ECEN 5023-001, -001B, -740

Mobile Computing & IoT Security

Lecture #13

28 February 2017

Be Boulder.

University of Colorado Boulder

Calculating the ADC Acquisition cycles and prescaler

- Assignment: OSR = 1, no over sampling, 12 bit resolution, 20,000 samples per second, minimum acquisition time of 3uS
  - Ta = Resolution*OSR*SamplePerSecond*Acquistion_Time / (1-OSR*SamplePerSecond*Acquisition_Time
  - Ta = 12*1*40,000*3x10^-6 / (1-1*40,000*3x10^-6)
  - Ta = 1.44 /0.88
  - Ta = 1.64
  - Cannot have a partial clock, so must always round up, so Ta = 2
- Now, go back and calculate the ADC clock frequency to determine prescaler

Calculating the ADC Acquisition cycles and prescaler

- Calculating the prescaler
  - (Ta + Resolution) * OSR * SamplesPerSecond = ADC_Clocks_Freq
  - (2 + 12) * 1 * 40,000 = ADC_Clocks_Freq
  - ADC_Clock_Freq = 560,000
  - Prescaler = CPU_Freq / ADC_Clock_Freq = 14,000,000 / 560,000
  - Prescaler = 25
  - Prescalers need to be integers, round down to obtain desired ADC_Clocks_Freq or greater. Rounding down will give you too few ADC clocks
- Verify
  - 14,000,000 / 25 = 560,000 ADC clocks which provides the required 260,000
  - Ta = 2 * 1/ADC_Clock_Freq = 2 * 1/560,000 = 3.57uS which is greater than 3uS

Verified

Verified

ADC prescaler

```
void ADC0_Setup(void){
    int ADC_Desired_Freq;

    ADC_Init_TypeDef          adc0_init;
    ADC_InitSingle_TypeDef    adc0_initsingle;

    ADC_Desired_Freq = Temp_Sense_Acq_Eval * Temp_Sense_Rate;

    adc0_init.lpfMode = adcLPFilterBypass;
    adc0_init.ovsRateSel = ADC0_osr;
    adc0_init.prescale = ADC_PrescaleCalc(ADC_Desired_Freq,0);
    adc0_init.tailgate  = false;
    adc0_init.timebase = ADC_TimebaseCalc(0);
    adc0_init.warmUpMode = ADC0_warmup;
```

ADC prescaler

Launcher | Simplicity IDE | Energy Profiler | Resource | Debu

Variables | Breakpoints | Registers | Expressions

Name	Value	Description
ADC0		ADC0
CTRL	0xD1800	Control Register
WARMUPMODE	0x0 - NORMAL (ADC is shut down af...	Warm-up Mode
TAILGATE	0x0	Conversion Tailgating
LPFMODE	0x0 - BYPASS	Low Pass Filter Mode
PRESC	0x18	Prescaler Setting
TIMEBASE	0xD	Time Base
OVSRSEL	0x0 - X2 (2 samples for each conversi...	Oversample Rate Select
CMD	0x0	Command Register

This is 24?

```
    hfperFreq = CMU_ClockFreqGet(cmuClock_HFPER);
  }

  ret = (hfperFreq + adcFreq - 1) / adcFreq;
  if (ret)
  {
    ret--;
  }
```

ADC prescaler

```
uint8_t ADC_PrescaleCalc ( uint32_t  adcFreq,
                           uint32_t  hfperFreq
                         )
```

Calculate prescaler value used to determine ADC clock.

The ADC clock is given by: HFPERCLK / (prescale + 1).

Parameters

[in]	**adcFreq**	ADC frequency wanted. The frequency will automatically be adjusted to be within valid range according to reference manual.
[in]	**hfperFreq**	Frequency in Hz of reference HFPER clock. Set to 0 to use currently defined HFPER clock setting.

Returns

Prescaler value to use for ADC in order to achieve a clock value <= `adcFreq`.

I2C TSL2651 plus Load Power Management Assignment Rubric

Questions are worth a total of 5.0pts

- Question 1: 3.8 – 3.9 (0.5 out of 1pt), 4.0 – 4.1 Energy Score (1.0 out of 1pt)
- Question 2: 3.8 – 3.9 (0.5 out of 1pt), 4.0 – 4.1 Energy Score (1.0 out of 1pt)
- Question 3: < 1.0uA (1pt)
- Question 4: < 275uS (1pt)
- Yes (0.5 pts)
- Yes (0.5 pts)

I2C TSL2651 plus Load Power Management Assignment Rubric

a. 1.0 pt total for the following defined statements:
   a. 0.34 points if code used #defined statements for the following:

```
#define I2C1_SCL_Port         gpioPortC
#define I2C1_SCL_Pin                5U
#define I2C1_SDA_Port         gpioPortC
#define I2C1_SDA_Pin                4U
```

   b. 0.33 points if code used 10 of 15 below #defined statements for the following or similar (they could be doing the shifting in the program):

```
#define TSL_Pwr_Port          gpioPortD
#define TSL_Pwr_Pin                 0U
#define TSL_Int_Port          gpioPortD
#define TSL_Int_Pin                 1U


#define TSL_Chip_Addr         (0x39 << 1) /* or some
variant of defining the TSL2651 I2C chip address */

#define TSL_Ctrl_Addr               0x00
#define TSL_Timing_Addr             0x01
#define TSL_ICR_Addr                0x06

#define TSL_THRESHLOWLOW_Addr       0x02
#define TSL_THRESHLOWHIGH_Addr      0x03
#define TSL_Low_Int_Level           0x000f
#define TSL_THRESHLOWLOW_Value      (TSL_Low_Int_Level &
0x00ff)
```

I2C TSL2651 plus Load Power Management Assignment Rubric

```
#define TSL_THRESHLOWHIGH_Value      ((TSL_Low_Int_Level >>
8) & 0x00ff)

#define TSL_THRESHHIGHLOW_Addr       0x04
#define TSL_THRESHHIGHHIGH_Addr      0x05
#define TSL_High_Int_Level           0x0800
#define TSL_THRESHHIGHLOW_Value      (TSL_High_Int_Level &
0x00ff)
#define TSL_THRESHHIGHHIGH_Value     ((TSL_High_Int_Level >>
8) & 0x00ff)

#define TSL_DATA0LOW_Addr            0x0c
#define TSL_DATA0HIGH_Addr           0x0d
```

b. 0.33 points if code used 6 of 9 below #defined statements for the following or similar (they could be doing the shifting in the program):

```
#define TSL_Pwr_Port           gpioPortD
#define TSL_Pwr_Pin                  0U
#define TSL_Int_Port           gpioPortD
#define TSL_Int_Pin                  1U
```

I2C TSL2651 plus Load Power Management Assignment Rubric

b. 1.0 pts for the correct sequence of disabling the TSL2651 for load power management
    a. Disable Interrupts
    b. Disable GPIO I/O pins (order can vary)
        i. SCL
        ii. SDA
        iii. INT
    c. Turn off power to the GPIO power pin by clearing the pin
        i. Do not disable, but clear the pin to 0

c. 1.0 pts for the correct sequence of enabling the TSL2651 for load power management
    d. Turn power onto the external device
    e. Wait for power to stabilize + external boot time
    f. Enable GPIO pins on MCU after peripheral to protect ESD diodes (order can vary)
        i. SCL
        ii. SDA
        iii. INT
    g. Initialize the TSL2651 for operation
    h. Enable Interrupts

I2C TSL2651 plus Load Power Management Assignment Rubric

b. 1.0 pts for correct ADC timing
    i. TimeAcq / Prescaler combinations (valid combinations) (0.66pts)
        1. 2 and 24/25
    ii. Number of conversions = 1000 (0.33pts) (measured out to 24.5 to 25.5mS)

c. 0.5 pts for LED0 turning on/off based on light level

d. 0.5 pts for LED1 turning on when temperature goes below temperature limits and turns off when the temperature sensor warms up

e. (-0.5 pts deduction) if the measurement between first LETIMER0 interrupt and the second interrupt is less than 5mS

f. (-0.5pts deduction) if there is not comment attributing the sleep routines to Silicon Labs or if it does not mention which routines are covered by it.

g. (-0.5pt deduction) if there is not comment attributing the temperature routines to Silicon Labs (It could be incorporated into the sleep routine IP statement)

Agenda

- I2C TSL2651 plus Load Power Management Assignment Review
- Class Announcements
- Reading assignment
- Mid-Term format
- Quiz 6 review
- What to expect between now and the course project
- Bluetooth Smart / Low Energy

Class Announcements

- No quiz for the week of February 27th!
- Atmel tutorial assignment due before March 7th
  - Questions regarding the tutorial will be on the mid-term
- Mid-term will be held in class on Tuesday, March 7th, at 6:30 in class
  - For on campus students, you must be in class for the exam
  - For distant learners, the mid-term will be due by 6:00pm on Thursday, March 9th

Assigned Reading

ECEN5023-001, -001B, -740 – Reading List
Mobile Computing and the Internet of Things Security
Week 7

1. "Bluetooth Low Energy, The Developer's Handbook," by Robin Heydon
   ISBN: 978-0-13-28836-3
   Chapter 5: The Physical Layer

2. "Bluetooth Low Energy, The Developer's Handbook," by Robin Heydon
   ISBN: 978-0-13-28836-3
   Chapter 6: Direct Test Mode

Mid-Term

- March 7th, 2017
  - For the distant learners, the Mid-Term will be available from March 7th at 6:30pm to Thursday the 9th at 6:30pm
- 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

Mid-Term

- Material covered will include:
  - All the readings from the first day of class
  - All the lectures through Thursday, March 2nd, 2017
  - All assignments
  - Atmel ATSAMB11 tutorial
- 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 6 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 will be used with equal priority
- Notice, that the advertising channel numbers are greater than 37, so they will never be used in this hop sequence

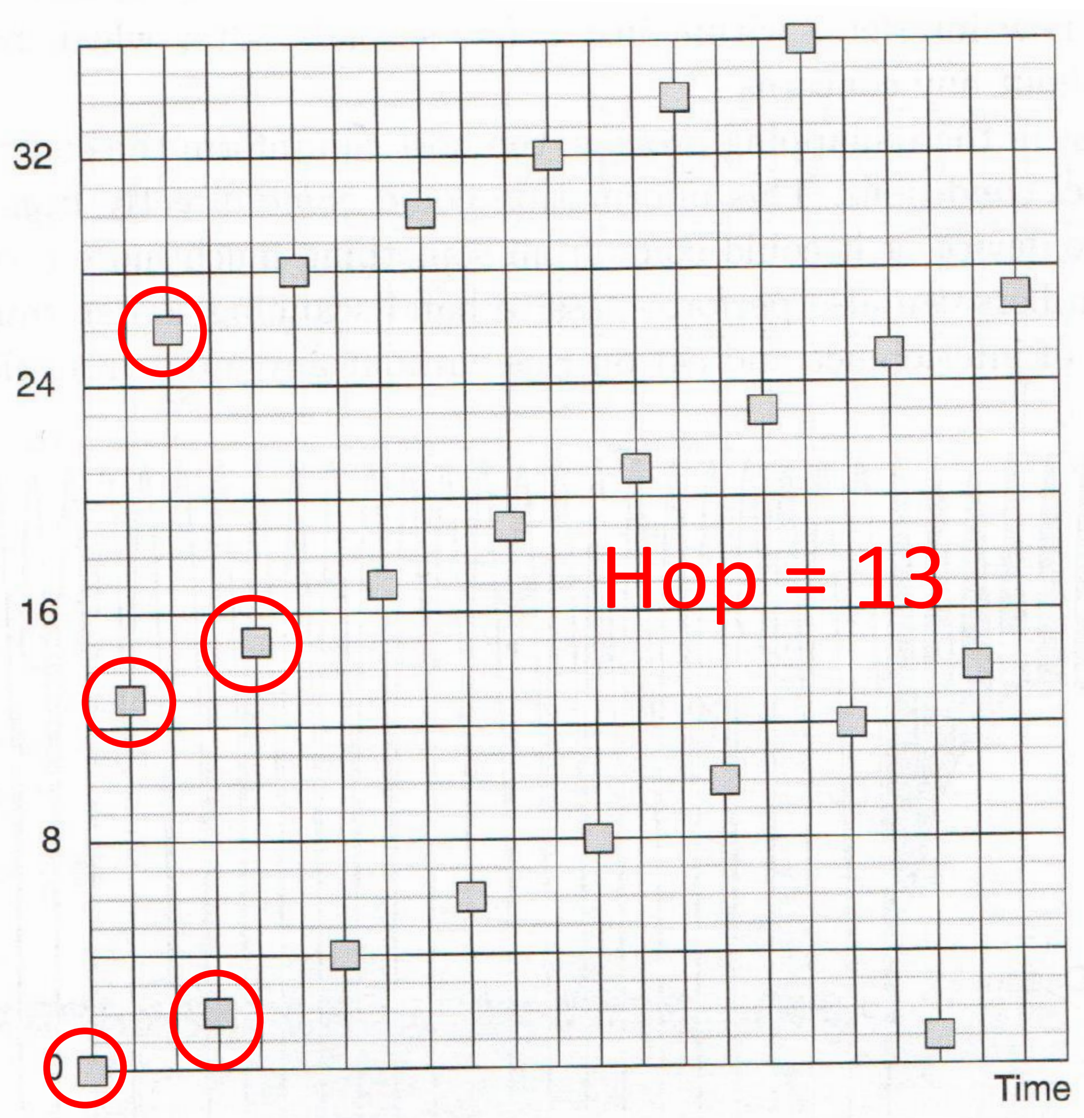


Figure 7–14 Frequency hopping of data channels over time

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

f_{n+1} = (f_n+hop) mod 37

- [x] n = 6
- [ ] n = 1
- [ ] n = 2
- [ ] n = 5
- [ ] n = 4
- [x] n = 3

F(0) = 9
F(1) = 23
F(2) = 0
F(3) = 14
F(4) = 28
F(5) = 5
F(6) = 19

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

f_{n+1} = (f_n+hop) mod 37

- [ ] n = 3
- [ ] n = 1
- [ ] n = 4
- [x] n = 2
- [ ] n = 6
- [x] n = 5

F(0) = 8
F(1) = 20
F(2) = 32
F(3) = 7
F(4) = 19
F(5) = 31
F(6) = 6

Quiz 6 review

$$f_{n+1} = (f_n + \text{hop}) \bmod 37$$

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 = 7
- hop = 14
- hop = 12
- ✔ hop = 10

F(0) = 7
F(8) hop 7 = 26
F(8) hop 10 = 13
F(8) hop 12 = 31
F(8) hop 14 = 8

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 = 13
- ○ hop = 7
- ○ hop = 10
- ✓ hop = 14

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 that apply to Bluetooth Classic

- [x] Each time slot is equal to 625uS
- [x] The Bluetooth radio hops 1600 times per second
- [ ] The 2.4GHz RF band is broken into 40 2MHz channels
- [ ] Master transmits on odd time slots

Quiz 6 review

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

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

Quiz 6 review

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

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

Quiz 6 review

What are the primary methods that the Link Layer reduces power?

- [x] Keeping data packets short
- [x] Using offline encryption
- [x] Using a high physical bid rate
- [x] Single-channel connection events

Quiz 6 review

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

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

Quiz 6 review

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

Quiz 6 review

Fundamentally, Bluetooth Smart is very **(simple, Simple.)** (single word answer).

Quiz 6 review

How long is the shortest data packet in Bluetooth Low Energy?

- 80uS
- 128uS
- 144uS
- 376uS

Quiz 6 review

How long is the typically advertising packet in Bluetooth Low Energy?

- 80uS
- 128uS
- 144uS
- 376uS

Quiz 6 review

A service is an (immutable encapsulation) of some atomic behavior of a device.

Quiz 6 review

(Encapsulation) means expressing features of something succinctly.

Quiz 6 review

(Immutable) means that once a service is published, it cannot change.

Quiz 6 review

In Bluetooth Low Energy, ______ means of or forming a singe irreducible unit of component of a larger system.

(atomic, atomic behavior, atomic operation, atomic operations)

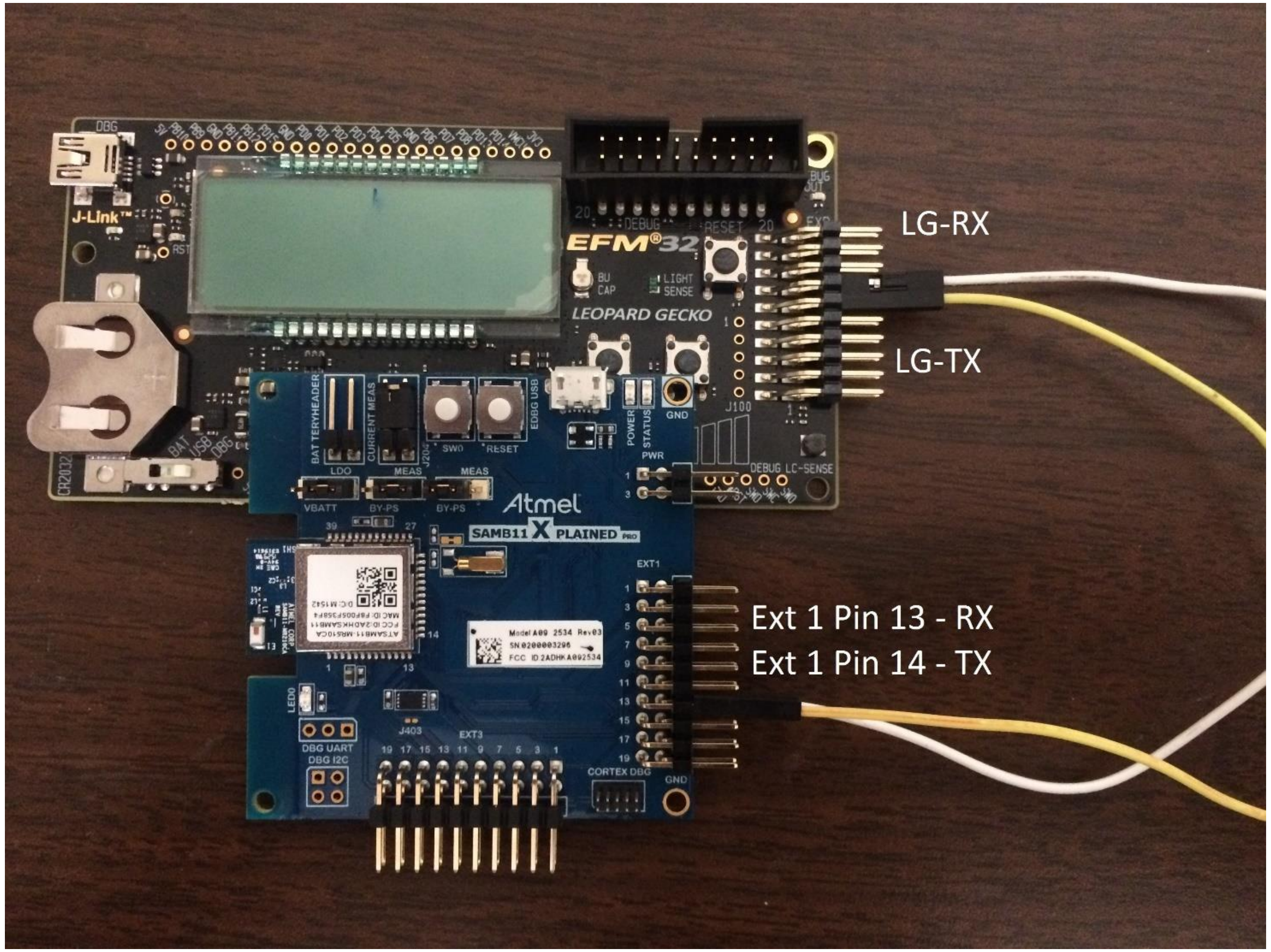
Quiz 6 review

The ultimate embodiment of a use case or application is a (profile, Profiles., profiles) (single word answer).

Planned assignments before/start of Course Project

- ATSAMB11 tutorial – not an assignment, but important for the following project assignments
- Assignment #5: Interfacing the ATSAMB11 and the Leopard Gecko using a UART interface
  - The values from the Leopard Gecko will be provided to the ATSAMB11 to include as the data for the Health Thermometer Service
  - 7-14 days will be planned for this assignment

Planned assignments before Course Project



Planned assignments before Course Project

- ATSAMB11 tutorial – not an assignment, but important for the following project assignments
- Assignment #5: Interfacing the ATSAMB11 and the Leopard Gecko using a UART interface
  - The values from the Leopard Gecko will be provided to the ATSAMB11 to include as the data for the Health Thermometer Service
  - 7-10 days will be planned for this assignment
- Assignment #6: Use a Smart Phone to turn on and off the Leopard Gecko LED
  - 5-7 days will be planned for this assignment

Course Project

- Utilizing both the Silicon Labs’ Leopard Gecko and the Atmel ATSAMB11 together in a Bluetooth Smart device application
  - The ATSAMB11 will be the master processor of the system
  - And, the Leopard Gecko will be the sensor hub
- The communication between the two devices will be a UART port
  - On the Leopard Gecko, the LEUART interface running in DMA mode

Course Project

- Minimum sensors required to operate on the Leopard Gecko are the following:
  - Internal Temperature Sensor
  - LESENSE – Ambient Light Sensor
  - LESENSE – Capacitive Sensor
  - I2C device other than the TSL2651
    - A selection of devices will available
    - Suggestions can be posted on the slack channel – courseproject

Course Project

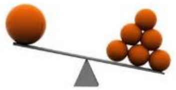
- Atmel ATSAMB11
  - will need to provide updates to a Smart Phone or tablet of sensor data
  - Receive commands to turn on or off the LED on the Leopard Gecko STK3600 starter kit
  - Change the security settings of the BLE connection
  - Modify the BLE connection interval, slave latency, and TX power to meet your project objectives
- Preliminary: The project will be graded on:
  - Functionality
  - Low power design of the project
  - And, the cohesive nature of how all the sensors and peripherals interact

Course Project

- Project kickoff
  - Expected to be Thursday, March 9th
- Due date:
  - Last week of semester - students will be signing up for demo times
- More details to follow

BLE: Peripheral (Connection)

- IMPORTANT: Slave Latency also sets the latency of the central device to the peripheral
  - For example: If a keyboard has an LED indicator that needs to be turned on within 500mS for the desired user experience, the connInterval X slavelatency must be less than 500mS



  - The slave has the option of jumping onto a connection event early, but the master does not!

BLE: Peripheral (Connection)

- IMPORTANT: Extremely long Slave Latency can actually consume more energy than they actually save due to the accuracy of the central and peripheral clocks
  - In worst case, the clock inaccuracies could be 500 parts per million on each device, and possibly in opposition directions
  - If a device was set with a connection interval of 15mS and a slave latency of 500, the slave only would be required to listen every 7.5s
  - At the end of 7.5s, the two devices could be out of synch by 3.75mS and possibly up to 7.5mS if the two devices are off in different directions
  - To resolve this possible timing synchronization issue, the peripheral would need to begin to listen 7.5mS before its expected connection time and possibly 7.5mS afterwards . This is called window widening.
    - For every 7.5s, the peripheral would have to listen from 7.5 to 15.0mS to synchronize
    - Resulting in significant energy loss

BLE: Peripheral (Connection)

- For practical purposes in terms of saving energy, it does not make sense to set the slave latency to have a maximum time between connections greater than 1s or fewer than 300mS
  - Below 300mS, the power used to repeatedly synchronize is higher than it would be to wait long
  - Above 1s, the power used by window widening does not save any significant amount of power, and the user experience is enhanced with a smaller slave latency

BLE: Peripheral (Stay Connected or Disconnect)

- Two main questions to answer:
  - Can the central device reconnect back to the peripheral in a reasonable latency if the peripheral starts to advertise?
  - If the peripheral does stay connected, can the peripheral inquire the connection latency being used or ask for a connection latency to enable an acceptable battery life?
- The peripheral can obtain from the central device the connection latency that it will honor to the peripheral when reestablishing a connection if the peripheral exposes the Scan Parameters Service
  - The central device that connects to this peripheral will discover the scan parameter service and provide to the peripheral its latency

BLE: Peripheral (Stay Connected or Disconnect)

- Example 1 disconnect:
  - Connection latency written to the peripheral from the central device is 100mS
  - The peripheral only requires a latency of 250mS
  - Theoretically, the peripheral could disconnect any time and reconnect within its 250mS requirement
  - Disconnecting saves power, but reconnect consumes lots of power
  - Within a 100mS connection latency, the peripheral will spend an average of 50mS to make the connection
  - During these 50mS, the peripheral will be sending Directed Advertising Packets 176 uS packet every 1.25mS resulting in the radio being powered for 8.64mS

BLE: Peripheral (Stay Connected or Disconnect)

- Example 1 stay connected:
  - Connection Interval is set to 250mS
    - If no data was to be sent, the radio would be on for 640uS every second
  - How long would you have to remain connected without sending data before it would become more energy expensive than reconnecting?
    - $\frac{Active\ Radio\ time\ to\ reconnect\ in\ uS}{Active\ Radio\ time\ in\ uS\ per\ second}$
    - $\frac{8640uS}{640uS}$ = 13.5s

BLE: Peripheral (Stay Connected or Disconnect)

- Example 1 disconnected, but connectable:
  - While disconnected, assume the peripheral wants to be connectable, so it advertises slowly, once every 1s
    - Advertising on all three advertising channels every second powers up the radio for 504uS per second
  - How long would you have to remain connected without sending data before it would become more energy expensive than reconnecting?
    - $\frac{Active\ Radio\ time\ to\ reconnect\ in\ uS}{Disconnected\ \&\ Advertising\ in\ uS\ per\ second}$
    - $\frac{8640uS}{504uS}$ = 17.1s


kasa
kasa
SMART LIGHTING
veho.
powered by veho.
0:07 / 1:55
CC
HD

Bluetooth-Classic: Profiles

- High level description to differentiate between Bluetooth-Classic and BLE profiles
- Why are there Bluetooth-Classic profiles?
  - They provide an interoperability between the master and slave
  - For example, enabling a Bluetooth-Classic headset to work with any Bluetooth-Classic or dual-mode phone
- How does the Bluetooth-Classic profile enable interoperability?
  - Clearly defines and states the responsibility of the master and its commands to the slave within a given profile
  - Clearly defines and states the responsibility of the slave and how it responses to the master within a given profile

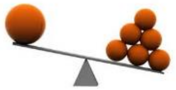
Bluetooth-Classic: Profiles

- Are there any drawbacks to the Bluetooth-Classic Profiles?
  - It does not allow, or at least easily, the change of roles or use cases
  - For example:
    - Bluetooth-Classic headsets support the Headset Profile (HSP) which enables interoperability with all Bluetooth-Classic phones
    - As an audio engineer, you discover a new way to send data to the headset that would increase audio fidelity
    - You develop the code on your phone, but since the definition of how the master (phone) operates with the slave (headset) is defined, your new and improved communications scheme will not work on any of the older HSP headsets
    - You will need to convince the headset manufacturers to support a new profile to enable your improved product to market

BLE: Profiles

- First, what is BLE service?
  - Defined state information on a server
  - Standard defined services on the server are immutable
- What are BLE profiles?
  - Client defined use of server services
  - The profile could use multiple server services or services across multiple servers
  - Note: No specification of what the server must do in support of a BLE profile

BLE: Profiles

- What is the advantage of moving the responsibility of the profile from both end points to the client?
  - Enabling the server to be used in a “limitless” number of profiles that exist today and in the future
  -  Minimize the code and responsibility of the server to save energy on the resource limited device
- For example:
  - A device provides the following services:
    - Temperature
    - Air quality
  - The client uses this devices services with a profile to provide an application with data on the temperature and air quality of a particular room in a building
  - Someday in the future, a new profile could be developed on the client that could take these device services and make it a fire or smoke alarm
- Moving the role to the client enables servers to be used in new roles that may not even be thought of today

BLE: Central (Discovering Devices)

- The first thing that a newly commissioned central device will do is to discover other devices
  - Passive Scanning: a central device passively listens to advertisement packets that peripherals are transmitting
  - Active Scanning: a central device, after hearing a peripheral, asks for more information
- If the Central device is only looking for what devices are around, such as when you open your Atmel Smart Connect application on your mobile device, is should only use passive scanning
  - Reduces the energy of the central and peripheral devices
  - If active scanning is used, the peripheral will need to listen to the central device and respond to request which will increase the radio active time consuming energy

BLE: Central (Discovering Devices)

- If the Central device is also populating a user interface with additional information on each device advertising, then active scanning should be used
  - Information that can be found in active scanning includes:
    - Name of the device
    - A unique number that identifies the device that can be used later to connect to the device
    - Discover some broadcast information data within the scan responses so that information that is being broadcasted can be obtained such as battery level or the current time

BLE: Central (Discovering Devices)

- During passive and active scanning, not only can the application obtain the contents of the advertising packets, but it can also receive the Received Signal Strength (RSSI) of these packets.
- If the Tx Power was included in the advertising packets, a basic estimate of the path loss and therefore an estimate of the distance between the device and central device can be determined
  - $path\ loss = TxPower\ - RSSI$
  - If the path loss is very small, between 0 and 20, the device is very close
  - If the path loss is very high, greater than 70, then the device is very far away
  - To eliminate multipath interference, these values should be averaged over a number of seconds

BLE: Central (Connecting to Devices)

- When initiating a connection, a set of connection parameters will need to be chosen depending on what the two devices are intending to do
- Typically, peripherals have a Client Preferred Parameters characteristic that gives a very strong hint to a central device about the type of connection parameters it prefers
  - Connect interval
  - Slave Latency
  - Etc
- When making the first connection with a device, this information is not available, so the central device should compromise between low power consumption and rapid characterization of a device

BLE: Central (Connecting to Devices)

- An example of a compromise would be a connection interval of 15mS to 30mS and a slave latency of 150mS
  - Allows both rapid collection of data about the peripheral using up to >60Hz connection intervals and a possible slave idle frequency of >6Hz
- The slave might request different parameters from those that an application on the central device has chosen after connection has been made
  - For example:
    - The peripheral may request a connection interval of 150mS and a slave latency of 750mS to minimize energy use
    - But, the application on the central device might require the data or state from the peripheral every 50mS, so the central sets up connection interval of 50mS and slave latency set to 0
    - The application will get the data it requires at the appropriate data rate, but the peripheral battery life will be negatively impacted

BLE: Central (What does this device do?)

- After connecting to the peripheral, the central device will need to discover what the device does using the following four procedures:
  - Primary Services Discovery
  - Relationship Discovery
  - Characteristic Discovery
  - And, Descriptor Discovery
- The first process is the Primary Services Discovery
  - These are the services that describe what the device does
  - For example:
    - If the device has a battery, the primary services would expose the Battery Service
    - If the device has a temperature sensor, the primary services would expose the Temperature Service
    - If the device had a temperature sensor within the battery, this secondary service of the battery would not be exposed through Primary Services Discovery