

# BioTelemetry



# BioTelemetry

## User Guide

**Copyright – Wireless Motorsport Pty Ltd – 2016**

The information in this document is subject to change without notice. While every effort is taken to ensure accuracy and correctness, we take no responsibility for the consequences of any inconsistencies or omissions in this User Guide.

May 2016

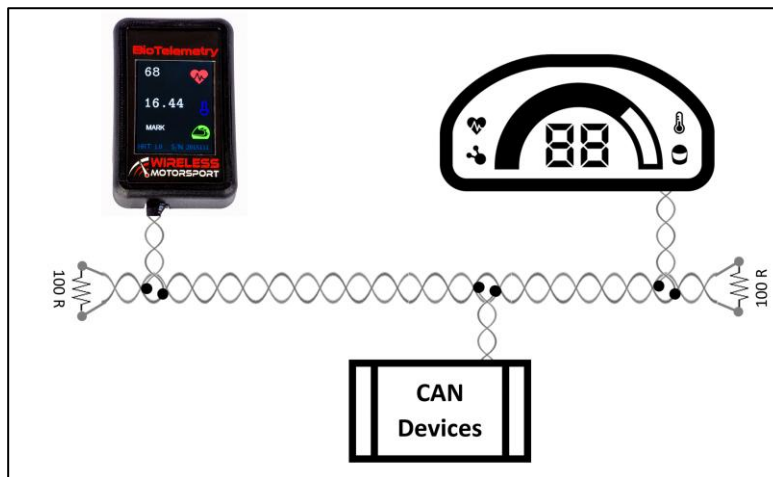
# Table of Contents

Pin-outs for the BioTelemetry Device .....	3
6 Pin .....	3
8 Pin .....	4
Pairing Sensors with the BioTelemetry Device .....	5
Pair a New Sensor .....	5
Pin-outs for the Deutsch Plug .....	6
CAN Protocol for Configuring Data Loggers .....	7
CAN Data Transmission Format .....	7
CAN Identifier per sensor Format .....	7
CAN Identifier per Driver Format (MoTeC & RacePak) .....	11
CAN Synchronisation .....	13
Advanced Configuration for the BioTelemetry Device .....	15
Configuring the BioTelemetry Device .....	15
Example Configuration File .....	16
Setting Descriptions and Values .....	17
Additional Settings .....	18
Updating the BioTelemetry Firmware .....	20

# Pin-outs for the BioTelemetry Device

## Overview

There are two possible pin-outs for the BioTelemetry device: the 6 and 8-pin Deutsch Connectors. The two are listed below with instructions and diagrams for each. The basic CAN device configuration is as follows:



## A. 6-pin DTM Deutsch Connector

GND Power	1 ■	■ 6	Sync
CAN LOW	2 ■	■ 5	GND Signal
CAN HIGH	3 ■	■ 4	12 V Power

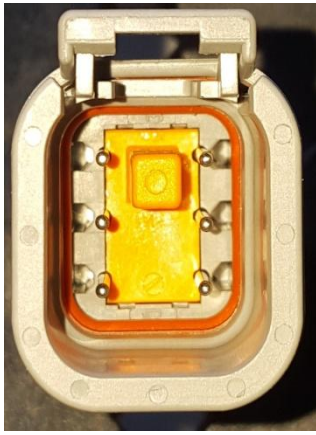
*DTM Rear View (Male Pins)*

### Power Requirements:

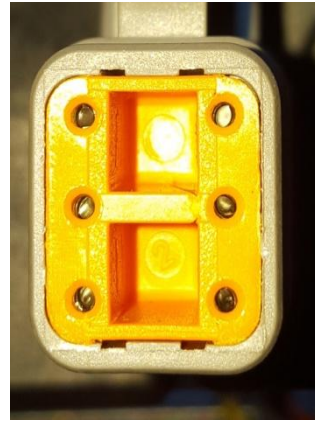
- Power is supplied to **Pin 4** of the DTM plug; power is between 9V to 32V.
- The current draw is 100mA at 12 volts.
- Ground is **Pin 1**.

### CAN Bus:

- **CAN Low** is **Pin 2** and **CAN High** is **Pin 3**.
- The default bit rate is 1Mbit, but can be set to 500Kbit via the configuration file (loaded via the micro SD card).
- The CAN bus must be correctly terminated at both ends for the bus to work correctly.



*Illustration 1:  
6-pin DTM Male Pins  
Attached to Telemetry Device*



*Illustration 2:  
6-pin DTM  
Female Mating Socket*

## B. 8-pin DTM Deutsch Connector

GND Power	1 ■	■ 8	12 V Power
CAN LOW	2 ■	■ 7	CAN HIGH
HRT Analogue	3 ■	■ 6	Temperature Analogue
GND Signal	4 ■	■ 5	Sync

*DTM Rear View (Male Pins)*

### Power Requirements:

- Power is supplied to **Pin 8** of the DTM plug; power is between 9V to 32V.
- The current draw is 100mA at 12 volts.
- Ground is **Pin 1**.

### CAN Bus:

- CAN Low is **Pin 2** and **CAN High** is **Pin 7**.
- The default bit rate is 1Mbit but can be set to 500Kbit via the configuration file (loaded via the micro SD card).
- The CAN bus must be correctly terminated at both ends for the bus to work correctly.

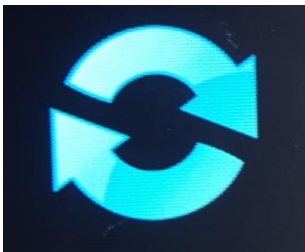
# Pairing Sensors with the BioTelemetry Device

## Overview

A Heart Rate Strap and Body Temperature Monitor can both be paired and used with the BioTelemetry device.

### Pair a new sensor:

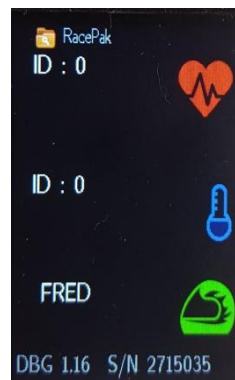
1. Place all the sensors, **including any current** and new sensor(s), within 1 meter of the BioTelemetry device.
2. Ensure that other sensors that are NOT to be paired are powered off (for the Temperature Sensor, remove the battery; this Sensor transmits up to 50 meters).
3. Power up the BioTelemetry device.
4. Power up both Sensors. For the Heart Rate Monitor, ensure it is also strapped on so the signal is transmitting.
5. Place the BioTelemetry device into Pairing Synchronisation mode by connecting the **Sync Pin** to **GND Signal** (see Pin-outs for Synchronisation *below*) for at least 10 seconds.
6. The BioTelemetry device will go into Search mode and once the Sensor is located, the screen will display the Pairing Icon.



Picture 1: Sensor Pairing Icon

7. When you release the Sync Pin, the BioTelemetry device will restart and go into sync mode to search for Sensors to pair with.

Picture 2: Sensor Search

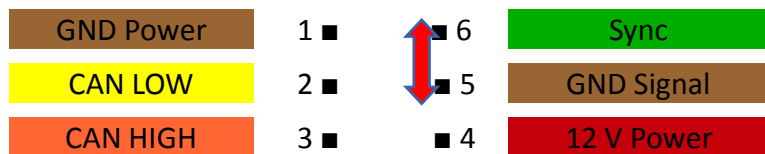


- After finding each Sensor, it will store the Sensor's ID into flash memory and resume normal operation.

*For additional drivers, please see the Advanced Configuration chapter.*

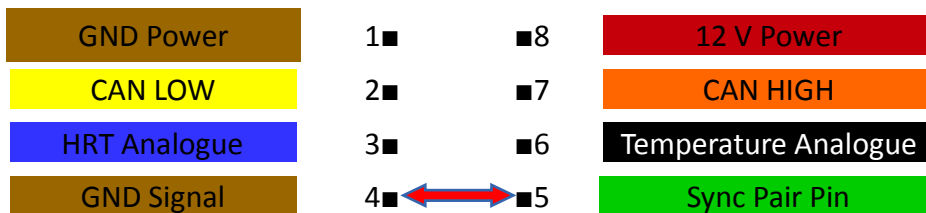
## Pin-outs for Synchronisation

### 6-pin DTM Connector



*DTM Rear View (Male Pins)*

### 8-pin DTM Connector



*DTM Rear View (Male Pins)*

# CAN Protocol for Configuring Data Loggers

## Overview

All data in this document is specified in Hexadecimal unless specified otherwise.

The byte order Big-endian network format.

## CAN Data Transmission Format

There are three modes of data transmission:

- Single driver (monitoring one sensor, one driver)
- Multiple drivers (monitoring multiple drivers, one at a time as in Endurance Racing)
- Rally mode (monitoring driver and navigator at the same time)

The mode is set by the **CAN\_LOGGER** tag in the configuration file:

**CAN\_LOGGER: 0** This has a unique CAN Identifier per sensor

**CAN\_LOGGER: 1** Each driver has unique CAN Identifier; used for the MoTeC dash.

**CAN\_LOGGER: 2** Each driver has unique CAN Identifier; used for the RacePak dash.

In all formats the CAN Identifiers start at the **CAN\_BASE\_ADDRESS**, which is set in the configuration file. The default address is **0x400**.

## CAN Identifier per Sensor Format

### CAN Sensor Identifier Format:

A block of eight identifiers starting at the CAN Identifier set in the configuration file by the **CAN\_BASE\_ADDRESS**.

Each sensor has a unique identifier and can be configured as follows:

**CAN SENSOR IDENTIFIER = CAN\_BASE\_ADDRESS + Sensor**

Where Sensor is:

**0 = Heart Rate**

**1 = Temperature**

**2 = Muscle Oxygen**

**3 = Board Parameters**

**4 = Future Sensor 1**

**5 = Future Sensor 2**

**6 = Future Sensor 3**

**7 = Future Sensor 4**

## CAN\_BASE\_ADDRESS is 0x400 Example:

CAN Identifier	Sensor
0x400	Heart Rate
0x401	Temperature
0x402	Muscle Oxygen
0x403	Board Parameters

## Heart Rate Message Format

Byte 0	1	2	3	4	5	6	7
Driver	0	HRT ID MSB	HRT ID LSB	HRT Value MSB	HRT Value LSB	Number Drivers Detected	Driver Priority

### Where-

**Driver:** This is the Driver number that was set in the configuration file. For example, if driver Joe has been assigned Driver 1 then this will be set to 1.

**HRT ID:** This is the Heart Rate Monitor's unique identifier.

If the value reads **0xFFFF** then no transmission is detected from the Heart Rate Monitor.

**HRT Value:** The computed heart rate in beats per minute.

If the value reads **0**, then there is a valid transmission; however, the strap is not making electrical contact with the skin.

**Number of Drivers Detected:** You can configure the device for a maximum of four drivers. This displays the total number of drivers detected.

**Driver Priority:** This is the detected driver with the highest priority. For example, the device detects Driver 1 and Driver 3. The byte is set to 1, because Driver 1 has the highest priority.

### Single driver example:

**Example A:** Driver 1 is the only driver present with a heart rate of 90 beats per minute (0x5A) and the sensor ID of 0x35BD (1376)

**0x400: 01 00 35 BD 00 5A 01 01**

**Example B:** Driver 2 is the only driver present with a heart rate of 82 beats per minute (0x52) and a sensor ID of 0x6839 (26681)

**0x400: 02 00 68 39 00 52 01 02**

### Two driver example:

Driver 1 Heart Rate 100 beats per minute (0x64) ID 0x35BD

Driver 2 Heart Rate 82 beats per minute (0x52) ID 0x6839

**0x400: 01 00 35 BD 00 64 02 01**



**0x400: 02 00 68 39 00 52 02 00**

Two drivers are detected; Driver 1 has the highest priority.

**Three driver example:**

Driver 1 Heart Rate 86 (0x56) ID 0x35BD

Driver 2 Heart Rate 147 (0x93) ID 0x6839

Driver 3 Heart Rate 72 (0x48) ID 0xA17D

**0x400: 01 00 35 BD 00 56 03 01**

**0x400: 02 00 68 39 00 93 03 00**

**0x400: 03 00 A1 7D 00 48 03 00**

**No drivers detected example:**

If no driver is detected, the CAN message displays:

**0x400: 00 00 FF FF 00 00 FF FF**

### Temperature Message Format

Byte 0	1	2	3	4	5	6	7
Driver	0	Temperature ID MSB	Temperature ID LSB	Temperature Value MSB	Temperature Value LSB	0	Status

**Where-**

**Driver:** This is the Driver number that was set in the configuration file. For example, if driver Joe has been assigned Driver 1 then this will be set to 1.

**Temperature ID:** This is the Temperature Sensor's unique identifier.

If the value reads **0xFFFF** then no transmission is detected from the Temperature Sensor.

**Temperature Value:** Temperature = **Temperature Value / 100**

**Status:**

*NOT\_CONNECTED* = 0,

*VALID\_DATA* = 1,

*RESEND* = 2

The *RESEND* means the data is valid, but it has not been updated. The Temperature Sensor sends data every four seconds. For example:

Driver 2 Temperature 25.20° C device ID 0x76C5 data has been resent

**0x401: 02 00 76 C5 09 D8 00 02**

Driver 1 Temperature 23.03° C device ID 0xD8F4 valid new data

**0x401: 01 00 D8 F4 08 FF 00 01**

## Muscle Oxygen Message Format

Byte 0	1	2	3	4	5	6	7
Driver	0	MOX ID MSB	MOX ID LSB	Total Hemoglobin MSB	Total Hemoglobin LSB	Current Saturated MSB	Current Saturated LSB

### *Where-*

**Driver:** This is the Driver number that was set in the configuration file. For example, if driver Joe has been assigned Driver 1 then this will be set to 1.

**MOX ID:** The Muscle Oxygen Sensor's unique identifier.

If the value reads **0xFFFF** then there has been no transmission detected from the MOX Sensor.

**Total Hemoglobin:** Divide by 100 to obtain the Total Hemoglobin.

**Current Saturated:** Divide by 10 to get the Percentage of Oxygen.

## Board Parameters

Byte 0	1	2	3	4	5	6	7
0	0	Board Temperature MSB	Board Temperature LSB				

### *Where-*

**Board Temperature:**  $T = \text{Board Temperature} / 1000$  in Degrees C

## CAN Identifier per Driver Format (MoTeC & RacePak)

Each driver has a unique CAN Identifier set in the configuration file starting at the **CAN\_BASE\_ADDRESS**.

**CAN\_BASE\_ADDRESS** is 0x400 Example:

CAN Identifier	Driver
0x400	Current Driver
0x401	Driver 1
0x402	Driver 2
0x403	Driver 3
0x404	Driver 4

### Where -

**Driver:** 1 has the highest priority, Driver 4 would be the lowest priority. If present, this data will be sent at the base address. If there are other drivers / navigators then it will be sent on its address, but the data format will be identical.

**Current Driver:** The current driver is the driver with the highest priority. Driver 1 is the highest and Driver 4 is the lowest. For Endurance Racing, there will be two drivers present during the change-over phase.

### Heart Rate Message Format

Byte 0	1	2	3	4	5	6	7
0	1	HRT ID MSB	HRT ID LSB	HRT Value MSB	HRT Value LSB	Number Drivers Detected	Driver Priority

### Where-

**HRT ID:** This is the Heart Rate Monitor's unique identifier.

If the value reads **0xFFFF** then there is no transmission detected from the Heart Rate Monitor.

**HRT Value:** The computed heart rate in beats per minute.

If the value reads **0**, then the transmission is valid, **but** the strap is not making electrical contact with the skin.

**Number of Drivers Detected:** You can configure the device for a maximum of four drivers. This displays the total number of drivers detected.

**Driver Priority:** This is the detected driver with the highest priority. For example, the device detects Driver 1 and Driver 3. The byte is set to 1, because Driver 1 has the highest priority.

### Single driver example:

Driver 1 Heart Rate 76 beats per minute (0x4C) Device ID 0x6839

**0x400: 00 01 68 39 00 4C 01 01**

**0x401: 00 01 68 39 00 4C 01 01**

**Two driver example:**

Driver 1 Heart Rate 173 beats per minute (0xAD) Device ID 0x6839

Driver 2 Heart Rate 79 beats per minute (0x4F) Device ID 0xA17D

**0x400: 00 01 68 39 00 AD 02 01**

**0x401: 00 01 68 39 00 AD 02 01**

**0x402: 00 01 A1 7D 00 4F 02 00**

**No driver detected example:**

If no driver is detected, the CAN message displays:

**0x400: 00 01 FF FF 00 00 FF FF**

### Temperature Message Format

Byte 0	1	2	3	4	5	6	7
0	2	Temperature ID MSB	Temperature ID LSB	Temperature Value MSB	Temperature Value LSB	0	Status

**Where-**

**Temperature ID:** Is the Temperature Sensor's unique identifier.

If the value reads **0xFFFF** then no transmission detected from the Temperature Sensor.

**Temperature Value:** Temperature = **Temperature Value / 100**

**Status:**

*NOT\_CONNECTED* = 0,

*VALID\_DATA* = 1,

*RESEND* = 2

**Example:**

Driver 1 24.55°C (0x997) device ID 0x76C5

**0x400: 00 02 76 C5 09 97 00 02**

### Muscle Oxygen Message Format

Byte 0	1	2	3	4	5	6	7
0	3	MOX ID MSB	MOX ID LSB	Total Hemoglobin MSB	Total Hemoglobin LSB	Current Saturated MSB	Current Saturated LSB

**Where -**

**MOX ID:** The Muscle Oxygen Sensor's unique identifier.

If the value reads **0xFFFF** then there has been no transmission detected from the MOX Sensor.

**Total Hemoglobin:** Divide by 100 to obtain the Total Hemoglobin.

**Current Saturated:** Divide by 10 to get the Percentage of Oxygen.

### Board Parameters

Byte 0	1	2	3	4	5	6	7
0	4	Board Temperature MSB	Board Temperature LSB				

**Where-**

**Board Temperature:**  $T = \text{Board Temperature} / 1000$  in degrees Celsius

### CAN Synchronisation

You can synchronise a new sensor through the CAN bus. The default CAN identifier is **0x500**

Byte 0	1	2	3	4	5	6	7
0	Trigger	ANT ID MSB	ANT ID LSB	0	Sensor	0	Driver

**Where -**

**Trigger:** When **Trigger = 0** the message is ignored.

A NON zero activates the synchronisation process. There must be at least 15 identical messages sent before the sensor enters synchronisation mode.

**ANT ID:** Each sensor has a unique identifier that is assigned at manufacture. Store this unique identifier in the BioTelemetry device's internal EEPROM.

If **ANT ID** is set to **0**, the device will pair with the sensor that is in closest proximity -- a wild card pairing.

**Sensor:** The type of sensor to pair

0. Sets all sensor identifiers for the driver to zero to enable all sensors to pair

1. Pair the Heart Rate Sensor
2. Pair the Temperature Monitor
3. Pair the Muscle Oxygen Sensor

**Driver:** Identifies which driver to synchronise (between 1 and 4).

**Example 1:**

Pair all Driver 1's sensors.

0x500: 00 00 00 00 00 00 00 00 // Idle

0x500: 00 01 00 00 00 00 00 01 // Trigger synchronisation for Driver 1

:

0x500: 00 01 00 00 00 00 00 01 // 15 Messages sent

0x500: 00 00 00 00 00 00 00 00 // Idle

**Example 2:**

Pair a new Heart Rate Monitor for Driver 2 with a wild card ANT ID.

0x500: 00 00 00 00 00 00 00 00

0x500: 00 01 00 00 00 01 00 02

:

0x500: 00 01 00 00 00 01 00 02 // 15 Messages sent

0x500: 00 00 00 00 00 00 00 00

# Advanced Configuration for the BioTelemetry Device

## Overview

The BioTelemetry device can be configured for up to eight drivers, each using three different sensors by using the micro SD card. When the device powers up, it reads the configuration file and transfers data to the device's internal EEPROM.

## Configuring the BioTelemetry device:

1. While the BioTelemetry device is powered on, make note of the serial number, which is displayed on the bottom line of the device.



Picture 3: Device Serial Number

2. Power off the BioTelemetry device.
3. Remove the micro SD card from the BioTelemetry device by gently pushing the end of the SD card with a small coin. The SD card is located on the right side of the device.
4. Insert the SD card into the PC.
5. Create a configuration file on the SD card with a simple text file editor such as Notepad.
6. Name the configuration file either: **serial\_number.txt** or **biotelm.txt**

Using the device in *Picture 3* for example, the name of the file would be: 2015018.txt for a unit with the S/N 201508. Alternatively, the default **biotelm.txt** can be used; however, the serial number file has precedence.

7. Once you have completed the configuration, insert the SD card back into the BioTelemetry device and power the device back up.

8. See below for the configuration file format, specific configuration options, values and examples.



Picture 4: SD Card with Pins Facing Up

**NOTE:** Insert the SD Card into the BioTelemetry device with the pins facing up.

### Example configuration file:

The following is a typical configuration file for two drivers, each using three sensors, transmitting to a MoTeC dash, in metric units. *(Information on each setting and value follows the example configuration file.)*

**DRIVER : 1**

**HRM\_ID : 5678**

**TEMP\_ID : 2345**

**MOX\_ID : 7180**

**NAME : FRED**

**DRIVER : 2**

**HRM\_ID : 8991**

**TEMP\_ID : 4531**

**MOX\_ID : 7187**

**NAME : JOE**

**CAN\_BASE\_ADDRESS: 400**

**CAN\_SYNC\_ADDRESS: 500**

**CAN\_LOGGER: 1**

**BAUD: 1000**

**UNIT\_TYPE: 0**



**DISPLAY\_MOX: 0**

**HRT\_COUNT\_TIMEOUT: 100**

## Setting descriptions and values:

### File Format:

The file consists of **TAG : VALUE** pairs for all entries. A **Driver** is defined from 1 to 8 followed by the unique identifiers for each of the Driver's sensors. All entries are case insensitive.

For example:

**DRIVER : 1**

**HRM\_ID : 5678**

**TEMP\_ID : 2345**

**MOX\_ID : 7180**

**NAME : FRED**

*Where:*

**HRM\_ID** is the heart rate strap ID

**TEMP\_ID** is the temperature monitor ID

**MOX\_ID** is the muscle oxygen sensor ID

**NAME** is the name of the driver

### CAN address settings:

The BioTelemetry data transmission address is defined using the **CAN\_BASE\_ADDRESS**. For example, if data is to be sent on CAN address 400, use the following:

**CAN\_BASE\_ADDRESS : 400**

To allow the BioTelemetry to synchronise to a new sensor via the CAN bus, your data logger can send a synchronise command via the CAN at a specified address. For example, to configure the BioTelemetry device to accept synchronisation messages on address 500, use:

**CAN\_SYNC\_ADDRESS : 500**

### CAN data logger type:

The BioTelemetry unit can send CAN messages in three formats:

**CAN\_LOGGER : 1** is used for a MoTeC data logger

**CAN\_LOGGER : 2** is used for a RacePak data logger

**CAN\_LOGGER: 0** all other data loggers such as AIM

Each type of sensor has a unique CAN address starting at the CAN Base address, for example:

**CAN Address 400 : heart rate monitor**

**CAN Address 401 : temperature sensor**

## CAN Address 402 : muscle oxygen sensor

*See the CAN Protocol Chapter for more details.*

### CAN baud rate:

To set the CAN baud rate, use the tag **BAUD**. The **VALUE** is in kilobits. For example, to set the CAN baud rate at 1Mbit:

**BAUD : 1000**

### Display units:

To display metric, use:

**UNIT\_TYPE : 0**

To display imperial:

**UNIT\_TYPE : 1**

### Display muscle oxygen:

To display the muscle oxygen values, set the **DISPLAY\_MOX: 1**

The second driver's data will not be displayed. The default value is **DISPLAY\_MOX: 0**

### Heart rate time out:

If there is a bad connection between the heart rate strap and the driver's skin, the strap does not send any new data and the BioTelemetry device will display a ~ . Use the **TIMEOUT** to set the number of seconds before this occurs. The seconds are multiplied by 0.25. For example:

**HRT\_COUNT\_TIMEOUT: 100** will allow for a 25 second time out.

## Additional Settings

### Scanning for a new sensor:

To search for a new sensor when the identifier is unknown, use **0** for the device's **VALUE**.

*Note: ensure that no other sensor of that type is within 30 meters of the BioTelemetry device while searching for the new sensor.*

If you were adding a new heart rate strap for **Driver 2** and did not know the unique identifier, the configuration file would be as follows:

**DRIVER : 2**

**HRM\_ID : 0**

**TEMP\_ID : 4531**

**MOX\_ID : 71802**

**NAME : JOE**

### **Removing a sensor:**

If the Driver is not using a device, enter **-1** for the **VALUE**. For example, if Driver 3 is not using the Moxy sensor:

**DRIVER : 3**

**HRM\_ID : 2785**

**TEMP\_ID : 3532**

**MOX\_ID : -1**

**NAME : STEVE**

### **Placing device in demonstration mode:**

To place the device into demonstration mode, add the following setting:

**DEMO : 1**

This will increment the heart rate and temperature every half a second until it reaches a maximum of 180 for the heart rate, and then it will reset it to zero and repeat.

### **Resetting:**

When the firmware is in **ANT device sync mode** the corresponding ANT identifiers in the configuration file will be ignored. The sync mode can then be entered in one of three ways:

1. Using the CAN sync command
2. Pulling the sync hardware line low
3. Setting the ANT device ID to **0** in the configuration file

# Updating the BioTelemetry Firmware

## Overview

As we offer enhancements to the device, we will make firmware updates available from the Wireless Motorsport website.

## Updating the firmware requires:

1. Micro SD card
2. A PC to download the firmware.
3. The latest version of the firmware file **hrt.wms**. The file can be downloaded from the Wireless Motorsport web site.
4. A SD card adaptor. This can be either a USB device that plugs into a PC; some laptops have this facility built in.
5. A small coin to push the SD card into the slot on the BioTelemetry device.



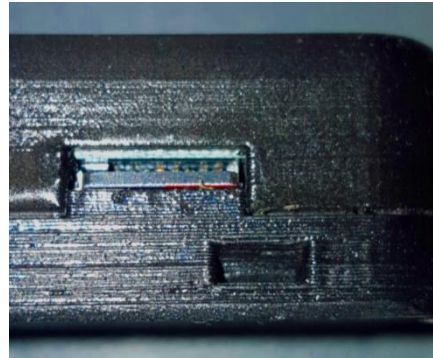
*Picture 5: SD Adaptor and USB Card Reader*

## Steps:

1. Download the **hrt.wms** file from the Wireless Motorsport web site to a PC.
2. Insert the SD Card into the PC and copy the **hrt.wms** to the SD Card.
2. **Power off** the BioTelemetry device.
3. Insert the SD card into the BioTelemetry device's SD slot as seen in *Picture 6* below.



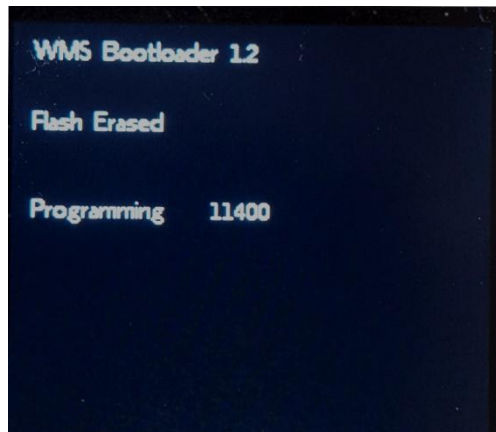
Picture 6: SD Card Positioning



Picture 7: Close up of the SD Card Slot

4. Power up the BioTelemetry device. The Boot Loader will program the BioTelemetry's internal flash memory and the screen will briefly display the status as seen in *Picture 8*. This process takes less than a minute, and the device will then reboot with the new firmware installed.

**NOTE: Ensure the power is not interrupted while programming the device.**



Picture 8: Programming the flash