AN08

CATERLINK COMMUNICATION PROTOCOL

Release 0.3 – October 2002

Index

1. GENERAL INFORMATION

This document defines a Serial Communication Protocol for electrically connected Devices.

The interface is arranged as a Master-Slave protocol. The ZIP Caterlink is the Slave and the Master Controller can be for example a Standard Cash Register or a PC Cash Register.

The Serial Communication Protocol is based on the Poll-Select communication procedure.

2. PROTOCOL STRUCTURE

The Communication Protocol has a structure defined by the following layers:

- Physical Layer
- Data Link Layer
- Application Layer

2.1. Physical Layer

The Communication is based on asynchronous serial communication using the following specification:

- Start bits: 1
- Data bits: 8
- Parity bits: None
- Stop bits: 1
- Communication Speed: 9600

The hardware connection is a RS 232 connection; the ZIP Caterlink connector is D-SUB 9 pin Female type with the following assignments:

Pin number	Signal Name	Signal Direction (ZIP Caterlink view)	Description
2	TxD	Output	Transmit data to the master
3	RxD	Input	Receive data from the master
5	Gnd	-	Signal GND

2.2. Data Link Layer

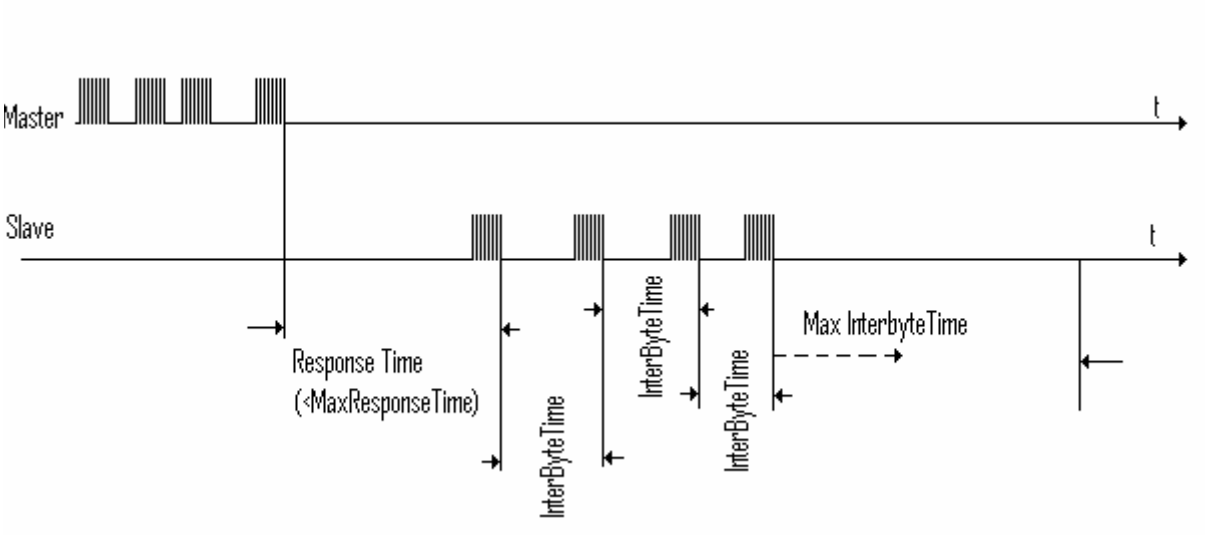
2.2.1. Block Format

POLL Command

- Master polls Slave by sending a message in order to receive data from the addressed Slave. Message specifies the control character “Poll” and the Slave address (In this document Slave address is 0x00).
- Slave responds with a message - if there is a pending one - starting with the Start of Header control character. The message specifies the slave address and a Data Block.
- Master acknowledges the Slave response.
- Slave ends the transmission by sending an End of Transmission sequence.
- If there’s no pending message slave responds with an End of Transmission.

SELECT Command

- Master selects Slave by sending a message in order to transmit data to the addressed Slave. Message specifies the control character “Select” and the Slave address.
- Slave either acknowledges the Master selection if it is ready to receive or responds with a Negative Acknowledge if it’s not.
- If Slave is ready to receive, Master sends a message starting with a Start of Header control character. The message specifies the slave address and a Data Block.
- Slave acknowledges the received message.
- Master ends the transmission by sending an End of Transmission sequence.
- Slave confirms the end of transmission with an End of Transmission sequence.
- Master can transmit the next Data Block after the previous one has been acknowledged.
- If the received Data Block is corrupted, Slave does not acknowledge and gets rid of the received Data Block.
- If Master receives a not acknowledge, it transmits the Data Block once again.
- If Master does not receive an acknowledge within a specified time-out interval, called Maximum Response Time, it transmits the Data Block once again.
- Each Data Block starts with a Start of Text control character and ends with an End of Text control character.
- After each Data Block follows a 8 bits CRC calculated making a bitwise XOR on the data between Start of Text control character and End of Text control character.
- The communication between Master and Slave is composed of message of different length. The end of the message is caught using the InterByteTime. The InterByteTime is a time that starts when a byte arrives and stops with the arrival of the next byte. So when the InterByteTime becomes greater than the Maximum-InterByteTime the message is considered finished. The default value for the Maximum-InterByteTime is 5 ms.
- ResponseTime is the time that the Master waits to receive an answer from the Slave. If the ResponseTime becomes greater the Maximum-ResponseTime, it is supposed that the Slave is busy or not connected and the question is repeated. The default value for Max-ResponseTime is 100 ms.



2.2.2. Control characters List

Name	Hex Value	Meaning
SOH	0x81	Start Of Heading.
STX	0x82	Start Of Text.
ETX	0x83	End Of Text.
EOT	0x84	End Of Transmission.
ENQ	0x85	Enquire.
ACK	0x86	Acknowledge.
NAK	0x95	Negative Acknowledge.
DLE	0x90	Data Link Escape to get control characters STX, ETX transparent: STX, ETX, DLE are transmitted as DLE STX, DLE ETX, DLE DLE.
POLL	0x70	Poll command
SELECT	0x71	Select command

2.2.3. Examples of Poll Command

Master polls Slave which is busy or not connected:

Master: **EOT**, Address, Poll, **ENQ**.

Slave: No Response.

Master polls Slave which has nothing to transmit:

Master: **EOT**, Address, Poll, **ENQ**.

Slave: **EOT**, Address

Master polls slave which replays with a single Data Block:

Master: **EOT**, Address, Poll, **ENQ**.

Slave: **SOH**, Address, **DLE**, **STX**, Data, Data, .. , Data, **DLE**, **ETX**, CRC7..0.

Master: **ACK**, Address.

Slave: **EOT**, Address.

Master polls Slave which replays with a single Data Block containing a Checksum error (Improbable event due probably to external noise):

Master: **EOT**, Address, Poll, **ENQ**.

Slave: **SOH**, Address, **DLE**, **STX**, Data, Data, ...Data, **DLE**, **ETX**, CRC7..0.

Master: **NACK,** Address.

Slave: **SOH**, Address, **DLE**, **STX**, Data, Data, ... Data, **DLE**, **ETX**, CRC7..0.

Master: **ACK,** Address.

Slave: **EOT**, Address.

2.2.4.Examples of Select Command

Master selects Slave but Slave is busy or not connected:

Master: **EOT**, Address, Select, **ENQ**.

Slave: no response.

Master selects Slave in order to transmit a Data Block and Slave is ready to receive:

Master: **EOT**, Address, Select, **ENQ**.

Slave: **ACK**, Address.

Master: **SOH**, Address, **DLE, STX**, Data, Data, ..., Data, **DLE**, **ETX**, $CRC_{7..0}$.

Slave: **ACK**, Address.

Master: **EOT**, Address.

Slave: **EOT**, Address.

Master selects Slave in order to transmit a Data Block containing a wrong Checksum (Improbable event due probably to external noise):

Master: **EOT**, Address, Select, **ENQ**.

Slave: **ACK**, Address.

Master: **SOH**, Address, **DLE, STX**, Data, Data, ... Data, **DLE**, **ETX**, $CRC_{7..0}$.

Slave: **NAK**, Address.

Master: **SOH**, Address, **DLE, STX**, Data, Data, ... Data, **DLE**, **ETX**, $CRC_{7..0}$.

Slave: **ACK**, Address.

Master: **EOT**, Address.

Slave: **EOT**, Address.

2.3. Application layer

The Application Layer is the highest level of the protocol messages.

2.3.1. Master to Slave available commands

In the following table are defined the available commands.

A communication Data Block from Master to Slave can be composed of:

- Only one Command Byte;
- A Command Byte followed by Data Bytes. The number of Data Bytes depends on the Command.

List of available commands.

Command	Name	Hex value
Status	STS	0x73
Read Media	RDM	0x65
Increment Credit	INC	0x79
Decrement Credit	DEC	0x78
Beep	BEP	0x76
System Identifier	SYS	0x88

2.3.2. Slave to Master Data Block format

A communication Data Block from Slave to Master consists of:

- Data Bytes;
- ACK or NAK.

2.3.3. Status command (STS)

Master asks Slave the Status of the Media (key/card) and of the Slave. The Status sequence is the following:

Master:	STS				
Slave:		SWD_{LSB}	SWD_{MSB}		

where SWD_{LSB} has the following pattern:

bit 0: 1 means media is inserted in the Slave reader and it isn't in an error state;

bit 1,2,3,4,5: are reserved to codify the Media/Slave error (see following table);

Bit Configuration (b_5 b_4 b_3 b_2 b_1)	Number of Error	Meaning
0 0 0 0 0		No errors
0 0 0 0 1	Error 1	Media with altered credit area
0 0 0 1 0	Error 2	Media with wrong codes
0 0 0 1 1	Error 3	Credit not charged to not exceed limit
0 0 1 0 0	Error 4	Not enough credit in the media
0 0 1 0 1	Error 5	Not used
0 0 1 1 0	Error 6	Media with wrong function code
0 0 1 1 1	Error 7	Media with credit greater than usable credit
0 1 0 0 0	Error 8	Not used
0 1 0 0 1	Error 9	Media writing error
0 1 0 1 0	Error 10	System Error: Databox Connection Error
0 1 0 1 1	Error 11	Not used
0 1 1 0 0	Error 12	Media written but not checked
0 1 1 0 1	Error 13	Media with number inserted in Black List
0 1 1 1 0	Error 14	Not used
0 1 1 1 1	Error 15	Media with one purse in a system of 3 purses
1 0 0 0 0	Error 16	Media must be read, before to be decremented
1 0 0 0 1	Error 17	Not used
……..	…….	……
1 1 1 1 1	Error 31	Not used

bit 6: writing enable. This flag is set to:

1 after a successful Media reading process. The media reading process follows a RDM command sent by the Master;

0 after the extraction of the Media from the reader of the Slave or before a RDM command;

bit 7: do not care.

The SWD_{MSB} is a number that has the following possibilities:

Code	*Function*		*Bonus setting*
	Slave SW version	*Purse number used*	
0	Sw 1 purse	-	without Bonus
1	Sw 1 purse	-	Bonus1
2	Sw 1 purse	-	Bonus2
3	Sw 1 purse	-	Bonus1and Bonus2
10	Sw 3 purse	1	Without Bonus
11	Sw 3 purse	1	Bonus1
12	Sw 3 purse	1	Bonus2
13	Sw 3 purse	1	Bonus1 and Bonus2
14	Sw 3 purse	2	without Bonus
15	Sw 3 purse	2	Drink/Food Bonus
16	Sw 3 purse	3	without Bonus
17	Not Used	…	…
…	…		
…	…		
255	Not used		

2.3.4.Read Media command (RDM)

Master sends a RDM command and Slave answers sending the Media data (Media has correct codes).

Master:	RDM							
Slave:		SWD_{LSB}	SWD_{MSB}	N_{LSB}	N	N_{MSB}	CC_{LSB}	CC_{MSB}
Slave:	DC	Ds	FC	CA_{LSB}	CA_{MSB}			

Master sends a RDM command but either the Media is not more present or Media codes are wrong.

Master:	RDM						
Slave:		SWD_{LSB}	SWD_{MSB}				

Where:

N: Number of the Media (it is a number between 1 and 999999);

CC: Customer Code (it is used for purse 1; it is set to 0 for purses 2 and 3);

DC: Department Code (for purse 1 it is a number between 0 and 255; for purse 2 and 3 it is a number between 0 and 99);

Ds: Discount Code (0, 1, 2, 3);

FC: Function Code;

Media Function	Code
Sale Key	0
Free sale Key	1
Revalue Key	4

CA: Credit Available (it is a number between 0 and 65.535);

X_{LSB}: X Low Significant Byte;

X_{MSB}: X Most Significant Byte.

2.3.5. Increment Credit (INC)

Master transmits to Slave the command Increment Credit followed by:

- *A* - the amount of the recharge (2 bytes);
- D - the discount (2 bytes);
- M - a message (10 bytes);
- DispT - the *"display" time* (1 byte) (the message is displayed on the second line of the Slave display for *"display" time* seconds).

In order to check if the increment of the credit has been successful, Master can send command status and Slave answers with SWD.

Master:	INC	A_{LSB}	A_{HSB}	D_{LSB}	D_{MSB}	$M_{chr\ 0}$	…..	$M_{chr\ 9}$	DispT
Slave:									
Master:	STS								
Slave:		SWD_{LSB}	SWD_{MSB}						

2.3.6. Decrement Credit (DEC)

Master transmits to Slave the command Decrement Credit followed by:

- A - the amount to be subtracted (2 bytes);
- D - the discount (2 bytes);
- M - a message (10 bytes);
- DipT - the *"display" time* (1byte).

In order to check if the decrement of the credit has been successful, Master can send command status and Slave answers with SWD.

Master:	DEC	A_{LSB}	A_{HSB}	D_{LSB}	D_{MSB}	$M_{chr\ 0}$	…..	$M_{chr\ 9}$	DispT
Slave:									
Master:	STS								
Slave:		SWD_{LSB}	SWD_{MSB}						

2.3.7.System Identifier (SYS)

Master sends System Identifier command to Slave in order to have the hardware and software configuration of the Slave.

Master:	SYS							
Slave:		SYSID	$Name_1$	$Name_2$	$Name_3$	$Name_4$	$Name_5$	$Name_6$
Slave:	Sw_1	…….	……	…….	Sw_{10}	Man_1	Man_2	Man_3

Where:

- SYSID: 111 - Slave with 8 Kbytes External RAM;
  112 - Slave with 32 Kbytes External RAM;
  114 - Slave with 512Kbytes External RAM;
- Name: it is the name of the Slave (6 bytes);
- Sw: it is the software release programmed in the Slave (10 bytes);
- Man: it is the Manufacturer name (3 bytes).