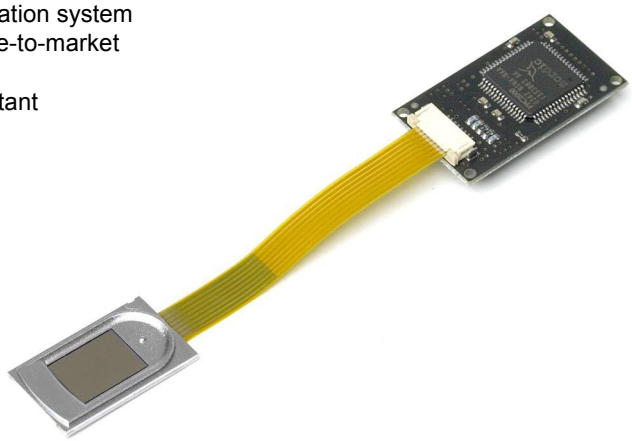
FPC-AM Area Sensor Module

Product Specification

Features

- Embedded stand-alone fingerprint identification system
- Extremely easy to integrate minimizing time-to-market
- Small size
- Thick protective sensor coating, ESD resistant
- One-To-One verification mode
- One-To-Many identification mode
- Onboard template storage (188 templates)
- Straightforward serial command interface
- Download/upload template functionality



Application examples

- Access control systems
- Time & Attendance
- Locks, safes

General description

The FPC-AM module acts as a biometric sub-system with onboard template storage.

Integrating the FPC-AM module into a product drastically reduces time-to-market with its easy-to-integrate serial command interface and proven robust fingerprint sensor solution.

FPC-AM features the rugged fingerprint sensor, FPC1011C with its unique thick protective coating, which prevents the user from directly touching the CMOS circuitry. The coating also protects the sensor against ESD well above 15 kV and everyday wear-and-tear. The FPC1011C has an acknowledged very high mage quality with a bit depth of 256 values per pixel.

FPC-AM can easily be integrated into virtually any application and be controlled by a host sending basic commands for enrolment and verification via the serial interface. FPC-AM comes preloaded with software and is ready to use at delivery. Fingerprint templates are automatically created and stored in the internal flash memory. Templates used for verification can also be imported from an external storage, e.g. central database, smart card or portable flash memory.

FPC-AM can be connected to a host via a board-to-board connector or by using a standard ribbon cable.

Quick reference data

PARAMETER		VALUE
Dimension	(L x W x T)	40 x 23 x 6 mm (Processor board)
Number of templates		188
Verification time	(1:1)	0.5 s (typical)
Identification time	(1:100)	1 s (typical)
Enrolment time		5 s (typical)
False-Rejection-Rate	(FRR)	Adjustable
False-Acceptance-Rate	(FAR)	Adjustable
Interface		Serial UART
Supply voltage		3.3 VDC
Supply current		Active: 100 mA, Power save mode: 10 mA, Sleep mode: 30 µA
Active sensing area		10.64 x 14.00 mm
Pixel resolution		256 gray scale values (8 bit)
ESD protection		> 15 kV
Wear-and-tear		> 1 million wear cycles

This page is intentionally left blank.

Table of contents

Functional description

System overview

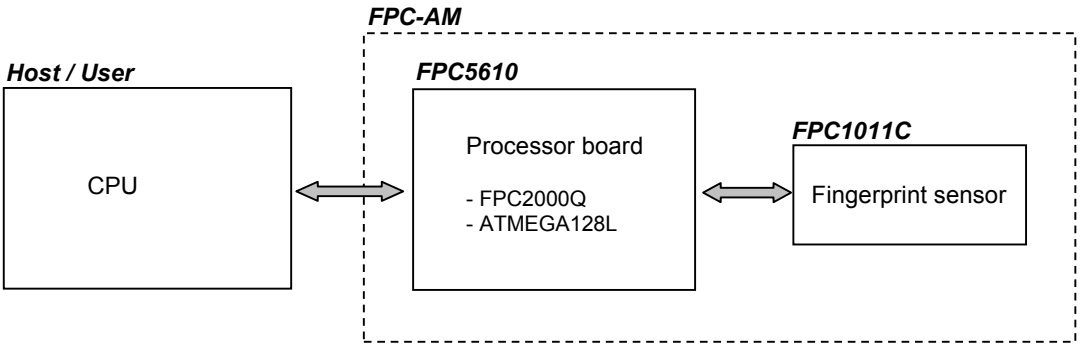


Figure 1
System overview

General description
The FPC-AM area sensor module is a versatile stand-alone fingerprint verification system.

The FPC-AM consists of two parts; the area sensor FPC1011C and the processor board FPC5610.

The host CPU, selected and provided by the customer, is executing the main application, interfacing the FPC-AM module. Requirements on the host processor associated with the module communication are extremely low. Hence the host processor can be selected entirely to suit the main application at hand.

All CPU capabilities on the FPC5610 processor board are used to perform biometric related functions. Software (firmware) is pre-loaded by Fingerprint Cards and there is no space available for customer applications.

The FPC5610 processor board acquires the fingerprint image from the fingerprint sensor. Thus there is no direct interaction between the host processor and the fingerprint sensor.

The interface between the host processor and the FPC5610 processor board is based on a simple-to-use serial UART command interface.

Algorithm performance
The pre-loaded firmware is based on the high performing DAD algorithm developed by Fingerprint Cards. Details are available in:

- REP2009 Performance white paper

FPC1011C Area sensor
The FPC1011C is a small and easy-to-integrate area sensor. The sensor chip is based on capacitive technology and utilizes a reflective measurement method. This method requires a galvanic contact point outside the sensor chip, seen as a conductive frame in the sensor package.

Details on the FPC1011C sensor and the flex film package can be found:

- FPC1011C Product specification

In order to obtain good quality images it is important that the sensor is correctly mounted in the enclosure. A number of general recommendations and useful hints can be found in the ergonomic guideline for area sensor based systems.

- TNT2005 Area sensor guidelines

Details on appropriate enclosure thickness and other important measures are given in the *Mechanical integration* section of this document.

FPC5610 Processor board
The FPC5610 make use of the FPC2000Q fingerprint ASIC for all biometric operations and a basic ATMEGA128L for external communication.

Performance characteristics

			Verification	*Identification*	*Identification*	
SYMBOL	**PARAMETER**	**CONDITION**	**1 USER**	**10 USERS**	**100 USERS**	**UNIT**
FRR	False-rejection-rate	High convenience	< 1	< 1	< 4	%
		Default	< 1	< 1	< 5	%
		High security	< 2	< 2	< 7	%
FAR	False-acceptance-rate	High convenience	< 0.2	< 1	1	%
		Default	< 0.04	< 0.2	0.2	%
		High security	< 0.004	< 0.02	0.02	%

Table 1
Algorithm performance characteristics

Electrical characteristics

Normal operation

Operating temperature: -20°C to +60°C

SYMBOL	PARAMETER	CONDITION	MIN	TYP	MAX	UNIT
V_{DD}	Supply voltage		3.0	3.3	3.45	V
I_{DD}	Supply current, total	Active high		100		mA
		Active low		20		mA
		Power save mode		10		mA
		Sleep mode		30		µA
Digital inputs						
V_{IL}	Logic '0' voltage				$0.2V_{DD}$	V
V_{IH}	Logic '1' voltage		$0.8V_{DD}$			V
I_{IL}	Logic '0' current (V_I = GND)				±10	µA
I_{IH}	Logic '1' current (V_I = V_{DD})				±10	µA
Digital outputs						
V_{OL}	Logic '0' output voltage			0.2	0.4	V
V_{OH}	Logic '1' output voltage		$0.85V_{DD}$	$0.90V_{DD}$		V

Table 2
Electrical characteristics

Note:
During normal operation current consumption varies between the ACTIVE HIGH and ACTIVE LOW rating depending on workload (type of command). Available commands and corresponding current consumption is listed later in this specification. If the module is configured for POWER SAME MODE, some components are shut down between each command and thus the current consumption is lowered in the ACTIVE LOW state (image RAM is lost). In SLEEP MODE a system interrupt (INT_N) is required for wake up.

Absolute maximum ratings

Operating temperature	-20°C to +85°C
Storage temperature	-40°C to +85°C
Supply voltage	-0.5 V to +6.0 V
Input voltage	-0.5 V to V_{DD} + 0.5 V
DC current any I/O pin	40 mA
DC current V_{DD} and GND pins	200 mA

Note:
Stress beyond values listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated as normal operation this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 3
Absolute maximum ratings

Software command interface

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

Serial interface settings
The software settings for the serial protocol is:

- Communication speed: factory default baud rate set to 9600 baud (range from 9600 to 115200 baud)
- Format: 8 data bits, odd parity, one stop bit.
- Bit order: least significant bit first

Command send structure
Command, sent from host:

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

STX: Start byte, should be 0x02
IDX-LSB: Index value, least significant byte
IDX-MSB: Index value, most significant byte
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

If PAYLOAD != 0, then additional data should follow in the stream according to the following:

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

Note:
1) The CRC size (4 bytes) is not included in the payload counter.

2) Do not send a new command before a response has been received from the previous command. One exception is when using the CANCEL-command, see “Command description / API_CANCEL”.

Response structure
Response from device:

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

STX: Start byte, should be 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 the following:

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

Note:
The CRC size (4 bytes) is not included in the payload counter.

Table of commands

BIOMETRIC COMMANDS	HEX	MODE	DESCRIPTION
API_CAPTURE_IMAGE	0x80	ACTIVE HIGH	Capture image from sensor (before enrol), returns “finger present” TRUE or FALSE
API_CAPTURE_AND_ENROL_RAM	0x81	ACTIVE HIGH	Enrol into RAM (includes Capture Image)
API_CAPTURE_AND_VERIFY_RAM	0x82	ACTIVE HIGH	Verify against RAM (includes Capture Image)
API_CAPTURE_AND_VERIFY_FLASH	0x83	ACTIVE HIGH	Verify against single FLASH slot (includes Capture Image) Set slot number (0 to 187) in IDX
API_CAPTURE_AND_IDENTIFY_FLASH	0x84	ACTIVE HIGH	Identify against all FLASH slots (includes Capture Image)
API_ENROL_RAM	0x85	ACTIVE HIGH	Enrol into RAM
API_VERIFY_RAM	0x86	ACTIVE HIGH	Verify against RAM
API_VERIFY_FLASH	0x87	ACTIVE HIGH	Verify against single FLASH slot Set slot number (0 to 187) in IDX
API_IDENTIFY_FLASH	0x88	ACTIVE HIGH	Identify against all FLASH slots

IMAGE TRANSFER	HEX	MODE	DESCRIPTION
API_UPLOAD_IMAGE	0x90	ACTIVE HIGH	Upload image from RAM
API_DOWNLOAD_IMAGE	0x91	ACTIVE HIGH	Download image to RAM

TEMPLATE HANDLING	HEX	MODE	DESCRIPTION
API_UPLOAD_TEMPLATE	0xA0	ACTIVE LOW	Upload template from RAM
API_DOWNLOAD_TEMPLATE	0xA1	ACTIVE LOW	Download template to RAM
API_COPY_TEMPLATE_RAM_TO_FLASH	0xA2	ACTIVE LOW	Copy template from RAM to permanent FLASH storage Set slot number (0 to 187) in IDX
API_UPLOAD_TEMPLATE_FROM_FLASH	0xA3	ACTIVE LOW	Upload template from single FLASH slot Set slot number (0 to 187) in IDX
API_DELETE_TEMPLATE_RAM	0xA4	ACTIVE LOW	Erase template from RAM
API_DELETE_SLOT_IN_FLASH	0xA5	ACTIVE LOW	Delete single slot in FLASH Set slot number (0 to 187) in IDX
API_DELETE_ALL_IN_FLASH	0xA6	ACTIVE LOW	Delete all FLASH slots

ALGORITHM SETTINGS	HEX	MODE	DESCRIPTION
API_SECURITY_LEVEL_RAM	0xB0	ACTIVE LOW	Set security level, setting saved in RAM IDX-LSB: 0x04 = high convenience 0x05 = default 0x06 = high security
API_SECURITY_LEVEL_STATIC	0xB1	ACTIVE LOW	Set security level, setting saved in non-volatile (static) memory
API_GET_SECURITY_LEVEL	0xB2	ACTIVE LOW	Get current security level, value sent as payload data

Table 4
Table of commands

FIRMWARE COMMANDS	HEX	MODE	DESCRIPTION
API_FIRMWARE_VERSION	0xC0	ACTIVE LOW	Upload the version string for this device
API_FIRMWARE_UPDATE	0xC1	ACTIVE LOW	Start download of new firmware Use MS-DOS application supplied by Fingerprint Cards

COMMUNICATION COMMANDS	HEX	MODE	DESCRIPTION
API_SET_BAUD_RATE_RAM	0xD0	ACTIVE LOW	Set baud rate, setting saved in RAM IDX-LSB: 0x10 = 9600 baud 0x20 = 14400 baud 0x30 = 19200 baud 0x40 = 28800 baud 0x50 = 38400 baud 0x60 = 57600 baud 0x70 = 76800 baud 0x80 = 115200 baud
API_SET_BAUD_RATE_STATIC	0xD1	ACTIVE LOW	Set baud rate, setting saved in non-volatile (static) memory IDX-LSB: 0x10 = 9600 baud 0x20 = 14400 baud 0x30 = 19200 baud 0x40 = 28800 baud 0x50 = 38400 baud 0x60 = 57600 baud 0x70 = 76800 baud 0x80 = 115200 baud
API_TEST_HARDWARE	0xD2	ACTIVE HIGH	Test hardware components

OTHER COMMANDS	HEX	MODE	DESCRIPTION
API_CANCEL	0xE0	ACTIVE LOW	Cancel ongoing command, only valid for: API_CAPTURE_AND_ENROL_RAM, API_CAPTURE_AND_VERIFY_RAM, API_CAPTURE_AND_VERIFY_AGAINST_FLASH, API_CAPTURE_AND_IDENTIFY_AGAINST_FLASH
API_ENTER_SLEEP_MODE	0xE1	ACTIVE LOW	Enter sleep mode (wake up by activating INT_N pin)
API_POWER_SAVE_MODE_RAM	0xE2	ACTIVE LOW	Set power save mode, setting saved in RAM IDX-LSB: 0x00 = enable 0x01 = disable
API_POWER_SAVE_MODE_STATIC	0xE3	ACTIVE LOW	Set power save mode, setting saved in non-volatile (static) memory IDX-LSB: 0x00 = enable 0x01 = disable
API_GET_POWER_SAVE_MODE	0xE5	ACTIVE LOW	Get current power save mode, value sent as payload data

Table 5
Table of commands (continued)

Note:
The MODE column indicates corresponding current rating as defined in *Electrical characteristics.*

Table of response bytes

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_FW_CRC_FAIL	0x14

Table 6
Table of response bytes

Command description

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.

Response command:
- API_NO_FINGER_PRESENT = No finger present on sensor
- API_FINGER_PRESENT = Finger present on sensor

Note:
If power save mode is enabled the image will NOT be saved in RAM. This means that power save mode must be disabled to use this command.

Capture and enrol (RAM) **API_CAPTURE_AND_ENROL_RAM**
An image is captured from the fingerprint sensor and enrolment of this image is done. The command waits for “finger present” before it starts the enrolment. This means that images are captured in a loop from the sensor until a finger is present. 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

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

Capture and verify (RAM) **API_CAPTURE_AND_VERIFY_RAM**
A template must be present in RAM before starting the verification, either by using the Download Template command (API_DOWNLOAD_TEMPLATE) OR the command 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.

Response command:
- API_VERIFICATION_OK = Verification successful
- API_VERIFICATION_FAIL = Verification failed
- API_NO_TEMPLATE_PRESENT = No template present

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

Capture and verify (FLASH) **API_CAPTURE_AND_VERIFY_FLASH**
The FLASH slot number must be given in the IDX bytes. This command first 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.

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

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

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.

Response command:
- API_ IDENTIFY _OK = Identification successful (slot index is also sent as payload).
- API_ IDENTIFY _FAIL = Identification failed

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

Enrol (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 command API_CAPTURE_IMAGE OR by using the Download Image Command (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.

Response command:
- API_ENROL_OK = Enrolment successful
- API_ENROL_FAIL = Enrolment failed

Verify (RAM) **API_VERIFY_RAM**
A template AND a fingerprint image must be present in RAM before starting the verification. To handle the template use the command Download Template API_DOWNLOAD_TEMPLATE) OR the command API_CAPTURE_ENROL_RAM. To handle the image use the command Download Image API_DOWNLOAD_IMAGE) OR the command API_CAPTURE_IMAGE. Thereafter the verification can be started. The command returns with response when the verification is complete or if the verification fails for any reason.

Response command:
- API_VERIFICATION_OK = Verification successful
- API_VERIFICATION_FAIL = Verification failed
- API_NO_TEMPLATE_PRESENT = No template present

Verify (FLASH) **API_VERIFY_FLASH**
A fingerprint image must be present in RAM before starting the verification, use the command Download Image API_DOWNLOAD_IMAGE) OR the command 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.

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

Identify (FLASH) **API_IDENTIFY_FLASH**
A fingerprint image must be present in RAM before starting the verification, use the command Download Image API_DOWNLOAD_IMAGE) OR the command API_CAPTURE_IMAGE. Identification is made against all FLASH slots. The command returns with response when the identification is complete or if the identification fails for any reason.

Response command:
- API_ IDENTIFY _OK = Identification successful (slot index is also sent as payload).
- API_ IDENTIFY _FAIL = Identification failed
- API_NO_TEMPLATE_PRESENT = All FLASH slots are empty

Upload image **API_UPLOAD_IMAGE**
By using this command it is possible to upload the fingerprint image present in RAM The response is the API_SUCCESS command followed by the image data. The size of image data is 30400 bytes. The first byte is the upper left pixel and then data follows row-wise (X-direction). Each pixel has one byte value (256 gray scales). There is no image header.

Response command:
- API_ SUCCESS = Upload successful
- API_ FAILURE = Upload failed

Download image **API_DOWNLOAD_IMAGE**
By using this command it is possible to download a fingerprint image to RAM. The size of image data must be 30400 bytes. The first byte is the upper left pixel and then data follows row-wise (X-direction). Each pixel has one byte value (256 grey scales). There is no image header.

Response command:
- API_ SUCCESS = Download successful
- API_ FAILURE = Download failed

Note:
If power save mode is enabled the image will NOT be saved in RAM. This means that power save mode must be disabled to use this command.

Upload template **API_UPLOAD_TEMPLATE**
After a successful enrolment the template is uploaded from RAM using the Upload Template command (API_UPLOAD_TEMPLATE). The response is the API_SUCCESS command followed by the template data. The template consists of 451 bytes binary data with no public information.

Response command:
- API_ SUCCESS = Upload successful
- API_ FAILURE = Upload failed

Download template **API_DOWNLOAD_TEMPLATE**
Before verification the template is downloaded to RAM using the Download Template command (API_DOWNLOAD_TEMPLATE). The size of the template is 451 bytes.

Response command:
- API_ SUCCESS = Download successful
- API_ FAILURE = Download failed

Copy template from RAM to FLASH **API_COPY_TEMPLATE_FROM_RAM_TO_FLASH**
This command copies the template currently in RAM to FLASH. To download and store a template into FLASH first use the Download Template command, then copy it from RAM to FLASH. The FLASH slot number must be given in the IDX bytes.

Response command:
- API_ SUCCESS = Template storage successful
- API_ FAILURE = Template storage failed
- API_INVALID_SLOT_NR = Wrong slot number

Upload template from FLASH **API_UPLOAD_TEMPLATE_FROM_FLASH**
This command uploads the template from FLASH. The FLASH slot number must be given in the IDX bytes. The size of the template is 451 bytes.

Response command:
- API_ SUCCESS = Upload successful
- API_ FAILURE = Upload failed
- API_INVALID_SLOT_NR = Wrong slot number

Delete template in RAM **API_DELETE_TEMPLATE_RAM**
This command deletes the template currently stored in RAM.

Response command:
- API_ SUCCESS = Template removal successful
- API_ FAILURE = Template removal failed

Delete single template in FLASH **API_DELETE_SLOT_IN_FLASH**
By using the command Delete slot in FLASH one can choose which slot to delete (include slot number in index value of command).

Response command:
- API_ SUCCESS = Template removal successful
- API_ FAILURE = Template removal failed
- API_INVALID_SLOT_NR = Wrong slot number

Delete all templates in FLASH **API_DELETE_ALL_IN_FLASH**
It is possible to delete all templates in FLASH by issuing the command Delete all in FLASH.

Response command:
- API_ SUCCESS = Template removal successful
- API_ FAILURE = Template removal failed

Security level (RAM) **API_SECURITY_LEVEL_RAM**
The security level to be used during verification and identification can be set by the command Set security level. The value of the security level should be set in the index value (IDX-LSB) of the command. The factory default security level is set to value 0x05. The value is stored in RAM and the setting is lost after reset. The security level is not stored together with the template. During enrolment there is no effect when changing the security threshold. The created template will support all security settings.

VALUE (IDX-LSB)	SECURITY LEVEL
0x04	High convenience
0x05	Default
0x06	High security

Response command:
- API_ SUCCESS = New security level set
- API_ FAILURE = Security level out of range

Security level (STATIC) **API_SECURITY_LEVEL_STATIC**
The security level to be used during verification and identification can be set by the command Set security level. The value of the security level should be set in the index value (IDX-LSB) of the command. The factory default security level is set to value 0x05. The value is stored in non-volatile memory and the setting is saved even after reset. This means that the factory default value will be changed. The security level is not stored together with the template. During enrolment there is no effect when changing the security threshold. The created template will support all security settings.

VALUE (IDX-LSB)	SECURITY LEVEL
0x04	High convenience
0x05	Default
0x06	High security

Response command:
- API_ SUCCESS = New security level set
- API_ FAILURE = Security level out of range

Get current security level **API_GET_SECURITY_LEVEL**
This command returns the value of the current security setting that the module uses. The value is sent as payload data.

VALUE	SECURITY LEVEL
0x04	High convenience
0x05	Default
0x06	High security

Response command:
- API_ SUCCESS = Command OK
- API_ FAILURE = Command fail

Firmware version **API_FIRMWARE_VERSION**
This command returns the firmware version of the main application. The response is the API_SUCCESS command followed by the firmware version string.

Response command:
- API_ SUCCESS = Request successful, version string follows as payload
- API_ FAILURE = Request failed

Firmware update **API_FIRMWARE_UPDATE**
It is possible to update the pre-loaded firmware. Fingerprint Cards supplies a PC application that performs firmware updates.

DOS application: UPDATE_FPC7610.EXE.

Application is executed from DOS according to the following:

UPDATE_FPC7610.EXE *firmware_name –COM<n>* *(<n> = COM port number)*

A CRC check is performed by FPC-AM at start-up to verify that correct firmware is present.
If the firmware is corrupt all commands will receive the response:

Response command:
- API_ FW_CRC_FAIL = Firmware is corrupt

If the firmware is corrupt it is still possible to set baud rate and perform firmware updates.

Set baud rate (RAM) **API_SET_BAUD_RATE_RAM**
It is possible to change baud rate for the serial communication between host and FPC-AM. The table below shows the available baud rates. Factory default baud rate is 9600. The selected value should be set in the IDX-LSB byte of the command. The value is stored in RAM and the setting is lost after reset.

VALUE (IDX-LSB)	BAUD RATE
0x10	9600 (factory default)
0x20	14400
0x30	19200
0x40	28800
0x50	38400
0x60	57600
0x70	76800
0x80	115200

Response command:
- API_ SUCCESS = Baud rate change accepted
- API_ FAILURE = Baud rate out of range

Note:
Once a baud rate change has been accepted, next command must be sent with new baud rate.
However the response command above is sent with the old baud rate.

Set baud rate (STATIC) **API_SET_BAUD_RATE_STATIC**

It is possible to change baud rate for the serial communication between host and FPC-AM. The table below shows the available baud rates. Factory default baud rate is 9600.
The selected value should be set in the IDX-LSB byte of the command. The value is stored in non-volatile memory and the setting is saved even after reset. This means that the factory default value will be changed.

VALUE (IDX-LSB)	BAUD RATE
0x10	9600 (factory default)
0x20	14400
0x30	19200
0x40	28800
0x50	38400
0x60	57600
0x70	76800
0x80	115200

Response command:

- API_ SUCCESS = Baud rate change accepted
- API_ FAILURE = Baud rate out of range

Note:
Once a baud rate change has been accepted, next command must be sent with new baud rate.
However the response command above is sent with the old baud rate.

Test hardware **API_TEST_HARDWARE**

This command tests the FPC-AM hardware. It performs check of the different components on the module.

Response command:

- API_HW_TEST_OK = Hardware check successful
- API_HW_TEST_FAIL = Hardware check failed, contact technical support

Cancel current command **API_CANCEL**

It is possible to cancel the following ongoing commands:

API_CAPTURE_ENROL_RAM
API_CAPTURE_VERIFY_RAM
API_CAPTURE_VERIFY_AGAINST_FLASH
API_CAPTURE_IDENTIFY_AGAINST_FLASH

The module will respond with API_CANCEL_SUCCESS and the return to normal command loop.

Response command:

- API_ CANCEL_SUCCESS = Cancel successful
- API_ FAILURE = Cancel failed

Enter sleep mode **API_ENTER_SLEEP_MODE**

SLEEP MODE is entered by issuing the command API_ENTER_SLEEP_MODE. In SLEEP MODE the device runs on low power. To wake up the device INT_N pin must be activated. This is triggered by a falling edge on INT_N. Before the device enter SLEEP MODE it responds with one of the following:

Response command:

- API_ SUCCESS = Request accepted, entering SLEEP MODE
- API_ FAILURE = Request failed

Note:
There is no way of waking up the device via serial commands. The INT_N pin must be used.

Power save mode (RAM) **API_POWER_SAVE_MODE_RAM**

In POWER SAVE MODE the module automatically shuts down some of the components between each command to lower power consumption. To enter POWER SAVE MODE issue the command Power Save Mode with the value 0 in the IDX-LSB byte. To exit POWER SAVE MODE issue the command Power Save Mode with the value 1 in the IDX-LSB byte. The setting is stored in RAM and the setting is lost after reset. The factory default setting is that the module is NOT in power down mode (value=1).

VALUE (IDX-LSB)	DESCRIPTION
0	Enable power save mode
1	Disable power save mode

Response command:

- API_ SUCCESS = Request accepted, entering POWER SAVE MODE
- API_ FAILURE = Request failed

Note:
If POWER SAVE MODE is enabled and the command CAPTURE_IMAGE is issued the image will NOT be saved in RAM since the needed component will be shut down.

Power save mode (STATIC) **API_POWER_SAVE_MODE_STATIC**

In POWER SAVE MODE the module automatically shuts down some of the components between each command to lower power consumption. To enter POWER SAVE MODE issue the command Power Save Mode with the value 0 in the IDX-LSB byte. To exit POWER SAVE MODE issue the command Power Save Mode with the value 1 in the IDX-LSB byte. The setting is stored in non-volatile memory and the setting is saved even after reset. This means that the factory default setting (value=1) will be changed.

VALUE (IDX-LSB)	DESCRIPTION
0	Enable power save mode
1	Disable power save mode

Response command:

- API_ SUCCESS = Request accepted, entering POWER SAVE MODE
- API_ FAILURE = Request failed

Note:
If POWER SAVE MODE is enabled and the command CAPTURE_IMAGE is issued the image will NOT be saved in RAM since the needed component will be shut down.

Get current power save mode **API_GET_POWER_SAVE_MODE**

This commands returns the value of the current setting of power save mode.
The value is sent as payload data.

VALUE	DESCRIPTION
0	Power save mode enabled
1	Power save mode disabled

Response command:

- API_ SUCCESS = Command OK
- API_ FAILURE = Command fail

CRC calculation

The CRC calculation uses 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 * 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 FPC-AM.

The source code for the CRC implementation is available on the CD, or as a download from the Fingerprint Cards homepage. It should compile with no or small changes on most environments.

Mechanical outline

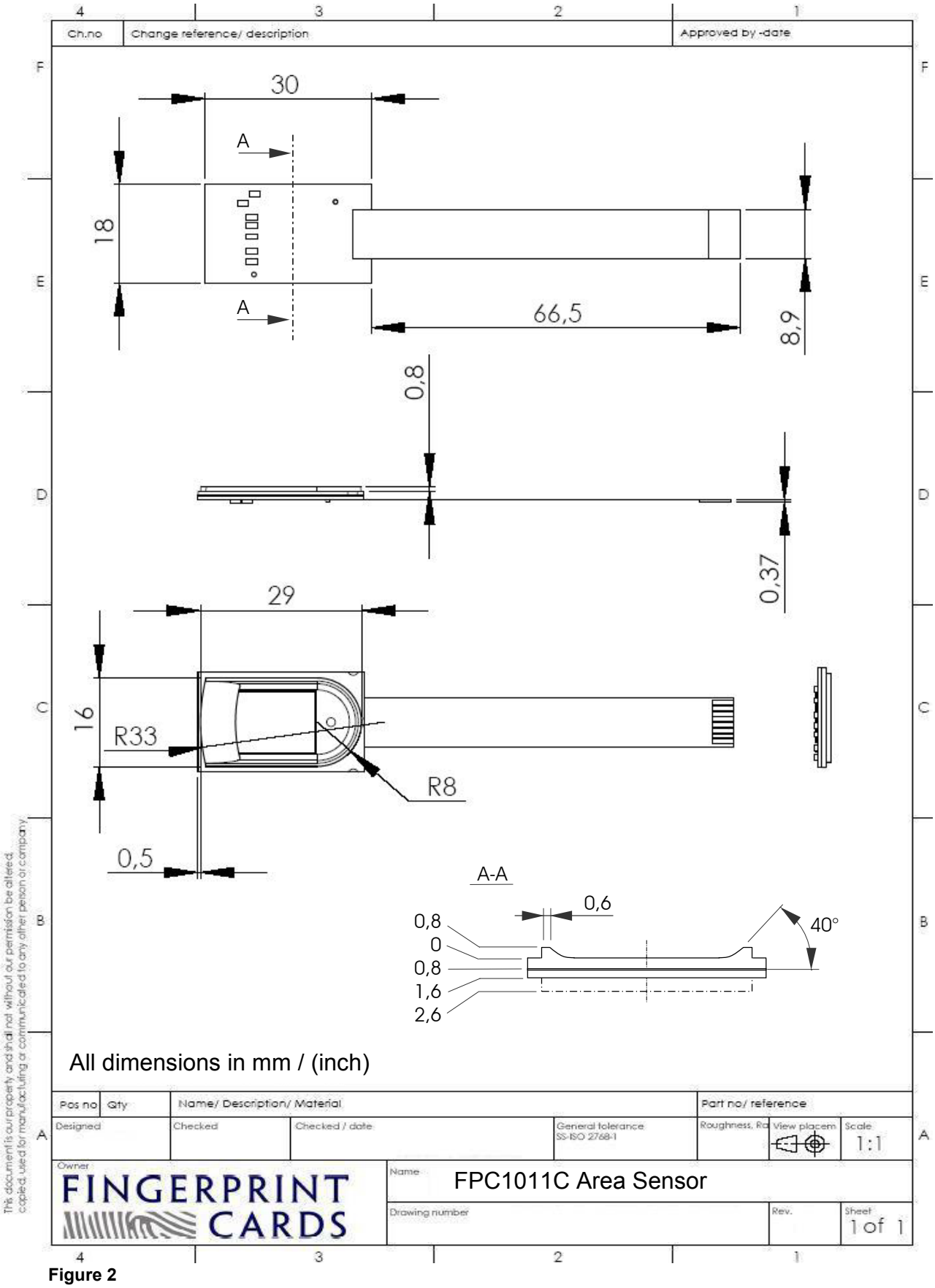


Figure 2
Mechanical drawing FPC1011C Area sensor. Drawing is not to scale.

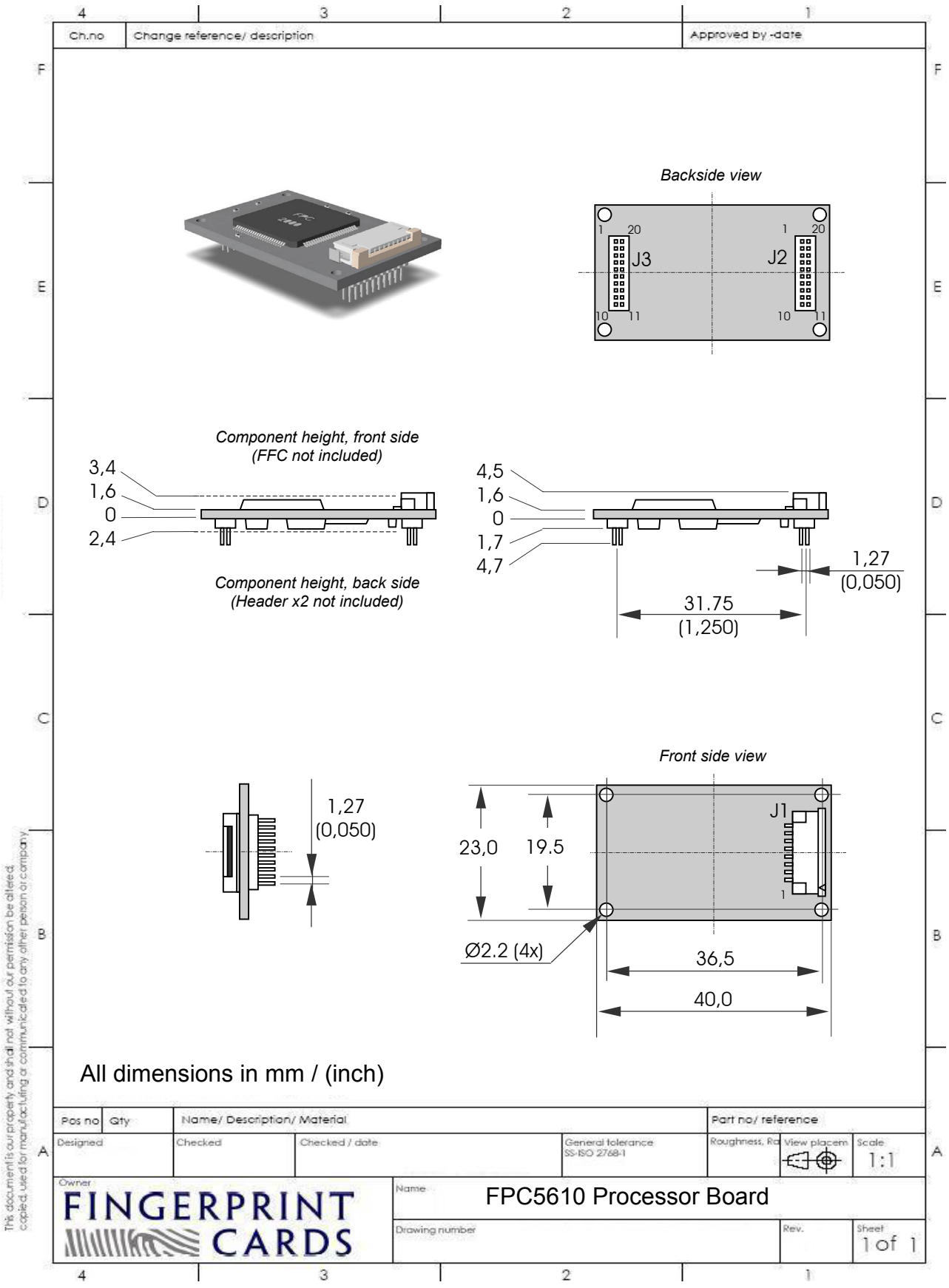


Figure 3
Mechanical drawing FPC5610 Processor board. Drawing is not to scale.

Hardware interface

Package details

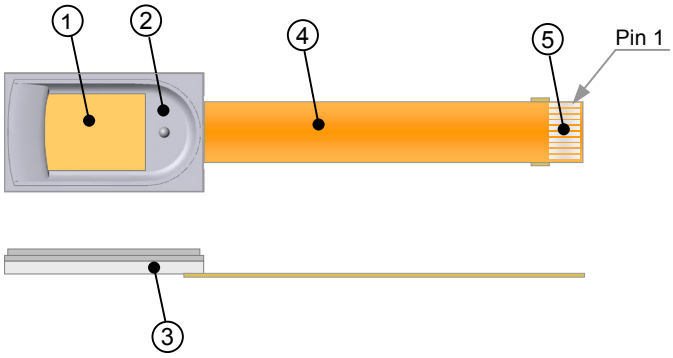


Figure 4
FPC1011C Device overview

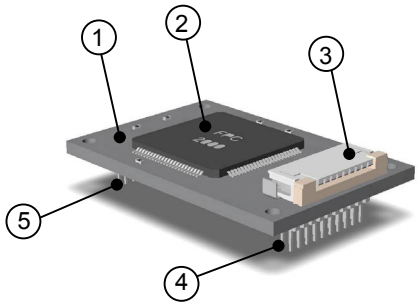


Figure 5
FPC5610 Device overview

ITEM	DESCRIPTION	
1	FPC1011 fingerprint area sensor chip	
2	Conductive frame	
3	Ceramic substrate:	0.63 mm
4	Flex film cable:	150 um
	Minimum recommended bend radius	
	Static application:	0.3 mm
	Dynamic application:	1.5 mm
5	Tin plated connectors:	4.0 x 0.7 mm

Table 7
FPC1011C Package details

ITEM	DESCRIPTION	
1	FR4 substrate, black, 1.6 mm	
2	FPC2000Q processor ASIC	
3	ZIF connector, 8 pin, 1.0 mm pitch: e.g. Molex 52207-0890	J1
4	Dual header, 20 pin, 1.27 mm (0,05") pitch:	J2-1
5	Dual header, 20 pin, 1.27 mm (0,05") pitch:	J3-1

Mating SMD socket: e.g. Harwin M50-3151022
Mating ribbon cable: e.g. Samtec FFSD-10-D-02.00-01-N

Table 8
FPC5610 Package details

Pin assignment

PIN	SIGNAL NAME	DESCRIPTION
J3:1	GND	Signal ground
J3:2	GND	Signal ground
J3:3	GND	Signal ground
J3:4	GPIO_3	*Future functionality*
J3:5	GPIO_2	*Future functionality*
J3:6	GPIO_1	*Future functionality*
J3:7	GPIO_0	*Future functionality*
J3:8	RD_N	*Future functionality*
J3:9	WR_N	*Future functionality*
J3:10	ALE	*Future functionality*
J3:11	RST_N	System reset, active low
J3:12	SPI_MISO	*Future functionality*
J3:13	SPI_MOSI	*Future functionality*
J3:14	SPI_SCK	*Future functionality*
J3:15	SPI_SS_N	*Future functionality*
J3:16	INT_N	Interrupt, falling edge
J3:17	UART_TX	Serial data output
J3:18	UART_RX	Serial data input
J3:19	VDD	Power supply 3.3 V
J3:20	VDD	Power supply 3.3 V

PIN	SIGNAL NAME	DESCRIPTION
J2:1	GND	Signal ground
J2:2	DATA_7	*Future functionality*
J2:3	DATA_6	*Future functionality*
J2:4	DATA_5	*Future functionality*
J2:5	DATA_4	*Future functionality*
J2:6	DATA_3	*Future functionality*
J2:7	DATA_2	*Future functionality*
J2:8	DATA_1	*Future functionality*
J2:9	DATA_0	*Future functionality*
J2:10	GND	Signal ground
J2:11	GND	Signal ground
J2:12	S_GND	Sensor ground
J2:13	S_CS_N	Chip select, active low
J2:14	S_SI	SPI data input
J2:15	S_GND	Sensor ground
J2:16	S_SCK	SPI clock input
J2:17	S_RST_N	Sensor reset, active low
J2:18	S_VDD	Sensor power supply
J2:19	S_SO	SPI data output
J2:20	GND	Signal ground

Table 9
FPC5610 Pin configuration

Note:
1) Several pins (J3:4 - J3:10, J3:12 - J3:15, J2:2 - J2:9) are reserved for future functionality and are not necessary to connect.

2) An alternative sensor interface is available on pin J2:12 - J2:19. This feature facilitates a remote sensor connection if this is more convenient, e.g. the ZIF connector can be placed on the opposite side of the main board. Unless these pins are used instead of the topside connector (J1), they are not necessary to connect.

Hardware control pins
- RST_N = A system (module) reset is carried out by pulling RST_N pin low
- INT_N = INT_N interrupt pin is pulled low for system wake up, when module is placed in SLEEP MODE

Reset to factory defaults
If INT_N is held low when RST_N is pulled low, a hard ware reset will be performed. All static variables are returned to factory defaults; i.e. baud rate = 9600, power save mode = 0x01, security level = 0x05. Module will remain in boot mode until a new firmware is downloaded or RST_N is pulled low once again, and normal command mode is entered.

Application information

Sensor assembly
Thanks to the conductive frame, containing micro ergonomics, a smooth transition to external mechanics can easily be obtained.

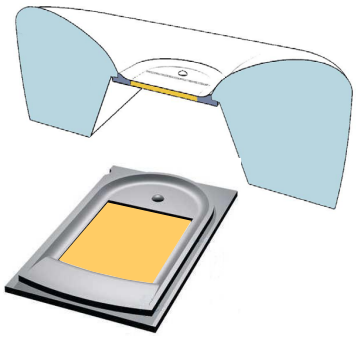


Figure 6
Mounting sensor in enclosure

Note that the sensor and its conductive frame must be mounted in such way that electrical isolation to adjacent conductive surfaces is achieved. Otherwise the sensor operation may be degraded.

Figure 7
Sensor keep-out and isolation

On the backside of the sensor substrate a number of passive components are placed. The overall maximum height is **1.0 mm**.

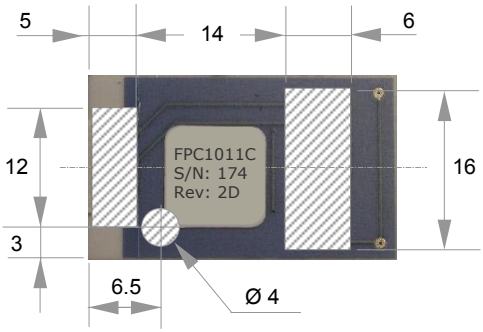


Figure 8
Sensor keep-out and back side layout

Board assembly
The processor board is furnished with two dual row pin headers. These miniaturized connectors facilitate a number of mounting alternatives. For instance, the processor board can be attached directly to the main board using a board-to-board connector.

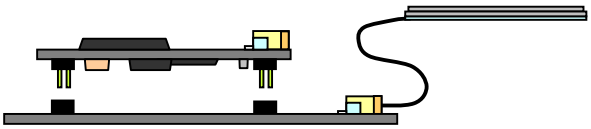


Figure 9
Board-to-board and external flex connector

Above the sensor is connected directly to the main board and the processor board is treated as an ordinary component.

Keep in mind that the separate flex connector **cannot** be placed to far away from the processor board. Less than **200 mm** is recommended.

By selecting sockets with proper height, the space underneath the processor board can also used for placing components on the main board. Required clearance below the processor board is **2.4 mm**.

Naturally the on-board flex connector (FFC) can be used as well if this is more convienient.

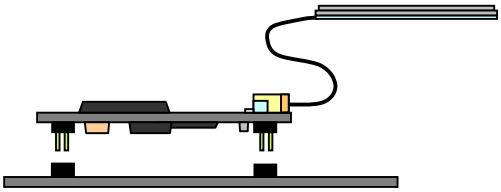


Figure 10
Board-to-board and on-board flex connector

If remote assembly is preferred, a standard high pitch ribbon cable can be used. Only the left header (J3) needs to be connected.

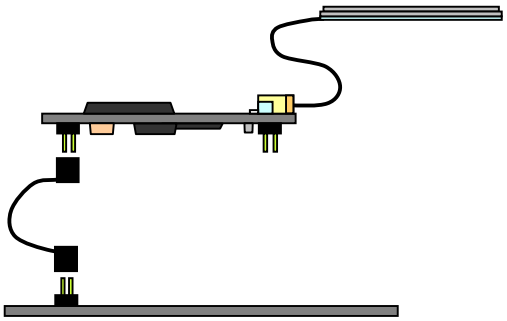


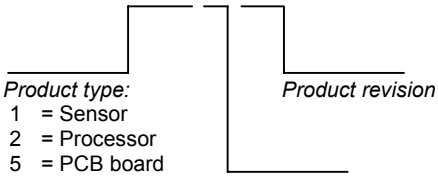
Figure 11
Ribbon cable and on-board flex connector

Ordering information

The FPC-AM module consists of an area sensor and a processor board.

Product no:

FPC1011C	-2A	*Area sensor*
FPC5610	-0A	*Processor board*



Product type:
1 = Sensor
2 = Processor
5 = PCB board
6 = Assembly

Product revision

Packaging option:
W = Wafer
N = Naked die
C = package type C, ceramic substrate

Contact information

Main office
Fingerprint Cards AB
P.O. Box 2412
SE-403 16 Göteborg
Sweden

Visiting address
Västra Hamngatan 8

Telephone
+46 (0)31 60 78 20

Fax
+46 (0)31 13 73 85

Web site
www.fingerprints.com

E-mail
sales@fingerprints.com
tech@fingerprints.com