METROlab

Magnetic precision has a name

Hall Magnetic Camera HMC9076

User's Manual

Version 1.0

(Revision 1.0)

February 2017

REVISION HISTORY

v. 1.0 r. 1.0	February 2017	First release

CONTENTS

GETTING STARTED
1-Introduction

The Hall Magnetic Camera HMC9076 is a platform for building semi-custom magnetic-field mapping systems.

The key system features are:

- Multiple Hall sensors provide high-precision, simultaneous 3-axis field measurements at multiple points.
- USB provides both system power and data communication.
- An optional thermal sensor allows for temperature compensation or regulation.
- An optional low-noise voltage regulator provides optimal measurement stability.
- Firmware and an associated software API provide straightforward and flexible software control.

The platform consists of:

- Hardware:
  - Magnetic field sensors: MagVector™ MV2 [1]
  - Microprocessor: Silicon Labs C8051F380 [2]
  - Voltage regulator: Texas Instruments REG102 [3]
  - Temperature sensor: Texas Instruments LM71CIMF [4]
- Firmware:
  - Write and read SPI data to/from selected device
  - Synchronize acquisitions
  - Loop-back test of USB communication

[1] See http://www.metrolab.com/products/magvector-mv2/.
[2] See https://www.silabs.com/Support%20Documents/TechnicalDocs/C8051F38x.pdf.
[3] See http://www.ti.com/lit/ds/sbvs024f/sbvs024f.pdf.
[4] See http://www.ti.com/lit/ds/symlink/lm71.pdf

  - Retrieve hardware/firmware versions
  - Get device information
  - Read or write arbitrary data, e.g. calibration tables, to ROM
- Software:
  - LabVIEW API to access all firmware features
  - MagVector MV2 register access
  - Programming examples

The contents of this manual are as follows:

- Chapter 1-Introduction: this chapter.
- Chapter 2-Installation Guide: how to install the system.
- Chapter 3-Software: documentation of software API and examples.
- Chapter 3-Firmware Interface: documentation of firmware interface.
- Chapter 5-Implementations: Specifications for HMC9076 implementations.

For the details of the Hall and temperature sensors, such as command and register definitions, please refer to the data sheets.

Updates to the documentation and software are posted on the Metrolab website, www.metrolab.com, and can be downloaded free of charge.

Unlike other Metrolab products, the firmware in the HMC9076 cannot be upgraded in the field.

For any other questions, please feel free to contact us at any time at contacts@metrolab.com.

GETTING STARTED
2-Installation Guide

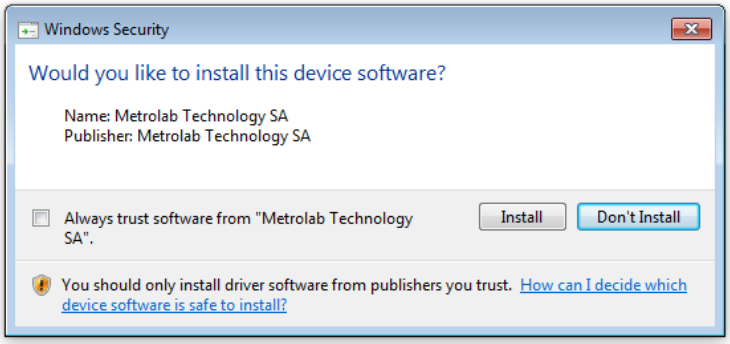
2-1 DRIVER INSTALLATION

The software driver must be installed before plugging in the hardware.

Insert the software CD and right-click on the file:

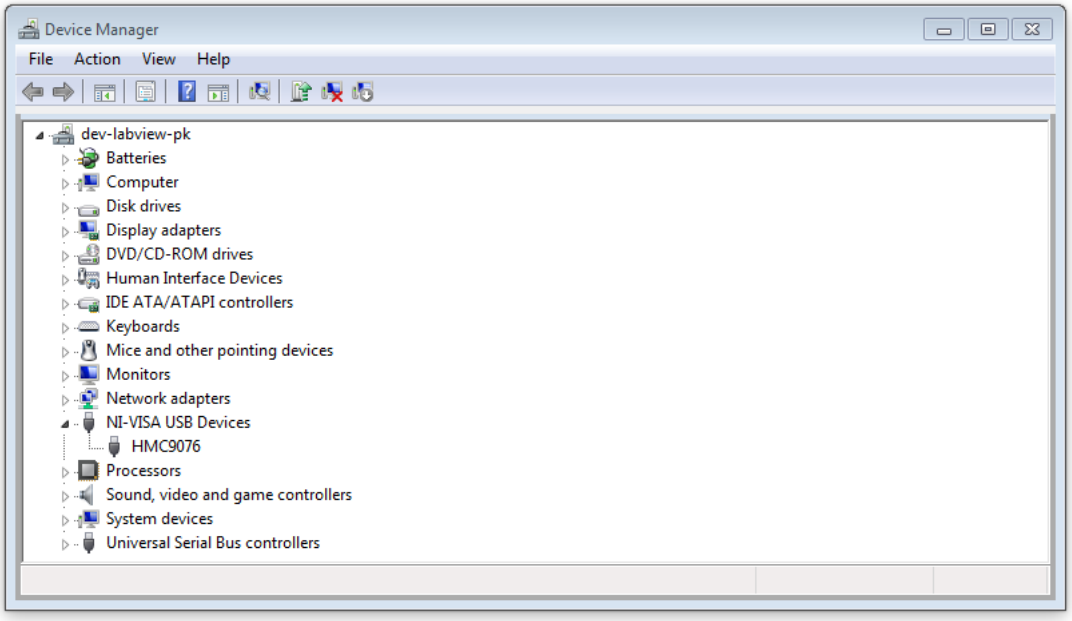
HMC9076\Inf\MTL-HMC9076.inf

From the pop-up menu, select "Install", and confirm the installation dialog:



2-2 HARDWARE INSTALLATION

Plug the HMC9076 USB connector into the computer. After device installation – which can take several minutes – the Windows Device Manager should enumerate the HMC9076 under "NI-VISA USB Devices":

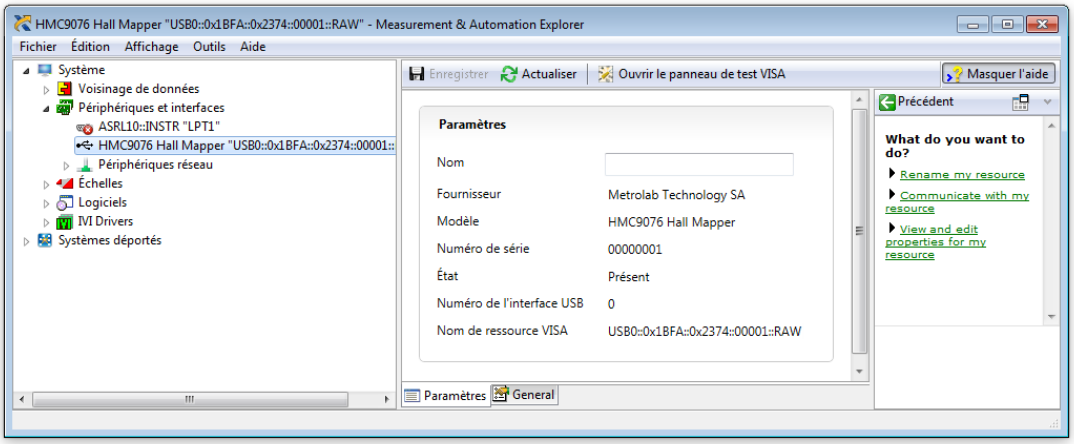


The USB Vendor ID is 0x1BFA and the Product ID is 0x2374 (0x2374 = 9076).

The National Instruments Measurement Automation Explorer (NI-MAX) should enumerate the HMC9076 as:

HMC9076 Hall Mapper "USB0::0x1BFA::0x2374::xxxxx::RAW"

where "xxxxx" is the serial number.



Make other electrical connections as appropriate for your implementation of the HMC9076 (see Chapter 5-Implementations).

2-3 SOFTWARE INSTALLATION

Copy the folder HMC9076\ to your hard disc.

Besides the "Inf" subfolder with the USB driver (see Section 2-1), this folder contains three subfolders:

- API\ : LabVIEW Application Programming Interface
- Examples\ : Sample programs
- MV2\ : MagVector MV2 register access

REFERENCE
3-Software

This chapter contains the documentation for the software supplied with the HMC9076.

This software allows users to write instrument control programs for the HMC9076, using the LabVIEW® graphical programming environment from National Instruments (www.ni.com).

The software is included in source code format (LabVIEW 2015 SP1) on the CD that came with your HMC9076 – see Section 2-3 for installation instructions. If upgrades become available, you will be able to download them from the Metrolab website, http://www.metrolab.com/downloads/.

This chapter is divided into the following sections:

- Section 3-1: LabVIEW API to access all firmware features
- Section 3-2: MagVector MV2 register access
- Section 3-3: Programming examples

3-1 APPLICATION PROGRAMMING INTERFACE

The following subsections describe the function of every Virtual Instrument (VI) in the API, and that of each terminal.

The four corner terminals are highly standardized in LabVIEW: "VISA resource name," "Duplicate VISA resource name," "error in," and "error out." These standard terminals are omitted from the documentation.

To initiate communication with the HMC9076, call VISA Open followed by SilabsUSBXpressInitialize.vi (see Section 3-1-7). To terminate, call VISA Close.

3-1-1 GetDeviceInfo.vi

Get device information.			
Max Write Length	Out	U16	Maximum write length.
Max Read Length	Out	U16	Maximum read length.
No Hall Sensors	Out	U8	Number of Hall sensors.
No Temp Sensors	Out	U8	Number of temperature sensors.
ROM size	Out	U16	Size of ROM available to user.

3-1-2 LoopBack.vi

Loop-back test of USB communication.			
In	In	U8[ ]	Arbitrary data to be sent for loop-back test.
Out	Out	U8[ ]	Data returned from loop-back test.
Equals?	Out	Boolean	Confirmation that input equals output.

3-1-3 ReadROM.vi

Read data from ROM.			
ROM Address	In	U16	Starting read address.
Length	In	U16	Read length.
Max Read Length	In	U16	Maximum read length, as returned by GetDeviceInfo.vi.
ROM	Out	U8[ ]	Returned ROM data.

3-1-4 ReadVersions.vi

Fetch hardware and firmware versions.			
HW Version	Out	U16	Hardware version.
FW Version	Out	U16	Firmware version.

3-1-5 Resetdevice.vi

Reset device.

3-1-6 SetADCInit.vi

Set/reset MagVector MV2 INIT signal. Setting INIT halts A/D conversion.			
INIT value	In	Boolean	Whether to set or reset INIT.

3-1-7 SilabsUSBXpressInitialize.vi

Performs low-level initialization of the Silabs USBXpress protocol. Call this VI immediately after the VISA Open.

3-1-8 SPIComplexTransfer.vi

Complex SPI transfer: multiple devices, multiple transfers, and with repeat count.			
SPI Cplx Transfer	In	Cluster	Transfer specification.
Repeat Count		U8	Number of times to repeat transmitting of all Frames. Note that "Repeat Count = 0" means "transmit once".
DR Timeout [ms]		U8	Data Ready wait timeout.
Frames		Cluster[ ]	Each Frame defines one transfer to one device.
Device Number		U8	Device number to transfer to/from.
Wait for DR		Boolean	Wait for Data Ready before initiating transfer?
MOSI		U8[ ][ ]	Data to be sent to the device. Each row in the array is a successive SPI transaction (= one Chip Select).
MISO	Out	U8[ ]	Data returned.
SPI Transfer Size	Out	U8	Number of bytes returned.

3-1-9 SPITransfer.vi

Single SPI transfer.			
Device Info	In	Cluster	Transfer specification.
Device Number		U8	Device number to transfer to/from.
Wait for DR		Boolean	Wait for Data Ready before initiating transfer?
DR Timeout [ms]		U8	Data Ready wait timeout.
MOSI	In	U8[ ]	Data to be sent to the device (one Chip Select).
MISO	Out	U8[ ]	Data returned.

3-1-10 WaitForDR.vi

Wait for Data Ready.			
Device Number	In	U8	Device number.
Timeout [ms]	In	U8	Wait timeout, in milliseconds.

3-1-11 WriteROM.vi

Write data to ROM.			
ROM	In	U8[ ]	ROM data to write.
Max Write Length	In	U16	Maximum write length, as returned by GetDeviceInfo.vi.

3-1-12 Utilities / Concatenate Equivalent Frames.vi

Concatenates the data returned in multiple frames by a single sensor.			
Frames	In	Cluster	All data returned by the device.
Frames		Array[ ]	An array of frames.
MISO		U8[ ]	Data returned in one frame.
MISO	Out	U8[ ]	Data of all frames, concatenated.

3-1-13 Utilities / Parse SPI Cplx Transfer MISO.vi

Parse the data returned from SPIComplexTransfer.vi, by collecting all the data returned by each device. Optionally concatenates this data to the parsed data from previous transfers.			
SPI Cplx Transfer	In	Cluster	Transfer specification, as used by SPIComplexTransfer.vi.
Repeat Count		U8	Number of times to repeat transmitting of all Frames. Note that "Repeat Count = 0" means "transmit once".
DR Timeout [ms]		U8	Data Ready wait timeout.
Frames		Cluster[ ]	Each Frame defines one transfer to one device.
Device Number		U8	Device number to transfer to/from.
Wait for DR		Boolean	Wait for Data Ready before initiating transfer?
MOSI		U8[ ][ ]	Data to be sent to the device. Each row in the array is a successive SPI transaction (= one Chip Select).
MISO	In	U8[ ]	Data returned by SPIComplexTransfer.vi.
Last Cplx Transfer MISO	In	Cluster[ ]	Data returned by previous transfers, parsed one set of frames per device.
Frames		Array[ ]	An array of frames.
MISO		U8[ ]	Data returned in one frame.
Cplx Transfer MISO	Out	Cluster[ ]	Output data, parsed one set of frames per device.
Frames		Array[ ]	An array of frames.
MISO		U8[ ]	Data returned in one frame.

3-2 MAGVECTOR™ MV2 REGISTER ACCESS

3-2-1 REG00.vi

Build command to read or write Register 00 of MagVector MV2.			
Measurement Axis	In	enum – U8	Measurement axis: - 3 axes - X axis - Y axis - Z axis
Resolution	In	enum – U8	ADC bit resolution and corresponding conversion rate: - 14 bits - 3 kHz - 15 bits - 1.5 kHz - 16 bits - 750 Hz - 16 bits - 375 Hz
Range	In	enum – U8	Measurement range: - 100 mT - 300 mT - 1 T - 3 T
Output Selection	In	enum – U8	Converted output value to be returned: - Bx - By - Bz - T (Temperature)
Access	In	enum – U8	Whether to read or write the register: - Write - Read
Request (u16)	Out	U16	Command encoded as unsigned 16-bit word
Request (u8[2])	Out	U8[ ]	Command encoded as array of unsigned 8-bit values

1-1-1 REG01.vi

Build command to read or write Register 01 of MagVector MV2.			
Large Measurement Range (x10)	In	boolean	Increase selected measurement range by 10x.
Extended Measurement Range (+30%)	In	boolean	Increase selected measurement range by 30%.
High Clock	In	boolean	Doubles the analog clock.
Invert	In	boolean	Reverse the Hall bias currents.
Low Power	In	boolean	Reduce power consumption.
Permanent Output	In	boolean	Permanently activate MISO output.
Status Position	In	enum - U8	The pin used for Data Ready signal: - DR pin - MISO/DR multiplexed
Access	In	enum – U8	Whether to read or write the register: - Write - Read
Request (u16)	Out	U16	Command encoded as unsigned 16-bit word
Request (u8[2])	Out	U8[ ]	Command encoded as array of unsigned 8-bit values

3-2-2 REG10.vi

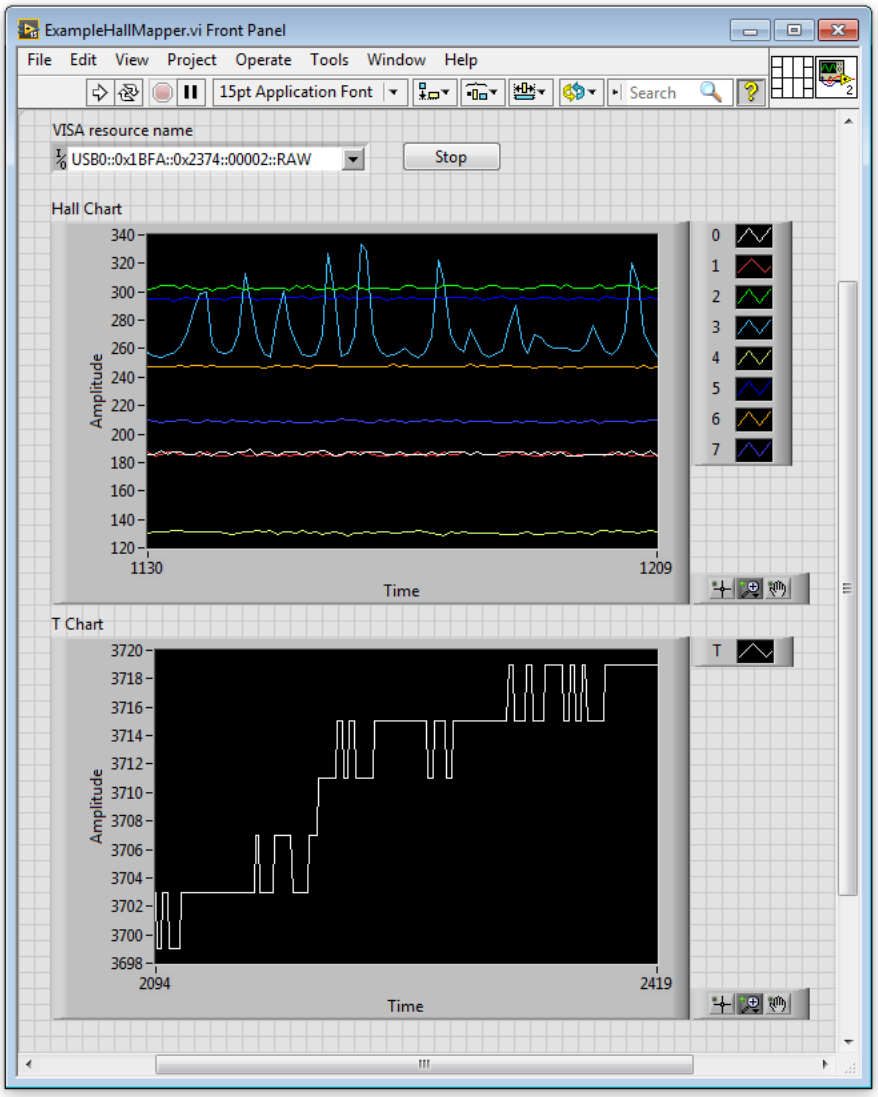
Build command to read or write Register 10 of MagVector MV2.			
Temperature Compensation	In	U8	Program temperature coefficient of Hall bias current.
Access	In	enum – U8	Whether to read or write the register: - Write - Read
Request (u16)	Out	U16	Command encoded as unsigned 16-bit word
Request (u8[2])	Out	U8[ ]	Command encoded as array of unsigned 8-bit values

3-3 SAMPLE PROGRAMS

The following sample programs are supplied on the HMC9076 product CD:

3-3-1 ExampleHallMapper.vi

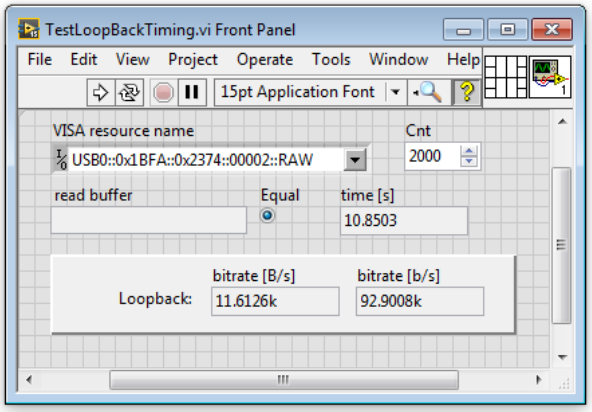
This VI measures and plots the magnetic-field magnitude for 8 Hall sensors, as well as the output of the temperature sensor.



The program acquires the data in two steps: first devices 0, 2, 4, 6, 8 (even-numbered Hall sensors and temperature sensor), and then devices 1, 3, 5, 7, 8 (odd-numbered Hall sensors and temperature sensor). Each acquisition step is executed ten times. The example illustrates how the two acquisitions can be concatenated, and how the resulting data can be separated into the individual channels.

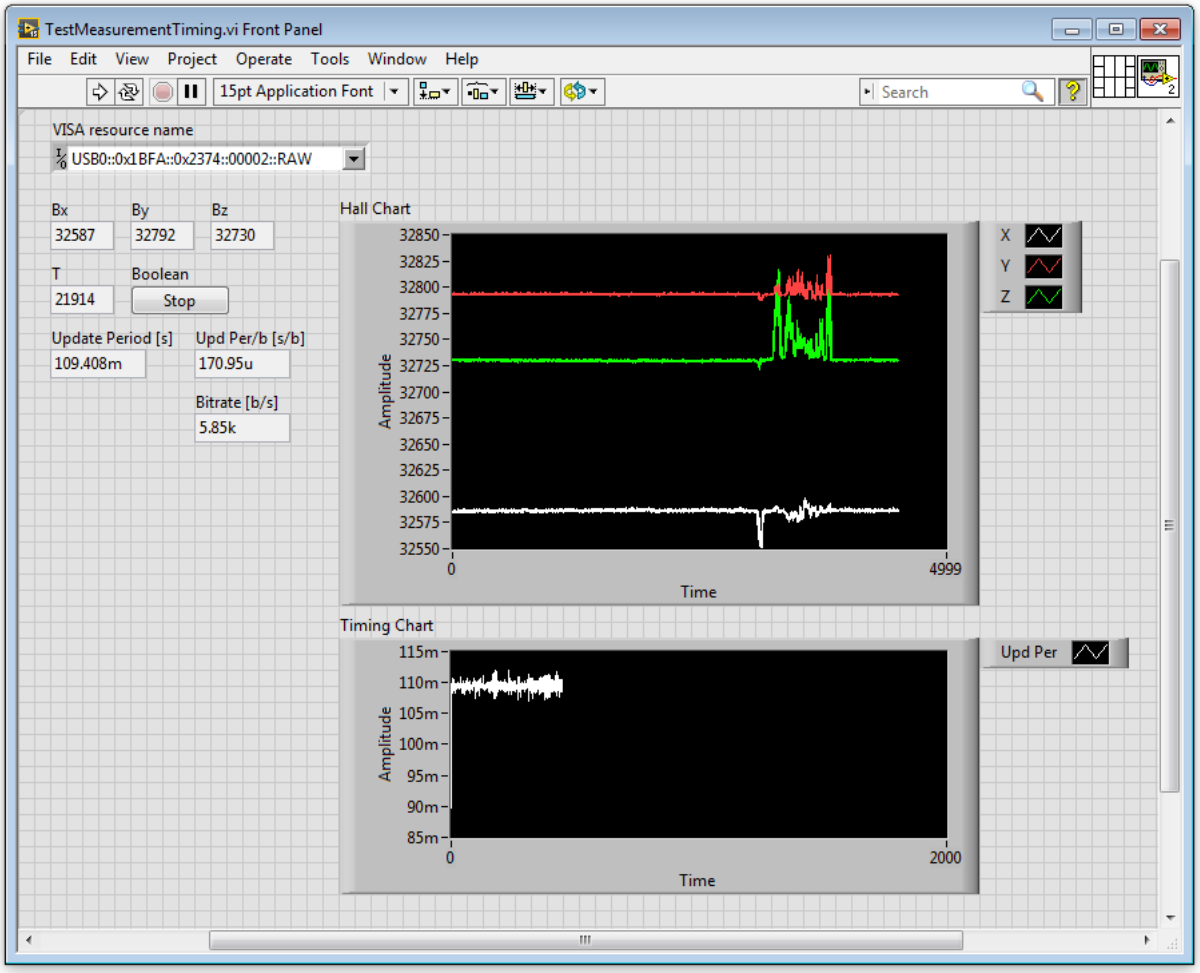
3-3-2 TestLoopBackTiming.vi

This VI uses the loop-back test to time the USB transfer rate.



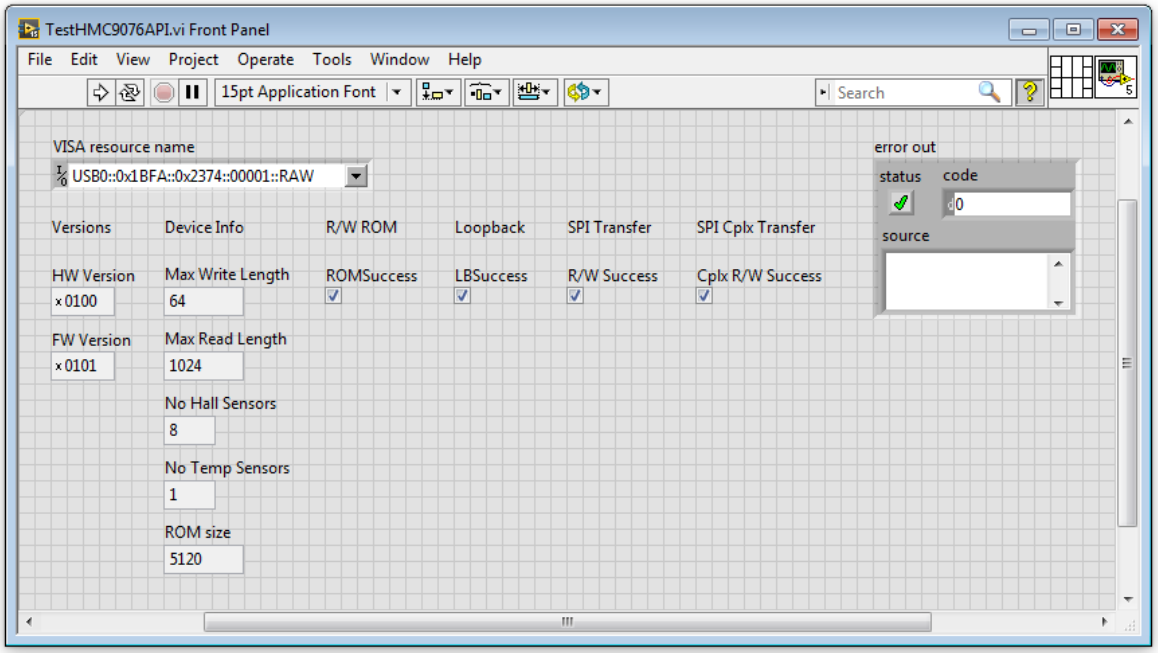
3-3-3 TestMeasurementTiming.vi

This VI performs continuous measurements on Device 0 (first Hall sensor), to measure the measurement update rate.



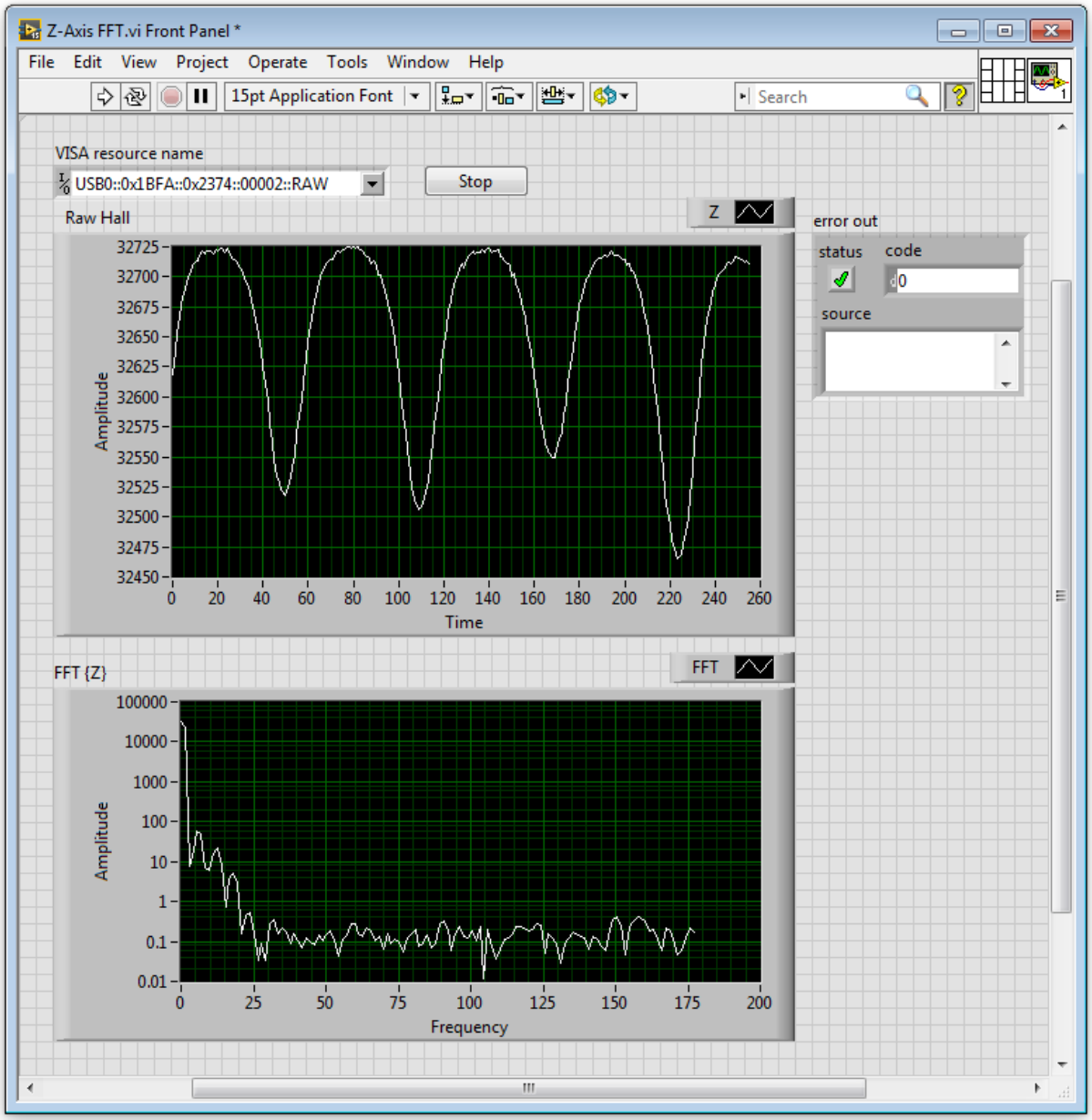
3-3-4 TestHMC9076API.vi

This VI executes all the API functions.



3-3-5 Z-Axis FFT.vi

This VI acquires blocks of 256 B_z measurements of Device 0 (first Hall sensor), and computes the FFT, to explore periodic noise (e.g. heater current).



REFERENCE
4-Firmware Interface

The HMC9076 firmware accepts commands transmitted via the Silabs USBXpress driver,[5] parses and executes these commands, and returns the results.

Each command has a one-byte command code, structured as follows:

- Bit 7: Command contains data
- Bit 6: Command returns data
- Bits 5-0: Command number

Each command may take parameters, transmitted after the command code, and may return results, returned to the host after execution.

The commands, including the definitions of the associated parameters and return values, are described in the following sections. The Error type, which may be part of the return values, is a single byte, defined as follows:

- 0: No error
- 1: Error

4-1 COMMANDS RETURNING DATA

Name	Code	Parameters	Return values
Read Versions	0x41	None	HM Hm FM Fm - HM: Hardware Major - Hm: Hardware Minor - FM: Firmware Major - Fm: Firmware Minor
Get Device Info	0x42	None	WL WL RL RL NH NT RS RS - WL: Maximum Write Length - RL: Maximum Read Length - NH: Number of Hall sensors - NT: Number of temperature sensors - RS: ROM Size

[5] See http://www.silabs.com/Support%20Documents/TechnicalDocs/an169.pdf.

4-2 COMMANDS TAKING PARAMETERS

Name	Code	Parameters	Return value
Reset Device	0x80	Must be "RESET"	None
Set ADC Init	0x81	00 or 01	None

4-3 COMMANDS TAKING PARAMETERS AND RETURNING DATA

Name	Code	Parameters	Return value
SPI Transfer	0xC1	See Section 4-4	ER MI MI MI … (n bytes) - ER: Error - MI: MISO data
SPI Complex Transfer	0xC2	See Section 4-5	ER MI MI MI … (n bytes) - ER: Error - MI: MISO data
Wait for Data Ready	0xC3	DN TO - DN: Device Number - TO: timeout in ms	ER - ER: Error
Read ROM Block	0xC4	RA RA LN LN - RA: ROM Address - LN: Length[6]	ER XX XX XX … (n bytes) - ER: Error - XX: Content of the ROM block
Write ROM Block	0xC5	See Section 4-6	ER BT BT - ER: Error - BT: Busy Time [ms][7]
Loop Back	0xFF	XX XX XX … (n bytes) - XX: Any byte	XX XX XX … (n bytes) - XX: Any byte (same as parameters)

4-4 SPI TRANSFER COMMAND

The SPI Transfer command is structured as follows:

<table>
<tr><td rowspan="3">Command code [1B]</td><td colspan="4">Command data [0-63B]</td></tr>
<tr><td colspan="3">Device Info [16b]</td><td rowspan="2">MOSI [0-61B]</td></tr>
<tr><td>Unused [2b]</td><td>Device Number [5b]</td><td>Wait For DR [1b]</td><td>DR Timeout [8b]</td></tr>
</table>

where:

- Device Info: Contains information about data to be transmitted:
  - Unused: Unused bits
  - Device Number: Indicates the number of the device to be activated for the next SPI transfer. See Chapter 5-Implementations for the device numbering of your instrument.

[6] Zero to Maximum Read Length, as given by Get Device Info (see Section 4-1).
[7] The HMC9076 may not respond to USB during this time.

- Wait For DR: Indicates whether the Data Ready must be present before each SPI transfer. Only valid for devices with a DR line (e.g. Hall sensors).
- DR Timeout: Timeout, in [ms], for the Data Ready signal, when requested.

- MOSI: Data transferred to the device.

4-5 SPI COMPLEX TRANSFER COMMAND

The SPI Complex Transfer command is structured as follows:

<table>
<tr><td rowspan="5">Command code [1B]</td><td colspan="9">Command data [0-63B]</td></tr>
<tr><td rowspan="4">Repeat Count [1B]</td><td rowspan="4">DR Timeout [1B]</td><td colspan="7">Frames [0-61B]</td></tr>
<tr><td colspan="5">Frame 1</td><td rowspan="3">…</td><td rowspan="3">Frame n</td></tr>
<tr><td colspan="4">Frame Info [16b]</td><td rowspan="2">MOSI</td></tr>
<tr><td>Device Number [5b]</td><td>Wait For DR [1b]</td><td>No Trans [5b]</td><td>Trans Size [5b]</td></tr>
</table>

where:

- Repeat Count: Allows the frame sequence to be repeated N times. When Repeat Count is zero, the sequence is executed once.
- DR Timeout: Timeout, in [ms], for the Data Ready signal, when requested.
- Frames: The command contains one or several frames. Each frame is destined for a given device, and uses a given transfer size, as specified in the Frame Info header:
  - Frame Info:
    - Device Number: Indicates the number of the device to be activated for the next SPI transfer. See Chapter 5- Implementations for the device numbering of your instrument.
    - Wait For DR: Indicates whether the Data Ready must be present before each SPI transfer. Only valid for devices with a DR line (e.g. Hall sensors).
    - No Trans: Number of SPI transactions in this frame.

- Trans Size: Number of bytes in an SPI transaction. The SPI transaction is defined by the number of bytes transmitted during the activation of the Chip Select. For SPI, sending *n* transactions of *m* bytes is different from sending *m* transactions of *n* bytes, even if the total number of bytes transmitted would be the same.

- MOSI: Contains the bytes to be transferred to the device. Its size, in bytes, is defined by the product "No Trans x Trans Size".

In SPI, each byte transmitted to a device results in a byte returned from the device. Therefore, the number of bytes returned by this commands corresponds to the total number of bytes transmitted:

$$Return\ Length = (1 + Repeat\ Count) \times \left(\sum_{0}^{No.frames} No\ Trans \times Trans\ Size \right)$$

The Return Length must not exceed the Maximum Read Length, as given by the "Get Device Info" command (see Section 4-1).

4-6 WRITE ROM BLOCK COMMAND

The Write ROM Block command allows the user to store data – such as calibration data – in the non-volatile memory (ROM) of the instrument itself. This data can be retrieved at any time. The user defines the format of the stored data.

Since the ROM pages are larger than the maximum allowable USB transaction size, this command must be called multiple times to fill the desired ROM area, and the Write operation is terminated by a special "Null Packet" command:

Command code [1B]	Command data [0-63B]
	Block Data [1-63B]
	Null Packet [0B]

The current ROM page size is 512 bytes. The USB protocol permits transmitting packets of at most 64 bytes, which, after deduction of the Command Code, leaves 63 bytes of data that can be transmitted in a single Write ROM Block command.

Each transfer is directed to a temporary RAM buffer that can hold a page of 512 bytes + a block of 63 bytes. The current ROM address starts at 0 and is auto-incremented for each Write ROM Block command. When at least 512 bytes have

been transferred to the device, a ROM page is erased and written from the RAM buffer.

The status returned by this command informs the caller how long the device will be busy with the ROM erase/write procedure; the instrument will not respond to USB during this time.
Because the total size of data which is transmitted is not necessarily a multiple of a ROM page, the "Null Packet," containing no data, is used to purge any remaining data from the RAM buffer to ROM. The current ROM address is then reset to zero.

APPENDIX 5-Implementations

This chapter describes the components, geometry, connectors and device numbering of specific implementations of the HMC9076.

5-1 SERIAL NUMBERS 1, 2 – "HFM-8-76"

5-1-1 Components

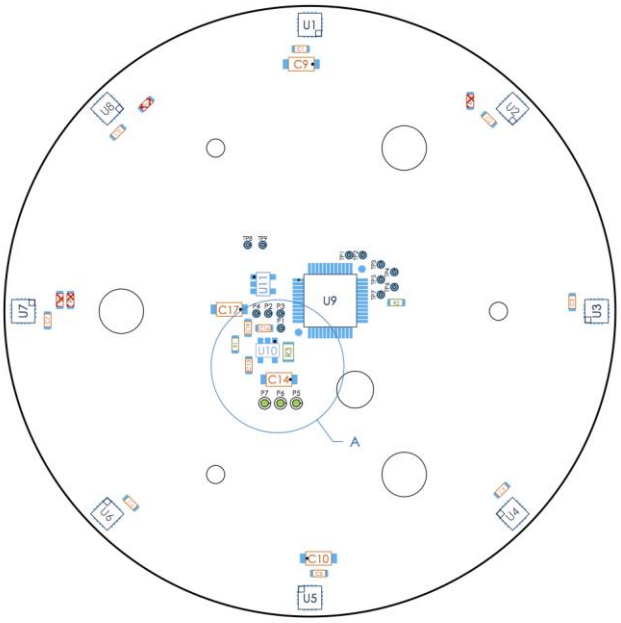
The HFM-8-76 contains the following components:

- 8 magnetic field sensors: MagVector™ MV2
- 1 microprocessor: Silicon Labs C8051F380
- 1 voltage regulator: Texas Instruments REG102
- 1 temperature sensor: Texas Instruments LM71CIMF
- Metal-Core PCB: 3 mm thick copper core
- Heater: 12 V / 36 W
- PT100 temperature sensor: glued to backside of heater [8]

5-1-2 Geometry

The 8 Hall sensors of the HFM-8-76 are spaced at 45° angles, with the field-sensitive points on a diameter of 76 mm, mounted on a disc 80 mm in diameter:

- [8] RS Pro order number 666-7362, see http://ch.rs-online.com/web/p/platin-temperatursensoren/6667362/.



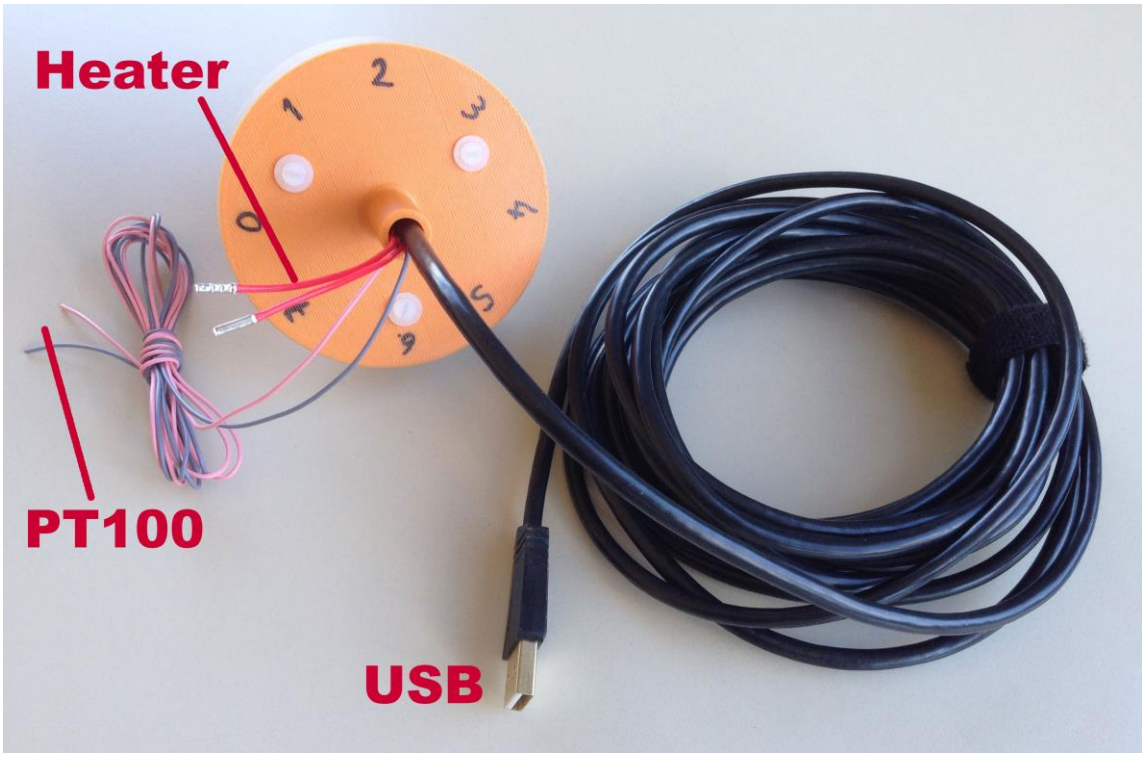
In the above diagram, Pin 1 of the Hall sensor is indicated by the small square in the corner (for example, bottom-right corner of U1). Consequently, the sensor axes are oriented as follows:

- B_x: radially outward
- B_y: clockwise around the edge of the disc
- B_z: downward

The positioning accuracy of the Hall sensors is estimated to be better than 100 µm horizontally, and better than 10 µm vertically.

5-1-3 Connectors

The electrical connections of the HFM-8-76 assembly are as follows:



5-1-4 Device numbering

By convention, the device numbers start with the Hall sensors, at zero, and are followed by the temperature sensor (if any). For the HFM-8-76, the resulting numbering of the devices for the firmware and software is as follows:

- Device Numbers 0 – 7: MagVector MV2 Hall sensors 1 – 8
- Device Number 8: Texas Instruments LM71CIMF temperature sensor