ECEN 5823-001 / -001B

Internet of Things Embedded Firmware

Lecture #8

21 September 2017

Be Boulder.

University of Colorado Boulder

Agenda

- Class announcements
- Review updated Energy Mode Rubrix
- BMA280 – Accelerometer / Tap Sensor assignment
- Low Energy Sensors and Low Energy Sensor Communication buses
- Bluetooth Classic

Class Announcements

- Quiz #4 is due at 11:59pm on Sunday, September 24th, 2017
- SPI tap sensor Assignment is due at 11:59pm on Wednesday, September 27th, 2017

Managing Energy Mode Review

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Current consumption in EM0 Active mode with all peripherals disabled	I_{ACTIVE}	38.4 MHz crystal, CPU running while loop from flash[1]	—	130	—	µA/MHz
		38 MHz HFRCO, CPU running Prime from flash	—	88	—	µA/MHz
		38 MHz HFRCO, CPU running while loop from flash	—	100	105	µA/MHz
		38 MHz HFRCO, CPU running CoreMark from flash	—	112	—	µA/MHz
		26 MHz HFRCO, CPU running while loop from flash	—	102	106	µA/MHz
		1 MHz HFRCO, CPU running while loop from flash	—	222	350	µA/MHz
Current consumption in EM1 Sleep mode with all peripherals disabled	I_{EM1}	38.4 MHz crystal[1]	—	65	—	µA/MHz
		38 MHz HFRCO	—	35	38	µA/MHz
		26 MHz HFRCO	—	37	41	µA/MHz
		1 MHz HFRCO	—	157	275	µA/MHz

- Question 1: EM0
  - Energy score: 2.5-2.7
  - Current LED off: ~2.3 - ~2.9mA
  - Current LED on: current in (ii) plus 0.45 to 0.55mA
- Question 2: EM1
  - Energy score: 2.7-2.9
  - Current LED off: ~1.6 - ~2.1mA
  - Current LED on: current in 2(ii) plus 0.45 to 0.55mA
- Question 3: EM2
  - Energy score: 5.2-5.4
  - Current LED off: ~1.5 - ~2.1uA
  - Current LED on: current in 3(ii) plus 0.45 to 0.55mA
- Question 4: EM2: Period and On-time
  - Period: 2.48 – 2.52 S
  - On-time: 49.75 – 50.25 mS
- Question 5: EM3
  - Energy score: 5.2-5.4
  - Current LED off: should be ~1 to 1.4uA less than 3(ii)
  - Current LED on: current in 5(ii) plus 0.45 to 0.55mA
- Question 6: EM3: Period and On-time
  - Period: 2.48 – 2.52 S
  - On-time: 49.75 – 50.25 mS

Managing Energy Mode Review

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Current consumption in EM0 Active mode with all peripherals disabled	I_{ACTIVE}	38.4 MHz crystal, CPU running while loop from flash[1]	—	130	—	µA/MHz
		38 MHz HFRCO, CPU running Prime from flash	—	88	—	µA/MHz
		38 MHz HFRCO, CPU running while loop from flash	—	100	105	µA/MHz
		38 MHz HFRCO, CPU running CoreMark from flash	—	112	—	µA/MHz
		26 MHz HFRCO, CPU running while loop from flash	—	102	106	µA/MHz
		1 MHz HFRCO, CPU running while loop from flash	—	222	350	µA/MHz
Current consumption in EM1 Sleep mode with all peripherals disabled	I_{EM1}	38.4 MHz crystal[1]	—	65	—	µA/MHz
		38 MHz HFRCO	—	35	38	µA/MHz
		26 MHz HFRCO	—	37	41	µA/MHz
		1 MHz HFRCO	—	157	275	µA/MHz

Low Energy ADC Rubric

Question scoring. Max score is 5.0 pts.

- a. Question 1:
  - i. Energy mode: EM3 (0.4 pts)
  - ii. Current LED0 off: 250uA-350uA (0.8 pts)
    1. 350uA – 400uA (0.7 pts)
    2. 400uA – 450uA (0.6 pts)
  - iii. Current for peroid: < 350uA (0.6 pts)
    1. < 450uA (0.5 pts)
  - iv. Energy Score: 3.9 – 4.1 (0.4 pts)
- b. Question 2:
  - i. Current LED0 off: < 675uA (0.7 pts)
    1. < 775uA (0.6 pts)
    2. < 850uA (0.5 pts)
  - ii. Energy score: 3.3 – 3.5 (0.4 pts)
- c. Question 3:
  - i. Current LED0 off: 250uA – 350uA (0.4 pts)
    1. 350uA – 400uA (0.3 pts)
    2. 400uA – 450uA (0.2 pts)
  - ii. Current for peroid: < 400uA (0.3 pts)
    1. < 550uA (0.2 pts)
  - iii. Energy Score: 3.7 – 3.9 (0.1 pts)
  - iv. On-time: 675-725mS (0.3 pts)
- d. Question 4:
  - i. Current LED0 off: 250uA – 350uA (0.4 pts)
    1. 350A – 400uA (0.3 pts)
    2. 400uA – 450uA (0.2 pts)
  - ii. Current for peroid: < 350uA (0.3 pts)
    1. < 450uA (0.2 pts)
  - iii. Energy Score: 3.9 – 4.1 (0.1 pts)
  - iv. On-time: 175-225mS (0.3 pts)

Low Energy ADC Rubric

3. Functional code delivered per exercise. Max score is 5.0 pts.
   a. Change the period to 5 secs (total of 2.0 pts)
      i. Does the period change to 5 seconds? (1.0 pts)
      ii. Does the on-time remain at 200mS? (0.5 pts)
      iii. When increasing the LED on time by 0.5S, does it change to 0.7s (0.5 pts)

   b. Code functions in EM3 correctly from reset (total of 3.0 pts)
      i. LED1 turns on with north click of joy stick (0.5 pts)
      ii. LED1 turns off with south clock of joy stick (0.5 pts)
      iii. LED0 on-time increases by 0.5S with east click of joy stick (0.5 pts)
      iv. LED0 on-time decreases by 0.5S with west clock of joy stick (0.5 pts)
      v. East click joy stick so LED0 is 0.7S, does LED0 reset to 0.2S with push of joy stick (1.0 pts)

IP credit is not given to Silicon Labs for sleep routines (-1.0 pts)

Lessons Learnt 1

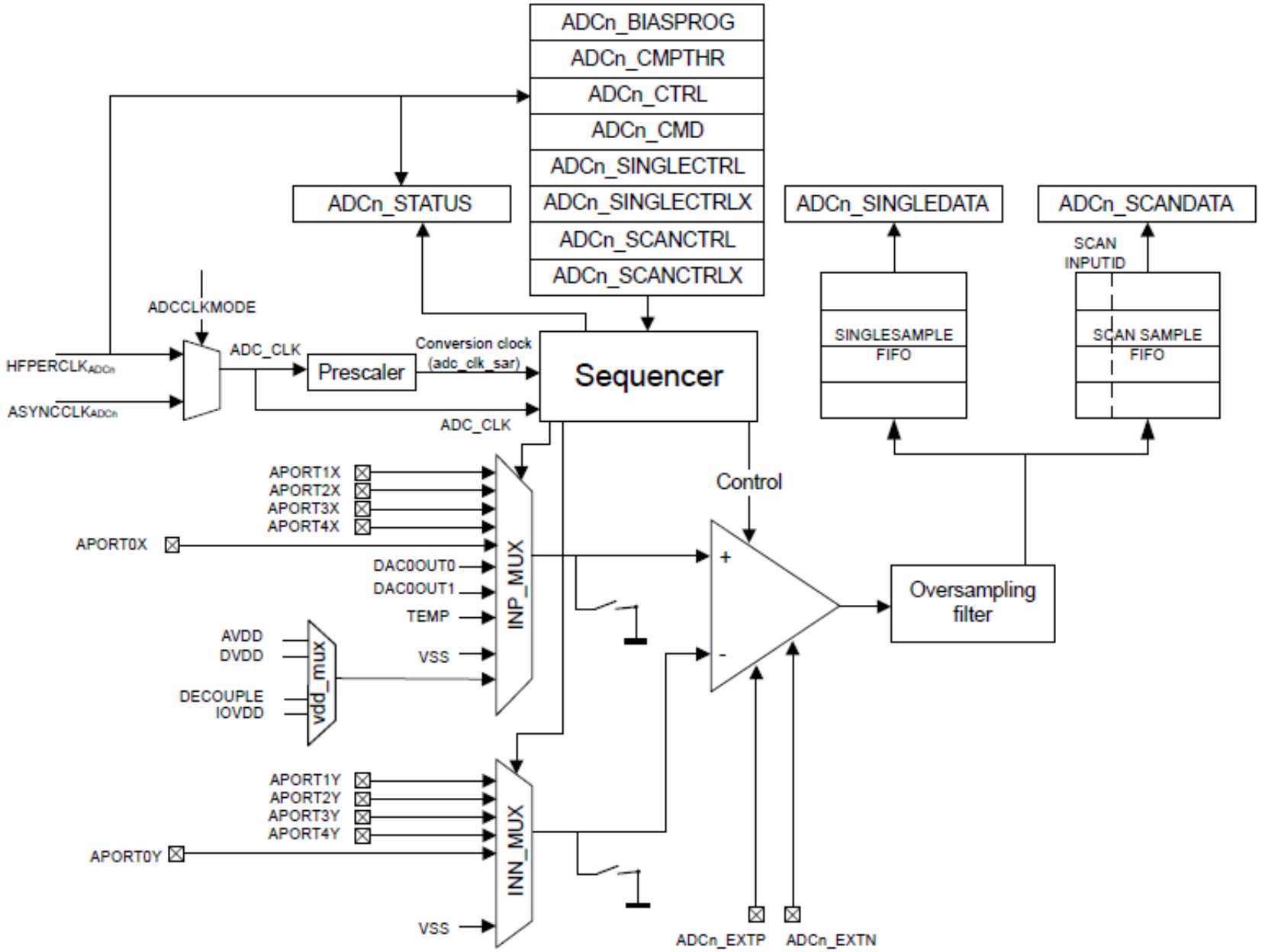


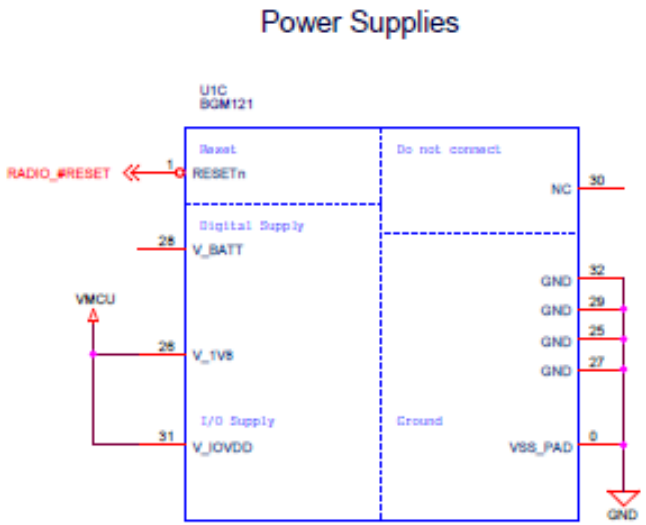
Figure 24.1. ADC Overview

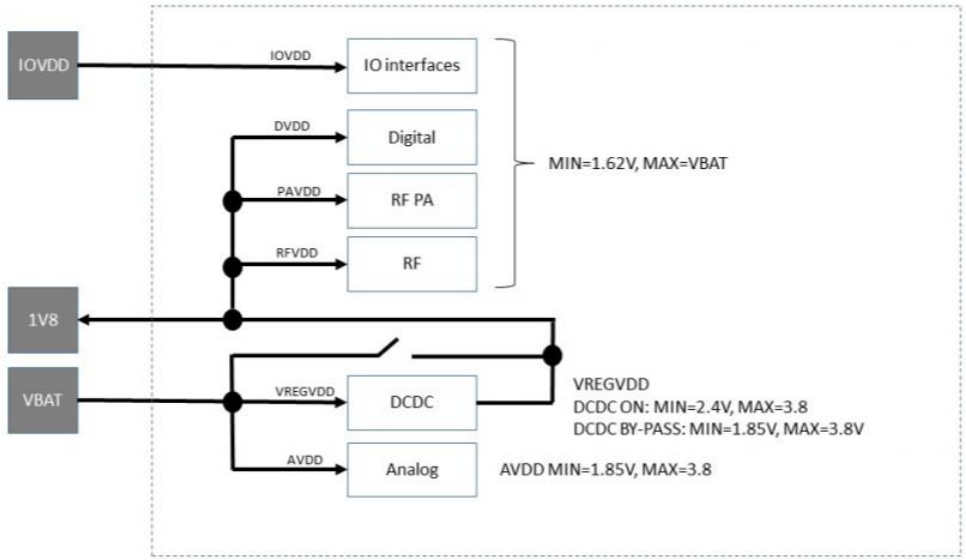
Lessons Learnt 1

- Debouncing a ADC Compre logic issue
  - While looking for the Joy Stick to be asserted and receiving the Window Inside interrupt
    - Stop the ADC
    - Change the Window compare to look outside the window instead of inside
    - Re-enable the ADC
    - Clear the interrupt
  - When receiving the interrupt while looking for an interrupt outside the window
    - Stop the ADC
    - Change the Window compare to look inside the window instead of outside
    - Re-enable the ADC
    - Clear the Interrupt
  - What else is required to make this work?

Lessons Learnt 2 – Is Vadd or Vbgr better to use?

- Two possible ADC references to measure a 3.3v input voltage using the full range of the ADC





- First: VFS = AVDD using AVDD as the reference source (AVDD ≤ 3.6 V)

Lessons Learnt 3

- Utilize the peripheral Type_Init_Def structures and the Peripherel_Init() functions
  - Minimizes mistakes in programming the peripheral
  - Minimizes missing to set up a value in the peripheral

Additional comments on coding style

For the instructing team to provide help and suggestions with programming assignments, the following documentation must be included. Without this documentation, it is very difficult and time consuming to review and provide help.

1. No use of Init TypeDef defaults. The constructs must be fully specified by one of the following methods. The instructing team does not memorize or know the defaults of the Init TypeDef defaults.

```
LETIMER_Init_TypeDef    LETIMER0_init;
int intFlags;
int Comp0_init;
int Comp1_init;
int LETIMER0_prescaler;
int ULFRCO_count_calibrated;

LETIMER0_init.bufTop = false;
LETIMER0_init.comp0Top = true;
LETIMER0_init.debugRun = false;
LETIMER0_init.enable = false;
LETIMER0_init.out0Pol = 0;
LETIMER0_init.out1Pol = 0;
LETIMER0_init.repMode = letimerRepeatFree;
LETIMER0_init.rtcComp0Enable = false;
LETIMER0_init.rtcComp1Enable = false;
LETIMER0_init.ufoa0 = letimerUFOANone;
LETIMER0_init.ufoa1 = letimerUFOANone;

LETIMER_Init(LETIMER0, &LETIMER0_init);
```

Or,

IoT Embedded Firmware – Fall 2017

```
/* Set configurations for LETIMER 0 */
 const LETIMER_Init_TypeDef letimerInit =
 {
 .enable         = true,
 .debugRun       = false,
 .rtcComp0Enable = false,
 .rtcComp1Enable = false,
 .comp0Top       = true,
 .bufTop         = false,
 .out0Pol        = 0,
 .out1Pol        = 0,
 .ufoa0          = letimerUFOAPwm,
 .ufoa1          = letimerUFOAPulse,
 .repMode        = letimerRepeatFree
 };

 /* Initialize LETIMER */
 LETIMER_Init(LETIMER0, &letimerInit);
```

2. No register bit manipulation with direct bit or hex values where appropriate and possible. Use of using the enumerations provided by the MCU/SOC provider to address bit names and fields. Similar to the use of Init TypeDef defaults, the instruction team does not memorize the value of each bit field in a register.

~~LETIMER0->IEN |= 0x00000004;~~ ~~Not acceptable~~
LETIMER0->IEN |= LETIMER_IEN_UF; Acceptable

SPI tap sensor assignment

ECEN 5823
SPI tap sensor Assignment
Fall 2017

Objective: Adding the BMA280 accelerator via the SPI bus and enabling / disabling the BMA280 to implement load power management.

Note: This assignment will begin with the completed Low Energy ADC assignment.

Instructions:

1. Make any changes required to the Low Energy ADC assignment.
2. Connect the STK6101C extension board to the main development kit board.

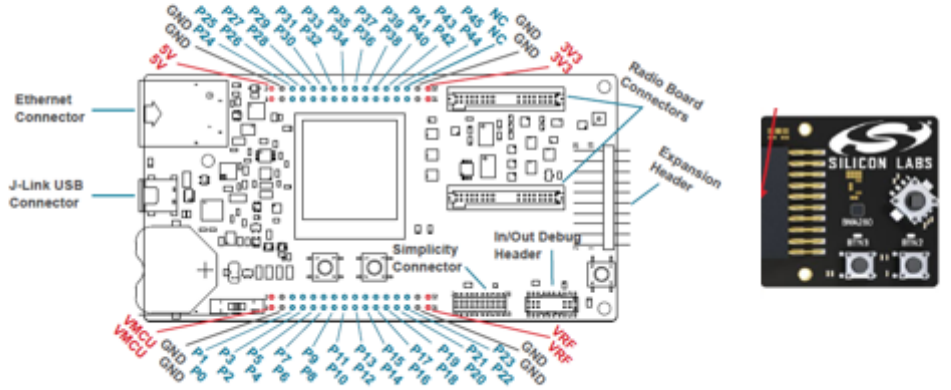


Figure 3.1. Mainboard Connector Layout

Setting up the TIMERn

- Third, the SPI / USARTn interrupts must be enabled if needed
  - Not necessarily with a peripheral that will be load power managed
    - Enable the Interrupts when you Load Power Manage ON the peripheral

Setting up the TIMERn

- Forth, the SPI / USARTn buffers must be reset
  - Clear out the transmit and receive buffers before enabling the SPI / USARTn peripheral to insure that the buffers are enable for this new instantiation of the SPI / USARTn
    - `spi->CMD`
  - With the buffers empty, enable the SPI / USART peripheral
    - `USART_Enable(USARTn, usartEnable);`

Setting up the USARTn

- Fifth, the USARTn interrupt handler must be included
  - Routine name must match the vector table name:

    ```
    Void USARTn_RX_IRQHandler(void) {
    }
    ```

    And,

    ```
    Void USARTn_TX_IRQHandler (void) {
    }
    ```

  - Inside this routine, you add the functionality that is desired for the USARTn interrupts

https://www.kickstarter.com/projects/802159142/nuimo-seamless-smart-home-interface?ref=discovery

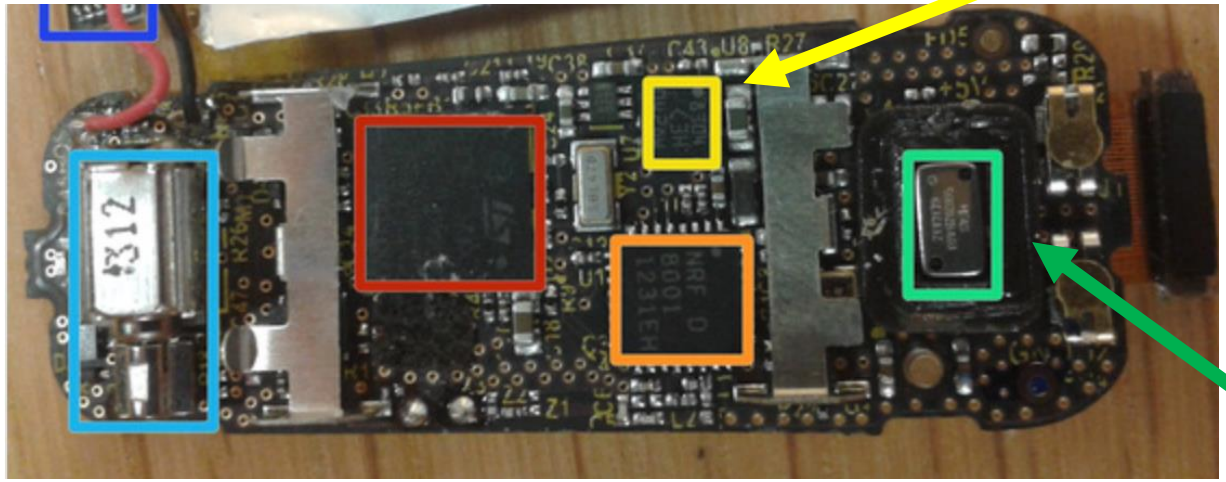
Sensors – For the low energy market

- Key Factors in a mobile sensor selection
  - End Product Features
    - What is the desired user experience?
    - What sensor features are required?
    - What is the energy budget for the product?
  - System considerations
    - What microcontroller interfaces available?
      - Analog IN
      - I2C
      - SPI
      - In what power mode?
    - What is the energy budget for the sensor?

Sensors – End Product Features

- What is the desired user experience?
  - What sensors are required to provide the user experience?
    - Example: Fitbit Odometer – Measures steps and staircases climbed
      - Sensors – Accelerometer and Altimeter

Accelerometer labelled 8304 AE D42 oW



Picture and p/ns from iFixIt

Measurement specialties MS5607-02BA03 altimeter

Sensors – End Product Features

- What sensor features are required?
  - To achieve the desired features, what are the required sensor specifications for the application?
  - Example: Accelerometer
    - Number of axis: 2, 3, etc.
    - Resolution: 10-bits, 12-bits, 14-bits, 16-bits, etc.
    - Range: +-2g, +-4g, +-8g, +-16g, etc.
    - Update rate: 1.25Hz, 5Hz, 10Hz, 20Hz, 40Hz, 400Hz, etc.

Freescale MMA8452Q, 3-Axis, 12-bit/8-bit Digital Accelerometer

Features

- 1.95V to 3.6V supply voltage
- 1.6V to 3.6V interface voltage
- ±2g/±4g/±8g dynamically selectable full-scale
- Output Data Rates (ODR) from 1.56 Hz to 800 Hz
- 99 µg/√Hz noise
- 12-bit and 8-bit digital output
- I^2C digital output interface
- Two programmable interrupt pins for six interrupt sources
- Three embedded channels of motion detection
  - Freefall or Motion Detection: 1 channel
  - Pulse Detection: 1 channel
  - Transient Detection: 1 channel
    - Orientation (Portrait/Landscape) detection with set hysteresis
    - Automatic ODR change for Auto-WAKE and return to SLEEP
    - High-Pass Filter Data available real-time
    - Self-Test
    - RoHS compliant
    - Current Consumption: 6 µA to 165 µA

Sensors – End Product Features

- What is the energy budget for the product?
  - Example: Fitbit Surge Watch
    - Battery life: last up to 7 days
    - GPS Battery life: last up to 10 hours
    - Battery type: Lithium-polymer
    - Charge time: One to two hours

Fitbit Surge photo & specifications from Fitbit Surge website

Sensors – System considerations

- What is the energy budget for the sensor?
  - Can drive the decision between active and passive sensor

Table 5. DC Characteristics
(Typical Operating Circuit, V_{DD} and V_{REG} = 1.8V, T_A = 25°C, unless otherwise noted.)

Parameter	Symbol	Conditions	Min	Typ	Max	Units
High Supply Voltage	V_{DD}		2.0	3.3	3.6	V
Low Supply Voltage	V_{REG}		1.71	1.8	2.75	V
Average Supply Current(1)	I_{DD}	Run Mode @ 1 ms sample period		393		µA
		Run Mode @ 2 ms sample period		199		µA
		Run Mode @ 4 ms sample period		102		µA
		Run Mode @ 8 ms sample period		54		µA
		Run Mode @ 16 ms sample period		29		µA
		Run Mode @ 32 ms sample period		17		µA
		Run Mode @ 64 ms sample period		11		µA
		Run Mode @ 128 ms sample period		8		µA
Measurement Supply Current	I_{DD}	Peak of measurement duty cycle		1		mA
Idle Supply Current	I_{DD}	Stop Mode		3		µA

Sensors – System considerations

- What microcontroller interfaces available?
  - Most common sensor interfaces
    - Analog In for passive sensors
      - LESENSE interface
    - Analog In for some active sensors such as audio and ambient light sensors
  - Active Sensors
    - I2C
    - SPI
    - UART

- **Communication interfaces**
  - 3× Universal Synchronous/Asynchronous Receiver/Transmitter
    - UART/SPI/SmartCard (ISO 7816)/IrDA/I2S
  - 2× Universal Asynchronous Receiver/Transmitter
  - 2× Low Energy UART
    - Autonomous operation with DMA in Deep Sleep Mode
  - 2× I^2C Interface with SMBus support
    - Address recognition in Stop Mode
  - Universal Serial Bus (USB) with Host & OTG support
    - Fully USB 2.0 compliant
    - On-chip PHY and embedded 5V to 3.3V regulator
- **Ultra low power precision analog peripherals**
  - 12-bit 1 Msamples/s Analog to Digital Converter
    - 8 single ended channels/4 differential channels
    - On-chip temperature sensor
  - 12-bit 500 ksamples/s Digital to Analog Converter
  - 2× Analog Comparator
    - Capacitive sensing with up to 16 inputs
  - 3× Operational Amplifier
    - 6.1 MHz GBW, Rail-to-rail, Programmable Gain
  - Supply Voltage Comparator
- **Low Energy Sensor Interface (LESENSE)**
  - Autonomous sensor monitoring in Deep Sleep Mode
  - Wide range of sensors supported, including LC sensors and capacitive buttons

Communication List from Silicon Labs' EFM32LG data sheet

What Digital Serial Bus is the lowest energy?

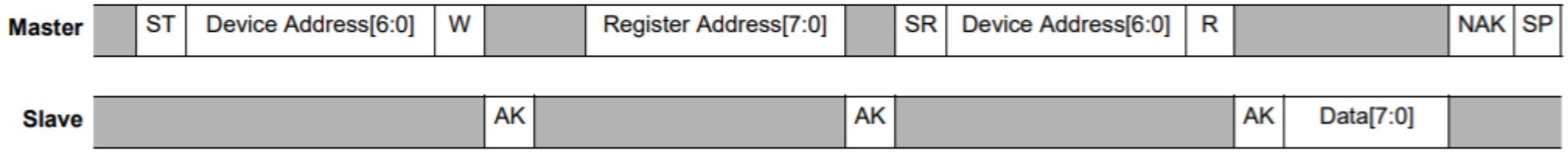
- The most common sensor buses are:
  - I2C
  - SPI
  - UART
- Energy = Power x Time

Comparing the Time of a single byte transfer

- I2C

I²C data sequence diagrams

< Single-byte read >



- A total of 40 cycles required for a single read transfer
- Time required:
  - 400KHz I2C = 100uS
  - 1,000KHz I2C = 40uS

Comparing the Time of a single byte transfer

- SPI
- 16 total cycles require for a single read transfer
- Time required:
  - 10,000 KHz = 1.6uS

The basic read operation waveform for 4-wire configuration is depicted in figure 15:

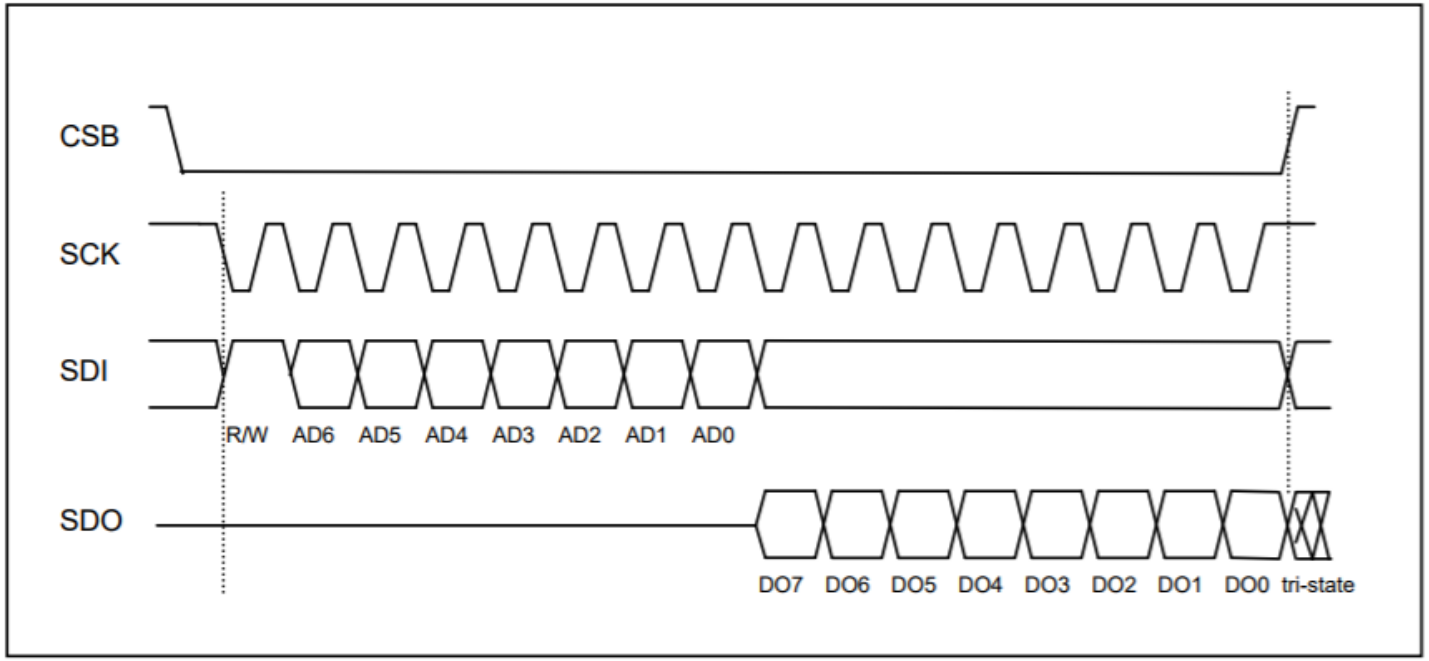


Figure 15: 4-wire basic SPI read sequence (mode ´11´)

Comparing the Time of a single byte transfer

- UART

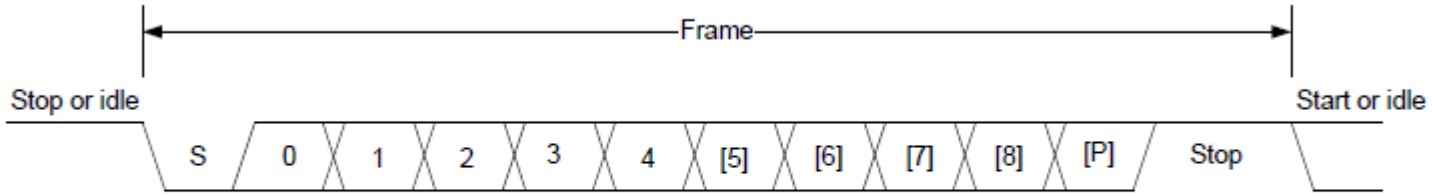


Figure 18.2. USART Asynchronous Frame Format

- A total of 11 cycles required for a single read transfer
- Time required:
  - 32.768KHz UART = 336uS
  - 115.2KHz UART = 95.5uS

Comparing the Power of a single byte transfer

- We will use the Silicon Labs Blue Gecko as a reference micro-controller due to its low energy design
- I2C can run as low as EM1 with DMA

Current consumption in EM1 Sleep mode with all peripherals disabled	I_{EM1}	38.4 MHz crystal[1]	—	65	—	µA/MHz
		38 MHz HFRCO	—	35	38	µA/MHz
		26 MHz HFRCO	—	37	41	µA/MHz
		1 MHz HFRCO	—	157	275	µA/MHz

- Standard HFXO crystal = 38.4MHz
  - Power = 65uA * 38.4 = 2.5mA
- Assume 50% of the time the pull ups are active (2 pull-ups: SCL and SDA)
  - 2* 0.50 * (3.3v / 10Kohm) = 0.330mA
- Total current:
  - EM1 + Pull ups = 2.83mA
- Total Power:
  - V * I = 3.3 * 2.83mA = 9.34mW

Comparing the Power of a single byte transfer

- We will use the Silicon Labs Blue Gecko as a reference micro-controller due to its low energy design
- SPI can run as low as EM1 with DMA

Current consumption in EM1 Sleep mode with all peripherals disabled	I_{EM1}	38.4 MHz crystal[1]	—	65	—	µA/MHz
		38 MHz HFRCO	—	35	38	µA/MHz
		26 MHz HFRCO	—	37	41	µA/MHz
		1 MHz HFRCO	—	157	275	µA/MHz

- Standard HFXO crystal = 38.4MHz
  - Power = 65uA * 38.4 = 2.5mA
- Total current
  - EM1 = 2.5mA
- Total power
  - V * I = 3.3 * 2.5mA =8.25mW

Comparing the Power of a single byte transfer

- We will use the Silicon Labs Blue Gecko as a reference micro-controller due to its low energy design
- LEUART can run as low as EM2 with DMA

Current consumption in EM2 Deep Sleep mode.	I_{EM2}	Full RAM retention and RTCC running from LFXO	—	3.3	—	µA
		4 kB RAM retention and RTCC running from LFRCO	—	3	6.3	µA

- HFXO oscillator has been turned off
- Total current
  - EM1 = 2.5mA
  - EM2 = 6.3uA
- Total Power
  - EM1 = 3.3v * 2.5mA = 8.25mW
  - EM2 = 3.3v * 0.0063mA = 0.02mW

What is the lowest Energy digital serial interface?

Digital Serial Interface	Power	Time	Energy
I2C @ 400KHz	9.34mW	100uS	934nJ
I2C @ 1,000KHz	9.34mW	40uS	374J
SPI @ 10,000KHz	8.25mW	1.6uS	13nJ
UART @ 115.2KHz	8.25mW	95.5uS	788nJ
LEUART @ 32.768KHz	0.02mW	336uS	7nJ

Other Digital Serial Bus Considerations

- I2C
  - Advantage: Addressable address bus up to 128 devices
    - Supports multiple sensors without increasing GPIO pin utilization or additional MCU resources
- SPI
  - Disadvantage: Requires additional GPIO pin for Chip Select for each additional addressable device
- UART
  - Disadvantage: Requires additional UART resource and GPIO pins for each device

Bluetooth Classic

- Perceived User Scenarios
  - Connection to peripheral devices
    - Wireless means no cables, and most likely battery operated
    - Low power wireless a must
  - Ad-hoc Networking
  - Bridging of Networks
    - Bluetooth has targeted lower cost, lower bandwidth applications
    - WiFi/WLAN designed for higher bandwidth, longer range, and larger devices

Bluetooth Classic – Technology Summary

- Globally free spectrum
  - 2.45 GHz, ISM band
  - GFSK modulation
  - Frequency Hopping (1600 hops/sec)
- Range
  - 10m piconet (0dBm)
  - 100m optional (+20dBm)
- Data and voice capable (1Mbps)
  - Full duplex: 478kbps, Asymmetric 721kbps
- Secure
  - Authentication
  - 128 Encryption
  - Limited Signal range 0 – dBm
  - Pseudo Random hop sequence

Bluetooth Classic - What does Bluetooth provide?

- Provides point-to-point connections.
- Provides ad-hoc networking capabilities.
- Bluetooth specification details how the technology works.
- Bluetooth Profiles detail how specific applications work to ensure interoperability.

Bluetooth Classic - Master /Slave Bluetooth Network Topology

- 1 master and up to 7 slaves
- Basic network structure – Star Network

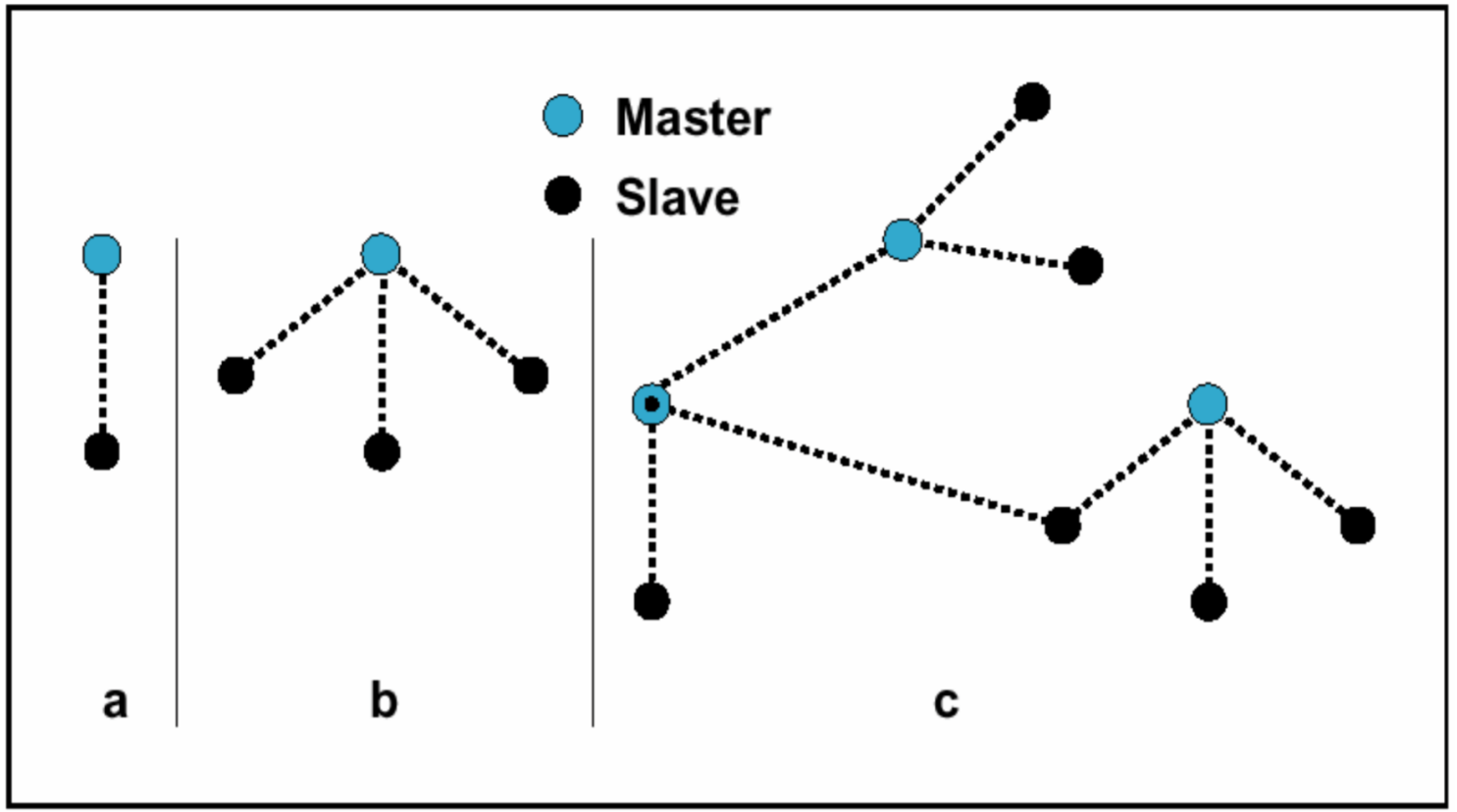
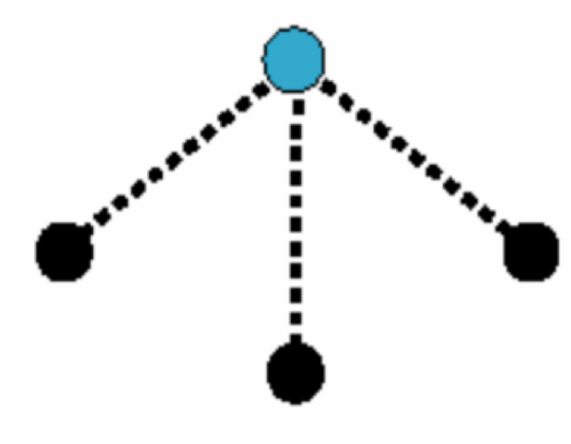


Figure 1.2: Piconets with a single slave operation (a), a multi-slave operation (b) and a scatternet operation (c).

Bluetooth Classic – Point-to-Point (Piconet)

- Two devices locate each other
- Form a connection and transfer data
- “Wireless cable replacement” scenario
- The device that initiates the connection is called the Master
- Any other devices the Master is connected to are referred to as Slaves.

Bluetooth Classic – Point-to-Multi-Point (The Piconet)

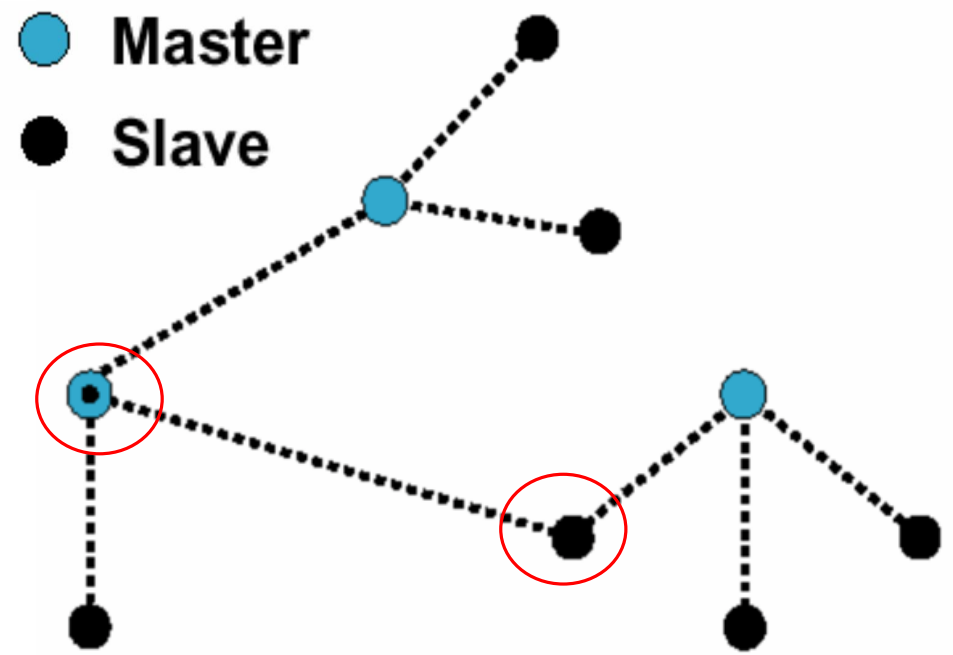


- Two devices create a point-to-point connection
- A third device comes into range
- The new device is discovered
- It is added to the piconet and data can be transferred
- Up to seven slaves can be connected to one master
- Slaves cannot pass data to other slaves without sending through the master
- The master defines the timing for the piconet
  - Each Piconet has a unique hopping pattern
- Piconets can collide if their unique hopping sequences overlap in a frequency band
  - Due to Ad-Hoc networking and not an infrastructure network!

IBM Bluetooth Wireless Technology presentation @ 2003
http://www.ncsysadmin.org/meetings/0506/slides.pdf
Mobile Communications: Second Edition by John Schiller

Bluetooth Classic - Piconet-to-Piconet: The Scatternet

- Scatternets allow devices to be active in numerous piconets
- The device can be a slave in one piconet and a master in another. It cannot be a master in two piconets which would result in two Piconets with the same frequency hopping sequence.
- The device can act as a gateway from one piconet to another.
- Before a slave leaves to participate in another Piconet, the slave must inform the current master that it will be unavailable for a period of time.
- As soon as a master leaves a Piconet to participate in another Piconet, all traffic within the original Piconet is suspended until the master returns.



IBM Bluetooth Wireless Technology presentation @ 2003
http://www.ncsysadmin.org/meetings/0506/slides.pdf
Mobile Communications: Second Edition by John Schiller

Bluetooth Classic – Identifying Bluetooth Devices

- Each Bluetooth device is assigned a unique 48-bit MAC address by the Bluetooth SIG
- This is enough addresses for 281,474,976,710,656 Bluetooth units, this should last a few years even with the optimistic predictions of the analysts!
- The address is split into three parts:
  - LAP: Lower Address Part - used to generate frequency hop pattern and header sync word.
  - UAP: Upper Address Part - used to initialize the HEC and lsb CRC engines.

Bluetooth Classic – Bluetooth Channels

- A master can create two types of logical channel with a slave device:
  - Asynchronous Connection Less (ACL): Packet Switched System provides a reliable data connection with a best effort bandwidth; depends on radio performance and number of devices in the piconet.
  - Synchronous Connection Oriented (SCO): Circuit Switched System provides real time reliable connection with a guaranteed bandwidth; usually used for voice based applications.
  - Would ACL or SCO be best for wireless speakers?
- The Bluetooth connections are limited to 1Mbps across the air
  - Giving a theoretical maximum of ~723kbps of useable data
  - Why is the theoretical maximum not 1Mbps?