The Synel Communication Protocol

User's Manual

Synel Industries Ltd.

This manual was issued on 4/28/08.
Catalog no. 660004 Document no. SYPROT-222-03

This document has been prepared for SY7XX firmware version 6.209.

Table of Contents

1. Overview

The protocol is the commands used by Synel's data collection terminals. All communication sequences are initiated by the host. The host computer polls the terminals, it queries each terminal for data records or fingerprint templates. If the terminal replies *Yes* then data transmission is permitted. A *No* reply means that the host moves on and polls the next terminal. This process is repeated continuously.

Host commands transmission can be performed comprehensively to all terminals or to a specific terminal. If the command is directed to a specific terminal (except the RESET LINE command), the terminal is obliged to produce a reply. The terminals are programmed by the host, therefore the host instructs the terminals to enter into programming mode before transmitting programming commands.

Commands transmitted by the host have a different structure than the commands received from the terminal, thus each command group is described separately.

1.1 Convention

Bit Type = [O..1); One bit
Nibble Type = [0..15); four bits
Byte Type = [0..255); eight bits
Word Type = Two Bytes; 16 bits

1.2 Commands sequence

The command sequence starts with a host command followed by a reply from the terminal. Both the command and the reply sequence includes the operation code followed by the terminal's multidrop ID. This is followed by applicable data, the sequence ends with a 4-byte Cyclic Redundancy Check (CRC) and a 4-byte - End of Transmission (EOT) character both in hexadecimal. Other characters are written in printable ASCII.

1.3 Format of commands and replies

Each command or reply is organized in the format shown below:

Command or reply	1 byte
Multidrop ID of a specific terminal or @ for all terminals	1 byte

Data field	up to 128 bytes
Communication CRC	4 bytes
End of command or reply (EOT)	1 byte

1.3.1 Command

A command is sent from the host computer to a specific or to all terminals.

1.3.2 Reply

A reply is the terminal's response to a command. It is sent in the same format as the command.

1.3.3 Standard reply

The terminal replays ACK (06h) unless a specific reply is requested, that is if the host command was received correctly. The host anticipates a reply within 50 milliseconds of the programmed time-out. If replay was not received within scheduled time, the host assumes that the terminal did not received the command and it will be resent.

1.3.4 Multidrop ID

The ID represents the terminal position on a multidrop line, definition range between 0 and F hexadecimal format. Up to 32 terminals may be connected to a single multidrop line, via a single computer port.

Synel's software identifies the communication port as well as the terminal's multidrop line position. Multidrop ID includes only terminal position, since the protocol is sent along one specific multidrop line.

The multidrop ID is exchanged with the @ character, this when command is transmitted to all terminals connected to the multidrop line and not designated to a specific terminal. Not in use with the commands: E, G, I, K and L.

1.3.5 Cyclic Redundancy Check (CRC)

The protocol's error-detecting capabilities, a number of 4 bytes generated and appended to the end of a block of data to provide error detection. The CRC performs 16 bit operations on each byte:

Definitions

1. RA, RB, RC, RD 16 bit registers.

2. Rn register
   Rn(H)- Least Significant Bit (LSB) 8 bit register.
   Rn(L)- Most Significant Bit (MSB) 8 bit register.
   Rn(SR)*i - Shift Right of i bits with leading zeros.
   Rn(SL)*i - Shift Left of i bits with trailing zeros.
3. **XOR -** OR operation:

A	B	A^B
0	0	0
0	1	1
1	0	1
1	1	0

CRC start registers:

RC register will contain the last computed CRC bytes (zero at initialization).
RA(L) register contains the next character (8 bits).

CRC end registers:

RC register will contain the new CRC bytes.

CRC Algorithm

RD<-----RC RA(H)<--0

SR(RC)*8

RC<-----RC XOR RA RB<-----RC SR(RC)*4

RC<-----RC XOR RB RB<-----RC SL(RC)*5

RC<-----RC XOR RB SL(RD)*8

RC<-----RC XOR RD RB<-----RC SL(RC)*12

RC<-----RC XOR RB

Transformation of the computed CRC into ASCII characters

The computed CRC can be represented by a sequence of four hexadecimal characters, each of which will be translated into an ASCII character by adding the nibble 3 hex to its left.

For example: The addition of the 3 hex nibble to the left of A hex results 3A hex.

Example 1

Computed CRC ASCII representation
1234 hex"1234"
9AF0 hex"9:?0"

Example 2

Two characters computed CRC ASCII representation
"A4" "482:"
"B4" "1=79"
"C4" "2>48"
"D4" ";7=?"

1.3.6 EOT

The End Of Transmission (EOT) is 04 hex.

2. Commands list

The commands tables contains a description of commands transmitted from the host to the terminal and the commands received from the terminal:

- Commands format is case sensitive.
- Lowercase letters cannot be replaced with uppercase letters and vice versa.

2.1 Host commands sent to the terminal(s)

	Definition	Description
A	SEND DATA	Initiates a sequence for terminal transmission of a full data block of up to 128 bytes.
B	SEND ALL	Initiates sequence for a terminal transmission of a partly-full data block.
C	CLEAR BUFFER	Directs terminal to clear all transmitted and acknowledged records stored at the memory buffer.

c	CLEAR BY DATE	Directs terminal to clear all transmitted and acknowledged records stored at the memory buffer according to the specified date.
D	GET STATUS	Initiates a sequence receiving data on the terminal's status.
E	SET STATUS	Initiates a sequence that sets the terminal's internal clock.
F	ACKNOWLEDGE LAST RECORD	The host has received the terminal’s last data record trasmission and it forwards a notification.
G	RESET BUFFER	Directs the terminal to reset all transmitted and acknowledged records and mark them as unsent stored records.
I	TABLE OPERATION	Initiates a sequence for table operation excution as; load, delete, replace.
K	HALT	Commands the terminal to terminate normal operation mode and proceed to programming mode.
L	RUN	Commands the terminal to terminate programming mode and proceed to normal operation mode.
O	SEND ONLY QUERY	Sends an existing query, not sent if data was not found. An N is sent if QUERY does not exist.
Q	QUERY REPLAY	The terminal produces the query response. There are three responses T=Timeout, O=Offline, B= Buzy for host, L=Long query
R	RESET LINE	Directs the terminal to terminate transmission, re-establish communication and waits for the next SEND DATA for SEND ALL command.
S	SYSTEM COMMANDS	Memory commands form the terminal to the host
V	Fingerprint V-HOST, v-TERMINAL	For SY780 terminals Version 5.0.3

2.2 Terminal commands to the host

2.2.1 Primary commands

These commands always appear in lowercase letters and may accompanied by secondary commands, See page -12.

Command	Definition	Description
b	BUSY	The terminal is currently busy and temporary unable to respond the host command.
d	DATA RECORD	The command is accompanied by a data record.
n	NO DATA	Data stored on the terminal that meets the host request parameters is not found.
q	QUERY FOR HOST	There is a terminal query directed to the host.
s	TERMINAL STATUS	The command is accompanied by a data that represents the terminal current status.
t	BLOCK RECEIVED	The terminal ackwnoladges receiving the transmitted host block.
v	LAST COMMAND	The command is accompanied by a data that represents the last received command from the host and whether it is comprehensible to terminal.
ACK (06hex)	ACKNOWLEDGE	Terminal acknowledgs that the host command was received clearly.
NACK (15hex)	NOT ACKNOWLEDGE	Terminal notifices that the host command was not received clearly.

2.2.2 Secondary commands

The secondary commands are transmitted when the terminal denies accepting the tables programming at the current time. These commands describes the cause for not accepting the programming tables, they are always appear in

uppercase letters.

Command	Definition	Description
C	IN PROGRAMMING MODE	The terminal is currently ready to receive programming tables and denies accepting transaction input.
E	TABLE ALREADY EXISTS	A table with the same ID already exists in the terminal's memory.
F	MEMORY FULL	The terminal's memory is full.
G	BLOCK REC. AND BEING STORED	A data block was received and processed to be stored at the terminal's memory.
L	PERFORMING LAST COMMAND	The command execution is in process by the terminal.
R	TRANSACTION IN PROGRESS	A transaction input is currently in progress at the terminal.
W	BLOCK RECEIVED AND STORED	A data block was received and already stored at the terminal's memory.

3. Host - Terminal Command sequences

The communication protocol between the host computer and the terminals is organized into five different groups:

- Terminal status
- Online communication and data collection
- Programming
- Command verification
- Special host displays

3.1 Terminal status sequences

There are two protocol sequences which work affect the terminal’s status:

- Send status
- Set status

3.1.1 Send Status

This sequence logs terminal status

.

Host command			**Terminal reply**		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
D (44*h*)			**s (73*h*)**		
	multidrop ID or @	1		multidrop ID or @	1
	CRC	4		model no. (0-9)	1
	EOT	1		version no. (0-9)	1
				terminal ID	5
				terminal status (year)	2
				terminal status (month)	2
				terminal status (day)	2
				terminal status (hour)	2
				terminal status (minute)	2
P				Active function P: **Z** = Crash status **i** = Init mode **T** = Technician mode **P** = No program mode **b** = Blocked **A....0** - Active functions A,B,	1
				Number of full buffers	3
				Number of faulty buffers	3
				Number of full buffers transmitted and not cleared	3
				Number of empty buffers	3
				Memory used for tables	5

				Station type: **A** - SY-110 **B** - DCM-100 **C** - SY-101x **D** - SY-70x **E** - SY-71x, SY-75x, SY-77x **M**- SY400 **m** - SY400 on SY78x	1
				hardware **V** - SY78x **v** - SY760 **?** - Unknown type **0** - DCT **2** - SY-125 **3** - SY-120h **4** - SY-130 **6** - SY-180 **8** - SY-101e **9** - SY-101h	

				Power status: **0** = Power off **1** = Power on	1
				UserDefinedField	4
				NetworkProtocol: **'-'** Network not supported **N** - Network card not present **T** - TCP/IP **U** - UDP	1
				Network Polling Interval: -- Network not supported	2

				Fingerprint mode: **M** - Terminal Master mode **S** - Terminal Slave mode **-** FPU not supported	1
				Reserved	12
				CRC	4
				EOT	1

3.1.2 Set status

This sequence sets the terminal's time and active function.

Host command			**Terminal reply**		
Command	**Data**	**bytes**	**Command**	**Data**	**bytes**
E (45*h*)			**ACK (06*h*)**		
	multidrop ID or @	1		M.ID (according to the ASCII table @=all	1
	Year	2		CRC	4
	Month	2		EOT	1
	Day	2			
	Hour	2			
	Minute	2			
	Active function	1			
	Seconds (optional)	2			
	CRC	4			
	EOT	1			

3.1.3 Set Technician Parameters

Sets new values for technician menu communication parameters.

Location	Length	Name	Name	
1	1	Op-code	S	Set hardware
2	1	ID	0-31	M.ID (according to the ASCII table @=all
3	3	Sub-code	HPS	Hardware Parameter Set
6	2	New ID	00-31, --	New M_id value
8	1	Serial Port #0	0	Used only for Host
9	1	Serial Port #1	1 or 3	FPU or Serieal smartcard reader.
10	1	Serial Port #2	2 or 3	Printer or Serieal smartcard reader.
11	1	Baud rate for host	-,0-7	See “Baud Rate Table”
12	1	Baud rate for FP	-,0-7	See “Baud Rate Table”
13	1	Baud rate for PRINTER	-,0-7	See “Baud Rate Table”
14	1	Number of modem ring	N,-,0-9	If modem does not exist put N Otherwise the number of rings to answer the call.
15	1	Net card type	A-F	A- Ethernet 10 Mbps B – Ethernet 10/100 Mbps C- N. Used D- Bluetooth E- Wi-Fi F – Ethernet 100 Mbps
16	4	CRC		
20	1	EOT		

The reply to this command is ACK or NACK with old setting, the setting is only changed after acknowledging.

Baud Rate Table

The following table displays the value to use and the baud rate it represents.

	Baud rate
0	1200
1	2400
2	4800
3	9600
4	19200
5	38400
6	57600
7	115200

3.1.4 Get Current Parameters

Returns the current values for technician menu communication parameters.

Location	Length	Value	Desc.
1	1	S	Opcode
2	1		M.ID (according to the ASCII table @=all
3	3	HPG	Sub code
6	4		CRC
10	1	04Hex	EOT

Reply from terminal:

Location	Length	Name	Value	Desc
1	1	Op-code	S	Set hardware
2	1	ID	0-31	M.ID (according to the ASCII table @=all
3	3	Sub-code	HPG	Hardware Parameter Set
6	2	New ID	00-31, --	New M.Id value
8	1	Serial Port #0	0	Used only for Host
9	1	Serial Port #1	1 or 3	FPU or Mifare R/W

10	1	Serial Port #2	2 or 3	Printer or Mifare R/W
11	1	Baud rate for host	-,0-7	See “Baud Rate Table”
12	1	Baud rate for FP	-,0-7	See “Baud Rate Table”
13	1	Baud rate for PRINTER	-,0-7	See “Baud Rate Table”
14	1	Number of modem ring	N,-,0-9	If modem does not exist put N Otherwise the number of rings to answer the call.
15	1	Net card type	A-F	A- Ethernet 10 Mbps B – Ethernet 10/100 Mbps C- N. Used D- Bluetooth E- Wi-Fi F – Ethernet 100 Mbps
16	4	CRC		
20	1	EOT		

3.1.5 Get Hardware Configuration Parameters from Terminal

Returns the current values for technician menu communication parameters.

Location	**Length**	**Value**	**Desc.**
1	1	S	System command
2	1	1	Multidrop ID
3	1	H	Hardware configuration
4	1	G	
5	3		SRC
8	1	04Hex	EOT

Reply from terminal:

Location	Length	Name	Value	Desc
1	1	System command	S	System command
2	1	Multidrop ID	1	Multidrop ID
3	1	Hardware configuration	H	Hardware configuration
4	1	General	G	
5	2	Terminal model	08/04	08 - SY-780/ SY-760/ SY-400 04 - SY-75x/ SY-777/
7	2	Firmware	00/02	00 - Firmware version 02 - Firmware subversion
9	1	Firmware subversion	0-2	0- SY-760 (1 COM port) 1- SY-780 (3 COM ports) and FPU type 1 or 2 2- SY-780 (3 COM ports) and FPU type 3
10	12	Date	@dd.mm.yyyy	Date proceeded by a @ symbol in dd.mm.yyyy format
22	1	Terminal type	M/V/v/E	M - SY-400 - Metal box V - SY-78x v - SY-760 E - TimeLOG

23	1	Keyboard type	F/D/B/b	F - Function only keyboard (6 keys + BAT) SY-71x/75x D - Dual keyboard function and numberic keys (12 keys + BAT) SY-71x/75x B - Big keyboard function, numberic and service keys (24 keys + BAT) SY-780/777 b - Big keyboard function, numberic and service keys (22 keys + BAT) SY-760
24	1	Display type	O/T/R/G	O - Single line display 1x16 T - Double line display 2x16 R - Double line display 2x16 with Russian charecters G - Graphic display (future use)
25	1	Seriel parameters	?	Bits
26	1		?	Parity
27	1		?	Stop bits
28	1		?	Baud rate index
29	1	Terminal ID	0-9 or A-P	
30	1	FPU type	1-3	?
31	1	FPU Terminal mode	M/S/N	M - master S - slave N - FPU not present
32	4	CRC		

21	1	EOT		

For example:08002@26.10.2004.

3.1.6 Get Network Configuration Parameters from Terminal

Returns the current values for technician menu communication parameters.

Location	Length	Value	Desc.
1	1	S	System command
2	1	1-9	Multidrop ID
3	1	H	Hardware configuration
4	1	N	Network
5	3		CRC
8	1	04Hex	EOT

Reply from terminal:

Location	Length	Name	Value	Desc
1	1	System command	S	
2	2	Multidrop ID	*1-9*	
4	1	Hardware configuration	H	
5	1	Network	N	
6	1	Network card	A/B/D/E	
7	1	Polling mode and MAC sending mode	*0-3*	0 - No 1 - Network polling 2 - Network MAC send 3 - Polling mode and Network MAC send
8	2	Polling interval	*n*	in seconds
10	1	Network Protocol	0/1	0 - TCP/IP 1 - UDP

11	12	MAC address	in Hex format *xx.xx.xx.xx.xx.xx*	
23	8	IP address	*xx.xx.xx.xx*	
31	8	Gateway	*xx.xx.xx.xx*	
39	8	IP remote address	*xx.xx.xx.xx*	
47	8	Subnet mask	*xx.xx.xx.xx*	
55	5	Terminal port	port number	
60	5	Remote port	port number	
65	5	Diconnect time		in seconds
70	1	DHCP	0/1	0 - Off 1 - On
71	2	Network card Firmware version		
73	3			CRC
76			04Hex	EOT

3.2 Online communication and data collection

The data collection process uses the following sequences.

3.2.1 Send data

The following sequence designated for a full-data block transmission. A complete sequence results with a sent and acknowledged representation for

all records transmitted to the host.

Host command			**Terminal reply (if there is data)**		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
A (41*h*)			**d (64*h*)**		
	Multidrop ID	1		Multidrop ID	1
	CRC	4		data	up to 128
	EOT	1		CRC	4
				EOT	1
			Go to the F or C command on page page -12 .		
			Terminal reply (if there is no data)		
			n (6E*h*)		
				multidrop ID	1
				CRC	4
				EOT	1
			Terminal reply (if online)		
			q (71*h*)		
				multidrop ID	1
				terminal ID	5
				last transaction (16-N)	
				CRC	4
				EOT	1

Terminal online connection promotes the following sequence interchange between host and terminal:

Host response			**Terminal acknowledgment**		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
Q (51*h*)			**ACK (06*h*)**		
	multidrop ID	1		multidrop ID	1

	allowed/not allowed (Y/N) or Long (L) - (Y/N) and length of date in 2 bytes	1		CRC	4
	message to be displayed (1)	16		EOT	1
	no. of seconds to be displayed 1-9,# (2)				
	reserved for future use	7			
	CRC	4			
	EOT	1			

Note:

1. If # is entered 16 times, the programming tables text definitions will be in use.
2. If # is entered, the programming tables seconds number definitions will be in use.

3.2.2 Send all (data)

The following sequence designated for a partly-full data block transmission. A complete sequence results with a sent and acknowledged representation for

all records transmitted to the host.

Host command			**Terminal reply (if there is data)**		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
B (42*h*)			**d (64*h*)**		
	multidrop ID	1		multidrop ID	1
	CRC	4		data	up to 128
	EOT	1		CRC	4
				EOT	1
			Go to the F or C command on page page -12.		
			Terminal reply (if there is no data)		
			n (6E*h*)		
				multidrop ID	1
				CRC	4

If the terminal is connected online, the host will respond to the terminal's reply as either by short or long message. Short message:

Host response			**Terminal acknowledgment**		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
Q (51*h*)			**ACK (06*h*)**		
	multidrop ID or @	1		multidrop ID	1
	allowed/not allowed (Y/N)	1		CRC	4
	message to be displayed	16		EOT	1
	no. of seconds to be displayed (1-9,#)	1			
	reserved for future use	7			
	CRC	4			
	EOT	1			

Long message:

Host response			Terminal acknowledgment		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
Q (51*h*)		**1**	**ACK (06*h*)**		
	multidrop ID or @	1		multidrop ID	1
	L (4Ch) Long message identification	1			
	Allowed/not allowed (Y/N)	1		CRC	4
	Data string: 99 max.	2		EOT	1
	Data as per length	up to 99			
	CRC	4			
	EOT	1			

3.2.3 Acknowledge last record

The host employs the following sequence to acknowledge receiving the last record from the terminal. A complete sequence results with a sent and acknowledged representation for all records transmitted to the host. The terminal will not clear the acknowledged data from its memory buffer until commanded to do so by the host.

Host response			Terminal acknowledgment		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
F (46*h*)			**ACK (06*h*)**		
	multidrop ID	1		multidrop ID	1
	CRC	4		CRC	4
	EOT	1		EOT	1

3.2.4 Clear buffer

This sequence commands the terminal to clear all data records marked as sent and acknowledged from its memory buffer, including the last data record transmitted but was not acknowledged. A complete sequence results with a

cleared memory buffer (page below).

Host command			Terminal reply		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
C (43h)			**ACK (06h)**		
	multidrop ID	1		multidrop ID	1
	CRC	4		CRC	4
	EOT	1		EOT	1

3.2.5 Clear by date

This sequence commands the terminal to clear data records marked as sent and acknowledged from its memory buffer, according to specified date.

Host command			Terminal reply		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
c (63h)			**ACK (06h)**		
	multidrop ID	1		multidrop ID	1
	Date as DDMMYY format	6			
	CRC	4		CRC	4
	EOT	1		EOT	1

3.2.6 Reset buffer

The sequence commands the terminal to reset all data records marked as sent and acknowledged to unsent records.

Host command			Terminal reply		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
G (47*h*)			**ACK (06h)**		
	multidrop ID	1		multidrop ID	1
	CRC	4			

	EOT	1		CRC	4

3.2.7 Reset line

The sequence commands the terminal to terminate all transmission to the host, initializing the communication settings, and wait for the host to send a SEND DATA or SEND ALL command before resuming transmission. The resumed transmission starts at the terminal with the first data record that has not be acknowledged by the host.

Host command			**Terminal does not reply or acknowledge**
Command	**Data**	**Bytes**	**None**
R (52*h*)			
	Multidrop ID	1	
	CRC	4	
	EOT	1	

3.3 Programming

The following sequences are used to program Synel's terminals.

3.3.1 Halt

This sequence is of terminating normal operation mode and proceeding to programming mode commands. *Programming* is displayed, accepting the terminal inputs are denied. It is essential to invoke terminal's programming mode in order to perform programming procedures.

Host command			**Terminal reply**		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
			If in programming mode		
K (4B*h*)			**ACK (06*h*)**		
	multidrop ID or @	1		multidrop ID	1
	CRC	4		CRC	4
	EOT	1		EOT	1

	When a data input is received (busy)		
	b (62*h*)		
		multidrop ID	1
	R (72*h*)		
		CRC	4
		EOT	1
	The HALT command may be resent		
	Reply if the terminal is being programmed		
	b (62*h*)		
		multidrop ID	1
	C (63*h*)		
		programming operation type (S, D or R)	1
		type of table	1
		table ID	3
		CRC	4
		EOT	1
	Reply if the terminal is processing the last command		
	b (62*h*)		
		multidrop ID	1
	Go to the L command page -12 .		
		CRC	4
		EOT	1

3.3.2 Table(s) Loading

This sequence loads new tables to the terminal. Creating the new tables and preparing them for transmission is performed using a programming tool, refer to software user's manual. It is essential to invoke terminal's programming mode in order to perform the sequence procedures.

Host command			**Terminal reply if free for programming**		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
I (49h)			t (74h)-received		
	multidrop ID or @	1		multidrop ID	1
	table type	1	**W (57h)** if the block was received and stored or **E (45h)** if the table already exists or **F (46h)** if the memory is full or **G (47h)** if a data block was received and processed to be stored at the terminal's memory.		
	table ID	3		CRC	4
S				EOT	1
	* Number of blocks; 03	2			
	Block serial no.	2			
	Block up to 119				
	CRC	4			
	EOT	1			
			Reply if the terminal was turned OFF and then ON, thereby terminating programming mode		
			b (62*h*)		
				multidrop ID	1
			R (72*h*)		

				CRC	4
				EOT	1

Host command			Terminal reply if free for programming		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
	* Number of blocks; 03	2			
	Block serial no.	2			
	Block up to 119				
	CRC	4			
	EOT	1			
			Reply if the terminal was turned OFF and then ON, thereby terminating programming mode		
			b (62*h*)		
				multidrop ID	1
			R (72*h*)		
				CRC	4
				EOT	1

* This is an SN field- a special numeric field for increasing the value range without increasing the number size. Only the most significant byte of the number is changed to a character as per the following algorithm: 10 is represented by ":", 11 ";" etc. according to the standard ASCII table. For example: If the records total in a table is 204, then it will be converted to E4. (See algorithms in Appendix - A).

Host command			Terminal reply if free for programming		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
			Reply if the terminal is being programmed		
			b (62*h*)		
				multidrop ID	1
			C (63*h*)		

	programming operation type (S, D or R)	1
	Table type	1
	Table ID	3
	CRC	4
	EOT	1
Reply if the terminal is processing the last command.		
b (62*h*)		
	multidrop ID	1
Go to the L command page -12 .		
	CRC	4
	EOT	1

3.3.3 Table(s) Deleting

This sequence commands to delete one or more tables from terminal memory.

Host command			**Terminal reply if free for programming**		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
I (49*h*)			t (74*h*)		
	Multidrop ID or *@*	1		Multidrop ID	1
	table type	1	**W (57h)** command was executed **G (47h)** if the command was received and it is in process of being executed		
	table ID	3		CRC	4
D				EOT	1
	CRC	4	Reply if the terminal was turned OFF and then ON, programming mode is terminated. @@@@D - deletes all tables!		
	EOT	1	b (62*h*)		

		multidrop ID	1
	R (72*h*)		
		CRC	4
		EOT	1
	The host may resend the HALT command		
	b (62*h*)		
		Multidrop ID	1
	C (63*h*)		
		Programming operation type (S,D or R)	1
		table type	1
		table ID	3
		CRC	4
		EOT	1
	Reply if the last command was responded		
	b (62*h*)		
		Multidrop ID	1
	Go to the L command on page -12		
		CRC	4
		EOT	1

3.3.4 Table(s) Replacing

This sequence replaces currently deposited terminal's programming tables. Creating the updated tables and preparing them for transmission is performed using a programming tool, refer to software user's manual. It is essential to invoke terminal's programming mode in order to perform the sequence procedures.

Host command			**Terminal reply if free for programming**		
Command	**Data**	**Bytes**	**Command**	**Data**	**bytes**
I (49*h*)			**t (74*h*)**		

	Multidrop ID or @	1		Multidrop ID	1
	table type	1	**W (57*h*)** *if the block was received and stored* or **F (46*h*)** *if the memory is full* or **G (47h)** if the block was received and is in the process of being stored		
	table ID	3		CRC	4
R				EOT	1
	number of blocks (i.e. 03)	2	Reply if the terminal was turned off and then on, thereby terminating the programming mode		
	block serial number	2	b (62*h*)		
	block contents	Up to 119		multidrop ID	1
	CRC	4	R (72*h*)		
	EOT	1		CRC	4
				EOT	1
			The host may resend the HALT command.		
			Reply if the terminal is being programmed		
			b (62*h*)		
	block contents	Up to 119		multidrop ID	1
	CRC	4	R (72*h*)		
	EOT	1		CRC	4
				EOT	1
			The host may resend the HALT command.		
			Reply if the terminal is being programmed		
			b (62*h*)		

	multidrop ID	1
C (63*h*)		
	programming operation type (S,D or R)	1
	table type	1
	table ID	3
	CRC	4
	EOT	1
Reply if the last command was responded		
b (62*h*)		
	multidrop ID	1
Go to the L page -12		

3.3.5 Run

This sequence commands the terminal to terminate the programming mode and proceed to normal operation mode.

Host command			**Terminal reply if it has resumed normal operation**		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
L (4C*h*)			ACK (06*h*)		
	Multidrop ID	1		multidrop ID	1
	CRC	4		CRC	4
	EOT	1		EOT	1
			If a data input was received: “terminal is busy”		
			b (62*h*)		
				multidrop ID	1
			R (72*h*)		
				CRC	4

		EOT	1
	The host may resend the HALT command.		
	Reply if the terminal is being programmed		
	b (62h)		
		multidrop ID	1
	C (63h)		
		programming operation type (S,D or R)	1
		table type	1
		table ID	3
		CRC	4
		EOT	1
	Reply if the last command was responded		
	b (62*h*)		
		multidrop ID	1
	Go to the L command page -12		
		CRC	4
		EOT	1

3.3.6 Command verification

This sequence requests that the terminal will verify a receipt of the last command.

Host command			**Terminal reply**		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
J (4A*h*)			v (76*h*)		
	multidrop ID or @	1		multidrop ID	1
	CRC	4		last command received	1

	EOT	1		was the command clear to the terminal? (Y/N)	1
				CRC	4
				EOT	1

3.4 Special host display

The sequence commands the terminal to display the transmitted message from the host.

Host command			**Terminal reply**		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
H (48h)			ACK (06h)		
	multidrop ID or @	1		multidrop ID	1
	message display interval (seconds) (1-9) or 0 for display until the next message	1		CRC	4
	message	16		EOT	1
	reserved for future use	2			
	CRC	4			
	EOT	1			

3.5 System Commands

3.5.1 Special System Command

This command initiate transmitting the entire terminal's memory to the host. This is performed when dump transmitting is required or after memory crash.

Host command			**Terminal reply**		
Command	**Data**	**Bytes**	**Command**	**Data**	**Bytes**
S (53h)			ACK (06h)		
	multidrop ID or @	1		multidrop ID	1
	C Real time clock tuning			CRC	4
	K dump, transmitting the entire memory as a block of transactions			EOT	1
	x memory crash, transmitting the entire memory after memory crash				
	CRC	4			
	EOT	1			

3.5.2 Save Parameters into Flash

Stores the current values in the actual memory into the into the Flash memory.

Location	**Length**	**Value**	**Description.**
1	1	S	Op-code
2	1		M.ID (according to the ASCII table @=all
3	2	LF	Sub code
5	4		CRC

9	1	04Hex	EOT

The replay is ACK or NACK (in case of invalid value or failure)

3.5.3 Restore parameters from flash

Restores the values from the Flash memory to the actual memory.

Location	**Length**	**Value**	**Description.**
1	1	S	Op-code
2	1		M.ID (according to the ASCII table @=all
3	2	LS	Sub code
5	4		CRC
9	1	04Hex	EOT

The replay is ACK or Nack (in case of invalid value or failure)

4. Fingerprint unit commands

The usage of finger print template deposited within the host computer or a terminal, generated via an additional terminal requires an extended host-terminal protocol. There are common commands for the host and the terminal, distinction is according to the OP code. The commands transmitted by host use uppercase V and commands transmitted by terminal use lowercase v. The scanned finger print image is deposited at the finger print unit memory as a template in a hexadecimal format and stored as 384 bytes. Synel's protocol allows transmission of characters ranging between 20 to 7F (ASCII visible characters), therefore a conversion is required. Type=1 represents the template trasmittion methode, the binary code is converted into ASCII.

The conversion functions executed via the user interface are designated to convert the deposited 384 bytes from a hexadecimal format into a pre-defined Synel's format (768 when packed) The conversion process is concluded with a deposit of a double sized file with a *fg1* suffixes, represented as Template ID.

	Left nibble				Right nibble				Hx	
Binary	0	1	0	0	0	0	1	0		
Hexadecimal calcuation	2^3=8	2^2=4	2^1=0	2^o=0	2^3=8	2^2=0	2^1=2	2^o=0	4	2

A byte is often interpreted as two adjacent nibbles

4.1 Conversion process

A byte is built of two nibbles a left and a right. Each nibble is converted into a byte a 6 or a 3 is added accordingly to nibbles. Converted bytes will range between 60-6F and 30-3F in accordance with nibble, for example: One byte in the hexadecimal format that is represented by the hexadecimal value as left nibble equals 4 and right nibble equals 2 is converted into two bytes in Synel's format 64 and 32. Synel's format is reversed into hexadecimal format, two bytes 64 and 32 are converted into one byte that is represented by 42.

4.1.1 Enrolment conversion procedure

This procedure converts the hexadecimal format into Synel's format:

1. Reading a byte
2. Perform AND operation with F0
3. Perform four times right shift
4. Perform an OR operation with 60
5. Store
6. Read the same byte
7. Perform AND operation with 0F
8. Perform OR operation with 30
9. Store

4.1.2 Verification conversion procedure

This procedure converts the Synel's format into hexadecimal format:

1. Reading a byte (bytes that starts with 6)
2. Perform AND operation with 0F
3. Perform four times left shift
4. Store in temporary byte
5. Read the next byte (bytes that starts with 3)
6. Perform AND operation with 0F
7. Perform OR Operation with the temporary byte
8. Store

4.1.3 Conversion algorithm

Hexadecimal to Synel's format

1. Get original format

									Hx	
0	1	0	0	0	0	1	0		4	2

2. Perform AND F0 operation to clear right nibble

									Hx	
0	1	0	0	0	0	1	0		4	2
1	1	1	1	0	0	0	0		F	0
0	1	0	0	0	0	0	0	Result	4	0

3. Perform Shift right four times; a>>=4

									Hx	
0	1	0	0	0	0	0	0		4	0

									Hx	
0	0	0	0	0	1	0	0	Result	0	4

Perform OR 60 operation; `a |= 0x60`

									Hx	
0	0	0	0	0	1	0	0		0	4
0	1	1	0	0	0	0	0		6	0
0	1	1	0	0	1	0	0	Result	6	4

4. Store byte
5. Get the same byte

									Hx	
0	1	0	0	0	0	1	0		4	2

6. Perform AND 0F operation

									Hx	
0	1	0	0	0	0	1	0		4	2
0	0	0	0	1	1	1	1		0	F
0	0	0	0	0	0	1	0	Result	0	2

7. Perform OR 30 operation; `a |= 0x30`

									Hx	
0	0	0	0	0	0	1	0		0	2
0	0	1	1	0	0	0	0		3	0
0	0	1	1	0	0	1	0	Result	3	2

8. Store byte
1. Get a byte (Starts with 6)

									Hx	
0	1	1	0	0	1	0	0		6	4

2. Perform AND 0F operation

									Hx	
0	1	1	0	0	1	0	0		6	4
0	0	0	0	1	1	1	1		0	F
0	0	0	0	0	1	0	0	Result	0	4

3. Perform Shift left four times; a<<=4

									Hx	
0	0	0	0	0	1	0	0		0	4

									Hx	
0	1	0	0	0	0	0	0		4	0

4. Store byte in a temporary variable
1. Get the next byte (starts with 3)

									Hx	
0	0	1	1	0	0	1	0		3	2

2. Perform AND 0F operation

									Hx	
0	0	1	1	0	0	1	0		3	2
0	0	0	0	1	1	1	1		0	F
0	0	0	0	0	0	1	0	Result	0	2

3. Perform OR

									Hx	
0	0	0	0	0	0	1	0		0	2
0	1	0	0	0	0	0	0		4	0
0	1	0	0	0	0	1	0	Result	4	2

4. Store byte

4.2 General characteristics

The fingertip unit works according to the following characteristics:

- Maximal template ID of 10 digits
- Transmission of identification templates of up to 4K.
- For verification, old as well as new commands can be used as long as the templates are 8 digit ID.

Fingertip units can be set to either Slave or Master mode:

Slave mode - The default terminal mode.

Master mode - A mode advanced to the default terminal mode with the following additions:

- If an unidentified finger template is enrolled a message is sent to the host computer with the new finger template. See “New Template” on page -50.
- If an unidentified finger template enrolled to matched to a known code at Identification mode, an “A” request is sent to the Host computer asking for the template. See “The Request template” on page -46.

4.3 Commands

As of firmware version 6.200 Synel terminals supports a 10 digit fingerprint template, thus upgrading communication to identification (no card needed) level as well as verification (previously).

Type1 -

- Is supported by 8 digit and 10 digit ID templates.
- Verifies with 348 bytes, identifies with 2352 bytes and higher.

FPU -

- Is supported by 10 digit ID templates.
- Verifies with 384 bytes.

For further information refer to “-Host-Terminal Communication” on page -99.

8 digit ID's

- Is used for backward compatibility and is not recommended for use otherwise.
- Supports Type1 fingerprints templates.
- Uses a 8 digit template id.
- Commander are in number form.

10 digit ID's

- Is the recommended template for use.
- Supports Type1 ad FPU-S fingerprints templates.
- Uses a 10 digit template id.
- Commander are in alphabetical form.

4.3.1 The Request template

This sequence requests a transmission of a fingerprint template according to ID number.

Command			**Reply**
Sequence	**String/size**	**Position in string**	
OP CODE	"V" " / "v"	1	***Get template*** See page -47 or Not available See page -58
ID	0-31	2	
CODE	"1"	3	
INDEX	"0 -9"	4	
Template ID		5-13	
CRC		13-16	
EOT	04 Hex	17	

A 10 digit ID template

This sequence requests a transmission of a fingerprint template according to ID number.

Command			**Reply**
Sequence	**String/size**	**Position in string**	***Get template*** See page -47 or **Not available See page -58**
OP CODE	“V" " / "v"	1	
ID	0-31	2	
CODE	"A”	3	
INDEX	"0 -9“	4	
Template ID		5-15	
CRC		15-18	
EOT	04 Hex	19	

4.3.2 Get template

An 8 digit ID verification template (348)

The sequence commence a transfer of a fingerprint template from terminal to the host computer or vice versa from host computer to terminal.

Command			**Reply**
Sequence	**String/size**	**Position in string**	
OP CODE	"V" / "v"	1	ACK- The template has been received. See page -58 Not available- Indication of the reason the template was not received. If no template is stored in the terminal a 105 UFPU_NOT_FOUND error is returned.
ID	0-31	2	
CODE	“2”	3	
TYPE	“1”	4	
Template ID		5-12	
TEMPLATE (BII)		13-709	
CRC		710-713	
EOT		714	

Type-	transmission method
1-	Binary code expanded for ASCII, each byte is divided into 2 nibbles. 6 is added to the higher value nibble and 3 to the lower value nibble and transmitted as 2 bytes.
Template Structure-	As per BII definitions. Length after expansion is 800 bytes.
ID template-	Will be stored in the PC as a file name.

A 10 digit ID template verification- 384 tø 348 x one

Command			**Reply**
Sequence	**String/size**	**Position in string**	ACKCurrentBLOCK- The specified block number was received. The next one can be sent. See page -49.
OP CODE	"V" / "v"	1	
ID	0-31	2	
CODE	“B”	3	
TYPE	“1” (packed)	4	
Template ID		5-14	
Number of blocks	01-99	15-16	
Current block	01-99	17-18	
Data size in block	768 (384 converted to ASCI) or 696 (348 converted to ASCI)	19-21	
TEMPLATE (BII)		22-718	
CRC		719-722	
EOT		723	

A 10 digit ID template identification- 2352 (or higher) x 8 blocks

Command			Reply
Sequence	**String/size**	**Position in string**	ACKCurrentBLOCK- The specified block number was received. The next one can be sent. See page -49.
OP CODE	"V" / "v"	1	
ID	0-31	2	
CODE	“B”	3	
TYPE	“1”	4	
Template ID		5-14	
Number of blocks	01-99	15-16	
Current block	01-99	17-18	
Data size in block		19-21	
TEMPLATE (BII)		22-610	
CRC		611-614	
EOT		615	

For example: V1B100000000010101768...

Long templates (identification)- After transmission of the last block the terminal will send an additional reply to inform the PC if the template was successfully transmitted to the PC. If transmission was unsuccessful it will also specify failure reason.

ACKCurrentBLOCK - reply for 10 ID verification+identification

Command			Reply
Sequence	**String/size**	**Position in string**	
OP CODE	"V" / "v"	1	
ID	0-31	2	

CODE	“b” (acknowledgment of reciving block)	3	ACKCurrentBLOCK- The specified block number was received. The next one can be sent. For the long template the terminal acknowledges the template’s successful/ unsuccessful receival after the last templates has been transmitted
TYPE	“0” (not packed)	4	
Template ID		5-14	
Number of blocks	01-99	15-16	
Current block	01-99	17-18	
CRC		19-22	
EOT		23	

For example: v1b0000000000101016342

4.3.3 New Template

This sequence transmits a notification that a new template was created at the terminal.

This Command is only available for terminals in Master mode.

The terminal does not expect or wait for a reply for this command from the Host computer.

8 digit ID

Command			**Reply**
Sequence	**String/size**	**Position**	
OP CODE	"V" / "v"	1	
ID	0-31	2	
CODE	“3”	3	
TYPE	“0”	4	
Template ID		5-12	
CRC		13-16	**ACK** notification was accepted. See page -58 or Request for template, transmit the template. See page -46
EOT	HEX 04	17	

10 digit ID

Command			**Reply**
Sequence	**String/size**	**Position in string**	
OP CODE	"V" / "v"	1	**ACK** notification was accepted. See page -58 or Request for template, transmit the template. See page -46
ID	0-31	2	
CODE	“C”	3	
TYPE	“0”	4	
Template ID		5-14	
CRC		15-18	
EOT	HEX 04	19	

4.3.4 Status Type

The FPU type parameter specifies the type of fingerprint command used as Verify, Identify, Universal and Universal for FPU-S version 35xx

The following is the structure of the **Request from the Host to the Terminal**

Name	**Position in string**	**Size**	**Value**
OP CODE	1	1	'V'
ID	2	1	0-9 A-P or @ all
CODE	3	1	'M'
TYPE	4	1	'0'
CRC	5-8	4	CRC
EOT	9	1	EOT

Reply from the Terminal to the Host

Name	Position in string	Size	Value
OP CODE	1	1	'v'
ID	2	1	0-9 A-P or @ all Terminal ID
CODE	3	1	'M'
TYPE	4	1	'0'
FPU Type	5	1	Comparison- 'V'- verify, 'I' – Identify 'U'-Universal 'S' – FPU-S for version 35xx
FPU firmware version	6-17	12	FPU firmware version
Template quantity	18-22	5	Number of templates
	23	1	'/'
Template space	24-28	5	Maximum available space for templates
FPU Terminal Mode	29	1	'M'- master 'S'- slave
Global Threshold level	30	1	'0' -Very Low '1' - Low '2' - Medium '3' - High '4' - Very High
Enroll Mode for version 35xx	31	1	'0' - One Time '1' - Two Times 'A' – Two Templates
CRC	32-35	4	CRC
EOT	36	4	EOT

For example:

"v1M0SB16C0603220000079/09090S00"[CRC],[EOT]

where:

S – FPU Type;

4.3.5 Set FPU Parameters/ Enrol Mode

This command us used to set or enroll FPU parameters to the terminals. Enrollment modes are set at production. The following enrollment methods are available.

One time – Enrolls a scanned fingerprint to a fingerprint template.

Two times – Scans the fingerprint twice. The two images are compared to each other and verified. If the fingerprint images do not match the user is rejected. If they match the better quality fingerprint is enrolled in the template.

Two templates – Scans the user finger twice and saves the each fingerprint scan as a template. The user has two templates stored. Then, each time the user is verified, the module can decide whether to replace the existing template with a new one. This update reflects the dynamic changes in the skin of the user's finger.

	One time	Two times	Two templates
Scan	1	2	2
Save template	1	1	2
Modification	-	-	+

The enrollment mode used is set as follows:

0 – One Time;

1 – Two Times;

A – Two Templates

The following is the structure for setting FPU parameters **from the Host to the Terminal**

Name	Position in string	Size	Value
OP code	1	1	'V'
ID	2	1	0-9 A-P or @ all
Sub CODE	3	1	'H'
Type	4	1	'0'
Parameter ID	5	1	'T' - Threshold 'F' - FPU mode 'E' - Enroll mode

Parameter ID Data	6	10	See Parameter ID Data table below
CRC	6	4	CRC
EOT	10	1	EOT

For example:

V1H0**E0**[CRC],[EOT]

Data Values for Threshold

Value must be entered is 10 spaces.

Value	**Desc.**
1-VERYHIGH	Very high
2-HIGH----	High
3-MEDIUM--	Medium
4-LOW-----	Low
5-VERYLOW-	Very low

Data Values for FPU Mode

Value must be entered is 10 spaces.

Value	**Desc.**
MASTERMODE	Master mode
SLAVE-MODE	Slave mode

Enroll Mode

For additional information see explanation above

Value must be entered is 10 spaces.

Value	**Desc.**
1-Template	**1** – Two Times
0-Template	**0** – One Time;
A-Template	**A** – Two Templates

Reply from the Terminal

The following is the structure of the **Reply from the Terminal to the Host**

Name	Position in string	Size	Value
OP CODE	1	1	'v'
ID	2	1	0-9 A-P or @ all Terminal ID
CODE	3	1	'M'
TYPE	4	1	'0'
FPU Type	5	1	Comparison- 'V'- verify, 'I' – Identify 'U'-Universal 'S' – FPU-S version 35xx - NEW
FPU firmware version	6-17	12	FPU firmware version
Template quantity	18-22	5	Number of templates
	23	1	'/'
Template space	24-28	5	Maximum available space for templates
FPU Terminal Mode	29	1	'M'- master 'S'- slave
Global Threshold level	30	1	'0' -Very Low '1' - Low '2' - Medium '3' - High '4' - Very High
Enroll Mode	**31**	**1**	**'0' - One Time** **'1' - Two Times** **'A' – Two Templates**
CRC	32-35	4	CRC
EOT	36	4	EOT

For example:
"v1M0SB16C0603220000079/09090S00"[CRC],[EOT]
where:
0 - "EnrollMode" – 1 Time to one Template;

4.3.6 BlockTemplateList

The PC requests the template list in the FP unit.
When sending a template you can send a string of up to 50 blocks of data at a time.
The blocks are sent indicating the following information:

- The number of spaces the block uses.
- The current block number in the sequence of blocks being sent.
- The total number of blocks in the sequence. If unknown 00000.

Command			**Reply**
Sequence	**String/size**	**Position in string**	**ACK** list. or Not available. See page -58 * 00000 indicates that the number of blockes is not known.
OP CODE	"V" / "v"	1	
ID	0-31	2	
CODE	“L”	3	
TYPE	“0”	4	
Size of the block	1-50	5-6	
Current block number in sequence	00000-99999	7-11	
Total number of blocks	00000*-99999	12-16	
CRC		17-21	
EOT	HEX 04	22	

For example: V1L02500001<785CRCEOT

Transmitting the template list from the FPU. Can contain up to 4000 items. Each ID consists of 10 bytes. Therefore, will be sent in blocks, each = 1024 bits.

Command			**Reply**
Sequence	**String/size**	**Position in string**	**ACK** list. or Not available. See page -58. Data format depends on the defined block size. * 00000 indicates that the number of blockes is not known.
OP CODE	"v"	1	
ID	0-31	2	
CODE	“L”	3	
TYPE	“0”	4	
Size of block	1-50	5-6	
Current block number in sequence	00000-99999	7-11	
Total number of blocks	00000*-99999	12-16	
Data	Variable		
CRC	Variable		
EOT	HEX 04		

For example: v1L0250000100101...(first 25 template blocks)...CRCEOT

4.3.7 Acknowledge

Command			**Reply**
Sequence	**String/size**	**Position in string**	Indicates whether the template was received/ deleted as per the specific command.
ACK	06 Hex	1	
ID	0-31	2	
CRC		3-6	
EOT	04 Hex	7	

4.3.8 Not available

This sequence triggers, accepting command was denied, detailing the cause according to NNN the FPU values.

8 Digit ID

Command			**Reply**
Sequence	**String/size**	**Position in string**	**NNN** , according to the FPU's return values
OP CODE	"V"/ "v"	1	
ID	0-31	2	
CODE	"6"	3	
TYPE	"0"	4	
Reason	NNN Error code See "Error Code List (NNN)" on page 64.	5-7	
CRC		8-11	
EOT	04 Hex	12	

10 Digit Template

Command			**Reply**
Sequence	**String/size**	**Position in string**	**NNN** , according to the FPU's return values
OP CODE	"V"/ "v"	1	
ID	0-31	2	
CODE	"F"	3	
TYPE	"0"	4	
Reason	NNN Error code See "Error Code List (NNN)" on page 64.	5-7	
CRC		8-11	
EOT	04 Hex	12	

4.3.9 Delete template

An 8 digit ID FP template

Command			**Reply**
Sequence	**String/size**	**Position in string**	ACK - Record was deleted! Not available- impossible to delete record.When the ID is @@@@@@@@ all templates will be deleted.
OP CODE	"V"/ "v"	1	
ID	0-31	2	
CODE	"7"	3	
TYPE	"0"	4	
Template ID		5-12	
CRC		13-16	
EOT	04 Hex	17	

A 10 digit ID FP template

Command			Reply
Sequence	**String/size**	**Position in string**	ACK - Record was deleted! Not available- impossible to delete record. When the ID is @@@@@@@@@@ all templates will be deleted.
OP CODE	“V"/ "v"	1	
ID	0-31	2	
CODE	"G”	3	
TYPE	"0”	4	
Template ID		5-14	
CRC		15-18	
EOT	04 Hex	19	

4.4 Set FPU Parameters

For setting a new FPU - Set FPU Parameters see “Set FPU Parameters/ Enrol Mode” on page -53.

4.4.1 SynGetFPUStatus

Command			Reply
Sequence	**String/size**	**Position in string**	
OP CODE	“V"	1	“v”
ID	0-P or @	2	0-P or @
CODE	"M”	3	“M”
TYPE	"0”	4	“0”
		5	Comparison- “U”-Universal “V”- verify, “I” - Identify
		6-17	Kernel version- 12 bytes
		18-22	TemplatesNow- 5 bytes
		23	Slash- 1 byte
		24-28	Max number of Templates - 5 bytes

		29	TerminalModeFPU- “M”- master, “S”- slave- 1 byte
		30	GlobalThreshold- 1/2/3/4/5 - 1 byte From 1= very high - 5 = very low
CRC	4	30-34	
EOT	04 Hex	35	

For example:

Sent - V1M0>=?4EOT (9)
Reply - v1M0UB15C0511110002520/09090S27809EOT

4.4.2 Check Template

Command			**Reply**
Sequence	**String/size**	**Position in string**	
OP CODE	“V"	1	“v”
ID	0-31	2	
CODE	"K”	3	“K”
Index	0-9 or “-”= All	4	0-9 or “-”= All
Template ID	10 char	5	10 char
		14	0- Invalid template 1- Valid template
CRC	4	15-18	
EOT	04 Hex	19	

4.4.3 Fingerprint Parameters

An 8 digit ID FP template

Command			**Reply**
Sequence	**String/size**	**Position in string**	ACK - Record was received! Not available- impossible to receive record.
OP CODE	“V"/ "v"	1	
ID	0-31	2	
CODE	"8”	3	
TYPE	"0”	4	
Parameter ID	*	5	
Data	*	6-16	
CRC		17-20	
EOT	04 Hex	21	

A 10 digit ID FP template

Command			**Reply**
Sequence	**String/size**	**Position in string**	ACK - Record was received! Not available- impossible to receive record.
OP CODE	“V"/ "v"	1	
ID	0-31	2	
CODE	"H”	3	
TYPE	"0”	4	
Parameter ID	The threshhold level as either **T** or **F** as in the **Threshhold Level Parameters** table below.	5	

Data	The data from the data column of the	6-16
CRC		17-20
EOT	04 Hex	21

Threshold Level Parameters

Parameters			
HEX	**ASCII**	**Data**	**Description**
“T”	54	“1-VERYHIGH”	
“T”	54	“2-HIGH----”	
“T”	54	“3-MEDIUM--”	
“T”	54	“4-LOW-----”	
“T”	54	“5-VERYLOW-”	
“F”	46	MASTERMODE	FPU mode for Master
“F”	46	SLAVE-MODE	FPU mode for Slave

For example: V1H0T2-HIGH----CRCEOT or V1H0FMASTERMODECRCEOT

Note: The text displayed in the terminal prompt must be identical to the text specified in the above data column (spacing-wise, case-wise).

Note: A 10 space section of the string is available for the parameters. In cases where the Data does not use up the 10 spaces dashes (-) are used to hold the space. See example above.

4.4.4 Error Code List (NNN)

No.	Error message	Description
001	Unknown ID	The command ID was not recognized.
002	Invalid Checksum	The checksum was not valid.
003	Invalid Length	The packet length was invalid (i.e. < 4).
004	Unit Busy	The command could not be placed on the command queue because either the command is non queueable or there are not enough resources available.
005	Unknown	The cause is unknown.
006	Download Time-out	A time out occurred during the download of raw data.
007	Invalid Baud	The request baud rate is not supported.
008	Invalid Stop Bit	The requested stop bit setting is not supported.
009	Invalid net ID	The requested net ID is out of range.
010	Invalid sensor param	One or more of the requested sensor parameters are out of range.
011	Invalid template ID	The specified template ID is not valid.
012	Invalid template Index	The specified template index does not exist for the specified template ID.
013	Error writing data	There was a general error writing to the non volatile storage.
014	Storage exhausted	There is not sufficient available space in non volatile memory to perform the operation.
015	Invalid thresh	The requested threshold setting is out of range.
016	No finger detected	Expected a finger on the sensor and none was detected.
017	Error reading data	Could not read the specified data from flash memory.
018	Error transferring list	A list sent to the device is to large.

No.	Error message	Description
019	Wiegand ID Exist	A Wiegand card is already assigned a different use
020	Table Full	Current table is full
021	MEM allocation error	Error allocating memory on the unit.
022	Invalid template type	Template type does not match current unit configuration.
023	Time out error	Generic time out error.
024	Invalid option	A parameter was specified that is not supported.
025	Max index	The maximum template index of 255 was exceeded.
026	Invalid data size	An invalid data size was specified.
027	Updatingtemplate Err	Unable to updated templates to new format.
028	Admin. level differs	Mismatch with the current template admin level and those stored under the same ID.
029	Invalid number of bits	The number of bits for Wiegand in or out is invalid.
030	Invalid Wiegand TPI	The Wiegand out TPI time is invalid.
031	Invalid Wiegand TPW	The Wiegand out TPW time is invalid.
032	Invalid Wiegand format	The requested Wiegand format is invalid.
033	Invalid sensor	Command not appropriate for current sensor type.
034	Invalid key	Key is not correct. Used for lockable functions.
035	Error input locked	The requested Wiegand format combination is not allowable.
036	Error output locked	The input option is locked, for Wiegand structure (output format).
037	Invalid Wiegand format combination	The requested Wiegand Format combination is not allowable.
038	Invalid program	The downloaded loader file was not valid for the hardware.
039	Image acquire error	Error acquiring image from the sensor.

No.	Error message	Description
040	Port locked	Port is locked.
041	Port is locked need password	Port is locked need to supply password to unlock the port.
042	Error operation failed	The requested operation was not completed.
043	Error finger on sensor	A finger was detected on the sensor but the requested operation requires that a finger not be on the sensor.
044	Error PIC does not support.	The PIC version does not support the requested option.
045	Error not a searching template	The sent template format is not supported, only searching templates can be used.
046	Error not a 1 to 1 template	The sent template format is not supported, only 1 to 1 templates can be used.
047	Error not a valid security level	Invalid template security level 1.
049	Error invalid enroll type	An invalid enroll wiegand type was received.
050	Error invalid delete type	An invalid delete wiegand type was received.
71	Error FPU answer erase templates 1	Template no. 1 can not be deleted from FPU memory.
72	Error FPU answer erase templates 2	Template no. 2 can not be deleted from FPU memory.
73	Error FPU answer erase templates 3	Template no. 3 can not be deleted from FPU memory.
74	Error FPU answer erase templates 4	Template no. 4 can not be deleted from FPU memory.
75	Error FPU answer download template	Template can not be downloaded from FPU memory.
76	Error FPU answer unknown command	The command is not recognized by the FPU.
92	Terminal no master mode	The terminal is not set to master mode.

No.	Error message	Description
93	Sent request for template to host	The terminal is in master mode, if template was not found in FPU it sends a request to locate the template at the host.
94	Error operation	Firmware is not operational
95	FPU no response	No response from FPU after time-out and number of retries
96	FPU busy	FPU is engaged with another process
97	BAD checksum	The terminal detected a non valid checksum.
99	BAD syncword	Incorrect SYncomm message
100	Error template block	Requested block does not exist

UniFinger Error Code:

No.	Error message	Description
97	UFPU_SUCCESS	Process is successfully completed.
98	UFPU_SCAN_SUCCESS	Fingerprint input has succeeded.
99	UFPU_SCAN_FAIL	Sensor or fingerprint input has failed.
105	UFPU_NOT_FOUND	There is no requested data found.
106	UFPU_NOT_MATCH	Fingerprint does not match.
107	UFPU_TRY_AGAIN	Fingerprint image is not good.
108	UFPU_TIME_OUT	Timeout for fingerprint input.
109	UFPU_MEM_FULL	Maximum template capacity exceeded.
110	UFPU_EXIST_ID	The requested user ID has been found.
113	UFPU_ADD_NEW	Adding more fingerprints to a current existing user ID.
114	UFPU_FINGER_LIMIT	The number of fingerprint templates per user ID has exceeded its limit.
116	UFPU_CONTINUE	There is more data to be sent.
117	UFPU_UNSUPPORTED	The command is not supported.
118	UFPU_INVALID_ID	The requested user ID is invalid or missing.

121	UFPU_AUTO_ID	Automatically assign user ID in enrollment.
122	UFPU_TIME_OUT_MATCH	Timeout for matching in identification.

UniFinger 3xxx Error Code:

No.	Error message	Description
128	UFPU_BUSY	Module is processing another command.
129	UFPU_CANCELED	The command is canceled.
130	UFPU_DATA_ERROR	The checksum of a data packet is incorrect.
131	UFPU_DATA_OK	The checksum of a data packet is correct.
132	UFPU_EXIST_FINGER	The finger is already enrolled.
132	UFPU_DURESS_FINGER	A duress finger is detected.

5. Firmware upgrade commands

These commands are based on the SDCPII. To simplify loading use LRC instead of CRC.

5.1 Commands

5.1.1 SwitchToFirmwareUpdateMode

The terminal now relies on its Boot Loader flash while its Main flash receives new firmware blocks.

Command			**Reply**
Sequence	**String/size**	**Position in string**	See paragraphs 1-3 below.
OP CODE	“P"	1	
ID	0-31	2	
CODE	"1”	3	
CRC		4-7	
EOT	HEX 04	8	

Replies:

1. Busy: The terminal is currently receiving data.

Sequence	String/size	Position in string
OP CODE	“P"	1
ID	0-31	2
CODE	"1”	3
Busy	B	4
CRC		5-8
CRC	HEX 04	9

2. MemoryNotEmpty=D - There is data in the terminal’s memory.

Sequence	String/size	Position in string
OP CODE	“P"	1
ID	0-31	2
CODE	"1”	3
Data	D	4
CRC		5-8
CRC	HEX 04	9

3. Ack = The command was received, the terminal is currently loading the firmware.

Sequence	String/size	Position in string
OP CODE	“P"	1
ID	0-31	2
CODE	"1”	3
Data	D	4
LRC		5-6
CRC	HEX 06	7

Note: The terminal will reply only after switching to the loading software. In ACK reply the transition will be to LRC.

5.1.2 SendFirmwareBlock

The terminal downloads new firmware from host and burns it into its flash memory. A transmission procedure in up to 512 bit data blocks. A block must be sequential! Do not send an additional block if there is no reply to the previous block (with an exception in cases of retries).

Sequence	**String/size**	**Position in string**
OP CODE	“P"	1
ID	0-31	2
CODE	"2”	3
Type	F/B/A/E	4
Page	0-7	5
Offset	08000-0bfff	6-10
Length	0001-0512	11-14
Data		15-527
LRC		528-529
EOT	04 HEX	530

Types are as follows:

F- Flash- loading of main software

B- Boot- opening loading siftware

E- End- The ending of each block will be marked accordingly.

A- Peripherals (burning of additional components).

Replies:

1. Busy=D - If the terminal is still burning the previous block. The terminal should send the sequential block.

Sequence	**String/size**	**Position in string**
OP CODE	“P"	1
ID	0-31	2
CODE	"2”	3
Busy	B	4

Type	F/B/A/E	5
Page	0-7	6
Offset	08000-0bfff	7-11
LRC		12-13
EOT	04 HEX	14

The PC will send the following block to the block specified in this reply.

2. Error burning the previous block-
   When the terminal can not write the data as it was sent, it will let the host know the address. Updating procedure will be stopped.

Sequence	**String/size**	**Position in string**
OP CODE	“P"	1
ID	0-31	2
CODE	"2”	3
Data	W	4
Type	F/B/A/E	5
Page	0-7	6
Offset	08000-0bfff	7-11
LRC		12-13
EOT	04 HEX	14

3. ACK- the block was received and burnt-
   After receiving this reply, the host can send another block!

Sequence	**String/size**	**Position in string**
OP CODE	“P"	1
ID	0-31	2
CODE	"2”	3
ACK	Hex 06	4
Type	F/B/A/E	5
Page	0-7	6
Offset	08000-0bfff	7-11
LRC		12-13
EOT	04 HEX	14

5.1.3 GetFirmwareBlock

The procedure during which the terminal requests the firmware from the terminal. The host will set the pages and bit amount.

Sequence	String/size	Position in string
OP CODE	“P"	1
ID	0-31	2
CODE	"3”	3
Type	F/B/A/E	4
Page	0-7	5
Offset	08000-0bfff	6-11
Length	0001-0512	12-15
LRC		16-17
EOT	04 HEX	18

Replies:

1. Busy- If the terminal is still occupied with a previous command.

Sequence	String/size	Position in string
OP CODE	“P"	1
ID	0-31	2
CODE	"3”	3
Busy	B	4
Type	F/B/A/E	5
Page	0-7	6
Offset	08000-0bfff	7-11
LRC		12-13
EOT	04 HEX	14

The host will wait and resend the command

2. The required block-
The terminal will send the required bits, if there is a data overflow it

will adjust page size. The host will create a file from the received blocks in a format that will allow transmission to another terminal.

Sequence	**String/size**	**Position in string**
OP CODE	“P"	1
ID	0-31	2
CODE	"3”	3
Type	F/B/A/E	4
Page	0-7	5
Offset	08000-0bfff	6-10
Length	0001-0512	11-14
Data		15-527
LRC		528-529
EOT	04 HEX	530

3. NACK, Error defining memory parameters (wrong page or offset)- If the terminal receives a request for memory content for an illegal area it will reply NACK.

Sequence	**String/size**	**Position in string**
OP CODE	“P"	1
ID	0-31	2
CODE	"3”	3
ACK	Hex 15	4
Type	F/B/A/E	5
Page	0-7	6
Offset	08000-0bfff	7-11
LRC		12-13
EOT	04 HEX	14

5.1.4 SwitchToWorkingMode

The terminal no longer relies on its Boot Loader flash but on its Main flash memory.

Sequence	String/size	Position in string
OP CODE	“P"	1
ID	0-31	2
CODE	"4”	3
LRC		4-5
EOT	Hex 04	6

Replies:

1. Busy- the host is still burning

Sequence	String/size	Position in string
OP CODE	“P"	1
ID	0-31	2
CODE	"4”	3
Busy	B	4
Type	F/B/A/E	5
Page	0-7	6
Offset	08000-0bfff	7-11
LRC		12-13
EOT	04 HEX	14

2. NoFirmware- no software was loaded-
   If the terminal was instructed to clear memory and/or started loading, but did not receive a type E (End) block, it will not run commands

other than for normal mode.

Sequence	String/size	Position in string
OP CODE	“P"	1
ID	0-31	2
CODE	"4”	3
Busy	N	4
Type	F/B/A/E	5
Page	0-7	6
Offset	08000-0bfff	7-11
LRC		12-13
EOT	04 HEX	14

3. ACK, the transmission was performed-
The terminal will clear the memory and prepare the parameters, so it will immediately revert to NoProg mode.

Sequence	String/size	Position in string
OP CODE	“P"	1
ID	0-31	2
CODE	"4”	3
ACK	Hex 06	4
CRC		5-7
EOT	Hex 04	8

5.1.5 ClearMainFlash

This command is used for clearing the memory before loading.

Sequence	String/size	Position in string
OP CODE	“P"	1
ID	0-31	2
CODE	"5”	3

LRC		4-5
EOT	Hex 04	6

Reply:

ACK- Memory cleared.

Sequence	**String/size**	**Position in string**
OP CODE	“p"	1
ID	0-31	2
CODE	"5”	3
ACK	Hex 06	4
LRC		5-6
EOT	Hex 04	7

6. Glossary

Command
A transmission from the host to the station.

Reply
A transmission from the station to the host.

Data collection routine or sequence
An action or series of actions performed woth an S-120 station in which data, or several pieces of data, are entered into the station.

Transaction or Report
The data received from a data collection routine.

Data record or Record
A group of reports which cannot exceed 128 characters (bytes).

Full record
As data is entered into the station it is entered into a RECORD. Eventually, the size of the RECORD will reach 128 bytes. When the size of the RECORD is such that the next REPORT will make the size of the RECORD exceed 128 bytes, the station will close the RECORD and place the next REPORT in a new RECORD. The RECORD which was closed is a FULL RECORD. It is also known as a STORAGE BUFFER.
For example:
A RECORD has been filled to 115 bytes and the next REPORT which is entered into the station contains 20 bytes. Since this TRANSACTION will cause the open RECORD to contain 135 bytes, thereby exceeding the 128 byte maximum, the station automatically closes the open RECORD at 115 bytes, making it a FULL RECORD, and places the 20 byte TRANSACTION into a new open RECORD.
Thus, FULL RECORDS can be of various lengths less than or equal to 128 bytes.

6.1 Data Records Status:

Ready to send or Unsent
A RECORD which has not been transmitted to the host.

Sent and Unacknowledged
A RECORD which has been sent to the host, but has not been acknowledged as received by the host.

Sent and Acknowledged
A RECORD which has been received by the host, and has been acknowledged as received by the host.

Cleared
A record which has been wiped clean from the buffer memory of the station. Only SENT AND ACKNOWLEDGED RECORDS may be cleared.

Multidrop I.D.
A character from 0 to F in ASCII table (30 hex - 3F hex) which designates a station's location in a multidrop connection.
@=all

Station I.D.
A 5 digit number which is burnt into the station's programming.

Appendix A - SY-7xx/SY4xx - RDY Format

A.1 General

Synel's terminal has a special format for internal tables (files). The table contains a header that determines its general information and structure, and data records. The table can be divided into two groups: Tables that are handled directly by the firmware and user defined tables. The first group – System tables has a pre-defined structure that cannot be changed by the user or the application. The user uses the SAL application to adjust the tables to his requirements.

This document will describe the general structure of the tables and the structure of system tables.

Special characters:

Character	**Limitation**
> 7F	A character greater than 7F hex can not be sent to the terminal.
EOT	04 hex – End of Transmission
ACK	06 hex - Acknowledge
CR	OD hex will be ignored – not sent to the terminal
LF	OA hex will be ignored – not sent to the terminal
__	5F hex will be ignored – not sent to the terminal
A string beginning with f1 (01 hex) will be considered as a comment and will not be sent to the terminal.	

A.2 Header structure

Each table has a header. The header is a 23 bytes string divided into the following fields:

A.2.1 Table A – Header structure

Byte	Length	Value	Type	Explanation
1	1	A-z	A&N	Table type – Used as part of file identification
2	3	001-999	N	Table ID – for file identification
5	5	00023-99999	N	Total table characters
10	1	0-z	A&N	Table version
11	2	23	N	Header Size
13	2	00-99	N	Record size - total number of characters in a record
15	3	000-|99	SN	Number of records in table
18	2	00-99	N	Key length
20	2	00	N	Key offset (a fixed value - not in use).
22	2	00-03	N	00 – Not sorted, not packed 01 – Not sorted, packed (Only in numeric records!) 02 – Sorted, not packed 03 – Sorted, packed

A&N- Numbers and characters are allowed in this field.

N- Numeric field

SN- Special Numeric field. This format is used in order to increase the value range in a numeric field without increasing the size of the field for compatibility. The algorithm is simple, there is no change in the least significant bytes of the number, and only the most significant byte of the number is changed to a character according to the following algorithm:
10 is represented by ":", 11 ";" etc. according to the standard ASCII table.
For example: If the records total in a table is 2049, then it will be converted to D49. (See "Algorithm for Synel's numeric fields" on page -95).

In a sorted table the records are in incremental order as per the defined field key. The sorted table search is a binary search, therefore there must not be a duplicate of the same record with the same key. In such cases add a digit to the key to make it unique.

A pack table must consist of numeric characters only. Compression is simple, the terminal will store 2 digits in one byte by converting ASCII into BCD. The compression ratio is 1:2. There will be no data record merging. In odd record length the last nibble will be empty.

A.3 System tables

System tables are handled directly by the firmware. The structure of the table is constant and cannot be changed, however the number of records is not constant. There are 5 System tables: Task Scheduler Table (FTS), Messages table (MPL), System Parameters Table (SYS) and Program tables like JPL, JPR, TRS/TRP Font translation table files (FNT).

A.3.1 Table B- Header of system tables

Field	FNT	FTS	SYS	MPL	JPL	JPR	TRS*	TRP*
Table type	g	e	p	d	j	v	m	m
Table ID	001	001	001	995	001	001	001	001
Total number of characters in table**	-----	-----	-----	-----	-----	-----	-----	-----
Version of table	A	A	A	A	A	A	A	A
Header Size	23	23	23	23	23	23	23	23
Record size	16	23		68	16	---	04	06
Number of records in table***					---	---	---	---
The key length	01	04	00	04	04	00	00	02
Key offset (a fixed value - not in use).	00	00	00	00	00	00	00	00
Table attributes:	02	02	00	02	02	00	02	02

* Either TRS or TRP files will be sent.

** A 5 digit numeric field

*** A 3 special numeric field: 1^{st} =ASCII, 2^{nd}+3^{rd}==numeric (see Appendix A)

Task Scheduler table

This table is used by the firmware to perform automatic operations per a specific day and time. There are 3 types of operations: Changing the active function, activating the relay (for bell purposes or door control) and setting the modem to auto-answer.

Task Scheduler record format

The record is 23 bytes long. The data field is modified according to the

operation code. See the record format in the page below:

Byte	Length	Value	Type	Explanation
1	4	0000-2359	Time	Event time in military format
5	15		A&N	Data depends on the operation type
20	2	00	N	Currently not in use
22	1	K, O, M	A&N	Device type K- Key function, O-Output, M-Modem
23	1	0-7	N	Day of week, 0- all week, 1- Sunday, 7-Saturday

Data field structure for function key operation:

Each function can have the following operation modes:

A Active: The function can be activated by pressing the function key

P Passive: The function is disabled in the defined interval.

D Default: The terminal will return to this function.

Each byte in this field represents a function key that can be an actual or a virtual key. The 1st byte is dedicated to the IN function, the 2nd to OUT, etc. In cases of more than one default key, after using non-default keys (as per user's requirements) the firmware will revert to the first key defined as default.

Byte	Length	Value	Type	Explanation
1	4	0000-2359	Time	Event time in military format
5	15	A, D, P	A&N	A- Active, D- Default, P –Passive
20	2	00	N	Not in use
22	1	K	A&N	K- Key function
23	1	0-7	N	Day of week, 0- all week, 1- Sunday, 7-Saturday

For Example:

Switch automatically to the OUT key during the whole week at 17:15 and

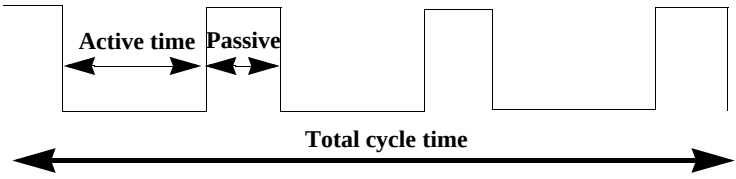
keep the IN key active.
1715ADPPPPPPPPPPPPP00**K0**

Data field structure for Output operations:

Each relay can be activated at a specific time. The relay can control the bell, door, turnstile or any other device. It has 3 operation modes:

S- Set (activate), open permanently.
R- Reset (passive), close permanently.
P- Pulse, activate for a period.

Byte	Length	Value	Type	Explanation
1	4	0000-2359	Time	Event time in military format
5	1	S, R, P	A&N	Operation type
6	2	01-02	N	Relay number
8	3	000-999	N	Active time in cycle, in units of 200 msec.
11	3	000-999	N	Passive time in cycle, in units of 200 msec.
14	3	000-999	N	Total cycle time, in units of 200 msec.
17	3	000	N	Not in use
20	2	00	N	Not in use
22	1	O	A&N	O-Output
23	1	0-7	N	Day of week, 0- all week, 1- Sunday, 7- Saturday



For Example: 1405**R01**00000000000000**O5**
For Example: 1405**S01**00000000000000**O5**
For Example: 1405**P012003005000**0000**O5**

Data field structure for Modem operations:

Currently, two commands are available:
1. Set into Auto-Answer
2. Send any AT Command

Auto answer - data field structure

Byte	Length	Value	Type	Explanation
1	4	0000-2359	Time	Event time in military format
5	1	A	A&N	Auto answer
6	1	0-9, N	A&N	Number of rings, N- Disables auto answer function.
7	13	0000000	N	Currently not in use
20	2	00	N	Not in Use
22	1	M	A&N	M-Modem
23	1	0-7	N	Day of week, 0- all week, 1- Sunday, 7-Saturday

AT commands - data field structure

Byte	Length	Value	Type	Explanation
5	1	I	A&N	Instructions (AT commands)
6	14		A&N	String to send

A.3.2 System parameters table record format

The system parameters table has only one record. This record is a string of parameter numbers and their values. Each parameter has a default value which is used when it is not defined. Each number starts with a “^” sign and is followed by a 3 digit number. The table below explains the existing

parameters:

Param. no.	Description	Value	Default
^000	Technician card length		6
^001	Tech. card authorization	D- Only Synel's technician card (000000) is authorized P- Only the value that is defined in parameter #2 is authorized B – Both values are valid S – Both values are valid and can set the time from the technician mode menu ver. 4.10 and upwards	B
^002	Tech. card number	Number length identical to parameter #0 length	000000
^003	Badge reading error notification	Y- Yes N- No	Y
^004	Shut-off timeout (sec.)	00-98 –The terminal will be turned off when it is working on its back-up battery and is not active for longer than the defined value. 99- Keeps the terminal ON continuously.	15
^005	Day light saving time	See "Day light saving time parameter structure" below.	Disabled
^006	Return to default function (sec.)	00-99, When a function key is pressed but not used.	50
^007	Memory alarm	000-999, turns ON an alarm LED whenever there is a memory overflow.	075
^008	Online_tmo	Online to off-line time out (sec.)	15

^009	Activate function led	Y- The LED is controlled by the firmware N- The LED is controlled by SAL	Y
^010	Memory overwrite	Y- Clears data that is temporarily acknowledged by the host (Op-code F) when memory is full. N- Clears commands from the host only (Op-code C,c)	N
^011	Date format	0- DDMMYY 1-YYMMDD 2- DDMMYYYY 3- YYYYMMDD	0
^012	nnnn	User defined code to be displayed	
^013	nn	Polling interval -- =Possible to change in Master mode value= Impossible changed in Master mode	--
^^	Sign for ending of parameters		

Record length is the total number of characters in all defined parameters (includes the parameter number and the ending of parameter sign "^^"). Make sure you define the constant value for default parameters.

A.3.3 Day Light Saving Time parameter structure

Setting the daylight savings time control

SY7xx terminals can be programmed to automatically move the clock forward or backward at given dates. The daylight saving time fields define when to move the SY7xx terminal's clock forward by one hour (Spring) and backward by one hour (Fall). This feature is optional. There are two methods of defining whether the clock will be moved: at a specified date (method 1) or using a fixed algorithm (method 2). This parameter can include more than one definition, each definition is a 13 bytes length string regardless of the

method used, but the structure is different.

Method 1 – Specific date

The clock will be moved per a specified date (DD/MM/YY):

Byte	Length	Value	Type	Explanation
1	1	1	N	Method type - 1
2	2	01-31	N	DD - day
4	2	01-12	N	MM - Month
6	2	01-99	N	YY - Year
8	2	00-23	N	HH – Hour (military format)
10	2	00-59	N	mm- Minutes
12	1	+, -	A&N	Forward (+), Backward (-)
13	1	0-9	N	Amount of change

Method 2 – Fixed algorithm

The clock will be moved on the day of the week on a specified week of a specific month:

Byte	Length	Value	Type	Explanation
1	1	2	N	Method type - 2
2	2	01-31	N	MM - Month
4	1	0	N	Constant
5	1	0-6	N	Day of week, from **0** = Sunday to **6** = Saturday
6	1	0	N	Constant
7	1	0-6	N	Week of month, from **1** for the 1st week to **L** for the last week
8	2	00-23	N	HH – Hour (military format)
10	2	00-59	N	mm- Minutes
12	1	+, -	A&N	Forward (+), Backward (-)
13	1	0-9	N	Amount of change

Sample setting using method 2

In the example below, when the terminal switches from 23:59 on the 1st Saturday of April to 00:00, it's internal clock automatically skips to 1:00 of the following day. When the terminal switches from 23:59 on the 1st Saturday of October to the next minute, it's internal clock automatically returns to 23:00 of the same day.

Method: 1 / 2	**Day**	**Month**	**Year**	**0**	**Day of week**	**0**	**Week of month**	**Time HH:mm**	**Change**
2	--	04	--	0	6	0	1	23:59	+1
2	--	10	--	0	6	0	1	23:59	-1

TRS record structure

This table enables using special characters that can not be transmitted by the protocol in the display/printer. Before displaying this character it replaces the transmitted code with the required character. There are two TRS table types: for display only, for display+printer. (Display only)

Byte	**Length**	**Value**	**Type**	**Explanation**
1	2	20-7F	A&N	ASCII code HEX of character to be translated.
3	2	20-FF	A&N	ASCII code HEX of character to display.

A.3.4 TRP record structure

Byte	**Length**	**Value**	**Type**	**Explanation**
1	2	20-7F	A&N	ASCII code HEX of character to be translated
3	2	20-FF	A&N	ASCII code HEX of character to display
5	2	20-FF	A&N	ASCII code HEX of character to print

Note: To disable translation either to display or print, you must fill-in the value to be translated (see 1st row in the table above) in the field you do not want to translate.

FNT record structure

This table enables creating (drawing within a 5x7 matrix) fonts that are not

display supported as per the user's specific requirements. You must define a specific designated character to enable this option.

Byte	Length	Value	Type	Explanation
1	1	Char	A&N	Symbol to be replaced
2	1	0	A&N	Type field – reserved for future use.
3	2	00-1F	A&N	Row 1 (hex)
5	2	00-1F	A&N	Row 2 (hex)
7	2	00-1F	A&N	Row 3 (hex)
9	2	00-1F	A&N	Row 4 (hex)
11	2	00-1F	A&N	Row 5 (hex)
13	2	00-1F	A&N	Row 6 (hex)
15	2	00-1F	A&N	Row 7 (hex)

Example- Synel's FNT00110.RDY file – see below:

_Special Char.

?-	character	
0-	type	
OE	01110	
11	10001	
01	00001	
02	00010	
04	00100	= **?**
00	00000	
04	00100	

^-	character	
0-	type	
04-	00100	
OE	01110	
15	10101	
04	00100	
04	00100	=
04	00100	
04	00100	

{-	character	
0-	type	
00-	00000	
01	00000	
09	00000	
1F	00000	
08	00000	=
00	00000	
00	11111	

}-	character	
0-	type	
00-	00000	
00	00000	
00	00000	
00	00000	
00	00000	= _____
00	00000	
1F	11111	

~-	character	
0-	type	
04-	00100	
04	00100	
04	00100	
04	00100	
15	10101	=
OE	01110	
04	00100	

A.4 Record structure of MPL

The MPL table is used for displaying messages. The messages can include @-Sequences that enable creating complex formats:

The following list outlines the available printed/displayed message file programming codes, their description and the placeholder position. An action depicts the information printed from a printer, which is connected to the terminal as a result of activating terminal reprogramming.

A message can contain up to 76 characters. 4 characters are the message number place holder and the remaining 72 will be used for the body of the message. In the page below please find common message programming codes:

A.4.1 Display @-Sequences formats

The following lists the available display message file programming codes, description and place-holder. Action depicts the information that is presented on the terminal display as a result of activating terminal reprogramming.

Code	Name	Length	Action
@D	DAY	02	Display the current day
@M	MONTH	02	_____ month two digits format
@Y	YEAR	02	_____ year two digit format
@y	YEAR	04	_____ year four digit format

@H	MILIT.HOUR	02	_____ hour (24) military format
@h	STAND.HOUR	02	_____ hour (12) format
@I	MINUTES	02	_____ minutes
@S	SECONDS	02	_____ seconds
@W	WEEK DAY	02	_____ day of the week Su Mon Tu We Th Fr Sa
@#..	IMMEDIATE		Display the text that follows the sign #
@X	SEQ. END		End of sequence
@P	POSITION		Start position: (rr-row) (cc-column)
@F_(9)	FRAME		Display frame buffer format in with parameter: '-' - source type (J-Jobbing ,K-keyboard) '--' - source offset '--' - frame length
			Attributes: '-' - char attribute: buffer(#)/another '-' - cursor state: on/off (Y/N) '-' - scroll left: on/off (Y/N) '-' - reserve
@T--	TEMPORARY		Display temporary message -- (measured in 1/5 Seconds)
@t--			-. -. -. - -- (Sec.) (From beginning of sequence)
@A	ALTERNATE		Beginning of alternate sequences
Code	**Name**	**Length**	**Action**
	begin		(Permit only once in @_sequence.)
@(	SEG.ALTER		Beginning of alternates sequences
	begin		
@)--	SEG.ALTER		End of alternate sequence block
	end		'--' (1/5sec)- Display time of alternate

@[	ALTERNATE		Used with @A: (@A@[@(@…@)- -@(@...@)@])
	begin		
@]	ALTERNATE		End of alternates sequence
	end		
|- -	BLINK		Blink for previous sequence block. first “-”-active time second”-”-passive

A.5 Printer @-Sequences formats

Code	Name	Length	Action
?	Begin		Beginning of all printer data-strings
@D	DAY	02	Display the current day
@M	MONTH	02	_____month two digits
@Y	YEAR	02	_____year two digits
@y	YEAR	04	_____year four digits
@H	MILIT.HOUR	02	_____24 hour military format
@h	STAND.HOUR	02	_____12 hour format
@I	MINUTES	02	_____minutes
@S	SECONDS	02	_____seconds
@W	WEEKDAY	02	_____day of the week: Su Mo Tu We Th Fr Sa
@#.....	IMMEDIATE		Print the text that follows the sign # To next @- sequence
@*nn	Hexadecimal	01	'nn'-one byte in hexadecimal coding To next @- sequence

@FPool l	FRAME		Print buffer in frame with parameters: 'oo' - source offset 'll' - frame length
Code	**Name**	**Length**	**Action**
@X	SEQ.END		End of String
@?	Report End		End of Print Report

A.6 Algorithm for Synel's numeric fields

A.6.1 Multi-drop ID algorithm

If multi-drop ID is < '@' & > '0' then
Multi-drop ID = - 30 Hex

If multi-drop ID is > '@' then
Multi-drop ID = - 31 Hex

A.6.2 HighByte + LowByte algorithm

If HighByte is > '9' then
Length is (HighByte - 48 dec.)x 10 dec. + LowByte x 1

A.6.3 HighByte + MiddleByte + LowByte algorithm*

If HighByte is > '9' then

Length is (HighByte - 48 dec.) x 100 dec. + MiddleByte x 10 + LowByte x 1
* Refers to the total of records number, see table in the page below:

No. of records	Actual no. in table
1-9nn	1-999
10nn	:nn
11nn	;nn
12nn	<nn
13nn	=nn
14nn	>nn
15nn	?nn
16nn	@nn
17nn	Ann
18nn	Bnn
19nn	Cnn
20nn	Dnn
21nn	Enn
22nn	Fnn
23nn	Gnn
24nn	Hnn
25nn	Inn
26nn	Jnn
27nn	Knn
28nn	Lnn
29nn	Mnn
30nn	Nnn
31nn	Onn

No. of records	Actual no. in table
32nn	Pnn
33nn	Qnn
34nn	Rnn
35nn	Snn
36nn	Tnn
37nn	Unn
38nn	Vnn
39nn	Wnn
40nn	Xnn
41nn	Ynn
42nn	Znn
43nn	[nn
44nn	\nn
45nn	]nn
46nn	^nn
47nn	`nn
48nn	ann
49nn	bnn
50nn	cnn
51nn	dnn
52nn	enn
53nn	fnn
54nn	gnn

No. of records	Actual no. in table
55nn	hnn
56nn	inn
57nn	jnn
58nn	knn
59nn	lnn
60nn	mnn
61nn	nnn
62nn	onn
63nn	pnn
64nn	qnn
65nn	rnn
66nn	snn
67nn	tnn
68nn	unn
69nn	vnn
70nn	wnn
71nn	xnn
72nn	ynn
73nn	znn
74nn	{nn
75nn	|nn

For example:

_
d
005

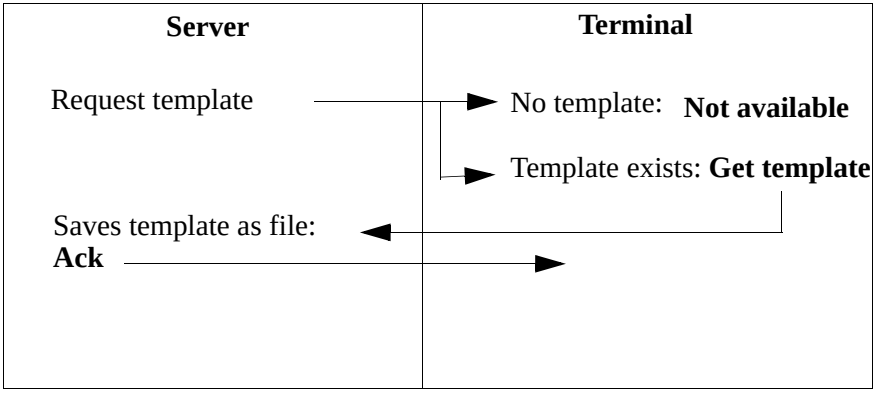
_
62981 (Total no. of characters)
A
23
42
>99 (Total record=1499)
06
00
02
000001TimeLOG Sup0001 10001200013000140001
000002TimeLOG Sup0002 100022000230002400002…

Appendix B --Host-Terminal Communication

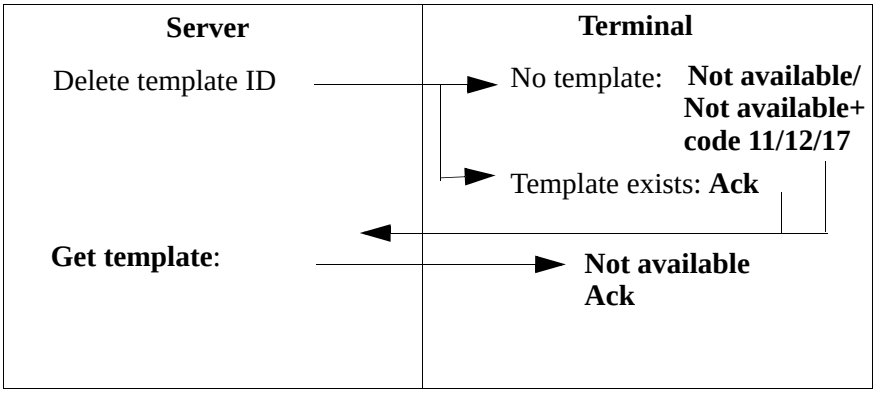
B.1 Fingerprint templates

Note: ***From SY-780/A firmware version 8.xxx, <u>prior</u> to working with the FP unit, it is recommended to perform the Halt command, and <u>after</u> working with the FP unit perform the Run command!***

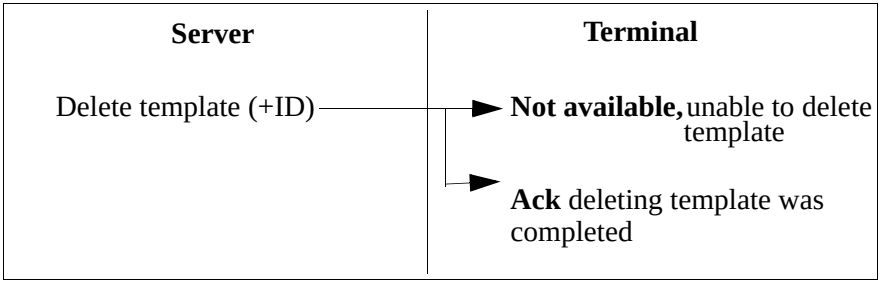
Server requests template from terminal



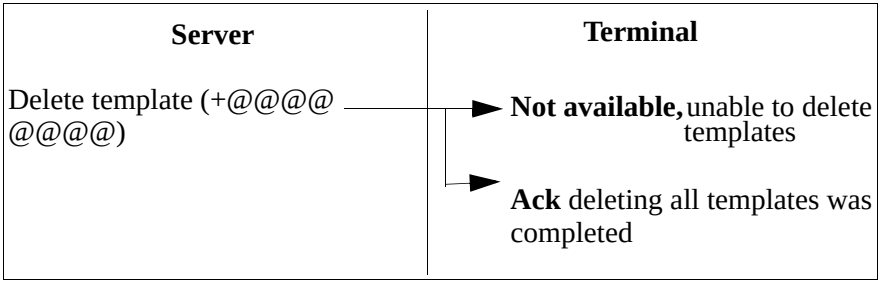
Server sends a template to terminal



Server deletes a template/all templates from the terminal



Server sends a template to terminal



Appendix C - FP Templates - Maintenance

C.1 Template Header Structure (SYNEL)

Byte	Length	Description	Remarks
1	8	Template ID	Card number
9	8	Employee ID	Currently not in use
17	8	Password	Currently not in use
25	2	Sensor version	Remains unchanged
27	2	Template version	Remains unchanged
29	3x2=6	FP quality	Fields: Name [0] - Name [2]
35	3x2=6	FP content	Fields: Name [3] - Name [5]
41	16	10x [Name], Finger, Admin level, Schedule	Currently not in use
67	1	Padded	
68	1	Sec. Thresh	Security threshold can be changed here as follows*: From **1**= Very high To **5**= Very low

* - Security level can be changed only using your PC!

C.2 Change security threshold for a single template:

A template's security threshold can not be changed from the SY-780/A terminal, only via your PC as follows:

Step 1: Enrol your finger. Instruct the terminal to save its quality

and content data in a record.

Step 2: You must collect this data from your PC and using your PC change byte 68 to one of the values from 1-5 (1=very high).

Step 3: Send the template to the terminal.

C.3 Replacing a template ID (card number):

A template ID number will usually be replaced if the employee card was lost/ changed for any other reason. The fingerprint remains unchanged, only the card number must be replaced as follows:

A Template ID = 32 bit. It is a long integer converted to 8 bytes. Write a short conversion program to perform the following:

Step 1: Convert the decimal number of the card to a long integer → hex 4 byte number. →

Step 2: Convert the 4 byte number to an Synel ASCII 8 byte number.

Step 3: Write this 8 byte number in the first 8 bytes of your template header.

Step 4: Send the template to the terminal.

Examples of the conversion process:

Decimal		**Hexa (4 byte)**		**Synel ASCII format* (8 byte)**
1234567890		0x499602D2		d9i6‘2m2
654321		0x0009FBF1		‘0‘9o;o1

* See “Fingerprint unit commands” on page -41