

Ultra Low Power Capacitive Touch Sensor Interface IC

Description

The EM6420 is an ultra low power Touch Sensor Interface IC able to scan sequentially up to 16 capacitive sensors. The device parameters (number of used sensors, sensors scan frequency, sensors sensitivity level, IRQ condition) are configurable either from a host microcontroller through a communication port or through configuration inputs.

Recognised touch inputs will be signaled with an active edge at the IRQ pad and data are ready to be read through the communication port by the host MCU. Conditions for the IRQ to get active are configurable : at the end of every scan, at the end of a scan if at least one sensor is active or at the end of a scan if the sensors state has changed.

The EM6420 can also detect the most active sensor in applications where sensors are tightly spaced. It compares relative levels among sensors and selects the sensor with the largest signal strength.

To increase the number of sensors >16, use several EM6420 in parallel.

Depending on the selected supply voltage range, 3 or 4 decoupling capacitors are required for the entire functionality of the EM6420 from -40 to + 85°C.

Features

- Up to 16 analogue sensor inputs
- User selectable communication interfaces: 4-wire SPI, I²C, 4-bit parallel interface and 8-bit direct output
- User-selectable active edge IRQ output signal
- Active high enable input
- No software development and tuning required
- Development tools and documentations available
- Complete touch module available: IC + electrodes design on various non-conductive substrates

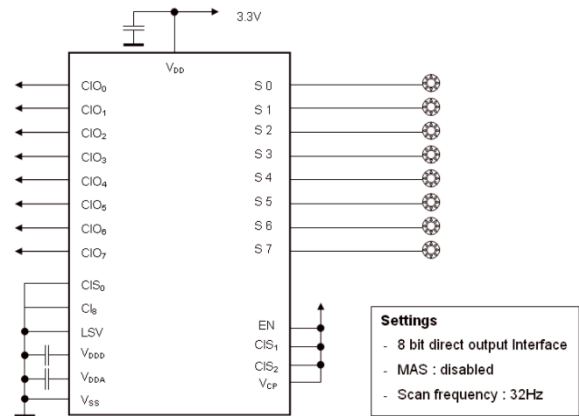
Electrical Characteristics

- Supply voltage 1.2 V to 2.0 V or 2.2 to 3.6 V
- Power consumption 8.0 µA @ 3.0 V (14.5 µA @1.5 V) for 16 sensors scanned at 8 Hz
Low Power Mode
- Ultra Low Power Mode 2.0 µA @ 3.0 V (5.0 µA @1.5 V) for 16 sensors scanned at 8 Hz
- Nominal sensor capacitance 3 to 31 pF
- Sensors scan frequency 1 Hz to 128 Hz *frequency depending on number of sensors
- COM clock frequency up to 400 kHz

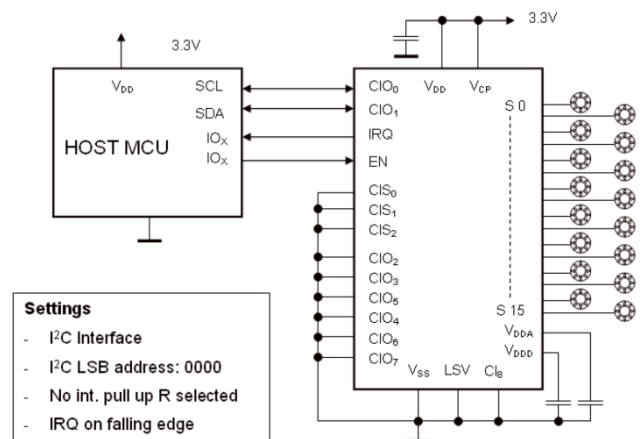
Availability

- Naked die
- Gold bump
- SMT package MLF40

Typical Application in Stand Alone Mode and 8 Sensing Inputs



Typical Application with a Host MCU and 16 Sensing Inputs



Design Considerations

The EM6420 is well suited for battery and mains powered applications where the following features are important :

- Tamper proof applications
- Nice and clean designs
- Touch function to avoid buttons and keys
- Slider functions
- Hygienic issues, cleaning aspects
- Waterproof designs

Applications

- Mobile phones, cordless phones
- PDA, keyboards
- White & brown goods
- Toys
- Lighting - Sliders for dimming



TABLE OF CONTENTS

1.	PRELIMINARIES	4
1.1	Reference.....	4
1.2	Conventions	4
2.	GENERAL DESCRIPTION	5
3.	FEATURES	6
3.1	Key elements.....	6
3.2	Power Supply	6
3.3	Interfaces	6
3.4	Development Tools	6
3.5	Touch modules based on EM6420 IC	6
4.	BLOCK DIAGRAM	7
5.	PAD DESCRIPTION	7
6.	ELECTRICAL SPECIFICATIONS	9
6.1	Absolute Maximum Ratings.....	9
6.2	Handling Procedures.....	9
6.3	Supply Voltage Configurations	9
6.4	Standard Operating Conditions	11
6.5	DC Characteristics – Power Supply	12
6.6	POR	12
6.7	Touch Screen Interface.....	13
6.8	Input pads CIS _x , CI ₈ and LSV	13
6.9	Input pad En	13
6.10	Output pad IRQ.....	14
6.11	Bidirectional pads CIO ₂ ... CIO ₇	14
6.12	Bidirectional pads CIO ₀ and CIO ₁	15
7.	TIMING SPECIFICATIONS	16
7.1	Standard Operating Conditions	16
7.2	Communication Interface	16
7.3	8-bit Direct Output Interface	16
7.4	Slave I2C Interface.....	16
7.5	Slave SPI Interface.....	17
7.6	Slave 4-bit Parallel Interface	19
8.	EM6420 TO HOST CONTROLLER COMMUNICATION	20
8.1	Introduction	20
8.2	EM6420 Communication Interfaces	21
8.2.1	Slave I2C Interface.....	21
8.2.2	Slave SPI Interface.....	22
8.2.3	Slave 4-bit Parallel Interface.....	24
8.2.4	8-bit Direct Output Interface	25
8.2.5	Communication interface initialization	27
8.3	EM6420 Commands.....	28
8.3.1	Command startTS	29
8.3.2	Command stopTS	29
8.3.3	Command setTSMODE.....	29
8.3.4	Command selectBaseSettings	30
8.3.5	Command selectAltSettings	30
8.3.6	Command setBaseScanFreq	31
8.3.7	Command setAltScanFreq	31
8.3.8	Command setBaseHiSensNb.....	32
8.3.9	Command setAltHiSensNb.....	32
8.3.10	Command setBaseIRQCond.....	33
8.3.11	Command setAltIRQCond.....	33
8.3.12	Command next (SPI protocol only).....	34
8.3.13	Command end.....	34
8.3.14	Command setThreshold	35
8.3.15	Command getAppSettings.....	35
8.3.16	Command getVersion.....	36
8.3.17	Command getStatus.....	36
9.	EM6420 COMMUNICATION FRAMES	38
9.1	Slave I ² C communication frame	39
9.2	Slave SPI communication frame	39
9.3	Slave 4-bit parallel communication frame.....	40
10.	TYPICAL APPLICATIONS	42
11.	PAD LOCATION DIAGRAM	45



12.	PACKAGE INFORMATION	47
12.1	Sawn 40-pin Micro Lead Frame 2 – 6 x 6 mm body	47
13.	ORDERING INFORMATION	49
14.	SPECIFICATION HISTORY	49

1. PRELIMINARIES

1.1 REFERENCE

[1] “*The I2C-Bus Specification – Version 2.1*”, Philips Semiconductors, January 2000

1.2 CONVENTIONS

The following conventions will be used in this document:

- Signals which are active low have names which start with the prefix “n_”. Example: n_rst. Signal names without this prefix are active high.
- When qualifying a signal, the term “asserted” means that the signal is active, while the term “deasserted” or “negated” means that the signal is inactive regardless of whether the active state is represented by a high or low voltage.
- When qualifying a bit within a register, the term “set” or “activated” means that the bit value is a high logic level, while the term “cleared” means that the bit value is a low logic level.
- Signal busses are denoted with the range “[MSB:LSB]” where the index of the **Most Significant Bit (MSB)** is given first and the index of the **Least Significant Bit (LSB)** is given last.
- Bit group within a register are denoted **B_{MSB} ... B_{LSB}** where the index of the **Most Significant Bit (MSB)** is given first and the index of the **Least Significant Bit (LSB)** is given last.
- Hexadecimal numbers are followed by the index “H”. Example: **1F5A_H**.
- Binary numbers are followed by the index “B”. Example: **1011_B**.
- Register names followed by the index “H” refers to the high byte of a 16-bit register.
- Register names followed by the index “L” refers to the low byte of a 16-bit register.

2. GENERAL DESCRIPTION

The **EM6420** is a very low power ASIC that includes a Touch Screen interface able to handle up to 16 capacitive sensors. Several devices can be used in parallel to manage more than 16 sensors. The application parameters (number of used sensors, sensors scan frequency, sensors sensitivity level, IRQ condition ...) are fully configurable either from a host microcontroller through a communication port or from several configuration inputs.

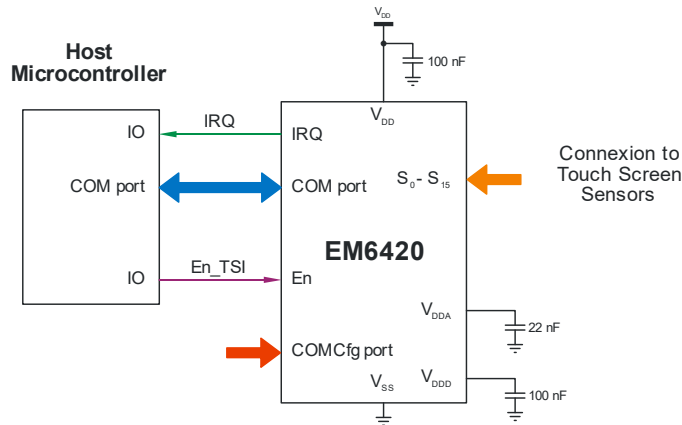


Figure 2-1: Typical Operating Configuration

Depending on the IRQ condition parameter, a user-selectable IRQ active edge can be generated:

- At the end of every scan.
- At the end of a scan if the sensors state has changed.
- At the end of a scan if either the sensors state has changed or at least one sensor is active.

The sensors state can then be read by the host microcontroller through the communication port.

The **EM6420** can also detect the most activated sensor in applications where sensors are tightly spaced by comparing relative levels among sensors and by selecting the one with the largest signal strength.

Supply voltage range can be selected either from 1.2 V to 2.0 V or from 2.2 V to 3.6 V. Depending on selected supply voltage range, 3 or 4 decoupling capacitors are required for overall functionality. No other external component is needed.

The **EM6420** can operate over a wide temperature range, from -40°C to $+85^{\circ}\text{C}$. It is available in die form or in different SMT packages.

Ultra low current consumptions have been achieved with the **EM6420** starter kit¹, typically² :

- 8.0 μA @ 3.0 V (14.5 μA @1.5 V) for 16 sensors scanned at 8 Hz in Low Power Mode
- 2.0 μA @ 3.0 V (5.0 μA @1.5 V) for 16 sensors scanned at 8 Hz in Ultra Low Power Mode
- 5.0 μA @ 3.0 V (9.0 μA @1.5 V) for 8 sensors scanned at 2 Hz in Low Power Mode
- 0.7 μA @ 3.0 V (1.2 μA @1.5 V) for 8 sensors scanned at 2 Hz in Ultra Low Power Mode

¹ Please ask EM Microelectronic-Marin SA for EM6420 starter kit availability

² Other values may be obtained depending on electrode design and selected parameters

3. FEATURES

3.1 KEY ELEMENTS

- Ultra low power and ultra low voltage Touch Screen interface
- Up to 16 sensor inputs per device
- Increased number of sensors can be addressed with more devices in parallel
- User selectable power supply range (see below)
- User-selectable active edge IRQ output signal
- User-selectable communication interface (see below)
- Active high enable input
- Maximum 4 external components needed (decoupling capacitors only)
- No software development or tuning required

3.2 POWER SUPPLY

- Low supply voltage range : 1.2 V to 2.0 V
- High supply voltage range : 2.2 V to 3.6 V
- Disabled Mode consumption : $I_{DD \text{ Disabled}} < 50 \text{ nA}$
- $I_{DD} = 8.0 \mu\text{A @ } 3.0 \text{ V}$ ($14.5 \mu\text{A @ } 1.5 \text{ V}$) for 16 sensors scanned at 8 Hz in Low Power Mode
- $I_{DD} = 2.0 \mu\text{A @ } 3.0 \text{ V}$ ($5.0 \mu\text{A @ } 1.5 \text{ V}$) for 16 sensors scanned at 8 Hz in Ultra Low Power Mode
- $I_{DD} = 5.0 \mu\text{A @ } 3.0 \text{ V}$ ($9.0 \mu\text{A @ } 1.5 \text{ V}$) for 8 sensors scanned at 2 Hz in Low Power Mode
- $I_{DD} = 0.7 \mu\text{A @ } 3.0 \text{ V}$ ($1.2 \mu\text{A @ } 1.5 \text{ V}$) for 8 sensors scanned at 2 Hz in Ultra Low Power Mode
- Internal voltage regulator for logic supply when used in high supply voltage range
- Internal voltage multiplier for analog supply when used in low supply voltage range
- Internal voltage regulator for analog supply

3.3 INTERFACES

- 4-wire SPI
- I²C (Standard-Mode or Fast-Mode compatible)
- 4-bit parallel interface
- 8-bit direct output (Standalone Mode)

3.4 DEVELOPMENT TOOLS

- **EM6420** starter kit with its related documentation
- Ultra low power User Interface reference design with **EM6420**-based Touch solution, EM6110 LCD driver and EM6819 host MCU

3.5 TOUCH MODULES BASED ON EM6420 IC

- Capacitive electrodes design capability on various non-conductive substrates (according customer's requirements)"
- Transparent / Invisible electrodes possible
- Application-specific touch modules development: contact EM-Microelectronic HQ

4. BLOCK DIAGRAM

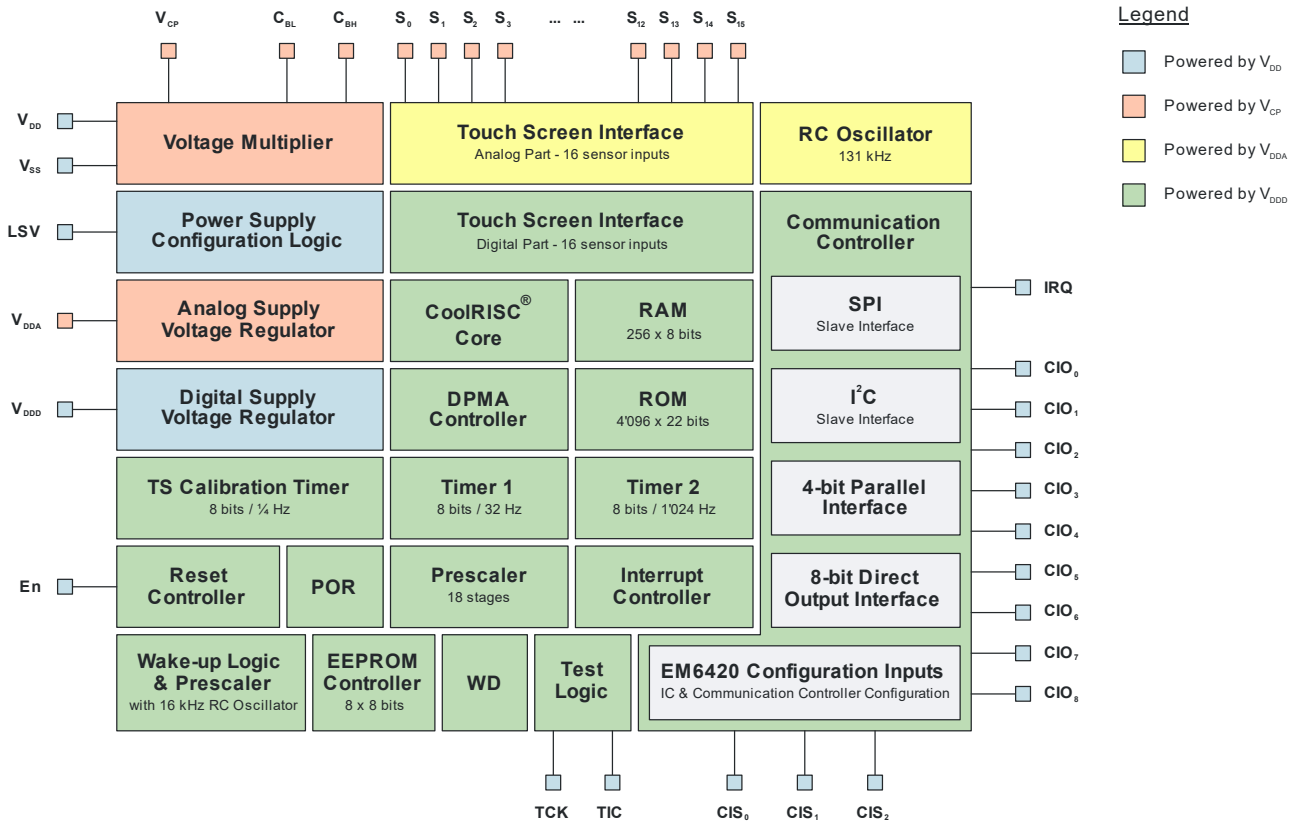


Figure 4-1 : EM6420 Block Diagram

5. PAD DESCRIPTION

PAD				
Number	Name	Type	Description	Note
1	V_{SS}	Supply	Negative power supply, bulk	Internal reference potential (ground)
2	C_{BL}	Analog	Charge pump B ooster C apacitor connection	Low voltage side
3	C_{BH}	Analog	Charge pump B ooster C apacitor connection	High voltage side
4	V_{CP}	Supply	Unregulated C harge P ump output voltage, capacitor connection	
5	V_{DDA}	Supply	Regulated A nalog supply voltage, capacitor connection	
6	S_{15}	Analog	Touch Screen S ensor 15 connection	Pull-down when not selected – See Note 3
7	S_{11}	Analog	Touch Screen S ensor 11 connection	Pull-down when not selected – See Note 3
8	S_{14}	Analog	Touch Screen S ensor 14 connection	Pull-down when not selected – See Note 3
9	S_{10}	Analog	Touch Screen S ensor 10 connection	Pull-down when not selected – See Note 3
10	S_7	Analog	Touch Screen S ensor 7 connection	Pull-down when not selected – See Note 3

Table 5-1 : EM6420 pad description



PAD				
Number	Name	Type	Description	Note
11	S ₆	Analog	Touch Screen Sensor 6 connection	Pull-down when not selected – See Note 3
12	S ₅	Analog	Touch Screen Sensor 5 connection	Pull-down when not selected – See Note 3
13	S ₄	Analog	Touch Screen Sensor 4 connection	Pull-down when not selected – See Note 3
14	S ₉	Analog	Touch Screen Sensor 9 connection	Pull-down when not selected – See Note 3
15	S ₁₃	Analog	Touch Screen Sensor 13 connection	Pull-down when not selected – See Note 3
16	S ₃	Analog	Touch Screen Sensor 3 connection	Pull-down when not selected – See Note 3
17	S ₁₂	Analog	Touch Screen Sensor 12 connection	Pull-down when not selected – See Note 3
18	S ₂	Analog	Touch Screen Sensor 2 connection	Pull-down when not selected – See Note 3
19	S ₈	Analog	Touch Screen Sensor 8 connection	Pull-down when not selected – See Note 3
20	S ₁	Analog	Touch Screen Sensor 1 connection	Pull-down when not selected – See Note 3
21	S ₀	Analog	Touch Screen Sensor 0 connection	Pull-down when not selected
22	TIC	Input	Factory – Reserved IC Test input	Pull-down – See Note 1
23	TCK	Input	Factory – Reserved IC Test Clock input	Pull-down – See Note 1
24	LSV	Input	Low Supply Voltage selection input	
25	IRQ	Output	Interrupt Request Output	Push-pull or open-drain with internal pull-up resistor
26	CI ₈	Input	Communication Controller Input 8	
27	CI ₇	Bidir	Communication Controller IO 7	See Note 2
28	CI ₆	Bidir	Communication Controller IO 6	See Note 2
29	CI ₅	Bidir	Communication Controller IO 5	See Note 2
30	CI ₄	Bidir	Communication Controller IO 4	See Note 2
31	En	Input	IC Enable input	
32	V _{DD}	Supply	Positive power supply	
33	CI ₃	Bidir	Communication Controller IO 3	See Note 2
34	CI ₂	Bidir	Communication Controller IO 2	See Note 2
35	CI ₁	Bidir	Communication Controller IO 1	See Note 2
36	CI ₀	Bidir	Communication Controller IO 0	See Note 2
37	CIS ₂	Input	Communication Interface Selector input 2	
38	CIS ₁	Input	Communication Interface Selector input 1	
39	CIS ₀	Input	Communication Interface Selector input 0	
40	V _{DDD}	Supply	Regulated Digital supply voltage, capacitor connection	
41	V _{DD}	Supply	Positive power supply	

Note 1 : Connect this pad to V_{SS} for better ESD protection in customer application

Note 2 : Depending on selected communication interface, pad type may be either Input, Output or Bidirectional

Note 3 : This pin must be left unconnected when not used

Table 5-2 : EM6420 pad description (cont'd)

6. ELECTRICAL SPECIFICATIONS

6.1 ABSOLUTE MAXIMUM RATINGS

Parameter	Conditions	Symbol	Min	Max	Units
Storage Temperature		T_{Store}	-40	125	°C
Supply Voltage	$V_{SS} = 0\text{ V}$	V_{DD}	-0.2	4.6	V
Voltage on any pin		V_{MAX}	$V_{SS} - 0.2$	$V_{DD} + 0.2$	V

Stresses above these listed maximum ratings may cause permanent damage to the device. Exposure beyond specified electrical characteristics may affect device reliability or cause malfunction.

6.2 HANDLING PROCEDURES

This device has built-in protection against high static voltages or electric fields; however, anti-static precautions should be taken as for any other CMOS integrated circuit. Unless otherwise specified, proper operation can only occur when all terminal voltages are kept within the supply voltage range.

6.3 SUPPLY VOLTAGE CONFIGURATIONS

The **EM6420** is supplied by a single external power supply between V_{DD} and V_{SS} (Ground). A voltage multiplier and two built-in voltage regulators provide supply voltages V_{DDD} for the internal logic and V_{DDA} for the analog Touch Screen blocks as well as for the system clock RC oscillator. These two regulator outputs are connected to the V_{DDD} and V_{DDA} pads respectively, through internal resistors R_{VDDD} and R_{VDDA} . Together with external capacitors C_{VDDD} and C_{VDDA} , these internal resistors implement a low pass filter function to protect the internal circuit against parasitic over and under voltages. When used, the voltage multiplier, clocked by the wake-up RC oscillator, needs an external booster capacitor C_B (typ. 47 nF) to double the input voltage and an external buffer capacitor C_{VCP} to smooth the newly generated voltage V_{CP} . Recommended values for the external capacitors C_{VDDD} , C_{VDDA} and C_{VCP} are 100 nF, 22 nF and 100 nF.

The power supply configuration depends on the selected supply voltage range (**LSV** input state). When the **LSV** input is connected to V_{DD} , the low supply voltage range is selected. The voltage regulator V_{DDD} is disabled (output tri-stated) to avoid an additional dropout voltage between V_{DD} and V_{DDD} supply voltages. In that case, the internal logic is supplied directly by V_{DD} . The voltage multiplier is enabled and the generated voltage V_{CP} supplies the voltage regulator V_{DDA} , the Touch Screen sensor pads as well as several power pads. When the **LSV** input is connected to V_{SS} , the high supply voltage range is selected. The internal logic is supplied by the voltage regulator V_{DDD} to reduce overall power consumption. The voltage multiplier is disabled (output tri-stated) and the voltage regulator V_{DDA} is supplied directly by V_{DD} .

Depending on the selected supply voltage range, 3 or 4 decoupling capacitors are required for the entire functionality of the EM6420 from -40 to + 85°C. Refer to the schematics below for proper mode of operation.

In high supply voltage range (**LSV** is deasserted), connect:

- a 100nF decoupling capacitor to V_{DDD}
- a 100nF decoupling capacitor to V_{DD}
- a 22nF decoupling capacitor to V_{DDA}

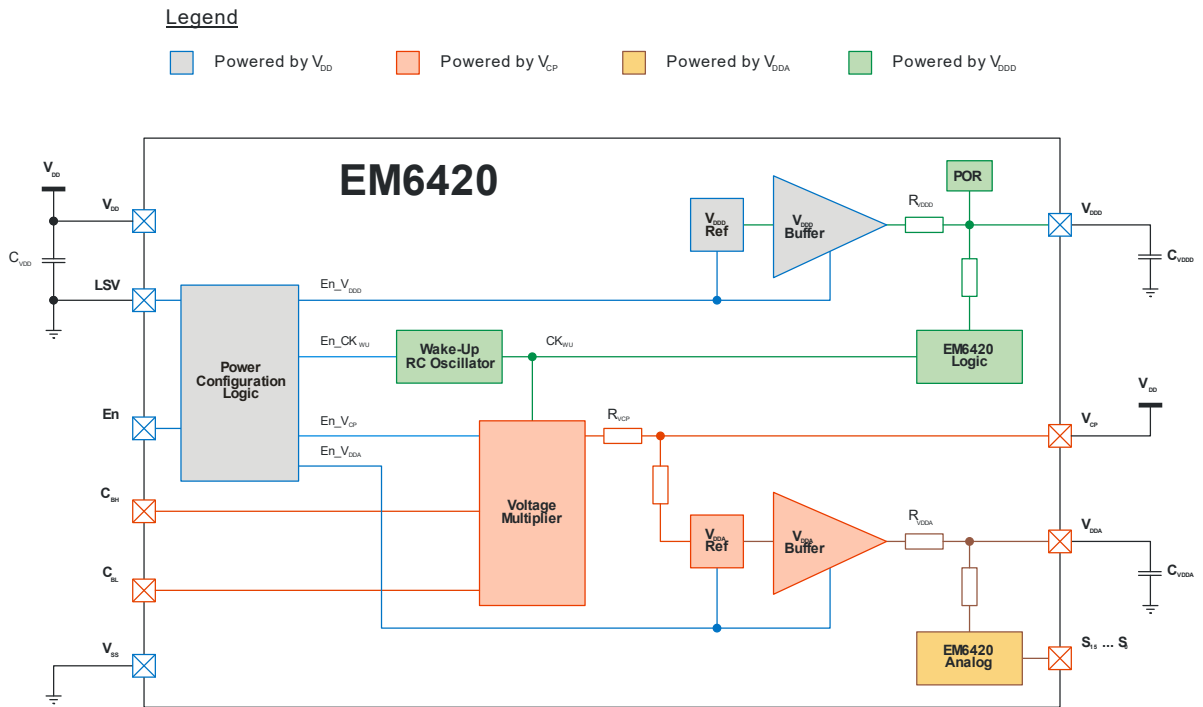


Figure 6-1: EM6420 power supply configuration when the high voltage supply range is selected

In low supply voltage range (**LSV** is asserted), connect:

- a 47nF capacitor to pins C_{BH} and C_{BL}
- a 100nF decoupling capacitor to V_{CP}
- a 100nF decoupling capacitor to V_{DD}
- a 22nF decoupling capacitor to V_{DDA}

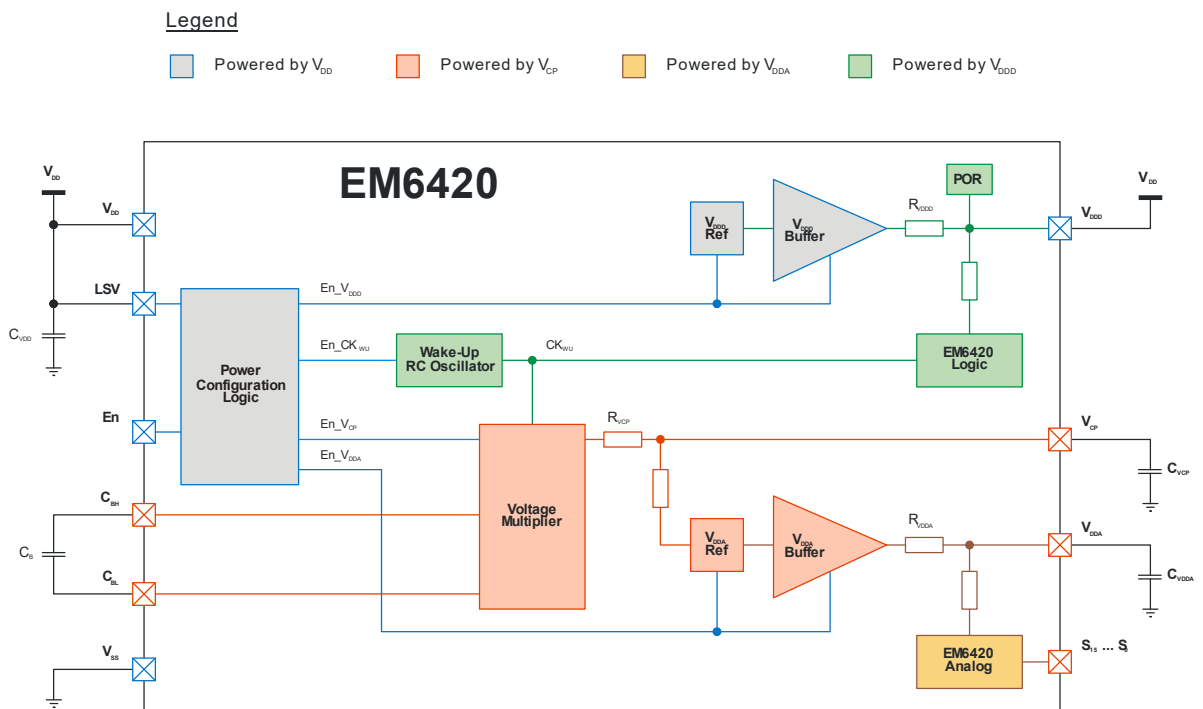


Figure 6-2: EM6420 power supply configuration when the low voltage supply range is selected

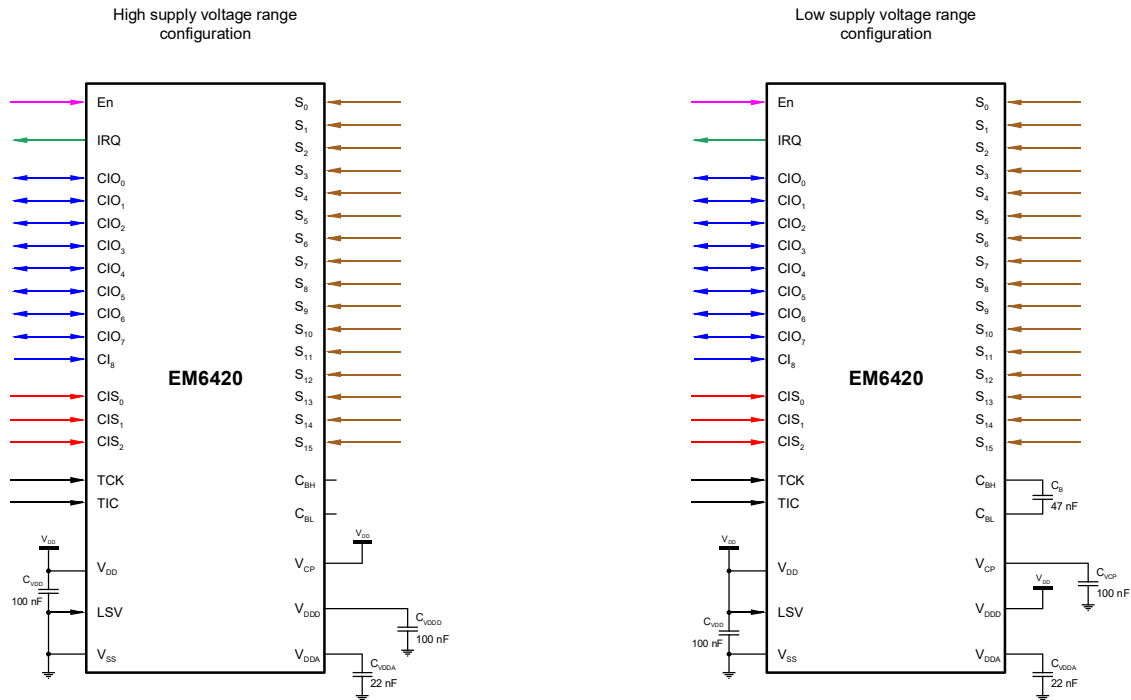


Figure 6-3 : **EM6420** simplified schematic of both supply voltage configurations

6.4 STANDARD OPERATING CONDITIONS

The **EM6420** can be used in two different modes according to customer application requirements: Low Power Mode or Ultra Low Power Mode (see § 8.3.3).

In Low Power Mode, the **EM6420** device remains always in Active Mode, i.e. during the scans of the sensors and also between them. The Touch Screen interface settings are internally chosen in order to minimize the current consumption. Furthermore, the communication between the host microcontroller and the **EM6420** is more efficient than in Ultra low Power Mode. The **EM6420** is indeed always active and so it takes less time to reply to a received command.

In Ultra Low Power Mode, the **EM6420** device remains in Active Mode only during the scans of the sensors and goes in Sleep Mode between them. The Touch Screen interface settings are internally chosen in order to scan as fast as possible the sensors, thus shortening as much as possible the time the **EM6420** device remains in Active Mode. Furthermore, receiving a command while in Sleep Mode may slow down the communication between the host microcontroller and the **EM6420**, as it has first to return in Active Mode before preparing and sending the reply. But when this mode is selected, it reduces the **EM6420** power consumption with full functionality to the minimum (see typical values above).

If the Touch Screen interface has to be switched off for a long time, it is strongly recommended to put the **EM6420** in Disable Mode by putting the **En** input to V_{SS} instead of simply sending a **stopTS** command (see § 8.3.2). In this case, the current consumption is reduced to a few nA, but the **EM6420** loses the application parameters and the host microcontroller must send them again next time the **En** input is set to V_{DD} .

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Operating temperature		T_{Op}	-40		85	°C
Low supply voltage range		V_{DDL}	1.2		2.0	V
High supply voltage range		V_{DDH}	2.2		3.6	V
Reference terminal		V_{SS}		0		V
Regulated voltage V_{DDD} capacitor		C_{VDDD}	100		470	nF
Regulated voltage V_{DDA} capacitor		C_{VDDA}		22		nF
Unregulated voltage V_{CP} buffer capacitor		C_{VCP}		100		nF
Voltage multiplier booster capacitor		C_B		47		nF

6.5 DC CHARACTERISTICS – POWER SUPPLY

Conditions unless otherwise specified : $V_{DD} = 3.0\text{ V}$, $T = 25^\circ\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Supply voltage	$T = -40\text{ to }+85^\circ\text{C}$	V_{DD}	2.2	3.0	3.6	V
Disable mode current consumption ³	En input connected to V_{SS}	I_{DD_Dis}		2	10	nA
Sleep mode current consumption		I_{DD_Slp}		470	580	nA
Active mode current consumption	Touch Screen OFF	I_{DD_RUN}		7.5	9.5	μA

Conditions unless otherwise specified : $V_{DD} = 1.5\text{ V}$, $T = 25^\circ\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Supply voltage	$T = -40\text{ to }+85^\circ\text{C}$	V_{DD}	1.2	1.5	2.0	V
Disable mode current consumption ⁴	En input connected to V_{SS}	I_{DD_Dis}		2	10	nA
Sleep mode current consumption		I_{DD_Slp}		380	530	nA
Active mode current consumption	Touch Screen OFF	I_{DD_RUN}		12.0	16.5	μA

6.6 POR

Conditions unless otherwise specified : $V_{DD} = 1.5\text{ V}$, $T = 25^\circ\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
High threshold voltage		V_{IH_POR}	0.75	0.90	1.10	V
Threshold voltage hysteresis		V_{Hys_POR}	50		110	mV

³ This value is guaranteed by design

⁴ This value is guaranteed by design

6.7 TOUCH SCREEN INTERFACE

Conditions unless otherwise specified : $V_{DD} = 3.0\text{ V}$ or $V_{DD} = 1.5\text{ V}$, $T = 25^\circ\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Reference capacitor	$TS_RCap = 00_H$	C_R		0.5		pF
	$TS_RCap = 3F_H$			31.5		pF
Reference capacitor increment	$\Delta TS_RCap = 1$	ΔC_R		0.5		pF
Pad S_x input current	Pull-down activated, analog switch turned OFF $V_{IN} = 0.3\text{ V}$	I_{IN_SPdON}	100	180	260	μA

6.8 INPUT PADS CIS_x , CI_8 AND LSV

Conditions unless otherwise specified : $V_{DD} = 3.0\text{ V}$ or $V_{DD} = 1.5\text{ V}$, $T = 25^\circ\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Low level input voltage		V_{IL_CIS}	V_{SS}		$0.3 \cdot V_{DD}$	V
High level input voltage		V_{IH_CIS}	$0.7 \cdot V_{DD}$		V_{DD}	V
Static input current	$V_{IN} = V_{SS} \dots V_{DD}$	I_{IN_CIS}	-100		100	nA

6.9 INPUT PAD EN

Conditions unless otherwise specified : $V_{DD} = 3.0\text{ V}$, $T = 25^\circ\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Low level input voltage		V_{IL_En}	V_{SS}		0.7	V
High level input voltage		V_{IH_En}	2.2		V_{DD}	V
Schmitt trigger hysteresis		V_{Hys_En}		0.8		V
Static input current	$V_{IN} = V_{SS} \dots V_{DD}$	I_{IN_En}	-100		100	nA
Min filtered glitches width		t_{FG_En}			10	μs
Valid reset pulse width ($En = V_{SS}$)		t_{En}	50			μs

Conditions unless otherwise specified : $V_{DD} = 1.5\text{ V}$, $T = 25^\circ\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Low level input voltage		V_{IL_En}	V_{SS}		0.4	V
High level input voltage		V_{IH_En}	1.1		V_{DD}	V
Schmitt trigger hysteresis		V_{Hys_En}		0.2		V
Static input current	$V_{IN} = V_{SS} \dots V_{DD}$	I_{IN_En}	-100		100	nA
Min filtered glitches width		t_{FG_En}			10	μs
Valid reset pulse width ($En = V_{SS}$)		t_{En}	50			μs

6.10 OUTPUT PAD IRQ

Conditions unless otherwise specified : $V_{DD} = 3.0\text{ V}$, $T = 25^\circ\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Low level output current	$V_{OUT} = 0.3\text{ V}$	I_{OL_IRQ}	3.0			mA
High level output current	Push-pull configuration $V_{OUT} = V_{DD} - 0.3\text{ V}$	I_{OH_IRQP}			-3.0	mA
	Open drain with internal pull-up configuration $V_{OUT} = V_{SS}$	I_{OH_IRQQ}	-140		-75	μA
Internal pull-up resistance		R_{PU_IRQ}		30		$\text{k}\Omega$

Conditions unless otherwise specified : $V_{DD} = 1.5\text{ V}$, $T = 25^\circ\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Low level output current	$V_{OUT} = 0.3\text{ V}$	I_{OL_IRQ}	1.6			mA
High level output current	Push-pull configuration $V_{OUT} = V_{DD} - 0.3\text{ V}$	I_{OH_IRQP}			-1.6	mA
	Open drain with internal pull-up configuration $V_{OUT} = V_{SS}$	I_{OH_IRQQ}	-75		-35	μA
Internal pull-up resistance		R_{PU_IRQ}		30		$\text{k}\Omega$

6.11 BIDIRECTIONAL PADS CIO₂ ... CIO₇

Conditions unless otherwise specified : $V_{DD} = 3.0\text{ V}$, $T = 25^\circ\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Low level output current	$V_{OUT} = 0.3\text{ V}$	I_{OL_CIO}	3.0			mA
High level output current	$V_{OUT} = V_{DD} - 0.3\text{ V}$	I_{OH_CIO}			-3	mA
Low level input voltage		V_{IL_CIO}	V_{SS}		$0.3 \cdot V_{DD}$	V
High level input voltage		V_{IH_CIO}	$0.7 \cdot V_{DD}$		V_{DD}	V
Static input current	$V_{IN} = V_{SS} \dots V_{DD}$	I_{IN_CIO}	-100		100	nA

Conditions unless otherwise specified : $V_{DD} = 1.5\text{ V}$, $T = 25^\circ\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Low level output current	$V_{OUT} = 0.3\text{ V}$	I_{OL_CIO}	1.6			mA
High level output current	$V_{OUT} = V_{DD} - 0.3\text{ V}$	I_{OH_CIO}			-1.6	mA
Low level input voltage		V_{IL_CIO}	V_{SS}		$0.3 \cdot V_{DD}$	V
High level input voltage		V_{IH_CIO}	$0.7 \cdot V_{DD}$		V_{DD}	V
Static input current	$V_{IN} = V_{SS} \dots V_{DD}$	I_{IN_CIO}	-100		100	nA

6.12 BIDIRECTIONAL PADS CIO0 AND CIO1

 Conditions unless otherwise specified : $V_{DD} = 3.0\text{ V}$, $T = 25^{\circ}\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Low level output current	$V_{OUT} = 0.3\text{ V}$	I_{OL_I2C}	3.0			mA
High level output current	Push-pull configuration $V_{OUT} = V_{DD} - 0.3\text{ V}$	I_{OH_I2CP}			-3.0	mA
	Open drain with internal weak pull-up configuration $V_{OUT} = V_{SS}$	I_{OH_I2CWR}	-120		-50	μA
	Open drain with internal strong pull-up configuration $V_{OUT} = V_{SS}$	I_{OH_I2CSR}	-200		-110	μA
Internal weak pull-up resistance		R_{I2C_W}		40		$\text{k}\Omega$
Internal strong pull-up resistance		R_{I2C_S}		20		$\text{k}\Omega$
Low level input voltage		V_{IL_I2C}	V_{SS}		$0.3 \cdot V_{DD}$	V
High level input voltage		V_{IH_I2C}	$0.7 \cdot V_{DD}$		V_{DD}	V
Static input current	Open drain with no internal pull-up configuration $V_{IN} = V_{SS} \dots V_{DD}$	I_{IN_I2C}	-100		100	nA
Schmitt trigger hysteresis		V_{Hys_I2C}	$0.05 \cdot V_{DD}$			V

 Conditions unless otherwise specified : $V_{DD} = 1.5\text{ V}$, $T = 25^{\circ}\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Low level output current	$V_{OUT} = 0.3\text{ V}$	I_{OL_I2C}	1.6			mA
High level output current	Push-pull configuration $V_{OUT} = V_{DD} - 0.3\text{ V}$	I_{OH_I2CP}			-1.6	mA
	Open drain with internal weak pull-up configuration $V_{OUT} = V_{SS}$	I_{OH_I2CWR}	-60		-25	μA
	Open drain with internal strong pull-up configuration $V_{OUT} = V_{SS}$	I_{OH_I2CSR}	-100		-55	μA
Internal weak pull-up resistance		R_{I2C_W}		40		$\text{k}\Omega$
Internal strong pull-up resistance		R_{I2C_S}		20		$\text{k}\Omega$
Low level input voltage		V_{IL_I2C}	V_{SS}		$0.3 \cdot V_{DD}$	V
High level input voltage		V_{IH_I2C}	$0.7 \cdot V_{DD}$		V_{DD}	V
Static input current	Open drain with no internal pull-up configuration $V_{IN} = V_{SS} \dots V_{DD}$	I_{IN_I2C}	-100		100	nA
Schmitt trigger hysteresis		V_{Hys_I2C}	$0.1 \cdot V_{DD}$			V

7. TIMING SPECIFICATIONS

7.1 STANDARD OPERATING CONDITIONS

Parameter	Symbol	Min	Typ.	Max	Units
Operating Temperature	T_{Op}	-40		85	°C
Low Supply Voltage Range	V_{DDL}	1.2	1.5	2.0	V
High Supply Voltage Range	V_{DDH}	2.2	3.0	3.6	V
Reference terminal	V_{SS}		0		V

7.2 COMMUNICATION INTERFACE

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Communication interface start-up time	8-bit Direct Output Interface selected	t_{Cl_St}			70	ms
	SPI, I2C or Parallel Interface selected				10	ms
IRQ start-up pulse width		t_{IRQ_SIPW}	100		150	µs

7.3 8-BIT DIRECT OUTPUT INTERFACE

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
IRQ pulse width		t_{IRQ_PW}	100		150	µs

7.4 SLAVE I2C INTERFACE

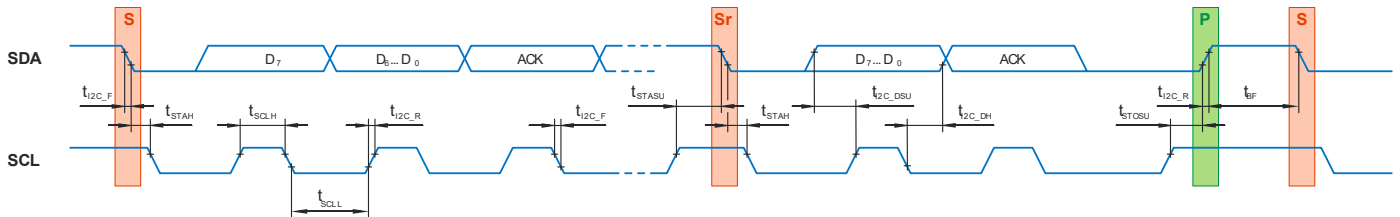


Figure 7-1: I2C interface timings

Conditions unless otherwise specified : $V_{DD} = 3.0\text{ V}$ or $V_{DD} = 1.5\text{ V}$, $T = 25^\circ\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
SCL clock frequency		f_{SCL}	0		400	kHz
Hold time (repeated) START condition.		t_{STAH}	0.6			µs
Low period of the SCL clock		t_{SCLL}	1.3			µs
High period of the SCL clock		t_{SCLH}	0.6			µs

Conditions unless otherwise specified : $V_{DD} = 3.0\text{ V}$ or $V_{DD} = 1.5\text{ V}$, $T = 25^\circ\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Setup time for a repeated START condition		t_{STASU}	0.6			μs
Data hold time		t_{I2C_DH}	0		0.9	μs
Data setup time		t_{I2C_DSU}	100			ns
Rise time of both SDA and SCL signals		t_{I2C_R}	$20 + 0.1 \cdot C_b$		300	ns
Fall time of both SDA and SCL signals		t_{I2C_F}	$20 + 0.1 \cdot C_b$		300	ns
Setup time for a STOP condition		t_{STOSU}	0.6			μs
Bus free time between a STOP and a START condition		t_{BF}	1.3			μs
Capacitive load for each bus line	With internal pull-up resistors	C_b			200	pF
	With external pull-up resistors	C_b			400	pF

7.5 SLAVE SPI INTERFACE

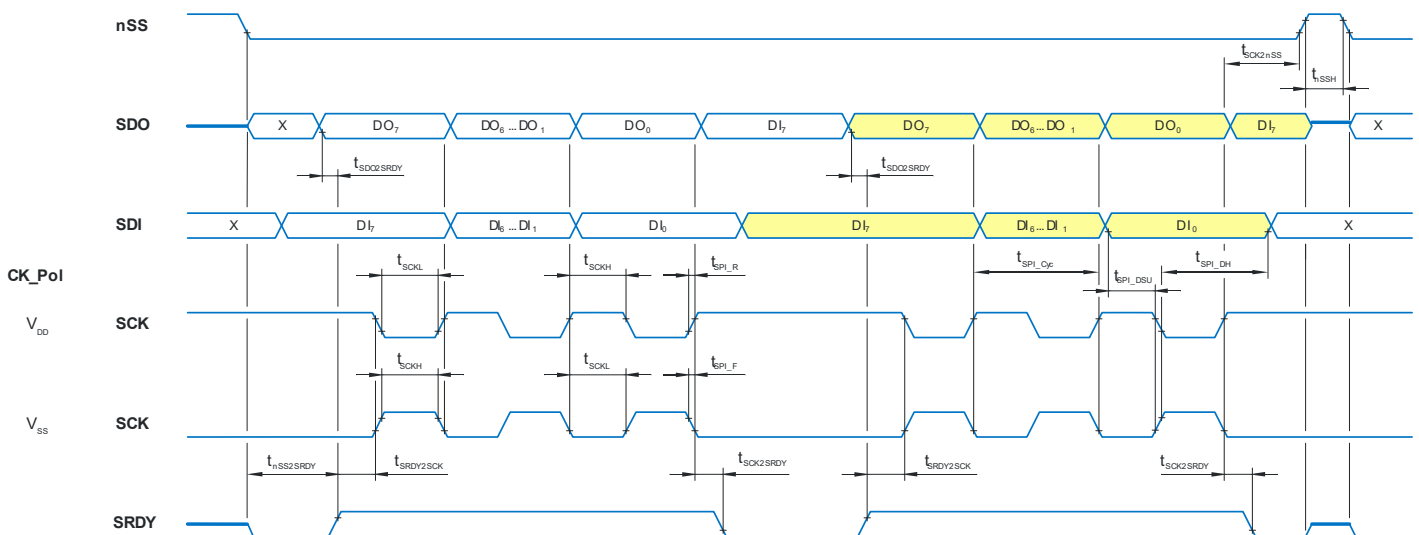


Figure 7-2 : SPI Interface timings when **CK_Pha** input is set to V_{SS} .

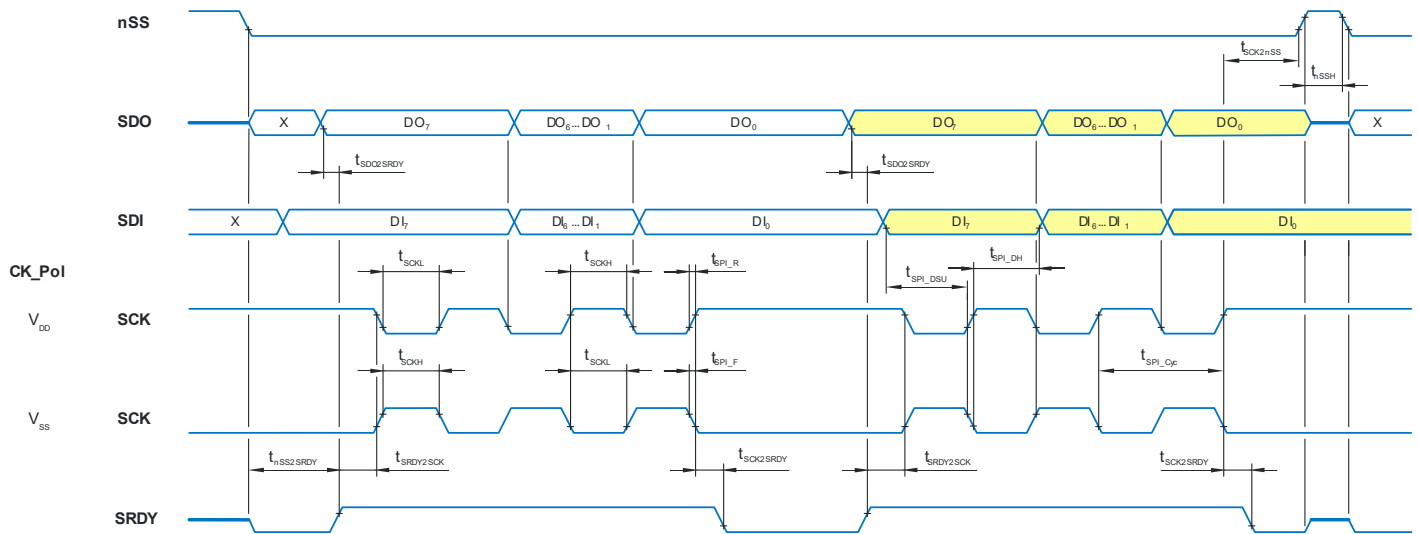


Figure 7-3 : SPI Interface timings when CK_Pha input is set to V_{DD}.

Conditions unless otherwise specified : V_{DD} = 3.0 V or V_{DD} = 1.5 V, T = 25°C

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Operating frequency		f _{SPI_Op}			400	kHz
Cycle time		t _{SPI_Cyc}	2.5			μs
Rise time of inputs SCK, SDI and nSS		t _{SPI_R}			250	ns
Fall time of inputs SCK, SDI and nSS		t _{SPI_F}			250	ns
Delay from nSS low to SRDY high		t _{nSS2SRDY}			200	ns
Low period of the SCK clock		t _{SCKL}	1			μs
High period of the SCK clock		t _{SCKH}	1			μs
Data setup time		t _{SPI_DSU}	200			ns
Data hold time		t _{SPI_DH}	200			ns
Delay from valid data to SRDY high		t _{SDO2SRDY}			200	ns
Delay from SRDY high to first SCK edge		t _{SRDY2SCK}	200			ns
Delay from last SCK edge to SRDY low		t _{SCK2SRDY}			200	ns
Delay from last SCK edge to nSS high		t _{SCK2nSS}			200	ns
nSS high time (Bus free time between communication frames)		t _{nSSH}	2			μs

7.6 SLAVE 4-BIT PARALLEL INTERFACE

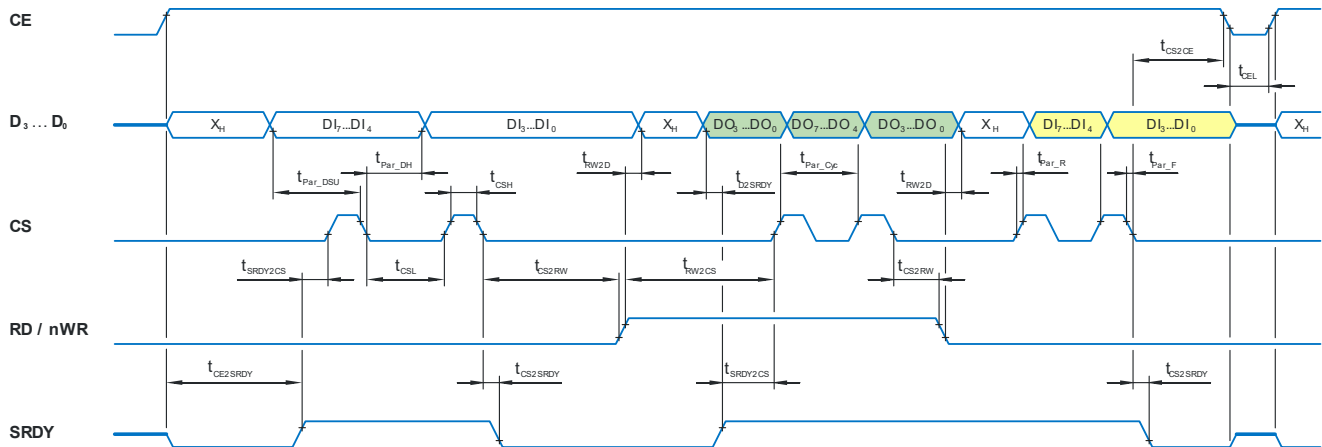


Figure 7-4 : Parallel Interface timings

Conditions unless otherwise specified : $V_{DD} = 3.0\text{ V}$ or $V_{DD} = 1.5\text{ V}$, $T = 25^\circ\text{C}$

Parameter	Conditions	Symbol	Min	Typ.	Max	Units
Operating frequency		$f_{\text{Par_Op}}$			400	kHz
Cycle time		$t_{\text{Par_Cyc}}$	2.5			μs
Rise time of inputs CE , CS , RD / nWR and D_x		$t_{\text{Par_R}}$			250	ns
Fall time of inputs CE , CS , RD / nWR and D_x		$t_{\text{Par_F}}$			250	ns
Delay from CE high to SRDY high		t_{CE2SRDY}			200	ns
Time interval between CS strobes		t_{CSL}	1			μs
CS strobe width		t_{CSH}	1			μs
Data setup time		$t_{\text{Par_DSU}}$	200			ns
Data hold time		$t_{\text{Par_DH}}$	200			ns
Delay from valid data to SRDY high		t_{D2SRDY}			200	ns
Delay from SRDY high to CS strobe		t_{SRDY2CS}	200			ns
Delay from CS strobe to SRDY low		t_{CS2SRDY}			200	ns
Delay from RD / nWR low to valid data		t_{RW2D}			200	ns
Delay from RD / nWR high to CS strobe		t_{RW2CS}	200			ns
Delay from CS strobe to RD / nWR low		t_{CS2RW}	200			ns
Delay from CS strobe to CE low		t_{CS2CE}	200			ns
CE low time (Bus free time between communication frames)		t_{CEL}	2			μs

8. EM6420 TO HOST CONTROLLER COMMUNICATION

8.1 INTRODUCTION

The **EM6420** can communicate with a host processor through several communication interfaces, mainly to receive application parameters, to signal sensors activity or to send **EM6420** status / error flags. Only one communication interface can be active at a time, as they share the same **EM6420** IO pads **CIO₇ ... CIO₀** and input pad **CI₈**. During a communication, the host processor is always considered as the master device and the **EM6420** as the slave one. Thus, the **EM6420** may never initiate a communication. However, by asserting its output pad **IRQ**, the **EM6420** can signal to the host processor that a predefined condition or an error occurred and that a communication may be initiated, normally by a **getStatus** command.

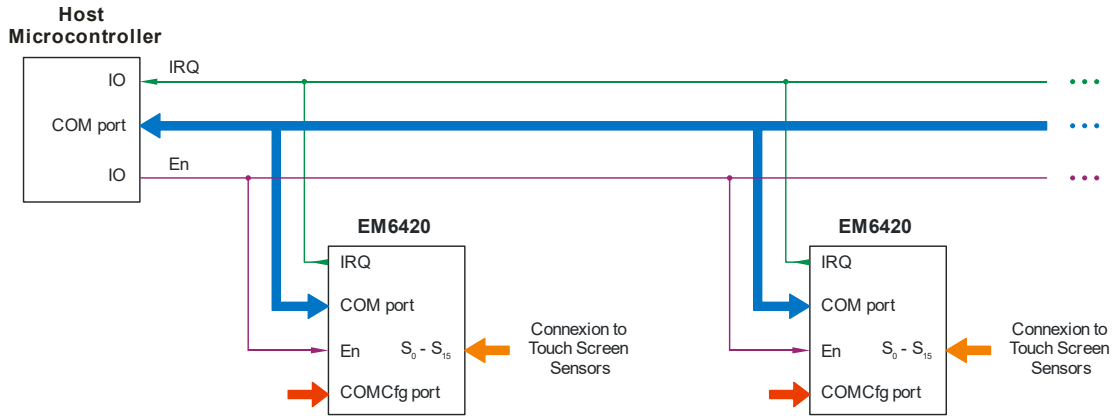


Figure 8-1 : Multi **EM6420** configuration

In applications where several **EM6420** are used, the open-drain with internal pull-up resistor configuration must be selected for **IRQ** output pads, to allow connecting all these output pads to a unique host **IRQ** input (see *Figure 8-1*).

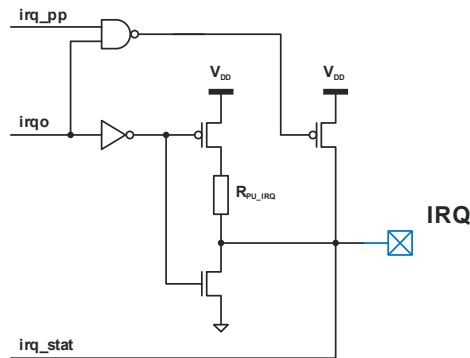


Figure 8-2 : Configuration of **EM6420** **IRQ** output pad

In this case, all **IRQ** output pads should be asserted by default (wired-OR), and each **EM6420** can signal to the host processor that a communication may be initiated by deasserting its output pad **IRQ**.

8.2 EM6420 COMMUNICATION INTERFACES

Active **EM6420** communication interface is selected by input pads **CIS₂ ... CIS₀** state, according to the Table 8-1.

CIS₂	CIS₁	CIS₀	Active Communication Interface
0	0	0	Slave I ² C Interface
0	0	1	Slave 4-bit Parallel Interface
0	1	X	Slave SPI Interface
1	X	X	8-bit Direct Output Interface

Table 8-1 : Communication interface selection

As input pads **CIS₂ ... CIS₀** do not include pull resistors, they must be connected either to **V_{SS}** or **V_{DD}** in customer application. Selecting a communication interface will directly define the functionality of communication input pad **CI₈** and IO pads **CIO₇ ... CIO₀**, thus configuring IO pads either as input, output or bidirectional pad.

8.2.1 Slave I2C Interface

When slave I²C interface is selected, communication pads **CI₈** and **CIO₇ ... CIO₀** are configured for specific I²C functions or define I²C interface options, according to the Table 8-2:

PAD			
Name	Alternate name	Type	Specific function or defined option for slave I²C interface
CIO₀	SCL	Bidir	I ² C Serial Clock
CIO₁	SDA	Bidir	I ² C Serial Data
CIO₂	EN_IWPU	Input	Enable Internal Weak Pull-Up resistors
CIO₃	EN_ISPU	Input	Enable Internal Strong Pull-Up resistors
CIO₇ ... CIO₄	A₃ ... A₀	Input	Low 4-bit I ² C Address. Default high 3-bits I ² C address are 100 _B
CI₈	IRQ_PoI	Input	IRQ Polarity

Table 8-2 : Defined I²C options and specific functions for communication pads **CI₈** and **CIO₇ ... CIO₀** when slave I²C interface is selected

This I²C interface fulfills the I²C specification (see [1] “The I²C-Bus Specification – Version 2.1”, Philips Semiconductors, January 2000) with the following restrictions:

- Only Standard-mode and Fast-mode are supported. Thus, the maximum clock frequency is 400 kHz.
- Only standard 7-bit addressing is supported. The default values of the higher three bits are 100_B⁵ while the lower 4 bits are defined by **A₃...A₀** input pads.
- General Calls are ignored.
- Each I²C bidir pad has a weak and a strong internal pull-up resistor. They can be enabled by connecting the **En_IWPU** and / or **EN_ISPU** input pads to **V_{DD}**. However, these internal pull-up resistors have been designed to minimize power consumption. As such, they can only drive capacitive bus loads up to 200 pF, even when both pull-up resistors are simultaneously enabled. For higher capacitive bus loads, external I²C pull-up resistors must be added.

⁵ Please contact EM Microelectronic-Marin SA for setting other values to the three higher address bits

The **IRQ_PoI** input defines the polarity of the **IRQ** output:

- The positive IRQ polarity is selected when the **IRQ_PoI** input is connected to V_{DD} , and a rising edge is generated when the **EM6420** asserts its **IRQ** output.
- The negative IRQ polarity is selected when the **IRQ_PoI** input is connected to V_{SS} , and a falling edge is generated when the **EM6420** asserts its **IRQ** output.

8.2.2 Slave SPI Interface

When slave SPI interface is selected, communication pads **CI₈** and **CIO₇ ... CIO₀** are configured for specific SPI functions or define SPI interface options, according to the Table 8-3:

PAD			
Name	Alternate name	Type	Specific function or defined option for slave SPI interface
CIO₀	SCK	Input	SPI Serial Clock
CIO₁	SDI	Input	SPI Serial Data Input
CIO₂	SDO	Output	SPI Serial Data Output
CIO₃	nSS	Input	SPI Slave Select (active low)
CIO₄	SRDY	Output	SPI Slave Ready (see below)
CIO₅	CK_PoI	Input	SPI Clock Polarity (see below)
CIO₆	CK_Pha	Input	SPI Clock Phase (see below)
CIO₇	MSB_First	Input	SPI data are sent MSB First (see below)
CI₈	IRQ_PoI	Input	IRQ Polarity (see below)

Table 8-3 : Defined SPI options and specific functions for communication pads **CI₈** and **CIO₇ ... CIO₀** when slave SPI interface is selected

This 4-wire SPI interface allows full-duplex, synchronous, serial communication between the host and the **EM6420**. The clock signal **SCK** generated by the host synchronizes data transmission.

The **nSS** input is the control signal used to enable the **EM6420** SPI interface. When set to V_{DD} , the **SDO** and the **SRDY** outputs are tri-stated, thus allowing another **EM6420** to take control of these lines in applications where several devices are used (see Figure 8-3).

- The positive IRQ polarity is selected when the **IRQ_Pol** input is connected to V_{DD} , and a rising edge is generated when the **EM6420** asserts its **IRQ** output.
- The negative IRQ polarity is selected when the **IRQ_Pol** input is connected to V_{SS} , and a falling edge is generated when the **EM6420** asserts its **IRQ** output.

8.2.3 Slave 4-bit Parallel Interface

When slave 4-bit parallel interface is selected, communication pads **CI₈** and **CIO₇ ... CIO₀** are configured for specific 4-bit parallel functions and define **IRQ** output pad polarity, according to the Table 8-4:

PAD			
Name	Alternate name	Type	Specific function for slave 4-bit parallel interface
CIO₃ ... CIO₀	D₃ ... D₀	Bidir	4-bit Data bus
CIO₄	CE	Input	Chip Enable control signal
CIO₅	RD / nWR	Input	ReaD / not WRite control signal
CIO₆	CS	Input	Chip Select control signal
CIO₇	SRDY	Output	Slave ReaDY
CI₈	IRQ_Pol	Input	IRQ Polarity

Table 8-4 : Defined **IRQ** polarity and specific functions for communication pads **CI₈** and **CIO₇ ... CIO₀** when slave 4-bit parallel interface is selected

This parallel interface allows fast bidirectional and synchronous communication between the host and the **EM6420**.

The **CE** input is the control signal used to enable the **EM6420** parallel interface. When set to V_{SS} , the data lines **D₃ ... D₀** as well as the **SRDY** output are tri-stated, thus allowing another **EM6420** to take control of these lines in applications where several devices are used.

When the **CE** input is set to V_{DD} , the **EM6420** drives its **SRDY** output and also the data lines **D₃ ... D₀** if the **RD / nWR** input is set to V_{DD} too. The data lines **D₃ ... D₀** are driven by the host when the **RD / nWR** control signal is set to V_{SS} .

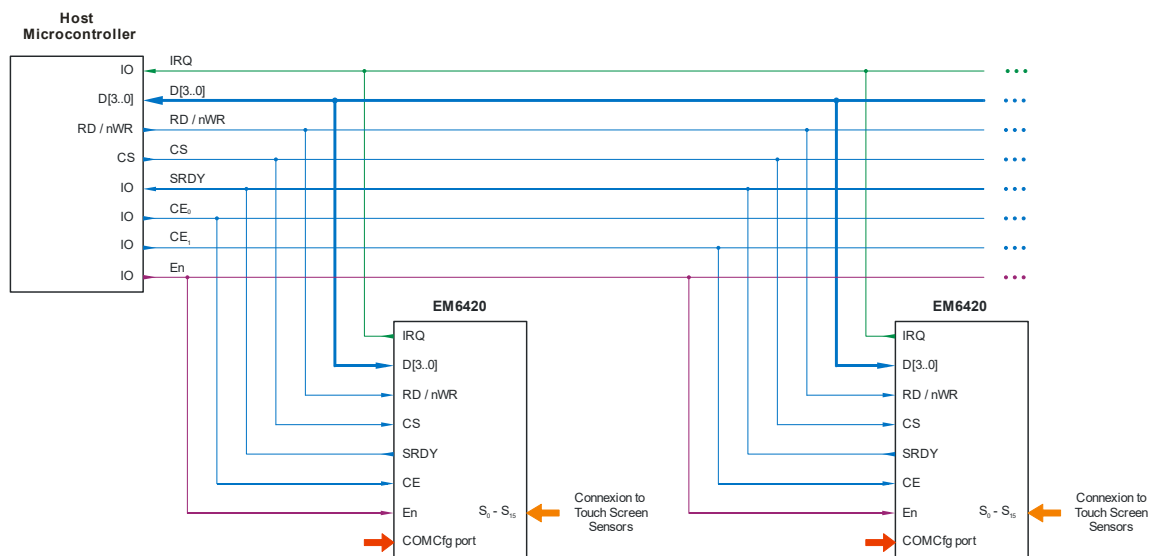


Figure 8-5: Multi EM6420 configuration using the 4-bit parallel interface

The **CS** input is the control signal used to effectively read or write a data nibble on the data bus. Data lines **D₃ ... D₀** can only change at the **CS** rising edge, and they are sampled at the **CS** falling edge.

The **SRDY** output indicates to the host that the **EM6420** is ready to send or receive a data byte. The host must always check that **SRDY** is set to **V_{DD}** before generating the two **CS** strobes needed to transfer a data byte. Data byte is sent high nibble first.

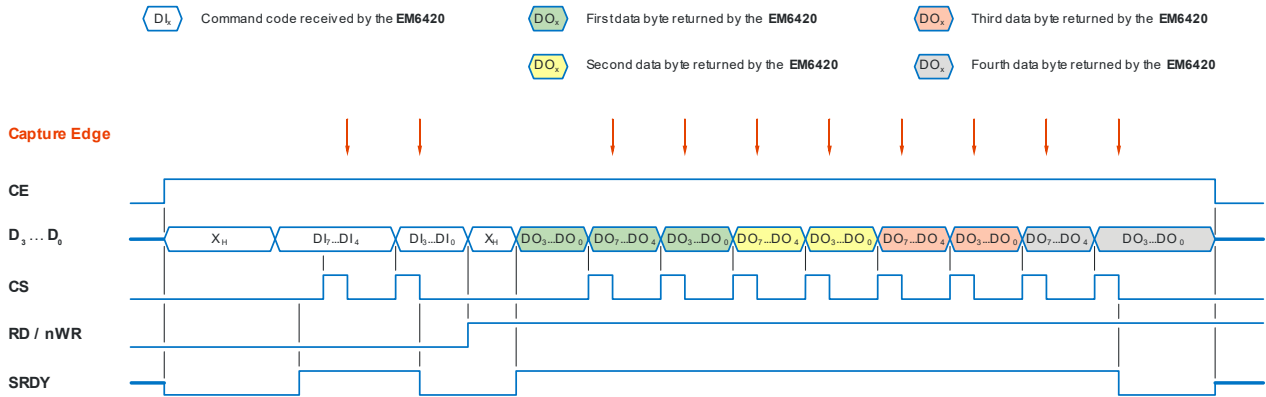


Figure 8-6: Typical data transfer using the parallel 4-bit interface.

The **IRQ_PoI** input defines the polarity of the **IRQ** output:

- The positive **IRQ** polarity is selected when the **IRQ_PoI** input is connected to **V_{DD}**, and a rising edge is generated when the **EM6420** asserts its **IRQ** output.
- The negative **IRQ** polarity is selected when the **IRQ_PoI** input is connected to **V_{SS}**, and a falling edge is generated when the **EM6420** asserts its **IRQ** output.

8.2.4 8-bit Direct Output Interface

When 8-bit direct output interface is selected, communication pads **CI₈** and **CIO₇ ... CIO₀** are configured for specific 8-bit direct output functions and define Touch Screen **IRQ** condition, according to the Table 8-5:

PAD			
Name	Alternate name	Type	Specific function for 8-bit direct output interface
CIO₇ ... CIO₀	SStO₇ ... SStO₀	Output	Sensors Status Output port
CI₈	MAS	Input	Touch Screen Most Activated Sensor feature

Table 8-5 : Touch Screen feature and specific functions for communication pads **CI₈** and **CIO₇ ... CIO₀** when 8-bit direct output interface is selected

In this configuration, the **EM6420** can only provide sensors status on an 8-bit output port. As no application parameters can be received from the host processor, the **EM6420** defines itself the number of sensors to be scanned (up to 8 sensors) while the Touch Screen scan frequency is defined from configuration inputs **CIS₀** and **CIS₁**, according to Table 8-6. Therefore, the **EM6420** may also be used in a standalone configuration, i.e. without any host processor connection.

CIS₁	CIS₀	Touch Screen scan frequency
0	0	2 Hz
0	1	8 Hz
1	0	32 Hz
1	1	128 Hz

Table 8-6 : Touch Screen scan frequency when 8-bit Direct Output interface is selected (Standalone configuration)

When 8-bit direct output interface is selected, the Touch Screen interface is always ON. Activating a sensor will directly asserts its corresponding bit on output port **SStO₇ ... SStO₀**. When input pad **MAS** is connected to **V_{DD}**, only the bit corresponding to the most activated sensor is asserted, even if other sensors are also active. By default, the output port polarity is positive⁶, i.e. the **SStO_x** outputs are asserted when they are set to **V_{DD}**.

A pulse of at least 100 μs is generated on the **IRQ** output every time a Touch Screen IRQ condition occurred. By default, the pulse polarity is negative and the open-drain with internal pull-up resistor configuration is selected by embedded software for the **IRQ** output pad⁷.

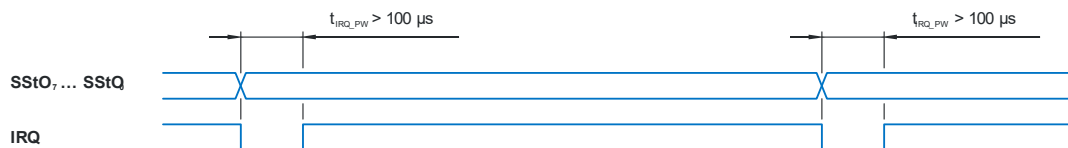


Figure 8-7: Default IRQ output pad timing when 8-bit direct output is selected

⁶ Please contact EM Microelectronic-Marin SA to change the output port polarity from positive to negative, in order to have the **SStO_x** outputs set to **V_{SS}** when they are asserted

⁷ Please contact EM Microelectronic-Marin SA to change the IRQ polarity from negative to positive and to select the push-pull instead of the open-drain configuration for the **IRQ** output pad

8.2.5 Communication interface initialization.

At start-up, the **EM6420** can determine which communication interface is active. It then sets the required communication options according to the communication pads **CI₈** and **CI₀ ... CI₇** state. Finally, the **EM6420** **IRQ** output pad is activated and special code 18_H is returned in the next **getStatus** command, thus signaling to the host controller that it is ready to accept communication frames. See Table 8-16 on page 38 for a complete list of the possible special codes.

When the positive **IRQ** polarity is used, the push-pull configuration is selected by the embedded software for the **IRQ** output pad and the timings shown in Figure 8-8 are generated at startup.

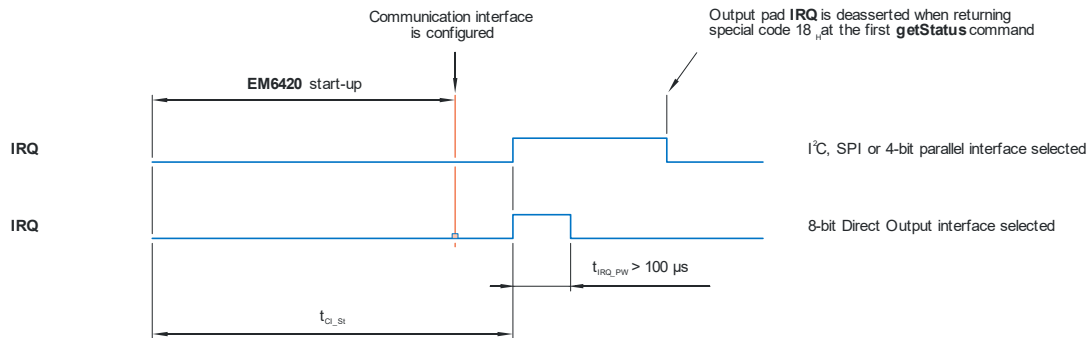


Figure 8-8 : **IRQ** output startup timings when positive **IRQ** polarity is selected by embedded software

When the negative **IRQ** polarity is used, the open-drain with internal pull-up resistor configuration is selected by the embedded software for the **IRQ** output pad and the timings shown in Figure 8-9 are generated at startup.

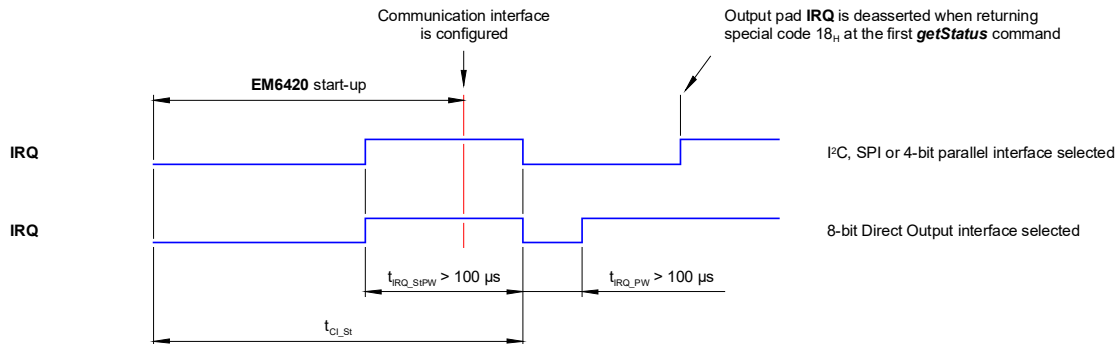


Figure 8-9 : **IRQ** output pad start-up timings when negative **IRQ** polarity is selected

8.3 EM6420 COMMANDS

Communication commands interpreted by the **EM6420** can be grouped into 3 command sets:

- The first command set includes all single byte commands, as described in Table 8-7. These commands are mainly used to send a new parameter value to the **EM6420**.

Command name	Command description
<u>startTS</u>	Start the Touch Screen interface
<u>stopTS</u>	Stop the Touch Screen interface
<u>setTSMODE</u>	Select the Touch Screen running features
<u>selectBaseSettings</u>	Select the base settings as the current settings
<u>selectAltSettings</u>	Select the alternate settings as the current settings
<u>setBaseScanFreq</u>	Set the Touch Screen base scan frequency
<u>setAltScanFreq</u>	Set the Touch Screen alternate scan frequency
<u>setBaseHiSensNb</u>	Set the base highest sensor number to be scanned
<u>setAltHiSensNb</u>	Set the alternate highest sensor number to be scanned
<u>setBaseIRQCond</u>	Set the base IRQ condition
<u>setAltIRQCond</u>	Set the alternate IRQ condition
<u>next</u>	Request the next data byte within a multiple data byte read sequence (SPI interface only)
<u>end</u>	End a multiple data byte read sequence (SPI interface only)

Table 8-7 : **EM6420** single byte command set

- The second command set includes two-byte commands, as described in the following table. These commands are used to get any parameter value from the **EM6420** or to send a more than 4-bit parameter value to the **EM6420**.

Command name	Command description
<u>setThreshold</u>	Set Touch Screen threshold
<u>getAppSettings</u>	Get current application settings

Table 8-8 : **EM6420** two-byte command set

- The third command set includes multiple byte commands, as described in the following table. These commands are used to get multiple parameters from or to send multiple parameters to the **EM6420**, thus reducing communication traffic and overall system consumption.

Command name	Command description
<u>getVersion</u>	Get EM6420 HW and SW version
<u>getStatus</u>	Get EM6420 status

Table 8-9 : **EM6420** multiple bytes command set

8.3.1 Command startTS

This command starts the **EM6420** Touch Screen interface.

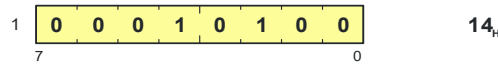


Figure 8-10 : **startTS** command format

At power-up, the Touch Screen interface is stopped. Some settings must be defined before the Touch Screen is started, and cannot be changed later on. Therefore, the commands **setTSMODE**, **setBaseScanFreq**, **setBaseHiSensNb**, **setBaseIRQCond** and **setThreshold**, if used, must be sent before the **startTS** command. Attempting to send one of these commands while the Touch Screen interface is running will cause the **EM6420** to assert its **IRQ** output and return error code 06_H.

The Touch Screen base settings are checked when the Touch Screen interface is started. If they are invalid, the **EM6420** asserts its **IRQ** output and returns error code 02_H.

During the Touch Screen startup, the **EM6420** checks the presence of each sensor and establishes a sensor map. If the total number of sensors wasn't specified before the **startTS** command (with the **setBaseHiSensNb** command), the device scans all sixteen sensors and determines the number of sensors by itself. To be valid, the sensor map must have at least sensor 0 connected and there must be no lack between the first and the last used sensor. If the highest sensor number was specified, the following sensors will never be scanned. If the **EM6420** detects a problem with the sensor map, it asserts its **IRQ** output and returns error code 01_H.

If the alternate settings are already selected before the **startTS** command is used, that is if the command **selectAltSettings** was sent before the **startTS** command, the **EM6420** will automatically apply the alternate settings immediately after the Touch Screen startup. However, initialization of the sensors is always performed according to the base settings. Sending the **startTS** command while the Touch Screen interface is already running will restart it according to the base settings for initialization, and then it will use the previous selected settings.

Sensors that are already activated during the Touch Screen startup procedure will not be detected until they are released and the **EM6420** has had enough time to initialize them properly.

8.3.2 Command stopTS

This command stops the **EM6420** Touch Screen interface.

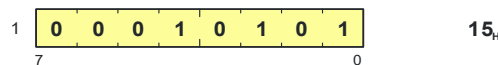


Figure 8-11 : **stopTS** command format

This command has no effect if the Touch Screen interface is already stopped. Stopping the Touch Screen interface allows the host to change the base settings with the commands **setTSMODE**, **setBaseScanFreq**, **setBaseHiSensNb**, **setBaseIRQCond** and **setThreshold**. However, if some sensors are activated during the next startup procedure (initiated by the **startTS** command), they will not be detected by the **EM6420** until they are released and the **EM6420** has had enough time to initialize them properly.

8.3.3 Command setTSMODE

This command sets the **EM6420** Touch Screen running features.

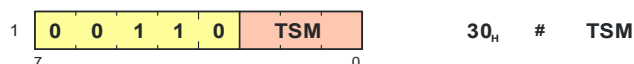


Figure 8-12 : **setTSMODE** command format

Table 8-10 gives the mapping between the **TSM** bits and the selected Touch Screen running features.

Bit	Behavior if bit is set	Behavior if bit is cleared
0	EM6420 works in Ultra Low Power mode	EM6420 works in Low Power mode
1	Each sensor sensitivity is continuously optimized, taking into account actual room temperature and supply voltage	Each sensor sensitivity is optimized only when the Touch Screen interface is started
2	Each sensor has its own activation threshold which is continuously adapted to sensor sensitivity	All sensors have the same activation threshold which is a fixed value

Table 8-10 : Mapping between the **TSM** bits and the Touch Screen running features

The Touch Screen running features must be defined before starting the Touch Screen interface. By default, the **TSM** bits values are 111_B when 8-bit direct output interface is selected and 110_B when any other communication interface is selected⁸. Attempting to modify the Touch Screen running features while the Touch Screen interface is running will cause the **EM6420** to assert its **IRQ** output and return the error code 06_H.

8.3.4 Command *selectBaseSettings*

This command selects the basic settings as the Touch Screen current settings.

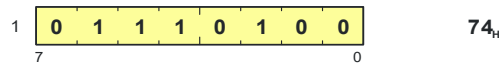


Figure 8-13 : *selectBaseSettings* command format

The Touch Screen base settings can be defined by **setBaseScanFreq**, **setBaseHiSensNb** and **setBaseIRQCond** commands. If used, these commands must be sent before the Touch Screen interface is started. If they aren't used, default values are supplied for the base settings. At power-up, the base settings (as opposed to the alternate settings) are selected, so that this command is only needed after a **selectAltSettings** command, in order to switch back to the base settings.

The Touch Screen base settings are checked when the Touch Screen interface is started. If they are invalid, the **EM6420** asserts its **IRQ** output and returns error code 02_H. Attempting to change the base settings while the Touch Screen interface is running will cause the **EM6420** to assert its **IRQ** output and return error code 06_H.

It is possible to switch from the alternate settings to the base settings at any time, even when the Touch Screen interface is running. In that case, the base settings are applied immediately. This command has no effect if the base settings are already selected.

8.3.5 Command *selectAltSettings*

This command selects the alternate settings as the Touch Screen current settings.

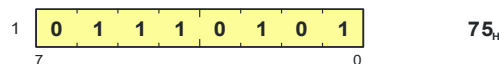


Figure 8-14 : *selectAltSettings* command format

The Touch Screen alternate settings are defined by **setAltScanFreq**, **setAltHiSensNb** and **setAltIRQCond** commands. They can be modified when the Touch Screen interface is stopped as well as when it's running even if some other alternate settings are currently active.. However, the new alternate settings will not take effect immediately after one of these three

⁸ Please contact EM Microelectronic-Marin SA to change the default values of **TSM** bits 1 and 2

commands. Instead, if the Touch Screen interface is running, the new alternate settings will be applied at the next occurrence of a **setAltSettings** command. If the Touch Screen interface is stopped and the alternate settings are already selected, they will be applied directly after the next **startTS** command.

Each alternate parameter that has never been explicitly set through the appropriate command, when applied, will be substituted by the corresponding base settings. Therefore, toggling between base and alternate settings without having ever sent any of the three **setAlt...** commands won't have any effect.

The validity of the new alternate settings is checked when they are applied, that is either after a **selectAltSettings** or a **startTS** command. If they are invalid, the **EM6420** asserts its **IRQ** output and returns error code 03_H.

8.3.6 Command setBaseScanFreq

This command sets the basic scan frequency of Touch Screen sensors.

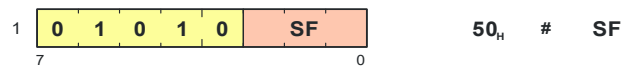


Figure 8-15 : **setBaseScanFreq** command format

Table 8-11 lists valid values for parameter **SF**. Note that a 64 Hz or 128 Hz scan frequency can only be used with a reduced number of sensors, i.e. 8 at 64 Hz and 4 at 128 Hz.

SF	Touch Screen scan frequency	SF	Touch Screen scan frequency
000 _B	1 Hz	100 _B	16 Hz
001 _B	2 Hz	101 _B	32 Hz
010 _B	4 Hz	110 _B	64 Hz
011 _B	8 Hz	111 _B	128 Hz

Table 8-11 : Selection of Touch Screen scan frequency

The **setBaseScanFreq** command, if used, must be sent before the Touch Screen interface is started. If the **setBaseScanFreq** command isn't used, the base scan frequency is 8 Hz by default.

Attempting to modify the base scan frequency while the Touch Screen interface is running will cause the **EM6420** to assert its **IRQ** output and return error code 06_H.

8.3.7 Command setAltScanFreq

This command sets the alternate scan frequency of Touch Screen sensors.



Figure 8-16 : **setAltScanFreq** command format

Table 8-12 lists valid values for parameter **ASF**. Note that a 64 Hz or 128 Hz scan frequency can only be used with a reduced number of sensors, i.e. 8 at 64 Hz and 4 at 128 Hz. Moreover, the alternate scan frequency cannot be greater than the base scan frequency. The **EM6420** will assert its **IRQ** output and return error code 03_H if these conditions are not met when the alternate settings are applied.

ASF	Touch Screen scan frequency	ASF	Touch Screen scan frequency
000 _B	1 Hz	100 _B	16 Hz
001 _B	2 Hz	101 _B	32 Hz
010 _B	4 Hz	110 _B	64 Hz
011 _B	8 Hz	111 _B	128 Hz

Table 8-12 : Selection of alternate Touch Screen scan frequency

The alternate scan frequency can be modified at any time, even if some other Touch Screen alternate settings are already selected. However, the new alternate scan frequency will be taken into account only next time a **setAltSettings** command is issued. By default, if the **setAltScanFreq** command has never been sent, the alternate scan frequency is the same as the base scan frequency (no change in scan frequency when switching from base to alternate settings).

8.3.8 Command setBaseHiSensNb

This command sets the highest sensor number to be scanned when base settings are selected. Sensors are numbered from 0 to 15.



Figure 8-17 : setBaseHiSensNb command format

Valid values for parameter **HSN** range from 0 to 15, allowing the host to select from one to sixteen sensors.

The base highest sensor number can only be defined before the Touch Screen interface is started. Attempting to modify the base highest sensor number while the Touch Screen interface is already running will cause the **EM6420** to assert its **IRQ** output and return error code 06_H.

If the base highest sensor number hasn't been defined when the Touch Screen interface is started, the **EM6420** will determine the number of connected sensors by itself. Otherwise, if the number of sensors has been defined, it will check that these sensors are effectively connected. An error in the sensor map (due to inappropriate settings or to sensors failure) will cause the **EM6420** to assert its **IRQ** output and generate error 01_H at the next **startTS** command.

8.3.9 Command setAltHiSensNb

This command sets the alternate highest sensor number to be scanned.



Figure 8-18 : setAltHiSensNb command format

Valid values for parameter **ASHN** range from 0 to 15, allowing the host to select from one up to sixteen sensors. The alternate highest sensor number must not be greater than the highest sensor number selected in the base settings.

The alternate highest sensor number can be modified at any time, even if some other alternate settings are currently being used. However, the new alternate highest sensor number will be taken into account only next time a **setAltSettings** command is issued. By default, if the **setAltHiSensNb** command has never been sent, the alternate sensor number is the same as the base sensor number (no change in number of sensors when switching from base to alternate settings).

8.3.10 Command setBaseIRQCond

This command sets the basic Touch Screen IRQ condition.

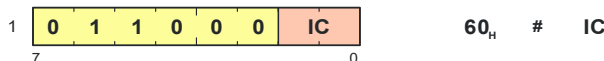


Figure 8-19 : setBaseIRQCond command format

Table 8-13 gives the mapping between IC parameter values and the selected Touch Screen IRQ condition.

IC	Touch Screen IRQ condition
00 _B	EM6420 output pad IRQ is asserted at the end of each sensors scan
01 _B	EM6420 output pad IRQ is asserted at the end of a sensors scan, when at least one sensor state has changed
10 _B	EM6420 output pad IRQ is asserted at the end of a sensors scan, when either at least one sensor is active or at least one sensor state has changed
11 _B	EM6420 output pad IRQ is asserted at the end of a sensors scan, when the most activated sensor has changed

Table 8-13 : Selection of EM6420 IRQ condition

The Touch Screen IRQ condition defines under which circumstances the EM6420 will assert its IRQ output to signal events happening on the Touch Screen. The IRQ output remains asserted until the new Touch Screen state is returned in response to a getStatus command.

The base Touch Screen IRQ condition should be defined before starting the Touch Screen interface. Two separate default values exist when 8-bit direct output interface is selected: the default value is 11_B when MAS input is asserted and 01_B when MAS input is deasserted or when any other communication interface is selected⁹.

Attempting to modify the base IRQ condition while the Touch Screen interface is running will cause the EM6420 to assert its IRQ output and return error code 06_H.

8.3.11 Command setAltIRQCond

This command set the alternate Touch Screen IRQ condition.

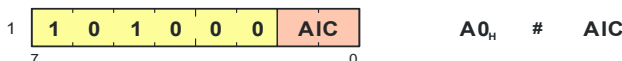


Figure 8-20 : setAltIRQCond command format

Table 8-14 gives the mapping between AIC parameter values and the selected Touch Screen IRQ condition.

When alternate settings are selected, the alternate Touch Screen IRQ condition defines under which circumstances the EM6420 will assert its IRQ output to signal events happening on the Touch Screen. The IRQ output remains asserted until the new Touch Screen state is returned in response to a getStatus command.

The alternate Touch Screen IRQ condition can be modified at any time, even if some other alternate settings are currently being used. However, the new alternate IRQ condition will be taken into account only next time a setAltSettings command is issued. By default, if the setAltIRQCond command has never been sent, the alternate IRQ condition is the same as the base IRQ condition (no change in IRQ generation mode when switching from base to alternate settings).

⁹ Please contact EM Microelectronic-Marin SA to change the default Touch Screen IRQ condition values

AIC	Touch Screen IRQ condition
00 _B	EM6420 output pad IRQ is asserted at the end of each sensors scan
01 _B	EM6420 output pad IRQ is asserted at the end of a sensors scan, when at least one sensor state has changed
10 _B	EM6420 output pad IRQ is asserted at the end of a sensors scan, when either at least one sensor is active or at least one sensor state has changed
11 _B	EM6420 output pad IRQ is asserted at the end of a sensors scan, when the most activated sensor has changed

Table 8-14 : Selection of **EM6420** alternate IRQ condition

When alternate settings are selected, the alternate Touch Screen IRQ condition defines under which circumstances the **EM6420** will assert its **IRQ** output to signal events happening on the Touch Screen. The **IRQ** output remains asserted until the new Touch Screen state is returned in response to a **getStatus** command.

The alternate Touch Screen IRQ condition can be modified at any time, even if some other alternate settings are currently being used. However, the new alternate IRQ condition will be taken into account only next time a **setAltSettings** command is issued. By default, if the **setAltIRQCond** command has never been sent, the alternate IRQ condition is the same as the base IRQ condition (no change in IRQ generation mode when switching from base to alternate settings).

8.3.12 Command next (SPI protocol only)

This is a dummy command that has to be sent while fetching all the bytes of a response, but the last one (see § 8.3.13 and also § 9.2). This command will request the next output byte to be prepared on the **EM6420**, thus indicating that the transfer isn't finished. The **next** command may not be used when the communication bus is idle and no response is expected. If the **EM6420** receives a **next** command in such circumstances, it will assert its **IRQ** output and return error code 05_H.

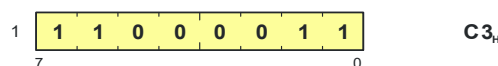


Figure 8-21 : **next** command format

This command can only be used when SPI communication interface is selected, and will cause the **EM6420** to assert its **IRQ** output and return error code 05_H if used with any other communication protocol.

8.3.13 Command end

This is a dummy command that has to be sent in order to fetch the last desired byte of a response and close the transfer. It indicates the end of a SPI communication frame and lets the **EM6420** stop sending data (see § 9.2)

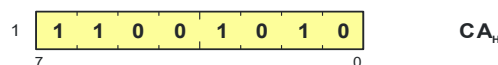


Figure 8-22 : **end** command format

It is not necessary to send an **end** for each intermediate command when several commands are chained. The following command code can be sent directly in place of the **end** code. That way, the last response byte to the previous command will be retrieved during the transfer of the following command code, and the reception of a new command will automatically close the previous one on the **EM6420** side.

The **end** command may not be used when the communication bus is idle and no response is expected. If the **EM6420** receives an **end** command in such circumstances, it will assert its **IRQ** output and return error code 05_H. Moreover, this command can only be used in SPI mode, and will cause the **EM6420** to assert its **IRQ** output and return error code 05_H if used with any other communication protocol.

8.3.14 Command setThreshold

This command sets the initial sensor activation threshold.

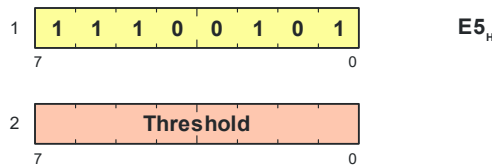


Figure 8-23 : **setThreshold** command format

Threshold values must be in the range from 3 to 200. Invalid values will cause the **EM6420** to assert its **IRQ** output and return error code 04_H. The default threshold value is 6¹⁰.

The threshold value can only be defined before the Touch Screen interface is started. Attempting to modify that value while the Touch Screen interface is already running will cause the **EM6420** to assert its **IRQ** output and return error code 06_H.

8.3.15 Command getAppSettings

This command gets the current application settings, i.e. the current Touch Screen scan frequency and the current highest scanned sensor number.

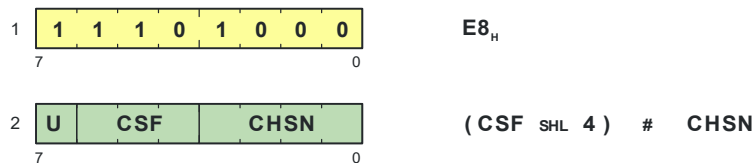


Figure 8-24 : **getAppSettings** command format

Valid values for parameter **CHSN** range from 0 to 15, thus indicating a number of scanned sensors comprised between 1 and 16. Note that the current sensor number could also be unknown, if it hasn't been specified by the host processor and the Touch Screen has not been started yet. In that case, the flag **U** is set, and **CHSN** contains the highest possible sensor number, depending on the scan frequency, i.e. 15 at scan frequencies up to 32 Hz, 7 at 64 Hz and 3 at 128 Hz.

Table 8-15 lists valid values for parameter **CSF**. Note that a 64 Hz or 128 Hz scan frequency can only be obtained with a reduced number of sensors, i.e. 8 at 64 Hz and 4 at 128 Hz.

CSF	Touch Screen scan frequency	CSF	Touch Screen scan frequency
000 _B	1 Hz	100 _B	16 Hz
001 _B	2 Hz	101 _B	32 Hz
010 _B	4 Hz	110 _B	64 Hz
011 _B	8 Hz	111 _B	128 Hz

Table 8-15 : **Current Touch Screen scan frequency**

¹⁰ Please contact EM Microelectronic-Marin SA to change the default threshold value

8.3.16 Command *getVersion*

This command gets the hardware as well as the software version of the **EM6420**.

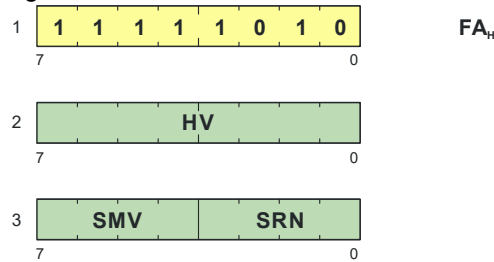


Figure 8-25 : *getVersion* command format

The hardware version is a single 8 bits value. The software version is composed of two nibbles: the **SMV** nibble is the software major version and the **SRN** nibble is the software revision number.

8.3.17 Command *getStatus*

This command gets the **EM6420** status and deasserts the **IRQ** output pad, if asserted. If the **IRQ** was not asserted, the *getStatus* will return null bytes.

NB: For better performance, do not send *getStatus* requests while no **IRQ** is asserted. In particular, do not try to read the Touch Screen status by sending this command repeatedly, as it will only slow down the **EM6420** and increase its power consumption.

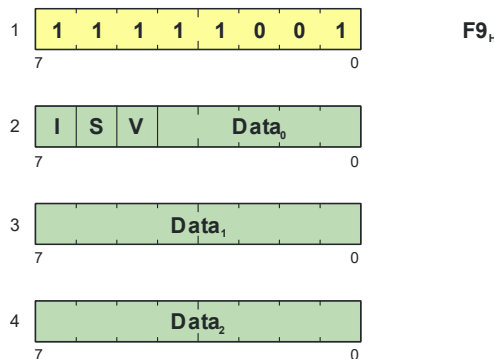


Figure 8-26 : *getStatus* command format

Flag **I** (Interrupt) is set if the **EM6420 IRQ** output was asserted when a *getStatus* command was received. In a multi **EM6420** configuration (all **IRQ** output pads connected to a unique host IRQ input) this flag allows the host to determine which **EM6420** device has asserted the IRQ line.

When flag **I** is set, flag **S** (**S**pecial) defines which kind of information is returned to the host processor:

When flag **S** is cleared, a Touch Screen IRQ condition occurred. In this case, parameter **Data₀** indicates the number of the most activated sensor, parameter **Data₁** gives the state of sensors 0 to 7 and parameter **Data₂** gives the state of sensors 8 to 15, as shown in Figure 8-27.

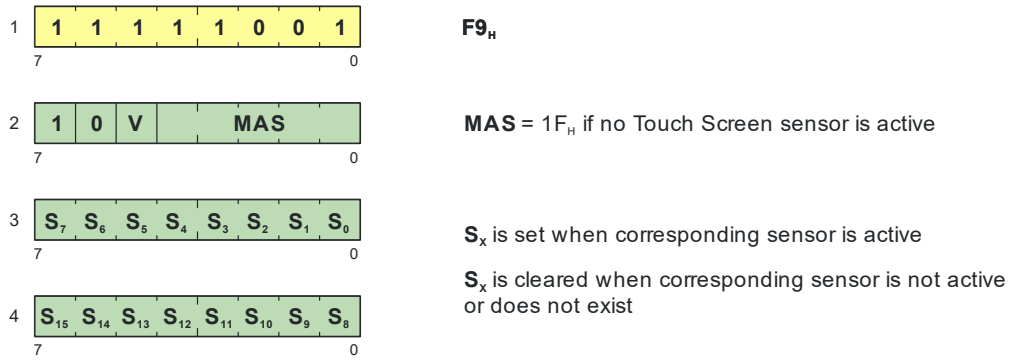


Figure 8-27 : **getStatus** command format when flag **I** is set and flag **S** is cleared

When command parameter **S** is set, the **EM6420** returns device status information, mainly error codes, but also the **READY** status after device startup. In this case, parameter **Data₀** gives a special code which valid values are listed in Table 8-16. For some special codes, parameters **Data₁** and **Data₂** may contain more information as shown in Figure 8-28. For special codes that do not provide complementary information, these parameters are null.

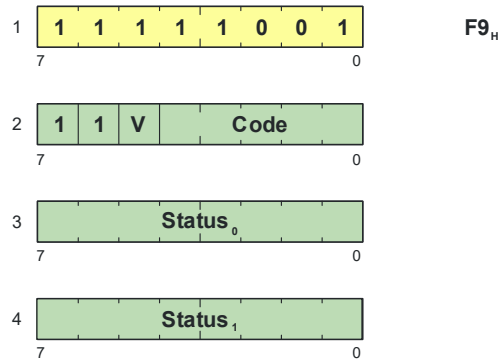


Figure 8-28 : **getStatus** command format when flags **I** and **S** are set

The flag **V** (oVerrun) indicates that the host has missed one or more significant status messages. This happens when the host takes too much time to react to an **IRQ**, and the **EM6420** wants to signal another event while the **IRQ** line is still active. In this case, the previous important message is deleted, and the host won't be able to retrieve it anymore. It will only have an indication, through the overflow bit, that at least one message was lost. In case of an overrun, the retrieved message is the most recent message containing the **S** flag. If none of them contains the **S** flag, it is the most recent Touch Screen message.

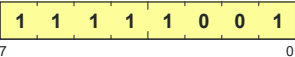
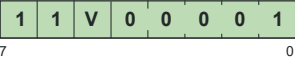
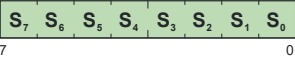
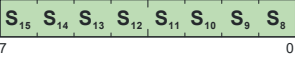
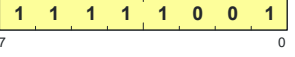
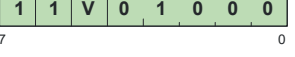
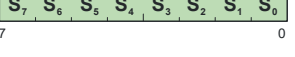
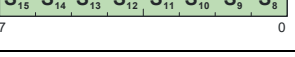
Code	Special code description
01 _H	<p>At least one Touch Screen sensor is not correctly connected to the EM6420 device. Command parameters Status₀ and Status₁ show which sensors seem to be connected.</p> <p>1  F9_H</p> <p>2 </p> <p>3  S_x is cleared when corresponding sensor seems to be defect S_x is set when corresponding sensor seems to be connected</p> <p>4 </p>
02 _H	Base settings are invalid. These settings are checked when the Touch Screen interface is started
03 _H	Alternate settings are invalid. These settings are checked once selected as current settings or when the Touch Screen interface is started, if alternate settings are already selected at this moment
04 _H	Bad initial sensor activation threshold
05 _H	Unexpected command received
06 _H	Parameter modification not allowed. Command parameter Status₀ returns the command code
07 _H	Reserved for debug purposes
08 _H	<p>Major overflow occurred. Command parameters Status₀ and Status₁ show which sensor has generated the major overflow. Normally, only one bit is set</p> <p>1  F9_H</p> <p>2 </p> <p>3  S_x is cleared when corresponding sensor is working properly S_x is set when corresponding sensor has generated a major overflow</p> <p>4 </p>
0F _H	Unknown command received. Command parameter Status₀ returns the command code
10 _H	Reserved for debug purposes
11 _H	Reserved for debug purposes
18 _H	EM6420 is READY to accept communication frames

Table 8-16 : **EM6420** special codes description

9. EM6420 COMMUNICATION FRAMES

All **EM6420** commands may be sent by the host using one of the two following communication frames:
A Write-Only (WO) communication frame is used to send commands that do not return any values to the host.

- A Write-Read (WR) communication frame is used to send commands that return one or more values to the host.
- Depending of the selected communication interface, these two communication frames may slightly differ, as explained in details hereafter.

9.1 SLAVE I²C COMMUNICATION FRAME

Each I²C communication frame must begin with a START condition followed by an I²C header and must end with a STOP condition.

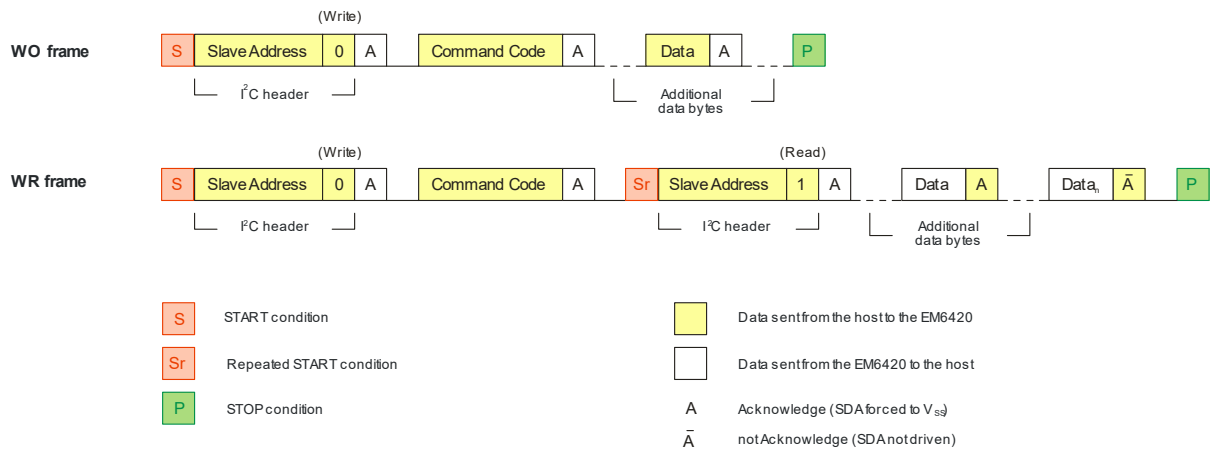


Figure 9-1: I2C WO and WR communication frames

When several commands are sent to the same EM6420 device, the STOP condition is not necessarily be generated between the concatenated communication frames.

9.2 SLAVE SPI COMMUNICATION FRAME

As the SPI is a full duplex interface, WO and WR communication frames looks quite the same. However, the host should ignore the values returned by the **EM6420** in a WO communication frame.

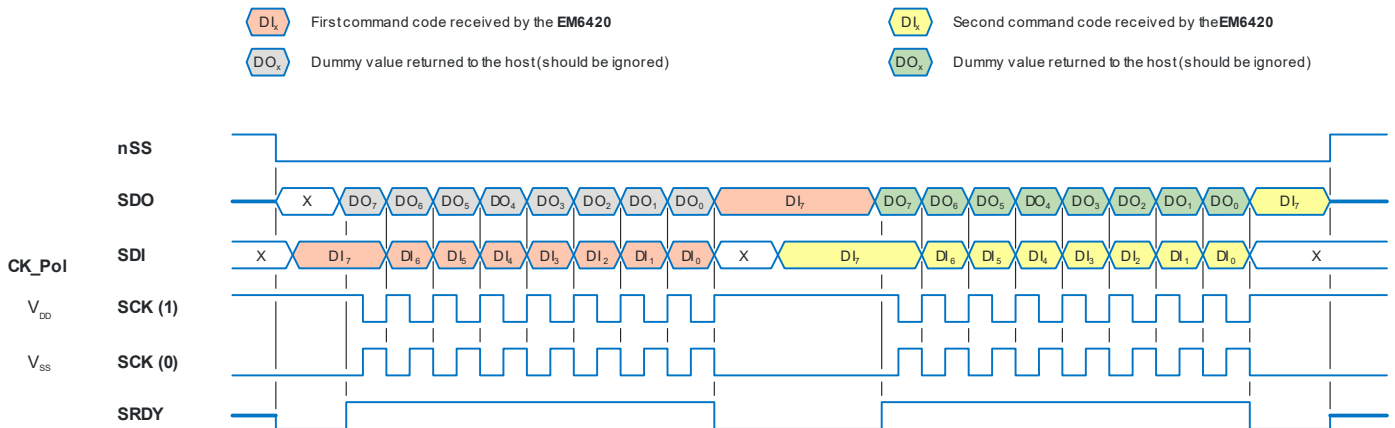


Figure 9-2: SPI WO communication frames when CK_Pha input is set to VSS

- DI_k First command code received by the EM6420
- DO_x Dummy value returned to the host (should be ignored)
- DI_k Second command code received by the EM6420
- DO_y Dummy value returned to the host (should be ignored)

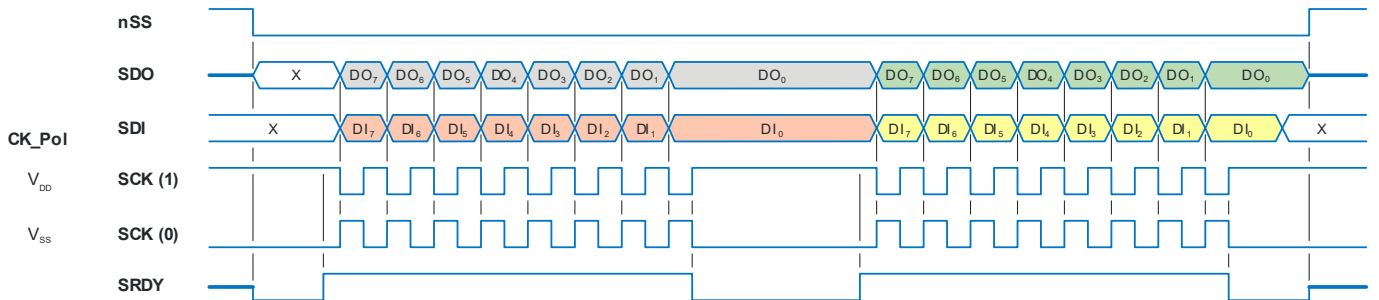


Figure 9-3: SPI WO communication frames when CK_Pha input is set to VDD

In a WR communication frame, the EM6420 return the requested values as long as it receives a **next** command code. Receiving any other command code will terminate the current command and immediately start the new one.

- DI_k First command code received by the EM6420
- NX_x Next command code
- DI_k Second command code received by the EM6420
- DI_k Parameter value associated to the second command
- DO_x Dummy value returned to the host (should be ignored)
- DO₀ First value returned to the host in response to the first command
- DO₁ Second value returned to the host in response to the first command
- DO₂ Third value returned to the host in response to the first command
- DO₃ Fourth value returned to the host in response to the first command
- DO₀ Dummy value returned to the host (should be ignored)

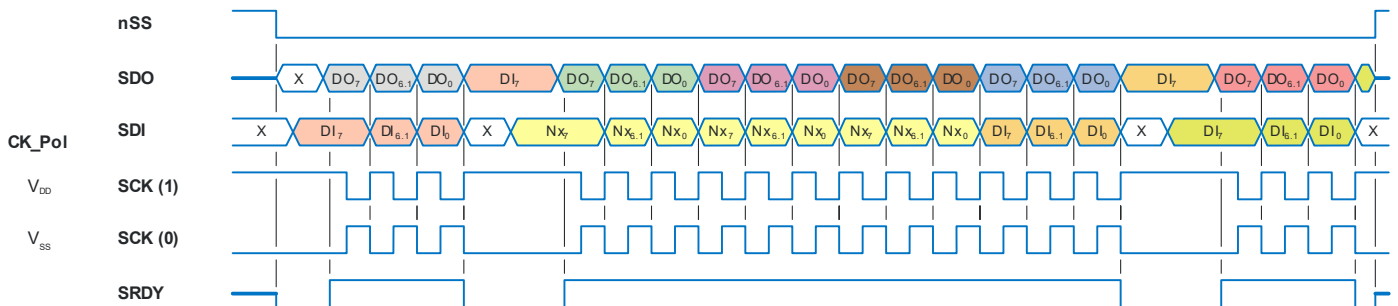


Figure 9-4: SPI WR followed by a WO communication frame when CK_Pha input is set to VSS

The nSS input of the EM6420 does not need to be deasserted between two communication frames.

9.3 SLAVE 4-BIT PARALLEL COMMUNICATION FRAME

The CE input of the EM6420 does not need to be deasserted between two communication frames.

- DI_k First command code received by the EM6420
- DI_k Second command code received by the EM6420
- DI_k Parameter value associated to the first command code
- DI_k Third command code received by the EM6420

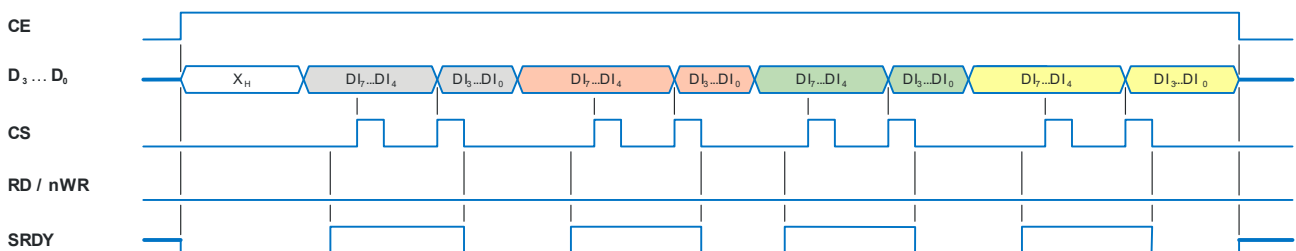


Figure 9-5 : Multiple 4-bit parallel WO communication frames

The falling edge of the RD / nWR input defines the end of a WR communication frame, and therefore the end of the current command.

- DI_k First command code received by the EM6420
- DO_x First data byte returned by the EM6420
- DI_x Second command code received by the EM6420
- DO_x Second data byte returned by the EM6420

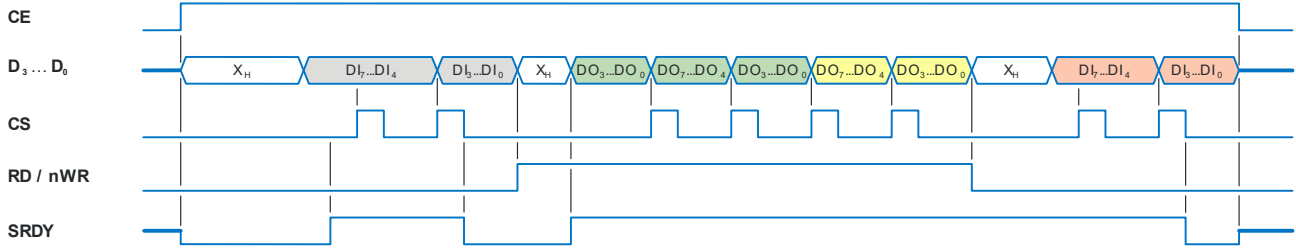
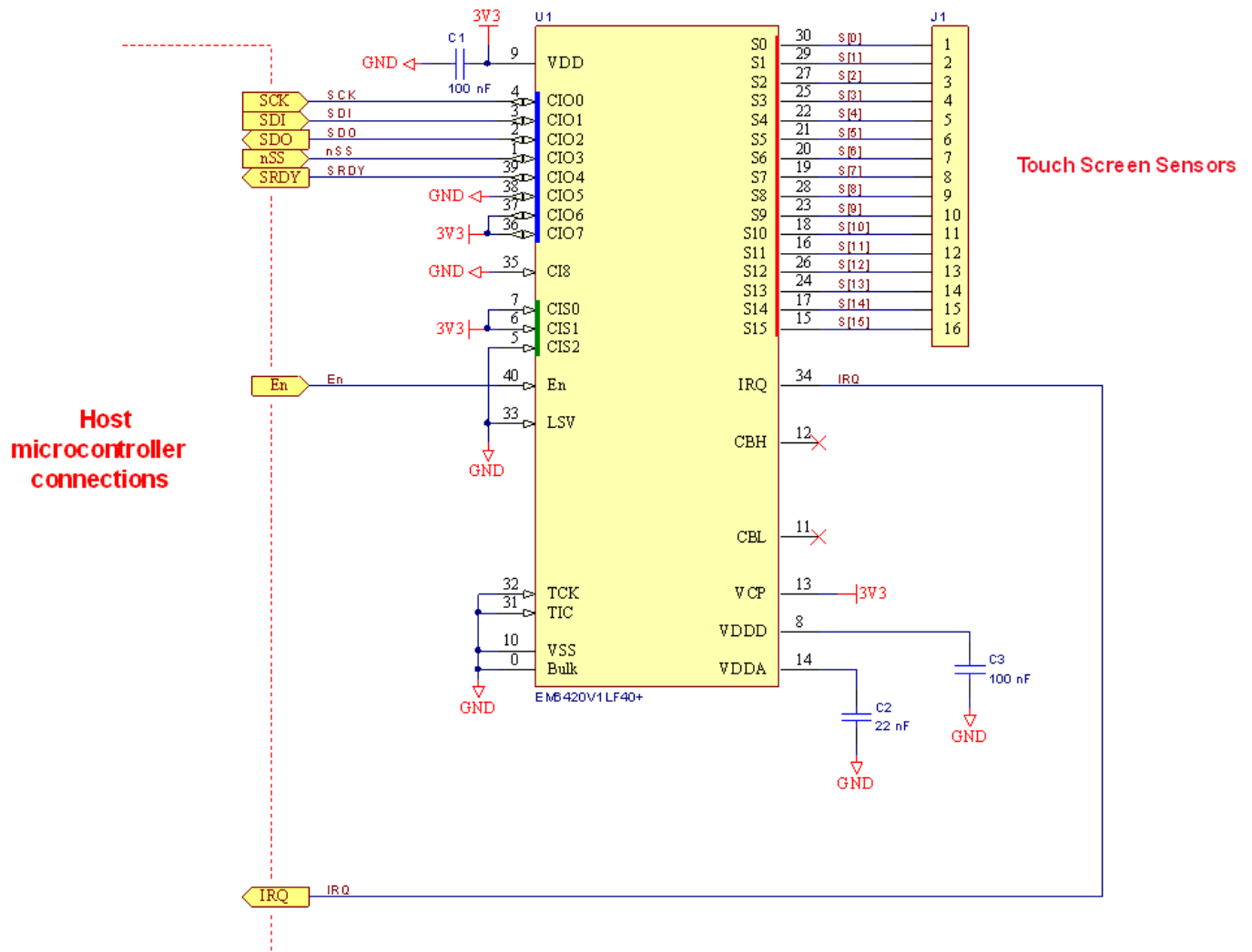


Figure 9-6: 4-bit parallel WR followed by a WO communication frame

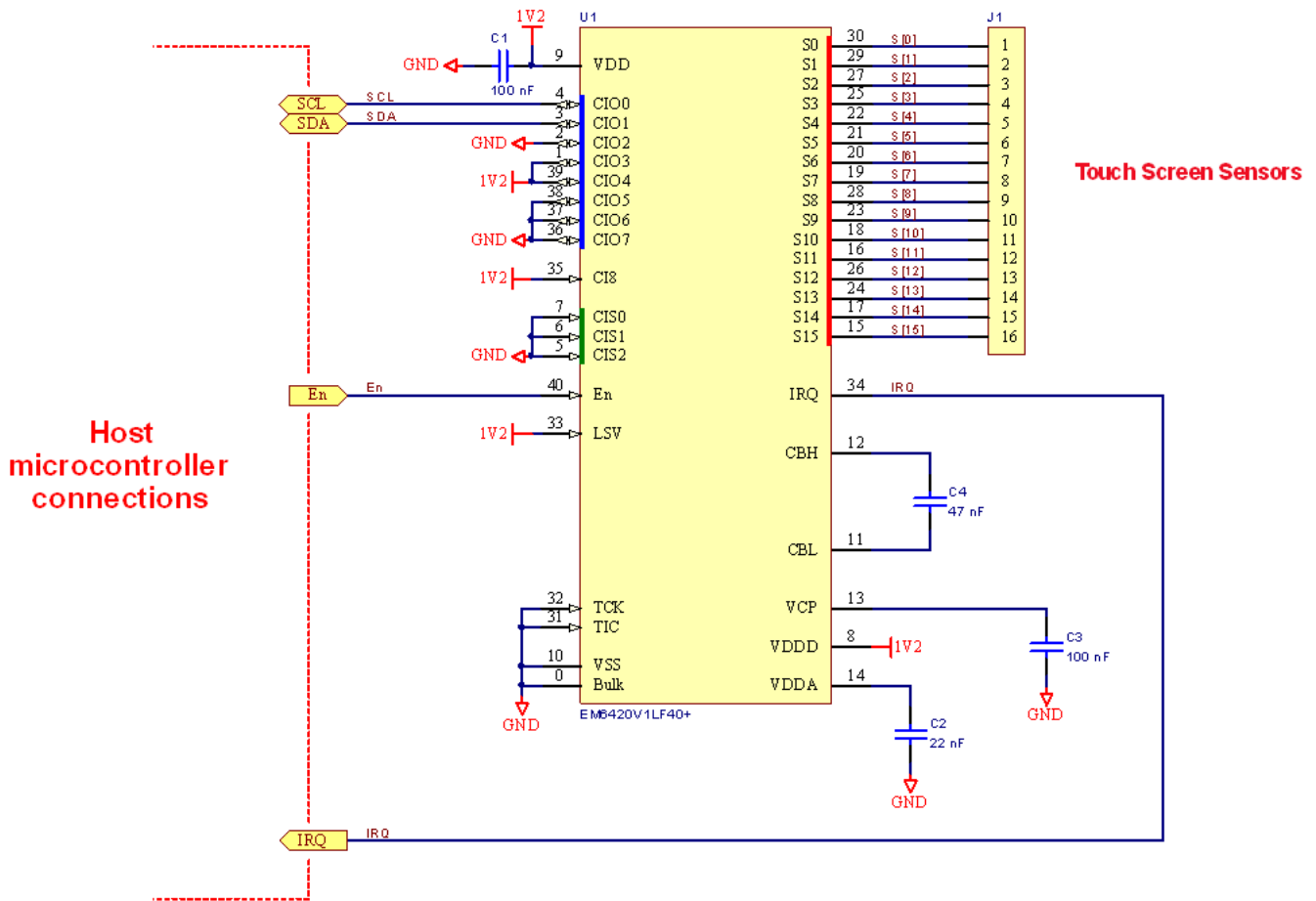
10. TYPICAL APPLICATIONS



SPI interface settings :

- SCK clock is set to V_{SS} between transmissions (**CK_Pol** [CI_0] input connected to V_{SS})
- Data on SDI input is latched at the second SCK clock edge (**CK_Pha** [CI_0] input connected to V_{DD})
- Data bytes are sent MSB first (**MSB_First** [CI_0] input connected to V_{DD})
- Negative IRQ polarity is selected (**IRQ_Pol** [CI_0] input connected to V_{SS})

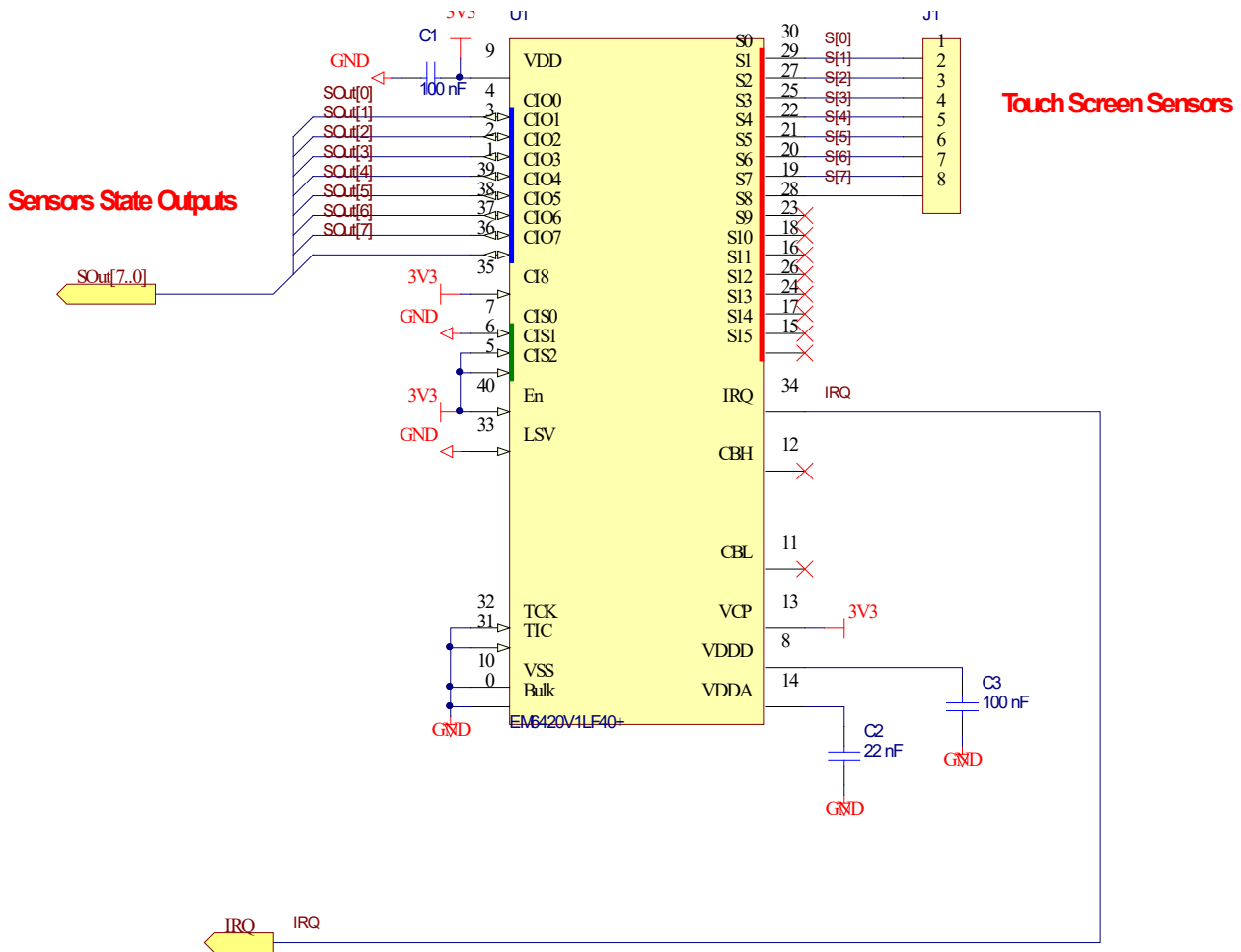
Figure 10-1: **EM6420** typical application powered by a 3.3 V supply voltage and using the SPI communication interface



I2C interface settings :

- Internal strong pull-up resistors are enabled on bidir pads **SDA** and **SCL** (**En_IWPU** [**CIO₂**] input connected to **V_{SS}** and **En_ISPU** [**CIO₃**] input connected to **V_{DD}**)
- 7-bit I2C address is set to 41_h (**A₃ ... A₁** [**CIO₇ ... CIO₅**] inputs connected to **V_{SS}** and **A₀** [**CIO₄**] input connected to **V_{DD}**)
- Positive IRQ polarity is selected (**IRQ_PoI** [**CI₁**] input connected to **V_{DD}**)

Figure 10-2: EM6420 typical application powered by a 1.2 V supply voltage and using the I2C communication interface



Standalone settings :

- Scan frequency = 32 Hz (CIS_s input connected to V_{SS} and CIS_i input connected to V_{DD})
- Most Activated Feature selected (MAS [C₁₈] input connected to V_{DD})
- IRQ generated each time the Most Activated Sensor has changed

Figure 10-3: EM6420 typical standalone application powered by a 3.3 V supply voltage

For better ESD protection in customer application, it is strongly recommended to connect the bulk of the EM6420 to V_{SS}.

11. PAD LOCATION DIAGRAM

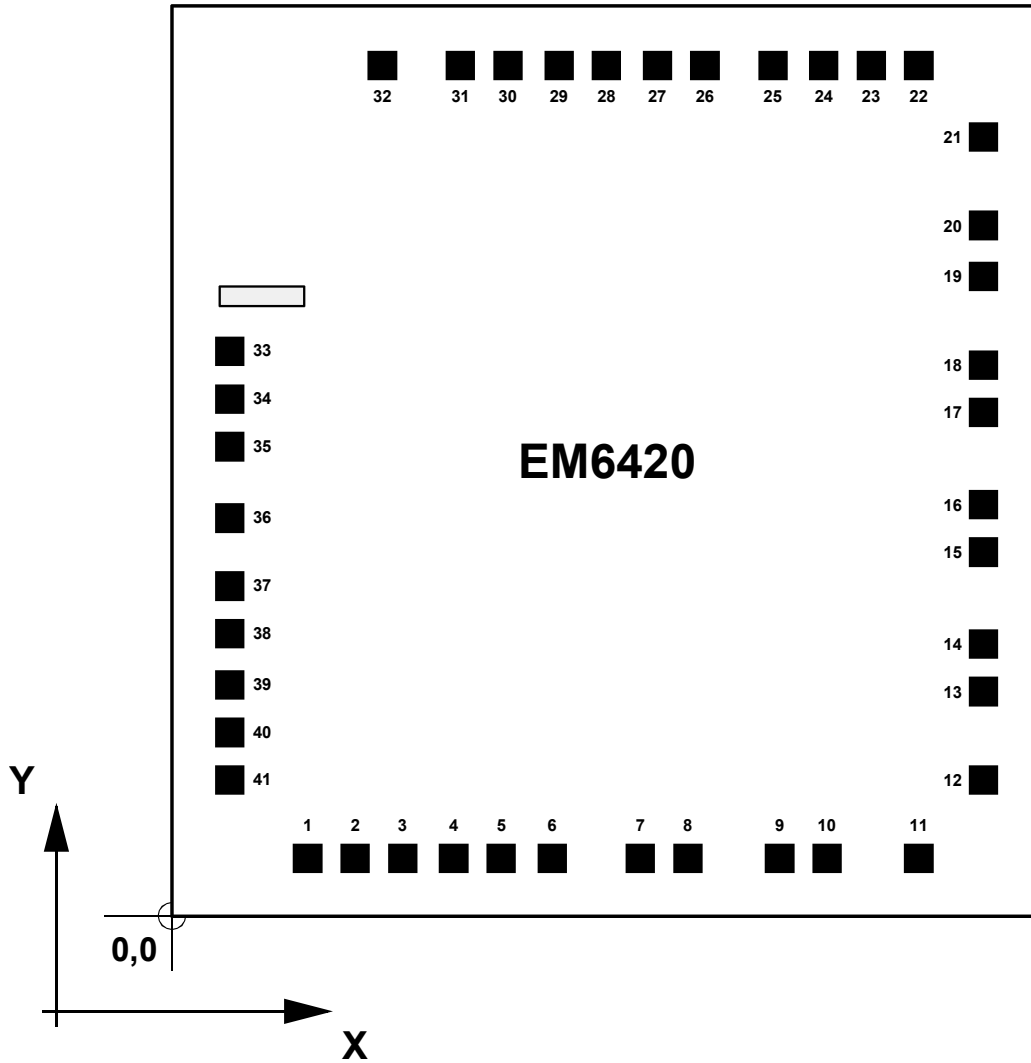


Figure 11-1 : **EM6420** pad location diagram (TOP view)

Chip dimensions		
Die size :	X = 2'130 μm \pm 100 μm	(83.86 mils \pm 3.94 mils)
	Y = 2'224 μm \pm 100 μm	(87.56 mils \pm 3.94 mils)
Die thickness :	279.4 μm \pm 25.4 μm	(10 mils \pm 1 mils)

Table 11-1: **EM6420** chip dimensions

Pad Number	Pad Name	X [μm]	Y [μm]	Pad Number	Pad Name	X [μm]	Y [μm]
1	V _{SS}	312.000	143.625	22	TIC	1'818.000	2'080.375
2	C _{BL}	432.000	143.625	23	TCK	1'698.000	2'080.375
3	C _{BH}	552.000	143.625	24	LSV	1'578.000	2'080.375
4	V _{CP}	672.000	143.625	25	IRQ	1'458.000	2'080.375
5	V _{DDA}	792.000	143.625	26	CI ₈	1'290.000	2'080.375
6	S ₁₅	912.000	143.625	27	CIO ₇	1'170.000	2'080.375
7	S ₁₁	1'132.000	143.625	28	CIO ₆	1'050.000	2'080.375
8	S ₁₄	1'252.000	143.625	29	CIO ₅	930.000	2'080.375
9	S ₁₀	1'472.000	143.625	30	CIO ₄	810.000	2'080.375
10	S ₇	1'592.000	143.625	31	En	690.000	2'080.375
11	S ₆	1'812.000	143.625	32	V _{DD}	500.000	2'080.375
12	S ₅	1'986.375	312.000	33	CIO ₃	143.625	1'368.000
13	S ₄	1'986.375	532.000	34	CIO ₂	143.625	1'248.000
14	S ₉	1'986.375	652.000	35	CIO ₁	143.625	1'128.000
15	S ₁₃	1'986.375	872.000	36	CIO ₀	143.625	960.000
16	S ₃	1'986.375	992.000	37	CIS ₂	143.625	792.000
17	S ₁₂	1'986.375	1'212.000	38	CIS ₁	143.625	672.000
18	S ₂	1'986.375	1'332.000	39	CIS ₀	143.625	552.000
19	S ₈	1'986.375	1'552.000	40	V _{DDD}	143.625	432.000
20	S ₁	1'986.375	1'672.000	41	V _{DD}	143.625	312.000
21	S ₀	1'986.375	1'892.000				

X, Y coordinates refers to the center of the pads. The origin (0, 0) is the bottom left corner of the circuit scribe line.

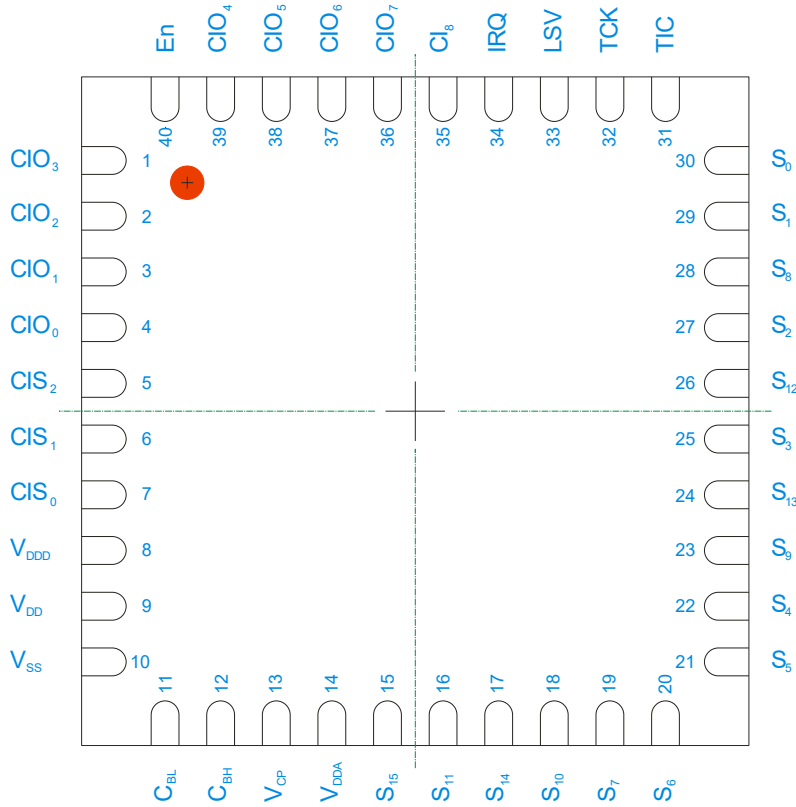
Table 11-2 : EM6420 pads coordinates

Standard die version		Gold bump version	
Pads opening	72 μm x 72 μm	Bump size :	68 μm x 68 μm \pm 5 μm
Minimum pad pitch :	120 μm	Bump height :	17.5 μm \pm 3 μm
		Bump height co-planarity	< 2 μm within die < 4 μm within wafer
		Bump roughness	< 2 μm
		Bump hardness :	30 – 90 HV (soft bump)
		Minimum bump space	52 μm edge to edge
		Shear force :	> 7.2 mg / μm^2
		PI thickness	No PI layer

Table 11-3 : EM6420 pads and gold bumps additional information

12. PACKAGE INFORMATION

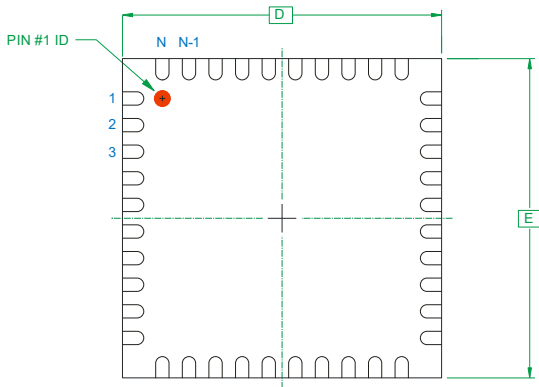
12.1 SAWN 40-PIN MICRO LEAD FRAME 2 – 6 X 6 MM BODY



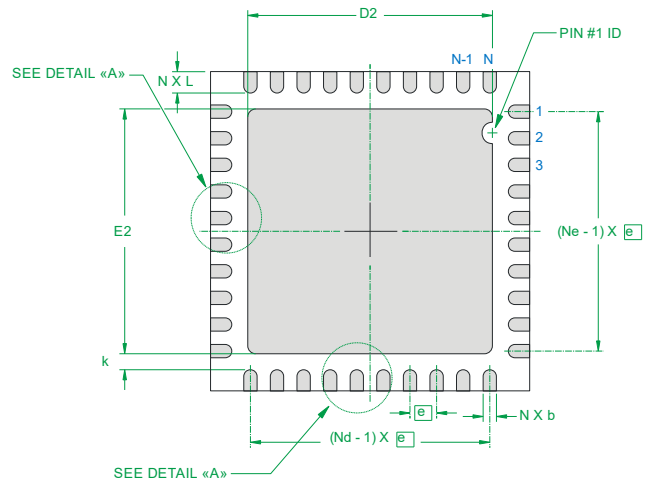
The exposed pad of the package is connected to the bulk of the device

Figure 12-1 : 40-pin Micro Lead Frame 2 Pin Assignment (TOP view)

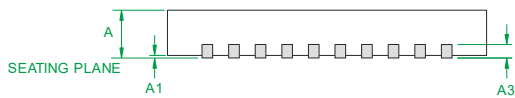
The exposed pad of the package is connected to the bulk of the device



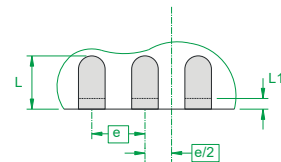
TOP VIEW



BOTTOM VIEW



SIDE VIEW



DETAIL «A»

Symbol	Dimensions		
	Min	Nom	Max
A	0.80	0.85	0.90
A1	0.00	0.02	0.05
A3	0.20 REF		
D	6.00 BSC		
D2	4.00	4.10	4.20
E	6.00 BSC		
E2	4.00	4.10	4.20

Symbol	Dimensions		
	Min	Nom	Max
N ¹¹	40		
Nd ¹²	10		
Ne ¹³	10		
e	0.50 BSC		
L	0.35	0.40	0.45
L1	0.00		0.15
b	0.18	0.25	0.30

All dimensions are in mm

¹¹ N is the number of terminals

¹² Nd is the number of terminals in X-direction

¹³ Ne is the number of terminals in Y-direction

13. ORDERING INFORMATION

Part number	Delivery Form
EM6420V4WS10	Sawn wafer, 10 mils thickness
EM6420V4WP10E	Sawn wafer with gold bumps, 10 mils thickness
EM6420V4LF40D+	40-pin sawn Micro Lead Frame 2 (40-pin MLF2), Tray
EM6420V4XXXX+	For other options please contact the EM Microelectronic-Marin SA sales representative.

Part number	Hardware version	Software version	I ² C Multi Chip mode	Minimum scan frequency
EM6420V4WS10 EM6420V4WP10E EM6420V4LF40D+ EM6420V4XXXX+	3	2-0	Supported	1 Hz

14. SPECIFICATION HISTORY

Rev.	E.C.N.	Pages	Date	Resp.	Description
1.0	25-1049	1, 49	20.08.2025	Schnider	New document naming, refreshed EM template Software version 2-0 Add history

EM Microelectronic-Marin SA ("**EM**") makes no warranties for the use of EM products, other than those expressly contained in EM's applicable General Terms of Sale, located at <http://www.emmicroelectronic.com>. EM assumes no responsibility for any errors which may have crept into this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein.

No licenses to patents or other intellectual property rights of EM are granted in connection with the sale of EM products, neither expressly nor implicitly.

In respect of the intended use of EM products by customer, customer is solely responsible for observing existing patents and other intellectual property rights of third parties and for obtaining, as the case may be, the necessary licenses.

Important note: The use of EM products as components in medical devices and/or medical applications, including but not limited to, safety and life supporting systems, where malfunction of such EM products might result in damage to and/or injury or death of persons is expressly prohibited, as EM products are neither destined nor qualified for use as components in such medical devices and/or medical applications. The prohibited use of EM products in such medical devices and/or medical applications is exclusively at the risk of the customer.