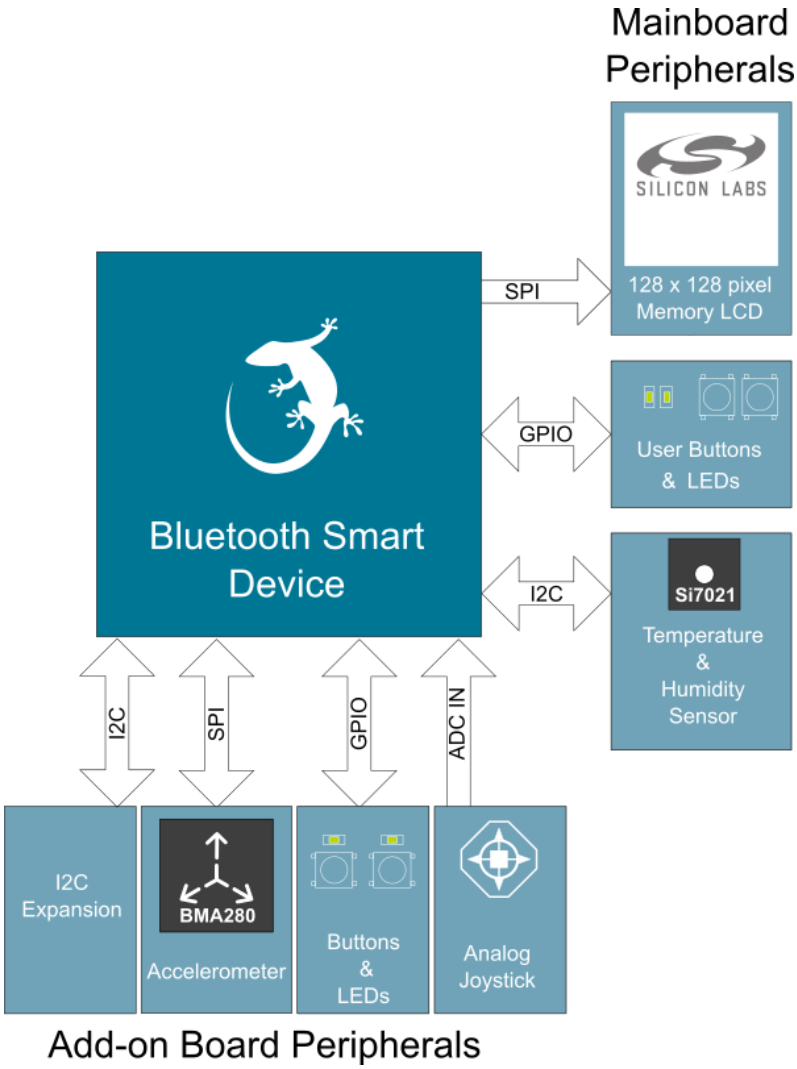
UG119: Blue Gecko *Bluetooth*® Smart Device Configuration User's Guide

This document describes how to start a software project for your Blue Gecko Bluetooth Smart devices, how to include the necessary resources in the project, and also how to configure the hardware interface settings for the Blue Gecko Bluetooth Smart devices.

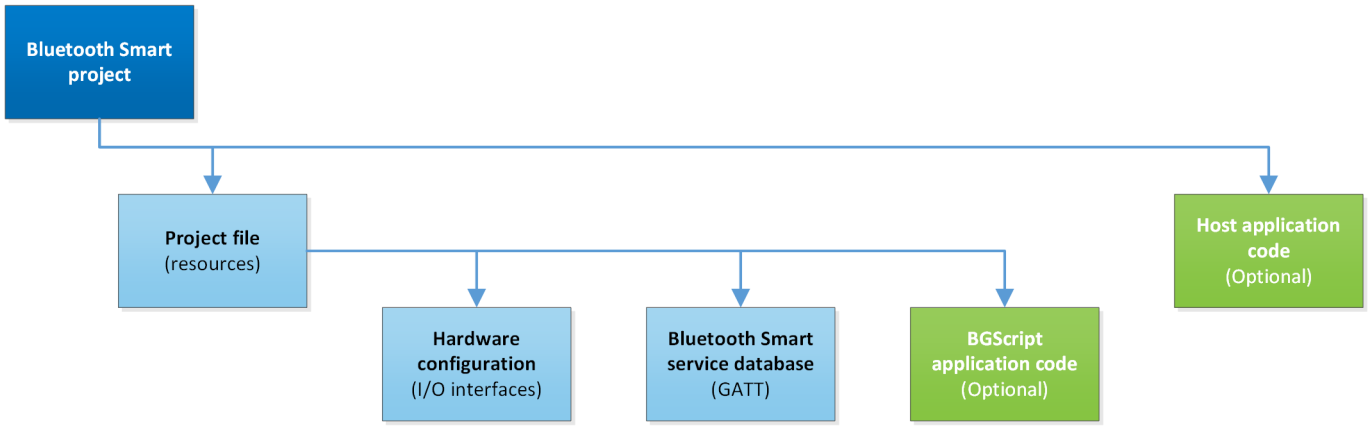
KEY POINTS
- Start a software project
- Include the necessary resources
- Configure hardware interface settings



1. Project Structure

The flowchart below illustrates the Bluetooth Smart software project structure and the mandatory and optional resources required. The structure is relatively simple and consists of the following components:

- Project file
- Hardware configuration file
- Bluetooth Smart service and characteristics database (GATT database)
- BGScript application code (optional and exclusive to host application code)
- Host application code (optional and exclusive to BGScript code)



1.1 Project File

The project file simply defines the resources included in the project and their physical locations.

1.2 Hardware Configuration File

The hardware configuration file defines the host and peripheral interfaces like UART, SPI, I^2C and GPIO used by the application and their physical locations (pins) and settings.

1.3 Bluetooth Smart Service Database

The Bluetooth Smart service database (GATT database) file is an XML based file prepared by the user (e.g. by using XML editor or Bluetooth Developer Studio) and defines the contents and structure of the Bluetooth GATT services and characteristics implemented by the application. The Bluetooth Smart SDK compiles the contents of this XML file and embeds the result into the firmware image.

Guidelines and syntax for defining the GATT database using XML can be found from the *Profile Toolkit User's Guide*.

1.4 BGScript Application Code

BGScript is a BASIC-style event driven application programming language, which allows simple applications to be embedded into Bluetooth Smart devices. In case BGScript is used to implement the application logic, the source files need to be included in the project file.

1.5 Host Application Code

An alternative way to using BGScript based application code is to use an external host (typically an MCU), and use the Bluetooth Smart device as a modem or in so called Network Co-Processor (NCP) mode. In this case the application code runs outside the Bluetooth device and the BGScript source code files do not need to be included in the Bluetooth Smart project When the BGScript code is not defined in the project file the Bluetooth Smart device will be put into the NCP mode.

2. Project File Syntax

The project file (typically **project.xml** or **project.bgproj**) is the file that describes all the components included in your Bluetooth Smart project. Typically these files are named as follows:

- **Hardware.xml** — Hardware configuration file for interfaces like UART, SPI, I^2C and GPIO
- **GATT.xml** — GATT database file for *Bluetooth* Smart services and characteristics
- **Script.bgs** — Optional BGScript application source code

The project file also defines other features of the project like the hardware version, firmware output files, and the selected boot loader. The project file itself is a simple XML file with only a few attributes on it, which are described in the following sections.

2.1 <project>

The XML attribute **<project>** is used to mark the beginning of the definition of a project file and also includes as a parameter the hardware device type the project is intended for. All the other definitions need to be inside the **<project>** and **</project>** tags.

Parameter	Description
device	This parameter and its value define the hardware device type the project is intended for. Options: **bgm111**: Blue Gecko BGM111 Bluetooth Smart Module **bluegecko** : A Blue Gecko Bluetooth Smart SoC
Example: A project file definition for BGM111 Bluetooth Smart module. <project device=**"bgm111"**> … </project>	
Example: A project file definition for EFR32 Bluetooth Smart SoC <project device=**"bluegecko"**> … </project>	

2.2 <hardware>

The XML attribute **<hardware>** defines the location of the hardware configuration file.

Parameter	Description
in	This parameter and its value are used to define the XML file which contains the hardware configuration for your *Bluetooth* Smart device.
Example: Defining the hardware configuration file <hardware in=**"hardware.xml"** />	

2.3 <gatt>

The XML attribute **<gatt>** defines the location of the GATT database definition file.

Parameter	Description
in	This parameter and its value point to the XML file which contains the Bluetooth Smart GATT database definition.
Example: Defining the GATT database definition file <gatt in=**"gatt.xml"** />	

2.4 <script>

The XML attribute **<script>** defines the location of the (optional) BGScript application code file.

Parameter	Description
in	This parameter and its value point to the .BGS file which contains the BGScript application code. This definition is only needed for projects where BGScript applications are used.
Example: Defining the BGScript file <scripting> <script in=**"bgm111demo.bgs"** /> </scripting>	

Note: If you are making a project using NCP mode simply leave the <script> attribute out from the project file.

2.5 <software>

The XML attribute **<software>** defines configuration options for the Bluetooth Smart stack.

Parameter	Description
max_connections value	This parameter defines how many simultaneous BLE connections are allowed by the Bluetooth Smart stack. **Range**: 0 - 8
Example: Setting the maximum number of connections to 8. <software> <max_connections value=**"8"** /> </software>	

2.6 <image>

The XML attribute **<image>** defines the name of the firmware file output by the BGBuild compiler.

Parameter	Description
out	This parameter and its value define the name of the firmware output file.
Example: Naming the firmware output file <image out=**"firmware.bin"** />	

2.7 Project File Examples for BGM111

The example below shows how to create a project file for BGM111 Blue Gecko Bluetooth Smart module when the application code is implemented with BGScript scripting language.

```
<?xml version="1.0" encoding="UTF-8" ?>

<!—Project for BGM111 Bluetooth Smart Module -->
<project device="bgm111">

      <!-- GATT service database -->
      <gatt in="gatt.xml" />

      <!-- Local hardware configuration file -->
      <hardware in="hardware.xml" />

      <!-- BGScript source code -->
      <scripting>
            <script in="bgm111demo.bgs" />
      </scripting>

      <!-- Firmware output file -->
      <image out="bgm111demo.bin" />

</project>
```

Example 1: A BGScript-Based Project

The example below on the other hand shows how to create a project where the application runs on an external host.

```
<?xml version="1.0" encoding="UTF-8" ?>

<project device="bgm111">

      <!-- GATT service database -->
      <gatt in="gatt.xml" />

      <!-- Local hardware configuration file -->
      <hardware in="hardware.xml" />

      <!-- Firmware output file -->
      <image out="WSTK_BGAPI.bin" />

</project>
```

Example 2: A Host-Based Project

The example below shows how to make a project file for EFR32BG and NCP mode.

```
<!-- Project file for EFR32BG -->
<project device="bluegecko">

    <!-- GATT service database -->
    <gatt in="gatt.xml" />

    <!-- Local hardware configuration file -->
    <hardware in="hardware.xml" />

    <!-- Firmware output file -->
    <image out="pcb4100a_wstk_bgapi.bin" />

</project>
```

Figure 2.1. Project example for EFR32BG SoC

3. Hardware Configuration File Syntax

The hardware configuration file (typically named **hardware.xml**) is the file that describes the hardware interface configuration and settings for both the host and the peripheral interfaces.

The hardware configuration file is a simple XML file. The syntax is described in the following subsections.

3.1 <sleep>

The XML attribute **<sleep>** can be used to allow or disallow the use of EM2 sleep mode.

Parameter	Description
enable	This parameter is used to allow or disallow the use of EM2 sleep mode. **true**: EM2 sleep mode is allowed **false**: EM2 sleep mode is not allowed **Default**: false
wake_up	This parameter defines the wake-up pin
Example: Allowing the use of EM2 sleep mode <sleep enable="**true**" />	

3.2 <wake-up>

The XML attribute **<wake-up>** can be used to configure an I/O pin to wake the device up from EM2 mode. This is needed if the device is used in NCP mode and EM2 power save mode is enabled with <**sleep**>.

When the device is in EM2 the host UART will be disabled and the host MUST use the wake-up pin to wake the device up before sending any BGAPI commands to it.

Parameter	Description
port	Defines the port of the wake-up pin **Range**: A - F **Default**: -
pin	Defines the pin of the wake-up pin **Range**: 0 - 15 **Default**: -
state	Defines the I/O state when wake-up occurs **up:** wake-up occurs when pin is high **down**: wake-up occurs when pin is low
Example: enabling EM2 and wake-up pin in PB1 <sleep enable="**true**" /> <wake-up port="**b**" pin="**0**" state="**up**" />	

3.3 <uart>

The XML attribute <uart> and its parameters are used to configure the UART interface settings.

Note: The indicated ranges for ports and pins are for BGM111 and EFR32BG. For other models and variants check the devices datasheet for applicable ranges.

Parameter	Description
index	This parameter is used to select which of the two UARTs is configured. **0:** UART 0 **1:** UART 1
baud	This parameter defines the UART baud rate. **Range**: 600 – 6000000 **Default**: 115200
rx_pin	This parameter defines the UART RX pin location. **Range**: PA0 – PF7 **Default**: PA1
tx_pin	This parameter defines the UART TX pin location. **Range**: PA0 – PF7 **Default**: PA0
rts_pin	This parameter defines the UART RTS pin location. **Range**: PA0 – PF7 **Default**: PA5
cts_pin	This parameter defines the UART CTS pin location. **Range**: PA0 – PF7 **Default**: PA4
flowcontrol	This parameter defines the RTS/CTS flow control state as enabled or disabled. RTS/CTS flow control is recommend to be set to enabled whenever possible to ensure reliable data transmission over the UART interface. **true**: RTS/CTS enabled **false**: RTS/CTS disabled **default**: false
bgapi	This parameter defines whether the BGAPI serial protocol (NCP mode) is enabled or disabled over the UART interface. When an external host is used to control the Bluetooth Smart module over UART, the BGAPI serial protocol must be enabled. When a BGScript application runs in the device the BGAPI serial protocol should be disabled. **true**: BGAPI serial protocol is enabled over UART **false**: BGAPI serial protocol is disabled for UART **default**: false
Example: Enabling UART at 115200 bps, disabling RTS/CTS flow control and disabling BGAPI serial protocol. Default pin mappings are used. (BGScript application has control over UART). <uart index=**"1"** baud=**"115200"** flowcontrol=**"false"** bgapi=**"false"**/>	
Example: Enabling UART at 460800 bps, enabling RTS/CTS flow control and enabling BGAPI serial protocol. Default pin mappings are used. <uart index=**"1"** baud=**"460800"** flowcontrol=**"true"** bgapi=**"true"**/>	

3.4 <i²c>

The XML attribute **<i2c>** and its parameters are used to configure the I²C interface settings.

Note: The indicated ranges for ports and pins are for BGM111 and EFR32BG. For other models and variants check the devices data-sheet for applicable ranges.

Parameter	Description
scl_pin	This parameter defines the I²C SCL pin location **Range**: PA0 – PF7 **Default**: PC11
sda_pin	This parameter defines the I²C SDA pin location **Range**: PA0 – PF7 **Default**: PC10
Example: Enabling I²C into pins PC10 and PC11. Note that on the WSTK main board the RHT sensor uses these pins. <i2c scl_pin=**"PC11"** sda_pin=**"PC10"**/>	

3.5 <gpio>

The XML attribute **<gpio>** and its parameters are used to configure the GPIO pin settings.

Note: The indicated ranges for ports and pins are for BGM111 and EFR32BG. For other models and variants check the devices data-sheet for applicable ranges.

Parameter	Description
port	This parameter is used to select the port to configure. **Range:** A – F
pin	This parameter is used to select the pin within the defined port to configure. **Range:** 0 – 15
out	This parameter configures the default value of the data out register (DOUT) **Values:** **0:** DOUT register value is 0 **1:** DOUT register value is 1 **Default:** 0
mode	This parameter is used to configure the GPIO mode of the pin to configure. **Values:** **DISABLED** Input disabled. Pull-up if DOUT is set. **INPUT** Input enabled. Filter if DOUT is set. **INPUTPULL** Input enabled. DOUT determines pull direction. **INPUTPULLFILTER** Input enabled with filter. DOUT determines pull direction. **PUSHPULL** Push-pull output. **PUSHPULLALT** Push-pull using alternate control. **WIREDOR** Wired-or output. **WIREDORPULLDOWN** Wired-or output with pull-down. **WIREDAND** Open-drain output. **WIREDANDFILTER** Open-drain output with filter. **WIREDANDPULLUP** Open-drain output with pull-up. **WIREDANDPULLUPFILTER** Open-drain output with filter and pull-up. **WIREDANDALT** Open-drain output using alternate control. **WIREDANDALTFILTER** Open-drain output using alternate control with filter. **WIREDANDALTPULLUP** Open-drain output using alternate control with pull-up. **WIREDANDALTPULLUPFILTER**Open-drain output using alternate control with filter and pull-up.

Parameter	Description
interrupt	This parameter is used to enable or disable interrupts and select rising, falling or both edge triggering of the pin to configure. **Values**: **none**: Interrupts are disabled. **rising**: Interrupts generated on rising edge. **falling**: Interrupts generated on falling edge. **both**: Interrupts generated on both rising and falling edges. **Default**: none
The examples below apply for BGM111 and WSTK main board. **Example**: Enabling the built-in UART (UART-to-USB bridge) on the WSTK main board requires that PA5 is configured as an output and PA3 as an input. The following GPIO configuration must be used with WSTK if UART functionality is needed. <gpio port=**"A"** pin=**"6"** mode=**"pushpull"** out=**"1"**/> <gpio port=**"A"** pin=**"3"** mode=**"pushpull"** out=**"0"**/>	
Example: Configuring both LED0 (PF6) and LED1 (PF7) on the WSTK main board with default state 0, which means the LED is on (active low). <!-- LED0 --> <gpio port=**"F"** pin=**"6"** mode=**"pushpull"** out=**"0"**/> <!-- LED1 --> <gpio port=**"F"** pin=**"7"** mode=**"pushpull"** out=**"0"**/>	

Pin configuration

Table 3.1. Pin Configuration

MODEn	Input	Output	DOUT	Pull-down	Pull-up	Alt. strength	Input Filter	Description
DISABLED	Disabled	Disabled	0					Input disabled
			1		On			Input disabled with pull-up
INPUT	Enabled		0					Input enabled
			1				On	Input enabled with filter
INPUTPULL			0	On				Input enabled with pull-down
			1		On			Input enabled with pull-up
INPUTPULLFILTER			0	On			On	Input enabled with pull-down and filter
			1		On		On	Input enabled with pull-up and filter
PUSHPULL		Push-pull	x					Push-pull
PUSHPULLALT			x			On		Push-pull with alt. drive strength
WIREDOR		Open Source (Wired-OR)	x					Open-source
WIREDORPULLDOWN			x	On				Open-source with pull-down
WIREDAND		Open Drain (Wired-AND)	x					Open-drain
WIREDANDFILTER			x				On	Open-drain with filter
WIREDANDPULLUP			x		On			Open-drain with pull-up
WIREDANDPULLUPFILTER			x		On		On	Open-drain with pull-up and filter
WIREDANDALT			x			On		Open-drain with alt. drive strength
WIREDANDALTFILTER			x			On	On	Open-drain with alt. drive strength and filter
WIREDANDALTPULLUP			x		On	On		Open-drain with alt. drive strength and pull-up
WIREDANDALTPULLUPFILTER			x		On	On	On	Open-drain with alt. drive strength, pull-up and filter

3.6 <adc>

The XML attribute **<adc>** and its parameters are used for configuring the ADC settings.

Note: Please refer to your Bluetooth devices data sheet for available ADC ports and pins which may vary depending on the model variant.

Parameter	Description
enable	This parameter is used to enable or disable the ADC functionality. **Values:** **true:** ADC is enabled **false:** ADC is disabled **Default:** false
Example: Enabling ADC <!-- ADC enabled --> <adc enable=**"true"**	

3.7 <ctune>

The XML attribute **<ctune>** can be used to configure the crystal (XTAL) frequency tuning value.

The crystal tuning value must be set correctly for every design, which is especially important when making your own EFR32BG based hardware.

Parameter	Description
value	This parameter value is used to configure the crystal tune value. **Range**: 0 - 511 **BGM111A256V1**: 0 **BGM111A256V2**: 322 **EFR32BG radio board**: 325
Example: Setting crystal tune value for EFR32BG Radio Board <ctune **value** ="325" />	

3.8 <lfxo>

The XML attribute **<lfxo>** can be used to define if the Low Frequency Crystal (LFXO) is present in the system or not.

All Blue Gecko modules have the LFXO in them so this must be set to true (default value), but with the Blue Gecko SoC designs you can decide not to use the LFXO to optimize BoM cost. Notice however that EM2 or more aggressive sleep modes are not available in LFXO is not used.

Parameter	Description
enable	This parameter is used to define if LFXO is present or not. **true**: LFXO is present **false**: LFXO is not present **Default**: true
Example: Disabling LXFO <lfxo enable="**false**" />	

3.9 <dcdc>

The XML attribute **<dcdc>** is used to enable or disable the dc/dc bypass mode.

Note: The data below applies for BGM111. For other device types consult the device's datasheet.

When DC/DC is enabled, supply voltage range is 2.4 to 3.8 V.

When DC/DC is disabled, supply voltage range is 1.8 to 3.8 V.

When DC/DC is disabled the RX sensitivity improves by 1 dB.

Parameter	Description
enable	This parameter is used to enable or disable the built-in dc/dc converter. **Values:** **true:** DC/DC is enabled **false:** DC/DC is disabled **Default**: enabled
Example: Enabling the built-in dc/dc <!-- Enable DC/DC converter --> <dcdc enable=**"true"** />	
Example: Disabling the built-in dc/dc <!-- Disable DC/DC converter --> <dcdc enable=**"false"** />	

3.10 <pti>

The XML attribute **<pti>** can be used to enable or disable the packet trace feature on the Blue Gecko Bluetooth Smart hardware. Packet trace outputs the Bluetooth packets sent and received by the Blue Gecko hardware via a special hardware interface and they can be viewed in the Simplicity Studio's Network Analyzer tool. Packet trace feature is useful for debugging the Bluetooth smart communications and error situations.

Parameter	Description
enable	This parameter defines if packet trace feature is enabled or disabled. **true**: Packet Trace is enabled **false**: Packet Trace is enabled **Default**: false
mode	This parameter configures the PTI format **uart**: UART format: TX + frame **spi**: SPI format: Data + Clock + Frame **onewire**: one wire format: Data only **Default**: uart
baud	This parameter configures the PTI baud rate in Hz. **Range** : 9600 - 2000000 **Default**: 1600000
data	This parameter selects the data output pin **Range**: PA0 - PA7 **Default**: PA4
clock	This parameter selects the clock output pin **Range**: PA0 - PA7 **Default**: PB11
frame	This parameter selects the frame output pin **Range**: PA0 - PA7 **Default**: PB13
Example: Enabling packet trace with default settings <pti enable="**true**" />	

3.11 Hardware Configuration File Examples

The example below shows how to create a hardware configuration file for a Blue Gecko Bluetooth Smart device, where the application is implemented with BGScript scripting language and both UART and I²C interfaces are enabled and controlled by the script application.

```
<?xml version="1.0" encoding="UTF-8" ?>

<hardware>

      <!-- UART configuration -->
      <!-- Settings: @115200bps, no RTS/CTS and BGAPI serial protocol is
      disabled -->
      <uart index="1" baud="115200" flowcontrol="false" bgapi="false"/>

      <!-- I2C configuration -->
      <!-- Settings: SCL pin PC11 and SDA pin PC10-->
      <i2c scl_pin="PC11" sda_pin="PC10"/>

</hardware>
```

Example 3: Hardware Configuration with both UART and I²C Enabled

```
<?xml version="1.0" encoding="UTF-8" ?>

<hardware>

      <!-- UART enabled @115200bps, RTS/CTS disabled, BGAPI enabled -->
      <uart index="1" baud="115200" flowcontrol="false" bgapi="true"/>

      <!-- Enable UART on WSTK-->
      <!-- PA5 as input, PA3 as output -->
      <gpio port="A" pin="5" mode="pushpull" out="1"/>
      <gpio port="A" pin="3" mode="pushpull" out="0"/>

</hardware>
```

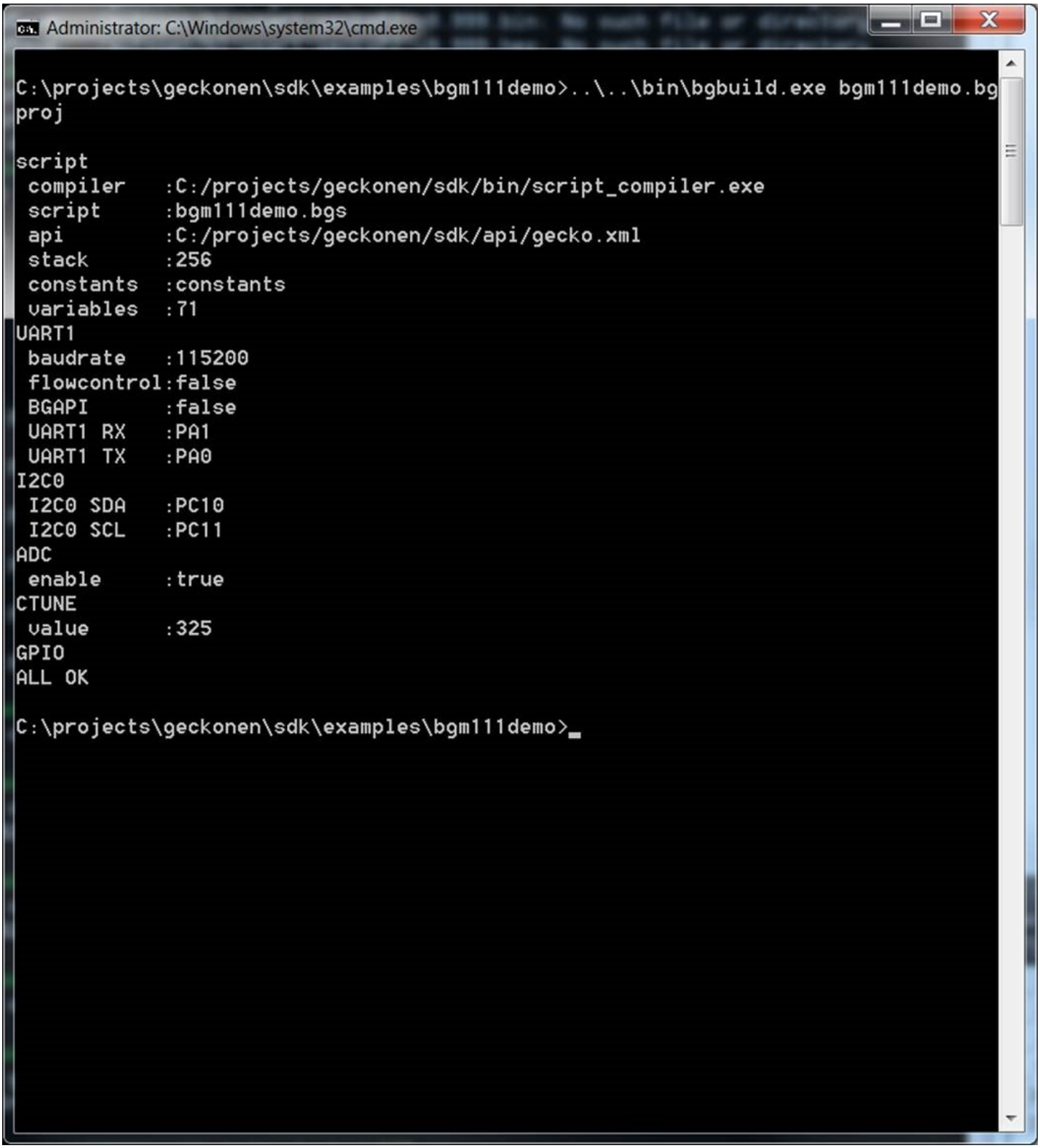
Example 4: A Host-Based Project with BGAPI Serial Protocol Enabled

4. Verifying Configuration from BGBuild Compiler Output

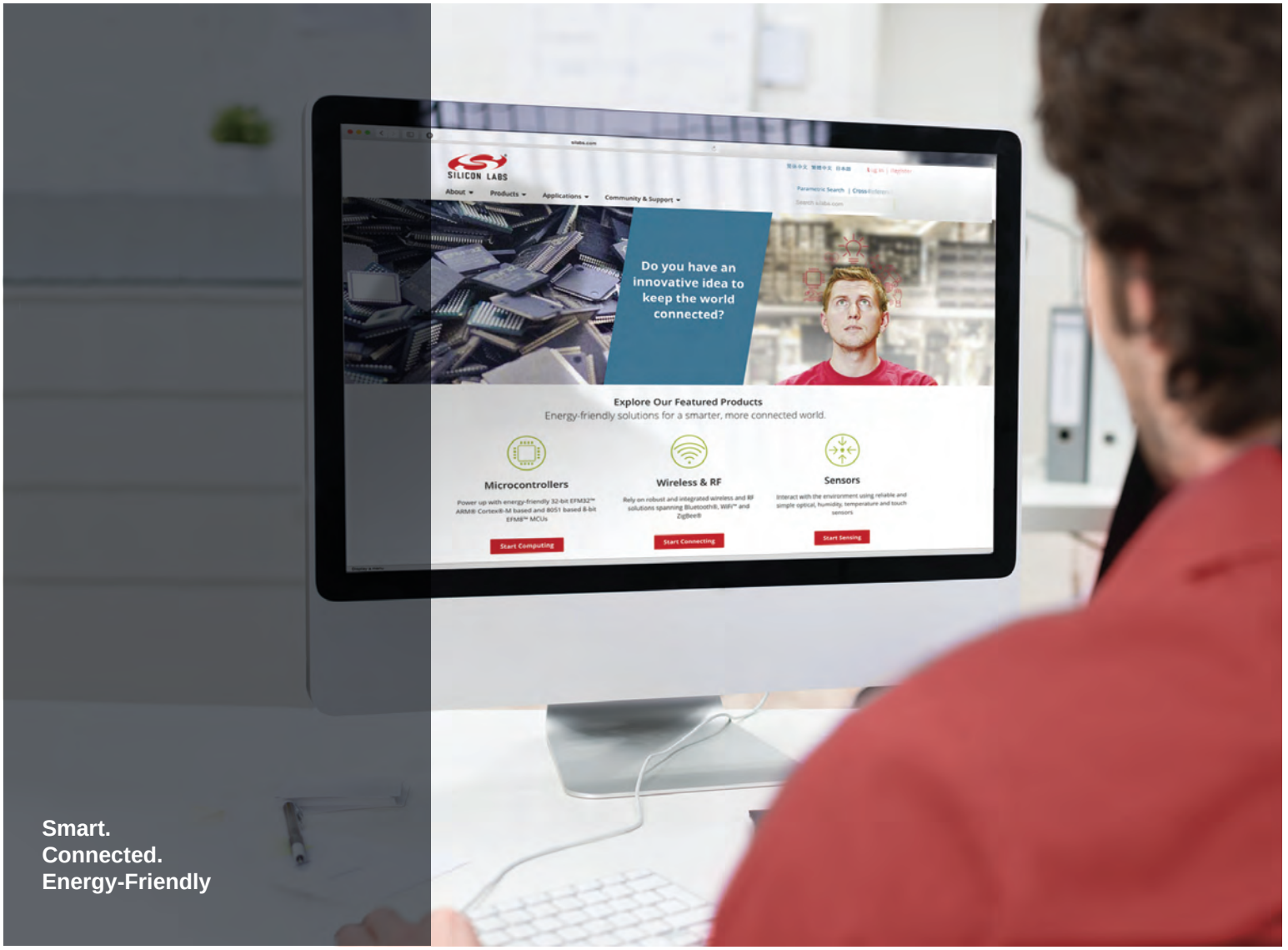
When you compile the project file with the BGBuild compiler it will output the project configuration, BGScript, BGAPI and hardware interface settings, so you can verify the correctness of your configuration files.

An example of a compiler output is shown in the image below.

Figure 4.1. BGBuild Compiler Output



Output	Description
compiler	Location of the BGBuild compiler used in the compilation.
script	Main BGScript source code file. This is displayed ONLY if BGScript is defined in the project configuration file.
api	Location of the used API definition file.
stack	Available stack allocated for BGScript applications.
constants	Name of the file which contains the IDs and attribute handles defined in the GATT database .
variables	Indicates the amount of memory reserved by variables.
UART0 ***or*** ***UART1***	UART interface settings **baudrate:**UART baud rate **flowcontrol:**RTS/CTS flow control configuration **BGAPI**: BGAPI serial protocol configuration **UART RX:**UART RX pin **UART TX:**UART TX pin **UART CTS:**UART CTS pin **UART RTX:**UART RTS pin
I2C0	I2C interface settings **I2C0 SDA:**I2C SDA pin **I2C0 SCL:**I2C SCL pin
ADC	Indicates whether the device's ADC is enabled or disabled.
CTUNE	Configured crystal tune value.
GPIO	GPIO configuration and pin assignment check result.
SLEEP	Indicates whether EM2 (sleep mode) is allowed or disallowed. **TBD** and **MISSING FROM SCREENSHOT**



Products
www.silabs.com/products

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