Datasheet

PicoLabel 2.7″ v2.x and v3.x

Classification: **Public**

Document Revision: **A**

Table of Contents

1 Features

PicoLabel 2.7″ is an NFC-based passive RFID tag with a 2.7″ ePaper display. PicoLabel 2.7″ integrates low-power digital signage with secure wireless communication.

Key features:

- passive power supply via NFC reader – no battery
- Secure Smart Card Controller NXP A7005 (in PLS-P27_v3.0)
- ISO 14443A-4 communication standard
- password authentication
  -
- 2.7″ Electronic Paper Display (EPD):
  - 264 × 176 pixel resolution, 117 dpi
  - 2-bit color (4 grayscales)
  - ultra-low-power consumption
  - powerless image sustaining
  - very good readability in most lighting conditions

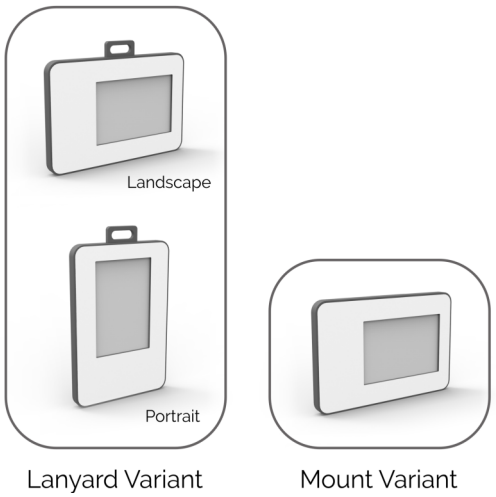
2 Ordering Information

PicoLabel 2.7″ can be ordered in lanyard (landscape/portrait) and mount variants. For details regarding different versions (different level of security) see 6.3.2 Authentication and Authorization.

PicoLabel 2.7″ v2.0 & v3.0 Product Codes

PLS-P27_v2.0 / PLS-P27_v3.0 Landscape Lanyard Variant:	**PLS-P27_v2.0_rA1 / PLS-P27_v3.0_rA1**
PLS-P27_v2.0 / PLS-P27_v3.0 Portrait Lanyard Variant:	**PLS-P27_v2.0_rA2 / PLS-P27_v3.0_rA2**
PLS-P27_v2.0 / PLS-P27_v3.0 Mount Variant:	**PLS-P27_v2.0_rA3 / PLS-P27_v3.0_rA3**

PicoLabel 2.7″ Variants



Lanyard Variant Mount Variant

3 Outline

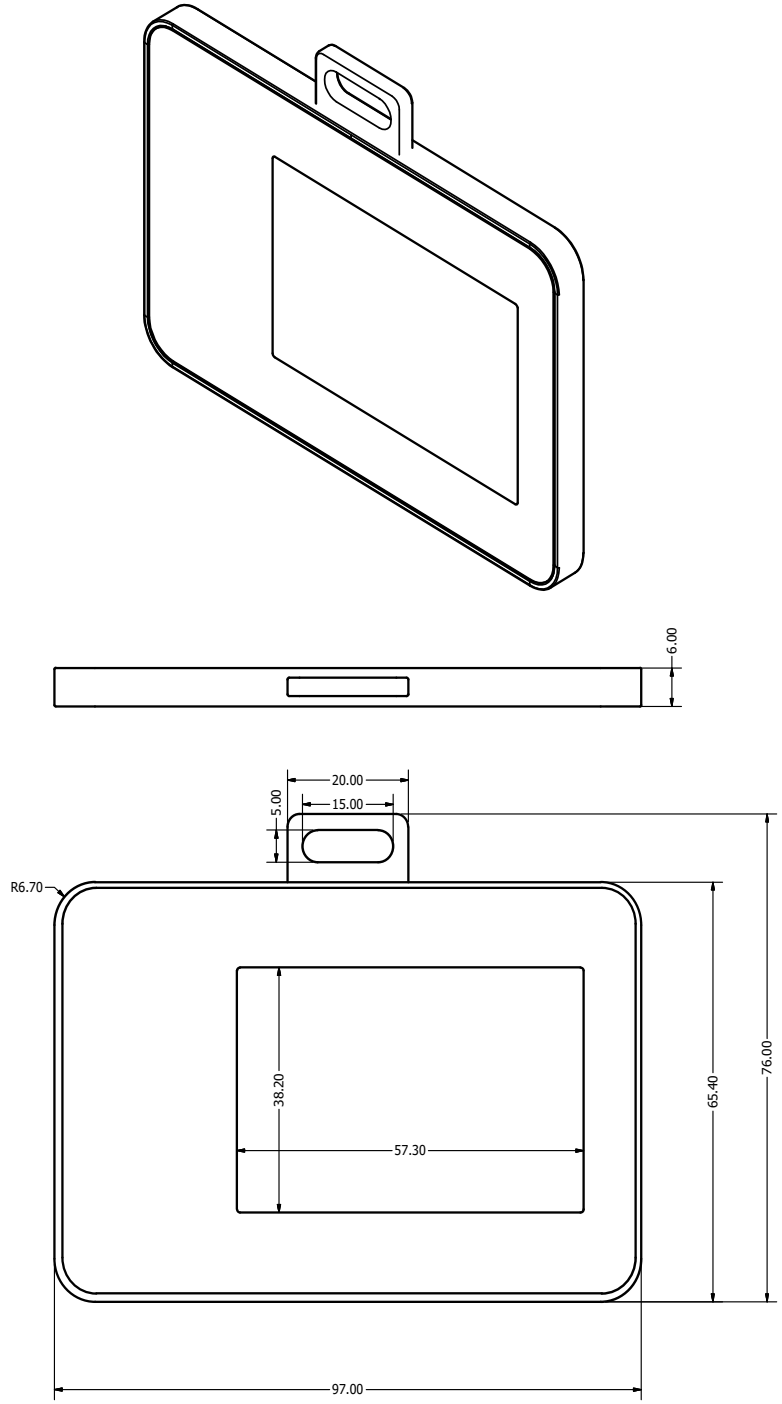


Figure 3.1: PicoLabel 2.7″ main dimensions. Landscape Lanyard Variant. All dimensions are in mm

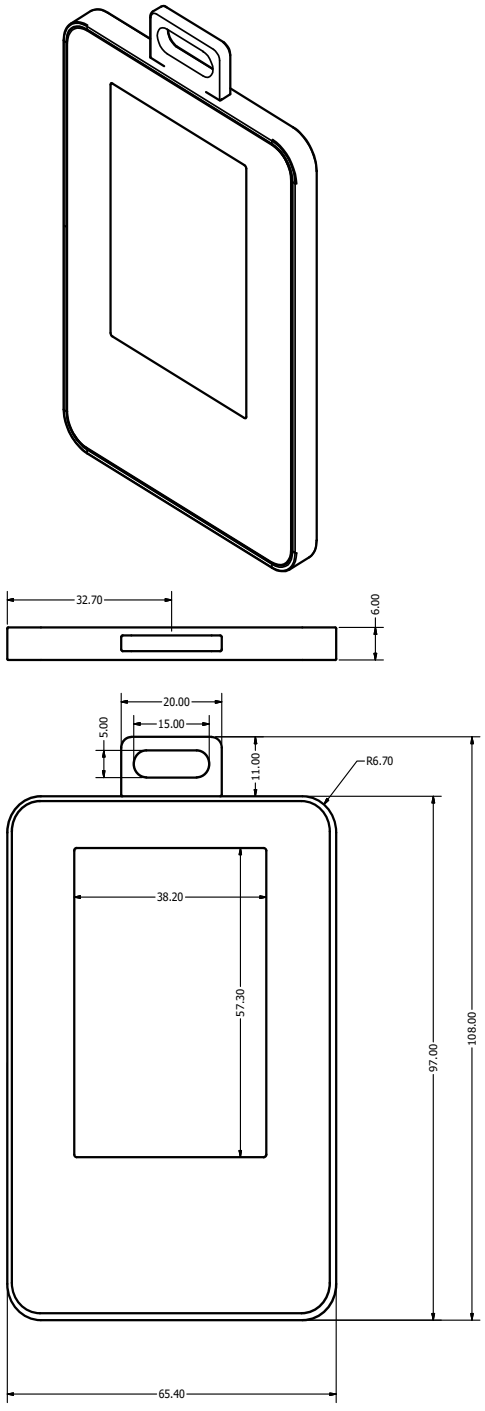


Figure 3.2: PicoLabel 2.7″ main dimensions. Portrait Lanyard Variant. All dimensions are in mm

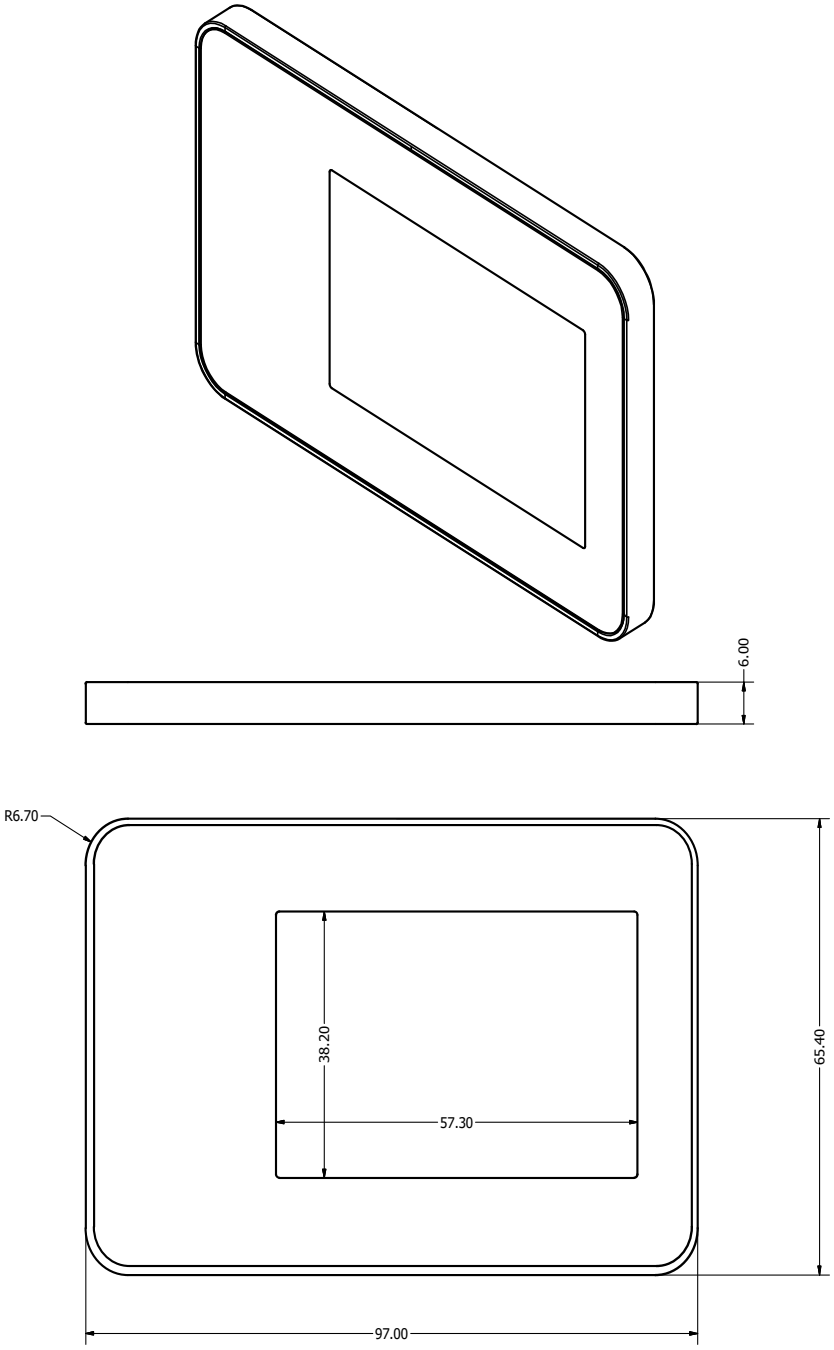


Figure 3.3: PicoLabel 2.7″ main dimensions. Mount Variant. All dimensions are in mm

4 Specification

4.1 Absolute Maximum Ratings

Symbol	Description	Min	Typ	Max	Unit
T_{st}	Storage temperature	-20	–	60	°C
T_{op}	Operating ambient temperature	0	–	50	°C
V_{in}	Input voltage	0	–	6.5	V

4.2 Recommended Operating Conditions

Symbol	Description	Min	Typ	Max	Unit
T_{op}	Operating ambient temperature	0	–	50	°C
V_{in}	Input voltage[1)]	2.4	–	6.0	V

4.3 Characteristics

Description	Min	Typ	Max	Unit
Color depth	1	–	2	bit
Display size	–	2.7	–	inch
Image resolution	–	264 × 176	–	pixel
Display contrast ratio	7:1	–	–	–
Display density	–	117	–	dpi
Display active area size	–	57.3 × 38.2	–	mm
Weight	–	45	–	g
Image upload time	–	0.8	–	s
Display refresh time	–	1.6	–	s
Flash data retention time (powered)	10	–	–	year
Flash data retention time (unpowered)	20	–	–	year
EEPROM data retention time[2)]	10	–	–	year
Flash write endurance[3)]	20,000	200,000	–	cycle
EEPROM write endurance[4)]	100,000	–	–	cycle

1 PicoLabel 2.7″ is contactlessly powered via the reader's magnetic field. There is no physical power connector.
2 Tested at 125 °C. Varies with ambient temperature.
3 Varies with ambient temperature. Flash memory is managed with 2 framebuffer slots, which doubles the original write endurance.
4 Tested at 125 °C. Varies with ambient temperature.

5 NFC Reader

PicoLabel 2.7″ works in cooperation with Omnikey 5021 CL reader.

Graphic below shows the correct alignment of PicoLabel and reader. PicoLabel can be placed either underneath the reader or on top of it.

Figure 6.1: PicoLabel and reader proper alignment

6 Communication Protocol

This section describes the PicoLabel communication protocol for interfacing with a back-end or host.

6.1 Control and Low Level

6.1.1 NFC IC

PicoLabel utilizes ISO 7816-5 APDU protocol for communication with the back-end. Please consult the ISO 7816-5 standard for reference regarding the communication protocol.

6.2 APDU Protocol Description

This paragraph describes the basics of APDU protocol, however it is required to familiarize with the complete APDU protocol specification.

6.2.1 General Protocol Concepts

1) The protocol is designed based on ISO14443/ISO7816 T=1 protocol standard.
   a) Protocol works in half-duplex (request-response) mode and in master-slave configuration:

- An implementation may not include response channel in certain designs (one-way communication possible).

b) The protocol may be overlapped over other higher or lower level protocol and also may overlap other protocol by itself (e.g. may be transmitted over TCP/IP protocol).

2) The protocol is designed for binary transfers (binary protocol).
3) The protocol shall be used for data transfers between the host controller and MCU.

6.2.2 Command Flow

The communication between the host system and the PicoLabel is in half-duplex mode. By default, the communication sequence is as follows:

- Host system sends Command-APDU to the PicoLabel
- Subsequently, PicoLabel sends Response-APDU to the host system.

6.2.3 Protocol Definition

1) The maximum length of APDU isrestricted to 255 bytes.
   a) The protocol may allow the future extension to exceed this limitation.
2) The protocol may optionally not support response data (R-APDU) – by default, R-APDU is supported.
   a) Every command shall have an SW code.
3) The answer will be sent after command handling, by the protocol which supports bi-directional communication.
4) If class/instruction/parameter is not supported it is notified with a proper SW code.
5) End of data transmission is determined by the length of the command.
6) Command execution time should not exceed 10 seconds. 10 seconds it is a maximum time to wait for a response.
7) Every command which does not comply with the APDU protocol reports an error as SW code.

6.2.3.1 Command-APDU (C-APDU)

Field name	Size [bytes]	Description	Presence
`CLA`	1	Class byte – defines set of instructions	Required
`INS`	1	Instruction byte – designates command code	Required
`P1`	1	Parameter 1	Required
`P2`	1	Parameter 2	Required
`Lc`	1	Data length	Optional
`CData`	0 ÷ 250	Data passed in command	Optional
`Le`	1	Expected response data length (number of bytes)	Optional

Table 6.1: C-APDU structure

NOTE `CData` field maximum length may vary according to used variation of APDU. It is required that whole C-APDU has a maximum length of 255 bytes. If response data length is undetermined, `Le` value should equal 0.

Allowed variants:

Short C-APDU	`CLA` `INS` `P1` `P2`
Short C-APDU with data	`CLA` `INS` `P1` `P2` `Lc` `Data...`
Full C-APDU	`CLA` `INS` `P1` `P2` `Lc` `Data...` `Le`

Table 6.2: C-APDU variants

6.2.3.2 Response-APDU (R-APDU)

Field name	Size [bytes]	Description	Presence
`RData`	1 ÷ 253	Response data returned from command	Optional
`SW`	1	Status word	Required

Table 6.3: R-APDU structure

NOTE `RData` length is equal to C-APDU `Le` field value.

If C-APDU `Le` field value equals 0, `RData` length is undetermined.

6.3 C-APDU Fields Description

6.3.1 CLA Handling

`CLA` is responsible for distributing the commands between the EPD controller or and the Smart Card Controller. `CLA` value 0xB5 redirects the command to the EPD controller. All other `CLA` values are redirected to Smart Card Controller though ISO7816 or are stubbed with *Class Not Supported* error (0x6E00) if Smart Card Controller was not detected.

6.3.2 Authentication and Authorization

Executing EPD commands requires an established authenticated channel between back-end and PicoLabel. The authentication mechanism depends on the product variant.

6.3.2.1 Password Security

By default the password is disabled. Enabling password requires executing the SetPassword (see 6.4.7 SetPassword) command. Executing SetPassword without data will disable the password.

Establishing authentication and authorization requires executing UnlockWithPassword (see 6.4.8 UnlockWithPassword) command. GetUID is the only command which does not require authentication.

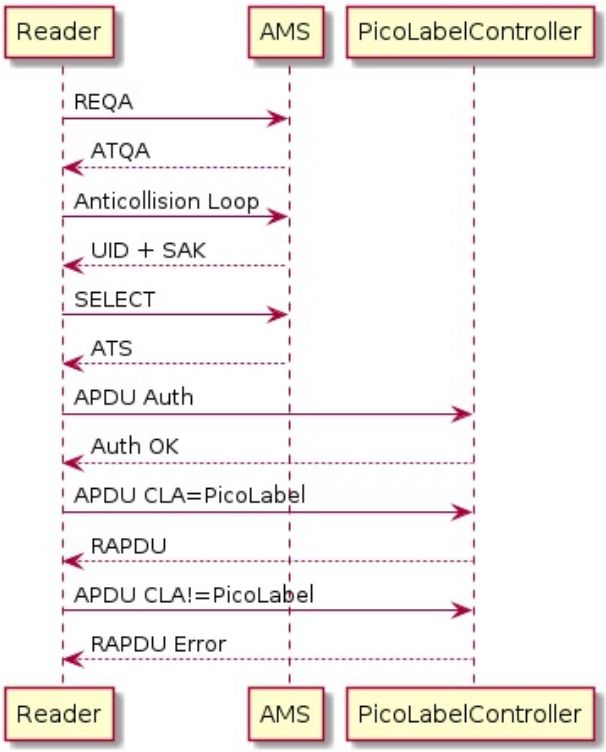


Figure 6.2: PicoLabel 2.7" password security communication flow chart

6.3.2.2 Cryptography-based Security Version (only in v3.0)

After receiving a command, the EPD command parser issues the authorization command of the below-described format to the Smart Card Controller and receives R-APDU from Smart Card Controller. The authorization status is encoded on the R-APDU:

- 0x9000: authorized, command processed
- 0x9804: authorization failure, command not processed

The authorization is valid until the next command sent to the Smart Card Controller. This means that after a successful authorization with Smart Card Controller, all consecutive EPD commands are authorized, until the next command to SCC. GetUID is the only command which does not require authentication.

Identifying the authorization failure reason requires sending the authorization command to PicoLabel.

APDU header of EPD command is encapsulated in authorization APDU issued by EPD controller. This is to allow fine-grained authorization per each EPD command.

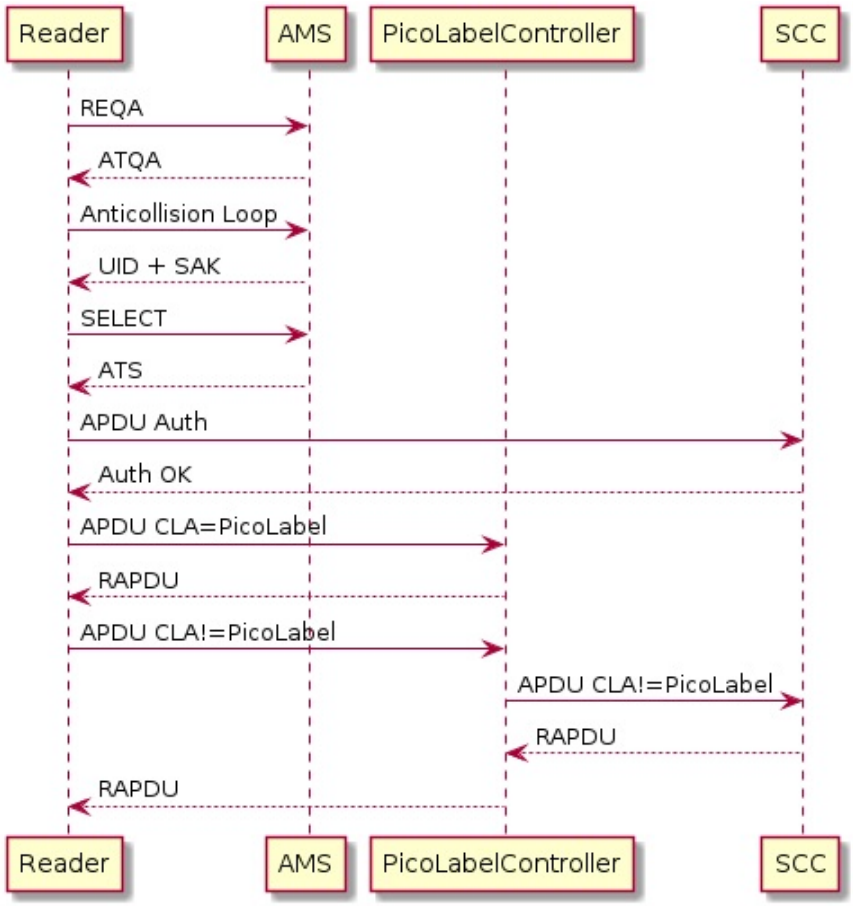


Figure 6.3: PicoLabel 2.7" v3.0 (with cryptography-based security) communication flow chart

6.3.2.2.1 Example Security Applet

Example security applet handling authentication and authorization is pre-loaded on the PicoLabel. Source code is available at https://github.com/MpicoSys/PicoLabel.

6.3.3 EPD Controller C-APDU Instruction List

The table below lists the supported C-APDU instructions (`INS` field values):

Command name	Instruction code (`INS`)	Description
UploadImageData	`0x20`	Uploads image data to PicoLabel
GetImageData	`0xA0`	Downloads image data from PicoLabel
DisplayUpdate	`0x24`	Updates (refreshes) the ePaper display
GetSystemInfo	`0x31`	Returns firmware version
GetUID	`0x30`	Returns unique ID
GetSensorData	`0xE5`	Reads data from the selected sensor
DriveLED	`0x29`	Controls LED
SetPassword	`0x32`	Sets or deletes password phrase
UnlockWithPassword	`0x33`	Authorize and Authenticate command

Table 6.4: C-APDU instruction list

6.3.4 C-APDU Parameters

The parameters have a similar construction in most of the commands.

6.3.4.1 P1 Parameter List

Bit No.	Field name	Size [bits]	Description	Values	Commands
0	RFU	1	Reserved for future use	0	All
1	`P1_SAVE_TO_FLASH`	1	Forces saving new data from RAM to flash memory	0 – none 1 – save data to flash	UploadImageData
2	RFU	1	Reserved for future use	0	All
3	`P1_RESET_POINTER`	1	Clear (reset) pointer (used to receive/send data)	0 – none 1 – clear data pointer	UploadImageData, DownloadImageData
4 ÷ 7	`P1_DISPLAY_SCHEME`	4	Determinate type of display drive scheme	See 6.3.4.2.1 Display Scheme	UploadImageData, DisplayUpdate

Table 6.5: C-APDU P1 parameter list

6.3.4.2 P2 Parameter List

Bit No.	Field name	Size [bits]	Description	Values	Commands
0 ÷ 2	P2_NUMBER_SLOT	3	Slot number (place in memory) to send, create or display image (only one available)	0 – slot No. 0	UploadImageData, DisplayUpdate
0 ÷ 2	RFU	3	Reserved for future use	0	All
3	P2_BORDER_CONTROL	1	Determines the ePaper display border colour	0 – white 1 – black	UploadImageData, DisplayUpdate
4 ÷ 5	P2_COMPRESSION_TYPE	2	Compression type	See 6.3.4.2.2 P2 Parameter P2_COMPRESSION_TYPE Values List	UploadImageData
4 ÷ 5	RFU	2	Reserved for future use	0	All
6 ÷ 7	P2_DATA_TYPE	2	Data type	0	UploadImageData

Table 6.6: C-APDU P2 parameter list

6.3.4.2.1 Display Scheme

P1 parameter `P1_DISPLAY_SCHEME` values list:

Value	Name	Number of stages	Description
0x0x	NO_DISPLAY	–	No display update
0x1x	DS_QUALITY_THROUGH_BLACK	4 stages	Quality through black
0x2x	DS_QUALITY_THROUGH_WHITE	4 stages	Quality through white
0x3x	DS_NORMAL_THROUGH_WHITE	2 stages	Normal through white
0x4x	DS_NORMAL_THROUGH_BLACK	2 stages	Normal through black
0x5x	DS_FAST_DIRECT	1 stage	Flashless
0x6x	DS_SCROLL	-	Scroll
0x7x	DS_NORMAL_THROUGH_WHITE_BLACK	3 stages	Normal through white/black
0x8x	DS_NORMAL_THROUGH_BLACK_WHITE	3 stages	Normal through black/white
0x9x	DS_FAST_DIRECT_INVERT	2 stages	Flashless inverted

Table 6.7: Display scheme parameter values

6.3.4.2.2 P2 Parameter P2_COMPRESSION_TYPE Values List

Value	Name	Description
0x00 << 4	COMPRESSION_NO	Data is not compressed
0x01 << 4	COMPRESSION_RLE7	RLE7 compression
0x02 << 4	RFU	Reserved for future use
0x03 << 4	RFU	Reserved for future use

Table 6.8: Compression type parameter values

`COMPRESION_RLE7` - image data in Type0 format (see 6.6 EPD File Format), compressed with RLE7 algorithm. Image header is also compressed.

RLE7 scheme encodes runs of multiple bits in a scan line and ignores byte and word boundaries. RLE7 encodes runs of one to 128 bits in length in a single-byte packet. The seven most significant bits contain the run count minus one, and the least significant bit contains the value of the bit run, either 0 or 1 (shown in table). A run longer than 128 pixels is split across several RLE-encoded packets.

Bit number	Name	Range
0	Value	0 or 1
1 ÷ 7	Counter	0 ÷ 127

Examples

- before compression:
  `0x00 0x00`
  after compression:
  `16 x 0` ⇒ `0x00 x 0x00` ⇒ `(0x10 - 1)<<1 | 0x00` ⇒ `0x1E | 0x00` ⇒ `0x1E`

- before compression:
  `0xFF 0xFF 0xFF 0xFF`
  after compression:
  `32 x 1` ⇒ `0x20 x 0x01` ⇒ `(0x20 - 1)<<1 | 0x01` ⇒ `0x3E | 0x01` ⇒ `0x3F`

- before compression:
  `0xFF 0xFF 0x00 0x0F 0xFF`
  after compression:
  `0x1F 0x16 0x17`

6.4 Command Instruction Description

6.4.1 UploadImageData

Command

INS	P1	P2	Lc	Data
0x20	P1	P2	Length	[Lc data bytes]

Description

The command uploads image data (in EPD file format) to PicoLabel image memory. The data needs to be divided into packets and transferred with multiple UploadImageData commands. In order to send the full image data, the user has to ensure to send it packet by packet.

Parameters

- P1: `P1_RESET_POINTER | P1_NO_ANSWER | P1_DISPLAY_SCHEME | P1_SAVE_TO_FLASH`
- P2: `P2_COMPRESSION_TYPE | P2_NUMBER_SLOT | P2_DATA_TYPE | P2_BORDER_CONTROL`

Data

Image file in EPD format, see 6.6 EPD File Format).

Response

- 2-byte status code (see 6.5 Response Status Codes)

GetImageData

Command

INS	P1	P2	Le
0xA0	P1	0x00	Length (max 0xFA / 250)

Description

Download image data from PicoLabel memory.

Parameters

- P1: `P1_RESET_POINTER`

Data

None.

Response

- Data (length specified by Le field, max 250 bytes),
- 2-byte status code (see 6.5 Response Status Codes)

6.4.2 DisplayUpdate

Command

INS	P1	P2
Transition	0x01	Si

Description

Refresh display using uploaded image.

- If data is uploaded, the new data will be displayed.
- If no data is sent, currently visible image will be refreshed (cleared and displayed again).
  - If `DS_FAST_DIRECT` is selected, there will be no visible effect on the display.

Parameters

- P1: `P1_NO_ANSWER | P1_DISPLAY_SCHEME`
- P2: `P2_BORDER_CONTROL`

Data

None.

Response

- 2-byte status code (see 6.5 Response Status Codes)

6.4.3 GetSystemInfo

Command

INS	P1	P2	Le
0x31	0x00	0x00	0x10

Description

The command returns information on system firmware version.

Parameters

- P1, P2: Constant values

Data

None.

Response

- `E0 E0 21 xx 00 00 00 00 36 01 04 02 00 00 00 00` (`xx` – firmware version number) – PLS-P27_v2.0
- `E0 E0 22 xx 00 00 00 00 36 01 04 02 00 00 00 00` (`xx` – firmware version number) – PLS-P27_v3.0
- 2-byte status code (see 6.5 Response Status Codes)

6.4.4 GetUID

Command

INS	P1	P2	Le
0x30	0x00	0x00	0x0C

Description

The command returns unique EPD MCU identification.

Parameters

- P1, P2: Constant values

Data

None.

Response

- UID data:
  - 0xC002 + 10-byte UID
- 2-byte status code (see 6.5 Response Status Codes)

6.4.5 GetSensorData

Command

INS	P1	P2	Le
0xE5	0x00	0x02	read_nb

Description

The command returns MCU supply voltage value.

Parameters

- P1, P2: Constant values

Data

None.

Response

- Raw data from voltage measurement – 2 bytes (MSB first), 10mV resolution
- 2-byte status code (see 6.5 Response Status Codes)

6.4.6 SetLED

Command

INS	P1	P2	Lc	Data
0x29	0x00	0x00	0x01	[Lc data bytes]

Description

The command allows controlling the LED pin output state which is stored in flash memory, therefore is preserved after a power cycle.

By default the blinking green LED indicates communication (from antenna alignment guidance to command execution), while the red LED indicates authorization status and is turned off after a successful authentication (either successful UnlockWithPassword command execution or successful authentication by Smart Card Controller).

Parameters

- P1, P2: Constant values

Data

Data byte 0bDxxxxxRG:

- D: reset LED functionality to default regardless of other bits (1 – reset, 0 – no reset)
- R: red LED (1 – on, 0 – off)
- G: green LED (1 – on, 0 – off)
- x: irrelevant bits

Response

- 2-byte status code (see 6.5 Response Status Codes)

6.4.7 SetPassword

Command

INS	P1	P2	Lc	Data
0x32	0x00	0x00	Length	[Lc data bytes]

Description

The command allows setting a password phrase used for password authentication. Executing the command without data results in disabling password security.

Parameters

- P1, P2: Constant values

Data

Password phrase – maximum 246 bytes, all values within byte range allowed.

Response

- 2-byte status code (see 6.5 Response Status Codes)

6.4.8 UnlockWithPassword

Command

INS	P1	P2	Lc	Data
0x33	0x00	0x00	Length	[Lc data bytes]

Description

The command must be used to authenticate and authorize connection with PicoLabel which requires a password.

Parameters

- P1, P2: Constant values

Data

Password phrase – maximum 246 bytes, all values within byte range allowed.

Response

- 2-byte status code (see 6.5 Response Status Codes)

6.5 Response Status Codes

Response SW Code	SW	Description
`EP_SW_NORMAL_PROCESSING`	`0x9000`	Command successfully executed
`EP_SW_INSTRUCTION_NOT_SUPPORTED`	`0x6D00`	Command under specified `INS` is not supported
`EP_SW_INVALID_LE`	`0x6C00`	Specified value for `Le` field is invalid
`EP_SW_WRONG_LENGTH`	`0x6B01`	Specified APDU has incorrect format, wrong length
`EP_SW_INVALID_LC`	`0x6700`	`Lc` field has incorrect value
`EP_SW_WRONG_PARAMETERS_P1P2`	`0x6A00`	Parameters specified for selected `INS` are not supported
`EP_SW_DATA_POINTER_OVER`	`0x5000`	Data pointer overflow
`EP_SW_WRONG_CRC`	`0x5003`	Error in CRC16 comparison
`EP_DISPLAY_NOT_AVAILABLE`	`0x9D54`	Display is disconnected or driver is broken
`EP_DISPLAY_BROKEN`	`0x9D55`	Display is broken
`EP_DISPLAY_HV_NOT_OK`	`0x9D56`	High voltage does not work

Table 6.9: Response status codes list

6.6 EPD File Format

EPD is a specific raster graphics image file format, accepted by the ePaper display driver. EPD file format was developed to maximize the decoding efficiency on the low-resource embedded platform. The EPD file comprises of two parts:

- Header
- Image data

Table below describes the various panels resolution and corresponding image data array sizes, as well as EPD files sizes.

Display size	Image resolution [px]	Image color depth [bit]	Header size [bytes]	Image data array size [bytes]	EPD file size [bytes]
2.7″	264×176	1	16	5,808	5,824
		2	16	11,616	11,632

Table 6.10: PicoLabel display data

6.6.1 Header

EPD file begins with a header. The header size is 16 bytes. The consecutive bytes are described in the table below:

Field name	Size	Possible values	Description
panel type	1 byte	0x32	Panel code 2.7″
X res	2 bytes	0x0108	264 px
Y res	2 bytes	0x00B0	176 px
color depth	1 byte	0x01 or 0x02	Image color depth (black and white) – either 1-bit or 2-bit
pixel data format	1 byte	0x00	Image pixel data format type 0
RFU	9 bytes	0x00	Reserved for future use

Table 6.11: Header values

Based on the information from the table above, here are complete header values for PicoLabel 2.7” v2.0/v3.0:

- 0x 32 01 08 00 B0 01 00 00 00 00 00 00 00 00 00 00 (1-bit color depth)
- 0x 32 01 08 00 B0 02 00 00 00 00 00 00 00 00 00 00 (2-bit color depth)

6.6.2 Image Data

Each byte of the image data encodes information on eight pixels (a single pixel is described by one bit of a single byte).

2-bit grey scale provides 4 colors. Value 0b00 corresponds to white colour, value 0b11 represents black color, while the values inbetween represents grey colors.

6.6.2.1 Pixel Data Format Type 0

Each byte of image data shall convey information on 8 consecutive pixels of the RAW image. Pixel data format type 0 for 2-bit images consists of combined data of each bit converted to this format.

Conversion Algorithm

The algorithm for conversion from standard RAW 4-bit data to EPD format is described below.

- Start with a byte array of image data which is already downsampled to 1-bit monochrome; each byte conveys information on 1 pixel.

1) Get a single row of 8 bytes (8 pixels):

Input byte No.:	0	1	2	3	4	5	6	7
Pixel value:	0	1	1	1	0	1	1	0

Table 6.12: Input data – 8 bytes

2) Merge the input byte values (numbering from 0 to 7) into one output byte, conveying information on 8 pixels

Input byte No.:	0	1	2	3	4	5	6	7
Pixel value:	0	1	1	1	0	1	1	0
Output byte value:	0x76 | 0b01110110							

Table 6.13: Output data – single byte

3) Go back to Step 1), getting the following row; repeat until all the bytes are processed

Sample Code

Below is sample Java code for image conversion:

```
static byte[] convertTo1bit_PixelFormatType0(byte[] picData, int w, int h)
{
   byte[] newRow = new byte[picData.length * 1 / 8];
   // join nibbles (so 1 byte is 8 pixels)
   int j = 0;
   for (int i = 0; i < picData.length; i += 8)
   {
          newRow[j] = (byte)( ((picData[i + 0] << 7) & 0x80) |
                              ((picData[i + 1] << 6) & 0x40) |
                              ((picData[i + 2] << 5) & 0x20) |
                              ((picData[i + 3] << 4) & 0x10) |
                              ((picData[i + 4] << 3) & 0x08) |
                              ((picData[i + 5] << 2) & 0x04) |
                              ((picData[i + 6] << 1) & 0x02) |
                              ((picData[i + 7])      & 0x01));
          j++;
   }
   return newRow;
}
```

7 PicoLabel Version Change Log

PicoLabel Version	Change Log
3.0	2-bit grey scale Cryptography-based security
2.0	2-bit grey scale Password security
1.1	GetUID command correction GetSystemInfo command update Display error return statuses added
1.0	1-bit grey scale

Table 7.1: Change log

8 Revision History

Document Revision	Release Date	Document Status	Supersedes
A	2017-04-27	Approved	–

Table 8.1: Revision history

Document Revision	Change Log
A	Initial version

Table 8.2: Change log

9 Legal Information

Copyright

PicoLabel and the PicoLabel logo are trademarks or registered trademarks of MpicoSys Solutions B.V. in the United States and in other countries.

The N-Mark is a trademark or registered trademark of NFC Forum, Inc. in the United States and in other countries.

HID, the HID logo, OMNIKEY and the OMNIKEY logo are trademarks or registered trademarks of HID Global in the U.S. and/or other countries.

All other trademarks, service marks and product or service names are trademarks or registered trademarks of their respective owners.

General

Information in this document is believed to be accurate and reliable. However, MpicoSys does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Right to make changes

MpicoSys reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use

MpicoSys products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of a MpicoSys product can reasonably be expected to result in personal injury, death or severe property or environmental damage. MpicoSys accepts no liability for inclusion and/or use of MpicoSys products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk. Product described in this document is intended for development purposes only and comes without any warranty. MpicoSys accepts no liability for inclusion and/or use of MpicoSys products in commercial products or applications and therefore such inclusion and/or use is at the customer's own risk.

Any software is provided "as is" and any expressed or implied warranties are disclaimed. In no event shall MpicoSys be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, arising in any way out of the use of the software.

Applications

Applications that are described herein for any of these products are for illustrative purposes only. MpicoSys makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Absolute maximum ratings

Stress above one or more limiting values of Absolute Maximum Ratings System (as defined in the Absolute Maximum Ratings System of IEC 60134) may cause permanent damage to the device. Limiting values are stress ratings only and operation of the device at these or any other conditions above those given in the Characteristics sections of this document is not implied. Exposure to limiting values for extended periods may affect device reliability.

Terms and conditions of sale

MpicoSys products are sold subject to the general terms and conditions of commercial sale, as published at http://www.mpicosys.com/terms, including those pertaining to warranty, intellectual property rights infringement and limitation of liability, unless explicitly otherwise agreed to in writing by MpicoSys. In case of any inconsistency or conflict between information in this document and such terms and conditions, the latter will prevail.

No offer to sell or license

Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

9.1 Third-party Licences

The following open-source components are used in PicoLabel 2.7″ Demo Kit:

9.1.1 TI Graphics Library

Software Name	Version	License Type	Delivered As	Obtained From	Modified by MpicoSys
Graphics Library Core	1.20.03.05	BSD	Source	Texas Instruments	Yes

9.1.2 DejaVu Font

Fonts are © Bitstream (see below). Glyphs imported from Arev fonts are © Tavmjung Bah (see below).

10 Contact Information

For additional information please visit www.mpicosys.com.

Please contact sales@mpicosys.com for commercial information.