SQ52912

0.5°C Accuracy, JEDEC DDR5 Grade B, I3C/I²C Interface Digital Temperature Sensor in DSBGA Package

General Description

The SQ52912 is a high-accuracy temperature sensor featuring a I²C/I3C-compliant digital interface. Designed in line with the JEDEC JESD302-1 standard for Grade B devices, the SQ52912 surpasses the temperature accuracy requirements, making it a key component for optimizing DDR5 memory module performance. The SQ52912 is designed for low-power, high-accuracy thermal monitoring applications and is available in a compact 6-ball DSBGA package.

The SQ52912 operates with a core power supply of 1.8V and an I/O power supply of 1V. It features a low typical average quiescent current of 5.8μA during each 125ms conversion.

The SQ52912 offers a typical accuracy of ±0.3°C across the full temperature range of –40 °C to +125 °C. It integrates an on-chip 11-bit analog-to-digital converter (ADC) designed to operate with a 1.8V core power supply and a 1.0V I/O power supply. The SQ52912 also supports advanced features such as in-band interrupts, parity checks, and packet error checks.

Features

- JEDEC JESD302-1 DDR5 Grade B Support
- Exceeds JEDEC Temperature Accuracy Standards:
  - ±0.3°C Typical from –40°C to +125°C
  - ±0.5°C Max from +75°C to +95°C
  - ±0.8°C Max from –40°C to +125°C
- Low Quiescent and Stand by Currents:
  - 5.7μA Average Quiescent Current (typ.)
  - 3.8μA Standby Current (typ.)
- Power Supplies:
  - 1V I/O Power Supply
  - 1.8V Core Power Supply
- Two-Wire Serial Bus Interfaces (I²C and I3C Basic Operation Modes)
- Data Transfer Rate: Up to 12.5MHz in I3C Basic Mode
- In-Band Interrupt (IBI) for Alerting Host
- Error Detection Functions:
  - Parity Check for Host Writes
  - Packet Check for Host Reads and Writes
- 11-bit Resolution with 0.25°C per LSB
- Standard 6-Ball DSBGA (WCSP) Package With 0.5mm Pitch

Applications

- Servers, Laptops, Workstations
- DDR5 DIMM Modules
- SSDs

Typical Application

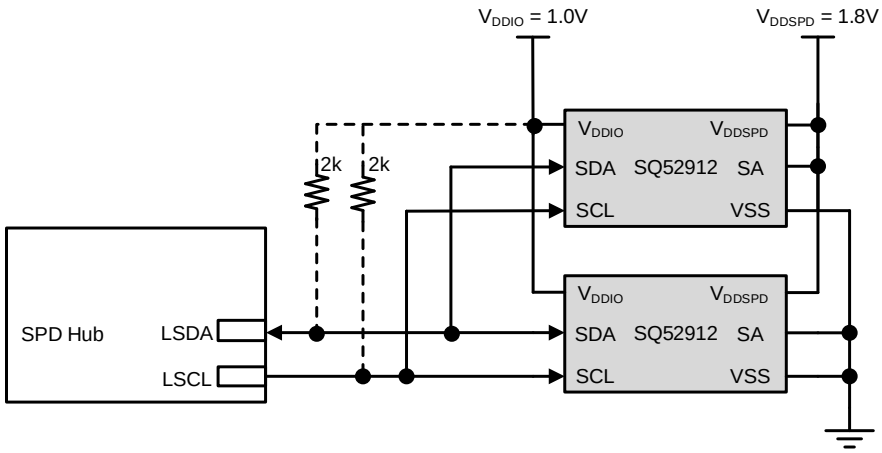


Figure 1. Typical Application Circuit

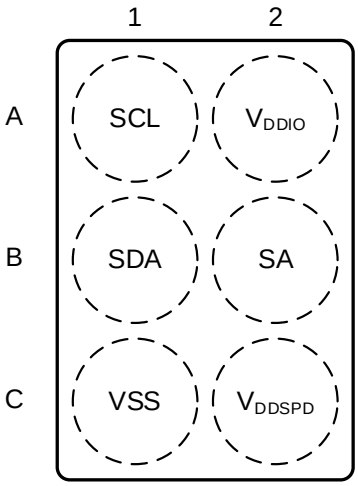
Ordering Information

Ordering Part Number	Package Type	Top Mark
SQ52912WBS	CSP1.338x0.838-6	**gxyz**

Device code: g

x=year code, y=week code, z= lot number code.

Pinout (Top View)



CSP1.338x0.838-6

Pin Descriptions

Pin No.	Pin Name	IO Type	Pin Description
B2	SA	I	Address selection. Connect to either V_{DDSPD} or GND to set the device address.
A1	SCL	I	Serial clock.
B1	SDA	I/O	Serial data input and output. The pin functions as either open-drain or push-pull in I3C mode and open-drain in I^2C mode.
A2	VDDIO	I	Supply voltage for sensor I/Os.
C2	VDDSPD	I	Supply voltage for sensor core.
C1	VSS	-	Ground.

Absolute Maximum Ratings

Parameter (Note 1)	Min	Max	Unit
VDDIO to GND	-0.5	2.1	V
VDDSPD to GND	-0.5	2.1	
SA	-0.5	2.1	
SCL, SDA	-0.5	V_{DDIO}+0.3	
Junction Temperature Range	-55	150	°C
Storage Temperature Range	-65	150	
ESD: HBM (Human Body Model)	± 2000		V
ESD: CDM (Charged Device Model)	± 1000		V

Thermal Information

Parameter (Note 2)	Value	Unit
θ_{JA} Junction-to-Ambient Thermal Resistance	311.9	°C/W
θ_{JC} Junction-to-Case (top) Thermal Resistance	0.98	
θ_{JB} Junction-to-Board Thermal Resistance	29.8	
Ψ_{JT} Junction-to-Top Characterization Parameter	0.6	
Ψ_{JB} Junction-to-Board Characterization Parameter	29.3	
P_D Power Dissipation T_A = 25°C	0.4	W

Recommended Operating Conditions

Parameter (Note 3)	Min	Max	Unit
V_{DDIO} to GND	0.95	1.05	V
V_{DDSPD} to GND	1.7	1.98	V
SA	0	V_{DDSPD}+0.3	V
SCL, SDA	0	V_{DDIO}+0.3	V
Operating Free-Air Temperature Range	-40	125	°C

Electrical Characteristics

At T_A = -40°C to +125°C, V_{DDIO} = 0.95V to 1.05 V and V_{DDSPD} = 1.7V to 1.98V (unless otherwise noted); typical specifications are at T_A = 25°C, V_{DDIO} = 1V and V_{DDSPD} = 1.8V (Note 4).

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
Temperature Sense						
Temperature Accuracy	T_{ERR}	+75°C to +95°C		±0.25	±0.5	°C
		-40°C to +125°C		±0.3	±0.8	°C
Resolution	T_{RES}	1LSB(11-bit)		0.25		°C
Repeatability	T_{REPEAT}			1		LSB
Active Conversion Time	t_{ACT}			10		ms
Conversion Interval	t_{CONV}			125		ms
Temperature Hysteresis	T_{HYST}		1			°C
Power Supply						
Average Current (Serial Bus Inactive)	I_Q	125ms conversion interval		5.7	55	μA
Average Current (Read Operation)	I_{DDR}	125ms conversion interval, f_{SCL}=12.5MHz		5.7		μA
Average Current (Write Operation)	I_{DDW}	125ms conversion interval, f_{SCL}=12.5MHz		5.7		μA
Active Current	I_{ACT}	During 10ms active conversion		29	81	μA
Standby Current	I_{DD1}	Between active conversions, while in continuous conversion mode		3.8	53	μA
Power-on Reset Threshold	V_{PON}	Monotonic rise between V_{PON} and V_{DDSPD} (MIN)	1.6			V
Power-off Reset Threshold for Warm Power on Cycle	V_{POFF}	Monotonically rise above V_{POFF}			0.3	V
Initialization Time after Power-on Reset	t_{INIT}				10	ms
Warm Power Cycle off Time	t_{POFF}		1			ms
Time from Valid V_{DDSPD} Supply to Sense SA Pin for LID Code Assignment	t_{SENSE_SA}				5	ms
Device Reinitialization Time	t_{RST}				40	μs
Digital Part						
Input Capacitance	C_{IN}	Input capacitance (SCL and SDA)			4	pF
Leakage Input Current	I_{LI}		-1	0	1	μA
Leakage Output Current	I_{LO}		-1	0	1	μA
Low-level Input Logic	V_{IL}		-0.4		0.4	V
High-level Input Logic	V_{IH}		0.7		1.35	V
Input Voltage Hysteresis	V_{HYS}	SCL and SDA pins		100		mV
Low-level Output Logic	V_{OL}	SDA pin, I_{OL} = -3mA	0		0.33	V
High-level Output Logic	V_{OH}	SDA pin, I_{OH} = 3mA	0.75			V
Output Slew Rate	SLEW_RATE	SDA pin	0.1		1	V/ms
I^2C Mode Open Drain (Note 5)						
SCL Operating Frequency	f_{SCL}		0.01		1	MHz
Clock Pulse Width High Time	t_{HIGH}		260			ns
Clock Pulse Width Low Time	t_{LOW}		500			ns
Detect Clock Low Timeout	$t_{TIMEOUT}$		10		50	ms
SDA Rise Time	t_R				120	ns
SDA Fall Time	t_F		4		120	ns
Data Setup Time	t_{SUDAT}		50			ns
Data Hold Time	t_{HDDI}		0			ns
START Condition Setup Time	t_{SUSTA}		260			ns
Hold time after repeated START condition. After this period, the first clock is generated.	t_{HDSTA}		260			ns

STOP Condition Setup Time	t_{SUSTO}		260			ns
Time between STOP Condition and next START Condition	t_{BUF}		500			ns
SCL Falling Clock into SDA Data Out Hold Time	t_{HDDAT}		0.5		350	ns
DEVCTRL CCC followed by DEVCTRL CCC or register read/write command delay	$t_{DEVCTRLCCC\_PEC\_DIS}$		3			µs
SETHID CCC or SETAASA CCC to any other CCC or read/write command delay	$t_{I2C\_CCC\_UPDATE\_DELAY}$		2.5			µs
I3C Mode Push Pull (Note 5)						
SCL Operating Frequency	f_{SCL}		0.001		12.5	MHz
Clock Pulse Width High Time	t_{HIGH}		35			ms
Clock Pulse Width Low Time	t_{LOW}		35			ns
Detect Clock Low Timeout	$t_{TIMEOUT}$		10		50	ns
SDA Rise Time	t_R				5	ns
SDA Fall Time	t_F				5	ns
Data Setup Time	t_{SUDAT}		8			ns
Data Hold Time	t_{HDDI}		3			ns
START Condition Setup Time	t_{SUSTA}		19.2			ns
Hold time after repeated START condition. After this period, the first clock is generated.	t_{HDSTA}		38.4			ns
STOP Condition Setup Time	t_{SUSTO}		19.2			ns
Time between STOP condition and next START condition	t_{BUF}		500			ns
Bus available time (no edges seen in SDA and SCL)	t_{AVAL}		1			µs
Time to issue IBI after an event is detected when bus is available	t_{IBI_ISSUE}				15	µs
Time from Clear Register Status to any I3C operation with START condition. PEC disabled	$t_{CLR\_I3C\_CMD\_DELAY}$		4			µs
Time from Clear Register Status to any I3C operation with START condition. PEC enabled			15			µs
SCL falling clock into SDA valid data out time	t_{DOUT}		0.5		12	ns
SCL rising clock into SDA output off	t_{DOFFS}		0.5		12	ns
SCL rising clock into host controller SDA output off	t_{DOFFM}		0.5		30	ns
SCL rising clock into host controller driving SDA low	$t_{CL\_R\_DAT\_F}$		40			ns
DEVCTRL CCC followed by DEVCTRL CCC or register read/write command delay	$t_{DEVCTRLCCC\_PEC\_DIS}$		3			ns
Register write command followed by register read command delay in PEC-enabled mode	$t_{WR\_RD\_DECLAY\_PEC\_EN}$		8			ns
RSTDAA CCC or ENEC CCC or DISEC CCC to any other CCC or read/write command delay	$t_{I3C\_CCC\_UPDATE\_DELAY}$		2.5			µs
Any CCC to RSTDAA CCC delay	t_{CCC_DELAY}		2.5			µs
Switching Characteristics						
Spike filter for I3C compatibility valid in I^2C mode only	t_{LPF}	SCL=12.5MHz			50	ns

Note 1: Stresses beyond the “Absolute Maximum Ratings” may cause permanent damage to the device. These are stress ratings only. Functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Note 2: Tested with natural convection and chip mounted on low effective single-layer PCB in accordance with JESD5-3.

Note 3: The device is not guaranteed to function outside its recommended operating conditions.

Note 4: Unless otherwise stated, limits are 100% production tested under pulsed load conditions such that $T_A \cong T_J = 25℃$. Limits over the operating temperature range (see recommended operating conditions) and relevant voltage range(s) are guaranteed by design, test, or statistical correlation.

Note 5: Guaranteed by design or statistical correlation and not production tested.

Communication Interface Timing

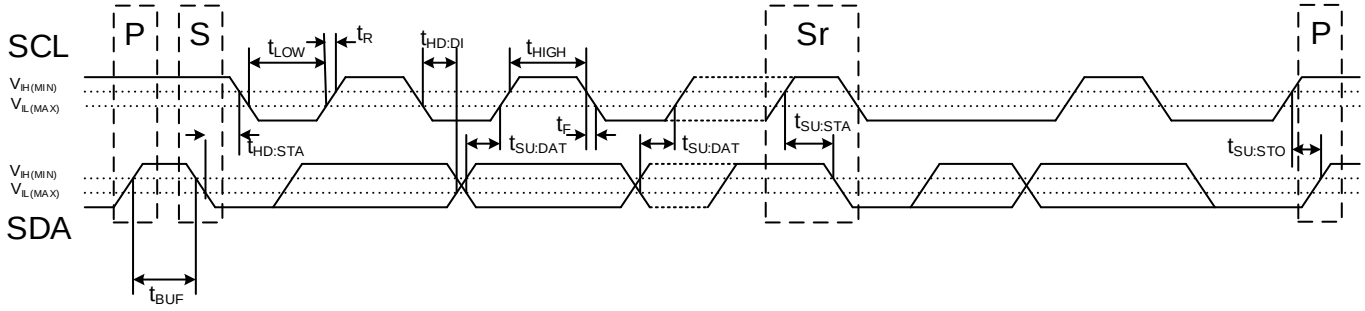


Figure 2. I²C and I3C Basic Bus Input Timing Diagram

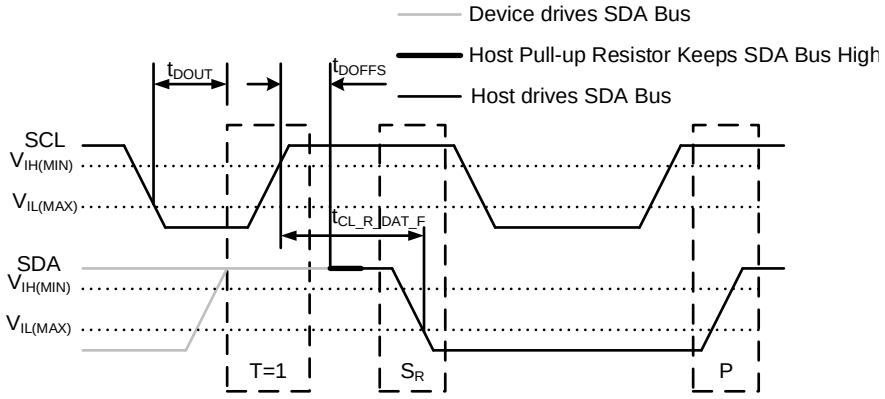


Figure 3. T = 1 Host Ends Read with Repeated Start and Stop Timing Diagram

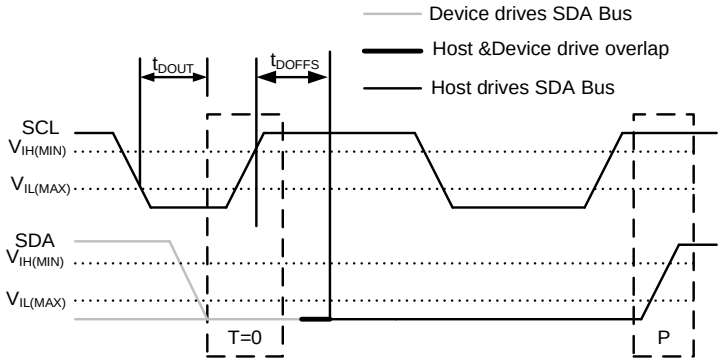


Figure 4. T = 0 Host Ends Read with Repeated Start and Stop Timing Diagram

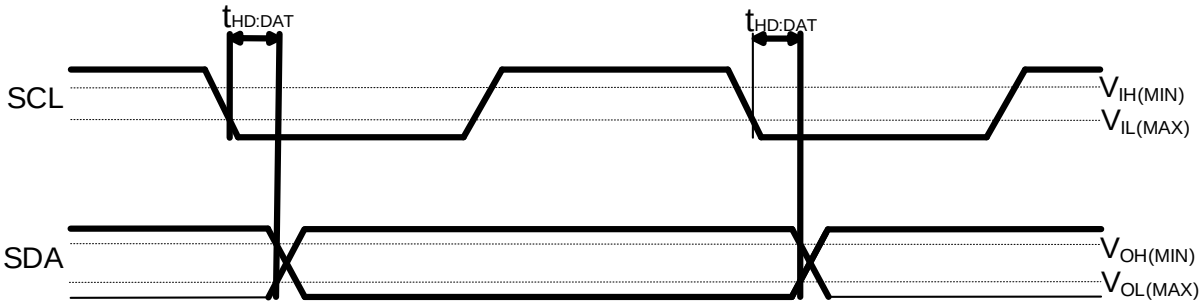


Figure 5. I²C Basic Bus Output Timing Diagram

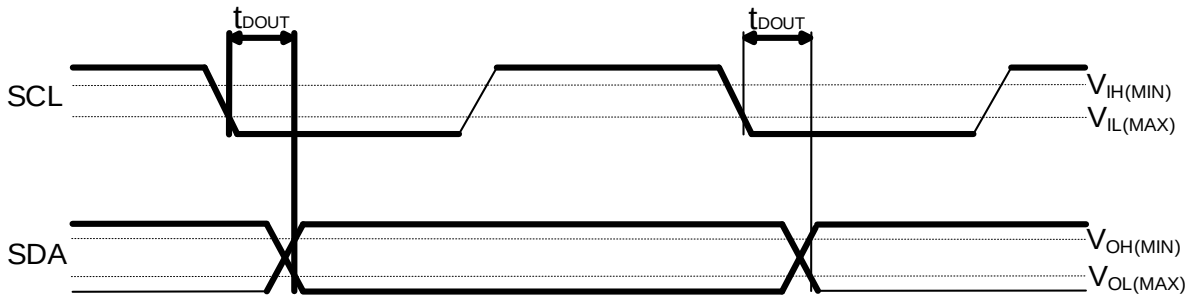


Figure 6. I3C Basic Bus Output Timing Diagram

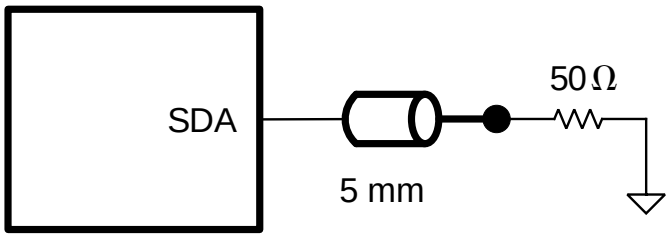


Figure 7. Output Slew Rate and Timing Reference Load for SDA

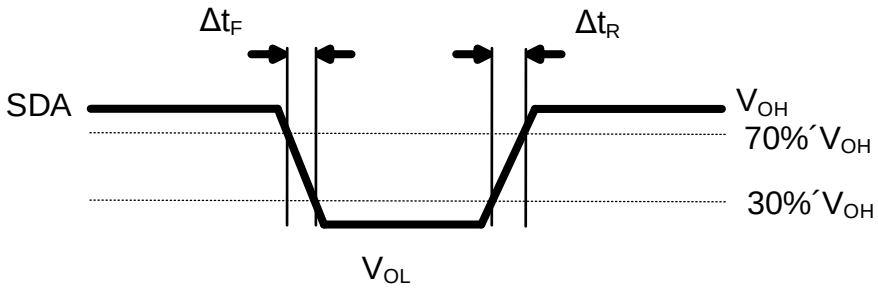


Figure 8. Output Slew Rate Measurement Points

Detailed Description

Overview

The SQ52912 is a high-accuracy temperature sensor featuring a I²C/I3C compatible interface, power-down capability, device reset, parity check, packet error check, In-Band Interrupts (IBI), and support for common command codes (CCC).

Functional Block Diagram

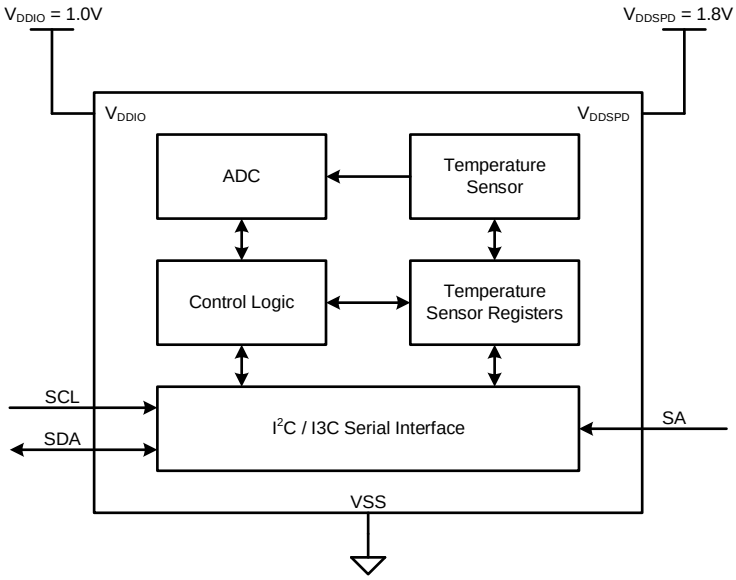


Figure 9. SQ52912 Functional Block Diagram

Power-Up Sequence

The SQ52912 is equipped with two supply pins: V_{DDSPD}, which is responsible for supplying power to the sensor core, and V_{DDIO}, which powers the IO. For proper startup of the device, it is crucial that V_{DDSPD} is powered up first, followed by V_{DDIO}. To prevent any improper operation due to an incorrect power-up sequence, the SQ52912 includes a power-on reset (POR) circuit.

Figure 10 shows that the VDDSPD supply should be applied first and must rise steadily within the range between VPON(min) and VDDSPD(min) without oscillation. The VDDIO supply must ramp up to the correct level before any operation begins.

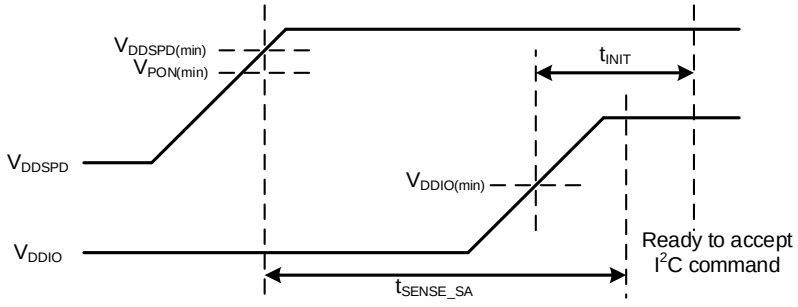


Figure 10. Power-Up Sequence

Once the V_{DDSPD} and V_{DDIO} supplies have ramped up above their minimum threshold values, the SQ52912 performs the following steps:

1. Within t_{SENSE_SA}, the device samples the SA pin to configure the LID code, which is part of the device address.
2. Within t_{INIT}, the device enables the interface to accept the host's commands.

The device consistently powers up in I²C mode.

Power-Down and Device Reset

If the V_{DDSPD} supply drops below the $V_{DDSPD(min)}$ level, device operation cannot be guaranteed. To ensure proper functionality, the application must maintain V_{DDIO} and V_{DDSPD} below V_{POFF} for the T_{POFF} duration, as depicted in *Figure 11*. Once this condition is met, the device will reset, and the power-up sequence will initialize the device.

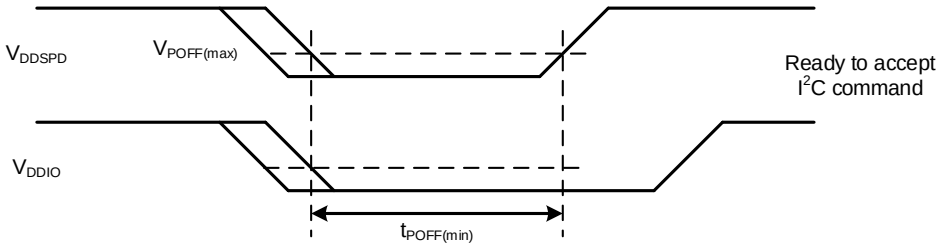


Figure 11. Power-Down and Reset Sequence

Temperature Result and Limits

The temperature result and limit registers contain 11-bit values stored across two consecutive registers. The low byte register precedes the high byte register, as detailed in Table 1 of the register map. The data is formatted as an 11-bit signed number, with the most significant bit acting as the sign bit. Each bit in the temperature value corresponds to a specific weight, which can be used to calculate the temperature.

Unused bits in the registers will read as 0, and writing to an unused bit will have no effect. The temperature result and limit registers offer a resolution of 0.25°C, allowing for values between -255.75°C and +255.75°C, though the recommended operating range is from -40°C to +125°C.

Table 1. Temperature Register Format

Register Byte	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Low Byte	8	4	2	1	0.5	0.25	RSVD=0	RSVD=0
High Byte	RSVD=0	RSVD=0	RSVD=0	Sign	128	64	32	16

Table 2 provides examples of temperature register readings and their corresponding conversions in °C.

Table 2. Temperature Register Examples

Temperature (°C)	High Byte	Low Byte
+255.75	0000 1111	1111 1100
+125	0000 0111	1101 0000
+95	0000 0101	1111 0000
+85	0000 0101	0101 0000
+75	0000 0100	1011 0000
+1	0000 0000	0001 0000
+0.25	0000 0000	0000 0100
0	0000 0000	0000 0000
-0.25	0001 1111	1111 1100
-1	0001 1111	1111 0000
-25	0001 1110	0111 0000
-40	0001 1101	1000 0000
-255.75	0001 0000	0000 0000

Bus Reset

To prevent the serial bus from becoming stuck, the SQ52912 supports a bus reset mechanism. Since the devices on the bus do not control the SCL line, this mechanism relies on a timeout scheme on the SCL line, as shown in Figure 12. If the host controller holds the SCL line low for a duration exceeding $T_{TIMEOUT(max)}$, the SQ52912 will be reset and initiate the following actions:

- The interface is reset, and since the bus reset is treated as a Stop condition, any pending internal transactions are cleared.
- The SQ52912 returns to I²C mode and resets the following registers:
  a) MR7 register: DEV_HID_CODE [2:0] is set to 3'b111.
  b) MR18 register: PEC_EN, PAR_DIS and INF_SEL bits are set to 1'b0.
  c) MR27 register: IBI_ERROR_EN is set to 1'b0.
  d) MR52 register: PEC_ERROR_STATUS and PAR_ERROR_STATUS are set to 1'b0.
- The SQ52912 does not resample the SA pin.
- The SQ52912 floats the SDA pin, allowing the bus controller to pull up the line.

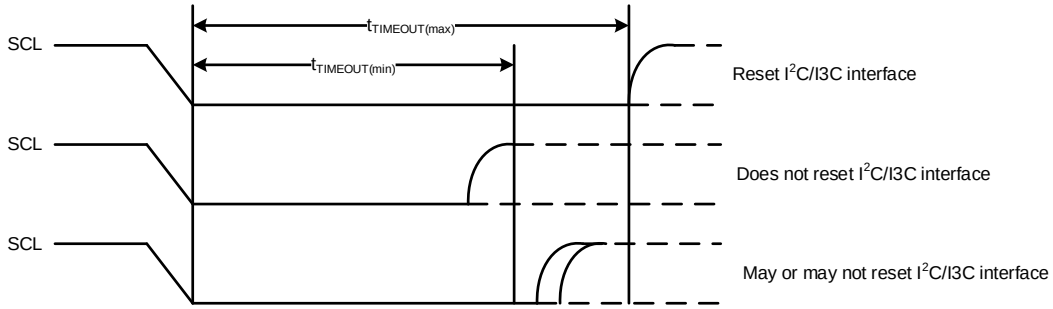


Figure 12. I²C or I3C Basic Bus Reset

Interrupt Generation

The SQ52912 lacks a dedicated interrupt or alert pin but supports interrupt generation through In-Band Interrupts (IBI) on the SDA pin. Interrupt generation using IBI is supported only in I3C mode. Therefore, the application must ensure that the device is first programmed to operate in I3C mode before enabling IBI functionality. An arbitration process is required because multiple devices on the I3C bus can generate an IBI.

If the SQ52912 detects an idle state on the bus for T_{AVAL}, it generates an IBI. To indicate the presence of an IBI to the host, the device will pull the SDA line low for the T_{IBI_ISSUE} duration. Subsequently, the host initiates the Start bus condition by driving the SCL low. At this point, the device transmits its device address on the bus with the R/W bit set.

A situation may arise where the host commences a new bus transaction simultaneously with the SQ52912 generating an IBI. In such an event, the SQ52912 and the host participate in arbitration for the device address byte.

Parity Error Check

The SQ52912 implements an odd parity error check. In I²C mode, parity error check is supported only for the supported common command codes (CCC). However, in I3C mode, it is supported for both CCC and host-to-device data transfers. The parity bit is transmitted only during a host write operation, and the SQ52912 verifies the parity to ensure the accuracy of the received data or CCC. The device uses odd parity, meaning that if an odd number of bits in a byte are set to 1, the parity bit is set as 0. Conversely, if an even number of bits in a byte is set as 1, the parity bit is set as 1.

If a parity error occurs during a data transfer or CCC, the SQ52912 will discard the bytes following the detection of the parity error and wait for a Stop condition on the bus. When a parity error is detected, the device will set the IBI_STATUS bit in the MR48 register and the PAR_ERROR_STATUS bit in the MR52 register.

Packet Error Check

The packet error check (PEC) is implemented using a CRC-8 algorithm with the polynomial specified in **Table 3**.

Table 3. PEC Rule Table

PEC Rule	Attributes
PEC width	8-bits
PEC polynomial	$X^8+X^2+X^1+1$

Initial seed value	00h
Input data reflected	No
Output data reflected	No
XOR value	00h

The PEC is exclusively supported in I3C mode and is calculated based on the device address, the R/W bit, and the data packet. The initial seed value for the PEC function is reset to zero whenever a Start or Repeated Start bus condition occurs.

Whenever a PEC enable or disable operation is executed during a host transaction, it is imperative to follow it promptly with a Stop condition on the bus. This is necessary to ensure that the PEC control bit in the MR18 register is updated accordingly.

Device Functional Modes

This section covers the serial address structure of the SQ52912 and its operation in both I²C mode and I3C basic mode, including the transition between these modes. Additionally, it discusses the behavior of the SQ52912 during In-Band Interrupts (IBI) and the bus reset sequence.

Conversion Mode

Upon startup, the SQ52912 is set to continuous conversion mode. In this mode, the device performs temperature conversions at regular intervals of 125 ms, as illustrated in *Figure 13*.

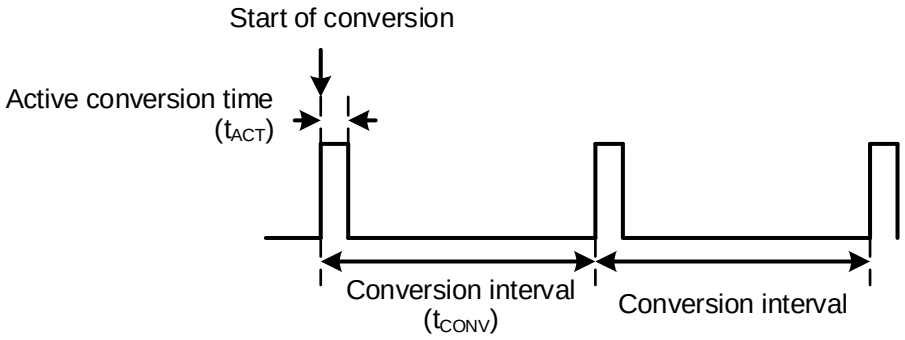


Figure 13. Continuous Conversion Timing Diagram

To halt the conversion process, the application software can clear bit 0 of the MR26 register. When this feature is disabled, the device doesn't update the result registers. After disabling the temperature sensor, the host must wait for at least one active conversion cycle before any additional writes are executed on the device for the disable command to become effective. Similarly, when re-enabling the temperature sensor for continuous conversion mode, the host must wait for at least one complete conversion interval before reading the temperature results. During this waiting period, it is permissible to read from other registers.

Serial Address

The SQ52912 features a 7-bit serial address that facilitates communication between the host and the device in both I²C and I3C basic modes. The serial address format for the SQ52912 is presented in **Table 4**. As detailed in the power-up sequence, the sampled value of the SA pin during power-up is employed to choose between the two available sections of the local device type ID (LID) within the serial address. The LID is then combined with the host ID (HID) to form a 7-bit serial address that is unique to the device.

Table 4. Serial Address Format

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	SA	1	0	1	1	1	R/W
Local Device Type ID (LID)				Host ID (HID)			Read/Write

If the SA pin is connected to GND, the serial address for the SQ52912 is encoded as 7'b0010111. If the SA pin is connected to VDDSPD, the serial address is encoded as 7'b0110111.

I²C Mode Operation

The device defaults to I²C mode after powering up, undergoing a bus reset, or issuing a RSTDAA CCC while in I³C mode. In this mode, the maximum supported bus speed is 1.0MHz. The following functionalities are not supported in this mode:

1. Packet Error Check: This feature is not supported. If the host includes a PEC byte when writing data, the PEC byte will be interpreted as a data byte and written to the next register address in sequence.
2. Parity Error Check: Parity error check is supported only for the CCC listed in Table 6.
3. IBI: When IBI is enabled in I³C basic mode, switching to I²C mode will disable the IBI functionality. If events occur that would normally generate an IBI, the device will log the event status in the appropriate register.

When operating in I²C mode, the SQ52912 supports SETHID, DEVCTRL, and SETAASA CCCs, as well as data transfer packets without PEC. Moreover, it is only permissible to use a Start or Repeated Start followed by 7'h7E with W = 0 in order to issue the supported CCCs. Any other use of a Repeated Start is deemed invalid.

Host I²C Write Operation

To perform an I²C write operation, the host controller initiates communication by sending the device address with the R/W bit set to 0 after a Start or Repeated Start, as depicted in *Figure 14*. Subsequently, the 8-bit register address is transmitted, followed by the data. The SQ52912 receives the data and writes it to the specified register address.

After each data byte is written, the internal write register address pointer is incremented. In the event that the write operation causes an address rollover, the device will reset the internal write register address pointer and, if possible, continue the write operation. It is worth noting that for reserved or read-only registers, the SQ52912 does not NACK the data byte. Instead, it discards the data byte and does not update the register.

S or Sr	0	SA	1	0	HID2	HID1	HID0	R/W=0	ACK/NACK
	RA = Register Address [7:0]								ACK
	$Data_{[RA]}$								ACK
	$Data_{[RA+1]}$								ACK
	...								ACK
	$Data_{[RA+N]}$								ACK

Figure 14. I²C Write Operation

Host I²C Read Operation

To perform an I²C read operation, the host controller initiates communication by sending the device address with the R/W bit set to 0 after a Start or Repeated Start. This is followed by transmitting the 8-bit register address. Once the register address is received by the SQ52912, the host controller issues a Repeated Start and sends the device address with the R/W bit set to 1. Then the device sends the data from the register address incrementally until the host controller sends a NACK. If the read operation causes the internal read register address pointer to rollover, the behavior of the SQ52912 device is undefined.

S or Sr	0	SA	1	0	HID2	HID1	HID0	R/W=0	ACK/NACK	
	RA = Register Address [7:0]								ACK	
Sr	0	SA	1	0	HID2	HID1	HID0	R/W=1	ACK/NACK	
	$Data_{[RA]}$								ACK	
	$Data_{[RA+1]}$								ACK	
	...								ACK	
	$Data_{[RA+N]}$								NACK	Sr or P

Figure 15. I²C Read Operation

Host I²C Read Operation in Default Read Address Pointer Mode

Figure 16 illustrates the default read address pointer mode provided by the SQ52912 for reading a specific register on the I²C bus. This mode offers a more efficient polling mechanism since the number of bytes to be sent by the host is two less

than a standard I²C read operation. To enable this mode, the DEF_RD_ADDR_POINT_EN bit in the MR18 register is utilized, while the DEF_RD_ADDR_POINT_Start bits are used to specify the default read address pointer set to a particular register in the register map. Once enabled, the SQ52912 will set the internal read address pointer to the specified register when a Stop condition occurs on the bus.

S or Sr	0	SA	1	0	HID2	HID1	HID0	R/W=1	ACK/NACK	
	Data [RA]								ACK	
	Data [RA+1]								ACK	
	...								ACK	
	Data [RA+N]								NACK	Sr or P

Figure 16. I²C Default Read Address Pointer Mode

In this mode of operation, two specific scenarios can occur. The first scenario, shown in Figure 17, involves a standard I²C read operation before entering the default read mode. If a Stop condition is issued before the Start condition, the internal read address pointer will reset to the default address. Subsequent data reads will return data bytes corresponding to this default read address. Alternatively, if a Repeated Start is issued instead of a Stop, the SQ52912 will transmit data based on the default read address pointer.

S or Sr	0	SA	1	0	HID2	HID1	HID0	R/W=0	ACK/NACK	
	RA = Register Address [7:0]								ACK	
Sr	0	SA	1	0	HID2	HID1	HID0	R/W=1	ACK/NACK	
	Data [RA]								ACK	
	Data [RA+1]								ACK	
	...								ACK	
	Data [RA+N]								NACK	P
S	0	SA	1	0	HID2	HID1	HID0	R/W=1	ACK/NACK	
	Data [DEF_ADDR_POINTER]								ACK	
	Data [DEF_ADDR_POINTER+1]								ACK	
	...								ACK	
	Data [DEF_ADDR_POINTER+N]								NACK	Sr or P

Figure 17. I²C Normal Read Followed by Default Read Address

In the second case, depicted in Figure 18, a normal I²C write operation is performed before entering the default read mode. If a Stop condition is encountered, followed by a write bus operation, and then a Repeated Start for the read mode, the SQ52912 will update its internal read address pointer to the default read address. Subsequently, it will transmit bytes to the host accordingly.

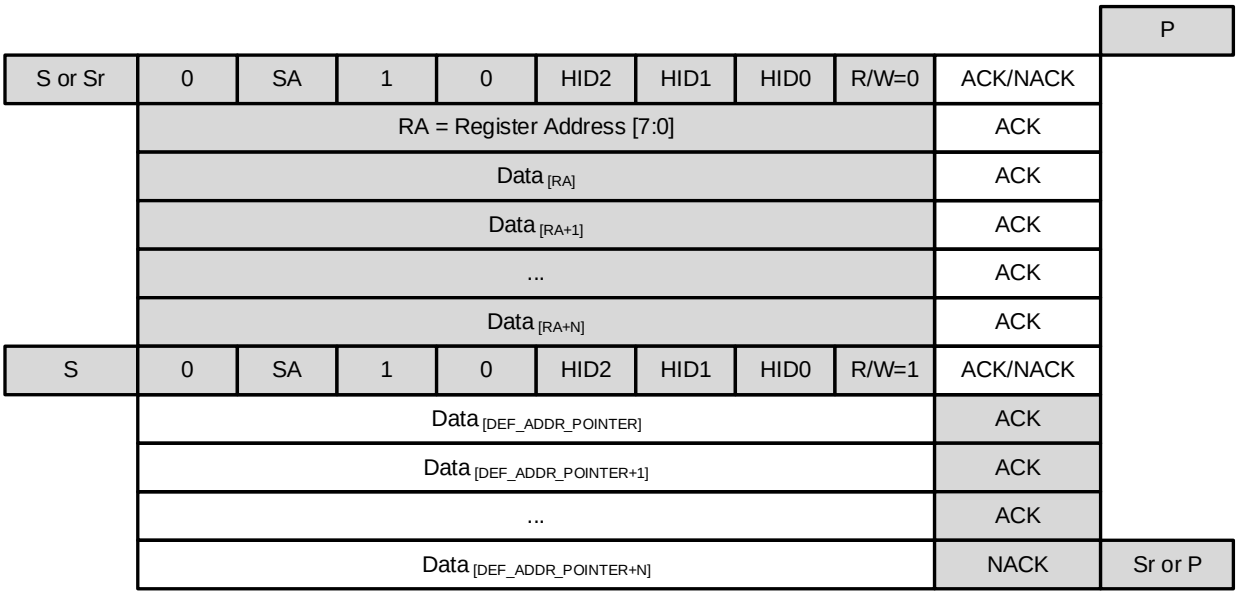


Figure 18. I²C Normal Write Followed by a Default Read Address

Switching from I²C Mode to I3C Basic Mode

Table 6 demonstrates that in I²C mode, only DEVCTRL, SETHID, and SETAASA CCCs are supported. The host may issue DEVCTRL and/or SETHID before it can issue SETAASA to switch the device from I²C mode to I3C basic mode. Once the host issues SETAASA, the device will register the command by setting the INF_SEL bit in the MR18 register to 1'b1 upon detecting a Stop condition on the bus. Subsequently, the SQ52912 will operate in the I3C basic mode.

I3C Basic Mode Operation

As explained in the preceding section, the transition to the I3C basic mode of operation is always entered from the I²C mode of operation. In the I3C basic mode, the device can accommodate data transfer rates of up to 12.5 MHz by utilizing a push-pull SDA driver. Furthermore, the following features may be supported either by default or when enabled:

1. IBI: Initially disabled, IBI can be enabled.
2. Packet error check: Disabled by default, but the SQ52912 can support the PEC feature when the host enables it.
3. Parity check: Always enabled by default.

In I3C basic mode, the structure of read and write packets may vary. The data payload structure is determined by the enabled feature.

Host I3C Write Operation without PEC

As shown in Figure 19 and*Figure 20*, an I3C basic write operation is identical to an I²C write operation. For all bytes following the device address field, the 9th bit is the parity bit sent by the host. If the host enables the Interrupt Bit Information (IBI), it must first transmit the IBI header byte composed of 7'h7E+R/W=0 prior to sending the device address. This allows devices on the bus to arbitrate among themselves if multiple devices have an interrupt condition that must be communicated to the host.

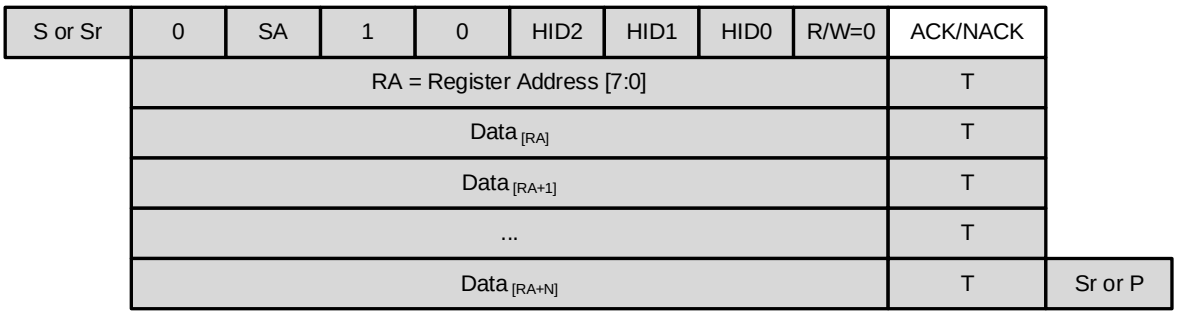


Figure 19. I3C Basic Mode Write

S	1	1	1	1	1	1	0	R/W=0	ACK	
Sr	0	SA	1	0	HID2	HID1	HID0	R/W=0	ACK/NACK	
	RA = Register Address [7:0]								T	
	Data [RA]								T	
	Data [RA+1]								T	
	...								T	
	Data [RA+N]								T	Sr or P

Figure 20. I3C Basic Mode Write with IBI Header

In the event of a parity error during data transfer, the device discards all bytes, including the byte where the parity error was detected, and flag the occurrence of a parity error. If the host attempts a new transaction with a Repeated Start to the same device, the SQ52912 will NACK the device address to signal an error condition. Before performing any new transfer to SQ52912, the host must first clear the parity error condition. When IBI is enabled, the device can use it to report error conditions to the host. However, in the absence of IBI, it is highly recommended that the host verifies the error status register to confirm that no parity error occurred on the bus.

Host I3C Write Operation with PEC

Figure 21 andFigure *22* illustrate that when the host enables the PEC, it appends an extra byte after transmitting the register address. The format for this additional byte is shown in Table 5.

Table 5. Command Truth Table for PEC Enabled Mode

CMD	RW	Command Name	Command Description
000	0	W1R	Write 1 byte to register address specified in data packet
	1	R1R	Read 1 byte from register address specified in data packet
001	0	W2R	Write 2 bytes to register address specified in data packet
	1	R2R	Read 2 bytes from register address specified in data packet
010 - 111	X	RSVD	Reserved

In cases where the CMD value transmitted by the host is not valid for SQ52912, the device does not write any data to the specified register.

S or Sr	0	SA	1	0	HID2	HID1	HID0	R/W=0	ACK/NACK	
	RA = Register Address [7:0]								T	
	CMD			W=0	0	0	0	0	T	
	Data [RA]								T	
	...								T	
	Data [RA+N]								T	
	PEC								T	Sr or P

Figure 21. I3C Basic Mode Write with PEC Enabled

S	1	1	1	1	1	1	0	R/W=0	ACK	
Sr	0	SA	1	0	HID2	HID1	HID0	R/W=0	ACK/NACK	
	RA = Register Address [7:0]								T	
	CMD			W=0	0	0	0	0	T	
	Data [RA]								T	
	...								T	
	Data [RA+N]								T	
	PEC								T	Sr or P

Figure 22. I3C Basic Mode Write with IBI Header and PEC Enabled

In the event of a parity error during data transfer, the device will discard all bytes, including the byte where the parity error was detected, and set the parity error condition. Similarly, if a PEC error occurs, the SQ52912 will discard the entire data packet and set the PEC error condition. In both cases, if the host attempts to initiate a new transaction with a Repeated Start to the same device, the SQ52912 will NACK the device address to indicate an error condition to the host. Before performing any new transfer to the SQ52912, the host must first clear the respective error condition, whether parity or PEC error.

When IBI is enabled, the device can utilize IBI to communicate any error conditions observed to the host. However, in cases where IBI is not enabled, it is strongly recommended that the host checks the error status register to ensure that no parity or PEC error was detected on the bus.

Host I3C Read Operation without PEC

Figure 23 andFigure *24* demonstrate that an I3C basic mode read operation is identical to an I^2C read operation. In this operation, for all bytes transmitted by the device, the 9^{th} bit serves as the T-bit. The device and host utilize this bit to negotiate the continuation of the read transfer. During the read phase, the device sets the T-bit to 1 before the rising edge to inform the host that it can send more bytes. Alternatively, the device sets the T-bit to 0 to indicate to the host its desire to terminate the transfer. In response, the host can either issue a Stop or Repeated Start on the bus. The host also has the option to terminate the transfer by setting the T-bit to 0, but only when the device sends the T-bit as 1, creating a repeated start condition on the bus. Furthermore, the host can send a Stop on the bus.

When the host enables IBI, it must first transmit the IBI header byte, which consists of 7'h7E+R/W = 0, before sending the device address. This allows the participating devices on the bus to arbitrate among themselves if more than one device has an interrupt condition that needs to be communicated to the host.

S or Sr	0	SA	1	0	HID2	HID1	HID0	R/W=0	ACK/NACK	
	RA = Register Address [7:0]								T	
Sr	0	SA	1	0	HID2	HID1	HID0	R/W=1	ACK/NACK	
	Data [RA]								T=1	
	Data [RA+1]								T=1	
	...								T=1	
	Data [RA+N]								T=1	Sr or P

Figure 23. I^3C Basic Mode Read

S	1	1	1	1	1	1	0	R/W=0	ACK	
Sr	0	SA	1	0	HID2	HID1	HID0	R/W=0	ACK/NACK	
	RA = Register Address [7:0]								T	
Sr	0	SA	1	0	HID2	HID1	HID0	R/W=1	ACK/NACK	
	Data [RA]								T=1	
	Data [RA+1]								T=1	
	...								T=1	
	Data [RA+N]								T=1	Sr or P

Figure 24. I3C Basic Mode Read with IBI Header

If a parity error occurs in the write phase before the repeated start, the SQ52912 issues a NACK during the read phase of the transaction. Furthermore, if the host attempts to continuously read data to the point where the internal read address pointer reaches 255 (which corresponds to the last register in the register map), the device will send the T-bit as 0. When the host initiates a new transaction with a Repeated Start to the same device after a previous transaction had a parity error, the SQ52912 will NACK the device address to indicate an error condition to the host. The host must first clear the parity error condition before performing any new transfer to the SQ52912.

When IBI is enabled, the device can utilize IBI to communicate any error conditions observed to the host. However, in cases where IBI is not enabled, it is strongly recommended that the host checks the error status register to ensure that no parity error was detected on the bus.

Host I3C Read Operation with PEC

Figure 25 andFigure *26* illustrate that when the host enables PEC, the host adds an extra byte after transmitting the register address. The format for this additional byte is detailed in **Table 5**. Due to the limitations imposed by the CMD byte, which only allows for one or two byte reads, the device should terminate the read phase by sending one byte of data and the PEC byte, or two bytes of data and the PEC byte, followed by setting the T-bit to 0. However, it should be noted that if the host sets the register address as 255 and attempts a read of two bytes, the device cannot guarantee the result.

S or Sr	0	SA	1	0	HID2	HID1	HID0	R/W=0	ACK/NACK	
	RA = Register Address [7:0]								T	
	CND			R=1	0	0	0	0	T	
	PEC								T	
Sr	0	SA	1	0	HID2	HID1	HID0	R/W=1	ACK/NACK	
	Data [RA]								T=1	
	...								T=1	
	Data [RA+N]								T=1	
	PEC								T=0	Sr or P

Figure 25. I3C Basic Mode Read with PEC enabled

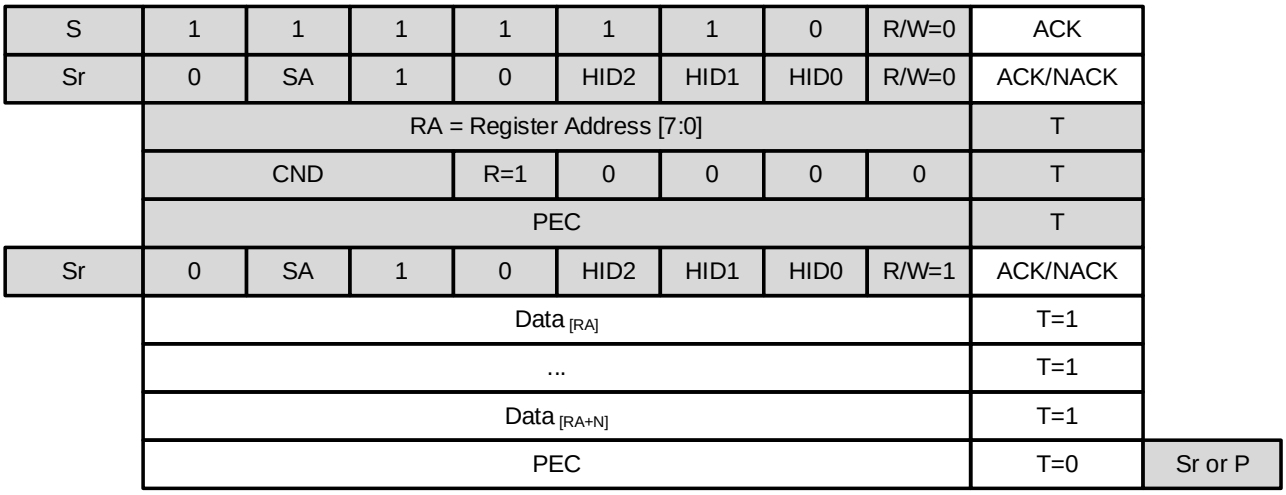


Figure 26. I3C Basic Mode Read with PEC enabled and IBI Header

If the CMD value sent by the host is invalid for the SQ52912, the device will NACK the read phase.

In the event of a parity error occurring in the write phase before the Repeated Start, the SQ52912 will NACK the read phase of the transaction. If the host attempts to initiate a new transaction with a Repeated Start to the same device, the SQ52912 should NACK the device address to indicate an error condition to the host. The host must first clear the parity error condition before performing any new transfer to the SQ52912.

If a PEC error occurs, the SQ52912 will NACK the read phase of the transaction. Similarly, if the host attempts to start a new transaction with a Repeated Start to the same device, the SQ52912 should NACK the device address to indicate the an existing error condition to the host. The host must first clear the PEC error condition before performing any new transfer to the SQ52912.

When IBI is enabled, the device can utilize it to report any error conditions observed to the host. However, in cases where IBI is not enabled, it is strongly recommended that the host checks the error status register to ensure that no parity or PEC error was detected on the bus.

Host I3C Read Operation in Default Read Address Pointer Mode

The default read address pointer mode in I3C basic mode functions similarly to I^2C mode, as shown in Figures 27 through 30. If the host continuously reads data, causing the internal read address pointer to reach 255 (the last register in the register map), the device will send the T-bit as 0. The host can terminate the transfer by setting the T-bit to 0, but this option is only available when PEC is not enabled.

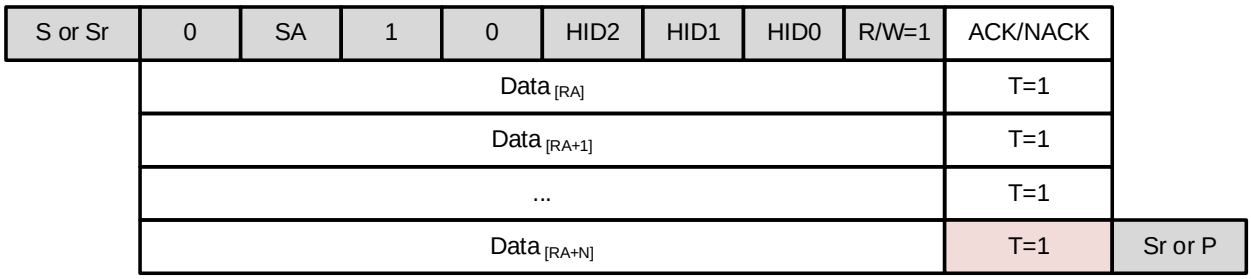


Figure 27. I3C Basic Mode Default Read Address Enabled

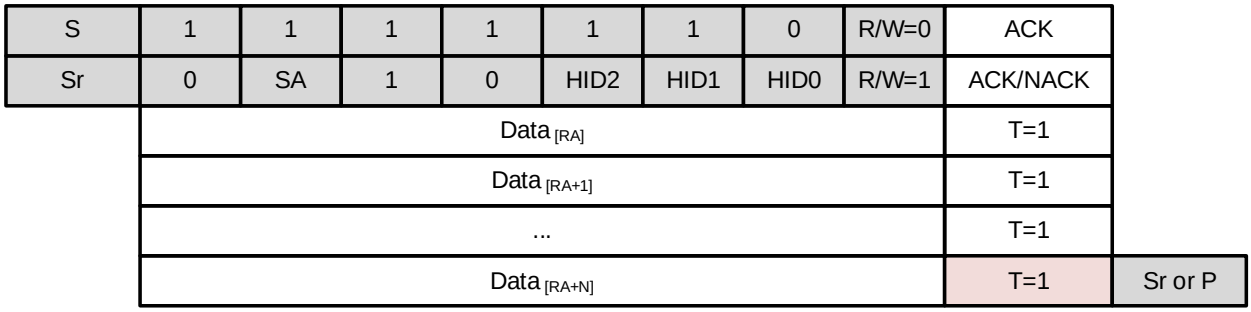


Figure 28. I3C Basic Mode Default Read Address Enabled with IBI Header

If PEC is enabled, the MR18 register determines the default number of bytes to be sent. Following the transmission of these bytes, the device will send the PEC byte with the T-bit set to 0.

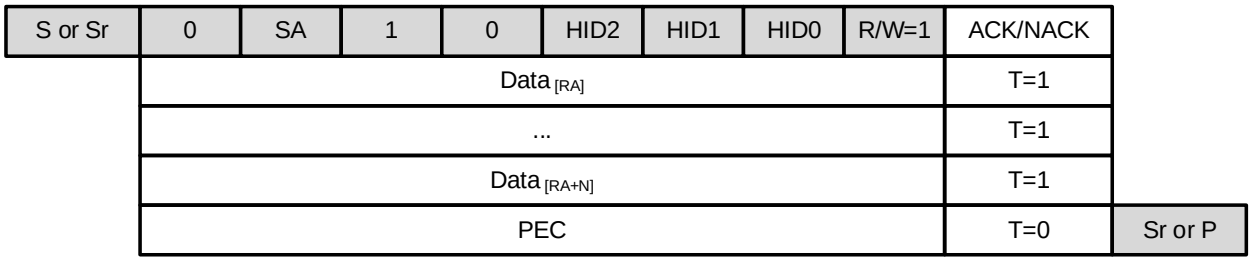


Figure 29. I3C Basic Mode Default Read Address Enabled with PEC Enabled

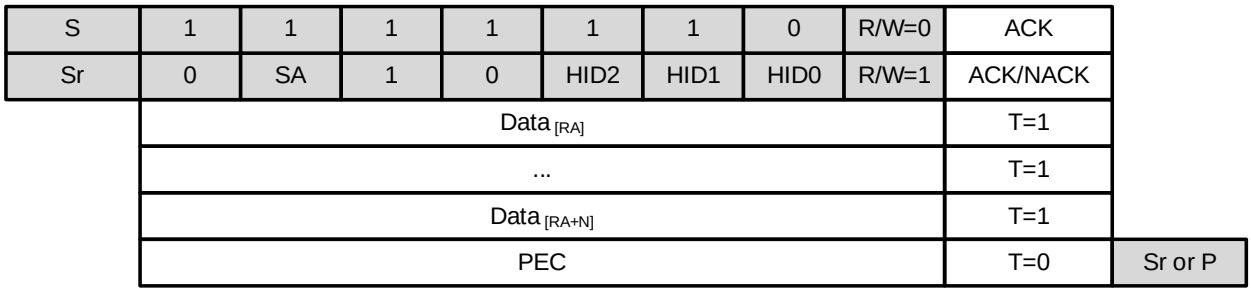


Figure 30. I3C Basic Mode Default Read Address Enabled with PEC Enabled and IBI Header

The SQ52912 will NACK the address phase during a Repeated Start if an error occurred in the previous transaction.

In-Band Interrupt

The In-Band Interrupt (IBI) is a sophisticated means of notifying the host about an event occurring in the SQ52912. The SQ52912 generates two distinct types of events:

1. Error event: Events corresponding to a PEC or parity error.
2. Temperature event: Events triggered by the temperature exceeding the upper limits or dropping below the lower limits.

When the device powers up, all interrupt sources are disabled by default. Enabling an interrupt source is only possible when the device operates in I3C basic mode, as allowing the interrupt source would generate an In-Band Interrupt (IBI), which is not permitted in I^2C mode.

The SQ52912 can only request an IBI when the bus remains inactive for the T_{AVAL} period. When the bus meets the condition for an inactive state, and no bus transaction is ongoing, the SQ52912 can trigger an IBI by driving the SDA low. This action serves as an indication to the host that an IBI is pending.

In Band Interrupt Arbitration Rules

The generation and arbitration of In-Band Interrupts (IBI) must follow specific rules outlined below, considering the state of the host controller readiness and the presence of multiple devices on the bus. It is important to note that all of these conditions are applicable only when the bus has remained inactive for the T_{AVAL} period.

1. Upon initiating a write or read operation with an IBI header, the host controller should enable SQ52912 to drive its own address on the bus. If the host controller detects a value other than the IBI header, it will stop driving the SDA line, enabling the SQ52912 to transmit its device header with the R/W bit set to 1.
2. If the host controller can accept the IBI from the device, it should acknowledge the device address, release the bus at the falling edge of the SCL, and receive the bytes transmitted by the SQ52912.
3. If the host controller is unable to accept the IBI from the device, it should send a NACK to the device address and generate a Stop condition on the bus. The SQ52912 will attempt another IBI only after the T_{AVAL} period.
4. Upon initiating a write or read operation without an IBI header to a device on the bus with a lower device address than the SQ52912, the device should cease its participation on the bus and attempt another IBI only after the T_{AVAL} period, upon detecting a mismatch.
5. If the host controller initiates a write or read operation without an IBI header to a device on the bus with a higher device address than the SQ52912, the device will win the bus arbitration, and the host will no longer be involved. The host can acknowledge the IBI by sending an ACK or ignore it by sending a NACK. In the latter case, the SQ52912 will retry another IBI only after the T_{AVAL} period.
6. If the host controller initiates a write or read transaction without an IBI header to the SQ52912, which also requests an IBI, the host or the SQ52912 can win in the bus arbitration.
7. If the host controller initiates a write transaction, it will win the bus arbitration, causing the SQ52912 to release the bus. The SQ52912 will then attempt another IBI only after T_{AVAL} period.
8. If the host controller initiates a read transaction, all the bits should match. However, at this stage, the host is anticipating an ACK from the SQ52912 for the read request, while the SQ52912 is waiting for an ACK from the host for the IBI. Consequently, a NACK will be present on the bus. In this case, the SQ52912 will retry the IBI only after the T_{AVAL} period. However, if the host issues a start (or Repeated Start) and attempts the read transaction before the T_{AVAL} period, it will receive an ACK from the SQ52912, and the host's read should win the arbitration on the bus.
9. When multiple devices attempt to initiate an IBI simultaneously, the device with the lowest device address will be the winner in the bus arbitration. If the SQ52912 detects a loss in the bus arbitration, it will retry another IBI only after the T_{AVAL} period.

In-Band Interrupt Bus Transaction

As depicted in Figure 31 andFigure *32*, when the device needs to transmit an IBI and successfully wins arbitration on the bus, receiving an ACK from the host, it is required to always send the mandatory data byte (MDB) as 8'h00. The MDB is then followed by the values of the MR51 and MR52 registers. Upon transmitting the final byte, the device should set the T-bit to 0. Subsequently, the host controller must send a Stop condition to the bus.

S	0	SA	1	0	HID2	HID1	HID0	R/W=1	ACK/NACK	
	MDB = 0x0								T=1	
	MR51 [7:0]								T=1	
	MR52 [7:0]								T=0	P

Figure 31. IBI Payload Packet with PEC Disabled

If PEC (Packet Error Checking) is enabled, following the transmission of the MR52 register value, the PEC byte is sent with the T-bit set as 0. Again, the host must send a Stop condition on the bus.

S	0	SA	1	0	HID2	HID1	HID0	R/W=1	ACK/NACK	
	MDB = 0x0								T=1	
	MR51 [7:0]								T=1	
	MR52 [7:0]								T=1	
	PEC								T=0	P

Figure 32. IBI Payload Packet with PEC Enabled

After the device asserts an IBI and successfully transmits the IBI, which includes the MDB, MR51, MR52, and PEC (if PEC mode is enabled) bytes, the IBI_STATUS bit in the MR48 register should be automatically cleared by the device.

Common Command Codes Support

The SQ52912 supports a limited set of CCCs as defined in the I3C basic specification and listed in Table 6. It only supports the CCCs specified in JESD302-1. If an unsupported CCC is encountered on a generic I3C bus, the SQ52912 will either NACK it (if possible) or ignore its associated actions. Similarly, for supported CCCs, if a CCC that doesn't apply to the current mode (I^2C or I3C) is sent, the device will disregard the corresponding actions.

Before the SQ52912 can process any device-specific read or write operation, a Stop condition on the bus is required after receiving any CCC. When processing a device-specific read or write condition, a Stop on the bus should be followed before any CCC can be issued.

The SQ52912 can receive a direct CCC with a Repeated Start condition following another direct CCC. It is acceptable to send a broadcast CCC following another broadcast CCC with a Repeated Start in between. In such cases, the device actions will only be updated following a Stop condition on the bus. However, if a direct CCC is followed by a broadcast CCC or vice versa with a Repeated Start, the behavior of the SQ52912 is not defined. For instance, in I^2C mode, it is acceptable to send a SETHID CCC, followed by a Repeated Start, then a SETAASA CCC followed by a Stop condition. However, in I3C mode, sending a direct ENEC CCC followed by a Repeated Start and then a broadcast DEVCTRL CCC is not a valid condition for the SQ52912. In this case, the host must issue a Stop after the ENEC CCC before sending a broadcast DEVCTRL CCC.

The CCC sent to the SQ52912 can be either a broadcast code or a direct code. All CCC operations require the host to send 7'h7E with R/W = 0, followed by the specific CCC and its payload bytes. For a direct CCC, the host should issue a Repeated Start on the bus after sending the CCC byte and then transmit the payload bytes.

Table 6. Supported CCCs

CCC	Operating Mode	Code	Description	Supported in I^2C Mode	Supported in I3C Mode
ENEC	Broadcast	0x00	Enable event interrupts	No	Yes
	Direct	0x80			
DISEC	Broadcast	0x01	Disable event interrupts	No	Yes
	Direct	0x81			
RSTDAA	Broadcast	0x06	Put the device in I^2C mode	No	Yes
	Direct	0x86			
SETAASA	Broadcast	0x29	Put the device in I3C Basic Mode	Yes	No
GETSTATUS	Direct	0x90	Get device status	No	Yes
DEVCAP	Direct	0xE0	Get device capability	No	Yes
SETHID	Broadcast	0x61	SQ52912 updates 3-bit HID field	Yes	No
DEVCTRL	Broadcast	0x62	Configure device	Yes	Yes

ENEC CCC

The host controller sends the ENEC CCC to enable event interrupt generation. The ENEC CCC becomes effective once the host controller issues a Stop. Upon receiving the ENEC CCC, the SQ52912 updates the IBI_ERROR_EN bit in the MR27 register to 1'b1. It is not permissible for the host controller to send the ENINT bit as 0.

The command can be issued as either a broadcast command or a direct command to the SQ52912, as depicted in Figure 33 andFigure *34*.

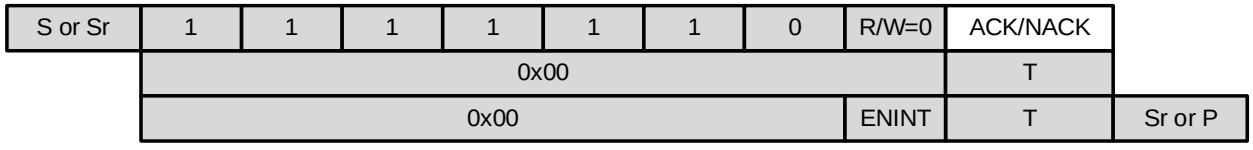


Figure 33. ENEC CCC Broadcast

Figure 34. ENEC CCC Direct

The command can be issued as either a broadcast or a direct command, as depicted in Figure 35 andFigure *36* when PEC is enabled. In this situation, the host controller must include the PEC byte, calculated on all bytes except the byte with 7'h7E and R/W = 0, after the Start or Repeated Start. If the previous transaction had a parity or PEC error and the host initiates the transaction with a Repeated Start, the SQ52912 will NACK the ENEC CCC.

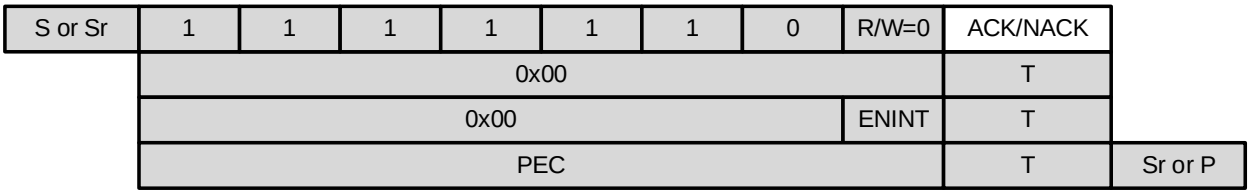


Figure 35. ENEC CCC Broadcast with PEC Enabled

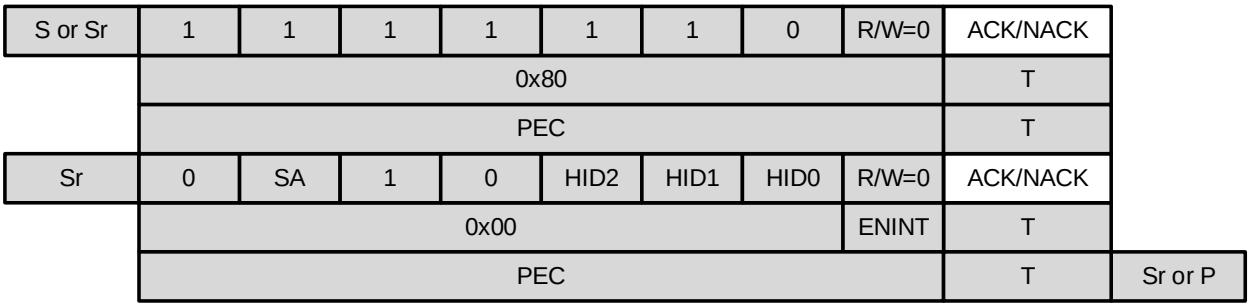


Figure 36. ENEC CCC Direct with PEC Enabled

DISEC CCC

The host controller sends the DISEC CCC to disable event interrupt generation. The DISEC CCC becomes effective once the host controller issues a Stop. Upon receiving the DISEC CCC, the SQ52912 will update the IBI_ERROR_EN bit in the MR27 register to 1'b0. It is not permissible for the host controller to send the DISINT bit as 0.

The command can be issued as either a broadcast or direct command to a specific device, as depicted in Figures 37 and 38.

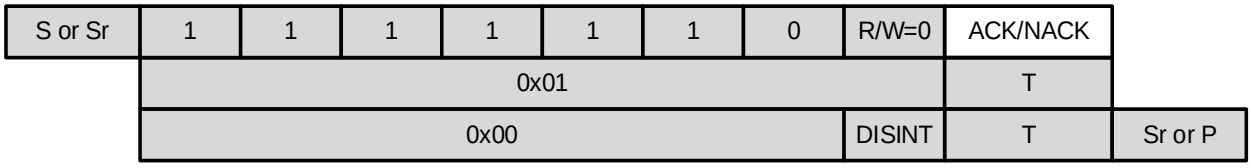


Figure 37. DISEC CCC Broadcast

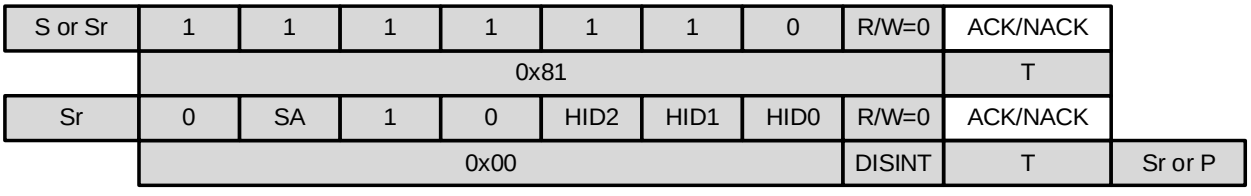


Figure 38. DISEC CCC Direct

The command can be issued as either a broadcast or a direct command, as depicted in Figure 39 andFigure *40* when PEC is enabled. In this case, the host controller must attach the PEC byte, calculated on all bytes except the byte with 7'h7E and R/W = 0, after the Start or Repeated Start.

If the previous transaction had a parity or PEC error and the host initiates the transaction with a Repeated Start, the SQ52912 will NACK the DISEC CCC.

S or Sr	1	1	1	1	1	1	0	R/W=0	ACK/NACK	
	0x01								T	
	0x00							DISINT	T	
	PEC								T	Sr or P

Figure 39. DISEC CCC Broadcast with PEC Enabled

S or Sr	1	1	1	1	1	1	0	R/W=0	ACK/NACK	
	0x81								T	
	PEC								T	
Sr	0	SA	1	0	HID2	HID1	HID0	R/W=0	ACK/NACK	
	0x00							ENINT	T	
	PEC								T	Sr or P

Figure 40. DISEC CCC Direct with PEC Enabled

RSTDAA CCC

Upon receiving the RSTDAA CCC from the host controller, the SQ52912 will transition from I3C basic mode to I²C mode. The RSTDAA CCC becomes effective after the host controller issues a Stop. Once the SQ52912 receives the RSTDAA CCC, it will carry out the following actions:

- Set the INF_SEL bit in the MR18 register to 1'b0 for I²C mode.
- Set the PEC_EN bit in the MR18 register to 1'b0 to disable PEC if it was previously enabled.
- Set the PAR_DIS bit in the MR18 register to 1'b0 to enable parity check if it was previously disabled.
- Set the IBI_ERROR_EN bit in the MR27 register to 1'b0 to disable IBI if it was previously enabled.

The command can be issued as either a broadcast or a direct command to the SQ52912 device, as depicted in Figure 41 andFigure *42*.

S or Sr	1	1	1	1	1	1	0	R/W=0	ACK/NACK	
	0x06								T	Sr or P

Figure 41. RSTDAA CCC Broadcast Command

S or Sr	1	1	1	1	1	1	0	R/W=0	ACK/NACK	
	0x86								T	
Sr	0	SA	1	0	HID2	HID1	HID0	R/W=0	ACK/NACK	Sr or P

Figure 42. RSTDAA CCC Direct Command

The command can be issued as either a broadcast or a direct command, as depicted in Figure 43 and Figure 44, when PEC is enabled. In this case, the host controller must include the PEC byte, calculated on all bytes except the byte with 7'h7E and R/W=0, after the Start or Repeated Start.

If the previous transaction encountered a parity or PEC error and the host initiates the transaction with a Repeated Start, the SQ52912 will NACK the RSTDAA CCC.

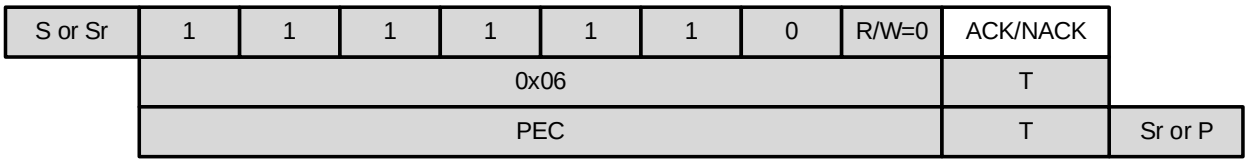


Figure 43. RSTDAA CCC Broadcast with PEC Byte

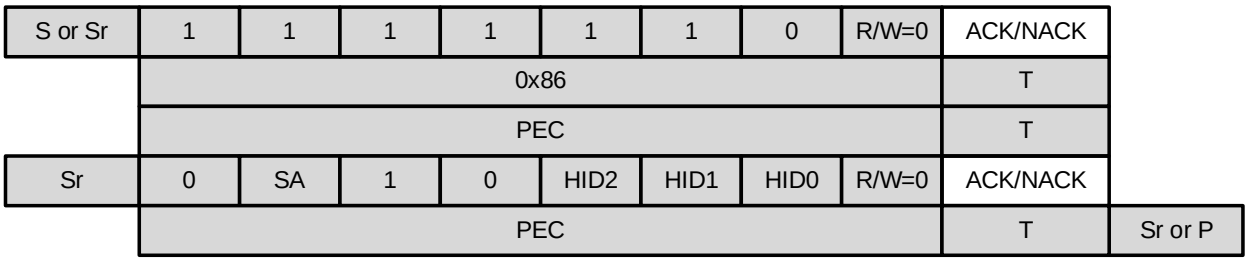


Figure 44. RSTDAA CCC Direct with PEC Byte

SETAASA CCC

Upon receiving the SETAASA CCC from the host controller, the SQ52912 will transition from I^2C mode to I3C basic mode. The SETAASA CCC becomes effective after the host controller issues a Stop. Once the SQ52912 receives the SETAASA CCC, it will set the INF_SEL bit of the MR18 register to 1'b1 for the I3C basic mode.

The SETAASA CCC is always issued as a broadcast command, as depicted in Figure 45, and does not require PEC bytes since it is only applicable during I^2C mode. If the previous CCC transaction encountered a parity error and the host initiates the transaction with a Repeated Start, the SQ52912 will NACK the SETAASA CCC.

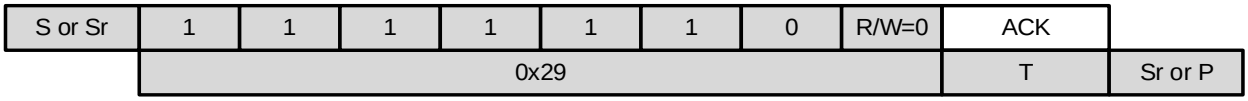


Figure 45. SETAASA CCC Command

GETSTATUS CCC

The host controller issues the GETSTATUS CCC to the SQ52912 to retrieve the status of any pending parity error, PEC error, or interrupt event. Upon receiving the GETSTATUS CCC, the SQ52912 will not clear the status bits, and the host must perform additional transactions on the bus to individually clear the status flags or write 1'b1 to the CLR_GLOBAL bit of the MR27 register.

The command is exclusively issued in direct mode, as depicted in Figure 46 when PEC is disabled and Figure 47 when PEC is enabled. In the latter case, the host controller must include the PEC byte, calculated on all bytes except the 7'h7E with R/W=0 byte, after the Start or Repeated Start. The SQ52912 calculates the PEC on the data bytes sent to the host.

If the previous transaction encountered a parity or PEC error and the host initiates the transaction with a Repeated Start, the SQ52912 will NACK the GETSTATUS CCC.

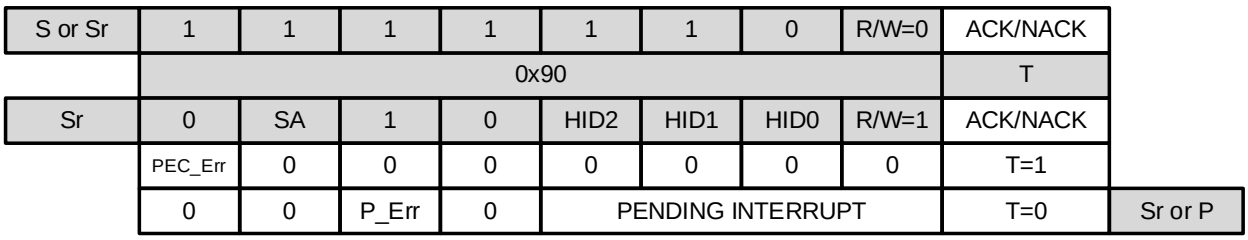


Figure 46. GETSTATUS CCC Direct Command

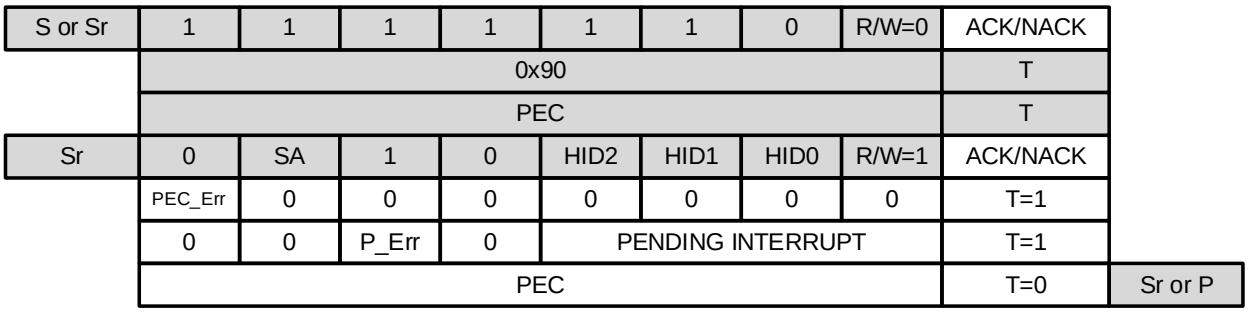


Figure 47. GETSTATUS CCC Direct with PEC Enabled

DEVCAP CCC

The host controller sends the DEVCAP CCC to the SQ52912 to obtain the supported optional device capabilities, as specified in Table 7.

The DEVCAP CCC is only issued in direct mode, as depicted in Figure 48 when PEC is disabled and in Figure 49 when PEC is enabled. In the latter case, the host controller must include the PEC byte, calculated on all bytes except the byte with 7'h7E and R/W=0, after the Start or Repeated Start. The SQ52912 calculates the PEC on the data bytes sent to the host. If the previous transaction encountered a parity or PEC error and the host initiates the transaction with a Repeated Start, the SQ52912 will NACK the DEVCAP CCC.

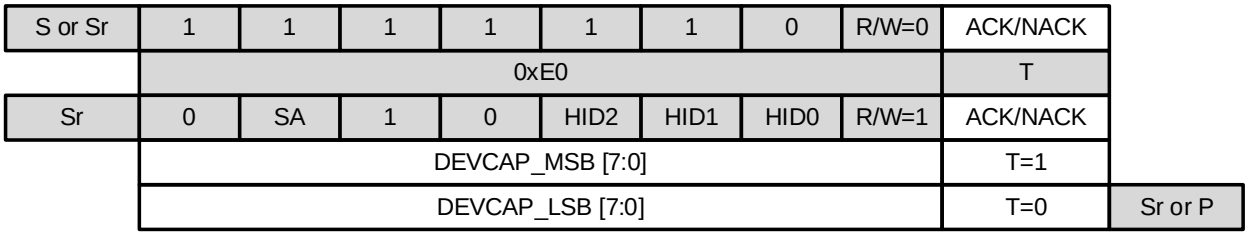


Figure 48. DEVCAP CCC Direct Command

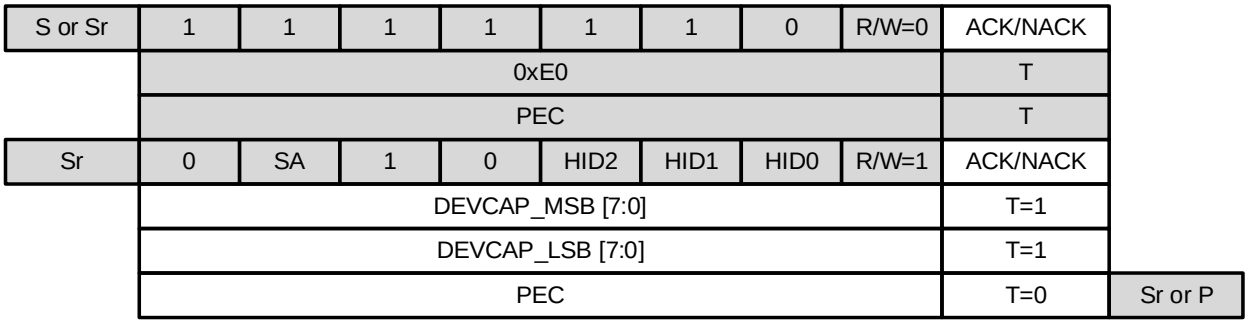


Figure 49. DEVCAP CCC Direct with PEC Byte

Table 7. DEVCAP Data Byte Description

Bit	Value	Comments
DEVCAP_MSB [7:3]	00000	Reserved
DEVCAP_MSB [2]	1	0 = Timer-based reset not supported 1 = Timer-based reset supported
DEVCAP_MSB [1:0]	00	Reserved
DEVCAP_LSB [7:0]	8'h00	Reserved

SETHID CCC

To update the HID code of the device serial address, the host controller sends the SETHID CCC to the SQ52912. The SETHID CCC becomes effective after the host controller issues a Stop. After receiving the SETHID CCC, the SQ52912 will update the MR7 register bits DEV_HID_CODE [2:0] to the value HID [2:0] provided in the CCC data payload when the host

sends a Stop bus condition. If the previous transaction encountered a parity error and the host initiates the transaction with a Repeated Start, the SQ52912 will NACK the SETHID CCC.

S or Sr	1	1	1	1	1	1	0	R/W=0	ACK	
	0x61								T	
	0	0	0	0	HID2	HID1	HID0	0	T	Sr or P

Figure 50. SETHID CCC Broadcast Command

DEVCTRL CCC

The host controller issues the DEVCTRL CCC to enable or disable operations common to devices on the bus and SQ52912 should recognize the DEVCTRL CCC.

The DEVCTRL CCC is typically issued in broadcast mode, but it can also be sent in unicast or multicast mode. The host can use the DEVCTRL CCC as a generic access with the RegMod field set to 0 or a specific register access with the RegMod field set to 1. When the RegMod field is set to 0, the structure of the DEVCTRL CCC packet is shown in Figure 51 when PEC is disabled. Figure 52 illustrates the structure of the DEVCTRL CCC when the RegMod field is set to 0 and PEC is enabled. In the latter case, the host controller is required to append the PEC byte, calculated on all bytes except the byte with 7'h7E and R/W=0, after the Start or Repeated Start.

S or Sr	1	1	1	1	1	1	0	R/W=0	ACK	
	0x62								T	
	ADDRMASK [2]	ADDRMASK [1]	ADDRMASK [0]	STOFFSET [1]	STOFFSET [0]	PECBL [1]	PECBL [0]	REGMOD=0	T	
	DEVADDR							0	T	
	DEVCTRL DATA 0								T	
	DEVCTRL DATA 1								T	
	DEVCTRL DATA 2								T	
	DEVCTRL DATA 3								T	Sr or P

Figure 51. DEVCTRL CCC with REGMOD = 0 and PEC Disabled

S or Sr	1	1	1	1	1	1	0	R/W=0	ACK	
	0x62								T	
	ADDRMASK [2]	ADDRMASK [1]	ADDRMASK [0]	STOFFSET [1]	STOFFSET [0]	PECBL [1]	PECBL [0]	REGMOD=0	T	
	DEVADDR							0	T	
	DEVCTRL DATA 0								T	
	DEVCTRL DATA 1								T	
	DEVCTRL DATA 2								T	
	DEVCTRL DATA 3								T	
	PEC								T	Sr or P

Figure 52. DEVCTRL CCC with REGMOD = 0 and PEC Enabled

Figure 53 illustrates the structure of the DEVCTRL CCC packet when PEC is disabled when the RegMod field is set to 1. When PEC is enabled, Figure 54 displays the structure of the DEVCTRL CCC for the same case. In the latter case, the host controller must append the PEC byte, calculated on all bytes except the byte with 7'h7E and R/W = 0, after the Start or Repeated Start. However, if the CMD field indicates that only one byte needs to be written, the optional register data should not be sent by the host. If the previous transaction encountered a parity or PEC error and the host initiates the transaction with a Repeated Start, the SQ52912 will NACK the DEVCTRL CCC.

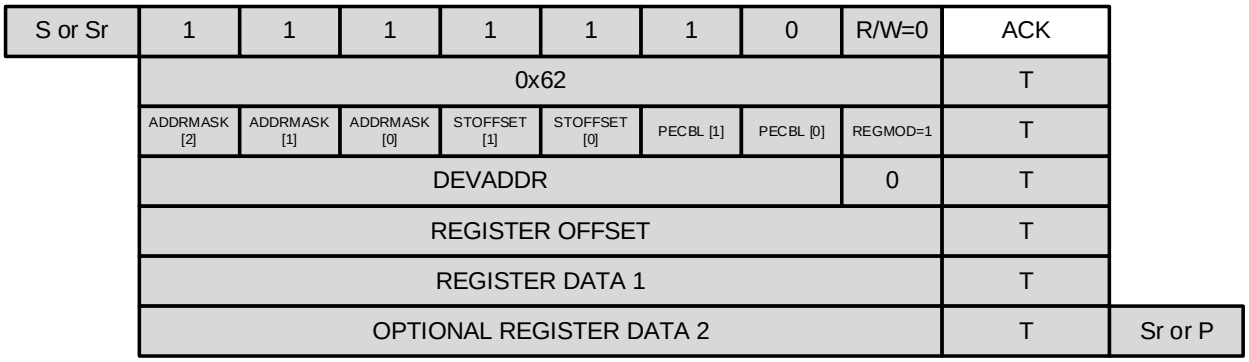


Figure 53. DEVCTRL CCC with REGMOD = 1 and PEC Disabled

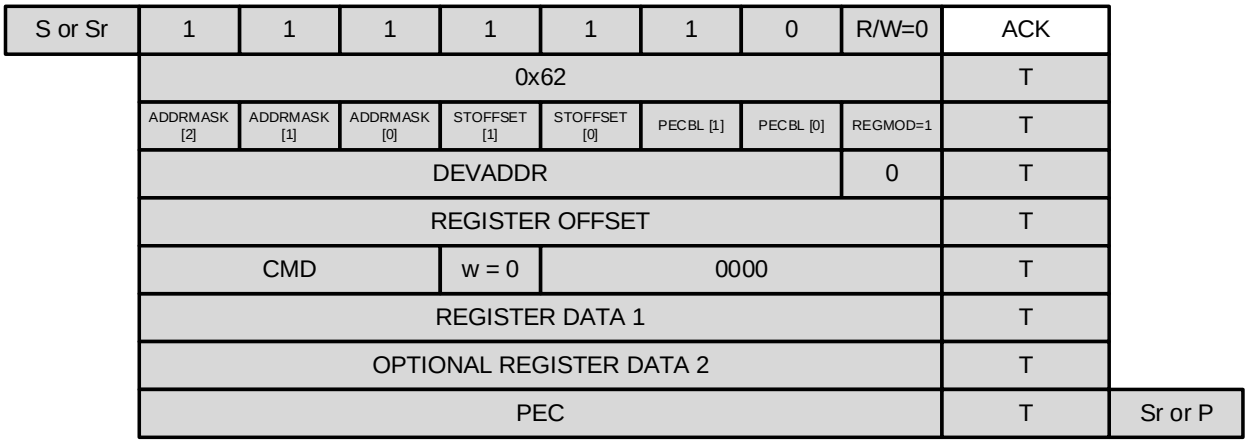


Figure 54. DEVCTRL CCC with REGMOD = 1 and PEC Enabled

Table 8 describes the definitions of the command fields.

Table 8. DEVCTRL CCC Command Definitions

Field	Description	Values	Action
ADDRMASK [2:0]	Broadcast, multicast, or unicast selection	000 = Unicast command	The SQ52912 matches the DEVADDR[6:0] field with its assigned serial address.
		011 = Multicast command	The SQ52912 matches the DEVADDR[6:3] field with its assigned LID code in the serial address.
		111 = Broadcast command	The SQ52912 ignores the DEVADDR[6:0] and performs the required action.
STOFFSET [1:0]	Start offset byte	00 = Byte 0	The SQ52912 identifies which byte is the first byte among DEVCTRL DATA 0, DEVCTRL DATA 1, DEVCTRL DATA 2, and DEVCTRL DATA 3 and updates its register accordingly. This field is valid only when REGMOD = 0.
		01 = Byte 1	
		10 = Byte 2	
		11 = Byte 3	
PECBL [1:0]	Identifies the burst length for the PEC byte position	00 = 1 Byte	The SQ52912 identifies the position of the PEC byte after the DEVCTRL DATA bytes are sent. This field is valid only when REGMOD = 0 and PEC is enabled.
		01 = 2 Byte	
		10 = 3 Byte	
		11 = 4 Byte	
REGMOD	Identifies if it is a generic or specific register access	0 = Generic Access	The SQ52912 interprets the DEVCTRL DATA byte as generic data bytes described in Table 9.
		1 = Register Access	The SQ52912 interprets the DEVCTRL DATA byte as specific register access bytes.

			If PEC is disabled, the format used for specific register access follows Figure 19. If PEC is enabled, the format used for specific register access follows Figure 21.

Table 9. Generic Data Byte Format

DEVCTRL_DATA Bit	Function	Values	Action
DEVCTRL DATA 0 [7]	PEC Enable	0 = Disable 1= Enable	MR18 register PEC_EN bit is updated
DEVCTRL DATA 0 [6]	Parity Disable	0 = Disable 1= Enable	MR18 register PAR_DIS bit is updated
DEVCTRL DATA 0 [5:0]	Reserved	Reserved	
DEVCTRL DATA 1 [7:4]	Reserved	Reserved	
DEVCTRL DATA 1 [3]	Global IBI Clear	0 = No action 1 = Clear all events and pending IBI	MR27 register CLR_GLOBAL bit is updated
DEVCTRL DATA 1 [2:0]	Reserved	Reserved	
DEVCTRL DATA 2 [7:0]	Reserved	Reserved	
DEVCTRL DATA 3 [7:0]	Reserved	Reserved	

I/O Operation

The device initially operates in I²C mode, using an open-drain I/O for its interface. However, when the device switches to the I3C mode, the I/O can be either open drain or push-pull. This dynamic switching between open drain and push-pull modes is primarily designed to support In-Band Interrupts (IBI). Table 10 details the different I/O modes for each cycle of operation.

Table 10. SQ52912 Dynamic I/O Operation for I3C Mode

Operation	Open-Drain Mode	Push-Pull Mode
Start + Device Address	Yes	No
Strat + 7'h7E IBI Header Byte	Yes	No
Repeat Start + Device Address	No	Yes
Repeat Strat + 7'h7E IBI Header Byte	No	Yes
CCC Bytes (after 7h'7E +R/W = 0 + ACK)	No	Yes
Stop	No	Yes
ACK/NACK Responses	Yes	No
Interrupt Request by SQ52912 + Device Address	Yes	No
Command and Address Operations	No	Yes
IBI Payload	No	Yes
Write Data, T-bit sequence	No	Yes
Read Data, T-bit sequence	No	Yes
PEC, T-bit sequence	No	Yes

Timing Diagrams

The SQ52912 is compatible with both I²C and I3C interfaces. Figures 2 and 4 provide descriptions of the different bus conditions supported by the device. The definitions for each of these bus conditions are listed below:

1. **Bus Idle:** Both the SDA and SCL lines remain high after a Stop condition.
2. **Start (S) condition:** A start condition is defined by the SDA line transitioning from high to low while the SCL line is high. The Start condition is preceded by a bus idle state.
3. **Stop (P) condition:** A stop condition is defined by the SDA line transitioning from low to high while the SCL line is high.
4. **Repeated Start (SR) condition:** A Repeated Start condition is defined by the SDA line transitioning from high to low while the SCL line is high, following a data transfer.
5. **Data Transfer:** The number of data bytes transferred between a Start and Stop condition, as determined by the host or device.
6. **Acknowledge:** When addressed, each receiving device is required to generate an acknowledge (ACK) bit during device address and host-to-device write transfer. A device that acknowledges must pull down the SDA line during

the acknowledge clock pulse, ensuring that the SDA line is stable low during the high period of the acknowledge clock pulse. When the host is receiving data, it can signal the end of the data transfer by generating a Not-Acknowledge (NACK) on the last byte transmitted by the target device. This behavior is as per I²C mode of operation. In I3C mode, each receiving device should only acknowledge its device address. Additionally, the host should acknowledge the device address during a successful IBI address arbitration.

7. **T-Bit:** The T-bit is only relevant in I3C mode or when the host sends a common command code (CCC) in I²C mode. It contains parity information when the host writes to the targeted device(s). If the T-bit is sampled during a read as 1 on the rising edge of the 9th clock, it indicates that the device will continue the read operation.

   If the host wants to terminate the read, it can activate the pull-up while the device drives the line high, as shown in Figure 3. When the device stops driving the line and tri-states its output, the pull-up momentarily keeps the line high until the host takes control of the bus to generate a Repeated Start and Stop, ending the read. If the host can accept more data from the device, it must not drive the line. The device samples the SDA on the falling edge of the 9th clock, and if the T-bit is sampled as 1, the device resumes driving the SDA for the next byte. If the T-bit is sampled as 0 on the rising edge of the 9th clock during a read, it indicates that the device is terminating the read, as shown in Figure 4. The host should also drive the SDA low so that when the device stops driving the line and tri-states its output, the host can take control of the bus to generate a Stop, ending the read.

Programming

This section outlines the programming model for specific operations of the SQ52912.

Enabling Interrupt Mechanism

IBI can be enabled only in I3C basic mode. Figure 55 illustrates the programming model the host controller must follow to enable IBI for the SQ52912 properly.

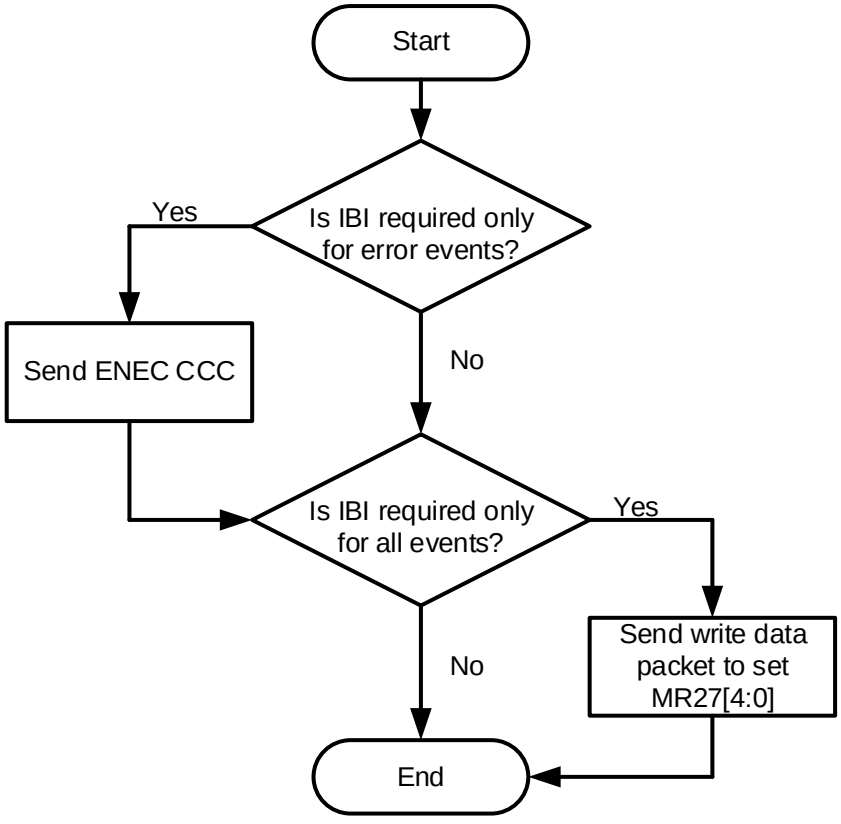


Figure 55. Interrupt Enable Flowchart

Clearing Interrupt

Although In-Band Interrupts (IBI) can be generated in I3C basic mode, the SQ52912 is designed to update the status bit for various events (excluding PEC error) even when operating in I²C mode. Figure 56 illustrates the programming model for

the host controller to clear an IBI in I3C basic mode. In I²C mode, the host controller can poll the SQ52912 by reading the register data, as explained in Section 0.

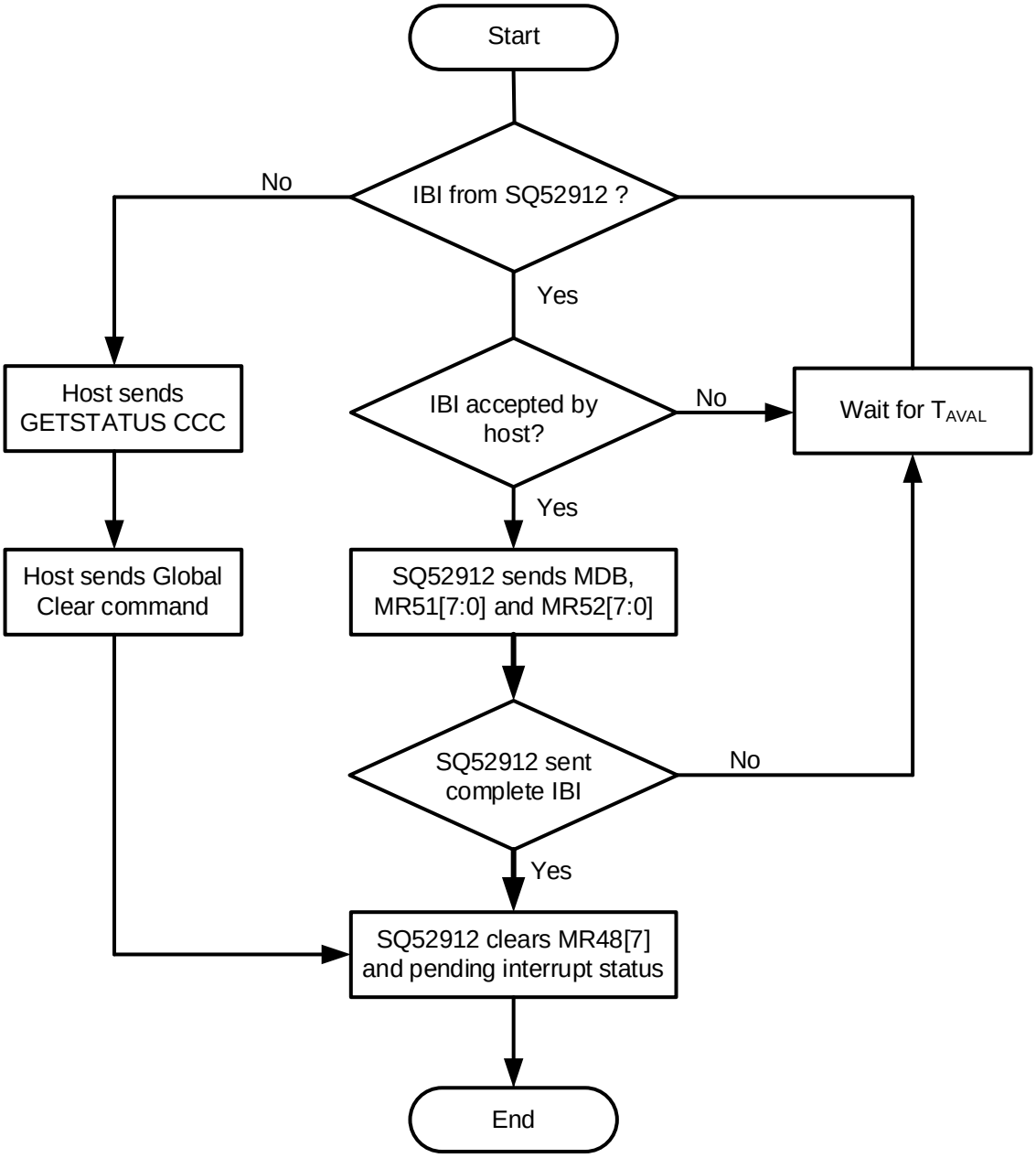


Figure 56. Interrupt Clear Process Flowchart

Register Map

Table 11. SQ52912 Register Map

Address	Type	Reset Value	Register Name	SS Description
00h	R	ACh	MR0	Device Type: Most Significant Byte
01h	R	05h	MR1	Device Type: Least Significant Byte
02h	R	02h	MR2	Device Revision
03h	R	15h	MR3	Vendor ID Byte 0
04h	R	64h	MR4	Vendor ID Byte 1
07h	RW	0Eh	MR7	Device Configuration - HID
12h	RW	00h	MR18	Device Configuration
13h	W1C	00h	MR19	Clear Register MR51 Temperature Status Command
14h	W1C	00h	MR20	Clear Register MR52 Error Status Command
1Ah	RW	00h	MR26	TS Configuration
1Bh	RW	00h	MR27	Interrupt Configurations
1Ch	RW	70h	MR28	TS Temp High Limit Configuration - Low Byte
1Dh	RW	03h	MR29	TS Temp High Limit Configuration - High Byte
1Eh	RW	00h	MR30	TS Temp Low Limit Configuration - Low Byte
1Fh	RW	00h	MR31	TS Temp Low Limit Configuration - High Byte
20h	RW	50h	MR32	TS Critical Temp High Limit Configuration – Low Byte
21h	RW	05h	MR33	TS Critical Temp High Limit Configuration - High Byte
22h	RW	00h	MR34	TS Critical Temp Low Limit Configuration – Low Byte
23h	RW	00h	MR35	TS Critical Temp Low Limit Configuration – High Byte
30h	R	00h	MR48	Device Status
31h	R	00h	MR49	TS Current Sensed Temperature - Low Byte
32h	R	00h	MR50	TS Current Sensed Temperature - High Byte
33h	R	00h	MR51	TS Temperature Status
34h	R	00h	MR52	Miscellaneous Error Status

Table 12. Register Section Access Type Codes

Access Type	Code	Description
Read type		
R	R	Read
RC	R C	Read to clear
RV	RV	Reserved for future expansion
Write type		
W	W	Write
W1C	W 1C	W 1 to clear
Reset or default value		
-n		Value after reset or the default value

MR0: Device Type, Most Significant Byte (address = 00h) [reset = ACh]

Table 13. MR0: Device Type Register

7	6	5	4	3	2	1	0
MSB_DEV_TYPE[7:0]							
R-ACh							

Table 14. MR0: Device Type Field Descriptions

Bit	Field	Type	Reset	Description
7:0	MSB_DEV_TYPE[7:0]	R	ACh	Device type, most significant byte. Used in conjunction with MR1 register.

MR1: Device Type, Least Significant Byte (address = 01h) [reset = 05h]

Table 15. MR1: Device Type Register

7	6	5	4	3	2	1	0
LSB_DEV_TYPE[7:0]							
R-05h							

Table 16. MR1: Device Type Field Descriptions

Bit	Field	Type	Reset	Description
7:0	MSB_DEV_TYPE[7:0]	R	05h	Device type, least significant byte. Used in conjunction with MR0 register. Indicates a Grade-B temperature sensor

MR2: Device Revision (address = 02h) [reset = 02h]

Table 17. MR2: Device Revision Register

7	6	5	4	3	2	1	0
Reserved		DEV_REV_MAJOR[1:0]		DEV_REV_MINOR[2:0]			Reserved
R-00		R-00		R-001			R-0

Table 18. MR2: Device Revision Field Descriptions

Bit	Field	Type	Reset	Description
7:6	Reserved	R	00	Reserved
5:4	DEV_REV_MAJOR[1:0]	R	00	Indicates the major revision number
3:1	DEV_REV_MINOR[2:0]	R	001	Indicates the minor revision number
0	Reserved	R	0	Reserved

MR3: Vendor ID Byte 0 (address = 03h) [reset = 15h]

Table 19. MR3: Vendor ID Byte 0 Register

7	6	5	4	3	2	1	0
VENDOR_ID_BYTE0[7:0]							
R-15h							

Table 20. MR3: Vendor ID Byte 0 Field Descriptions

Bit	Field	Type	Reset	Description
7:0	VENDOR_ID_BYTE0[7:0]	R	15h	Indicates the lower byte of the Vendor ID.

MR4: Vendor ID Byte 1 (address = 04h) [reset = 64h]

Table 21. MR4: Vendor ID Byte 1 Register

7	6	5	4	3	2	1	0
VENDOR_ID_BYTE1[7:0]							
R-64h							

Table 22. MR4: Vendor ID Byte 1 Field Descriptions

Bit	Field	Type	Reset	Description
7:0	VENDOR_ID_BYTE1[7:0]	R	64h	Indicates the upper byte of the Vendor ID.

MR7: Device Configuration - HID (address = 07h) [reset = 0Eh]

The MR7 register reads the HID configured by the host controller. This register can be updated only by the SETHID CCC when the device is in I²C mode, by the RSTDAA command when in I3C mode, or through a bus reset.

Table 23. MR7: Device Configuration – HID Register

7	6	5	4	3	2	1	0
Reserved				DEV_HID_CODE[2:0]			Reserved
R-0h				RW-111			R-0

Table 24. MR7: Device Configuration – HID Field Descriptions

Bit	Field	Type	Reset	Description
7:4	Reserved	R	0h	Reserved
3:1	DEV_HID_CODE[2:0]	RW	111	Device HID Code. The SQ52912 device responds to a unique 7-bit address formed by a 4-bit LID code as Table 4 and 3-bit ID code configured in this register.
0	Reserved	R	0	Reserved

MR18: Device Configuration (address = 12h) [reset = 00h]

The MR18 register configures the device features. In I3C mode, this register enables PEC and disables Parity (T-bit). It also sets the default read address mode for both I²C and I3C bus operations. The burst length for the PEC byte applies only in I3C mode; in I²C mode, the host controller must not modify this bit.

Table 25. MR18: Device Configuration Register

7	6	5	4	3	2	1	0
PEC_EN	PAR_DIS	INF_SEL	DEF_RD_ADDR_POINT_EN	DEF_RD_ADDR_POINT_Start[1:0]		DEF_ED_ADDR_POINT_BL	Reserved
RW-0	RW-0	R-0	RW-0	RW-0		RW-0	R-0

Table 26. MR18: Device Configuration Field Descriptions

Bit	Field	Type	Reset	Description
7	PEC_EN	RW	0	PEC enable 0 = PEC is disabled 1 = PEC is enabled
6	PAR_DIS	RW	0	Parity (T-bit) disable 0 = Parity or T-bit is enabled 1 = Parity or T-bit is disabled
5	INF_SEL	R	0	Interface selection 0 = I²C protocol (maximum speed of 1 MHz) 1 = I3C basic protocol
4	DEF_RD_ADDR_POINT_EN	RW	0	Default read address pointer enable 0 = Disable default read address pointer (host sets address pointer) 1 = Enable default read address pointer (address selected by MR7 register, DEF_RD_ADDR_POINT_Start[1:0] bits
3:2	DEF_RD_ADDR_POINT_Start[1:0]	RW	00	Default read address pointer starting address 00 = MR49 register 01 = Reserved 10 = Reserved 11 = Reserved
1	DEF_RD_ADDR_POINT_BL	RW	0	Burst length for read pointer address for PEC calculation 0 = 2 bytes 1 = 4 bytes
0	Reserved	R	0	Reserved

MR19: Clear MR51 Temperature Status Command (address = 13h) [reset = 00h]

The MR19 register is written by the host to clear the temperature comparison status following the most recent conversion.

Table 27. MR19: Clear MR51 Temperature Status Command Register

7	6	5	4	3	2	1	0
Reserved				CLR_TS_CRIT_LOW	CLR_TS_CRIT_HIGH	CLR_TS_LOW	CLR_TS_HIGH
R-00h				R0-W1C	R0-W1C	R0-W1C	R0-W1C

Table 28. MR19: Clear MR51 Temperature Status Command Field Descriptions

Bit	Field	Type	Reset	Description
7:4	Reserved	R	00h	Reserved
3	CLR_TS_CRIT_LOW	R0-W1C	0	Clear temperature sensor critical low status 1 = Write '1' to clear MR51 TS_CRIT_LOW_STATUS bit Writing a '0' does not affect MR51 TS_CRIT_LOW_STATUS bit
2	CLR_TS_CRIT_HIGH	R0-W1C	0	Clear temperature sensor critical high status 1 = Write '1' to clear MR51 TS_CRIT_HIGH_STATUS bit Writing a '0' does not affect MR51 TS_CRIT_HIGH_STATUS bit
1	CLR_TS_LOW	R0-W1C	0	Clear temperature sensor low status 1 = Write '1' to clear MR51 TS_LOW_STATUS bit Writing a '0' does not affect MR51 TS_LOW_STATUS bit
0	CLR_TS_HIGH	R0-W1C	0	Clear temperature sensor high status 1 = Write '1' to clear MR51 TS_HIGH_STATUS bit Writing a '0' does not affect MR51 TS_HIGH_STATUS bit

MR20: Clear MR52 Error Status Command (address = 14h) [reset = 00h]

In I3C mode only, the host writes to the MR20 register to clear error conditions when the PEC checksum is incorrect or when the last write from the host causes a parity error in the T-bit. This register is not applicable in any other mode of operation.

Table 29. MR20: Clear MR52 Error Status Command Register

7	6	5	4	3	2	1	0
Reserved						CLR_PEC_ERROR	CLR_PAR_ERROR
R-00h						W1C	W1C

Table 30. MR20: Clear MR52 Error Status Command Field Descriptions

Bit	Field	Type	Reset	Description
7:2	Reserved	R	00h	Reserved
1	CLR_PEC_ERROR	R0-W1C	0	Clear packet error status 1 = Write '1' to clear MR52 PEC_ERROR_STATUS bit Writing a '0' does not affect MR52 PEC_ERROR_STATUS bit
0	CLR_PAR_ERROR	R0-W1C	0	Clear parity error status 1 = Write '1' to clear MR52 PEC_ERROR_STATUS bit Writing a '0' does not affect MR52 PAR_ERROR_STATUS bit

MR26: TS Configuration (address = 1Ah) [reset = 00h]

The host can utilize the DIS_TS bit in the MR26 register to disable the temperature sensor. When the bit is set, the device will either halt an ongoing temperature conversion or, if a conversion is in progress, it will finish the current conversion and subsequently deactivate the temperature sensor.

Table 31. MR26: Temperature Sensor Configuration Register

7	6	5	4	3	2	1	0
Reserved							DIS_TS
R-0h							RW-0

Table 32. MR26: Temperature Sensor Configuration Field Descriptions

Bit	Field	Type	Reset	Description
7:1	Reserved	R	0h	Reserved
0	DIS_TS	RW	0	Disable temperature sensor 0 = Enable temperature sensor 1 = Disable temperature sensor

MR27: Interrupt Configuration (address = 1Bh) [reset = 00h]

Table 33. MR27: Interrupt Configuration Register

7	6	5	4	3	2	1	0
CLR_GLOBAL	Reserved		IBI_ERROR_EN	IBI_TS_CRIT_LOW_EN	IBI_TS_CRIT_HIGH_EN	IBI_TS_LOW_EN	IBI_TS_HIGH_EN
W1C	R-00		R-0	RW-0	RW-0	RW-0	RW-0

Table 34. MR27: Interrupt Configuration Field Descriptions

Bit	Field	Type	Reset	Description
7	CLR_GLOBAL	R0-W1C	0	Global clear event status and In-Band Interrupt (IBI) status 1 = Write '1' to clear the MR48, MR51 and MR52 registers. Writing a '0' does not affect MR48, MR51 and MR52 registers.
6:5	Reserved	R	0h	Reserved
4	IBI_ERROR_EN	R	0	In-band interrupt (IBI) enable for MR52 error log. 0 = Disable. Errors logged in MR52 register bits do not generate an IBI to host. 1 = Enable. Errors logged in MR52 register bits generate an IBI to host.
3	IBI_TS_CRIT_LOW_EN	RW	0	In-band interrupt (IBI) enable for temperature sensor critical low. 0 = Disable. MR51 register TS_CRIT_LOW_STATUS bit does not generate an IBI to host. 1 = Enable. MR51 register TS_CRIT_LOW_STATUS bit generates an IBI to host.
2	IBI_TS_CRIT_HIGH_EN	RW	0	In-band interrupt (IBI) enable for temperature sensor critical high. 0 = Disable. MR51 register TS_CRIT_HIGH_STATUS bit does not generate an IBI to host. 1 = Enable. MR51 register TS_CRIT_HIGH_STATUS bit generates an IBI to host.
1	IBI_TS_LOW_EN	RW	0	In-band interrupt (IBI) enable for temperature sensor low. 0 = Disable. MR51 register TS_LOW_STATUS bit does not generate an IBI to host. 1 = Enable. MR51 register TS_LOW_STATUS bit generates an IBI to host.
0	IBI_TS_HIGH_EN	RW	0	In-band interrupt (IBI) enable for temperature sensor high. 0 = Disable. MR51 register TS_HIGH_STATUS bit does not generate an IBI to host. 1 = Enable. MR51 register TS_HIGH_STATUS bit generates an IBI to host.

MR28: Temperature Sensor High Limit-Low Byte Configuration (address = 1Ch) [reset = 70h]

The status flag indicating a temperature high limit is triggered when the outcome of the temperature conversion exceeds the programmed value in the MR29 and MR28 registers. The application must guarantee that the critical temperature high limit registers are set with a value higher than the high limit registers.

Table 35. MR28: Temperature Sensor High Limit-Low Byte Configuration Register

7	6	5	4	3	2	1	0
TS_HIGH_LIMIT_LOW[7:0]							
RW-70h						R-0	R-0

Table 36. MR28: Temperature Sensor High Limit-Low Byte Field Descriptions

Bit	Field	Type	Reset	Description
7:0	TS_HIGH_LIMIT_LOW[7:0]	RW	70h	Low byte of the high limit temperature for the thermal sensor. MR29 and MR28 together define the high limit for the thermal sensor.

MR29: Temperature Sensor High Limit-High Byte Configuration (address = 1Dh) [reset = 03h]

When the outcome of the temperature conversion surpasses the programmed value in the MR29 and MR28 registers, the status flag for the temperature high limit is activated. The application must guarantee that the critical temperature high limit registers are set with a value higher than the high limit registers.

Table 37. MR29: Temperature Sensor High Limit-High Byte Configuration Register

7	6	5	4	3	2	1	0
TS_HIGH_LIMIT_HIGH[7:0]							
R-0	R-0	R-0	RW-03h				

Table 38. MR29: Temperature Sensor High Limit-High Byte Field Descriptions

Bit	Field	Type	Reset	Description
7:0	TS_HIGH_LIMIT_HIGH[7:0]	RW	03h	High byte of the high limit temperature for the thermal sensor. MR29 and MR28 together define the high limit for the thermal sensor.

MR30: Temperature Sensor Low Limit-Low Byte Configuration (address = 1Eh) [reset = 00h]

The status flag for the critical low limit is set when the temperature conversion result is less than the programmed value in the MR31 and MR30 registers. The application must guarantee that the critical temperature low limit registers have a value lower than the low limit registers.

Table 39. MR30: Temperature Sensor Low Limit-Low Byte Configuration Register

7	6	5	4	3	2	1	0
TS_LOW_LIMIT_LOW[7:0]							
RW-00h						R-0	R-0

Table 40. MR30: Temperature Sensor Low Limit-Low Byte Field Descriptions

Bit	Field	Type	Reset	Description
7:0	TS_LOW_LIMIT_LOW[7:0]	RW	00h	Low byte of the low limit temperature for the thermal sensor. MR31 and MR30 together define the low limit for the thermal sensor.

MR31: Temperature Sensor Low Limit-High Byte Configuration (address = 1Fh) [reset = 00h]

The status flag for the critical low limit is set when the temperature conversion result is less than the programmed value in the MR31 and MR30 registers. The application must guarantee that the critical temperature low limit registers have a value lower than the low limit registers.

Table 41. MR31: Temperature Sensor Low Limit-High Byte Configuration Register

7	6	5	4	3	2	1	0
TS_LOW_LIMIT_HIGH[7:0]							
R-0	R-0	R-0	RW-00h				

Table 42. MR31: Temperature Sensor Low Limit-High Byte Field Descriptions

Bit	Field	Type	Reset	Description
7:0	TS_LOW_LIMIT_HIGH[7:0]	RW	00h	The high byte of the low limit temperature for the thermal sensor. MR31 and MR30 together set the low limit for the thermal sensor.

MR32: Temperature Sensor Critical High Temperature Limit-Low Byte Configuration (address = 20h) [reset = 50h]

The status flag for the critical temperature high limit is set when the temperature conversion result is greater than the programmed value in the MR33 and MR32 registers. The application must ensure that the critical temperature high limit registers have a value greater than the high limit registers.

Table 43. MR32: Temperature Sensor Critical Temperature High Limit-Low Byte Configuration Register

7	6	5	4	3	2	1	0
TS_CRIT_HIGH_LIMIT_LOW[7:0]							
RW-50h						R-0	R-0

Table 44. MR32: Temperature Sensor Critical Temperature High Limit-Low Byte Field Descriptions

Bit	Field	Type	Reset	Description
7:0	TS_CRIT_HIGH_LIMIT_LOW[7:0]	RW	50h	Low byte of the critical high limit temperature for the thermal sensor. MR33 and MR32 together set the critical high limit temperature for the thermal sensor.

MR33: Temperature Sensor Critical Temperature High Limit-High Byte Configuration (address = 21h) [reset = 05h]

The status flag for the critical temperature high limit is set when the temperature conversion result is greater than the programmed value in the MR33 and MR32 registers. The application must ensure that the critical temperature high limit registers have a value greater than the high limit registers.

Table 45. MR33: Temperature Sensor Critical Temperature High Limit-High Byte Configuration Register

7	6	5	4	3	2	1	0
TS_CRIT_HIGH_LIMIT_HIGH[7:0]							
R-0	R-0	R-0	RW-05h				

Table 46. MR33: Temperature Sensor Critical Temperature High Limit-High Byte Field Descriptions

Bit	Field	Type	Reset	Description
7:0	TS_CRIT_HIGH_LIMIT_HIGH[7:0]	RW	05h	The high byte of the critical high limit temperature for the thermal sensor. MR33 and MR32 together set the critical high limit temperature for the thermal sensor.

MR34: Temperature Sensor Critical Temperature Low Limit-Low Byte Configuration (address = 22h) [reset = 00h]

The status flag for the critical temperature low limit is set when the temperature conversion result is less than the programmed value in the MR35 and MR34 registers. The application must guarantee that the critical temperature low limit registers have a value lower than the low limit registers.

Table 47. MR34: Temperature Sensor Critical Temperature Low Limit-Low Byte Configuration Register

7	6	5	4	3	2	1	0
TS_CRIT_LOW_LIMIT_LOW[7:0]							
RW-00h						R-0	R-0

Table 48. MR34: Temperature Sensor Critical Temperature Low Limit-Low Byte Field Descriptions

Bit	Field	Type	Reset	Description
7:0	TS_CRIT_LOW_LIMIT_LOW[7:0]	RW	00h	The low byte of the critical low limit temperature for the thermal sensor. MR35 and MR34 together set the critical low limit temperature for the thermal sensor.

MR35: Temperature Sensor Critical Temperature Low Limit-High Byte Configuration (address = 23h) [reset = 00h]

The status flag for critical temperature low limit is set when the temperature conversion result is less than the programmed value in the MR35 and MR34 registers. The application must guarantee that the critical temperature low limit registers have a value lesser than the low limit registers.

Table 49. MR35: Temperature Sensor Critical Temperature Low Limit-High Byte Configuration Register

7	6	5	4	3	2	1	0
TS_CRIT_LOW_LIMIT_HIGH[7:0]							
R-0	R-0	R-0	RW-00h				

Table 50. MR35: Temperature Sensor Critical Temperature Low Limit-High Byte Field Descriptions

Bit	Field	Type	Reset	Description
7:0	TS_CRIT_LOW_LIMIT_HIGH[7:0]	RW	00h	The high byte of the critical low limit temperature for the thermal sensor. MR35 and MR34 together set the critical low limit temperature for the thermal sensor.

MR48: Device Status (address = 30h) [reset = 00h]

The MR48 register provides the status of the IBI when the SQ52912 is in I3C mode.

Table 51. MR48: Device Status Register

7	6	5	4	3	2	1	0
IBI_STATUS	Reserved						
R-0	R-00h						

Table 52. MR48: Device Status Field Descriptions

Bit	Field	Type	Reset	Description
7	IBI_STATUS	R	0	Device event In-Band Interrupt (IBI) status. 0 = No pending IBI. 1 = Pending IBI.
6:0	Reserved	R	00h	Reserved

MR49: Current Sensed Temperature Low Byte (address = 31h) [reset = 00h]

The MR49 register stores the lower 8-bits temperature output from the most recent conversion.

Table 53. MR49: Current Sensed Temperature Low Byte Register

7	6	5	4	3	2	1	0
TS_SENSE_LOW[7:0]							
R-00h							

Table 54. MR49: Current Sensed Temperature Low Byte Field Descriptions

Bit	Field	Type	Reset	Description
7:0	TS_SENSE_LOW[7:0]	R	00h	The low byte of the current temperature returned after the most recent conversion by the thermal sensor. MR50 and MR49 together provide the temperature returned after the most recent conversion.

MR50: Current Sensed Temperature High Byte (address = 32h) [reset = 00h]

The MR50 register stores the upper 8-bits temperature output from the most recent conversion.

Table 55. MR50: Current Sensed Temperature High Byte Register

7	6	5	4	3	2	1	0
TS_SENSE_HIGH[7:0]							
R-00h							

Table 56. M50: Current Sensed Temperature High Byte Field Descriptions

Bit	Field	Type	Reset	Description
7:0	TS_SENSE_HIGH[7:0]	R	00h	The high byte of the current temperature returned after the most recent conversion by the thermal sensor. MR49 and MR50 together provide the temperature returned after the most recent conversion.

MR51: Temperature Status (address = 33h) [reset = 00h]

The MR51 register stores the most recent temperature comparison status against the four threshold levels defined in MR28 through MR35.

Table 57. MR51: Temperature Status Register

7	6	5	4	3	2	1	0
Reserved				TS_CRIT_LOW_STATUS	TS_CRIT_HIGH_STATUS	TS_LOW_STATUS	TS_HIGH_STATUS
R-0h				R-0	R-0	R-0	R-0

Table 58. MR51: Temperature Status Field Descriptions

Bit	Field	Type	Reset	Description
7:4	Reserved	R	00h	Reserved
3	TS_CRIT_LOW_STATUS	R	0	Temperature sensor critical low status. 0 = Temperature is above the limit set in registers MR35 and MR34. 1 = Temperature is below the limit set in registers MR35 and MR34.
2	TS_CRIT_HIGH_STATUS	R	0	Temperature sensor critical high status. 0 = Temperature is below the limit set in registers MR33 and MR32. 1 = Temperature is above the limit set in registers MR33 and MR32.
1	TS _LOW_STATUS	R	0	Temperature sensor low status. 0 = Temperature is above the limit set in registers MR31 and MR30. 1 = Temperature is below the limit set in registers MR31 and MR30.
0	TS _HIGH_STATUS	R	0	Temperature sensor high status. 0 = Temperature is below the limit set in registers MR29 and MR28. 1 = Temperature is above the limit set in registers MR29 and MR28.

MR52: Miscellaneous Error Status (address = 34h) [reset = 00h]

The MR52 register stores the status of PEC checksum failures when PEC mode is enabled and parity errors on the T-bit when the host writes to the device in I3C mode.

Table 59. MR52: Miscellaneous Error Status Register

7	6	5	4	3	2	1	0
Reserved						PEC_ERROR_STATUS	PAR_ERROR_STATUS
R-00h						R-0	R-0

Table 60. MR52: Miscellaneous Error Status Field Descriptions

Bit	Field	Type	Reset	Description
7:2	Reserved	R	00h	Reserved
1	PEC_ERROR_STATUS	R	0	Packet error status. 0 = No PEC error. 1 = PEC error in one or more packets.
0	PAR_ERROR_STATUS	R	0	Parity check error status

				0 = No parity error. 1 = Parity error in one or more bytes.

Application Schematic

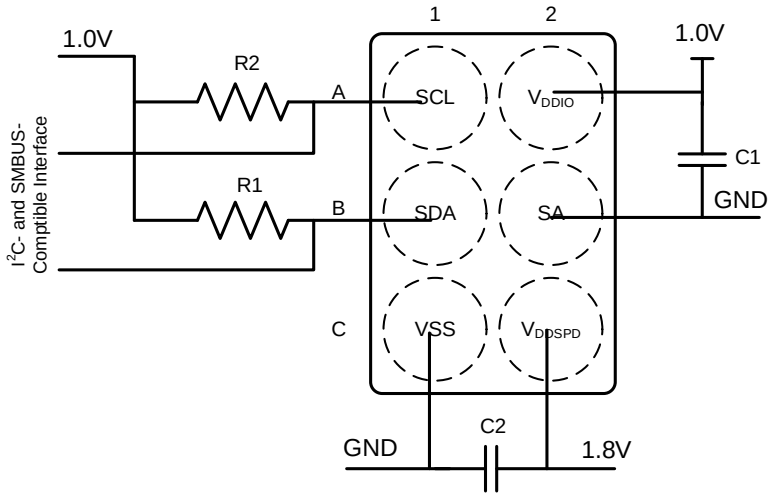


Figure 57 I2C /SMBus Application Schematic

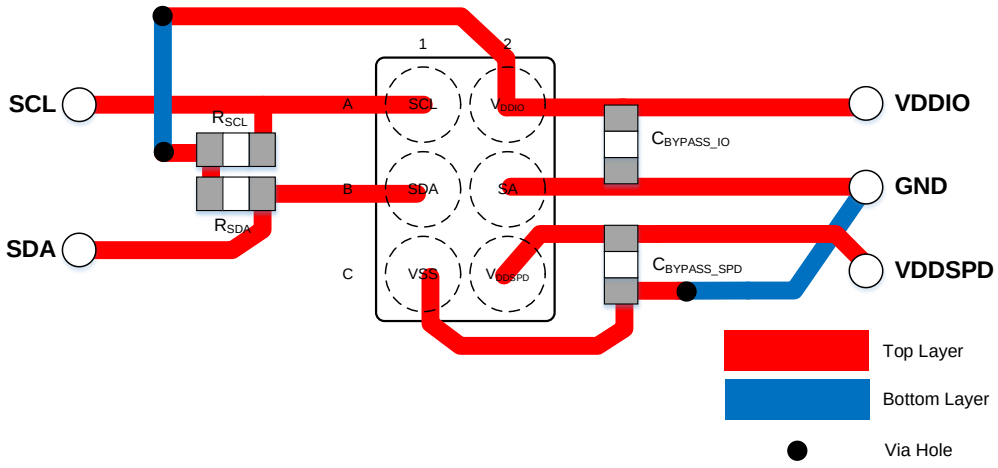
BOM List

Designator	Description	Part Number	MFR
C1, C2	100nF/50V, 0603		
R1, R2	2kΩ, 0603, 1%		

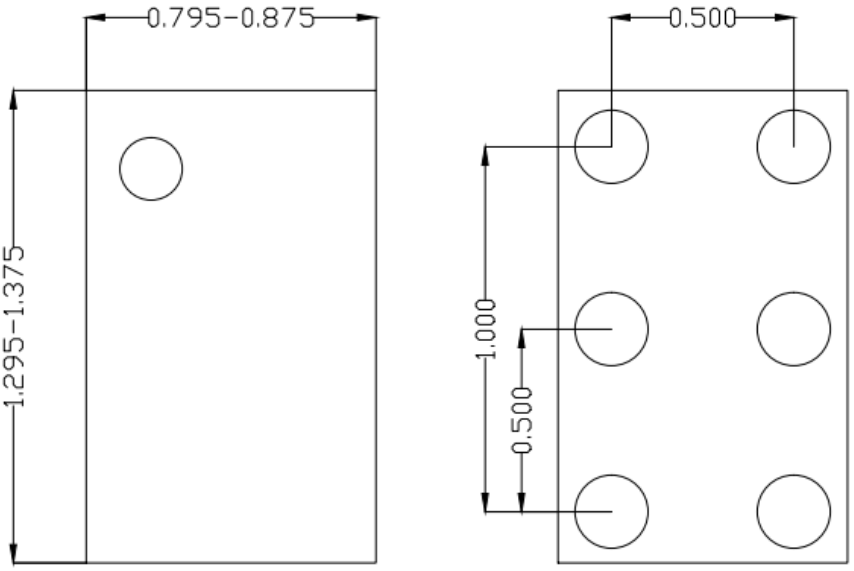
Layout Design

For optimal design, follow these PCB layout guiedlines:

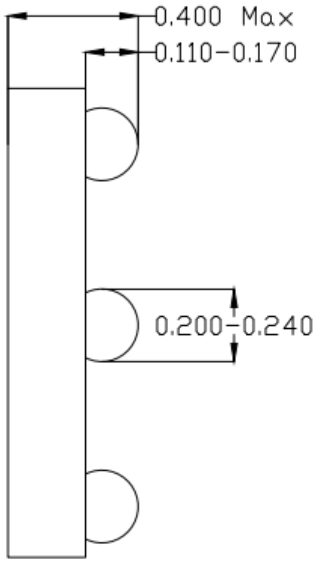
- Place the bypass capacitor (a 0.1μF MLCC is recommended) as close as possible to the VDDSPD and GND pins.
- Place the bypass capacitor (a 0.1μF MLCC is recommended) as close as possible to the VDDIO and GND pins.
- Thermal contact is required between the part and the sensed region. For improved thermal contact, a wide copper trace has to be provided between the GND pin of the temperature sensor and the area being monitored.
- The sensor has to be placed in close proximity to the parts that dissipate heat.



CSP1.338×0.838-6 Package Outline Drawing



Top View **Bottom View**

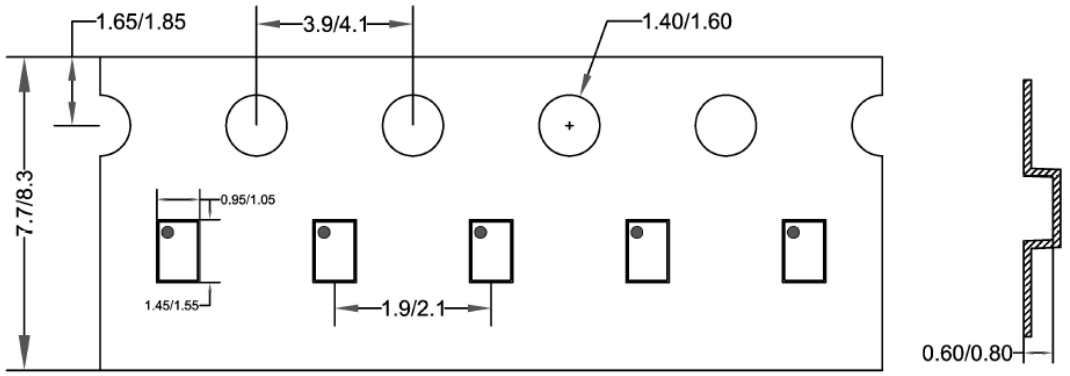


Side View

Note: All dimensions are in millimeters and exclude mold flash and metal burr.

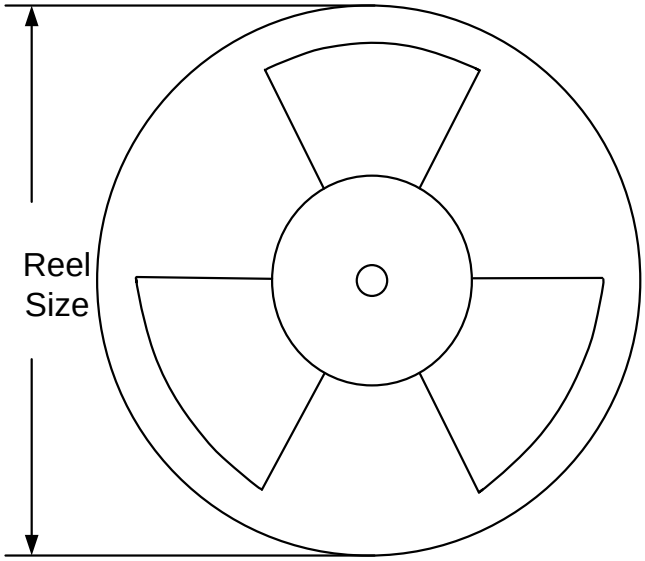
Tape and Reel Information

Tape Dimensions and Pin 1 Orientation



Direction of Feed ⟶

Reel Dimensions



Package type	Tape width (mm)	Pocket pitch (mm)	Reel size (Inch)	Trailer * length (mm)	Leader * length (mm)	Qty per reel (pcs)
CSP1.338*0.838	8	2	7"	280	160	10000

Revision History

The revision history provided is for informational purposes only and is believed to be accurate; however, not warrantied. Please make sure that you have the latest revision.

Date	Revision	Change
Oct. 26, 2024	Revision 1.0	Initial Release

IMPORTANT NOTICE

1. **Right to make changes.** Silergy and its subsidiaries (hereafter Silergy) reserve the right to change any information published in this document, including but not limited to circuitry, specification and/or product design, manufacturing or descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products are sold subject to Silergy's standard terms and conditions of sale.

2. Applications. Application examples that are described herein for any of these products are for illustrative purposes only. Silergy makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification. Buyers are responsible for the design and operation of their applications and products using Silergy products. Silergy or its subsidiaries assume no liability for any application assistance or designs of customer products. It is customer's sole responsibility to determine whether the Silergy product is suitable and fit for the customer's applications and products planned. To minimize the risks associated with customer's products and applications, customer should provide adequate design and operating safeguards. Customer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Silergy assumes no liability related to any default, damage, costs or problem in the customer's applications or products, or the application or use by customer's third-party buyers. Customer will fully indemnify Silergy, its subsidiaries, and their representatives against any damages arising out of the use of any Silergy components in safety-critical applications. It is also buyers' sole responsibility to warrant and guarantee that any intellectual property rights of a third party are not infringed upon when integrating Silergy products into any application. Silergy assumes no responsibility for any said applications or for any use of any circuitry other than circuitry entirely embodied in a Silergy product.

3. **Limited warranty and liability.** Information furnished by Silergy in this document is believed to be accurate and reliable. However, Silergy makes no representation or warranty, expressed or implied, as to the accuracy or completeness of such information and should have no liability for the consequences of use of such information. In no event should Silergy be liable for any indirect, incidental, punitive, special or consequential damages, including but not limited to lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges, whether or not such damages are based on tort or negligence, warranty, breach of contract or any other legal theory. Notwithstanding any damages that customer might incur for any reason whatsoever, Silergy' aggregate and cumulative liability towards customer for the products described herein should be limited in accordance with the Standard Terms and Conditions of Sale of Silergy.

4. **Suitability for use.** Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of Silergy components in its applications, notwithstanding any applications-related information or support that may be provided by Silergy. Silergy products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of a Silergy product can reasonably be expected to result in personal injury, death or severe property or environmental damage. Silergy assumes no liability for inclusion and/or use of Silergy products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

5. **Terms and conditions of commercial sale**. Silergy products are sold subject to the standard terms and conditions of commercial sale, as published at https://www.silergy.com, unless otherwise agreed in a valid written individual agreement specifically agreed to in writing by an authorized officer of Silergy. In case an individual agreement is concluded only the terms and conditions of the respective agreement should apply. Silergy hereby expressly objects to and denies the application of any customer's general terms and conditions with regard to the purchase of Silergy products by the customer.

6. **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. Silergy makes no representation or warranty that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right. Information published by Silergy regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from Silergy under the patents or other intellectual property of Silergy.

For more information, please visit: www.silergy.com