BioHarness BT Android API User Guide

Document History

Version	Date	Description
1.0	April 26th 2011	First Draft
1.1	April 27th 2011	Added Events and Summary Data Packet. Added *PacketTypeRequest* class and its methods
1.2	June 23rd 2011	Added classes to unpack ECG, R-R, Accelerometry 100mg, Summary and Packet Events Added Support to Enable or disable logging
1.3	June 24 2011	Reformat – no content change

Zephyr technology

Contents

1. Introduction

This document describes the functionality of the BioHarness BT API. It provides an overview of all the classes and methods which have been implemented in the API to communicate with the BioHarness 2 and the BioHarness 3. Please note that the example application currently supports General Packet Data only. The API however supports all the messages listed below. By following the example application, the user can see how to enable the other packet types. For a complete list of Messages and their descriptions you will need to refer to the General Comms Link document.

The intended audience of this document is an Android programmer who needs a quick introduction to communicate with the BioHarness. Along with the API, support is also provided via some source files to get a jump start for communicating with the BioHarness.

The BioHarness BT application package enables a user to receive the following messages concurrently from a paired BioHarness device:

1. General Data Packet
2. ECG Data Packet
3. Breathing Data Packet
4. R-R Data Packet
5. Accelerometer Data Packet
6. Summary Data Packet
7. Event Data Packet

2. The Example Application

The example application utilizes the API and enables the BioHarness to transmit the different packet types listed above. The following steps below are a description of the most important aspects of the source code in the example application used to enable the General Packet and display the data on the Android Phone.

1. Upon clicking the Connect button, a Bluetooth adaptor type object is created and passed to an object of the *BT Client* class type. The *BT Client* object is essentially a thread that manages the overall Bluetooth connectivity of the phone with the BioHarness device.

2. Next, an object of the *NewConnectedListener* class needs to be created which essentially implements the *ConnectedListener* interface and one that extends the *ConnectedListener* class. This object is responsible for reacting differently to different kinds of messages. In this object we override the parent class's *connected* method and define our own method. In this method we create a *ZephyrProtocol* object and call its *addZephyrPacketEventListener* method. This method will take a *ZephyrPacketListener* argument ,in whose

ReceivedPacket method we define what message we are interested in, and how we want the data to be displayed on the phone screen.

3. This *ConnectedListenerImpl* object then needs to be connected to the *BTClient* object type via *addConnectedEventListener* function call to tie this object to respond to a received packet from the BioHarness.

4. Calling the *start* function of the *BTClient* thread kicks off the communication of the Application with the BioHarness device.

For a complete description of the different classes along with their variables and methods, please refer to the Appendix.

3. Appendix

3.1. The BTClient Class

The BT client class is a thread that is used to manage the Bluetooth communication between the phone and the BioHarness. The object of this class is responsible for kick-starting the communication procedure to get data from the BH. The variables of this class are as follows:

Variable Name	Variable Class Type	Variable Type	Purpose
_adapter	`BluetoothAdapter`	**Android**	Holds the adapter variable defined in the MainActivity class.
_btstream	`BluetoothSocket`	**Android**	Stores the RFComm Socket to communicate to a remote device(the BioHarness)
_comms	`BTComms`	**User-defined**	Stores a BTComm variable object
_connectionString	`String`	**Java**	Stores the MacID of the BioHarness
_device	`BluetoothDevice`	**Android**	Stores the instantiation object of the remote Bluetooth device
_isConnected	`Boolean`	**Java**	Variable to check connectivity of the phone to the remote device
_isValidBlueToothAddress	`Boolean`	**Java**	Checks the validity of the MacAddress of the remote device
eventSubscribers	`Vector`	**Java/ User-defined**	Stores a list of objects that implement the *ConnectedListener* Interface

The methods for this class are:

Method Name	Method purpose
addConnectedEventListener	Adds objects to the eventSubscribers list that implement the *ConnectedListener* Interface
Close	Closes the _comms thread and the communication link with the remote

	Bluetooth device.
getComms	Returns the _comms variable of this class.
getDevice	Returns the BluetoothDevice object representing the remote Bluetooth device.
IsConnected	Returns the _isConnected variable of this class.
IsValidBlueToothAddress	Returns a Boolean indicating whether the Mac Address of remote device is valid or not.
OnConnected	Iterates through the eventSubscribers list of objects that implement the *ConnectedListener* Interface and calls the ConnectedMethod for those objects.
removeConnectedEventListener	Removes objects from the eventSubscribers list that implement the *ConnectedListener* Interface
run	This is the run method for this thread. Calls the StartCommunication method if this device is connected to the phone.
StartCommunication	Instantiates the BTComms thread and calls the OnConnected method.
BTClient	In the constructor for this object, the socket is created using MY_UUID

3.2. The BTComms Class

The BTComms class is a thread that is used to read from the input stream and write to the output stream to communicate with the remote Bluetooth device. The variables of this class are as follows:

Variable Name	Variable Class Name	Variable Class Type	Purpose
_btstream	`BluetoothSocket`	**Android**	Stores the RFComm Socket to communicate to a remote device(the BioHarness)
_istream	`InputStream`	**Java**	Input Stream to read data coming from the remote device.
_ostream	`OutputputStream`	**Java**	Output Stream to write data to the remote device.
_queue	`LinkedBlockingQueue`	**Java**	Queue of bytes to read data from and process it.
eventSubscribers	`Vector`	**Java/ User-defined**	Stores a list of objects that implement the *ReceivedListener* Interface

The methods for this class are:

Method Name	Method purpose
addReceivedEventListener	Adds objects to the eventSubscribers list that implement the *ReceivedListener* Interface
CallingReceivers	Executes the OnReceived method forever
canRead	Returns status to test whether the input stream for the remote device can be read from.
canWrite	Returns status to test whether the output stream for the remote device can be written to.
Close	Closes the input and output stream used for communicating with the remote device
OnReceived	Iterates through the eventSubscribers list of objects that implement the *ReceivedListener* Interface and calls the Received Method for those objects.
removeReceivedEventListener	Removes objects from the eventSubscribers list that implement the

	ReceivedListener Interface
Run	Forever reads from the input stream and stores it in the queue to be processed.
Write	Method used to write information bytes to the output stream and to the remote device.

3.3. The ConnectedEvent Class

The methods for this class are:

Method Name	Method purpose
getSource	Returns the event source passed into this object.

3.4. The ConnectedListener Interface

The method defined for this interface is:

Method Name	Method purpose
Connected	Connected Method defined with ConnectedEvent<T> type argument as an input parameter.

3.5. The ConnectedListenerImpl Class

This class handles the processing of the input packets from the BioHarness and is responsible for creating objects that parse the input data stream and other objects and methods that display the data on the phone screen. The variables defined for this class are:

Variable Name	Variable Class Name	Variable Class Type	Purpose
GPInfoPacket	GeneralPacketInfo	User-defined	Stores the General Packet Information for each received General Packet data from the BioHarness
_handler	Handler	Android/ User-defined	Used to process the input packet type based upon the received packet type from the BioHarness.
ConnectListenerImpl	Constructor	User-defined	Used to instantiate a GeneralPacketInfo object and initialize local variables.
GP_ENABLE	Boolean	User-defined	Enables or disables a General Packet transmission
ECG_ENABLE	Boolean	User-defined	Enables or disables a ECG Packet transmission
BREATHING_ENABLE	Boolean	User-defined	Enables or disables a Breathing Packet transmission
RtoR_ENABLE	Boolean	User-defined	Enables or disables a R to R Packet transmission
ACCELEROMETER_ENABLE	Boolean	User-defined	Enables or disables a Accelerometer Packet transmission
SUMMARY_ENABLE	Boolean	User-defined	Enables or disables a Summary Data Packet transmission
LOGGING_ENABLE	Boolean	User-defined	Enables or disables Data logging in the BioHarness

The method defined for this object is:

Method Name	Method purpose
Connected	1. Create a new ZephyrProtocol object. 2. Calls the addZephyrPacketEventListener method.

3.6. The CRC8 Class

This class defines methods and variables for computation of the CRC of each received packet from the BioHarness. The variables defined for this class is:

Variable Name	Variable Class Name	Variable Class Type	Purpose
_crc8Poly	`Integer`	**`User-defined`**	Stores the CRC polynomial to perform CRC calculation for each packet.

The method defined for this class is:

3.7. The PacketTypeRequest Class

This class defines methods and variables to enable/disable the different Packet types supported by the API. The variables defined for this class are:

Variable Name	Variable Class Name	Variable Class Type	Purpose
GP_ENABLE	`Boolean`	**`User-defined`**	Stores the Enable or Disable status of the General Packet.
ECG_ENABLE	`Boolean`	**`User-defined`**	Stores the Enable or Disable status of the ECG Packet
BREATHING_ENABLE	`Boolean`	**`User-defined`**	Stores the Enable or Disable status of the Breathing Packet
RtoR_ENABLE	`Boolean`	**`User-defined`**	Stores the Enable or Disable status of the R to R Packet
ACCELEROMETER_ENABLE	`Boolean`	**`User-defined`**	Stores the Enable or Disable status of the R to R Packet
SUMMARY_ENABLE	`Boolean`	**`User-defined`**	Stores the Enable or Disable status of the Summary Packet
EVENT_ENABLE	`Boolean`	**`User-defined`**	Stores the Enable or Disable status of the Event Packet
LOGGING_ENABLE	`Boolean`	**`User-defined`**	Enables or disables Data logging in the BioHarness

The methods for this class are:

Method Name	Method purpose
EnableGP	Takes in a true or false Boolean argument to enable or disable General Packet

EnableECG	Takes in a true or false Boolean argument to enable or disable ECG Packet
EnableBreathing	Takes in a true or false Boolean argument to enable or disable Breathing Packet
EnableTtoR	Takes in a true or false Boolean argument to enable or disable R to R Packet
EnableAccelerometry	Takes in a true or false Boolean argument to enable or disable Accelerometer 100mg Packet
EnableSummary	Takes in a true or false Boolean argument to enable or disable Summary Data Packet
EnableEvent	Takes in a true or false Boolean argument to enable or disable Event Packet
EnableLogging	EnableLogging

3.8. The ReceivedEvent Class

The methods for this class are:

Method Name	Method purpose
getBytes	Returns the bytes associated with this event type.

3.9. The ReceivedListener Interface

The method defined for this interface is:

Method Name	Method purpose
Received	This method takes in argument of type ReceivedEvent and processes it based upon the object that implements this interface.

3.10. The ZephyrPacket Class

This class defines methods and variables for processing every received Packet from the BioHarness and creating objects and methods that check for packet sanctity such as valid Packet Start, Packet Length, Payload length, CRC etc. The important variables are listed as below:

Variable Name	Variable Class Name	Variable Class Type	Purpose
_buffer	`ByteArrayOutputStream`	**Java**	Stores the parsed and serialized data packet.
_crc8	`CRC8`	**User-defined**	Object to perform the CRC of the received data packet
_length	`Integer`	**Java**	Temporary variable to store the length of the received packet.
ACK	`Integer`	**Java**	ACK constant from the BioHarness
CRC8_POLY	`Integer`	**Java**	CRC constant
ETX	`Integer`	**Java**	End of Transmission Constant
MINIMUM_LENGTH	`Integer`	**Java**	Minimum Length of Packet Constant
NAK	`Integer`	**Java**	NAK constant from BioHarness Constant
STX	`Integer`	**Java**	Start of Transmission Constant

The methods defined for this class are:

Method Name	Method purpose
getSetLifeSignMessage	Returns a byte array with LifeSign message bytes stored.
getSetAccelerometerPacketMessage	Returns a byte array filled with Accelerometer Data Request Packet enabled/disabled.
getSetBreathingPacketMessage	Returns a byte array filled with Breathing Data Request Packet enabled/disabled.
getSetECGPacketMessage	Returns a byte array filled with ECG Data Request Packet enabled/disabled.
getSetGeneralPacketMessage	Returns a byte array filled with General Packet Data Request Packet enabled/disabled.
getSetRtoRPacketMessage	Returns a byte array filled with R to R Data Request Packet enabled/disabled.
getSetSerialNumberMessage	Returns a byte array filled with Serial Number Data Request Packet enabled/disabled.
getSetSummaryPacketMessage	Returns a byte array filled with Summary Packet Data Request Packet enabled/disabled.
getSetLoggingDataMessage	Returns a byte array filled with Enable or Disable logging Data Request Packet enabled/disabled.
Parse	Returns an object of type ZephyrPacketArgs upon success. Parses the received byte array and checks for accurate STX, ETX, payload length, CRC etc. Throws an exception if failure is seen.
Serialize	Reads from the input stream and returns a Vector of bytes.

3.11. The ZephyrPacketArgs Class

This class defines methods and variables for processing every received Packet from the BioHarness and storing certain information fields for each received packet

The important variables are listed as below:

Variable Name	Variable Class Name	Variable Class Type	Purpose
_bytes	`Byte Array`	**User-defined**	Stores the payload associated with each packet
_msgID	`Integer`	**User-defined**	Stores the Message ID of each received packet from the BioHarness
_status	`Byte`	**User-defined**	Stores the status of each message whether it passed or failed the checks per packet
_NumRcvdBytes	`Byte`	**User-defined**	Stores the number of bytes received in the payload for each packet
_CrcStatus	`Byte`	**User-defined**	Stores the CRC Pass/Fail status for each packet

The methods defined for this class are:

Method Name	Method purpose
getBytes	Returns the payload for each data packet in _bytes field of the class
getMsgID	Returns the Message ID for each data packet in _msgID field of the class
getStatus	Returns the Status for each data packet in _status field of the class
getNumRvcdBytes	Returns the Status for each data packet in the _NumRvcdBytes field of the class
getCRCStatus	Returns the CRC Status for each data packet in the _CRCStatus field of the class

3.12. The ZephyrPacketEvent Class

This class is used in conjunction to store the received packets from the BioHarness. The methods for this class are:

Method Name	Method purpose
getPacket	Returns the packet associated with this received packet event type in the _packet variable for this class.

3.13. The ZephyrPacketListener Interface

The method defined for this interface is:

Method Name	Method purpose
ReceivedPacket	This method takes in argument of type ZephyrPacketEvent and processes it based upon the object that implements this interface.

3.14. The ZephyrProtocol Class

This class is one that implements the ReceivedListener Interface. It is responsible for enabling and disabling messages from the BioHarness. The variables for this class are:

Variable Name	Variable Class Name	Variable Class Type	Purpose
_comms	BTComms	User-defined	Stores the BTComms object
_packet	ZephyrPacket	User-defined	Stores the ZephyrPacket object for obtaining the packets from the BH.
BREATHING_PACKET_ACK	Integer	Java	Stores the Breathing Packet Ack constant
RtoR_PACKET_ACK	Integer	Java	Stores the R to R Packet Ack constant
ACCELEROMETER_PACKET_ACK	Integer	Java	Stores the Accelerometer Packet Ack constant
SUMMARY_DATA_PACKET_ACK	Integer	Java	Stores the Summary Data Packet Ack constant
_GP_ENABLE	Boolean	Java	Variable to enable or disable the General Packet from the BioHarness
_ECG_ENABLE	Boolean	Java	Variable to enable or disable the ECG Packet from the BioHarness
_BREATHING_ENABLE	Boolean	Java	Variable to enable or disable the Breathing Packet from the BioHarness
_RtoR_ENABLE	Boolean	Java	Variable to enable or disable the R to R Packet from the BioHarness
_ACCELEROMETER_ENABLE	Boolean	Java	Variable to enable or disable the Accelerometer Packet from the BioHarness
_SUMMARY_DATA_ENABLE	Boolean	Java	Variable to enable or disable the Summary Data Packet from the BioHarness
eventSubscribers	Vector	Java/ User-defined	Stores a list of objects that implement the *ZephyrPacketListener* Interface

The methods defined for this class are:

Method Name	Method purpose
addZephyrPacketEventListener	Adds objects to the eventSubscribers list that implement the *ZephyrPacketListener* Interface
GetSerialNumber	Enables/Disables the Get Serial Number Message from the BioHarness
OnZephyrPacket	Takes in a ZephyrPacketArgs object and iterates through the eventSubscribers list of objects that implement the *ZephyrPacketListener* Interface and calls the ReceivedPacket Method for those objects.
Received	Takes in a ReceivedEvent object and calls the Methods to parse it. It then calls the OnZephyrPacket method.
removeZephyrPacketEventListener	Removes objects from the eventSubscribers list that implement the *ZephyrPacketListener* Interface
SendLifeSign	Sends a lifesign message to the BioHarness
SetAccelerometerPacket	Enables/Disables the Accelerometer Packet from the BioHarness
SetBreathingPacket	Enables/Disables the Breathing Packet from the BioHarness
SetECGPacket	Enables/Disables the ECG Packet from the BioHarness
SetGeneralPacket	Enables/Disables the General Packet from the BioHarness
SetRtoRPacket	Enables/Disables the R to R Packet from the BioHarness
SetSummaryDataPacket	Enables/Disables the Summary Data Packet from the BioHarness
SetLoggingDataPacket	Enables/Disables the Data Logging from the BioHarness

3.15. The GeneralPacketInfo Class

This class stores the data information from the General Packet and provides methods to parse the General Packet and obtain the data stored in the packet. For details of each variable to the ***General Communications Link Specification*** document. The variables for this class are:

Variable Name	Variable Class Name	Variable Class Type
_SequenceNum	Byte	Java
_TSYear	Int	Java
_TSMonth	Byte	Java
_TSDay	Byte	Java
_MsOfDay	Long	Java
_HeartRate	Int	Java
_RespirationRate	double	Java
_SkinTemperature	double	Java
_Posture	Int	Java
_VMU	double	Java
_PeakAcceleration	double	Java
_BatteryVoltage	double	Java
_BreathingWaveAmpl	double	Java
_ECGAmplitude	double	Java
_ECGNoise	double	Java
_XAxis_Accn_Min	double	Java
_XAxis_Accn_Peak	double	Java
_YAxis_Accn_Min	double	Java
_YAxis_Accn_Peak	double	Java
_ZAxis_Accn_Min	double	Java
_ZAxis_Accn_Peak	double	Java
_ZephyrSysChan	Int	Java
_GSR	Int	Java
_ROGStatus	Byte	Java
_AlarmSts	Byte	Java
_WornStatus	Byte	Java
_UserIntfBtnStatus	Byte	Java
_BHSigLowStatus	Byte	Java
_BHSensConnStatus	Byte	Java
_BatteryStatus	Byte	Java

The **Get**variablename methods provide access to each of the above mentioned variables.

3.16. The ECGPacketInfo Class

This class stores the data information from the ECG Data Packet and provides methods to parse the ECG Packet and obtain the data stored in the packet. For details of each variable to the ***General Communications Link Specification*** document. The variables for this class are:

Variable Name	Variable Class Name	Variable Class Type
_SequenceNum	Byte	Java
_TSYear	Int	Java
_TSMonth	Byte	Java
_TSDay	Byte	Java
_MsOfDay	Long	Java
NUM_ECG_SAMPLES_PER_PACKET	Short	Java

_ECGSamples	Array of short	Java

The **Get**variablename methods provide access to each of the above mentioned variables.

3.17. The BreathingPacketInfo class

This class stores the data information from the Breathing Data Packet and provides methods to parse the Breathing Packet and obtain the data stored in the packet. For details of each variable to the ***General Communications Link Specification*** document. The variables for this class are:

Variable Name	Variable Class Name	Variable Class Type
_SequenceNum	Byte	Java
_TSYear	Int	Java
_TSMonth	Byte	Java
_TSDay	Byte	Java
_MsOfDay	Long	Java
NUM_BREATHING_SAMPLES_PER_PACKET	Short	Java
_BreathingSamples	Array of short	Java

The **Get**variablename methods provide access to each of the above mentioned variables.

3.18. The RtoRPacketInfo class

This class stores the data information from the R to R Data Packet and provides methods to parse the R to R Packet and obtain the data stored in the packet. For details of each variable to the ***General Communications Link Specification*** document. The variables for this class are:

Variable Name	Variable Class Name	Variable Class Type
_SequenceNum	Byte	Java
_TSYear	Int	Java
_TSMonth	Byte	Java
_TSDay	Byte	Java
_MsOfDay	Long	Java
NUM_RtoR_SAMPLES_PER_PACKET	Short	Java
_RtoRSamples	Array of Int	Java

The **Get**variablename methods provide access to each of the above mentioned variables.

3.19. The AccelerometerPacketInfo class

This class stores the data information from the Accelerometer 100mg Data Packet and provides methods to parse the Accelerometer 100mg Packet and obtain the data stored in the packet. For details of each variable to the ***General Communications Link Specification*** document. The variables for this class are:

Variable Name	Variable Class Name	Variable Class Type
_SequenceNum	Byte	Java
_TSYear	Int	Java
_TSMonth	Byte	Java
_TSDay	Byte	Java

_MsOfDay	Long	Java
NUM_ACCN_SAMPLES	Short	Java
XYZ_AccnDataSamples	User-defined Inner Class XYZ_AccelerationData	User-defined

The XYZ_AccelerationData class

This is an Inner Class of the AccelerometerPacketInfo class. The variables for this class are:

Variable Name	Variable Class Name	Variable Class Type
X_axisAccnData	double	Java
Y_axisAccnData	double	Java
Z_axisAccnData	double	Java

The **Get**variablename methods provide access to each of the above mentioned variables.

3.20. The SummaryPacketInfo Class

This class stores the data information from the Summary Packet and provides methods to parse the Summary Packet and obtain the data stored in the packet. For details of each variable to the ***General Communications Link Specification*** document. The variables for this class are:

Variable Name	Variable Class Name	Variable Class Type
_SequenceNum	Byte	Java
_TSYear	Int	Java
_TSMonth	Byte	Java
_TSDay	Byte	Java
_MsOfDay	Long	Java
_HeartRate	Int	Java
_RespirationRate	double	Java
_SkinTemperature	double	Java
_Posture	Int	Java
_Activity	double	Java
_PeakAcceleration	double	Java
_BatteryVoltage	double	Java
_BatteryStatus	Byte	Java
_BreathingWaveAmpl	double	Java
_BreathingWaveNoise	double	Java
_BreathingRateConfidence	Byte	Java
_ECGAmplitude	double	Java
_ECGNoise	double	Java
_HeartRateConfidence	Byte	Java
_HRV	Int	Java
_SystemConfidence	Byte	Java
_GSR	Int	Java
_ROGStatus	Byte	Java
_ROGTime	short	Java
_Vertical_AxisAccnMin	double	Java
_Vertical_AxisAccnPeak	double	Java
_Lateral_AxisAccnMin	double	Java
_Lateral_AxisAccPeak	double	Java

_Sagittal_AxisAccnMin	double	Java
_Sagittal_AxisAccnPeak	double	Java
_Device_Internal_Temperature	double	Java
_Status_Worn_Det_Level	Byte	Java
_Status_Button_Press_Det_Flag	Byte	Java
_Status_Fitted_to_Garment_Flag	Byte	Java
_Status_Heart_Rate_Unreliable_Flag	Byte	Java
_LinkQuality	short	Java
_RSSI	Byte	Java
_TxPower	short	Java
_Reserved	short	Java

The **Get**variablename methods provide access to each of the above mentioned variables.

3.21. The EventPacketInfo Class

This class stores the data information from the Event Packet and provides methods to parse the Event Packet and obtain the data stored in the packet. For details of each variable to the ***General Communications Link Specification*** document. The variables for this class are:

Variable Name	Variable Class Name	Variable Class Type
_SequenceNum	Byte	Java
_TSYear	Int	Java
_TSMonth	Byte	Java
_TSDay	Byte	Java
_MsOfDay	Long	Java
_EventCode	short	Java
_EventSpecificData	Array of bytes	Java

The **Get**variablename methods provide access to each of the above mentioned variables. Since this is a variable length packet, the constructor to this class needs to be initialized with number of bytes required to store the EventSpecificData bytes. This can be calculated from the DLC of the packet.