Embedded Pico Systems

Developer's Guide

Timing Controller Solutions – Generation 2 for Pervasive Displays 4.41″ and 10.2″ Panels

TCM2-P441-231_v1.1, TC2-P441-231_v1.1 TCM2-P102-231_v1.1

Classification: **Public**

Document Revision: **B**

© MpicoSys – 2015
All rights reserved. Reproduction in whole or in part is prohibited without the written consent of the copyright owner.

Table of Contents

1 Introduction

E-paper Timing Controller Solutions Generation 2 provide timing controller (TCon) functionality for **Pervasive Displays**' large size panels (**4.41″ and 10.2″**). Solution for each of the panels provides identical functionality, command set and physical interface. Offered as a chip only (**Timing Controller – TC**) or as fully-assembled PCB module (**Timing Controller Module – TCM**), the solution allows a quick and easy integration with your host system, minimizing the cost and time-to-market.

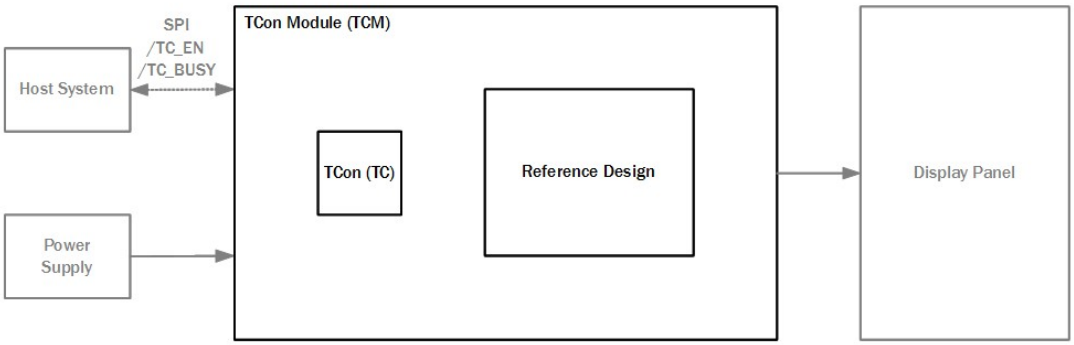


Figure 1.1: TCS block diagram

TCon (as well as TCM) can be connected to a host system via fast and reliable Serial Peripheral Interface (SPI). TCon is controlling both the source and gate drivers, composing waveforms required to generate high quality images on the display.

1.1 Ordering Information

Product Family: Timing Controller Solutions Generation 2 (TCS2)

Product Line: TCS2 for Pervasive Displays (TCS2-P)

TCS2-P441-231 Part Numbers

Timing Controller Module: **TCM2-P441-231_v1.1**

Timing Controller: **TC2-P441-231_v1.1**

TCS2-102-231 Part Numbers

Timing Controller Module: **TCM2-P102-231_v1.1**

Timing Controller: ***Not Available as TCon***

1.2 Compatibility Note

Timing Controller Solutions Generation 2 products supersede the respective Generation 1 products, being fully backwards compatible[1] in terms of host-side hardware and software.

The Generation 2 TCMs can be used as drop-in replacements for Generation 1 devices.

The TC reference designs have changed in Generation 2 and need to be adopted accordingly – this applies only to the users utilizing TCs integrated into their own designs.

1) Exception: GetSensorData command – see section 5.7.5.1

1.2.1 Superseded Products Part Numbers

Product Family: Timing Controller Solutions (TCS)

Product Line: TCS for Pervasive Displays (TCS-P)

TCS-P441 Part Numbers

Timing Controller Module: **TCM-P441-110_v1.1; TCM-P441-230_v1.1; TCM2-P441-231_v1.0**

Timing Controllers: **TCM-P441-110_v1.1; TC-P441-230_v1.1; TC2-P441-231_v1.0**

TCS-102-220 Part Numbers

Timing Controller Modules: **TCM-P102-220_v1.1; TCM2-P102-230_v1.0**

Timing Controllers: ***Not Available as TCon***

1.3 Supported Display Panels

MpicoSys TCon Solution	PDI Display Size	PDI Display Material (FPL)	PDI Display Part No.	PDI Display Resolution [px]	PDI Display Density [dpi]
TCS2-P441-231_v1.1	4.41″	Aurora Mb – v231	E1441CS021	400×300	113
TCS2-P102-231_v1.1	10.2″		EZ102CT011	1024×1280	160

Table 1.1: Supported display panels summary

1.4 Features

- Supporting the state-of-the-art PDI Aurora Mb panels
- SPI interface to host – slave device with additional /TC_EN and /TC_BUSY lines
- 1-bit color (black and white)
- Complete solution including:
  - Temperature compensation
  - Common electrode voltage compensation
  - All voltages needed for the display
- Internal image buffer retains content during system power down
- Backward-compatible with Generation 1

New In Generation 2

- Direct display update – no flashing during the image transition
- Partial image upload – no need to send the full image every time
- Reduced power consumption due to reactive implementation
- Multiple image slots
- Image data checksum calculation ensuring data integrity

1.5 Characteristics

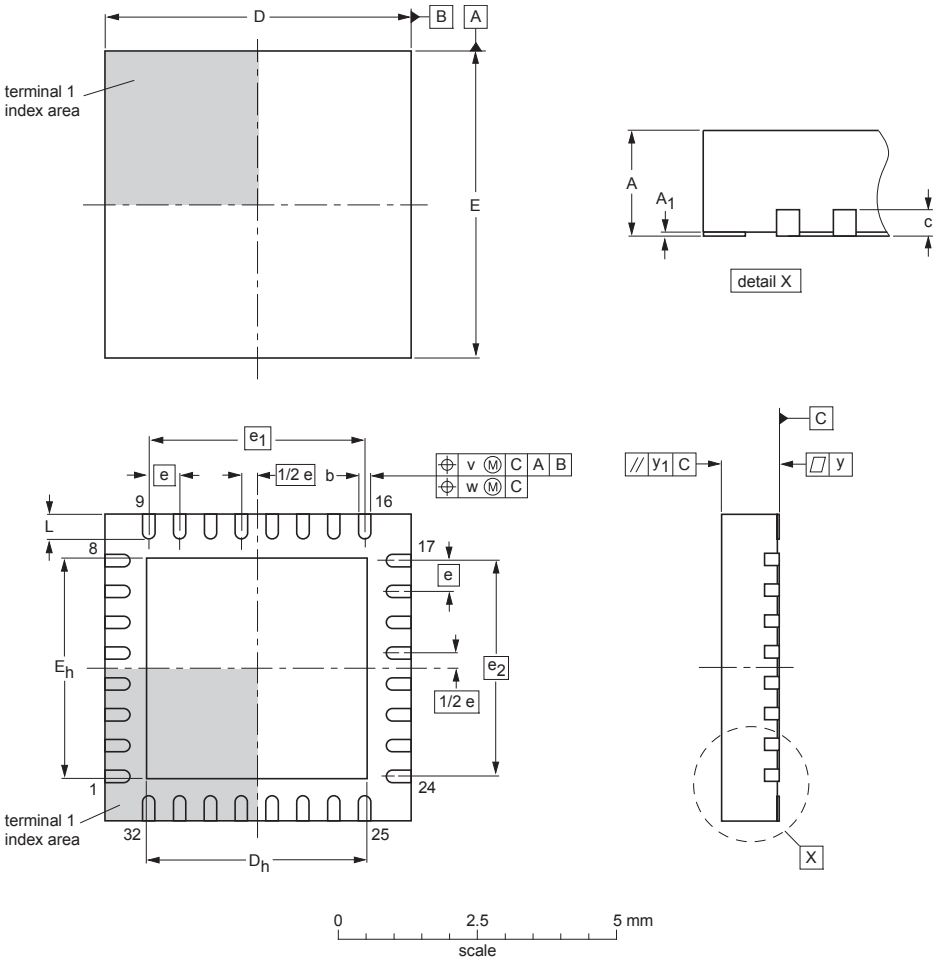
- From 2.7 to 3.3 V supply voltage
- From 0 to 50 °C operating temperature range

2 Outline

2.1 TCon

The information below applies to TC2-P441-231 product.

HVQFN33: plastic thermal enhanced very thin quad flat package; no leads; 32 terminals; body 5 x 5 x 0.85 mm



Unit[1]		A[1]	A_1	b	c	D[1]	D_h	E[1]	E_h	e	e_1	e_2	L	v	w	y	y_1
mm	max		0.05	0.30		5.1	3.75	5.1	3.75				0.5				
	nom	0.85			0.2					0.5	3.5	3.5		0.1	0.05	0.05	0.1
	min		0.00	0.18		4.9	3.45	4.9	3.45				0.3				

Table 2.1: Dimensions (mm are the original dimensions)[2]

1) Plastic or metal protrusions of 0.075 mm maximum per side are not included.
2) © NXP Semiconductors N.V. 2014. All rights reserved.

2.2 TCon Module

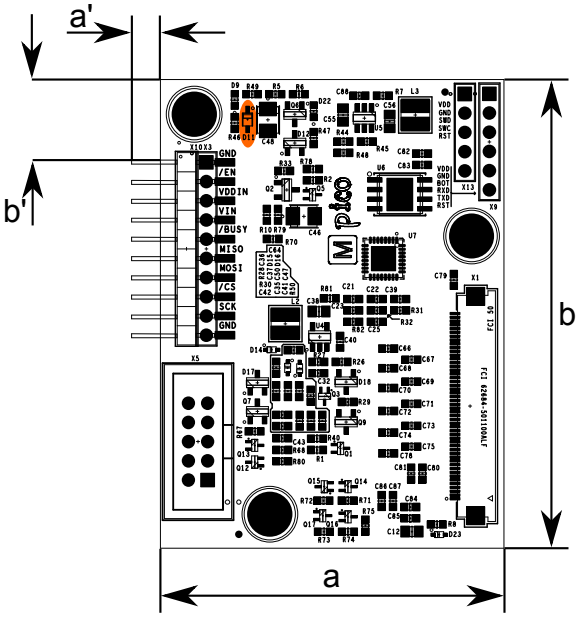


Figure 2.1: TCM2-P441 Outline

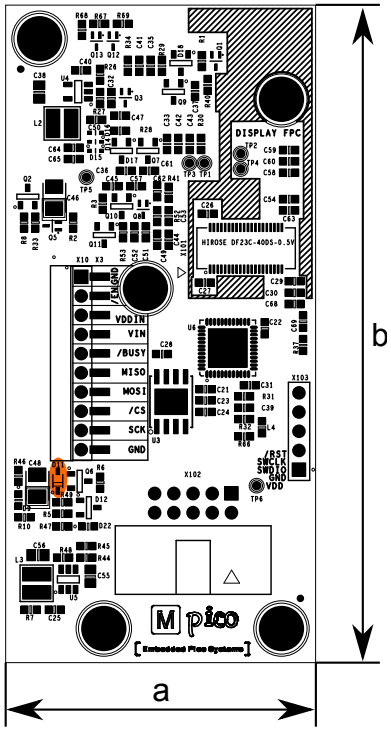


Figure 2.2: TCM2-P102 Outline

Module	Dimensions (mm)			
	a	a'	b	b'
TCM-P441-230	45.5	4.0	62.0	11.0
TCM-P102-220	41.0	–	87.0	–

Table 2.2: TCM2 physical dimensions

NOTE TCM2 features solder pads for overvoltage protection 3.6 V Zener diode (D11). The diode is by default not mounted to limit the TCM2 current consumption. If required, the diode can be mounted in the designated spot at the Customer's own account. The diode placement is marked orange on the Figure 2.1. It is recommended to use BZX384 3V6 diode. This will increase the average current consumption by 1 mA during all operations.

3 Electrical Characteristics

Unless specified otherwise, the values in this chapter are applicable to the whole product family (TCS2), that is both to TC2 and TCM2.

3.1 Absolute Maximum Ratings

TCS2-P441-231

Symbol	Description	Min	Typ	Max	Unit
VDDIN / VIN[1]	Supply voltage[2]	–0.5	–	4.0	V
V_I[2]	Logic input voltage	–0.5	–	4.6	V
T_{st}	Storage temperature	–20	–	+60	°C

Table 3.1: Absolute maximum ratings – TCS2-P441-231

TCS2-P102-231

Symbol	Description	Min	Typ	Max	Unit
VDDIN	Digital supply voltage	–0.5	–	4.0	V
VIN	Analog supply voltage	–0.5	–	6.0	V
V_I[2]	Logic input voltage[2]	–0.5	–	4.6	V
T_{st}	Storage temperature	–20	–	+60	°C

Table 3.2: Absolute maximum ratings – TCS2-P102-231

3.2 Operating Conditions

TCS2-P441-231

Symbol	Description	Min	Typ	Max	Unit
VDDIN / VIN[2]	Standard operating voltage[2]	2.7	3.0	3.3	V
T_{op}	Operating temperature	0	+21	+50	°C

Table 3.3: Typical operating conditions – TCS2-P441-231

TCS2-P102-231

Symbol	Description	Min	Typ	Max	Unit
VDDIN	Standard digital operating voltage	2.7	3.0	3.3	V
VIN	Standard analog operating voltage	2.0	3.0	5.5	V
T_{op}	Operating temperature	0	+21	+50	°C

Table 3.4: Typical operating conditions – TCS2-P102-231

3.3 TCM2 Supply Current Characteristics

Measurement Setup

Current consumption measured with Agilent 34411A Multimeter;

VDDIN shorted with VIN; range from 2.7 V to 3.3 V.

NOTE Values vary with ambient temperature, supply voltage, the displayed pattern, and the host controller settings.

1) Due to the fact TCS2 4.41″ v231 utilize internal display panel charge pump, the supply pins VDDIN and VIN are shorted with a 0R resistor.
2) Only valid when the VDDIN supply voltage is present.

TCS2-P441-231

Symbol	Description	Operation	Min	Max	Unit
IDD	Average current consumption	Display update – Quality	5.7	7.3	mA
		Display update – Flashless	6.9	8.7	mA
		Display update – Flashless-Inverted	7.3	12.8	mA
		Data reception on SPI	5.5	6.5	mA
		Disabled (/TN_EN inactive)	<1	<1	µA
E	Average energy consumption in room temperature	Display update – Quality	56	72	mJ
		Display update – Flashless	23.5	36	mJ
		Display update – Flashless-Inverted	45	79	mJ

Table 3.5: Supply current characteristics – TCS2-P441-231

TCS2-P102-231

Symbol	Description	Operation	Min	Max	Unit
IDD	Average current consumption	Display update – Quality	35	156	mA
		Display update – Flashless	33	347	mA
		Display update – Flashless-Inverted	38	172	mA
		Data reception on SPI	15.1	17	mA
		Disabled (/TN_EN inactive)	<1	<1	µA
E	Average energy consumption in room temperature	Display update – Quality	245	1168	mJ
		Display update – Flashless	101	313	mJ
		Display update – Flashless-Inverted	235	1976	mJ

Table 3.6: Supply current characteristics – TCS2-P102-231

Measurement Results Conditions

The below table describes conditions at which the results from tables above were achieved. *ESL* images are presented below the table. *Checkerboard* image is a 1 pixel by 1 pixel black and white checkerboard fulfilling the whole display area.

Measurement		Value	Power Supply (VDD = VIN) [V]	Image Used for Measurement	Ambient Temp. [°C]
Average current consumption	Display update	Min	3.3	Transition *ESL* to *ESL*	21
		Max	2.7	Transition *Checkerboard* to *Checkerboard*	21
	Data reception on SPI	Min	2.7	*ESL*	21
		Max	3.3	*Checkerboard*	21
Average energy consumption in room temperature	Display update	Min	2.7	ESL	21
		Max	3.3	*Checkerboard*	21

Table 3.7: Measurement results conditions

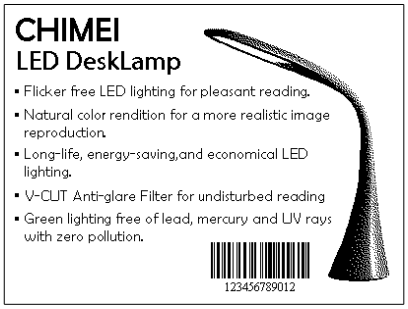


Figure 3.1: 4.41″ ESL image

Figure 3.2: 10.2″ ESL image

3.4 DC Characteristics

Symbol	Description	Min	Max	Unit
VIH	Input high level voltage	0.7×VDD	–	V
VIL	Input low level voltage	–	0.3×VDD	V
VOH	Output high level voltage	VDD-0.4	–	V
VOL	Output low level voltage	–	0.4	V

Table 3.8: Typical operating conditions

4 Display Refresh Time

The T_{amb} temperature value indicates the middle of the range.

Example: T_{amb}=20 indicates range between 17.5 and 22.5°C.

4.1 Default Transition

T_{amb} [°C]		0	5	10	15	20	25	30	35	40	45	50
Time [s]	TCS2-P441-231	9.7	7.3	5.5	4.2	3	2.7	2.4	2.1	2.1	2.1	2.1
	TCS2-P102-231	6.22	5.6	4.9	3.3	2.3	1.7	4.7	3.7	3.7	4.7	5.5

Table 4.1: Display refresh time versus ambient temperature – Default transition

4.2 Flashless Transition

T_{amb} [°C]		0	5	10	15	20	25	30	35	40	45	50
Time [s]	TCS2-P441-231	3.5	2.7	1.9	1.5	1.3	1.2	1.2	1.0	1.0	1.0	1.0
	TCS2-P102-231	2.1	1.7	1.5	1.4	1.0	1.0	1.0	0.9	0.9	0.9	0.7

Table 4.2: Display refresh time versus ambient temperature – Flashless transition

4.3 Flashless-Inverted Transition

T_{amb} [°C]		0	5	10	15	20	25	30	35	40	45	50
Time [s]	TCS2-P441-231	6.3	4.8	3.3	2.5	2.1	1.9	1.7	1.5	1.5	1.5	1.5
	TCS2-P102-231	2.8	2.5	2.3	1.5	1.5	1.5	1.5	1.2	1.2	1.2	1.0

Table 4.3: Display refresh time versus ambient temperature – Flashless-Inverted transition

5 TCS2 Hands-on

Unless specified otherwise, the values in this chapter are applicable to the whole product family (TCS2), that is both to TC2 and TCM2.

5.1 TC2 Integration

TC2 together with the reference schematic can be integrated with user's own host system. This enables the user to develop their own application utilizing e-paper technology.

Reference design is included in the Design Guide, distributed separately. Please contact sales@mpicosys.com for more information.[1]

1) TCon for 10.2″ display is not available

5.2 TCM2 Interconnection

Use the below described host connector to connect TCM2 to your host system. It is a 10-pin single-row 2.54 mm-pitch male header.

NOTE Forward slash "/" in front of the pin name indicates the signal is active low

Pin #	Pin Name	Remarks
1	GND	Supply ground
2	/TC_EN	TC2 enable
3	VDDIN[1]	Power supply for digital part[1]
4	VIN[1]	Power supply for analog part[1]
5	/TC_BUSY	Host interface busy output
6	TC_MISO	Host interface data output
7	TC_MOSI	Host interface data input
8	/TC_CS	Host interface chip select input
9	TC_SCK	Host interface clock input
10	GND	Supply ground

Table 5.1: TCM2 host connector

5.3 TCM2 Power On

Connect your power supply to the VDDIN and VIN pins.

VDDIN supply for digital part has to be supplied from a stable power supply, e.g. stabilized by a DC/DC converter or a low-dropout regulator (LDO).

In TCS2 4.41″ v231, VIN is internally connected to VDDIN, so this pin may be left not connected.

In TCS2 10.2″ v231, VIN can either be supplied directly from the battery (e.g. coin-cell) for improved efficiency, or can be shorted to VDDIN.

When connected to power supply, TCM2 is by default turned off to conserve energy. To switch it on, activate the /TC_EN signal.

5.4 Interface

Connection To Host

User's host system can communicate with TCon via Serial Peripheral Interface (SPI) with additional /TC_EN and /TC_BUSY line. TCon works as a SPI slave device. TCon power has to be supplied by the host system. The SPI supports 8-bit frames of data flowing from the master to the slave and from the slave to the master.

Signals

Inputs:

- /TC_EN – active low

1) Due to the fact TCS2 4.41″ v231 utilize internal display panel charge pump, the supply pins VDDIN and VIN are shorted with a 0R resistor.

- /TC_CS – active low
- TC_SCK
- TC_MOSI

Outputs:

- TC_MISO
- /TC_BUSY – active low

Startup and Initialization Sequence

TC2-P441-231:

The below timing diagram (Figure 5.1) represents the TCon startup and initialization sequence after power-up. The TCon is ready for communication after $T_{STARTUP}$ + T_{INIT} which is indicated by /TC_BUSY rising edge.

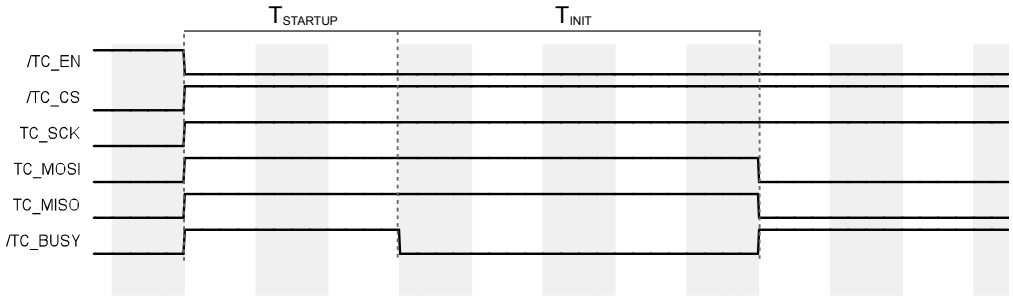


Figure 5.1: TC2-P441-231 initialization sequence

Time	Min	Max	Unit
$T_{STARTUP}$	1.5	1.5	ms
T_{INIT}	4.2	70	ms

Table 5.2: TC2-P441-231 startup and initialization times

TC2-P102-231:

The below timing diagram (Figure 5.2) represents the TCon startup and initialization sequence after power-up. The TCon is ready for communication after $T_{STARTUP+INIT}$ which is indicated by /TC_BUSY rising edge.

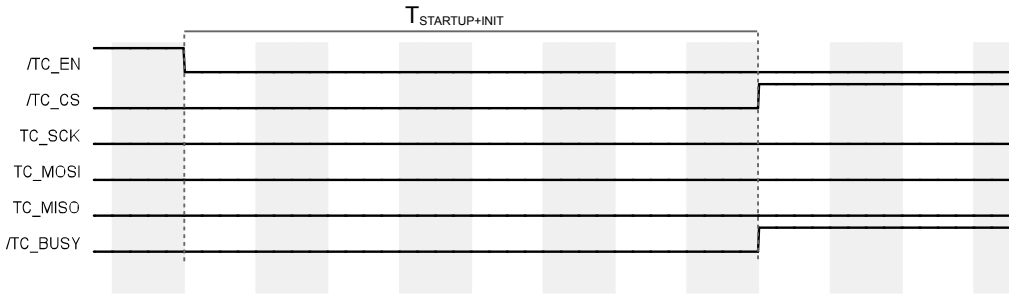


Figure 5.2: TC2-P102-231 initialization sequence

Time	Min	Max	Unit
$T_{STARTUP+INIT}$	2.5	22	ms

Table 5.3: TC2-P102-231 startup and initialization time

SPI Settings

- Bit rate – up to 12 MHz
  - Effective bit rate: 3 MHz
- Polarity – CPOL = 1; clock transition high-to-low on the leading edge and low-to-high on the trailing edge
- Phase – CPHA = 1; setup on the leading edge and sample on the trailing edge
- Bit order – MSB first
- Chip select polarity – active low

Reference SPI timing diagram below:

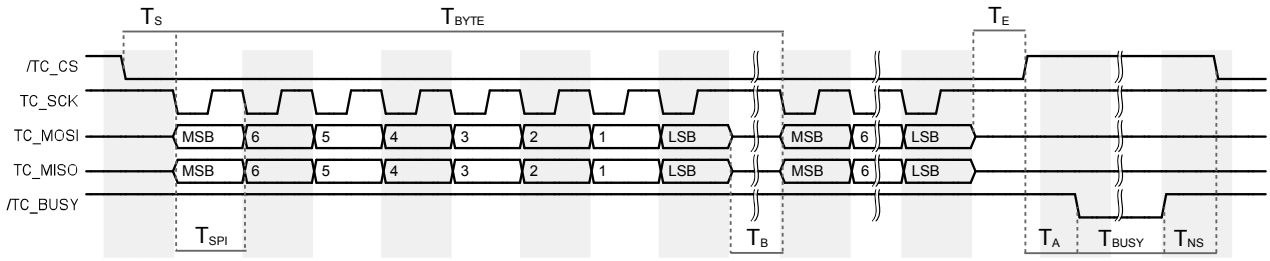


Figure 5.3: SPI timing diagram

Time	T_S	T_{BYTE}	T_{SPI}	T_B	T_E	T_A	T_{BUSY}	T_{NS}
Min.	2.67 µs	2.67 µs	83 ns[1]	0[2]	2.4 µs	–	255 µs	50 ns
Max.	–	–	550 ns	–	–	14.5 µs	–	–

Table 5.4: TCS2-P441-231_v1.1 SPI timing description

Time	T_S	T_{BYTE}	T_{SPI}	T_B	T_E	T_A	T_{BUSY}	T_{NS}
Min.	3.5 µs	2.67 µs	83 µs[3]	0[4]	0.4 µs	–	12.4 µs	0.4 µs
Max.	–	–	1 ms	–	–	3.8 µs	–	–

Table 5.5: TCS2-P102-231_v1.1 SPI timing description

Communication Flow

TCon is able to communicate to the host system if /TC_BUSY signal is inactive. To start communication, the /TC_CS line has to be activated by the host. Then the command data can be passed. There is no timeout during the communication, so delays between the consecutive bytes are allowed. Only when /TC_CS line is deactivated, is the command interpreted by the TCon.

1) Minimum T_{SPI} value reflects the maximum supported bit rate of 12 MHz. In this case T_B has to be greater than or equal to 2.0 µs.
2) T_{SPI} value can equal 0 if frequency is lower than or equal to 3 MHz.
3) Minimum T_{SPI} value reflects the maximum supported bit rate of 12 MHz. In this case T_B has to be greater than or equal to 2.0 µs.
4) T_{SPI} value can equal 0 if frequency is lower than or equal to 3 MHz.

After passing the command, it is being interpreted and executed by the TCon. The time of execution is indicated by /TC_BUSY signal active. During this time, the TCon does not accept any new commands.

5.5 Framebuffer Slots

TCons feature image (framebuffers) slots for storing image data. The main purpose of enabling multiple slots is to increase the product use cycles, limited by the flash memory endurance. With multiple slots, the newly uploaded images (when addressed with slot number 0) are cycling through the slots, effectively reducing the necessity of constantly erasing the same memory sectors. The number of the available framebuffer slots is indicated in the Table 5.6 below. The images are stored in non-volatile memory, thus are retained when the system is not powered.

TCon Solution Type	TCon Solution Type
TCS2-P441-231	16
TCS2-P102-231	3

Table 5.6: Number of available image slots

When addressed with the default slot number 0, the TCon automatically assigns the slot number so that the new data is always stored in the oldest used slot, and correctly referred to in case of display update command. In this case the host does not need to pay any attention to in which slot the data is stored.

Framebuffer slots can also be addressed directly by the slot number (1, 2, 3, ...) or relatively by the last displayed: *-1* addresses the currently displayed slot, *-2* addresses the previously displayed slot, and so on. In this case the host needs to keep information on to which slot the data has been written and which slot is available for upload. It is important to note that the slot containing the data of the image currently displayed on the display cannot be overwritten.

5.5.1 Current Slot Restriction

The slot that has been displayed latest, cannot be modified. This is because the last displayed image data is needed to properly refresh the display, minimizing the ghosting phenomenon.

Thus all the commands attempting to modify the last displayed slot, will respond with 0x6981 error status code.

5.6 Use Cases

5.6.1 Partial Image Upload and Display

The partial upload and display allows to:

- Define any rectangular area within the display (Region of Interest – ROI)
- Fill the area with specific data, or
- Fill the area with data from another, previously populated image slot, or
- Fill the area with uniform data (black, white, or pattern)

The commands can be run in a sequence to generate the image, and then display the final result.

NOTE The generated image can be displayed either with full-quality display refresh (with black and white flashes), or with flashless update. Using the flashless update will ensure that only the changed part of the display is refreshed – see 5.7.2.1 DisplayUpdate.

Each image building sequence needs to be proceeded by memory slot erasing.

With the use of partial update commands described below, user can e.g. upload part of an image to the buffer, copy some image from another slot and fill the rest of the image with black, white or with given pattern.

Example use case:

1) Erase chosen slot (see 5.7.1.5 ImageEraseFrameBuffer) – erase is mandatory before any partial upload operation;
2) Evoke 5.7.1.6 UploadImageSetROI command to define the area (e.g. 100x200 px in upper right corner) where the changes in framebuffer should apply;
3) Evoke 5.7.1.1 UploadImageData commands to fill the above defined area with image data;
4) Evoke 5.7.1.6 UploadImageSetROI command to define the next area (areas cannot intersect!);
5) Use 5.7.1.8 UploadImageCopySlots command to copy image data from another slot to the defined area;
6) Use 5.7.1.6 UploadImageSetROI command to define next area (areas cannot intersect!);
7) Use 5.7.1.7 UploadImageFixVal to e.g. fill the defined area with white color;
8) Evoke 5.7.2.1 DisplayUpdate command to display created image;
9) The sequence of commands (2-3, 4-5, 6-7) can be run in any order.

5.7 Command Description

Command Format

Each command is built up from 3 to 255 bytes. The command is divided into six fields.

The first three fields are used in each command:

- *INS* – command group specific
- *P1* – parameter
- *P2* – parameter

whereas the next three fields are only used by some particular commands:

- *Lc* – number of bytes in *Data* field
- *Data* – bytes forming command data; number of bytes determined by *Lc*
- *Le* – number of bytes of expected response

Response

Upon each command, TCon responds with a 2-byte command status code. The command status code is not included in the *Le* (expected response length).

Possible status codes list:

Status code	Status mnemonic	Description
0x9000	EP_SW_NORMAL_PROCESSING	Command successfully executed
0x6581	EP_SW_MEMORY_FAILURE	An error occurred while interfacing external memory
0x6700	EP_SW_WRONG_LENGTH	Incorrect length (invalid *Lc* value or command too short or too long)
0x6981	EP_FRAMEBUFFER_SLOT_NOT_AVAILABLE	Frambuffer slot number is either the last displayed slot, or the number is out of range
0x6A00	EP_SW_WRONG_PARAMETERS_P1P2	Invalid *P1* or *P2* field
0x6A84	EP_FRAMEBUFFER_SLOT_OVERRUN	Framebuffer slot overridden
0x6C00	EP_SW_INVALID_LE	Specified value for *Le* field is invalid
0x6D00	EP_SW_INSTRUCTION_NOT_SUPPORTED	Command not supported
0x6F00	EP_SW_GENERAL_ERROR	Internal TCon reset triggered due to abnormal behavior; the command was not executed properly

Figure 5.4: List of response codes

If a command returns specific data, the status code is attached to the end of the data.

Data Readout

During each SPI clock cycle, a full-duplex data transmission takes place: the host sends a bit on the MOSI line, and the TCon sends a bit on the MISO line at the same time.

Thus, the command status should be read after the command is executed. To read the command status, the host should send the expected number of 0x00 bytes to TCon. The amount of bytes to be sent is dependent on the type of a command:

- If a command does not use the *Le* field, it will return only the two-byte status code; thus only two bytes should be sent by the host
- When *Le* field is used and set to 0x00, the response length is not determined; then the response should be read until 0x00 is encountered, indicating the response termination, and two additional bytes should be sent to acquire the command status
- When *Le* field is set to a value other than 0x00, the response length is determined by the value at *Le* field. The host should send the number of bytes indicated by the *Le* field, and two additional bytes to acquire the command status

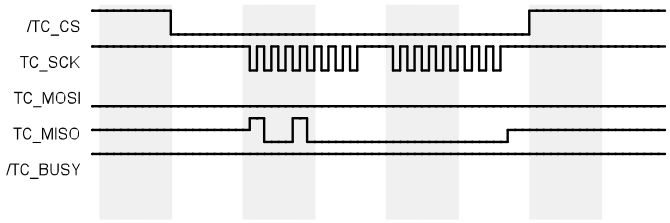


Figure 5.5: Example readout - 0x9000 response

5.7.1 Image Data Commands

This group of commands, handles the process of data transfer between the host and the TCon.

5.7.1.1 UploadImageData

Command

INS	P1	P2	Lc	Data
0x20	0x01	Si	Data packet size (max 0xFA)	[Lc Data bytes]

Description

The command uploads image data (in EPD file format) to TCon 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 make sure to send it packet by packet.

While writing to the TCon internal memory, the TCon data pointer will be internally increased by the size of the current packet, until reaching maximum of slot memory. When the slot memory size is exceeded, EP_FRAMEBUFFER_SLOT_OVERRUN status code will be returned as response.

Parameters

- P2: Si – framebuffer slot number (see 5.5 Framebuffer Slots)

Data

Image file in EPD format, see 6 EPD File Format). Maximum packet size is 251 bytes (as maximum command size is 255 bytes.)

NOTE If this command is used in partial update (i.e. following the UploadImageSetROI command), the data **should not** contain the EPD header, and should be encoded in EPD format type 0.

Response

- 2-byte status code

5.7.1.2 GetImageData

Command

INS	P1	P2	Le
0xA0	0x00	Si	Length

Description

Get image data from specified slot. The data is divided into packets – similarly to UploadImageData command. To get the full image data, multiple GetImageData commands need to be sent, until the full image is received.

Each time the command is called, the TCon data pointer is increased by the size of the read data packet, until reaching the maximum of slot size.

Parameters

- P1: Constant value
- P2: Si – framebuffer slot number (see 5.5 Framebuffer Slots)

Data

Image file in EPD format (see 6 EPD File Format.) Maximum packet size is 251 bytes (as maximum command size is 255 bytes.)

Response

- 2-byte status code

5.7.1.3 GetChecksum

Command

INS	P1	P2	Le
0x2E	0x01	Si	0x02

Description

Get 16-bit checksum of an image stored in the TCon memory.

NOTE TCS2-P441-231_v1.1 supports two types of Pixel Data Format Type – 0 and 1. Type 2 data is converted to Type 0 during data upload, thus checksum is calculated on Type 0 data, even if Type 2 was uploaded. Header data is not converted, thus remains the same as uploaded.

Initial checksum value is 0x6363. Checksum is calculated on raw image data (image data deconversion performed on chip.)

Checksum implementation:

```
uint16_t crc16_add(uint8_t byte, uint16_t acc)
{
   acc ^= byte;
   acc  = (acc >> 8) | (acc << 8);
   acc ^= (acc & 0xff00) << 4;
   acc ^= (acc >> 8) >> 4;
   acc ^= (acc & 0xff00) >> 5;
   acc = acc;
   return acc;
}
```

Parameters

- P1: Constant value
- P2: Si – framebuffer slot number (see 5.5 Framebuffer Slots)

Response

- [2 bytes: (0xHH, 0xLL), where 0xHH is the upper byte, and 0xLL is the lower byte of the 16 bit checksum.] + 0x9000 status code, or
- 2-byte error status code

5.7.1.4 ResetDataPointer

Command

INS	P1	P2
0x20	0x0D	0x00

Description

The command resets data pointer for Upload Image Data command.

NOTE Data pointer is automatically reset when TCon is enabled by /TC_EN activation

Response

- 2-byte status code

5.7.1.5 ImageEraseFrameBuffer

Command

INS	P1	P2
0x20	0x0E	Si

Description

Command resets data pointer to the beginning of the chosen image slot (similarly to ResetDataPointer command) and erases the entire image slot. The erased slot is filled with 0xFF, which if displayed is shown as a full-black image.

Parameters

- P1: Constant value
- P2: Si – framebuffer slot number (see 5.5 Framebuffer Slots)

Response

- 2-byte status code

5.7.1.6 UploadImageSetROI

Command

INS	P1	P2	Lc	Data
0x20	0x0A	0x00 or Si	0x08	[ROI data]

Description

Command sets region of interest for image upload. The framebuffer pointer is set to the beginning of ROI buffer: after the command each UploadImageData command will fill framebuffer in ROI region only.

- EPD image data header should not be sent in image data after ImageUploadSetROI command
- Set ROI region is valid until ResetDataPointer, ImageEraseFrameBuffer, DisplayUpdate or next ImageUploadSetROI command
- X coordinate value for ROI must be dividable by 8
- At startup and after ResetDataPointer, ImageEraseFrameBuffer and DisplayUpdate command ROI is not set and points to the whole framebuffer

Parameters

- P2: Constant value in TCS2-P441, or Si – framebuffer slot number (see 5.5 Framebuffer Slots) in TCS2-P102

Data

ROI_data: four 16-bit (MSB first) values that define ROI area: Xmin (inclusive), Xmax (exclusive) (from 0 to 1024), Ymin (inclusive), Ymax (exclusive) (from 0 to 1280). Max has to be greater than min value.

Example: [01 C0 02 40 01 EC 03 14] defines ROI: Xmin = 448, Xmax = 576, Ymin = 492, Ymax = 788, which can fit 128x296 px image.

Response

- 2-byte status code

5.7.1.7 UploadImageFixVal

Command

INS	P1	P2	Lc	Data
0x20	0x0B	Si	Length	[Pattern]

Description

Command copies and replicates the given data buffer (max 250 bytes specified by Data field, without EPD header) to frame buffer slot Si area specified by ROI which was set by ImageUploadSetROI. Can be used to clear framebuffer to white (Data = 0x00), black (Data = 0xFF) or pattern defined by Data. Framebuffer slot needs to be erased prior to partial image upload commands. Example use case of this command is barcode rendering.

Parameters

- P1: Constant value
- P2: Si – framebuffer slot number (see 5.5 Framebuffer Slots)

Data

Pattern data in EPD format type 0 (see 6 EPD File Format), without the EPD header. Maximum pattern size is 250 bytes.

Response

- 2-byte status code

5.7.1.8 UploadImageCopySlots

Command

INS	P1	P2	Lc	Data
0x20	0x0C	Si	0x01	Si_Source

Description

Command copies image from a selected slot to framebuffer. If ROI was specified prior to this command, only this area is copied. ROI can be set by ImageUploadSetROI command. Otherwise the whole slot is copied.

Parameters

- P1: Constant value
- P2: Si – destination framebuffer slot number (see 5.5 Framebuffer Slots)

Data

- Si_Source – source framebuffer slot number (see 5.5 Framebuffer Slots)

Response

- 2-byte status code

5.7.2 Display Control Commands

5.7.2.1 DisplayUpdate

Command

INS	P1	P2	Lc	Data
Transition	0x01	Si	Length	Temperature (optional)

Description

The command starts the display refresh sequence, displaying the current content of the image memory, using the display transition sequence chosen with the INS value.

- If data was uploaded, the new data is going to be displayed
- If no data was sent, currently visible image will be refreshed (cleared and displayed again)

Parameters

- INS – Specifies the display refresh transition sequence according to the table below:

Transition	Name	Description
0x24, 0x82 (interchangeably)	Default	Default transition sequence – with black-white-black screen flashing. Offers the best image quality.
0x85	Flashless	Direct image to image transition (without the blank black or white screen in between.) The fastest and the most energy-efficient transition, at the cost of image quality.
0x86	Flashless-Inverted	Transition from the current image to the inverted new image, followed by the new image (without the blank black or white screen in between). Compromise between the Default and the Flashless – both in terms of energy consumption and image quality.

- P2: Si – slot number (see 5.5 Framebuffer Slots)

Data

Temperature (optional parameter) – temperature overriding the internal temperature measurement used for display driving temperature compensation. Leave this field empty to use the internal temperature measurement.

Temperature in degrees Celsius, encoded in U2 code (values spanning from -128 to 127.)

WARNING Populating this field alters the display driving time, which will compromise the image quality. It is recommended to leave this field empty.

Response

- 2-byte status code

5.7.3 Device Info Commands

This group of commands, starting with INS = 0x30 byte, manages the acquirement of hardware information from TCon.

5.7.3.1 GetDeviceInfo

Command

INS	P1	P2	Le
0x30	0x01	0x01	0x00

Description

The command returns information on system hardware. String data is specific for the particular device type and is constant for the same type of devices if no hardware differences occur.

Response

- [String: "MpicoSys TC2-P441-231_v1.1" terminated by 0x00 byte] + 0x9000 status code in case of TC2-P441-231_v1.0, or
- [String: "MpicoSys TC2-P102-231_v1.1" terminated by 0x00 byte] + 0x9000 status code in case of TC2-P102-231_v1.0, or
- 2-byte error status code

5.7.3.2 GetDeviceId

Command

INS	P1	P2	Le
0x30	0x02	0x01	0x14

Description

The command returns unique device ID number.

Response

- [20 bytes of data] + 0x9000 status code, or
- 2-byte error status code

5.7.4 System Info Commands

This group of commands, starting with INS = 0x31 byte, deals with acquirement of firmware information from TCon.

5.7.4.1 GetSystemInfo

Command

INS	P1	P2	Le
0x31	0x01	0x01	0x00

Description

The command returns information on system firmware.

Response

- [String: "MpicoSys TC2-P441-231_fB_BIN" terminated by 0x00 byte] + 0x9000 status code in case of TC2-P441-231_v1.1, or
- [String: "MpicoSys TC2-P102-231_fB_BIN" terminated by 0x00 byte] + 0x9000 status code in case of TC2-P102-231_v1.1, or
- 2-byte error status code

5.7.4.2 GetSystemVersionCode

Command

INS	P1	P2	Le
0x31	0x02	0x01	0x10

Description

The command returns information on system version.

Response

- 0x D0 AE 01 00 00 00 00 00 33 10 04 00 00 00 00 00 + 0x9000 status code in case of TC2-P441-231_v1.1, or
- 0x D0 AF 01 00 00 00 00 00 3D 20 04 00 00 00 00 00 + 0x9000 status code in case of TC2-P102-231_v1.1, or
- 2-byte error status code

5.7.5 Sensor Data Commands

5.7.5.1 GetSensorData

Command

INS	P1	P2	Le
0xE5	0x01	0x00	0x02

Description

The command returns the temperature value measured by the TCon temperature sensor. The sensor is built in the TCM2 board and is included in the TCon reference design. The measurement is based on a NCP18WB473E03RB thermistor and 8-bit ADC.

TCS2-P441-231_v1.0 returns temperature in degrees Celsius, U2 encoded, whereas TCS2-P102-231_v1.0 returns raw ADC data, that needs to be mapped to temperature value according to the chart on Figure 5.6.

NOTE The temperature readout functionality will be modified in TCS2 version v1.1. Bot the 4.41 and 10.2 solutions will return raw ADC data (for backwards compatibility with Gen1), and value in degrees Celsius if the command is sent with different parameter.

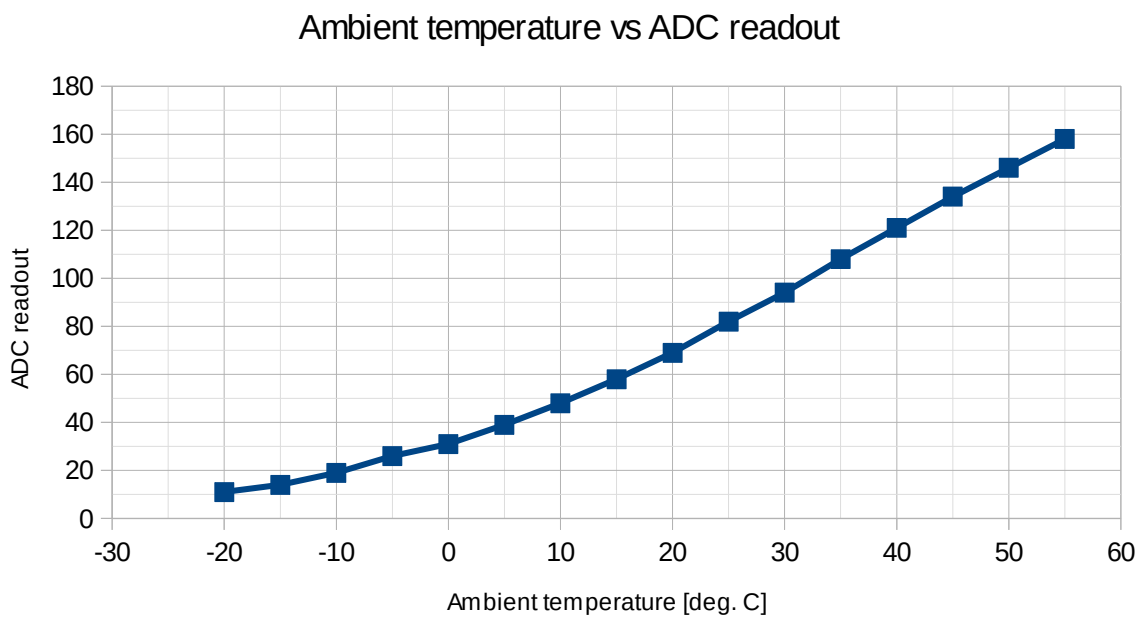


Figure 5.6: Ambient temperature versus ADC readout chart

Temperature [°C]	ADC value [dec]	ADC value [hex]
-20	11	0B
-15	14	0E
-10	19	13
-5	26	1A
0	31	1F
5	39	27
10	48	30
15	58	3A

Temperature [°C]	ADC value [dec]	ADC value [hex]
20	69	45
25	82	52
30	94	5E
35	108	6C
40	121	79
45	134	86
50	146	92
55	158	9E

Response

- [2 bytes of temperature measurement in deg. C, u2 encoded[1]] + 0x9000 status code, or
- [2 bytes of RAW sensor data[2]] + 0x9000 status code, or
- 2-byte error status code

6 EPD File Format

EPD is a specific raster graphics image file format, accepted by TCon. EPD file format was developed to maximize the decoding efficiency on the target platform. The EPD file comprises of two parts:

- Header
- Image data

1) Valid for TCS2-P441-231
2) Valid for TCS2-P102-231

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

Panel size	Image resolution [px]	Image color depth [bit]	Header size [bytes]	Image data array size [bytes]	EPD file size [bytes]
4.41″	400×300	1	16	15,000	15,016
7.4″	480×800			48,000	48,016
10.2″	1024×1280			163,840	163,856

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	0x33	Panel code	4.41″
		0x3D		10.2″
X res	2 bytes	0x0190	400 px	
		0x0400	1024 px	
Y res	2 bytes	0x012C	300 px	
		0x0500	1280 px	
color depth	1 byte	0x01	Image color depth – 1-bit (black and white)	
pixel data format	1 byte	0x00	Image pixel data format type 0	
		0x02	Image pixel data format type 2	
		0x04	Image pixel data format type 4	
RFU	9 bytes	0x00	Reserved for future use	

Based on the information from the table above, here are complete header values depending on the panel size:

- TC2-P441-231: 0x 33 01 90 01 2C 01 00 00 00 00 00 00 00 00 00 00
  0x 33 01 90 01 2C 01 02 00 00 00 00 00 00 00 00 00
- TC2-P102-231 0x 3D 04 00 05 00 01 00 00 00 00 00 00 00 00 00 00

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).

1-bit gray scale provides 2 colors. Bit value 0 corresponds to white color while value 1 represents black color.

6.2.1 Pixel Data Format Type 0

This format is used in TC2-P441-231 and TC2-P102-231. Each byte of image data shall convey information on 8 consecutive pixels of the RAW image.

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 6.1: 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 6.2: 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;
}
```

6.2.2 Pixel Data Format Type 2

This format is used in TC2-P441-231, for backwards compatibility with Generation 1.

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 6.3: Input data – 8 bytes

2) Assign the input byte values (numbering from 0 to 7) to the output byte, conveying information on 8 pixels, as follows:
   - Input byte 0: assign to output byte bit 0
   - Input byte 1: assign to output byte bit 2
   - Input byte 2: assign to output byte bit 4
   - Input byte 3: assign to output byte bit 6
   - Input byte 4: assign to output byte bit 1
   - Input byte 5: assign to output byte bit 3
   - Input byte 6: assign to output byte bit 5
   - Input byte 7: assign to output byte bit 7

Output byte bit No.:	0	1	2	3	4	5	6	7
Input byte No.:	0	4	1	5	2	6	3	7
Pixel value:	0	0	1	1	1	1	1	0
Output byte value:	0x3E | 0b00111110							

Table 6.4: 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_PixelFormatType2(byte[] picData, int w, int h)
{
   byte[] newRow = new byte[picData.length * 1 / 8];
   // join nibbles (so 1 byte is 8 pixels) and interlace at the same time
   int j = 0;
   for (int i = 0; i < picData.length; i += 8)
   {
          newRow[j] = (byte)( ((picData[i + 0] << 7) & 0x80) |
                              ((picData[i + 4] << 6) & 0x40) |
                              ((picData[i + 1] << 5) & 0x20) |
                              ((picData[i + 5] << 4) & 0x10) |
                              ((picData[i + 2] << 3) & 0x08) |
                              ((picData[i + 6] << 2) & 0x04) |
                              ((picData[i + 3] << 1) & 0x02) |
                              ((picData[i + 7])      & 0x01));
          j++;
   }
   return newRow;
}
```

7 Revision History

Document Revision	Release Date	Document Status	Supersedes
B	2015-11-23	Approved	A
A	2015-07-17	Approved	All the draft versions

Table 7.1: Revision history

Document Revision	Change Log
B	Update for v1.1 versions
A	Initial version

Table 7.2: Change log

8 Legal Information

Draft

The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. MpicoSys does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet

A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local MpicoSys sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

Data sheet

A document intended to give a full description of the product details that a customer needs to implement the product in their design.

8.1 Disclaimers

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 Contact Information

If you have any technical questions, please send an email to support@mpicosys.com.

Please contact sales@mpicosys.com for commercial information.