



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



Heart Rate Monitor Interface User Manual

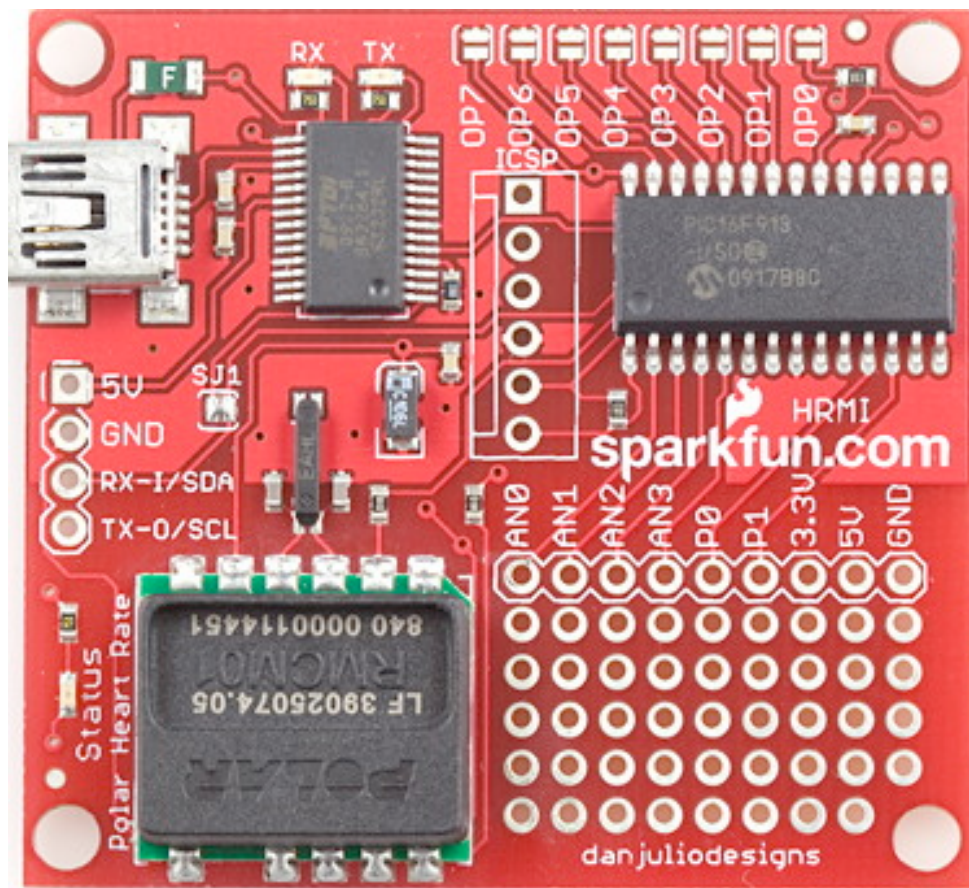


Table of Contents

Heart Rate Monitor Interface	3
Description	3
Features	3
Applications	3
Version Information	4
Disclaimer	4
Contact	4
Electrical Specifications	5
Mechanical Specifications	6
Getting Started	7
Block Diagram	7
Connections	7
Communication/Power	7
Configuration Jumpers	10
Status Indicators	11
Prototype Connections	11
Orientation with Polar Transmitter	12
Quick Test	13
Quick Test with Host Computer	13
Quick Test using HyperTerminal (Microsoft Windows)	14
Quick Test using QuickTerm (Apple Mac OS X)	18
Operation	20
Host Interface Operation	20
Serial Interface (USB or Logic-level physical interface)	20
I2C Interface	21
Heart Rate Monitor Algorithms	22
Command List	23
Read Analog Input	24
Set Utility Port Directions	25
Get Heart Rate Data	26

Get Utility Port	27
Get Mode	28
Set Utility Port	29
Set Mode	30
Get Version	31
Appendix A: Troubleshooting	32
Appendix B: Code Examples	34
Unix/Mac OS X using the serial interface	34
Processing using the serial interface	38
Arduino using I2C	40
Appendix C: Schematic	43

Heart Rate Monitor Interface

Description

The Heart Rate Monitor Interface (HRMI) is an intelligent peripheral device that converts the ECG signal from Polar Electro Heart Rate Monitor (HRM) transmitters into easy-to-use heart rate data. It implements a sophisticated algorithm for computing an average heart rate even with noisy or intermittent data from the transmitter. The HRMI also provides analog inputs and a digital input/output utility port to ease integration into custom applications.

Features

- Multiple interfaces: USB, Logic-level serial and I²C™
- Dual heart rate processing algorithms: averaged and raw
- Uses the RMCM01 Polar OEM receiver
- Compatible with coded and non-coded Polar transmitters including T31, T31C, T61C and Wearlink®
- 32-entry heart rate data history buffer
- Four 8-bit ADC inputs
- Up to a 5 channel digital input/output utility port
- Simple command/response interface
- Programmable power-on default operation

Applications

- Custom exercise equipment
- Portable heart rate monitoring devices
- Bio-feedback devices
- Heart beat aware body-worn electronics

Version Information

Datasheet Revision	HRMI Firmware Version	Comments
1.0	0x01	Initial Release
1.1	0x01	Updated schematic and product image to match Sparkfun v17 design Minor typographical corrections

Disclaimer

Copyright © danjuliodesigns, LLC, 2008-2010, All rights reserved.

Neither the whole nor any part of the information contained in, or the product described in this manual, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder.

This product and its documentation are supplied on an as-is basis and no warranty as to their suitability for any particular

This product is not a medical grade ECG or EKG monitor. This product or any variant of it is not intended for use in any medical appliance, device or system in which the failure of the product might reasonably be expected to result in personal injury.

purpose is either made or implied.

danjuliodesigns, LLC. will not accept any claim for damages howsoever arising as a result of use or failure of this product. Your statutory rights are not affected.

This document and the functionality of the product may be subject to change without notice.

Contact

Email: info@danjuliodesigns.com

Website: <http://www.danjuliodesigns.com/sparkfun/hrmi.html>

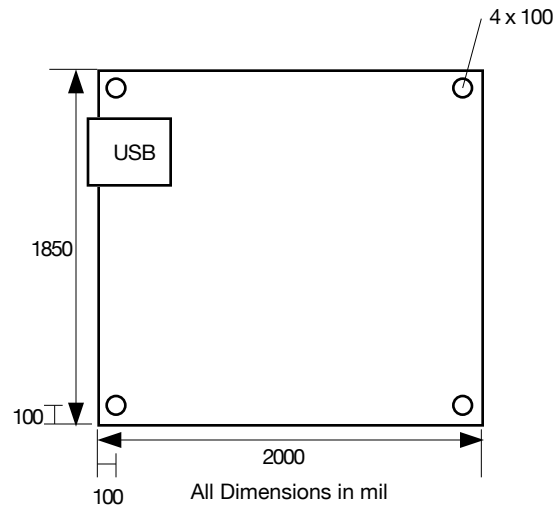
Electrical Specifications

Parameter	Min	Typ	Max	Unit	Conditions
External Voltage Supply	3.6		5.5	V	When powered from the external power connection
USB Voltage Supply	4.5	5	5.5	V	When powered from a USB host
External Current		30	45	mA	No connection to any Analog or Utility port signal
Reception Range	80	92	100	cm	Typical using a T31 transmitter
Reception Frequency		5.5		kHz	
Analog Voltage Range	VSS		VCC	V	Note 1
Utility Port Input Low Level	VSS		0.15* VCC	V	$3.3V \leq VCC \leq 4.5V$
	VSS		0.8	V	$4.5V \leq VCC \leq 5.5V$
Utility Port Input High Level	0.25* VCC + 0.8		VCC	V	$3.3V \leq VCC \leq 4.5V$
	2		VCC	V	$4.5V \leq VCC \leq 5.5V$
Utility Port Output Low Level			0.6	V	$I_{OL} = 8.5 \text{ mA}$, $VCC = 4.5V$
Utility Port Output High Level	VCC - 0.7			V	$I_{OH} = -3.0 \text{ mA}$, $VCC = 4.5V$
Maximum Analog Port Source Impedance			10k	Ω	Note 2
Maximum output current sunk by any Utility Port signal			25	mA	Note 3
Maximum output current sourced by any Utility Port signal			25	mA	Note 3
Capacitive Loading Analog and Utility ports			55	pF	
Configuration EEPROM write cycles	100K	1M		Write Cycles	$T_A \leq 85^\circ\text{C}$
Configuration EEPROM characteristic retention	40			Year	Provided no other specifications are violated
Operating Temperature Range	0	20	60	$^\circ\text{C}$	
Storage Temperature Range	-30	20	70	$^\circ\text{C}$	

Notes

1. VSS is on-board ground, VCC is on-board voltage supply.
2. The maximum Analog Port Source impedance is required in order for the ADC sampling time to be met.
3. Total current sourced or sunk by all Utility Port signals must be less than 80 mA.

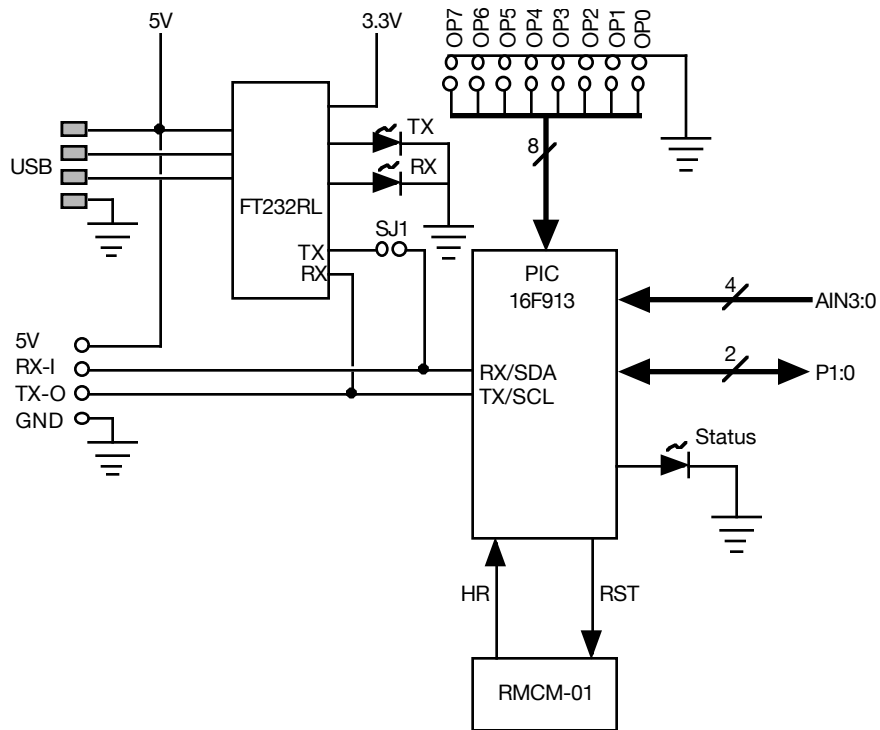
Mechanical Specifications



Board Dimensions

Getting Started

Block Diagram

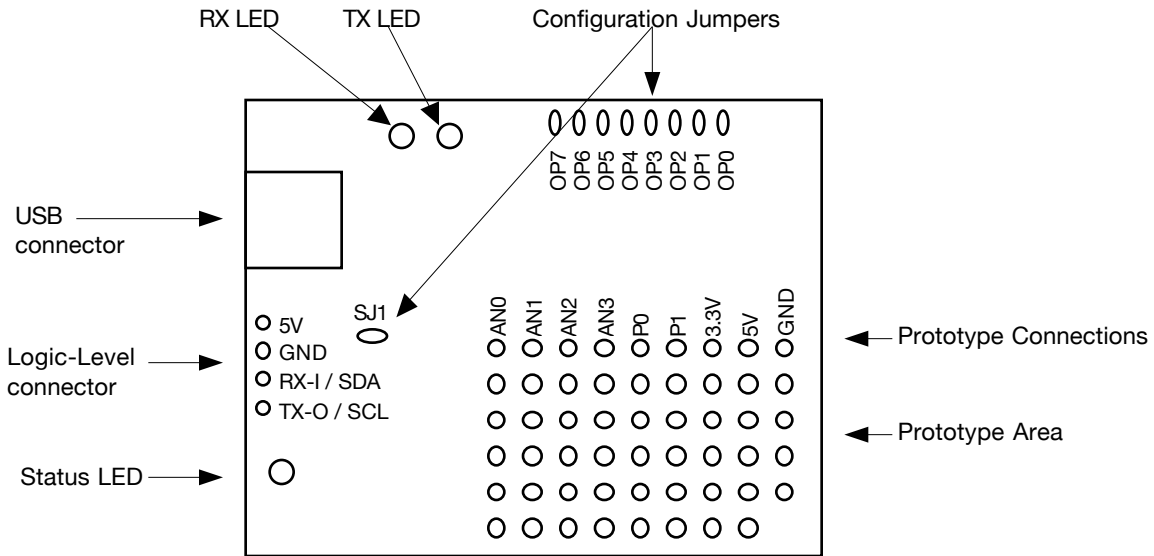


HRMI Block Diagram

Connections

Communication/Power

The HRMI receives power and data through a communication interface. It provides two interface connectors: USB and a Logic-level header. Use of the two physical interfaces is mutually exclusive, only one can be in use at a time.



Board Layout (Top View)

The HRMI communicates using either a serial interface or an I²C interface. The HRMI is configured at power-on reset to use a particular interface with the OP0 - OP7 jumper block as described in section [“Configuration Jumpers”](#) of this chapter. It communicates using a serial interface through either the USB connector or the Logic-level header. It communicates using the I²C interface through the Logic-level header.

USB

A USB Mini-B connector allows quick connection to a host computer. The USB interface provides both power and a data connection to the HRMI. The USB interface uses a FTDI FT232RL interface IC that exposes the HRMI as a serial device to the host computer. FTDI provides royalty free drivers for Microsoft Windows, Apple Macintosh and Linux operating systems.



USB Cable with Mini-B connector

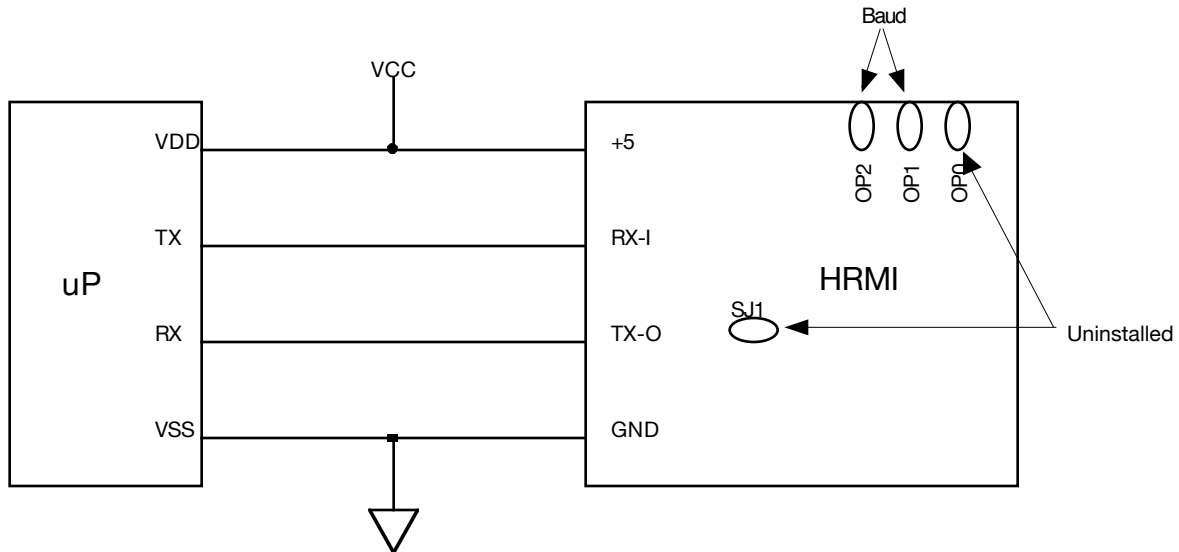
To use the USB interface a jumper should be installed in the SJ1 position (factory default). The jumper in the SJ1 position connects the TX line on the FTDI FT232RL interface IC to the HRMI micro-controller RX input (the micro-controller TX output is permanently connected to the FT232RL RX input). No jumper should be installed in the OP0 position. This configures the micro-controller to use a serial interface at power-on. The desired baud rate is configured on the OP1 and OP2 jumper positions as described in section [“Configuration Jumpers”](#).

A 4-pin header allows direct connection for access to Logic-level serial or I²C interfaces. It also provides Ground and +5 volt power input connections.

Label	Function
5V	+5 volt power input
GND	Ground
RX-I / SDA	HRMI Serial TTL RX input HRMI I ² C SDA (data)
TX-O / SCL	HRMI Serial TTL TX output HRMI I ² C SCL (clock)

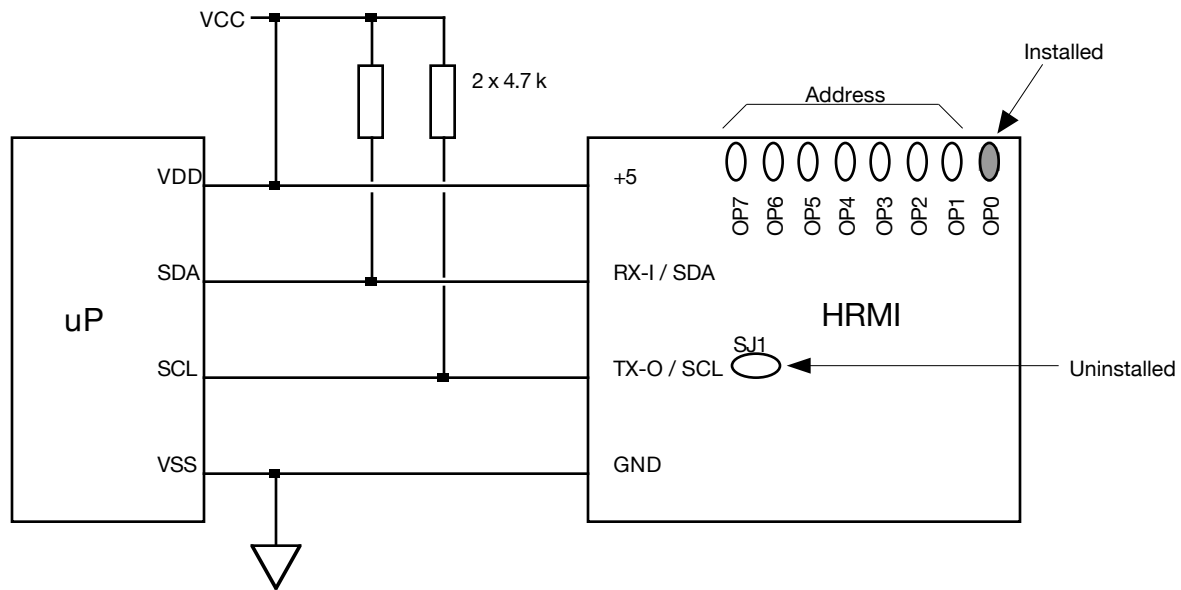
Logic-level header pin description

The HRMI is a serial device operating at baud rates from 2400 - 38400 when jumper OP0 is uninstalled. Data is transmitted at TTL signal levels using 8 data bits, 1 stop bit and no parity (8N1). Jumper positions OP1 - OP2 configure the baud rate. An example of interfacing the HRMI to an external microprocessor using a serial interface is shown below.



Interfacing with a microprocessor through a TTL serial port

The HRMI is a 7-bit I²C slave operating at up to a 100 kHz data rate when jumper OP0 is installed. Jumper positions OP1 - OP7 configure the I²C address (OP1 is LSb). The HRMI does not provide pull-up resistors on the I²C signals. It may be necessary to add these externally. An example of interfacing the HRMI to an I²C device is shown below. More than one I²C device can be wired in parallel observing signal line capacitance limits. Each device must respond to a different address. For more information about I²C please see <http://www.standardics.nxp.com/support/documents/i2c>.



Interfacing with an I²C master

Jumper SJ1 should be removed when using the Logic-level 4-pin header (to disconnect the output port on the FT232RL).

Configuration Jumpers

Configuration jumpers are used to configure the HRMI for operation. The jumpers are two exposed pads situated close together. To install a jumper use a soldering iron to dab a small amount of solder that covers both pads. To remove a jumper heat and remove the liquid solder by dragging the soldering iron tip quickly sideways across the pads to allow the surface tension of the solder to separate it into two blobs, one on each pad. Alternatively use the soldering iron to melt the solder and remove it using a small piece of solder-wick. Be careful not to apply too much heat or the solder pads may pull up off of the board .

Configuration jumper SJ1 is used to connect the TX line on the FTDI FT232RL interface IC to the HRMI micro-controller RX input when using the USB interface.

Configuration jumpers OP0 - OP7 configure the micro-controller communication interface at power-up. They are only read at power-up. OP0 configures the interface type. Without a jumper across OP0 the micro-controller communicates using a serial interface through either the Logic-level connector or USB connector. With a jumper across OP0 the micro-controller communicates using the I²C protocol.

With OP0 removed, OP1 and OP2 configure the baud rate. OP3 - OP7 are unused. The data protocol is 8 data bits, 1 stop bit and no parity (8N1). By default, with no jumpers installed, the baud rate is 9600 baud.

OP2	OP1	Baud Rate
installed	installed	2400 baud
installed	uninstalled	19200 baud
uninstalled	installed	38400 baud
uninstalled	uninstalled	9600 baud

Serial Interface (OP0 uninstalled)

Note: The baud rate configured with jumpers OP1 and OP2 must match the baud rate selected by the software running on the host computer.

With OP0 installed, OP1 - OP7 configure the 7-bit I²C slave address. OP1 is the least-significant bit. The maximum I²C transfer rate is 100 kHz. By default with no jumpers (other than OP0) installed, the address is 127.

OP7	OP6	OP5	OP4	OP3	OP2	OP1	Address
installed	installed	installed	installed	installed	installed	installed	0
installed	installed	installed	installed	installed	installed	uninstalled	1
installed	installed	installed	installed	installed	uninstalled	installed	2
installed	installed	installed	installed	installed	uninstalled	uninstalled	3
...							...
uninstalled	uninstalled	uninstalled	uninstalled	uninstalled	uninstalled	uninstalled	127

I²C Interface (OP0 installed)

Status Indicators

The HRMI contains three status LEDs.

1. USB TX: Red LED that flashes for data transmitted from the host to the HRMI through the USB interface.
2. USB RX: Green LED that flashes for data transmitted from the HRMI to the host through the USB interface.
3. Status: Green LED that indicates the status of the HRMI. It is on when the HRMI is searching for valid heart beat data. It blinks at the detected heart rate when a signal is acquired. For each heart beat it is on for 100 mSec.

Prototype Connections

The HRMI provides four analog inputs and two programmable digital input/outputs along with a small prototyping area to include local circuitry. The analog inputs are connected to the HRMI micro-controller 8-bit ADC inputs. The programmable digital input/output ports are connected to general purpose I/O pins on the HRMI micro-controller and

can be configured to be either a digital input or digital output and drive logic-level signals. Both +5 volts and +3.3 volts are available.

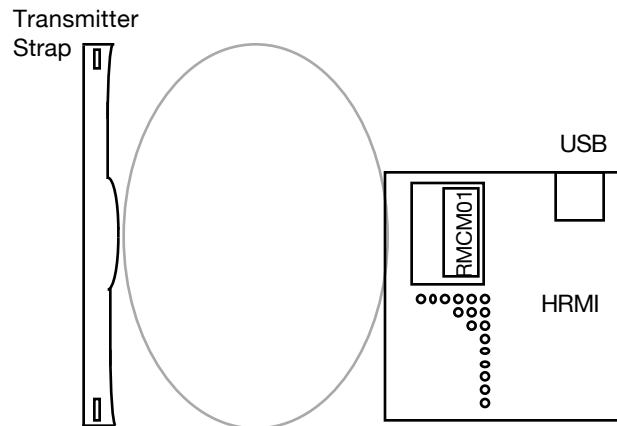
Label	Function
AN0 - AN3	Analog Input
P0 - P1	Programmable digital I/O (Utility port)
3.3V	+3.3 volt supply rail. Power draw from this pin should be limited to a maximum of 50 mA.
5V	+5 volt supply rail. Connected to the 5V input connector and to the USB power rail. Care must be taken not to exceed USB current limits.
GND	Board Ground

Prototype Signals

Orientation with Polar Transmitter

Polar transmitters use a magnetic field to transmit data to the RMCM-01 OEM receiver on the HRMI board. The following rules help maximize signal transfer between a polar transmitter and the HRMI.

1. The maximum distance between the transmitter and HRMI should not exceed 80 cm.
2. The magnetic field is generated and detected using coils in the transmitter and receiver. The receiver coil should be in parallel with the magnetic flow generated by the transmitter for maximum energy transfer as illustrated in the following diagram.



Transmitter and receiver orientation

3. Metal casing around the receiver may form a Faraday cage around the receiver attenuating the signal. A metal cage may also change the orientation of the magnetic field coming from the transmitter.

4. Interference created by other electronic devices (such as motors, displays and power supplies) may interfere with the transmission of information from the transmitter to receiver. Optimally the HRMI will be physically separated from such sources of electro-magnetic energy.

Quick Test

The ability of the HRMI to detect a heart rate can be tested with just a Polar transmitter and 5 volt regulated power supply.

Attach the HRMI to the 5 volt power supply and switch on the power supply. Alternatively plug the HRMI board into a computer USB port. The HRMI Status LED should light.

Strap on the Polar transmitter and orient yourself near the HRMI board. Within 15 seconds the HRMI Status LED should begin to flash indicating it has detected heart rate data from the transmitter. The Polar transmitter operation can be verified with a Polar wristwatch receiver.

Quick Test with Host Computer

A computer with USB interface and terminal emulator program can access the HRMI. A USB cable and Polar transmitter are required in addition to the computer. The basic steps are listed below. Some specific instructions for particular platforms follow in subsequent sections.

1. Install a jumper at position SJ1. Make sure there are no jumpers installed on positions OP0 - OP7.
2. Before attaching the HRMI to the computer download and install the appropriate driver from the FTDI website: <http://www.ftdichip.com/FTDrivers.htm>. The website contains the most current driver and instructions for each supported operating system. Be sure to load the Virtual COM Port Driver (VCP) for this test instead of the D2XX driver. The VCP driver makes the HRMI appear as a standard serial device (COM port).
3. Once the driver has been installed then attach the HRMI to the computer's USB interface. The HRMI Status indicator should light and blink when a heart beat is detected from a Polar transmitter.
4. Start a serial terminal emulator program on the host computer (for example, Hyperterminal on Microsoft Windows). Make sure it is configured to use the USB-based communication port associated with the HRMI at 9600 baud, 8 bit data, 1 stop bit and no parity. It is handy but not necessary to configure the terminal emulator program to echo back characters typed locally and to append a linefeed character to received line ends. This allows display of the command sent to the HRMI as well as the result returned by the HRMI.
5. Type the following command at the terminal emulator: "G1 <CR>". That is the three characters "G" followed by "1" followed by the carriage return ("Enter" key or control character for carriage return, CTRL-M). This should result in a string such as the following.

```
1 15 63
```

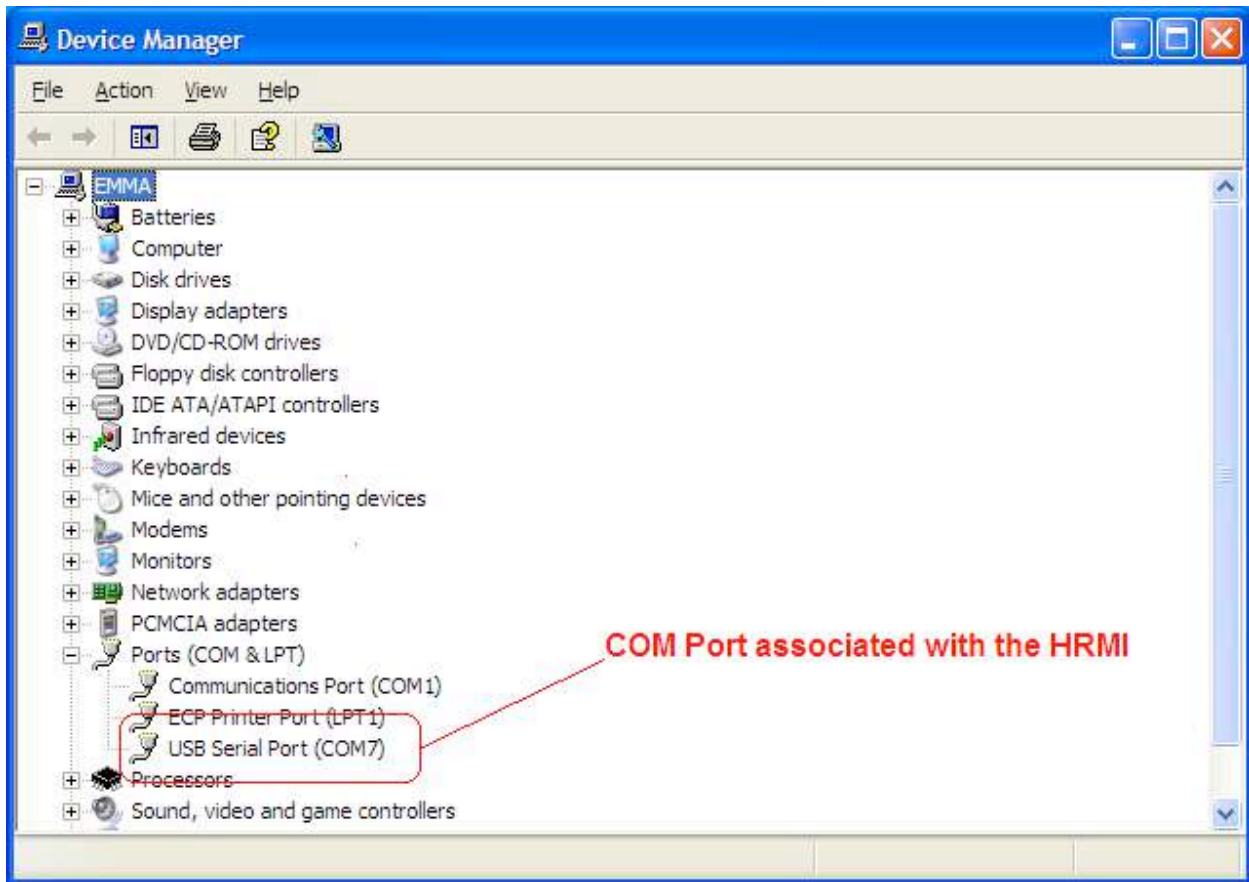
where the first number, "1", indicates the current heart rate algorithm mode, the second number, "15", indicates the current timer value and the third number, "63", the current heart rate (your actual heart rate may vary). See the section "[Command List](#)" for a detailed description of this and other commands.

Quick Test using HyperTerminal (Microsoft Windows)

These examples are shown using Microsoft Windows XP™.

Use Device Manager to identify the COM port associated with the HRMI before starting HyperTerminal.

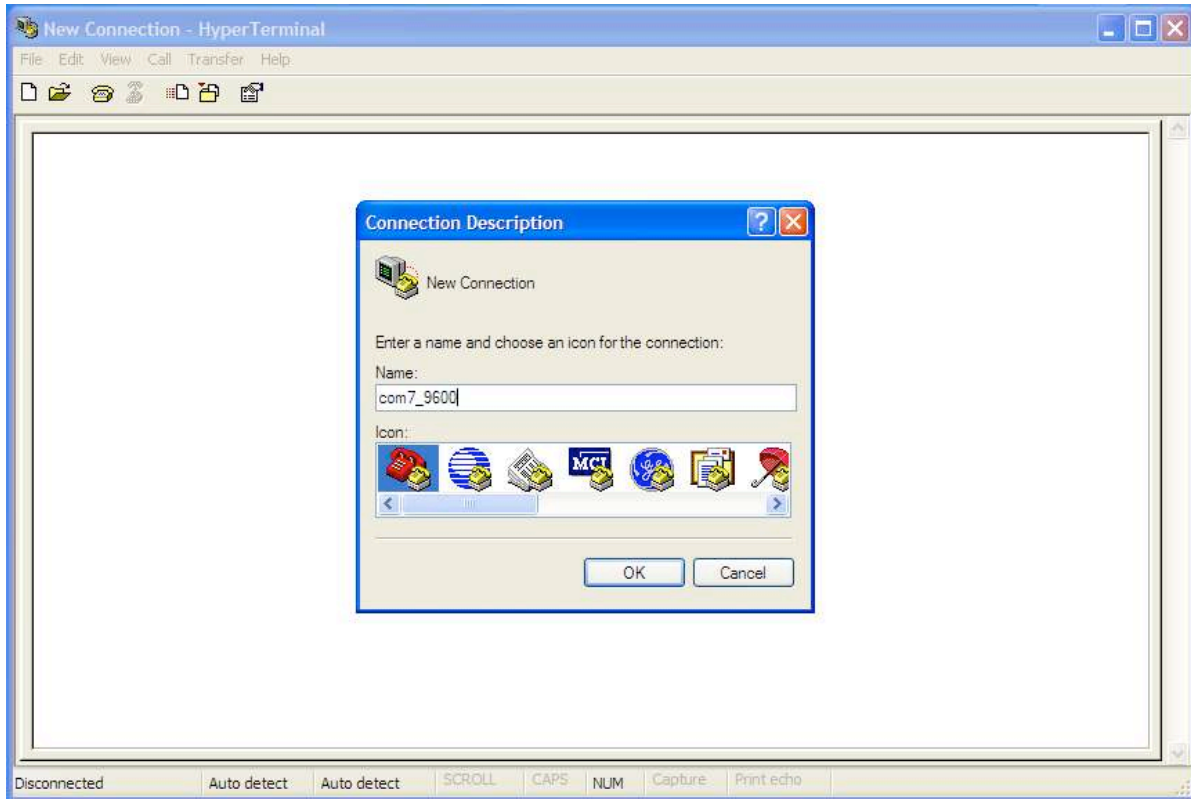
1. Plug the HRMI into a USB port (after the FTDI driver has been loaded). Verify the Status LED is lit. It may be helpful if the HRMI is the only USB peripheral attached as other USB peripherals may also use a FTDI USB Interface IC.
2. Left-click on the My Computer icon on the desktop or Start menu and select “Properties”. This will bring up the “System Properties” window.
3. Select “Device Manager” from the “System Properties” Window. The “Device Manager” window will appear as shown in the following picture. Click the “+” symbol next to “Ports (COM & LPT)” to display a list of ports the system has identified. A new COM port will be listed as shown (assuming the HRMI is the only USB peripheral attached using a FTDI USB Interface IC).



Device Manager showing a USB COM port

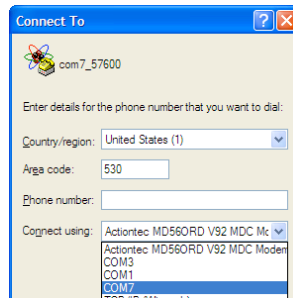
Start and configure HyperTerminal (HyperTerminal is located in “All Programs -> Accessories -> Communications”). Configuration consists of the following illustrated steps.

1. Create a new connection configuration, for example “com7_9600”, in the “Connection Description” dialogue box. HyperTerminal can save this configuration to disk for use at a later time.



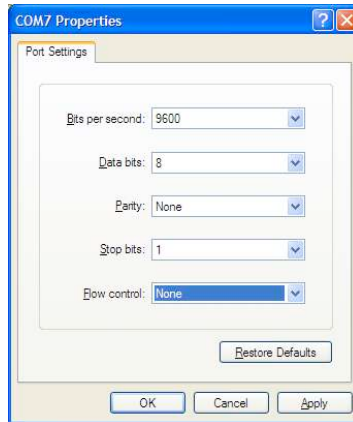
Configuring HyperTerminal with a new configuration

2. Select the COM port associated with the HRMI in the “Connect To” dialogue box.



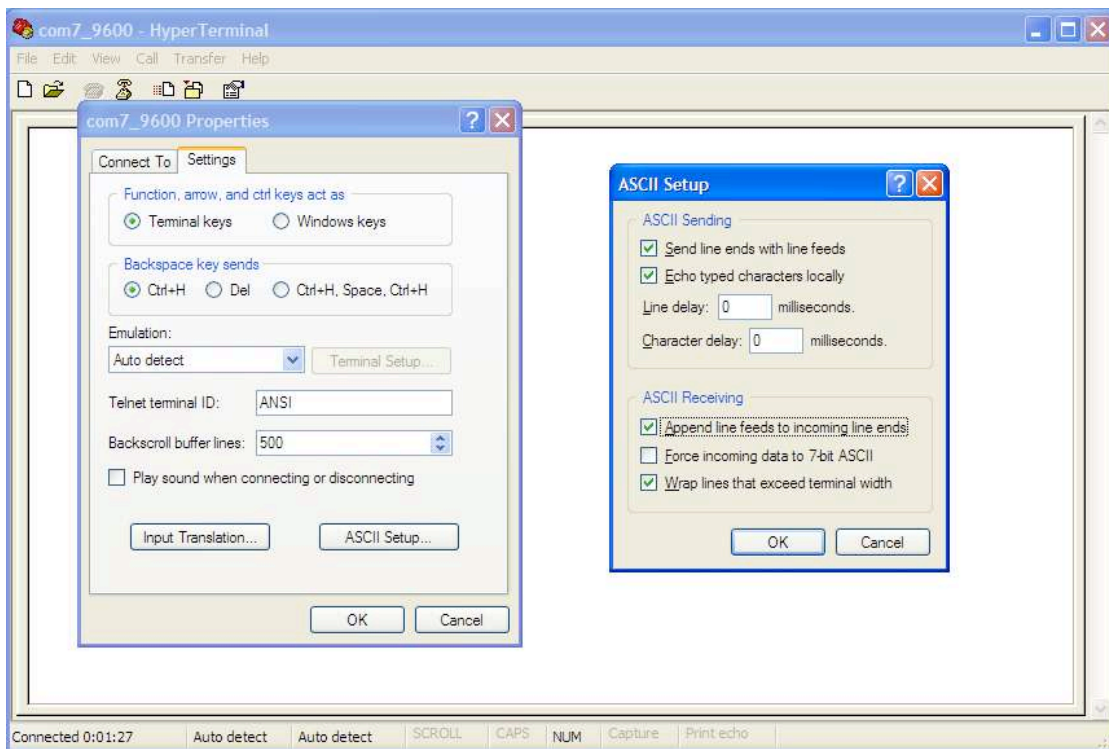
Configuring HyperTerminal to use the USB COM port

3. Configure the correct serial port characteristics in the “Properties” dialogue box. Be sure that flow control is set to “None”.



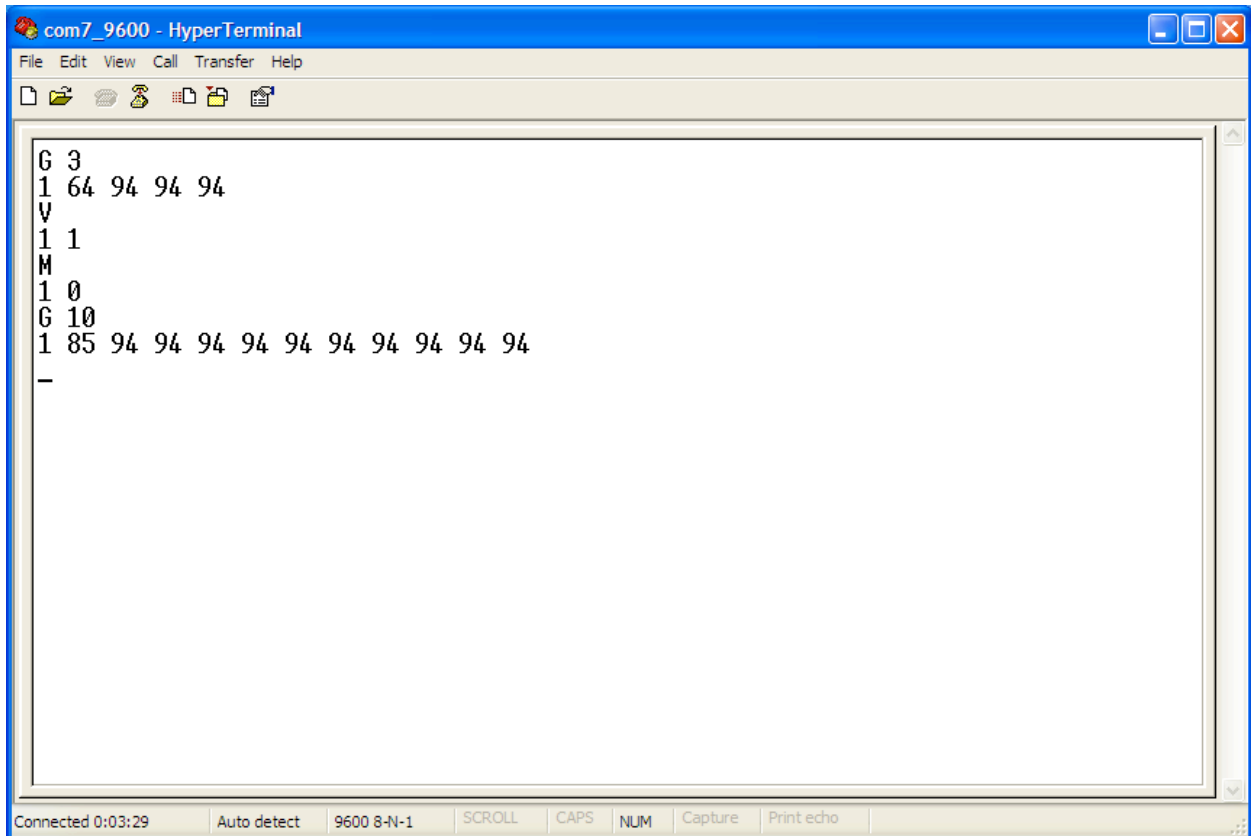
Configuring HyperTerminal for 9600 baud, 8N1

4. Select “Properties” for HyperTerminal (File->Properties) then click on “ASCII Setup...” to bring up the “ASCII Setup” dialogue box. Click the “Echo typed characters locally” checkbox to cause HyperTerminal to echo the characters transmitted to the HRMI on the screen. Click the “Send line ends with line feeds” and “Append line feeds to incoming line ends” checkboxes to cause HyperTerminal to put each text string on a separate line for easy viewing.



Configuring HyperTerminal to echo characters

Send commands to the HRMI followed by the Carriage Return character (CTRL-M) and observe the results. See the section "[Command List](#)" for a detailed description of each command.



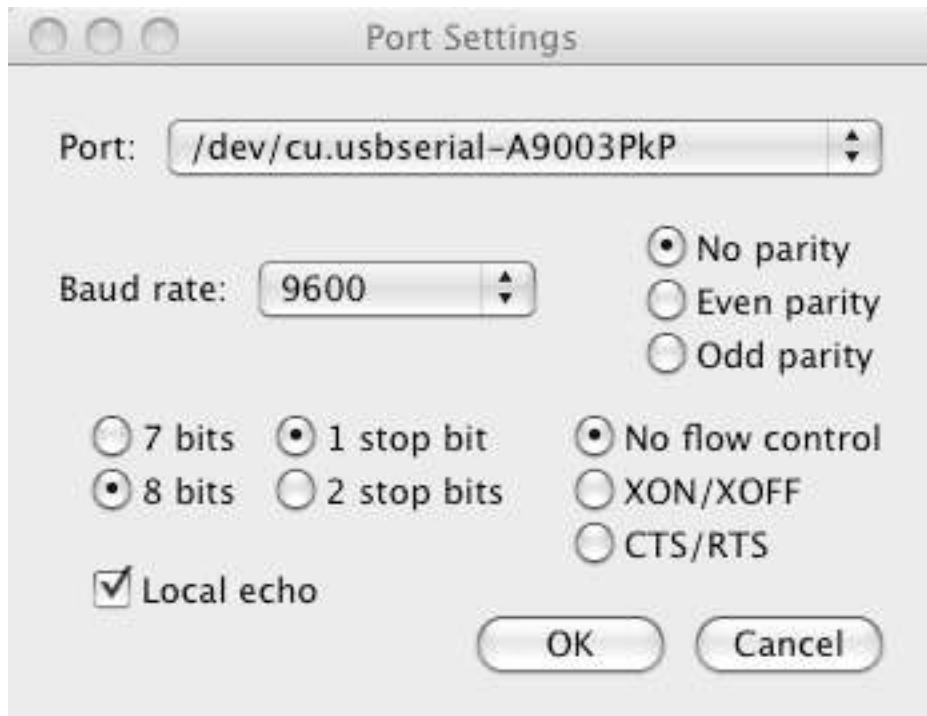
HRMI Commands and Responses

Quick Test using QuickTerm (Apple Mac OS X)

QuickTerm is a simple universal terminal application compatible with Mac OS X 10.2 - 10.5. It can be found at the developer's website <http://www.gelhaus.net/cgi-bin/showpage.py?cocoa/+quickterm.html> or through the version tracker website (<http://www.versiontracker.com>).

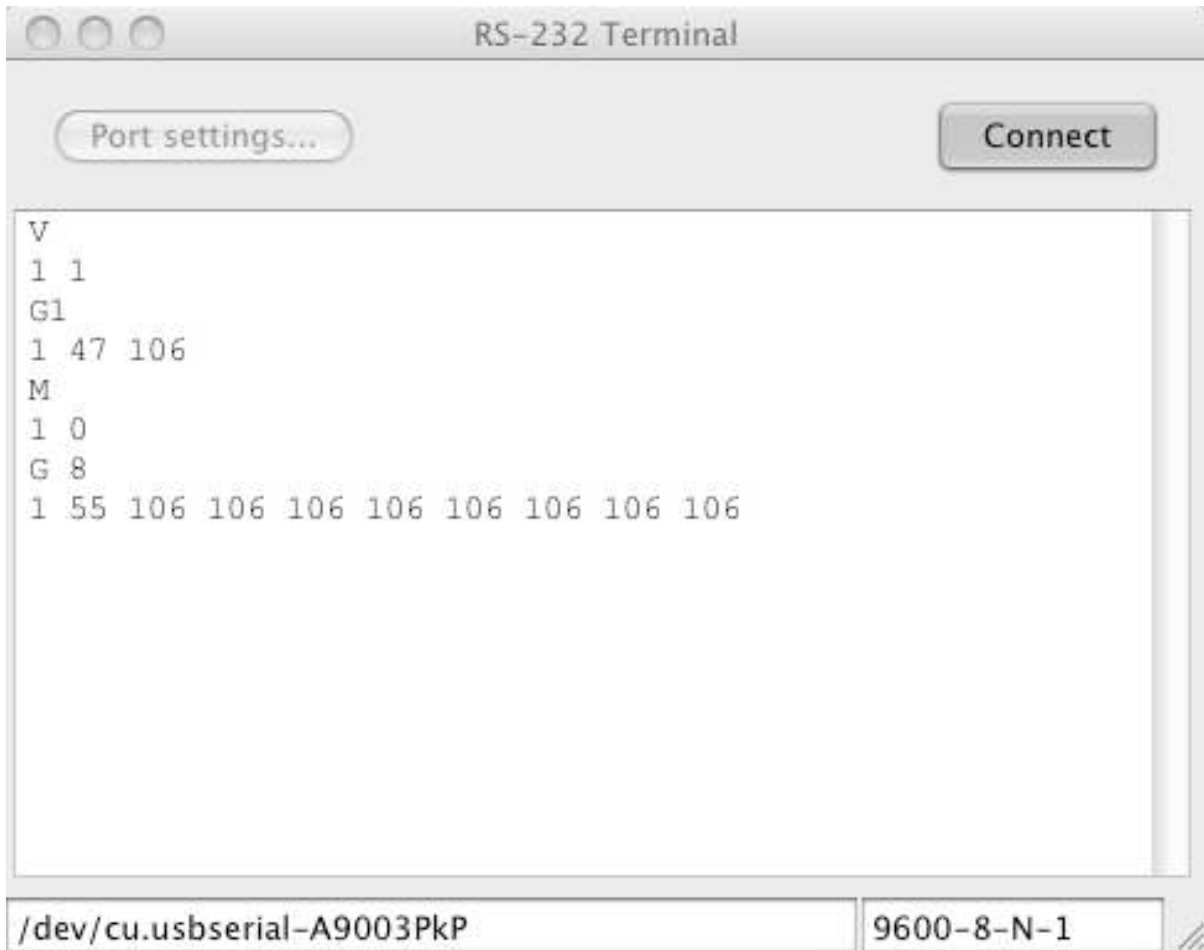
Currently QuickTerm (version 1.01) is limited to a maximum of 19200 baud. The HRMI must be configured to work at 19200 baud or less. This example uses the default baud rate of 9600 baud.

1. Plug the HRMI into a USB port (after the FTDI driver has been loaded). Verify the Status LED is lit. It may be helpful if the HRMI is the only USB peripheral attached as other USB peripherals may also use a FTDI USB Interface IC.
2. Start QuickTerm and configure the serial port by clicking on the "Port settings...". Select the appropriate device from the "Port" pull-down menu. The entry will have the form "/dev/cu.usbserial-XXXXXXX" where the string "XXXXXXX" is unique to each FTDI FT232RL IC. Select 9600 baud. Make sure there is no flow control enabled and select the "Local echo" checkbox.



Example QuickTerm Port Settings

3. Click the “Connect” button and send commands to the HRMI followed by the Carriage Return character (CTRL-M) and observe the results. See the section “[Command List](#)” for a detailed description of each command.



HRMI Commands and Responses

Note: Do not disconnect the HRMI from the Macintosh USB port while any application is using the serial port (the serial port is open). This may cause the system to freeze or crash.

Operation

Host Interface Operation

The HRMI is a slave device. It responds to commands issued to it from a host controller. It interprets commands from either a serial or I²C interface. Both interfaces use the same physical I/O ports on the HRMI micro-controller. The interface type is selected by sampling jumper OP0 at power-on. The interface type persists until the next power-on reset. The HRMI is capable of receiving commands 250 mSec after being powered up.

Unknown or illegal commands are ignored. Some commands cause the HRMI to generate a response which it transmits back to the host controller after processing the command.

Commands are 1-byte (8-bits in length). They have ASCII representations that are an abbreviation of their function. Some commands are followed by an argument representing a single 8-bit numeric quantity.

Responses consist of one or more numeric values.

Commands and responses are encoded slightly differently for serial and I²C communication.

Serial Interface (USB or Logic-level physical interface)

Commands and arguments sent through the serial interface are encoded as ASCII values. Commands are a single ASCII character with values between 'A' and 'Z'. Arguments are one to three ASCII numbers "0" through "255" representing an 8-bit value. Commands are issued in Command Sequences. A Command Sequence consists of the command value followed by any required argument terminated with the Carriage Return character (0x0D). Space characters may be included between the command and argument or between the argument and the Carriage Return (<CR>). Any other characters or additional arguments cause the HRMI to ignore the Command Sequence up to the next Carriage Return.

Responses consist of ASCII numbers ("0" through "255") separated by spaces terminated with the Carriage Return character.

For example the ASCII character 'G' is used for the Get Heart Rate command. It is always followed by an argument with a value of 0 - 32 commanding the HRMI to return 0 - 32 heart rate values from its heart rate history buffer. The HRMI also includes two status bytes prior to the requested heart rate data.

Sending the command

```
G3<CR>
```

or

```
G 3 <CR>
```

generates a response with the form

```
1 25 63 64 67 <CR>
```

where the values "1" and "25" are the status bytes (explained later in this chapter) and the values "63", "64" and "67" are the requested heart rate values (in beats-per-minute) from the heart rate history buffer.

The HRMI can buffer multiple serial commands and arguments in an internal 16-byte FIFO. It processes one command at a time (including transmission of a required response). The HRMI will continue to push new commands into the FIFO while it is processing a command. It will drop commands when the FIFO is full. Care must be taken when sending commands immediately following a Get Heart Rate command and before receiving the response from the Get Heart Rate command so as not to overflow the FIFO.

I²C Interface

The HRMI implements a simple 7-bit slave I²C device. It is configured with any 7-bit I²C address on the OP1-7 configuration jumpers at power-on. The HRMI implements only the mandatory I²C slave functionality described in the NXP Semiconductor document "UM10204 I²C-bus specification and user manual". These functions include START condition, STOP condition, Acknowledge and 7-bit slave address. It does not support General Call address, 10-bit slave address, Software reset or Device ID. The maximum bit rate is 100 kHz.

Commands and arguments are encoded as 8-bit binary values. Command Sequences consist of 1- or 2-byte I²C write sequences (the command byte followed by any required argument byte) followed by the STOP condition.

The host controller must issue a read of the expected number of data bytes for commands that generate a response. The read should be issued following the write and before subsequent commands. The HRMI stretches the clock on the SCL line during the read while it fetches data bytes internally to satisfy the read. The HRMI will return the value 0x00 for host controller I²C reads that request more data than the is contained in the response or for reads without a prior command.

The example shown in the Serial Interface section above would consist of the following bytes transmitted on the I²C interface for the command.

```
<0x47><0x03>
```

The following response data would be read (the host controller must read 5 bytes for this example).

```
<0x01><0x19><0x3F><0x40><0x43>
```

The HRMI can buffer multiple I²C commands and arguments in an internal 16-byte FIFO. However it can only process one command at a time that requires a response. This is because it resets the internal buffer it uses to hold responses

each time a command is received from the I²C interface. The host controller must execute a read and obtain the response data for any command that generates a response before issuing another command.

Heart Rate Monitor Algorithms

The HRMI computes the heart rate by measuring the time from one heart beat to the next as received from the Polar transmitter. The transmitter sends a pulse for each heart beat it detects. During normal operation the data from the Polar transmitter may contain a significant amount of spurious information. For example a slightly loose transmitter strap may generate extra pulse information each time the strap contacts are jostled (as can occur frequently when running). Other devices can also generate electromagnetic pulses that appear to the receiver as heart rate data.

The HRMI allows access to the received data in either raw or averaged form (Raw or Average mode). In Raw mode the instantaneous heart rate based on each pulse is calculated and stored in a 32-entry history buffer. Raw mode data is useful for observing dynamic heart response on a beat-by-beat basis although the data may include erroneous heart rate values due to spurious pulses. It is the responsibility of software running on the host controller or user analysis to identify possibly spurious heart rate values in raw data.

In Average mode the instantaneous heart rate data is averaged and the average heart rate, computed once per second, is pushed into the 32-entry history buffer. The averaging algorithm identifies and ignores spurious heart rate data when generating the average by using a pulse rate discriminator and a dual set of averaging buffers (a primary buffer and an acquisition buffer). The acquisition buffer is used to determine the initial average. Once an average has been computed the primary buffer is used to maintain calculations of the average. The primary buffer is dynamically sized from 6 to 16 entries using more entries for higher average heart rates to prevent slightly spurious values from affecting the average too much while allowing the average to be responsive to quickly changing heart rates. Once an average has been computed the acquisition buffer is used when spurious values too large for the primary averaging buffer are detected. The acquisition buffer requires a certain number of values within a range before determining that there is a new average. The discriminator is used to detect spurious values for both the primary and acquisition buffer.

The algorithm is selected from a built-in EEPROM at power-up. The algorithm may also be changed while the HRMI is running by issuing a command. From the factory the power-up value configures the HRMI to use the Averaged algorithm. The power-up value may also be changed by issuing a command to the HRMI.

In raw mode the current instantaneous heart rate value is pushed into the heart rate history buffer each time a pulse is received. In average mode the current average is pushed into the heart rate history buffer each second while non-spurious data is being received. The history buffer is not loaded in average mode while spurious data is being received. The last entry pushed into the history buffer represents the most recent accurate average heart rate. The HRMI clears the heart rate history buffer when there has been a loss-of-signal from the transmitter for more than fifteen seconds. It also clears the heart rate history buffer if more than 64 spurious pulse values are received. When the heart rate history buffer has been cleared or before any values have been loaded the HRMI will generate responses with the value of 0 for the heart rate.

The status indicator LED is pulsed on for 100 mSec when a pulse is received. It is set to a solid on condition when no pulse has been received in more than 1792 mSec setting a lower bound to detectable heart rate values at 34 beats-per-minute (bpm). The upper bound to detectable heart rate values is 239 bpm.

Command List

The HRMI implements the following commands, summarized in the following table and described more fully in the following sections.

Command	ASCII Command Value	Hex Command Value	Command Argument	Response	Description
Read Analog Input	A	0x41	Channel 0-3	8-bit value	Trigger the ADC to read the specified analog input.
Set Utility Port Directions	D	0x44	5-bit direction mask	-	Set the direction, either input or output, of each Utility port pin.
Get Heart Rate Data	G	0x47	Number of heart rate history buffer entries 0-32	2 - 34 8-bit values	Get the specified number of heart rate values and status information from the heart rate history buffer.
Get Utility Port	I	0x49	-	5-bit value	Get the current value of each Utility port pin.
Get Mode	M	0x4D	-	1 1-bit value followed by 1 5-bit value	Get the current heart rate algorithm and Utility port direction.
Set Utility Port	O	0x4F	5-bit value	-	Set the value of each Utility port pin.
Set Mode	S	0x53	3-bit mode	-	Set the heart rate algorithm and optionally set the power-on defaults for heart rate algorithm and Utility port direction and value.
Get Version	V	0x56	-	2 8-bit values	Get information about the firmware running in the HRMI.

HRMI Command List

Read Analog Input

Description

Trigger the HRMI to read the analog voltage level on one of the analog inputs and return the value as an 8-bit number.

Serial command

A<N><CR>

I²C command

<0x41><N>

Argument

<N> specifies the analog input to read, 0-3. Other values cause the command to be ignored.

Serial response

Returns 1 8-bit byte with the ADC value encoded as an ASCII string "0 " to "255 " followed by <CR>.

I²C response

Read 1 8-bit byte with the ADC value.

Notes

1. The HRMI delays 5 uSec after configuring the specified analog input before triggering the ADC to allow the sampling capacitor in the ADC time to charge. The maximum input impedance seen by any analog input pin should be less than 10 k Ω to allow the capacitor to fully charge in this time.
2. The internal micro-controller ADC is referenced to the 5 volt power rail which means the analog inputs are scaled relative to the power rail. It is possible to use a precision reference voltage by attaching the voltage reference to one of the analog inputs. Voltage levels attached to other analog inputs can be scaled by the voltage reference by dividing the result read from the desired analog input by the result read from the reference analog input.