Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

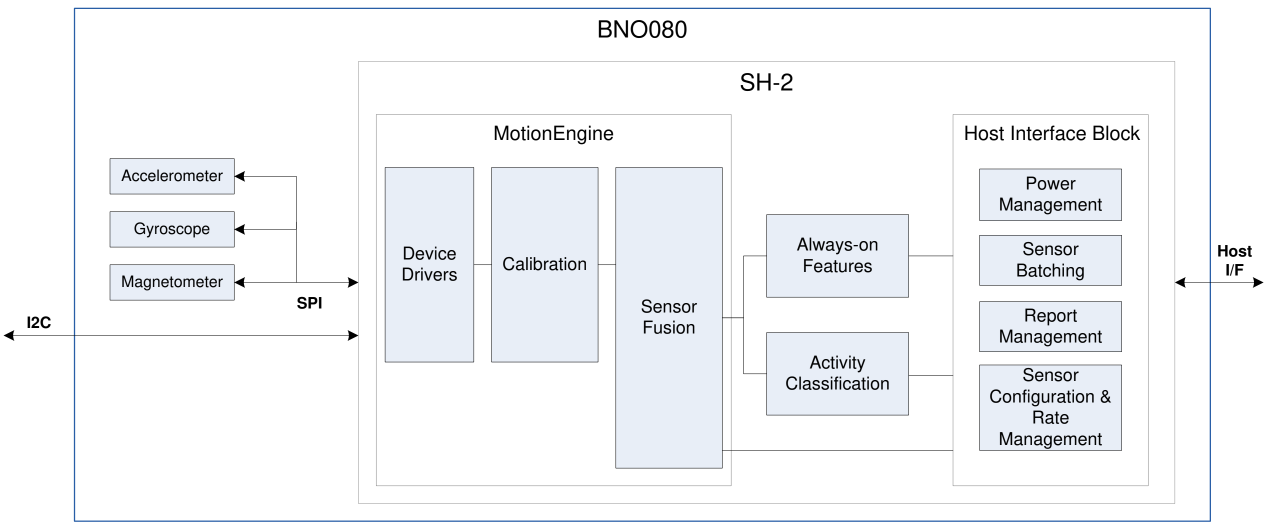
Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

BNO080 Data Sheet

The BNO080 is a System in Package (SiP) that integrates a triaxial accelerometer, triaxial gyroscope, magnetometer and a 32-bit ARM® Cortex™-M0+ microcontroller running Hillcrest's SH-2 firmware. The SH-2 includes the MotionEngine™ software, which provides sophisticated signal processing algorithms to process sensor data and provide precise real-time 3D orientation, heading, calibrated acceleration and calibrated angular velocity, as well as more advanced contextual outputs. The BNO080 is integrated into a single 28 pin LGA 3.8mm x 5.2mm x 1.1mm package. It is compatible with Android and provides a turn-key sensor hub solution, eliminating the complexity and investment associated with a discrete design.

Turn-key Sensor Hub Solution



- Android compliant 4.4 KitKat and above
- Variety of 3D orientation outputs (including linear acceleration, rotation vectors, gravity)
- Includes "always-on" and classification features (step counter, stability and tap detectors, and a variety of gestures)
- Dynamically calibrates sensor data for temperature and aging

Sensor Hub Features
- Three sensors and microcontroller in a single device
- Supports new sensors and features in Android 4.4 KitKat
- Includes power management to optimize power
- Support for device firmware upgrade (DFU)

Extensive Data Modes
- Linear acceleration
- Angular velocity
- Angular position (quaternion)
- Data returned at configurable sample rates
- Timestamps attached to sensor reports
- AR/VR stabilization applied to rotation vectors provides visually improved angular position output
- Low latency, 1kHz gyro rotation vector for AR/VR applications

"Always-on" and Classification Features
- Built-in stability detector, tap detector, and step counter

Applications
- Wearables such as head trackers for AR/VR applications, smartwatches, fitness bands, audio headsets
- Smartphones
- Tablets
- Ultrabooks
- Robotics

Table of Contents

List of Figures

1 Functional Overview

The BNO080 is manufactured by Bosch Sensortec and runs software provided by Hillcrest Labs. The BNO080 integrates a triaxial 12-bit accelerometer with a range of ±8g, triaxial 16-bit gyroscope with a range of ±2000 degrees per second, a triaxial geomagnetic sensor, and a 32-bit ARM® Cortex™-M0+ microcontroller. The sensors are provided by Bosch Sensortec GmbH and the Cortex M0+ processor by Atmel Corporation.

A system diagram of the BNO080 is shown in Figure 1-1.

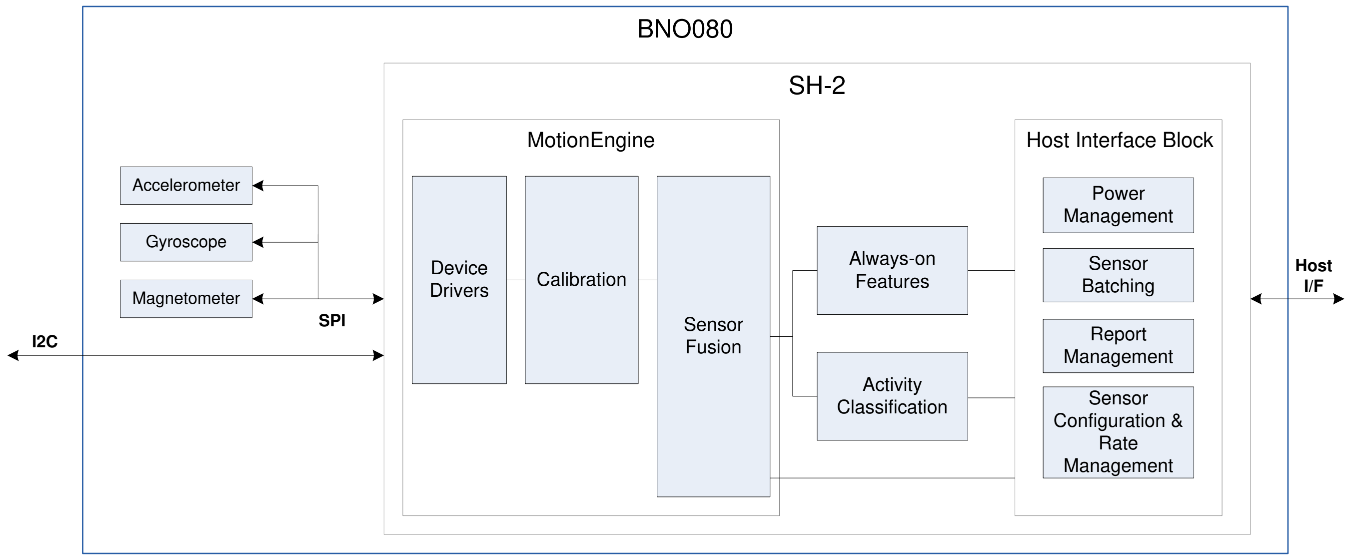


Figure 1-1: BNO080 block diagram

At the heart of the BNO080 is Hillcrest's SH-2 software. The SH-2 software includes MotionEngine™, 'always-on' features, activity classification and the host interface software. MotionEngine is digital signal processing software that takes raw motion data from the MEMS sensors and translates this data into precise motion information. These accurate motion outputs can be used for gesture detection and a variety of advanced motion-controlled applications. The 'always-on' and activity classification features include step counter, stability detector, tap detector, and gestures. The host interface module includes sophisticated power management functionality, configures sensors, and handles communication with the system host. The SH-2 supports sensor types defined in Android 4.4 KitKat. Android 5.0 defined additional methods of configuration which the BNO080 supports.

The BNO080 supports the addition of environmental sensors on a secondary I²C interface. See 2.3.

The BNO080 can communicate with the system host over various serial interfaces: SPI, I²C and UART.

1.1 Reference Design Configurations

1.1.1 Standalone Sensor Hub Solution in Mobile Devices

The BNO080 can be integrated as a co-processor in a mobile device such as a smartphone, tablet or ultrabook. This means the host processor can offload the sensor management and processing functions to the BNO080. The BNO080 constantly processes sensor data in the background while the host processor is in sleep state and can provide the host processor with real-time sensor data on demand. This low power paradigm is useful to enable a number of 'always-on' applications such as activity tracking, gesture recognition, fitness and pedestrian dead reckoning.

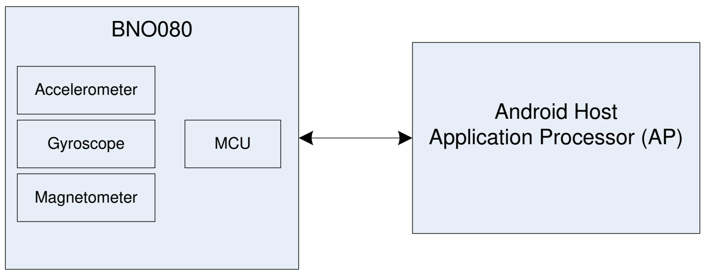


Figure 1-2: BNO080 in a mobile device

1.1.2 Virtual Reality Head Tracker

The BNO080 can be integrated into an HMD as a head tracker to allow the user to be immersed in either a virtual reality or an augmented reality application. The BNO080 provides accurate angular position data to allow navigation of this virtual world.

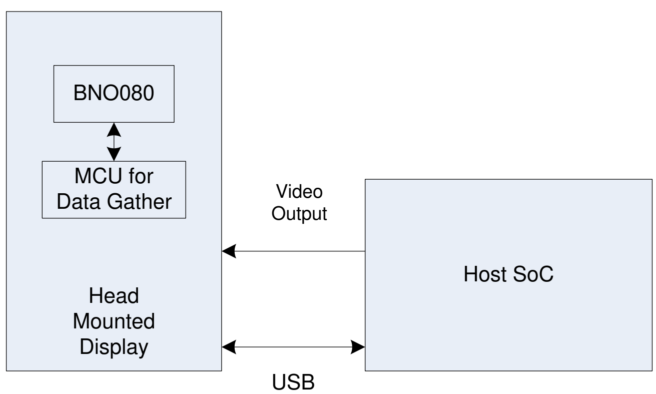


Figure 1-3: Virtual reality head tracker

1.1.4 Robot Vacuum Cleaner

The BNO080 can be integrated into a robot vacuum cleaner, providing heading information to fuse with cameras to allow navigation (SLAM – Simultaneous Localization and Mapping application). The BNO080 can also provide details on the tilt of the vacuum and by virtue of the included accelerometer can serve as a bump detector.

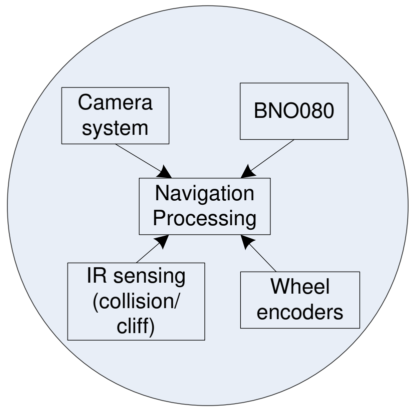


Figure 1-4: Robot vacuum cleaner

1.3 BNO080 Connectivity

The BNO080 can support connections to a host microcontroller through various serial interfaces:

- I²C interface
- UART interface
- SPI interface
- UART-RVC interface – a simplified UART interface for Robot Vacuum Cleaners

In addition, the BNO080 includes a bootloader to allow for firmware upgrades. The bootloader can support I²C, SPI or UART. Access to the bootloader is achieved by setting BOOTN to 0.
Configuration of the communication interface is achieved by setting the protocol selection (PS1/0) pins appropriately:

PS1	PS0	BNO080 (BOOTN=1)	BNO080 bootloader (BOOTN=0)
0	0	I²C	I²C
0	1	UART-RVC	Reserved
1	0	UART	UART
1	1	SPI	SPI

Figure 1-5: Protocol selection for BNO080

The protocol selection and BOOTN pins are sampled at reset.
PS0 is repurposed as a WAKE signal in SPI mode following reset (see 6.5.3 for timing).

1.3.1 Pin Descriptions

Figure 1-6 describes the function of each pin.

Pin Number	BNO080 Name	Mode	Description
1	RESV_NC	NC	Reserved. No connect.
2	GND	Input	Ground
3	VDD	Input	Supply voltage (sensors) (2.4V to 3.6V)
4	BOOTN	Input	Bootloader mode select
5	PS1	Input	Protocol Select pin 1
6	PS0/WAKE	Input	Protocol Select pin 0, also used to wake processor in SPI mode
7	RESV_NC	Input	Reserved. No connect.
8	RESV_NC	NC	Reserved. No connect.
9	CAP		External capacitor (100nF to GND)
10	CLKSEL0	Input	Clock source selection. Internal pulldown.
11	NRST	Input	Active low reset
12	RESV_NC	NC	Reserved. No connect.
13	RESV_NC	NC	Reserved. No connect.
14	H_INTN	Output	Interrupt to host device
15	ENV_SCL	Bidirectional	Environmental sensor I^2C clock
16	ENV_SDA	Bidirectional	Environmental sensor I^2C data
17	SA0/H_MOSI	Input	Lower address bit of device address. In SPI mode, data input
18	H_CSN	Input	SPI chip select, active low
19	H_SCL/SCK/RX	Bidirectional	Host Interface I^2C clock, SPI clock or UART RX
20	H_SDA/H_MISO/TX	Bidirectional	Host Interface I^2C data, SPI data out or UART TX
21	RESV_NC	NC	Reserved. No connect.
22	RESV_NC	NC	Reserved. No connect.
23	RESV_NC	NC	Reserved. No connect.
24	RESV_NC	NC	Reserved. No connect.
25	GND	Input	Ground
26	XOUT32/CLKSEL1	Output	32K crystal output / clock source selection. Internal pulldown.
27	XIN32	Input	32K crystal input. / external clock
28	VDDIO	Input	Supply voltage (core and I/O domain) (1.65V to 3.6V)

Figure 1-6: BNO080 pin descriptions

The BNO080 can operate from an internal oscillator, an external 32.768 kHz clock or an external 32.768 kHz crystal. If an external clock is used it must be connected to pin 27. Hillcrest recommends a tolerance of 50ppm. If a crystal is used it must be connected across pins 26 and 27. Hillcrest recommends using a crystal with tolerance 50ppm with 22pF capacitor loading.

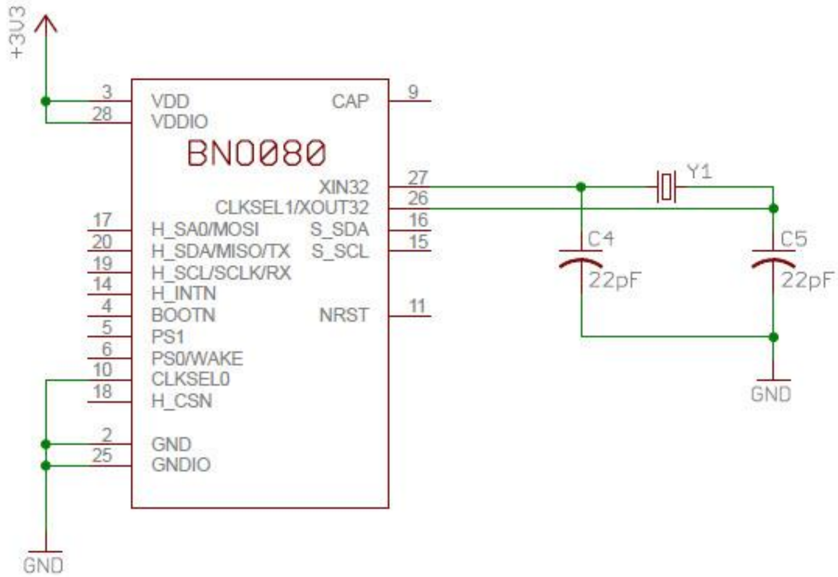


Figure 1-7: 32.768kHz crystal connection

Clock source selection is done during startup using the CLKSEL0 and CLKSEL1 pins. Figure 1-8 shows how to select each source.

Source	CLKSEL0	CLKSEL1
Crystal	0 or unconnected	Connected to crystal
External	1	1
Internal	1	0 or unconnected

Figure 1-8: Clock Source Selection

CLKSEL0 is configured with an internal pulldown at startup. After CLKSEL0 has been sampled, if it is found to be a 1, the internal pulldown is disabled to avoid unnecessary power consumption. If either the external or internal clock sources are selected then CLKSEL1 is configured as an input with a pulldown. CLSKSEL1 is then sample and the clock source determined. After CLKSEL1 has been sampled, if it is found to be a 1, the internal pulldown is disabled to avoid unnecessary power consumption.

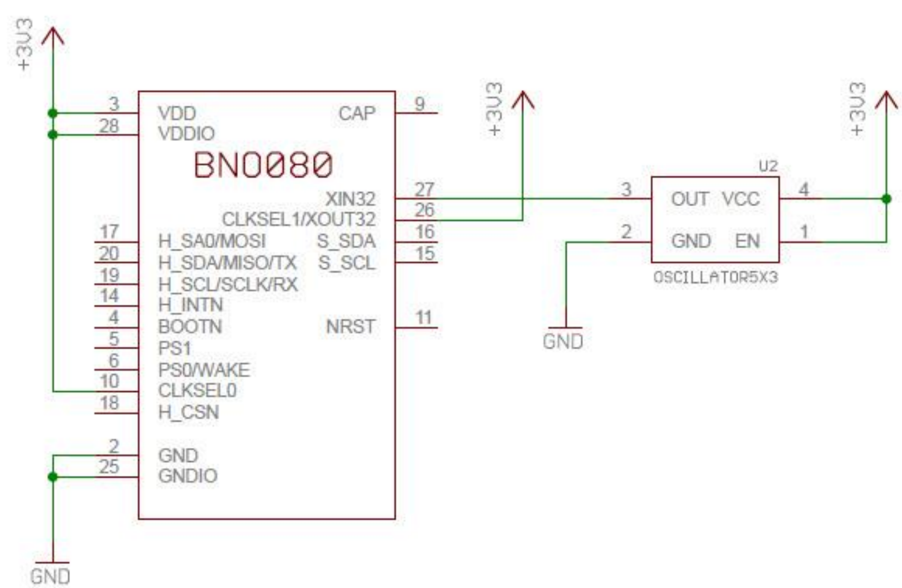


Figure 1-9: External clock connection

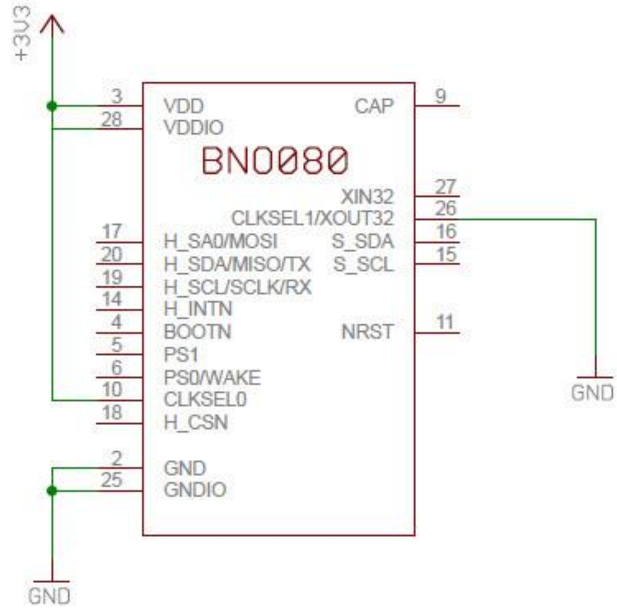


Figure 1-10: Internal clock selection

1.3.2 I²C interface

The BNO080 supports a standard Fast mode I²C interface and can communicate over this interface at up to 400kb/s.

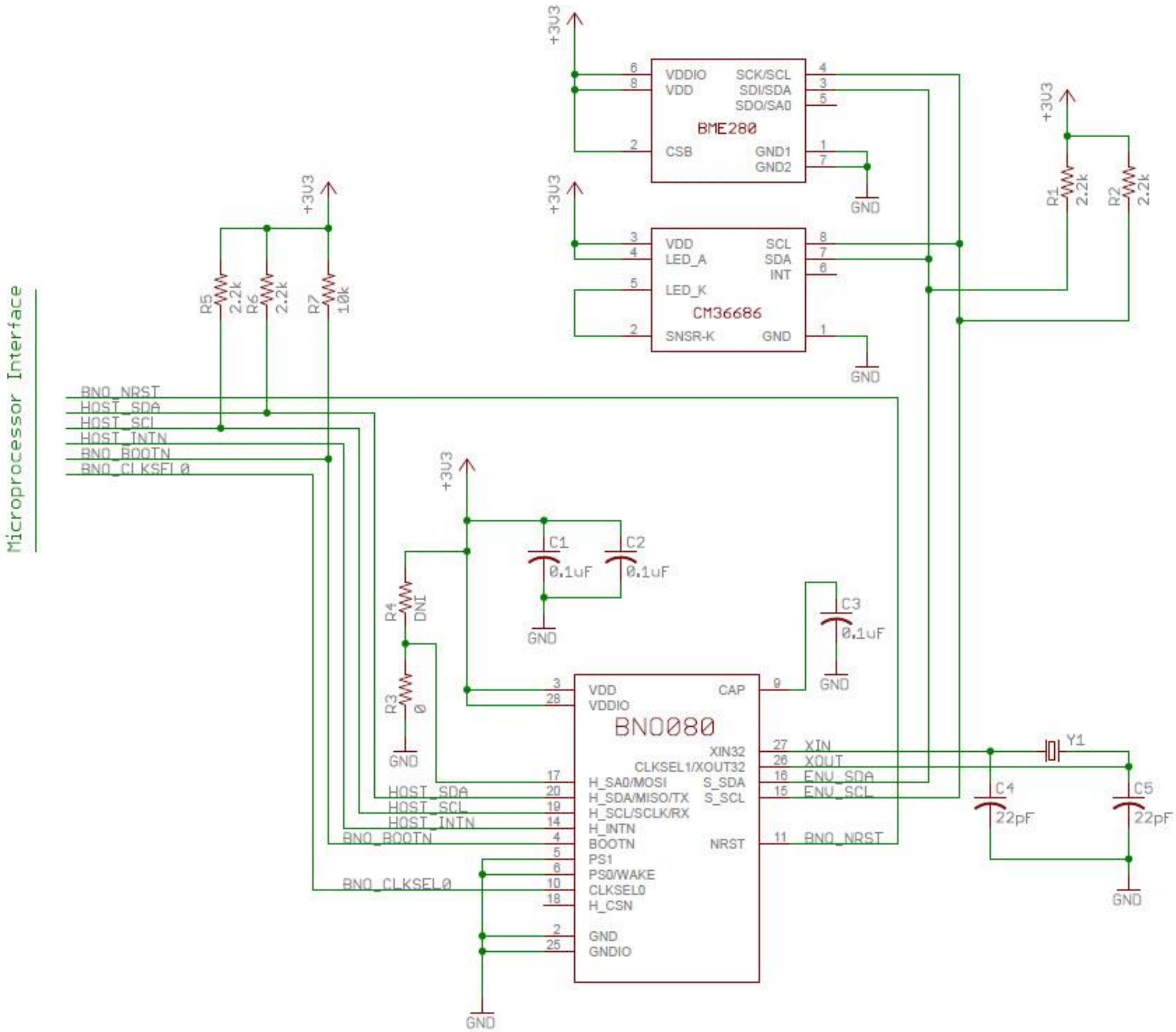


Figure 1-11: BNO080 I²C connection diagram

Figure 1-11 shows how the BNO080 can be connected to an external microcontroller via the I²C interface. The following notes are provided as guidelines for connecting the BNO080 in a system design.

1. The H_INTN pin is the application interrupt line that indicates the BNO080 requires attention. This should be tied to a GPIO with wake capability. The interrupt is active low.
2. NRST is the reset line for the BNO080 and can be either driven by the application processor or the board reset.
3. BOOTN is sampled at reset. If low the BNO080 will enter bootloader mode.
4. Pin 4 (BOOTN) should be pulled high through a 10K Ohms resistor. To use the device firmware update (DFU) capability of the BNO080, it is recommended to connect Pin 4 to a GPIO pin on the external microcontroller.
5. Pin 5 (PS1) and Pin 6 (PS0/WAKE) are the host interface protocol selection pins. These pins should be tied to ground to select the I²C interface.
6. Pin 17 (SA0) can be used to select the lower bit of the 7-bit I²C slave device address.

7. Pullup resistors (R1 and R2) are needed on the I²C communication lines – Pin 19 (HOST_SCL) and Pin 20 (HOST_SDA). These values may vary depending on the board design and bus capacitance, but typical values are between 2KΩ and 4KΩ.
8. The BNO080 supports environmental sensors (e.g. pressure sensors, ambient light sensors) on a secondary I²C interface. This interface should be pulled up via resistors regardless of the presence of the external sensor as the SW polls for sensors at reset.

1.3.2.1 I²C Operation

The I²C specification is documented in [4]. The BNO080 provides a slave interface to the application processor and supports 100kbps Standard mode (Sm) and 400kbps Fast mode (Fm).

I²C is a two-wire serial interface that consists of a serial clock line (SCL) and a serial data line (SDA). I²C allows for multi-master, multi-slave communication and uses an open-drain architecture to enable this capability. All devices that drive the SDA line or SCL line can only drive the line low ('0'); the bus is released and pulled high by the pull-up resistor for '1's. The master device places the slave address on the data bus and the slave device with the corresponding address acknowledges the master.

The BNO080 I²C interface answers to a 7-bit address of either 0x4A or 0x4B:

6	5	4	3	2	1	0
1	0	0	1	0	1	SA0

Figure 1-12: BNO080 I²C address

The lower address bit of the device address is provided by the SA0 pin. This pin is sampled at reset and should be tied to either a logic high or logic low.

1.3.2.2 I²C Protocol

In general the data line should be stable while the clock is high, data transitions should therefore occur when the clock is low. Communication on an I²C bus is initiated with the master device presenting a START command and terminated with a STOP command. The bus is busy between the two commands. A START command (or condition) is defined as a transition on the SDA signal from high to low while the SCL signal is high (Figure 1-13). A STOP command (or condition) is defined as a low to high transition on the SDA signal while the SCL signal is high (Figure 1-14).

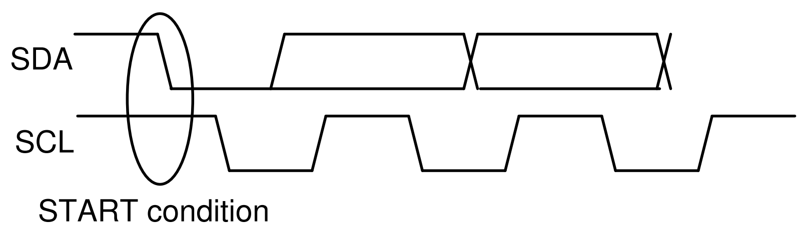


Figure 1-13: I²C START condition

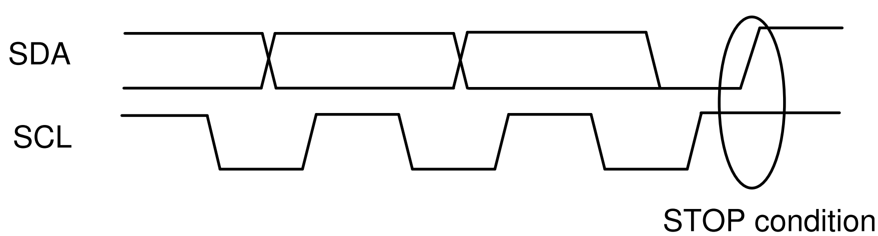


Figure 1-14: I²C STOP condition

I²C is a byte oriented protocol. Hence each element passed between the master and slave is 8-bits long. The bits within the byte are transmitted most-significant bit (MSB) first. For multi-byte words the data is presented in littleendian format – least-significant byte first. The master device generates the clock. Every byte transmitted must be acknowledged. An acknowledgement is generated by the device receiving the data and is formed by the receiver driving the SDA line low during the ninth bit.

The master device generally drives the clock. However, if the slave device requires additional time to respond it can force the clock low, only releasing the line when it is prepared to deliver more data. The master device MUST support clock stretching.

The first byte presented after the start condition contains the device address and the read/write bit. The least-significant bit (LSB) of this first byte is the read/write indication (‘0’ corresponding to write)

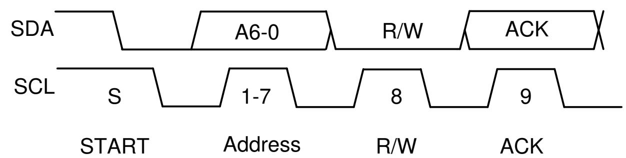


Figure 1-15: Device addressing

For a write cycle data is provided after the device address. The slave (BNO080 in this instance) provides an ACK for every byte received. A typical I²C write cycle is provided below (S=start, P=stop, AD=Address).

Master	**S**	**AD+W**		**DATA**		**DATA**		**DATA**		○ ○	**DATA**		**P**
Slave			**ACK**		**ACK**		**ACK**		**ACK**			**ACK**	

Figure 1-16: I²C write cycle

A read cycle requires that the device must first be selected with its device address. Following the device address the BNO080 will provide data. Every byte should be acknowledged by the master:

Master	**S**	**AD+R**			**ACK**		**ACK**		**ACK**	○ ○		**NAK**	**P**
Slave			**ACK**	**DATA**		**DATA**		**DATA**			**DATA**		

Figure 1-17: I²C read cycle

The BNO080 uses Hillcrest’s SHTP (Sensor Hub Transport Protocol) protocol to communicate. The BNO080 application does not support the repeated start method for typical I²C register based interfaces. More details are available in [2].

1.3.3 UART-SHTP interface

The BNO080 provides a UART communication interface that supports Hillcrest’s SHTP protocol. The UART interface operates at 3Mb/s. A typical connection is shown in Figure 1-18

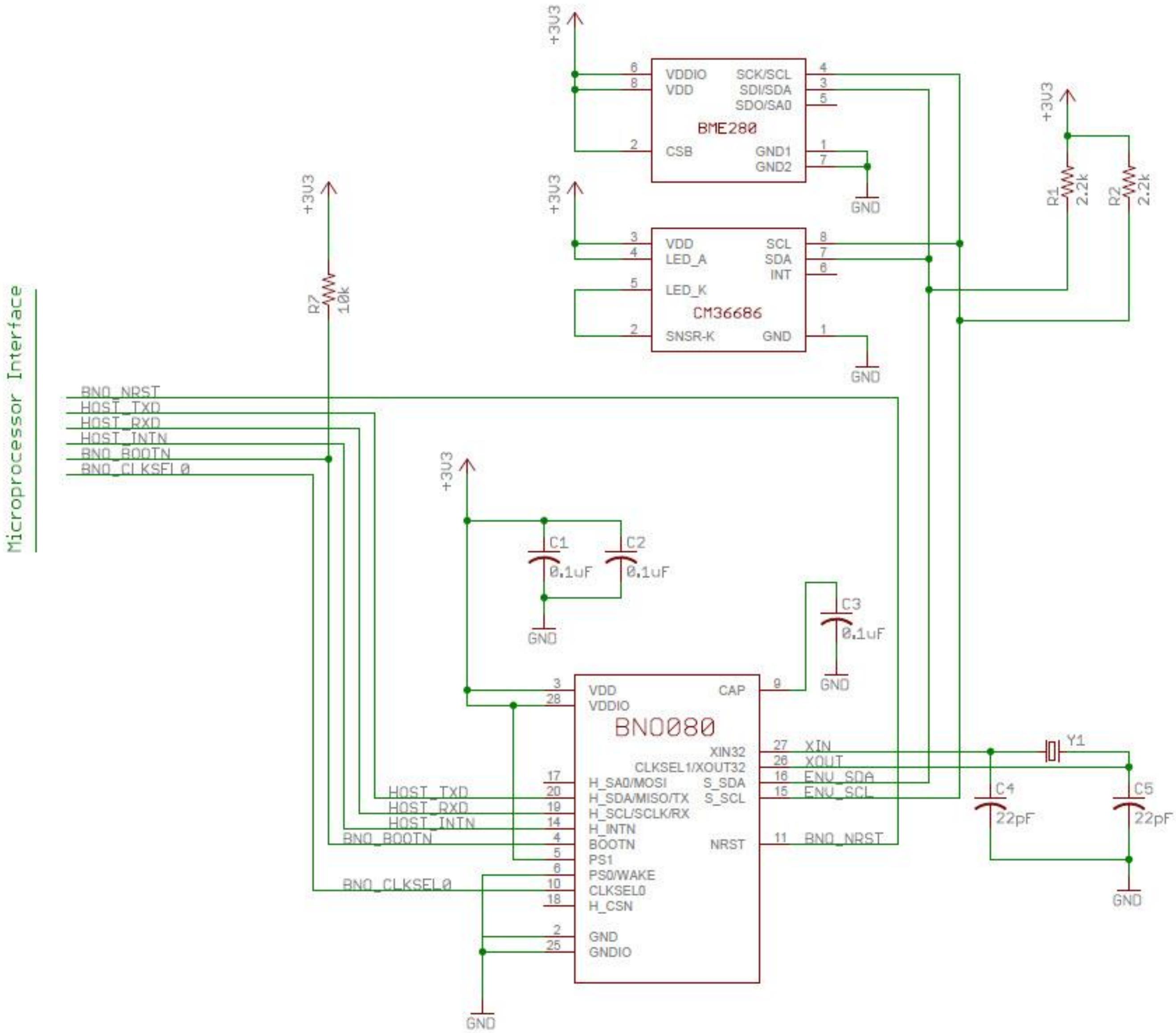


Figure 1-18: BNO080 UART-SHTP connection diagram

Figure 1-18 shows how the BNO080 can be connected to an external microcontroller via a UART interface. The following notes are provided as guidelines for connecting the BNO080 in a system design.

1. The H_INTN pin is driven low prior to the initial byte of UART transmission. It will deassert and reassert between messages. It is used by the host to timestamp the beginning of data transmission.
2. NRST is the reset line for the BNO080 and can be either driven by the application processor or the board reset.
3. BOOTN is sampled at reset. If low the BNO080 will enter bootloader mode.
4. Pin 4 (BOOTN) should be pulled high through a 10K Ohms resistor. To use the device firmware update (DFU) capability of the BNO080, it is recommended to connect Pin 4 to a GPIO pin on the external microcontroller.
5. Pin 5 (PS1) and Pin 6 (PS0/WAKE) are the host interface protocol selection pins. These pins should be tied to VDDIO and ground respectively to select the UART-SHTP interface.

6. The BNO080 supports environmental sensors (e.g. pressure sensors, ambient light sensors) on a secondary I^{2}C interface. This interface should be pulled up via resistors regardless of the presence of the external sensor as the SW polls for sensors at reset.

1.3.3.1 UART operation

The UART is configured for 3Mkb/s, 8 data bits, 1 stop bit and no parity. The UART protocol relies on an idle line being 'high'. A transmission is started with the assertion of a start bit (pulling the line low), followed by the data, LSB first. After the data segment is sent (in this case 8-bits), the line is pulled high (the stop signal) for a minimum number of bits (1 for the BNO080) to indicate end of that segment. Bytes sent from the host to the BNO080 must be separated by at least 100us. Bytes sent from the BNO to the host have no extra spacing.

Figure 1-19: UART signaling

The BNO080 uses Hillcrest's SHTP protocol to communicate. The UART protocol makes use of framing bytes at the start and end of transmission. More details are available in [2].

1.3.4 SPI Interface

The BNO080 supports 4-wire Serial Peripheral Interface (SPI) for host communication. A typical connection diagram is provided in Figure 1-20.

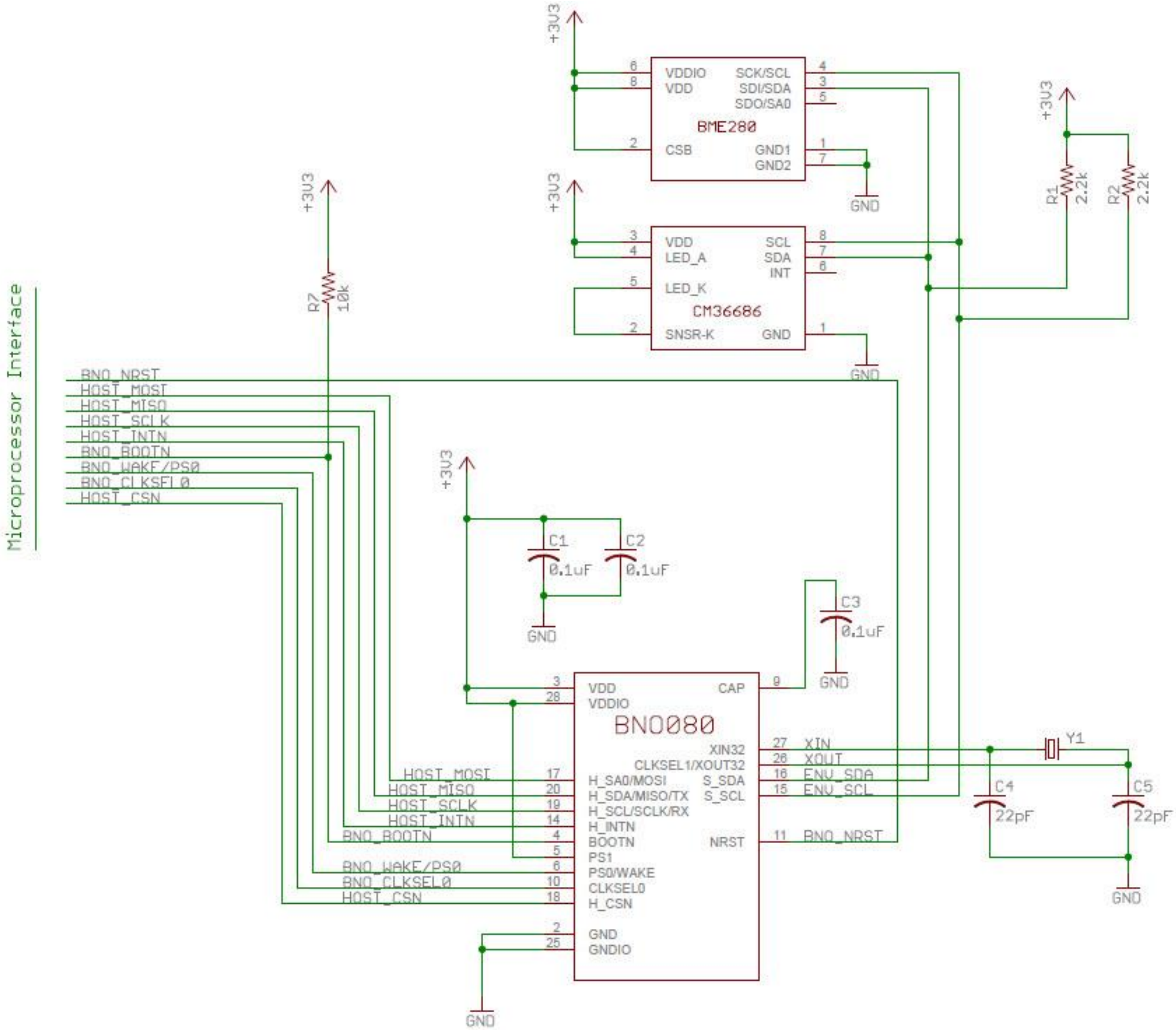


Figure 1-20: BNO080 SPI connection diagram

Figure 1-20 shows how the BNO080 can be connected to an external microcontroller via a SPI interface. The following notes are provided as guidelines for connecting the BNO080 in a system design.

1. The H_INTN pin is the application interrupt line that indicates the BNO080 requires attention. This should be tied to a GPIO with wake capability. The interrupt is active low.
2. NRST is the reset line for the BNO080 and can be either driven by the application processor or the board reset.
3. BOOTN is sampled at reset. If low the BNO080 will enter bootloader mode.
4. Pin 4 (BOOTN) should be pulled high through a 10K Ohms resistor. To use the device firmware update (DFU) capability of the BNO080, it is recommended to connect Pin 4 to a GPIO pin on the external microcontroller.
5. Pin 5 (PS1) and Pin 6 (PS0/WAKE) are the host interface protocol selection pins. Both pins must be high from before reset until after the first assertion of H_INTN to select the SPI interface. Pin 5 may be tied to VDDIO. Pin 6 must be connected to a GPIO so that the WAKE functionality can be performed.

6. After reset the PS0/WAKE signal is used as a 'wake' signal taking the BNO080 out of sleep if the host wants to initiate communication with the BNO080.
7. The BNO080 supports environmental sensors (e.g. pressure sensors, ambient light sensors) on a secondary I²C interface. This interface should be pulled up via resistors regardless of the presence of the external sensor as the SW polls for sensors at reset.

1.3.4.1 SPI Operation

SPI is a 4-wire synchronous serial protocol. SPI provides a full duplex communication path and has a master/slave relationship. The master provides the clock for all transactions. Multiple slave devices can exist on a SPI interface by the use of a chip select signal. The BNO080 is the slave in all communications.

SPI allows for two clock polarities and clock edge sampling. These options are typically called CPOL and CPHA. The BNO080 uses CPOL = 1 and CPHA = 1. In this configuration the clock idles high and data is captured on the rising edge of the clock:

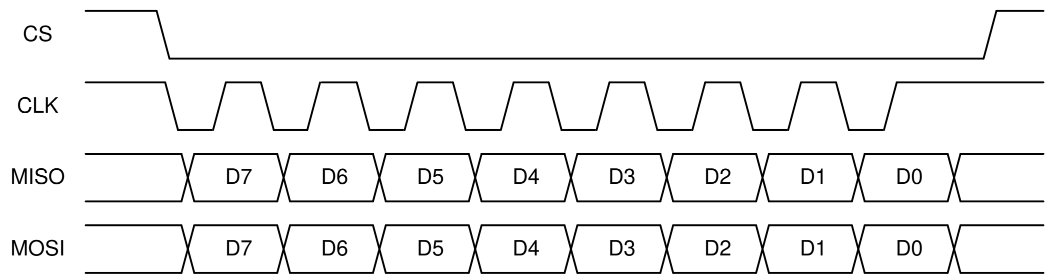


Figure 1-21: BNO080 SPI signaling

SPI transmits data MSB first and is byte oriented (i.e. all data is passed in 8-bit segments). Any number of bytes can be transferred in a single transaction (chip select assertion).

MISO is the data transferred from the slave to the master and MOSI from master to slave.

The BNO080 uses Hillcrest's SHTP protocol to communicate. More details are available at [2].

1.3.4.2 Wake operation

When the host want to initiate contact with the BNO080 it may be necessary to wake the processor from a sleep mode. To enable this function the BNO080 uses the PS0/wake pin. The pin is active low and should be driven low to initiate the wake procedure. The BNO080 will respond by asserting the interrupt line at which point the host can initiate SPI accesses. The BNO080 will de-assert the interrupt line as soon as the chip select is detected.

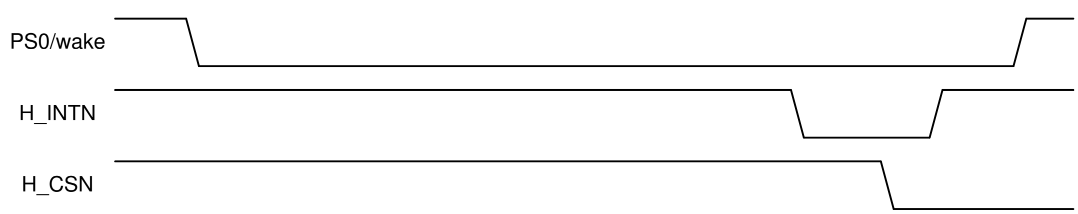


Figure 1-22: SPI Wake operation

1.3.5 UART-RVC interface

The BNO080 provides a simplified UART interface for use on robot vacuum cleaners (RVC). When configured in this mode the BNO080 simply transmits heading and sensor information at 100Hz over the UART TX pin. A typical connection is shown in Figure 1-23

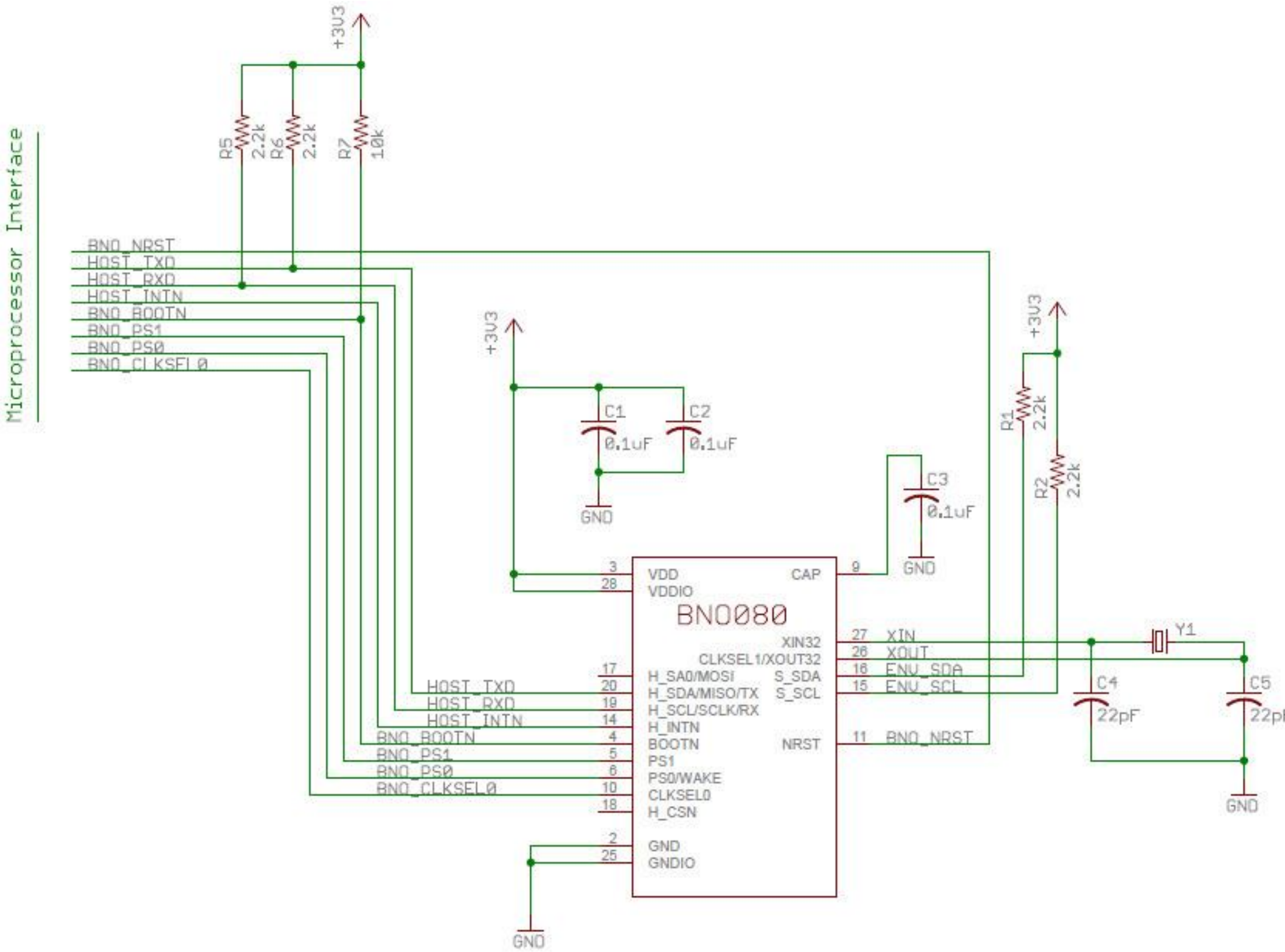


Figure 1-23: BNO080 UART-RVC connection diagram

Figure 1-23 shows how the BNO080 can be connected to an external microcontroller via a UART interface. The following notes are provided as guidelines for connecting the BNO080 in a system design.

1. NRST is the reset line for the BNO080 and can be either driven by the application processor or the board reset.
2. BOOTN is sampled at reset. If low the BNO080 will enter bootloader mode.
3. Pin 4 (BOOTN) should be pulled high through a 10K Ohms resistor. To use the device firmware update (DFU) capability of the BNO080, it is recommended to connect Pin 4 to a GPIO pin on the external microcontroller.
4. Pin 5 (PS1) and Pin 6 (PS0) are the host interface protocol selection pins. These pins should be tied to ground and VDDIO respectively to select the UART-RVC interface.
5. The BNO080 supports environmental sensors (e.g. pressure sensors, ambient light sensors) on a secondary I^2C interface. This interface should be pulled up via resistors regardless of the presence of the external sensor as the SW polls for sensors at reset.

1.3.5.1 UART-RVC operation

The UART operates at 115200 b/s, 8 data bits, 1 stop bit and no parity. The UART protocol relies on an idle line being ‘high’. A transmission is started with the assertion of a start bit (pulling the line low), followed by the data, LSB first. After the data segment is sent (in this case 8-bits), the line is pulled high (the stop signal) for a minimum number of bits (1 for the BNO080) to indicate end of that segment.

Figure 1-24: UART signaling

1.3.5.2 UART-RVC protocol

The BNO080 transmits the following data at a rate of 100Hz.

Header	Index	Yaw		Pitch		Roll		X-axis accel		Y-axis accel		Z-axis accel		Reserved			Csum
0xAAAA		LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB	LSB	MSB	0	0	0	

Figure 1-25: BNO080 UART-RVC packet format

The 19-byte message has the following fields:

Header: Each report is prefixed with a 0xAAAA header
Index: A monotonically increasing 8-bit count is provided (0-255) per report
Yaw: The yaw is a measure of the rotation around the Z-axis since reset. The yaw has a range of +/- 180˚ and is provided in 0.01˚ increments, i.e. a report of 8734 is equivalent to 87.34˚.
Pitch: The pitch is a measure of the rotation around the Y-axis. The pitch has a range of +/- 90˚ and is provided in 0.01˚ increments, i.e. a report of 1072 is equivalent to 10.72˚.
Roll: The roll is a measure of the rotation around the X-axis. The roll has a range of +/- 180˚ and is provided in 0.01˚ increments, i.e. a report of 1072 is equivalent to 10.72˚.
X-axis acceleration: The acceleration along the X-axis, presented in mg
Y-axis acceleration: The acceleration along the Y-axis, presented in mg
Z-axis acceleration: The acceleration along the Z-axis, presented in mg
Reserved: The message is terminated with three reserved bytes, currently set to zero
Checksum (Csum): The Index, yaw, pitch, roll, acceleration and reserved data bytes are added to produce the checksum.

To determine the actual orientation of the module, the rotations should be applied in the order yaw, pitch then roll.

An example complete message and checksum calculation is as follows:

Message: 0xAA AA DE 01 00 92 FF 25 08 8D FE EC FF D1 03 00 00 00 E7

Where:

Index = 0xDE = 222
Yaw = 00.01˚ (1 = 0x0001)
Pitch = -1.10˚ (-110 = 0xFF92)
Roll = 20.85˚ (2085 = 0x0825)
X-acceleration = -371 mg = -3.638 m/s^2 (-371 = 0xFE8D)
Y-acceleration = -20 mg = -0.196 m/s^2 (-20 = 0xFFEC)
Z-acceleration = 977 mg = -9.581 m/s^2 (977 = 0x03D1)
Checksum = 0xE7

1.4 Host Communication

1.4.1 SHTP

The BNO080 uses Hillcrest's SHTP (Sensor Hub Transport Protocol) to communicate for all interface styles except UART-RVC. SHTP provides a means of passing data between the BNO080 and a host with support for multiple channels. The BNO080 does not currently support the inclusion of 3rd party applications, but the SHTP protocol allows for separation of traffic via these channels such that applications on a host can communicate over this channel.

All data is prefixed with a 4-byte header:

Byte	Field
0	Length LSB
1	Length MSB
2	Channel
3	SeqNum

Figure 1-26: SHTP Header

Length	Bit 15 of the length field is used to indicate if a transfer is a continuation of a previous transfer. Bits 14:0 are used to indicate the total number of bytes in the cargo plus header, which may be spread over multiple messages. The bytes in the header field are counted as part of the length. A length of 65535 (0xFFFF) is reserved because a failed peripheral can too easily produce 0xFFFF. Therefore, the largest cargo that can be transported is 32766 minus the header bytes. The BNO080 does not support receiving fragmented messages but it does support sending them.
Channel	The channel number of the cargo. Channel 0 is the command channel and is used by the SHTP.
SeqNum	The sequence number of the cargo. The sequence number is a monotonically incrementing number that increments once for each cargo sent or cargo continuation sent. Each channel and each direction has its own sequence number. The sequence number is used to detect duplicate or missing cargoes and to associate segmented cargoes with each other.

The length field allows a host to schedule the correct number of clocks to generate for reads over I^2C and SPI. The BNO080 supports partial reads if the host cannot accept all the bytes in one read, the length will be updated on a subsequent read. So, for instance a host could read the first 4 bytes to determine the number of clocks to generate and then repeat the read with the required number of clocks. The protocol is fully described in [2].

The BNO080 supports 6 channels:

- Channel 0: the SHTP command channel
- Channel 1: executable
- Channel 2: sensor hub control channel
- Channel 3: input sensor reports (non-wake, not gyroRV)
- Channel 4: wake input sensor reports (for sensors configured as wake up sensors)
- Channel 5: gyro rotation vector

The SHTP control channel provides information about the applications built into the BNO080 firmware image. The BNO080 uses advertisements to publish the channel maps and the names of the built-in applications.

The executable channel allows the host to reset the BNO080 and provide details of its operating mode. Figure 1-27 provides details.
:

SHTP Channel	Use	Direction
1 (executable)	0 – reserved 1 – reset 2 – on 3 – sleep 4-255 – reserved	Write
	0 – reserved 1 – reset complete 2-255 – reserved	Read

Figure 1-27: SHTP executable commands and response

The use of the 'on' and 'sleep' commands provides support when the host is entering and exiting sleep. When the 'sleep' command is issued all sensors that are configured as always on or wake (see 1.4.5.1) will continue to operate all, other sensors will be disabled. All configured sensors will be enabled when the host issues the 'on' command.

The sensor hub control channel is used to configure the BNO080. Responses from the BNO080 in reaction to configuration requests are also sent over this channel.

The input sensor report channel is unidirectional, passing data from the BNO080 to the host. All input reports pass over this channel except for sensors that are configured as wake sensors and the gyro rotation vector. A wake sensor is defined in Android as a sensor that will remain awake during periods when a mobile device is asleep and can wake the processor when a particular event occurs (the event being defined by the sensor).

The wake input sensor report channel is used for sensors that are configured as wake sensors.

The gyro rotation vector is a channel used for the gyro rotation vector. A separate channel is provided to allow applications on a host processor to prioritize this data over others to ensure low latency.

1.4.2 Report Structure

The BNO080 runs Hillcrest's SH2 protocol. Full details are available in [1].
All commands, responses, sensor input reports are prefixed with a report number. The report number follows the SHTP header.
For example, the command to read the Product ID is:

Byte	Description
0	Report ID = 0xF9
1	Reserved

Figure 1-28: Product ID request

The BNO080 will respond to the Product ID request with the following response:

Byte	Description
0	Report ID = 0xF8
1	Reset Cause
2	SW Version Major
3	SW Version Minor
4	SW Part Number LSB
5	SW Part Number …
6	SW Part Number …
7	SW Part Number MSB
8	SW Build Number LSB
9	SW Build Number …

Byte	Description
10	SW Build Number …
11	SW Build Number MSB
12	SW Version Patch LSB
13	SW Version Patch MSB
14	Reserved
15	Reserved

Figure 1-29: Product ID Response

The list of currently supported commands/configurations is:

SHTP Channel	Direction	Report ID	Description
2 (SH-2 control)	Host to BNO	0xFE	Get Feature Request
2 (SH-2 control)	Host to BNO	0xFD	Set Feature Command
2 (SH-2 control)	BNO to host	0xFC	Get Feature Response
2 (SH-2 control)	Host to BNO	0xF9	Product ID Request
2 (SH-2 control)	BNO to host	0xF8	Product ID Response
2 (SH-2 control)	Host to BNO	0xF7	FRS Write Request
2 (SH-2 control)	Host to BNO	0xF6	FRS Write Data
2 (SH-2 control)	BNO to Host	0xF5	FRS Write Response
2 (SH-2 control)	Host to BNO	0xF4	FRS Read Request
2 (SH-2 control)	BNO to host	0xF3	FRS Read Response
2 (SH-2 control)	Host to BNO	0xF2	Command Request
2 (SH-2 control)	BNO to host	0xF1	Command Response

Figure 1-30: BNO080 Commands

1.4.3 BNO080 Configuration

Motion analysis systems must process data from sensors that can be mounted in an arbitrary manner and in systems with characteristics that affect the way the data is delivered. For example a tap detector could behave differently in a small form factor device made of metal than a tap detector in a larger form factor device made of plastic. Another example is the static calibration record. This record provides a description of the sensor and its orientation; necessary details to provide calibration of the sensor data.

The BNO080 contains a Flash Record System (FRS) to store these configurations.

Complete documentation of these records is provided in [1]. Supported FRS records in the BNO080 are:

Record ID	Description
0x7979	Static calibration – AGM
0x4D4D	Nominal calibration – AGM
0x8A8A	Static calibration – SRA
0x4E4E	Nominal calibration - SRA
0x1F1F	Dynamic calibration
0xD3E2	MotionEngine power management
0x2D3E	System orientation
0x2D41	Primary accelerometer orientation
0x2D46	Gyroscope orientation
0x2D4C	Magnetometer orientation
0x3E2D	AR/VR stabilization – rotation vector
0x3E2E	AR/VR stabilization – game rotation vector
0xC274	Significant Motion detector configuration
0x7D7D	Shake detector configuration

Record ID	Description
0xD7D7	Maximum fusion period
0x4B4B	Serial number
0x39AF	Environmental sensor - Pressure calibration
0x4D20	Environmental sensor - Temperature calibration
0x1AC9	Environmental sensor - Humidity calibration
0x39B1	Environmental sensor - Ambient light calibration
0x4DA2	Environmental sensor - Proximity calibration
0xD401	ALS Calibration
0xD402	Proximity Sensor Calibration
0xED85	Stability detector configuration
0x74B4	User record
0xD403	MotionEngine Time Source Selection
0xA1A2	Gyro-Integrated Rotation Vector configuration

Figure 1-31: FRS records

FRS records are written by use of FRS read and write requests. Writing and reading FRS records follows a multi-step protocol.

For writes an FRS Write Request is issued indicating which FRS record to write and the amount of data to be written. The BNO080 will respond with a write response acknowledging the request. The host will then issue a number of FRS Write Data packets to which the BNO080 will respond. The final response from the BNO080 will indicate a successful conclusion to the flash writes or if the process failed a failure indication.

An FRS read follows a similar pattern. An FRS Read request is issued and FRS Read responses provide the data with the final response indicating completion of the record retrieval.

1.4.4 Sensor Metadata

The BNO080 operates as a sensor hub and as such provides processing of sensor data. In order for a user to understand the capabilities of the sensors, metadata is provided that describes those capabilities. The SH-2 reference manual provides full details of the metadata record but as an example Figure 1-32 provides the metadata for the rotation vector. Metadata is stored as an FRS record and is therefore retrieved as an FRS read.

Word	Description			
	MSB			LSB
0	Version = 0x00010001			
1	Range = 1.0			
2	Resolution = 6.103515625e-05			
3	Revision = 4		Power = 5.3017578125	
4	Minimum period = 1000			
5	FIFO reserved count = 1		FIFO max count = 0	
6	Vendor ID length = 0		Batch buffer bytes = 14	
7	Q point 2 = 12		Q point 1 = 14	
8	Q point 3 = 13		Sensor-Specific Metadata Length = 0	
9	Maximum period = 10000			
10	Sensor-Specific metadata = 0			
N	Vendor ID			

Figure 1-32: BNO080 rotation vector metadata

Version	Identifies the physical sensor/driver/fusion versions for a given sensor. The elements within this field are updated when a component changes in a manner that affects the output of the sensor.

