

Index

1	User serial port usage	3
2	FPS Module protocol answers.....	3
3	Command list.....	4
4	Command description	4
4.1	<i>GetVersion.....</i>	<i>4</i>
4.2	<i>SetLeds</i>	<i>4</i>
4.3	<i>GetPushButton.....</i>	<i>5</i>
4.4	<i>SetAuxOut.....</i>	<i>5</i>
4.5	<i>GetAuxOut.....</i>	<i>6</i>
4.6	<i>SetAuxPulse</i>	<i>7</i>
4.7	<i>GetAuxPulse</i>	<i>7</i>
4.8	<i>ContinousModeOff.....</i>	<i>7</i>
4.9	<i>GetQualityContinous</i>	<i>7</i>
4.10	<i>MatchContinous.....</i>	<i>8</i>
4.11	<i>MatchSingle.....</i>	<i>8</i>
4.12	<i>SetFID.....</i>	<i>9</i>
4.13	<i>EnrollSingle.....</i>	<i>9</i>
4.14	<i>TplDownload</i>	<i>9</i>
4.15	<i>TplUpload.....</i>	<i>10</i>
4.16	<i>TplErase</i>	<i>11</i>
4.17	<i>DbReset</i>	<i>11</i>
4.18	<i>DbInfo</i>	<i>11</i>
4.19	<i>DbMode.....</i>	<i>11</i>
4.20	<i>MatchContinuousPermanent.....</i>	<i>12</i>
4.21	<i>GetMatchContinous</i>	<i>12</i>
4.22	<i>SetBaudRate</i>	<i>12</i>
5	Compatibility issues.....	13
5.1	<i>Version 1.20.....</i>	<i>13</i>

1 User serial port usage

The serial port operates at 8 data bits, no parity, 1 stop bit.

The baudrate can be set with the SetBaudRate command (see 4.22 on page 12). Factory setting is 57600 bps.

During module boot a few text lines describing sensor and libraries initialization are sent to the serial port. At the end of this initialization (if everything was ok) a character with the code of 0x07 (BEL) is sent. A microcontroller can detect this character and use it to detect the end of the module boot.

2 FPS Module protocol answers

Except for the Fingerprint Identification String returned by the MatchSingle or MatchContinuous commands and the Version command (see below), the FPS module replies using uniquely (?) defined characters.

Reply	Char	ASCII	Notes
ACMD_NAK	NAK	0x15	Command not executed. Normally this indicates that an unknown command has been received by the FPS module
ACMD_ACK	ACK	0x06	Command executed successfully
ACMD_ENROLLOK	@	0x40	Enroll operation completed successfully.
ACMD_ENROLLFAIL	-	0x2D	Enroll operation failed, possibly due to insufficient features read by the FP sensor.
ACMD_MATCHOK	O	0x4F	Match completed successfully and fingerprint found in the module database. This reply is followed by the 8 characters Fingerprint Identification String saved in the database during the Enroll process.
ACMD_MATCHFAIL	K	0x4B	Match failed: the fingerprint has not been found in the module database.
ACMD_GOTFINGER	*	0x2A	A finger has been sensed by the sensor and captured successfully. This reply is sent during the execution of the MatchContinuous, MatchSingle, EnrollSingle and GetQualityContinuous commands.
ACMD_NOFINGER	-	0x2D	There is no finger on the sensor. Returned during the GetQualityContinuous command.
ACMD_DBFOUND	F	0x46	Finger found in database
ACMD_DBNOTFOUND	N	0x4E	Finger not found in database
ACMD_SENDDATA	S	0x53	Upload data to sensor

3 Command list

The following commands are defined in FPS module firmware v. 1.30

ContinousModeOff		MatchContinous	
0x63	7	0x4D	8
DbInfo		MatchContinuousPermanent	
0x49	11	0x53	12
DbMode		MatchSingle	
0x64	11	0x6D	8
DbReset		SetAuxOut	
0x52	11	0x61	5
EnrollSingle		SetAuxPulse	
0x65	9	0x50	7
GetAuxOut		SetBaudRate	
0x41	6	0x42	12
GetAuxPulse		SetFID	
0x10	7	0x69	9
GetMatchContinous		SetLeds	
0x11	12	0x6C.....	4
GetPushButton		TplDownload	
0x70	5	0x44	9
GetQualityContinous		TplErase	
0x43	7	0x45	11
GetVersion		TplUpload	
0x76	4	0x55	10

4 Command description

4.1 GetVersion

Command: v (ASCII 0x76)
 Parameters: None
 Reply: 3 bytes
 Example: ▷ **0x76**
 ◁ **0x31 0x30 0x30**

Notes: The first byte is the major version number, the remaining two are the minor version number. So the version replied in the example is version 1.00

4.2 SetLeds

Command: l (ASCII 0x6c)

Parameters: 1 byte, bit mapped as follow:

7	6	5	4	3	2	1	0
LED 4		LED 3		LED 2		LED 1	

Bits		Operation made
0	0	The LED is left unchanged
0	1	The LED is toggled (switched OFF if it was ON and viceversa)
1	0	The LED is switched OFF
1	1	The LED is switched ON

Reply: 1 byte (ACK)

Example: ▷ **0x6C 0xE4**

◁ **0x06**

Notes: In this example, LED 1 is left unchanged, LED 2 is toggled, LED 3 is switched OFF and LED 4 is switched ON.

4.3 GetPushButton

Command: p (ASCII 0x70)

Parameters: None

Reply: 1 byte

Example: ▷ **0x70**

◁ **0x31**

Notes: The returned value is 0x31 if the button is pressed, or 0x30 if the button is released.

4.4 SetAuxOut

Command: a (ASCII 0x61)

Parameters: 2 byte, from the following tables

Byte 1: Output to configure:

Value	Output
0x00	Aux Output #0
0x01	Aux Output #1
0x02	Led #1
0x03	Led #2
0x04	Led #3
0x05	Led #4

Byte 2: Operation to perform:

Parameter	Operation
0x00	Set Aux output OFF
0x01	Set Aux output ON
0x02	Toggle Aux output

Parameter	Operation
0x03	Give a positive going pulse on Aux output ⁽¹⁾
0x04	Give a negative going pulse on Aux output ⁽¹⁾
0x05	Set Aux output ON when a successful match occurs
0x06	Set Aux output OFF when a successful match occurs
0x07	Give a positive going pulse on Aux output when a successful match occurs ⁽¹⁾
0x08	Give a negative going pulse on Aux output when a successful match occurs ⁽¹⁾
0x09	Set Aux output ON when a failed match occurs
0x0A	Set Aux output OFF when a failed match occurs
0x0B	Give a positive going pulse on Aux output when a failed match occurs ⁽¹⁾
0x0C	Give a negative going pulse on Aux output when a failed match occurs ⁽¹⁾
0x0D	Set Aux ON when a successful enroll occurs
0x0E	Set Aux OFF when a successful enroll occurs
0x0F	Give a positive going pulse on Aux output when a successful enroll occurs ⁽¹⁾
0x10	Give a negative going pulse on Aux output when a successful enroll occurs ⁽¹⁾
0x11	Set Aux ON when a failed enroll occurs
0x12	Set Aux OFF when a failed enroll occurs
0x13	Give a positive going pulse on Aux output when a failed enroll occurs ⁽¹⁾
0x14	Give a negative going pulse on Aux output when a failed enroll occurs ⁽¹⁾
0x15	Set Aux ON when the sensor sense a finger
0x16	Set Aux OFF when the sensor sense a finger
0x17	Give a positive going pulse on Aux output when a finger is sensed ⁽¹⁾⁽²⁾
0x18	Give a negative going pulse on Aux output when a finger is sensed ⁽¹⁾⁽²⁾

Reply: ACK

Example: ▷ 0x61 0x00 0x05

◁ 0x06

Notes: (1) See the *SetAuxPulse* command to set the length of this pulse
 (2) This event occurs both in Match, Enroll and GetQuality commands.

4.5 GetAuxOut

Command: A (ASCII 0x41).

Parameters: 1 byte: Output to get configuration. See table in SetAuxOut.

Reply: 2 bytes with the ASCII value of the hex number of the function associated with this output, from the table in 4.4 SetAuxOut.

Example: ▷ **0x41 0x01**
 ◁ **0x30 0x07** (*Function 0x07 is associated with output 1*)

4.6 SetAuxPulse

Command: P (ASCII 0x50)

Parameters: 2 bytes, Output to configure and Pulse length in tenth of a second

Reply: ACK

Example: ▷ **0x50 0x02 0x0A** (*Set Pulse length for LED1 to 1 sec*)
 ◁ **0x06**

Notes: See table in SetAuxOut on page 3

4.7 GetAuxPulse

Command: CTRL-P (ASCII 0x10)

Parameters: 1 byte: Output to get configuration. See table in SetAuxOut.

Reply: 2 bytes with the ASCII value of the hex number of the impulse len in tenth of a second.

Example: ▷ **0x10 0x00**
 ◁ **0x36 0x34** (*Impulse len of 0x0A tenth of a second for Aux0*)

4.8 ContinuousModeOff

Command: c (ASCII 0x63)

Parameters: None

Reply: ACK

Example: ▷ **0x63**
 ◁ **0x06**

Notes: Disables continuous mode. In Continuous mode the module is continuously reading the FP sensor and optionally doing a match operation against fingerprints stored in its internal database, reporting asynchronously the result of the operation. See the *MatchContinuous* and the *GetQualityContinuous* commands.

4.9 GetQualityContinuous

Command: C (ASCII 0x43)

Parameters: None

Reply: ACK

Example: ▷ **0x43**
 ◁ **0x06**

Notes: Activating the GetQualityContinuous mode, the FPS module will continuously reading the sensor outputting ACMD_NOFINGER or

ACMD_GOTFINGER if a finger is sensed. Note that no match operation is performed in this case against the internal fingerprint database.

If Aux0 or Aux1 outputs are programmed to perform an operation when a finger is sensed, the programmed operation will be performed.

4.10 MatchContinuous

Command: M (ASCII 0x4D)

Parameters: None

Reply: ACK

Example: ▷ 0x4D
 ◁ 0x06

Notes: When the MatchContinuous mode is activated, the FPS module will continuously read the fingerprint sensor.

When a finger is sensed, the system will output a ACMD_GOTFINGER byte, followed either by ACMD_MATCHFAIL if the finger is not found in the fingerprint database, or by ACMD_MATCHOK if the finger is found.

The ACMD_MATCHOK will be followed by the 8 bytes Fingerprint Identification string stored with that fingerprint during the enroll process.

If Aux0 or Aux1 outputs are programmed to perform an operation when a finger is sensed, the programmed operation will be performed.

4.11 MatchSingle

Command: m (ASCII 0x6D)

Parameters: None

Reply: 1 or more bytes. See the example below

Example: ▷ 0x6D

Now the match process has begun. The reply from the FPS module depends on what happens: there are 3 cases:

- Case #1: Match process aborted
 - ▷ 0x1B
 - ◁ 0x06
- Case #2: Finger sensed but not found in the database
 - ◁ 0x2A (ACMD_GOTFINGER)
 - ◁ 0x4B (ACMD_MATCHFAIL)
- Case #3: Finger sensed and found in the database.
 - ◁ 0x2A (ACMD_GOTFINGER)
 - ◁ 0x4F (ACMD_MATCHOK)
 - ◁ 0x.. .. 0x.. (Finger Identification string, 8 bytes)

Notes: If Aux0 or Aux1 outputs are programmed to perform an operation when a finger is sensed or matched, the programmed operation will be performed.

4.12 SetFID

Command: i (ASCII 0x69)
 Parameters: FID string, 8 bytes
 Reply: ACK
 Example: ▷ 0x69 0x... .. 0x...
 ◁ 0x06

Notes: The 8 bytes Finger Identification string is internally stored and will be associated with all the following enrolled fingerprints.

4.13 EnrollSingle

Command: e (ASCII 0x65)
 Parameters: None
 Reply: 1 or more bytes. See the example below
 Example: ▷ 0x65

Now the enroll process has begun. The reply from the FPS module depends on what happens: there are 3 cases:

- Case #1: Enroll process aborted
 ▷ 0x1B
 ◁ 0x06
- Case #2: Finger sensed but the enroll process failed
 ◁ 0x2A (ACMD_GOTFINGER)
 ◁ 0x2D (ACMD_ENROLLFAIL)
- Case #3: Finger sensed and enroll succeeded.
 ◁ 0x2A (ACMD_GOTFINGER)
 ◁ 0x40 (ACMD_ENROLLOK)

Notes: The fingerprint will be stored in the internal database associated with the currently active Finger Identification String (see the *SetFID* command).

If Aux0 or Aux1 outputs are programmed to perform an operation when a finger is sensed and/or enrolled, the programmed operation will be performed.

4.14 TplDownload

Command: D (ASCII 0x44)
 Parameters: None
 Reply: 1 or more bytes. See the example below
 Example: ▷ 0x44

The reply from the FPS module depends on what happens: there are 2 cases:

- Case #1: Template not found in the Database
 ◁ **0x4E** (ACMD_DBNOTFOUND)
- Case #2: Template found in database. It will be downloaded.
 ◁ **0x46** (ACMD_DBFOUND)
 ◁ **0x31, 0x34, 0x38** (Bytes that will be downloaded: '148' ASCII)
 ◁ **0x.. ... 0x..** (Template data, 148 bytes)

Notes: Template Data associated with the currently active Finger Identification String (see the *SetFID* command) will be read from the internal database and downloaded. Template Data consists of both the “Finger Identification String” and its associated fingerprint data. The number of bytes to be downloaded depends on several factors and it is NOT constant. The current upper limit is 300 bytes.

4.15 TplUpload

Command: U (ASCII 0x55)
 Parameters: 3 bytes (template len) followed by template data
 Reply: 1 byte.
 Example: Sending data for a 154 bytes template
 ▷ **0x55, 0x31, 0x35, 0x34**
 The reply from the FPS module depends on what happens: there are 3 cases:

- Case #1: Template length is correct and template data accepted by the FPS Module
 ◁ **0x53** (ACMD_SENDDATA)
 ▷ **0x.. ... 0x..** (Template data, 154 bytes)
 ◁ **0x06** (ACMD_ACK)
- Case #2: Template length is correct but template data NOT accepted by the FPS Module
 ◁ **0x53** (ACMD_SENDDATA)
 ▷ **0x.. ... 0x..** (Template data, 154 bytes)
 ◁ **0x15** (ACMD_NAK)
- Case #3: Template length is NOT correct.
 ◁ **0x15** (ACMD_NAK)

Notes: Uploaded Template will be saved into the internal database. The “Finger Identification String” for this finger print has been saved inside the template and does not require to be set before the upload.

The number of bytes to be uploaded depends on several factors and it is NOT constant. The current upper limit is 300 bytes.

4.16 TplErase

Command: E (ASCII 0x45)
 Parameters: None
 Reply: Acknowledge
 Example: ▷ 0x45
 ◁ 0x06

Notes: Template Data associated with the currently active Finger Identification String (see the *SetFID* command) will be deleted from the internal database and downloaded.
 This call will return success even if the requested FID is not present in the database (i.e.: deleting a non-existing FID will not return an error).

4.17 DbReset

Command: R (ASCII 0x52)
 Parameters: None
 Reply: Acknowledge
 Example: ▷ 0x52
 ◁ 0x06

Notes: The full database content is erased. No fingerprint will remain in the database after this command has been executed.

4.18 DbInfo

Command: I (ASCII 0x49)
 Parameters: None
 Reply: 'F' (0x46) or 'R' (0x52) indicating Flash (permanent) or Ram (volatile) database, followed by comma and two ascii numbers, separated by comma, indicating the number of fingerprints stored in the database and the number of free bytes in the database memory, followed by *ACMD_ACK*.

Example: ▷ 0x49
 ◁ 0x52 0x2c 0x31 0x2c 0x34 0x30 0x39 0x32 0x06
 (Ram database, 1 fingerprint, 4092 free bytes)

Notes: Deleting a fingerprint will not necessary free the memory used (i.e.: you will not generally see the free memory increase when deleting fingerprints).

4.19 DbMode

Command: d (ASCII 0x64)

Parameters: 1 byte , database mode: 'F' (0x46) for Flash (permanent) database or 'R' (0x52) for Ram (volatile) database.
 Reply: *ACMD_ACK.*, if parameter is correct, *ACMD_NAK* otherwise
 Example: ▷ **0x64 0x46** (*Select Permanent database*)
 ◁ **0x06** (*ACMD_ACK*)
 Notes: The database mode change is not done 'on the fly' but the new setting will become operational after a FPS module reset.

4.20 MatchContinuousPermanent

Command: S (ASCII 0x53)
 Parameters: None
 Reply: ACK
 Example: ▷ **0x53**
 ◁ **0x06**
 Notes: This command is the same as MatchContinuous (see paragraph 4.10 on page 3) with the difference that this setting will be saved in non volatile memory, so the FPS module will start a MatchContinuous automatically at power on.

4.21 GetMatchContinuous

Command: CTRL-Q (ASCII 0x11)
 Parameters: None
 Reply: ACK or NAK
 Example: ▷ **0x53**
 ◁ **0x06** (*Continuous mode is active*)
 ▷ **0x53**
 ◁ **0x11** (*Continuous mode is NOT active*)
 Notes: This command returns the ContinuousMode status. When an *ACMD_ACK* is returned, either a *MatchContinuous* or a *MatchContinuousPermanent* is active.

4.22 SetBaudRate

Command: B (ASCII 0x42)
 Parameters: 0x41 0x55 0x44 Baudrate
 Reply: ACK or NAK
 Example: ▷ **0x42 0x41 0x55 0x44 0x31** (*Set baud to 9600*)
 ◁ **0x06** (*ACMD_ACK*)
 Notes: The '0x41 0x55 0x44' constant parameters are meant to avoid erroneous baudrate changes.
 Available baudrates are the following:

Parameter value	Baudrate
0x31	9600
0x32	19200

Parameter value	Baudrate
0x33	38400
0x34	57600
0x35	115200

Any value not defined here will default to 57600 bps.

Note that the PC demonstration program will use the 57600 bps rate only.

5 Compatibility issues

Since up to version 1.20 this protocol was considered “preliminary”, there will be no guarantee regarding command compatibility with previous firmware versions.

This section will list, version by version, the differences in command syntax and/or reply for incompatible commands.

5.1 Version 1.20

- *DbInfo* command response
- *SetAuxOut*, *SetAuxPulse*, *GetAuxOut*, *GetAuxPulse* commands.

5.2 Version 1.30

No relevant changes.