Device Operational Overview HMC6352

HMC6352 has two parameters; *Operational Mode* and *Output Mode*, which control its operation.
The Operational Mode is a RAM byte (0x74) and is shadowed in EEPROM location 0x08. This byte can be used to control the Measurement rate, Set/reset function, and to command the device into the three allowed operating modes; Standby, Query, and Continuous. The current *Op Mode* RAM value can be saved in the EEPROM using the "L" command, and will become the default mode on subsequent power up. Also, HMC6352 can be put in to Sleep mode for the lowest power consumption.

The Output Mode Byte is located in RAM 0x4E and is not shadowed in the EEPROM, and upon power up the device is in the Heading output mode. This byte can be changed to get magnetometer data if necessary.

The application environment of the HMC6352 will dictate the most suitable operational mode.
In the Standby Mode the HMC6352 is not performing measurements and is waiting for a command, and can be commanded in to making a heading measurement by issuing the "A" command. This mode is useful to get data on demand or at random intervals as long as the application can withstand the time delay in getting the data.

With the Query Mode, the HMC6352 will make a fresh measurement after it is read by the host processor. In this mode the data are available for immediate read.

The above two modes are the most power efficient readout modes.

In the Continuous Mode the user can choose 1,5,10,or 20 Hz output rate and the HMC6352 will make continuous measurements and update the output registers. This mode is useful for data demanding applications. In this mode the output can be read by writing 0x43 to the HMC6352 I2C bus.

I^{2}C Bus Overview

HMC6352 employs the 2-wire I^{2}C bus protocol (http://www.semiconductors.philips.com/acrobat/literature/9398/39340011.pdf) in the 100 Kb/s data rate, 7 bit addressing mode.

There is a clock line (SCL) and a data SDA line in this bus specification and a host of devices can be connected. The bus can be a single Master – multi Slave or it can be a Multi-Master configuration. All data transfers are initiated by the Master device which is responsible for generating the clock signal, and the transfers are 8 bit long. All devices are addressed by its unique 7 bit **Address**. After each 8-bit transfer, the Master generates a 9 th clock pulse, and the transmitting device releases the SDA line. The receiving device will pull the SDA line low to acknowledge (**ACK**) the successful transfer or leave the SDA high to **NACK**.

All transitions in the SDA line must occur when SCL is low. This requirement leads to two unique conditions on the bus associated with the SDA transitions when SCL is high. Master device pulling the SDA low while SCL high is the **Start (S)** condition, and the **Stop(P)** condition when the SDA is pulled high while SCL is high. The I^{2}C protocol also allows for the **Restart** condition in which the master device issues a second Start condition without issuing a Stop.

All bus transactions begin with the Master issuing the Start sequence followed by the slave **address-byte**. The address-byte contains the slave address; the upper 7 bits (bits7-1), and the LSb. The LSb of the address-byte designates if the operation is read (LSb=1) or write (LSb=0). At the 9 th clock pulse, the transmitting device will issue the ACK (or NACK). Following these bus events, the master will send data bytes for a write operation, and the slave will transmit data for a read operation. All bus transactions are terminated with the Master issuing a Stop sequence.

I^{2}C Implementation

I2C bus can be implemented with either a hardware module or in software. Typical hardware modules will release the SDA and SCL lines as appropriate to allow the slave device to manipulate these lines. In software implementation care must be taken to perform these tasks in software.

HMC6352 Interface Commands (Table 1)

Command	Argument1	Argument2	Response1	Response2	Description
(0x77) w	EEPROM Address	Data			Write to EEPROM.
(0x72) r	EEPROM Address		Data		Read from EEPROM.
(0x47) G	RAM Address	Data			Write to Register.
(0x67) g	RAM Address		Data		Read from Register.
(0x53) S					Sleep.
(0x57) W					Wake Up.
(0x4F) O					Update the Bridge Offset.
(0x43) C					Enter the User Calibration Mode.
(0x45) E					Exit the User Calibration Mode.
(0x4C) L					Save the current MODE into EEPROM
(0x41) A			MSByte	LSByte	Get Data. Compensate and Calculate Heading

EEPROM Content

EE Address (hex)	Byte Description	Factory Default
00	I2C Slave Address	0x42
01	Magnetometer X Offset MSB	**
02	Magnetometer X Offset LSB	**
03	Magnetometer Y Offset MSB	**
04	Magnetometer Y Offset LSB	**
05	Time Delay (0 – 255 ms)	0x01
06	Number of Summed measurements(1-16)	0x04
07	Software Version Number	> 0x01
08	Operation Mode Byte	0x50

Timing Requirements

Below are the time delays required by HMC6352 upon receipt of the command to either perform the commanded task or to have the response available on the I2C bus

Command	Description	Time Delay
(0x77) w	Write to EEPROM.	70 uS
(0x72) r	Read from EEPROM.	70 uS
(0x47) G	Write to Register.	70 uS
(0x67) g	Read from Register.	70 uS
(0x53) S	Sleep.	10 uS
(0x57) W	Wake Up.	100 uS
(0x4F) O	Update the Bridge Offset.	6 mS
(0x43) C	Enter the User Calibration Mode.	10 uS
(0x45) E	Exit the User Calibration Mode.	14 mS
(0x4C) L	Save the current MODE into EEPROM	125 uS
(0x41) A	Get Data. Compensate and Calculate Heading	6 mS

Command and Operation Mode Interactions

All commands are acceptable in the Standby Mode. Honeywell strongly recommends using this mode during initial setup stage. Setting up of the HMC6352 operation mode and its slave address are set up examples. Although execution of all commands in the Query and Continuous Modes is acceptable, the outcome is not guaranteed.

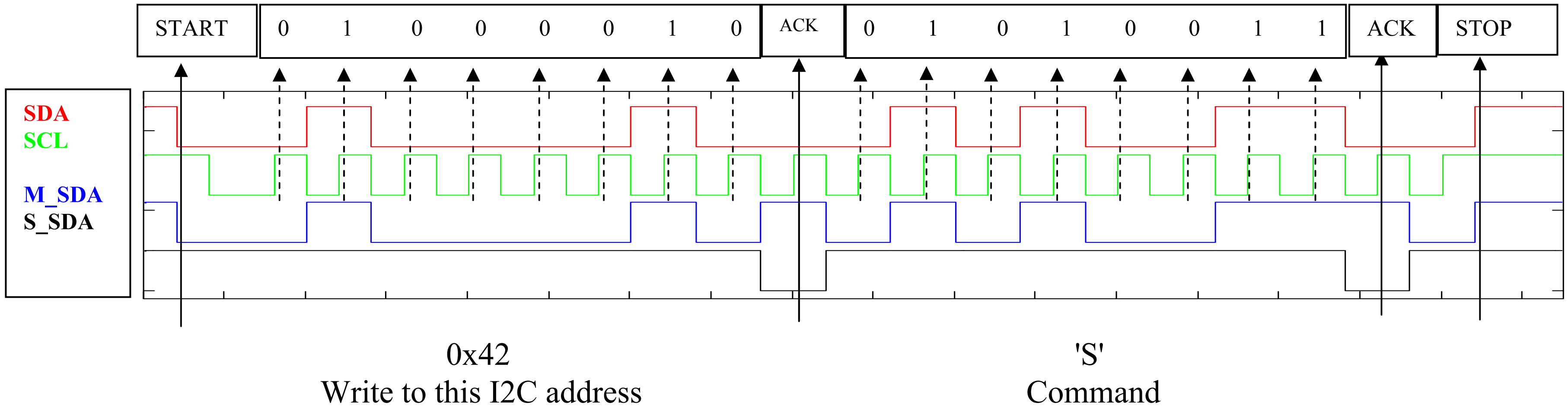
How to Read Data from HMC6352

1) In Standby Mode
   Use “A” command
2) In Query Mode
   Send 0x43 and clock out data (See Example 5)
3) In Continuous Mode
   Send 0x43 and clock out data (See Example 5)
   A is not allowed

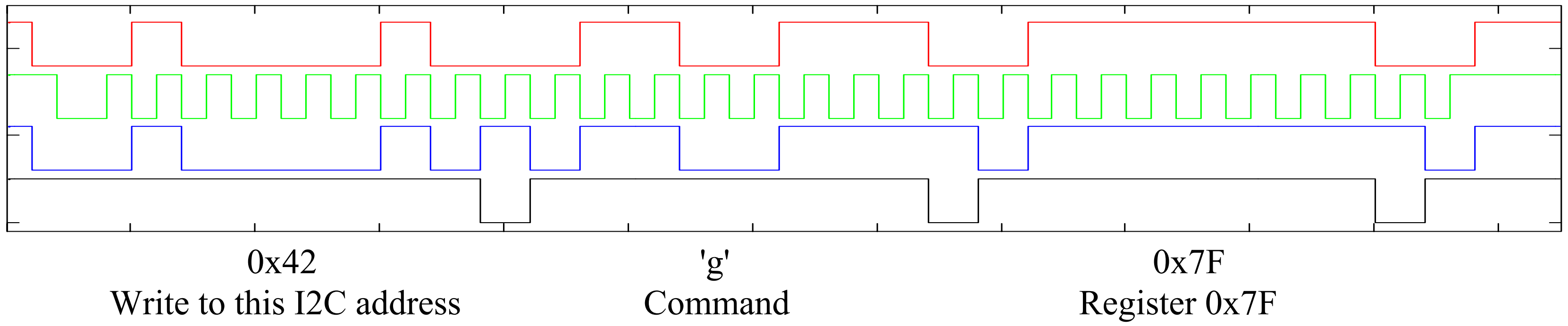
Waveform Examples

Red: This is what actually happens on the SDA line.
Green: This is what actually happens on the SCL line.
Blue: This is what the Master tries to make happen on the SDA line.
Black: This is what the senso (Slave) tries to make happen on the SDA line.

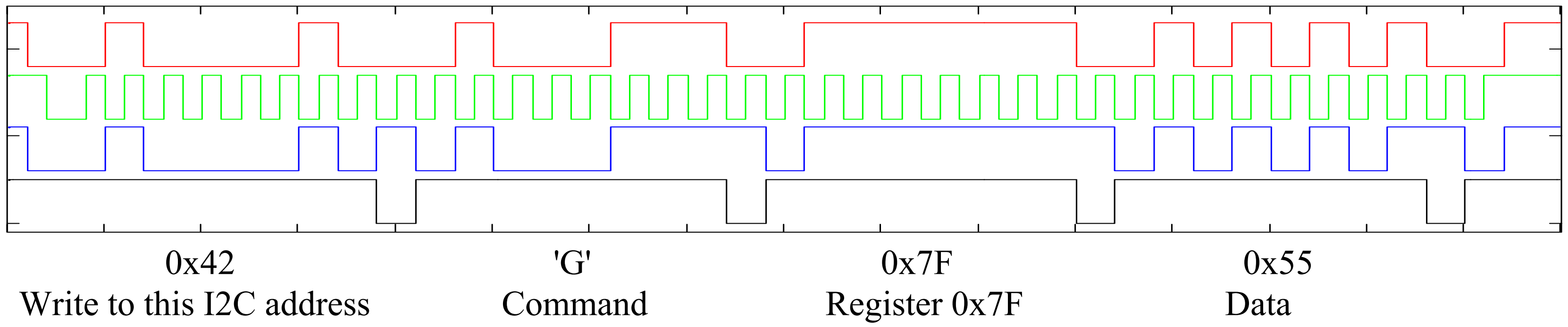
Example 1: This example shows how to command the HMC6352 in to Sleep mode by writing the 'S' command to the slave.



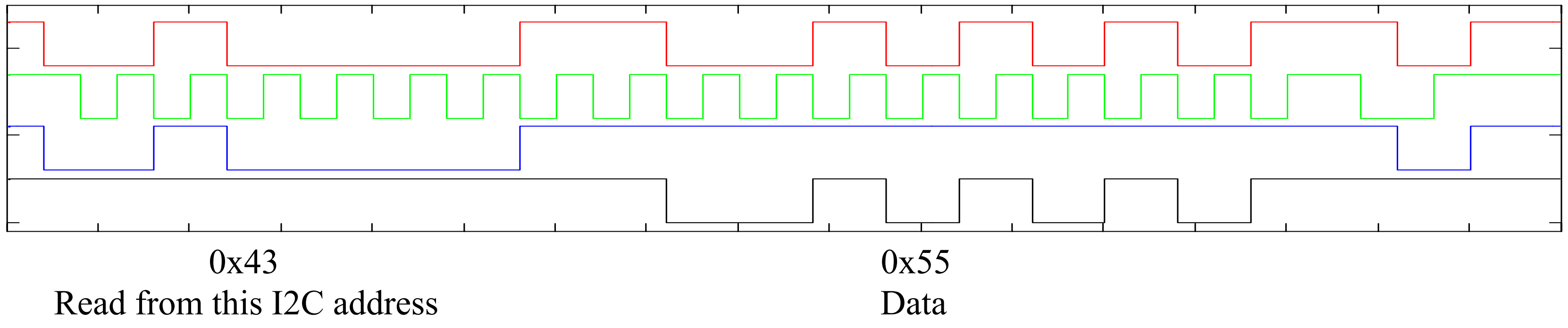
Example 2: This example shows how to command HMC6352 to read a RAM register by sending the 'g' command and the register address (0x7F). Note that this example does not show the process of reading the answer. See below for reading.



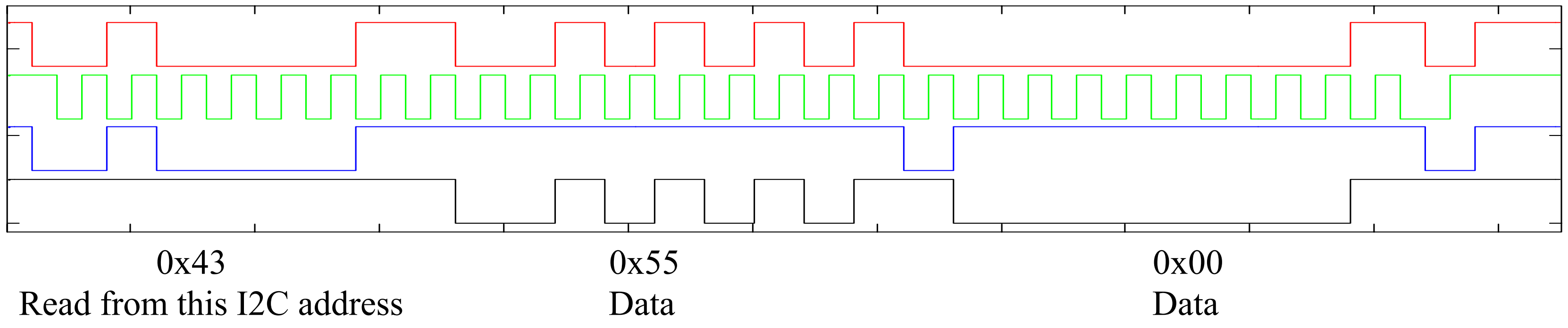
Example 3: This example shows how to write to a RAM register in the HMC6352 by sending the 'G' command, the register address (0x7F), and the data byte (0x55) to the sensor.



Example 4: This example shows how to read a single byte from the HMC6352. The Slave(HMC6352) continues to hold the SDA line low after the acknowledge (ACK) because the first bit of the data byte is a zero.

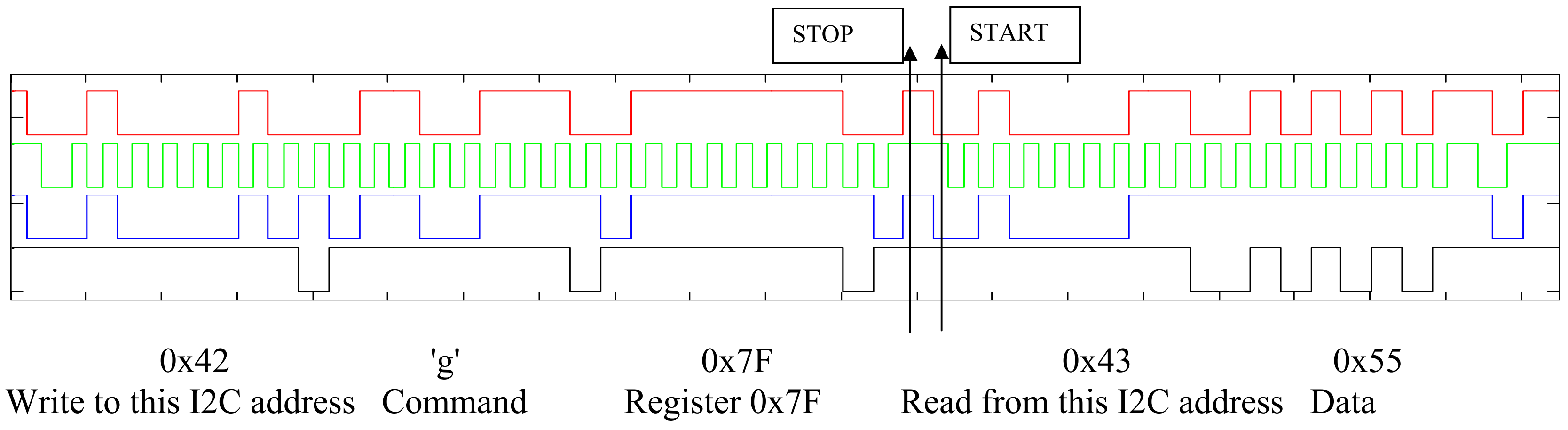


Example 5: This example shows how to read two bytes from HMC6352 (slave). The slave continues to hold the SDA line low after the acknowledge because the first bit of the data byte is a zero.



Honeywell

Example 6: The final example shows how to read RAM register 0x7F. First perform a write operation to command the HMC6352 to read a RAM register and define which register to read (Example 2). The sensor puts the answer in the data buffer. Then perform a read operation to clock out the answer (Example 4). There is a Stop / Start event in between the write operation and the read operation. This example is just a combination of Examples 2 and 4, but it is provided to show that reading a register involves both a write and a read operation.


STOP
START
0x42
Write to this I2C address
'g'
Command
0x7F
Register 0x7F
0x43
Read from this I2C address
0x55
Data