MD46 Bluetooth Server

Rev C

Mid-Continent Instruments and Avionics

Model Series MD46
Governor Controller / Data Recorder
(EMU)

Revision History

Revision	SW Ver	Description	Author	Date
A	1.0.0.0	Initial Release	KEY	06/01/18
B	1.0.1.0	Updates for changes to Register API Rev D	KEY	07/05/18
C	1.0.1.1	Change endianness on Register Address, Add examples and current limitations Added **Identifier** support	KEY	07/23/18

Table of Contents

1   MD46 Bluetooth Server

The MD46 Bluetooth Server GUI application (hereby the application) called emulates the behavior of the MD46 Governor Controller / Data Logger Bluetooth peripheral in order to assist with development of the Bluetooth Low Energy client application.

1.1   Installation and Setup

The application requires a Windows 7 or newer PC and the Microsoft .Net Framework version 4.6.2 installed on said PC.

The application can be installed by extracting the program files into the desired location or by running the self-extracting executable.

The BlueGiga BLED112 USB CDC driver must also be installed, which can be obtained from Silicon Labs.

1.2   Server Application

The MD46 Bluetooth Server GUI application can be started by launching the MD46Bluetooth.exe file.
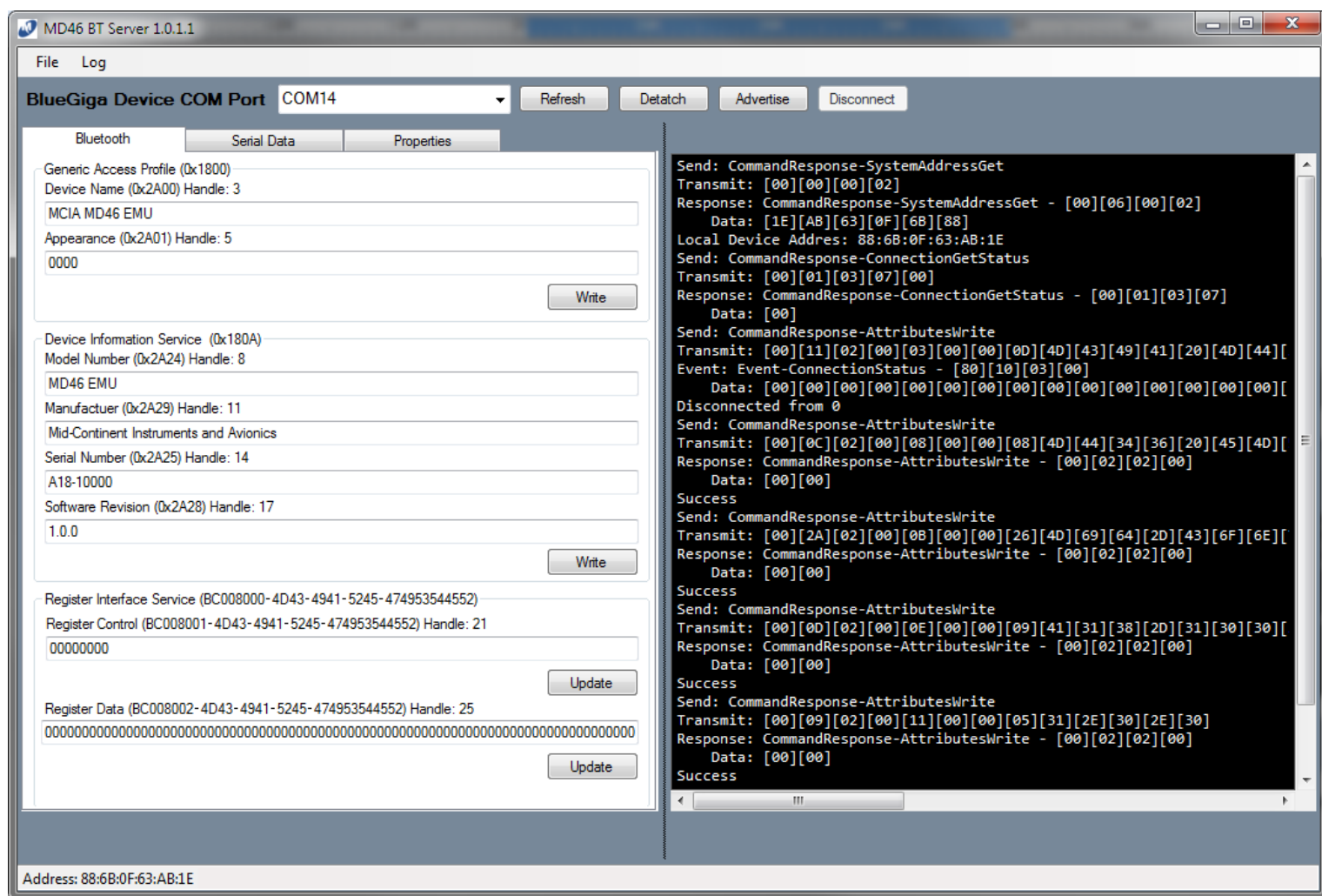
1.2.1 Connecting to the BLED112 Dongle

The BLED112 USB dongle is identified on the Windows platform as a serial COM port. Connect to the dongle by selecting the COM port that matches the dongle's COM port and clicking the **Attach** button.

If the COM port does not show up in the drop down box, hitting **Refresh** will rescan for available COM ports.

Once the application has been attached to the BLED112 dongle, the server BLE address should be displayed at the bottom of the UI.
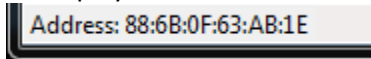
1.2.2 BLE Advertising

To put the dongle into advertising mode, clock the **Advertise** button. Once a device connects to the dongle, the application will automatically disable the **Advertise** button until the remote device is disconnected.

To stop advertising, press the **Advertise** button again (which should say **Stop Adv**).

1.2.3 GAP / Advertising Properties

The *Device Name* and *Appearance* properties used for advertising and discovery of the BLE server are part of the Generic Access Profile (GAP) and can be modified by modifying text in the appropriate field and clicking the **Write** button for the GAP group.

If these properties are changed while the dongle is advertising or if a remote client is attached, the change may not be observable on the remote client side until the command to advertise is sent again.

1.2.4 Device Information Service

The Device Information Service is a Bluetooth accepted service providing additional device information. This data is automatically written by the GUI to the dongle when attaching and can be updated via the GUI by changing the appropriate setting text and clicking the **Write** button for the Device Information Service group.

The change in value can be observed on the client side by reading the corresponding property.

1.2.5 Register Interface Service

The Register Interface Service is an MCIA custom BLE service that provides an interface to the MCIA Register API. The UI values represent the current "internal" server value. Changing the value and hitting the corresponding **Update** button will update the internal server value. When issuing a read from the client, this is the value that is returned. When issuing a write from the client, this value will be automatically updated in the UI.

1.2.5.1 Register Control Characteristic

This characteristic is 4 bytes in length and has the Read, Write, and Notify properties. Note that the MD46 will support writing only the 16 bit address to this characteristic.

> *The usage of this characteristic can be found in the MD46 Register API document.*

Register Control Characteristic (Read/Write/Notify)	
Bytes	Description
0:1	16 bit Address **(Big Endian)**
2	Register Control Byte (Optional)
3	Register Length (Optional) Use for reading contiguous registers in sequence

1.2.5.2 Register Data Characteristic

This characteristic can be from 1 to 20 bytes of data and its values are dependent on the current value of the Register Control characteristic.

For values that require more than 20 bytes of data to be transmitted, the MD46 will send BLE messages in multiples of 20 bytes, with the first byte in each message indicating the zero indexed packet number. The last message in each packet may be shorter than the maximum 20 bytes allowed.

> *The usage of this characteristic can be found in the MD46 Register API document.*

Register Data Characteristic (Read/Write/Notify)	
Bytes	Description
19:0	Variable length data (1-20 bytes)

1.2.6 Serial over BLE Service

The Serial over BLE service provides a method to pass serial data between the server and the client. This is achieved by exposing a Write No Response "Receive" characteristic and a Notify "Transmit" characteristic.
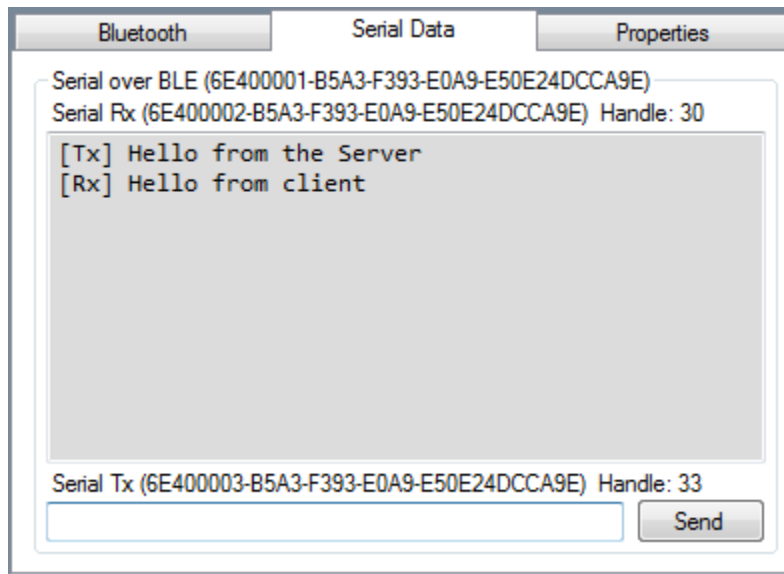
The Tx and Rx characteristics are from the viewpoint of the server. The client should write data to the receive characteristic and subscribe to notifications from the transmit characteristic.

This characteristics and the UUID's chosen match the Nordic UART Service. More details can be found here:

https://infocenter.nordicsemi.com/index.jsp?topic=%2Fcom.nordic.infocenter.sdk52.v0.9.2%2Fble_sdk_app_nus_eval.html

1.2.6.1 Receive Characteristic

Serial over BLE Receive Characteristic (Write No Response)	
Bytes	Description
0:19	Variable length data (1-20 bytes)
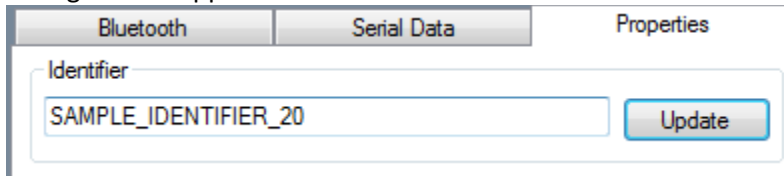
1.2.6.2 Transmit Characteristic

Serial over BLE Transmit Characteristic (Notify)	
Bytes	Description
0:19	Variable length data (1-20 bytes)

1.2.7 Server Properties

1.2.7.1 Identifier

The **Identifier** property will update the internal setting and save the value to the persistent storage of the application.

1.2.8 Usage Examples

- Read Log Entry Count
  - Write **0x50 0x10** to Register Control
  - Issue Read to Register Data
- Set Log Read Start Time to first entry
  - Write **0x50 0x20** to Register Control
  - Write **0x00 0x00 0x00 0x00** to Register Data
- Set Log Read End Time to last entry
  - Write **0x50 0x30** to Register Control
  - Write **0xFF 0xFF 0xFF 0xFF** to Register Data
- Read a single log entry from the current **Log Read Start Time**
  - Write **0x51 0x00** to Register Control
  - Issue Read to Register Data
    - Two transmissions per Log Entry (packet) will be transmitted, with the first byte indicating which portion of the packet (0x00 or 0x01) the transmission is.
- Start automated log dump
  - Subscribe to Register Data notifications
  - Write **0x51 0x00** to Register Control
  - Write **0x01** to Register Data
    - Two transmissions per Log Entry (packet) will be transmitted, with the first byte indicating which portion of the packet (0x00 or 0x01) the transmission is.
    - The log dump will continue until either a 0x00 is written to the Register Data characteristic or any other value is written to the Register Control characteristic.
- Read all limits
  - Write **0x30 0x00 0x00 0x14** to Register Control
  - Issue Read to Register Data
    - Receive 20 bytes in register address order – 0x3000, 0x3004, 0x3008, etc.

1.3   Application Limitations

- The log entry interval setting will currently be ignored
- The log entry data is not valid and no exceedance entries are currently generated.
- The generated sample log entries are only on 1 second boundaries.