ECEN 5823-001, -001B
Internet of Things Embedded Firmware

Lecture #2

31 August 2017

Be Boulder.

University of Colorado Boulder

Agenda

- Class Announcements
- Silicon Labs’ Exercise
- Low Power versus Low Energy Design
- What Makes a Low Energy Microcontroller
- Clock tree
- How to address register’s and their bits

Class Announcements

- Quiz #1 is due at 11:59pm on Sunday, September 3rd, 2017
- Survey is due at 11:59pm on Sunday, September 3rd, 2017
- Register for ESE lab card access at:
  - https://goo.gl/forms/8rTSW3hfbPUb4RFh1
- Waiting list is 4 students
  - Wait list students should be able to take the D2L quiz as well as submit assignments via the D2L drop box
  - Most likely will increase enrollment by 5 from the original total of 35
- Blue Gecko STK6101C development boards will be available after class

Simplicity Studio Exercise

Due: Wednesday, September 6th, 2017 at 11:59pm

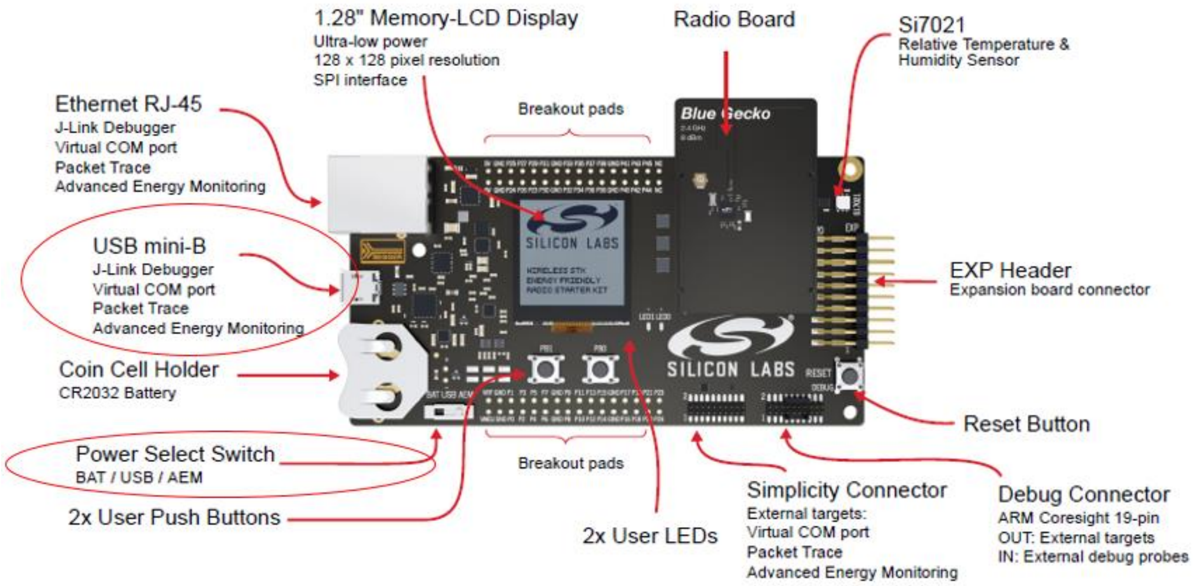
Objective: Install and become familiarized with the Silicon Labs' Simplicity Studio development environment

Due: Wednesday, September 6th, at 11:59pm

Instructions:

1. Install Silicon Labs' Simplicity Studio 4 development environment. You can download the software from the following site:
    a. http://www.silabs.com/products/mcu/Pages/simplicity-studio.aspx
    b. Insure that you select all EFR32 files as a minimum

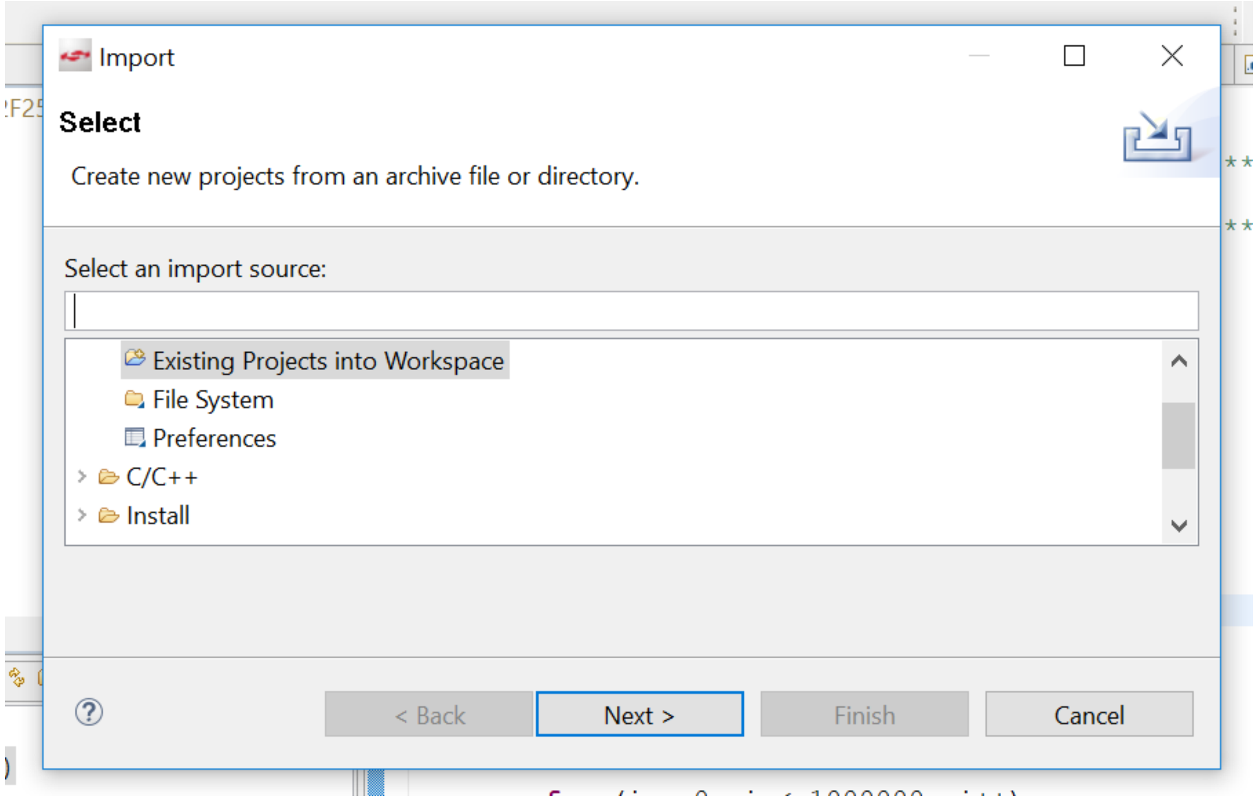
2. Connect your Silicon Labs' Blue Gecko SLWSTK6101C starter kit, USB Interface, to the computer



Silicon Lab SLWSTK6101CUser Manual

Simplicity Studio Exercise

3. Insure that the Power Source Select is to the AEM position
4. Select Import… under file, and highlight Existing Projects into Workspace before clicking on Next
5. Select the radio dial “Select Archive File:” and browse to select the downloaded archive project file for this exercise, LED_Blinking. Then click on Finish. Now, the demo / example project should be loaded into your workspace.



Simplicity Studio Exercise

6. Expand the LED_Blinking project and then open up the /src folder. Open up the gpio.h file and complete the following #define statements by tracing the trace from the LEDs to the Blue Gecko
   a. #define LED0_port
   b. #define LED0_pin
   c. #define LED1_port
   d. #define LED1_pin

7. Now, click on build to build the project and under the Run pull down menu, select "Profile."
   a. Simplicity IDE should begin to compile the project, and then download and flash the code into Blue Gecko on the SLWSTK6101C
   b. After flashing the microcontroller, the code should begin to run, and the LED on the SLWSTK6101C should begin to flash
   c. Simplicity should now open the Energy Profiler

Simplicity Studio Exercise

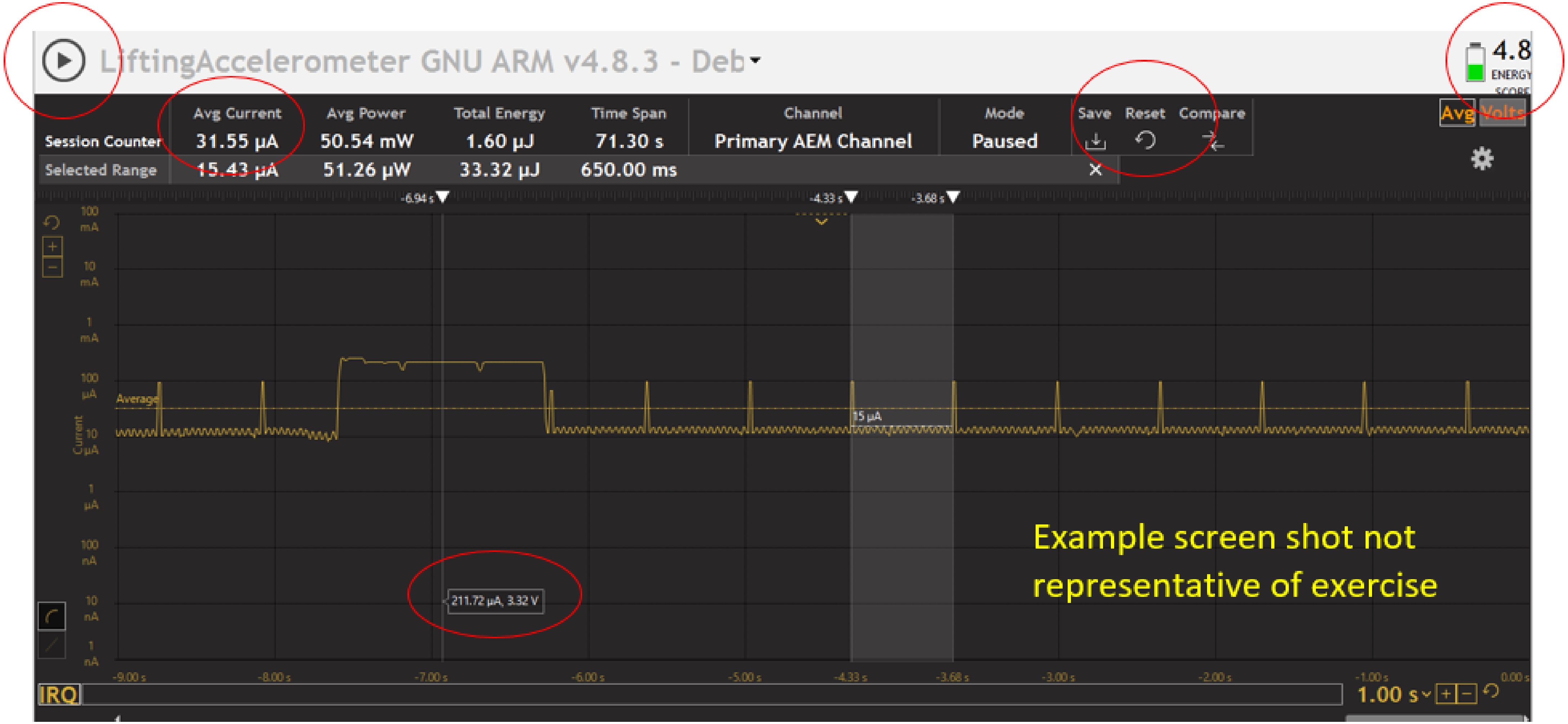
8. In the Energy Profiler, click the pause button towards the upper left corner.

9. Click once somewhere after the program has started. Towards the bottom of the marker, the instantaneous current measurement can be found.

10. Click the play button towards the upper left corner to restart the Energy Profiler measurements.

11. Towards the right end of the session counter, click on the counter clockwise arrow to reset the session counters and the Energy Profiler score. Wait 30 seconds after resetting to determine the Energy Score and average current

Simplicity Studio Exercise



Simplicity Studio Exercise

12. Pausing the Energy Profiler, without zooming in, use the instantaneous current measurement to determine how much current of a single LED draws.

12. In gpio.c, comment out these lines of code:
    - GPIO_DriveStrengthSet(LED0_port, gpioDriveStrengthStrongAlternateStrong);
    - GPIO_DriveStrengthSet(LED1_port, gpioDriveStrengthStrongAlternateStrong);

13. In gpio.c, uncomment out these lines of code:
    - GPIO_DriveStrengthSet(LED0_port, gpioDriveStrengthWeakAlternateWeak);
    - GPIO_DriveStrengthSet(LED1_port, gpioDriveStrengthWeakAlternateWeak);

14. Pausing the Energy Profiler, without zooming in, use the instantaneous current measurement to determine how much current of a single LED draw at the lower output current setting.

Simplicity Studio Exercise

15. Now go back into the Simplicity IDE, and comment out the following line of code in the main.c routine
    - GPIO_PinOutSet(LED1_port, LED1_pin); => //GPIO_PinOutSet(LED1_port, LED1_pin);

16. Now click on Run for Simplicity Studio to compile, flash, and start the updated program.

17. click on the counter clockwise arrow to reset the session counters and the Energy Profiler score. Wait 30 seconds after resetting to determine the Energy Score and average current.

Simplicity Studio Exercise

Questions:

In a separate document to be placed in the drop box with the program code, please answer the following questions:

1. How much current does a single LED draw when the output drive is set to “Strong” with the original code?

2. After commenting out the standard output drive and uncommenting “Weak” drive, how much current does a single LED draw?

3. Is there a difference in current between the answers for question 1 and 2? And, explain your answer, why or why not?
   - Due to measurement accuracy, a difference is defined as currents measured with a difference greater than 75uA

3. Using the Energy Profiler with “weak” drive LEDs, what is the Energy Score and average current measured before commenting out turning on LED1?

4. Using the Energy Profiler with “weak” drive LEDs, what is the Energy Score and average current measured after commenting out turning on LED1?

Simplicity Studio Exercise

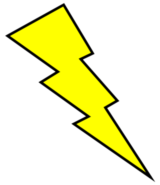
Deliverables:

In the D2L drop box for this assignment, please include two files:

1. Answers to the 5 assignment questions
2. Modified sample program with all modifications made (set to a single LED and "weak" drive)
   - Export the project and submit the .zip file via the D2L drop box

The battle between Real-Time versus Reliability?

- In making the architecture tradeoffs, which requirement wins the battle, Real-Time versus Reliability?
  - Real-Time? The asset needs to be protected
    - Example: Imagine a driverless car not able to make the decision in real-time to prevent an accident
  - Reliability? The asset needs to be protected throughout the asset lifecycle
    - Example: Imagine a driverless car whose sensor stopped working which resulted in an accident



  - Real-Time and Reliability both need to win!
    - Real-Time by architecture
    - Reliability by design

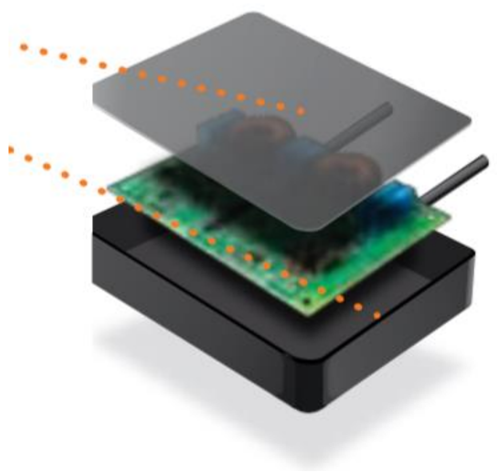
IoT / IIoT critical differences

Consumer IoT

- Benign ambient temperatures
  - 20C to 40C
- Lifecycles < 5 years
- Priority => Cost
- Wireless connectivity
  - Bluetooth, WiFi, Zwave, ZigBee

Industrial IoT

- Extreme temperatures both low and high
  - -25C to +85C
- Life cycles > 25 years
- Priority => High reliability
  - Mission critical
- Wireless connectivity
  - WiFi, ISM, Licensed



Features:

- High efficiency power conversion
- Fast and accurate MPPT per PV module
- Ampt Mode™ technology to increase inverter output power
- String Stretch™ technology to build longer strings
- Output voltage and current limits
- Instrument-grade precision measurement
- Optional two-way wireless communication
- Independent power optimization without reliance on communication
- Inverter and PV module compatible
- Compatible with 3rd party monitoring
- Utility-strength operation and stability
- High reliability
- 25 year warranty

Ampt-x Converter Model	V40-x	V50-x	V100-x
Electrical*			
Input			
Maximum module power (Pmax) at STC	260 W	320 W	360 W
Maximum module voltage (Voc) at coldest design temperature	46 V	58 V	102 V
Module MPP DC voltage range	10 - 38 V	17 - 48 V	25 - 80 V
Maximum module current (Imp) at STC	8.5 A	9.2 A	6.1 A**
Maximum module short circuit current (Isc) at STC	9.2 A	9.2 A	6.7 A
Output			
Maximum converter output voltage	33.3 V	40.6 V	63.6 V
Maximum converter output current	9.4 A	9.2 A	6.7 A**
Maximum converter output power	260 W	320 W	360 W
Maximum operating efficiency	99.0%	99.2%	99.2%
Mechanical			
Ambient temperature operating range	-40°F to +158°F (-40°C to +70°C)		
Dimensions	5.9 x 4.7 x 1.4 in. (15 x 11.9 x 3.6 cm)		
Weight	12 oz. (0.3 kg)		
Cooling	Convection		
General			
Communication	Two-way wireless (optional)		
Compliance	CSA to UL 1741, FCC Part 15 Class B IEC 62109, 61000-6-1, 61000-6-3		
Demonstrated MTBF at 40°C continuous	90 million hours		
Warranty	25 years		

* Standard test condition (STC) irradiation level of 1000 W/m² at 25°C.
** 6.1 A input and 6.7 A output at 60°C. 5.45 A input and 5.55 A output at 70°C.

> 85C with temperature rise inside of box

Xeta2-E

9.6 kbps - 3.5 Mbps Ethernet Radio
Broad/Narrowband 216 MHz-232 MHz

Security

- AES 128/ 256-bit encryption
- Password authentication
- VLAN network segregation

Transmitter	
Frequency Range	216 – 232 MHz
Output Power	10mW to 8W, step size 10mW
Range – Line of Sight	70+ miles
Modulation	BPSK, QPSK, 8-PSK, 16 QAM, 2-4 Level GFSK, 32-QAM
RF Data Rate	9.6 kbps to 3.5 Mbps

Environmental

- -40°C to +85°C operating temperature range. *-55°C available.*
- 95% operating humidity @ 40°C non-condensing.

T2 TRANSPARENT TECHNOLOGIES

BEYOND AMR

Register features:

- Compact, fully encapsulated package for all meter environments.
- Large 8-digit LCD display showing totalization, measuring units and direction of flow.
- Zero-drag magnetic sensor for meter magnet tracking.
- Embedded high-resolution datalogging down to magnet-by-magnet turns.
- Flowrate and meter test functions.
- 20-year warranted battery life (10-yr full/10-yr prorated).

Specifications:

Temperature:	-40F° to 185F° (-40C° to +85C°)
Humidity:	100%
Submersion:	IP68 Fully Submersible
Battery:	19.0Ah D-Cell

http://www.metronfarnier.com/wp-content/themes/metron_theme/PDF%20Downloads/Innov8%20Registers/Innov8%20Datasheet.pdf

T701i WIFI Thermostat

T701i Specifications

Specifications

T701i Thermostat

The display range of temperature	41°F to 95°F (5°C to 35°C)
The control range of temperature	44°F to 90°F (7°C to 32°C)
Load rating	1 amp per terminal, 1.5 amp maximum all terminals combined
Display accuracy	± 1 °F
Swing (cycle rate or differential)	Heating is adjustable from 0.2°F to 2.0°F Cooling is adjustable from 0.2°F to 2.0°F
Power source	18 to 30 VAC, NEC Class II, 50/60 Hz for hardwire (common wire) Battery power from 2 AA Alkaline Energizer batteries
Operating ambient	32° to +105° (O° to + 41°C)

Kwikset Kevo
Bluetooth Door Lock

What is Kevo's warranty?

Your Kwikset Kevo products are backed by one of the most comprehensive warranty programs available. You can feel confident that with the purchase of Kwikset you have selected the best quality product, backed by the best customer service available. Your Kwikset Kevo deadbolt product comes with a lifetime mechanical and finish warranty along with a 1 year electronic warranty to the original residential user of the product against defects in material and workmanship as long as the original user occupies the residential premises upon which the product was originally installed.

Security and Data Privacy

- Digital models of the physical world including the collection of new customer usage and product data creates security concerns:
  - The use of “big data” in the backend must be managed in a secure fashion as well as individual’s data
  - In the distributed world of IoT/IIoT, the connections between the device and the backend need to be secured
  - The IoT/IIoT hardware and software must be secured or at a minimum detect intrusion and be able to be isolated from the network
  - If the IoT/IIoT is controlling an asset, the asset hardware and software must be secured

IoT security breach

- **Building Infrastructure:** The Department of Homeland Security recently disclosed a 2012 breach in which cybercriminals managed to penetrate the thermostats of a state government facility and a manufacturing plant in New Jersey. The hackers exploited vulnerabilities in industrial heating systems, which were connected to the Internet and then changed the temperature inside the buildings. *(On the surface, that might seem harmless, but think about the damage that cybercriminals could do with unfettered access to the controls that govern most major buildings today. The smart building might not seem so smart if for example, the bad guys activate the water sprinkler systems in a data centre or mess with the elevators.)*

https://www.linkedin.com/pulse/10-most-terrifying-iot-security-breaches-so-far-you-arent-montgomery

IoT security breach

- **The Kitchen:** This breach that recently occurred in the UK boggles the mind. Hackers attacked IoT-connected devices in kitchens across the country, with almost comical outcomes. Smart toasters are forcing consumers into reconsidering eating habits by refusing to toast any bread that isn't considered 'healthy'. Smart Fridges and freezers across the UK are shutting down as soon as ice cream is detected. *(The message is abundantly clear. Leave that white bread on the grocery store counter and stock up on whole wheat, and while you're there, put down those high-fat/high-calorie frozen goodies in favour of good old wholesome fruit).*

https://www.linkedin.com/pulse/10-most-terrifying-iot-security-breaches-so-far-you-arent-montgomery

IoT/IIoT topics to be covered

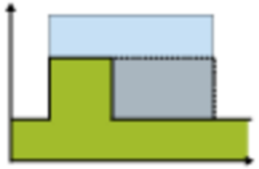
- Choosing the right controller for the application based on network, system, and end application requirements
- Designing reliability to meet the target asset or end application
- Security at the embedded product level

Low Energy

Energy = Power x Time

- Battery Capacity is measured in Energy such as mA-h
- To increase Batter Life, Energy must be reduced
  - Decrease Power
  - Decrease Time
- Low Energy firmware design minimizes both Power and Time

How does a microcontroller reduce energy?



- Reduce ON Time:
  - Highest energy consuming peripheral in a microcontroller is the CPU
  - A higher computational CPU will reduce the time required for a fix amount of work
  - Thus, a higher computational CPU => Reduced computational time
  - Resulting in minimizing CPU on time minimizes energy

How does a microcontroller reduce energy?

- MSP430
  - DMIPS/MHz = 0.288
  - 330uA/MHz active
  - Or $\frac{uA/MHz}{DMIPS/MHz}$
  - Or $\frac{330uA/MHz}{0.288DMIPS/MHz}$
  - 1,146uA/DMIP
- Silicon Labs’ Leopard Gecko Cortex-M3
  - DMIPS/MHz = 1.25
  - 216uA/MHz
  - 173uA/DMIP
- Cortex M3 is 662% more energy efficient than the MSP430
  - The Cortex-M3 is more energy efficient than the MSP430 not just due to the current per MHz (~ 50%), but due to the 400+% higher computational performance

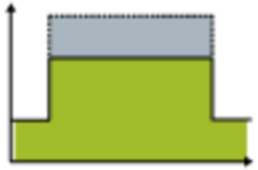
Comparison of MSP430 and MSP432

	MSP430	MSP430X	MSP432
Address space	16 bits	20 bits	32 bits
Memory address space	64kB	1MB	4GB
Clock speed	25 MHz		48 MHz
Floating Point	soft		IEEE754 32 bit FPU
Typical Dhrystone 2.1 (DMIPS/MHz)	0.288 [3]		1.196
ULPBench low power score	120		167.4

https://en.wikipedia.org/wiki/TI_MSP430, https://en.wikipedia.org/wiki/TI_MSP432, http://www.ti.com/lit/ds/symlink/msp430f1611.pdf, http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32LG990.pdf

How does a microcontroller reduce energy?

- Reduce Power:
  - Very low active power consumption



    - Specifications can be in uA/MHz or mA per clock speed
      - Silicon Labs’ EFM32LG (Leopard Gecko) Cortex-M3

14 MHz HFRCO, all peripheral clocks disabled, V_{DD}= 3.0 V, T_{AMB}=25°C			216		µA/ MHz

Compared to the MSP430 from the previous slide at 330uA/MHz, the Cortex-M3 is a lower energy MCU architecture

      - NXP LPC15xx Cortex-M3

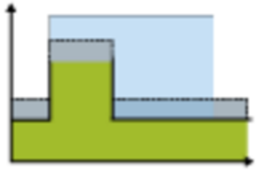
I_{DD}	supply current	Active mode; code `while(1){}` executed from flash;					
		system clock = 12 MHz; default mode; V_{DD} = 3.3 V	[3][4][5] [7][8]	-	4.3	-	mA
		system clock = 12 MHz; low-current mode; V_{DD} = 3.3 V	[3][4][5] [7][8]	-	2.7	-	mA

- 2.7mA / 12MHz = 0.225mA/MHz or 225uA/MHz

http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32LG990.pdf
http://www.nxp.com/documents/data_sheet/LPC15XX.pdf
http://www.nxp.com/documents/user_manual/UM10736.pdf

How does a microcontroller reduce energy?

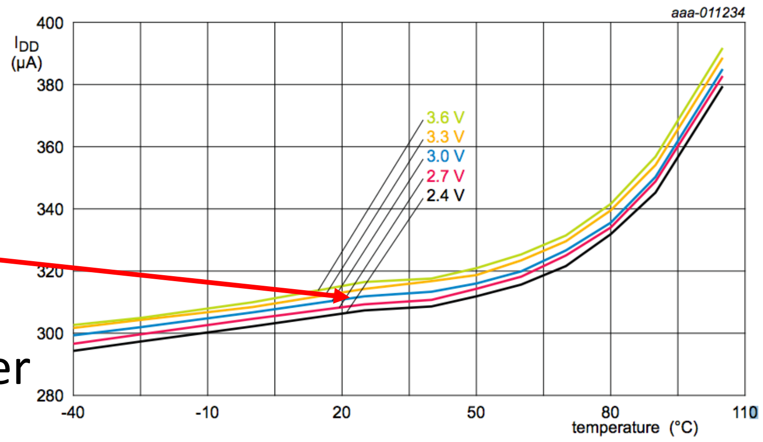
- Reduce Power:



  - Ultra-Low power sleep modes
    - Silicon Labs' EFM32LG (Leopard Gecko) Cortex-M3

I_{EM2}	EM2 current	EM2 current with RTC prescaled to 1 Hz, 32.768 kHz LFRCO, V_{DD}= 3.0 V, T_{AMB}=25°C			0.95[1]	1.7[1]	μA

- NXP LPC15xx Cortex-M3
  - Deep-sleep mode is similar EM2 above
  - 310uA at 25C



Conditions: BOD disabled; all oscillators and analog blocks disabled. Use API power_mode_configure() with mode parameter set to DEEP_SLEEP and peripheral parameter set to 0xFF.

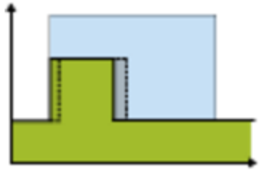
Fig 21. Deep-sleep mode: Typical supply current I_{DD} versus temperature for different supply voltages V_{DD}

- If being in sleep mode is a significant period of operation, then the Leopard Gecko may be a better choice

http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32LG990.pdf
http://www.nxp.com/documents/data_sheet/LPC15XX.pdf
http://www.nxp.com/documents/user_manual/UM10736.pdf

How does a microcontroller reduce energy?

- Reduce Power:
  - Fast wake up times from sleep or low energy modes



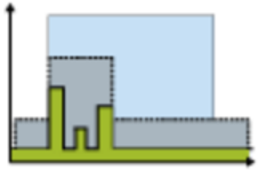
    - Time to wake up from a low energy mode is wasted energy -> no work is being accomplished
      - Silicon Labs' EFM32LG (Leopard Gecko) Cortex-M3

t_{EM20}	Transition time from EM2 to EM0		2		µs

      - NXP LPC15xx Cortex-M3
        - "In Deep-sleep mode, the LPC15xx is in Sleep-mode and all peripheral clocks and all clock sources are off except for the IRC. The IRC output is disabled unless the IRC is selected as input to the watchdog timer. "

http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32LG990.pdf
http://www.nxp.com/documents/data_sheet/LPC15XX.pdf
http://www.nxp.com/documents/user_manual/UM10736.pdf

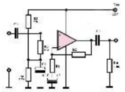
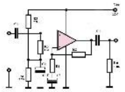
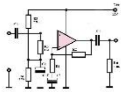
How does a microcontroller reduce energy?



- Reduce Power:
  - Autonomous Peripherals
    - Highest energy consuming peripheral in a microcontroller is the CPU
    - Do more work with the CPU off, the more energy efficient
      - Silicon Labs' EFM32LG (Leopard Gecko) Cortex-M3
        - Peripherals operating in EM2 mode (not complete list):
          - USB, I2C, Low Energy UART, Low Energy Sense, Low Energy Timer, Interrupts, DAC, Analog Comparators, Voltage Comparators
      - NXP LPC15xx Cortex-M3
        - Peripherals operating in Deep-Sleep mode (not complete list):
          - USB, I2C, SPI, USART, RTC, Interrupts

http://www.silabs.com/Support%20Documents/TechnicalDocs/EFM32LG990.pdf
http://www.nxp.com/documents/data_sheet/LPC15XX.pdf
http://www.nxp.com/documents/user_manual/UM10736.pdf

What are the characteristics that the firmware engineer can take advantage?

- Low Energy Microcontroller characteristics:
  -  Higher Computational CPU
  -  Very Low Active Power Consumption
  -  Ultra Low Power Sleep Modes
  -  Fast wake up times from sleep or low energy modes
  -  Autonomous Peripherals

Additional considerations to select a Low Energy Microcontroller

- Advanced autonomous peripheral functions:
  - Passive sensor state machines - LESENSE
  - Peripheral Intercommunication - PRS
- Well architected Energy Modes
- Energy or Current monitors

Advanced autonomous peripheral functions

- Passive sensor state machines
  - State machine equates to no CPU resources required
    - Highest energy peripheral, CPU, can be turned off
  - Passive equates to no active energy required
    - Energy is only required when sensor is turned on
  - Silicon Labs Leopard Gecko's Low Energy Sensor Interface (LESENSE)
    - Up to 16 passive sensor
    - Excites, Measures, and Decodes passive sensors without the CPU
    - Operates down to Energy Mode 2 with the CPU turned off

Advanced autonomous peripheral functions

- Peripheral Intercommunication
  - Create new autonomous functions by combining individual autonomous peripherals
  - Increase functionality without CPU involvement allowing the microcontroller to remain in a low energy state
  - Silicon Labs Leopard Gecko Peripheral Reflex System
    - 12 interconnect channels
    - Any producing peripheral can connect to any consuming peripheral
    - Example of ”new” autonomous function
      - LESENSE ambient light sensor can trigger an ADC conversion whose result is stored in memory via DMA
      - Complete operation can occur while in EM1 with no CPU involvement
        - 938uA typically versus 3,024uA in EM0 with the CPU

Well architected Energy Modes

- Well architected Energy Modes equates to the appropriate peripherals for your application that will enable you to remain in the lowest possible energy mode for the longest period of time
  - For example, if an Analog Comparator is required to monitor the system to initiate an activity, having this peripheral available in a low energy state would be important
    - The Silicon Labs Leopard Gecko enables Analog Comparator down to EM3

I_{EM3}	EM3 current	V_{DD}= 3.0 V, T_{AMB}=25°C			0.65	1.3	µA
		V_{DD}= 3.0 V, T_{AMB}=85°C			2.65	4.0	µA

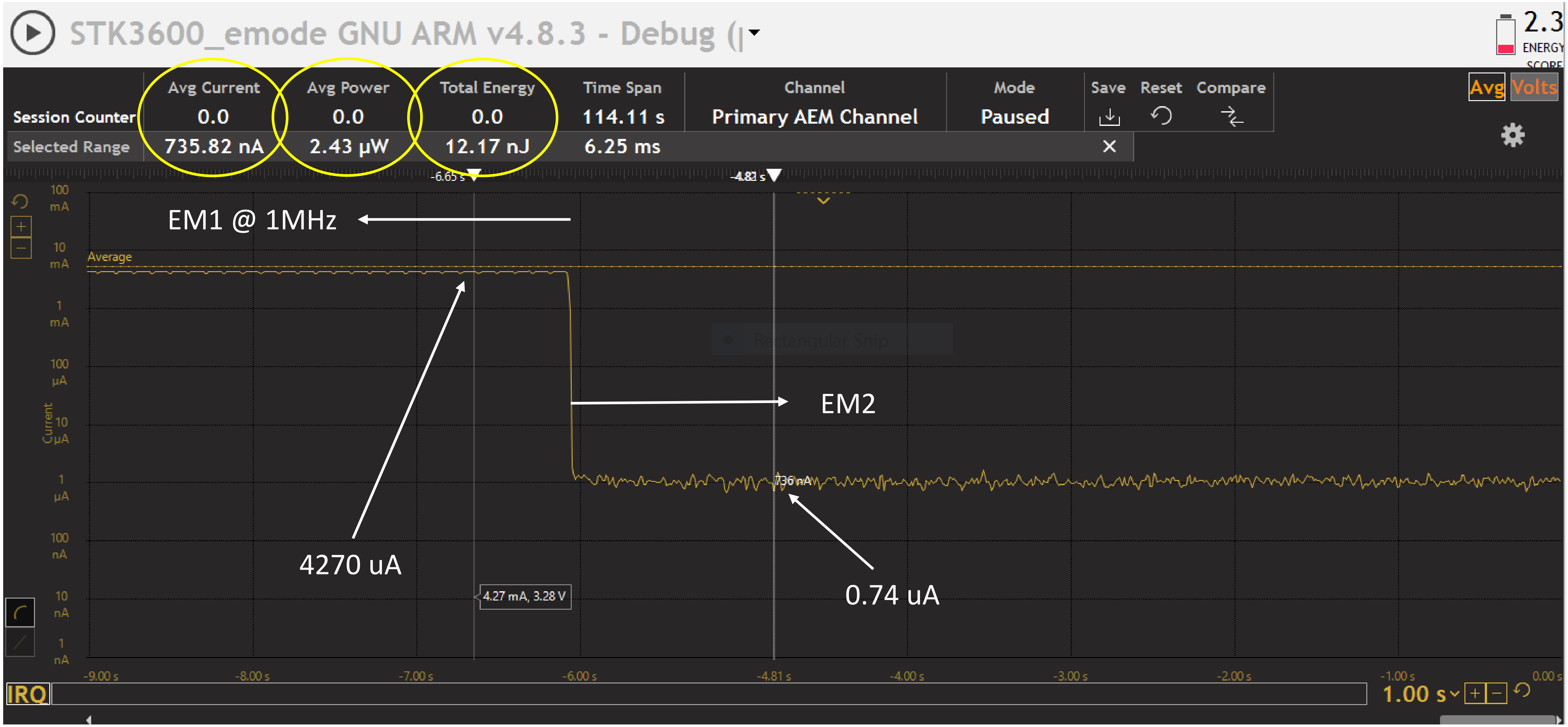
    - While the NXP LPC15xx enables its Analog peripherals only to Sleep Mode

system clock = 12 MHz; low-current mode; V_{DD} = 3.3 V	[3][4][5] [7][8]	-	2.7	-	mA

    - Silicon Labs' ACMP lower energy mode is typically 42 times more energy efficient

Energy or Current Monitors

- Energy or Current Monitors provide real time data on the use of energy
- Can be used to determine which routines consume a significant amount of energy by using code correlation
  - Enables the firmware engineer to pinpoint where the code is spending energy
  - And, thus, focus attention to reduce energy in those routines
- Verifies the energy efficiency of the firmware design



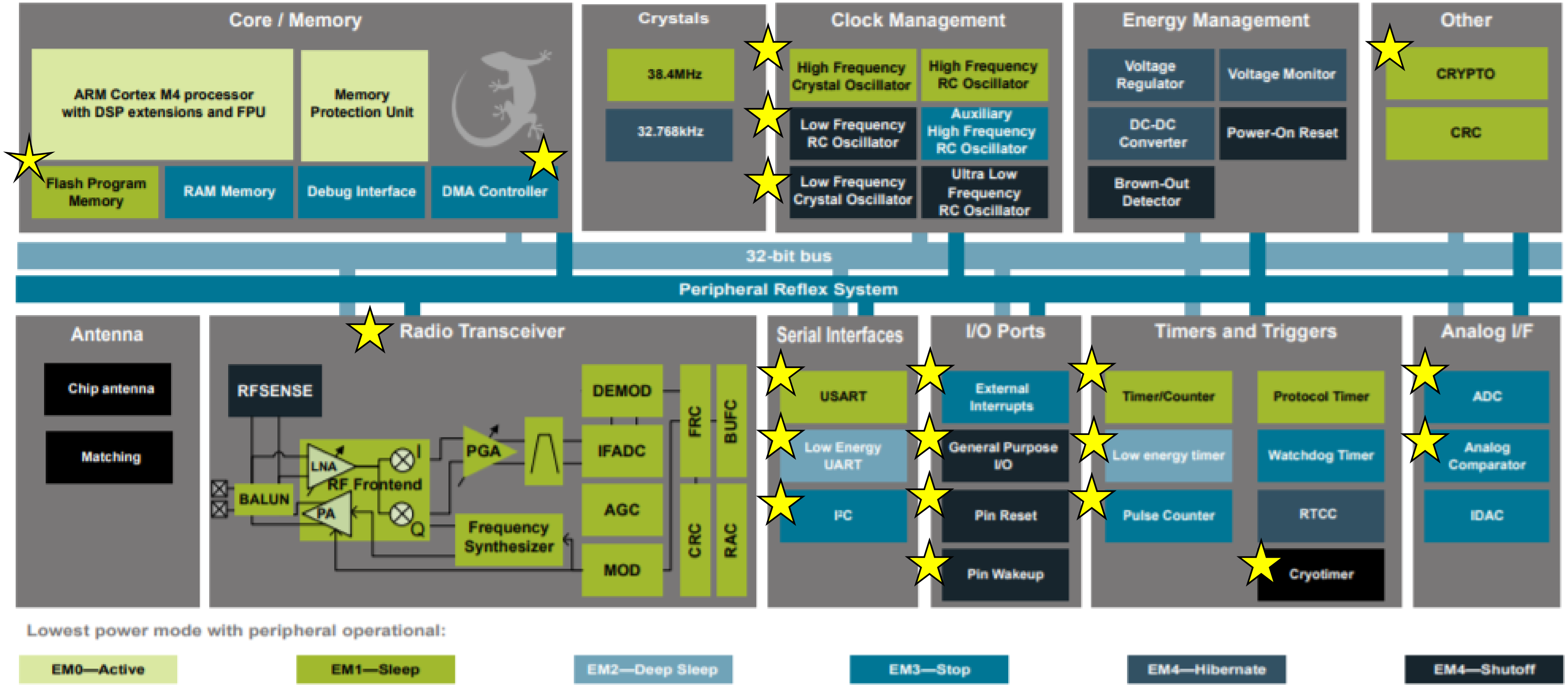
Silicon Labs' Energy Profiler of the Energy Mode sample code on the Leopard Gecko

What are the characteristics that the firmware engineer can take advantage?

- Low Energy Microcontroller characteristics:
  - Higher Computational CPU
  - Very Low Active Power Consumption
  - Ultra Low Power Sleep Modes
  - Fast wake up times from sleep or low energy modes
  - Autonomous Peripherals
- Advanced autonomous peripheral functions:
  - Passive sensor state machines - LESENSE
  - Peripheral Intercommunication - PRS
- Well architected Energy Modes
- Energy or Current monitors

Example project from fall 2016

https://www.youtube.com/watch?v=J5XtAHLOXU4&feature=youtu.be


Core / Memory
ARM Cortex M4 processor with DSP extensions and FPU
Memory Protection Unit
Flash Program Memory
RAM Memory
Debug Interface
DMA Controller
Crystals
38.4MHz
32.768kHz
Clock Management
High Frequency Crystal Oscillator
High Frequency RC Oscillator
Low Frequency RC Oscillator
Auxiliary High Frequency RC Oscillator
Low Frequency Crystal Oscillator
Ultra Low Frequency RC Oscillator
Energy Management
Voltage Regulator
Voltage Monitor
DC-DC Converter
Power-On Reset
Brown-Out Detector
Other
CRYPTO
CRC
32-bit bus
Peripheral Reflex System
Antenna
Chip antenna
Matching
Radio Transceiver
RFSENSE
DEMOD
IFADC
AGC
MOD
FRC
BUFC
CRC
RAC
PGA
LNA
PA
BALUN
RF Frontend
Frequency Synthesizer
I
Q
Serial Interfaces
USART
Low Energy UART
I²C
I/O Ports
External Interrupts
General Purpose I/O
Pin Reset
Pin Wakeup
Timers and Triggers
Timer/Counter
Protocol Timer
Low energy timer
Watchdog Timer
Pulse Counter
RTCC
Cryotimer
Analog I/F
ADC
Analog Comparator
IDAC
Lowest power mode with peripheral operational:
EM0—Active
EM1—Sleep
EM2—Deep Sleep
EM3—Stop
EM4—Hibernate
EM4—Shutoff

Ideal CMOS FET

- CMOS logic reduces power consumption because no current flows (ideally), and thus no [power is consumed, except when the inputs to logic gates are being switched. CMOS accomplishes this current reduction by complementing every nMOSFET with a pMOSFET and connecting both gates and both drains together.](https://en.wikipedia.org/wiki/MOSFET)
- Thus, power/energy is reduced when:
  - CMOS logic is not clocked or switched
  - Lower switching frequency will result in reduced current/power/energy
    - Caveat: Only if it does not extend the time the CPU remains in EM0