

Contents

.....	1
1 General	2
2 Architecture.....	3
3 BLE Service Descriptions	5
4 Connection Setup	6
5 Data Passing	8
6 Terminating Connections	10
7 Limitations.....	11
Appendix A: iOS Example Code.....	12
Appendix B: Revision History.....	14

This is the advanced Developer Guide (SDK) for those seeking to write custom mobile apps to communicate with the VEX IQ Robot Brain via the Smart Radio.

For instructions on how to setup and use the Smart Radio with your robot, please refer to the Smart Radio User Guide:

 link.vex.com/vexiq/pdf/228-3530-Smart-Radio-User-Guide

228-3530

Smart Radio Developer Guide (SDK)

1 General

The VEX IQ Smart Radio (228-3530) uses the Bluetooth® Low Energy (BLE) protocol to enable the Robot Brain to connect to a BLE-enabled device, such as a smartphone, tablet, or desktop computer, using the BLE GATT profile. Supported operating systems include, but are not limited to, iOS, OS X, Android 5.0+ and Windows 8+.

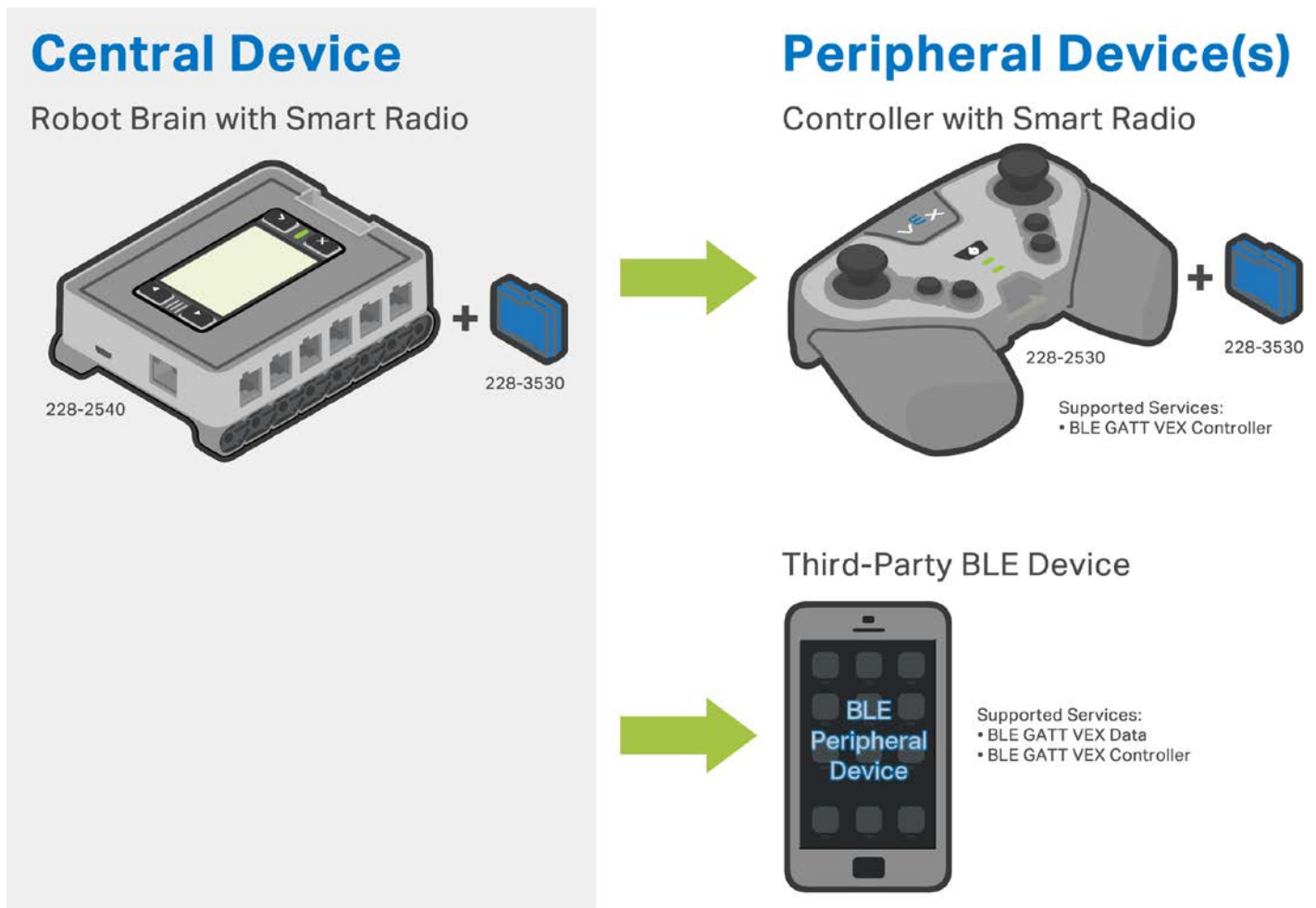


Smart Radio Developer Guide (SDK)

2 Architecture

The BLE architecture has two types of devices, Centrals and Peripherals. Centrals listen for advertising Peripherals and choose which ones to make connections with. A Peripheral can only connect to one Central at a time, but a Central can connect to multiple Peripherals simultaneously.

In the VEX IQ Control System, the Smart Radio enables the use of BLE connections in the VEX IQ Robot Brain (228-2540) and VEX IQ Controller (228-2530). The Robot Brain is always the Central device, and the Controller and any BYOD (bring your own device) / third-party connections are the Peripheral devices.



Smart Radio Developer Guide (SDK)

The Smart Radio supports two types of BLE GATT service connections between the Central and Peripheral devices: Data and Controller. The VEX IQ Controller will always connect via the BLE Controller service, while a third-party device (such as a mobile app) can connect to the Robot Brain as either

- BLE VEX Data service
- BLE VEX Controller service
- Simultaneous BLE VEX Data and VEX Controller services

The current Smart Radio implementation only allows the Central device (Robot Brain) to connect to one Peripheral of each service type at a time. This means that if you have a Robot Brain connected to a VEX IQ Controller, a second Peripheral can only connect to the Robot Brain via the BLE Data service. If there is no VEX IQ Controller present, a third-party BLE device can connect with both VEX Data and VEX Controller services.

Please refer to Section 7 about limitations of multiple simultaneous Peripheral devices at once connected to a single Robot Brain.



3 BLE Service Descriptions

3.1 BLE VEX Controller Service

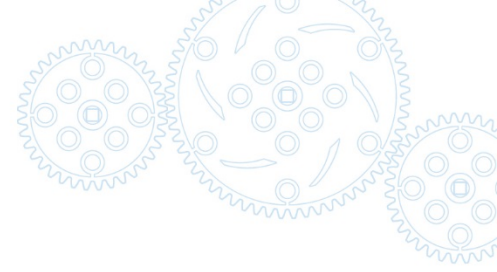
The VEX Controller Service (UUID = 08590F7E-DB05-467E-8757-72F6FAEB13A5) has the following BLE GATT characteristics:

Name	Properties Read / Write / Notify / Indicate	UUID	Default Value	Data Type	Description
JS_Data	R/W/N/I	08590F7E-DB05-467E-8757-72F6FAEB13B5	All zeros	Joystick Structure	Structure of data representing current state of VEX IQ Controller

3.2 BLE VEX Data Service

The VEX Data Service (UUID = 08590F7E-DB05-467E-8757-72F6FAEB13D5) has the following BLE GATT characteristics:

Name	Properties	UUID	Default Value	Data Type	Description
Data_Brain_RX	R/W/N/I	08590F7E-DB05-467E-8757-72F6FAEB13F5	None	Undefined	Binary data transmitted from the Peripheral to the Robot Brain
Data_Brain_TX	R/W	08590F7E-DB05-467E-8757-72F6FAEB1306	None	Undefined	Structure of data representing the current state of VEX IQ Controller



4 Connection Setup

4.1 Standard Connection Setup

To keep the connection of these devices very simple, bonding is not performed. All Smart Radio enabled Peripherals announce the Unique ID (SSN) of the desired Central (Robot Brain) in the BLE advertisement data. This allows the Central (Robot Brain) to listen and connect to only the Peripherals that wish to connect to it. Version information is also transmitted in this data to make sure the Robot Brain only connects to Peripherals with a compatible Smart Radio firmware version.

The following is a C code example of the advertisement data:

```
// GAP - Advertisement data
static uint8 advertData[] =
{
    // Flags; this sets the device to use limited discoverable
    // mode (advertises for 30 seconds at a time) instead of general
    // discoverable mode (advertises indefinitely)
    0x02, // length of this data
    0x01, // GAP_ADTYPE_FLAGS,
    0x05, // DEFAULT_DISCOVERABLE_MODE | BREDR_NOT_SUPPORTED,

    0x0B, // Length of the following data
    0xFF, // GAP_ADTYPE_MANUFACTURER_SPECIFIC,
    0x11, // VEX Company ID byte1
    0x11, // VEX Company ID byte0
    0x01, // MAJOR_VERSION of the radio firmware,
    0x01, // MINOR_VERSION of the radio firmware,
    0x06, // The following 32 bits are out SSN (Little Endian)
    0x12, // Example SSN = 987654
    0x0F,
    0x00,
    0x01, // MAJOR_VERSION of the peripheral's software
    0x05 // MINOR_VERSION of the peripheral's software
};
```

Once the Robot Brain's Smart Radio receives an advertisement which contains the correct version and SSN information, it will attempt to establish a connection. After the BLE connection is initiated, the Robot Brain's Smart Radio then queries the Peripheral first for VEX Controller BLE service, then for the VEX Data BLE service.

If the Robot Brain's Smart Radio finds one or both of these services, it will then subscribe to the characteristics of each service. Once the subscription is established, the connection is considered complete and data transmissions can start.

Smart Radio Developer Guide (SDK)

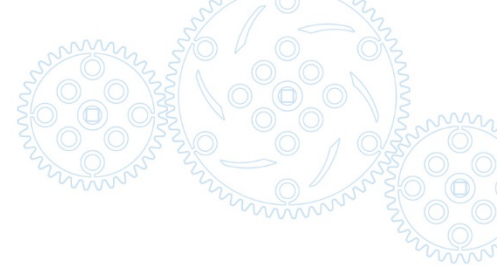
4.2 iOS Connection Setup

When programming an iOS device to connect as a Peripheral, there is no direct access to the advertisement data. Due to this limitation, the Robot Brain's Smart Radio will also look for a "DataLocalName" advertisement section with the following format: "SSN-Peripheral_Major_Version-Peripheral_Minor_Version-Radio_Major_Version-Radio_Minor_Version" (ex. "987654-1-5-1-1").

The following is an Objective-C example of this modified iOS advertisement section:

```
[self.peripheralManager startAdvertising:@{ CBAvertisementDataServiceUUIDsKey : @[[CBUUID  
UUIDWithString:JS_SERVICE_UUID]], CBAvertisementDataLocalNameKey:[NSString  
stringWithFormat:@"%d-%d-%d-%d-%d", (unsigned long)self.SSN, JS_MAJOR_VERSION,  
JS_MINOR_VERSION,RADIO_MAJOR_VERSION,RADIO_MINOR_VERSION]}}];
```

The Robot Brain's Smart Radio will look for both of these sections in every advertisement packet it receives, but will only use the first valid section.



5 Data Passing

5.1 BLE VEX Controller Service

Once the Robot Brain's Smart Radio has subscribed for indications for the JS_Data characteristic, the VEX IQ Controller Peripheral must start sending the JS_Data immediately at the configured rate (Default is 25ms intervals).

The following is a C code structure of JS_Data:

```
typedef struct //Warning: Keep these long word aligned (4 byte)
{
    UInt8  j1_y;           // Right Joystick Y value (0-255)
    UInt8  j1_x;           // Right Joystick X value (0-255)
    UInt8  j2_y;           // Left Joystick Y value (0-255)
    UInt8  j2_x;           // Left Joystick X value (0-255)
    UInt8  buttons;        // Bit representation of the JS buttons
    UInt8  battery;        // JS battery power level
    UInt8  mainPwr;        // Main Power level
    UInt8  idleTime;       // Idle time in minutes
    UInt8  pwrOffDelay;    // Power off delay: value is ++ when power button is pressed
    UInt8  contCount;      // 1 up counter that rolls over at 255
    UInt32 tickCount;      //Timing information used by radios only
} JoyStickRecord;
```

The Robot Brain's Smart Radio will timeout and close the connection if a new JS_Data packet is not received within 1 second.

Smart Radio Developer Guide (SDK)

5.2 BLE VEX Data Service

Once the Robot Brain's Smart Radio has subscribed for indications for the Data_Brain_RX characteristic, the VEX Data Peripheral may start transmitting the data to the Robot Brain's Smart Radio.

Data is transmitted to the Robot Brain's Smart Radio via indications on the Data_Brain_RX characteristic. Data is received from the Robot Brain's Smart Radio via "WriteNoResponse" ATT messages from the Robot Brain. This connection is exactly the same as the USB connection to the Robot Brain; the Robot Brain will respond to the same commands sent over USB.

The primary difference between connections established over the Smart Radio versus direct USB, is that the Smart Radio data rate is much slower and does not have any flow control. The data buffers on the Robot Brain's Smart Radio are ~64 bytes so it will fill quickly. If the buffers fill up, the data will be lost and the message will have to be resent.

BLE limits the size of transmitted packets to 20 bytes. All messages larger than that will have to be broken up into 20 byte packets in both directions. It is important to note that all BLE indications are "Best Effort" – this means that the BLE stack will do its best to get the packet to the other party, but there is no guarantee.

The USB connection to the Robot Brain has priority over Smart Radio connection. **If a Robot Brain has an active USB connection, all messages from the BLE VEX Data Service will be dropped.**



6 Terminating Connections

Peripherals can terminate connections at any time with the Robot Brain. When a Peripheral terminates the BLE Connection with the Robot Brain, the Robot Brain will notice the termination and immediately start looking for other Peripherals. The Robot Brain will only terminate a BLE connection if the BLE VEX Controller Service connection times out.



7 Limitations

7.1 Data Latency from Multiple Connections

With the current Robot Brain firmware version (v1.16), if more than one BLE Peripheral is connected to the Robot Brain at once, both connections will see noticeable latency (~2-3 seconds). It is currently recommended to only connect one device at a time.

In the scenario of having a VEX IQ Controller connected at the same time as a wireless program downloader, it is recommended that the wireless downloader disconnect shortly after the download is complete. Once the second device is disconnected, the latency disappears.



Appendix A: iOS Example Code

A.1 Requirements

A1.1 iOS Software

The example code contains Xcode projects and source code that will compile and run on iOS devices. This code has been tested on iOS 8.1-8.4, and should be able to run on any iOS 5.0+ or newer device. A zip archive of these sample applications is included with this Developer Guide (SDK).

A1.2 Apple Hardware

The Smart Radio will communicate with the following mobile Apple devices:

- iPhone 4S and newer
- iPad (3rd Generation) and newer
- iPad Mini (all generations)

The VEXSmartRadio static library code can be compiled as a Mac OS X static library and run on any OS X device that supports Bluetooth[®] 4.0 and newer hardware.

A1.3 VEX Firmware

The Smart Radio requires the following VEX firmware versions (or newer):

- Robot Brain v1.16 or newer
- Controller v1.05 or newer
- Smart Radio v1.1 or newer

A.2 User Interface

The sample applications were designed and tested on an iPhone 6. The UI for the applications was created for the iPhone 6 screen. Some modifications may be required to run the same application on other devices.

Smart Radio Developer Guide (SDK)

A.3 File Descriptions

A.3.1 VEXSmartRadio Project

This project contains a static iOS library that can be added to your custom iOS application project. The public interface for this library is documented in the following file:

~/VEXSmartRadio/VEXSmartRadio/VEXSmartRadio.h

This library is an easy way to get started with communicating with a Smart Radio enabled VEX IQ Robot. All other VEX-provided sample applications use this library to communicate with the VEX IQ Robot Brain.

A.3.2 VEXJoystickFramework

This project contains a sample iOS application that connects to a Smart Radio enabled VEX IQ Robot Brain as a virtual VEX IQ Controller, using the simultaneous BLE VEX Controller and BLE VEX Data services.

To use this application, enter the Unique ID (SSN) of the VEX IQ Robot Brain that you wish to connect to in the text box, then press the virtual "Power" button on the App to connect. Once connected, the status label text will change from "Connecting" to "Both Connected". "Both Connected" indicates that the iOS app has connected via both the BLE VEX Controller and BLE VEX Data services.

While connected, the app has virtual buttons and analog joysticks to fully simulate the VEX IQ Controller. All button presses and joystick values will be transmitted to the Robot Brain using the BLE VEX Controller service. Pressing the virtual "Power" button will disconnect the iOS device.

A.3.3 VEXDownloadDemoFramework

This project demonstrates how to download a new User Program to the VEX IQ Robot Brain from a precompiled .bin file. This project also demonstrates how to query the Robot Brain for which User Programs are present, and then Run/Stop a desired User Program.

WARNING! The file downloaded to the VEX IQ Robot Brain MUST be in the VEX IQ User program format. Downloading any other type of data could corrupt your User Program Catalog, and remove any other User programs currently installed on your Robot Brain.



Appendix B: Revision History

Date	Revision	Changes
2015-08-31	1	Initial document release
2016-03-02	2	Corrected order of data in example code in Section 4.1