

BM-Lite (FPC SafeTouch®)

Product Specification & Integration Guideline

Contents

1	Overview	5
1.1	General Description	5
1.2	Technical Features – BM-Lite	6
1.3	Technical Features – BM-Lite sensor	6
2	Functional Description	7
2.1	Operation	7
2.2	Fingerprint Sensor	7
2.3	Performance Characteristics	8
2.4	Verification Times	9
2.5	Enrollment	9
2.6	Expected lifetime	9
3	Signal List	10
4	Electrical Characteristics	11
4.1	Supply Voltage	11
4.2	Supply Current	11
4.3	Current Consumption graph	12
4.4	Digital Inputs	12
4.5	Digital Outputs	12
5	Software Command Interface	13
5.1	UART Serial Interface Settings	13
5.2	SPI Interface Settings	13
5.3	Command Send Structure	14
5.4	Frame format	16
5.5	Runtime modes	17
5.6	Example usage - Biometric functions	18
5.7	Example usage - Sensor operations	21
6	Command Tables	22
6.1	Biometric Commands	22
7	Command Descriptions	23
7.1	Capture	23
7.2	Enroll	23
7.3	Identify	23
7.4	Info	24
7.5	Image	24
7.6	Template	25
7.7	Wait	25
7.8	Sensor	25
7.9	Storage Template	26
7.10	UART Speed	27
8	Argument description	28
8.1	Finger down	28
8.2	Finger up	28
8.3	Start	28
8.4	Add	28
8.5	Finish	28
8.6	Id	28
8.7	All	28

8.8	Extract.....	29
8.9	Match	29
8.10	Get Version	29
8.11	Upload	29
8.12	Download	29
8.13	Delete	29
8.14	Data	29
8.15	Result.....	29
8.16	Count	30
8.17	Sleep	30
8.18	Timeout	30
8.19	Status.....	30
8.20	Version information.....	30
8.21	Unique ID information.....	31
8.22	Sensor Production Test.....	31
8.23	UART speed	31
9	CRC Calculation	32
10	Mechanical Properties	33
10.1	Part Drawings – BM-Lite.....	33
11	Integration and Configuration	36
11.1	Pin Configuration.....	36
11.2	Connector pin numbering.....	36
11.3	Power Supply Settings	36
11.4	Unused Pins.....	36
11.5	Mechanical Support.....	37
11.6	Galvanic Contact.....	37
11.7	ESD Protection.....	37
11.8	Mechanical Integration	37
11.9	Mounting recommendation	38
11.10	Connector	40
11.11	Flex Cable.....	41
12	HW Setup.....	42
13	SW Setup	45
13.1	Communication	45
13.2	Reference Code	45
14	Biometrics – Using BM-Lite.....	46
14.1	Usage guidelines.....	46
14.2	Finger placement.....	47
14.3	Enroll	47
14.4	Verify/Authenticate.....	48
14.5	Template Update.....	48
14.6	Performance	48
14.7	Use Cases.....	50
15	Product Identification	54
15.1	QR code	54
16	Product Updates	56
16.1	Product Configurations.....	56
17	Regulatory Compliance	57
17.1	EU RoHS2 (2011/65/EU).....	57



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(4/69)

17.2	Halogen Free	57
17.3	ODS	57
17.4	China RoHS (SJ/T 11363-2006)	57
17.5	REACH	57
18	Known Limitations	58
18.1	UART (only for CSW1.0)	58
18.2	UART Baudrate	58
18.3	Identify at Enroll	58
18.4	Delay After Matching	58
19	Revision history	60
20	Appendix – Python Host evaluation Application	61
20.1	HW setup	61
20.2	SW setup	63
20.3	Arguments for bep_ref.py	63
20.4	Startup procedure	64
20.5	Example procedures	64
21	Appendix – C-code reference application	66
21.1	HW setup FPC5832	66
21.2	HW setup FPC5924	67
21.3	SW setup	67
21.4	Arguments for bep_host_com[.exe]	68
21.5	Startup procedure	68
21.6	Example procedures	68

Terminology

BM	Biometric Module
CPU	Central Processing Unit
ESD	Electrostatic Discharge
FAR	False Acceptance Rate
FPC	Fingerprint Cards
FRR	False Rejection Rate
FW	Firmware, software that is an integrated part of a hardware component.
HCP	Host Command Protocol
HW	Hardware
LED	Light Emitting Diode
MCU	Microcontroller Unit
PCB	Printed Circuit Board
SPI	Serial Peripheral Interface
SW	Software
TU	Template Update
UART	Universal Asynchronous Receiver/Transmitter



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(5/69)

1 Overview

The BM-Lite module is a complete biometric fingerprint system, including a robust fingerprint sensor solution and on-board template storage. The biometric module is used connected to a host CPU over UART or SPI. Simple serial commands are used to enroll and verify.

Application examples

- Access control systems
- Time & Attendance
- Locks, safes
- USB tokens
- POS terminals

1.1 General Description

The BM-Lite sensor has a protective coating which helps to protect against ESD, scratches, impact and everyday wear-and-tear. All Fingerprints sensors feature 3D pixel sensing technology that can read virtually any finger; dry or wet.

The BM-Lite includes the following features:

- Embedded / stand-alone fingerprint identification system
- Compact sensor module package
- Protective sensor coating: scratch and ESD resistant
- One-to-one verification mode
 - Matching against 1 template
- Identify verification mode
 - 50 templates
- On-board template storage
 - Max. 50 templates
- Straightforward serial command interface
- Finger detect functionality
- EU RoHS compliant, China RoHS compliant, REACH compliant, ODS compliant and Halogen Free
- Front side IP rating: IPX7
- Easy to integrate and minimize time-to-market



The BM-Lite can be integrated into virtually any application and controlled by a host sending basic commands for enrollment and verification via the serial interface. Fingerprint templates are automatically created and stored in the internal flash memory.

1.2 Technical Features – BM-Lite

An overview of the technical features of the BM-Lite module is shown in Table 1.

Parameter	Description	Value	Unit
Processor board dimension	(W x L x H)	15 x 15 x 2.88	mm
Supply voltage	VDD typical	3.3	V
Supply Current	Wait-for-finger mode (default, 20 ms polling time)	0.05	mA
	Wait-for-finger mode (200 ms polling time)	0.01	mA
	Wait-for-command	5.4	mA
	Active mode	13	mA
ESD**	IEC61000-4-2, level X, Class B, air discharge	> ± 15	kV
	IEC61000-4-2, Class B, Contact discharge	± 8	kV

**With proper integration as outlined in this specification.

Table 1. Technical features overview

1.3 Technical Features – BM-Lite sensor

Sensor features of the BM-Lite is shown in Table 2.

SENSOR FEATURE	BM-Lite
Sensor matrix	160 x 160 pixels
Number of pixels	25 600 pixels
Active sensing area	8.0 x 8.0 mm
Spatial resolution	508 DPI
Pixel resolution	256 8-bit grayscale levels
Durability	10 million finger placements
Scratch resistance	Pencil hardness: 4H
Resistance to household substances	Water, carbonated soft drinks, coffee, oil, petrol, soap, artificial sweat, orange juice
RoHS / ACPEIP	EU RoHS 2 / China RoHS
Waterproofing	IPX7, front side (IP67 tests will be made on request)

Table 2. Sensor features of the BM-Lite

See section 10 for information on the mechanical properties of BM-Lite.

2 Functional Description

The BM-Lite module is a versatile fingerprint verification system which consists of two main components: a Fingerprints fingerprint sensor and an MCU.

The BM-Lite MCU acquires the fingerprint image from the fingerprint sensor. The flash memory on the BM-Lite is pre-loaded with firmware from Fingerprints and is used for all biometric operations and template storage.

2.1 Operation

BM-Lite operates connected to a host over UART or SPI.

The illustration in Figure 1 shows a system overview of BM-Lite connected to host.

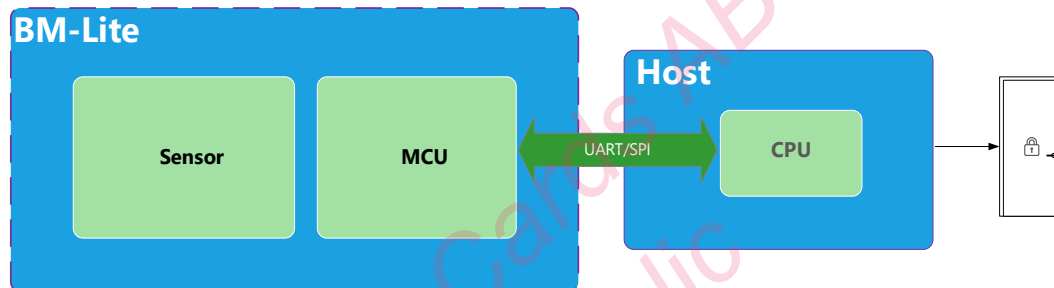


Figure 1. BM-Lite system overview

2.1.1 BM-Lite and Host

The BM-Lite is used as a slave towards a host. A host - selected and provided by the customer - executes the main application which interfaces with the BM-Lite module. Requirements on the host processor associated with the module communication are extremely low; therefore, the host processor can be selected entirely to suit the primary application.

Command Interface

The interface between the host and the processor board is based on a serial UART or an up to 20 MHz SPI command interface. There is no direct interaction between the host processor and the fingerprint sensor.

BM-Lite supports both UART and SPI communication interface. BM-Lite automatically detects the specific communication interface in use. However, it is not possible to use both interfaces at the same time. Also note that the input pins of the unused interface must be held at fixed levels to avoid interferences.

2.2 Fingerprint Sensor

The BM-Lite sensor is based on capacitive technology and utilizes a reflective measurement method. This method requires a galvanic contact point outside the sensor chip – this is achieved by means of a conductive bezel integrated into the sensor package.

To obtain good quality images, it is important that the sensor is correctly mounted in an enclosure. See section 11 for more information on integration.

2.2.1 Absolute Maximum Ratings

Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Stress beyond the values outlined in this section may cause permanent damage to the fingerprint sensors. Operation of the device in conditions beyond those indicated as normal operation in this specification is not implied or supported.



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(8/69)

BM-Lite ratings

The values in Table 3 indicate absolute maximum ratings for the BM-Lite.

Parameter	Absolute maximum value	Unit
VDDIO	-0.3 to +4.0	V
Operating temperature	-40 to +70	°C
Storage temperature	-40 to +85	°C

Table 3. Absolute maximum ratings for BM-Lite

2.3 Performance Characteristics

This section describes the performance characteristics for the BM-Lite sensor.

The FRR and FAR values for the verification mode is shown in Table 4.

FRR and FAR

Parameter	1 User	Unit
FRR (False-rejection-rate)	3.5 (after enroll) 1.4 (after template update (TU) = after 50 successful touches)	%
FAR (False-acceptance-rate)	1/500k (per finger)	-

Table 4. Performance characteristics – FRR / FAR



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(9/69)

2.4 Verification Times

Typical processing times for the biometric verification are shown in Table 5. Processing time starts at finger touch and ends with output to host via communication interface.

Process	Min	Typical	Max	Unit
Verification (last template of 50 enrolled fingers)	-	550	-	ms

Table 5. Biometric processing times

2.5 Enrollment

BM-Lite supports 3 enrollment touches.

2.6 Expected lifetime

The expected lifetime of the module is 5 years in a typical consumer usage/indoor environment.

3 Signal List

This section gives an overview of the signals sent between the BM-Lite and a host.

Pin number	Signal Name	Function	Description	Min. Voltage [V]	Max. Voltage [V]	Max. frequency [MHz]
1	GND	Supply	GND pin	0	0	DC
2	SPICLK	Digital in	Serial clock used for SPI	GND	3.6	20
3	MISO	Digital out	The SPI Serial tri-state output which is enabled with CS_N low	GND	3.6	SPICLK
4	MOSI	Digital in	The SPI Serial data Input	GND	3.6	SPICLK
5	CS_N	Digital in	Chip Select active when low	GND	3.6	~0
6	READY	Digital out	Data ready output (SPI only)	GND	3.6	~0
7	RST_N	Digital in	Reset sensor. Active when low*	GND	3.6	~0
8	UART_RX	Digital in	UART receive data pin	GND	3.6	921600** (Baud)
9	UART_TX	Digital out	UART transmit data pin	GND	3.6	921600** (Baud)
10	VDDIO	Supply	Core supply voltage. Typical 3.3V	2.0	3.6	DC

Table 6. Host signals

* Keep low at least 20µs to reset whole module.

** Default baud rate is 115200.

Switches

The host system is responsible for handling any ESD events that may happen via the external switches.

4 Electrical Characteristics

The electrical characteristics of the BM-Lite are described in this section. All signals except the power supply rails and ground are nominally 3.3 V.

4.1 Supply Voltage

Supply voltage for the BM-Lite are shown in Table 7.

Symbol	Parameter	Minimum	Typical	Maximum	Unit
VDD	Core supply, regulated input 3.3 V	2.0	3.3	3.6	V

Table 7. Supply Voltage

See section 11.3 for more information on power supply settings.

Ripple and Noise Level

Ripple and noise level must be ≤ 20 mVpp with 20 MHz bandwidth. The rise and fall time of the host power interface must be slower than 10 μ s/V at power-up and power-down to ensure MCU integrity.

4.2 Supply Current

Supply current for the BM-Lite (@3.3V) is shown in Table 8.

SYMBOL	PARAMETER	CONDITION	Minimum	Typical	Maximum	UNIT
IDD Supply current		Wait-for-finger mode (default, 20 ms polling time)*	-	0.05	-	mA
		Wait-for-finger mode (200 ms polling time)*	-	0.01	-	mA
		Wait-for-command		5.4		mA
		Active mode	-	13	18	mA

Table 8. Supply Current. *) Wait-for-finger mode is entered when CMD_CAPTURE or CMD_WAIT(ARG_FINGER_DOWN) command is sent to BM-Lite. Note! In previous BM-Lite documentations, this mode was called "Idle mode".

4.3 Current Consumption graph

An example current consumption graph for authentication is shown Figure 2.

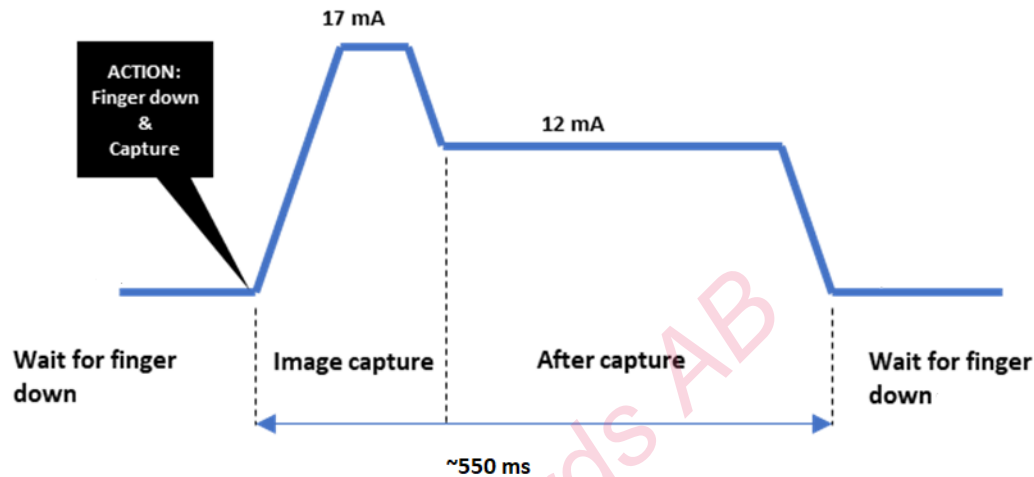


Figure 2. Illustration BM-Lite Current Consumption graph, Identify procedure (UART mode)

4.4 Digital Inputs

Digital inputs for the BM-Lite are shown in Table 9.

SYMBOL	PARAMETER	Minimum	Typical	Maximum	UNIT
V_{IL}	Logic '0' voltage	0	-	$0.3 V_{DD}$	V
V_{IH}	Logic '1' voltage	$0.7 V_{DD}$	-	$V_{DD}+0.3$	V
I_{IL}	Logic '0' current ($V_I = GND$)	-	-	± 10	μA
I_{IH}	Logic '1' current ($V_I = V_{DD}$)	-	-	± 10	μA

Table 9. Digital Inputs

4.5 Digital Outputs

4.5.1 Digital Output voltage

Digital output voltage specifications for the BM-Lite are shown in Table 10.

SYMBOL	PARAMETER	Minimum	Typical	Maximum	UNIT
V_{OL}	Logic '0' output voltage	-	-	0.4	V
V_{OH}	Logic '1' output voltage	$V_{DD}-0.4$	-	-	V

Table 10. Digital Output voltage

4.5.2 Digital Output current

Digital output current specifications for the BM-Lite are shown in Table 11.

SYMBOL	Ratings	Maximum	UNIT
$I_{IO}(PIN)$	Output current sourced by any I/O and control pin	20	mA
$\Sigma I_{IO}(PIN)$	Total output current sourced by sum of all I/Os and control pins ⁽¹⁾	100	mA

Table 11. Digital Output current

- (1) This current consumption must be correctly distributed over all I/Os and control pins. The total output current must not be sunk/sourced between two consecutive power supply pins referring to high pin count QFP packages.

5 Software Command Interface

To communicate with BM-Lite, a serial command interface is used between a host processor and BM-Lite. This interface is designed to be easy to use and performs the basic biometric functions needed in a fingerprint authentication system. Communication can be either UART or SPI. BM-Lite will detect which protocol being used and continue using that protocol for the power-on session. It is important that the input pins of the unused protocol are fixed and not floating. This is to avoid interferences.

5.1 UART Serial Interface Settings

The software settings for the serial protocol using UART are outlined in Table 12.

Parameter	Value
Communication speed	Factory default baud rate set to 115200* baud.
Format	8 data bits, parity none, one stop bit.
Bit order	Least significant bit first

Table 12. UART interface settings

* Valid for SW version ≥ 1.2 (See chapter 18.2 for more information)

5.1.1 UART Timing Diagram

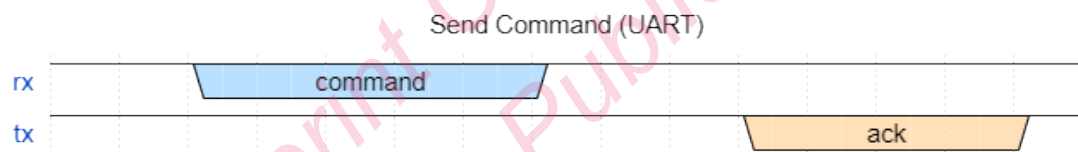


Figure 3. UART timing diagram (Send Command)

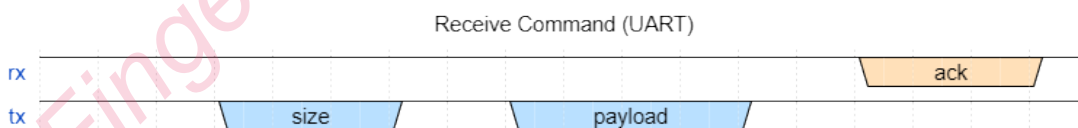


Figure 4. Figure 4 UART timing diagram (Receive Command)

5.2 SPI Interface Settings

The software settings for using SPI commands are outlined in Table 13.

Note that the length of the SPI connection cable shall be shorter than 20cm.

Parameter	Value
Communication speed	Guaranteed maximum speed: 20 MHz. (measured at 5cm cable length)
SPI Mode	Mode 0
Chip Select	Active Low

Table 13. SPI interface settings

The data ready signal (pin 6: READY) is used to signal to the host when BM-Lite has data available for the host to read over SPI.

5.2.1 SPI Timing Diagram

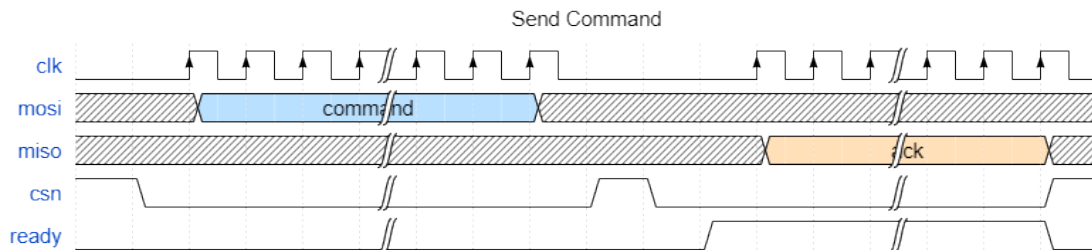


Figure 5. SPI Timing Diagram for SPI mode 0 (Send Command)

5.2.2 SPI Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$t_{su}(CS_N)$	CS_N setup time	Slave mode	$4 \times T_{PCLK}$	-	-	ns
$t_{h}(CS_N)$	CS_N hold time	Slave mode	$2 \times T_{PCLK}$	-	-	ns
$t_{su}(SI)$	Data input setup time	Slave mode	1.5	-	-	ns
$t_{h}(SI)$	Data input hold time	Slave mode	1.5	-	-	ns
$t_{a}(SO)$	Data output access time	Slave mode	9	-	36	ns
$t_{dis}(SO)$	Data output disable time	Slave mode	9	-	16	ns
$t_{v}(SO)$	Data output valid time	Slave mode $2.7 < V_{DD} < 3.6$ V Voltage Range 1	-	12.5	13.5	ns
		Slave mode $1.71 < V_{DD} < 3.6$ V Voltage Range 1	-	12.5	24	
		Slave mode $1.71 < V_{DD} < 3.6$ V Voltage Range 2	-	12.5	33	
$t_{h}(SO)$	Data output hold time	Slave mode	7	-	-	ns

Figure 6. SPI Characteristics

5.3 Command Send Structure

5.3.1 Overview

The communication stack implemented on the embedded devices by Fingerprints follows this specification.

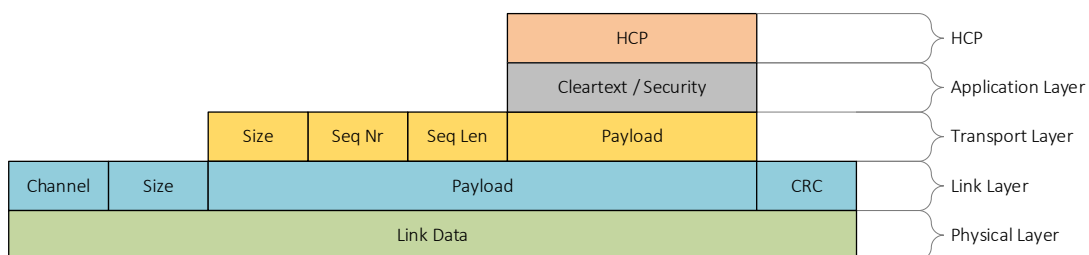


Figure 7. Overview of the command send structure



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(15/69)

5.3.2 Physical layer

The physical layer has a fixed size buffer of 256 bytes.

5.3.2.1 ACK packet

After each received HCP packet the receiving side shall respond with an ACK packet. This is a 4-byte packet, with data as follows:

0x7F	0xFF	0x01	0x7F
------	------	------	------

Note that this packet is sent standalone, and not as an HCP encapsulated packet.

5.3.3 Link layer

The link layer handles packet consistency.

Each packet received is acknowledged on the link layer, if an error occurs no retransmissions is done on this level instead the error is propagated upwards.

Channel	Size	Payload	CRC
2 bytes	2 bytes	size bytes	4 bytes

Table 14. Link layer

- Channel is not used in BM-Lite and can be set to zero.
- Size reflects the Payload size only.
- CRC is calculated on the Payload only.

5.3.4 Transport layer

The transport layer handles packet segmentation.

As the PHY MTU is 256 bytes the maximum payload per segment is 242 bytes.

Errors are propagated upwards.

Size	Seq Nr	Seq Len	Payload
2 bytes	2 bytes	2 bytes	size bytes

Table 15. Transport layer

- Size is the size of Payload in bytes.
- Seq Nr is the current packet chunk number, in case of a split packet. Starts at 1.
- Seq Len is the total number of chunks, in case of a split packet.

5.4 Frame format

Each HCP frame consist of a command and a variable number of arguments.

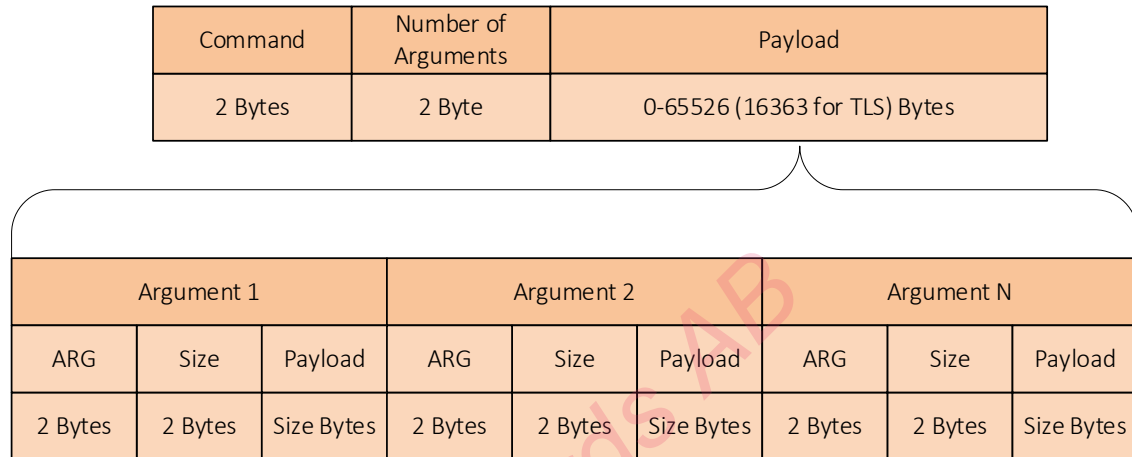


Figure 8. Frame format illustration

The Commands define the general action that is going to be executed. However, each command can have several arguments each, with their own data attached.

CMD	Number of arguments	Payload
2 bytes	2 bytes	arg_size bytes

Table 16. Command

The Argument is used as a complement to the command if it is needed and can contain arbitrary data. The order of the arguments is not relevant. It is not allowed to have the same argument key multiple times. The size of the argument payload can be zero.

ARG	Size	Payload
2 bytes	2 bytes	size bytes

Table 17. Argument

5.5 Runtime modes

BM-Lite has two runtime modes; Wait-for-finger mode (previously named “Idle mode”) and Active mode. They are described in this section.

5.5.1 Wait-for-finger mode

In Wait-for-finger mode (previously named “Idle mode”) BM-Lite is in its lowest power consumption state. BM-Lite wakes up if the user puts a finger on the sensor or if the host MCU starts communication with BM-Lite on one of the communication interfaces.

BM-Lite will not enter Wait-for-finger mode automatically, it must be initiated by the host. Wait-for-finger mode is entered when `CMD_CAPTURE` or `CMD_WAIT(ARG_FINGER_DOWN)` command is sent to BM-Lite. (There is no need to execute `CMD_WAIT` before `CMD_CAPTURE` as they both wait for finger in Wait-for-finger mode!)

Note: Make sure to put BM-Lite in Wait-for-finger mode, to save power, after all major procedures.

5.5.2 Active mode

In Active mode BM-Lite is in its highest power consumption state, and all major procedures are performed in this mode:

- **Await command state**
Await command state is the default state after BM-Lite starts up (boot) and after Enroll-, Identify- and Delete procedure. In Await command state BM-Lite waits for instructions from the host.
- **Enroll procedure**
In the Enroll procedure a user fingerprint is enrolled into BM-Lite.
- **Identify procedure**
In the Identify procedure a user fingerprint is matched against the enrolled fingerprints to decide if it is a match or not.
- **Delete procedure**
In the Delete procedure all enrolled fingerprints are deleted from BM-Lite

5.6 Example usage - Biometric functions

Examples of how to use the biometric functions are described in this section. See section 7 on detail about the different commands, and the reference code.

5.6.1 Capture

To start a capture *CMD_CAPTURE(ARG_TIMEOUT(timeout in ms))* shall be sent from host to BM-Lite. BM-Lite will wait for finger down, and when a finger is present on the sensor a capture will be executed. When image is captured BM-Lite returns *CMD_CAPTURE(ARG_RESULT(bep_result))* to host.

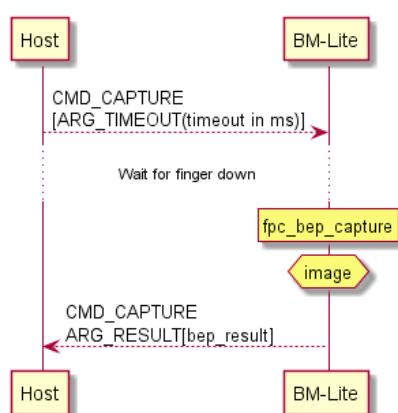


Figure 9. Capture sequence

5.6.2 Enroll

To start the enrollment *CMD_ENROLL(ARG_START)* shall be sent from host to BM-Lite. BM-Lite will start enroll session and send *CMD_ENROLL(ARG_RESULT(bep_result))* back to host. BM-Lite waits for finger down and when a finger is present on the sensor an image capture will execute (see section 5.6.1). After image capture is complete host sends *CMD_ENROLL(ARG_ADD)* to BM-Lite and the image is enrolled. BM-Lite sends *CMD_ENROLL(ARG_COUNT(samples_remaining), ARG_RESULT(bep_result))* back to host with information about how many finger touches are left in the enrollment process. BM-Lite waits for finger up (see section 5.7.1) and if more images are necessary to complete the enrollment the procedure will repeat itself. When BM-Lite has returned that 0 samples are remaining, Host sends *CMD_ENROLL(ARG_FINISH)* to BM-Lite which ends the enrollment procedure. BM-Lite sends *CMD_ENROLL(ARG_RESULT(bep_result))* back to host to verify that the enrollment procedure has ended.

Note: BM-Lite supports 3 enrollment touches.

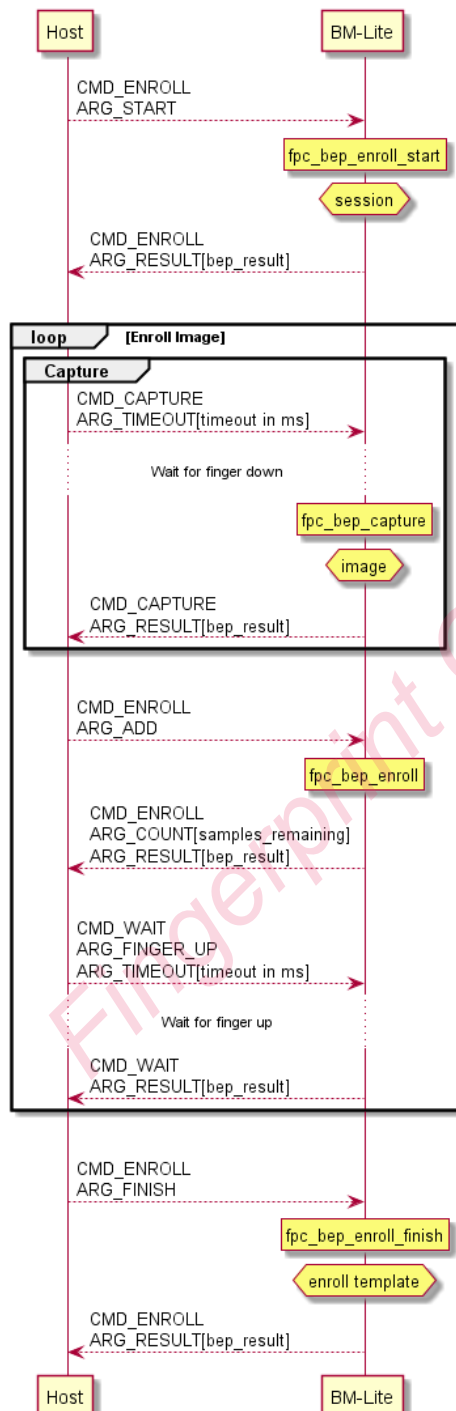


Figure 10. Enroll sequence

5.6.3 Identify

An Identity procedure starts with an image capture (see section 5.6.1). When image capture is done host sends `CMD_IMAGE(ARG_EXTRACT)` to BM-Lite to extract the image. When the image is extracted BM-Lite sends `CMD_IMAGE(ARG_RESULT(bep_result))` back to host. To start the identification process Host sends `CMD_IDENTIFY` to BM-Lite. When the identification process is done BM-Lite sends `CMD_IDENTIFY(ARG_RESULT(bep_result), ARG_ID(id))` back to host with the result of the identification process. If there was a match with an existing enrolled finger image an id will be provided.

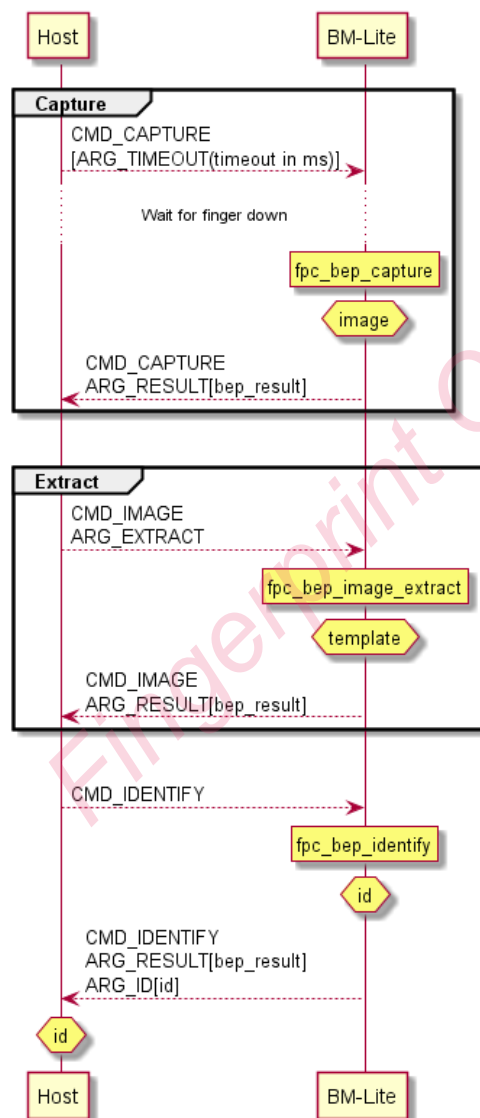


Figure 11. Identify sequence

5.7 Example usage - Sensor operations

5.7.1 Wait for finger up

Host sends `CMD_WAIT(ARG_FINGER_UP(ARG_TIMEOUT(timeout in ms)))` to BM-Lite to start waiting for finger up procedure. When BM-Lite senses finger up, `CMD_WAIT(ARG_RESULT(bep_result))` is sent back to host.

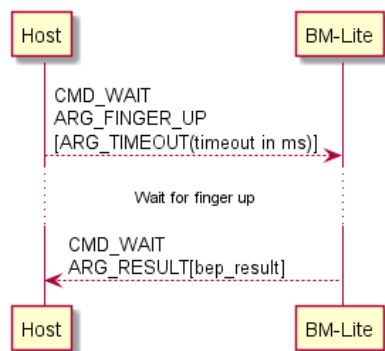


Figure 12. Wait for finger up sequence

5.7.2 Wait for finger down

Host sends `CMD_CAPTURE` or `CMD_WAIT(ARG_FINGER_DOWN(ARG_TIMEOUT(timeout in ms)))` to BM-Lite to start waiting for finger down procedure. While waiting for finger down BM-Lite is in Wait-for-finger mode.

(The difference between `CMD_CAPTURE` and `CMD_WAIT(ARG_FINGER_DOWN)` is that `CMD_CAPTURE` does capture after detecting finger touch.)

When BM-Lite senses finger down, `CMD_WAIT(ARG_RESULT(bep_result))` is sent back to host.

Note:

- **These commands must be sent from host to BM-Lite after startup (Boot), Identify or Enroll to set BM-Lite in Wait-for-finger mode.**
- **BM-Lite will not enter Wait-for-finger mode automatically.**
- **The only way to break the waiting for a timeout or a finger touch is to reset BM-Lite.**

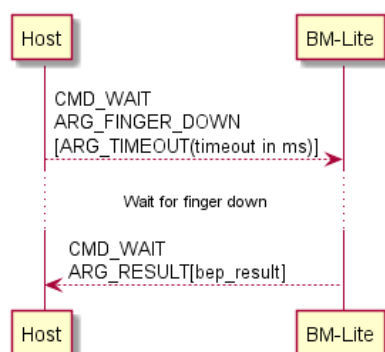


Figure 13. Wait for finger down sequence

6 Command Tables

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

6.1 Biometric Commands

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

Command	Hex	Note
CMD_CAPTURE	0x0001	Capture
CMD_ENROLL	0x0002	Enroll
CMD_IDENTIFY	0x0003	Identify
CMD_INFO (read out version of BM-lite)	0x3004	Info
CMD_IMAGE	0x0005	Image
CMD_TEMPLATE (manage templates)	0x0006	Template
CMD_WAIT (finger events)	0x0007	Wait
CMD_SENSOR (manage sensor)	0x1002	Sensor
CMD_STORAGE_TEMPLATE (manage template storage)	0x4002	Storage
CMD_COMMUNICATION (manage UART communication)	0x6001	UART baud rate

Table 18. Biometric Commands

7 Command Descriptions

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

7.1 Capture

Command name: CMD_CAPTURE

Value: 0x0001

7.1.1 Capture with wait for finger down

Command: CMD_CAPTURE

Argument: [ARG_TIMEOUT], [ARG_SLEEP]

Response: ARG_RESULT

Description: Wait in Wait-for-finger mode until a finger is detected on the sensor and then capture an image. There is an optional argument ARG_TIMEOUT that indicates a timeout value. If the timeout is provided, then the function returns without capturing an image if the timeout expires before the finger is detected on the sensor. If no timeout is provided the function waits infinite until a finger is detected on the sensor and then captures an image. The polling interval is dictated by the optional argument [ARG_SLEEP].

7.2 Enroll

Command name: CMD_ENROLL

Value: 0x0002

7.2.1 Enroll start

Command: CMD_ENROLL

Argument: ARG_START

Response: ARG_RESULT

Description: Starts the enroll phase in the module. Resources are allocated and initialized.

7.2.2 Enroll add image

Command: CMD_ENROLL

Argument: ARG_ADD

Response: ARG_COUNT, ARG_RESULT

Description: Extracts a template from the captured image. The template is added to the enroll template. The response argument ARG_COUNT informs about the remaining enroll templates required.

7.2.3 Enroll finish

Command: CMD_ENROLL

Argument: ARG_FINISH

Response: ARG_COUNT, ARG_RESULT

Description: Finishes the enroll sequence.

7.3 Identify

Command name: CMD_IDENTIFY

Value: 0x0003

7.3.1 Identify template

Command: CMD_IDENTIFY
Argument: none
Response: ARG_MATCH, ARG_ID, ARG_RESULT
Description: Runs an identification of the template in memory against the templates in storage. Extracts features from an image into a template.

7.4 Info

Command name: CMD_INFO
Value: 0x3004

7.4.1 Read version information

Command: CMD_INFO
Argument: ARG_GET
Response: ARG_VERSION
Description: Reads out version information from the device. The response contains a variable length string that contains version information of the device.

7.4.2 Read unique id

Command: CMD_INFO
Argument: ARG_UNIQUE_ID
Response: ARG_UNIQUE_ID, ARG_RESULT
Description: Reads out unique ID information from the device. The response contains 12 bytes of LSB data that contains unique ID information of the device.

7.4.3 Read reset reason

Command: CMD_INFO
Argument: ARG_STATUS
Response: ARG_STATUS
Description: Reads out reason for the last reset, see Table 19 Reset reasons for more information.

7.5 Image

Command name: CMD_IMAGE
Value: 0x0005

7.5.1 Feature extraction

Command: CMD_IMAGE
Argument: ARG_EXTRACT
Response: ARG_RESULT
Description: Extracts features from an image into a template.

7.5.2 Image upload to host

Command: CMD_IMAGE
Argument: ARG_UPLOAD
Response: ARG_DATA, ARG_RESULT

Description: Uploads an image to the host. The image is sent as pixel data with the size of the image in the ARG_DATA response. The image size is provided as argument size for ARG_DATA in the response.

7.6 Template

Command name: CMD_TEMPLATE

Value: 0x0006

7.6.1 Save template

Command: CMD_TEMPLATE

Argument: ARG_SAVE, ARG_ID

Response: ARG_RESULT

Description: Saves the enrolled template to the local storage. The template is assigned the ID provided by the argument ARG_ID.

7.7 Wait

Command name: CMD_WAIT

Value: 0x0007

7.7.1 Wait for finger on sensor

Command: CMD_WAIT

Argument: ARG_FINGER_DOWN, [ARG_TIMEOUT], [ARG_SLEEP]

Response: ARG_RESULT

Description: Waits in Wait-for-finger mode until a finger is detected on the sensor. There is an optional argument that indicates a timeout value. If the timeout is provided then the function returns if the timeout expires or a finger is detected on the sensor, whatever comes first. The polling interval is dictated by the optional argument [ARG_SLEEP].

7.7.2 Wait until no finger on sensor

Command: CMD_WAIT

Argument: ARG_FINGER_UP, [ARG_TIMEOUT], [ARG_SLEEP]

Response: ARG_RESULT

Description: Waits until no finger is detected on the sensor. There is an optional argument that indicates a timeout value. If the timeout is provided then the function returns if the timeout expires or no finger is detected on the sensor, whatever comes first. The polling interval is dictated by the optional argument [ARG_SLEEP].

7.8 Sensor

Command name: CMD_SENSOR

Value: 0x1002

7.8.1 Sensor reset

Command: CMD_SENSOR

Argument: ARG_RESET

Response: ARG_RESULT

Description: Performs a reset of the sensor.

7.8.2 Sensor production test

Command: CMD_SENSOR
Argument: ARG_PROD_TEST
Response: ARG_PROD_TEST, ARG_RESULT
Description: Read sensor production test result. The arguments ARG_PROD_TEST in the command does not contain any data.
 The response contains a sensor production test result in the argument ARG_PROD_TEST. The value is provided in the format as uint32_t value.

7.9 Storage Template

Command name: CMD_STORAGE_TEMPLATE
Value: 0x4002

7.9.1 Delete a specific template

Command: CMD_STORAGE_TEMPLATE
Argument: ARG_DELETE, ARG_ID
Response: ARG_RESULT
Description: Deletes the template with the ID provided in the argument ARG_ID.

7.9.2 Delete all templates

Command: CMD_STORAGE_TEMPLATE
Argument: ARG_DELETE, ARG_ALL
Response: ARG_RESULT
Description: Deletes all templates.

7.9.3 Load a template from flash to RAM

Command: CMD_STORAGE_TEMPLATE
Argument: ARG_UPLOAD, ARG_ID
Response: ARG_RESULT
Description: Loads the template with the ID provided in the argument ARG_ID into the local RAM.

7.9.4 Read template ID's

Command: CMD_STORAGE_TEMPLATE
Argument: ARG_ID
Response: ARG_DATA, ARG_RESULT
Description: Returns a list with all template ID's where a template is stored in the local flash memory.
 The response contains a list of all ID's in the argument ARG_DATA. The values are provided in the same format as ARG_ID as uint16_t values.

7.9.5 Get number of templates in local storage

Command: CMD_STORAGE_TEMPLATE
Argument: ARG_COUNT
Response: ARG_COUNT, ARG_RESULT
Description: Returns the number of templates in the local storage as parameter in the response argument ARG_COUNT. The argument ARG_COUNT in the command does not contain any data.

7.10 UART Speed

Command name: **CMD_COMMUNICATION**

Value: **0x6001**

7.10.1 Get supported UART baud rates

Command: **CMD_COMMUNICATION**

Argument: **ARG_SPEED, ARG_GET**

Response: **ARG_DATA, ARG_RESULT**

Description: Returns supported baud rates. The arguments ARG_SPEED and ARG_GET in the command do not contain any data. The response contains a list of all supported baud rates in the argument ARG_DATA. The values are provided in the format as uint32_t values.

7.10.2 Set UART baud rate

Command: **CMD_COMMUNICATION,**

Argument: **ARG_SPEED, ARG_SET, ARG_DATA**

Response: **ARG_RESULT**

Description: Returns supported baud rates. The arguments ARG_SPEED and ARG_SET in the command do not contain any data. The new baud rate is provided by the argument ARG_DATA.

8 Argument description

8.1 Finger down

Argument name: ARG_FINGER_DOWN

Value: 0x0001

Size: zero

Description: Indicates that the device will wait in Wait-for-finger mode until the finger is detected on the sensor before the command is executed.

8.2 Finger up

Argument name: ARG_FINGER_UP

Value: 0x0002

Size: zero

Description: Indicates that the device will wait until the finger is not anymore detected on the sensor before the command is executed.

8.3 Start

Argument name: ARG_START

Value: 0x0003

Size: zero

Description: Starts the specific command.

8.4 Add

Argument name: ARG_ADD

Value: 0x0004

Size: zero

Description: Adds some data to the function in progress that is specified by the command.

8.5 Finish

Argument name: ARG_FINISH

Value: 0x0005

Size: zero

Description: Finishes the specific command.

8.6 Id

Argument name: ARG_ID

Value: 0x0006

Size: uint16_t

Description: Specifies a template ID.

8.7 All

Argument name: ARG_ALL

Value: 0x0007

Size: zero

Description: Specifies that the command is applicable to all items. For more details see command description.

8.8 Extract

Argument name: ARG_EXTRACT

Value: 0x0008

Size: zero

Description: Requests the feature extraction from an image into a template.

8.9 Match

Argument name: ARG_MATCH

Value: 0x000A

Size: uint8_t

Description: Specifies a match result. The parameter value is 0 for NO_MATCH and 1 for MATCH_OK.

8.10 Get Version

Argument name: ARG_GET

Value: 0x1004

Size: zero

Description: Requests version information. See ARG_VERSION for response information.

8.11 Upload

Argument name: ARG_UPLOAD

Value: 0x1005

Size: zero

Description: Requests some data to be uploaded to the host. The data is specified by the command. See command description for more details.

8.12 Download

Argument name: ARG_DOWNLOAD

Value: 0x1006

Size: zero

Description: Requests some data to be downloaded from host. The data is specified by the command. See command description for more details.

8.13 Delete

Argument name: ARG_DELETE

Value: 0x1009

Size: zero

Description: Deletes the item(s) indicated by the command.

8.14 Data

Argument name: ARG_DATA

Value: 0x100A

Size: Variable. See command description for more details.

Description: Standard argument to provide unspecified data buffers in a command result. The format of the data is context specific and described in the command description where this argument is listed as response.

8.15 Result

Argument name: ARG_RESULT



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(30/69)

Value: 0x2001
 Size: uint16_t
 Description: Standard argument to provide a command result. The values of data attached to this argument are defined in the header file fpc_bep_types.h.

8.16 Count

Argument name: ARG_COUNT
 Value: 0x2002
 Size: uint32_t
 Description: Specifies a number. See command description for usage.

8.17 Sleep

Argument name: ARG_SLEEP
 Value: 0x4002
 Size: uint16_t
 Description: Specifies sleep time in milliseconds in the range of [0, 1020]. Default value is 20 milliseconds.

8.18 Timeout

Argument name: ARG_TIMEOUT
 Value: 0x5001
 Size: uint16_t
 Description: Timeout in milliseconds

8.19 Status

Argument name: ARG_STATUS
 Value: 0x6002
 Size: uint16_t
 Description: Reset reason defined in the table below:

Value	Description
0x0004	Set by hardware when a reset from the NRST pin occurs.
0x0008	Set by the hardware when brown out reset (BOR) occurs.
0x0010	Set by hardware when a software reset occurs.
0x0080	Set by hardware when a reset occurs due to illegal Stop, Standby or Shutdown mode entry.
0x0100	Set when hard fault exception has occurred.
0x0200	Set when bus fault exception has occurred.
0x0400	Set when mem manage exception has occurred.
0x0800	Set when usage fault exception has occurred.

Table 19 Reset reasons

8.20 Version information

Argument name: ARG_VERSION
 Value: 0x6003
 Size: Variable
 Description: Contains the BM-Lite firmware version information as a character string. The string is built up according to the following syntax:
 <name>:
 <release><major>.<minor>.<build><SPACE>(<date><SPACE><hh:mm:ss>)

Example: FPC BM-Lite: 1.1.20 (Mar 5 2019 01:19:47)

8.21 Unique ID information

Argument name: ARG_UNIQUE_ID

Value: 0x6004

Size: 12 bytes

Description: Get device unique ID information.

8.22 Sensor Production Test

Argument name: ARG_PROD_TEST

Value: 0x2009

Size: uint32_t

Description: Get sensor production test result as format as uint32_t. The parameter value is "zero" for OK and "non-zero" for FAIL.

8.23 UART speed

Argument name: ARG_SPEED

Value: 0x2008

Size: Variable

Description: Set or get UART baud rate



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(32/69)

9 CRC Calculation

The CRC calculation can be implemented as a table of pre-computed effects to ensure efficiency. The CRC value is 32 bits long. The table is indexed by the byte to be encoded and thus the table contains 256 double words (256 x 32 bits).

The CRC algorithm implementation was initially developed by the University of California, Berkeley and its contributors, but has been changed and somewhat simplified to fit the embedded nature of BM-Lite. The algorithm uses the CCITT-32 CRC Polynomial.

The source code for the CRC implementation is available from Fingerprints and can be compiled with limited impact in most environments.

10 Mechanical Properties

This section gives an overview of the mechanical properties for BM-Lite.

- All measurements are in millimeters.
- Dimensional data is based on nominal values.

10.1 Part Drawings – BM-Lite

The BM-Lite module (R03) is illustrated in this section as mechanical part drawings.

10.1.1 Top View – BM-Lite

A part drawing for the top view of the BM-Lite module is shown in Figure 14.

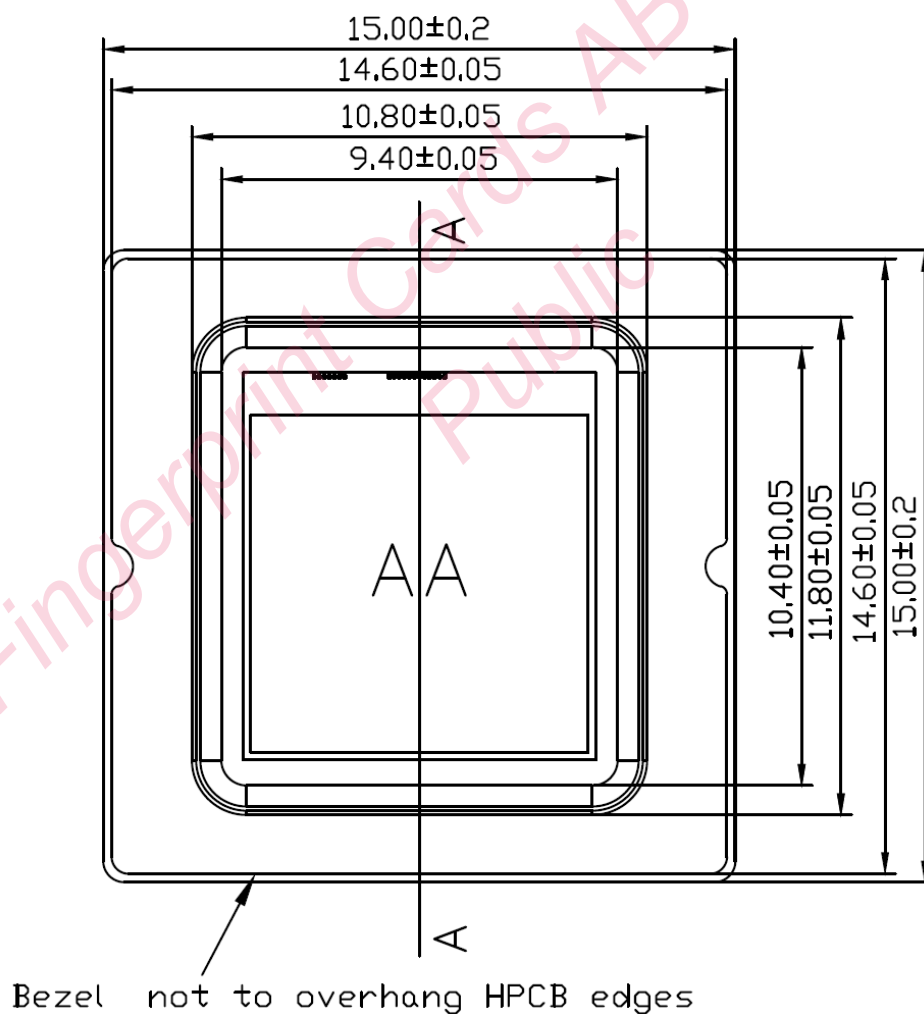


Figure 14. Part drawing – top view

10.1.2 Bottom View – BM-Lite

A part drawing for the bottom view of the BM-Lite module is shown in Figure 15.

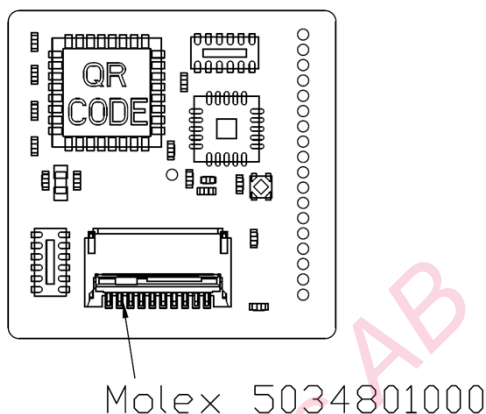


Figure 15. Part drawing – bottom view

Pinout

For details of pinout configuration on the connector, see section 11.1.

Note: The pitch of the test points (on the left-hand side) is 0.7mm.

10.1.3 Side View – BM-Lite

A part drawing for the side view of the BM-Lite module is shown in Figure 16.

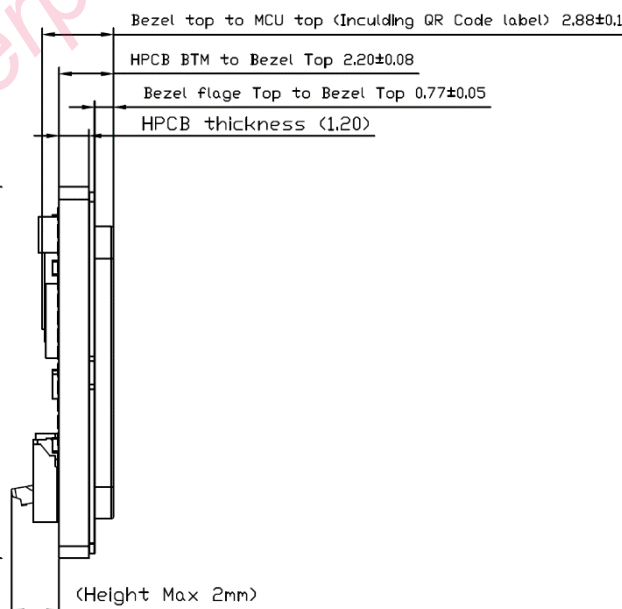


Figure 16. Part drawing – side view

10.1.4 Bezel

A part drawing for the side view of the BM-Lite Bezel is shown in Figure 17.

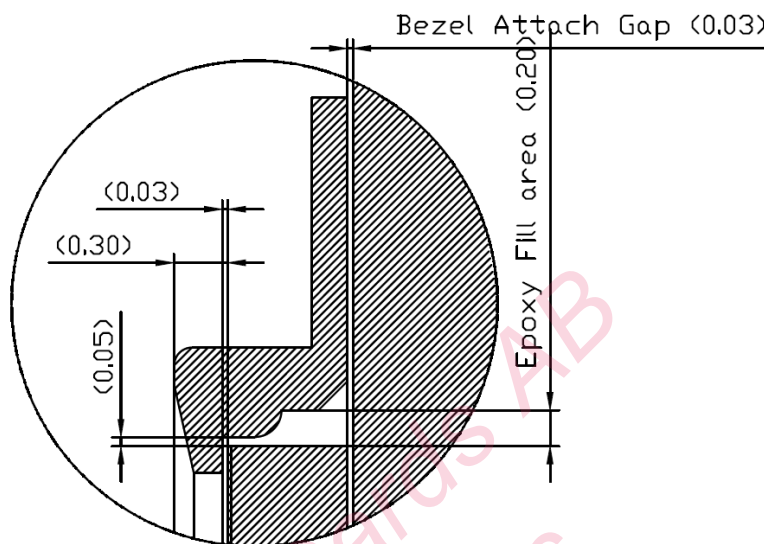


Figure 17. BM-Lite Bezel details

11 Integration and Configuration

This section contains recommendations for integrating the BM-Lite, including an overview of signaling, pin-pad and connector configuration. See section 12 for more information about HW setup.

11.1 Pin Configuration

BM-Lite pin configuration is outlined in Table 6 in section 3.

11.2 Connector pin numbering

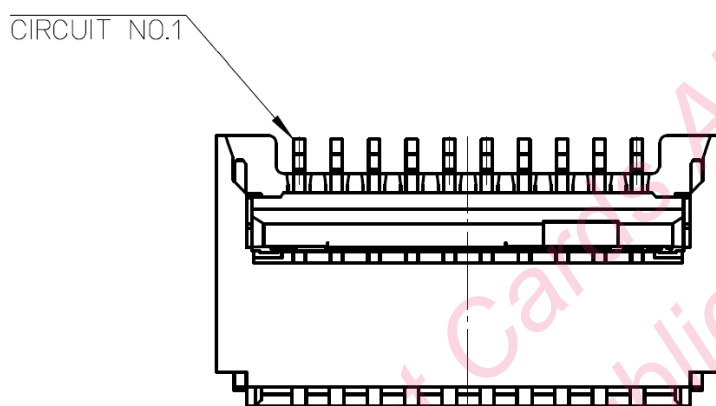


Figure 18. Pin numbering on connector, Molex 5034801000.

11.3 Power Supply Settings

See section 4.1 for more information on supply voltage.

11.4 Unused Pins

This section contains information on unused pins.

11.4.1 UART & SPI

If the UART or SPI host interface is not used, the unused pins are recommended to be connected as follows:

Unused UART

- UART_RX (8) pull-up to 3.3V

See section 12.1.1 for more information.

Unused SPI

- CS_N (5) pull-up to 3.3V

See section 12.1.3 for more information.

Note: A pull-up will possibly increase the power consumption.



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(37/69)

11.5 Mechanical Support

The best way to ensure a solid mount is to apply a stable, non-conductive, support to the back side of the BM-Lite module or by gluing it on the sides of the module towards the casing. See section 11.9 for more information.

Important! Mechanical force outside the maximum rating may cause permanent damage to the sensor.

11.6 Galvanic Contact

Due to the conductive bezel on the sensor module, a smooth transition to exterior mechanics can easily be obtained.

11.7 ESD Protection

BM-Lite has integrated ESD protection. ESD current is deflected across a protective coating on the sensor surface to the bezel surrounding the sensor. From the bezel, the discharged current is conducted via a Transient Voltage Suppressor (TVS) to the local ESD drain node ground pin (GND) on the PCB.

Power Supply Rails

The power supply rails do not need to be protected with an ESD device as they have bulk capacitors that will efficiently store any charge delivered by the ESD.

11.8 Mechanical Integration

Note. All images in this section are illustrations.

11.8.1 Positioning

The whole sensor area should be exposed to the finger.



Figure 19. Positioning of the sensor

11.8.2 Force

The module needs structural support from underneath. The mechanical integration shall withstand a pressure of at least 25N.



Figure 20. Integration shall withstand a pressure of at least 25N

11.9 Mounting recommendation

11.9.1 Key notes for mounting

- The BM-Lite module should be integrated into the system by strong adhesives, physical holder or the combination of both
- Twisting or bending of the module should be avoided during the integration.
- Excessive bending of flex should be avoided during assembly.

11.9.2 Integration Option #1

Suggestions for the internally assembled module:

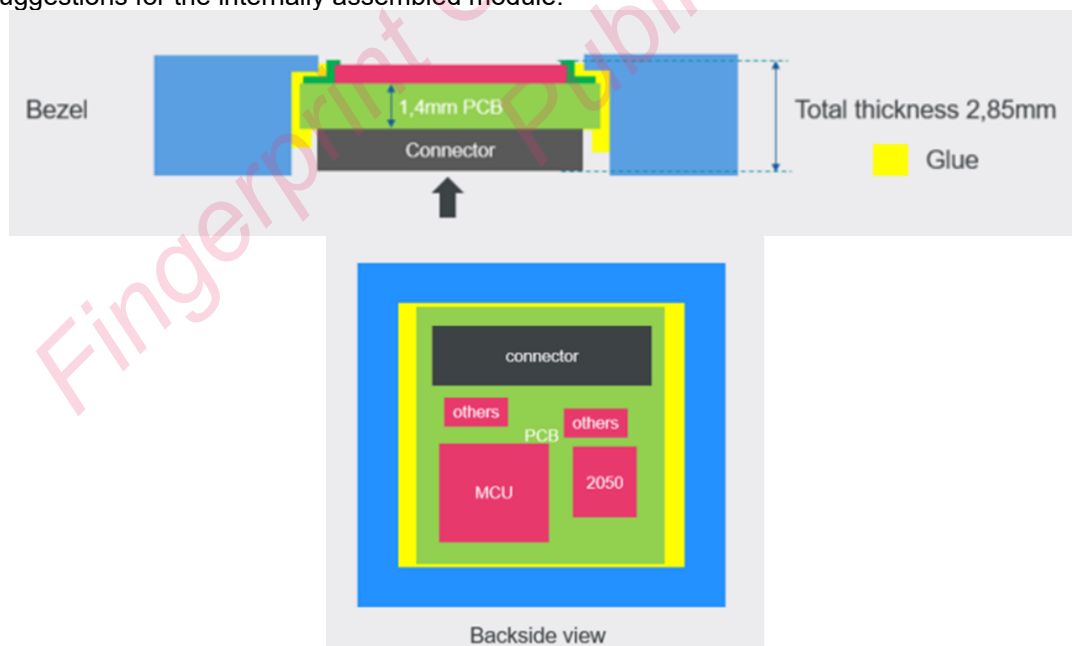


Figure 21. Integration option 1

This is an option where the module is integrated from the bottom of the host system's casing directly. It is suggested that the casing of the product has an opening that fits the dimension of BM-lite module. The attachment of BM-lite module is achieved by applying epoxy adhesive to both the "bezel" part and the side of "PCB" part. In addition, if applicable, extra adhesives could be applied to fill the space on the back side of the module, after attachment mentioned above, to increase the reliability of the integrated system.

11.9.3 Integration Option #2

Suggestions for the externally assembled module:

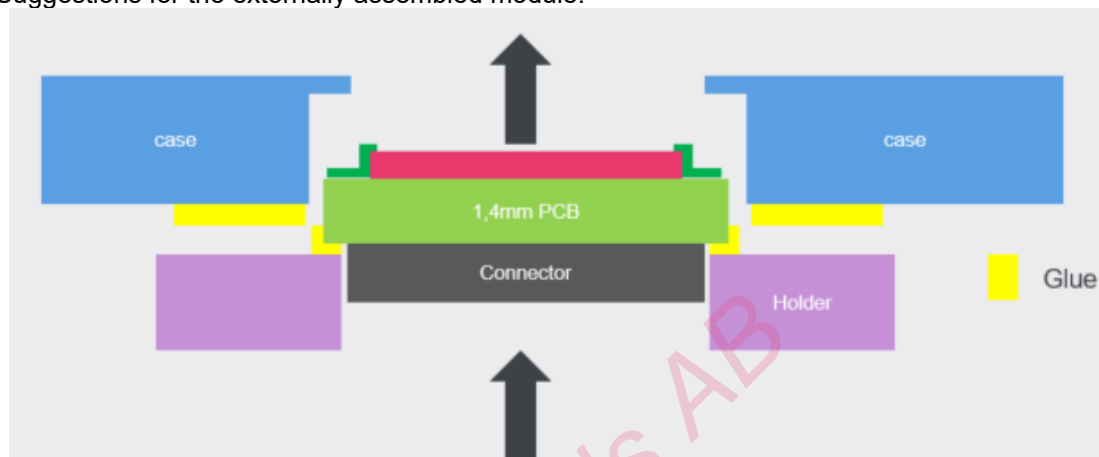


Figure 22. Integration option 2

As shown in above illustration, for the second option of integration includes one extra step before mounting the sensor module into product system from bottom.

BM-lite module will be sitting on a holder except the connector side, so that extra support is added to the module from bottom side. Then the holder and the module will be mounted into the system with epoxy adhesives that have larger surface contact, hence better reliability towards excessive pushing from top side.

11.10 Connector

The connector used on the BM-Lite module is a 10-pin Molex 5034801000. Details about the connector can be found at:

https://www.molex.com/molex/products/datasheet.jsp?part=active/5034801000_FFC_FPC_CO_NNECTORS.xml

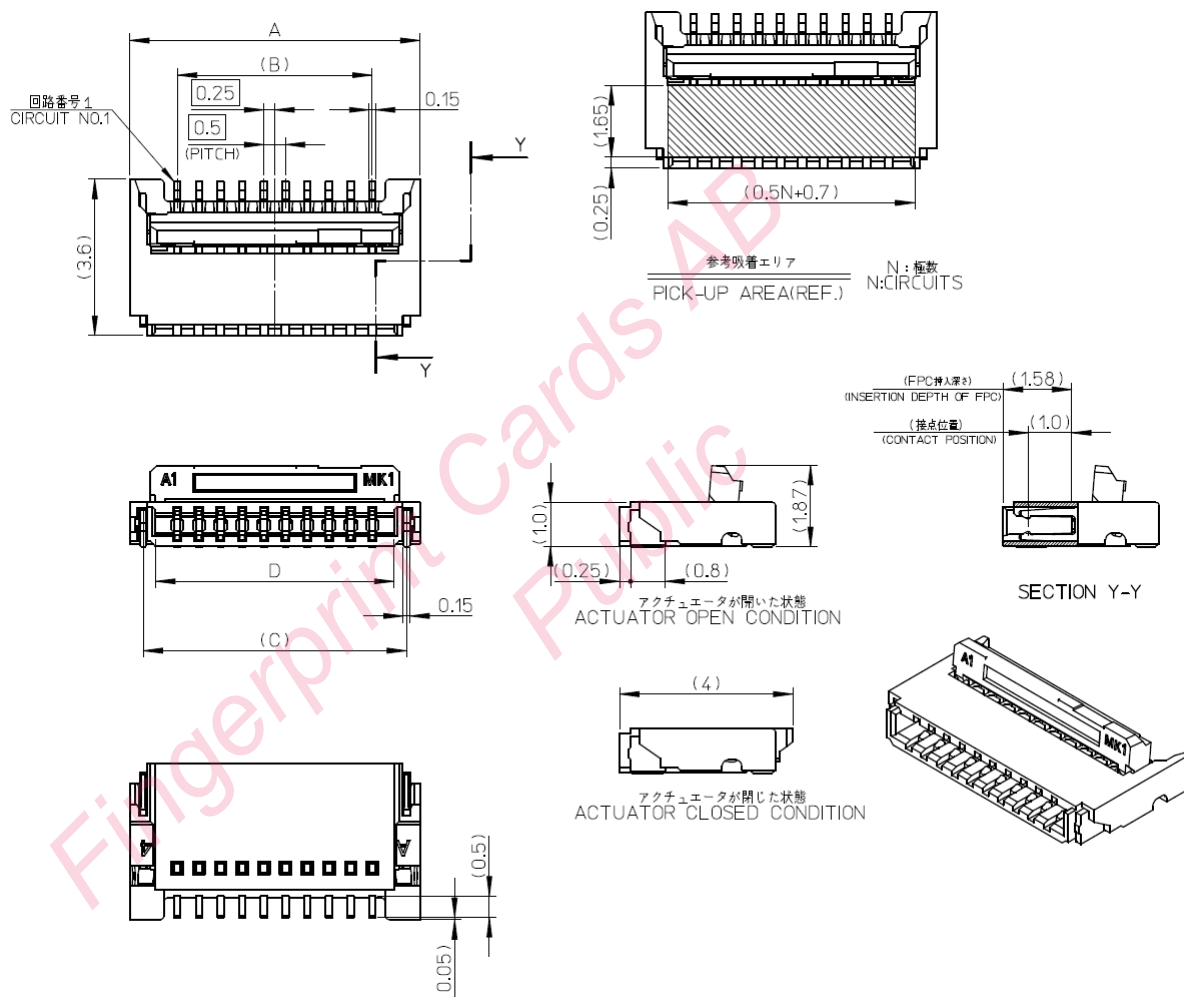


Figure 23. 10-pin Molex connector details.

11.11 Flex Cable

11.11.1 Flex Cable Design

The flex cable is a standard 10-pin with 0.5 mm pitch. The details for the recommended Fingerprints cable for the BM-Lite connector is shown in Figure 24.

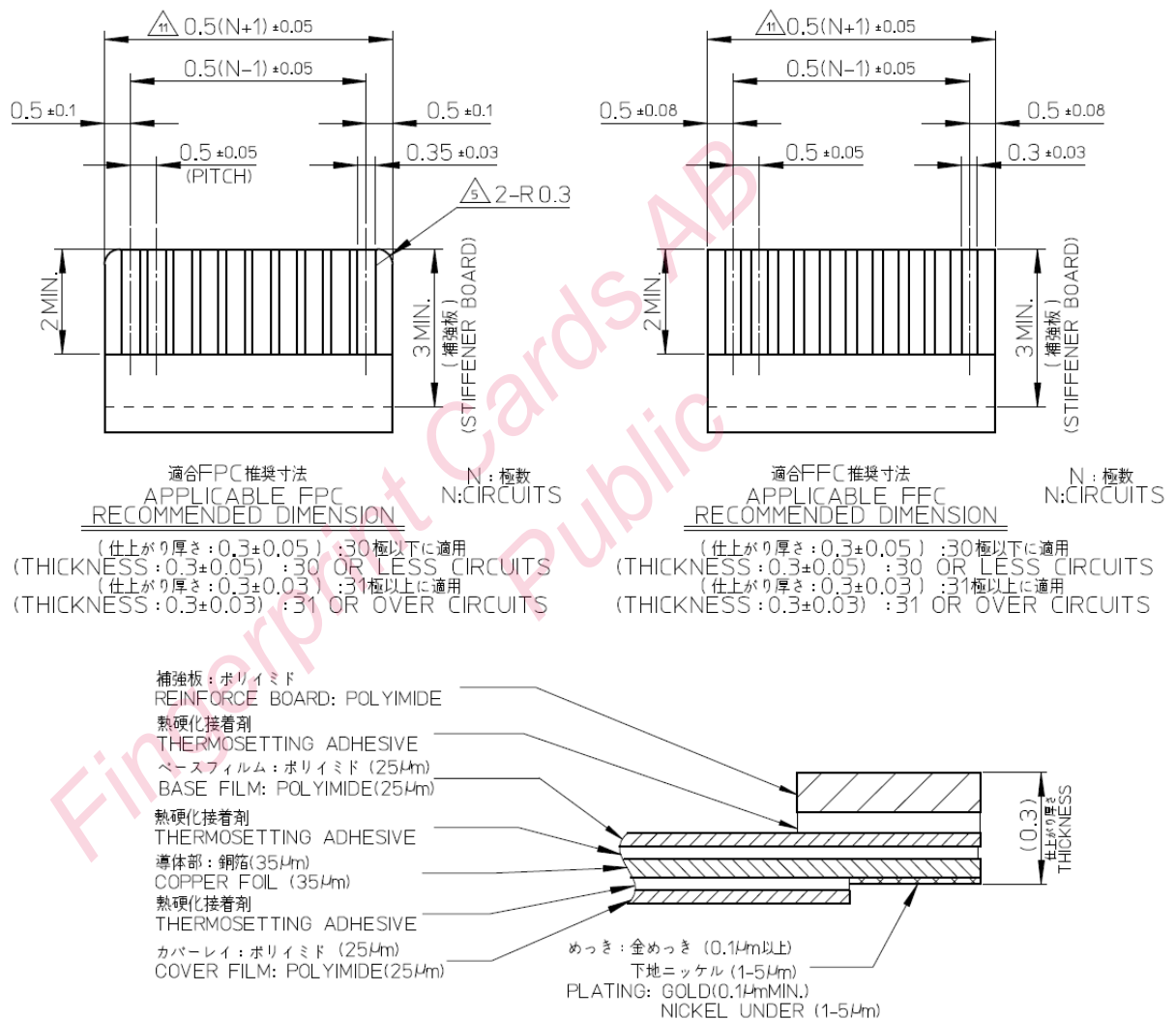


Figure 24. Cable details for the BM-Lite connector.

12 HW Setup

HW setup for version R03 and onwards is described in this appendix.

BM-Lite support both UART and SPI communication interface. BM-Lite automatically detects the specific communication interface in use. However, it is not possible to use both interfaces at the same time!

12.1.1 Communication pin specification

The BM-Lite internal configuration of the communication pins are presented in Table 20 for UART and Table 21 for SPI. This information is necessary to consider when designing the communication interface to another device, in regard to signal levels and states of the pins. Recommended interface connection details can be found in chapter 12.1.2 for UART and chapter 12.1.3 for SPI.

1.1.1.1 UART

Table 20. UART communication pin specification

Pin	Function	Configuration
UART_RX	UART receive data. (Default baud rate is 115200 baud from SW Version 1.2.X)	High impedance input mode, pull-up.
UART_TX	UART transmit data. (Default baud rate is 115200 baud from SW Version 1.2.X)	Push-Pull output mode, no pull-up or pull-down.

1.1.1.2

1.1.1.3 SPI

Table 21. SPI communication pin specification

Pin	Function	Configuration
SPICLK	Serial Clock input for SPI slave.	High impedance input mode, no pull-up or pull-down.
MISO	Master In / Slave Out data.	Push-Pull output mode, no pull-up or pull-down.
MOSI	Master Out / Slave In data.	High impedance input mode, no pull-up or pull-down.
CS_N	Chip Select, active when low.	High impedance input mode, no pull-up or pull-down.
READY	Sensor data available, active when high. (Stays high until BM-Lite has data to send)	Push-Pull output mode, no pull-up or pull-down.
RST_N	Sensor Module reset, active when low (Keep low at least 20μs to reset whole module).	High impedance input mode, no pull-up or pull-down.

12.1.2 UART interface

UART interface connection details are shown in Figure 25.

When using the UART interface; it is recommended to inhibit the SPI interface by adding a pull-up (<300kΩ) to the CS_N pin. *Note: A pull-up will possibly increase the power consumption.*

It is necessary for the host side to make sure that the RST_N is managed so that no unwanted resets will occur, i.e. RST_N shall be kept at VDD-level (high).

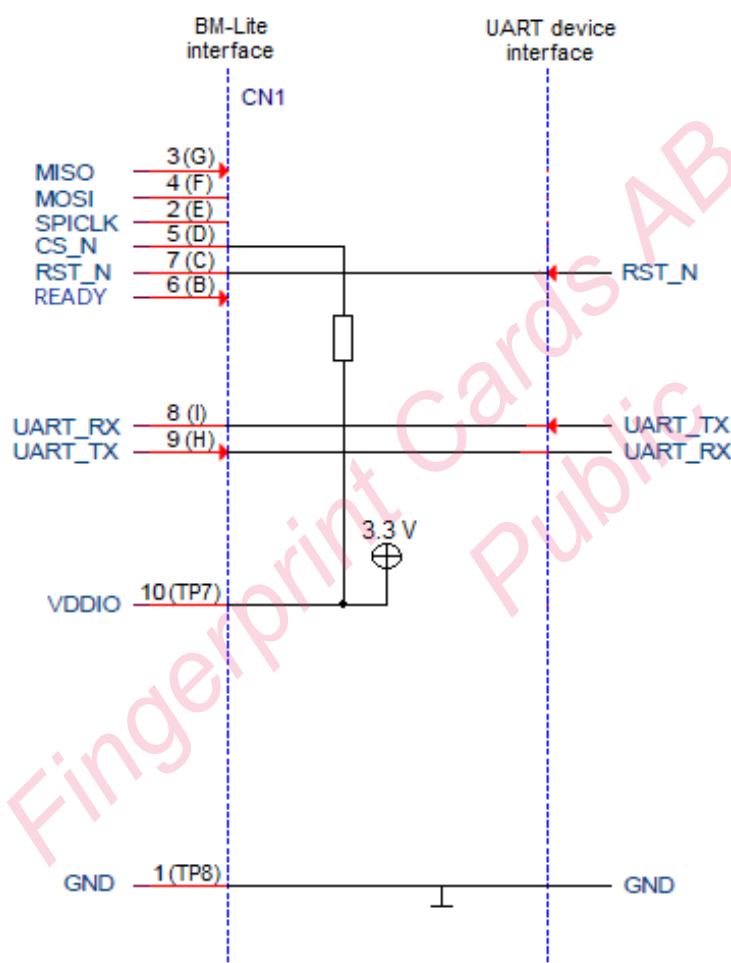


Figure 25. UART connection details

12.1.3 SPI interface

SPI interface connection details are shown in Figure 26.

Note that the length of the SPI connection cable shall be shorter than 20cm!

When using the SPI interface; it is recommended to inhibit the UART interface by adding a pull-up (<300k Ω) to the UART_RX pin. *Note: A pull-up will possibly increase the power consumption.*

It is necessary for the host side to make sure that the RST_N is managed so that no unwanted resets will occur, i.e. RST_N shall be kept at VDD-level (high). (*Note: RST_N is connected to an internal permanent pull-up resistor*)

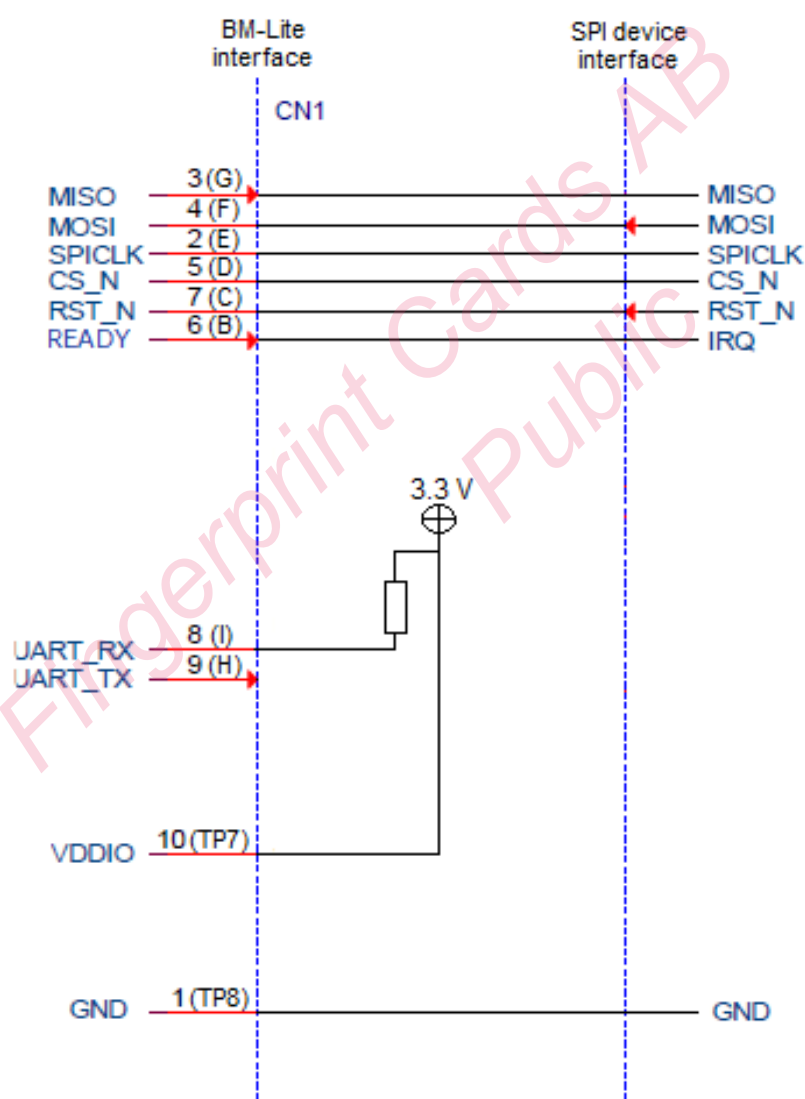


Figure 26. SPI interface connection details



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(45/69)

13 SW Setup

13.1 Communication

BM-Lite support both UART and SPI communication interface. BM-Lite automatically detects the specific communication interface in use. However, it is not possible to use both interfaces at the same time! See section 12 for HW setup information.

See section 5 - 7 for information about SW command interface, command tables and command descriptions.

13.2 Reference Code

The reference code includes examples of all functionalities of BM-Lite, and it is recommended to follow these examples to get started with BM-Lite.

The reference code can be found as part of the BM-Lite release.

14 Biometrics – Using BM-Lite

- Fingerprints are unique for every individual
 - Biometrics is by definition directly linked to the specific person
- Prevents identity fraud
 - Unlike biometrics; passwords and codes can be easily stolen or guessed
 - False Acceptance rate (FAR):
 - PIN code (4-digits): 0.01 %
 - Biometrics: typically 0.002 %
- High convenience
 - Fast verifications
 - Automated recognition of an individual's characteristics
 - False rejection rate (FRR): 0.5 – 2.5 %
 - Easy to use
 - No codes or passwords to remember
 - Biometrics is always available and easy to use

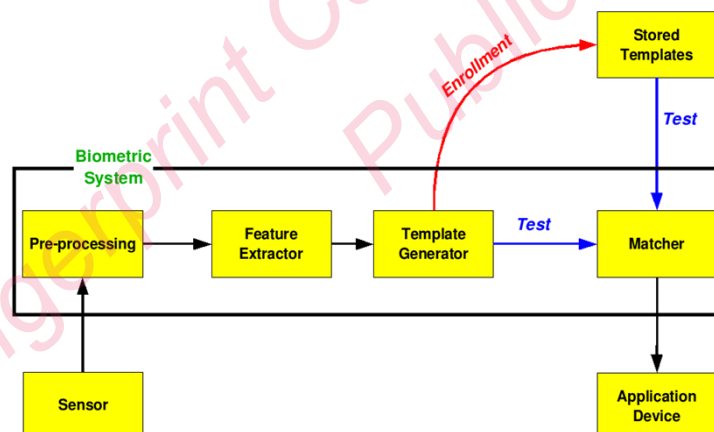


Figure 27. Biometric system illustration

14.1 Usage guidelines

Following are what information the end users might need before and during usage of the integrated fingerprint sensor

- The user needs to know:
 - *That* enrollment must be done
 - *Why* enrollment must be done
 - *How* enrollment should be done
- To register one's fingerprint could be considered leaving sensitive and personal data, therefore the user needs to know *what* data is stored and *how* it is stored
- A thorough guiding (e.g. on a screen) will be necessary for the user to complete the enrollment successfully and thereby acquire full performance of the sensor
- The user will need feedback during verification to understand when to verify, if the verification succeeded etc. e.g. visual feedback, haptic feedback and/or auditory feedback

14.2 Finger placement

Following are some general information on how to properly put your finger on the sensor

- The lower part of the fingertip should be exposed to the sensor, as shown in the upper left picture
- The sensor needs to be fully covered by the finger
- Keep the finger straight and fully parallel to the sensor, as shown in the upper right and lower pictures

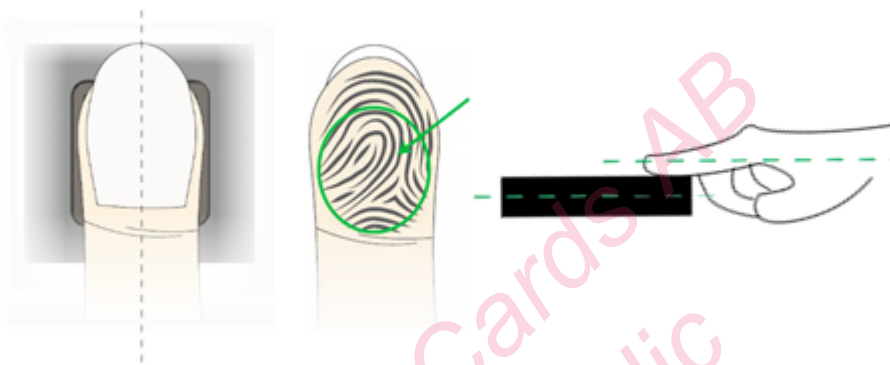


Figure 28. Finger placement recommendations

14.3 Enroll

- Before first usage at least one fingerprint needs to be registered (i.e. enrolled) on the sensor
- This is stored as a template, of key features in the fingerprint, and cannot be reversed to a fingerprint image again
- When enrolling (i.e. registering your fingerprint on the sensor):
- Approach BM-Lite as you would when using it
- Enroll the same finger as you want to put on the sensor when using BM-Lite for verification
- Keep the finger still on the sensor for approximately 1 second and then completely lift it
- This must be repeated until the enrollment is finished and a complete template has been acquired
- For enrolling more fingers, restart the reader and repeat the process with another finger
- For enrolling more fingers, restart the reader and repeat the process with another finger

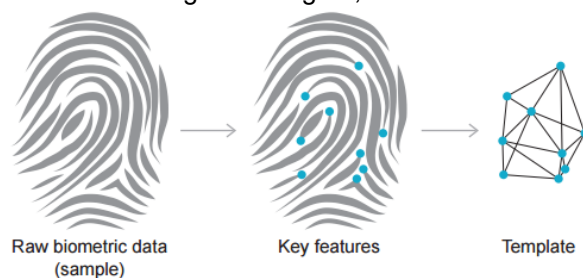


Figure 29. Template generation illustration



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(48/69)

14.4 Verify/Authenticate

- When the sensor is used to verify a user, features on the fingerprint presented are compared to features in the stored template
- If there are enough similar features it is considered a match
- Make sure you use the same finger as when you enrolled, and that the same part of the fingertip is exposed to the sensor.

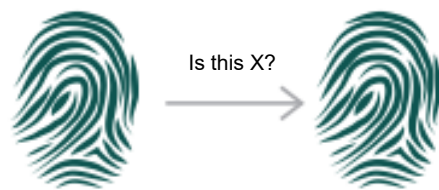


Figure 30. Verification illustration

14.5 Template Update

Template updating adapts the template information.

Template updating enables:

- Increasing the amount of information (template growing)
 - Little coverage during initial enroll
 - Successful verify (unlock attempts) provide additional coverage (covering additional parts of the finger), or yielding better quality information of already covered parts
 - Template updating captures this information and updates templates accordingly
- Updating already enrolled information:
 - Some of the fingerprint characteristics may change over time (e.g. changes in creases due to seasonal differences in humidity)
 - The user may change behavior (grip) over time
 - In consequence, the most suitable template representation changes over time
 - Successful verifies provide updated information

Template updating captures these changes and updates templates accordingly.

14.6 Performance

- FAR (False Accept Rate): the probability that the system incorrectly matches the input pattern to a non-matching template in the database. It measures the percent of invalid inputs that are incorrectly accepted. In case of similarity scale, if the person is an imposter, but the matching score is higher than the threshold, then he is treated as genuine. This increases the FAR, which thus also depends upon the threshold value. FAR also increases with increased number of enrolled templates. *In summary, FAR is a measurement on security.*
- FRR (False Reject Rate): the probability that the system fails to detect a match between the input pattern and a matching template in the database. It measures the percent of valid inputs that are incorrectly rejected. *In summary, FRR is a measurement on convenience or user friendliness.*



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(49/69)

The errors that are observed in FRR are typically not evenly distributed between the users, most of the users have good biometric performance but few users might have an observed high FRR. The reason for this is that the fingers of the users with high FRR might have features that are hard to capture e.g. from worn fingers, dry skin or damages.

One possible implementation that further improve the FRR for BM-Lite is to during the enrollment to enroll the same finger twice, i.e. allow one finger to get two Id. See example of flow charts in chapter 14.7.2.

14.7 Use Cases

14.7.1 Use Case – 3 touches to enroll

All procedures/functions are initiated by the host. (see section 5.5 for examples on how the interaction between host and BM-Lite shall be executed)

Note: BM-Lite supports 3 enrollment touches.

14.7.1.1 Enrollment

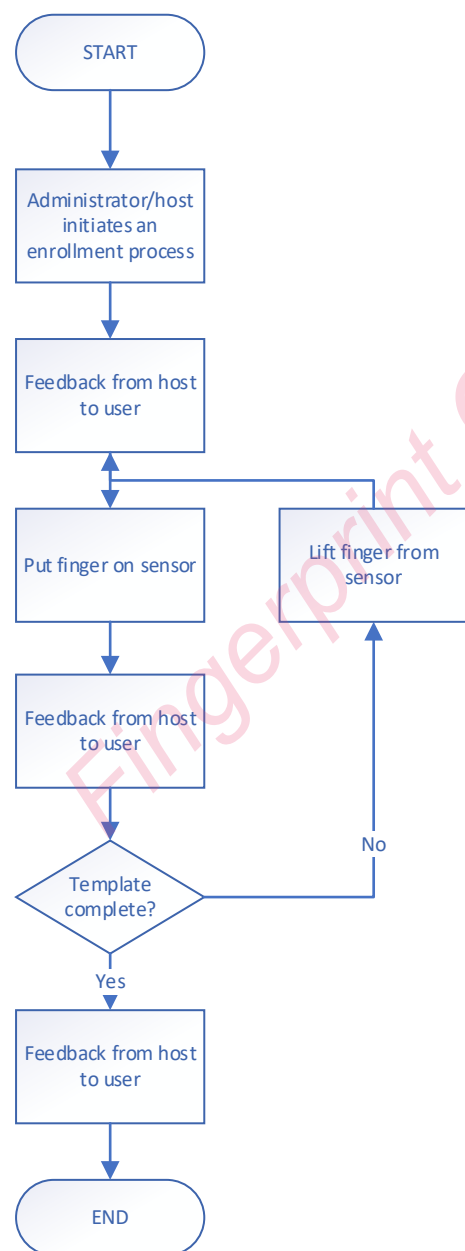


Figure 31. Enrollment flow chart

14.7.1.2 Identification

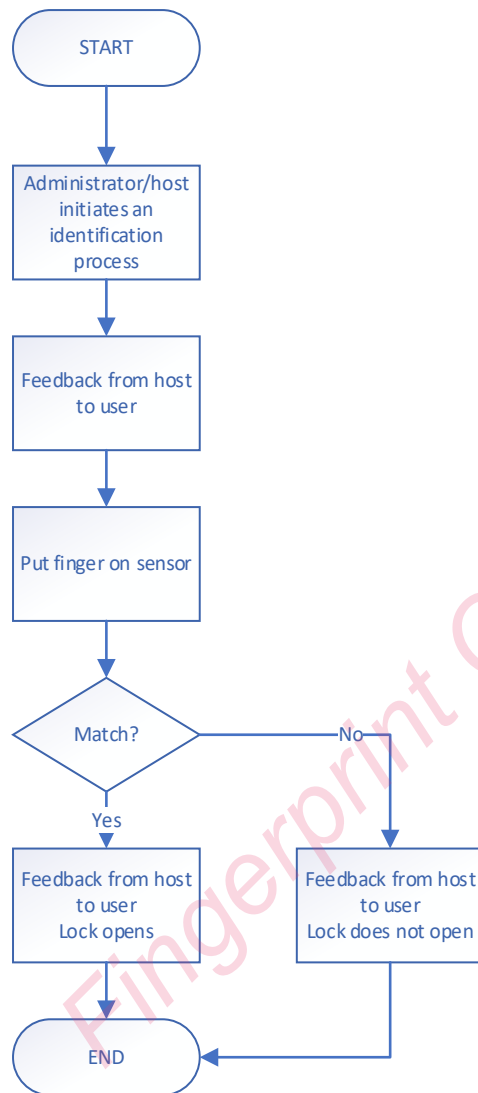


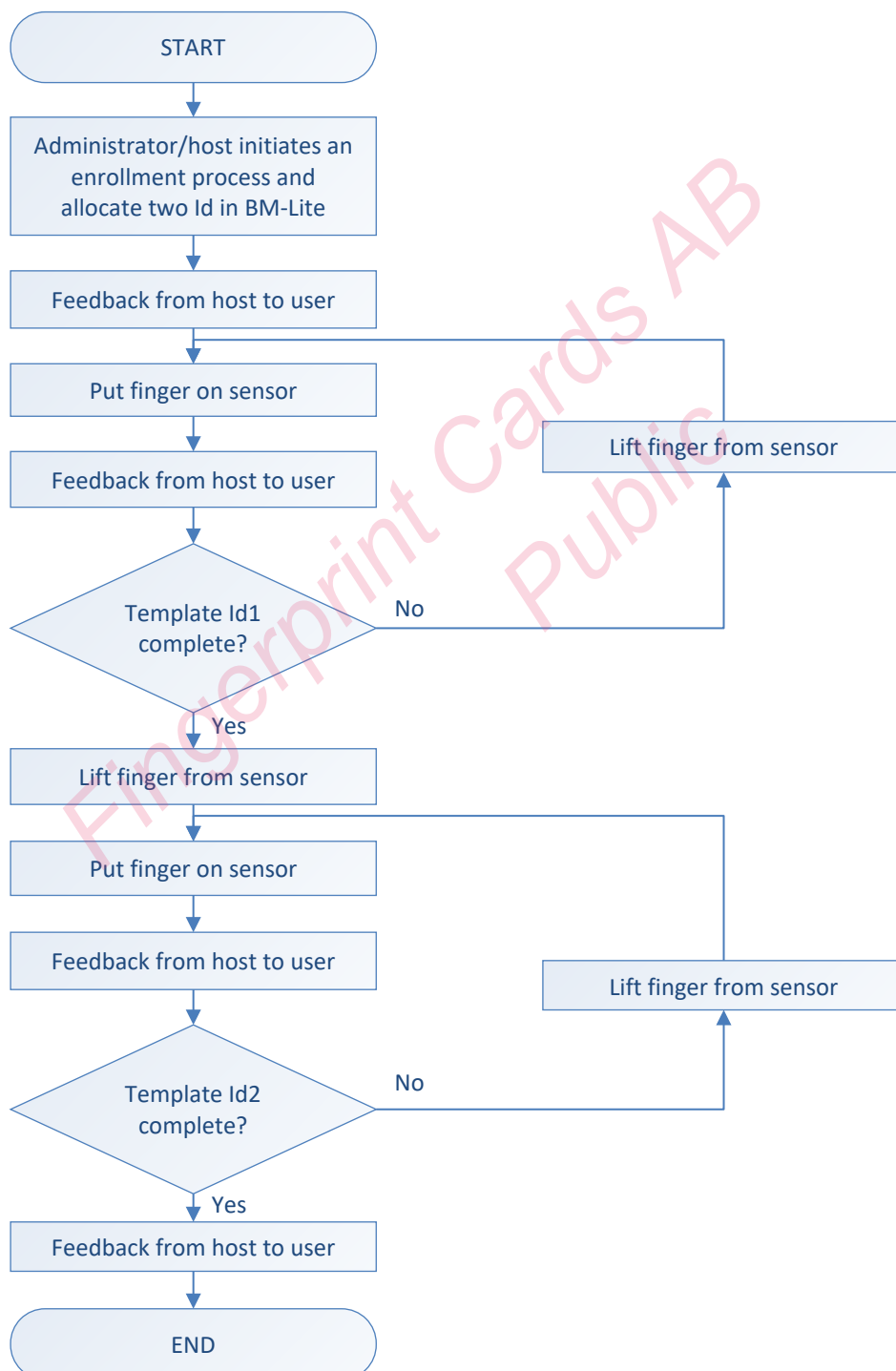
Figure 32. Identification flow chart

14.7.2 Use Case – 6 touches to enroll

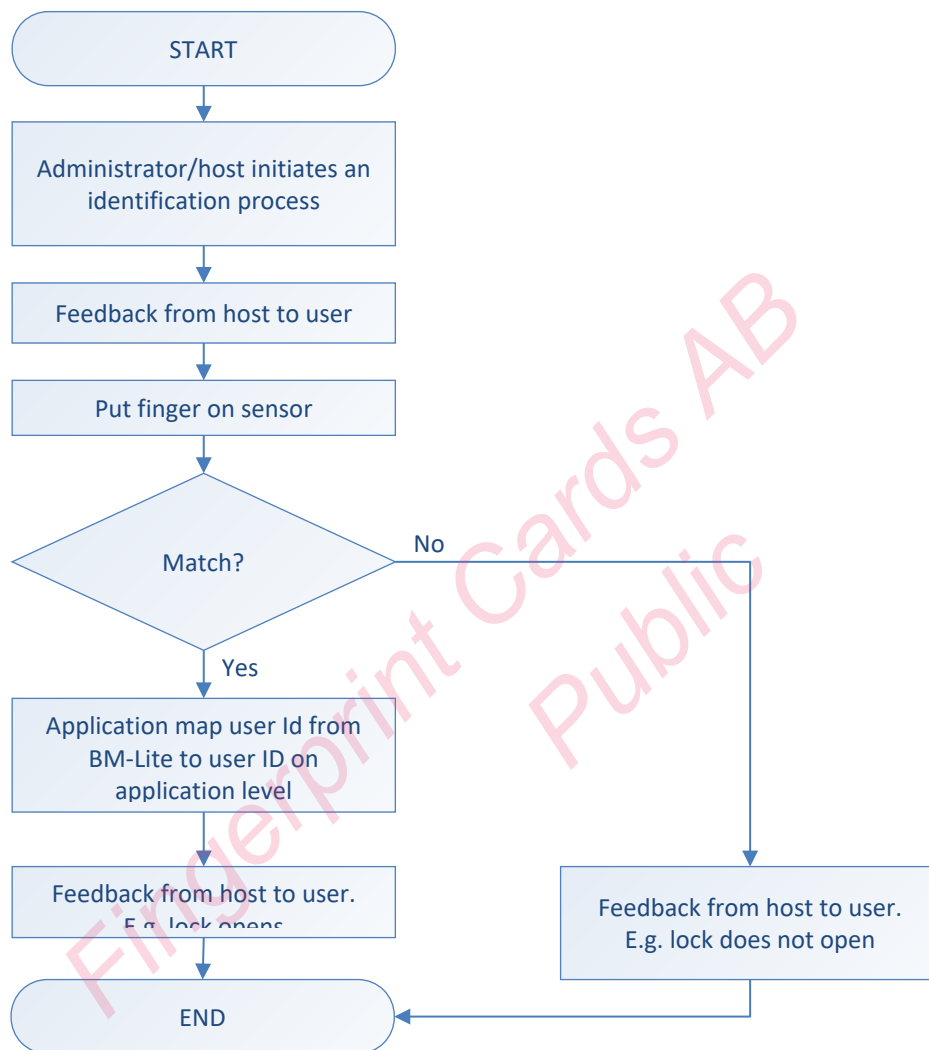
All procedures/functions are initiated by the host. (see section 5.5 for examples on how the interaction between host and BM-Lite shall be executed)

Note: BM-Lite supports 3 enrollment touches.

14.7.2.1 Enrollment



14.7.2.2 Identification



15 Product Identification

To allow for visual recognition and traceability, the BM-Lite is delivered with the following marking on the surface of the BM-Lite PCB (shown in Figure 33):

- Device name
- Module part number
- Company information
- Test point identification
- QR code

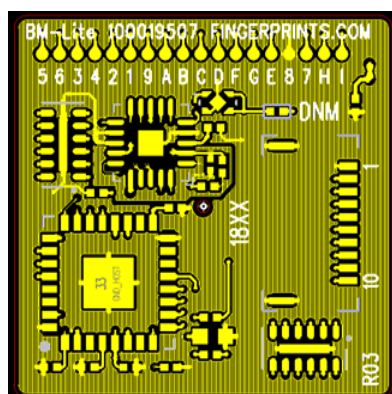


Figure 33. Example of PBC marking

15.1 QR code

BM_Lite QR

Classification	Byte	#1
		SUS316+PVD
Prefix	2	FP
Customer P/N	13	100018754CM01
Revision	4	ES--
Sensor	12	DAE370D13025
Date	3	87D
QR_Serial#	4	0001 ~ ZZZZ
Total	38	

Figure 34. QR code information



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(55/69)

Public

16 Product Updates

An overview of updates for the BM-Lite can be found in this section.

16.1 Product Configurations

An overview of the latest product configurations for the BM-Lite is shown in Table 22.

FPC-BM-Lite (P/N 100026040) replaces FPC-BM-Lite (P/N: 100018754)

Product Number	Commercial Name	Description	Bezel Color	Sensor Color	Color Code
100026040	FPC-BM-Lite	Match-on-Chip module with SUS bezel	Matte black	Matte black	Pantone no 8603C

Table 22: Product Configurations

17 Regulatory Compliance

17.1 EU RoHS2 (2011/65/EU)

Is compliant

17.2 Halogen Free

Is compliant

17.3 ODS

Is compliant

17.4 China RoHS (SJ/T 11363-2006)

Is compliant

17.5 REACH

Is compliant



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(58/69)

18 Known Limitations

18.1 UART (only for CSW1.0)

In some cases, when BM-Lite is connected to a UART to USB converter, and when disconnecting the USB from the computer BM-Lite will stop to respond. This could be solved by adding a pull-up resistor to UART_RX pin or just power cycle BM-Lite.

18.2 UART Baudrate

FW version 1.1 only supports 921600 baud rate, and version 1.2 or later version keeps 115200 as default baud rate, which can be switched to 921600 at run-time.

18.3 Identify at Enroll

Identify at enroll means that when enrolling a finger, a check is automatically made to make sure that the specific finger is not already enrolled.

When enrolling a finger on BM-Lite this check is not automatically done. Instead, host must make sure to make the check when enrolling:

CMD_CAPTURE ⇒ CMD_IDENTIFY ⇒ CMD_ENROLL

However, BM-Lite can handle multiple enrolled versions of the same fingers. But to save memory space it is important not to allow multiple enrolled versions of the same finger.

18.4 Delay After Matching

After finger matching, BM_LITE may trigger TU (template update), so any sudden power cut/reset procedure may result in losing updated template. Hence, it is recommended to add at least 60ms delay to wait for TU to be done.



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(59/69)

Public

19 Revision history

Revision	Date	Changes
1	2023-09-27	First revision of SPC26627 (BM-Lite product specification and integration guideline). For older version see SPC20596-21.
2	2023-10-16	Added chapter "HW setup FPC5924"
3	2023-10-25	Updated with the name FPC SafeTouch

20 Appendix – Python Host evaluation Application

Host evaluation application enables users to easily evaluate biometric functionality such as:

- Image capture
- Enroll procedure
- Identify

The Host evaluation application is a python application (bep_ref.py) with corresponding UI (see Figure 35).

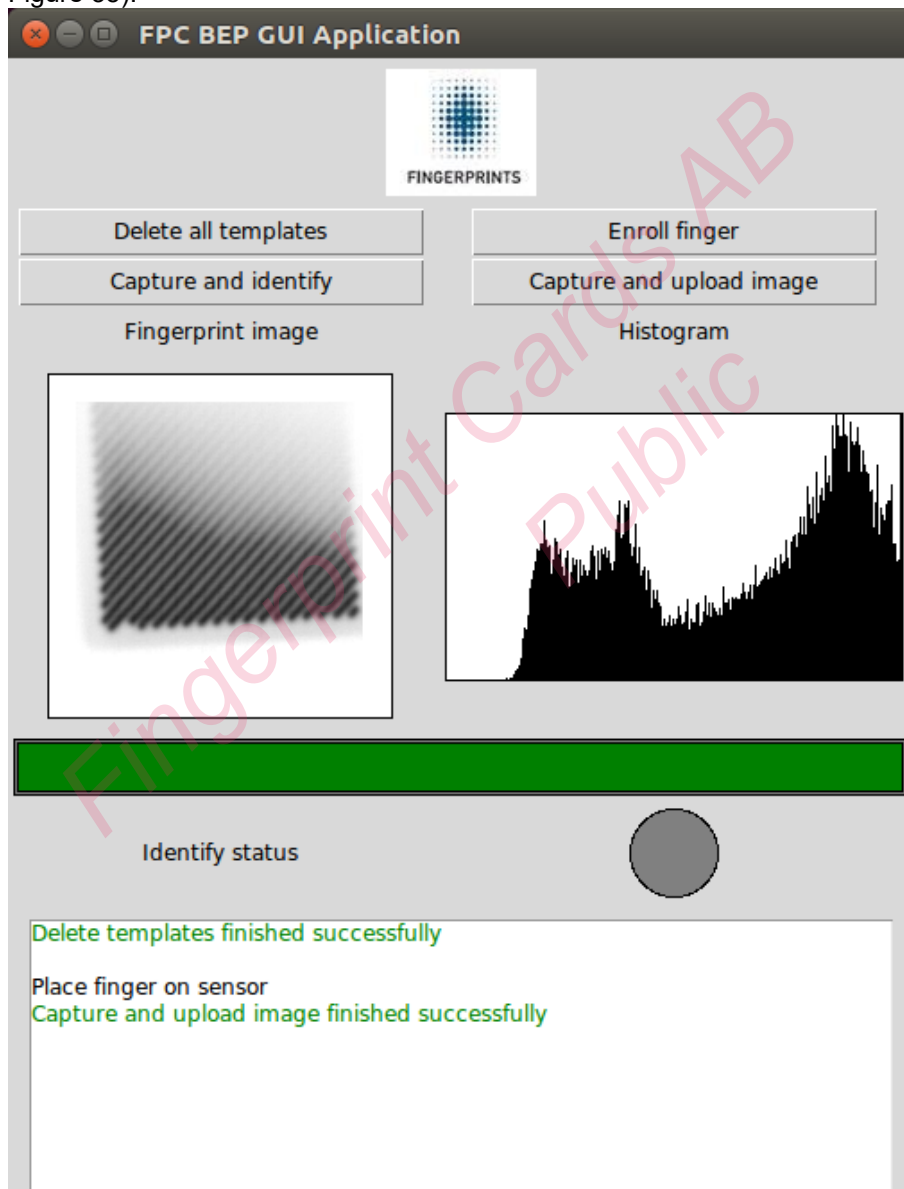


Figure 35. Host evaluation Application

20.1 HW setup

In order to use the Demo application, the FPC5832 SPI to USB device must be connected between BM-Lite and the PC.

When using the SPI interface; it is recommended to inhibit the UART interface by adding a pull-up (<300k Ω) to the UART_RX pin. *Note: A pull-up will possibly increase the power consumption.*

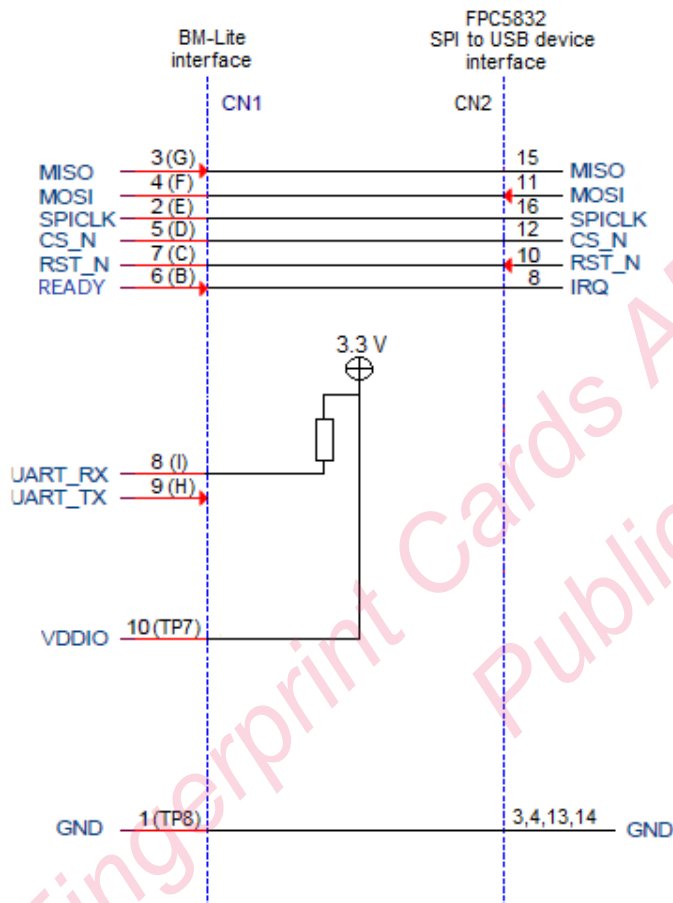


Figure 36. FPC5832 connection details

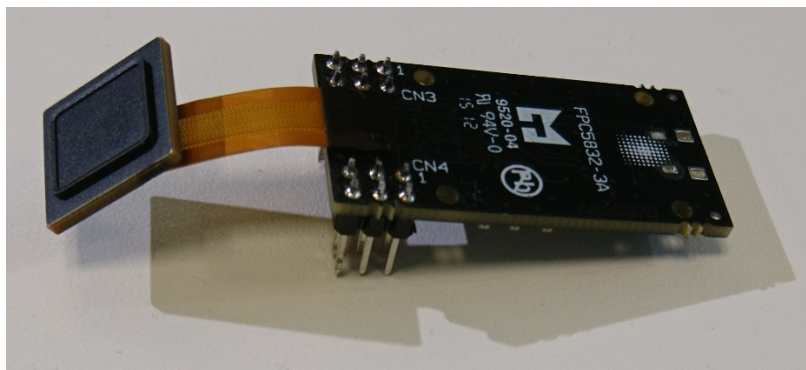


Figure 37. BM-Lite with FPC5832

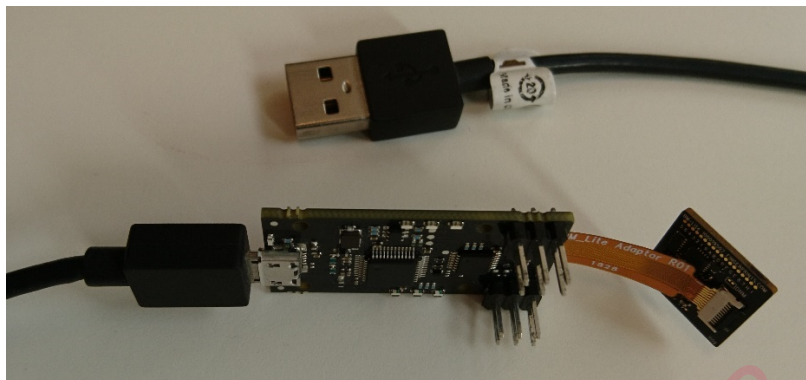


Figure 38. BM-Lite with FPC5832 and USB-cable

20.2 SW setup

Python application works on Linux and Windows platform and has dependencies to the following packages:

- pexpect
- pyserial
- numpy
- pillow
- pyftdi
- spidev

20.2.1 Linux

1. Install Python3.
2. Install pip3.
3. Install the required packages.
4. Run bep_ref.py with the appropriate arguments.

20.2.2 Windows

1. Install Python3.
2. Install pip3.
3. Install the required packages.
4. If using SPI: Install SPI-filter*
5. Run bep_ref.py with the appropriate arguments.

- * In addition to packages described in 20.2, a filter driver for SPI is needed. Download and install libusb-win32-devel-filter-1.2.6.0.exe.

<https://sourceforge.net/projects/libusb-win32/files/libusb-win32-releases/1.2.6.0/>
The filter should be installed on the FTDI driver and the FPC5832 board should be installed as "USB Serial Port"

20.3 Arguments for bep_ref.py

Arguments	Description
-i, --interface	Host interface for communication with BM-Lite. Default is 'uart'. Other alternatives: 'spi' for the FPC5832 FTDI-card, 'rpispi' for the Raspberry Pi SPI



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(64/69)

-p, --port	Input port name (COMX, /dev/ttyACMX, /dev/ttyUSBX)
-s, --speed	Input communication speed
-t, --timeout	Timeout for 'UART receive' in seconds (>=2)
--id	Partial or full Jlink device id
--debug	Print debug information
--log	Collect KPI data and target logs. Port number must be entered "--log=[PORT]"
--gui	Start application with graphical user interface

20.4 Startup procedure

20.4.1 SPI Startup example

(Replace the comport name with the one you are using)

Linux: `sudo python3 ./bep_ref.py -i spi -p /dev/ttyUSB0`

Win: `py ./bep_ref.py -i spi -p COM5`

20.4.2 UART Startup example

Note 1: Using BM-Lite with FW version 1.2 and later must start at baud rate 115200. The baudrate is configurable after start, see Command Description in Chapter 7.

Using BM-Lite with older FW version 1.1 must use 921600. (see Chapter 18.2 for more information)

Note 2: UART is not possible to use, if using the flexcable provided in Fingerprints' BM-Lite Development Kit. To use UART you have to make sure to have a physical UART connection between BM-Lite and host.

(Replace the comport name with the one you are using)

Linux: `sudo python3 ./bep_ref.py -i uart -p /dev/ttyUSB0 -s 115200`

Win: `py ./bep_ref.py -i uart -p COM5 -s 115200`

20.5 Example procedures

After starting the bep_ref.py application (see section 20.4), "Possible options" will be presented for controlling the BM-Lite.

Note: Please follow the procedures in numerical order

20.5.1 Sanity check

1. In the main options menu of bep_ref.py, send command 'g' for 'System sub-menu' and then option 'a' for 'Get software version'.
2. If the communication is working properly, Fingerprints BM-Lite version number will be received.

20.5.2 Performing Enroll

1. In the main options menu of bep_ref.py, send command 'a' for "Enroll finger".
2. You will receive the messages "Enroll Started" and "Place finger on the sensor".
3. Place your finger on the sensor. There is a timeout on this action which will trigger unless there is a finger placed on the sensor after a given time period. If this happens, just try again.
4. When you've placed your finger on the sensor, you will receive the messages "Finger captured", "Image enrolled, X enrollments left" and "Waiting for finger not present".
5. Lift your finger from the sensor.



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(65/69)

6. Repeat step 3-5 until enough images have been saved for a successful enrollment. This will be communicated with the message "Enroll finished. Finger enrolled".
7. Press enter to return to the main options menu.
8. To store your fingerprint template in the flash memory, send command 'd' for "Save template to flash and remove from RAM".
9. Enter a number which will be the enrolled fingerprint templates ID-number.
10. You will receive the message "Template saved with id 4. Template removed from RAM." Press enter to return to the main options menu. Your finger is now enrolled.

20.5.3 Performing Identify on an enrolled finger

1. In the main options menu of bep_ref.py, send command 'b' for "Capture and identify by template".
2. You will receive the message "Place finger on the sensor".
3. Place your finger on the sensor. There is a timeout on this action which will trigger unless there is a finger placed on the sensor after a given time period. If this happens, just try again.
4. When you've placed your finger on the sensor, you will receive the messages "Finger captured", "Image extracted" and "Identify: match against id X" where X is the ID-number of your fingerprint template. If your finger does not match an enrolled template, you will instead receive the message "Identify: no match".
5. If no fingerprint templates are enrolled and you attempt to perform Identify, you will receive the error message "Failure. Result code -7, "Resources are not available". If that happens, please repeat the enrollment procedure to make sure you have successfully enrolled a fingerprint.
6. Congratulations, your product is now safeguarded by the most secure fingerprint sensor on the market.

21 Appendix – C-code reference application

C-code reference application enables users to easily evaluate biometric functionality such as:

- Image capture
- Enroll procedure
- Identify

21.1 HW setup FPC5832

In order to use the C-code reference application, the FPC5832 SPI to USB device must be connected between BM-Lite and the PC.

When using the SPI interface; it is recommended to inhibit the UART interface by adding a pull-up (<300kΩ) to the UART_RX pin. *Note: A pull-up will possibly increase the power consumption.*

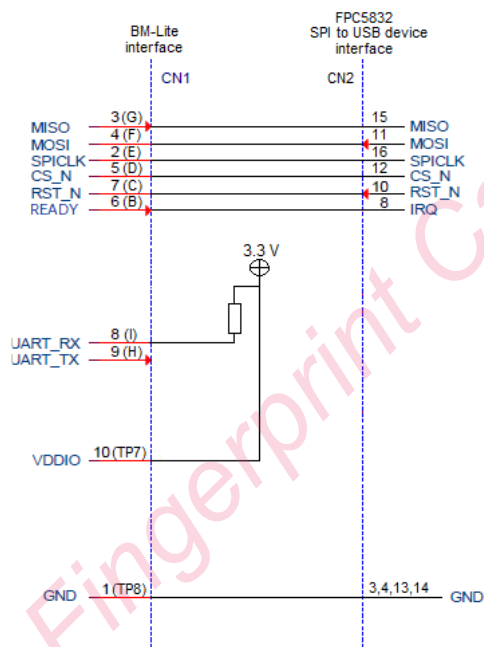


Figure 39. FPC5832 connection details

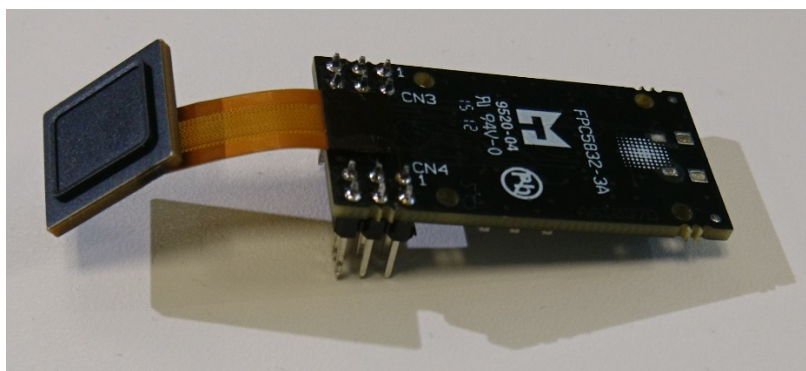


Figure 40. BM-Lite with FPC5832

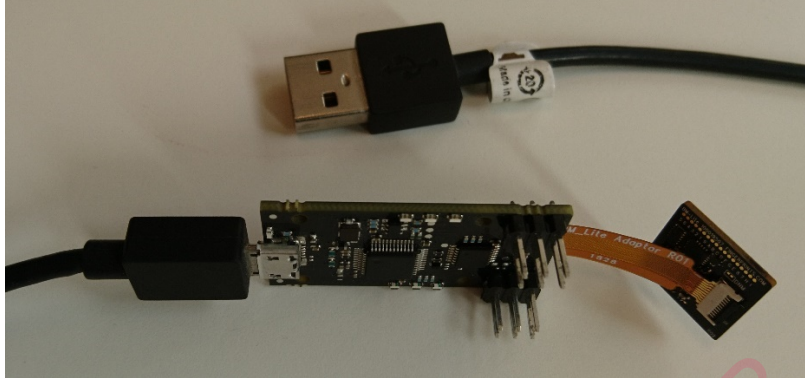
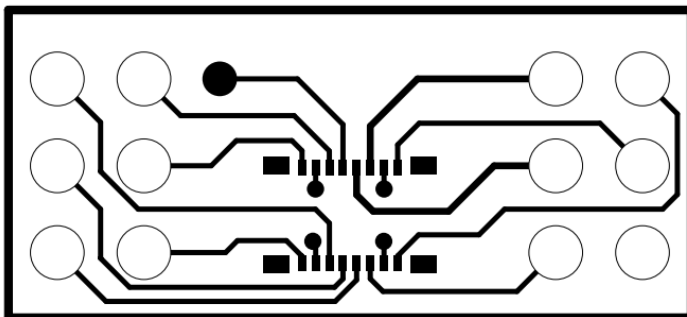
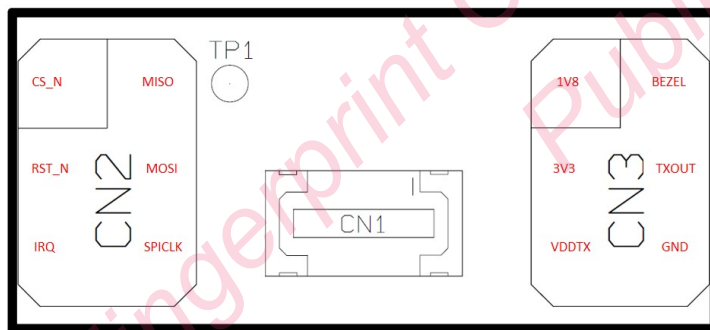


Figure 41. BM-Lite with FPC5832 and USB-cable

21.2 HW setup FPC5924

In case using the FPC5924 connector, the following is the pin assignment between the Hirose B2B connector and the FPC5924 pinout.



21.3 SW setup

The FPC5832 uses standard FTDI drivers and should be automatically installed on both Linux and Windows when plugging it in.



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(68/69)

21.3.1 Linux

1. "sudo rmmod ftdi_sio" and "sudo rmmod usbserial" needs to be run before starting the application.
2. Run `bep_host_com` with the appropriate arguments.

21.3.2 Windows

1. Run `bep_host_com.exe` with the appropriate arguments.

21.4 Arguments for `bep_host_com[.exe]`

Arguments	Description
-s	Input communication speed
-t	Timeout for 'SPI receive' in seconds (≥ 2)

21.5 Startup procedure

21.5.1 SPI Startup example

Linux: `bep_host_com -s 1000000`Win: `bep_host_com.exe -s 1000000`

21.6 Example procedures

After starting the `bep_host_com` application, "Possible options" will be presented for controlling the BM-Lite.

Note: Please follow the procedures in numerical order

21.6.1 Sanity check

1. In the main options menu, choose option 'h' for 'Get software version'.
2. If the communication is working properly, FPC BM-Lite version number will be received.

21.6.2 Performing Enroll

1. In the main options menu, choose option 'a' for "Enroll finger".
2. You will receive the messages "Enroll start".
3. Place your finger on the sensor. There is a timeout on this action which will trigger unless there is a finger placed on the sensor after a given time period. If this happens, just try again.
4. Lift your finger from the sensor.
5. Repeat step around 3 times until enough images have been saved for a successful enrollment. This will be communicated with the message "Command succeeded, press enter to continue".
6. Choose option 'd' to save template
7. Enter a number which will be the enrolled fingerprint template ID-number.
8. Press enter to return to the main options menu

21.6.3 Performing Identify on an enrolled finger

1. In the main options menu, choose option 'b' for "Capture and identify finger".
2. You will receive the message "Put finger on sensor".
3. Place your finger on the sensor.
4. When you've placed your finger on the sensor, you will receive the message "Match with template id: X" where X is the ID-number of your fingerprint template. If your finger does not match an enrolled template, you will instead receive the message "No match".
5. If no fingerprint templates are enrolled and you attempt to perform Identify, you will receive



Doc number/doc name:

SPC26627-3

Document Type:

Specification

Author

Henrik Nilsson

State:

Approved

Approver:

Henrik Nilsson

Date:

2023-10-25

Page:

(69/69)

Public

the error message "Command failed with error code No Resource (-7) ". If that happens, please repeat the enrollment procedure to make sure you have successfully enrolled a fingerprint.

6. Congratulations, your product is now safeguarded by the most secure fingerprint sensor on the market.

Fingerprint Cards AB
Public