

Biometric Project

-D.Sankirth Reddy(14/IT/85)

1)AIM:

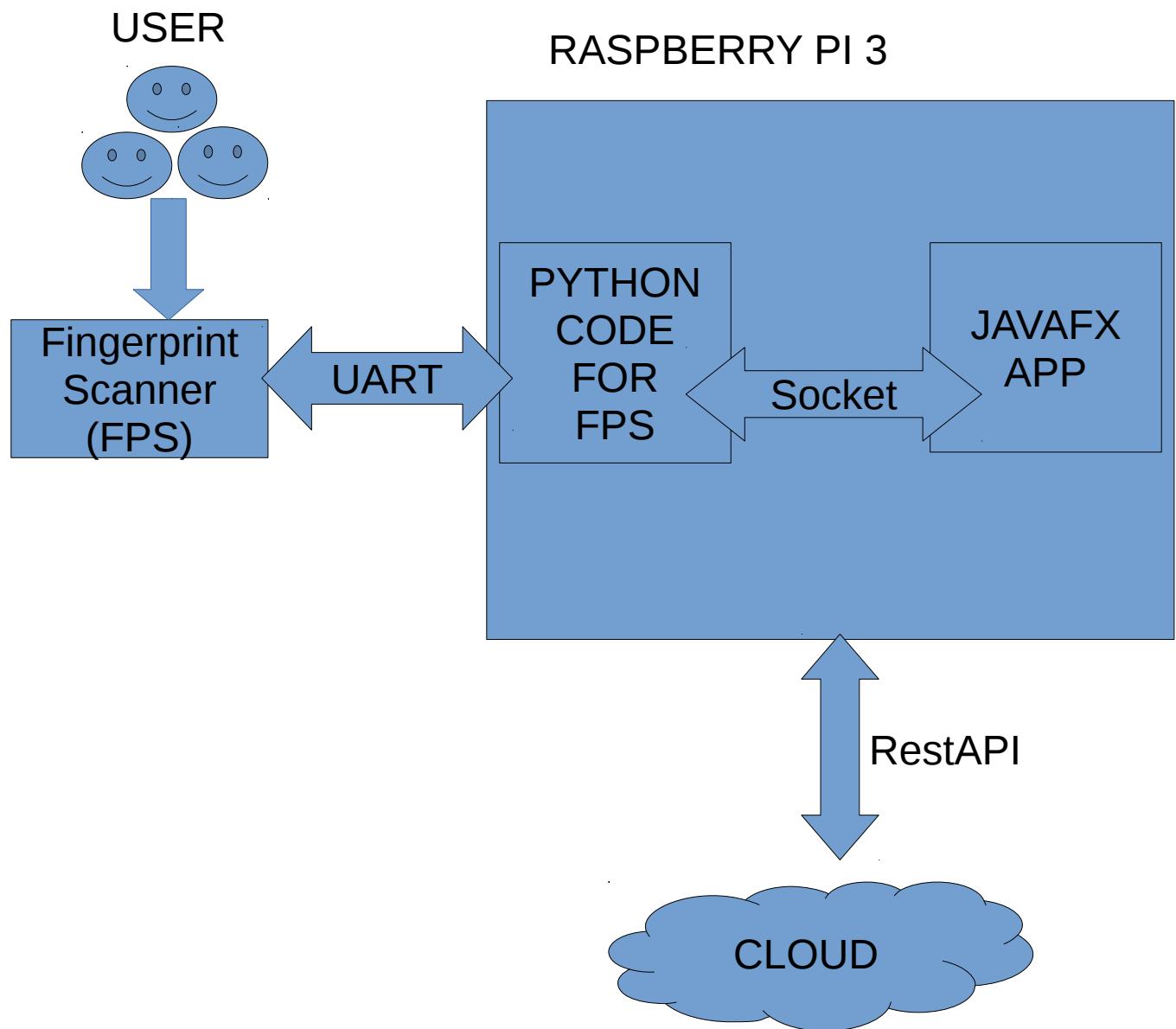
Attendance Management System using Raspberry pi and FingerPrint Scanner (GT-511C3).

2)ABSTRACT:

Here, we are going to make an application which has features such as enrollment,attendance,delete and Sync.

- Enrollment is for enrolling the student,after enrollment is done,he will be given a unique enroll id.
- In Attendance,student keeps his finger on the scanner and if the finger is enrolled,attendance is taken,else it returns error.
- Delete is for deleting the fingerprint of the student from the database,here the enroll ID is sent as parameter.
- Sync is to sync the data from the server such as studentname,batchID,etc from server during first login and to send the attendance details to the server after the attendance is taken.

The block diagram of the entire process is as below:

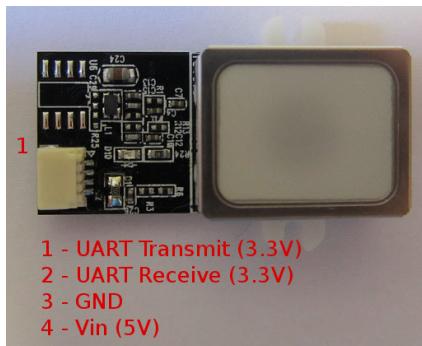


3)REQUIRED RESOURCES:

- Raspberry pi 3
- Fingerprint Scanner(GT-511C3)
- Raspberry pi 3 Touchscreen
- MicroSD Card

4)Connecting Raspberry pi 3 and FingerPrint scanner:

- Tx-->Rx
- Rx-->Tx
- GND-->GND
- Vin-->5V power

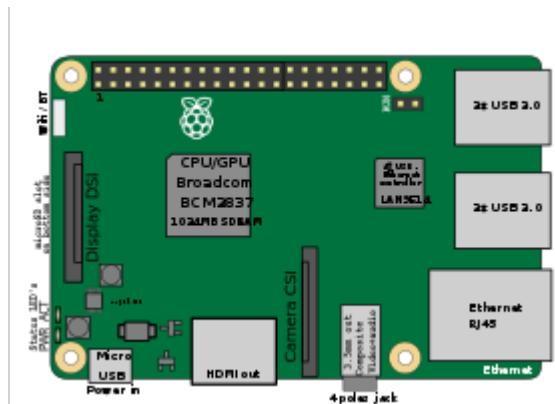


Alternate Function			Alternate Function
	3.3V PWR	1	2 5V PWR
I2C1 SDA	GPIO 2	3	4 5V PWR
I2C1 SCL	GPIO 3	5	6 GND
	GPIO 4	7	8 UART0 TX
	GND	9	10 UART0 RX
	GPIO 17	11	12 GPIO 18
	GPIO 27	13	14 GND
	GPIO 22	15	16 GPIO 23
	3.3V PWR	17	18 GPIO 24
SPI0 MOSI	GPIO 10	19	20 GND
SPI0 MISO	GPIO 9	21	22 GPIO 25
SPI0 SCLK	GPIO 11	23	24 GPIO 8 SPI0 CS0
	GND	25	26 GPIO 7 SPI0 CS1
	Reserved	27	28 Reserved
	GPIO 5	29	30 GND
	GPIO 6	31	32 GPIO 12
	GPIO 13	33	34 GND
SPI1 MISO	GPIO 19	35	36 GPIO 16 SPI1 CS0
	GPIO 26	37	38 GPIO 20 SPI1 MOSI
	GND	39	40 GPIO 21 SPI1 SCLK

Here, the communication between the two devices is in UART protocol (Refer UART communication)

5)About Raspberry pi 3:

- The **Raspberry Pi** is a series of small **single-board computers** developed by the Raspberry pi foundation
- Raspbian OS is 32-bit ARMv7 compatible mode and so the OS
- Dimensions: 86.9mm x 58.5mm x 19.1mm / 3.4" x 2.3" x 0.8"
- Weight: 41.2g / 1.5oz
- RAM: 1GB
- Processor: quad-core Cortex-A53 processor



6) About FingerPrint Scanner(GT-511C3):

This device is one chip module with;

- fingerprint algorithm
- optical sensor

The major functions are the followings.

- High-accuracy and high-speed fingerprint identification technology
- Ultra-thin optical sensor
- 1:1 verification, 1:N identification
- downloading fingerprint image from the device
- Reading & writing fingerprint template(s) from/to the device
- Simple UART & USB communication protocol

Technical Specification

Item	Value
CPU	ARM Cortex M3 Core
Sensor	optical Sensor
Effective area of the Sensor	14 x 12.5(mm)
Image Size	202 x 258 Pixels
Resolution	450 dpi
The maximum number of fingerprints	200 fingerprints
Matching Mode	1:1, 1:N
The size of template	496 Bytes (template) + 2 Bytes (checksum)
Communication interface	UART, default baud rate = 9600bps after power on USB Ver1.1, Full speed
False Acceptance Rate (FAR)	< 0.001%
False Rejection Rate(FRR)	< 0.1%
Enrollment time	< 3 sec (3 fingerprints)
Identification time	< 1.0 sec (200 fingerprints)
Operating voltage	DC 3.3~6V
Operating current	< 130mA

6.1)Functionalities of the FingerPrint Scanner:

- Enrollment
- Identification
- Verification
- DeleteID
- DeleteALL
- CheckEnrolled
- GetEnrollCount

The code for the above functionalities is to be written in a library.

- Every functionality follows pre-defined protocols.
- There are three types of packets.
 - Command Packet
 - Response Packet
 - Data Packet(Currently, we deal with Command and Response packet).

6.2) PACKET STRUCTURE

Command Packet

OFFSET	ITEM	TYPE	DESCRIPTION
0	0x55	BYTE	Command start code1
1	0xAA	BYTE	Command start code2
2	<i>Device ID</i>	WORD	Device ID: default is 0x0001, always fixed
4	<i>Parameter</i>	DWORD	Input parameter
8	<i>Command</i>	WORD	Command code
10	<i>Check Sum</i>	WORD	Check Sum (byte addition) OFFSET[0]+...+OFFSET[9]= <i>Check Sum</i>

Response Packet

OFFSET	ITEM	TYPE	DESCRIPTION
0	0x55	BYTE	Response start code1
1	0xAA	BYTE	Response start code2
2	<i>Device ID</i>	WORD	Device ID: default is 0x0001, always fixed
4	<i>Parameter</i>	DWORD	Response == 0x30: (ACK) Output Parameter Response == 0x31: (NACK) Error code
8	<i>Response</i>	WORD	0x30: Acknowledge (ACK). 0x31: Non-acknowledge (NACK).
10	<i>Check Sum</i>	WORD	Check Sum (byte addition) OFFSET[0]+...+OFFSET[9]= <i>Check Sum</i>

Data Packet

OFFSET	ITEM	TYPE	DESCRIPTION
0	0x5A	BYTE	Data start code1
1	0xA5	BYTE	Data start code2
2	<i>Device ID</i>	WORD	Device ID: default is 0x0001, always fixed
4	<i>Data</i>	N BYTES	N bytes Data The size is pre-defined per protocol stage
4+N	<i>Check Sum</i>	WORD	Check Sum (byte addition) $\text{OFFSET}[0] + \dots + \text{OFFSET}[4+N-1] = \text{Check Sum}$

6.2.1)Commands:

Number (HEX)	Alias	Description
01	<i>Open</i>	Initialization
02	<i>Close</i>	Termination
03	<i>UsbInternalCheck</i>	Check if the connected USB device is valid
04	<i>ChangeBaudrate</i>	Change UART baud rate
05	<i>SetIAPMode</i>	Enter IAP Mode In this mode, FW Upgrade is available
12	<i>CmosLed</i>	Control CMOS LED
20	<i>GetEnrollCount</i>	Get enrolled fingerprint count
21	<i>CheckEnrolled</i>	Check whether the specified ID is already enrolled
22	<i>EnrollStart</i>	Start an enrollment
23	<i>Enroll1</i>	Make 1 st template for an enrollment
24	<i>Enroll2</i>	Make 2 nd template for an enrollment
25	<i>Enroll3</i>	Make 3 rd template for an enrollment, merge three templates into one template, save merged template to the database
26	<i>IsPressFinger</i>	Check if a finger is placed on the sensor
40	<i>DeleteID</i>	Delete the fingerprint with the specified ID
41	<i>DeleteAll</i>	Delete all fingerprints from the database
50	<i>Verify</i>	1:1 Verification of the capture fingerprint image with the specified ID
51	<i>Identify</i>	1:N Identification of the capture fingerprint image with the database
30	<i>Ack</i>	Acknowledge.
31	<i>Nack</i>	Non-acknowledge.
60	<i>CaptureFinger</i>	Capture a fingerprint image(256x256) from the sensor

6.2.2)Error Codes:

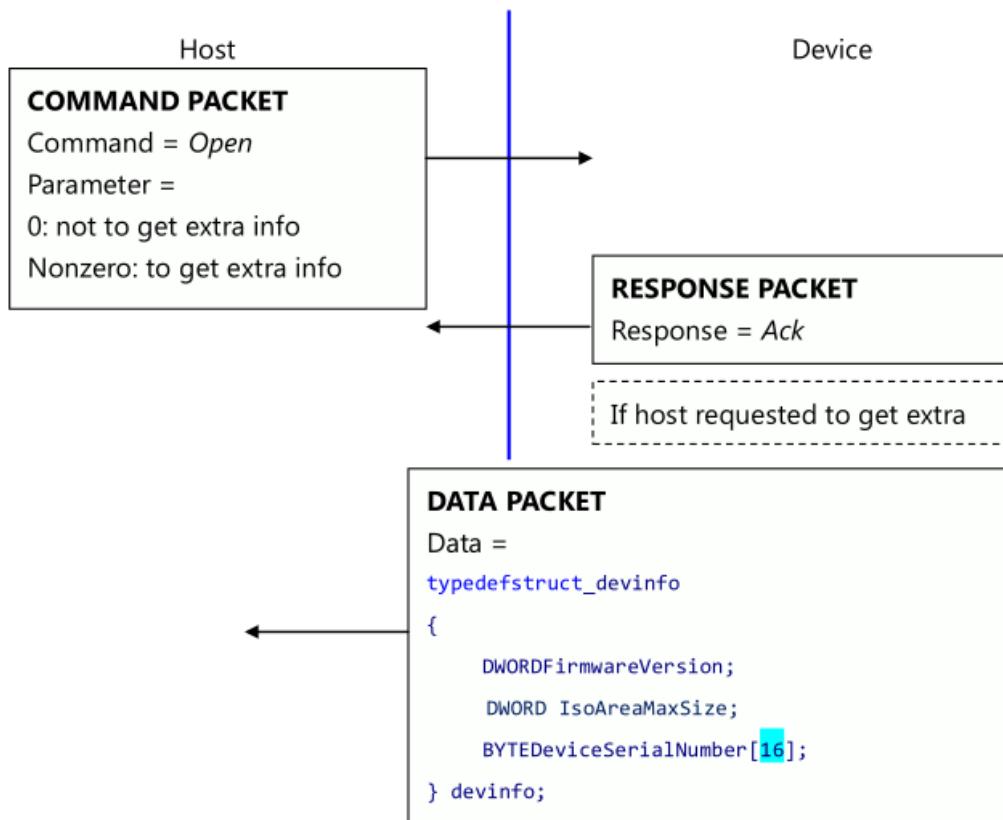
NACK Parameter	Value	Description
NACK_TIMEOUT	0x1001	Obsolete , capture timeout
NACK_INVALID_BAUDRATE	0x1002	Obsolete , Invalid serial baud rate
NACK_INVALID_POS	0x1003	The specified ID is not between 0~199
NACK_IS_NOT_USED	0x1004	The specified ID is not used
NACK_IS_ALREADY_USED	0x1005	The specified ID is already used
NACK_COMM_ERR	0x1006	Communication Error
NACK_VERIFY_FAILED	0x1007	1:1 Verification Failure
NACK_IDENTIFY_FAILED	0x1008	1:N Identification Failure
NACK_DB_IS_FULL	0x1009	The database is full
NACK_DB_IS_EMPTY	0x100A	The database is empty
NACK_TURN_ERR	0x100B	Obsolete , Invalid order of the enrollment (The order was not as: EnrollStart -> Enroll1 -> Enroll2 -> Enroll3)
NACK_BAD_FINGER	0x100C	Too bad fingerprint
NACK_ENROLL_FAILED	0x100D	Enrollment Failure
NACK_IS_NOT_SUPPORTED	0x100E	The specified command is not supported
NACK_DEV_ERR	0x100F	Device Error, especially if Crypto-Chip is trouble
NACK_CAPTURE_CANCELED	0x1010	Obsolete , The capturing is canceled
NACK_INVALID_PARAM	0x1011	Invalid parameter
NACK_FINGER_IS_NOT_PRESSED	0x1012	Finger is not pressed
Duplicated ID	0 – 199	There is duplicated fingerprint (while enrollment or setting template), This error describes just duplicated ID

6.3)Command Details:

Here, basically we have to build a library in which we send a Command packet to device and receive an acknowledgement(Response packet)

The structure of the packet is shown above(Ref:Packet Structure)

Open:



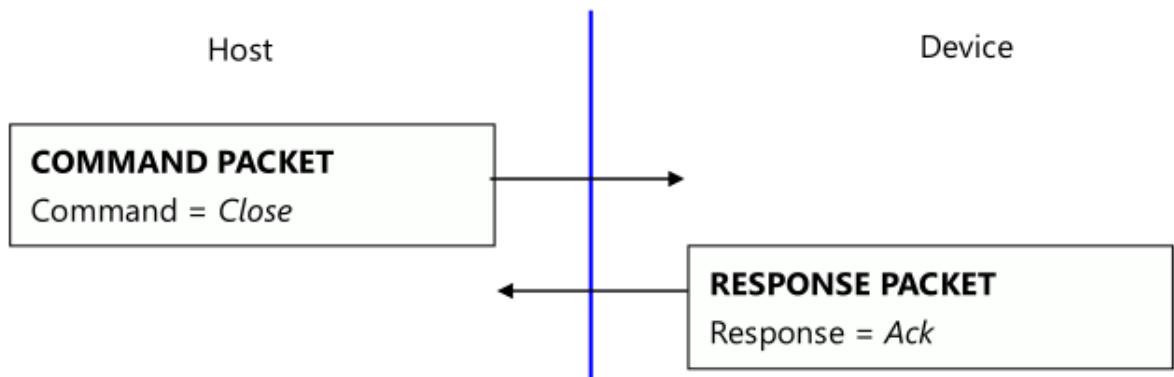
Open command is used to initialize the device; especially it gets device's static info.

Description of *devinfo* structure

Field	Sample	Description
<i>FirmwareVersion</i>	FirmwareVersion: 20120225	Firmware version
<i>IsoAreaMaxSize</i>	IsoAreaMaxSize: 0 KB	Maximum size of ISO CD image
<i>DeviceSerialNumber</i>	DeviceSN: EF15EF4016C66250-888F1A4139000000	Unique serial number of the device

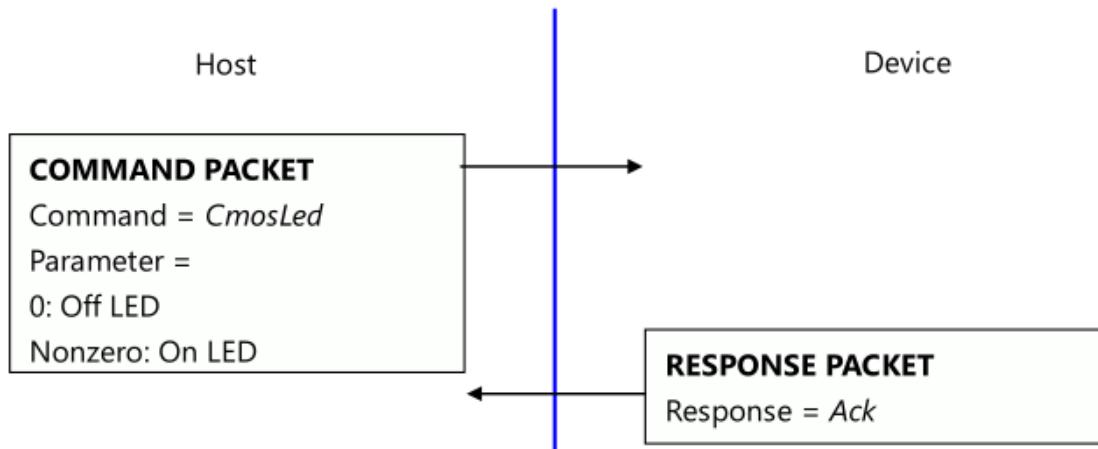
If the Device's Serial Number is zero, then there is no guarantee for stable operation of the device.

Close:



Close command does nothing.

CmosLed:

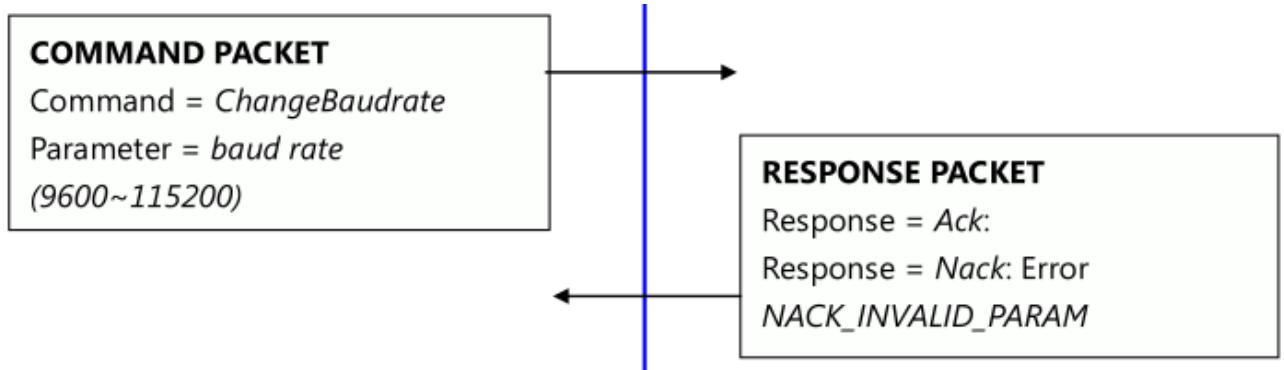


Default state of CMOS (Sensor) LED is OFF state.

(But while booting, LED blinks once, this says the LED is OK.)

Therefore, please issue LED ON command prior to any capture.

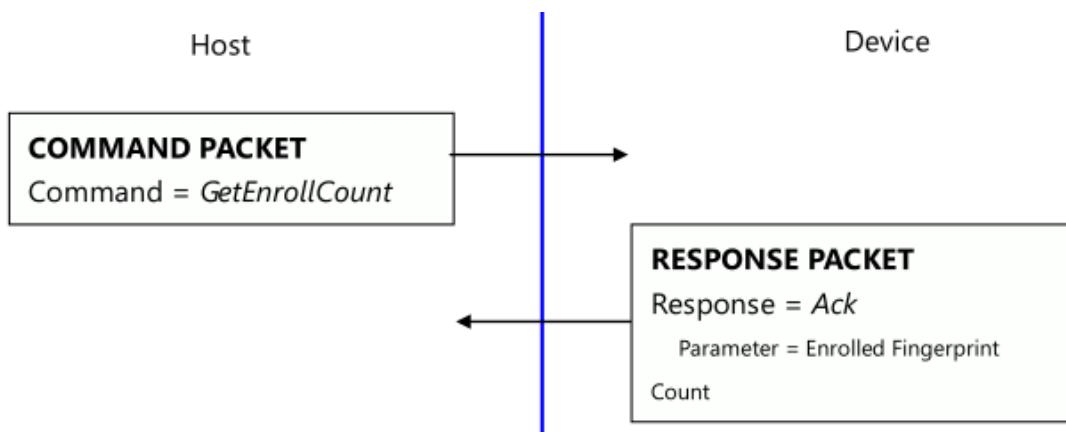
ChangeBaudrate:



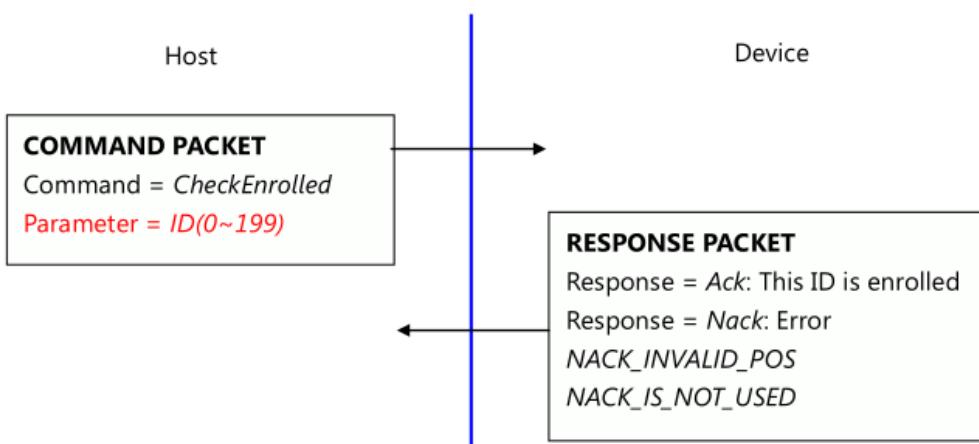
This command changes the UART baud rate at the run-time.

The device initializes its UART baud rate to 9600 bps after power on.

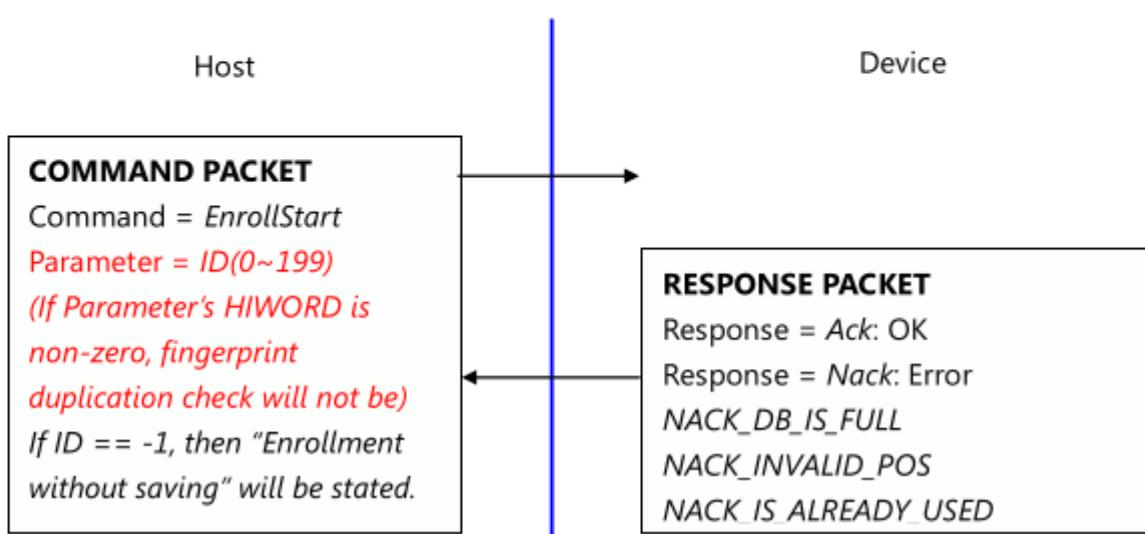
GetEnrollCount:



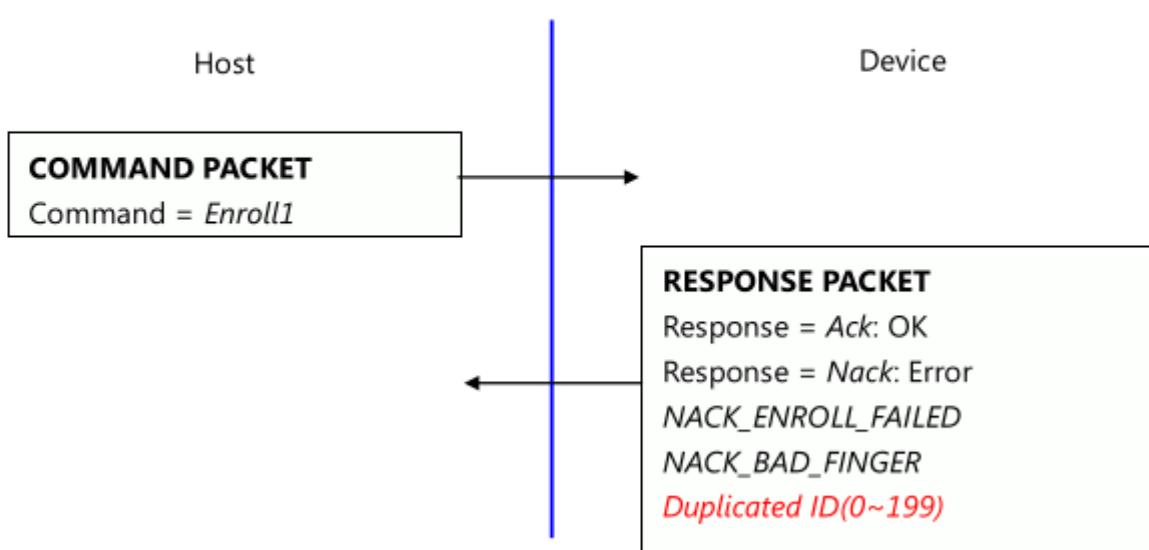
CheckEnrolled :



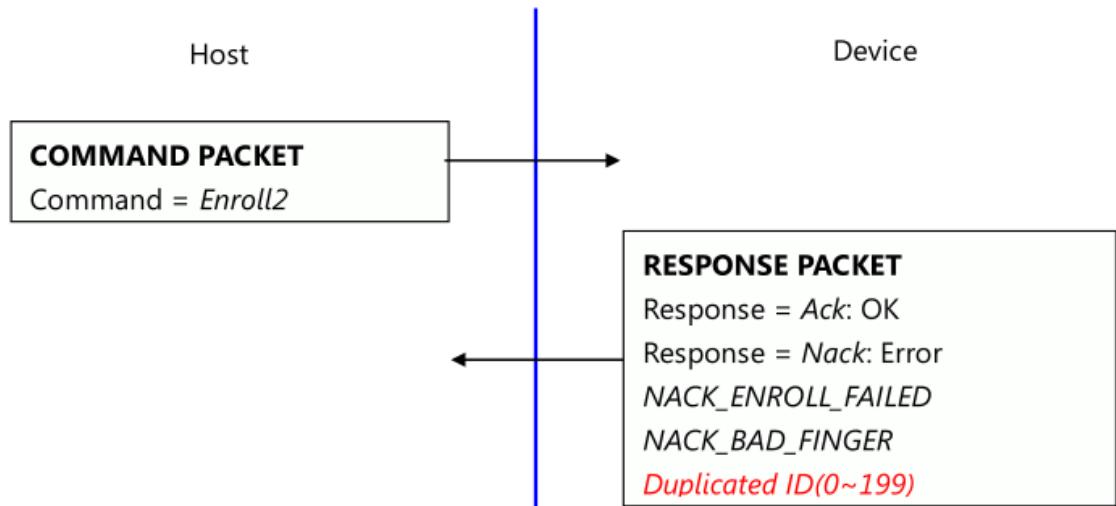
EnrollStart:



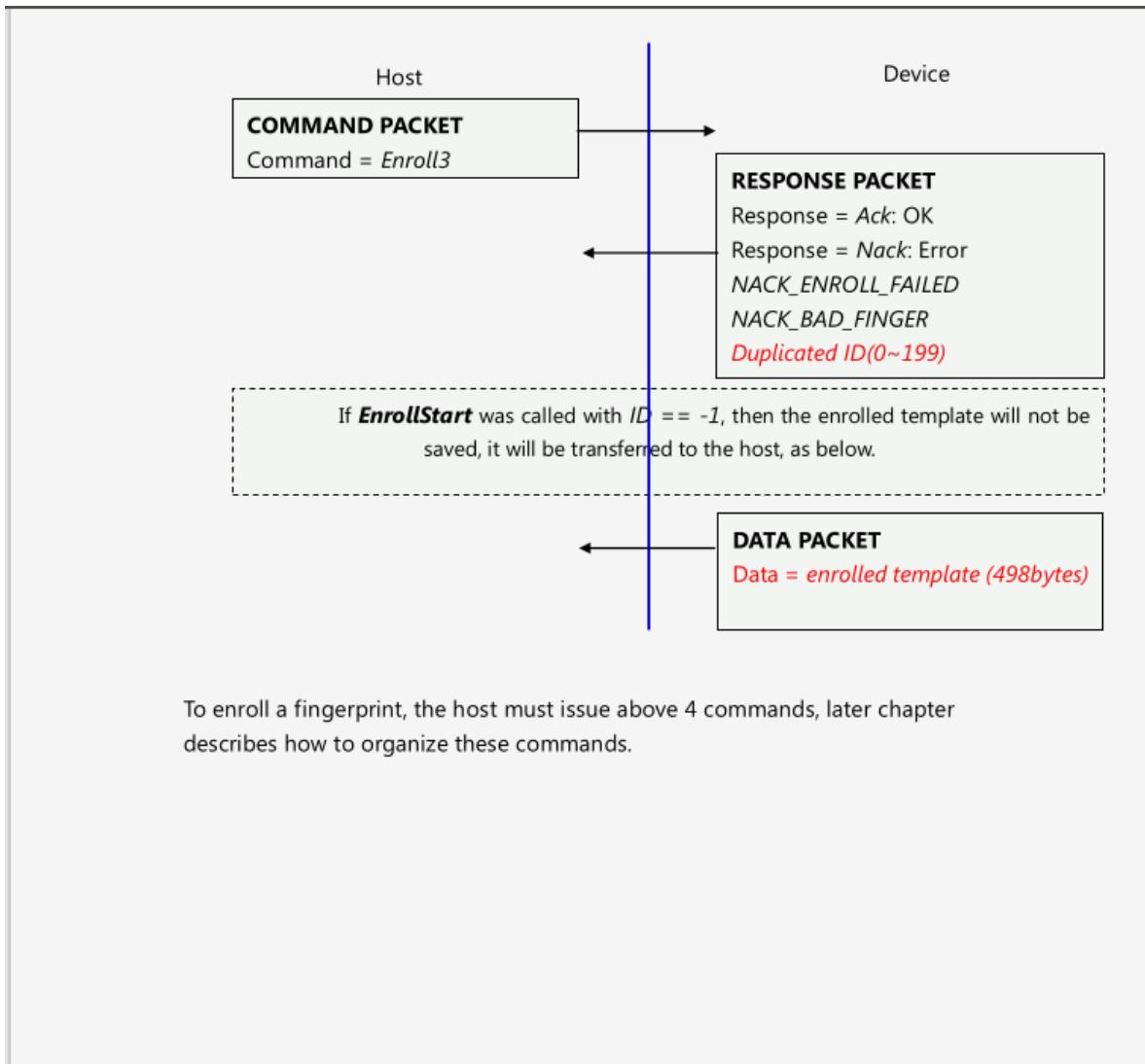
Enroll1:



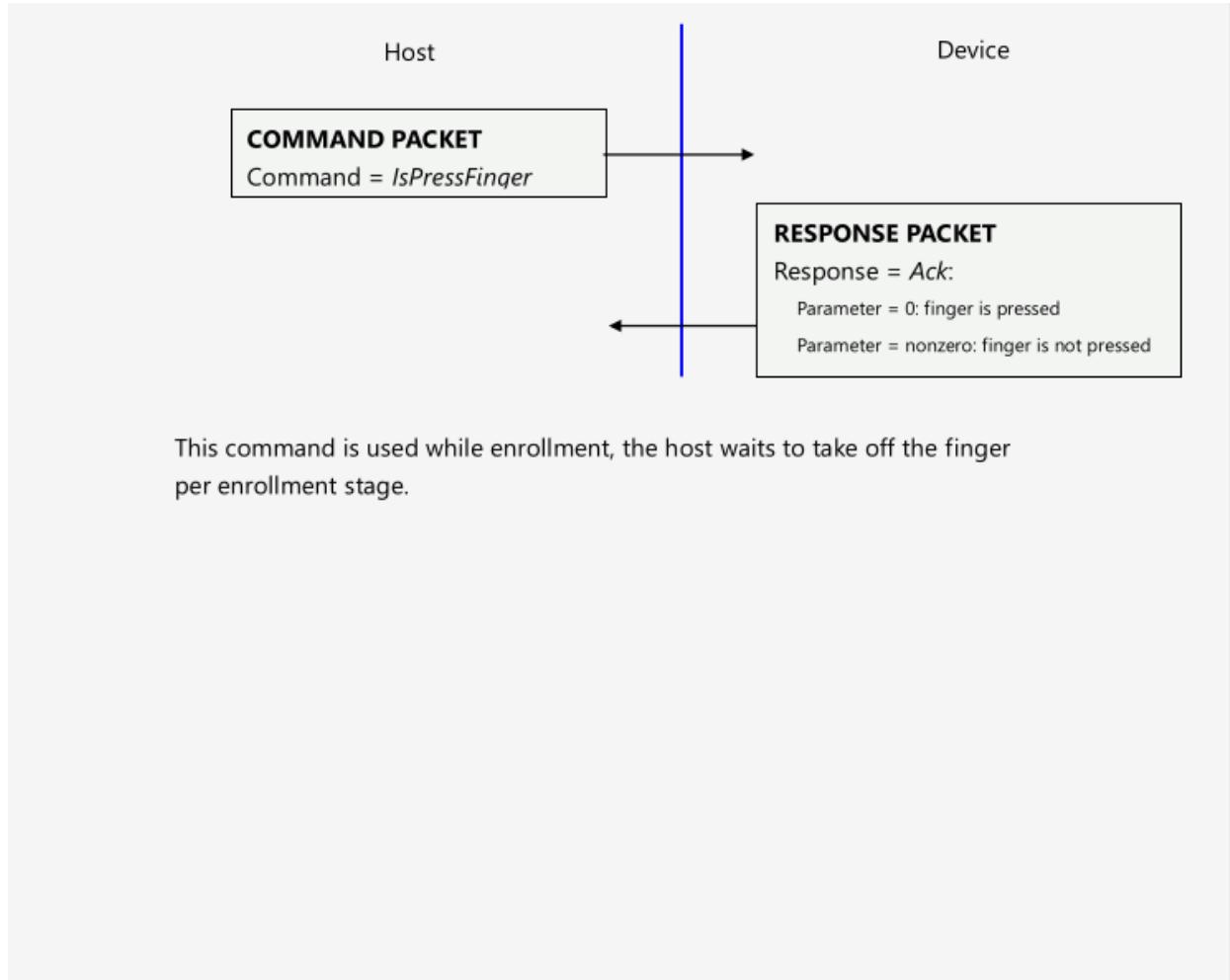
Enroll2



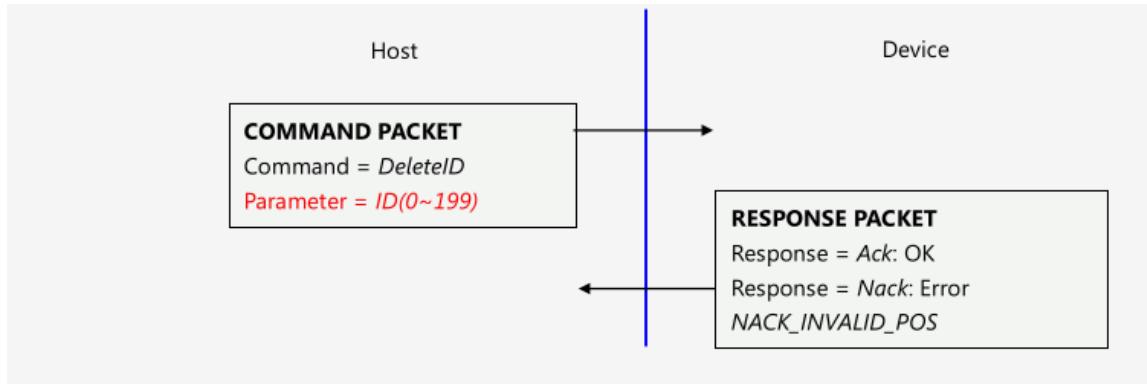
Enroll3:



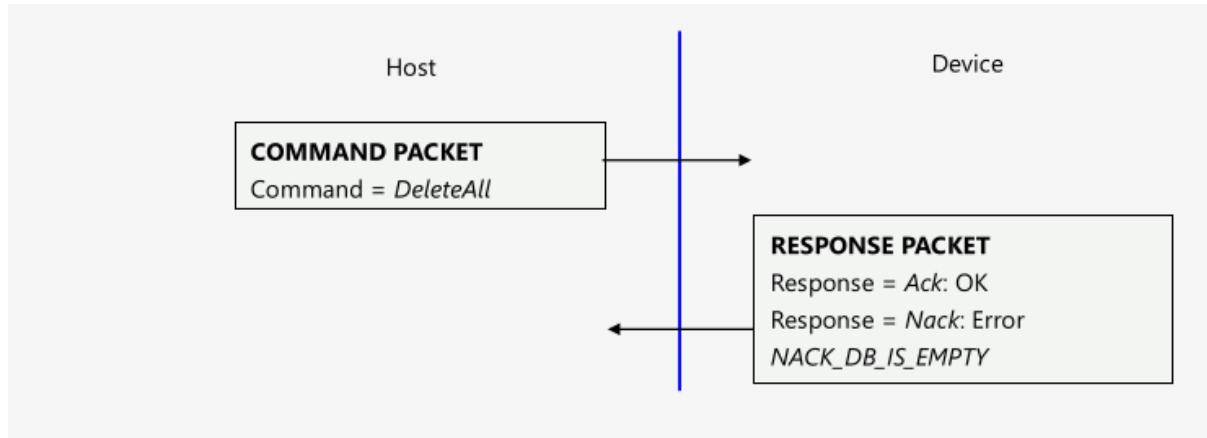
IsPressFinger:



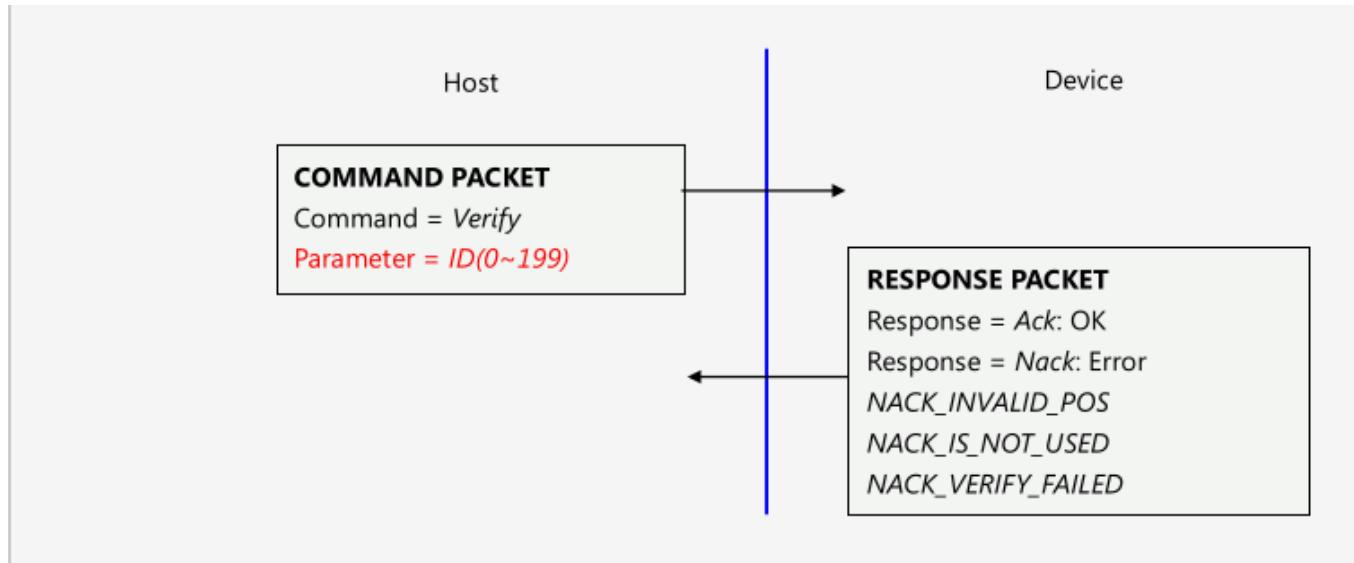
DeleteID:



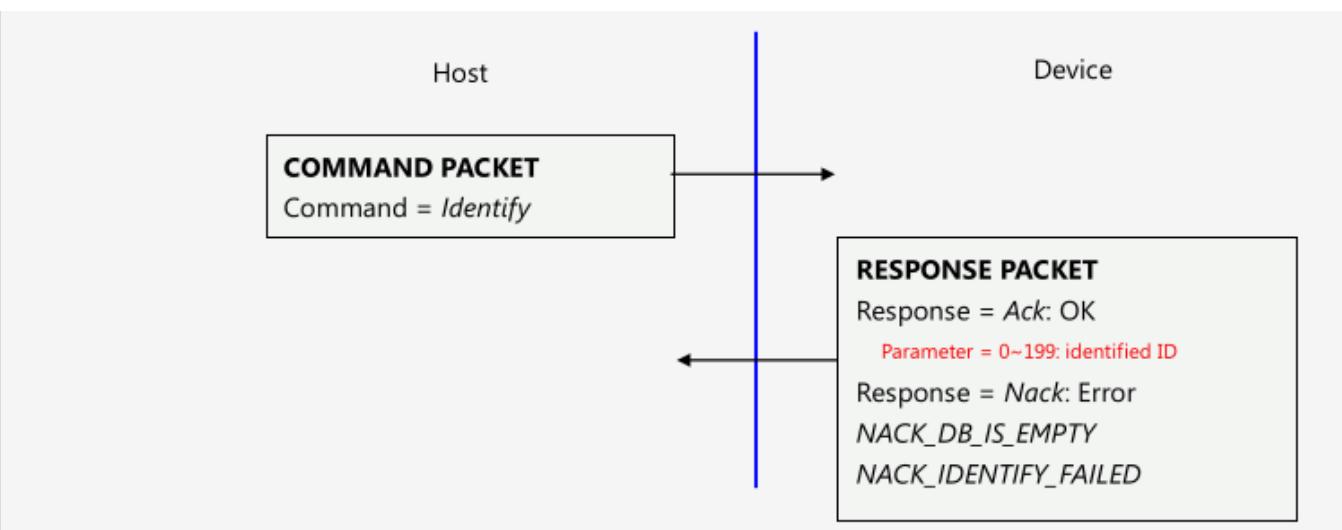
DeleteALL:



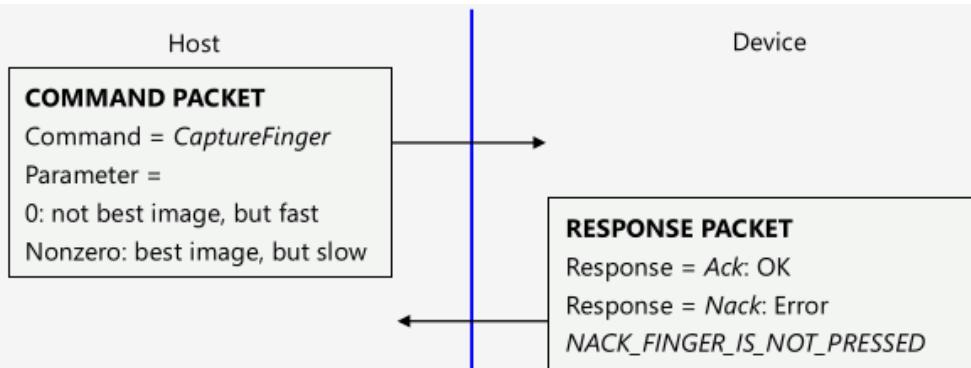
Verify(1:1):



Identify(1:N):



CaptureFinger:



The fingerprint algorithm uses **450dpi** 256x256 image for its input.

This command captures raw image from the sensor and converts it to 256x256 image for the fingerprint algorithm. If the finger is not pressed, this command returns with non-acknowledge.

Please use best image for enrollment to get best enrollment data.

Please use not best image for identification (verification) to get fast user sensibility.

6.4)Description of Functionalities

How Enrollment is done?

The flowchart for enrollment process is as follows

- EnrollStart with a ID(not used)
- CaptureFinger
- Enroll1
- Wait to take off the finger using IsPressfinger
- CaptureFinger
- Enroll2
- Wait to take off the finger using IsPressfinger
- CaptureFinger
- Enroll3

What is IsPressfinger,CaptureFinger?

- IsPressFinger checks whether a finger place on the sensor.This function is used especially while enrollment.
- CaptureFinger captures a fingerprint image(256X256),if a finger isn't placed on the sensor,it returns with error.

What is Identify and Verify?

- Identify perform 1:N matching operation.
- Verify perform 1:1 matching operation.
- Just before calling these functions,host must call CaptureFinger.

Using all the protocols and functions mentioned above [Chapter 6], a library is built in PYTHON, for the functioning of fingerprint scanner.

