

# ECEN 5823-001 / -001B

Internet of Things Embedded Firmware

Lecture #21

7 November 2017

# Agenda

- Class announcements
- Reading Assignment
- Project Proposal feedback
- ASIP/IP Processor Design Course, ECEN 5013-003
- Bluetooth DFU OTA demo
- Bluetooth Mesh

# Class Announcements

- Quiz #8 is due at 11:59pm this Sunday, November 125<sup>th</sup>, at 11:59pm
- Any questions regarding the Course Project?
- Any questions regarding Bluetooth Mesh?

# Reading Assignment

ECEN5823-001, -001B – Reading List  
Internet of Things Embedded Firmware  
Week 11

Note: There is a quiz this week. The material covered on the quiz will be from the last several chapters of the course textbook as well as the below readings and course lectures.

1. Silicon Labs' "AN1045: Bluetooth® Over-the-Air Device Firmware Update for EFR32xG1 and BGM11x Series Products.pdf"  
- Located on the D2L week 11 reading assignment folder
  
1. Silicon Labs' White Paper, "Bluetooth Mesh Technology Wireless Technology for the World of IoT.pdf"  
- Located on the D2L week 11 reading assignment folder

# Project Proposal Feedback

- Fantastic project ideas!
- Two pieces of feedback
  - For Bluetooth Mesh, I am looking for who will do the Mesh Proxy Node and who will be doing the software for the sensor/actuator Mesh Node
  - Schedule requires a date, not just a list of activities
- If I provided feedback asking for additional information, you do not need to provide an update proposal
  - You will provide the answer / additional information in your first Course Project Update
- Course Project Update #1 is due Saturday, November 18<sup>th</sup>, at 11:59pm

# ECEN 5013-13: ASIP and IP Core Processor Design

**Custom solution that provides  
The required performance  
At lower power and cost**



Contadina Tomato Sauce, 8-Ounce

\$0.75



Muir Glen Organic Tomato Sauce, 15 oz

\$0.95



Gebardi Chicago Pizza Sauce, 16 oz

\$0.75



Muir Glen Organic Chunky Tomato Sauce, 28 Ounce

\$2.99

# ECEN 5013-13: ASIP and IP Core Processor Design

- A graduate level course in developing a optimized micro-controller to satisfy an application performance, power, and/or cost goal
- Over the course of 15 weeks, the students will learn a high level descriptive language and tool to design the micro-controller which will generate the processor's RTL as well as c-compiler and assembler
- With the knowledge of a high level descriptive language, the students will evaluate the performance of the micro-controller through profiling their application
  - With the profiled analysis, the student will modify the processor design to meet their application performance, power, and/or cost objectives
  - This optimization may include a new processor instruction, adding a Multiply Accumulate, or Vector Processing functions

# ECEN 5013-13: ASIP and IP Core Processor Design

- With the processor completed, the student will then realize their custom processor in a FPGA to realize their own optimized micro-controller
- The instructing team will provide base files or starting points along the way to enable this journey to be completed
- When: Spring 2018, MWF, at 10:00 to 10:50am
- Where: FLMG 103
- Note: PSoC is not a prerequisite to this course.

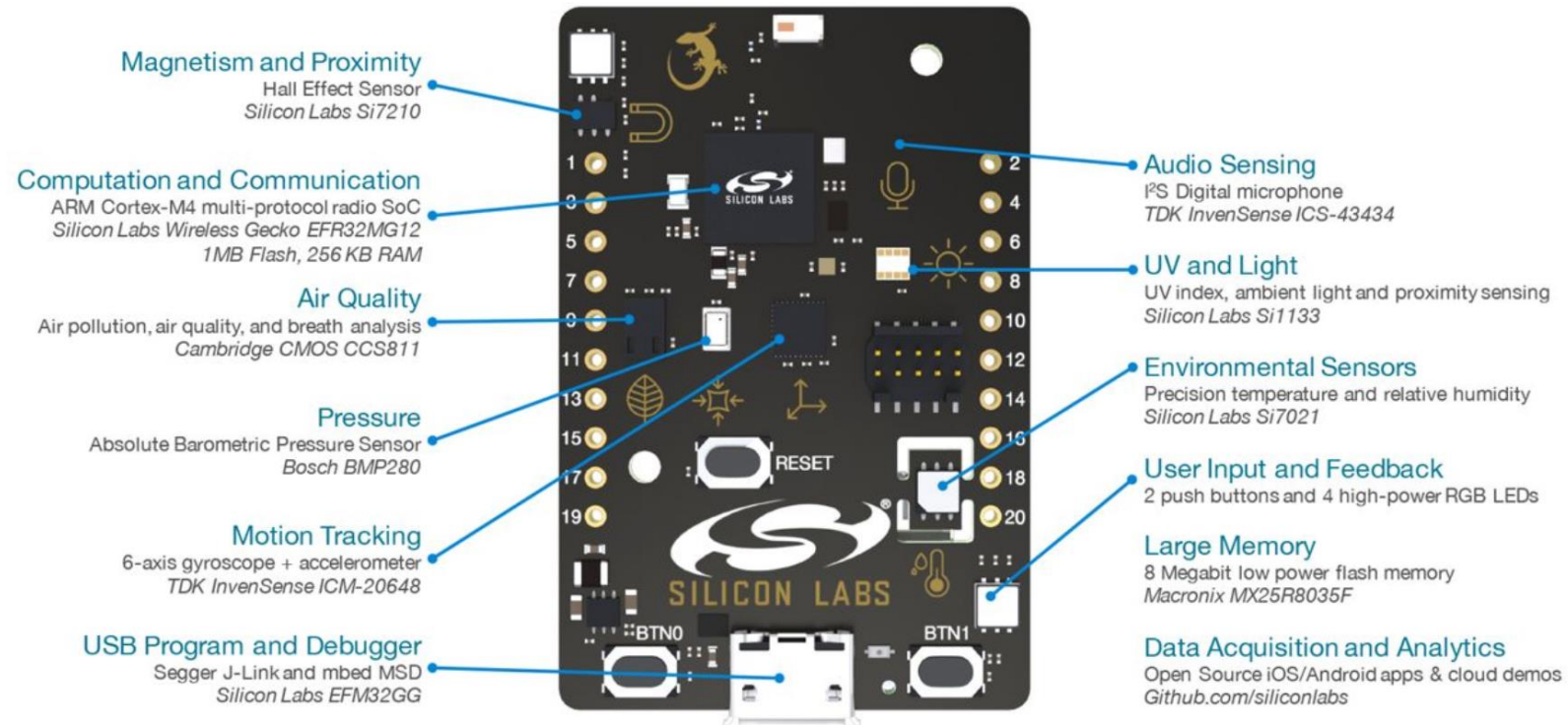


# Project Proposal Feedback

- All sensors have been ordered
- Working with Silicon Labs to verify best Bluetooth Mesh compatible Radio board to order

# Project Proposal Feedback

- An alternative radio board solution would be the Silicon Labs Thunderboard Sense



# BLE OTA demo

File Edit Navigate Search Project Run Window Help

Project Explorer

- Assignment\_1 [GNU ARM v4.9.3 - Default] [EFR32BG1B23]
- Assignment\_2 [GNU ARM v4.9.3 - Default] [EFR32BG1B23]
- Assignment\_3 [GNU ARM v4.9.3 - Default] [EFR32BG1B23]
- Assignment\_4 [GNU ARM v4.9.3 - Default] [EFR32BG1B23]
- Assignment\_5 [GNU ARM v4.9.3 - Default] [EFR32BG1B23]
- Assignment\_6 [GNU ARM v4.9.3 - Default] [EFR32BG1B23]
- Assignment\_OTA [GNU ARM v4.9.3 - Default] [EFR32BG1B23]
  - Binaries
  - Includes
  - bgapi
  - device
  - emlib
  - GNU ARM v4.9.3 - Default
  - inc
  - kit\_flashpwr
  - linker
  - output\_ebl
  - output\_gbl
  - src
    - adc.c
    - adc.h
    - bluetooth.c

Profiler README.txt main.h soc-empty.isc main.c native\_gecko.h InitDevice.c bootloader-storage-intern... soc-empty.isc

Bluetooth SDK, version:2.4.1.0

General Other

### BLE GATT Configurator

Source filters

☒ SIG ☒ Silicon Labs

Profiles Services Characteristics Descriptors

type filter text

- 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
- Tx Power
  - Tx Power Level
- Health Thermometer
  - Temperature Measurement
- Client Characteristic Configuration

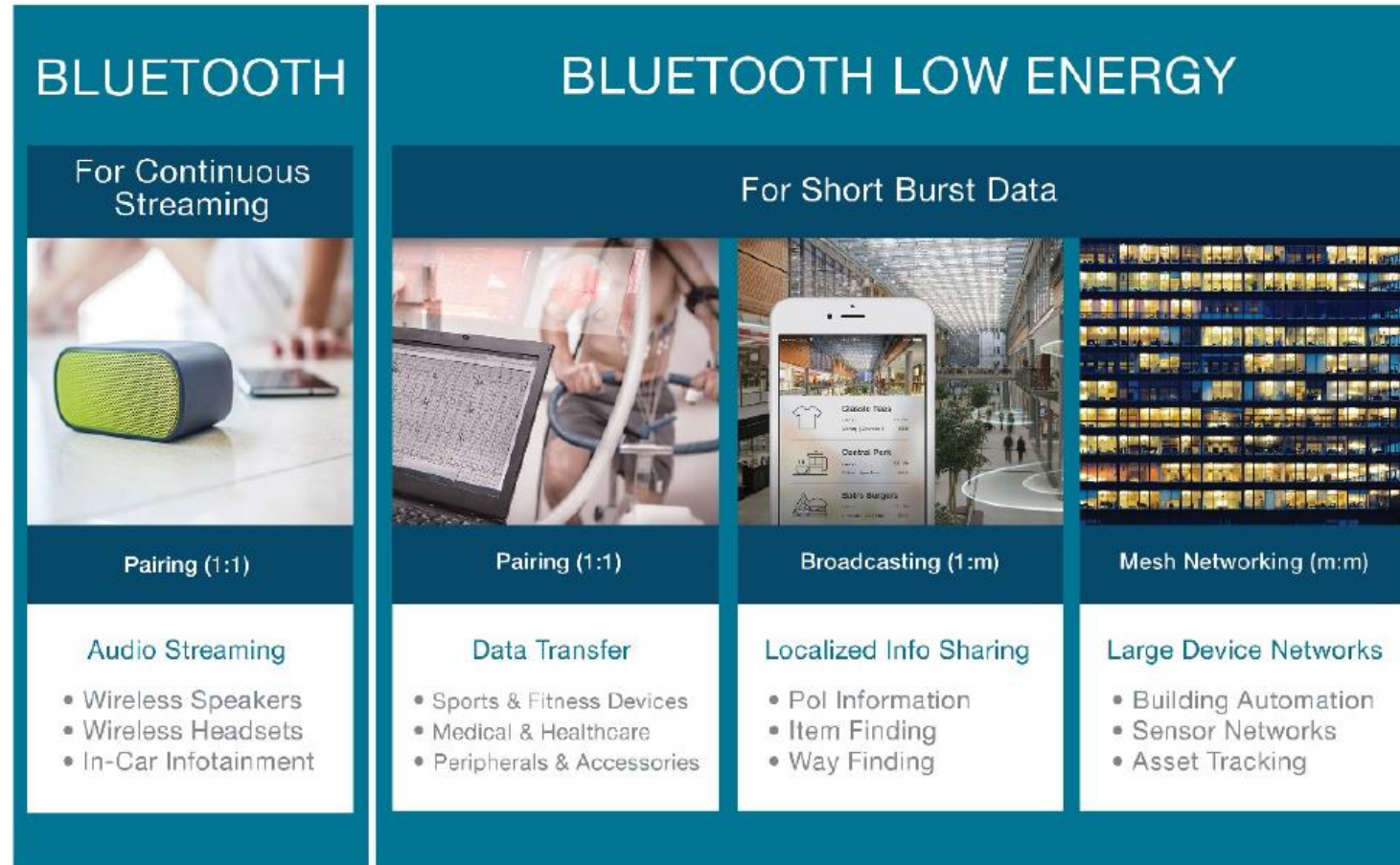
Select a GATT item to configure

Generate Preview

# Silicon Labs' OTA resources

- To learn more about setting up your Application for OTA:
  - <https://www.silabs.com/documents/login/application-notes/an1045-bt-ota-dfu.pdf>
- To learn more about Gecko Bootloader, please refer to these documents:
  - <https://www.silabs.com/documents/public/user-guides/ug266-gecko-bootloader-user-guide.pdf>
  - [https://www.silabs.com/community/wireless/bluetooth/knowledge-base.entry.html/2017/06/22/secure\\_ota\\_dfu-Wb22](https://www.silabs.com/community/wireless/bluetooth/knowledge-base.entry.html/2017/06/22/secure_ota_dfu-Wb22)
  - [https://www.silabs.com/community/wireless/bluetooth/knowledge-base.entry.html/2017/04/12/adding\\_gecko\\_bootloa-osqt](https://www.silabs.com/community/wireless/bluetooth/knowledge-base.entry.html/2017/04/12/adding_gecko_bootloa-osqt)

# Bluetooth Mesh



*Development of Bluetooth standard from Bluetooth BR/EDR (left) to LE and Bluetooth mesh (right)*

# Bluetooth Technology Summary

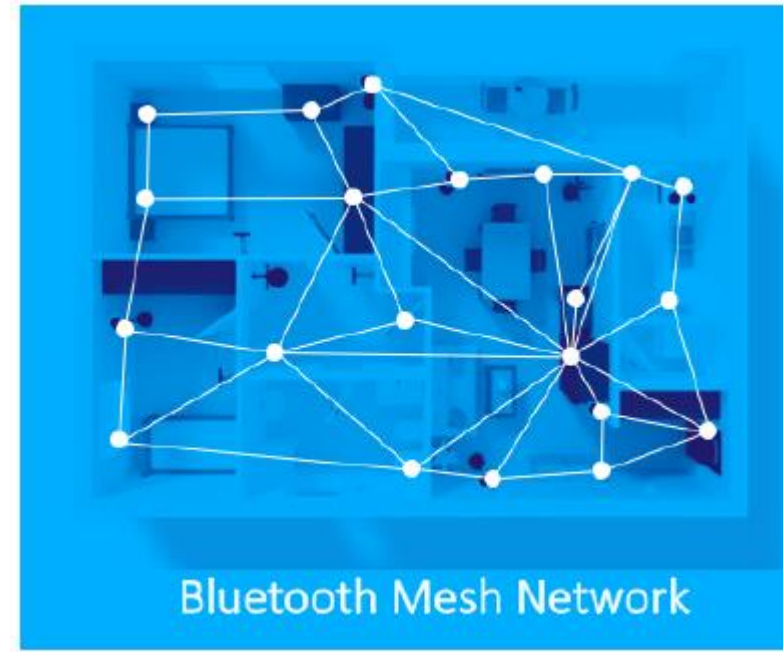
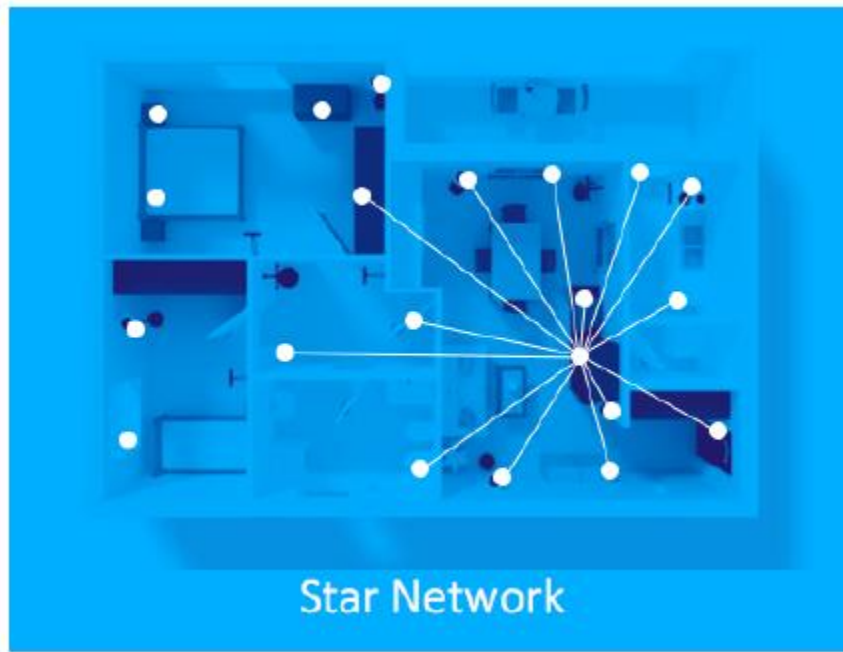
- Bluetooth Classic BR/EDR
  - de facto wireless technology for voice and audio streaming and use cases like wireless speakers, headphones and in-car infotainment
  - not optimized for low power applications IoT devices
- Bluetooth Low Energy
  - short burst wireless connections and data broadcast for the wearable devices, sports and fitness sensors and beacon applications
  - Limited range and number of connections

# Bluetooth Technology Summary

- Bluetooth Mesh
  - addresses the needs of the quickly growing IoT market
  - Increased connectivity
  - Increased reach – mesh network
  - Increase functionality – publish-subscribe



# Bluetooth Mesh



- Bluetooth mesh nodes can also support existing Bluetooth LE topologies and use cases such as point-to-point connectivity and Bluetooth beaconing



# How does Bluetooth Mesh Route?

- **Flooding type meshing:**
  - Broadcast channels are used to transmit messages from nodes
  - As messages are received they are relayed further making it possible to extend the range by adding more nodes
  - If the density of the mesh nodes is sufficient that any node can send messages to any other node directly it is **full mesh**
  - or first to nodes close-by after which the nodes relay the messages onward to further nodes it is **partial mesh**
- Flooding-based mesh networks are considered easier to implement and more resilient if compared with routing-based mesh networks
- But, routing-based networks have better scalability

# Bluetooth Mesh – Managed Flooding

Feature	Description
Heartbeats	<ul style="list-style-type: none"> <li>Nodes send heartbeat messages periodically to indicate activity.</li> <li>Heartbeat messages contain data to allow receiving nodes to determine the number of hops between the nodes which can be taken into account using the Time to Live (TTL) parameter (see below).</li> </ul>
Time to Live (TTL)	<ul style="list-style-type: none"> <li>All Bluetooth mesh PDU's contain this field.</li> <li>Controls maximum number of hops.</li> <li>Compulsory for each node.</li> <li>All recently received messages are stored in the cache.</li> <li>If the received message is already in the cache it is an indication that the message has been processed already and is therefore discarded immediately.</li> </ul>
Network Message Cache	
Friendship	<ul style="list-style-type: none"> <li>Used in combination with Low Power nodes.</li> <li>Friendship node receives and buffers addressed to the configured Low Power node and forwards buffered messages to the Low Power node when it becomes active after waking up.</li> </ul>