

# TDS

## Ticket tearing and issuing machine

### **COMMUNICATION PROTOCOL**

PRODUCED BY

ADEL S.r.l.

Legal address: Via Saffi N. 58

Operative address Via Nonantolana N. 970/1

41122 MODENA

Tel 059-2550137 Fax 059-2551207

e-mail adel@adel2000.it

**REVISION**

<b>DATE</b>	<b>AUTOR</b>	<b>VER.</b>	<b>CHANGES</b>
02/09/2009	GB	1.2	First edition
02/05/2010	GB	1.3	Modified par 4.3

## INDEX

1.	COMUNICATION PROTOCOL .....	3
1.1.	INTERFACE.....	3
1.2.	TRANSMISSION MODE .....	3
2.	COMMANDS .....	3
3.	ANSWERS.....	4
4.	COMMANDS AND ANSWERS LIST .....	4
4.1.	RESET COMMAD .....	5
4.2.	VERSION REQUEST .....	5
4.3.	STATUS REQUEST.....	5
4.4.	FEEDING COMMAND .....	6

## 1. COMUNICATION PROTOCOL

### 1.1. INTERFACE

The standard interface is EIA RS232C with standard electrical levels (+/- 12V)

### 1.2. TRANSMISSION MODE

Communication is serial, asynchronous, full duplex, with the following characteristics:

Speed	19200 bauds
Bit per character	7
Stop Bit	1
Parity	Even

## 2. COMMANDS

The “command” is the message sent from the host to the module to execute a specific operation

The commands format is:

`<stx> cc data <etx>`

Where:	<code>&lt;stx&gt;</code>	Start of the message (character 02 HEX)
	<code>cc</code>	Code of the command to execute composed by two ascii characters.
	<code>data</code>	Data and parameters associated to the command, of variable length according to the type of command.
	<code>&lt;etx&gt;</code>	End of message (character 03 HEX)

In the indicated format the spaces between the fields are inserted for greater clarity, and they do not make part of the message; for example the reset command is composed by four characters ( 02HEX 30 HEX 31 HEX 03 HEX), and the format is indicated with “`<stx>01 <etx>`”.

When the host sends a command, the following cases can be verified:

- 1) The module answers with the character **ack** (06HEX). This answer indicates that there are no communication errors and that the command is formally correct. Now the host is waiting for the answer message that the module will send at the end of the execution of the command. The answer time changes from a few milliseconds to several seconds, according to the command. The execution time of the command can vary from a few milliseconds up to some seconds according to the command type.
- 2) The module answers with the character **nak** (15 HEX). This answer shows when there are communication errors, or if the command is not valid. The host must transmit the command again, and eventually signal the state of out of service of the module if the error is repeated.
- 3) The module does not answer within 300 msec. This happens when the module is not connected or is down. At this point the host can transmit the command again, or send a status request, and eventually signal the state of out of service of the module if the error is repeated.

### 3. ANSWERS

The answer is the message sent from the module to confirm to the host the execution of the received command, or if there is an asynchronous and independent events from the execution of commands (for example a hardware reset).

The format of the answer is:

*<stx> cc rr data <etx>*

Where:

- <stx>* Start of the message (character 02 HEX)
- cc* Code of the executed command composed by two ascii characters. If the answer is generated from an asynchronous event, the code command assumes value " 00 ". For example a hardware reset sends the message " stx 00 51 all etx ".
- rr* Feedback command code, composed by two ASCII characters. This code takes the value of the running command plus 50. This code allows to understand wich asynchronous event occurred.
- data* Data associated to the command of variable length according to the type of answer. The answer can also have no data.
- <etx>* End of message (character 03 HEX)

In the indicated format the spaces between the fields are inserted for clarity, and they are not part of the message. For example the answer to the reset command is composed by six characters (02 HEX, 30 HEX, 31 HEX, 35 HEX, 31 HEX, all, 03 HEX), and the format is indicated with “*<stx> 01 51 all <etx>*”.

When the module receives a valid message from host, it sends the character **ack** (06 HEX), executes the command, and finally sends the answer. At this point the following cases can be verified:

- 1) The host receives the answer without errors. In this case it analyses the answer and it can send a new command to the module.
- 2) The host receives the answer with communication errors, or the answer is not valid. In this case it can send the **nak** character (15 HEX) in order to ask the repetition of the answer from the module, and eventually to signal the out of service of the module if the error is repeated.
- 3) The host does not receive the answer within 5seconds: in this case it can send the **nak** character (15 HEX) to ask the repetition of the answer from the module, or to send the status request, and eventually to signal the out of service of the module if the error is repeated.

### 4. COMMANDS AND ANSWERS LIST

In the following paragraphs the command and the associated answer available for the TDS module are described.

Command	Answer	Description
<i>&lt;stx&gt;01&lt;etx&gt;</i> ”.	<i>&lt;stx&gt;01 51 al &lt;etx&gt;</i> ”	Software reset
<i>&lt;stx&gt;02&lt;etx&gt;</i> ”.	<i>&lt;stx&gt;02 52 ve &lt;etx&gt;</i> ”	Version request
<i>&lt;stx&gt;03&lt;etx&gt;</i> ”.	<i>&lt;stx&gt;03 53 al op ap ms&lt;etx&gt;</i> ”	Status request
<i>&lt;stx&gt;04 ap &lt;etx&gt;</i> ”.	<i>&lt;stx&gt;04 54 status &lt;etx&gt;</i> ”	Feeding command

#### **4.1. RESET COMMAD**

This command executes a software reset of the CPU. The module is initialized.

Command: *<stx>01<etx>*

Answer: *<stx>0151 al<etx>*

Where : *al* A character to indicate the alarm code

"0" No Error

"7" Ticket jam

When the module is reset by the button SW1 or when is turned on, it sends the messages

*<stx>0051 al<etx>*

#### **4.2. VERSION REQUEST**

The host may request the firmware version by sending this command

Command: *<stx>02<etx>*

Answer: *<stx>02 ve<etx>*

Where: *ve* Firmware version

#### **4.3. STATUS REQUEST**

The host can use this command to have the realtime status of module. The host can send this command at any time also while command is in execution.

Command: *<stx>03<etx>*

Answer: *<stx>03 53 al op ap ms [rp]<etx>*

Where: *al* A character to indicate the alarm code

"0" No error

"1" The command is in execution

"2" Command not executed ticket not present

"3" Command not executed ticket already present

"7" Ticket is jammed

*op* A character to indicate wich operation the module is running

"0" No operation

"1" Reset operation

"3" Feeding operation

*ap* A character to indicate the feeding status

"0" Ticket not present

"1"	Ticket present
"2"	Ticket inside the module
<i>ms</i>	A character to indicate front opening status
"0"	Front opening free
"1"	Front opening busy
<i>rp</i>	An optional character to indicate the reserve paper status
"0"	The paper is in reserve
"1"	The paper is full

NOTA The *rp* status is optional it could be activated by moving the dip switch 2 in ON position

#### **4.4. FEEDING COMMAND**

Tickets to be issued are loaded when they are inserted into the module

Command: `<stx>04 ap<etx>`

Where: *ap* A character to indicate feeding procedure type

"A"	Load the ticket inside the module and keep it ready to be issued
"E"	Load and issue a ticket

Answer: `<stx>04 status<etx>`

Where: *status* See the parameters of status command