 64 Address	SINCLAIR (TM) ZX SPECTRUM Port
CONTROL	DE04h	9Fh
STATUS	DE06h	DFh
RECOGNITION	DE07h	FFh
TRANSMISSION	DE05h	BFh
MIDI "DIN" CONTROL	DE03h	7Fh

CONTROL REGISTER SPECIFICATIONS (write only)

COUNTER DIVIDE SELECT BITS

D1 CD1	D0 CD0	F U N C T I O N S (Master Clock 2,00 MHZ)
0	0	Division for 1
0	1	Division for 16
1	0	Division for 64
1	1	Master Reset

MIDI COMPUTER INTERFACE

WORD SELECT BITES

D4 CD4	D3 CD3	D2 CD2	FUNCTIONS
0	0	0	7 Bits + Even Parity + 2 Stop Bits
0	0	1	7 Bits + Odd Parity + 2 Stop Bits
0	1	0	7 Bits + Even Parity + 1 Stop Bits
0	1	1	7 Bits + Odd Parity + 1 Stop Bits
1	0	0	8 Bits + 2 Stop Bits
1	0	1	8 Bits + 1 Stop Bits
1	1	0	8 Bits + Even Parity + 1 Stop Bit
1	1	1	8 Bits + Odd Parity + 1 Stop Bit

TRASMITTED CONTROL BITES

D6 CD6	D5 CD5	FUNCTIONS
0	0	RTM = Low , Transmitting interrupt Disabled
0	1	RTS = Low , Transmitting interrupt Enabled
1	0	RTS = High , Transmitting interrupt Disabled
1	1	RTS = Low , Transmitting a Break level on the Transmit Data Output.Transmit.Inter.Disabled

D7/CD7 = High RECEIVE INTERRUPT ENABLE BIT

STATUS REGISTER SPECIFICATIONS (read only)

Table V°

Data Bus zx	Data Bus cbm64	FUNCTIONS Data Bits Status Register
D0	CD0	Receive Data Register Full
D1	CD1	Transmit Data Register Empty
D2	CD2	Data Carrier Detect
D3	CD3	Clear To Send
D4	CD4	Framing Error
D5	CD5	Receive Overrun
D6	CD6	Parity Error
D7	CD7	Interrupt Request

For further technical information on U.A.R.T. refer to MOTOROLA (TM) I.C. 68B50 specifications.

STATUS REGISTER SPECIFICATIONS (read only)

Table V°

Data Bus zx	Data Bus cbm64	FUNCTIONS Data Bits Status Register
D0	CD0	Receive Data Register Full
D1	CD1	Transmit Data Register Empty
D2	CD2	Data Carrier Detect
D3	CD3	Clear To Send
D4	CD4	Framing Error
D5	CD5	Receive Overrun
D6	CD6	Parity Error
D7	CD7	Interrupt Request

For further technical information on U.A.R.T. refer to MOTOROLA (TM) I.C. 68B50 specifications.