

((CABLE))



North Pole Engineering

Rick Gibbs

8/31/2016

Product Overview

CABLE (Convert ANT to BLE) is a low power radio product used to receive data from ANT+ enabled sensors and convert the data to the appropriate BLE services via a BLE peripheral connection. The supported ANT+ profiles are Heart Rate, Bike Speed, Bike Cadence, Bike Speed/Cadence, Bike Power. Any or all of the ANT+ device profiles are received by the CABLE and relayed to BLE endpoint via the appropriate BLE services. The supported Bike Power profiles include Power Only, Wheel Torque, Crank Torque, and Crank Torque Frequency.

Purpose and Scope of this Specification

This document is intended to detail the features of the IP provided by NPE in support of the CABLE product.

Product/Service Description

The primary low power radio technology used in the sports and fitness market for years has been ANT+. This is a radio technology pioneered by Dynastream in the mid 2000's. Adoption by the sports and fitness market started specifically in the biking and running communities with heart rate, speed/cadence, and stride/distance monitoring devices. Garmin and Timex were two large players in the marketplace to adopt the technology. Many watches and bike computers have used the radio technology to wirelessly transport the data, display it real time, and record it for later playback and analysis.

With the advent of the smart phone many consumers have switched from the sport specific displays to using their smart phone as the display of choice. With this shift in technology came a new low energy radio technology known as Bluetooth Low Energy or BLE. It is also known as Bluetooth Smart. The two radio technologies have similar power characteristics, but have very different usage models for connectivity. The iOS platform being a major player in the smart phone market has been driving the radio choice in the direction of BLE by selecting it as the only low energy radio technology it supports. As a result of this the demand for BLE enabled devices has surged leaving consumers with legacy ANT+ enabled products out of the loop.

Product Context

CABLE is IP capable of bridging the gap between the existing ANT+ enabled devices and the BLE only smart phones. By utilizing the Bluetooth SIG standardized services for each of the sensor types CABLE integrates with all existing applications currently

supporting BLE sensors. This means the product works off the shelf with hundreds of apps already in existence.

User Characteristics

The most compelling group of customers is current owners of ANT+ enabled power meters. These power meters typically cost from \$1000-2000. Because of the high cost to replace the power meters, consumers have been looking for cost effective ways to get their ANT+ sensor data, specifically power meters into their smart phones.

Additional consumers of the product are people with multiple bikes which utilize ANT+ sensors. To replace all of the sensors on all of their bikes would be expensive. CABLE IP allows the user to save configurations for multiple setups. This allows the user to transition from bike to bike with very little effort needed to change configurations.

The third category of user is the indoor cycling enthusiast. Many of the bikes on the market currently only support ANT+ transmissions. A user with a CABLE IP enabled device is able to pair with the ANT+ transmission and receive their workout data directly to their smart phone.

Assumptions

- The user of the product has at least one ANT+ sensor they would like to access via a BLE enabled device capable of interpreting the transmitted BLE services.
- The user wants to receive the data via their BLE enable smart phone.

Constraints

The design is constrained by a few factors including:

- The reception ranges of the ANT+ transmissions are dependent on the RF characteristic of the operational environment.
- BLE connectivity can be confusing when multiple BLE central devices are attempting to gain access to a BLE peripheral device.
- Up to 4 ANT+ devices can be converted to BLE services concurrently.
- Multiple sensors of the same type cannot be sent via the BLE peripheral channel.
- BLE connectivity range is dependent on the RF characteristic of the operational environment.

- Battery life of the unit is dependent on the number of ANT+ devices being tracked.

Dependencies

The CABLE IP implementation in the device is based on loading the IP into a Nordic nRF422 enable chip or module

Requirements

1. BLE Advertising **must have**
 - 1.1. Start (When the enable GPIO line is activated)
 - 1.2. Fast (20ms at startup for 30 seconds)
 - 1.3. Slow (200ms for 4 minutes)
 - 1.4. Stopped after 4.5 minutes with no connection
2. ANT+ Channel Scanning **must have**
 - 2.1. Wildcard scanning with RSSI threshold set to -XXdb
 - 2.2. Assigned scanning based on saved settings
 - 2.3. Each ANT+ channel has a set of characteristics
 - 2.3.1. Device Type
 - 2.3.2. Device Number
 - 2.4. ANT+ scanning times out after 30 seconds of searching.
 - 2.4.1. Scanning can be restarted by pulsing the GPIO line into the module.
3. BLE Services **must have**
 - 3.1. Source ANT+ channel for each service
 - 3.2. Custom Control Service
4. Supported ANT+ profiles
 - 4.1. Heart Rate (required)
 - 4.2. Bike Speed (required)
 - 4.3. Bike Cadence (required)
 - 4.4. Bike Speed/Cadence (required)
 - 4.5. Bike Power (all required)
 - 4.5.1. Power Only
 - 4.5.2. Wheel Torque
 - 4.5.3. Crank Torque
 - 4.5.4. Crank Torque Frequency
 - 4.5.5. Calibration
5. **Must have** additional GPIO wake capabilities
6. Option to save device settings **must be** available for multiple configurations
7. **Must have** BLE OTA DFU capability

TY Module Support

The CABLE IP is hosted by the nRF51422 module supplied by Taiyo Yuden. The module utilizes an onboard 32MHz crystal, 32KB of RAM and 256KB of Flash. It also hosts the PCB antenna.

Memory Allocation

The Flash memory in the module is broken up into three sections: Softdevice, Bootloader, and Application. The softdevice is the S310 dual mode ANT/BLE stack and the bootloader supports the BLE OTA update capability. With the modules preloaded with the softdevice and the bootloader the application can be loaded at manufacturing time without need to physically connect to the device.

Prototype Setup

The TY development module can be utilized for testing out the CABLE IP implementation and integrate into the Solos product.

To load the TY module, first load the softdevice using the nRFgo Studio application to load the S310 v3.0.0.

Firmware Loading

Using nRFgo Studio:

1. Plug the JTAG programming board into the TY dev board using the supplied ribbon cable
2. Connect power to the dev board and the JTAG programmer using the USB cables
3. Select the nRF5x Programming area under the Device Manager on the left side of the screen.
4. When the programming controls appear in the right hand pane of the screen , select the soft device programming tab.
5. Select the option to “Erase All”
6. Browse to the location where the S310 v3.0.0 soft device is stored and program the soft device.
7. Select the bootloader tab for programming the bootloader code. Browse to the cable_v1_5_0_bl_nrf51422_xxac.hex file supplied and program the application.
8. The application code needs to be loaded via the DFU process because the Bootloader needs to update some values in the Flash memory based on the loading of the application FW. At this point the module is advertising as SolosDfu#<device_number>. Send the solos_1_4_0_fw.zip file to your email, and long press on the zip file. You will be prompted to send the file to the config application. The app will launch and you can connect to the module and perform the application FW update.

Once all the components are loaded the CABLE IP will be operational.

Configuring CABLE IP

CABLE setup is a two-step process involving controlling ANT radio in the TY module to first scan for available ANT+ devices and save the results of the scan, then setup the mobile app to pair with the BLE services. Since The Solos mobile app is setup to consume the data from the ANT+ sensors via the BLE connection the pairing with the BLE services can happen in conjunction with the configuration of the ANT+ receivers.

The CABLE IP supports two separate configuration sets so two different bikes can be saved and allow the Solos app the ability to quickly switch between the two sets.

UART

Configuration

BaudRate: 57600
Parity: None
Stop Bits: 1
Data Bits: 8
Hardware Flow Control: None

UART Messages

There are simple messages sent from the BLE module to the main processor via the UART interface. These messages provide feedback to the Bluetooth interface about the status of the BLE module. Each message has a <cr> (0x0D) at the beginning and end of the string.

Note: The device number is the decimal value of the lower 4 hex digits of the serial number.

- DFU_APP_START:*device number*
- GCD_ENABLED/GCD_DISABLED
- STARTING...
- ADVERTISING
- CONNECT:*device number*
- DISCONNECT:*device number*
- POWERING_DOWN
- DFU_WAITING
- DFU_ACTIVE

UART AT Command Set

The CABLE interface supports a UART interface to provide a basic set of status and control information between the BLE module and the main board processor. The UART supports a simple AT command set which utilized the CR (0x0D) character to terminate the commands.

Commands Send To The BLE/ANT Module

AT

Simple AT command to test the serial interface

Message Format: <cr>AT <cr>

Response: <cr>OK<cr>

ATEn (Enable/Disable Verbose Mode)

This command enables or disables the configuration status commands normally sent via the BLE configuration interface.

n = 0 – Disable the verbose mode (default)

n = 1 – Enable the verbose mode

Message Format: <cr>ATEn<cr>

Response: <cr>OK<cr>

ATH (Force Disconnect)

Force the BLE module to drop a connection if it is connected to a BLE central device. This will allow the configuration application to gain control of the BLE connection when it wants to scan for the CABLE associated with the Solos device.

Message Format: <cr>ATH<cr>

Response: <cr>OK<cr>

ATPD (Power Down)

Command the BLE module to go into low power mode. The only recovery from this command is to either cycle the power to the module, or toggle the ANT_CTS line from low to high.

Message Format: <cr>ATPD<cr>

Response: <cr>OK<cr>

AT+DFU0 Exit DFU Mode

Command the BLE module to exit DFU mode. Device firmware updates are performed via the BLE radio interface. In normal operational mode, the module can be put into DFU mode by enabling the DFU mode using the BLE DFU service. Once the command is accepted, the module resets itself and restarts in DFU mode. While in DFU mode the normal operations of the module are not active and the module awaits a command to download new firmware. The three ways to exit DFU mode are either perform a valid firmware update, remove power from the module, or issue the serial command to disable DFU mode.

Message Format: `<cr>AT+DFU0<cr>`

Response: `<cr>OK<cr>`

AT+GCDn Enable/Disable Generic Control Device

Command the module to enable/disable the ANT+ Generic Controllable Device interface. When enabled the module generates the ANT+ beacon the remote control device uses to connect.

n = 0 – Disable GCD Interface

n = 1 – Enable GCD Interface

Message Format: `<cr>AT+GCDn<cr>`

Response: `<cr>OK<cr>`

Messages Sent From The BLE/ANT Module

AT+V

When the Verbose mode is enabled the channel status messages normally present on the BLE interface are also sent across the UART interface. Refer to section Request Active Device Settings (Request/Response) for details on the binary BLE message format.

Response: `<cr>AT+V binary_BLE_message <cr>`

AT+GCD Generic Control Device Message

When the BLE/ANT receives a message from a Generic Control Device paired with the module a message is forwarded via the serial interface. The message contains the serial number, manufactureID, sequence number, and command. The contents of the message are sent as binary numbers in little endian format separated by ASCII commas.

Message Format: `<cr>AT+GCD<16 bit serial number>,<16 bit manID>,<8bit sequence number>,<16 bit command><cr>`

WAKE From Sleep

When the BLE module powers down, it can be woken up by toggling the ANT_CTS line low (pin 22 on the module). The line should normally be driven high and toggled low to wake the module. There is a weak pullup on the module pin 22.

BLE Configuration Protocol

Request Device Settings

```
<0xX0> X=1 : Active Setting; X=2 : Saved Setting; X=3 : Factory Default Settings  
<Set Index>  
<Device Index>
```

Request Active Device Settings (Request/Response)

```
<0x10>  
<Set Index>  
<Device Index>  
<7-4:Proximity Bin,  
  3:ANT Tx En,  
  2:BLE En,  
  1:ANT En,  
  0:Device En>  
<Device Type (Read Only)>  
<Device Input Status (Read Only)> // ANT Receive Channel Status  
<Device Output Status (Read Only)> // ANT Transmit Channel Status  
<DeviceID 0-3>  
<ANT Channel>  
<ANT Event>
```

Device Settings (Set)

```
<0x11>  
<Set Index>  
<Device Index>  
<7-4:Proximity Bin,  
  3:ANT Tx En,  
  2:BLE En,  
  1:ANT En,  
  0:Device En>  
<DeviceID 0-3>
```

Request Wildcard Scan (Request)

```
<0x12>  
<Device Index>  
<Proximity Bin>  
Note: Response is returned with Command Code <0x10>
```

Cancel Wildcard Scan (Request)

```
<0x22>  
<Device Index>
```

Save Device Config To Flash (Set)

```
<0x13>
```

Revert To Saved Config

<0x23>

Restore Factory Defaults

<0x24>

Select Active Config Set (Set)

<0x14>
<Set Index>

Get Active Set Index

<0x15>

Turn Off

<0x16>

Enter DFU Mode

<0x17>

Control Generic Control Device Mode (set/response)

<0x19><0==disable; 1==enable>

Control Generic Control Device Mode (get)

<0x1A>

BLE Supported Services

Device Information	0x180A
Battery Information	0x180F
Heart Rate	0x180D
Cycling Power	0x1818
Cycling Speed and Cadence	0x1816
CABLE Configuration	4B480001-6E6F-7274-6870-6F6C65656E67
DFU Operations	00001530-1212-EFDE-1523-6F6C65656E67

OTA DFU Operation

The CABLE implementation supports over-the-air firmware updates for all three system firmware components: Application, Bootloader, and Soft Device. The three components of the system need to stay in sync with each other. Each of the components needs to be updated individually because of space constraints on the module, but the BLE DFU service supports a sequential update process. Updates to the bootloader and application code are non-destructive, however, because of the size of the soft device, the application code will get erased when a soft device update

is performed. Following a soft device update, the application code will need to be reloaded.

Application

The application is the heart of CABLE functionality. It manages all of the BLE services as well and the ANT+ connectivity.

Bootloader

The bootloader is called each time the device starts up execution and verifies the validity of the application code before allowing it to execute. If the application code is either missing or corrupt the bootloader starts advertising the Dfu service and waits for a host device to connect to it and provide it with a new application image. If there is no activity for three minutes the bootloader powers the module down to save battery. It can be awakened by toggling the CTS line on the module.

Soft Device

The soft device provides the BLE and ANT radio stack support. It works in conjunction with the application and bootloader to manage system resources and radio protocols.

OTA DFU Application Support

The iOS application support allows iOS devices with BLE radios to manage the OTA DFU process. The application is setup to receive the firmware files using the stock email application in the iOS device. When a firmware file is included as an attachment to a document, the file can be sent to the configuration application by long pressing on the zip file for the firmware and selecting the configuration application as the destination. The application is launched and the FW Update option can be selected. The application is setup to select the latest file sent to it to perform the OTA DFU. If multiple components are updated, simply return the email application and repeat the process for each element.