BLEGUI APPLICATION

USER GUIDE

Monday, 15 October 2012

Version 1.7

TABLE OF CONTENTS

1 Version history

Version	Comments
1.7	Bluetooth Smart SDK v.1.1 beta II updates

2 Introduction

BLEGUI is a simple user interface application that allows a developer to quickly test and evaluate Bluegiga's *Bluetooth* Smart products. The main purpose of BLEGUI is to hide the complexity of the binary protocol (BGAPI) used by the host to control the *Bluetooth* Smart stack. BLEGUI offers a more user friendly approach to the usage of the BGAPI. BLEGUI can however be a very good tool for quick prototyping and debugging of *Bluetooth* Smart applications.

This user guide walks you through the basic usage of BLEGUI.

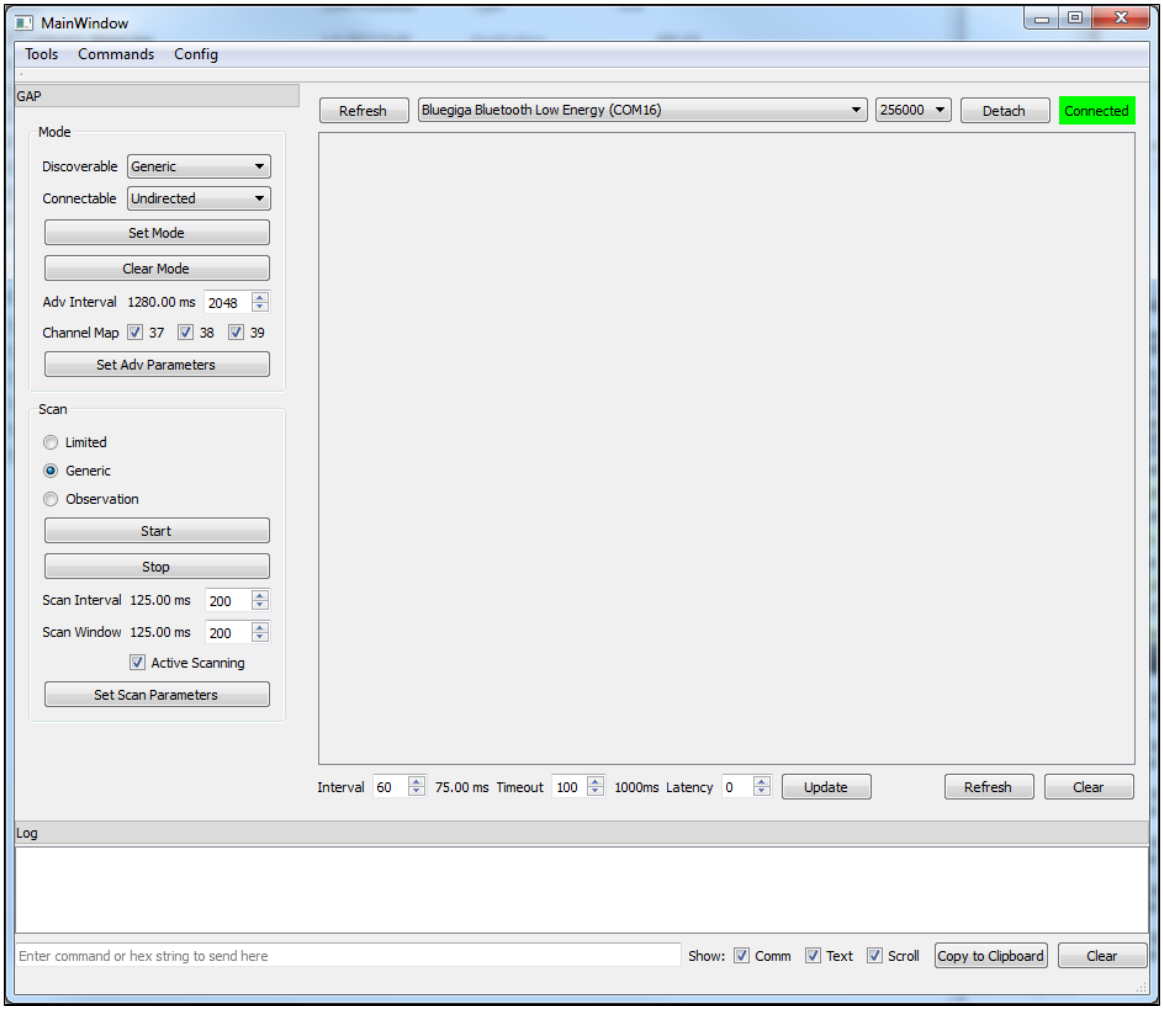


Figure 1: BLEGUI application

3 Getting started

BLEGUI works at the moment with the following products:

- BLE112 - *Bluetooth* Smart module
- BLED112 - *Bluetooth* Smart USB dongle
- DKBLE112 - BLE112 development kit

BLUGUI can control the above products via USB or UART interfaces, so therefore an appropriate firmware must be programmed into the hardware.

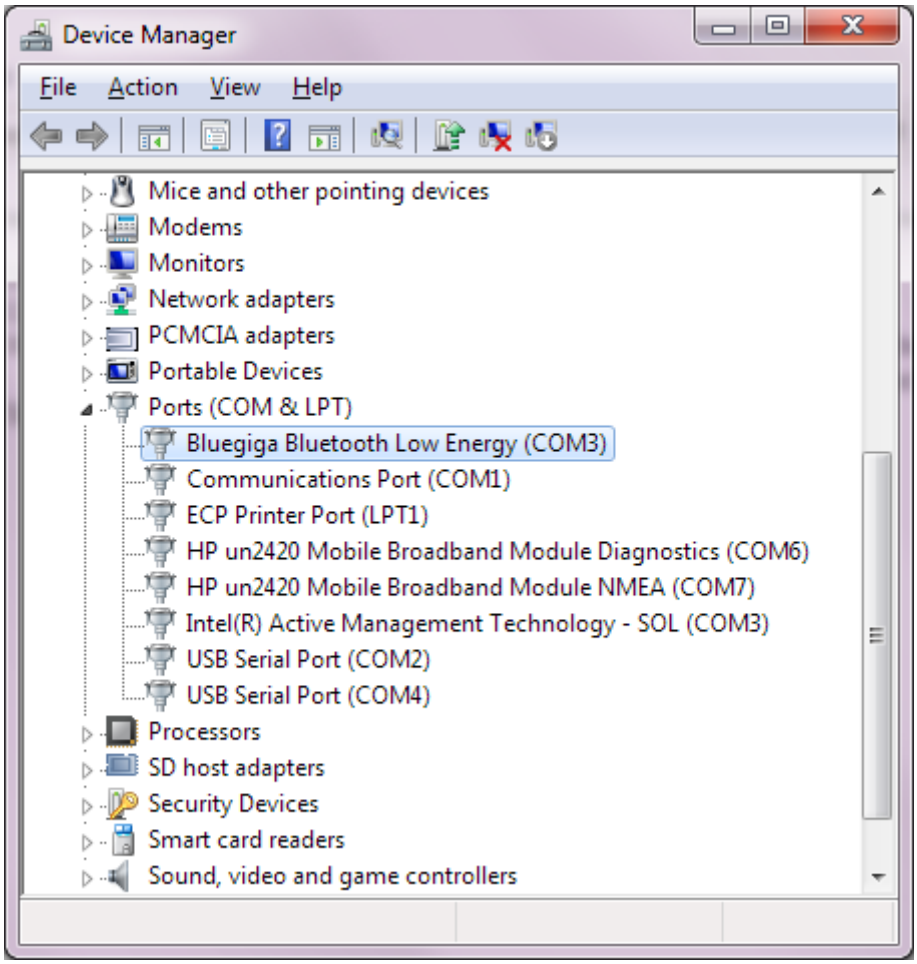
- USBCDC example allows the control over USB interface
- UARTDEMO example allows the control over UART interface

3.1 Preparations

If you have not used BLEGUI before, you first need to do some preparations

Using USB

1. Download the *Bluetooth* Smart software development kit from Bluegiga's Tech Forum
2. Extract it you your PC
3. Connect the BLED112 USB dongle or DKBLE112 via USB to your PC
4. Windows will recognize the device and prompt for a driver software, use the driver located in the ***windrv*** folder
    a. If Windows does not prompt for the driver, go to ***Device manager,*** select the unknown device and click ***Update driver software***
5. If Windows security gives a waring about the driver installation, choose ***Install this driver software anyway***
6. After driver installation a device called ***Bluegiga Bluetooth Low Energy*** *should be visible under* **Ports** *in Windows* **Device manager**
7. If you see this, the driver installation is complete



Using RS232

If you have BLE112 modules or DKBLE112 which use UART interface, no driver installation is needed.

1. Connect the BLE112 or DKBLE112 to your PC via RS232 (a 3.3V level shifter such as MAX3232 might be needed)
2. Make sure the DKBLE112 is power via USB, since a CR2032 cannot power a RS232 level sifter
3. Notice also that the default UARTDEMO firmware has power mode 3 (PM3) enabled, which disables the UART timings. In order to communicate with the BLEGUI you need to have the wake-up pin (P0_0) enabled

Starting BLEGUI

Finally start the **BLEGUI** application located under ***bin*** folder.

> ⚠ The DKLBE112 development kit comes pre-programmed with a Health Thermometer example. This is a standalone application and the DKBLE112 is not recognized over USB interface.

4 BLEGUI user interface

After starting the BLEGUI the main view is visible. The first thing you need to do is to select the correct device you want to use from the drop down menu.

1. Select one of the available **Bluegiga Bluetooth Low Energy** devices from the drop down menu
2. Click ***Attach*** to open the serial port connection
    a. A green ***Connected*** light should turn on in the user interface
3. Execute **Commands->Info** command to read out the firmware version, to make sure the communication works and that BLEGUI version matches to the firmware version.

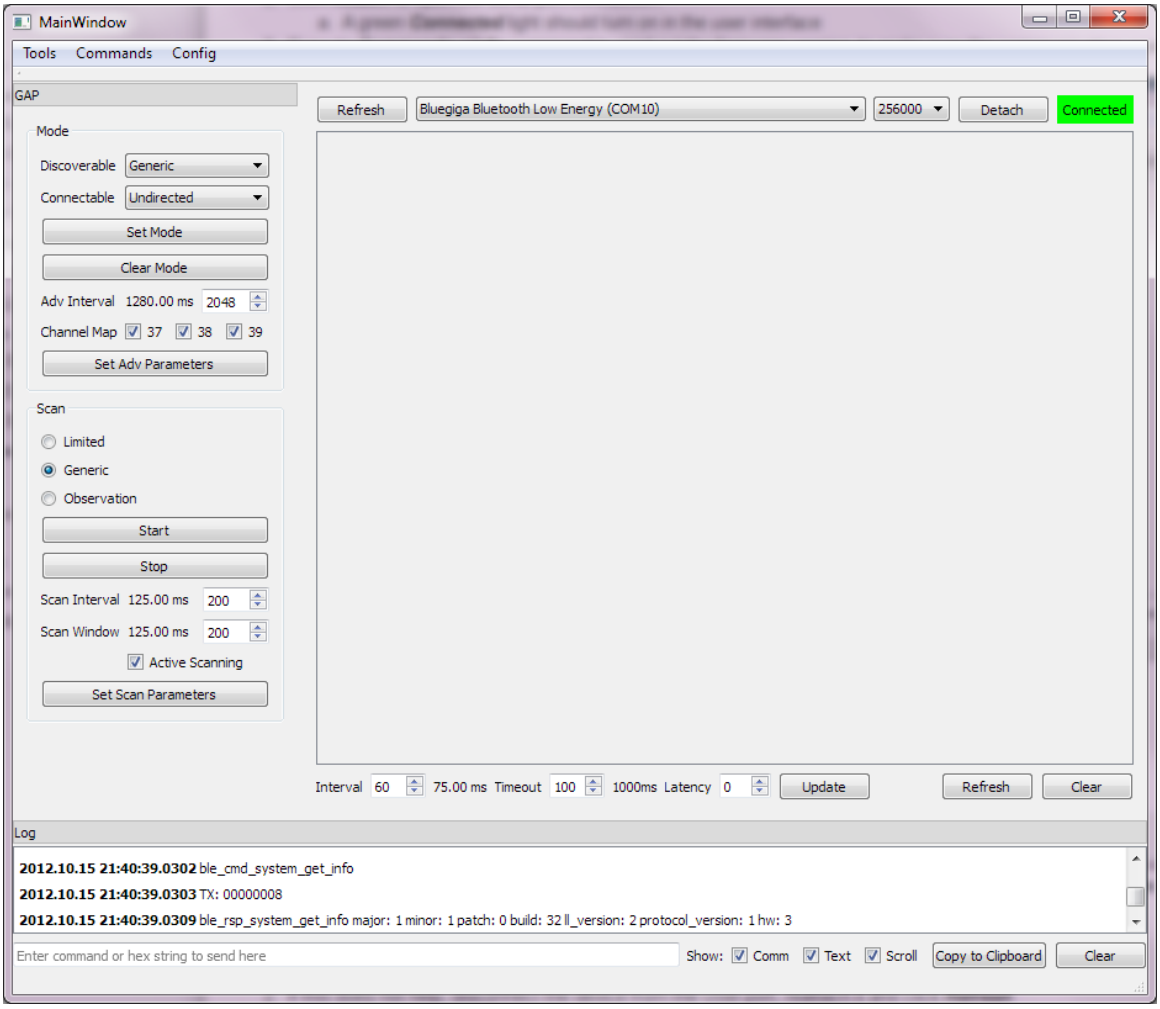


Figure 2: BLEGUI main view

If you do not see any Bluetooth low energy devices in the drop down menu, please try the following:

1. Click ***Refresh*** button
2. If this does not help, disconnect the device from the USB port, reattach it and click ***Refresh***
3. If this does not help, close BLEGUI, make sure the device is visible in the ***Device manager*** and restart BLEGUI

The main components in the BLEGUI are:

- **GAP** - Generic Access Profile toolbar, which allows you to control the visibility, connectability, broadcast and scanning modes of a device
- **Log -** Shows the raw and user friendly communications log between BLEGUI and the USB dongle
- **Action view** - The center part of the user interface changes depending on the performed action. It can show discovered Bluetooth low energy devices, GATT data bases etc.

4.1 Generic Access Profile controls

Generic Access Profile (GAP) controls allow you to change devices discoverability and connectability modes and scan for other *Bluetooth* Smart devices.

Discoverability modes:

Mode	Explanation
No	Device does not advertise itself
Limited	Device advertises itself in limited advertisement mode
Generic	Device advertises in generic advertisement mode
Broadcast	Device is in broadcast mode

Connectability modes:

Mode	Explanation
No	Device cannot be connected
Directed	Device can be connected only by a specific *Bluetooth* Smart device
Undirected	Device can be connected by any *Bluetooth* Smart device

Once the desired discoverability and/or connectability settings have been selected, **Set Mode** button enables or changes the configuration.

Clear mode button disables advertisements.

Advertisement parameters:

Option	Explanation
Advertisement interval	Configures the interval between advertisement events
Channel map	Selects which advertisement channels are in use
Set Adv Parameters	Send the BGAPI command to change the advertisement parameters

Set Adv Parameters changes the advertisement parameters.

Scanning options:

Mode	Explanation
Limited	Only devices advertising in limited advertisement mode are shown
Generic	All devices advertising are shown
Observation	Only broadcasting devices are shown

Once the desired settings have been selected, **Start** button starts scanner.

Stop button on the other hand will pause scanning.

Scan parameters:

Option	Explanation
Scan interval	Configures the interval at which the scan procedure is made
Scan windows	Configures the scan window, which defines how long the scan procedure lasts i.e. how long the RX is active.
Active scanning	Enables or disables the active scanning mode

Press **Set Scan Parameters** to send the BGAPI command to change the scan parameters

A device cannot perform scanning and advertisement at the same time.

4.1.1 Advertising

Advertising is an operation where ***advertiser*** starts to broadcasts advertisement packets. All Bluetooth low energy devices within the range can pick up these packets and discover the ***advertiser*** . The advertisement packets typically contain data telling if the device is connectable and bondable, transmit power level and supported services, but this may depend on the devices configuration.

To start advertising, simply:

- Select the device's discoverability and connectabilty mode
- Press **Set Mode** button
- Optionally you can set the advertisement parameters. If not set, firmware defaults will be used.
    - Advertisement parameters need to be configured before advertisement is started

The device starts to broadcast advertisement packets an all the advertisement channels.

.

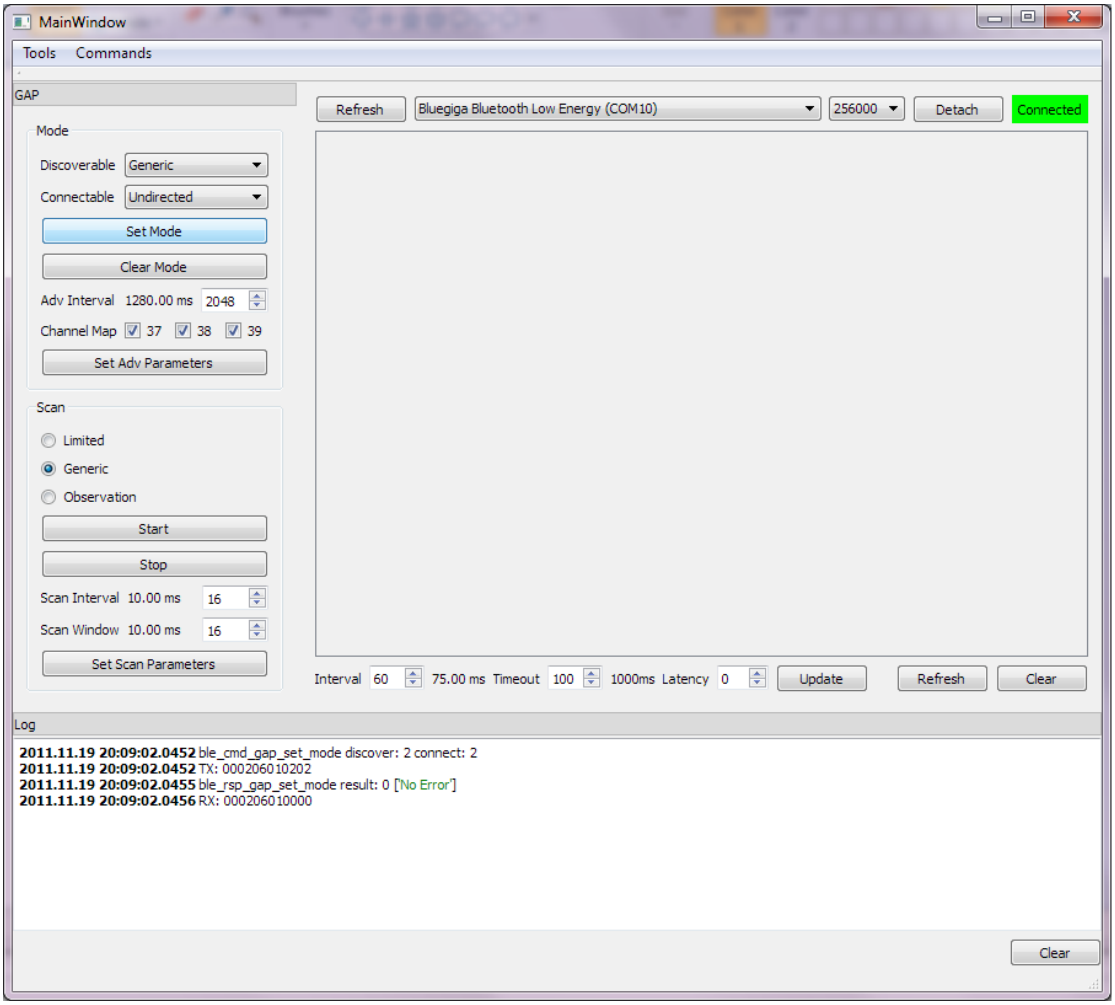


Figure 3: Starting advertisement

To stop advertising:

- Set the Discoverable mode to **No**
- Set the Connectable mode to **No**
- Press **Set Mode** button
- Or alternatively simply press **Clear Mode** button

To change advertisement parameters on the fly:

- Stop advertisements
- Change advertisement parameters

- Restart advertisement

4.1.2 Scanning devices

Scanning is an operation where a ***scanner*** listens on all the three advertisement channels for advertisement packets sent by the ***advertisers***. When a proper advertisement packet is received from an ***advertiser*** a scan request is made and a remote device is discovered.

To perform scanning, simply:

- Select the scanning mode (Limited, Generic, Observation)
- Press **Start** button
- Optionally you can configure the scan parameters. If not configured, firmware defaults will be used.
    - Scanning parameters need to be set before before scanning is started

If *Bluetooth* Smart devices are discovered, they will be displayed in the action view

.

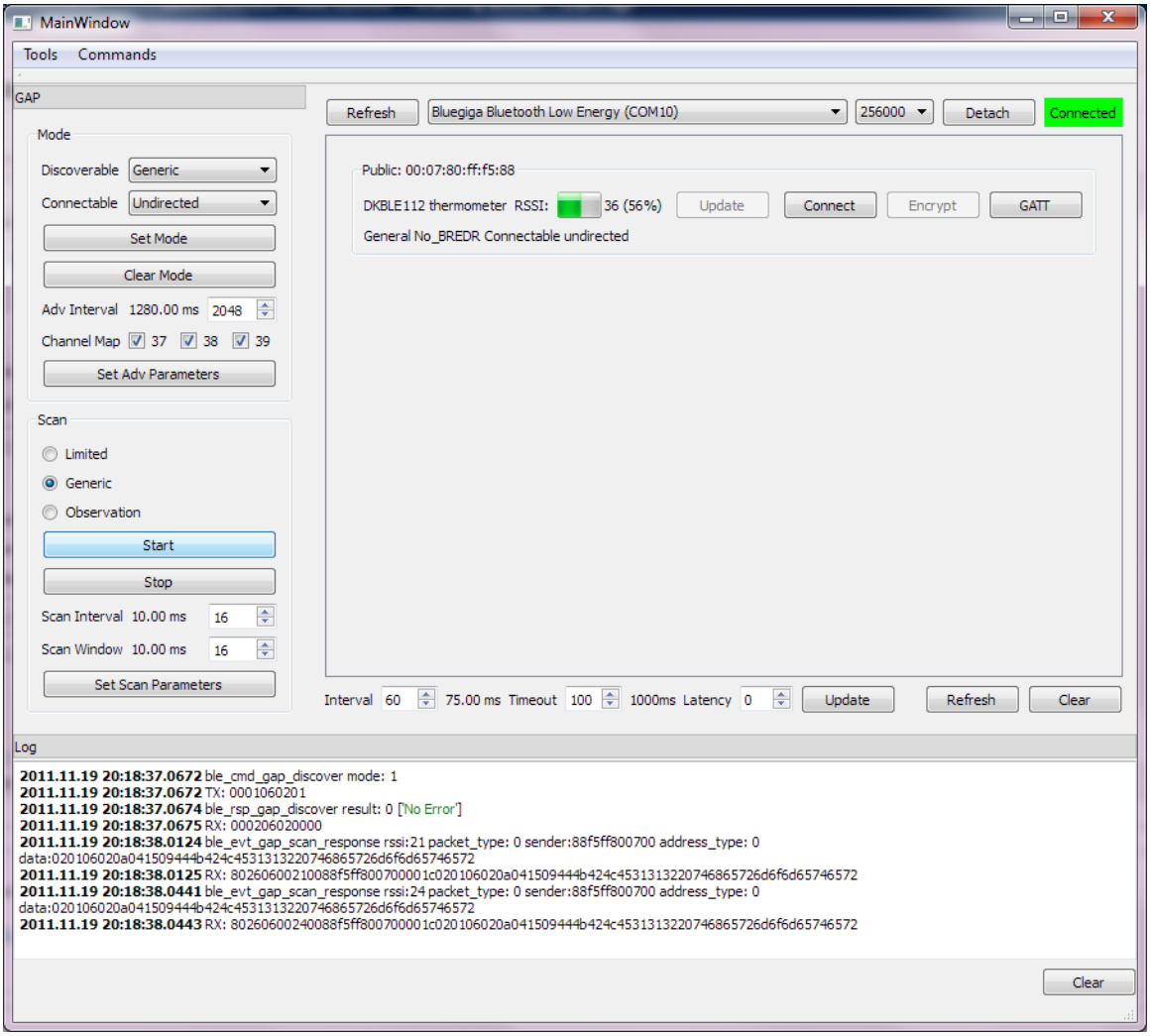


Figure 4: Performing scanning

In the figure above a single device has been discovered. The device has the following features:

- **Device name** : DKBLE112 thermometer
- **RSSI** : 36 (56%)
- The device does not support BR/EDR
- Device is generally connectable (undirected)

The log panel shows both human readable messages and raw (BGAPI) communications between the BLEGUI and the *Bluetooth* low energy stack running either on BLE112 or BLED112.

To stop scanning:

- Press the **Stop** button or
- connect the discovered device

To change advertisement parameters:

- Stop scanning
- Change parameters
- Restart scanning

4.1.3 Opening and closing connections

To connect another *Bluetooth* Smart device, simply select one of the discovered devices from the action view and press the **Connect** button.

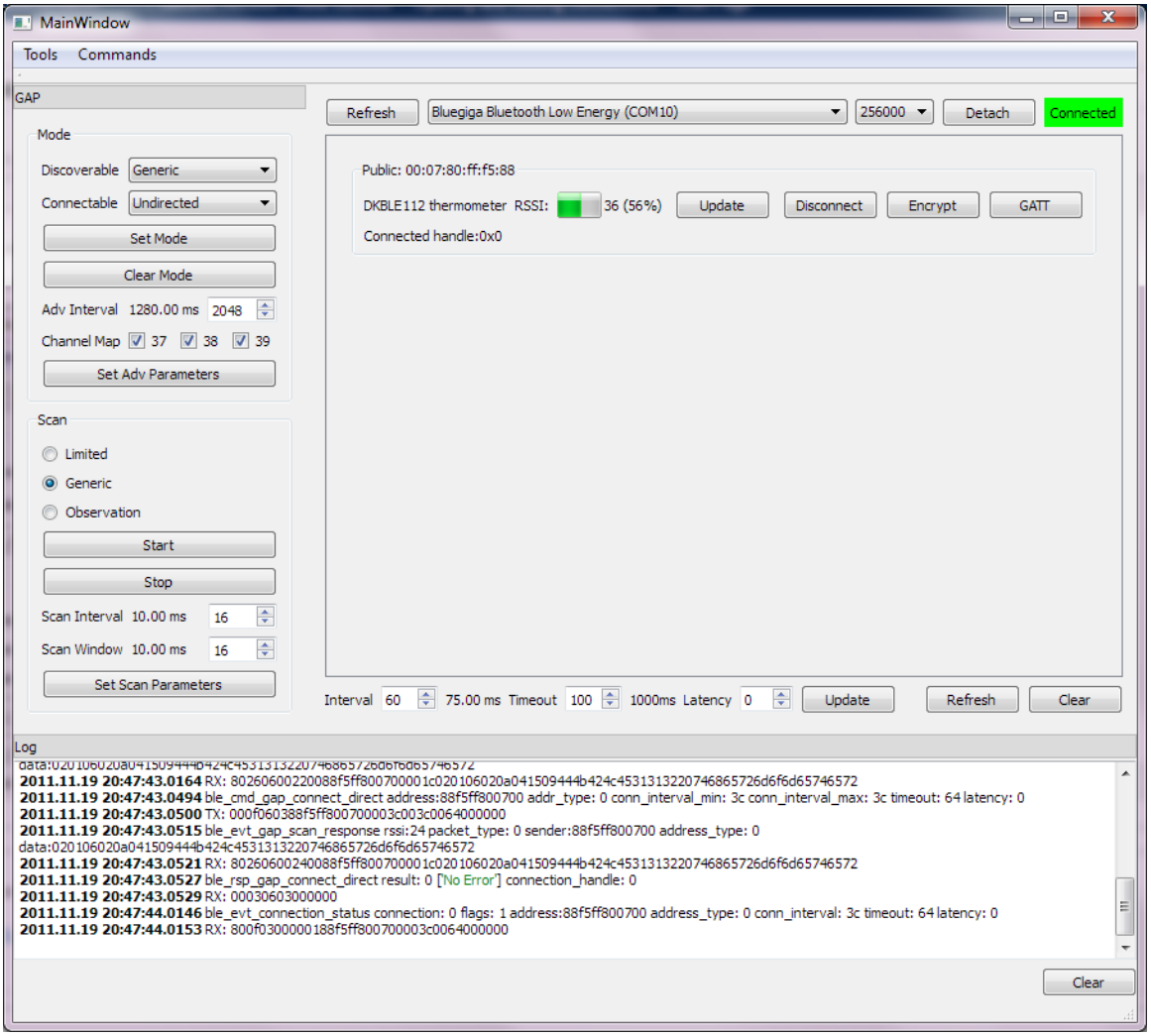


Figure 5: Successful connection to a device

A successful connection will change the **Connect** button to a **Disconnect** button, a connection handle is displayed in the action screen. In the log panel a direct connect and a connection status events are shown.

A nonzero return code to direct connect event indicates that the connection failed.

A disconnection event indicates that the connection has been closed or lost.

Connection parameters

The *Bluetooth* low energy connection parameters can be changed from the toolbar below the action view. The following parameters can be changed:

Option	Explanation
Interval	Defines the connection interval in units of 1.25ms. Connection interval defines how often data can be exchanged between the devices. Range: 7.5ms to 4000ms
Timeout	Defines the supervision timeout in units of 10ms. If the devices cannot communicate within the defined timeout, the connection will be terminated. Range: 7.5ms to 30200 ms
Latency	Defines the slave latency i.e. how many anchor points (connection intervals) the slave can skip if it has no data to send.

The connection parameters can be updated any time during the connection life time by changing the value of the parameters and pressing the **Update** button.

Connection termination

Terminating a connection is simply done by pressing the **Disconnect** button in the action view. A disconnect event that the connection has been closed.

4.2 Generic Attribute Profile controls

Once connected, the data transactions can be made using the Attribute Protocol (ATT). The Attribute Protocol gives access to the Generic Attribute Profile (GATT) remote database and allows operations like: read, write, indicate and notify. Generally speaking the ATT can be used to discover services on a device and exchange data.

4.2.1 Service discovery

Service Discovery

To discover the services that a device supports you need to do the following steps:

- With BLEGUI configure a *Bluetooth* Smart device to start advertisements (device A)
- With BLEGUI scan the device A with a second *Bluetooth* Smart device (device B)
- Connect device A from the device B
    - Optionally you can perform bonding and connection encryption
- Once connected simply press the **GATT** button in the action view
    - A GATT database browser will appear
- Press **Service Discover** button to perform a service discovery

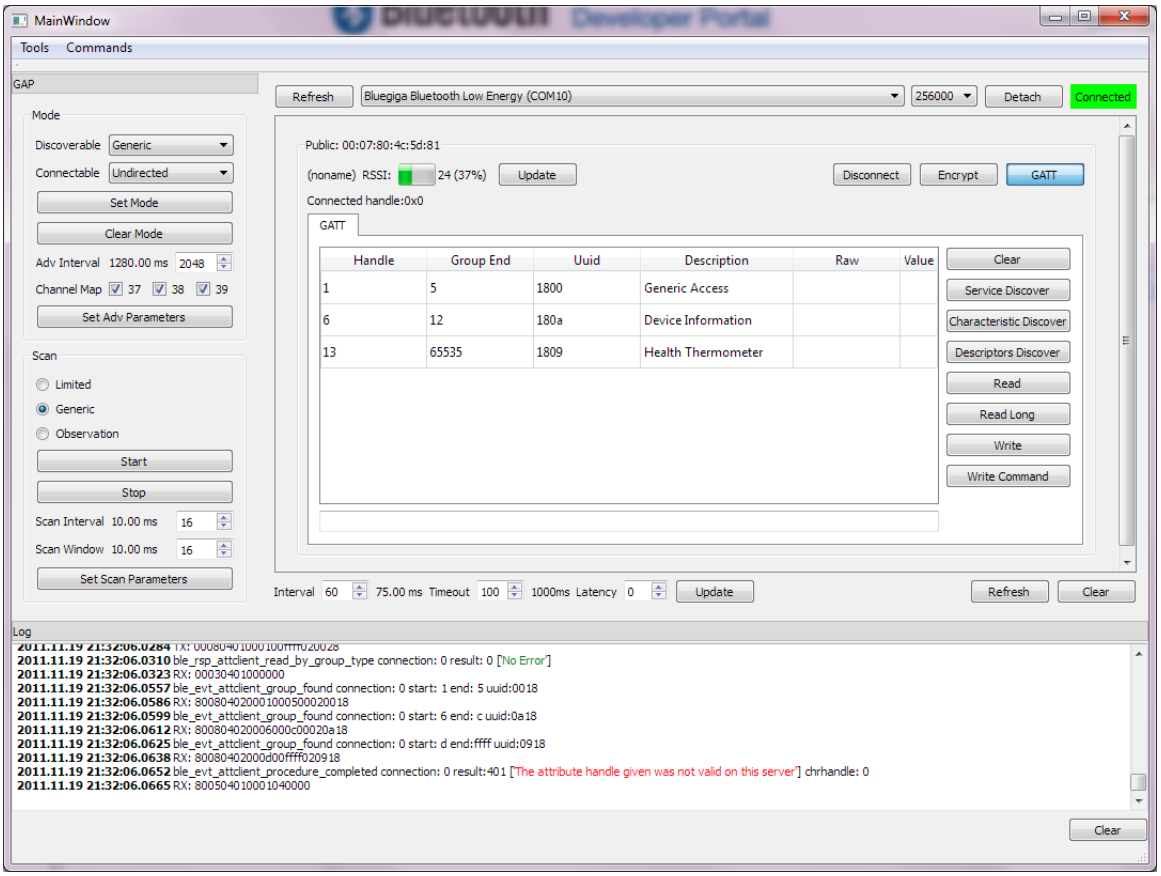


Figure 6: Service Discovery

In the above example a service discovery has been performed and three services have been found.

The services supported by the remote device are:

Service	UUID	Handles
Generic Access Profile service (GAP)	1800	0x01 to 0x05
Device Information Service (DIS)	180a	0x06 to 0x12
Health Thermometer service (HTM)	1809	0x13 to 0x65535

4.2.2 Characteristics and descriptors discovery

Characteristics discovery

Bluetooth Smart services consist of one or several characteristics. Characteristic is a value with a know type for example weight in kilograms or device name. To discover the characteristics of a service, simply select the desired service and press **Characteristics Discover** button.

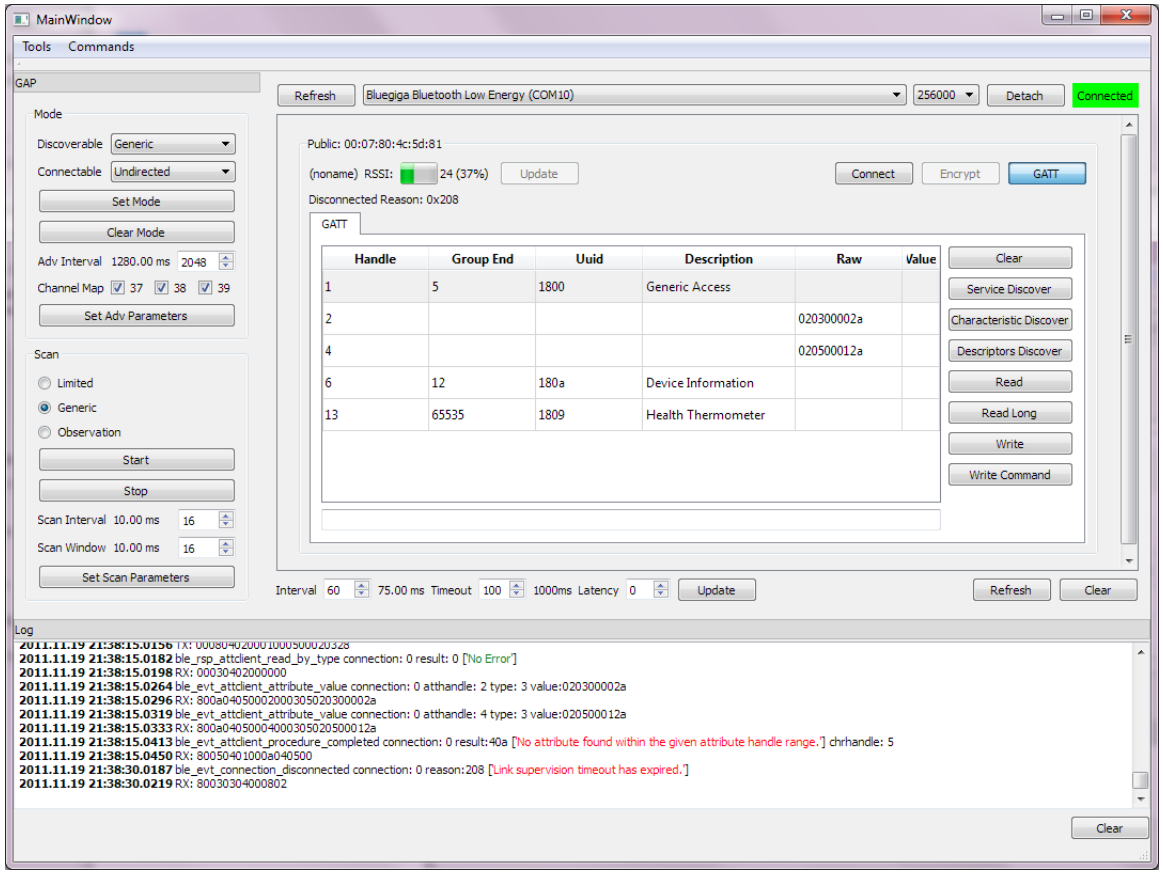


Figure 7: Characteristics declaration

In the figure above a characteristics discovery has been performed to the GAP service. GAP has two characteristics shown in the GATT tools view. The characteristic declarations refer to:

- **Device name** attribute which tells the friendly name of the device. From raw data we see that it has UUID: 2a00, handle: 0x0003 and properties: 0x02
- **Device appearance** which tells the device type. UUID: 2a01, handle: 0x0005; properties: 0x02

So the characteristics discovery can be used to find out what kind of data is exposed by a service and how the data can be accessed.

Descriptors discovery

The descriptors discovery on the other hand goes through the service handle by handle and discovers the UUID of every characteristic and data fields. Select first a Primary Service then click the Descriptor Discover button to find out more about the characteristics of that service.

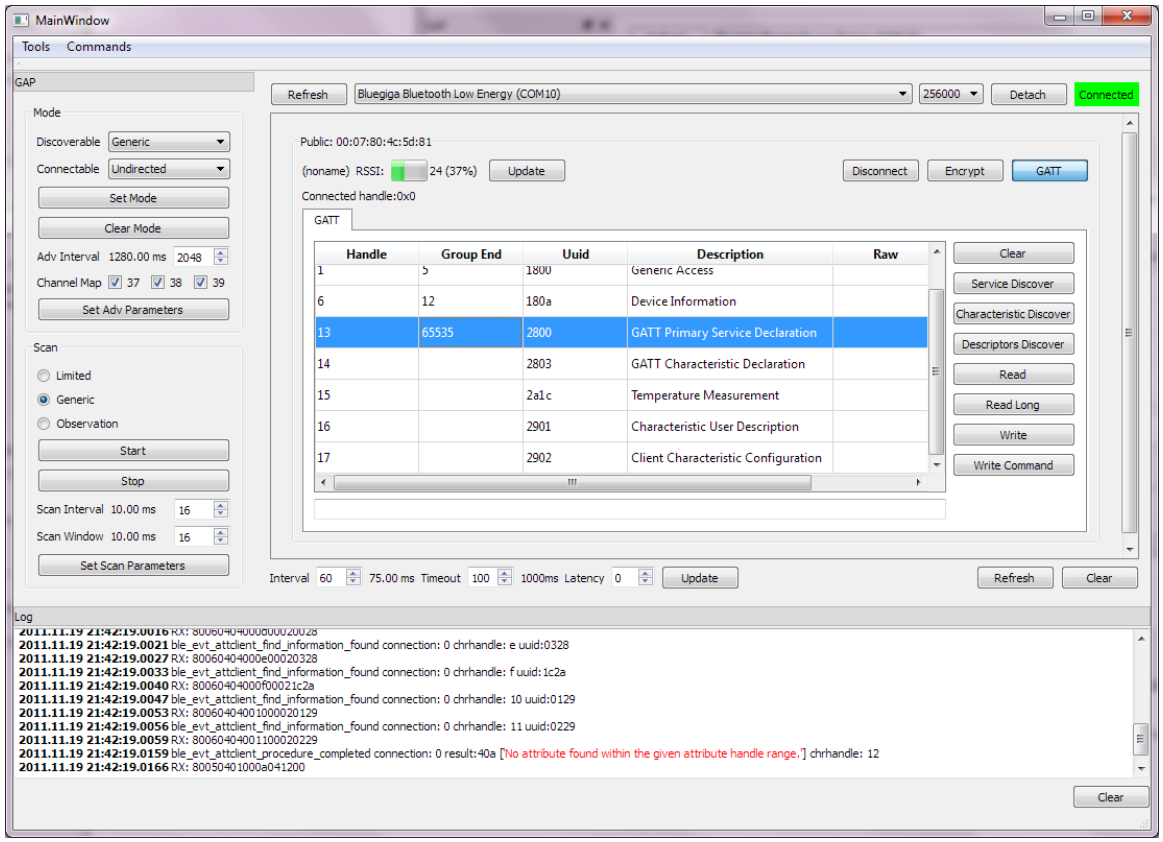


Figure 8: Descriptors discovery

In the figure above, after pressing the **Descriptors Discover** button the Characteristics Declarations of the GAP service are exposed in more detail, together with their related attributes.

4.2.3 GATT operations

GATT procedures

Generic Attribute Profiles offers several procedures for manipulating attribute values. The procedures include:

Operation	Description	Payload	Acknowledged
Read	Reads a characteristic value	22 bytes	Yes
Write	Write characteristic value	20 bytes	Yes
Write command	Write characteristic value up to 20 bytes without Acknowledgement	20 bytes	No
Read long	Read long characteristics up to 64 KBytes	Up to 64kB*	Yes
Indication	Start characteristic indication (max payload 20 bytes)	20 bytes	Yes
Notification	Start characteristic notification (max payload 20 bytes)	20 bytes	No

*) Depends how long attributes are supported by the attribute server

Read

To read a remote characteristic value the GATT tool contains a **Read** button. For example to read the Device name:

- Select UUID 2A00 from the GAP service
- Press **Read** button

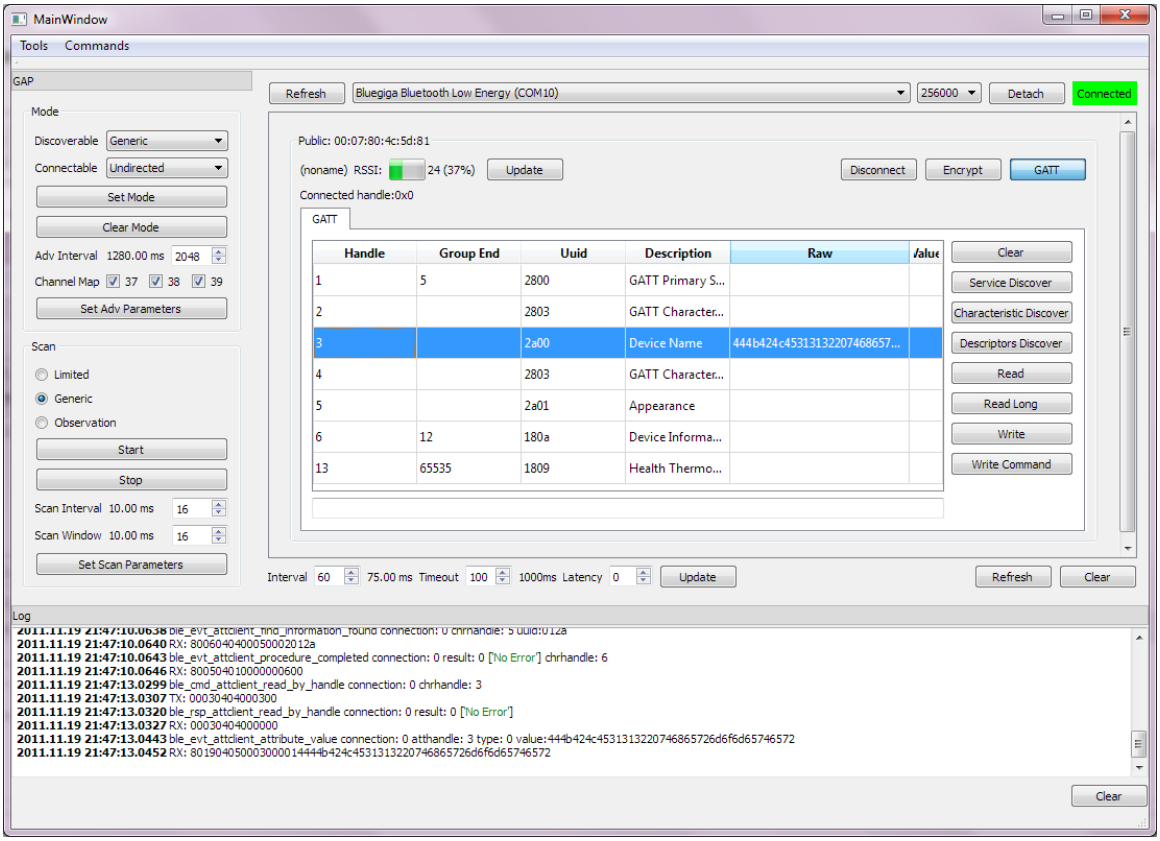


Figure 9: Read device name

The **Raw** field is updated with device name.

0x54656d706572617475726206d6561737572656d656e74 corresponds to "DKBLE112 thermometer" in ASCII

Read long

Read long is similar to read, but it can be used to read attributes, which are longer than 22 bytes. Read long can read attribute values up to 64 KBytes.

Write

If the attribute has a write property, then the remote value can also be written. To write a value:

- Select an attribute, which value can be written
- Select the field below the GATT tool
- Write the value in to the field
- Press **Write** button

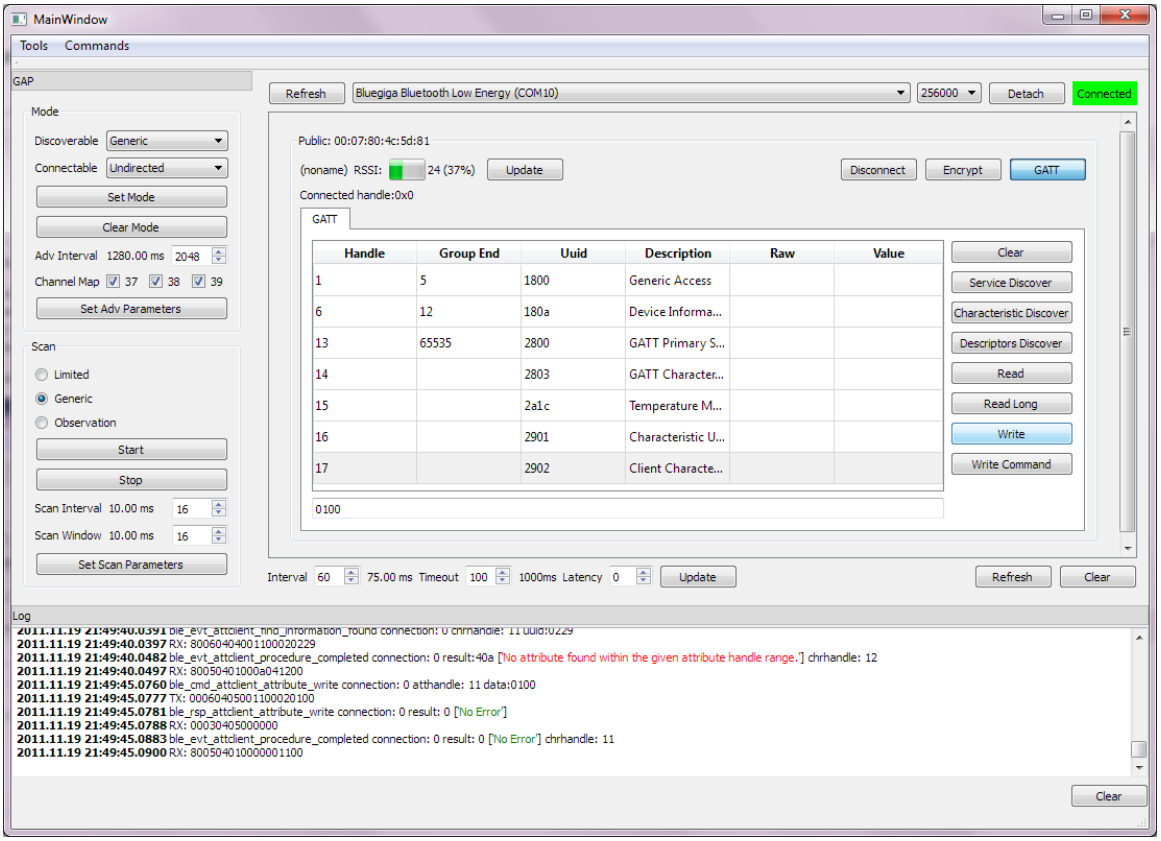


Figure 10: Writing a value

In the figure above value 0xc0ffeeee is written to characteristic with UUID 1811. The update the value on the GATT tool just press **Read**.

Write command

Write command is similar to Write, but Write command is not confirmed (acknowledged).

Notify

When a characteristic is configured to be notified, a server will automatically notify the characteristic value changes to the client. The client however has to first configure the server to start notifications.

If a characteristic has a notification property it will also have an additional property called **Characteristic client configuration**. To start the notifications the client needs to write 0x01 into that characteristic. After this is done the server will automatically notify the characteristic value changes.

In the example below, a Heart Rate Service is used to demonstrate how notifications are started. Notifications are enabled by writing 0x01 to Characteristic Client Configuration.

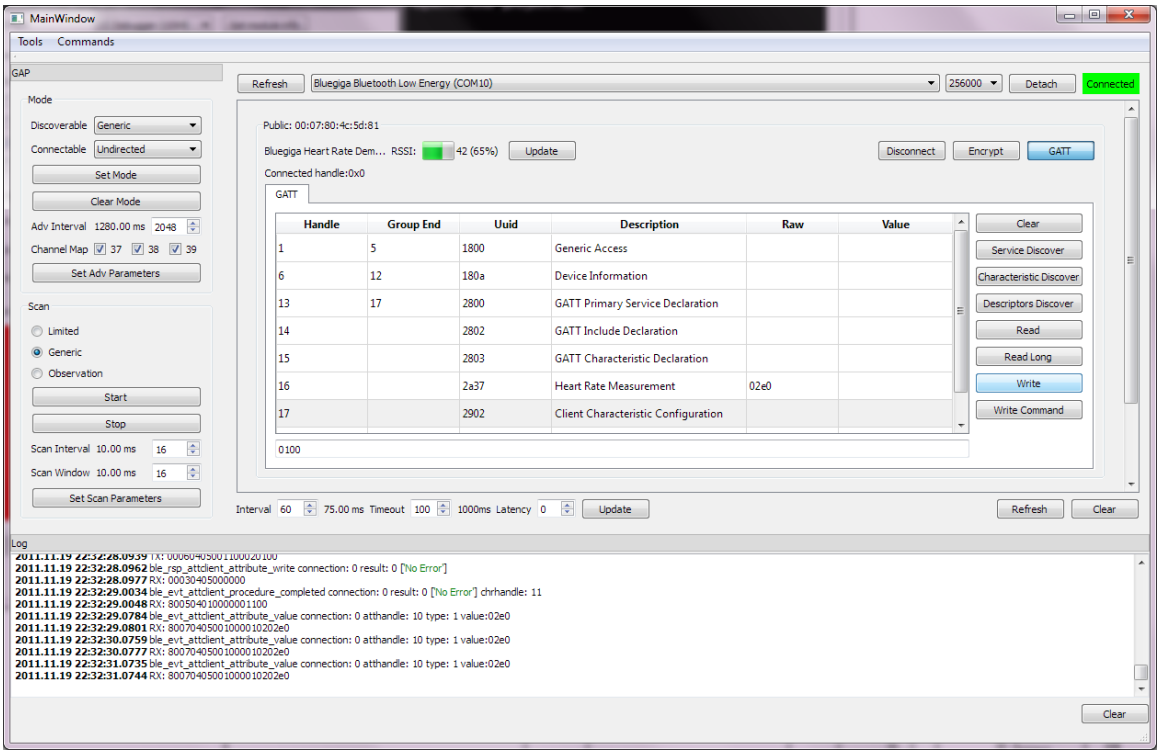


Figure 11: Enabling notifications

Value 0x01 must be written to UUID 2902 (Characteristic client configuration), for the notifications to start.

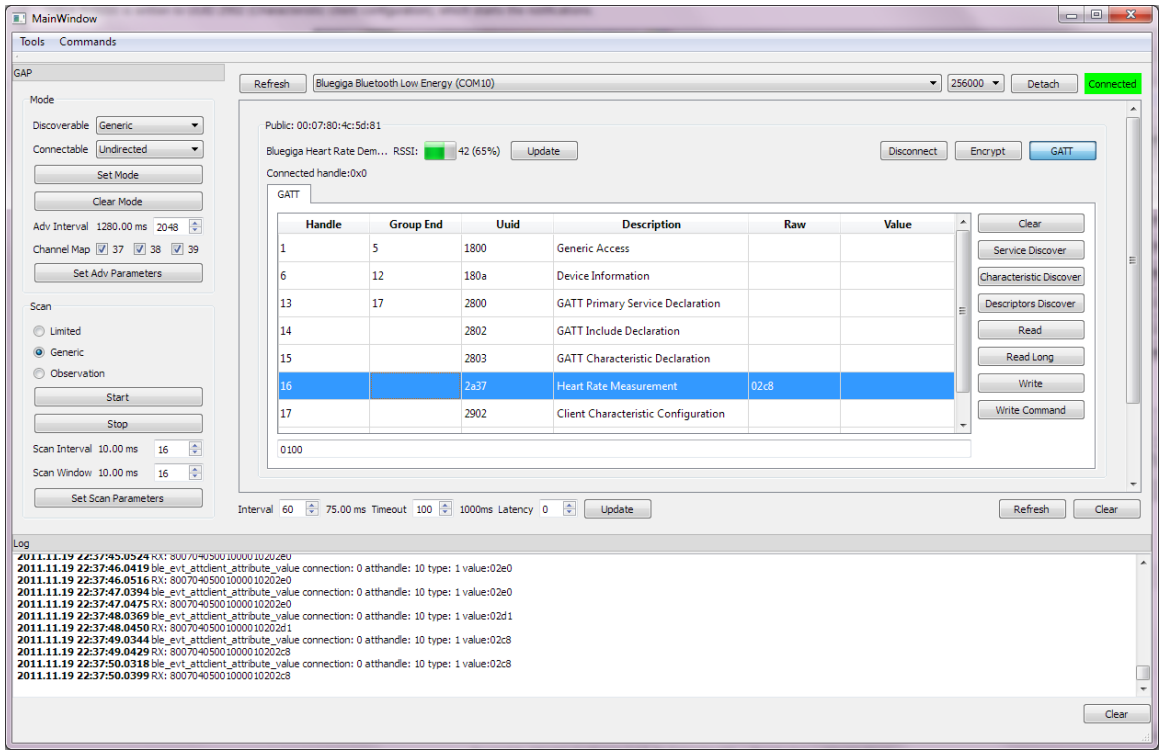


Figure 12: Notifications started

The attribute server starts the automatic notifications of characteristic with UUID 0x2a37 (Heart Rate Measurement).

Indicate

When a characteristic is configured to be indicated, a characteristic server will automatically indicate the characteristic value changes to the client. The client however has to first configure the server to start indications.

Indications are activated similarly to notifications, but instead of writing 0x01 to the **Characteristic client configuration** value 0x02 must be used.

The difference between indications and notifications is that notifications are NOT confirmed by the attribute client, whereas indications are.

4.3 Tools menu

Tools menu give access to various local functions such as GATT server, PS store, IO interfaces etc.

4.3.1 GATT server

The GATT server tool allows attribute **read** and **write** operation to the device's local GATT database. Read operation can be used to read the local attributes and their values and the write operation on the other hand to write attribute values to the local database.

The operations are done using the attribute handles.

Read

Select the desired attribute handle and press **Read** button. The attribute value is displayed in hex format.

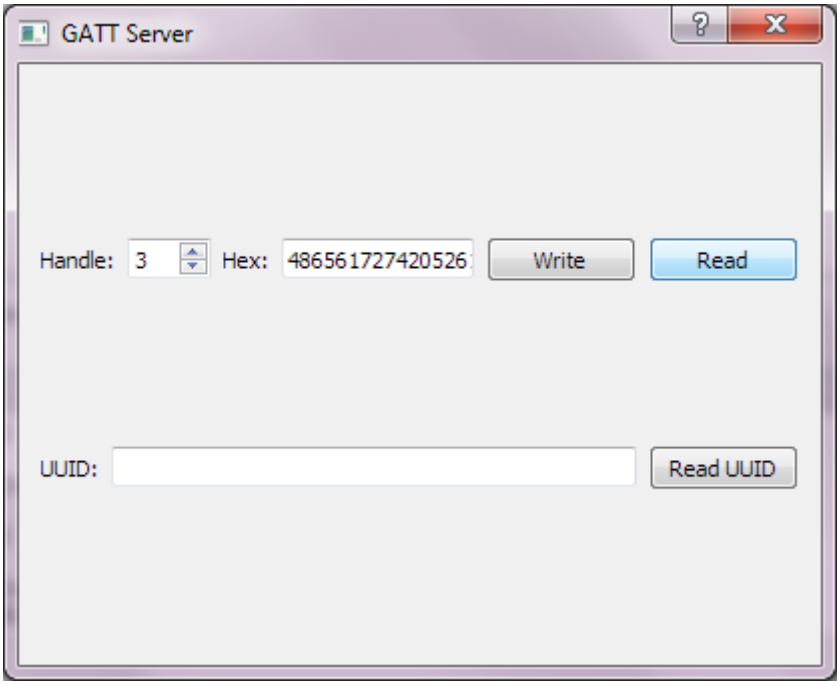


Figure 13: GATT server tool

Write

Select the desired attribute handle, write the new value into the **Hex** field and press **Write** button.

> ⚠ If the attribute is marked as **const** in the the GATT database, it cannot be written even with the local GATT database tool.

4.3.2 Security Manager

This window can be used to control the Bluetooth low energy stacks Security Manager Protocol. The Security Manager can be used to configure the devices I/O capabilities, enable or disable Man-in-the-middle protection or perform the passkey entry.

The I/O capability options are:

I/O capability	Explanation
NoInputOutput	The device does not have a user interface. "Just works" mode
DisplayOnly	The device has only a display
DisplayYesNo	The device has a display and yes/no button
KeyboardOnly	The device has only a keyboard
KeyboardDisplay	The device has both display and keyboard

Man-in-the-middle protection

Checking the Man-in-the-middle protection check-box enabled the man-in-the-middle protection.

Minimum key size

This configures the minimum acceptable PIN code size

Bondable

When this box is checked, the device can be bonded.

Set parameters

When the button is pressed the security configuration command is sent to the device.

Passkey entry

This button can be used to enter the PIN code.

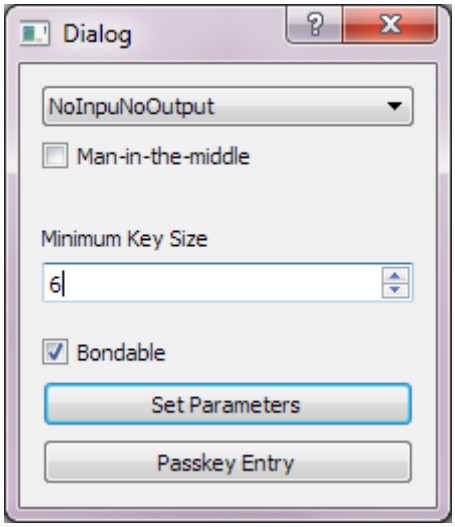


Figure 14: Security Manager dialog

Enabling security parameters and bonding mode on the local device

1. Select **I/O capabilities**

2. Check or un-check the **MITM** protection
3. Check **Bondable** mode
4. Press **Set Parameters** button

4.3.3 Persistent store

Persistent Store tool can be used to modify the local devices persistent store. Persistent store can be used to permanently store data on the local devices flash memory.

Available operations:

Mode	Explanation
Refresh	Read all the values from the Persistent Store
Delete	Delete the selected value from the Persistent Store
Add	Add a new value into Persistent Store
Write	Write the selected value into Persistent Store
Defrag	Defragments the Persistent Store

PS-keys form 8000 to 807F can be used by the applications and each of the keys can contain up to 32 bytes of data.

4.3.4 IO

The I/O dialog allows flexible control of the BLE112 I/O interfaces. It enables the following functions:

Function	Explanation
Port 0	Allows reading, writing and configuration of Port 0
Port 1	Allows reading, writing and configuration of Port 1
Port 2	Allows reading, writing and configuration of Port 2
ADC	Allows reading of ADC inputs as well ADC configuration
SPI	SPI selection, configuration and and SPI transfer functions
Endpoints	Enables data transmission to various endpoints. Can be used to for example send data to BGscript, USB or UART
I2C	I2C read and write functions

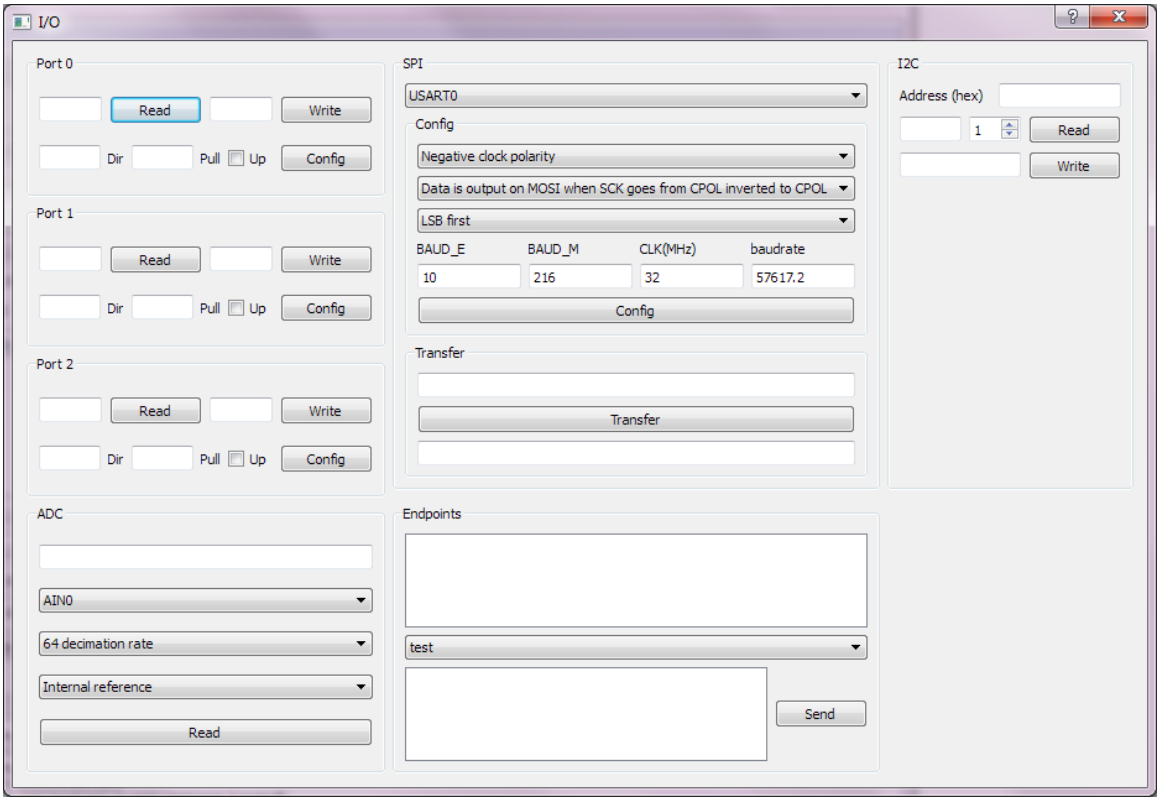


Figure 15: I/O configuration and usage dialog

4.4 Commands menu

The commands toolbar contains some useful functions.

Reset

Performs a software reset and detaches device from BLEGUI.

DFU

Boots the device into DFU mode.

> ⚠ With the BLE dongle a DFU driver needs to be install the first time the device to booted into DFU mode.

Installing the firmware using DFU:

1. Open **BLEGUI** application
2. Select the device you want to update (COM port)
3. From **Commands** menu boot the device to **DFU** mode
    a. Install the DFU driver if requested by the Windows operating system
4. Perform the DFU update as in the image below

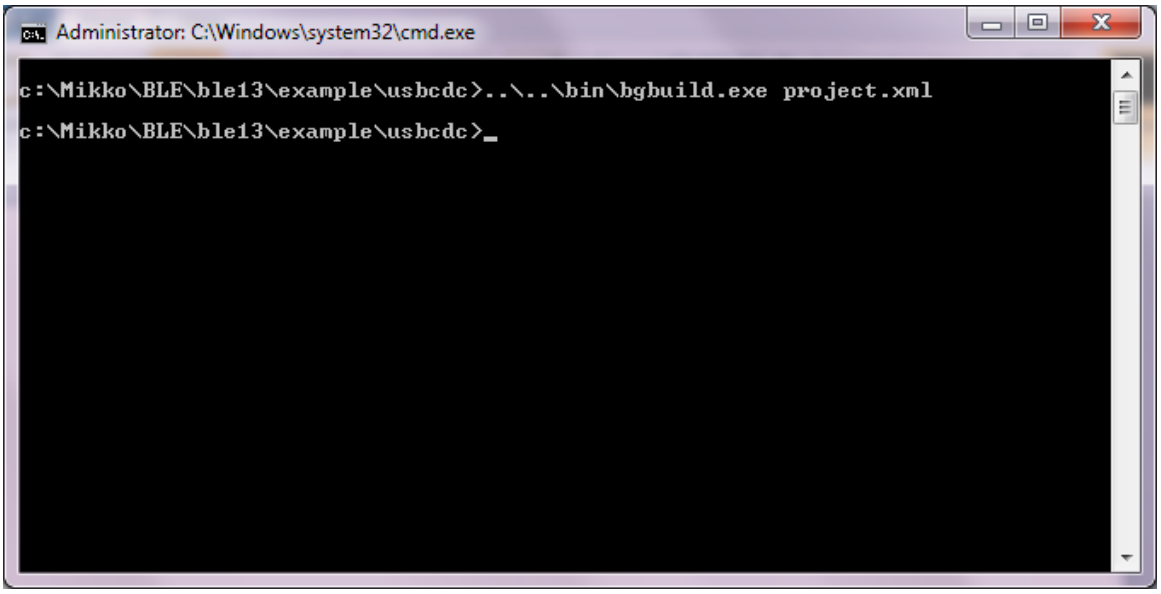


Figure 16: Compiling USB dongle software example with BGBuild

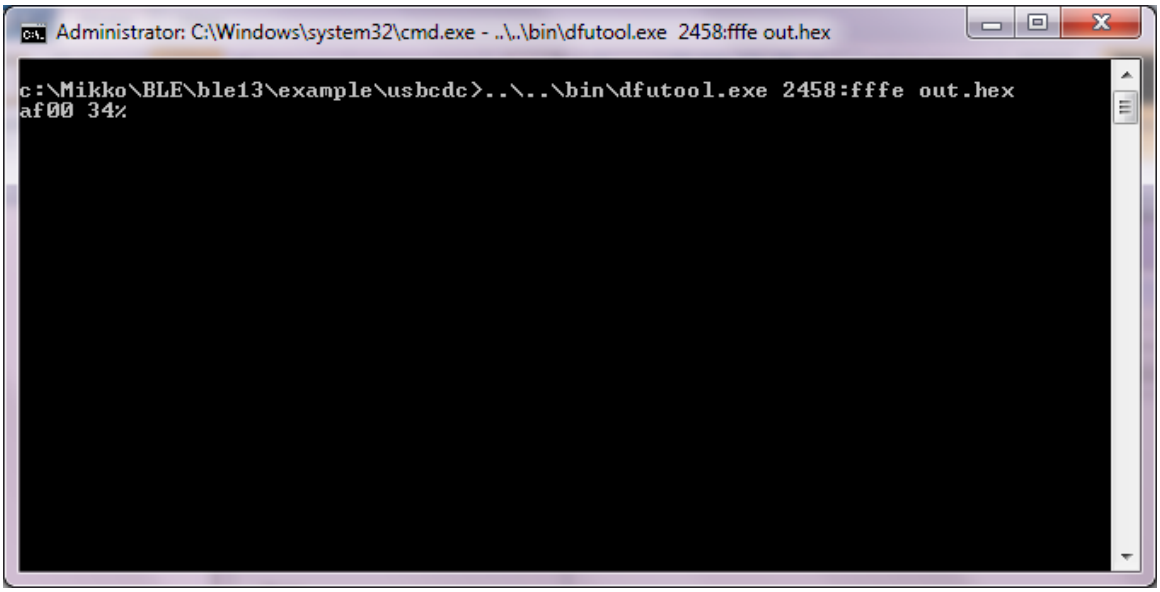


Figure 17: Performing DFU update with DFUTool

Info

Reads the firmware version information

> Always include the output of Info command when contacting Bluegiga support for any software related issues.

Get Address

Reads the *Bluetooth* address of local device.

4.5 Config menu

Config Menu allows the activation or deactivation of the BGAPI protocols packet mode.

Packet mode allows the use of BLE112 over UART interface without flow control by enabling UART DMA and adding a length byte into the beginning of all BGAPI commands.

> Packet mode must be enabled in the hardware configuration file of your Bluetooth Smart device

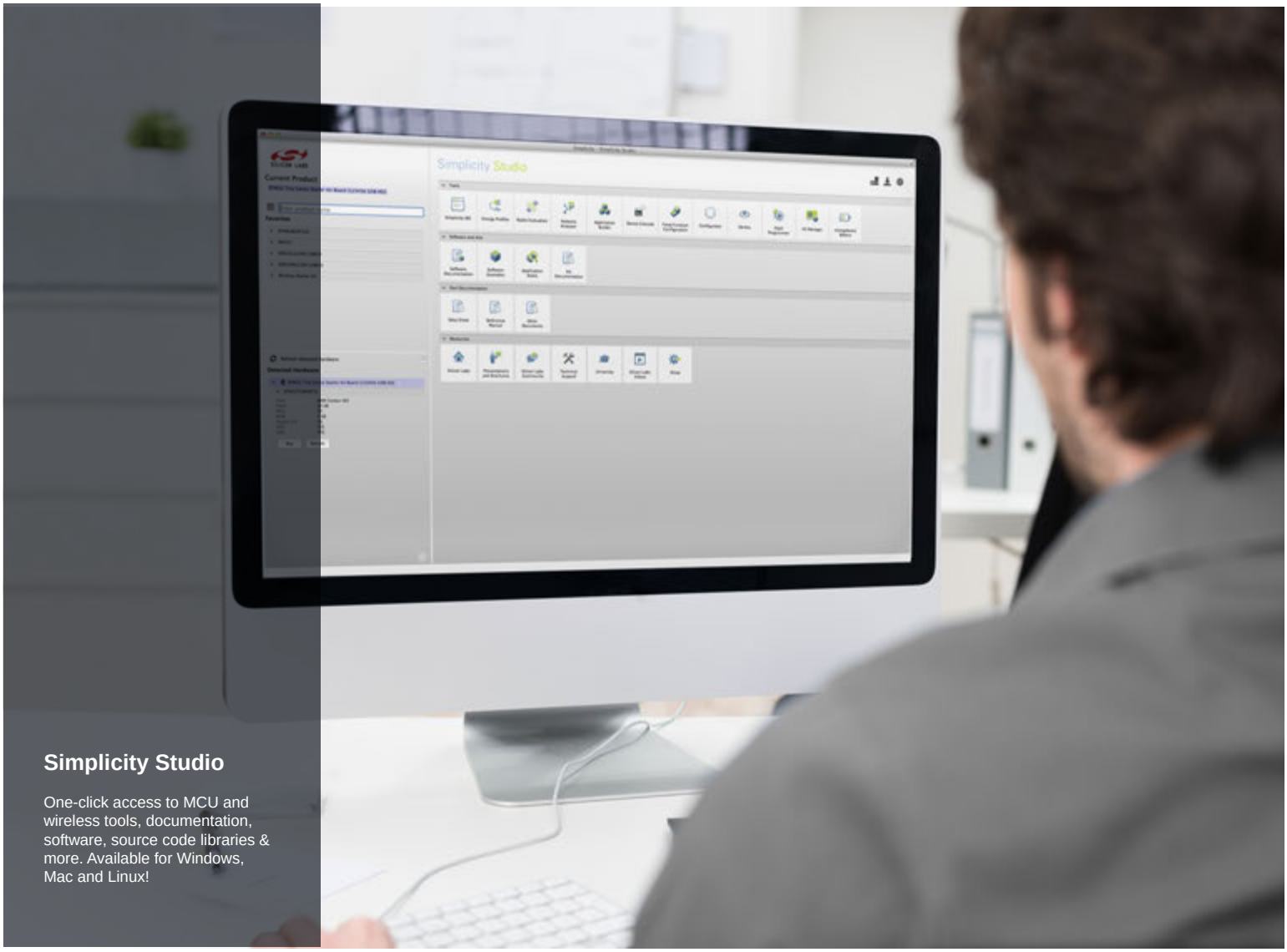
> Please refer to the BGAPI protocols description in the API reference manual for details of the packet mode.

5 Known issues

BLEGUI has the following known issues and problems:

1. Long device names are not parsed properly by BLEGUI and "(no name)" might be shown.

2. BLEGUI does not display secondary services properly. Only primary services are shown.



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS®

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com