ECEN 5823-001 / -001B

Internet of Things Embedded Firmware

Lecture #13

10 October 2017

Be Boulder.

University of Colorado **Boulder**

Agenda

- BLE Health Temperature Service Assignment demo
- Class announcements
- Quiz 6 review
- BLE Health Temperature Service Assignment
- Setting up a BLE Service
- Bluetooth Smart

Class Announcements

- Quiz #7 is due at 11:59pm on Sunday, October 15th, 2017
- BLE Health Temperature Service assignment is due at 11:59pm on Sunday, October 22nd, 2017
- Mid-term will be held in class on Thursday, October 19th, at 6:30pm in class
  - For on campus students, you must be in class for the exam
  - For distant learners, the mid-term will be due by 11:59pm on Saturday, October 21st, 2017
- There will be no homework assignment or quiz the week of October 16th

Mid-Term

- October 19th, 2017
  - For the distant learners, the Mid-Term will be available from October 19th at 6:30pm to Saturday the 21st at 11:59pm
- Will be administered by D2L
  - 75 minute time limit for the Mid-term
  - 5 minutes time limit for the bonus section
  - 1 attempt
- Open book, but not open people, not google

Mid-Term

- Material covered will include:
  - All the readings from the first day of class
  - All the lectures through Thursday, October 12th, 2017
  - All assignments
- Questions:
  - 33 questions that will represent 100% of the mid-term
    - Question pool will be over 100 questions
  - 10 bonus questions each worth 1 point
    - Comprised of a random selection from the first 7 week quiz questions (roughly 150 questions in the question library)

BLE: Frequency Hopping

- When in data connection, a frequency-hopping algorithm is used. Since there are 37 data channels which is a prime number, the hopping sequence is very simple
  - f_{n+1} = (f_n+hop) mod 37
  - The hop value can range from 5 to 16
  - This will result in every frequency be used with equal priority
- Notice, that the advertising channel numbers are greater than 37, so they will never be used in the data connection hop sequence

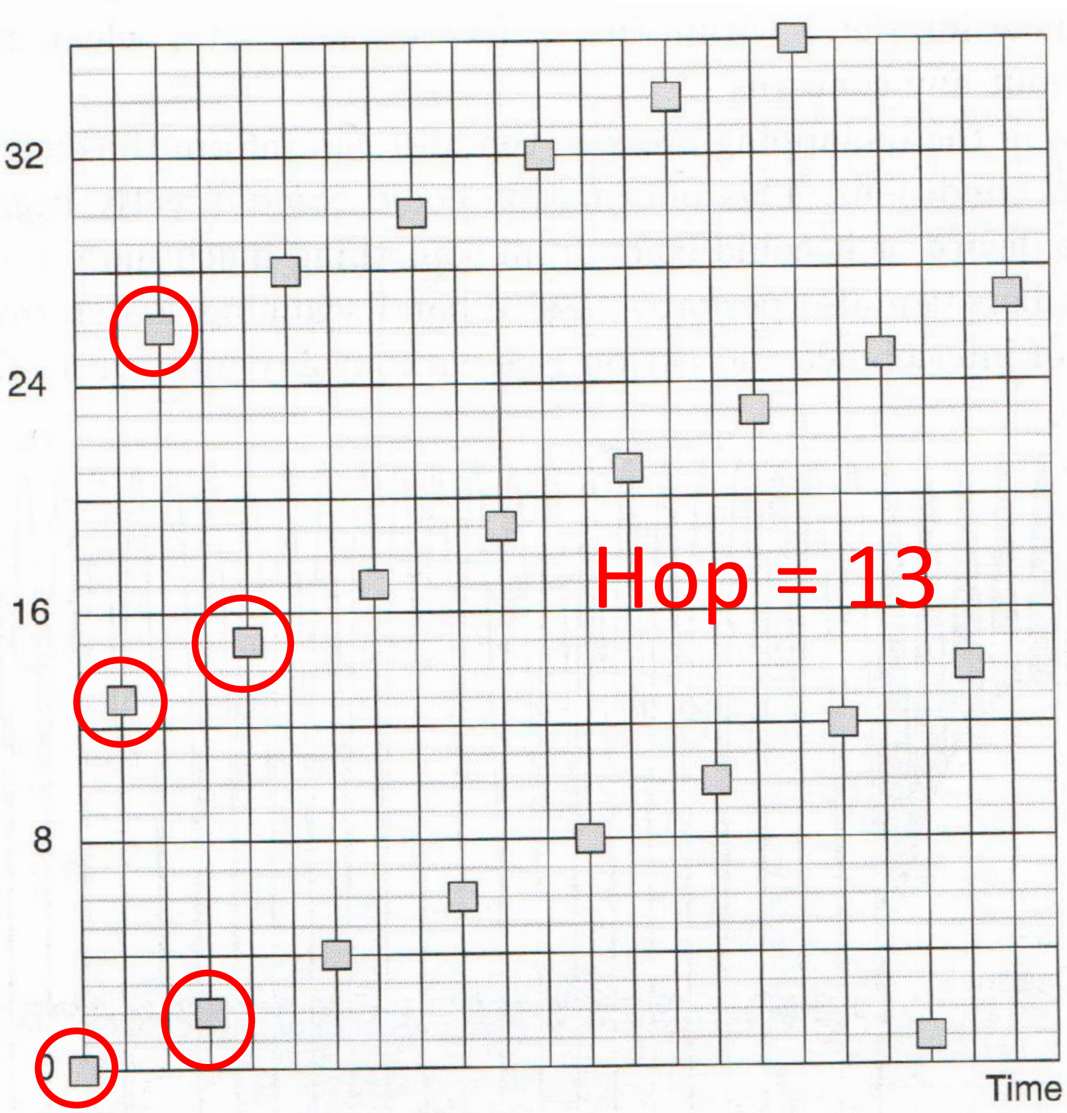


Figure 7–14 Frequency hopping of data channels over time

BLE: Adaptive Frequency Hopping Managing Interference and Coexistence

- WiFi access point typically use one of three 802.11 channels
- BLE Advertising channels are strategically placed to not be interfered by these WiFi channels (1, 6, and 11)
- Three advertising channels are designed into the BLE specification to provide robustness
- Without an effective advertising channel, BLE would not be an effective wireless network

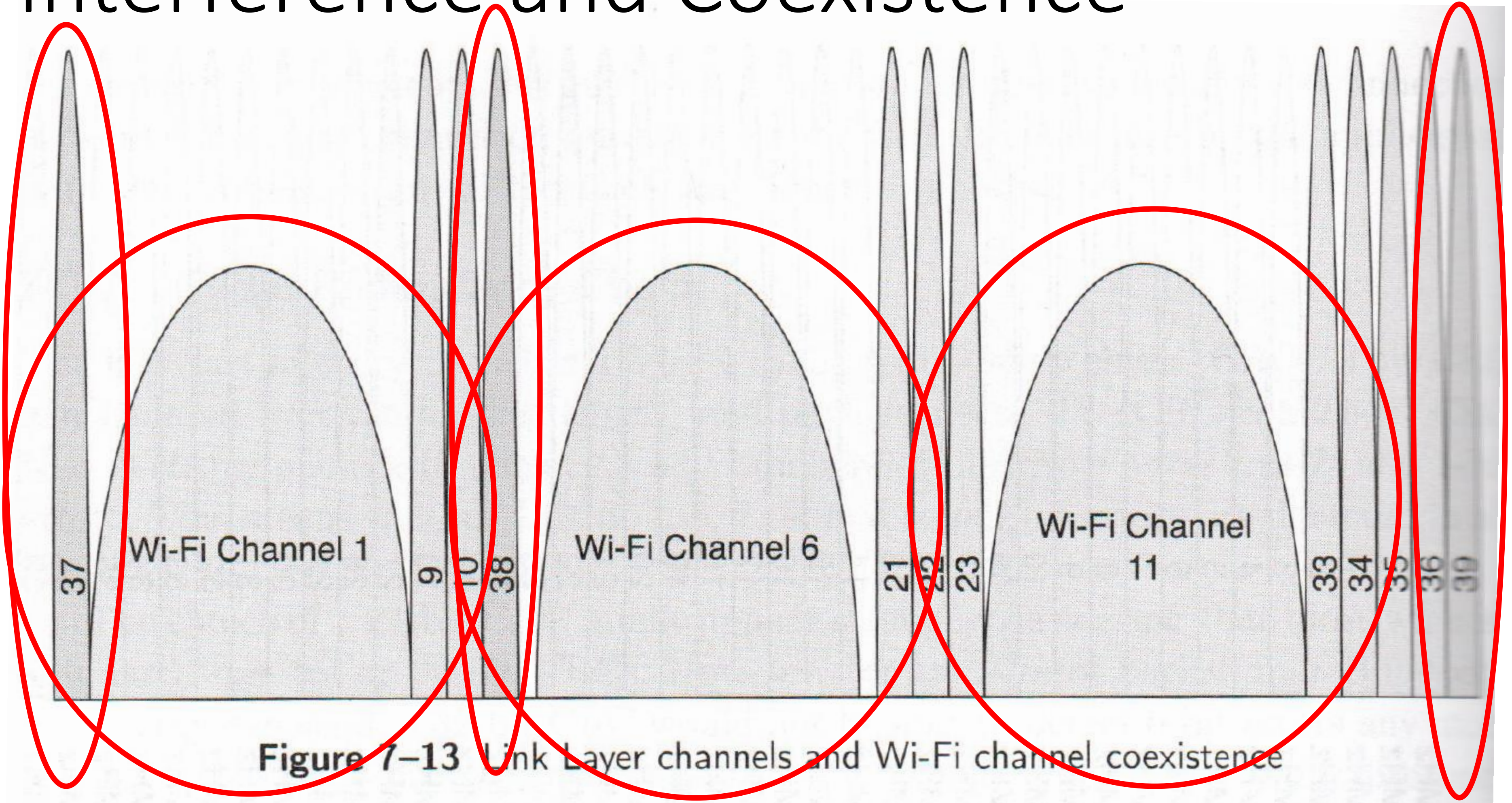
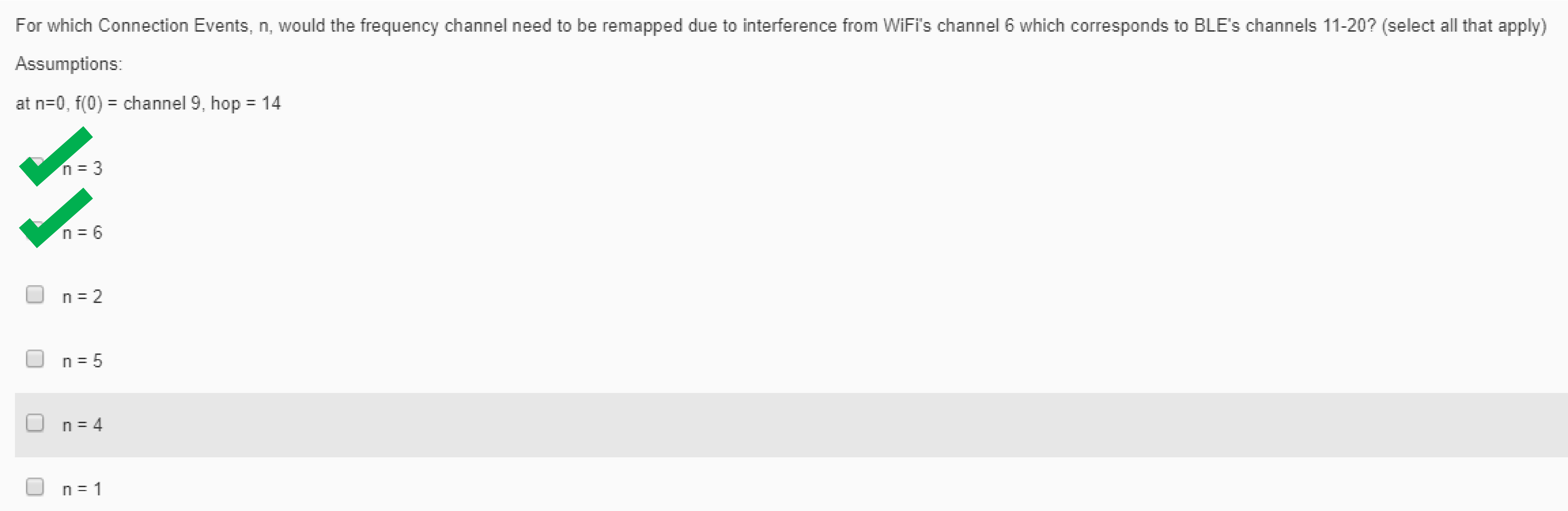


Figure 7–13 Link Layer channels and Wi-Fi channel coexistence

Quiz 6 review



For which Connection Events, n, would the frequency channel need to be remapped due to interference from WiFi's channel 6 which corresponds to BLE's channels 11-20? (select all that apply)

Assumptions:

at n=0, f(0) = channel 9, hop = 14

☑ n = 3

☑ n = 6

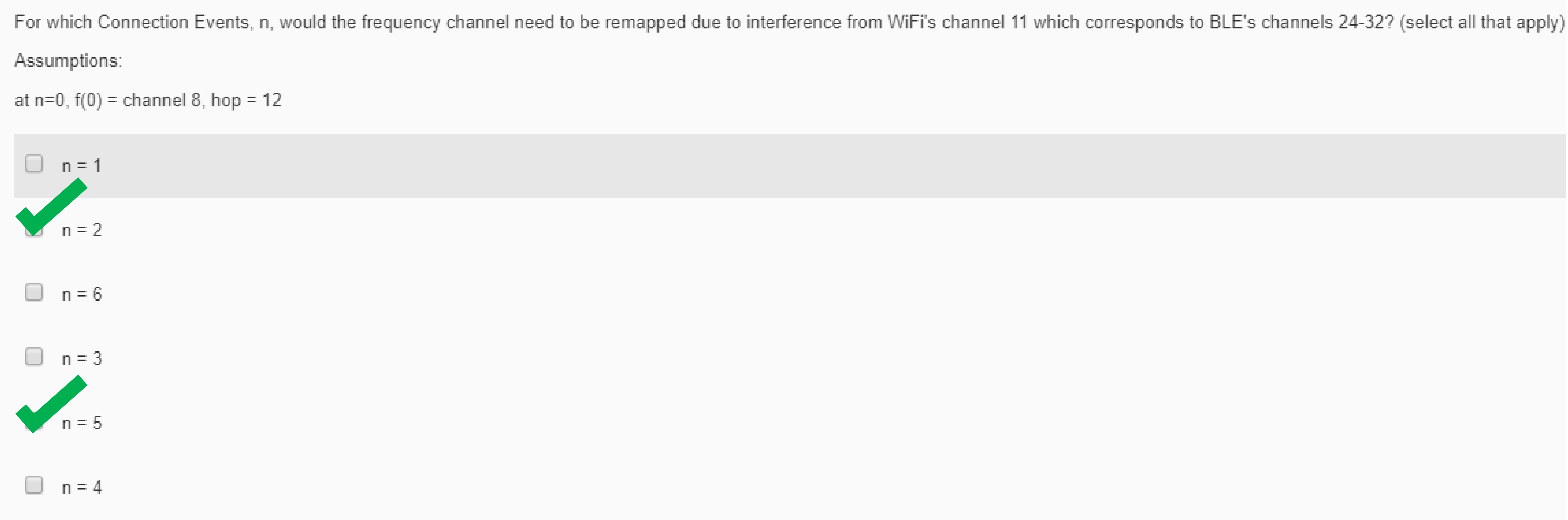
□ n = 2

□ n = 5

□ n = 4

□ n = 1

Quiz 6 review



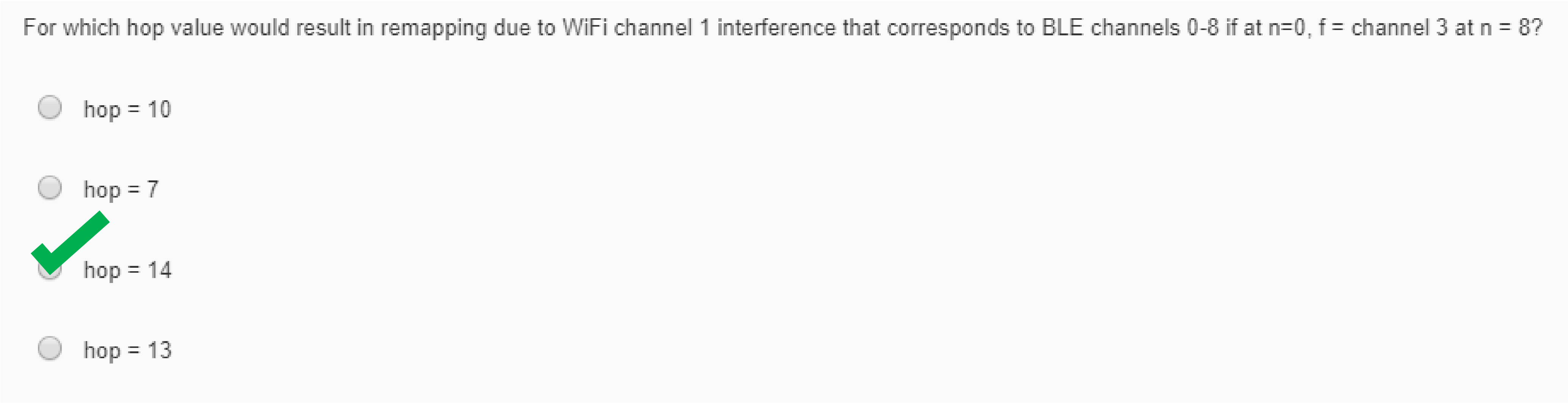
For which Connection Events, n, would the frequency channel need to be remapped due to interference from WiFi's channel 11 which corresponds to BLE's channels 24-32? (select all that apply)

Assumptions:

at n=0, f(0) = channel 8, hop = 12

- □ n = 1
- ✓ n = 2
- □ n = 6
- □ n = 3
- ✓ n = 5
- □ n = 4

Quiz 6 review



For which hop value would result in remapping due to WiFi channel 1 interference that corresponds to BLE channels 0-8 if at n=0, f = channel 3 at n = 8?

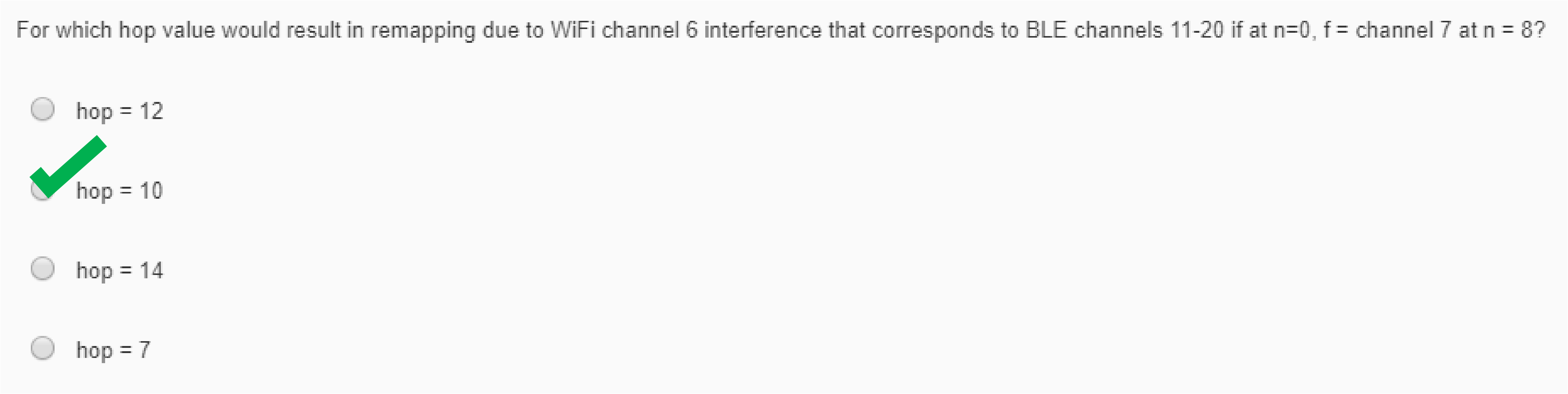
○ hop = 10

○ hop = 7

○ [✓ hop = 14]

○ hop = 13

Quiz 6 review



For which hop value would result in remapping due to WiFi channel 6 interference that corresponds to BLE channels 11-20 if at n=0, f = channel 7 at n = 8?

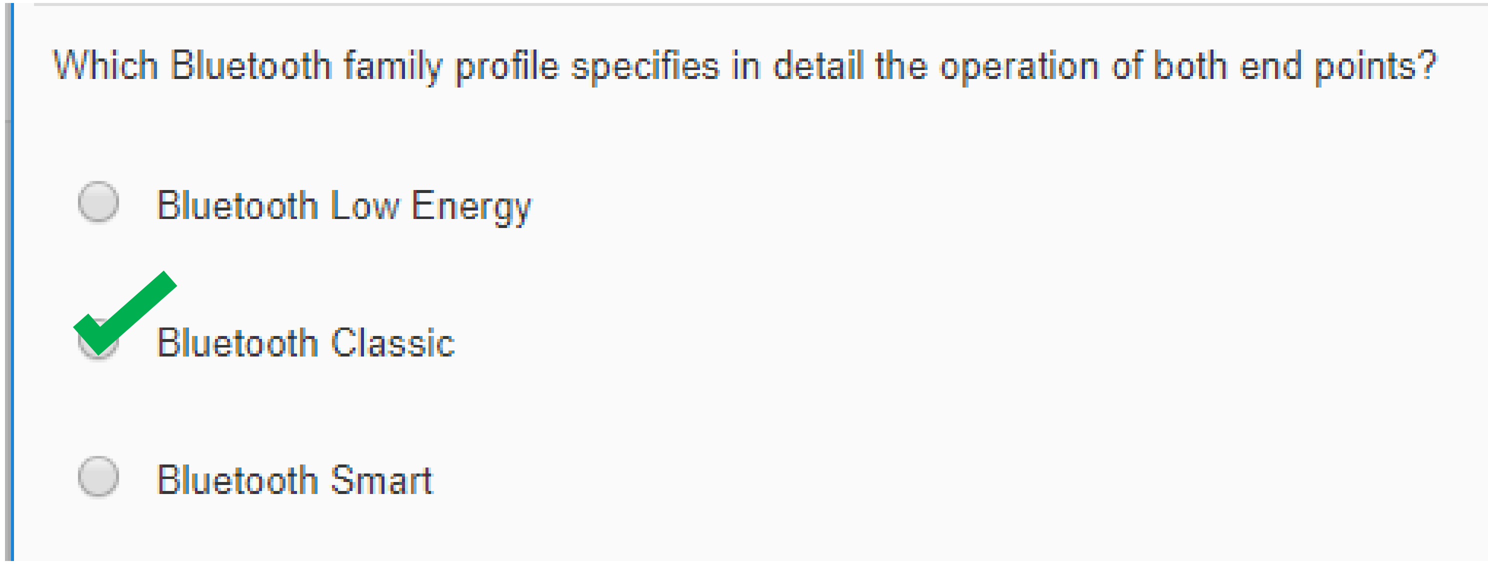
○ hop = 12

○ [✓ hop = 10]

○ hop = 14

○ hop = 7

Quiz 6 review



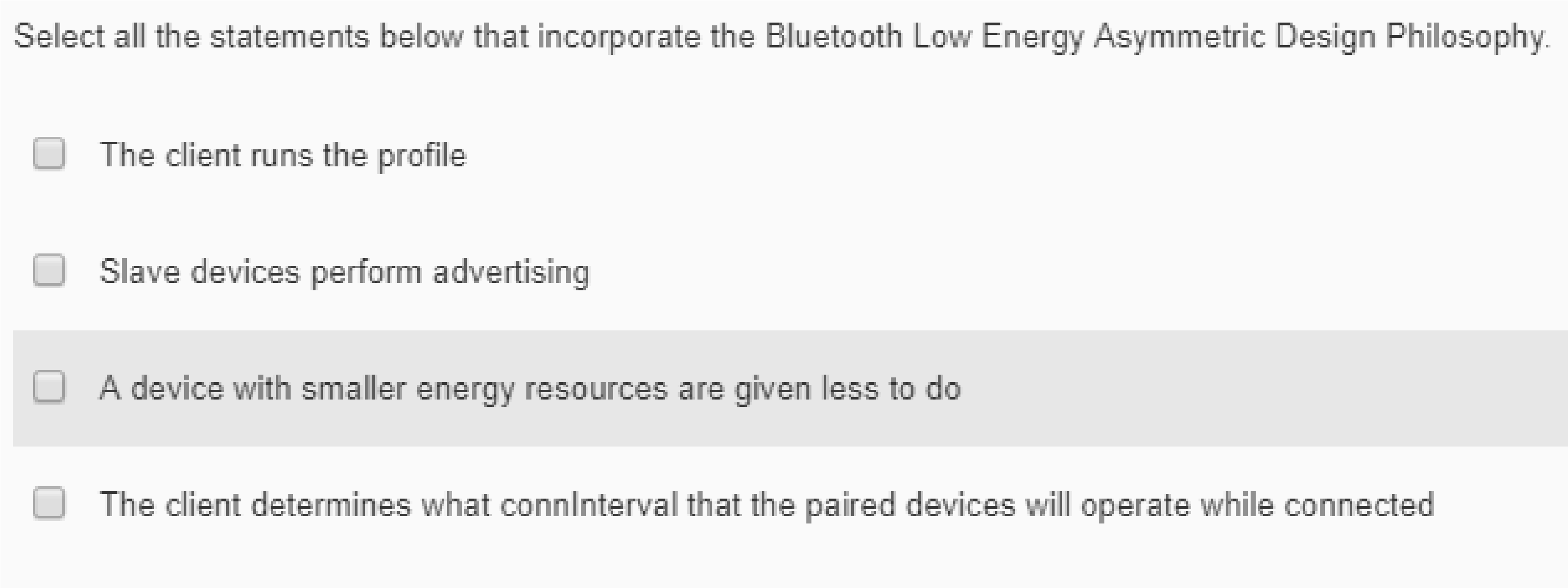
Which Bluetooth family profile specifies in detail the operation of both end points?

○ Bluetooth Low Energy

✓ Bluetooth Classic

○ Bluetooth Smart

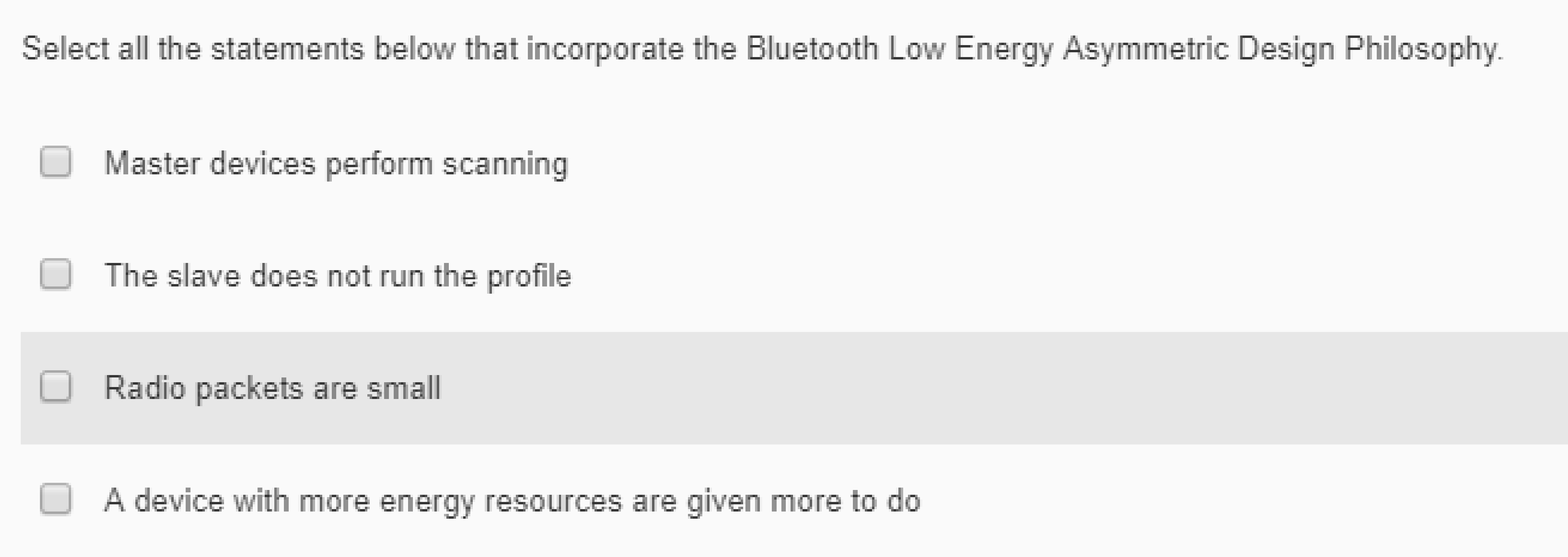
Quiz 6 review



Select all the statements below that incorporate the Bluetooth Low Energy Asymmetric Design Philosophy.

- □ The client runs the profile
- □ Slave devices perform advertising
- □ A device with smaller energy resources are given less to do
- □ The client determines what connInterval that the paired devices will operate while connected

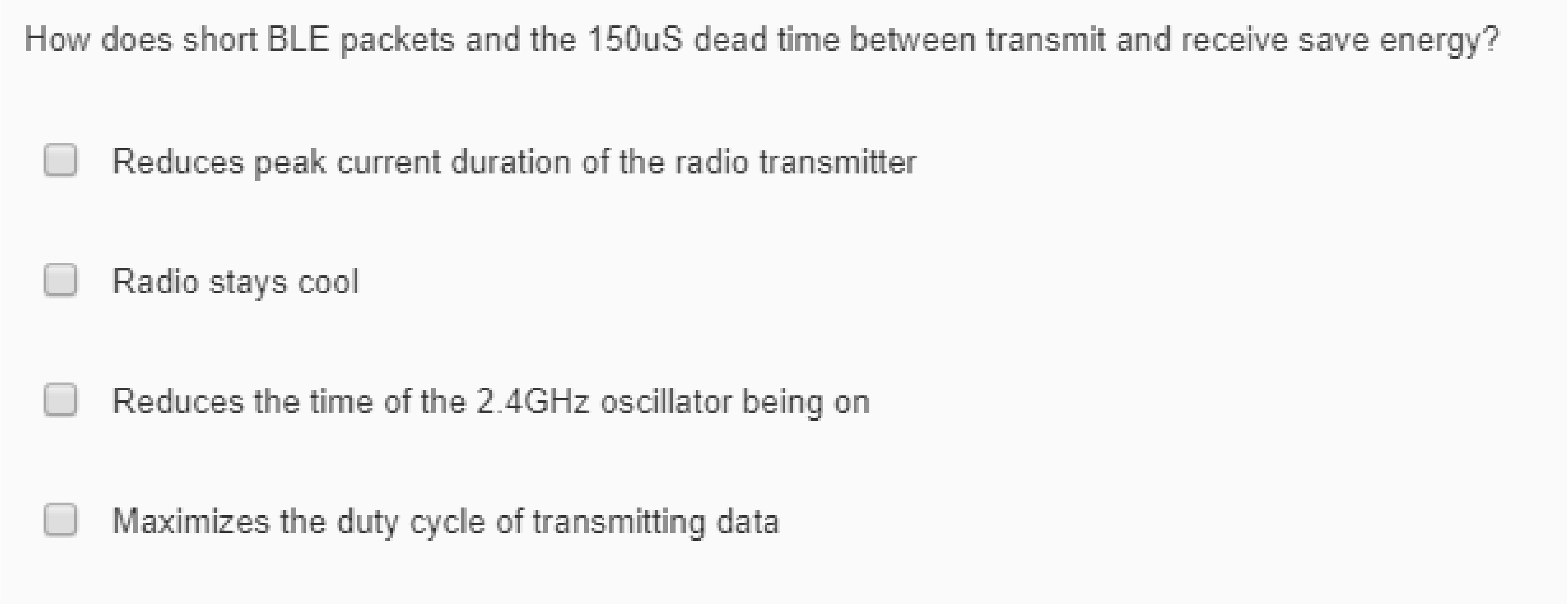
Quiz 6 review



Select all the statements below that incorporate the Bluetooth Low Energy Asymmetric Design Philosophy.

- □ Master devices perform scanning
- □ The slave does not run the profile
- □ Radio packets are small
- □ A device with more energy resources are given more to do

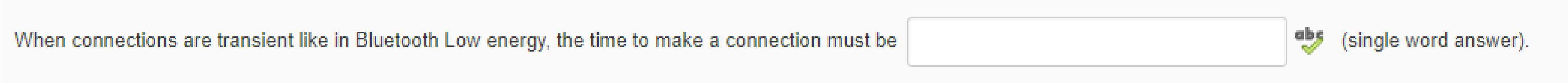
Quiz 6 review



How does short BLE packets and the 150uS dead time between transmit and receive save energy?

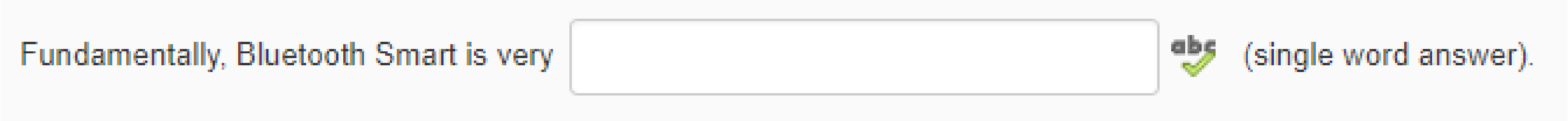
- □ Reduces peak current duration of the radio transmitter
- □ Radio stays cool
- □ Reduces the time of the 2.4GHz oscillator being on
- □ Maximizes the duty cycle of transmitting data

Quiz 6 review



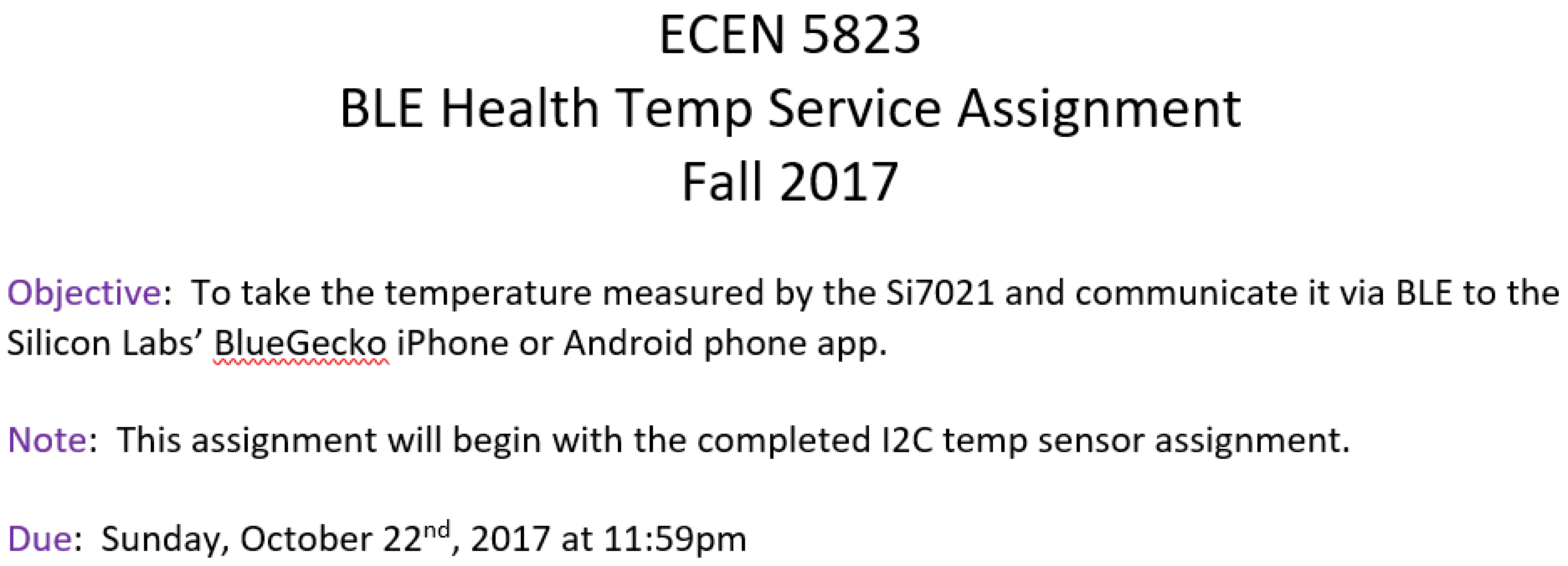
When connections are transient like in Bluetooth Low energy, the time to make a connection must be ______ (single word answer).

Quiz 6 review



Fundamentally, Bluetooth Smart is very ______ (single word answer).

Bluetooth Health Temp Service Assignment



ECEN 5823

BLE Health Temp Service Assignment

Fall 2017

Objective: To take the temperature measured by the Si7021 and communicate it via BLE to the Silicon Labs' BlueGecko iPhone or Android phone app.

Note: This assignment will begin with the completed I2C temp sensor assignment.

Due: Sunday, October 22nd, 2017 at 11:59pm

Using the Bluetooth Stack on the Blue Gecko

- What drives Low Energy Firmware?

- It saves energy by allowing which peripheral to be turned off as much as possible?

Using the Bluetooth Stack on the Blue Gecko

- With Bluetooth being a long energy radio protocol, its code is based on ….

Bluetooth Events

- Enabling the CPU to be off as much as possible with the following ….

evt = **gecko_wait_event();**

Using the Bluetooth Stack on the Blue Gecko

- evt = **gecko_wait_event();**
  - The Bluetooth stack must control the level of sleep to enable it to wake up on schedule to handle Bluetooth events
  - It sets a timer and only returns if a Bluetooth stack event has occurred
    - Similar to the Enter_Sleep() routine which only returns when an interrupt has occured

Using the Bluetooth Stack on the Blue Gecko

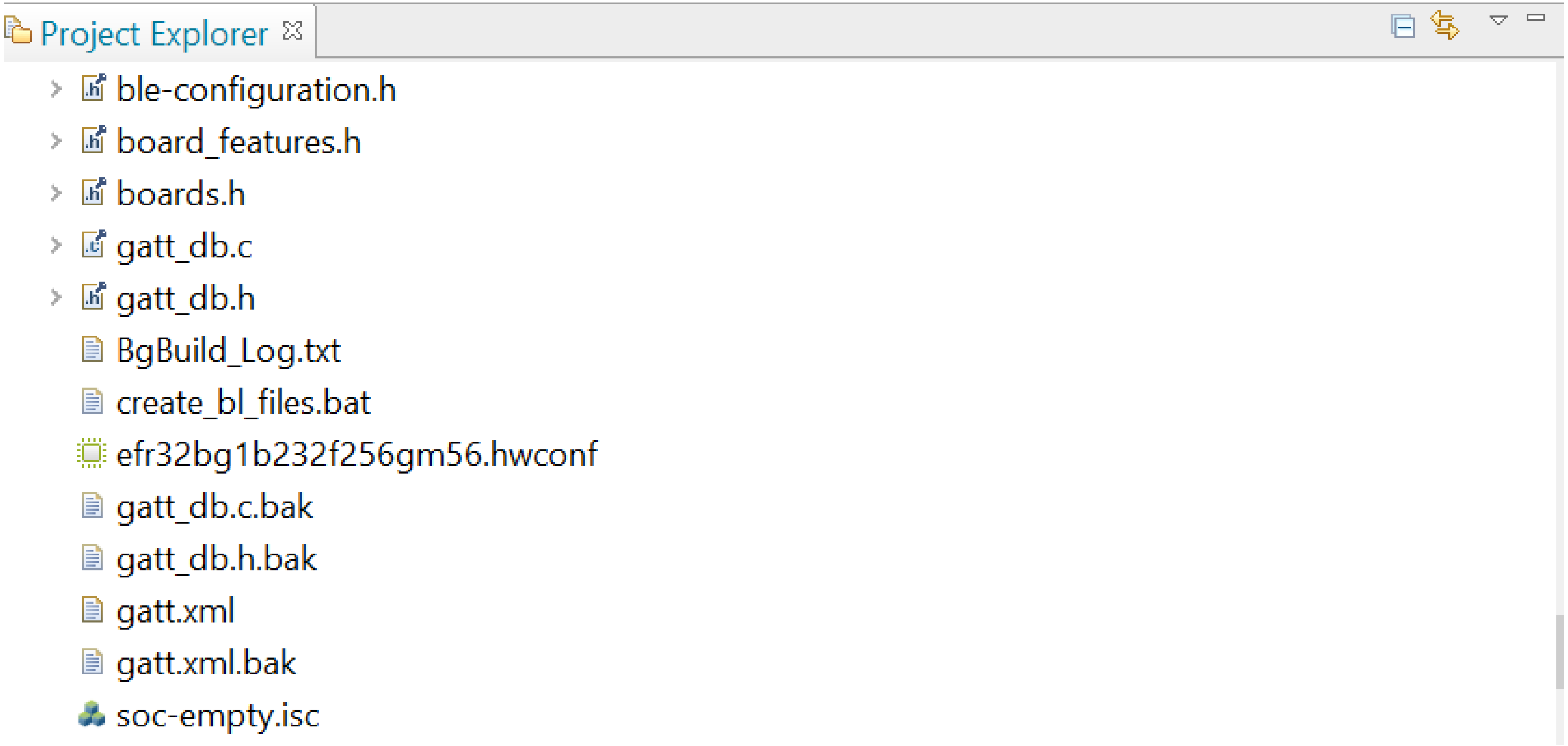
- evt = **gecko_wait_event();**
  - This will replace your Enter_Sleep() routine that you have created and being used
    - For now, in your code, you can simply comment out the code inside your Enter_Sleep() routines. We will discuss how to manage this in more detail in Thursday’s lecture.

Using the Bluetooth Stack on the Blue Gecko

- The Bluetooth Stack requires the following resources:
  - RAM memory – always needed, must be retained in sleep modes
  - RTCC – always needed for sleep timing
  - LDMA – used for handling BGAPI commands in NCP mode
  - UART – used for receiving/transmitting BGAPI commands/responses in NCP mode
  - PROTIMER – used for protocol timing when receiving/transmitting packets
  - RADIO – used for receiving/transmitting packets

Using the Bluetooth Stack on the Blue Gecko

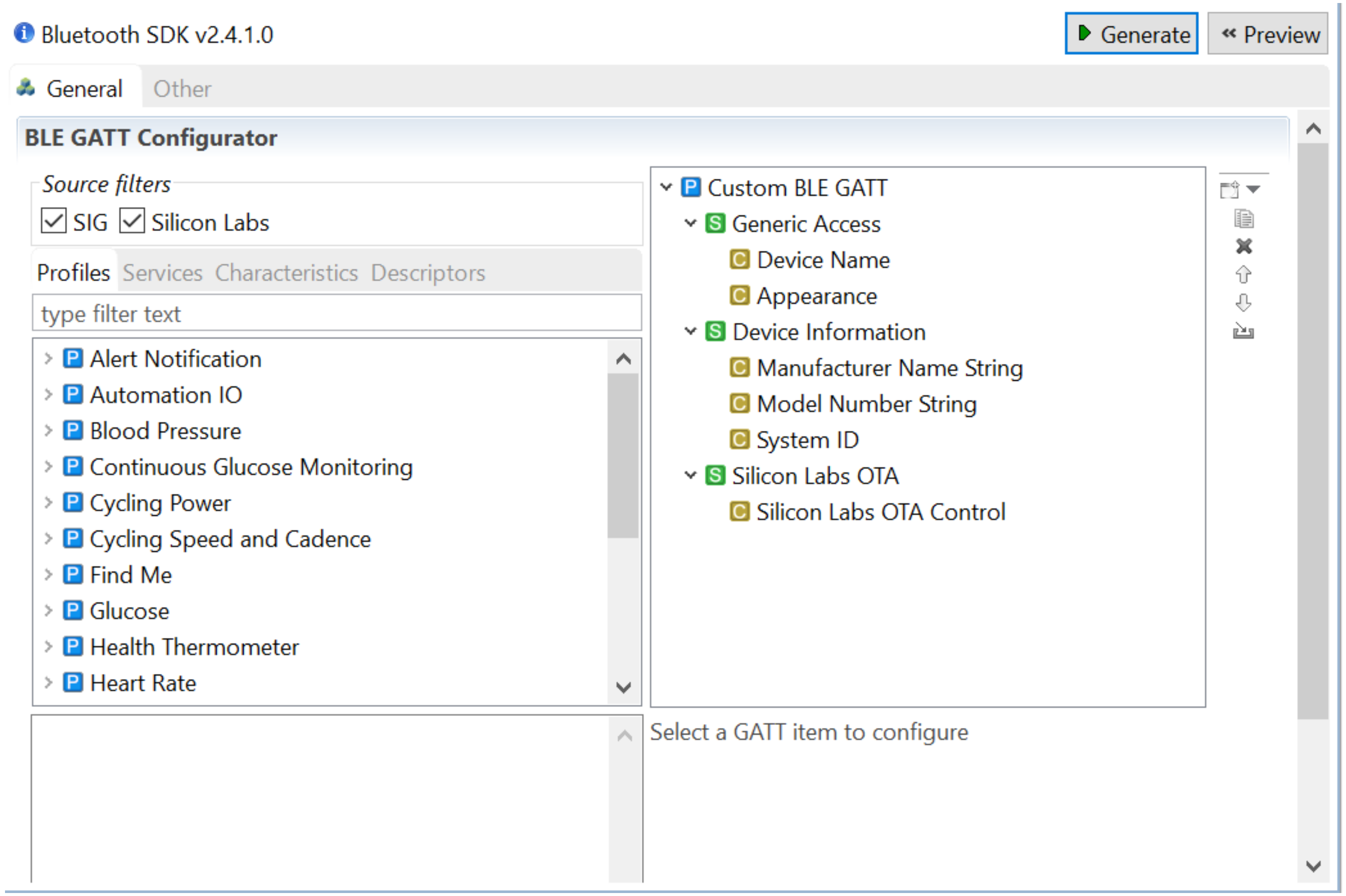
- How to install / specify a Bluetooth service or profile?
  - By its .isc file



```
ble-configuration.h
board_features.h
boards.h
gatt_db.c
gatt_db.h
BgBuild_Log.txt
create_bl_files.bat
efr32bg1b232f256gm56.hwconf
gatt_db.c.bak
gatt_db.h.bak
gatt.xml
gatt.xml.bak
soc-empty.isc
```

Using the Bluetooth Stack on the Blue Gecko

- The .isc file will open and provide the following gui interface


Bluetooth SDK v2.4.1.0

BLE GATT Configurator

Source filters

☑ SIG ☑ Silicon Labs

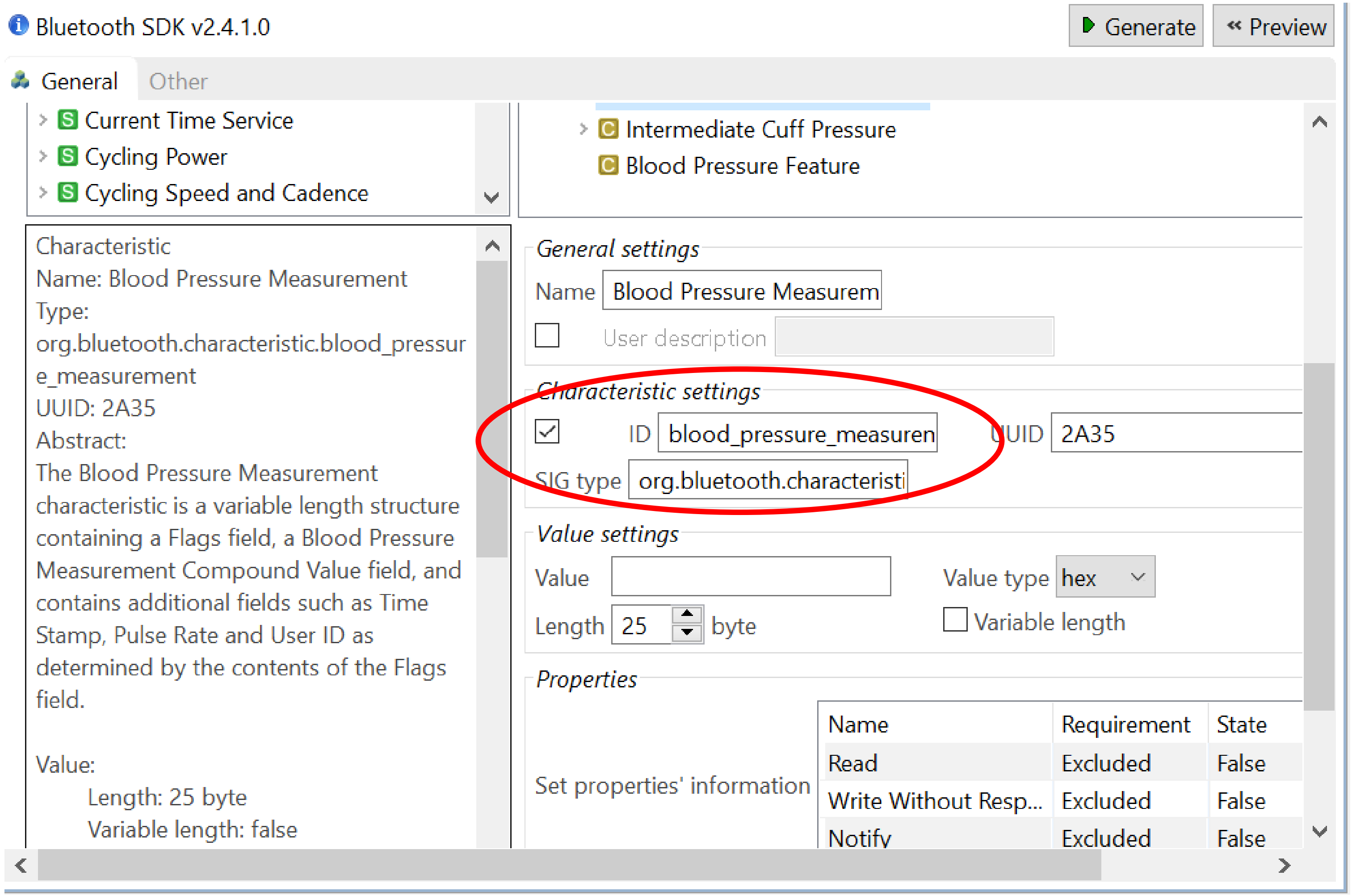
- Alert Notification
- Automation IO
- Blood Pressure
- Continuous Glucose Monitoring
- Cycling Power
- Cycling Speed and Cadence
- Find Me
- Glucose
- Health Thermometer
- Heart Rate

- Custom BLE GATT
  - Generic Access
    - Device Name
    - Appearance
  - Device Information
    - Manufacturer Name String
    - Model Number String
    - System ID
  - Silicon Labs OTA
    - Silicon Labs OTA Control

Select a GATT item to configure

Using the Bluetooth Stack on the Blue Gecko

- How do I know how to address an attribute or characteristic?
  - By its characteristic ID



Bluetooth SDK v2.4.1.0

Characteristic
Name: Blood Pressure Measurement
Type: org.bluetooth.characteristic.blood_pressure_measurement
UUID: 2A35
Abstract:
The Blood Pressure Measurement characteristic is a variable length structure containing a Flags field, a Blood Pressure Measurement Compound Value field, and contains additional fields such as Time Stamp, Pulse Rate and User ID as determined by the contents of the Flags field.

Value:
Length: 25 byte
Variable length: false

General settings

Name: Blood Pressure Measurem

□ User description

Characteristic settings

☑ ID: blood_pressure_measuren   UUID: 2A35

SIG type: org.bluetooth.characteristi

Value settings

Value: Value type: hex

Length: 25 byte   □ Variable length

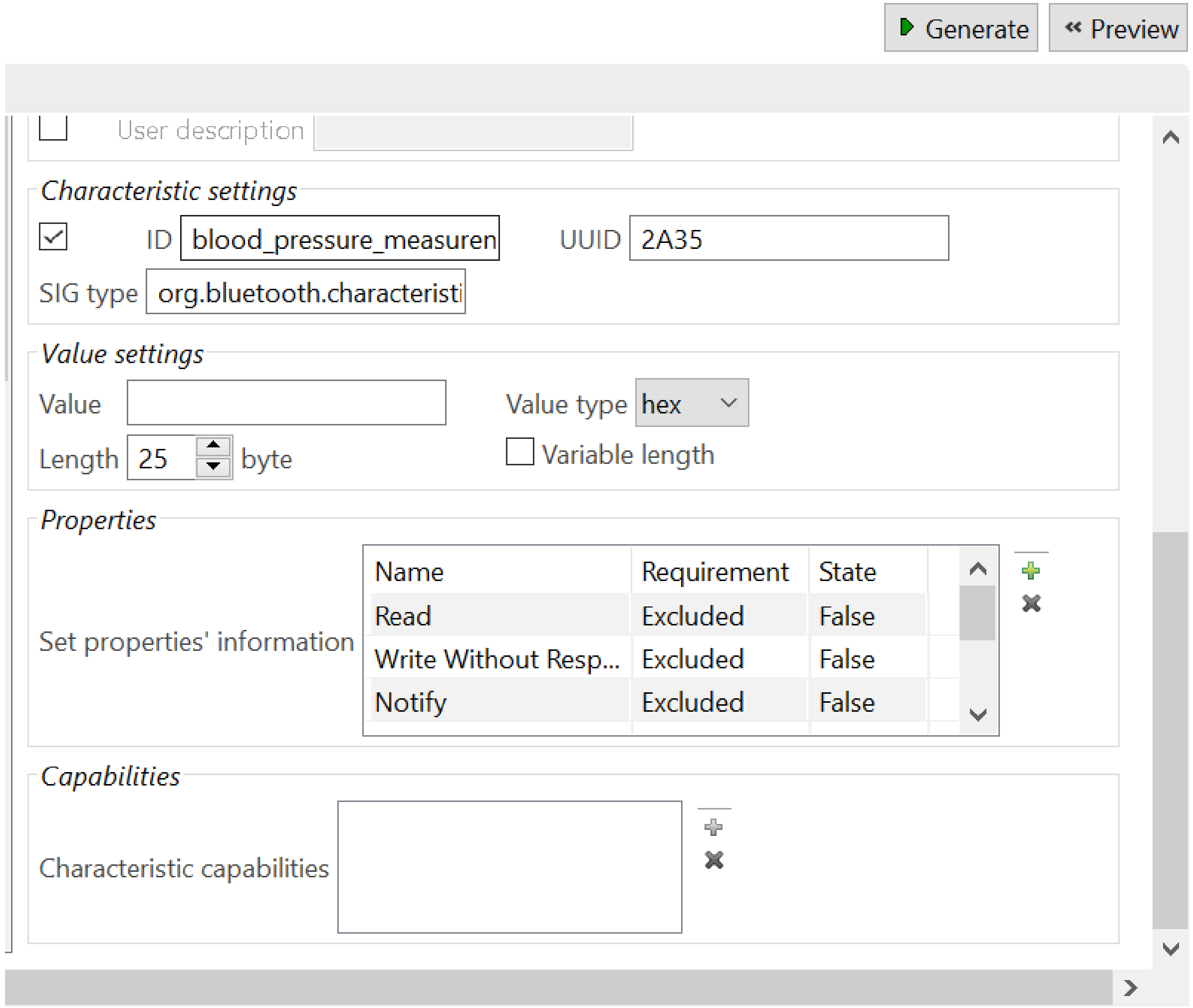
Properties

Set properties' information

Name	Requirement	State
Read	Excluded	False
Write Without Resp...	Excluded	False
Notify	Excluded	False

Using the Bluetooth Stack on the Blue Gecko

- From this GUI interface, you can set:
  - Characteristic ID
  - UUID
  - The initial value
  - Value type
  - Characteristic length
  - Its properties
- Then you must Generate the service/profile



□ User description

Characteristic settings

☑ ID blood_pressure_measuren UUID 2A35

SIG type org.bluetooth.characteristi

Value settings

Value Value type hex

Length 25 byte □ Variable length

Properties

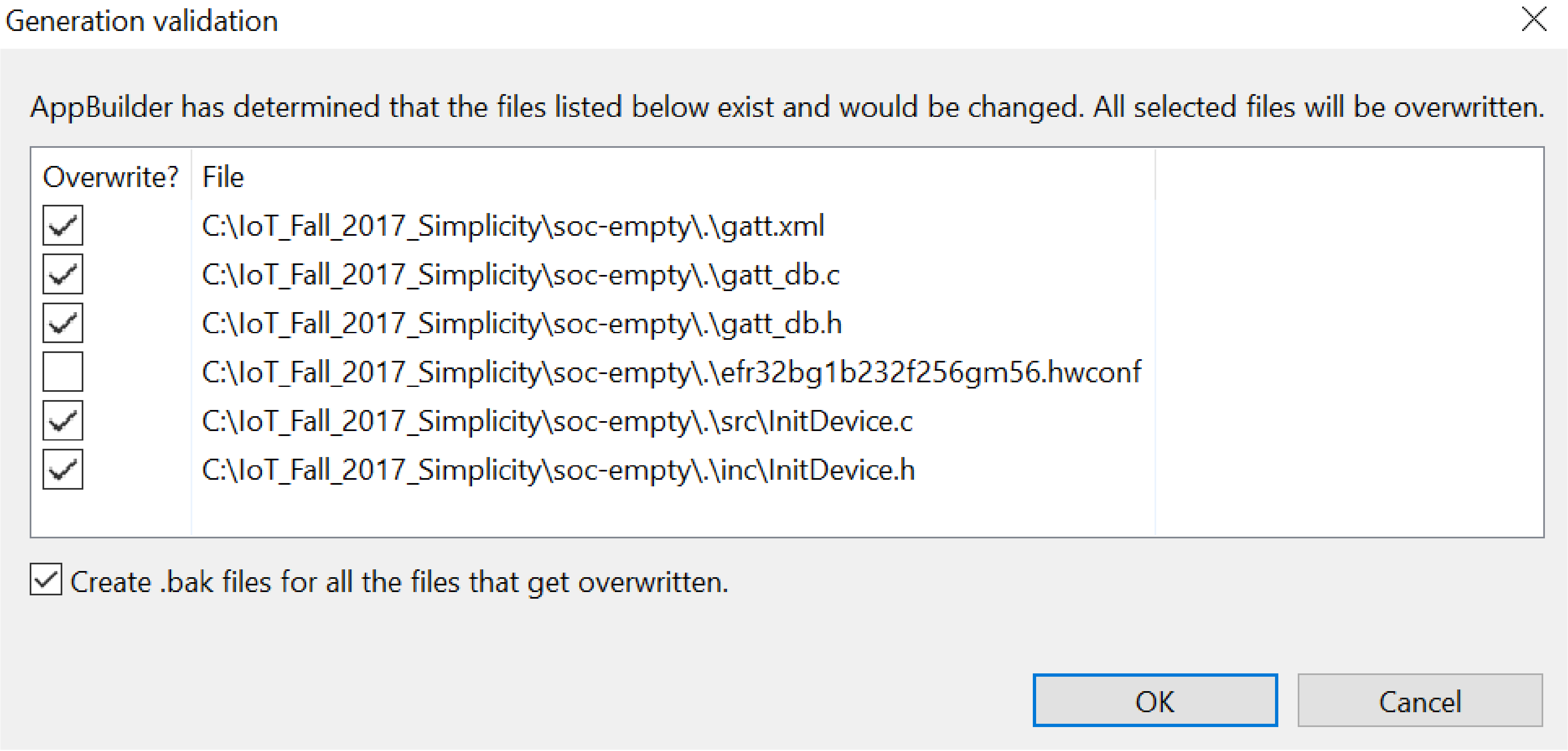
Set properties' information

Name	Requirement	State
Read	Excluded	False
Write Without Resp...	Excluded	False
Notify	Excluded	False

Capabilities

Characteristic capabilities

Using the Bluetooth Stack on the Blue Gecko



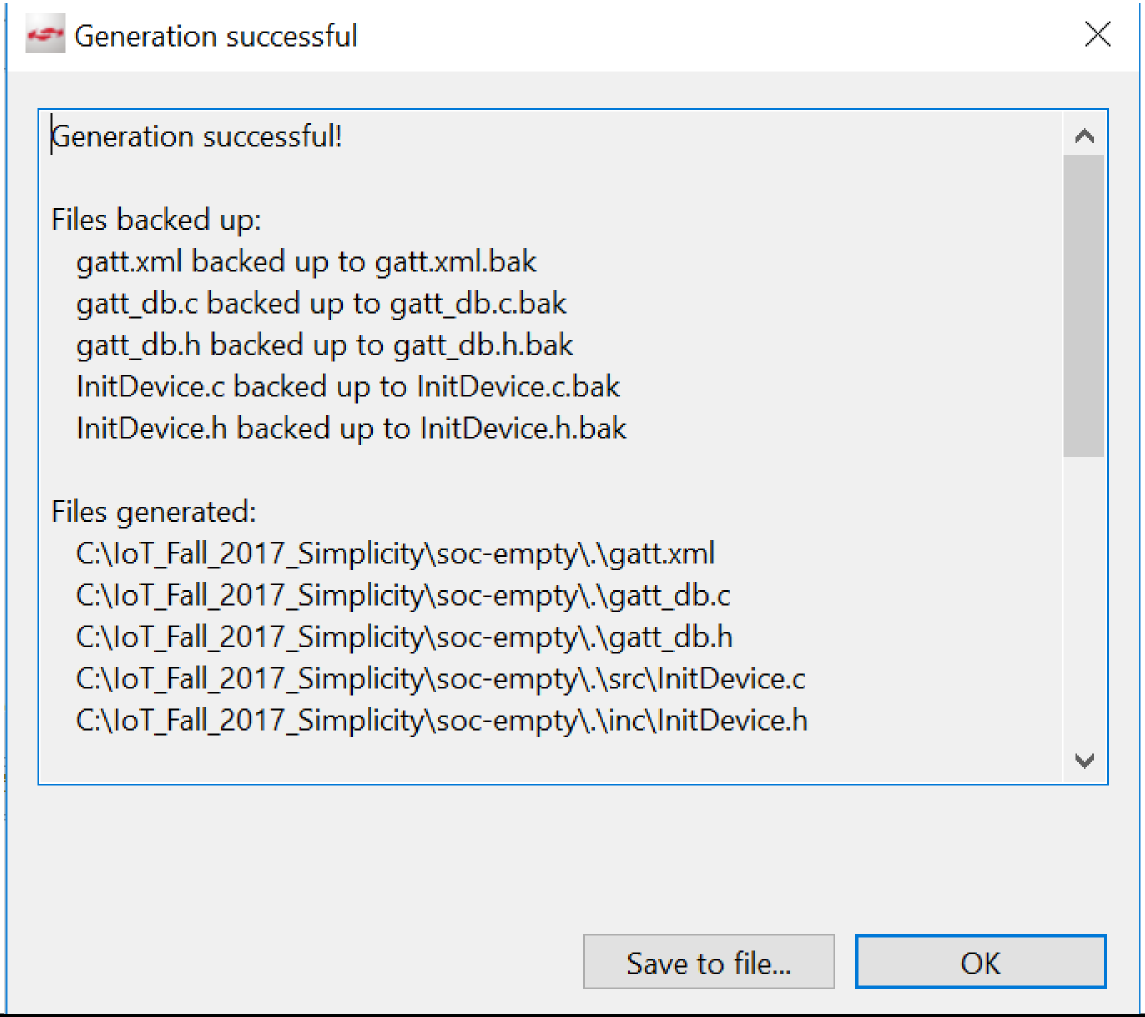
Generation validation

AppBuilder has determined that the files listed below exist and would be changed. All selected files will be overwritten.

Overwrite?	File
☑	C:\IoT_Fall_2017_Simplicity\soc-empty\.\gatt.xml
☑	C:\IoT_Fall_2017_Simplicity\soc-empty\.\gatt_db.c
☑	C:\IoT_Fall_2017_Simplicity\soc-empty\.\gatt_db.h
☐	C:\IoT_Fall_2017_Simplicity\soc-empty\.\efr32bg1b232f256gm56.hwconf
☑	C:\IoT_Fall_2017_Simplicity\soc-empty\.\src\InitDevice.c
☑	C:\IoT_Fall_2017_Simplicity\soc-empty\.\inc\InitDevice.h

☑ Create .bak files for all the files that get overwritten.

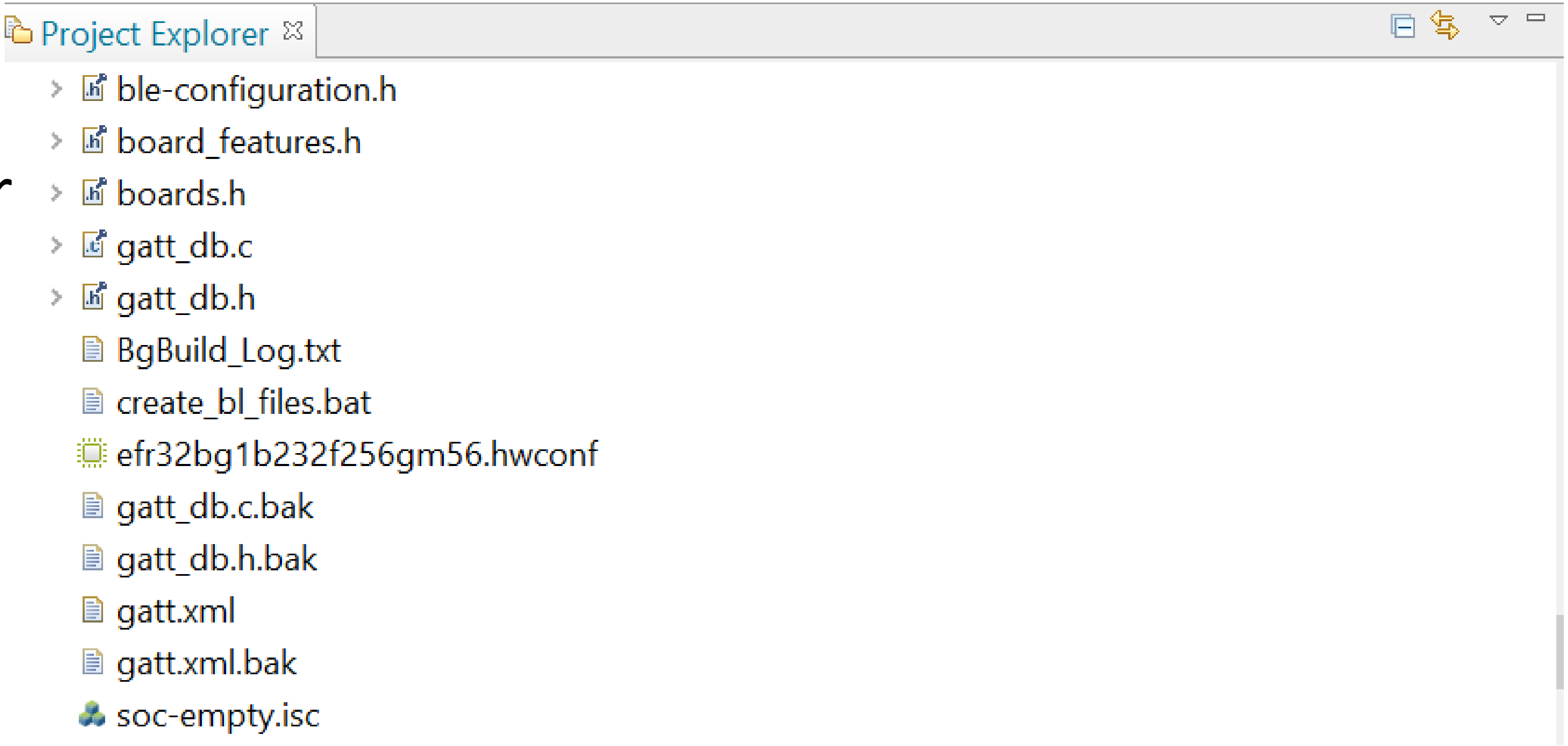
Using the Bluetooth Stack on the Blue Gecko



```
Generation successful!

Files backed up:
  gatt.xml backed up to gatt.xml.bak
  gatt_db.c backed up to gatt_db.c.bak
  gatt_db.h backed up to gatt_db.h.bak
  InitDevice.c backed up to InitDevice.c.bak
  InitDevice.h backed up to InitDevice.h.bak

Files generated:
  C:\IoT_Fall_2017_Simplicity\soc-empty\.\gatt.xml
  C:\IoT_Fall_2017_Simplicity\soc-empty\.\gatt_db.c
  C:\IoT_Fall_2017_Simplicity\soc-empty\.\gatt_db.h
  C:\IoT_Fall_2017_Simplicity\soc-empty\.\src\InitDevice.c
  C:\IoT_Fall_2017_Simplicity\soc-empty\.\inc\InitDevice.h
```

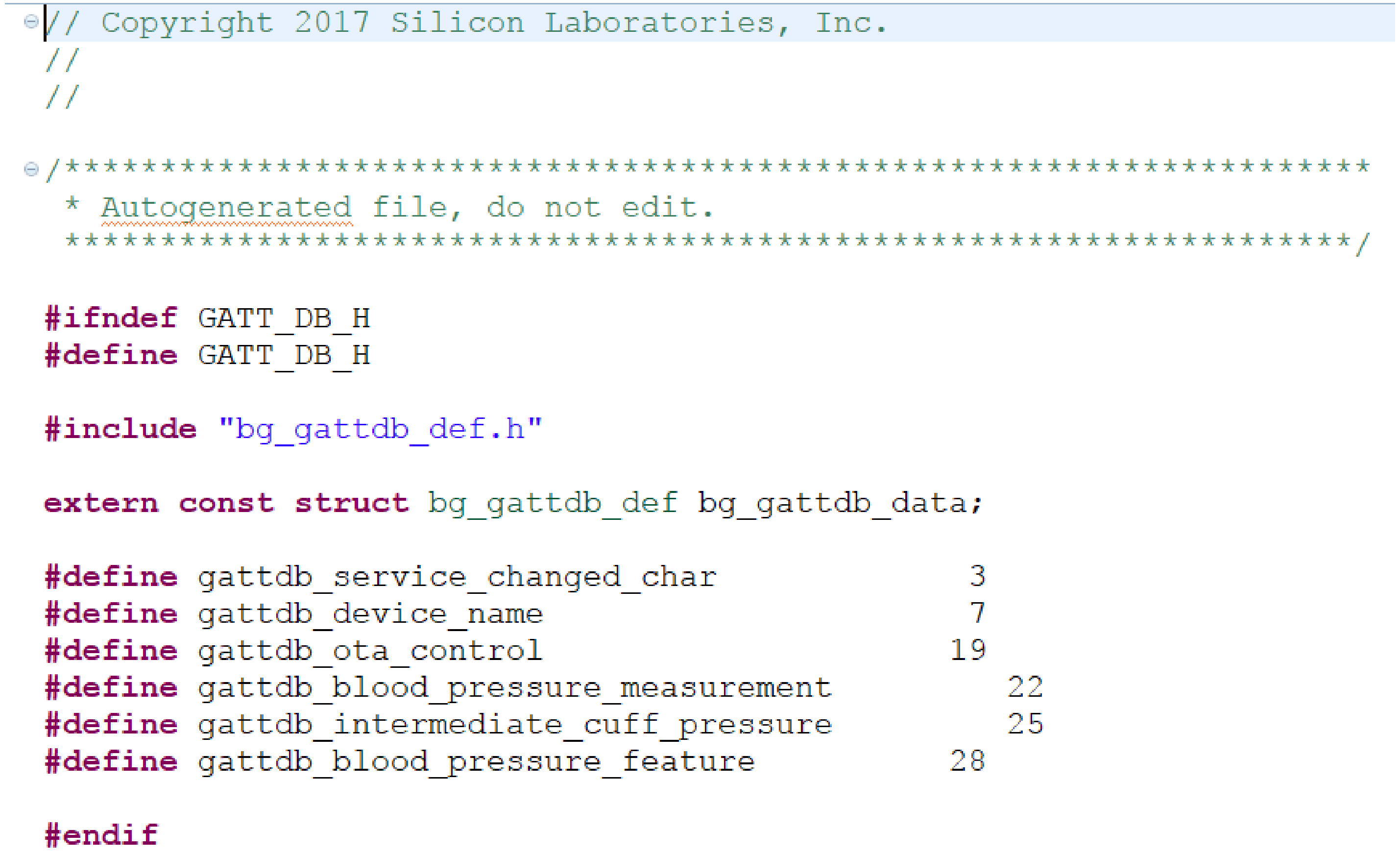
Using the Bluetooth Stack on the Blue Gecko

- Where do I find the characteristic or attribute address once it has been generated?
  - In its gatt_db.c file



```
ble-configuration.h
board_features.h
boards.h
gatt_db.c
gatt_db.h
BgBuild_Log.txt
create_bl_files.bat
efr32bg1b232f256gm56.hwconf
gatt_db.c.bak
gatt_db.h.bak
gatt.xml
gatt.xml.bak
soc-empty.isc
```

Using the Bluetooth Stack on the Blue Gecko



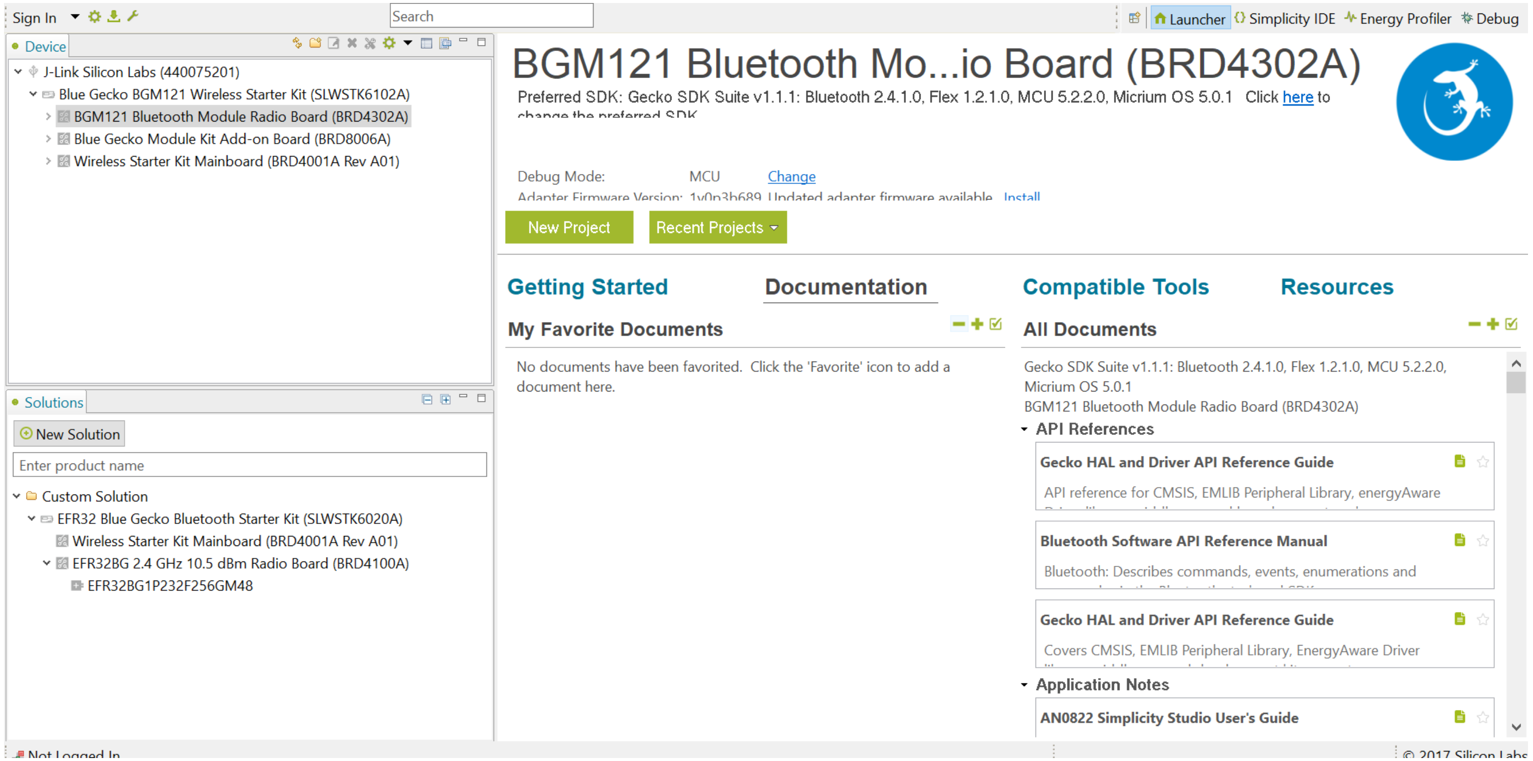
```
// Copyright 2017 Silicon Laboratories, Inc.
//
//

/*********************************************************************
 * Autogenerated file, do not edit.
 ********************************************************************/

#ifndef GATT_DB_H
#define GATT_DB_H

#include "bg_gattdb_def.h"

extern const struct bg_gattdb_def bg_gattdb_data;

#define gattdb_service_changed_char             3
#define gattdb_device_name                      7
#define gattdb_ota_control                     19
#define gattdb_blood_pressure_measurement         22
#define gattdb_intermediate_cuff_pressure         25
#define gattdb_blood_pressure_feature          28

#endif
```

Using the Bluetooth Stack on the Blue Gecko

- How does the application know when to respond to an even? An Interrupt?
- It knows to respond to an event because the following code returns:
  - evt = **gecko_wait_event();**
    - It sets a timer and only returns if a Bluetooth stack event has occurred
- Responding to the events will not be an ISR, but a switch statement in the applications while (1) routine
  - Remember that BLE responds to a single event, not retaining state

Using the Bluetooth Stack on the Blue Gecko

- The next question is how to determine which case statement will be executed



J-Link Silicon Labs (440075201)
Blue Gecko BGM121 Wireless Starter Kit (SLWSTK6102A)
BGM121 Bluetooth Module Radio Board (BRD4302A)
Blue Gecko Module Kit Add-on Board (BRD8006A)
Wireless Starter Kit Mainboard (BRD4001A Rev A01)

Custom Solution
EFR32 Blue Gecko Bluetooth Starter Kit (SLWSTK6020A)
Wireless Starter Kit Mainboard (BRD4001A Rev A01)
EFR32BG 2.4 GHz 10.5 dBm Radio Board (BRD4100A)
EFR32BG1P232F256GM48

BGM121 Bluetooth Mo...io Board (BRD4302A)

Preferred SDK: Gecko SDK Suite v1.1.1: Bluetooth 2.4.1.0, Flex 1.2.1.0, MCU 5.2.2.0, Micrium OS 5.0.1 Click here to change the preferred SDK

Debug Mode: MCU Change

Adapter Firmware Version: 1v0p3b689 Updated adapter firmware available Install

My Favorite Documents

No documents have been favorited. Click the 'Favorite' icon to add a document here.

All Documents

Gecko SDK Suite v1.1.1: Bluetooth 2.4.1.0, Flex 1.2.1.0, MCU 5.2.2.0, Micrium OS 5.0.1
BGM121 Bluetooth Module Radio Board (BRD4302A)

API References

Gecko HAL and Driver API Reference Guide
API reference for CMSIS, EMLIB Peripheral Library, energyAware

Bluetooth Software API Reference Manual
Bluetooth: Describes commands, events, enumerations and

Gecko HAL and Driver API Reference Guide
Covers CMSIS, EMLIB Peripheral Library, EnergyAware Driver

Application Notes

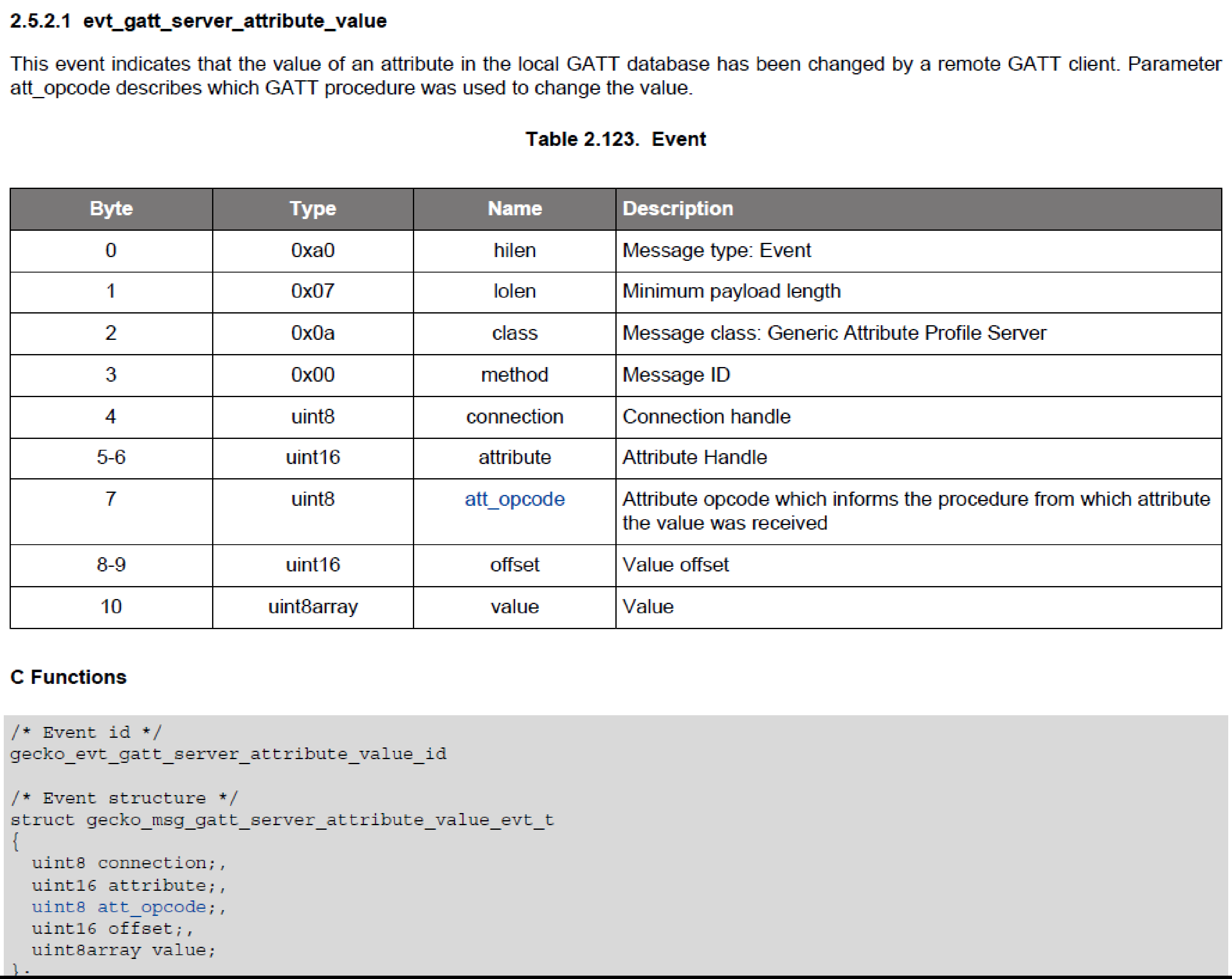
AN0822 Simplicity Studio User's Guide

Not Logged In

© 2017 Silicon Labs

Using the Bluetooth Stack on the Blue Gecko

- Through the Bluetooth API reference manual, you can locate:
  - the event ID
  - data structure



2.5.2.1 evt_gatt_server_attribute_value

This event indicates that the value of an attribute in the local GATT database has been changed by a remote GATT client. Parameter att_opcode describes which GATT procedure was used to change the value.

Table 2.123. Event

Byte	Type	Name	Description
0	0xa0	hilen	Message type: Event
1	0x07	lolen	Minimum payload length
2	0x0a	class	Message class: Generic Attribute Profile Server
3	0x00	method	Message ID
4	uint8	connection	Connection handle
5-6	uint16	attribute	Attribute Handle
7	uint8	att_opcode	Attribute opcode which informs the procedure from which attribute the value was received
8-9	uint16	offset	Value offset
10	uint8array	value	Value

C Functions

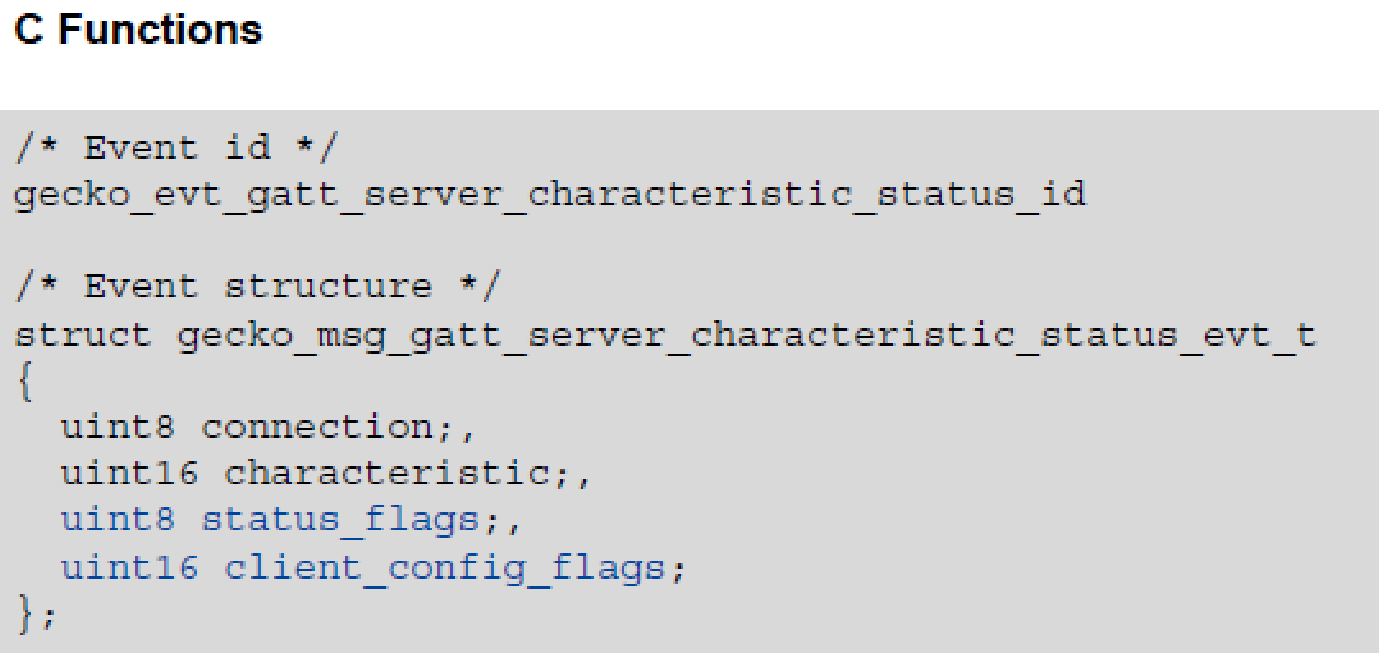
```
/* Event id */
gecko_evt_gatt_server_attribute_value_id

/* Event structure */
struct gecko_msg_gatt_server_attribute_value_evt_t
{
  uint8 connection;,
  uint16 attribute;,
  uint8 att_opcode;,
  uint16 offset;,
  uint8array value;
};
```

Silicon Labs Bluetooth API document

Using the Bluetooth Stack on the Blue Gecko

- Events include a header and data
- Header information is accessed by evt->header
- Data is accessed by
  - evt->data.event_id_minus_the_gecko_msg_in_front_and_minus_evt_t_at_the_end.data_element

Using the Bluetooth Stack on the Blue Gecko



C Functions

```
/* Event id */
gecko_evt_gatt_server_characteristic_status_id

/* Event structure */
struct gecko_msg_gatt_server_characteristic_status_evt_t
{
  uint8 connection;,
  uint16 characteristic;,
  uint8 status_flags;,
  uint16 client_config_flags;
};
```

characterist_status = evt->data.gatt_server_characterist_status.status_flags;

Silicon Labs Bluetooth API document