5 Software Command Interface

To communicate with the sensor module, a serial command interface is used between a host processor and the processor board. This interface is designed to be easy to use and performs the basic biometric functions needed in a fingerprint authentication system.

5.1 UART Serial Interface Settings

The software settings for the serial protocol using UART are outlined in

Parameter	Value
Communication speed	Range from 9600 to 115200 baud. Factory default baud rate set to 9600 baud.
Format	8 data bits, odd parity, one stop bit.
Bit order	Least significant bit first

Table 16: UART interface settings

5.2 SPI Interface Settings

The software settings for using SPI commands are outlined in

Parameter	Value
Communication speed	Guaranteed maximum speed: 20 MHz. For details, see SPI timing requirements, in section
SPI Mode	Mode 3, Clock Polarity High and Clock Phase Rising
Chip Select	Active Low

Table 17: SPI interface settings

5.3 Command Send Structure

The structure of commands sent from a host are shown in and described in this section:

0	1	2	3	4	5
STX	IDX-LSB	IDX-MSB	COMMAND	PAYLOAD-LSB	PAYLOAD-MSB

Table 18: Structure of commands sent from the host

STX
Start byte: 0x02.

IDX-LSB:

Index value, least significant byte.

If no specific value is used for a command setting, the IDX-LSB must be set to zero.

IDX-MSB:

Index value, most significant byte.

If no specific value is used for a command setting, the IDX-MSB must be set to zero.

COMMAND:

Command byte.

PAYLOAD-LSB:

If any additional data is sent, the payload is a counter of how many bytes that will be sent (not including the CRC-code), otherwise zero.

PAYLOAD-MSB:

Payload most significant byte, if no data, set to zero.

5.3.1 Additional Payload Data

If PAYLOAD != 0, then additional data should follow in the stream outline in

6	...	n	n+1	n+2	n+3	n+4
DATA	DATA-...	DATA-n	CRC-LSB	CRC-BYTE2	CRC-BYTE3	CRC-MSB

Table 19: Command Send - Additional Payload Data

The CRC size (4 bytes) is not included in the payload counter. Its value is calculated from all the data bytes, and is used for checking if an error occurred during the transmission. The default for IDX-LSB and IDX-MSB is 0x00, if nothing else is stated.

A new command cannot be sent before a response has been received from the previous command. An exception is when using the API_CANCEL command. See section for more information on the cancel command.

5.4 Response structure

The structure of the response from the FPC-BM is given in

0	1	2	3
STX	RESULT	PAYLOAD-LSB	PAYLOAD-MSB

Table 20: Structure of response from the FPC-BM

STX: Start byte: 0x02

RESULT: Result byte

PAYLOAD-LSB: If any additional data is sent, the payload is a counter of how many bytes that will be sent (not including the CRC-code), otherwise zero.

PAYLOAD-MSB: Payload most significant byte, if no data, set to zero. If PAYLOAD != 0, then additional data should follow in the stream according to

4	...	n	n+1	n+2	n+3	n+4
DATA-1	DATA-...	DATA-n	CRC-LSB	CRC-BYTE2	CRC-BYTE3	CRC-MSB

Table 21: Response Structure Additional Data

The CRC size (4 bytes) is not included in the payload counter. Its value is calculated from all the data bytes, and is used for checking if an error occurred during the transmission.

5.5 SPI Timing Requirements

The SPI interface of the FPC-BM is a slave interface, implying that the host (the master) determines when data is sent to and from the FPC-BM. Since the host cannot know when the FPC-BM has completed processing a given command, a polling process is implemented by the host when trying to read the response for a given command request.

The required implementation of the Request/Response process is as follows:

1. Let the host send the 6 command bytes
2. Wait a minimum of 20 ms for possible payload and CRC
3. Let the host send a byte with value *0x52* to the slave. The host asks the FPC-BM if it is ready to send the response back to the host.
4. Wait a minimum of 20 ms
5. Check that the received byte is *0x02*. If not, the slave is not ready, and requires more time to complete processing the command. Alternatively, the FPC-BM returns *0x52* to indicate that it is busy.
6. Repeat steps 4-6 until a *0x02* byte is received as response.
7. The *0x02* value is the first byte in the regular response consisting of 4 bytes (plus a possible payload and CRC).

The SPI data transfer speed is up to 20 MHz for all single -byte transmission.

UART Interface

The requirements outlined in this section do not apply for the UART interface, where the host is aware that the received response is the correct response.

6 Command Tables

This section gives an overview of the available commands that can be used with the FPC-BM.

6.1 Biometric Commands

This section describes the biometric commands for the FPC-BM.

Command	HEX	Description
API_CAPTURE_IMAGE	0x80	Capture image from sensor (before enroll).
API_CAPTURE_AND_ENROL_RAM	0x81	Enroll into RAM (includes Capture Image)
API_CAPTURE_AND_VERIFY_RAM	0x82	Verify against RAM (includes Capture Image)
API_CAPTURE_AND_VERIFY_FLASH	0x83	Verify against single FLASH slot (includes Capture Image) Set slot number (0 – 49) in IDX
API_CAPTURE_AND_IDENTIFY_FLASH	0x84	Identify (Few) against all FLASH slots (includes Capture Image)
API_ENROL_RAM	0x85	Enroll into RAM
API_VERIFY_RAM	0x86	Verify against RAM
API_VERIFY_FLASH	0x87	Verify against single FLASH slot. Set slot number (0 – 49) in IDX
API_IDENTIFY_FLASH	0x88	Identify (Few) against all FLASH slots
API_CAPTURE_IMAGE_FINGERPRESENT	0x89	Capture Image from sensor (once a finger is present)
API_ENROL_FLASH	0x92	Enroll into FLASH memory
API_CAPTURE_AND_ENROL_FLASH	0x93	Enroll into FLASH memory (includes Capture Image)

Table 22: Biometric Commands

6.2 Image transfer

This section describes the image transfer commands for the FPC-BM.

Command	HEX	Description
API_UPLOAD_IMAGE	0x90	Upload image from RAM to Host
API_DOWNLOAD_IMAGE	0x91	Download image to RAM to Host

Table 23: Image transfer commands

6.3 Template Handling

This section describes the template handling commands for the FPC-BM.

Command	HEX	Descriptions
API_UPLOAD_TEMPLATE	0xA0	Upload template from RAM to host
API_DOWNLOAD_TEMPLATE	0xA1	Download template to RAM to host
API_COPY_TEMPLATE_RAM_TO_FLASH	0xA2	Copy template from RAM to permanent FLASH storage Set slot number (0 to 49) in IDX
API_UPLOAD_TEMPLATE_FROM_FLASH	0xA3	Upload template from single FLASH slot to host. Set slot number (0 to 49) in IDX
API_DELETE_TEMPLATE_RAM	0xA4	Erase template from RAM
API_DELETE_SLOT_IN_FLASH	0xA5	Delete single slot in FLASH Set slot number (0 to 49) in IDX
API_DELETE_ALL_IN_FLASH	0xA6	Delete all FLASH slots
API_DOWNLOAD_TEMPLATE_TO_FLASH	0xA7	Download a template from host to FLASH

Table 24: Template handling commands

6.4 Algorithm Settings

This section describes the algorithm settings commands for the FPC-BM.

Command	HEX	Description
API_SECURITY_LEVEL_RAM	0xB0	Set security level, setting saved in RAM IDX-LSB: 0x04 = high convenience 0x05 = standard 0x06 = high security
API_SECURITY_LEVEL_STATIC	0xB1	Set security level, setting saved in non-volatile (static) memory.
API_GET_SECURITY_LEVEL	0xB2	Returns the current security level, value sent as payload data.
API_GET_DYNAMIC_UPDATE	0xB3	Returns the current dynamic update setting.
API_SET_DYNAMIC_UPDATE	0xB4	Sets the dynamic update IDX-LSB: 0x00 = Off 0x01 = On

Table 25: Commands for algorithm settings

6.5 Firmware Commands

This section describes the firmware commands for the FPC-BM.

Command	HEX	Description
API_FIRMWARE_VERSION	0xC0	Retrieve the version string for this device

Table 26: Firmware commands

6.6 Communication Commands

This section describes the communication commands for the FPC-BM.

Command	HEX	Description
API_SET_BAUD_RATE_RAM	0xD0	Set baud rate, setting saved in RAM. See section
API_SET_BAUD_RATE_STATIC	0xD1	Set baud rate, setting saved in non-volatile static Flash memory. See section
API_TEST_HARDWARE	0xD2	Test hardware components

Table 27: Communication commands

6.7 Power Commands

This section describes the commands for power settings for the FPC-BM.

Command	HEX	Description
API_ENTER_SLEEP_MODE	0xE1	Enter sleep mode (wake up by activating proper pin)
API_GET_POWER_SAVE_MODE	0xE5	Get current power save mode, value sent as payload data: IDX-MSB: 0x00 = Frequency 0x01 = LED 0x02 = Sleep Mode 0x03 = Sensor wake up (detection frequency)
API_POWER_SAVE_MODE_RAM	0xE2	Set power save mode, setting saved in RAM IDX-MSB: 0x00 = Frequency 0x01 = LED 0x02 = Sleep Mode 0x03 = Sensor wake up (detection frequency) IDX-LSB (frequency): 0x00 = Half 0x01 = Full IDX-LSB (LED): 0x00 = On 0x01 = Mode 0x02 = Off IDX-LSB (sleep mode): 0x00 = Run

		0x01 = Standby 0x02 = Sleep 0x03 Deep Sleep IDX-LSB (sensor wake up): 0x00 = Min (always active) … up to 0xFF = 2000 ms (each step = 7.8 ms)
API_POWER_SAVE_MODE_STATIC	0xE3	Set power save mode, setting saved in non-volatile (static) memory IDX-MSB: 0x00 = Frequency 0x01 = LED 0x02 = Sleep Mode 0x03 = Sensor wake up (detection frequency) IDX-LSB (frequency): 0x00 = Half 0x01 = Full IDX-LSB (LED): 0x00 = On 0x01 = Mode 0x02 = Off IDX-LSB (sleep mode): 0x00 = Run 0x01 = Standby 0x02 = Sleep 0x03 Deep Sleep IDX-LSB (sensor wake up): 0x00 = Min (always active) … up to 0xFF = 2000 ms (each step = 7.8 ms)

Table 28: Power commands

6.8 Miscellaneous Commands

This section describes other miscellaneous commands for the FPC-BM.

Command	HEX	Description
API_CANCEL	0xE0	Cancel ongoing command, only valid for the following commands: API_CAPTURE_AND_ENROL_RAM, API_CAPTURE_AND_VERIFY_RAM, API_CAPTURE_AND_VERIFY_AGAINST_FLASH, API_CAPTURE_AND_IDENTIFY_AGAINST_FLASH
API_ADVANCED_SETTINGS	0xE8	Managing advanced settings, see section
API_STAND_ALONE	0xEF	Toggle stand-alone functionality IDX-LSB: 0x00 = Off 0x01 = On

Table 29: Miscellaneous commands

6.9 Response Bytes

This section describes the possible response bytes from the FPC-BM to the host.

Command	HEX
API_FAILURE	0x00
API_SUCCESS	0x01
API_NO_FINGER_PRESENT	0x02
API_FINGER_PRESENT	0x03
API_VERIFICATION_OK	0x04
API_VERIFICATION_FAIL	0x05
API_ENROL_OK	0x06
API_ENROL_FAIL	0x07
API_HW_TEST_OK	0x08
API_HW_TEST_FAIL	0x09
API_CRC_FAIL	0x0A
API_PAYLOAD_TOO_LONG	0x0B
API_PAYLOAD_TOO_SHORT	0x0C
API_UNKNOWN_COMMAND	0x0D
API_NO_TEMPLATE_PRESENT	0x0E
API_IDENTIFY_OK	0x0F

API_IDENTIFY_FAIL	0x10
API_INVALID_SLOT_NR	0x11
API_CANCEL_SUCCESS	0x12
API_APPL_CRC_FAIL	0x14
API_SYS_CRC_FAILED	0x16
API_LOW_VOLTAGE	0x17
API_NO_SENSOR	0x18
API_BROKEN_SENSOR	0x19

Table 30: Response Bytes

6.10 No sensor: API_NO_SENSOR

The response **API_NO_SENSOR** is sent for the following commands if there is no active sensor present in the system during runtime.

- API_TEST_HARDWARE
- API_CAPTURE_IMAGE
- API_CAPTURE_AND_VERIFY_RAM
- API_CAPTURE_IMAGE_FINGERPRESENT
- API_CAPTURE_AND_VERIFY_FLASH
- API_CAPTURE_AND_IDENTIFY_FLASH
- API_CAPTURE_AND_ENROL_FLASH

Startup

It should be noted that if there is no sensor present at system startup, the boot sequence will end and the red LED lamp will blink until the system is powered down or until a sensor is connected. The response **API_NO_SENSOR** is not sent in this case.

6.11 Broken sensor: API_BROKEN_SENSOR

The response **API_BROKEN_SENSOR** is sent for the following commands if an improper or corrupted response is received from the sensor:

- API_TEST_HARDWARE
- API_CAPTURE_IMAGE
- API_CAPTURE_AND_VERIFY_RAM
- API_CAPTURE_IMAGE_FINGERPRESENT
- API_CAPTURE_AND_VERIFY_FLASH
- API_CAPTURE_AND_IDENTIFY_FLASH
- API_CAPTURE_AND_ENROL_FLASH

Startup

It should be noted that if an improper or corrupted response is received from a connected sensor at system startup, the boot sequence will end and the red LED lamp will blink until the system is powered down, or until a functioning sensor is connected to the system. The response **API_BROKEN_SENSOR** is not sent in this case.

7 Command Descriptions

This chapter describes the individual commands of the main application, along with their parameters, and responses.

7.1 Capture image: API_CAPTURE_IMAGE

An image is captured from the fingerprint sensor. The fingerprint image is placed in RAM and can be uploaded by the command API_UPLOAD_IMAGE. Calculation is done on the image to determine if a finger is present or not present on the sensor. No payload is sent with this command.

Response command

- API_NO_FINGER_PRESENT = No finger present on sensor
- API_FINGER_PRESENT = Finger present on sensor
- API_BROKEN_SENSOR = Improper or corrupted response from sensor

No payload is received with the response from this command.

7.1.1 Standalone Mode

Capture Image is not supported in Standalone mode. To use the API_CAPTURE_IMAGE command, the API_STAND_ALONE setting must be turned off (command sequence: *0x02 0x00 0x00 0xEF 0x00 0x00*).

See section for more information on API_STAND_ALONE.

7.2 Capture and Enroll (RAM): API_CAPTURE_AND_ENROL_RAM

An image is captured from the fingerprint sensor and enrolment of the image is performed. See for the number of finger placements required during enroll.

Sensor	Number of finger placements during enroll
FPC1011	1
FPC1020	8

Table 31: Finger placements during enrolment

Images are captured in a loop from the sensor until a finger is present. The command waits for "finger present" before it starts enrolment. No payload is sent with this command.

The command returns with response when the enrolment is complete or if the enrolment fails for any reason. After enrolment, the template is stored in RAM and can be uploaded or moved to FLASH storage.

Response command

- API_ENROL_OK = Enrolment successful
- API_ENROL_FAIL = Enrolment failed
- API_CANCEL_SUCCESS = API_CANCEL successful, or timeout of 6 seconds per finger placement.

No payload is received with the response from this command.

Cancel operation

It is possible to cancel the current enroll operation by sending the command API_CANCEL. This cancels the enrolment and the device returns to its normal command loop.

See Section for more information on the cancel command.

7.3 Capture and Verify (RAM): API_CAPTURE_AND_VERIFY_RAM

A template must be present in RAM before starting the verification, (use one of the following commands: **API_DOWNLOAD_TEMPLATE**, **API_CAPTURE_ENROL_RAM**). Thereafter the verification can be started. This command also captures an image from the fingerprint sensor. The command waits for "finger present" before it starts the verification. This means that images are captured in a loop from the sensor until a finger is present. The command returns with response when the verification is complete or if the verification fails for any reason. No payload is sent with this command.

Response command

- API_VERIFICATION_OK = Verification successful
- API_VERIFICATION_FAIL = Verification failed
- API_NO_TEMPLATE_PRESENT = No template present
- API_BROKEN_SENSOR = Improper or corrupted response from sensor
- API_CANCEL_SUCCESS = **API_CANCEL** successful, or timeout of 6 seconds per finger placement.

No payload is received with the response from this command. It is possible to cancel the current verification operation by sending the command **API_CANCEL**. This returns the device to its normal command loop.

See Section for more information on the cancel command.

7.4 Capture and Identify (FLASH): API_CAPTURE_ AND _IDENTIFY_FLASH

Identification is made against all FLASH slots. This command first captures an image from the fingerprint sensor. The command waits for "finger present" before it starts the identification. This means that images are captured in a loop from the sensor until a finger is present. The command returns with response when the identification is complete or if the identification fails for any reason. No payload is sent with this command.

Response command

- API_ IDENTIFY _OK = Identification successful
- API_ IDENTIFY _FAIL = Identification fails
- API_BROKEN_SENSOR = Improper or corrupted response from sensor
- API_CANCEL_SUCCESS = **API_CANCEL** successful, or timeout of 6 seconds per finger placement.

In a successful identification, the slot index is received as payload in two bytes (LSB first) plus the 4 CRC bytes. Maximum number of templates during identification is 50.

It is possible to cancel the current identification operation by sending the **API_CANCEL** command. The device returns to its normal command loop. See Section for more information on the cancel command.

7.5 Enroll (RAM): API_ENROL_RAM

A fingerprint image must be present in RAM before starting the enrolment, either by capturing an image from the fingerprint sensor using the **API_CAPTURE_IMAGE** command, or the **API_DOWNLOAD_IMAGE**. The command returns with response when the enrolment is complete or if the enrolment fails for any reason. After enrolment, the template is stored in RAM and can be uploaded or moved to FLASH storage. No payload is sent with this command.

Response command

- API_ENROL_OK = Enrolment successful
- API_ENROL_FAIL = Enrolment failed

No payload is received with the response from this command.

7.6 Verify (RAM): API_VERIFY_RAM

A template and a fingerprint image must both be present in RAM before verification starts; use the **API_DOWNLOAD_TEMPLATE** command, or the **API_CAPTURE_ENROL_RAM** command.

To process the image, use the **API_DOWNLOAD_IMAGE** command, or the **API_CAPTURE_IMAGE** command. The verification can be started after one of these commands has been sent.

The command returns with a response when the verification is complete or if the verification fails for any reason. No payload is sent with this command.

Response command

- API_VERIFICATION_OK = Verification successful
- API_VERIFICATION_FAIL = Verification failed
- API_NO_TEMPLATE_PRESENT = No template present

No payload is received with the response from this command.

7.7 Verify (FLASH): API_VERIFY_FLASH

A fingerprint image must be present in RAM before starting the verification, (use one of the following commands: **API_DOWNLOAD_IMAGE**, **API_CAPTURE_IMAGE**. The FLASH slot number must be given in the IDX bytes. The command returns with response when the verification is complete or if the verification fails for any reason. No payload is sent with this command.

Response command:

- API_VERIFICATION_OK = Verification successful
- API_VERIFICATION_FAIL = Verification failed
- API_NO_TEMPLATE_PRESENT = No template in given FLASH slot
- API_INVALID_SLOT_NR = Wrong slot number

No payload is received with the response from this command.