G.FIT

User Guide and Specification

Copyright Information and Usage Notice

Revision History

Revision	Effective Date	Description
1.0	June 2017	Initial Customer Sample Release.
1.1	September 2017	Customer Sampling Update
1.2	September 2017	Production Release

Table of Contents

List of Figures

List of Tables

1 Support

The D52 ANT SoC module series (including G.FIT) uses the nRF52832 from Nordic Semiconductor. You can seek technical support from Nordic Semiconductor, www.nordicsemi.com. G.FIT application support can be sought from Dynastream Innovations, via www.thisisant.com.

1.1 G.FIT Technical References

Documents

1. G.FIT User Guide and Specification
2. G.FIT Firmware Updater Application Note
3. G.FIT and Premium Module Manufacturing Considerations Application Note
4. nRF52832 Product Specification, Nordic Semiconductor
5. nRF52 Series Compatibility Matrix, Nordic Semiconductor Infocenter
6. nRF52832 Objective Product Specification, Nordic Semiconductor
7. nRF52832 S332 SoftDevice Specification, Dynastream Innovations
8. nRF52 Development Kit Documentation, Nordic Semiconductor Infocenter
9. ANT SoC Module Starter Kit User Manual, Dynastream Innovations
10. ANT Message Protocol and Usage, Dynastream Innovations
11. Interfacing with ANT General Purpose Chipsets and Modules, Dynastream Innovations
12. Application Note: Interpreting RF Radiation Patterns, Dynastream Innovations
13. Bluetooth Fitness Machine Service/Profile, Bluetooth SIG
14. Bluetooth Core Specification, Bluetooth SIG

Software

1. G.FIT SDK – simulators, G.FIT, iOS, Connect IQ demo code
2. G.FIT Library – G.FIT Library files, S332 ANT/Bluetooth SoftDevice, Evaluation G.FIT Network Processor binary
3. S332 nRF52832 SoftDevice, Dynastream Innovations
4. nRF5 SDK, Nordic Semiconductor
5. ANTwareII – a system testing and debugging tool, Dynastream Innovations
6. SimulANT+

Design models (all apply to G.FIT modules)

1. D52Q Altium library, Dynastream Innovations
2. D52Q module STEP model, Dynastream Innovations
3. D52M Altium library, Dynastream Innovations
4. D52M module STEP model, Dynastream Innovations

The above documents and software are available at www.dynastream.com, www.thisisant.com, www.bluetooth.com and/or www.nordicsemi.com / infocenter.nordicsemi.com. User registration may be required.

2 Use Case

G.FIT is a turnkey dual-protocol ANT/Bluetooth® low energy (BLE) solution for wireless fitness equipment, optimized for group training environments with 50+ fitness devices and multiple receivers.

As group fitness training grows in demand, many challenges to a good experience are presenting themselves:

- Gym and studio environments have lots of WiFi (and other) interference.
- Heart rate monitors from multiple manufacturers and wireless standards are on the market.
- Gym users and instructors are mostly non-technical users.
- Gym patrons bring in their own personal and fitness-specific devices (e.g. watches and phones).
- Gyms have many types of equipment from multiple manufacturers to maintain, many of them with different operating procedures.

With these challenges in mind, Dynastream Innovations Inc. has evolved individual fitness machines to incorporate group fitness capabilities. The G.FIT solution is the result of that evolution, and solves the challenges of group fitness:

- G.FIT works around WiFi interference to ensure large groups of **50+ devices can connect** concurrently.
- G.FIT solves pairing and setup difficulties with dual protocol support for both **ANT+ and BLE heart rate monitors**.
- G.FIT makes it easy for non-technical users to pair by using **proximity pairing and list pairing** features.
- G.FIT solves personal device pairing by **re-broadcasting ANT+ or BLE heart rate data** and works with the **ANT+ FE-C Device Profile** so that people can record entire workout sessions on their own personal devices.
- G.FIT enables technology maintenance with both **built-in support for easy wireless updates and the addition of custom tailored features and controls** using the G.FIT development SDK.

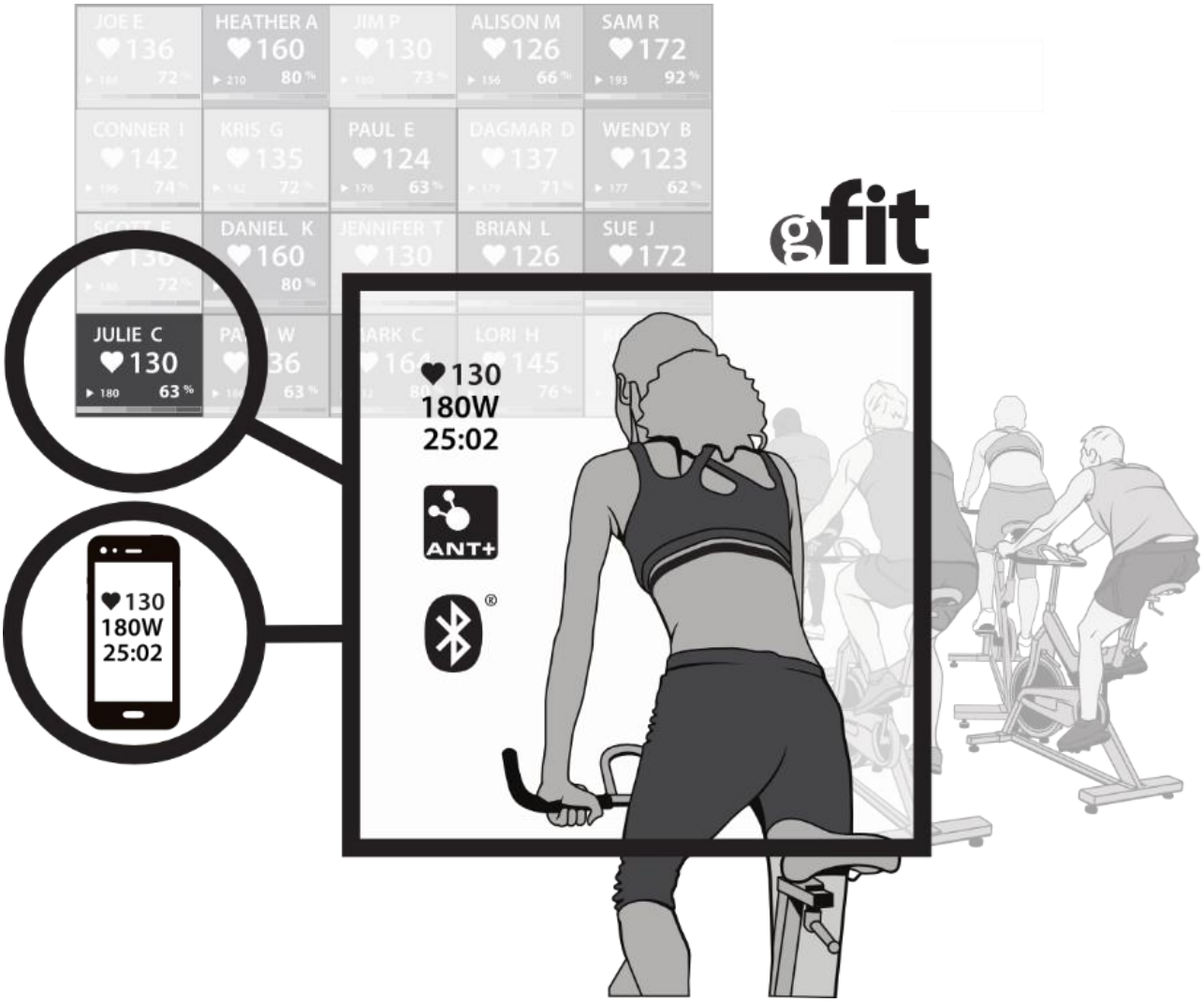


Figure 2-1. G.FIT Use Case Illustration

3 Development Kit Contents

3.1 D52DK2

The D52 starter kit (D52DK2) contains all the necessary equipment needed to get started on G.FIT development. See https://www.dynastream.com/d52starterkit.

3.1.1 G.FIT compatible module

- D52QSKM6IA-A Module

3.1.2 USB Interface Board

The USB interface board is required so that the G.FIT module can be used with the provided PC application. See section 5.2 for more details on installing the necessary drivers.

3.1.3 USB-m

The USB-m is required so that the developer can view the data being transmitted from the G.FIT module. The USB-m can be used with SimulANT+ to connect to one of the G.FIT modules ANT+ FE-C channels.

3.2 G.FIT Libraries

The contents of the G.FIT Libraries package are as follows:

- G.FIT static library: Pre-built library, compiled in GCC, that can be used by custom applications using G.FIT modules as an SOC. The static library, libgfit.a is available in the bin folder of the G.FIT_Library zip file.
- API Headers: These files, available in the inc folder of the G.FIT_Library zip file, need to be included by your application to access the G.FIT static library, and provide documentation on the available API calls.
    - gfit_interface.h: Function definitions and API documentation.
    - gfit_defines.h: Includes constants, enumerations and data structures.
- G.FIT Evaluation Network Processor: Pre-built hex file for the G.FIT network processor intended for evaluation purposes. This hex file can be used only in D52 development modules (D52QSKM6IA-A), e.g., for use with the D52 ANT SoC Module Series Starter Kit (D52DK2)
- ANT S332 SoftDevice

3.3 G.FIT Software Development Kit

The contents of the G.FIT Software Development Kit are as follows:

3.3.1 Example SoC Application (demo_gfit)

Shows how to integrate the G.FIT library into application. The demo implements a simple fitness equipment console with three buttons: Start (button A), Stop (button B) and Disconnect HR (button C). The demo shows how to initialize and configure G.FIT, trigger state transitions, and update data transmitted over the ANT+ FE-C and BLE FTMS profiles. Paired HR values are output over Segger RTT. The project file for the example application is located under apps\demo_gfit\armgcc. See section 8.1 for more details on compiling the demo.

3.3.2 Fitness Equipment Simulator PC Application

The fitness equipment simulator PC application simulates a fitness equipment console that uses the G.FIT module. See section 5 for more details.

3.3.3 Reference CIQ Application

Garmin's Connect IQ platform supports creating apps with custom ANT+ channels that can be used to connect Garmin's wearable and bike accessories with G.FIT enabled fitness equipment. If the user has a Connect IQ compatible Garmin

watch/bike computer, they are able to track their workout and save that activity for viewing on Garmin Connect over the ANT+ FE-C channel. See section 6.2 for more details.

3.3.4 Reference iOS Application

iOS devices supporting BT v4.0 and above are compatible with G.FIT over a BLE connection using the FTMS (Fitness Machine Service), HRS (Heart Rate Service), and DIS (Device Information Service). An iOS Application (including source code) is available in the download package to provide an example for scanning, pairing and connecting with a G.FIT enabled fitness equipment device.

3.4 SimulANT+ (available through Download)

SimulANT+ can be used to connect to one of the G.FIT ANT+ FE-C channels. See section 6.1 for more details. SimulANT+ is available for download at https://www.thisisant.com/developer/resources/downloads/. Before downloading SimulANT+ you will need to create an account on http://www.thisisant.com

4 G.FIT Operation

4.1 Fitness Equipment (FE) States

The G.FIT module has been designed so that it complies with the states defined in the ANT+ Fitness Equipment Device Profile. The ANT+ Fitness Equipment Device Profile defines four states that the fitness equipment can be in. Any FE device using the G.FIT module shall move between these states in response to user interaction as shown in Figure 4-1 below. These state transitions are completed using the GFIT_SetState command (section 7.3.1).

When the G.FIT is power cycled, the G.FIT will enter a configuration state. In this configuration stage the application shall then configure the G.FIT for the type of fitness equipment and desired fitness equipment behaviour. To move out of configuration mode, mode to state **READY**. When the fitness equipment is not used it can remain in a low power state, the **OFF** state. Typically, user activity such as pedaling will wake up the FE and activate the user interface. This would be an example of when to make the transition to the **READY** state. Some FE may never sleep, and would always default to the **READY** state when not in use.

From **READY**, a button press or sustained activity begins a workout session and puts the FE into the **IN USE** state. At the end of the workout a button press or user inactivity may end the session and put the FE into the **FINISHED** state. From **FINISHED**, another button press or elapsed time with no activity will return the FE to the **READY** or **OFF** state.

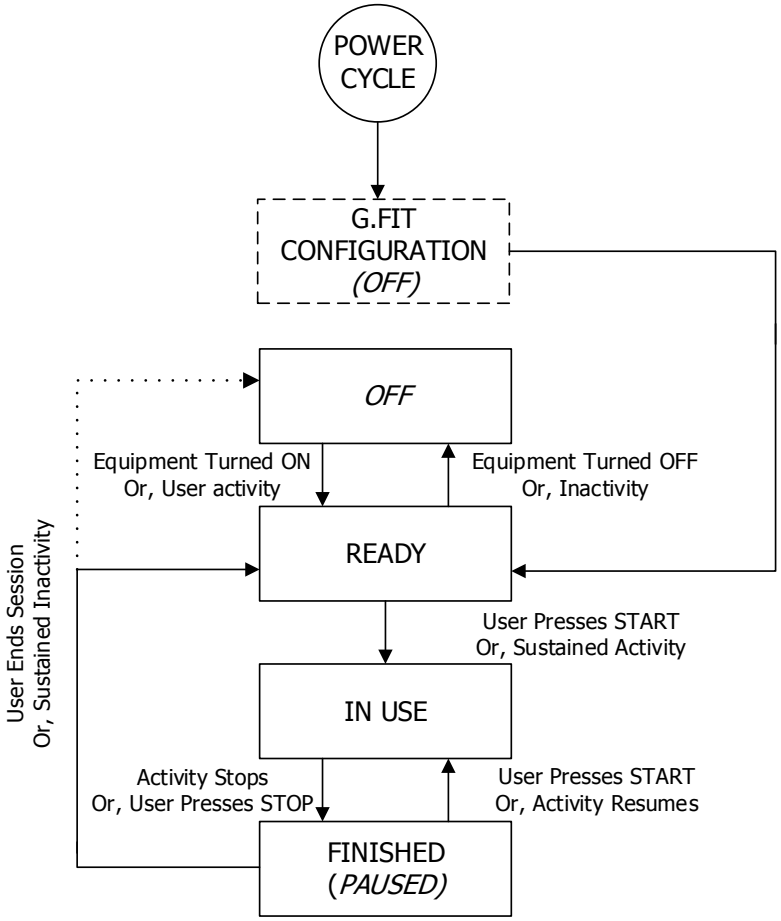


Figure 4-1. Fitness Equipment State Machine

4.1.1 G.FIT CONFIGURATION (OFF)

Each time G.FIT module is power cycled, the G.FIT module resets its settings and enters into a special OFF state; during this initial OFF state the G.FIT module is in its configuration mode. In this initial OFF state the application MCU configures the G.FIT to meet the desired behaviour for the given fitness equipment and required use case. All commands that are used to configure the G.FIT module must be sent in this state. There are some minimum requirements that the G.FIT requires before transitioning from OFF to READY, see section 4.3 for the minimum G.FIT configuration seaquence.

4.1.2 OFF

This state is used to conserve power and it is recommended that the G.FIT module be put into this state after sustained user inactivity.

4.1.2.1 Heart Rate Receive Channels

Both ANT+ and BLE heart rate scan channels are closed.

4.1.2.2 Fitness Equipment Transmit Channels

Both the ANT+ FE-C master channel and BLE FTMS peripheral transmit channels are closed.

4.1.3 READY

This state should be entered when there is initial user activity or when a user has recently ended a session. This state opens scanning channels that search for BLE and ANT+ heart rate monitors, allowing the user to connect their heart rate monitors as they prepare for a workout. It also turns on the ANT+ FE-C master channel and the BLE FTMS peripheral which transmits data describing the user's workout.

4.1.3.1 Heart Rate Receive Channels

ANT+ and BLE heart rate scan channels are opened and will remain open while in state READY. If a heart rate sensor is paired to the G.FIT module, a dedicated channel is opened to establish a connection with that heart rate sensor, both ANT+ and BLE heart rate scan channels will remain open and scan for heart rates when connected to a heartrate monitor.

4.1.3.2 Fitness Equipment Transmit Channels

The ANT+ FE-C master channel is opened and begins transmitting and BLE FTMS peripheral begins advertising. Both channels will remain open and continue transmitting while in state READY. If a phone is paired to the G.FIT module while in state IN USE, a connection is established with the phone and the BLE FTMS peripheral stops advertising.

4.1.4 IN USE

This state should be entered when the user has sustained activity or has indicated a desire to start a workout session by pressing the start button.

4.1.4.1 Heart Rate Receive Channels

The heart rate receive channels have the following behaviour depending on the user interaction.

4.1.4.1.1 No heart rate sensor connected

If the G.FIT module moves to state IN USE and there is no heart rate sensors connected, then the ANT+ and BLE heart rate scan channels will remain open for 30 seconds (by default, this can be adjusted using HR_SetInUseScanTimeout, see section 0). After 30 seconds the ANT+ and BLE heart rate scan channels will close preventing any heart rate sensors from pairing to the G.FIT.

4.1.4.1.2 Heart rate sensor paired while IN USE

If a heart rate sensor is paired to the G.FIT module while in state IN USE, a dedicated channel is opened to establish a connection with that heart rate sensor. After the connection is established to the heart rate sensor, both ANT+ and BLE heart rate scan channels will close immediately.

4.1.4.1.3 Heart rate sensor paired while READY

If a heart rate sensor was paired to the G.FIT while it was in state READY, when the G.FIT transitions to state IN USE both ANT+ and BLE heart rate scan channels will close immediately.

4.1.4.2 Fitness Equipment Transmit Channels

The ANT+ FE-C master channel remains open and continues transmitting while in state IN USE. The BLE FTMS peripheral has the following behaviour depending on the user's interactions.

4.1.4.2.1 No phone connected to peripheral

If the G.FIT modules moves to state IN USE and there is no phone connected to the BLE FTMS peripheral, the peripheral will continue to advertise for 30 seconds (by default, this can be adjusted using HR_SetInUseAdvertisingTimeout, see section 7.3.8). After 30 seconds the BLE FTMS peripheral will stop advertising preventing any phones from pairing to the G.FIT.

4.1.4.2.2 Phone connected while IN USE

If a phone is paired to the G.FIT module while in state IN USE, a connection is established with the phone and the peripheral stops advertising.

4.1.4.2.3 Phone connected while READY

If a phone is paired to the G.FIT module while in state READY, the BLE FTMS peripheral will not advertise in state IN USE.

4.1.5 FINISHED (PAUSED)

This state should be entered when a user presses stop or the user's activity stops. Sustained inactivity should cause the G.FIT module to transition state OFF or READY.

4.1.5.1 Heart Rate Receive Channels

The heart rate receive channels have the following behaviour depending on the user interaction.

4.1.5.1.1 No heart rate sensor connected

If the G.FIT module moves to state FINISHED while there is no heart rate sensors connected, and both the ANT+ and BLE heart rate scan channels are still open (the time spend in state IN USE was less than the InUseScanTimeout), the InUseScanTimeout will reset and the scan channels will remain open for the timeout duration.

4.1.5.1.2 Heart rate sensor paired

If a heart rate sensor was paired to the G.FIT when moving to state FINISHED, both ANT+ and BLE heart rate scan channels will remain closed. The G.FIT will maintain the connection to the heart rate sensor while in state FINISHED.

4.1.5.2 Fitness Equipment Transmit Channels

The ANT+ FE-C master channel remains open and continues transmitting while in state FINISHED. The BLE FTMS peripheral has the following behaviour depending on the user's interactions.

4.1.5.2.1 No phone connected to peripheral

If the G.FIT module moves to state FINISHED while there is no phone connected and the BLE FTMS peripheral is still advertising (the time spend in state IN USE was less than the InUseAdvertisingTimeout), the InUseAdvertisingTimeout will reset and the peripheral will continue to advertise for the timeout duration.

4.1.5.2.2 Phone connected

If a phone is paired to the G.FIT module when moving to state FINISHED, the BLE FTMS peripheral will not advertise in state FINISHED.

4.2 Pairing Heart Rate Sensors

The G.FIT module provides two methods of connecting to heart rate sensors to best fit the required use case:

- Proximity pairing allows the user to pair a heart rate monitor by physically moving the heart rate monitor to within the 'pairing zone' of the G.FIT module.
- Channel ID based pairing give the application the flexibility to select what heart rate sensor to pair to allowing the user to manually select the desired heart rate monitor by scrolling through all available heart rate monitors seen by the G.FIT module.

4.2.1 Proximity Pairing

The G.FIT module will pair with a heart rate monitor automatically based on the proximity to the G.FIT module. In order to successfully pair with a heart rate monitor, the received signal strength received from the heart rate monitor must maintain a specified signal strength during each phase of the connection process. There are three phases in the connection process: searching, pairing, and tracking. See Figure 4-2 for typical zone sizes.

Searching: This phase of the connection process searches for possible heart rate monitor to connect to. It has the strictest RSSI threshold so that a heart rate monitor must be brought in close proximity to the G.FIT module. Once a heart rate monitor is detected in the searching phase it then advances to the Pairing phase.

Pairing: This phase of the connection process ensures that the heart rate monitor found in the searching phase maintains a close proximity to the G.FIT module. This helps prevent accidental connections to other heart rate monitor. This threshold is less strict than searching, but should be defined so that the user must keep the heart rate monitor in the pairing zone during the initial connection process. Once the G.FIT module receives the sufficient number or packets from the heart rate monitor within the defined signal strength, a connection is established and the heart rate monitor advances to the tracking phase.

Tracking: The tracking phase is used to maintain a connection to the heart rate monitor while the user is using the fitness equipment.

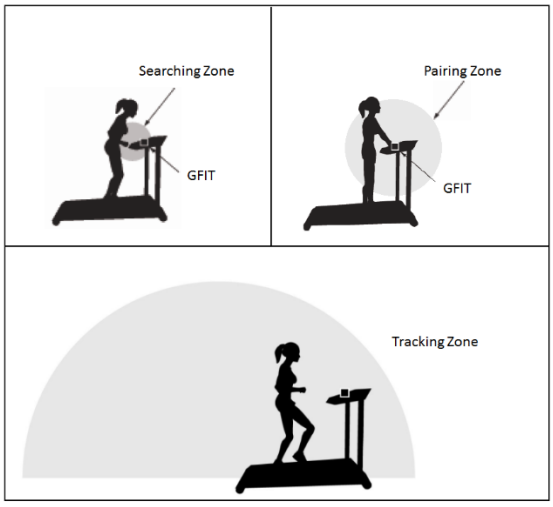


Figure 4-2. G.FIT Pairing Zones

4.2.1.1 RSSI Thresholds

The G.FIT module provides default RSSI thresholds that should work in most use cases. However, it is **highly recommended** that the settings be adjusted to account for difference in module installation, equipment type and desired pairing experience. The application should define RSSI values to have the following behaviour:

Searching: The minimum RSSI before heart rate monitor is considered for pairing. Values below this threshold will not be considered in the pairing algorithm.

Pairing: The signal strength required for the G.FIT module to pair to the heart rate monitor. The heart rate monitor must maintain a signal strength greater than **pairing** for the duration of the paring algorithm in order to successfully pair to the G.FIT module.

The RSSI values shall be set so that **Pairing < Searching.**

4.2.2 Channel ID based pairing

In channel ID pairing, the G.FIT module generates HR events (see section 7.5.2) for every heart rate message received on the ANT and BLE scan channels, and provides a method of connecting to the desired heart rate sensor using the GFIT_HRChannelIDTargetPair serial command (see section 7.3.5). This gives the application the flexibility to pair to any heart rate sensor within range of the G.FIT module. This method of pairing can be used to scan for heart rate sensors in the area and allow the user to select their own heart rate monitor from a list of devices on a screen.

Refer to section 7.5.2 for sequence diagrams of HR Events in Proximity and Channel ID based pairing modes.

4.3 G.FIT Configuration

The G.FIT module needs to be configured while in the initial OFF state. Figure 4-3 shows the order in which the G.FIT module should be configured.

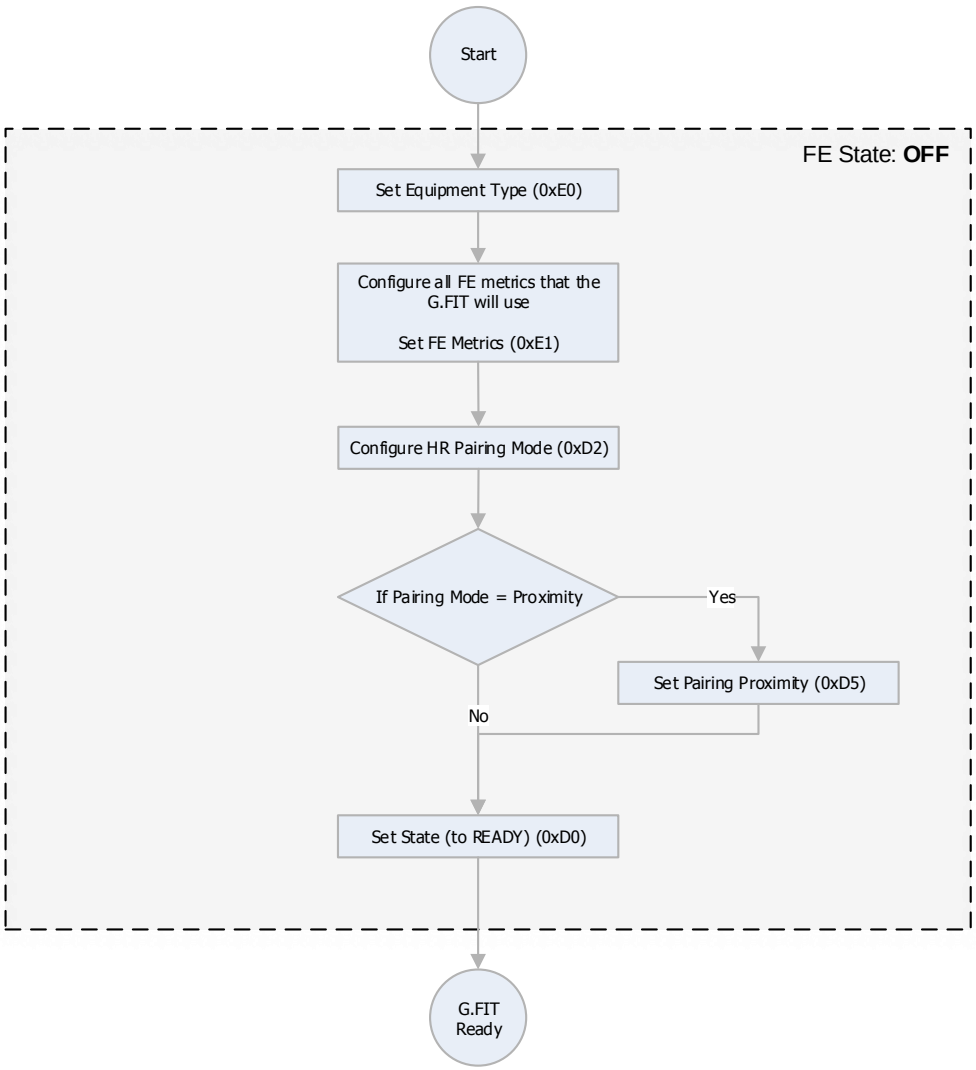


Figure 4-3. Order of G.FIT Configuration Commands

5 Fitness Equipment Console Simulator

5.1 Introduction

The G.FIT simulator is an application that demonstrates how the G.FIT module would operate for a given type of fitness equipment. As a test tool it allows developers to quickly configure the G.FIT module (D52QSKM6IA-A attached to a USB interface board) settings to see their affect.

5.2 Installing ANT USB Interface Board Driver

Before downloading the required software you will need to create an account on http://www.thisisant.com

Download the **ANT USB Interface Board Driver for Windows** package from http://www.thisisant.com/developer/resources/downloads and extract the entire contents onto the hard drive.

Note: The ANT USB Interface Board drivers are unsigned. Systems that require signed drivers for installation (e.g. Windows 8, Windows 10) are required to boot with driver signature enforcement disabled to complete the installation process. This procedure requires restarting the computer and booting up in a special mode. This additional step can be found by conducting a Google search.

Run the USBXpressInstaller.exe file contained in the folder. Install the drivers in the desired location.

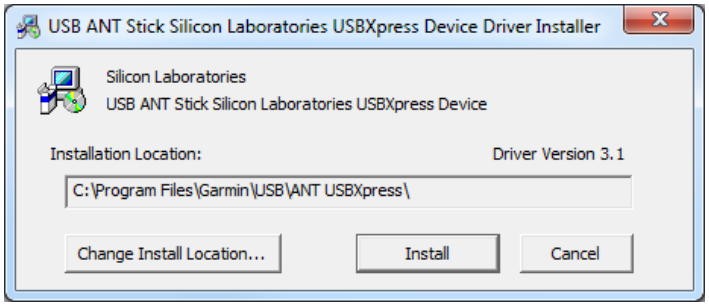


Figure 5-1. Install USB Interface Driver

A warning message may appear that indicates that Windows can't verify the publisher of the driver software. Click 'Install this driver software anyway' to continue.

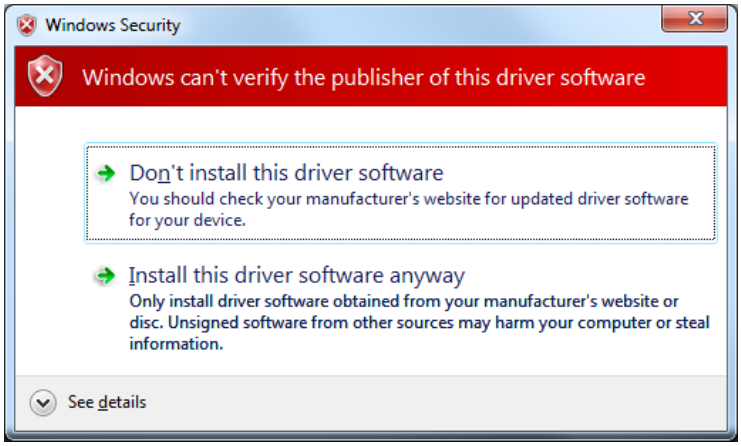


Figure 5-2. Bypass Publisher Verification

A window will indicate the drivers have installed correctly.

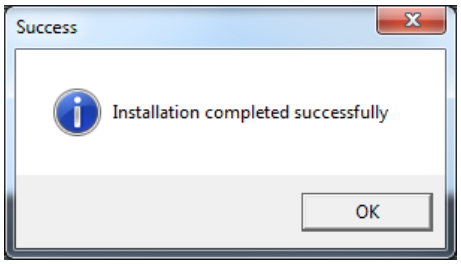


Figure 5-3. Driver Installation Success

Connect the G.FIT module to the USB interface board and insert into a USB port.

Figure 5-4. G.FIT Module Mounted on a USB Interface Board

The Driver Software Installation wizard should pop up and begin a search for drivers; the wizard will indicate the USB device is 'Ready to Use' when it detects the installed drivers on the PC.

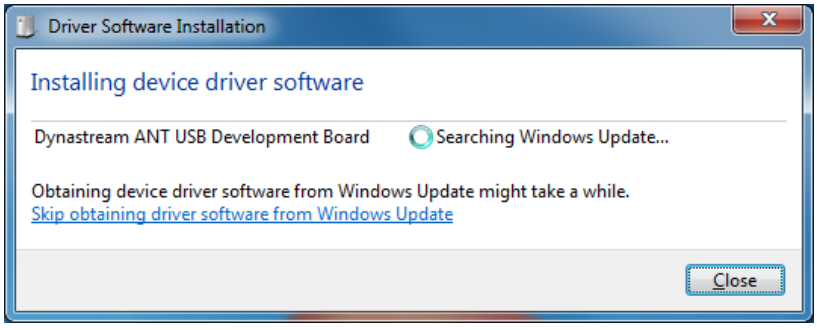


Figure 5-5. Driver Installation Wizard

5.3 Basic Operation

This section provides a high-level description of how to operate the simulated console application. It may be useful to reference section 4.1 (Fitness Equipment States), when reading this basic operation section. Start by opening the application (FitnessEquipmentSimulator.exe), shows the user interface that is provided by the simulator. The menus on the left side of the application are used to modify settings, and to view the commands that are being sent to the G.FIT module. The main center part of the application provides a simulated fitness equipment console implemented using a G.FIT module.

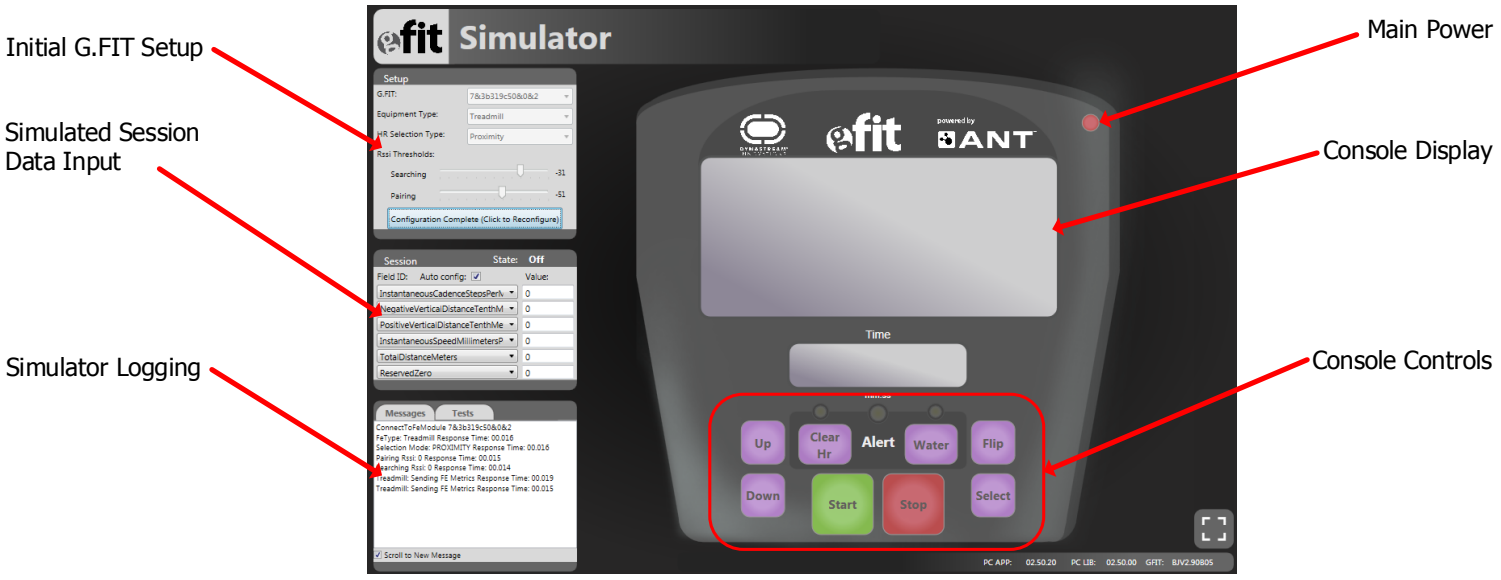


Figure 5-6. Overview of Fitness Equipment Simulator

5.3.1 Initial G.FIT Setup

The Setup section contains the advanced settings related to the G.FIT configuration. These settings must be adjusted before the simulator is turned on. Figure 5-7 shows the settings that must be set before the console main power is turned on. After each setting is selected the user must press the 'Apply G.FIT Setting Button' to complete that portion of the configuration.

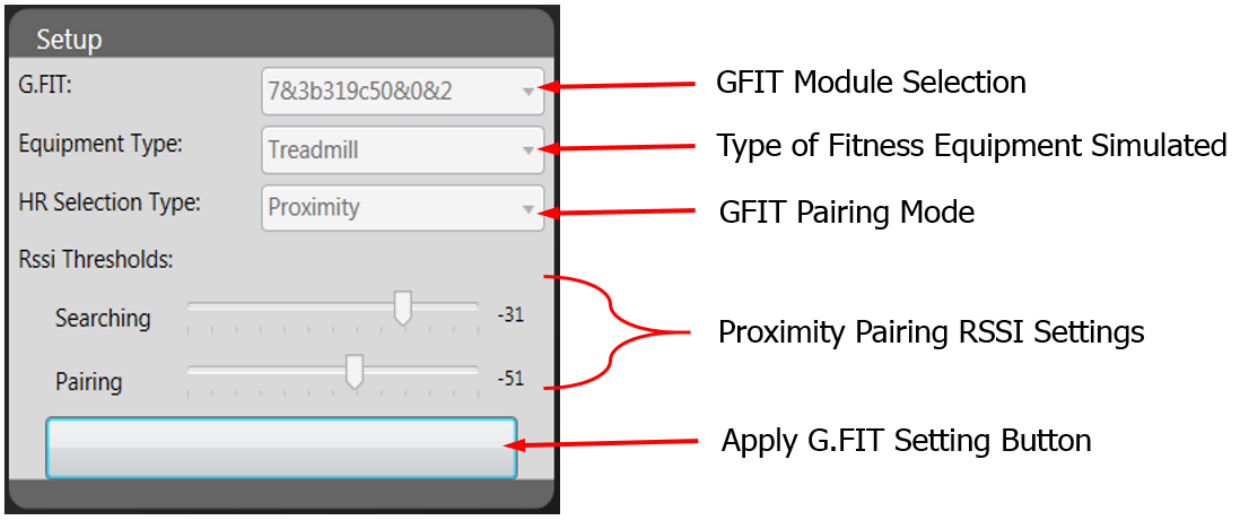


Figure 5-7. Overview of Initial G.FIT Setup

5.3.1.1 G.FIT

This shows which G.FIT module the simulator is connected to. If there is more than one G.FIT module connected to the computer, then select which module to use.

5.3.1.2 Equipment Type

This setting determines which type of fitness equipment is simulated.

5.3.1.3 HR Selection Type

Select the pairing mode used to connect the heart rate monitor to the G.FIT module. See section 4.2 for more details on the best choice for the given use case.

5.3.1.4 RSSI Thresholds

When pairing to a heart rate monitor using proximity mode, the simulated console will pair with a heart rate monitor automatically based on the proximity to the G.FIT module. See section 4.2.1.1 for more details on selecting RSSI values.

The RSSI values shall be set so that **Pairing < Searching.**

5.3.1.5 Configuring FE Metrics

Specific field IDs will need to be selected appropriately depending on what type of fitness equipment type was selected. For details on the setting FE Metrics see section 7.3.10. For a list of field IDs for the given fitness equipment type see section 7.3.10.2.

5.3.1.6 Auto Configuration

This provides the quickest and easiest method of configuring the FE Metrics for the given type of fitness equipment. Ensure that the Auto config checkbox is selected as in Figure 5-8. When selected, the simulator will configure every FE Metrics listed in the appropriate table (section 7.3.10.2) for the given fitness equipment type. Press the 'Apply G.FIT Setting Button' seen in Figure 5-7 to complete the G.FIT configuration.

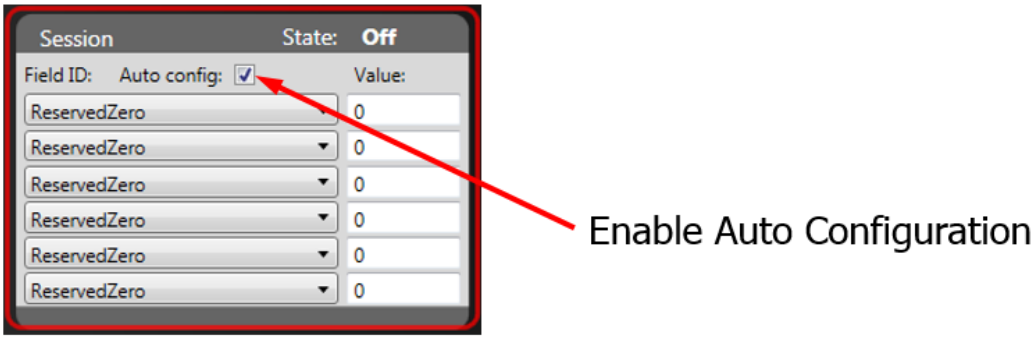


Figure 5-8. Auto Configuration of FE Metrics

5.3.1.7 Manual Configuration

If you would like to simulate a type of fitness equipment that does not fully support all types of FE Metrics, manual configuration is required. Manual configuration allows the user to individually select the FE Metrics that will be supported on the G.FIT. To use manual configuration ensure that the Auto config checkbox is deselected as in Figure 5-9. Field IDs can then be selected to configure which FE Metrics the G.FIT will support. It is important to select the proper field IDs for the given type of fitness equipment (see section 7.3.10.2). When a field ID is selected verify that it was sent properly and that there were no errors returned. When finished configuring the FE metrics manually, press the 'Apply G.FIT Setting Button' seen in Figure 5-7 to complete the G.FIT configuration.

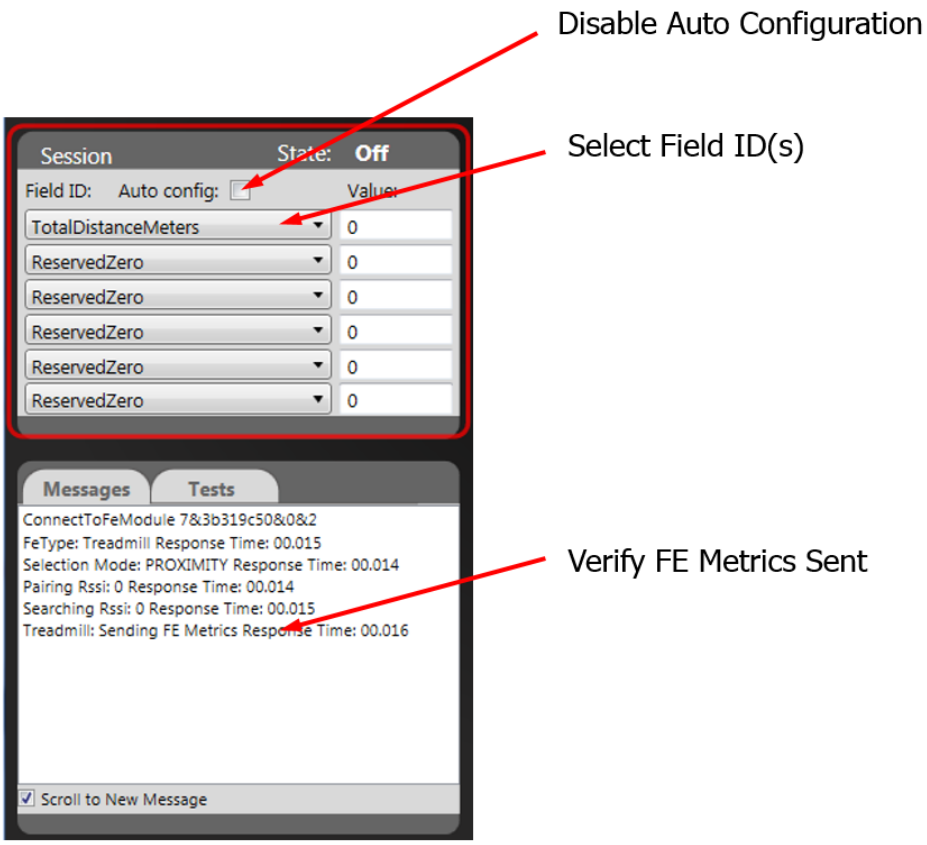


Figure 5-9. Manual Configuration of FE Metrics

5.3.2 Turning on the Simulator

To turn on the simulated console, press the red circle button  in the upper right section of the console. Turning on the console puts it into FE state **READY**. This has the following effects:

- Turns on the ANT+ FE-C channel
- Begins searching for ANT+ heart rate monitors
- Begins searching for BLE heart rate monitors
- Turns on the BLE FTMS peripheral

5.3.3 Pairing a Heart Rate Monitor

Depending on the **HR Selection Type** value specified in the **Setup** section of the application, the pairing of a HR strap will be completed in one of two ways.

5.3.3.1 Proximity Pairing

The simulated console will pair with a heart rate monitor automatically based on the proximity to the G.FIT module. See section 4.2.1 for more details.

5.3.3.2 Pairing from a List of Devices (Channel ID based pairing)

This allows the user to select which heart rate monitor to connect to via a user interface. In list mode the simulated console will display all the heart rate monitors in range of the G.FIT module and allow the user to select which heart rate monitor they want to pair to. See section 4.2.2 for more details.

5.3.3.2.1 Selecting a Heart Rate Monitor in List

When there is more than one heart rate monitor displayed on the screen, the user will have to select the one that they are using. Pressing the **Up** or **Down** buttons on the console will change which heart rate monitor is highlighted on the screen. Up to four heart rate monitors can be displayed on the screen at once. When the user has their heart rate monitor highlighted, they must press the **Select** button to connect to it.

5.3.4 Successfully Paired Heart Rate Monitor

After a heart rate monitor is paired, the connection is identified by the yellow light on the console as seen in Figure 5-10.

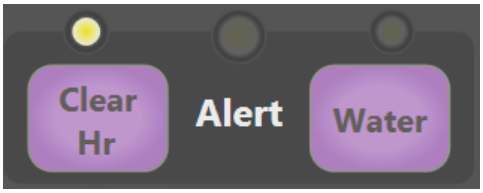


Figure 5-10. Indicator for a Connected Heart Rate Monitor

The current heart rate monitor data is displayed on the console display. The user is given information on their current heart rate, the type of heart rate monitor being used (ANT/BLE), the device ID, and the RSSI.

5.3.5 Disconnecting a Paired Heart Rate Monitor

If the user wants to disconnect their heart rate monitor or possibly another heart rate monitor that was accidentally connected to, they can press the **Clear Hr** button. After a heart rate monitor has been disconnected the simulated console returns to the pairing screen.

5.3.6 Starting a Workout/Session

A session can be started with or without a heart rate monitor paired. To start a session, press the **Start** button. The state will change from **READY** to **IN USE.**

5.3.6.1 Modifying FE Metrics

The simulated data that is sent over ANT+ FE-C channel and BLE FTMS peripheral can be modified at any time during the session. Simply change any of the values within the 'Session' section of the application. To update a FE Metric, simply select the required field ID and adjust its value. The simulator will allow up to 6 values to be adjusted simultaneously. See

Figure 5-11 for an example of setting some FE metrics for a treadmill. When in state IN USE the PC simulator will automatically update the FE Metric elapsed time to match the value that is shown in the simulated console in addition to the other FE metrics specified by the user.

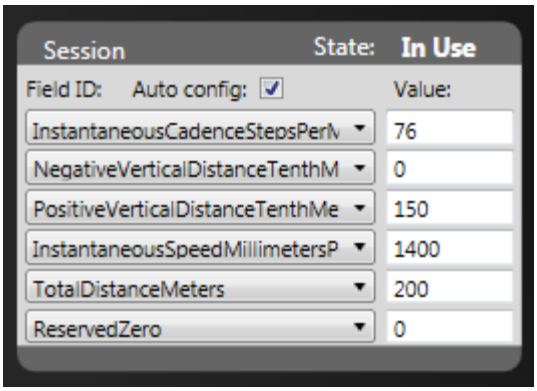


Figure 5-11. Modifying FE Metrics

5.3.6.2 Sending Commands to a Leader Board

These are custom messages that are sent from the console to the leaderboard and are not part of the ANT+ FE-C profile.

5.3.6.2.1 Request Water

Press the **Water** button to trigger the water request. Press the **Water** button again to cancel the water request.

5.3.6.2.2 Flip Users Leaderboard Tile

Press the **Flip** button to send a tile flip command. This allows the user to flip their tile on the leaderboard is to display more data.

5.3.7 Pausing/Resuming a Session

Active sessions (i.e. FE state **IN USE**) can be paused by pressing the **Stop** button once. Doing so pauses the session and puts the console into FE state **FINISHED (PAUSED)**. To resume the session the user will press the **Start** button and the console will go back into FE state **IN USE**.

5.3.8 Ending a Session

If the user wishes to end their session from:

- an active session (FE State **IN USE**) then the user will press the **Stop** button twice.
- a paused session (FE State **FINISHED (PAUSED)**) then the user will press the **Stop** button once.

Once the session has ended the console will go back into FE state **READY.**

If there was a connected heart rate monitor when the session was ended, that heart rate monitor will be disconnected.

5.3.9 Turning off the Simulator

To turn off the simulated console, press the red circle button  in the upper right section of the console. To successfully turn off the simulator, the G.FIT module must be in FE state **READY** or **FINISHED (PAUSED)**. Turning off the console puts it into FE state **OFF**. When in FE state **OFF**, the G.FIT can be reconfigured as described in section 0.

5.3.10 Enabling/Disabling Full Screen Mode

To enter full screen, simply press:

To exit out of full screen mode, hover the mouse over the lower right section to make the exit full screen button appear, press this button to exit full screen.

6 Dual frequency FE-C channel

G.FIT works around WiFi interference to ensure large groups of 50+ devices can connect. This means the G.FIT transmits ANT+ FE-C on 2 radio frequencies - one is ANT+ (2457MHz) and the other is 2472MHz. The two frequencies transmit the same information for redundancy. The second channel (2472MHz) has the same ANT+ FE-C channel parameters.

6.1 Using SimulANT+ to View Transmitted Data

SimulANT+ can be used to view and parse the data being transmitted by the G.FIT modules on the ANT+ frequency (2457MHz). SimulANT+ is available for download from https://www.thisisant.com/developer/resources/downloads/. Before downloading SimulANT+ you will need to create an account on http://www.thisisant.com. For details on using SimulANT+ see the user guide provided in the SimulANT+ download package.

Ensure that the USB-m is inserted into a USB port before the SimulANT+ application is opened. To view data being transmitted by the G.FIT module, select a Fitness Equipment Display simulator. Turn on the Fitness Equipment Display to start receiving data from the G.FIT module.

Figure 6-1 shows the parsed data being received from the G.FIT module.

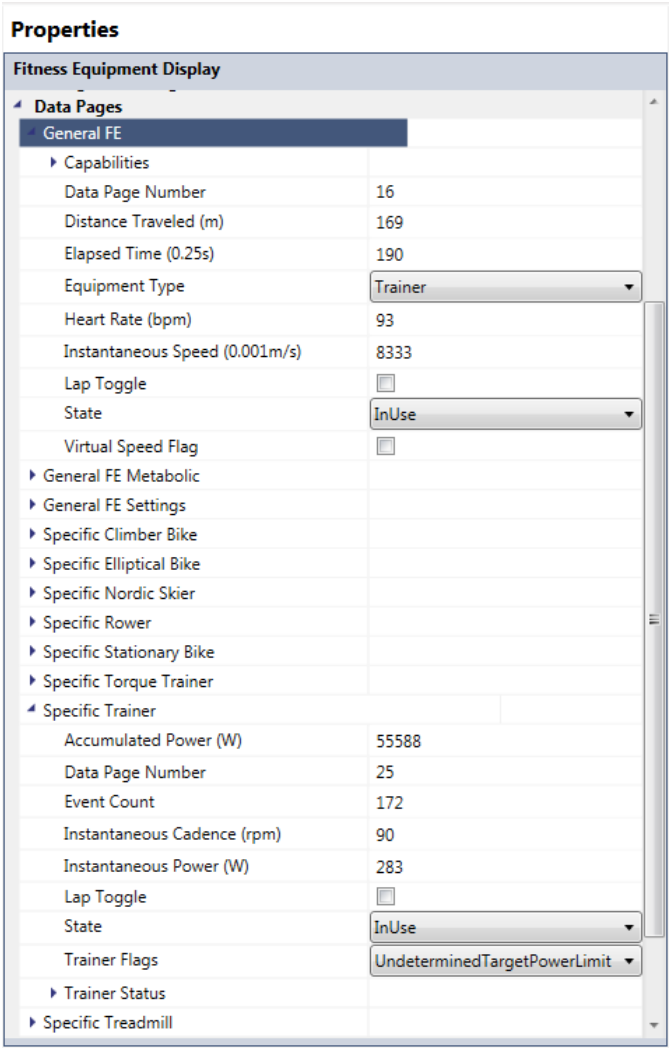


Figure 6-1. Viewing Received Data Pages on SimulANT+

6.2 Tracking FE-C Channel with CIQ Application

The CIQ application allows a user to connect to the G.FIT module ANT+ FE-C channel. This allows the user to view their workout in real time on their watch. At the end of a workout, the user can save their session on their device. Using a CIQ application can allow the user to track their workout from the ANT+ FE-C channel and later upload that workout to Garmin Connect.

6.2.1 Installing CIQ Application on a Compatible Garmin Watch

Connect the watch to a USB port on your computer. Select the appropriate PRG file for your specific device. Using a file browser place the PRG file into your device's GARMIN/APPS directory.

6.2.2 Opening the Application

Using the devices menu, find and select the newly installed CIQ application named **G.FIT Display**.

6.2.3 Connecting to the G.FIT ANT+ FE-C Channel

The CIQ application only searches for G.FIT modules that are in the READY state. If there is more than one G.FIT module in READY state than all available devices will be displayed in a list. Select the desired G.FIT module from the list.

6.2.4 Viewing Live Workout Data

When the G.FIT transitions from READY to IN USE the CIQ application detects that state transition and starts recording the user's workout. The user's workout is displayed on the watches screen. The user can pause/resume the workout timer by tapping the screen (on vivoactiveHR).

6.2.5 Saving Users Workout Session

If the back button is pressed the user has the option to save their data. Select yes to save the activity to the watch.

6.2.6 Viewing Workout Data

The app records FIT files during the workout. After a FIT file is saved, it can be found in the "GARMIN\ACTIVITY" subdirectory on the watch. The "Monkeygraph" section of the ConnectIQ Eclipse plugin can be used to view the workout data (see: https://developer.garmin.com/connect-iq/programmers-guide/positioning-sensors/). Alternatively, this data can be viewed by converting the FIT file to a CSV file using tools found in the "FIT SDK" from www.thisisant.com. Once a Connect IQ app is written and uploaded as a beta version to Garmin Connect, the data can be viewed through the Garmin Connect Dashboard.

7 Serial Interface

7.1 Using extended serial messages

To interface with the G.FIT module, the application MCU and the G.FIT module use the extended serial message protocol. In addition to using 2 byte message ID's, the extended serial protocol is more dynamic than the standard ANT serial protocol. For all commands outside of G.FIT functionality, the standard serial protocol still applies, as detailed by the "ANT Message Protocol and Usage" document.

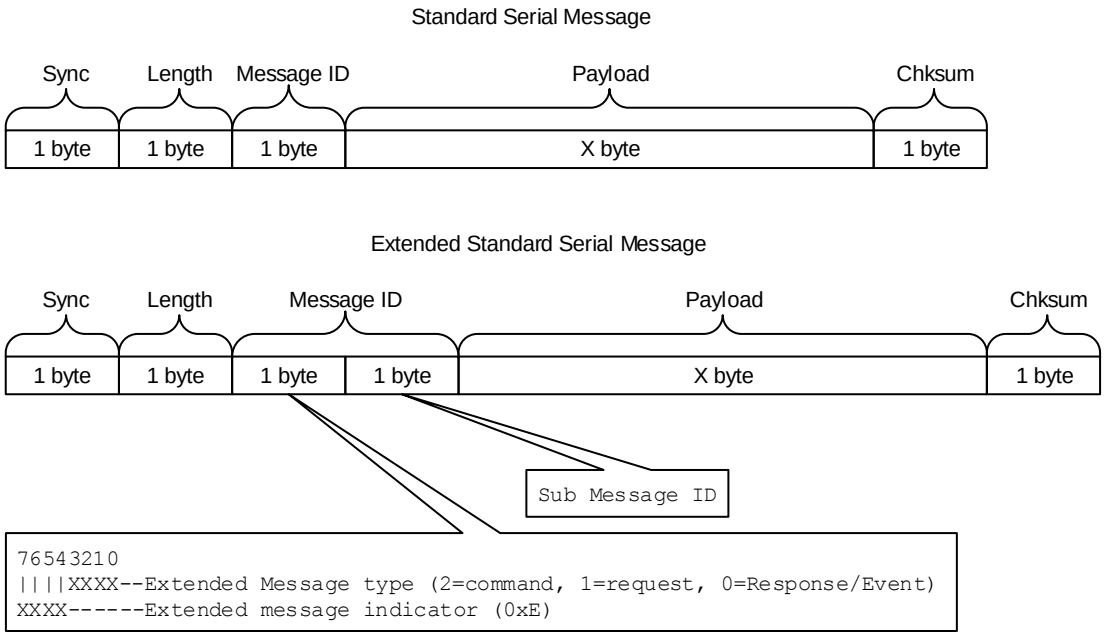


Figure 7-1. Serial Message General Packet Structure

The extended serial protocol defines three types of messages – commands, requests and responses/events. Compared to a standard serial message, an extended serial message is indicated by setting the top nibble of the message ID to 0xE. The type of extended message is indicated in the following nibble (Command = 0xE2, Request = 0xE1 and Response/Event = 0xE0). The following byte is the Sub-Message ID, followed by the payload of the message.

The checksum for a standard and extended serial message is calculated as the XOR of all bytes including the sync byte. The length of a standard message includes the number of bytes in the payload of the message, whereas for extended serial messages the **length** includes the **payload + 1 byte** (to account for the Sub-Message ID). As with the standard serial protocol, the use of optional padding bytes is also recommended for systems that are slow to react to hardware flow control (when using asynchronous serial interface).

In general, a command sent by the MCU to the G.FIT module elicits a response. A request elicits a command message and an event is an unsolicited message from the G.FIT module to the MCU. This behavior is illustrated in Figure 7-2.

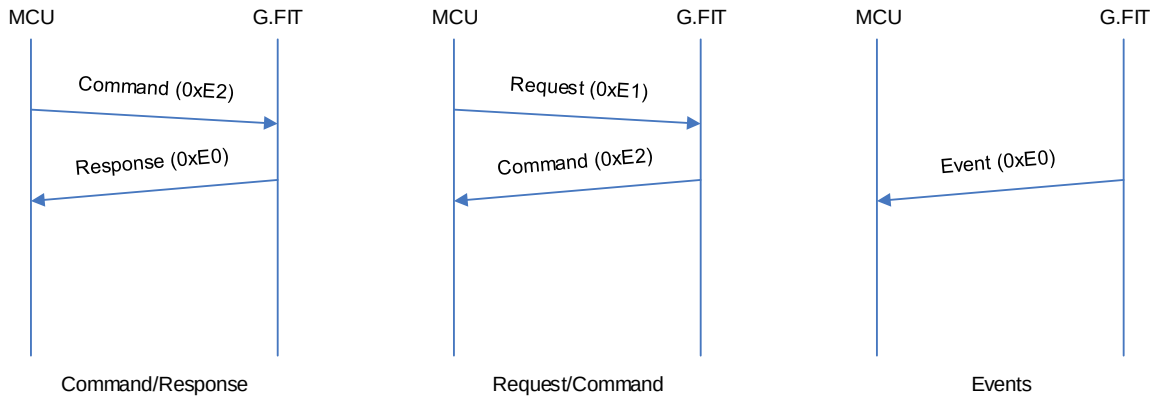


Figure 7-2. Extended Serial Message Types

7.1.1 Commands (0xE2)

Extended serial commands are used to execute specific commands on the G.FIT module, or to provide information about the settings/state of the G.FIT module. When sent by the application MCU they elicit a response from the G.FIT module. If a message request is made by the application MCU, then the G.FIT module will send a command message in response.

The structure of a command is identical to that described by Figure 7-1, with the extended message type set to 2. The specific command to execute is indicated by the second byte of the Message ID (Sub-Message ID). The payload is specific to each command and may be completely empty. Figure 7-3 describes an extended serial command.

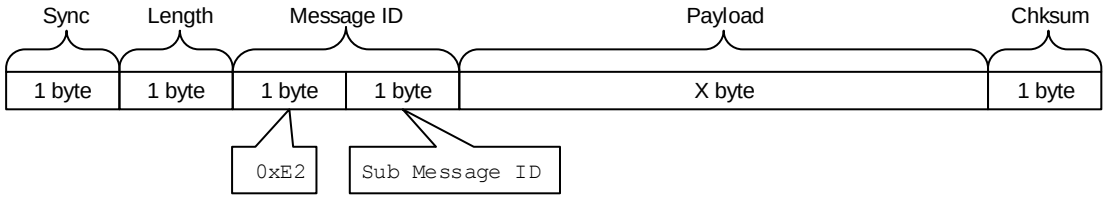


Figure 7-3. Extended Serial Command

7.1.2 Requests (0xE1)

Extended serial requests are used by the application MCU to request information from the G.FIT module. The information requested comes to the MCU in the form of a command.

The structure of a request command is identical to that described by Figure 7-4. The first two bytes of the payload indicate what command is being requested.

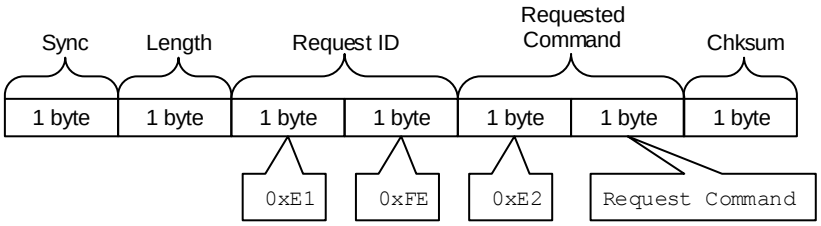


Figure 7-4. Extended Serial Request

7.1.3 Responses/Events (0xE0)

Responses are used to indicate the success or failure of a command. Events are a special form of response messages. They are unsolicited messages that come from the G.FIT module and indicate some type of information which the application MCU must handle.

The structure of a Response/Event message is identical to that described by Figure 7-1. The payload of responses includes the ID of the message being responded to as well as the response code and associated payload.

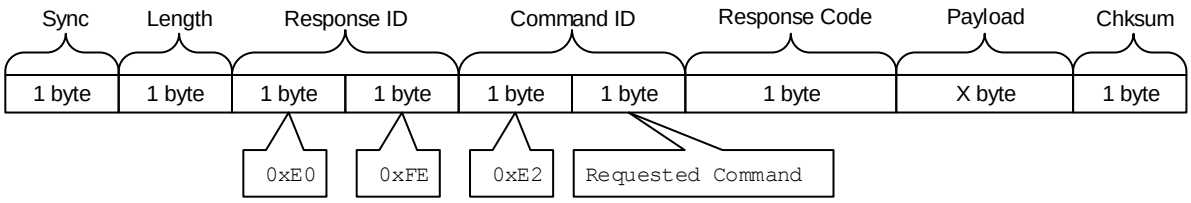


Figure 7-5. Extended Serial Response

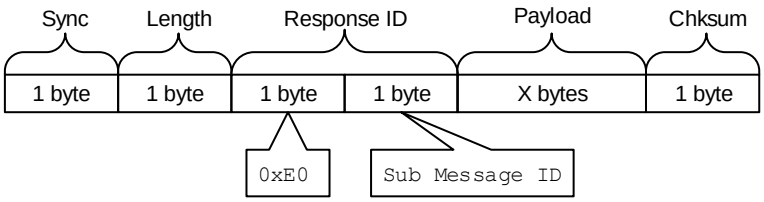


Figure 7-6. Extended Serial Event

7.2 G.FIT Response Codes

When a command is sent by the application MCU, the G.FIT module will respond with a response message. Within this response message is a response code. This response code informs the MCU of the success/failure of the command. Table 7-1 describes the meaning of the different response codes.

Table 7-1. G.FIT Response Codes

Code #	Name	Description
0x00	GFIT_SUCCESS	The command performed successfully
0x01	GFIT_ERROR INCORRECT_STATE	Many commands are only valid in specific states. This error is returned when command is called in a state where the command is not supported.
0x02	GFIT_ERROR INCORRECT_CONFIGURATION	Some commands require the G.FIT to be configured appropriately before the command can properly perform its task. This error is returned if a command is called to perform a task that the G.FIT is currently not configured to perform.
0x03	Reserved	This error code is not used and is reserved for future use.
0x04	GFIT_ERROR INVALID_DEVICE_ID	The G.FIT does not allow the G.FIT device ID to be set to zero. This error is returned if attempting to set a 4 byte device ID that has the two least significant bytes equal to zero.
0x05	Reserved	This error code is not used and is reserved for future use.
0x06	Reserved	This error code is not used and is reserved for future use.
0x07	GFIT_ERROR INVALID_COMMAND_VALUE	If a command is sent with an invalid parameter setting than this error will be returned. Check which values are supported for the command that is being sent.
0x08	GFIT_ERROR NO_SENSOR	This error is returned if a disconnect command is sent and there is no wireless heart rate sensor to disconnect.
0x09	GFIT_ERROR MODULE_NOT_SUPPORTED	This error is returned if the G.FIT library is used on a module that was not intended to run the G.FIT library.
0x0A	GFIT_ERROR INVALID_LICENSE_KEY	This error is returned if using the G.FIT library and an incorrect licence key is used.
0x0B	GFIT_ERROR INVALID_FIELD_ID_FOR_EQUIPMENT_TYPE	Each fitness equipment have unique FE Metrics specific to that type of equipment. If a field ID is being set that is not compatible for the configured fitness equipment type this error is returned. See section 7.3.10.2 for tables of valid field IDs for given fitness equipment types.
0x0C	GFIT_ERROR INCORRECT_FE_METRIC_PAYLOAD	This error is returned if the G.FIT is unable to decode the GFIT_SetFEMetrics serial command. This could be caused by incorrectly sized fields, or using invalid field IDs. See section 7.3.10 for more details on constructing the GFIT_SetFEMetrics serial command.
0x0D	GFIT_ERROR INVALID_EQUIPMENT_TYPE	This error is returned when attempting to configure the G.FIT to a fitness equipment type that is not supported by the G.FIT.
0x0E	GFIT_ERROR EQUIPMENT_TYPE_NOT_SET	The G.FIT needs to have an equipment type set before any FE Metrics can be configured towards a given equipment type. This error is returned if attempting to move to READY or setting FE metrics without having an equipment type set.

0x0F	GFIT_ERROR NO_FIELD_IDS_CONFIGURED	This error occurs if you attempt to transition from OFF to READY without setting/configuring any FE Metrics. FE Metrics must be configured on the G.FIT before the initial transition to READY to indicate which FE Metrics are supported by the fitness equipment. See section 7.3.10 for more details.
0x10	GFIT_ERROR FIELD_ID_NOT_CONFIGURED	This error occurs when attempting to update a FE Metric that was not configured on the G.FIT. FE Metrics must be configured before the initial transition from OFF to READY to indicate which FE Metrics are supported by the fitness equipment. See section 7.3.10 for more details.
0x11	GFIT_ERROR INVALID_MFG_SPECIFIC_PAGE_NUMBER	This error is returned if sending a manufacturer page over ANT+ FE-C that does not fall within the manufacturer page range. Manufacturer pages should be greater than 0xE0.
0x12	GFIT_ERROR INVALID_LENGTH	If sending a command with an invalid length then this error will be returned.
0x13	GFIT_ERROR INVALID_BIKE_POWER_USAGE	When equipment type is set to trainer, to comply with the ANT+ FE-C profile, the G.FIT enforces that if either instantaneous power or accumulated power is updated, that both instantaneous power and accumulated power be included in the same GFIT_SetFEMetrics serial command.
0x14	GFIT_ERROR HR_SENSOR_PAIRED	This error is returned if attempting to set the HR via a command and there is already a wireless HR sensor connected, or if attempting to target pair to a HR sensor and there is already one connected.
0x15	GFIT_ERROR EQUIPMENT_TYPE_ALREADY_SET	The fitness equipment type should only be set once in a power cycle. If the equipment type is changed or it is set twice than this error will be returned.

7.3 Commands (0xE2)

7.3.1 GFIT_SetState (Message ID = 0xD0)

Control Command: Commands the module to change states to the specified value. Only valid transitions are permitted. (See section 4.1 for more details)

Valid State: OFF, READY, IN USE, FINISHED

Table 7-2. GFIT_SetState Command Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x02
2	Message ID	1 byte	Set to 0xE2 (command)
3	Sub-Message ID	1 byte	Set to 0xD0 (GFIT_SetState)
4	Payload	1 byte	Set to represent the state transition: 0x01 – Transition to OFF 0x02 – Transition to READY 0x03 – Transition to IN USE 0x04 – Transition to PAUSED/FINISHED
5	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.3.1.1 Consecutive State Transitions

It is recommended that the application waits a period greater than 200ms before performing consecutive state transitions (i.e. Off → READY → OFF). This period allows the GFIT to perform state sensitive SoftDevice operations.

Table 7-3. GFIT_SetState Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x05
2	Response ID	2 bytes	0xE0 (response)
3			0xFE (G.FIT response)
4	Command ID	2 bytes	0xE2 (command)
5			0xD0 (GFIT_SetState)
6	Response Code	1 byte	0x00 - GFIT_SUCCESS 0x01 - GFIT_ERROR_INCORRECT_STATE 0x07 - GFIT_ERROR_INVALID_COMMAND_VALUE 0x12 - GFIT_ERROR_INVALID_LENGTH See Table 7-1 for descriptions.
7	Payload	1 byte	Indicates **current state** if response code is GFIT_ERROR_INCORRECT_STATE: Indicates **updated state** if response code is GFIT_SUCCESS: 0x00 – State is INVALID 0x01 – State is OFF 0x02 – State is READY 0x03 – State is IN USE 0x04 – State is PAUSED/FINISHED
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.3.2

7.3.2 GFIT_ConfigureHRPairingMode (Message ID = 0xD2)

Configuration Command: Choose between automated proximity pairing or channel ID based pairing (where the application will choose to pair to a specific heart rate sensor using its ID). See section 4.2 for more details.

Valid State(s): OFF

Default: Proximity Based

Table 7-4. GFIT_ConfigureHRPairingMode Command Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x02
2	Message ID	1 byte	Set to 0xE2 (command)
3	Sub-Message ID	1 byte	Set to 0xD2 (GFIT_ConfigureHRPairingMode)
4	Payload	1 byte	HR Pairing Mode: Set to 0x00 for **proximity** based pairing (default setting). Set to 0x01 for **channel ID** based pairing.
5	Checksum	1 byte	XOR of all previous bytes including the sync byte.

Table 7-5. GFIT_ConfigureHRPairingMode Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x05
2	Response ID	2 bytes	0xE0 (response)
3			0xFE (G.FIT response)
4	Command ID	2 bytes	0xE2 (command)
5			0xD2 (GFIT_ConfigureHRPairingMode)
6	Response Code	1 byte	0x00 - GFIT_SUCCESS 0x01 - GFIT_ERROR_INCORRECT_STATE 0x07 - GFIT_ERROR_INVALID_COMMAND_VALUE 0x12 - GFIT_ERROR_INVALID_LENGTH See Table 7-1 for descriptions.
7	Payload	1 byte	Current State: 0x00 – State is INVALID 0x01 – State is OFF 0x02 – State is READY 0x03 – State is IN USE 0x04 – State is PAUSED/FINISHED
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.3.3

7.3.3 GFIT_HRSetInUseScanTimeout (Message ID = 0xD3)

Configuration Command: Sets the scanning timeout for the scan channels once the module transitions to the IN USE state. After the timeout, the scan channels will close. Setting a timeout allows the G.FIT to continue searching for a sensor while in state IN USE for the specified time. The pairing behavior during the search depends on the configured HR pairing mode.

Valid State: OFF

Default: 30 second in use scan timeout.

Table 7-6. GFIT_HRSetInUseScanTimeout Command Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x02
2	Message ID	1 byte	Set to 0xE2 (command)
3	Sub-Message ID	1 byte	Set to 0xD3 (GFIT_HRSetInUseScanTimeout)
4	Payload	1 byte	Timeout Value: Set to 0 for no timeout (Scan channels stop when transitioning to IN USE state). Set to a value from 1-254 to set a timeout in units of 2.5 seconds Set to 255 for infinite timeout.
5	Checksum	1 byte	XOR of all previous bytes including the sync byte.

Table 7-7. GFIT_HRSetInUseScanTimeout Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x05
2	Response ID	2 bytes	0xE0 (response)
3			0xFE (G.FIT response)
4	Command ID	2 bytes	0xE2 (command)
5			0xD3 (GFIT_HRSetInUseScanTimeout)
6	Response Code	1 byte	0x00 - GFIT_SUCCESS 0x01 - GFIT_ERROR_INCORRECT_STATE 0x12 - GFIT_ERROR_INVALID_LENGTH See Table 7-1 for descriptions.
7	Payload	1 byte	Current State: 0x00 – State is INVALID 0x01 – State is OFF 0x02 – State is READY 0x03 – State is IN USE 0x04 – State is PAUSED/FINISHED
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.3.4

7.3.4 GFIT_SetPairingProximity (Message ID = 0xD5)

Configuration Command: Sets the RSSI pairing thresholds for the HR scan and proximity pairing. See section 4.2 and section 4.2.1.1 for more details on selecting RSSI values.

Required Configuration: HR Pairing Mode == Proximity.

Valid State: OFF

Table 7-8. GFIT_SetPairingProximity Command Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x03
2	Message ID	1 byte	Set to 0xE2 (command)
3	Sub-Message ID	1 byte	Set to 0xD5 (GFIT_SetPairingProximity)
4 – 5	Payload	2 bytes	Byte 4 - **Search** Threshold RSSI Value (dBm) Minimum RSSI before heart rate monitor is considered for pairing. Values below this threshold will not be considered in the pairing algorithm.
			Byte 5 - **Pairing** Threshold RSSI Value (dBm) Received signal strength required for the G.FIT module to pair to the heart rate monitor. The G.FIT module must continue to receive messages with received signal strength from the heart rate monitor greater than the Pairing Threshold value before G.FIT module pairs to sensor.
6	Checksum	1 byte	XOR of all previous bytes including the sync byte.

The response to GFIT_SetPairingProximity command will be returned as described in Table 7-9.

Table 7-9. GFIT_SetPairingProximity Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x05
2	Response ID	2 bytes	0xE0 (response)
3			0xFE (G.FIT response)
4	Command ID	2 bytes	0xE2 (command)
5			0xD5 (GFIT_SetPairingProximity)
6	Response Code	1 byte	0x00 - GFIT_SUCCESS 0x01 - GFIT_ERROR_INCORRECT_STATE 0x02 - GFIT_ERROR_INCORRECT_CONFIGURATION 0x12 - GFIT_ERROR_INVALID_LENGTH See Table 7-1 for descriptions.
7	Payload	1 byte	Current State: 0x00 – State is INVALID 0x01 – State is OFF 0x02 – State is READY 0x03 – State is IN USE 0x04 – State is PAUSED/FINISHED
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.3.5 GFIT_HRChannelIDTargetPair (Message ID = 0xD6)

Control Command: Commands the module to only establish a channel with a specific device channel ID in the scan. Module must be set to use Channel ID based pairing mode.

Required Configuration: HR Pairing Mode == Channel ID.

Valid State: READY

Table 7-10. GFIT_HRChannelIDTargetPair Command Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x08
2	Message ID	1 byte	Set to 0xE2 (command)
3	Sub-Message ID	1 byte	Set to 0xD6 (GFIT_HRChannelIDTargetPair)
4 - 10	Payload	7 bytes	Byte 4 – indicates the protocol to use for pairing. Set to 0x00 to use ANT protocol Set to 0x01 to use BLE protocol
			Set bytes 5-10 as described in Table 7-11
11	Checksum	1 byte	XOR of all previous bytes including the sync byte.

Table 7-11. ANT and BLE Channel IDs

Byte #	Length	If ANT protocol is used	If BLE protocol is used
5	1 byte	Set to device number LSB	Set to 6 Byte MAC Address LSB
6	1 byte	Set to device number MSB	…
7	1 byte	Set to transmission type	…
8	1 byte	Set to 0x00	…
9	1 byte	Set to 0x00	…
10	1 byte	Set to 0x00	Set to 6 Byte MAC Address MSB

The response to GFIT_HRChannelIDTargetPair command will be returned as described in Table 7-12.

Table 7-12. GFIT_HRChannelIDTargetPair Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x05
2	Response ID	2 bytes	0xE0 (response)
3			0xFE (G.FIT response)
4	Command ID	2 bytes	0xE2 (command)
5			0xD6 (GFIT_HRChannelIDTargetPair)
6	Response Code	1 byte	0x00 - GFIT_SUCCESS 0x01 - GFIT_ERROR_INCORRECT_STATE 0x02 - GFIT_ERROR_INCORRECT_CONFIGURATION 0x07 - GFIT_ERROR_INVALID_COMMAND_VALUE 0x12 - GFIT_ERROR_INVALID_LENGTH 0x14 - GFIT_ERROR_HR_SENSOR_PAIRED See Table 7-1 for descriptions.
7	Payload	1 byte	Current State: 0x00 – State is INVALID 0x01 – State is OFF 0x02 – State is READY 0x03 – State is IN USE 0x04 – State is PAUSED/FINISHED
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.3.6 GFIT_HRDisconnect (Message ID = 0xD7)

Control Command: Commands the module to disconnect from an already paired/connected heart rate monitor.

Valid State: READY, IN USE, FINISHED

Table 7-13. GFIT_HRDisconnect Command Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x01
2	Message ID	1 byte	Set to 0xE2 (command)
3	Sub-Message ID	1 byte	Set to 0xD7 (GFIT_HRDisconnect)
4	Checksum	1 byte	XOR of all previous bytes including the sync byte.

Table 7-14. GFIT_HRDisconnect Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x05
2	Response ID	2 bytes	0xE0 (response)
3			0xFE (G.FIT response)
4	Command ID	2 bytes	0xE2 (command)
5			0xD7 (GFIT_HRDisconnect)
6	Response Code	1 byte	0x00 - GFIT_SUCCESS 0x01 - GFIT_ERROR_INCORRECT_STATE 0x08 - GFIT_ERROR_NO_SENSOR 0x12 - GFIT_ERROR_INVALID_LENGTH See Table 7-1 for descriptions.
7	Payload	1 byte	Current State: 0x00 – State is INVALID 0x01 – State is OFF 0x02 – State is READY 0x03 – State is IN USE 0x04 – State is PAUSED/FINISHED
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.3.7 GFIT_EnterBootloader (Message ID = 0xD8)

Control Command: Restarts the G.FIT module and starts up the bootloader with the specified transport type. Entering the bootloader allows you to update the G.FIT FW and/or the application that is currently loaded on the module.

Valid State: OFF

Table 7-15. GFIT_EnterBootloader Command Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x02
2	Message ID	1 byte	Set to 0xE2 (command)
3	Sub-Message ID	1 byte	Set to 0xD8 (GFIT_EnterBootloader)
4	Payload	1 byte	Transport Type: Set to 0x01 to update via **BLE** Set to 0x02 to update via **UART**
5	Checksum	1 byte	XOR of all previous bytes including the sync byte.

If the GFIT_EnterBootloader command is successful, it will not return a GFIT_SUCCESS response message. The module will reboot in bootloader mode.

Table 7-16. GFIT_EnterBootloader Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x05
2	Response ID	2 bytes	0xE0 (response)
3			0xFE (G.FIT response)
4	Command ID	2 bytes	0xE2 (command)
5			0xD8 (GFIT_EnterBootloader)
6	Response Code	1 byte	0x01 - GFIT_ERROR_INCORRECT_STATE 0x02 - GFIT_ERROR_INCORRECT_CONFIGURATION 0x07 - GFIT_ERROR_INVALID_COMMAND_VALUE 0x12 - GFIT_ERROR_INVALID_LENGTH See Table 7-1 for descriptions.
7	Payload	1 byte	Current State: 0x00 – State is INVALID 0x01 – State is OFF 0x02 – State is READY 0x03 – State is IN USE 0x04 – State is PAUSED/FINISHED
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.3.8 GFIT_SetInUseAdvertisingTimeout (Message ID = 0xD9)

Configuration Command: Sets the advertising timeout for the peripheral once the module transitions to the IN USE state. After the timeout, the peripheral will stop advertising. Setting a timeout allows the G.FIT to continue advertising so that the user can pair their phone while in state IN USE for the specified time.

Valid State: OFF

Default: 30 second in use advertising timeout.

Table 7-17. GFIT_HRSetInUseScanTimeout Command Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x02
2	Message ID	1 byte	Set to 0xE2 (command)
3	Sub-Message ID	1 byte	Set to 0xD9 (GFIT_SetInUseAdvertisingTimeout)
4	Payload	1 byte	Timeout Value: Set to 0 for no timeout (Peripheral stops advertising when transitioning to IN USE state). Set to a value from 1-254 to set a timeout in units of 2.5 seconds Set to 255 for infinite timeout.
5	Checksum	1 byte	XOR of all previous bytes including the sync byte.

Table 7-18. GFIT_SetInUseAdvertisingTimeout Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x05
2	Response ID	2 bytes	0xE0 (response)
3			0xFE (G.FIT response)
4	Command ID	2 bytes	0xE2 (command)
5			0xD9 (GFIT_SetInUseAdvertisingTimeout)
6	Response Code	1 byte	0x00 - GFIT_SUCCESS 0x01 - GFIT_ERROR_INCORRECT_STATE 0x12 - GFIT_ERROR_INVALID_LENGTH See Table 7-1 for descriptions.
7	Payload	1 byte	Current State: 0x00 – State is INVALID 0x01 – State is OFF 0x02 – State is READY 0x03 – State is IN USE 0x04 – State is PAUSED/FINISHED
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.3.9 GFIT_SetEquipmentType (Message ID = 0xE0)

Configuration Command: Sets the FE equipment type. This updates which FE Specific Main Data Pages are transmitted on the ANT+ FE-C channel and which characteristics are enabled in the BLE FTMS.

Valid State: OFF

Table 7-19. GFIT_SetEquipmentType Command Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x02
2	Message ID	1 byte	Set to 0xE2 (command)
3	Sub-Message ID	1 byte	Set to 0xE0 (GFIT_SetEquipmentType)
4	Payload	1 byte	Set to represent the FE Type: 19 – Treadmill 20 – Elliptical 22 – Rower 23 – Climber 25 – Trainer / Indoor Bike
5	Checksum	1 byte	XOR of all previous bytes including the sync byte.

Table 7-20. GFIT_SetEquipmentType Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x05
2	Response ID	2 bytes	0xE0 (response)
3			0xFE (G.FIT response)
4	Command ID	2 bytes	0xE2 (command)
5			0xE0 (GFIT_SetEquipmentType)
6	Response Code	1 byte	0x00 - GFIT_SUCCESS 0x01 - GFIT_ERROR_INCORRECT_STATE 0x0D - GFIT_ERROR_INVALID_EQUIPMENT_TYPE 0x12 - GFIT_ERROR_INVALID_LENGTH 0x15 - GFIT_ERROR_EQUIPMENT_TYPE_ALREADY_SET See Table 7-1 for descriptions.
7	Payload	1 byte	Current State: 0x00 – State is INVALID 0x01 – State is OFF 0x02 – State is READY 0x03 – State is IN USE 0x04 – State is PAUSED/FINISHED
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.3.10 GFIT_SetFEMetrics(Message ID = 0xE1)

When comparing the ANT+ FE-C profile and the BLE FTMS, there is quite a bit of variance in the how data is represented and transmitted over the air. More specifically, there are discrepancies between the units, ranges and size of fields. The G.FIT interface for setting these fitness equipment metrics was designed to handle that complexity, as a result the application is only required to enter a single G.FIT defined value that is compatible with both profiles. G.FIT then formats and sends the data accordingly for each profile, reducing implementation efforts on the application layer.

Within this document **FE Metrics** refer to the data elements that are sent from the given type of fitness equipment such as speed, distance, power, etc.

Field IDs are used as a key to specify which FE metric the app is attempting to update on the G.FIT module. See section 7.3.10.2 for a list of field IDs that are applicable for a given fitness equipment type.

A **field** contains the value of the given FE metric based on the preceding field ID.

The GFIT_SetFEMetrics serial command has two purposes. When in the **initial** OFF state, GFIT_SetFEMetrics is used to configure the G.FIT, which specifies what FE Metrics are supported by the fitness equipment. Once the G.FIT moves to ready for the **first time**, it is no longer in configuration mode and the GFIT_SetFEMetrics serial command is used to set the values of the FE Metrics, including any subsequent OFF states.

Configuration Command (OFF): GFIT_setFEMetrics is used to configure which FE metrics the G.FIT will support prior to initial use. This serial command **must** be called once before the **first** transition from OFF to READY. This command is used to configure the G.FIT and specify which FE Metrics the fitness equipment supports. Configuring a FE Metric before the initial OFF to READY transition ensures that the FE Metric shows up as supported when transmitted over ANT+ FE-C and that it is included in the BLE FTMS. Any FE metrics not configured during first OFF (configuration) state, will be not be permitted to be updated in later states.

Set Data Command (OFF, READY, IN USE, FINISHED): Update the current fitness equipment metrics that the G.FIT module is reporting on the ANT+ FE-C channel and BLE FTMS.

Valid State: OFF, READY, IN USE, FINISHED

The set FE Metrics serial message is used to update the values that are transmitted by the G.FIT module over ANT+ FE-C and BLE FTMS. Any number of FE Metrics can be set in a single serial command. The serial message must be constructed as a series of pairs of a field ID followed by a field, so that the G.FIT knows which FE metric the value (field) is associated to. See Figure 7-7 for an example of how this is completed.

Table 7-21. GFIT_SetFEMetrics Command Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Length is variable. See Equation 7-1 for calculating length.
2	Message ID	1 byte	Set to 0xE2 (command)
3	Sub-Message ID	1 byte	Set to 0xE1 (GFIT_SetFEMetrics)
4 – N	Payload	N bytes	The payload contains a set of key value pairs that determine which FE metrics will be set on the G.FIT. The payload must start with a field ID followed by the dynamically sized field. If more than one FE metrics needs to be set you can continue placing field IDs followed by fields. See Figure 7-7 for an example of how to construct the payload.
N + 1	Checksum	1 byte	XOR of all previous bytes including the sync byte.

Table 7-22. GFIT_SetFEMetrics Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x05
2	Response ID	2 bytes	0xE0 (response)
3			0xFE (G.FIT response)
4	Command ID	2 bytes	0xE2 (command)
5			0xE1 (GFIT_SetFEMetrics)
6	Response Code	1 byte	See Table 7-1 for descriptions. 0x00 - GFIT_SUCCESS 0x0B - GFIT_ERROR_INVALID_FIELD_ID_FOR_EQUIPMENT_TYPE 0x0C - GFIT_ERROR_INCORRECT_FE_METRIC_PAYLOAD 0x0E - GFIT_ERROR_EQUIPMENT_TYPE_NOT_SET 0x0F - GFIT_ERROR_NO_FIELD_IDS_CONFIGURED 0x10 - GFIT_ERROR_FIELD_ID_NOT_CONFIGURED 0x12 - GFIT_ERROR_INVALID_LENGTH 0x13 - GFIT_ERROR_INVALID_BIKE_POWER_USAGE
7	Payload	1 byte	Current State: 0x00 – State is INVALID 0x01 – State is OFF 0x02 – State is READY 0x03 – State is IN USE 0x04 – State is PAUSED/FINISHED
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.3.10.1 Details on constructing the SetFEMetrics serial message

When the GFIT_SetFEMetrics payload (a series of pairs of a **field ID** followed by a **field**) is constructed, the fields must be represented using the appropriate length and encoded in **little endian**. See Table 7-23 for the length of a field for a given field ID.

Table 7-23. Field Lengths

Field ID used	Field length required
1 to 31	1
32 to 95	2
96 to 127	3

After the payload has been constructed, the length of the SetFEMetrics serial message can be calculated. The length of the serial message must take into account the Sub-Message ID, field IDs used, and the length of all fields combined. See Equation 7-1 for how to calculate the SetFEMetrics serial message length.

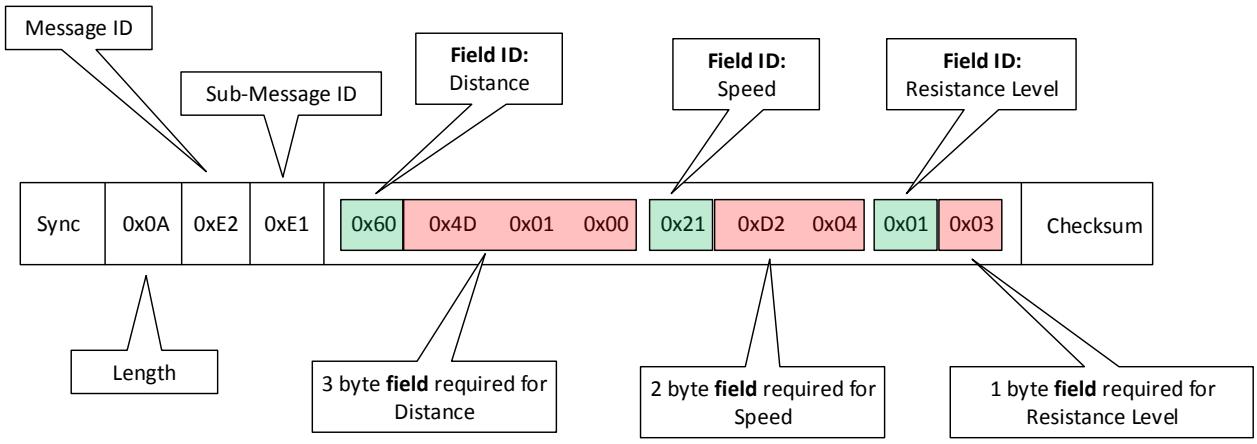
$$Total\ Field\ Length = (Number\ of\ 1\text{-}byte\ fields) \times 1 + (Number\ of\ 2\text{-}byte\ fields) \times 2 + (Number\ of\ 3\text{-}byte\ fields) \times 3$$

$$FE\_SetFEMetrics\ Length = 1 + Number\ of\ Updated\ FE\ Metrics + Total\ Field\ Length$$

Equation 7-1. Calculating GFIT_SetFEMetrics Length

Figure 7-7 below shows an example of how to construct a GFIT_SetFEMetrics serial message. In this example 3 FE metrics are set in a single serial message. The GFIT_SetFEMetrics serial message can be used to set as little as 1 FE metric or setting all FE Metrics for the given fitness equipment type.

Set FE Metrics Example:
Set Distance to 333
Set Speed to 1234
Set Resistance Level to 3



Item that contributes to length	Length
Sub-Message ID	1
Field ID	(3 Field IDs used) x 1 = 3
Total **Field** Length	(1 x 1) + (1 x 2) + (1 x 3) = 6

Total Length	1 + 3 + 6 = 10

Figure 7-7. Example: Setting FE Metrics Serial Command

7.3.10.2 Fitness Equipment Type Field IDs

Table 7-24. Treadmill Field IDs and Field Lengths

Treadmill	FE Metric	Input Units (G.FIT)	Output Units (ANT)	Output Units (BLE)	Field ID	Field Length (in bytes)	Set Value to Invalid
Treadmill Specific Data	Elapsed Time	Seconds	Quarter Seconds	Seconds	32 (0x20)	2	--
	Instantaneous Speed	Millimeters per Second	Millimeters per Second	Hundredth Kilometer per Hour	33 (0x21)	2	0xFFFF
	Instantaneous Cadence	Steps per Minute	Steps per Minute	--	35 (0x23)	2	0xFFFF
	Negative Vertical Distance	Tenth Meter	Tenth Meter	Tenth Meter	36 (0x24)	2	--
	Positive Vertical Distance	Tenth Meter	Tenth Meter	Tenth Meter	37 (0x25)	2	--
	Total Distance	Meters	Meters	Meters	96 (0x60)	3	--
Metabolic Data	Metabolic Equivalent	Hundredth MET	Hundredth MET	Tenth MET	44 (0x2C)	2	0xFFFF
	Caloric Burn Rate	Tenth kCal per Hour	Tenth kCal per Hour	kCal per Hour	45 (0x2D)	2	0xFFFF
	Calories Burned	kCal	kCal	kCal	46 (0x2E)	2	--
Settings	Cycle Length	Hundredth Meter	Hundredth Meter	--	2 (0x02)	1	0xFF
	Incline	Hundredth Percent	Hundredth Percent	Tenth Percent	47 (0x2F)	2 (signed)	0x7FFF
Product Information	HW Revision	Unitless	Unitless	--	3 (0x03)	1	--
	SW Number Major	Unitless	Unitless	--	4 (0x04)	1	--
	SW Number Minor	Unitless	Unitless	--	5 (0x05)	1	0xFF
	Manufacturer ID	Unitless	Unitless	--	48 (0x30)	2	--
	Model Number	Unitless	Unitless	--	49 (0x31)	2	--

Table 7-25. Trainer / Indoor Bike Field IDs and Field Lengths

Trainer / Indoor Bike	FE Metric	Input Units (G.FIT)	Output Units (ANT)	Output Units (BLE)	Field ID	Field Length (in bytes)	Set Value to Invalid
Bike Specific Data	Elapsed Time	Seconds	Quarter Seconds	Seconds	32 (0x20)	2	--
	Instantaneous Speed	Millimeters per Second	Millimeters per Second	Hundredth Kilometer per Hour	33 (0x21)	2	0xFFFF
	Instantaneous Cadence	Half Revolutions per Min	Revolutions per Min	Half Revolutions per Min	34 (0x22)	2	0xFFFF
	Instantaneous Power†	Watts	Watts (unsigned)	Watts	42 (0x2A)	2 (signed)	0x8000
	Accumulated Power†	Watts	Watts	--	43 (0x2B)	2	--
	Total Distance	Meters	Meters	Meters	96 (0x60)	3	--
Metabolic Data	Metabolic Equivalent	Hundredth MET	Hundredth MET	Tenth MET	44 (0x2C)	2	0xFFFF
	Caloric Burn Rate	Tenth kCal per Hour	Tenth kCal per Hour	kCal per Hour	45 (0x2D)	2	0xFFFF
	Calories Burned	kCal	kCal	kCal	46 (0x2E)	2	--
Settings	Resistance Level	Percent	Half Percent	Percent	1 (0x01)	1 (signed)	--
	Cycle Length	Hundredth Meter	Hundredth Meter	--	2 (0x02)	1	0xFF
Product Information	HW Revision	Unitless	Unitless	--	3 (0x03)	1	--
	SW Number Major	Unitless	Unitless	--	4 (0x04)	1	--
	SW Number Minor	Unitless	Unitless	--	5 (0x05)	1	0xFF
	Manufacturer ID	Unitless	Unitless	--	48 (0x30)	2	--
	Model Number	Unitless	Unitless	--	49 (0x31)	2	--

† To comply with the ANT+ FE-C profile, the G.FIT enforces that if either instantaneous power or accumulated power is updated, that both instantaneous power and accumulated power be included in the same GFIT_SetFEMetrics serial command. When updating either field ID 42 or 43, if both field IDs 42 and 43 are not updated in the same serial message, the G.FIT will return error: GFIT_ERROR_INVALID_BIKE_POWER_USAGE (0x13).

Table 7-26. Elliptical / Cross Trainer Field IDs and Field Lengths

Elliptical / Cross Trainer	FE Metric	Input Units (G.FIT)	Output Units (ANT)	Output Units (BLE)	Field ID	Field Length (in bytes)	Set Value to Invalid
Elliptical Specific Data	Elapsed Time	Seconds	Quarter Seconds	Seconds	32 (0x20)	2	--
	Instantaneous Speed	Millimeters per Second	Millimeters per Second	Hundredth Kilometer per Hour	33 (0x21)	2	0xFFFF
	Instantaneous Cadence	Steps per Minute	Stride† (Steps/2) per Min	Steps per Minute	35 (0x23)	2	0xFFFF
	Positive Vertical Distance	Tenth Meter	Tenth Meter	Meter	37 (0x25)	2	--
	Stride† Count	Tenth Stride†	Stride†	Tenth Stride†	39 (0x27)	2	--
	Instantaneous Power	Watts	Watts (unsigned)	Watts	42 (0x2A)	2 (signed)	0x8000
	Total Distance	Meters	Meters	Meters	96 (0x60)	3	--
Metabolic Data	Metabolic Equivalent	Hundredth MET	Hundredth MET	Tenth MET	44 (0x2C)	2	0xFFFF
	Caloric Burn Rate	Tenth kCal per Hour	Tenth kCal per Hour	kCal per Hour	45 (0x2D)	2	0xFFFF
	Calories Burned	kCal	kCal	kCal	46 (0x2E)	2	--
Settings	Resistance Level	Percent	Half Percent	Tenth Percent	1 (0x01)	1 (signed)	--
	Cycle Length	Hundredth Meter	Hundredth Meter	--	2 (0x02)	1	0xFF
	Incline	Hundredth Percent	Hundredth Percent	Tenth Percent	47 (0x2F)	2 (signed)	0x7FFF
Product Information	HW Revision	Unitless	Unitless	--	3 (0x03)	1	--
	SW Number Major	Unitless	Unitless	--	4 (0x04)	1	--
	SW Number Minor	Unitless	Unitless	--	5 (0x05)	1	0xFF
	Manufacturer ID	Unitless	Unitless	--	48 (0x30)	2	--
	Model Number	Unitless	Unitless	--	49 (0x31)	2	--

† A stride is defined as a full revolution of the elliptical rear pedal disk/flywheel, which is the equivalent of two steps.

Table 7-27. Rower Field IDs and Field Lengths

Rower	FE Metric	Input Units (G.FIT)	Output Units (ANT)	Output Units (BLE)	Field ID	Field Length (in bytes)	Set Value to Invalid
Rower Specific Data	Elapsed Time	Seconds	Quarter Seconds	Seconds	32 (0x20)	2	--
	Instantaneous Speed	Millimeters per Second	Millimeters per Second	--	33 (0x21)	2	0xFFFF
	Instantaneous Cadence	Half Strokes per Min	Strokes per Min	Half Strokes per Min	34 (0x22)	2	0xFFFF
	Stroke Count	Strokes	Strokes	Strokes	41 (0x29)	2	--
	Instantaneous Power	Watts	Watts (unsigned)	Watts	42 (0x2A)	2 (signed)	0x8000
	Total Distance	Meters	Meters	Meters	96 (0x60)	3	--
Metabolic Data	Metabolic Equivalent	Hundredth MET	Hundredth MET	Tenth MET	44 (0x2C)	2	0xFFFF
	Caloric Burn Rate	Tenth kCal per Hour	Tenth kCal per Hour	kCal per Hour	45 (0x2D)	2	0xFFFF
	Calories Burned	kCal	kCal	kCal	46 (0x2E)	2	--
Settings	Resistance Level	Percent	Half Percent	Percent	1 (0x01)	1 (signed)	--
	Cycle Length	Hundredth Meter	Hundredth Meter	--	2 (0x02)	1	0xFF
Product Information	HW Revision	Unitless	Unitless	--	3 (0x03)	1	--
	SW Number Major	Unitless	Unitless	--	4 (0x04)	1	--
	SW Number Minor	Unitless	Unitless	--	5 (0x05)	1	0xFF
	Manufacturer ID	Unitless	Unitless	--	48 (0x30)	2	--
	Model Number	Unitless	Unitless	--	49 (0x31)	2	--

Table 7-28. Step Climber Field IDs and Field Lengths

Step Climber	FE Metric	Input Units (G.FIT)	Output Units (ANT)	Output Units (BLE)	Field ID	Field Length (in bytes)	Set Value to Invalid
Step Climber Specific Data	Elapsed Time	Seconds	Quarter Seconds	Seconds	32 (0x20)	2	--
	Instantaneous Speed	Millimeters per Second	Millimeters per Second	Hundredth Kilometer per Hour	33 (0x21)	2	0xFFFF
	Instantaneous Cadence	Steps per Minute	Cycles† (Steps/2) per Minute	Steps per Minute	35 (0x23)	2	0xFFFF
	Positive Vertical Distance (Climber)	Meter	Meter‡	Meter	38 (0x26)	2	--
	Step Count	Steps	Cycle† (Steps/2)	Steps	40 (0x28)	2	--
	Instantaneous Power	Watts	Watts (unsigned)	--	42 (0x2A)	2 (signed)	0x8000
	Floors Climbed	Floors	--	Floors	50 (0x32)	2	--
Metabolic Data	Metabolic Equivalent	Hundredth MET	Hundredth MET	Tenth MET	44 (0x2C)	2	0xFFFF
	Caloric Burn Rate	Tenth kCal per Hour	Tenth kCal per Hour	kCal per Hour	45 (0x2D)	2	0xFFFF
	Calories Burned	kCal	kCal	kCal	46 (0x2E)	2	--
Settings	Cycle Length	Hundredth Meter	Hundredth Meter	--	2 (0x02)	1	0xFF
Product Information	HW Revision	Unitless	Unitless	--	3 (0x03)	1	--
	SW Number Major	Unitless	Unitless	--	4 (0x04)	1	--
	SW Number Minor	Unitless	Unitless	--	5 (0x05)	1	0xFF
	Manufacturer ID	Unitless	Unitless	--	48 (0x30)	2	--
	Model Number	Unitless	Unitless	--	49 (0x31)	2	--

† A cycle is defined as a one step taken with each foot, resulting in 1 cycle = 2 steps climbed.

‡ ANT+ FE-C climber specific page does not have vertical distance, and will transmit inputted vertical distance as distance in data page 16.

7.3.11

7.3.11 GFIT_SetHeartRate (Message ID = 0xE2)

Set Data Command: Update the current heart rate that the module is reporting on the ANT+ FE-C channel. This command will have no effect if there is an external ANT+ or BLE heart rate monitor connected to the G.FIT.

Valid State: OFF, READY, IN USE, FINISHED

Table 7-29. GFIT_SetHeartRate Command Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x02
2	Message ID	1 byte	Set to 0xE2 (command)
3	Sub-Message ID	1 byte	Set to 0xE2 (GFIT_SetHeartRate)
4	Payload	1 byte	Set to represent the heart rate in bpm. 0x00 indicates invalid
5	Checksum	1 byte	XOR of all previous bytes including the sync byte.

Table 7-30. GFIT_SetHeartRate Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x05
2	Response ID	2 bytes	0xE0 (response)
3			0xFE (G.FIT response)
4	Command ID	2 bytes	0xE2 (command)
5			0xE2 (GFIT_SetHeartRate)
6	Response Code	1 byte	0x00 - GFIT_SUCCESS 0x01 - GFIT_ERROR_INCORRECT_STATE 0x12 - GFIT_ERROR_INVALID_LENGTH 0x14 - GFIT_ERROR_HR_SENSOR_PAIRED See Table 7-1 for descriptions.
7	Payload	1 byte	Current State: 0x00 – State is INVALID 0x01 – State is OFF 0x02 – State is READY 0x03 – State is IN USE 0x04 – State is PAUSED/FINISHED
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.3.12 GFIT_SetDeviceID (Message ID = E3)

Configuration Command: Updates the G.FIT ANT+ FE-C device ID. This updates the 4 byte serial number transmitted in common page 81, and changes the ANT+ FE-C device number channel parameter.

The G.FIT takes the 4 byte device ID and casts it to a 2 byte device number which is used as the device number channel parameter when opening the ANT+ FE-C channel.

The device number of the fitness equipment shall not be 0x0000. Care should be taken to ensure that serial numbers that are multiples of 0x10000 (65536) are handled correctly so that the lower 16-bits of the device ID will not be equal to 0.

Valid State: OFF

Table 7-31. GFIT_SetDeviceID Command Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x05
2	Message ID	1 byte	Set to 0xE2 (command)
3	Sub-Message ID	1 byte	Set to 0xE3 (GFIT_SetDeviceID)
4	Payload	1 byte	Device number LSB
5		1 byte	…
6		1 byte	…
7		1 byte	Device number MSB
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

Table 7-32. GFIT_SetDeviceID Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x05
2	Response ID	2 bytes	0xE0 (response)
3			0xFE (G.FIT response)
4	Command ID	2 bytes	0xE2 (command)
5			0xE3 (GFIT_SetDeviceID)
6	Response Code	1 byte	0x00 - GFIT_SUCCESS 0x01 - GFIT_ERROR_INCORRECT_STATE 0x04 - GFIT_ERROR_INVALID_DEVICE_ID 0x12 - GFIT_ERROR_INVALID_LENGTH See Table 7-1 for descriptions.
7	Payload	1 byte	Current State: 0x00 – State is INVALID 0x01 – State is OFF 0x02 – State is READY 0x03 – State is IN USE

			0x04 – State is PAUSED/FINISHED
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.3.13 GFIT_SendCustomEvent (Message ID = 0xB0)

Custom Command: Instruct the module to send an event to the leaderboard application over the ANT+ FE-C channel. Note: This is a custom command and is not part of the ANT+ FE-C profile.

Valid State: READY, IN USE, FINISHED

Table 7-33. GFIT_SendCustomEvent Command Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x02
2	Message ID	1 byte	Set to 0xE2 (command)
3	Sub-Message ID	1 byte	Set to 0xB0 (send custom event)
4	Payload	1 byte	Application defined event
5	Checksum	1 byte	XOR of all previous bytes including the sync byte.

Table 7-34. GFIT_SendCustomEvent Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x05
2	Response ID	2 bytes	0xE0 (response)
3			0xFE (G.FIT response)
4	Command ID	2 bytes	0xE2 (command)
5			0xB0 (GFIT_SendCustomEvent)
6	Response Code	1 byte	0x00 - GFIT_SUCCESS 0x01 - GFIT_ERROR_INCORRECT_STATE 0x12 - GFIT_ERROR_INVALID_LENGTH See Table 7-1 for descriptions.
7	Payload	1 byte	Current State: 0x00 – State is INVALID 0x01 – State is OFF 0x02 – State is READY 0x03 – State is IN USE 0x04 – State is PAUSED/FINISHED
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.3.14 GFIT_SendManufacturerSpecificPage (Message ID = B3)

Command: Instruct the module to send a manufacturer specific over the ANT+ FE-C channel. The manufacturer specific page will replace the page to be transmitted; it is not recommended to insert manufacturer specific pages in a short rotation interval as that may affect the transmission pattern for ANT+ FE-C, see section xxx for more details. This command should only be called once per channel period of the ANT+ FE-C channel; the application can use the EVENT_TX as cue to send the manufacturer specific page, or send the command based on asynchronous events (e.g. button). Valid manufacturer specific pages numbers are 0xE0 to 0xFF.

Valid State: READY, IN USE, FINISHED

Table 7-35. GFIT_SendManufacturerSpecificPage Command Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x09
2	Message ID	1 byte	Set to 0xE2 (command)
3	Sub-Message ID	1 byte	Set to 0xB3 (GFIT_SendManufacturerSpecificPage)
4	Payload	1 byte	Manufacturer specific page number
5		1 byte	Manufacturer specific data
6		1 byte	Manufacturer specific data
7		1 byte	Manufacturer specific data
8		1 byte	Manufacturer specific data
9		1 byte	Manufacturer specific data
10		1 byte	Manufacturer specific data
11		1 byte	Manufacturer specific data
12	Checksum	1 byte	XOR of all previous bytes including the sync byte.

Table 7-36. GFIT_SendManufacturerSpecificPage Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x05
2	Response ID	2 bytes	0xE0 (response)
3			0xFE (G.FIT response)
4	Command ID	2 bytes	0xE2 (command)
5			0xB3 (GFIT_SendManufacturerSpecificPage)
6	Response Code	1 byte	0x00 - GFIT_SUCCESS 0x01 - GFIT_ERROR_INCORRECT_STATE 0x11 - GFIT_ERROR_INVALID_MFG_SPECIFIC_PAGE_NUMBER 0x12 - GFIT_ERROR_INVALID_LENGTH See Table 7-1 for descriptions.
7	Payload	1 byte	Current State: 0x00 – State is INVALID

			0x01 – State is OFF 0x02 – State is READY 0x03 – State is IN USE 0x04 – State is PAUSED/FINISHED
8	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.4 Requests (0xE1)

7.4.1 G.FIT Version Number

To request the version number from the G.FIT, use the request version number message.

Table 7-37. Request G.FIT Version Number Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x03
2	Request ID	2 bytes	Set to 0xE1 (request)
3			Set to 0xFE
4	Command ID	2 bytes	Set to 0xE2 (command)
5			Set to 0xC0 (G.FIT Version)
6	Checksum	1 byte	XOR of all previous bytes including the sync byte.

Table 7-38. Version Number Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x0B
2	Message ID	1 byte	0xE2 (command)
3	Sub-Message ID	1 byte	0xC0 (G.FIT Version)
4	Payload	3 bytes	Software version prefix
5			
6			
7		1 byte	Major version number
8		1 byte	Software version separator
9		2 bytes	Minor version number
10			
11		3 bytes	Software version postfix
12			
13			
14	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.5 Events (0xE0)

7.5.1 Custom FE Channel Leaderboard Event

Event: A custom event that is sent to the MCU when the leaderboard sends a notification to the G.FIT module that requires user's attention.

Table 7-39. Custom FE Channel Leaderboard Event Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x02
2	Message ID	1 byte	0xE0 (event)
3	Sub-Message ID	1 byte	0xB1 (receive custom event)
4	Payload	1 byte	Application defined custom event.
5	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.5.2 HR Event

Event: An event that is sent to the MCU when the G.FIT module receives HR data from the wireless heart rate monitor it is currently connected to, as well as scan packets for heart rate monitors in the vicinity.

Table 7-40. HR Event Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Varies
2	Message ID	1 byte	0xE0 (event)
3	Sub-Message ID	1 byte	0xB2 (HR event)
4	Flags	1 byte	Bit 0 – 0: Scan, 1: Pair Bit 1 – 0: ANT protocol, 1: BLE protocol Bit 2 – 0: Friendly name not available, 1: Friendly name available Bits 3-7 – Reserved for future use (set to 0)
5	Heart Rate	1 byte	Heart rate in bpm. 0x00 indicates invalid
6	RSSI	1 byte	RSSI in dBm 0x7F indicates invalid
7	HRM ID	6 bytes	HRM ID LSB
8 – 11			…
12			HRM ID MSB
13	Reserved	2 bytes	Reserved for future use (set to 0). Do not interpret.
14			
15	Reserved	1 byte	Reserved for future use (set to 0xFF). Do not interpret.
16-20	Reserved	5 bytes	Reserved for future use (set to 0). Do not interpret.
21	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.5.2.1 Handling HR Events

7.5.2.1.1 Proximity Pairing Mode

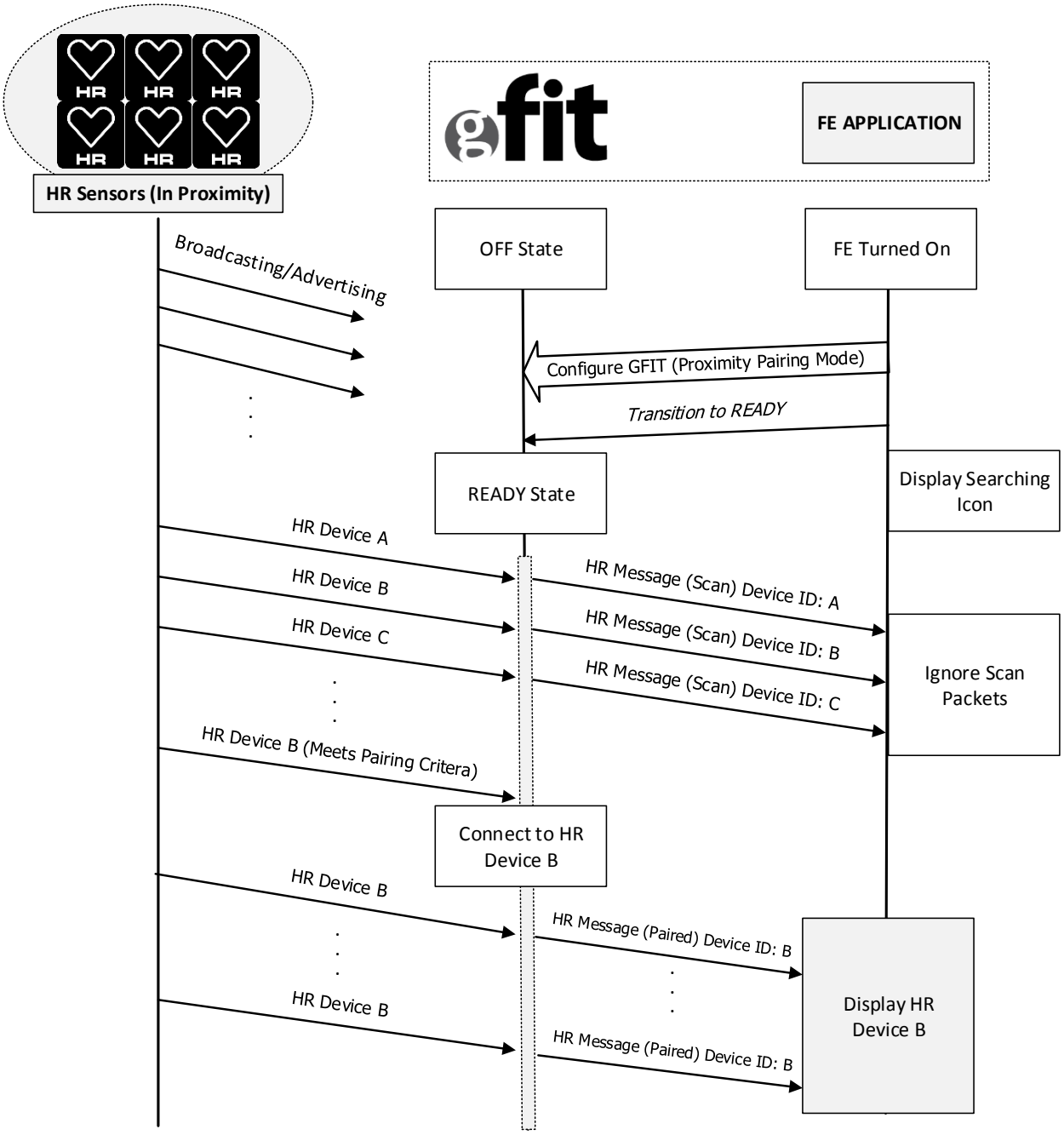


Figure 7-8. Sequence Diagram: Proximity Pairing Mode HR Events

7.5.2.1.2 List Pairing Mode

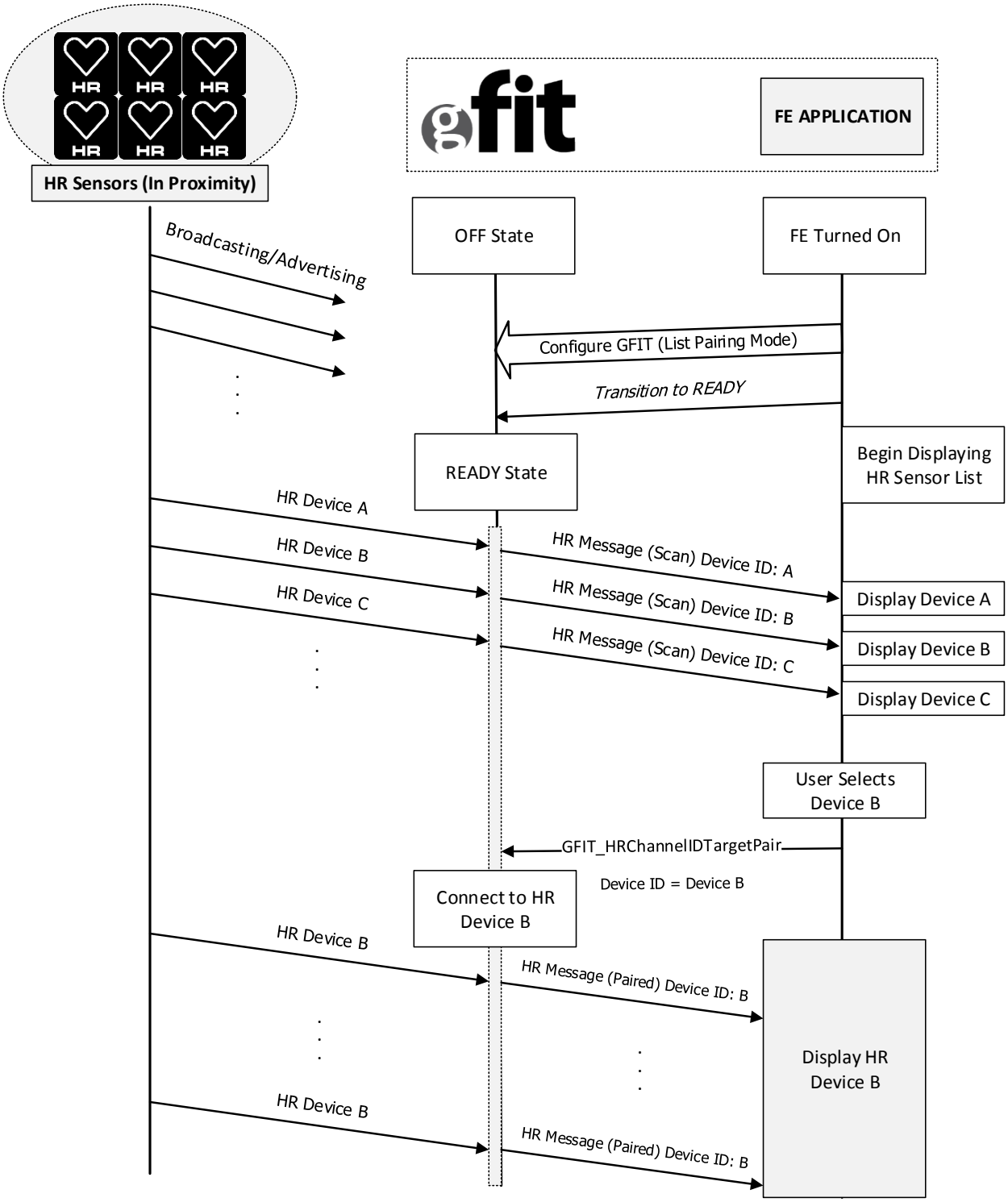


Figure 7-9. Sequence Diagram: Channel ID Based Pairing Mode Connection

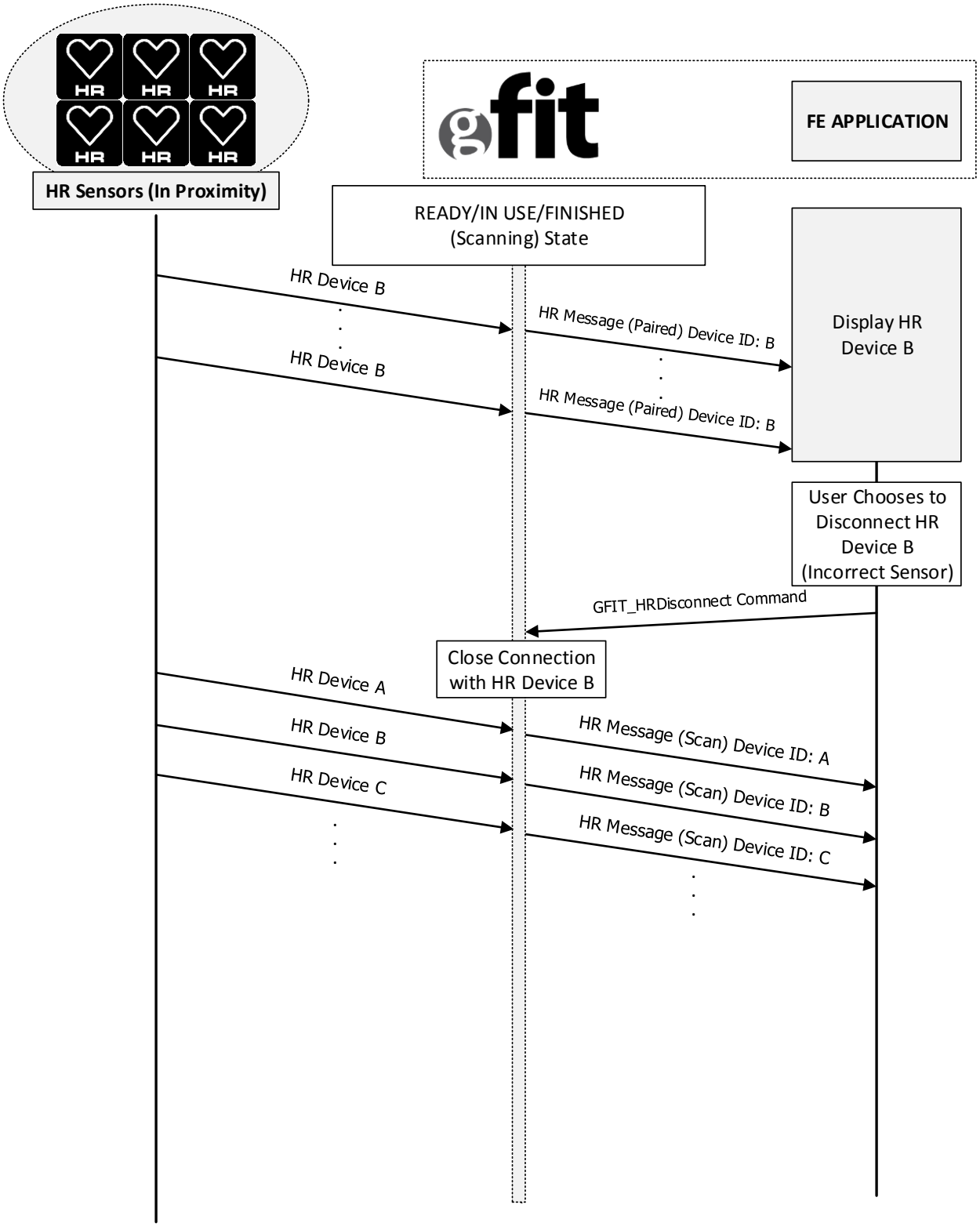


Figure 7-10. Sequence Diagram: Disconnecting from Connected Sensor Device B

7.5.2.1.3 Handling Disconnected Sensors

It is recommended that the application maintains an internal timeout (recommended ~15 seconds) once paired to an HR sensor. The timeout should be restarted every time a paired HR event is received from the G.FIT module. If a paired sensor disconnects from the G.FIT module (battery outage, user moves out of proximity), this timeout allows the application to prompt the user that the sensor has been removed, and action is required. If the fitness equipment is still in a scanning state (READY, IN USE/FINISHED before scanning timeout), it should go back to the search/pairing screen to indicate that the sensor is no longer paired, and a new sensor may be paired.

7.6 Other ANT Commands

7.6.1 Request Serial Number

Used to retrieve the serial number of the G.FIT module. The serial number is a 4-byte, little-endian encoded unsigned integer.

Table 7-41. Request Serial Number Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x01
2	Message ID	1 byte	Set to 0x4D (Request Message)
3	Payload	1 byte	Set to 0x61 (Serial Number)
4	Checksum	1 byte	XOR of all previous bytes including the sync byte.

Table 7-42. Serial Number Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x04
2	Message ID	1 byte	0x61 (MESG_GET_SERIAL_NUMBER_ID)
3	Payload	4 bytes	4 byte serial number LSB
4			…
5			…
6			4 byte serial number MSB
7	Checksum	1 byte	XOR of all previous bytes including the sync byte.

7.6.2 RSSI Calibration ID

Used to obtain the 1 byte calibration offset from the G.FIT module.

Table 7-43. RSSI Calibration ID Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	Set to 0x01
2	Message ID	1 byte	Set to 0xE4
3	Sub-Message ID	1 byte	Set to 0x02
4	Checksum	1 byte	XOR of all previous bytes including the sync byte.

Table 7-44. RSSI Calibration ID Response Message

Byte #	Name	Length	Description
0	Sync	1 byte	Fixed value of 10100100 or 10100101 (Refer to the "Interfacing with ANT General Purpose Chipsets and Modules" document for details.)
1	Length	1 byte	0x02
2	Message ID	1 byte	0xE4
3	Sub-Message ID	1 byte	0x02
4	Payload	1 byte	Calibration Offset
5	Checksum	1 byte	XOR of all previous bytes including the sync byte.

8 Appendix

8.1 Compiling the Demo Appication

Unzip the contents of the G.FIT_Library zip file inside the G.FIT Libraries Package into a directory on your hard drive (e.g. C:\gfit).

Download version 13.1 the Nordic nRF5 SDK for GCC from http://developer.nordicsemi.com/. Create a new directory called nrf5-sdk inside the gfit folder, and install the the Nordic nRF5 SDK in this directory.

Unzip the S332 SoftDevice package and copy the contents of the include directory directly into the components\softdevice\s332\headers directory of the nrf5-sdk.

Unzip the G.FIT SDK package, and copy the app directory into the gfit directory.

The final directory structure should look like Figure 8-1.

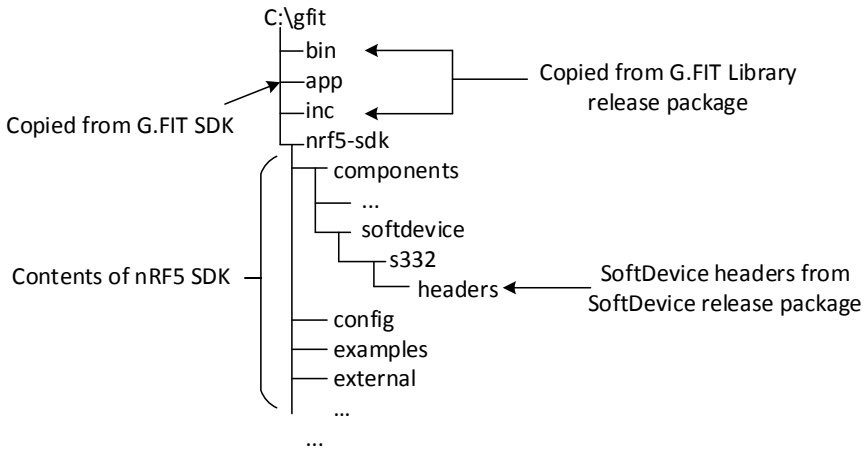


Figure 8-1. Final Directory Structure

Build the demo_gfit project. The first time you build the project, the compiler will point out two errors which serve as reminders that licenses must be obtained to use the S332 SoftDevice and the G.FIT libraries. Follow the instructions in the error messages to enable the evaluation keys and allow the project to build.

8.1.1 Library License Key

The G.FIT Library require a license key to operate. An evaluation key is available which will enable full functionality and is to be used for NON-COMMERCIAL USE ONLY. The license key required for the G.FIT Library is different and independent from the license key G.FIT operates exclusively on Dynastream G.FITmodules, once the Distribution Agreement for the G.FIT Module is completed, SoftDevice royalties will be waived for each instance of ANT SoftDevice used within a G.FIT module for commercial end product.

Further information about licensing can be found at: www.thisisant.com/developer/ant/licensing.

License validation can extend the library initialization time to up to 100 ms.

8.1.2 Compiler Compatibility

The G.FIT static library is built with GNU ARM Embedded Toolchain version 4.9 2015q3, and uses the settings in Table 8-1. See example application Makefile for further details.

Table 8-1. GCC Specific Compiler Options

Option	Description
-mcpu=cortex-m4	Cortex-M4
-mfloat-abi=hard -mfpu=fpv4-sp-d16	The libraries are built using hardware floating point numbers. Application settings must match.
-mthumb	Thumb mode