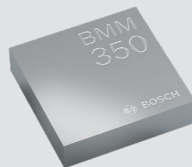


# BMM350

## 3-axis magnetic sensor with high data rate



### **BST-BMM350-DS001-21**

Document revision	1.21
Document release date	July 2023
Document number	BST-BMM350-DS001-21
Sales Part Number	0273.017.027

Notes Data and descriptions in this document are subject to change without notice. Product photos and pictures are for illustration purposes only and may differ from the real product appearance.

## 1. Basic Description

The BMM350 is a 3-axis magnetic sensor which operates in automatic mode or triggered mode. The magnetic-to-digital conversion technology is based on TMR (tunnel magneto resistance)

The BMM350 is designed to meet all requirements for high performance consumer applications such as magnetic compass, virtual, augmented and mixed reality applications and high-end gaming, platform stabilization applications such as image stabilization, or indoor navigation and dead-reckoning, for example in robotics applications.

Evaluation circuitry converts the output of the magnetic transducer sensing structures, developed, produced and tested in BOSCH facilities. The corresponding chip-sets are packed into one single WLCSP 1.28mm x 1.28mm x 0.5mm housing. For optimum system integration the BMM350 is fitted with digital interfaces (I2C, I3C), offering a wide VDDIO voltage range from 1.72V to 3.6V. To provide maximum performance and reliability each device is tested and ready-to-use calibrated.

The BMM350 has an excellent temperature behaviour with an outstanding low temperature coefficients of the offset (TCO) and of the sensitivity (TCS).

An API is provided for quick integration into the target system.

### Key features

Three-axis magnetic field sensor

- Ultra-Small package **Wafer Level Chip Scale Package**  
(9 pins, 0.4mm ball pitch)  
footprint 1.28 x 1.28 mm<sup>2</sup>, height 0.5 mm
- Digital interface I2C, I3C, 1 interrupt pin, 1 pin for I2C/I3C (legacy) address definition
- Low voltage operation V<sub>DD</sub> supply voltage range: 1.72V to 1.98V  
V<sub>DDIO</sub> interface voltage range: 1.72V to 3.6V
- Flexible functionality Magnetic field range typical:  
±2000µT (x, y-axis), ±2000µT (z-axis)  
Magnetic field resolution of ~0.1µT
- On-chip interrupt controller Interrupt-signal generation for new data
- Ultra-low power Low current consumption (typ. 200µA @ 100 Hz in regular power preset),  
short wake-up time
- Normal and forced mode periodic operation (normal mode) and triggered operation (forced mode)
- Temperature range -40 °C ... +85 °C
- RoHS compliant, halogen-free

## Typical applications

- Magnetic heading information
- Tilt-compensated electronic compass for map rotation, navigation and augmented reality
- Gyroscope calibration in 9-DoF applications for mobile devices
- In-door navigation, e.g. step counting in combination with accelerometer
- Gaming (AR/VR)

## General Description

The BMM350 is a standalone geomagnetic sensor for consumer market applications. It allows measurements of the magnetic field in three perpendicular axes. Based on Bosch's proprietary TMR technology, performance and features of BMM350 are carefully tuned and perfectly match the demanding requirements of all 3-axis mobile applications such as electronic compass, navigation or augmented reality.

An evaluation circuitry (ASIC) converts the output of the geomagnetic sensor to digital results which can be read out over the digital interfaces I2C or I3C.

Package and interfaces of the BMM350 have been designed to match a multitude of hardware requirements. As the sensor features an ultra-small footprint and a flat package, it is ingeniously suited for mobile applications. The wafer level chip scale package (WLCSP) with dimensions of only 1.28 x 1.28 x 0.5 mm<sup>3</sup> ensures high flexibility in PCB placement.

The BMM350 offers low voltage operation ( $V_{DD}$  voltage range from 1.72V to 1.98V,  $V_{DDIO}$  voltage range 1.72V to 3.6V) and can be programmed to optimize functionality, performance and power consumption in customer specific applications.

The BMM350 senses the three axis of the terrestrial field in cell phones, handhelds, computer peripherals, man-machine interfaces, virtual reality features and game controllers.

## Table of Contents

<b>1. Basic Description</b> .....	<b>2</b>
<b>2. Specification</b> .....	<b>8</b>
2.1 Electrical operation conditions .....	8
2.2 Magnetometer output signal specification .....	9
<b>3. Absolute maximum ratings</b> .....	<b>11</b>
<b>4. Block diagram</b> .....	<b>12</b>
<b>5. Functional description</b> .....	<b>13</b>
5.1 Power modes, boot procedure .....	14
5.1.1 Power off.....	14
5.1.2 Suspend mode.....	14
5.1.3 Normal mode .....	14
5.1.4 Forced mode.....	15
5.1.5 Magnetic reset in normal and forced mode/excessive field strengths handling .....	17
5.1.6 Self Test .....	17
5.2 Compensated data access from magnetic and temperature registers.....	18
5.2.1 Accessing compensation data.....	18
5.2.2 Sensortime .....	18
5.3 Excessive field strengths handling .....	19
5.4 Data rate and noise performance setting .....	19
5.5 Interrupt settings and interrupt status .....	20
5.5.1 Mapping Interrupt to INT pin .....	20
5.5.2 Interrupt polarity at INT pin .....	20
5.5.3 Interrupt drive type: CMOS or open drain.....	20
5.5.4 Interrupt latching type or non-latching .....	20
5.5.5 In-band interrupt .....	20
5.6 Pad drive settings .....	21
5.7 Enable/disable axis.....	21
5.8 Sensortime always available .....	21
5.9 I2C watchdog setting .....	21
5.10 Status information.....	21
5.10.1 CHIP_ID.....	21
5.10.2 ERR_REG: PMU and boot error.....	22
5.10.3 PMU_CMD_STATUS: PMU (command) status .....	22

5.10.4	I3C_ERR: I3C status/error reporting .....	22
5.10.5	Soft reset, reset detection.....	22
<b>6.</b>	<b>Pin and package description .....</b>	<b>23</b>
5.1	Landing pattern recommendation.....	24
5.2	Connection diagrams.....	24
5.3	Recommendations for PCB design .....	24
<b>7.</b>	<b>API functions .....</b>	<b>25</b>
<b>8.</b>	<b>Register Map.....</b>	<b>26</b>
8.1	Register (0x00) CHIP_ID.....	29
8.2	Register (0x02) ERR_REG.....	29
8.3	Register (0x03) PAD_CTRL.....	30
8.4	Register (0x04) PMU_CMD_AGGR_SET .....	30
8.5	Register (0x05) PMU_CMD_AXIS_EN.....	31
8.6	Register (0x06) PMU_CMD .....	32
8.7	Register (0x07) PMU_CMD_STATUS_0 .....	33
8.8	Register (0x08) PMU_CMD_STATUS_1 .....	34
8.9	Register (0x09) I3C_ERR.....	34
8.10	Register (0x0A) I2C_WDT_SET.....	34
8.11	Register (0x2E) INT_CTRL .....	35
8.12	Register (0x2F) INT_CTRL_IBI .....	36
8.13	Register (0x30) INT_STATUS.....	36
8.14	Register (0x31) MAG_X_XLSB .....	37
8.15	Register (0x32) MAG_X_LSB.....	37
8.16	Register (0x33) MAG_X_MSB.....	37
8.17	Register (0x34) MAG_Y_XLSB .....	38
8.18	Register (0x35) MAG_Y_LSB.....	38
8.19	Register (0x36) MAG_Y_MSB.....	38
8.20	Register (0x37) MAG_Z_XLSB .....	38
8.21	Register (0x38) MAG_Z_LSB.....	39
8.22	Register (0x39) MAG_Z_MSB.....	39
8.23	Register (0x3A) TEMP_XLSB.....	39
8.24	Register (0x3B) TEMP_LSB.....	39
8.25	Register (0x3C) TEMP_MSB.....	40
8.26	Register (0x3D) SENSORTIME_XLSB .....	40
8.27	Register (0x3E) SENSORTIME_LSB.....	40
8.28	Register(0x3F) SENSORTIME_MSB .....	40
8.29	Register (0x50) OTP_CMD_REG.....	41
8.30	Register (0x52) OTP_DATA_MSB_REG.....	41
8.31	Register (0x53) OTP_DATA_LSB_REG.....	41

8.32	Register (0x55) OTP_STATUS_REG .....	42
8.33	Register (0x60) TMR_SELFTEST_USER .....	42
8.34	Register (0x61) CTRL_USER.....	43
8.35	Register (0x7E) CMD.....	43
<b>9.</b>	<b>Digital Interfaces .....</b>	<b>44</b>
9.1	Interface selection.....	44
9.2	I <sup>2</sup> C interface .....	44
9.2.1	I <sup>2</sup> C write .....	45
9.2.2	I <sup>2</sup> C read.....	46
9.2.3	Dummy bytes in I2C mode .....	46
9.3	I3C interface .....	46
9.3.1	Supported CCC .....	47
9.3.2	Dummy bytes in I3C mode .....	48
9.3.3	I3C provisional ID .....	48
9.3.4	I3C read timing diagrams .....	49
9.3.5	I3C write timing diagrams .....	50
9.4	Interface parameter specification .....	50
9.4.1	General interface parameters.....	50
9.4.2	I <sup>2</sup> C timings.....	50
9.4.3	I3C timings.....	52
<b>10.</b>	<b>Package information.....</b>	<b>53</b>
10.1	Marking/Lasermarking, coordinate system.....	54
<b>11.</b>	<b>Examples of use .....</b>	<b>56</b>
<b>12.</b>	<b>Timing control .....</b>	<b>57</b>
12.1	Timing control Asynchronous mode .....	57
<b>13.</b>	<b>Recommendations for external capacitor CRST .....</b>	<b>58</b>
<b>14.</b>	<b>Environmental Safety.....</b>	<b>59</b>
14.1	RoHS .....	59
14.2	Halogen content.....	59
14.3	Internal Package Structure .....	59
<b>15.</b>	<b>Acronyms.....</b>	<b>60</b>
<b>16.</b>	<b>Legal disclaimer .....</b>	<b>61</b>
<b>17.</b>	<b>Document history and modification.....</b>	<b>62</b>

## List of figures

Figure 1: Block diagram .....	12
Figure 2: Power mode transitions.....	16
Figure 3: View from lasermarking side, balls invisible, required external components in operation.....	23
Figure 4: Pin names only, laser marking side .....	23
Figure 5: I2C connection diagram .....	24
Figure 6: I <sup>2</sup> C multiple byte write (not auto-incremented) .....	46
Figure 7: I <sup>2</sup> C single byte write.....	46
Figure 8: I <sup>2</sup> C multiple byte read, first 2 bytes transferred must be discarded, they are dummy bytes.....	46
Figure 9: I3C data package .....	46
Figure 10: I <sup>2</sup> C timing diagram.....	51
Figure 11: Pin out and package dimensions .....	53
Figure 12: Marking .....	54
Figure 13: Coordinate system definition, right-handed. lasermarking side up.....	55
Figure 14: Timing control async mode, from MIPI I3C spec .....	57
Figure 15. External capacitor .....	58

## List of tables

Table 1: Electrical operation conditions .....	8
Table 2: magnetometer property specification .....	9
Table 3: Magnetometer output signal.....	10
Table 4: Absolute maximum ratings .....	11
Table 5: Allowed combinations of output data rate ODR and noise configuration/averaging and related current consumption in $\mu\text{A}^*$ . .....	15
Table 6: Highest forced mode trigger rate versus noise performance/ averaging setting .....	16
Table 7: Power consumption versus data rate ODR and noise performance setting, allowed combinations of settings. n/a: not allowed. ....	19
Table 8 : 8-bit portrait register map .....	26
Table 9: I2C address definition.....	44
Table 10: I2C address roll-out.....	45
Table 11: MIPI I3C CCC supported commands.....	47
Table 12: Interface parameters .....	50
Table 13: I <sup>2</sup> C timings .....	51
Table 14: I3C timings .....	52

## 2. Specification

If not stated otherwise, the given values with minimum/maximum values are mean  $\pm 3\sigma$ , typical values are mean  $\pm 1\sigma$ . Room temperature (RT) refers to 25°C, a full-scale sweep (FSS) refers to  $\pm 2000 \mu\text{T}$ . Values are given after API correction. Data refers to parts soldered on PCB and without underfill or encapsulation.

### 2.1 Electrical operation conditions

Table 1: Electrical operation conditions

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Supply Voltage Internal Domains	VDD		1.72		1.98	V
Supply Voltage I/O Domain	VDDIO		1.72		3.6	V
Voltage Input Low Level	$V_{IL,a}$	I2C&I3C			0.3VDDIO	-
Voltage Input High Level	$V_{IH,a}$	I2C&I3C	0.7VDDIO			-
Voltage Output Low Level	$V_{OL,a}$	$I_{OL} \leq 2\text{mA}$			0.23VDDIO	-
Voltage Output High Level	$V_{OH}$	$I_{OH} \leq 2\text{mA}$	0.8VDDIO			-
Operating Temperature	$T_A$		-40		+85	°C
Current in normal mode	$I_{dd,n,rp}$ $I_{DD,ref}$	ODR=100, 25°C, averaging=2 samples (regular power setting)		200		$\mu\text{A}$
	$I_{dd,n,ln}$	ODR=100, 25°C, averaging=4 samples (low noise)		350		$\mu\text{A}$
Current in suspend mode	$I_{dd,sus}$	data retention		1.8		$\mu\text{A}$
Peak supply current in active mode (normal or forced mode)	$I_{DDpk,m}$	In measurement phase $T_A=25^\circ\text{C}$ , before bit reset		3.5		mA
Sensor start-up time (suspend to normal mode) <sup>1</sup>	$T_{sus2nm}$	Start-up time from suspend mode to normal mode		70		ms
Sensor start-up time (OFF to suspend mode)	$T_{pon2sus}$	Start-up time from OFF to suspend mode		2.5		ms



## 2.2 Magnetometer output signal specification

Table 2: magnetometer property specification

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Magnetic field range, all axes	$R_{ng}$		-2000		2000	$\mu T$
Zero-field offset drift before or after soldering	$off_{,ma}$	25°C		$\pm 25$		$\mu T$
Zero-B offset	$OFF_{m,cal}$	After software calibration with Bosch Sensortec eCompass software <sup>1</sup> -40°C ≤ T <sub>A</sub> ≤ +85°C		$\pm 2$		$\mu T$
ODRs in normal mode	ODR	At room temperature, after soldering	Typ -2 %	400 200 100 50 25 25/2 25/4 25/8 25/16	Typ+2%	Hz
Maximum forced mode trigger frequency	$FMT_f$	averaging=0 only		200		Hz
Relative ODR drift over temperature	$ODR_{,dev,lr}$	Reduced temperature sweep [-5°C, 65°C]	-3		3	%
	$ODR_{,dev,fr}$	Full temperature sweep [-40°C, 85°C]	-4		4	%
3dB bandwidth	BW	by design		ODR/2		Hz

<sup>1</sup>Magnetic zero-B offset assuming calibration with Bosch Sensortec eCompass software. Typical value after applying calibration movements containing various device orientations (typical device usage).

Table 3: Magnetometer output signal

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Sensitivity/gain error after soldering <sup>2</sup>	$G_{err,m}$	After API compensation $T_A=25^\circ\text{C}$ Nominal $V_{DD}$ supplies x/y axis		+/-1		%
Sensitivity/gain error after soldering <sup>3</sup>	$G_{err,m}$	After API compensation $T_A=25^\circ\text{C}$ Nominal $V_{DD}$ supplies Z axis		+/-3		%
Sensitivity Temperature Drift	$TCS_m$	After API compensation $-40^\circ\text{C} \leq T_A \leq +85^\circ\text{C}$ Nominal $V_{DD}$ supplies all axis		+/-0.010		%/K
TCO error	TCOerr	$-40^\circ\text{C}$ to $85^\circ\text{C}$		$\pm 200$		nT/K
Hysteresis <sup>4</sup>	$hyst_{xy, \%FS}$	RT, after FSS, in % of FSS		0.02		%
	$hyst_z, \%FS$	RT, after FSS, no reset, in % of FSS		0.4		%
Integral Nonlinearity	$INL_{m, FS}$	Max. Deviation to best fit straight line x/y axis		$\pm 10$		$\mu\text{T}$
Integral Nonlinearity	$INL_{m, FS}$	Max. Deviation to best fit straight line z axis		$\pm 20$		$\mu\text{T}$
Output noise rms 3dB BW= ODR/2	$n_{rms,xy}$ x.y channel	ODR=100 averaging=2 samples		$\pm 190$		nTrms
	$n_{rms,z}$ z channel	ODR=100 averaging=2 samples		$\pm 450$		nTrms
Sensitivity drift after magnetic field shock recovery		250 mT	-100		+100	$\mu\text{T}$
Sensitivity drift after magnetic field shock recovery		250 mT	-1		+1	%

<sup>2</sup> Definition: gain error = ( measured field after API compensation ) / ( applied field ) - 1, gain measured between +/-1.2 mT

<sup>3</sup> Definition: gain error = ( measured field after API compensation ) / ( applied field ) - 1, gain measured between +/-1.2 mT

<sup>4</sup> BMM350 z-axis hysteresis after full-scale sweep can be cleared using magnetic reset.

### 3. Absolute maximum ratings

The absolute maximum ratings are provided in Table 4. At or above these maximum ratings operability is not given. The specification limits in chapter 2 only apply under normal operating conditions.

Table 4: Absolute maximum ratings

Parameter	Condition	Min	Max	Unit
Voltage at Supply Pin	V <sub>DD</sub> Pin	-0.3	2	V
	V <sub>DDIO</sub> Pin	-0.3	3.6	V
Voltage at any Logic Pad	Non-Supply Pin	-0.3	V <sub>DDIO</sub> + 0.3	V
Operating Temperature, T <sub>A</sub>	Active operation	-40	+85	°C
Passive Storage Temp. Range	≤ 65% rel. H.	-50	+125	°C
None-volatile memory (OTP) Data Retention	T = 85°C	10		year
Mechanical Shock according to JESD22-B104C	Duration ≤ 500µs		20,000	g
	Drop in reel onto hard surfaces		2	m
ESD	HBM, at any Pin		±2	kV
	CDM		±500	V
Latch-up	LU		±100	mA
Magnetic field	Any direction <sup>5</sup>	-250	250	mT

<sup>5</sup> Field shock strengths up to BMFS = 400 mT tested; fields of 250 mT < BMFS < 400 mT will not damage the device but may lead to offset and sensitivity drifts above 100 µT and 1%, respectively.

## 4. Block diagram

Figure 1 shows the signal flow block diagram of the BMM350.

A magnetic transducer converts the magnetic field information in X, Y and Z direction into a differential voltage. An additional temperature transducer converts the ASIC die temperature into a differential voltage. A multiplexer selects the X, Y, Z or temperature channel in a predefined timing scheme and feeds the voltage into an amplifier which does a coarse correction for gain deviations. The ADC following that stage converts the analog differential signal into the digital domain. The API software running on the host processor reads out the X, Y, Z and temperature information and further applies gain, offset and temperature effect corrections. The API finally provides corrected X, Y, Z and temperature information to the host as floating point values.

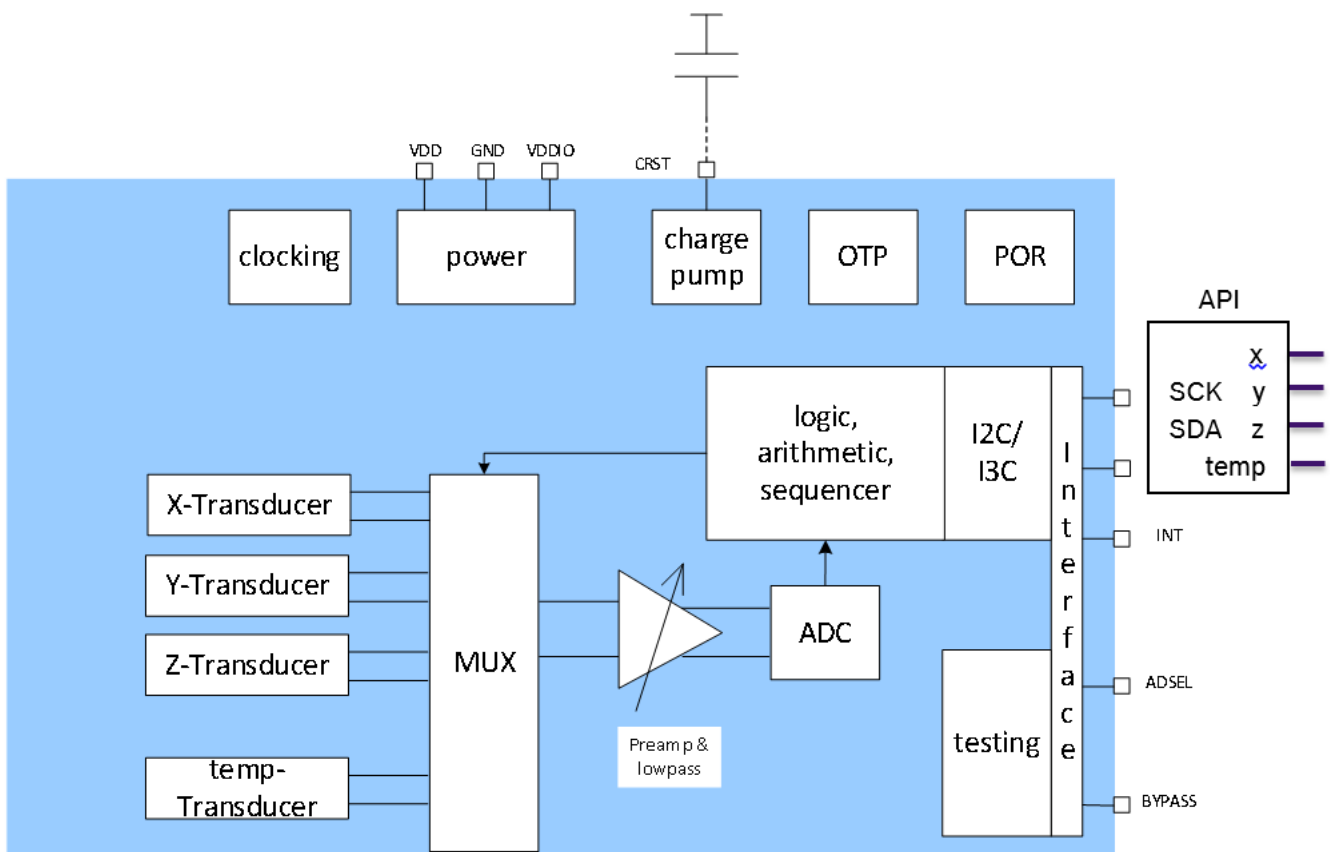


Figure 1: Block diagram

The data are made available via I3C or I2C interface. A data ready interrupt signals when new data are available, either via INT pin or via in-band-interrupt in I3C mode. If status polling is preferred a “data ready” status register can be read out on demand.

## 5. Functional description

BMM350 is a triaxial standalone geomagnetic sensor (sensing element and ASIC) in wafer-level-chip-scale package and can be operated via I2C or I3C as a slave device.

The BMM350 has two distinct power supply pins:

- ▶ VDD is the main power supply for all internal analog and digital functional blocks;
- ▶ VDDIO is a separate power supply pin, used for the supply of the digital interface.

The device can be completely switched off ( $V_{DD} = 0V$ ) while keeping the  $V_{DDIO}$  supply on ( $V_{DDIO} > 0V$ ) or vice versa. When  $V_{DDIO} = 0$  the voltages at IO pins must not exceed the specified limit of 300mV.

The BMM350 is highly configurable

### The major IC control settings are

- Power mode (normal, forced, suspend) see chapter 4.1
- Data rate ODR, see chapter 4.4
- Accuracy/noise, see chapter 4.4
- Interrupt settings, see chapter 4.5
- Pad drive strength, see chapter 4.6
- Enabled axes, see chapter 4.7
- I2C watchdog configuration, see chapter 9.1
- I3C in-band-interrupt (IBI) configuration, see chapter 4.5.5
- Sensortime always available, see chapter 4.8

Many registers provide status information:

- CHIP\_ID (product ID), see chapter 4.9.1
- Interrupt status, see chapters 4.5 and 4.10
- Error register, see chapter 4.9.3
- PMU (power management) status, effective ODR and accuracy settings, see chapter 4.9.4
- I3C errors, see chapter 4.9.5

Digital data registers provide **uncompensated** data about magnetic field strength, temperature and time:

Data registers for X,Y and Z magnetic channel and temperature channel provide data in 24bit registers in 21bit signed-integer format (21LSBs used)

Sensor time is provided in 24bit format, unsigned integer, with 40us resolution.

For proper temperature compensation is it mandatory to either use the provided API functions or at least look at the API code for own software developments.

To obtain compensated data use API functions, described in chapter 7.

## 5.1 Power modes, boot procedure

The BMM350 knows two major modes of operation, “normal mode” and “forced mode”. “suspend mode” brings the device into minimal power consumption, where settings are retained and communication is possible while data conversions are stopped.

After boot, the device will be in suspend mode, consuming the least current possible offered by this device.

Immediately after boot the API downloads the compensation coefficients from the BMM350 OTP (one-time programmable memory).

This is handled automatically in the API function `BMM350_init()`.

The boot phase must be terminated by writing `0x80` to `OTP_CMD_REG` (also done in `BMM350_init`).

From then on the OTP is inaccessible to the API unless a power reset or soft reset triggers another boot.

Power modes are selected using register `PMU_CMD.pmu_cmd`.

**An API function “BMM350\_set\_powermode” is provided which handles power mode settings.**

### 5.1.1 Power off

In Power off, VDD and/or VDDIO are unpowered/below brownout threshold and the device does not operate. When only one of VDD or VDDIO is supplied, the magnetic sensor will still be in Power off. Power on/boot is performed immediately after both VDD and VDDIO are raised above their reset release thresholds.

The sequence of raising VDD and VDDIO is left to the user, both voltages can be raised in any desired order.

If power is stable, the internal OTP becomes accessible (OTP access granted).

### 5.1.2 Suspend mode

Suspend mode is the default power mode of BMM350 after the chip is powered and has finished booting when using `BMM350_init` API function. This function reads out the OTP which stores compensation coefficients.

Suspend mode can be reached on demand via sending a suspend mode request to register `PMU_CMD.pmu_cmd`, or it is reached automatically after finishing a “forced mode” conversion.

Current consumption in suspend mode is minimal, so, this mode is useful for periods when data conversion is not needed. Read and write of all registers is possible.

### 5.1.3 Normal mode

Normal mode is activated via register `PMU_CMD.pmu_cmd`.

An API function `BMM350_set_powermode` is provided which helps doing that.

In normal mode the internal oscillator triggers conversions with the chosen ODR (output data rate) with the accuracy that can be expected from an RC oscillator.

The noise performance, tightly coupled with the current consumption, is controlled via the “averaging” setting, see register `PMU_CMD_AGGR_SET.avg`. As the name indicates 1, 2, 4 or 8 data samples are generated internally and averaged before the result is written into the data register. This improves the noise performance at the cost of current consumption, which is increasing proportionally with the averaging factor setting.

Not all combinations of output data rate and averaging are possible, see Table 5.

An API function `BMM350_set_performance` is provided which helps setting the averaging/noise performance. The function checks for the combination of ODR and averaging setting and chooses the highest possible averaging setting if the desired averaging setting is too high.

Table 5: Allowed combinations of output data rate ODR and noise configuration/averaging and related current consumption in  $\mu A^*$ .

Noise Mode, number of samples averaged	ODR 400Hz	ODR 200Hz	ODR 100Hz	ODR 50Hz	ODR 25Hz	ODR 12.5Hz	...	ODR 1.5625Hz
“Ultra Low Noise”, avg=3 8 samples	-	-	-	325	180	108		50
“Low Noise”, avg=2, 4 samples	-	-	335	175	96	57		25
“Regular Power”, avg=1, 2 samples	-	370	190	100	55	33		15
“Low Power”, avg=0, 1 sample	455	235	122	70	40	23		12

\* Approximate values in  $\mu A$ , measurements only for a couple of samples, deviation from specification table possible

### 5.1.4 Forced mode

By using forced mode, it is possible to trigger new measurements at any rate as long it is below 200Hz. In forced mode, which is available only when the device resides in suspend mode, a conversion of all selected channels ( register `PMU_CMD_AXIS_EN.en_x/y/z`) can be triggered via “forced mode request”, writing `0x03` or `0x04` ( FM or FM\_FAST) into register “`PMU_CMD`”. When finishing the forced mode data conversion the device returns into suspend mode.

The two flavors of forced mode, “`FORCED_MODE (FM)`” and “`FORCED_MODE_FAST (FM_FAST)`” shall be used for small (FM) or high (FM\_FAST) data rates. For data rates of 25Hz or higher FM\_FAST (`0x04`) can be used. For lower ODRs FM (`0x03`) must be used.

To reach ODR = 200Hz is only possible by using FM\_FAST.

In case the time between `BMM350_INIT` and the first FM trigger exceeds 0.1s, FM (`PMU_CMD = 0x03`) has to be used. All further FM calls can then use the FM or FM\_FAST depending on the ODR as described above.

A conversion (except the first sample after suspend mode) will be carried out within  $<5e-3$  seconds ( avg=1sample, FM\_FAST, up to 30ms for avg=8samples, FM) and data can be read via data register.

Conversion time will depend on the averaging setting and the choice of fast/normal forced mode; a high averaging value reduces the noise but increases the current consumption as well as the conversion time.

The conversion is guaranteed to finish within the relative timing precision/drift that can be expected from the internal RC oscillator, up to  $\pm 2\%$ .

The “data ready” interrupt or data ready status registers can be used to read data with minimum delay.

A programming example is provided which shows how to poll the status register to achieve high data rates and low latencies.

A temperature conversion will be done for every forced mode trigger.

When a forced mode request is issued before an ongoing forced mode conversion is finished the request will be ignored. Such a request will also be ignored when the device is in normal mode. The maximum forced mode trigger frequency (FMTf) permitted is 200Hz when selecting a noise performance/averaging setting of 1. See also Table 4 for other averaging settings.

Table 6: Highest forced mode trigger rate versus noise performance/ averaging setting

Noise performance/Averaging settings avg	Highest forced mode trigger rate [Hz] , forced mode fast
1	200
2	150
4	80
8	40

An API function is provided to request the BMM350 forced mode, `BMM350_set_powermode`.

The reading of results is handled via the API as well. Power mode transitions are depicted in Figure 2.

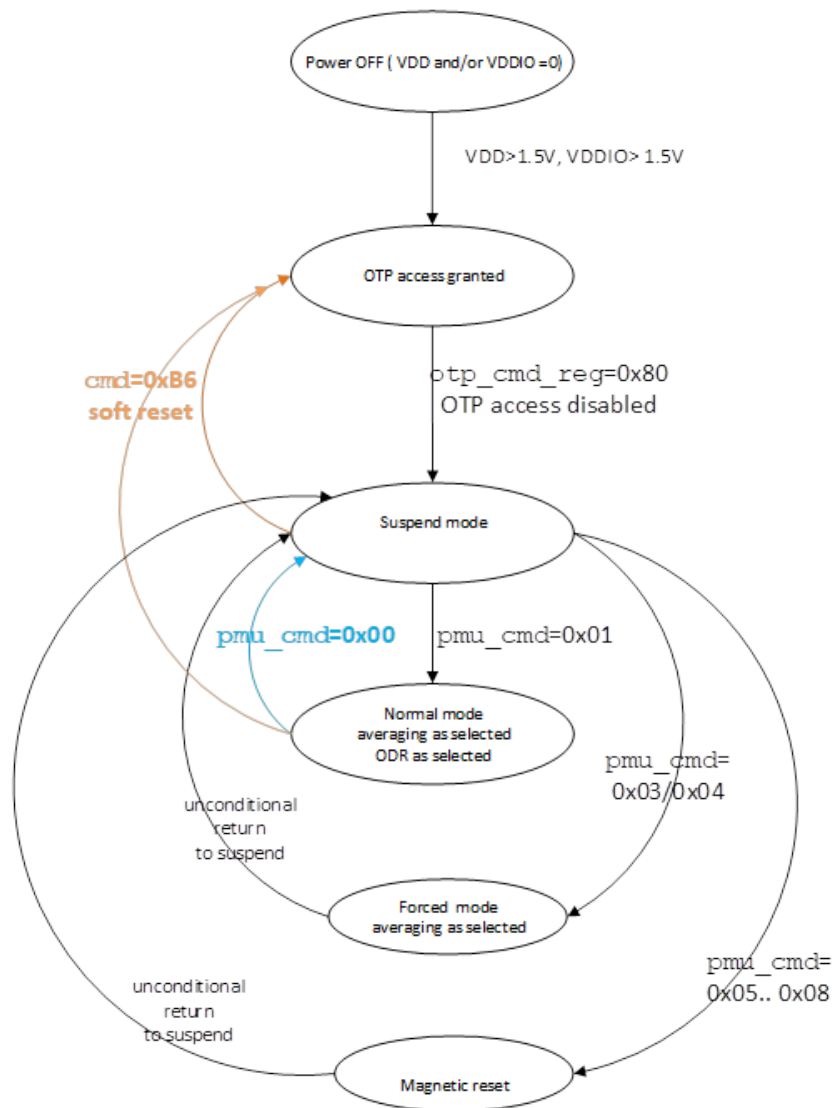


Figure 2: Power mode transitions

Only when being in suspend mode it is possible to switch to forced mode. A direct transition request from normal mode to forced mode will be ignored by the device.



The API function `BMM350_set_powermode()` allows to switch from normal to forced mode by transitioning via the “suspend mode” state.

### 5.1.5 Magnetic reset in normal and forced mode/excessive field strengths handling

The BMM350 has measures to recover from excessively strong magnetic fields which would lead to a performance deterioration in competitor products. These measures are called “magnetic reset”.

They can be triggered programmatically via the API function “`BMM350_check_for_highfield_and_reset`” which checks for excessive fields during operation and then triggers the reset. As said this works only during operation of the sensor, both in normal and forced mode.

If excessive field exposure has occurred out-of-operation, in a power-off state, users don't need to care: the magnetic reset is triggered after each boot by the device itself.

If the excessive field exposure has occurred in suspend mode, the sensor will not be able to detect that event itself. If large offset or sensitivity drift indicates that such an event has occurred user can call the API function “`BMM350_magnetic_reset_and_wait`”.

If the system resides in suspend over longer times (> 2s), the user may want to implement a preventive reset policy by issuing the magnetic reset before changing power mode to “forced” or “normal”. To this purpose the user can call the API function “`BMM350_magnetic_reset_and_wait`”.

This function will perform a full magnetic reset.

A programming example is provided with the API.

Hint: In normal and in forced mode the sensor automatically performs a magnetic reset for every ODR tick.

The magnetic reset is triggered using register `PMU_CMD.pmu_cmd`.

The magnetic reset generates magnetic fields inside the sensor which “reset” the magnetic orientation of the magnetic transducer.

Doing so requires the external 2.2uF capacitor attached to pin “CRST” to be charged. Depending on temperature and selected data rate this charging can take more or less time (less than 8ms by design). After finishing the API function `BMM350_init`, the said capacitor is charged, and data conversion can start right away after finishing the `BMM350_INIT`. The capacitor discharge time constant is about 1 second, so, calling “`BMM350_magnetic_reset_and_wait`” after longer time of inactivity recharges the capacitor.

### 5.1.6 Self Test

The self-test feature allows to test the function of the X and Y channel.

When activating self-test an internal magnetic field in the order of 130μT is generated which adds on to the ambient magnetic field.

Self-test result interpretation thus requires data before activating the self-test and data during activation of the self-test:

$$\Delta x = \text{mag}_x(\text{during self-test}) - \text{mag}_x(\text{before self-test}) \geq 130\mu\text{T}$$

$$\Delta y = \text{mag}_y(\text{during self-test}) - \text{mag}_y(\text{before self-test}) \geq 130\mu\text{T}$$

Self-test can only be activated for 1 axis at a time. The device must be in suspend mode and forced mode must be used to capture data before and after activating self-test.

The API function `BMM350_perform_self_test` carries the procedure out automatically.

## 5.2 Compensated data access from magnetic and temperature registers

Sensortime, temperature data and magnetic data are made available via data registers located next to each other. This allows reading said data within a single burst read.

Actually, burst read is an absolute “must” for data readout since the read stops data of being updated during readout and thus become inconsistent. The BMM350 halts data register update during a burst read starting at a sensortime, magnetic data or temperature data address. Reading data registers using single reads would allow for data updates between these reads and is strictly discouraged for normal mode.

The host must call API functions to correct data for temperature effects: offset and sensitivity errors will be corrected there as well as temperature effects. Trim values for the correction of offset, gain and temperature effects etc are stored in the BMM350 OTP and will be loaded when the API-function `BMM350_init()` is called.

For proper operation and best performance it is thus mandatory either to use the API functions or at least use API functions as template for own code, see chapter 7.

An API function “`BMM350_read_mag_data_and_compensate`” is provided which fetches data and calls all necessary compensation routines.

Magnetic data registers, temperature data registers and sensortime registers are located at addresses 0x31 to 0x3F.

### 5.2.1 Accessing compensation data

Compensation data for offset, gain and temperature effects are stored in the OTP.

The API function “`BMM350_get_compensation_data`” reads the data and stores them in the API memory space during the call of API function “`BMM350_init`”. So no explicit user call is needed. The data is used by API function “`BMM350_read_mag_data_and_compensate`”.

Compensation data is stored in the “`dev`” C structure, which allows to handle different sensors separately.

### 5.2.2 Sensortime

The API function “`BMM350_read_mag_data_and_compensate`” optionally reads the sensortime information. The 3 sensortime registers at address 0x3D-0x3F form a 24bit register when (and only when!) new data is generated.

If only sensortime is needed by the user the function “`bmm350_read_sensortime`” is provided.

If no new data is generated, either in normal or in forced mode, the sensortime data register is not updated. A sensortime read will provide the time when the last data has been generated.

Due to the synchronicity of ODR and sensortime, the XLSB value in normal mode is always the same when choosing ODR=100Hz or smaller. An ODR=200Hz leads to 2 different XLSB values, and ODR=400Hz to 4 different XLSB values.

The internal sensortime counter is only incremented in normal mode, not in suspend mode or forced mode unless setting register `CTRL_USER.cfg_sens_tim_aon` to 0x1.

After boot sensortime is 0.

**Sensortime information is ALWAYS related to new data generation.** When setting `CTRL_USER.cfg_sens_tim_aon` to 0x1 in suspend mode the internal sensortime is advanced, but reading sensortime always leads to the same result unless the device is e.g. sent into forced mode and new data is produced.

When reading sensortime after forced mode is finished one can observe that sensortime is now “pointing” to the timepoint where new data has been calculated.

Sensortime wraps around when the 24bit counter wraps around (no saturation).

### 5.3 Excessive field strengths handling

The BMM350 can handle magnetic field strengths up to  $\pm 2000\mu\text{T}$  in every X/Y/Z direction.

BMM350 has the limitation for measurement range that the sum of absolute values of each axis should be smaller.

$$\text{SQRT}(H_x^2 + H_y^2 + H_z^2) = 2000\mu\text{T}$$

An API function “`BMM350_check_for_highfield_and_reset`” is provided which checks whether the field range has been exceeded, and then automatically magnetically resets the sensor. See chapter 4.1.5 for details.

This API function must be called explicitly, it is not called automatically. It should be called directly after “`BMM350_read_mag_data_and_compensate`”.

A programming example is provided with the API.

### 5.4 Data rate and noise performance setting

Output data rate (ODR) is set using register `PMU_CMD_AGGR_SET.odr`.

Noise performance is set in register `PMU_CMD_AGGR_SET.avg`. In the following text this parameter is also called “averaging” ( $\text{avg} = 0/1/2/3 \rightarrow \text{averaging} = 1/2/4/8$ ).

An API function “`BMM350_set_odr_performance`” is provided which both handles data rate and noise performance setting.

A higher noise performance trades linearly against power consumption: to achieve a better noise performance several data samples are calculated internally and averaged. This is why current consumption is proportional to the averaging setting/noise performance setting, at least at higher ODR rates.

Table 7: Power consumption versus data rate ODR and noise performance setting, allowed combinations of settings. n/a: not allowed.

Averaging value/ noise performance setting <code>PMU_CMD_AGGR_SET.avg</code>	Noise Mode
8/3	Ultra Low Noise
4/2	Low Noise
2/1	Regular Power
1/0	Low Power

Power mode after boot is “suspend mode” when using API function `BMM350_init`.

When changing `PMU_CMD_AGGR_SET.odr` or `PMU_CMD_AGGR_SET.avg` it is necessary to send an “update” command via `PMU_CMD.pmu_cmd`

The API function “`BMM350_set_odr_performance`” does that automatically.

## 5.5 Interrupt settings and interrupt status

The BMM350 provides a “data ready” (abbreviated “drdy”) reset which is mapped to the interrupt status register and to the “INTerrupt” pin.

For I3C in-band interrupts see next chapter.

Two API functions are provided, “BMM350\_enable\_interrupt” and “BMM350\_configure\_interrupt” to configure and enable the “traditional” interrupt.

The API function “BMM350\_get\_interrupt\_status” can be used to obtain the interrupt status from register `INT_STATUS`: new data ready or not ready.

Reading `INT_STATUS.drdy_data_reg` retrieves the interrupt status, or in other words: whether new data is available since the last data read.

Reading this field clears it in latched mode.

In non-latched mode the bit field clears itself after 1.25ms.

### 5.5.1 Mapping Interrupt to INT pin

When enabling the data ready interrupt it is automatically mapped to the interrupt pin INT for A-samples. No extra action must be taken if mapping is desired.

Using C samples, mapping can be enabled or disabled using register `INT_CTRL.int_output_en`.

The setting can be changed using the API function “BMM350\_configure\_interrupt”.

### 5.5.2 Interrupt polarity at INT pin

The output at the interrupt pin “INT” can be configured to “active HIGH” or “active LOW” using register `INT_CTRL.int_pol`. Setting 0x0 is active LOW, 0x1 is active HIGH.

The setting can be changed using the API function “BMM350\_configure\_interrupt”.

### 5.5.3 Interrupt drive type: CMOS or open drain

The driver type at the interrupt pin “INT” can be configured to “push-pull” or “open drain” using register `INT_CTRL.int_od`. Setting 0x0 is “open drain”, 0x1 is push-pull.

The setting can be changed using the API function “BMM350\_configure\_interrupt”.

### 5.5.4 Interrupt latching type or non-latching

The interrupt can be configured to “latching” or “non-latching” using register

`INT_CTRL.int_mode`. Setting 0x0 is “pulsed”, 0x1 is “latched”.

The setting can be changed using the API function “BMM350\_configure\_interrupt”.

### 5.5.5 In-band interrupt

In I3C mode an in-band interrupt (“IBI”) can be sent to the host using the serial I3C interface. That IBI has to be enabled explicitly using register `INT_CTRL_IBI.drdy_int_map_to_ibi`.

Only enabling the IBI is not sufficient, the data ready interrupt itself has to be enabled using register `INT_CTRL.drdy_data_reg_en`.

The IBI uses the serial interface for the transmission for the interrupt information.

The time between issuing the interrupt in the sensor and the host receiving the interrupt is not deterministic since high I3C bus traffic can lead to a (repeated) non-acknowledgement of the sensor interrupt request.

`INT_CTRL_IBI.clear_drdy_int_status_upon_ibi` allows to clear the interrupt status register automatically upon sending an IBI.

An API function “`BMM350_set_int_ctrl_ibi`” is provided to configure and enable/disable the IBI.

## 5.6 Pad drive settings

To avoid problems with signal over- or undershoot with long wires the user can configure the relative pad driver strength via register `PAD_CTRL.drvc<2:0>`. A setting of 0 selects the weakest drive strength, a setting of 7 selects the strongest drive strength.

In I3C mode the user does not have control over the value of the pull-up resistor value, so care must be taken to not select a too weak driver strength.

The default relative drive strength is 7 after boot.

An API function “`BMM350_set_pad_drive`” is provided to change the drive strength.

## 5.7 Enable/disable axis

The user can choose between converting all axes or only a selected set of axes.

Configuration is handled via register `PMU_CMD.en_x/y/z`.

An API function “`BMM350_enable_axes`” is provided to handle that configuration.

The function will ignore requests to disable all axes.

Enabling or disabling an axis is only working in suspend mode, a change on-the-fly in normal mode is ignored by the sensor hardware.

The API function “`BMM350_enable_axes`” will turn the sensor into suspend mode, change the axes enable bits and re-enable normal mode when the sensor is in normal mode at the moment the said API function is called.

## 5.8 Sensortime always available

For best power saving the internal clocks are only turned on when needed.

In suspend mode, all fast internal clocks are turned off, meaning that the sensortime counter is stopped.

User can decide to keep the main internal oscillator running, which makes sensortime information available all time.

An API function “`BMM350_set_ctrl_user`” is provided which makes sensortime all time available or not. The sensortime reading is updated only when new data are available after data conversion.

## 5.9 I2C watchdog setting

When I2C communication stalls the bmm350 is able to reset the communication interface using a watchdog. It is possible to choose between 2 timeout intervals, see register description `I2C_WDT_SET`.

An API function “`BMM350_set_i2c_wdt`” is provided to enable or disable the watchdog and choose the interval length.

## 5.10 Status information

A few registers provide status or error information.

### 5.10.1 CHIP\_ID

`CHIP_ID` is located at address 0.

Its content is 0x33, read-only. This content is BST product-specific and helps e.g. to check initially whether the BMM350 is properly communicating with the host.

### 5.10.2 ERR\_REG: PMU and boot error

Register *ERR\_REG* indicates PMU- and boot-errors.

*ERR\_REG.pmu\_err* tells if PMU commands have been issued out-of-time (too early) or in the wrong context.

Error conditions are:

- ▶ Forced mode is requested in normal mode (sensor must be in suspend mode if doing that)

To prevent the occurrence of such errors it is recommended to use the API function `BMM350_set_powermode`.

### 5.10.3 PMU\_CMD\_STATUS: PMU (command) status

▶ *PMU\_CMD\_STATUS\_0.pmu\_cmd\_busy* signals that the PMU is not ready to take new commands. Such a busy condition can be encountered after triggering a magnetic reset, which can take up to 8ms. Another busy condition can be met after issuing a second forced mode request immediately after issuing a first forced mode request. The first forced mode request can take longer as expected, depending on the charging condition of the external 2.2uF capacitor.

▶ *PMU\_CMD\_STATUS\_0.cmd\_is\_illegal* indicates illegal commands. Illegal commands are commands which do not appear in the register description of register *PMU\_CMD*.

▶ *PMU\_CMD\_STATUS\_0.pwr\_mode\_is\_normal* indicates whether the sensor is in normal (power) mode.

▶ *PMU\_CMD\_STATUS\_1.pmu\_odr\_s* and *PMU\_CMD\_STATUS\_1.pmu\_odr\_s* indicate the actual noise/performance setting and data rate setting.

To prevent the occurrence of errors it is recommended to use the API function `BMM350_set_powermode` and `BMM350_set_odr_perfromance`.

### 5.10.4 I3C\_ERR: I3C status/error reporting

I3C is a complex protocol. When bringing up a new system status information may be helpful.

*I3C\_ERR.i3c\_err\_0* indicates an SDR parity error. See MIPI spec **5.1.2.3.2 Ninth Bit of SDR Master Written Data as Parity**.

*I3C\_ERR.i3c\_err\_1* indicates an S0/S1 error, see MIPI spec **5.1.10.1.2 Error Type S0, 5.1.10.1.3 Error Type S1**.

### 5.10.5 Soft reset, reset detection.

A “soft reset” can be used to reset all sensor settings issued so far.

The soft reset is triggered by sending

`CMD.cmd = 0xB6` followed by `CMD.cmd = 0x0`

An API function `BMM350_soft_reset_and_wait()` is provided.

## 6. Pin and package description

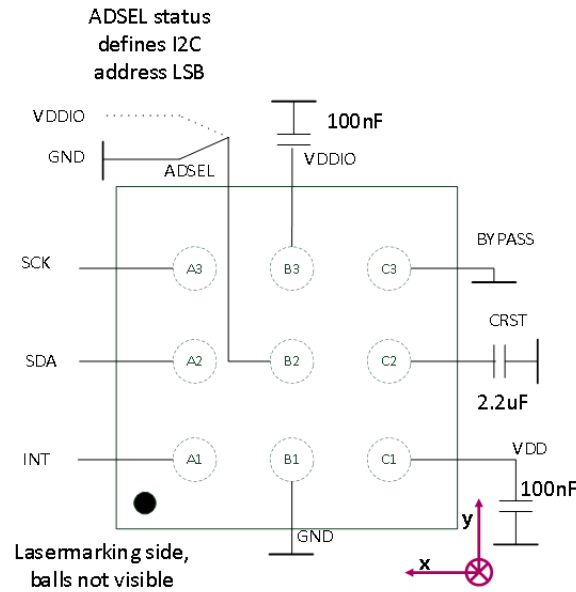
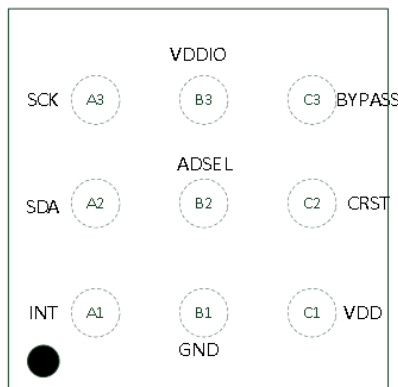


Figure 3: View from lasermarking side, balls invisible, required external components in operation

Pin name	Name	Function
C3	BYPASS	Connect to ground
C2	CRST	Connect to external 2.2uF capacitor, low inductance. e.g. see chapter 14
C1	VDD	Digital and analog supply, use decoupling capacitor
B3	VDDIO	IO supply, use decoupling capacitor
B2	ADSEL	The logic status of ADSEL defines the LSB to the I2C legacy address. Connect ADSEL to GND or VDDIO
B1	GND	Analog, digital and IO ground
A3	SCK	I2C / I3C clock
A2	SDA	I2C / I3C data
A1	INT	Interrupt output



Lasermarking side, balls not visible

Figure 4: Pin names only, laser marking side

### 5.1 Landing pattern recommendation

For the design of the landing pattern, the dimensions shown in the HSMI are recommended. It is important to note that areas marked in red are exposed PCB metal pads.

- ▶ In case of a solder mask defined (SMD) PCB process, the land dimensions should be defined by solder mask openings. The underlying metal pads are larger than these openings.
- ▶ In case of a non-solder mask defined (NSMD) PCB process, the land dimensions should be defined in the metal layer. The mask openings are larger than these metal pads.

### 5.2 Connection diagrams

100nF capacitors should be used for VDD and VDDIO decoupling.  
 CRST requires a 2.2uF capacitor, recommended type: see chapter 13.

### 5.3 Recommendations for PCB design

- ▶ S IC footprint: IC size 1.28 x 1.28mm, reserve space for 1.3 x 1.3 mm<sup>2</sup> IC size
- ▶ ball pitch is 400um (horizontal & vertical)
- ▶ the following peak current can flow:
  - ▶ 400mA current (very short time < 1us) between CRST capacitor (2.2uF) and CRST pin
  - ▶ 400mA current (very short time < 1us) between CRST capacitor (2.2uF) and GND pin
  - ▶ 20mA max current (short time, < 10ms) in VDD pin, 750uA maximum average current
- ▶ external required components:
  - ▶ 100nF capacitor decoupling VDD to GND
  - ▶ 100nF capacitor decoupling VDDIO to GND
  - ▶ 2.2uF capacitor to CRST, 200mA peak current, need low resistance, low inductance type and non-magnetic properties. Recommended type: see chapter 14.
- ▶ Connect pin “ADSEL” either to GND or VDDIO to change I2C legacy address.

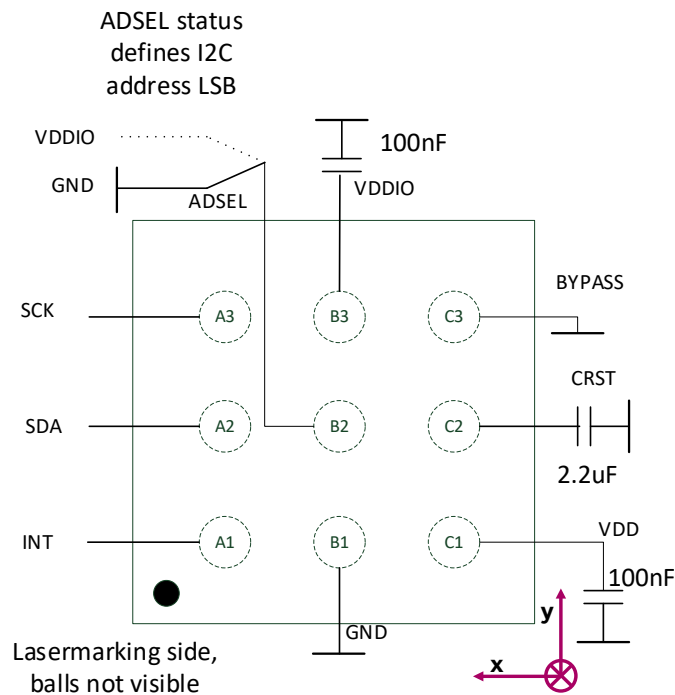


Figure 5: I2C connection diagram



## 7. API functions

API functions are documented in

<https://github.com/BoschSensortec/BMM350-Sensor-API/blob/master/README.md>

## 8. Register Map

Table 8 : 8-bit portrait register map

			read only	write only	reserved					
read/write										
Register Address	Register Name	Default Value	7	6	5	4	3	2	1	0
0x7E	<a href="#">CMD</a>	0x00	cmd							
0x7D	-	-	reserved							
...	-	-	reserved							
0x62	-	-	reserved							
0x61	<a href="#">CTRL_USER</a>	0x00	reserved							
0x60	<a href="#">TMR_SELFTES_T_USER</a>	0x00	reserved			ist_en_y	ist_en_x	st_p	st_n	st_igen_en
0x5F	-	-	reserved							
...	-	-	reserved							
0x56	-	-	reserved							
0x55	<a href="#">OTP_STATUS_REG</a>	0x10	error			cur_page_addr			otp_cmd_done	
0x54	-	-	reserved							
0x53	<a href="#">OTP_DATA_LSB_REG</a>	0x00	otp_mem_data_lsb							
0x52	<a href="#">OTP_DATA_MSB_REG</a>	0x00	otp_mem_data_msb							
0x51	-	-	reserved							
0x50	<a href="#">OTP_CMD_REG</a>	0x00	otp_cmd			word_addr				
0x4F	-	-	reserved							
...	-	-	reserved							
0x40	-	-	reserved							
0x3F	<a href="#">SENSE_RUNTIME_MSB</a>	0x7F	data_st_23_16							
0x3E	<a href="#">SENSE_RUNTIME_LSB</a>	0x7F	data_st_15_8							
0x3D	<a href="#">SENSE_RUNTIME_XLSB</a>	0x7F	data_st_7_0							

0x3C	<a href="#">TEMP_MSB</a>	0x7F	data_t_23_16						
0x3B	<a href="#">TEMP_LSB</a>	0x7F	data_t_15_8						
0x3A	<a href="#">TEMP_XLSB</a>	0x7F	data_t_7_0						
0x39	<a href="#">MAG_Z_MSB</a>	0x7F	data_z_23_16						
0x38	<a href="#">MAG_Z_LSB</a>	0x7F	data_z_15_8						
0x37	<a href="#">MAG_Z_XLSB</a>	0x7F	data_z_7_0						
0x36	<a href="#">MAG_Y_MSB</a>	0x7F	data_y_23_16						
0x35	<a href="#">MAG_Y_LSB</a>	0x7F	data_y_15_8						
0x34	<a href="#">MAG_Y_XLSB</a>	0x7F	data_y_7_0						
0x33	<a href="#">MAG_X_MSB</a>	0x7F	data_x_23_16						
0x32	<a href="#">MAG_X_LSB</a>	0x7F	data_x_15_8						
0x31	<a href="#">MAG_X_XLSB</a>	0x7F	data_x_7_0						
0x30	<a href="#">INT_STATUS</a>	0x00	reserved			drdy_data_reg	reserved	reserved	
0x2F	<a href="#">INT_CTL_IBI</a>	0x00	reserved		clear_drdy_int_status_upon_ibi	reserved		drdy_int_map_to_ibi	
0x2E	<a href="#">INT_CTL_RL</a>	0x00	drdy_data_reg_en	reserved		int_output_en	int_od	int_pol	int_mode
0x2D	-	-	reserved						
...	-	-	reserved						
0x0E	-	-	reserved						
0x0D	<a href="#">TRANS_DUCER_REV_ID</a>	0x33	trsdcr_rev_id_fixed				trsdcr_rev_id_otp		
0x0C	-	-	reserved						
...	-	-	reserved						
0x0B	-	-	reserved						
0x0A	<a href="#">I2C_WDT_SET</a>	0x00	reserved				i2c_wdt_sel	i2c_wdt_en	
0x09	<a href="#">I3C_ERR</a>	0x00	reserved			i3c_error_3	reserved		i3c_error_0
0x08	<a href="#">PMU_CMD_STATUS_1</a>	0x00	reserved	pmu_avg_s		pmu_odr_s			

0x07	<a href="#">PMU_C MD_ST ATUS_0</a>	0x00	pmu_cmd_val	ODR_ov wr	pmu_cm d_busy	AVG_ov wr	ODR_ov wr	pmu_cm d_busy
0x06	<a href="#">PMU_C MD</a>	0x00	reserved		pmu_cmd			
0x05	<a href="#">PMU_C MD_AXI S_EN</a>	0x70	reserved		En_z	En_y	En_x	
0x04	<a href="#">PMU_C MD_AG GR_SE I</a>	0x14	reserved	avg	odr			
0x03	<a href="#">PAD_CT RL</a>	0x07	reserved			drv		
0x02	<a href="#">ERR_R EG</a>	0x00	reserved					pmu_cm d_error
0x00	<a href="#">CHIP_ID</a>	0x33	chip_id_fixed		chip_id_otp			

### 8.1 Register (0x00) CHIP\_ID

DESCRIPTION: Chip identification code

RESET: 0x33

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4
Read/Write	R	R	R	R
Reset Value	0	0	1	0
Content	chip_id_fixed			

Bit	3	2	1	0
Read/Write	R	R	R	R
Reset Value	0	0	1	1
Content	chip_id_otp			

chip\_id\_otp: Programmable (NVM) part of chip id.

chip\_id\_fixed: Fixed part of chip id.

### 8.2 Register (0x02) ERR\_REG

DESCRIPTION: Reports Sensor Error Flag. Will be cleared on read. If the user writes a 1 into any status bit, this will also clear that bit.

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0	
Read/Write	n/a	n/a	n/a	n/a	n/a	n/a	R/W	R/W	
Reset Value	0	0	0	0	0	0	0	0	
Content	reserved							pmu_cmd_error	

► pmu\_cmd\_error: a new PMU\_CMD is issued when previous command has not been finished. The new PMU\_CMD will be ignored.

0x0: pmu\_ok

0x1: pmu\_error

### 8.3 Register (0x03) PAD\_CTRL

DESCRIPTION: Configure pad behavior

RESET: 0x07

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	n/a	n/a	n/a	n/a	n/a	R/W	R/W	R/W
Reset Value	0	0	0	0	0	1	1	1
Content	reserved					drv		

► drv: Set the pad drive capability

0x0: drv\_weakest, 0x7: drv\_strongest , 0xn:drv\_n\_7, n=1..6

### 8.4 Register (0x04) PMU\_CMD\_AGGR\_SET

DESCRIPTION: configuration of the ODR and AVG

RESET: 0x14

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	1	0	1	0	0
Content	reserved		avg		odr			

odr: Output Data Rate.

odr		
0x02	odr_400hz	400 Hz odr
0x03	odr_200hz	200 Hz odr
0x04	odr_100hz	100 Hz odr
0x05	odr_50hz	50 Hz odr
0x06	odr_25hz	25 Hz odr
0x07	odr_12_5hz	12.5 Hz odr
0x08	odr_6_25hz	6.25 Hz odr
0x09	odr_3_125hz	3.125 Hz odr
0x0a	odr_1_5625hz	1.5625 Hz odr

avg: Measurements Averages

avg		
0x00	no_avg	no average
0x01	avg_2	average between 2 samples
0x02	avg_4	average between 4 samples
0x03	avg_8	average between 8 samples

reserved: reserved.

## 8.5 Register (0x05) PMU\_CMD\_AXIS\_EN

DESCRIPTION: axis configuration

RESET: 0x07

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	n/a	R/W	R/W	R/W	W	W	W	W
Reset Value	0	0	0	0	0	1	1	1
Content	reserved					en_z	en_y	en_x

en\_x: Enable Axes X

en_x		
0x00	disable	Channel Disabled
0x01	enable	Channel Enabled

en\_y: Enable Axes Y

en_y		
0x00	disable	Channel Disabled
0x01	enable	Channel Enabled

en\_z: Enables Axes Z

en_z		
0x00	disable	Channel Disabled
0x01	enable	Channel Enabled

### 8.6 Register (0x06) PMU\_CMD

DESCRIPTION: PMU cmd configuration

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	n/a	RW	RW	RW	W	W	W	W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved				pmu_cmd			

pmu\_cmd: command for PMU mode switch

pmu_cmd		
0x00	SUS	go to SUSPEND mode
0x01	NM	go to NORMAL mode. Note that when being in normal mode, it is possible to require only the SUS and UPD_OAE commands. In case a different command is required, a protection mechanism prevents from propagating that command inside the device and the illegal_command flag is set. Even in this case, before sending another command after the illegal one, the busy bit must be low
0x02	UPD_OAE	update odr and avg parameter: after requiring one odr/avg update, it is mandatory to wait for the data ready interrupt after the busy flag was cleared before requiring the next odr/avg update (this means that the first data set has been generated with the new odr/avg). Note that looking only at the busy flag is not correct because the busy flag can be deasserted too early (A-Si implementation limitation)
0x03	FM	go to FORCED mode with full CRST recharge
0x04	FM_FAST	go to FORCED mode with fast CRST recharge
0x05	FGR	do flux-guide reset with full CRST recharge
0x06	FGR_FAST	do flux-guide reset with fast CRST recharge
0x07	BR	do bit reset with full CRST recharge
0x08	BR_FAST	do bit reset with fast CRST recharge



### 8.7 Register (0x07) PMU\_CMD\_STATUS\_0

DESCRIPTION: Sensor Status Flag

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	n/a	n/a	n/a	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved			cmd_is_ill egal	pwr_mod e_is_nor mal	AVG_ovw r	ODR_ovw r	pmu_cmd _busy

- ▶ pmu\_cmd\_busy: The previous PMU CMD is still in processing  
0x0: pmu\_ok 0x1: pmu\_busy
- ▶ ODR\_ovwr: The previous PMU\_CMD\_AGGR\_SET.odr has been overwritten  
0x0: odr\_nochange 0x1: odr\_ovwr
- ▶ AVG\_ovwr: The previous PMU\_CMD\_AGGR\_SET.avg has been overwritten  
0x0: avg\_nochange 0x1: avg\_ovwr
- ▶ pwr\_mode\_is\_normal: The chip is in normal power mode  
0x0: not\_normal 0x1: normal\_mode
- ▶ cmd\_is\_illegal: cmd value is not allowed  
0x0: cmd\_ok 0x1: illegal

## 8.8 Register (0x08) PMU\_CMD\_STATUS\_1

DESCRIPTION: PMU Status Flag

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	n/a	n/a	R	R	R	R	R	R
Reset Value	0	0	0	0	0	0	0	0
Content	reserved		pmu_avg_s		pmu_odr_s			

► pmu\_odr\_s: The actual effective PMU\_CMD\_AGGR\_SET.odr value

► pmu\_avg\_s: The actual effective PMU\_CMD\_AGGR\_SET.avg value

## 8.9 Register (0x09) I3C\_ERR

DESCRIPTION: I3C Bus Error Statistics. Will be cleared on read. If the user writes a 1 into any status bit, this will also clear that bit

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	n/a	n/a	n/a	n/a	RW	n/a	n/a	RW
Reset Value	0	0	0	0	0	0	0	0
Content	reserved				i3c_error_ 3	Reserved		i3c_error_ 0

► i3c\_error\_0: SDR parity error occurred.

0x0: i3c0\_noerror 0x1: i3c0\_error

► i3c\_error\_3: S0/S1 error occurred. S0/S1 error will be cleared automatically after 60 us or if we see a HDR-exit pattern on the bus.

0x0: i3c3\_noerror 0x1: i3c3\_error

## 8.10 Register (0x0A) I2C\_WDT\_SET

DESCRIPTION: i2c watchdog configure registers

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	n/a	n/a	n/a	n/a	n/a	n/a	RW	RW
Reset Value	0	0	0	0	0	0	0	0
Content	reserved						i2c_wdt_s el	i2c_wdt_e n

i2c\_wdt\_en: i2c watch dog enable.

i2c_wdt_en		
0x00	disabled	Disable I2C watch dog
0x01	enabled	Enable I2C watch dog

i2c\_wdt\_sel: i2c watch dog time out period.

i2c_wdt_sel		
0x00	short	I2C watch dog time out after 1.28ms
0x01	long	I2C watch dog time out after 40.96ms

### 8.11 Register (0x2E) INT\_CTRL

H DESCRIPTION: Configuration of interrupts for INT\_STATUS register and INT pin

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R/W	n/a	n/a	n/a	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	drdy_data_reg_en	reserved			int_output_en	int_od	int_pol	int_mode

int\_mode: Interrupt Mode

► If set, output is in Latched Mode, else in Pulsed Mode

int_mode		
0x00	pulsed	null
0x01	latched	null

int\_pol: Interrupt Polarity

► If set, output is Active High, else Active Low

int_pol		
0x00	active_low	null
0x01	active_high	null

int\_od: Configure output: open-drain or push-pull

int_od		
0x00	open-drain	null
0x01	push-pull	null

int\_output\_en: Enable mapping of int on INT pin. IMPORTANT: this is not supported in A-Si: this bit is reserved in A-Si.

int_output_en		
0x00	off	Output disabled
0x01	on	Output enabled

drdy\_data\_reg\_en: Enable Mag Data Ready interrupt onto INT pin and INT\_STATUS

drdy_data_reg_en		
0x00	disabled	null
0x01	enabled	null

### 8.12 Register (0x2F) INT\_CTRL\_IBI

DESCRIPTION: Configuration of interrupts features related to IBI

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	n/a	n/a	n/a	R/W	n/a	n/a	n/a	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved			clear_drd y_int_stat us_upon_i bi	reserved			drdy_int_ map_to_i bi

drdy\_int\_map\_to\_ibi: map the drdy interrupt to I3C IBI

drdy_int_map_to_ibi		
0x00	disabled	null
0x01	enabled	null

clear\_drdy\_int\_status\_upon\_ibi: clear INT\_STATUS.drdy upon I3C IBI

clear_drdy_int_status_upon_ibi		
0x00	disabled	null
0x01	enabled	null

### 8.13 Register (0x30) INT\_STATUS

DESCRIPTION: Interrupt Status. Will be cleared on read. If the user writes a 1 into any status bit, this will also clear that bit.

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	n/a	n/a	n/a	n/a	n/a	R/W	n/a	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved					drdy_data_ reg	reserved	reserved

► drdy\_data\_reg: Magnetic data ready interrupt

0x0: no\_new data

0x1: new\_data

#### 8.14 Register (0x31) MAG\_X\_XLSB

DESCRIPTION: magnetometer X-axis extreme LSB byte

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_x_7_0							

#### 8.15 Register (0x32) MAG\_X\_LSB

DESCRIPTION: magnetometer X-axis LSB byte

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_x_15_8							

#### 8.16 Register (0x33) MAG\_X\_MSB

DESCRIPTION: magnetometer X-axis MSB byte

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_x_23_16							

### 8.17 Register (0x34) MAG\_Y\_XLSB

DESCRIPTION: magnetometer Y-axis extreme LSB byte

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_y_7_0							

Data retrieval: see 8.14

### 8.18 Register (0x35) MAG\_Y\_LSB

DESCRIPTION: magnetometer Y-axis LSB byte

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_y_15_8							

### 8.19 Register (0x36) MAG\_Y\_MSB

DESCRIPTION: magnetometer Y-axis MSB byte

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_y_23_16							

### 8.20 Register (0x37) MAG\_Z\_XLSB

DESCRIPTION: magnetometer Z-axis extreme LSB byte

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_z_7_0							

### 8.21 Register (0x38) MAG\_Z\_LSB

DESCRIPTION: magnetometer Z-axis LSB byte

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_z_15_8							

### 8.22 Register (0x39) MAG\_Z\_MSB

DESCRIPTION: magnetometer Z-axis MSB byte

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_z_23_16							

### 8.23 Register (0x3A) TEMP\_XLSB

DESCRIPTION: Temperature extreme LSB byte

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_t_7_0							

data\_t\_7\_0: Temperature

### 8.24 Register (0x3B) TEMP\_LSB

DESCRIPTION: Temperature LSB byte.

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_t_15_8							

data\_t\_15\_8: Temperature.

### 8.25 Register (0x3C) TEMP\_MSB

DESCRIPTION: Temperature MSB byte.

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_t_23_16							

data\_t\_23\_16: Temperature.

### 8.26 Register (0x3D) SENSORTIME\_XLSB

DESCRIPTION: Sensor timer extreme LSB byte

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_st_7_0							

data\_st\_7\_0: Sensor timer.

### 8.27 Register (0x3E) SENSORTIME\_LSB

DESCRIPTION: Sensor timer LSB byte.

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_st_15_8							

data\_st\_15\_8: Sensor timer.

### 8.28 Register(0x3F) SENSORTIME\_MSB

DESCRIPTION: Sensor timer MSB byte.

RESET: 0x7F

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	1	1	1	1	1	1	1
Content	data_st_23_16							

data\_st\_23\_16: Sensor timer.



### 8.29 Register (0x50) OTP\_CMD\_REG

DESCRIPTION: OTP command and word address register

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	otp_cmd				word_addr			

word\_addr: word address (DIR\_READ, DIR\_PRGM\_1B, DIR\_PRGM) or word start address (EXT\_READ, EXT\_PRGM)

otp\_cmd: otp commands:

- 001: DIR\_READ
- 010: DIR\_PRGM\_1B
- 011: DIR\_PRGM
- 100: PWR\_OFF\_OTP
- 101: EXT\_READ
- 111: EXT\_PRGM

### 8.30 Register (0x52) OTP\_DATA\_MSB\_REG

DESCRIPTION: OTP data: most significant byte register: it contains the data to write for a prog command or the read data after a read command has been executed

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	otp_mem_data_msb							

otp\_mem\_data\_msb: Most significant byte of 16-bit memory word

### 8.31 Register (0x53) OTP\_DATA\_LSB\_REG

DESCRIPTION: OTP data: least significant byte register: it contains the data to write for a prog command or the read data after a read command has been executed

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	otp_mem_data_lsb							

otp\_mem\_data\_lsb: Least significant byte of 16-bit memory word

### 8.32 Register (0x55) OTP\_STATUS\_REG

DESCRIPTION: OTP status register

RESET: 0x10

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	R	R	R	R	R	R	R	R
Reset Value	0	0	0	1	0	0	0	0
Content	error			cur_page_addr				otp_cmd_done

otp\_cmd\_done: Command done flag. When this field is = 1, then a new command can be accepted.

Otherwise, SW must wait before sending a new command because the previous one is still being executed

cur\_page\_addr: Currently selected page out of 8 selectable pages

error: Error register

000: NO\_ERROR

001: BOOT\_ERR

010: PAGE\_RD\_ERR

011: PAGE\_PRG\_ERR

100: SIGN\_ERR

101: INV\_CMD\_ERR

### 8.33 Register (0x60) TMR\_SELFTEST\_USER

DESCRIPTION: TMR user selftest reg. Please also consider the following IMPORTANT NOTES for the definition of ST algorithm in API.

1. enabling of internal self test current generator is a precondition for correct functionality of self test
2. the bits st\_n and st\_p must never be asserted simultaneously, not only because they are conflicting configurations, but because they would generate an internal short during self test execution, yielding wrong results and potentially damaging the circuits

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	n/a	n/a	n/a	R/W	R/W	R/W	R/W	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved			ist_en_y	ist_en_x	st_p	st_n	st_igen_en

st\_igen\_en: Drives the dc\_st\_igen\_en signal (when 1 it enables the selftest internal current gen)

st\_n: when at 1 configures execution of negative field self test (Drives the dc\_st\_n signal)

st\_p: when at 1 configures execution of positive field self test (Drives the dc\_st\_p signal)

ist\_en\_x: when at 1 activates internally generated self-test field on X axis,  
assuming all other preconditions and configurations are already set (Drives the dc\_ist\_en\_x signal)

ist\_en\_y: when at 1 activates internally generated self-test field on Y axis  
transducer, assuming all other preconditions and configurations are already set (Drives the dc\_ist\_en\_y signal)

### 8.34 Register (0x61) CTRL\_USER

DESCRIPTION: user settings register

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	n/a	n/a	n/a	n/a	n/a	n/a	n/a	R/W
Reset Value	0	0	0	0	0	0	0	0
Content	reserved							cfg_sens_ tim_aon

► **cfg\_sens\_tim\_aon**: It forces the sensor timer to be always running, even in SUSPEND MODE.

► This field can be written only in SUSPEND mode.

0x0: sensortime\_while\_normal

0x1: sensortime\_always

### 8.35 Register (0x7E) CMD

DESCRIPTION: Command Register

RESET: 0x00

DEFINITION (See Chapter 8 Register Map)

Bit	7	6	5	4	3	2	1	0
Read/Write	W	W	W	W	W	W	W	W
Reset Value	0	0	0	0	0	0	0	0
Content	cmd							

cmd: Available commands (Note: Register will always return 0x00 as read result):

cmd		
0x00	nop	Reserved. No command.
0xb6	softreset	Configuring this command will trigger a power on reset

## 9. Digital Interfaces

The sensor supports the I<sup>2</sup>C and I<sup>3</sup>C digital interfaces, where it acts as a “slave” for both protocols. The I<sup>2</sup>C interface supports the Standard, Fast and Fast+ modes. The I<sup>3</sup>C interface supports the single-data-rate SDR, high data rate HDR is not supported. However, the device can detect entering HDR mode and immediately stops listening to I<sup>3</sup>C data traffic until it detects the “leave HDR” I<sup>3</sup>C message.

The following transactions are supported:

- ▶ Single byte write
- ▶ Multiple byte write (using pairs of register addresses and register data)
- ▶ Single byte read
- ▶ Multiple byte read (using a single register address which is auto-incremented)

I<sup>3</sup>C additionally supports

- ▶ in-band-interrupt IBI
- ▶ timing control ASYNC

### 9.1 Interface selection

By default/after reset the I<sup>2</sup>C interface is active. After receiving the standardized I<sup>3</sup>C enable message (see I<sup>3</sup>C specification), the I<sup>3</sup>C interface is activated and remains active until a reset/power down event.

The correct address must be applied in both communication modes. The I<sup>2</sup>C (legacy) address is a combination of a fixed value, the value of the level of the pin “ADSEL” at boot time (after reset).

### 9.2 I<sup>2</sup>C interface

For detailed timings, please review the NXP original document UM10204.pdf (registration at <https://www.nxp.com> required).

All modes (standard, fast, fast+) are supported. As the device does not perform clock stretching, the SCL structure is a high-Z input without open drain capability.

The 5 MSB bits are fixed. The last bit is changeable by the pin setting of “ADSEL”. ADSEL will be evaluated at boot time, the I<sup>2</sup>C address is fixed then.

The ADSEL pin must not be left floating; if left floating, the I<sup>2</sup>C address will be undefined.

The 7-bit I<sup>2</sup>C device address is defined according to the following tables.

Table 9: I<sup>2</sup>C address definition

bit<6>	bit <5>	bit <4>	bit <3>	bit <2>	bit <1>	bit <0>
0	0	1	0	1	OTP backed, can be 0 or 1	0: ADSEL=LOW 1:ADSEL=HIGH

By default bit 1 of the I<sup>2</sup>C address is 0. Please contact sales if you wish to obtain samples with bit1 = 1.

Table 10: I2C address roll-out

I2C address	OTP content	ADSEL
0x14	0	LOW
0x15	0	HIGH
0x16	1 (please contact sales)	LOW
0x17	1 (please contact sales)	HIGH

The I2C interface provides a “watchdog” which monitors the I2C traffic. In case that no “STOP” condition is detected within a predefined time the I2C core is resetted.

*I2C\_WDT\_SET.i2c\_wdt\_en* enables and disables the watchdog.

In I3C mode the watchdog is not active.

Use API function `it_bmm350_set_i2c_watchdog` to configure the watchdog.

*I2C\_WDT\_SET.i2c\_wdt\_en* allows to switch between a short (setting=0x0) time out (after 1.28ms) or a long (setting=0x1) time out (after 40.96ms) configuration.

The I<sup>2</sup>C interface uses the following pins:

- ▶ SCK: serial clock (SCL)
- ▶ SDI: data (SDA)
- ▶ ADSEL: Slave address LSB (GND = ‘0’, V<sub>DDIO</sub> = ‘1’)

The following abbreviations will be used in the I<sup>2</sup>C protocol figures:

- ▶ S Start
- ▶ P Stop
- ▶ ACKS Acknowledge by slave
- ▶ ACKM Acknowledge by master
- ▶ NACKM Not acknowledge by master

### 9.2.1 I<sup>2</sup>C write

Writing is done by sending the slave address in write mode (RW = ‘0’).

Then the master sends pairs of register addresses and register data. The transaction is ended by a stop condition.

This is depicted in Figure 6.

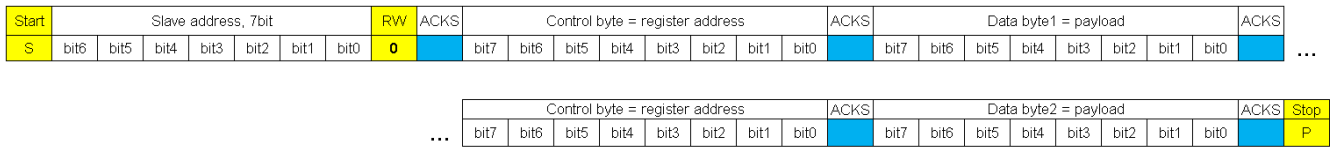


Figure 6: I2C multiple byte write (not auto-incremented)

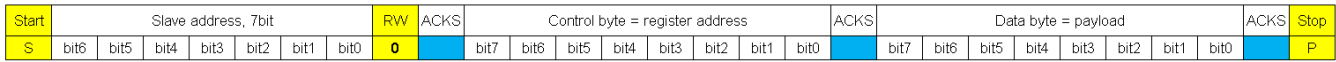


Figure 7: I2C single byte write

### 9.2.2 I2C read

To be able to read registers, first the register address must be sent in write mode. Then either a stop or a repeated start condition must be generated. After this the slave is addressed in read mode (RW = '1'), after which the slave sends out data from auto-incremented register addresses until a NOACKM and stop condition occurs. This is depicted in Figure 8, where register 0xF6 and 0xF7 are read.

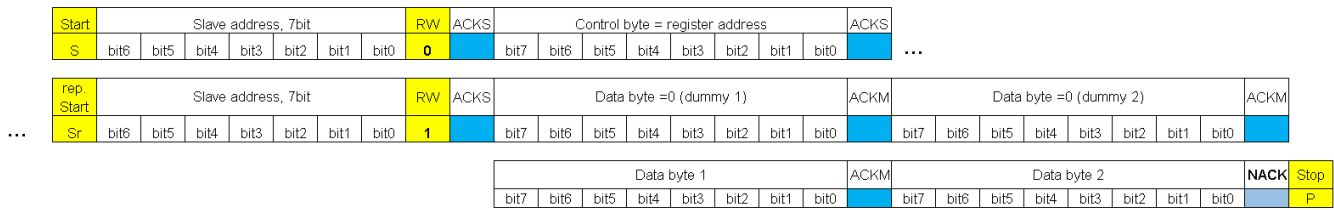


Figure 8: I2C multiple byte read, first 2 bytes transferred must be discarded, they are dummy bytes

### 9.2.3 Dummy bytes in I2C mode

It is important to note that two dummy bytes are sent when reading registers, which means that during an n byte read actually n+2 bytes have to be read, and the first 2 bytes be discarded.

### 9.3 I3C interface

The I3C interface is enabled using CCC commands as described in the I3C standard.

The I3C interface uses the following pins:

- ▶ SCK: serial clock
- ▶ SDA: serial data input/output, open drain or push-pull depending on communication state
- ▶ ADSEL: I2C legacy slave address LSB (GND = '0', VDDIO = '1')

A typical I3C data packet looks like this:

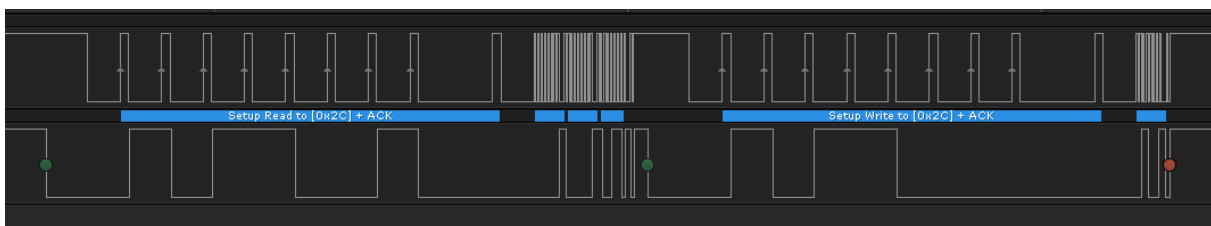


Figure 9: I3C data package

The open-drain address communication at the beginning is done with a low clock speed, while the push-pull data communication towards the end is done with 12.5MHz.

The possible benefit of the 12.5MHz clock is thus only achieved when sending large chunks of data at a time.

### 9.3.1 Supported CCC

Command	Command code	Default	Description
ENEC	0x00 / 0x80		Enable Target event driven interrupts (broadcast and direct)
DISEC	0x01 / 0x81		Disable Target event driven interrupts (broadcast and direct)
RSTDAA	0x06 / 0x86		Reset the assigned dynamic address (broadcast and direct)
ENTDAA	0x07		Enter the Dynamic Address Assignment procedure (broadcast)
SETAASA	0x29		Tell every target with a static address to use it as dynamic address (broadcast)
SETDASA	0x87		Assign dynamic address using static address (0x14 / 0x15 depending on ADSEL level)
SETNEWDA	0x88		Change dynamic address
RSTACT	0x2A / 0x9A		Configure Target Reset Action and query reset timing (broadcast and direct)
SETXTIME	0x28 / 0x98		Set exchange timing information (broadcast and direct)
GETXTIME	0x99		Get exchange timing information
GETPID	0x8D	0x07 0x70 0x10 0x33 0x00/0x10 0x00	Get target's Provisioned ID PID[1] = 0x00 / 0x10 depending on ADSEL level
GETBCR	0x8E	0x26	Get Bus Characteristic Register (BCR)
GETDCR	0x8F	0x43	Get Device Characteristics Register (DCR)
GETSTATUS	0x90		Get Operating Status
GETCAPS	0x95		Get Optional Capabilities

Table 11: MIPI I3C CCC supported commands

For all the unsupported set type CCCs we will still ACK to 0x7E and dynamic address (if it is a direct CCC), but the CCC will not be effective.

For all the unsupported get type CCCs we will still ACK to 0x7E and dynamic address (get type CCC can only be direct CCC), but will return all zeros.

### 9.3.2 Dummy bytes in I3C mode

It is important to note that two dummy bytes are sent when reading registers in I3C mode, which means that during an n byte read actually n+2 bytes have to be read, and the first 2 bytes be discarded.

### 9.3.3 I3C provisional ID

For dynamic address arbitration the product provides a “provisional ID” which is a 6-byte ID. (see MIPI chapter 5.1.4.1 “Device Requirements for Dynamic Address Assignment”)

byte6								byte5								byte4								byte3								byte2								byte1																																							
47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
MIPI member ID								IDT	Bosch Group ID								DEVICE ID								instance ID				reserved																																																		
0x03								0	0xb8								TBD								OTP/ADSEL				0x0																																																		

#### Bits [47:33]/ bytes 5-4: MIPI Member ID 15-bits:

The Bosch MIPI Member ID is 0x03b8.

#### Bit [32]/byte 4: Provisional ID Type selector 1-bit:

The Provisional ID type selector (IDT) is 0b0 (0= vendor fixed value for bits 31:0).

#### Bits [31:16] /bytes 3-2 : Part ID 16 bits:

The Part ID is divided into a Bosch Group ID (5 bit i.e. bits 31:27) and a Device ID (11 bit i.e. bits 26:16).

The Bosch Sensortec Group ID is 0b00010

The Device ID consists of an extended ID (3 bits i.e. bits 26:24), and the BMM350 CHIP\_ID, bits 23:16. The extended ID is mapped to 0b000.

#### Bits [15:12] Instance ID:

The LSB of the Instance ID (1-bit i.e. bit 12) is controlled by the level of the ADSEL pin. The other 3-bits of instance ID are mapped into OTP. The user register I3C\_trim\_0.i3c\_instance\_id<2:0> are used for this purpose. This register is OTP backed and can be modified when access to address page 1 is granted.

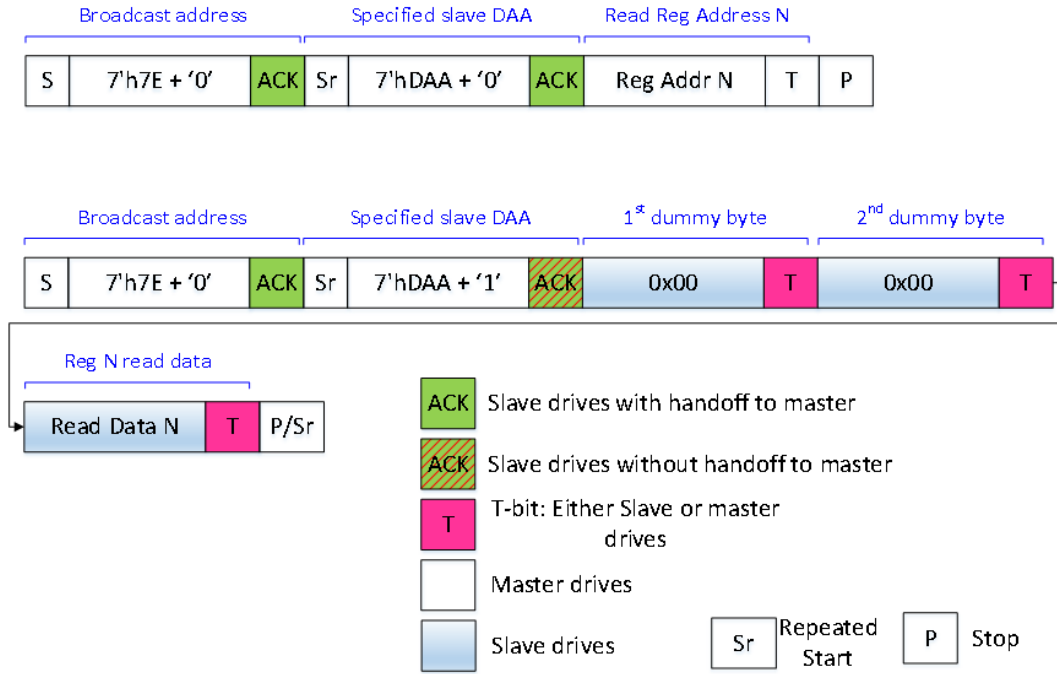
#### Bits [11:0] Reserved:

The reserved part is 0x00.

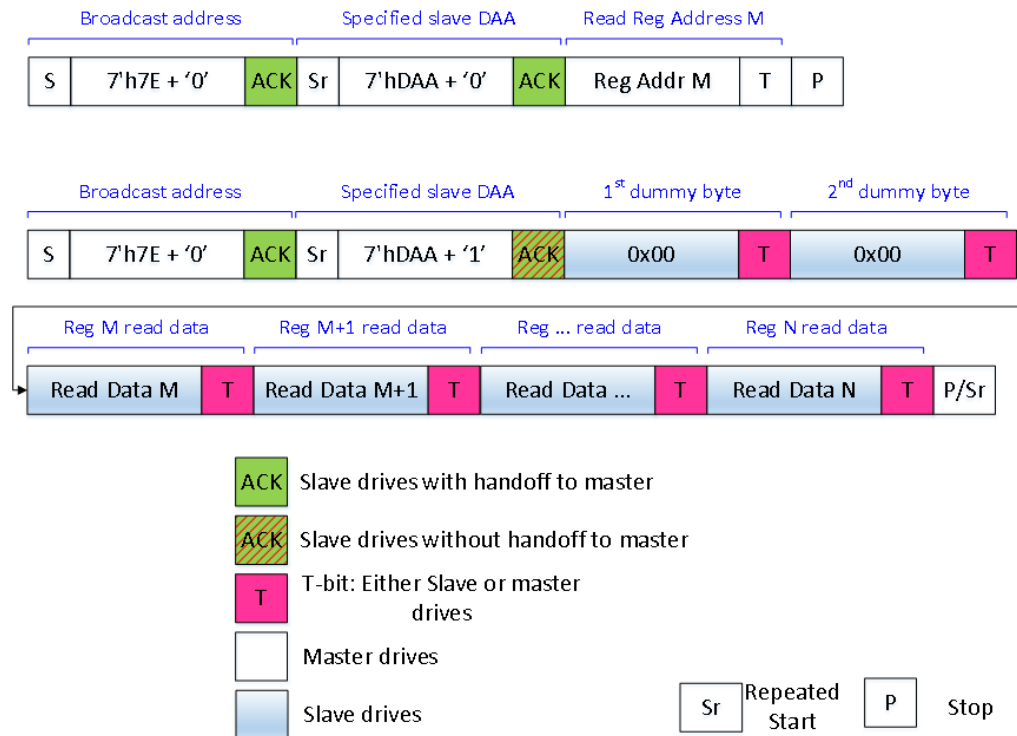


### 9.3.4 I3C read timing diagrams

#### 9.3.4.1 Single byte read

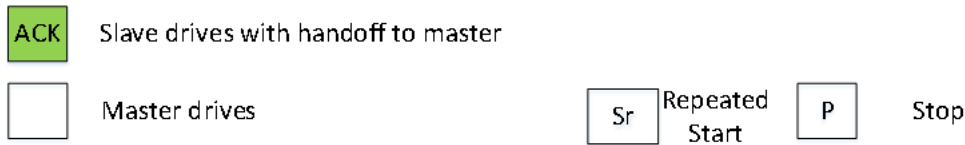
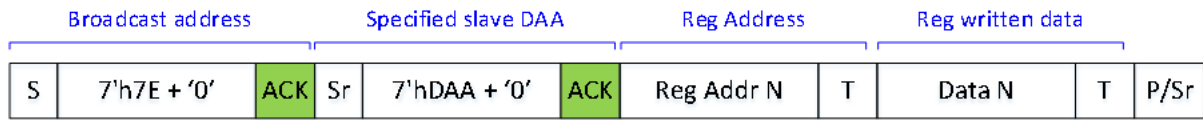


#### 9.3.4.2 Multiple byte read

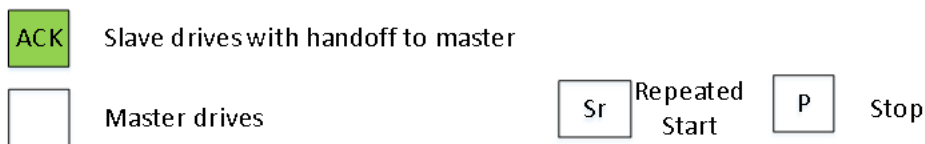
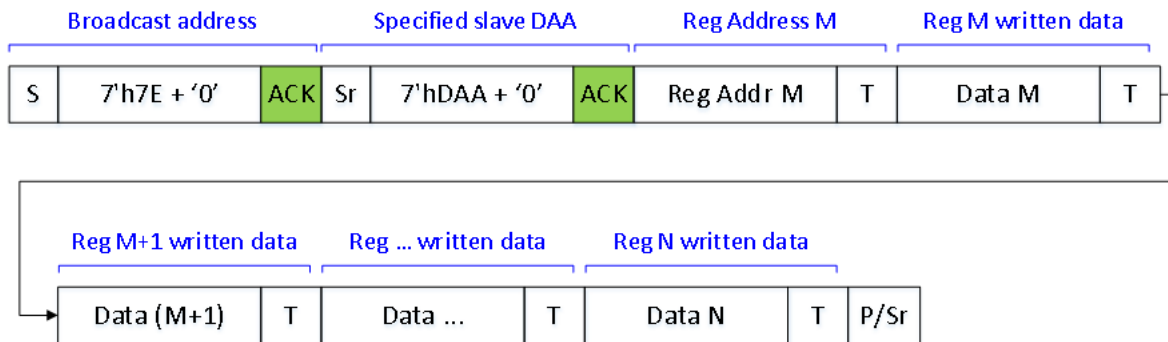


### 9.3.5 I3C write timing diagrams

#### 9.3.5.1 Single byte write



#### 9.3.5.2 Multiple byte write



## 9.4 Interface parameter specification

### 9.4.1 General interface parameters

Table 12: Interface parameters

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Input low level	$V_{il\_si}$	$V_{DDIO}=1.72\text{ V to }3.6\text{ V}$			20	% $V_{DDIO}$
Input high level	$V_{ih\_si}$	$V_{DDIO}=1.72\text{ V to }3.6\text{ V}$	80			% $V_{DDIO}$
Output low level I <sup>2</sup> C, I3C	$V_{ol\_SDI}$	$V_{DDIO}=1.72\text{ V}, I_{ol}=3\text{ mA}$			20	% $V_{DDIO}$
Output high level	$V_{oh}$	$V_{DDIO}=1.72\text{ V}, I_{oh}=1\text{ mA}$ (SDO, SDI)	80			% $V_{DDIO}$
Bus load capacitor	$C_b$	On SDA and SCK			20	pF

### 9.4.2 I<sup>2</sup>C timings

For I<sup>2</sup>C timings, the following abbreviations are used:

- ▶ “S&F mode” = standard and fast mode
  - ▶ “HS mode” = fast+ mode
  - ▶  $C_b$  = bus capacitance on SDA line
- All other naming refers to I<sup>2</sup>C specification 2.1 (January 2000).

The I<sup>2</sup>C timing diagram is in Figure 10. The corresponding values are given in Table 14.

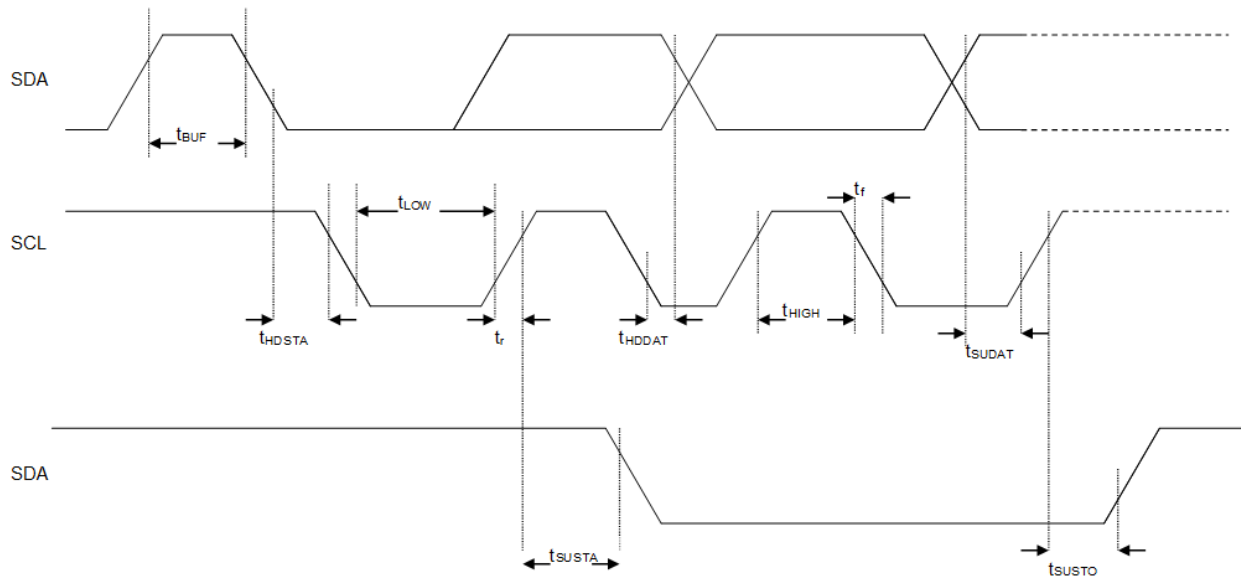


Figure 10: I<sup>2</sup>C timing diagram

Table 13: I<sup>2</sup>C timings

Parameter	Symbol	Condition	Min	Typ	Max	Unit
SDA setup time	$t_{SU,DAT}$	S&F Mode HS mode	160 30			ns ns
SDA hold time	$t_{HD,DAT}$	S&F Mode, $C_b \leq 100$ pF S&F Mode, $C_b \leq 400$ pF HS mode, $C_b \leq 100$ pF HS mode, $C_b \leq 400$ pF	80 90 18 24		115 150	ns ns ns ns
SCL low pulse	$t_{LOW}$	HS mode, $C_b \leq 100$ pF $V_{DDIO} = 1.62$ V	160			ns
SCL low pulse	$t_{LOW}$	HS mode, $C_b \leq 100$ pF $V_{DDIO} = 1.2$ V	210			ns
SDA setup time	$t_{SU,DAT}$	S&F Mode HS mode	160 30			ns ns

The above-mentioned I<sup>2</sup>C specific timings correspond to the following internal added delays:

- ▶ Input delay between SDA and SCK inputs: SDA is more delayed than SCK by typically 100 ns in Standard and Fast Modes and by typically 20 ns in Fast+ Mode.
- ▶ Output delay from SCL falling edge to SDA output propagation is typically 140 ns in Standard and Fast Modes and typically 70 ns in Fast+ Mode.

### 9.4.3 I3C timings

The I3C timing values are given in Table 15.

Table 14: I3C timings

Parameter	Symbol	Condition	Min	Typ	Max	Unit
I3C clock i/p frequency	F_i3c		160 30	0.01		12.9
SCL low pulse	T_low_scl		80 90 18 24			
SCL high pulse	T_high_scl		160			
SDA setup time	T_setup_sda		210			
SDA hold time	T_hold_sda		160 30			
SDA output delay	T_delay_sda	50 pF load, V <sub>DDIO</sub> =1.72V				12

## 10. Package information

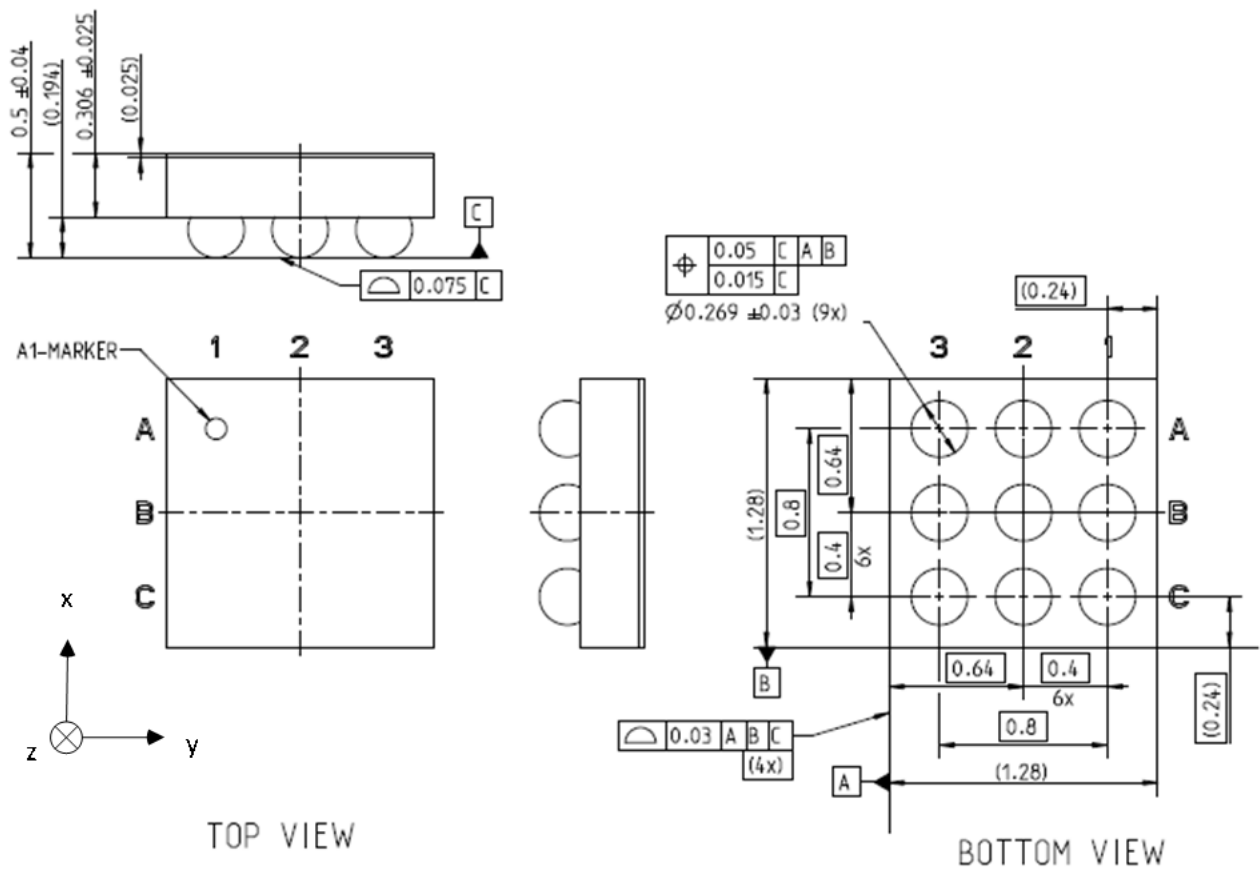


Figure 11: Pin out and package dimensions

## 10.1 Marking/Lasermarking, coordinate system

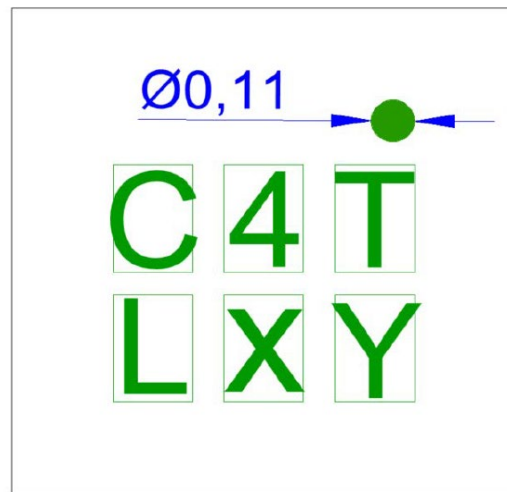


Figure 12: Marking

Lasermarking is done on a per-die basis for engineering samples. The 2<sup>nd</sup> digit, “variant identifier”, tells which TMR variant is on the die.

Marking of Engineering Samples BMM350 (A, C-samples)

Top view Labeling	Name	Marking position	Remark																																																											
<p style="text-align: center;">●</p> <p style="text-align: center;">123 456</p>	Sample stage, product	123	“C4T” : C samples, bmm350																																																											
	Lot identifier	4	Lot identifier																																																											
	Wafer identifier	5,6	(1) One code wafer ID convert table 一碼 ID 轉換表 <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>Wafer ID</th> <th>1-digit ID</th> <th>Wafer ID</th> <th>1-digit ID</th> <th>Wafer ID</th> <th>1-digit ID</th> </tr> </thead> <tbody> <tr><td>01</td><td>1</td><td>10</td><td>A</td><td>19</td><td>N</td></tr> <tr><td>02</td><td>2</td><td>11</td><td>C</td><td>20</td><td>P</td></tr> <tr><td>03</td><td>3</td><td>12</td><td>D</td><td>21</td><td>R</td></tr> <tr><td>04</td><td>4</td><td>13</td><td>E</td><td>22</td><td>T</td></tr> <tr><td>05</td><td>5</td><td>14</td><td>F</td><td>23</td><td>U</td></tr> <tr><td>06</td><td>6</td><td>15</td><td>H</td><td>24</td><td>X</td></tr> <tr><td>07</td><td>7</td><td>16</td><td>J</td><td>25</td><td>Y</td></tr> <tr><td>08</td><td>8</td><td>17</td><td>K</td><td></td><td></td></tr> <tr><td>09</td><td>9</td><td>18</td><td>L</td><td></td><td></td></tr> </tbody> </table>	Wafer ID	1-digit ID	Wafer ID	1-digit ID	Wafer ID	1-digit ID	01	1	10	A	19	N	02	2	11	C	20	P	03	3	12	D	21	R	04	4	13	E	22	T	05	5	14	F	23	U	06	6	15	H	24	X	07	7	16	J	25	Y	08	8	17	K			09	9	18	L	
Wafer ID	1-digit ID	Wafer ID	1-digit ID	Wafer ID	1-digit ID																																																									
01	1	10	A	19	N																																																									
02	2	11	C	20	P																																																									
03	3	12	D	21	R																																																									
04	4	13	E	22	T																																																									
05	5	14	F	23	U																																																									
06	6	15	H	24	X																																																									
07	7	16	J	25	Y																																																									
08	8	17	K																																																											
09	9	18	L																																																											

Marking of Mass Production Parts BMM350

Top view Labeling	Name	Marking position	Remark
<p style="text-align: center;">●</p> <p style="text-align: center;">4TY XXX</p>	Product identifier	T	2 digit (alphanumeric), fixed; for identification of device type (“A” = T is BMM350)
	Internal number	Y	internal
	Lot identifier	XXX	3 digits (alphanumeric 0–Z), variable, no reset
	Pin A1 identifier top side	●	n/a

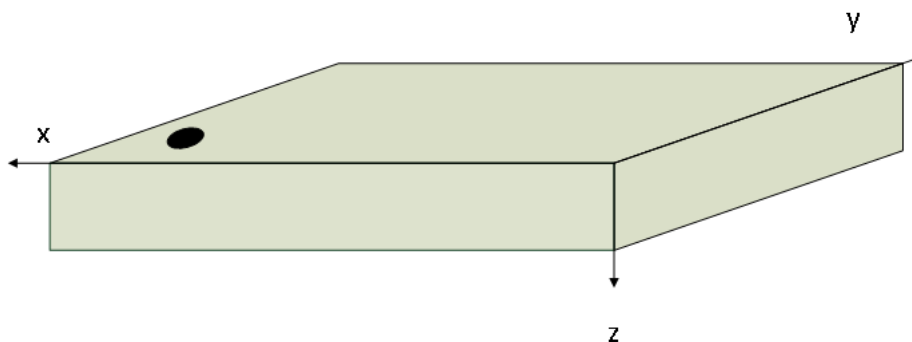


Figure 13: Coordinate system definition, right-handed. lasermarking side up

## 11. Examples of use

The examples are given in the API provided by Bosch Sensortec. Please contact the Sales representative for more Information.



## 12. Timing control

### 12.1 Timing control Asynchronous mode

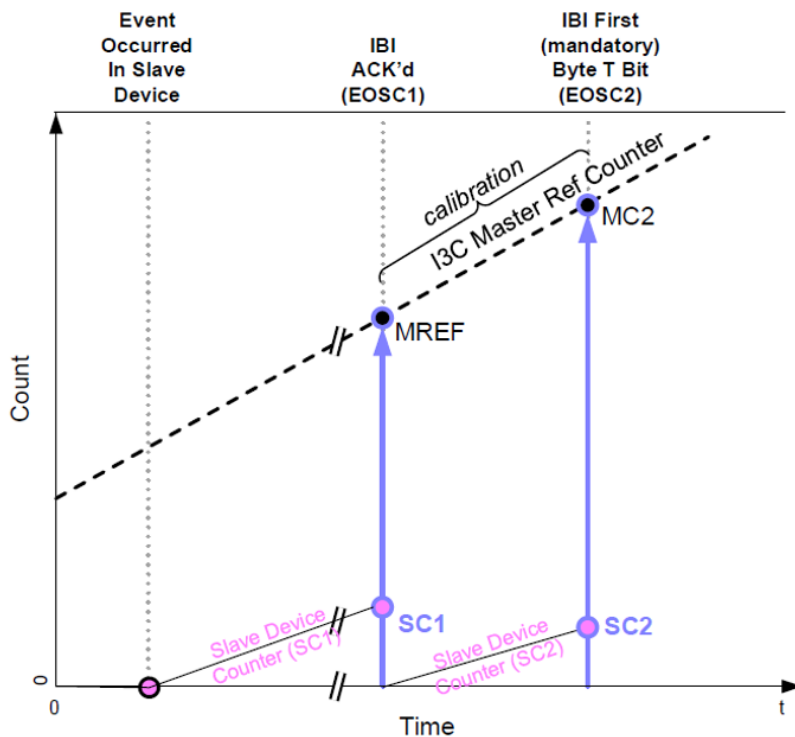


Figure 14: Timing control async mode, from MIPI I3C spec

The BMM350 supports timing control asynchronous mode 0.

In the MIPI I3C specification as of 1-1\_r13 (26 July 2019) the functions of this mode are described in chapter 5.1.8.3.1. The BMM350 supports this mode employing its 4.32MHz internal clock as “slave device counter” (see Figure 14). The BMM350 will report the counter values SC1 (time between magnetic data ready and IBI acknowledged) and SC2 (time between IBI acknowledged and IBI first mandatory byte T bit) on the basis of the said 4.32MHz oscillator. The counter value SC2 depicts the read time of the mandatory byte in slave clocks. When doing the math SC2 will vary between 2 and 3, assuming 12.5MHz I3C clock frequency. This small count will lead to a high variation of the timepoint calculation done in the master. To improve accuracy it would be an option to use clock stalling (see MIPI I3C spec 5.1.2.5 “Master clock stalling”), at least for a short initialization period, so that the master can perform an accurate calculation of the slave clock frequency. Clock stalling will reduce the bus bandwidth, so that option should be used with care. MIPI spec 5.1.2.5.3 “I3C Read Transfer, Transition Bit” explains how to do clock stalling during the mandatory byte read.

For proper function the I3C master will query the BMM350 using the GETXTIME CCC, asking for the oscillator accuracy (in the order of 5%), oscillator frequency in multiples of 500kHz, supported mode (async0 mode and sync mode).

The SETXTIME CCC command will be used to switch the BMM350 into timing control mode, in this chapter to async0 mode.

The timing control async0 functionality is based on in-band interrupts.

After SETXTIME CCC 0xDF (=enter async0 mode) it is not necessary to configure in-band interrupts.

### 13. Recommendations for external capacitor CRST

For the 2.2uF external capacitor a low-inductance low-ESR type is recommended.

The recommended type is

[https://product.tdk.com/de/search/capacitor/ceramic/mlcc/info?part\\_no=CGB4B3X7R0J225K055AB](https://product.tdk.com/de/search/capacitor/ceramic/mlcc/info?part_no=CGB4B3X7R0J225K055AB)

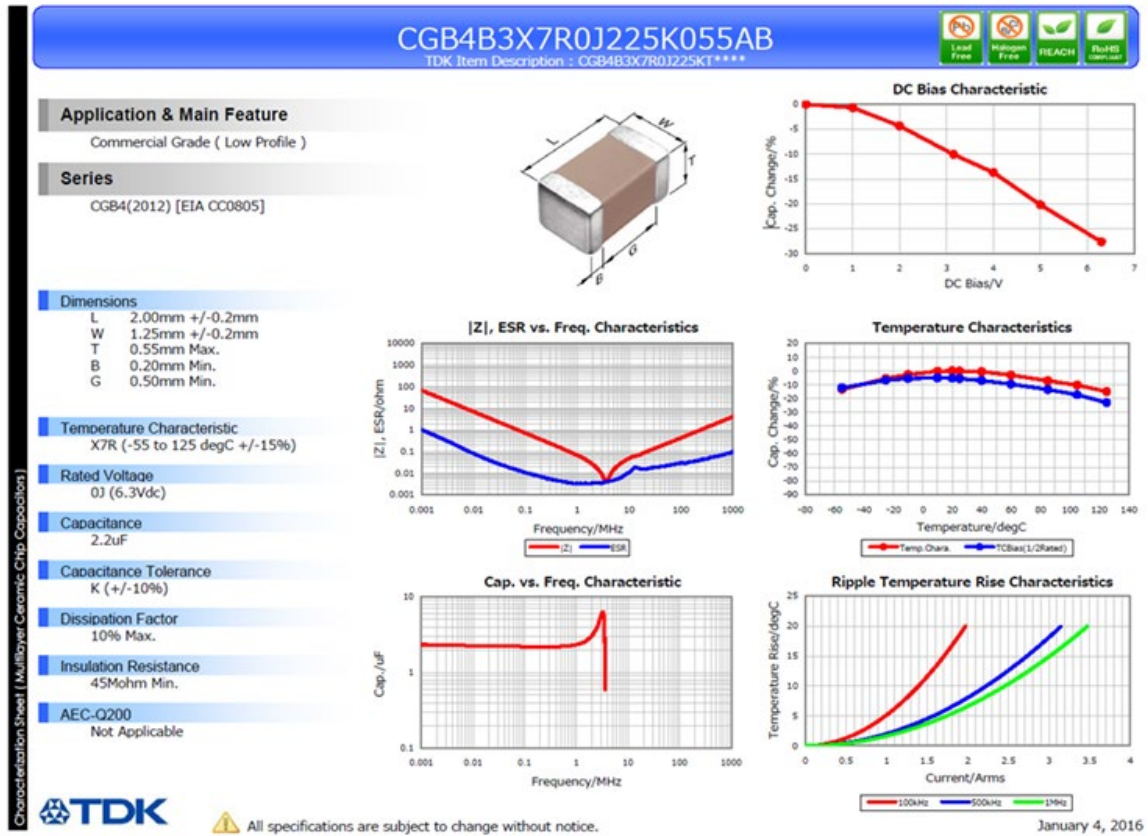


Figure 15. External capacitor

## 14. Environmental Safety

### 14.1 RoHS

The BMM350 sensor meets the requirements of the EC restriction of hazardous substances (RoHS) directive, see also: Directive 2015/863/EU (amending Annex II to Directive 2011/65/EU) of the European Parliament and of the Council on the restriction of the use of certain hazardous substances in electrical and electronic equipment.

### 14.2 Halogen content

The BMM350 is halogen-free. For more details on the analysis results please contact your Bosch Sensortec representative.

### 14.3 Internal Package Structure

Within the scope of Bosch Sensortec's ambition to improve its products and secure the mass product supply, Bosch Sensortec qualifies additional sources (e.g. 2nd source) for the package of the BMM350.

While Bosch Sensortec took care that all of the technical packages parameters are described above are 100% identical for all sources, there can be differences in the chemical content and the internal structural between the different package sources.

However, as secured by the extensive product qualification process of Bosch Sensortec, this has no impact to the usage or to the quality of the BMM350 product.

## 15. Acronyms

ADC	Analog to digital converter
API	application programming interface: set of C functions to configure bmm350 and read out compensated data
avg	A bit field inside the register map, controlling the internal averaging setting of bmm350 and thus the noise performance
AR/VR	Augmented reality/virtual reality
ASIC	Application specific integrated circuit
BW	bandwidth
CCC	Common Command Code (see I3C)
CDM	Charged device model (see ESD)
CRST	pin connected to external 2.2uF capacitor. Said capacitor is called "CRST capacitor"
drdy	Data ready, data ready interrupt
FM	Forced mode: an operation mode which performs a single magnetic-to-digital conversion. Must be used for ODR<25
FM-Fast	Forced mode fast: an operation mode which performs a single magnetic-to-digital conversion.
HBM	Human body model (see ESD)
IBI	In-band-interrupt, I3C serial interrupt
INL	Non-linearity error, here: maximum deviation from sensor output regression curve
ESD	Electrostatic discharge
HDR	High data rate (I3C)
INT	Interrupt, also : interrupt pin
ODR	Output data rate, in samples/s
OTP	One time programmable memory
PCB	Printed circuit board
PMU	Power management unit, not to mix with PMU_CMD: command processor register
RoHS	Restriction of Hazardous Substances
SDR	Single data rate (I3C mode)
TCO	Temperature coefficient offset
TCS	Temperature coefficient sensitivity
TMR	Tunnel Magnetoresistance
VDD	Main supply
VDDIO	IO supply
WLCSP	wafer level chip-scale package
XLSB	Extended LSB: Bits 17-24 of the data and temperature registers

## 16. Legal disclaimer

### i. Engineering samples

Engineering Samples are marked with an asterisk (\*), (E) or (e). Samples may vary from the valid technical specifications of the product series contained in this data sheet. They are therefore not intended or fit for resale to third parties or for use in end products. Their sole purpose is internal client testing. The testing of an engineering sample may in no way replace the testing of a product series. Bosch Sensortec assumes no liability for the use of engineering samples. The Purchaser shall indemnify Bosch Sensortec from all claims arising from the use of engineering samples.

### ii. Product use

Bosch Sensortec products are developed for the consumer goods industry. They may only be used within the parameters of this product data sheet. They are not fit for use in life-sustaining or safety-critical systems. Safety-critical systems are those for which a malfunction is expected to lead to bodily harm, death or severe property damage. In addition, they shall not be used directly or indirectly for military purposes (including but not limited to nuclear, chemical or biological proliferation of weapons or development of missile technology), nuclear power, deep sea or space applications (including but not limited to satellite technology).

Bosch Sensortec products are released on the basis of the legal and normative requirements relevant to the Bosch Sensortec product for use in the following geographical target market: BE, BG, DK, DE, EE, FI, FR, GR, IE, IT, HR, LV, LT, LU, MT, NL, AT, PL, PT, RO, SE, SK, SI, ES, CZ, HU, CY, US, CN, JP, KR, TW. If you need further information or have further requirements, please contact your local sales contact.

The resale and/or use of Bosch Sensortec products are at the purchaser's own risk and his own responsibility. The examination of fitness for the intended use is the sole responsibility of the purchaser.

The purchaser shall indemnify Bosch Sensortec from all third party claims arising from any product use not covered by the parameters of this product data sheet or not approved by Bosch Sensortec and reimburse Bosch Sensortec for all costs in connection with such claims.

The purchaser accepts the responsibility to monitor the market for the purchased products, particularly with regard to product safety, and to inform Bosch Sensortec without delay of all safety-critical incidents.

### iii. Application examples and hints

With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Bosch Sensortec hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights or copyrights of any third party. The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. They are provided for illustrative purposes only and no evaluation regarding infringement of intellectual property rights or copyrights or regarding functionality, performance or error has been made.

## 17. Document history and modification

Rev. No	Chapter	Description of modification/changes	Date
1.0	-	Initial release	August 2019
1.20	all	Final datasheet	Apr 2023
1.21	Table 2.1	Sensor start-up time description	July 2023
	15	Modified Acronyms	
	5	Deleted POR description	

**Bosch Sensortec GmbH**

Gerhard-Kindler-Straße 9  
72770 Reutlingen / Germany

[contact@bosch-sensortec.com](mailto:contact@bosch-sensortec.com)  
[www.bosch-sensortec.com](http://www.bosch-sensortec.com)

Modifications reserved

Preliminary - specifications subject to change without notice

Document number: BST-BMM350-DS001-21