G.FIT and Premium Module Manufacturing Considerations

Application Note

D00001706 **Rev 1.0**

T +1 403.932.9292 **F** +1 403.932.6521 **www.dynastream.com** **www.thisisant.com**
#201, 100 Grande Blvd., Cochrane, Alberta, Canada T4C 0S4

Copyright Information and Usage Notice

Revision History

Revision	Effective Date	Description
1.0	August 2017	Initial Release

Table of Contents

List of Figures

List of Tables

List of Equations

1 Introduction

During production, the G.FIT (D52xGF) and Premium (D52xPM and D52xSK) modules are programmed with additional data. This data contains information that is used by some of the features in the G.FIT library and factory-loaded software. If the modules are to be re-programmed with custom firmware using the SWD interface, some of these data fields may need to be retained manually. This document describes the functions of the data fields, as well as the back-up and restoration process.

The G.FIT modules contain a firmware updater which will preserve all fields. If G.FIT modules are to be updated using the updater, the backup and restoration processes are not necessary. However, if G.FIT modules are re-programmed using the SWD interface, the firmware updater and its settings will not be preserved.

2 Data Fields

2.1 RSSI Calibration Offset

RSSI readings can be used to gauge the distance between two wireless devices. To ensure that each module reports the same RSSI reading from a given transmitter, calibration is performed when the modules are produced. This calibration is stored in the form of a single byte within the UICR Customer Register space. The calibration byte is an offset to be applied to a raw reading using the following formula:

$$AdjustedRSSI = RawRSSI - (127 - RSSICalibrationOffset)$$

Equation 2-1. Applying the RSSI Calibration Offset

Care should be taken to ensure that the RSSI calibration has been assigned. If the data field corresponding to the RSSI calibration value reads 0xFF, no calibration value exists and calibration should not be applied.

The RSSI calibration offset is stored in the LSByte of UICR Customer Register 3, located at address 0x1000108C in the nRF52 memory. The value of this location cannot be read through the SWD interface once the readback protection is enabled. The location is erased when the "--recover" operation is carried out in nrfjprog.

2.1.1 Read RSSI Calibration Offset

The RSSI Calibration Offset can be retrieved from the module using a serial command if the stock firmware is present. To do so, a request must be sent (for full description of the serial protocol, refer to D00000652 – ANT Message Protocol and Usage and D00001699 – G.FIT User Guide and Interface Control):

Message Length	Message ID	Payload [0]	Payload [1]	Payload [2]
3	0xE1	0x00	0xE4	0x02

Figure 2-1. Request to Retrieve RSSI Calibration Offset

The response will arrive with Message ID 0xE4 and can be parsed as:

Message Length	Message ID	Payload [0]	Payload [1]
2	0xE4	0x02	RSSI Calibration Offset

Figure 2-2. RSSI Calibration Offset Read Response Message

2.1.2 Write RSSI Calibration Offset

The RSSI Calibration Offset can be written to the module using a serial command if the network processor interface is present. Note that attempting to overwrite an existing non-0xFF value will not be successful.

Message Length	Message ID	Payload [0]	Payload [1]
2	0xE4	0x02	RSSI Calibration Offset

Figure 2-3. Write RSSI Calibration Offset

2.2 ANT ID

Unique 32-bit ANT ID values have been written into the UICR Customer Register space for each module. The values are only guaranteed to be unique to 20 bits within one module family – for example, the lower 20 bits may be identical between G.FIT and Premium modules; however, M8 and M4 modules of the G.FIT family will never have identical lower 20 bits in their ANT IDs.

The ANT ID value is stored in the UICR Customer Register 4, located at address 0x10001090 in the nRF52 memory. The value of this location cannot be read through the SWD interface once the readback protection is enabled. The location is erased when the "--recover" operation is executed in nrfjprog.

The ANT ID value is used internally by the G.FIT library. Furthermore, the value can be used by a customer in order to start an ANT channel where the Device ID and Transmission Type are derived from the ANT ID. Subsection 2.2.1 describes this process.

2.2.1 Read ANT ID

The ANT ID can be retrieved from the module using a serial command if the stock firmware is present. To do so, a Request Message (0x4D) must be sent:

Message Length	Message ID	Channel	Payload [0]
2	0x4D	0x01	0x61

Figure 2-4. Request to Retrieve ANT ID

The response will arrive with Message ID 0x61 and can be parsed as:

Message Length	Message ID	Payload [0]	Payload [1]	Payload [2]	Payload [3]
4	0x61	ANT ID [0]	ANT ID [1]	ANT ID [2]	ANT ID [3]

Figure 2-5. ANT ID Read Response Message

2.2.2 Write ANT ID

The ANT ID can be written to the module using a serial command if the network processor interface is present. The ANT ID is a 32-bit value and is passed in LSByte first. A non-0xFFFFFFFF ANT ID cannot be overwritten. The message format is as follows:

Message Length	Message ID	Payload [0]	Payload [1]	Payload [2]	Payload [3]	Payload [4]	Payload [5]
6	0xE4	0x03	0x01	ANT ID [0]	ANT ID [1]	ANT ID [2]	ANT ID [3]

Figure 2-6. Write ANT ID

2.2.3 Setting up a Channel using ANT ID

It is possible to set up an ANT channel to use the ANT ID to derive the extended device number, which includes the Device ID and Transmission Type values. This can be done using the Serial Number Set Channel ID command (0x65). For more information on this command, please consult D00000652 – ANT Message Protocol and Usage.

Message Length	Message ID	Channel	Payload [0]	Payload [1]
3	0x65	Channel Number	Device Type	Trans Type

Figure 2-7. Serial Number Set Channel ID Command

For example, if ANT ID is 0x78563412, and the above command is called as 0x65-0x00-0x03-0x01, the device will be assigned:

- Device ID = 0x3412, derived from the lower 16 bits of ANT ID.
- Transmission Type = (0x6 << 4) | (0x01). 0x06 originates from bits 16-19 of the ANT ID, 0x01 is from the Serial Number Set Channel ID command.
- Device Type = 0x03, from the Serial Number Set Channel ID command.

Since ANT ID is used to set the transmission type, as well as the Device ID, it provides a 20-bit unique ID for the module.

3 Recommended Re-Programming Process

During the production process, all modules have readback protection enabled. This means that the values can no longer be read from the data fields described in Section 2 using the SWD interface. However, G.FIT and Premium modules contain a serial interface that can be used for this purpose. This section describes the process that can be used to read the desired data fields using existing serial commands.

3.1 Setup

3.1.1 Equipment

- J-Link programmer, connected to the SWD lines of the module
- UART-USB bridge. UART pins connected to the module as described in Section 3.1.2

3.1.2 Connections

The connections between required devices can be summarized as shown in Figure 3-1.

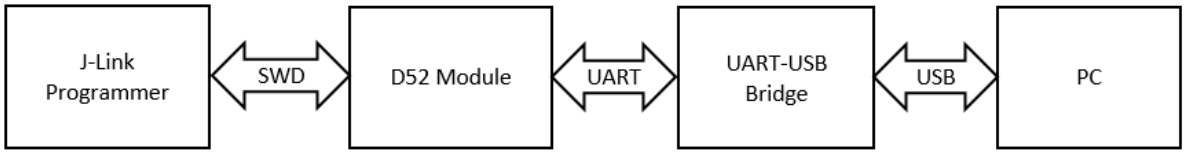

Figure 3-1. Overview of the Re-Programming Interconnect

The table below describes how the module's control pins must be configured in order to permit UART-based communication and allow the module to be re-programmed. BR1, BR2 and BR3 pins are tied to VCC in order to set the module to use 57600 baud, however, depending on the requirements of the setup, this can be changed. For more information on the baud rate settings as well as pin locations, please consult D00001705 – G.FIT Fitness Equipment Modules Datasheet, or D00001687 – D52 ANT SoC Module Series Datasheet.

Note the flow control requirements for the interface.

Module Pin	Voltage Level Connection	J-Link Programmer Connection	UART-USB Bridge Connection
VCC	VCC	--	--
GND	GND	GND	GND
TXD0	--	--	RX
RXD0	--	--	TX
RTS	--	--	CTS
BR1	VCC	--	--
BR2	VCC	--	--
BR3	VCC	--	--
SWDCLK	--	SWDCLK	--
SWDIO	--	SWDIO	--
nRESET	--	nRESET	--
nSUSPEND	VCC	--	--
SLEEP	GND	--	--
PORTSEL	GND	--	--

Table 3-1. Re-Programming Set up Connections

3.2 Process Overview

The re-programming process is described in the flowchart below. For information regarding the formatting of the serial commands, please consult the Reference Re-Programming Script (see Section 0).

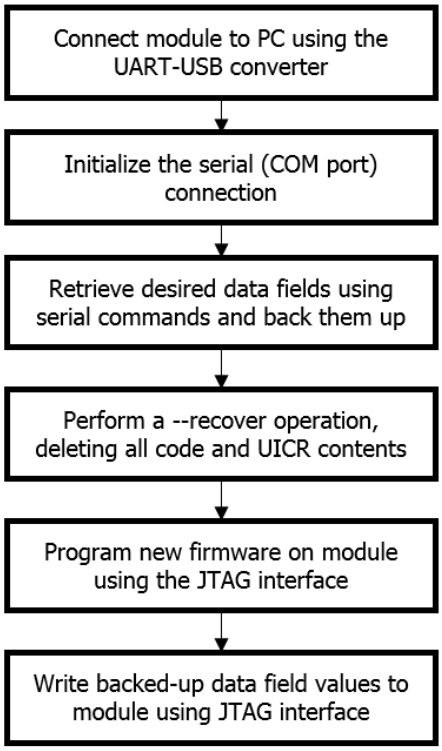

Figure 3-2. Overview of the Re-Programming Process

3.3 Reference Re-Programming Script

A reference re-programming script is provided with this document. The script relies on the UART-USB converter and the J-Link programmer being connected to a computer with the following software:

- Python (32-bit) with pyserial package
- nRF5x Command Line tools. These are available for download from the Nordic Semiconductor website.

Several sections of the script should be edited to match the setup configuration:

- J-Link Programmer serial number
- COM port used by the UART-USB converter
- Names and locations of the new firmware files (none are distributed with the script)

4 Enabling Readback Protection

All nRF52 devices can have readback protection enabled using the "--rbp ALL" command supplied through the nrfjprog command line interface. Doing so will disable the debug interface and not permit any memory to be read out of the device until a "--recover" command is issued, which will also erase **all** user code and data. This includes the UICR which contains the data fields described in Section 2.

Care must be taken to ensure that the desired data fields can still be recovered from the device after readback protection is enabled.