Concept2

Performance Monitor
Bluetooth Smart Communications Interface Definition

Filename: Concept2 PM Bluetooth Smart Communication Interface Definition.doc

Revision: 1.25
8/29/17 11:19 AM

Concept2

Table of Contents

List of Figures

List of Tables

Purpose and Scope

This document contains the communications interface definition for devices communicating with Performance Monitor Generation 5s (PM5s) using the wireless Bluetooth Smart technology, also known as Bluetooth Low Energy (BTS). Information in this document combined with the documents referred to in Table 2 should provide the developer with sufficient information to create applications that communicate with the PM over the BTS interface.

Document History

Table 1 - Document Modification History

Edit Date	Engineer	Description of Modification
9/4/14	Andrew Dombek	Initial outline created with design concept for Mobile Device interfacing. V0.01
9/16/14	Kurt Preiss	Added section on BTS link layer. General edits. V0.02
9/24/14	Andrew Dombek	Edits to characteristic table. V0.03
10/8/14	Andrew Dombek	Minor edits. V0.04
10/9/14	Andrew Dombek	Added more characteristic data. V0.05
10/11/14	Andrew Dombek	Deleted Impulse Drive Force, added more characteristic data. V0.06
10/14/14	Andrew Dombek	More additions and edits to characteristic data. This version of the spec corresponds to PM firmware version 999 build 12, and ErgData V1.2.19. V0.07
10/15/14	Andrew Dombek	Added machine type attribute. Deleted rows about writing values to turn on notifications because it was causing confusion. V0.08.
10/15/14	Andrew Dombek	Moved machine type attribute to Device Status area. V0.09
10/16/14	Andrew Dombek	Added characteristics to split/interval characteristics. Added CSAFE names to parameters. V0.10
10/22/14	Kurt Preiss	Updated units of measurement in the split characteristics
11/6/14	Andrew Dombek	Added characteristic for heart rate belt information. V0.12
11/18/14	Andrew Dombek	Added stroke data characteristics for Projected Work Distance and Projected Work Time. V0.13
2/4/15	Andrew Dombek	General clean up and additions to Appendix. First official release of specification, coinciding with PM firmware V17, PM Ski V717. V1.0
2/6/15	Andrew Dombek	Fix units in Workout Summary Characteristics. V1.01
2/8/2015	Scott Hamilton	Minor wording changes, unpair from PM5, references to Polar. V1.02
2/18/2015	Kurt Preiss	Added multiplexing concept. V1.03
2/20/2015	Andrew Dombek	Major rework to accommodate multiplexing. V1.04
2/25/2015	Kurt Preiss	Tweaks to multiplexed characteristic. V1.06
2/26/2015	Andrew Dombek	Deleted Model Number Characteristic, Added back Manufacturer Characteristic. V1.07
9/7/2016	Kurt Preiss	Added NFC NDEF record description for BLE pairing
1/31/2017	Andrew Dombek	Added Erg Machine Type and Workout Verified information. V1.20
2/3/2017	Andrew Dombek	Deleted Machine Type information in Device Info Service as firmware unable to support it. V1.21.
4/19/2017	Mark Lyons	Added Force Curve characteristic definition. V1.22.
4/25/2017	Andrew Dombek	Modified Force Curve characteristic definition. V1.23.
5/25/2017	Andrew Dombek	Added new machine types. V1.24.
8/29/2017	Andrew Dombek	Added model number and machine type. V1.25.

Related Documents

Table 2 - Related Documents

Document Title	Document Number - Date
CSAFE Protocol Technical Specification, V1.x	http://www.fitlinxx.com/csafe/
Getting Started with Bluetooth Low Energy.doc	
Concept2 PM Communications Interface Definition.doc	Rev 0.15 8/23/2010
Bluetooth Specification Version 4.1	Rev 4.1 12/3/2013
Android guide	https://developer.android.com/guide/topics/connectivity/bluetooth-le.html

Overview

PM5 devices are equipped with a Bluetooth low energy module that provides short range RF communications capability. The PM5 utilizes BTS for data transfer with mobile devices as well as configuration and control of the PM5 by mobile devices.

This document describes a proprietary Bluetooth profile utilized by the PM5 to provide a data and control interface with mobile devices. The profile defines the services and characteristics available to the mobile device developer.

This document assumes the reader has sufficient knowledge of the Bluetooth specification.

In Bluetooth terminology, the PM5 assumes the Peripheral role and the mobile device assumes the Central role.

Mobile Device Interface

The following BTS wireless scenarios are supported by the PM5:
1. **Single PM5 To Single Mobile Device** - A PM sends realtime data to a BTS-enabled mobile device.   The mobile device can also setup and start workouts on the PM. An example application is a specialized smartphone application for the visually impaired.

2. **Multiple PM5s To Single Mobile Device** – Multiple PMs send realtime data to a BTS-enabled mobile device. Example mobile device applications are coaching tools and racing controllers.

3. **[future] Multiple PM5s to single PM5 –** may be used for Wireless Racing (see existing PM4 ANT Wireless racing) or for Venue Race cut-wire backup.

The three key operational areas common to the scenarios defined above are Discovery, Enumeration and Data/Control. These are described for each scenario in the following sections.

Single PM5 To Single Mobile Device

Discovery
When an application running on a mobile device wants to communicate with a particular PM using BTS, it needs to uniquely identify the desired device before establishing the communication link. Since there may be other BTS devices transmitting, this involves a step in which the PM user selects and confirms the correct Mobile Device.

PM Logic – Not Previously Paired, or Reset Pairing
If the PM has not previously been paired with a device, or the user wants to pair with a different device, then the user must use the Main Menu and select "More Options" then "Turn Wireless ON". This will bring the user to the "Connect Device" screen on the PM to enable BTS and enter pairing mode. Once on that screen, the PM
   A) displays it's 'friendly ID number/string' to the user and
   B) broadcasts BTS data, advertising the services and characteristics it has to offer.

The mobile device listens for the BTS advertising information and completes the connection per the Bluetooth specification. The Mobile device may choose to pair based on 'very close' proximity of the PM5 BTS signal, or based on a user choice of which advertising PM5 to connect to. The mobile device application should use the proprietary C2 base UUID, advertised in a Scan Response packet, to filter the selection of BTS devices available to connect to.

PM Logic – Unpairing
The PM will unpair from the mobile device when it powers down, or when the Mobile Device signals to end the session. To Unpair from the PM5, select More Options, Turn Wireless OFF. With older firmware, this may be still labeled "Wireless ON".

Enumeration

Enumeration is not necessary in this use case with a single PM connected to a single Mobile Device.

Data/Control

Data transfer and PM control occurs using C2 proprietary BTS services. A C2 PM Control service is utilized to send CSAFE commands and receive CSAFE responses. This service is typically used to set up workouts on the PM or to retrieve workout data. The C2 Rowing Service is utilized to enable/disable broadcast of PM data at various data rates. These services are defined in the following sections.

Multiple PMs To Single Mobile Device

Discovery

Discovery of multiple PMs is based upon the process decribed in the single PM case defined in the previous section. In this scenario one PM at a time is put in the "Connect Device" screen to perform the pairing function, until all PM's have been paired.

Enumeration

Devices are enumerated using additional transactions after the Discovery/Pairing process has been completed. This involves sending additional CSAFE commands to the PM (TBD).

Data/Control

Receiving data from and sending commands to PMs works the same as with the single PM scenario.

Bluetooth Low Energy Link Layer

Bluetooth Smart is a low bandwidth interface capable of achieving speeds upto 16kBytes per second under ideal conditions. The protocol supports a single master and up to 8 slaves. For the single PM5 to Mobile device model, the PM5 will act as a slave to the mobile device master.   Below is a summary of how packets are transferred over the air between a master and slave.

- Once in a Connected state, either the master OR slave can terminate the connection.
- In a connected state, the rf link is broken up into connection events. One connection event occurs per connection interval. The premise of the connection event is to allow master and slave devices to exchange data then go back to sleep as quickly as possible to conserve power. As the connection interval decreases, the data throughput increases. The minimum connection interval is OS dependent. On Android, the minimum interval is 7.5msec and on iOS, the minimum is around 30msec.
- The slave can request the master to change the connection interval.
- The master and slave are synchronized in time when the connected state is established and rely on the time synchronization to wakeup, transmit and receive.

- Within a connection event, there are back-to-back time slots dedicated for transmission and reception of data packets. The master transmits in the first time slot and listens in the second. The slave listens in the first time slot and transmits in the second. The time between two consecutive packets is defined as the Inter Frame Space (T_IFS) time. It is specified to be 150usec. Even if the master has no data to send, it will still transmit a NULL packet to the slave allowing the slave to respond with data it may want to transmit.
- The data packets transmitted by both master and slave are acknowledged. A slave device upon receiving a packet from the master must ALWAYS send a response back to the master. There are 2 bits in EVERY data packet containing ACK information adjusted appropriately by both master and slave upon packet reception. If the master does not receive a data packet from the slave or determines the received data packet is bad (including ACK bits), it will close the connection event and retransmit it's previously transmitted packet at the next connection interval. This guarantees master and slave acknowledgement.
- A supervision timeout determines if the connection is good. Both the master and slave are aware of the timeout value. The supervision timer is reset whenever a valid packet is received. If the timer elapses, the master/slave issues a Disconnect event to the application layer and the radio returns to an unconnected state.
- Each data packet can contain up to 20 bytes of data. A data packet will be smaller if fewer data bytes are transmitted.
- Within a single connection event, multiple data packets may be transmitted by master or slave. Up to 6 data packets may be transmitted within a single connection event. This too appears to be platform dependent. Android allows 4 packets per event while iOS allows 6 packets. There is a bit (More Data) in every transmitted data packet indicating to the receiver if more packets should follow.
- The current Nordic S120 softstack used for communicating with BTS slaves, only allows one 20 byte packet per connection event. A packet is dedicated per connection within a single connection event. This means that the minimum connection interval cannot be 7.5msec if the master has made connections since 1.25msec/packet x 8 > 7.5msec. Below is a table of minimum connection intervals based upon the number of connected slaves.
-

Figure 1 – Minimum Connection Intervals

Protocol	Role	Method	Number of connected slaves	Interval (ms)	Maximum data throughput
GATT	Client	Receive Notification	1 - 8	20	7.8 kbps
			1 - 8	50	3.1 kbps
		Send Write command	1 - 8	20	7.8 kbps
			1 - 8	50	3.1 kbps
		Send Write request	1 - 8	20	3.8 kbps
			1 - 8	50	1.5 kbps
		Simultaneous receive Notification and send Write command	1	7.5	21 kbps (each direction)
			1 - 8	20	7.8 kbps (each direction)
			1 - 8	50	3.1 kbps (each direction)
GATT	Server	Send Notification	1 - 8	20	7.8 kbps
			1 - 8	50	3.1 kbps
		Receive Write command	1 - 8	20	7.8 kbps
			1 - 8	50	3.1 kbps
		Receive Write request	1 - 8	20	3.9 kbps
			1 - 8	50	1.5 kbps
		Simultaneous send Notification and receive Write command	1	7.5	21 kbps (each direction)
			1 - 8	20	6.7 kbps (each direction)
			1 - 8	50	3.1 kbps (each direction)

Concept2 PM Bluetooth Profile

Overview

The Concept2 PM Bluetooth Profile consists of three proprietary BTS services for device discovery, control and data transfer. These services are all based on the Generic Attribute Profile (GATT). GATT provides standard interfaces for discovering, reading, writing and indicating of service characteristics and attributes.

Supported Mobile Platforms

iPhone 4S and above (and similar class iPad), iOS7. Samsung S4 and above and similarly enabled phones/tablets; Nexus7 and MotoX.

References: According to developer.android.com, the minimum Android is 4.3 (API Level 18).

C2 PM Device Discovery

A mobile device uses the PM's unique 128-bit Bluetooth peripheral Universally Unique Identifier (UUID) to discover the PM. This UUID is specified as Version 1 by the Network Working Group specification RFC 4122. It is based upon the time of day and the MAC address of the computer upon which it was generated. The PM's UUID is CE06*xxxx*-43E5-11E4-916C-0800200C9A66, where xxxx is a 16-bit value used to identify the specific service or characteristic. The base UUID of the PM is CE060000 -43E5-11E4-916C-0800200C9A66.

C2 PM Device Information Service

The C2 PM Device Information Service provides model and version information. See Table 3 for details.

C2 PM Control Service

The C2 PM Control Service allows the mobile device to send CSAFE commands and receive CSAFE responses. See Table 3 for details.

C2 PM Rowing Service

The C2 PM Rowing Service provides broadcast of real time rowing data. Each characteristic contains multiple status bytes packed in an array of data. Each characteristic can also be enabled/disabled for broadcast, and the broadcast rate can be set. See Table 3 for details.

On some Android platforms, there is a limitation to the number of notification messages allowed. In Android 4.4, the limit is 7 and in Android 4.3 the limit is 4. To circumvent this issue, a single characteristic (C2 multiplexed data info) exists to allow multiple characteristics to be multiplexed onto a single characteristic. The last byte in the characteristic will indicate which data characteristic is multiplexed. Android applications should enable this notification in lieu of the following UUIDs; 0x31, 0x32, 0x33, 0x35, 0x36, 0x37, 0x38, 0x39, 0x3A, and 0x3B.

Table 3 – C2 PM BTS Peripheral : Attribute Table

C2 PM BTS Peripheral : Attribute Table

C2 PM Base UUID : CE06XXXX-43E5-11E4-916C-0800200C9A66

UUID	Type	Value	GATT Server Permissions	Notes
0x1800	GAP primary service	GAP_SERVICE_UUID	READ	Start of GAP Service (Mandatory)
0x2A00	GAP device name characteristic	"PM5 430000000" where 430000000 is the actual PM5 serial number.	READ	Device name characteristic value
0x2A01	GAP appearance characteristic	0x0000	READ	Appearance characteristic value
0x2A02	GAP peripheral privacy characteristic	0x00 (GAP_PRIVACY_DISABLED)	READ/WRITE	Peripheral privacy characteristic value
0x2A03	GAP reconnect address characteristic	00:00:00:00:00:00	READ/WRITE	Reconnection address characteristic value
0x2A04	Peripheral preferred connection parameters characteristic	0x0018 (30ms preferred min connection interval) 0x0018 (30ms preferred max connection interval) 0x0000 (0 preferred slave latency) 0x03E8 (10000ms preferred supervision timeout)	READ	Peripheral preferred connection parameters characteristic value
0x1801	GATT primary service	GATT_SERVICE_UUID	READ	Start of GATT Service (Mandatory)
0x2A05	Service changed characteristic	(null)	(none)	Service changed characteristic value
0x2902	GATT client configuration characteristic	00:00 (2 bytes)	READ/WRITE	Write 01:00 to enable notifications, 00:00 to disable
0x0010	C2 device information primary service	C2_DEVINFO_SERVICE_UUID	READ	Start of C2 Device Information Service

C2 PM BTS Peripheral : Attribute Table

C2 PM Base UUID : CE06**XXXX**-43E5-11E4-916C-0800200C9A66

UUID	Type	Value	GATT Server Permissions	Notes
0x0011	C2 module number string characteristic	(Model Number, "PM5") (16 bytes)	READ	Model number string (Valid for PM5 V150 – V299.99 only)
0x0012	C2 serial number string characteristic	(Serial Number) (9 bytes)	READ	Serial number string
0x0013	C2 hardware revision string characteristic	(Hardware Revision) (3 bytes)	READ	Hardware revision string
0x0014	C2 firmware revision string characteristic	(Firmware Revision) (20 bytes)	READ	Firmware revision string
0x0015	C2 manufacturer name string characteristic	"Concept2" (16 bytes)	READ	Manufacturer name string
0x0016	Erg Machine Type characteristic	(Erg Machine Type) (1 byte)	READ	Erg Machine Type enumerated value.[1] (Valid for PM5 V150 – V299.99 only)
0x0020	C2 PM control primary service	C2_PM_CONTROL_SERVICE_UUID	READ	Start of C2 PM Control Primary Service
0x0021	C2 PM receive characteristic	(Up to 20 bytes)	WRITE	Control command in the form of a CSAFE frame sent to PM.[2]
0x0022	C2 PM transmit characteristic	(Up to 20 bytes)	READ	Response to command in the form of a CSAFE frame from the PM.
0x0030	C2 rowing primary service	C2_PM_CONTROL_SERVICE_UUID	READ	Start of C2 Rowing Service

C2 PM BTS Peripheral : Attribute Table				
C2 PM Base UUID : CE06XXXX-43E5-11E4-916C-0800200C9A66				
UUID	**Type**	**Value**	**GATT Server Permissions**	**Notes**
0x0031	C2 rowing general status characteristic	(19 bytes)	READ	*Data bytes packed as follows:* Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Distance Lo (0.1 m lsb), Distance Mid, Distance High, Workout Type [3](enum), CSAFE_PM_GET_WORKOUTTYPE[4] Interval Type[5] (enum), CSAFE_PM_GET_INTERVALTYPE Workout State (enum), CSAFE_PM_GET_WORKOUTSTATE Rowing State (enum), CSAFE_PM_GET_ROWINGSTATE Stroke State (enum), CSAFE_PM_GET_STROKESTATE Total Work Distance Lo, CSAFE_PM_GET_WORKDISTANCE Total Work Distance Mid, Total Work Distance Hi, Workout Duration Lo (if time, 0.01 sec lsb), CSAFE_PM_GET_WORKOUTDURATION Workout Duration Mid, Workout Duration Hi, Workout Duration Type (enum), CSAFE_PM_GET_WORKOUTDURATION Drag Factor CSAFE_PM_GET_DRAGFACTOR

---

[1] See Appendix for enumerated values
[2] See Appendix for additional information on CSAFE commands
[3] See Appendix for enumerated values definitions
[4] For reference - The named CSAFE command returns the same value
[5] This value will change depending on where you are in the interval (work, rest, etc). Use workout type to determine whether the intervals are time or distance intervals.

C2 PM BTS Peripheral : Attribute Table				
C2 PM Base UUID : CE06XXXX-43E5-11E4-916C-0800200C9A66				
UUID	**Type**	**Value**	**GATT Server Permissions**	**Notes**
0x0032	C2 rowing additional status 1 characteristic	(16 bytes)	READ	*Data bytes packed as follows:* Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Speed Lo (0.001m/s lsb), CSAFE_GETSPEED_CMD[6] Speed Hi, Stroke Rate (strokes/min), CSAFE_PM_GET_STROKERATE Heartrate (bpm, 255=invalid), CSAFE_PM_GET_AVG_HEARTRATE Current Pace Lo (0.01 sec lsb), CSAFE_PM_GET_STROKE_500MPACE Current Pace Hi, Average Pace Lo (0.01 sec lsb), CSAFE_PM_GET_TOTAL_AVG_500MPACE Average Pace Hi, Rest Distance Lo, CSAFE_PM_GET_RESTDISTANCE Rest Distance Hi, Rest Time Lo, (0.01 sec lsb) CSAFE_PM_GET_RESTTIME Rest Time Mid, Rest Time Hi

C2 PM BTS Peripheral : Attribute Table				
C2 PM Base UUID : CE06XXXX-43E5-11E4-916C-0800200C9A66				
UUID	**Type**	**Value**	**GATT Server Permissions**	**Notes**
0x0033	C2 rowing additional status 2 characteristic	(20 bytes)	READ	*Data bytes packed as follows:* Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Interval Count, CSAFE_PM_GET_WORKOUTINTERVALCOUNT[7] Average Power Lo, CSAFE_PM_GET_TOTAL_AVG_POWER Average Power Hi, Total Calories Lo (cals), CSAFE_PM_GET_TOTAL_AVG_CALORIES Total Calories Hi, Split/Int Avg Pace Lo (0.01 sec lsb), CSAFE_PM_GET_SPLIT_AVG_500MPACE Split/Int Avg Pace Hi, Split/Int Avg Power Lo (watts), CSAFE_PM_GET_SPLIT_AVG_POWER Split/Int Avg Power Hi, Split/Int Avg Calories Lo (cals/hr), CSAFE_PM_GET_SPLIT_AVG_CALORIES Split/Interval Avg Calories Hi, Last Split Time Lo (0.1 sec lsb), CSAFE_PM_GET_LAST_SPLITTIME Last Split Time Mid, Last Split Time High, Last Split Distance Lo, CSAFE_PM_GET_LAST_SPLITDISTANCE (in meters) Last Split Distance Mid, Last Split Distance Hi

---

[6] For reference - The named CSAFE command returns the same value
[7] For reference - The named CSAFE command returns the same value

C2 PM BTS Peripheral : Attribute Table				
C2 PM Base UUID : CE06**XXXX**-43E5-11E4-916C-0800200C9A66				
UUID	**Type**	**Value**	**GATT Server Permissions**	**Notes**
0x0034	C2 rowing general status and additional status sample rate characteristic	(1 byte)	WRITE/Read	Determines how often slave sends general status and additional status data as notifications. Set rate as follows: 0 – 1 sec 1 - 500ms (default if characteristic is not explicitly set by the app) 2 – 250ms 3 – 100ms

		C2 PM BTS Peripheral : Attribute Table		
		C2 PM Base UUID : CE06**XXXX**-43E5-11E4-916C-0800200C9A66		
UUID	**Type**	**Value**	**GATT Server Permissions**	**Notes**
0x0035	C2 rowing stroke data characteristic	(20 bytes)	READ	*Data bytes packed as follows:* Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Distance Lo (0.1 m lsb), Distance Mid, Distance High, Drive Length (0.01 meters, max = 2.55m), CSAFE_PM_GET_STROKESTATS Drive Time (0.01 sec, max = 2.55 sec), Stroke Recovery Time Lo (0.01 sec, max = 655.35 sec), CSAFE_PM_GET_STROKESTATS Stroke Recovery Time Hi, CSAFE_PM_GET_STROKESTATS[8] Stroke Distance Lo (0.01 m, max=655.35m), CSAFE_PM_GET_STROKESTATS Stroke Distance Hi, Peak Drive Force Lo (0.1 lbs of force, max=6553.5m), CSAFE_PM_GET_STROKESTATS Peak Drive Force Hi, Average Drive Force Lo (0.1 lbs of force, max=6553.5m), CSAFE_PM_GET_STROKESTATS Average Drive Force Hi, Work Per Stroke Lo (0.1 Joules, max=6553.5 Joules), CSAFE_PM_GET_STROKESTATS Work Per Stroke Hi Stroke Count Lo, CSAFE_PM_GET_STROKESTATS Stroke Count Hi,

<table>
<tr><td colspan="5" align="center"><h1>C2 PM BTS Peripheral : Attribute Table</h1></td></tr>
<tr><td colspan="5" align="center">C2 PM Base UUID : CE06XXXX-43E5-11E4-916C-0800200C9A66</td></tr>
<tr>
<th>UUID</th>
<th>Type</th>
<th>Value</th>
<th>GATT Server Permissions</th>
<th>Notes</th>
</tr>
<tr>
<td>0x0036</td>
<td>C2 rowing additional stroke data characteristic</td>
<td>(15 bytes)</td>
<td>READ</td>
<td><i>Data bytes packed as follows:</i>
Elapsed Time Lo (0.01 sec lsb),
Elapsed Time Mid,
Elapsed Time High,
Stroke Power Lo (watts),
<small>CSAFE_PM_GET_STROKE_POWER</small>
Stroke Power Hi,
Stroke Calories Lo (cal/hr),
<small>CSAFE_PM_GET_STROKE_CALORICBURNRATE</small>
Stroke Calories Hi,
Stroke Count Lo, <small>CSAFE_PM_GET_STROKESTATS</small>
Stroke Count Hi,
Projected Work Time Lo (secs),
Projected Work Time Mid,
Projected Work Time Hi,
Projected Work Distance Lo (meters),
Projected Work Distance Mid,
Projected Work Distance Hi</td>
</tr>
</table>

---

[8] For reference - The named CSAFE command returns the same value

		C2 PM BTS Peripheral : Attribute Table		
		C2 PM Base UUID : CE06**XXXX**-43E5-11E4-916C-0800200C9A66		
UUID	**Type**	**Value**	**GATT Server Permissions**	**Notes**
0x0037	C2 rowing split/interval data characteristic	(18 bytes)	READ	*Data bytes packed as follows:* Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Distance Lo (0.1 m lsb), Distance Mid, Distance High, Split/Interval Time Lo (0.1 sec lsb), Split/Interval Time Mid, Split/Interval Time High, Split/Interval Distance Lo (1m lsb), Split/Interval Distance Mid, Split/Interval Distance High, Interval Rest Time Lo (1 sec lsb), Interval Rest Time Hi, Interval Rest Distance Lo (1m lsb), Interval Rest Distance Hi Split/Interval Type[9], Split/Interval Number,

		C2 PM BTS Peripheral : Attribute Table		
		C2 PM Base UUID : CE06**XXXX**-43E5-11E4-916C-0800200C9A66		
UUID	**Type**	**Value**	**GATT Server Permissions**	**Notes**
0x0038	C2 rowing additional split/interval data characteristic	(18 bytes)	READ	*Data bytes packed as follows:* Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Split/Interval Avg Stroke Rate, Split/Interval Work Heartrate, Split/Interval Rest Heartrate, Split/Interval Avg Pace Lo (0.1 sec lsb) Split/Interval Avg Pace Hi, Split/Interval Total Calories Lo (Cals), Split/Interval Total Calories Hi, Split/Interval Avg Calories Lo (Cals/Hr), Split/Interval Avg Calories Hi, Split/Interval Speed Lo (0.001 m/s, max=65.534 m/s) Split/Interval Speed Hi, Split/Interval Power Lo (Watts, max = 65.534 kW) Split/Interval Power Hi Split Avg Drag Factor, Split/Interval Number

---

[9] This value will change depending on where you are in the interval (work, rest, etc). Use workout type to determine whether the intervals are time or distance intervals

C2 PM BTS Peripheral : Attribute Table

C2 PM Base UUID : CE06**XXXX**-43E5-11E4-916C-0800200C9A66				
UUID	**Type**	**Value**	**GATT Server Permissions**	**Notes**
0x0039	C2 rowing end of workout summary data characteristic	(20 bytes)	READ	*Data bytes packed as follows:* Log Entry Date Lo, Log Entry Date Hi, Log Entry Time Lo, Log Entry Time Hi, Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Distance Lo (0.1 m lsb), Distance Mid, Distance High, Average Stroke Rate, Ending Heartrate, Average Heartrate, Min Heartrate, Max Heartrate, Drag Factor Average, Recovery Heart Rate, (zero = not valid data. After 1 minute of rest/recovery, PM5 sends this data as a revised End Of Workout summary data characteristic unless the monitor has been turned off or a new workout started) Workout Type, Avg Pace Lo (0.1 sec lsb) Avg Pace Hi

	C2 PM BTS Peripheral : Attribute Table			
	C2 PM Base UUID : CE06**XXXX**-43E5-11E4-916C-0800200C9A66			
UUID	**Type**	**Value**	**GATT Server Permissions**	**Notes**
0x003A	C2 rowing end of workout additional summary data characteristic	(19 bytes)	READ	*Data bytes packed as follows:* Log Entry Date Lo, Log Entry Date Hi, Log Entry Time Lo, Log Entry Time Hi, Split/Interval Type[10], Split/Interval Size Lo, (meters or seconds) Split/Interval Size Hi, Split/Interval Count, Total Calories Lo, Total Calories Hi, Watts Lo, Watts Hi, Total Rest Distance Lo (1 m lsb), Total Rest Distance Mid, Total Rest Distance High Interval Rest Time Lo (seconds), Interval Rest Time Hi, Avg Calories Lo, (cals/hr) Avg Calories Hi,
0x003B	C2 rowing heart rate belt information characteristic	(6 bytes)	WRITE/Read	Manufacturer ID, Device Type, Belt ID Lo, Belt ID Mid Lo, Betl ID Mid Hi, Belt ID Hi

C2 PM BTS Peripheral : Attribute Table

C2 PM Base UUID : CE06XXXX-43E5-11E4-916C-0800200C9A66

UUID	Type	Value	GATT Server Permissions	Notes
0x003D	C2 force curve data characteristic	(2 - 288 bytes separated into multiple successive notifications)	WRITE/Read	MS Nib = # characteristics, LS Nib = # words,[11] Sequence number, Data[n] (LS), Data[n+1] (MS), Data[n+2] (LS), Data[n+3] (MS), Data[n+4] (LS), Data[n+5] (MS), Data[n+6] (LS), Data[n+7] (MS), Data[n+8] (LS), Data[n+9] (MS), Data[n+10] (LS), Data[n+11] (MS), Data[n+12] (LS), Data[n+13] (MS), Data[n+14] (LS), Data[n+15] (MS), Data[n+16] (LS), Data[n+17] (MS)

---

[10] This value will change depending on where you are in the interval when the workout is terminated. Use workout type to determine whether the intervals are time or distance intervals.

[11] MS Nibble = Total number of characteristics for this force curve, LS Nibble = Number of 16-bit data points in the current characteristic

C2 PM BTS Peripheral : Attribute Table

C2 PM Base UUID : CE06XXXX-43E5-11E4-916C-0800200C9A66

UUID	Type	Value	GATT Server Permissions	Notes
0x0080	C2 multiplexed information characteristic	(Up to 20 bytes)	READ	The multiplexed information characteristic consists of an identification byte and up to 19 data bytes. The first byte identifies the payload as defined in the Data Definitions table in the following section. **Important note: The following identifiers will ONLY be multiplexed on this characteristic as long as the respective characteristic notification of the same ID is NOT enabled. Note: The byte length of the following multiplexed characteristics does not include the identifier byte. The total length of the data packet is N+1 bytes. 0x31 0x32 0x33 0x35 0x36 0x37 0x38 0x39 0x3A 0x3B 0x3C

Table 4 – C2 Multiplexed Information: Data Definitions

C2 Multiplexed Information: Data Definitions

ID	Name	Byte Length	Definitions
0x0031	C2 rowing general status	(19 bytes)	*Data bytes packed as follows:* Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Distance Lo (0.1 m lsb), Distance Mid, Distance High, Workout Type [12](enum), CSAFE_PM_GET_WORKOUTTYPE[13] Interval Type[14] (enum), CSAFE_PM_GET_INTERVALTYPE Workout State (enum), CSAFE_PM_GET_WORKOUTSTATE Rowing State (enum), CSAFE_PM_GET_ROWINGSTATE Stroke State (enum), CSAFE_PM_GET_STROKESTATE Total Work Distance Lo, CSAFE_PM_GET_WORKDISTANCE Total Work Distance Mid, Total Work Distance Hi, Workout Duration Lo (if time, 0.01 sec lsb), CSAFE_PM_GET_WORKOUTDURATION Workout Duration Mid, Workout Duration Hi, Workout Duration Type (enum), CSAFE_PM_GET_WORKOUTDURATION Drag Factor CSAFE_PM_GET_DRAGFACTOR

text]

23
23

[12] See Appendix for enumerated values definitions

C2 Multiplexed Information: Data Definitions

ID	Name	Byte Length	Definitions
0x0032	C2 rowing additional status 1	(18 bytes)	*Data bytes packed as follows:* Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Speed Lo (0.001m/s lsb), CSAFE_GETSPEED_CMD[15] Speed Hi, Stroke Rate (strokes/min), CSAFE_PM_GET_STROKERATE Heartrate (bpm, 255=invalid), CSAFE_PM_GET_AVG_HEARTRATE Current Pace Lo (0.01 sec lsb), CSAFE_PM_GET_STROKE_500MPACE Current Pace Hi, Average Pace Lo (0.01 sec lsb), CSAFE_PM_GET_TOTAL_AVG_500MPACE Average Pace Hi, Rest Distance Lo, CSAFE_PM_GET_RESTDISTANCE Rest Distance Hi, Rest Time Lo, (0.01 sec lsb) CSAFE_PM_GET_RESTTIME Rest Time Mid, Rest Time Hi, Average Power Lo, CSAFE_PM_GET_TOTAL_AVG_POWER Average Power Hi

---

[13] For reference - The named CSAFE command returns the same value

[14] This value will change depending on where you are in the interval (work, rest, etc). Use workout type to determine whether the intervals are time or distance intervals.

[15] For reference - The named CSAFE command returns the same value

C2 Multiplexed Information: Data Definitions

ID	Name	Byte Length	Definitions
0x0033	C2 rowing additional status 2	(18 bytes)	*Data bytes packed as follows:* Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Interval Count, CSAFE_PM_GET_WORKOUTINTERVALCOUNT[16] Total Calories Lo (cals), CSAFE_PM_GET_TOTAL_AVG_CALORIES Total Calories Hi, Split/Int Avg Pace Lo (0.01 sec lsb), CSAFE_PM_GET_SPLIT_AVG_500MPACE Split/Int Avg Pace Hi, Split/Int Avg Power Lo (watts), CSAFE_PM_GET_SPLIT_AVG_POWER Split/Int Avg Power Hi, Split/Int Avg Calories Lo (cals/hr), CSAFE_PM_GET_SPLIT_AVG_CALORIES Split/Interval Avg Calories Hi, Last Split Time Lo (0.1 sec lsb), CSAFE_PM_GET_LAST_SPLITTIME Last Split Time Mid, Last Split Time High, Last Split Distance Lo, CSAFE_PM_GET_LAST_SPLITDISTANCE (in meters) Last Split Distance Mid, Last Split Distance Hi
0x0034	Not used		

C2 Multiplexed Information: Data Definitions

ID	Name	Byte Length	Definitions
0x0035	C2 rowing stroke data	(18 bytes)	*Data bytes packed as follows:* Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Distance Lo (0.1 m lsb), Distance Mid, Distance High, Drive Length (0.01 meters, max = 2.55m), CSAFE_PM_GET_STROKESTATS Drive Time (0.01 sec, max = 2.55 sec), Stroke Recovery Time Lo (0.01 sec, max = 655.35 sec), CSAFE_PM_GET_STROKESTATS Stroke Recovery Time Hi, CSAFE_PM_GET_STROKESTATS[17] Stroke Distance Lo (0.01 m, max=655.35m), CSAFE_PM_GET_STROKESTATS Stroke Distance Hi, Peak Drive Force Lo (0.1 lbs of force, max=6553.5m), CSAFE_PM_GET_STROKESTATS Peak Drive Force Hi, Average Drive Force Lo (0.1 lbs of force, max=6553.5m), CSAFE_PM_GET_STROKESTATS Average Drive Force Hi, Stroke Count Lo, CSAFE_PM_GET_STROKESTATS Stroke Count Hi,

---

[16] For reference - The named CSAFE command returns the same value
[17] For reference - The named CSAFE command returns the same value

	C2 Multiplexed Information: Data Definitions		
ID	**Name**	**Byte Length**	**Definitions**
0x0036	C2 rowing additional stroke data	(17 bytes)	*Data bytes packed as follows:* Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Stroke Power Lo (watts), CSAFE_PM_GET_STROKE_POWER Stroke Power Hi, Stroke Calories Lo (cal/hr), CSAFE_PM_GET_STROKE_CALORICBURNRATE Stroke Calories Hi, Stroke Count Lo, CSAFE_PM_GET_STROKESTATS Stroke Count Hi, Projected Work Time Lo (secs), Projected Work Time Mid, Projected Work Time Hi, Projected Work Distance Lo (meters), Projected Work Distance Mid, Projected Work Distance Hi, Work Per Stroke Lo (0.1 Joules, max=6553.5 Joules), CSAFE_PM_GET_STROKESTATS Work Per Stroke Hi

C2 Multiplexed Information: Data Definitions

ID	Name	Byte Length	Definitions
0x0037	C2 rowing split/interval data	(18 bytes)	*Data bytes packed as follows:* Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Distance Lo (0.1 m lsb), Distance Mid, Distance High, Split/Interval Time Lo (0.1 sec lsb), Split/Interval Time Mid, Split/Interval Time High, Split/Interval Distance Lo (1m lsb), Split/Interval Distance Mid, Split/Interval Distance High, Interval Rest Time Lo (1 sec lsb), Interval Rest Time Hi, Interval Rest Distance Lo (1m lsb), Interval Rest Distance Hi Split/Interval Type[18], Split/Interval Number,

C2 Multiplexed Information: Data Definitions

ID	Name	Byte Length	Definitions
0x0038	C2 rowing additional split/interval data	(18 bytes)	*Data bytes packed as follows:* Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Split/Interval Avg Stroke Rate, Split/Interval Work Heartrate, Split/Interval Rest Heartrate, Split/Interval Avg Pace Lo (0.1 sec lsb) Split/Interval Avg Pace Hi, Split/Interval Total Calories Lo (Cals), Split/Interval Total Calories Hi, Split/Interval Avg Calories Lo (Cals/Hr), Split/Interval Avg Calories Hi, Split/Interval Speed Lo (0.001 m/s, max=65.534 m/s) Split/Interval Speed Hi, Split/Interval Power Lo (Watts, max = 65.534 kW) Split/Interval Power Hi Split Avg Drag Factor, Split/Interval Number

---

[18] This value will change depending on where you are in the interval (work, rest, etc). Use workout type to determine whether the intervals are time or distance intervals

C2 Multiplexed Information: Data Definitions

ID	Name	Byte Length	Definitions
0x0039	C2 rowing end of workout summary data characteristic	(18 bytes)	*Data bytes packed as follows:* Log Entry Date Lo, Log Entry Date Hi, Log Entry Time Lo, Log Entry Time Hi, Elapsed Time Lo (0.01 sec lsb), Elapsed Time Mid, Elapsed Time High, Distance Lo (0.1 m lsb), Distance Mid, Distance High, Average Stroke Rate, Ending Heartrate, Average Heartrate, Min Heartrate, Max Heartrate, Drag Factor Average, Recovery Heart Rate, (zero = not valid data. After 1 minute of rest/recovery, PM5 sends this data as a revised End Of Workout summary data characteristic unless the monitor has been turned off or a new workout started) Workout Type

C2 Multiplexed Information: Data Definitions

ID	Name	Byte Length	Definitions
0x003A	C2 rowing end of workout additional summary data characteristic 1	(18 bytes)	*Data bytes packed as follows:* Log Entry Date Lo, Log Entry Date Hi, Log Entry Time Lo, Log Entry Time Hi, Split/Interval Size Lo, (meters or seconds) Split/Interval Size Hi, Split/Interval Count, Total Calories Lo, Total Calories Hi, Watts Lo, Watts Hi, Total Rest Distance Lo (1 m lsb), Total Rest Distance Mid, Total Rest Distance High Interval Rest Time Lo (seconds), Interval Rest Time Hi, Avg Calories Lo, (cals/hr) Avg Calories Hi,
0x003B	C2 rowing heart rate belt information characteristic	(6 bytes)	Manufacturer ID, Device Type, Belt ID Lo, Belt ID Mid Lo, Betl ID Mid Hi, Belt ID Hi
0x003C	C2 rowing end of workout additional summary data characteristic 2	(10 bytes)	*Data bytes packed as follows:* Log Entry Date Lo, Log Entry Date Hi, Log Entry Time Lo, Log Entry Time Hi, Avg Pace Lo (0.1 sec lsb) Avg Pace Hi, Game Identifier/ Workout Verified (see Appendix), Game Score Lo, Game Score Hi Erg Machine Type (enum – see Appendix)

Near Field Communication NDEF Records

The PM5 unit configures itself as a Near Field Communication Tag A. The tag consists of two records.
The first record is an External record type used for pairing the mobile device with the PM5. The information in this record is sufficient for establishing a BLE connection between the mobile device and the PM5. The record format is as follows.

BLE Pairing Record Identifier String	PM5 BLE address (6 bytes)	PM5 BLE Address Type (1 byte)	PM5 Advertising Name (up to 31 bytes)
concept2.com:bleconnectinfo	*0x54 0x96 0xA2 0x56 0x10 0xF0	0x01	*PM5 430343693

*This data is used as an example and will vary unit to unit.

The second record is an Android Application Record (AAR) containing the following string. The record is used to launch the Ergdata application on an Android platform. The data in this record does not change.

BLE Launch App Identifier String	Android Package to Launch
android.com:pkg	com.concept2.ergdata

Appendix A

Enumerated Values

Erg Machine Type

```
typedef enum {
  ERGMACHINE_TYPE_STATIC_D,
  ERGMACHINE_TYPE_STATIC_C,
  ERGMACHINE_TYPE_STATIC_A,
  ERGMACHINE_TYPE_STATIC_B,
  ERGMACHINE_TYPE_STATIC_E = 5,
  ERGMACHINE_TYPE_STATIC_SIMULATOR = 7,
  ERGMACHINE_TYPE_STATIC_DYNAMIC = 8,
  ERGMACHINE_TYPE_SLIDES_A = 16,
  ERGMACHINE_TYPE_SLIDES_B,
  ERGMACHINE_TYPE_SLIDES_C,
  ERGMACHINE_TYPE_SLIDES_D,
  ERGMACHINE_TYPE_SLIDES_E,
  ERGMACHINE_TYPE_SLIDES_DYNAMIC = 32,
  ERGMACHINE_TYPE_STATIC_DYNO = 64,
  ERGMACHINE_TYPE_STATIC_SKI = 128,
  ERGMACHINE_TYPE_STATIC_SKI_SIMULATOR = 143,
  ERGMACHINE_TYPE_BIKE= 192,
  ERGMACHINE_TYPE_BIKE_ARMS,
  ERGMACHINE_TYPE_BIKE_NOARMS,
  ERGMACHINE_TYPE_BIKE_SIMULATOR = 207,
  ERGMACHINE_TYPE_NUM
} OBJ_ERGMACHINETYPE_T;
```

Workout Type

```
typedef enum {
  WORKOUTTYPE_JUSTROW_NOSPLITS = 0,
  WORKOUTTYPE_JUSTROW_SPLITS,
  WORKOUTTYPE_FIXEDDIST_NOSPLITS,
  WORKOUTTYPE_FIXEDDIST_SPLITS,
  WORKOUTTYPE_FIXEDTIME_NOSPLITS,
  WORKOUTTYPE_FIXEDTIME_SPLITS,
  WORKOUTTYPE_FIXEDTIME_INTERVAL,
  WORKOUTTYPE_FIXEDDIST_INTERVAL,
  WORKOUTTYPE_VARIABLE_INTERVAL,
  WORKOUTTYPE_VARIABLE_UNDEFINEDREST_INTERVAL,
  WORKOUTTYPE_FIXED_CALORIE,
  WORKOUTTYPE_FIXED_WATTMINUTES,
  WORKOUTTYPE_FIXEDCALS_INTERVAL,
  WORKOUTTYPE_NUM
} OBJ_WORKOUTTYPE_T;
```

Interval Type

```
typedef enum {
  INTERVALTYPE_TIME,
  INTERVALTYPE_DIST,
  INTERVALTYPE_REST,
```

```
  INTERVALTYPE_TIMERESTUNDEFINED,
  INTERVALTYPE_DISTANCERESTUNDEFINED,
  INTERVALTYPE_RESTUNDEFINED,
  INTERVALTYPE_CAL,
  INTERVALTYPE_CALRESTUNDEFINED,
  INTERVALTYPE_WATTMINUTE,
  INTERVALTYPE_WATTMINUTERESTUNDEFINED,
  INTERVALTYPE_NONE = 255
} OBJ_INTERVALTYPE_T;
```

Workout State

```
typedef enum {
  WORKOUTSTATE_WAITTOBEGIN,                                  // 0
  WORKOUTSTATE_WORKOUTROW,                                   // 1
  WORKOUTSTATE_COUNTDOWNPAUSE,                               // 2
  WORKOUTSTATE_INTERVALREST,                                 // 3
  WORKOUTSTATE_INTERVALWORKTIME,                             // 4
  WORKOUTSTATE_INTERVALWORKDISTANCE,                         // 5
  WORKOUTSTATE_INTERVALRESTENDTOWORKTIME,                    // 6
  WORKOUTSTATE_INTERVALRESTENDTOWORKDISTANCE,                // 7
  WORKOUTSTATE_INTERVALWORKTIMETOREST,                       // 8
  WORKOUTSTATE_INTERVALWORKDISTANCETOREST,                   // 9
  WORKOUTSTATE_WORKOUTEND,                                   // 10
  WORKOUTSTATE_TERMINATE,                                    // 11
  WORKOUTSTATE_WORKOUTLOGGED,                                // 12
  WORKOUTSTATE_REARM,                                        // 13
} OBJ_WORKOUTSTATE_T;
```

Rowing State

```
typedef enum {
 ROWINGSTATE_INACTIVE,
 ROWINGSTATE_ACTIVE,
} OBJ_ROWINGSTATE_T;
```

Stroke State

```
typedef enum {
 STROKESTATE_WAITING_FOR_WHEEL_TO_REACH_MIN_SPEED_STATE,
 STROKESTATE_WAITING_FOR_WHEEL_TO_ACCELERATE_STATE,
 STROKESTATE_DRIVING_STATE,
 STROKESTATE_DWELLING_AFTER_DRIVE_STATE,
 STROKESTATE_RECOVERY_STATE
} OBJ_STROKESTATE_T;
```

Workout Duration Type

```
enum DurationTypes {
  CSAFE_TIME_DURATION = 0,
  CSAFE_CALORIES_DURATION = 0X40,
  CSAFE_DISTANCE_DURATION = 0X80,
  CSAFE_WATTS_DURATION = 0XC0};
```

Game ID

enum {
        APGLOBALS_GAMEID_NONE,
        APGLOBALS_GAMEID_FISH,
        APGLOBALS_GAMEID_DART,
        APGLOBALS_GAMEID_TARGET_BASIC,
        APGLOBALS_GAMEID_TARGET_ADVANCED,
        APGLOBALS_GAMEID_CROSSTRAINING};

Game Identifier / Verified Information

The Game Identifier/Workout Verified byte in the *C2 rowing end of workout additional summary data characteristic 2* contains two independent data. The Game Identifer is contained in the lower nibble with the enumeration as defined above. The Workout Verified flag is contained in the upper nibble. See the additional definitions below.

```
#define LOGMAP_GAMETYPEIDENT_PM5_MSK                 0x0F
#define LOGMAP_LOGHEADER_STRUCT_VERIFIED_MSK         0xF0

#define LOGMAP_GET_GAMETYPEIDENT_M(gameid)           \
    ((UINT8_T)(gameid & LOGMAP_GAMETYPEIDENT_PM5_MSK))

#define LOGMAP_GET_WORKOUTVERIFIED_M(gameid)          \
     ((UINT8_T)(( gameid & LOGMAP_LOGHEADER_STRUCT_VERIFIED_MSK) >> 4))
```

Communicating with the PM using CSAFE Commands

The C2 PM Receive Characteristic and C2 PM Transmit Characteristic can be used to send and receive CSAFE frames. In general refer to the PM communications specification and the CSAFE protocol specification for information on how to do this. The following are some additional notes to supplement these specifications.

Retrieving Heartrate Belt Information

The PM Heart Rate Belt Information Characteristic will send data whenever it changes. You can also get this data using a CSAFE command. As the PM5 now supports the Polar H7 and similar Bluetooth Smart heart rate belts with 32-bit belt IDs, use this new CSAFE command: CSAFE_PM_GET_EXTENDED_HBELT_INFO – 0x57
This command returns a 1 byte user number, 1 byte manufacturer ID, 1 byte device type and 4-byte belt id.

Commanding the PM5 to Pair with a known Heartrate Belt

If your application saves the heart rate belt information then you can command the PM to automatically pair with the belt each time you connect with the PM. This will save a step for the user, as typically he had to pair the PM to a belt using the PM front panel menus. To do this use the CSAFE command CSAFE_PM_SET_EXTENDED_HRM – 0x39. This command uses the same parameters as the GET function in the previous paragraph.