# Rapid Prototyping with step-by-step Functionality using full TI Portfolio

**“Yes, even YOU can prototype with minimal or NO programming, even embedded and wirelessly” (as Launchpad meets Beaglebone)**

**Joe George, Northeast Digital Field Applications**

**Texas Instruments**

**Americas Sales and Marketing**

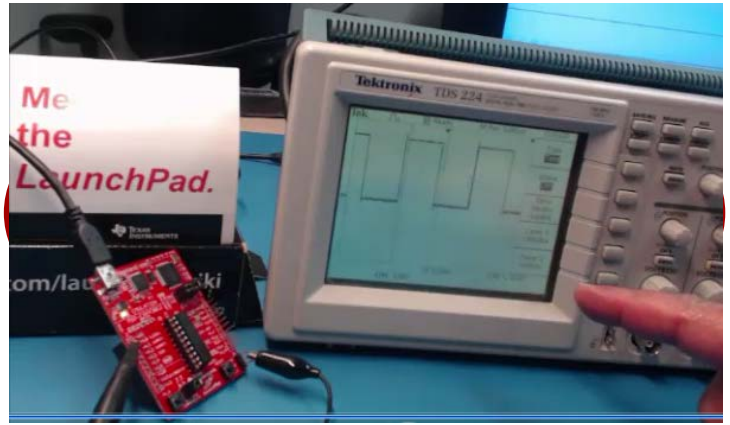
# Agenda

- Fundamentals (mostly for Analog)
  - Implementing  necessary prototyping functions such clocks/GPIO, Read A/D, I2C/SMBus, etc.
  - Seamless interface of various Analog EVM's for system "proof of concept"
  - Standalone UI - Button (GP Input - GPIO), LCD Display ("Hello"), Music, Serial Interface (Putty)

- More UI (i.e. GUI Advanced Comm Tab basically Putty/Serial I/F) - Lessons learned from home networking (if you can setup the WiFi in your house, you can prototype with a few steps)

- EP - Embedded prototyping (mostly for Digital)
  - Wired  and Wireless Control
  - Use of TI Cloud Computing Tools for prototype

- Advanced Topics
- Conclusion Demos (Simple and Complex)

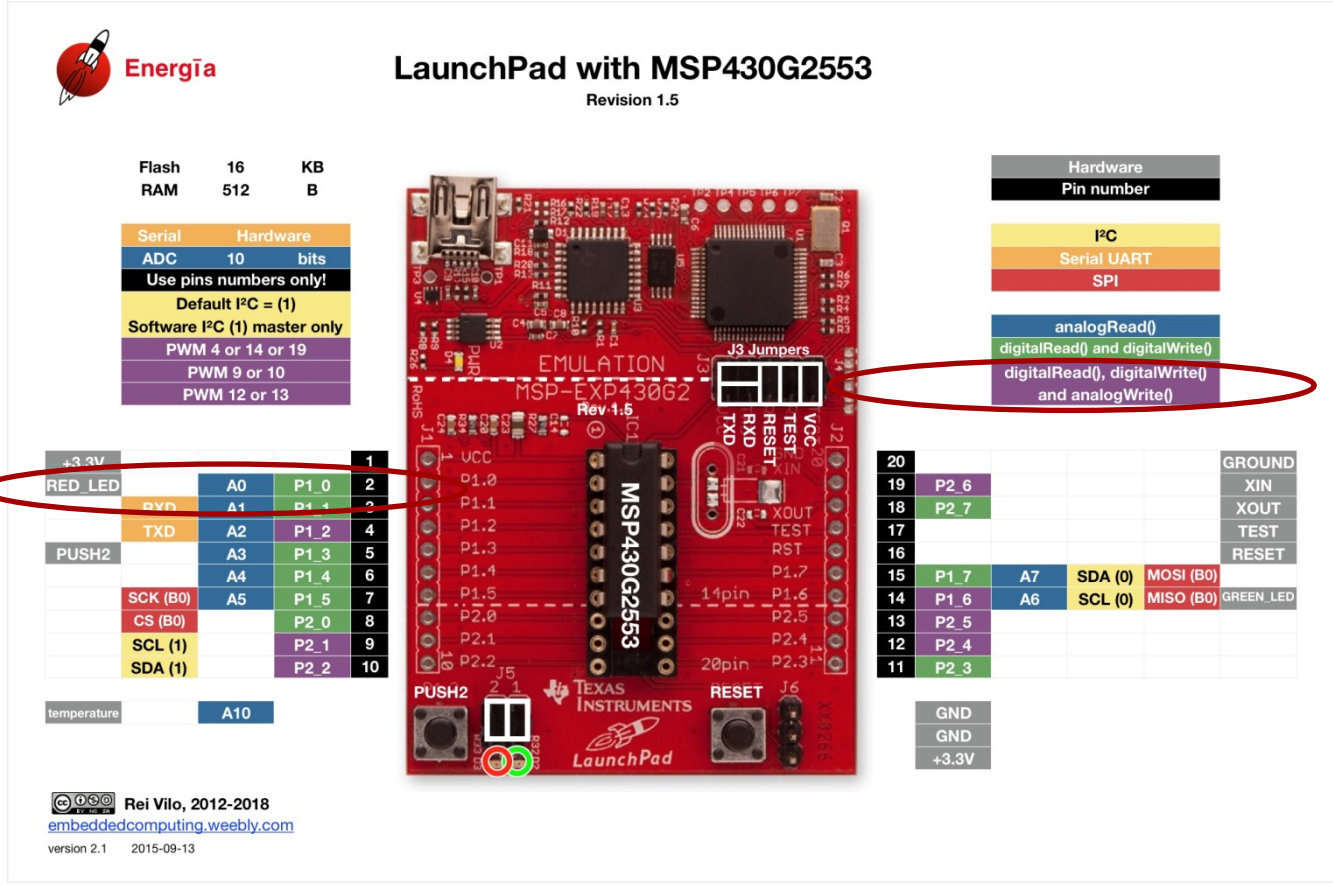


# Prototyping Functions

- Step-by-step Functionality (Demos)
  - GP Output (GPIO – General Purpose I/O)->Square Wave->Clock
  - Read A/D
  - I2C/SMBus (Wire)
- Optional WiFi
  - STA (station)
  - AP (access point)
- Optional Energia
- UI
  - Button (GP Input - GPIO, add debounce)
  - LCD Display (“Hello”)
  - Music
- UI - Serial Interface (i.e. Putty for echo “Hello World”)

# Example Pin Map – Digital I/O

- How can you not love an MCU in a DIP package for rapid prototyping?

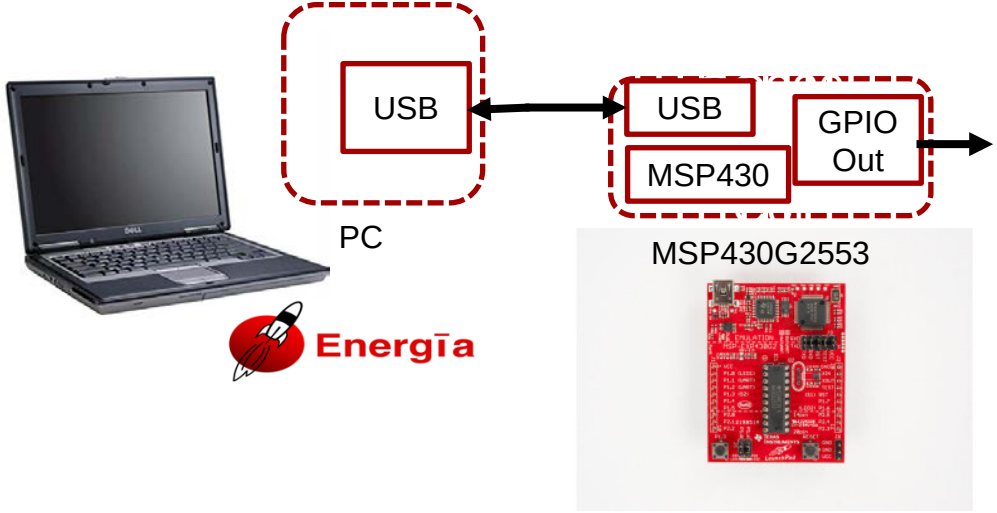


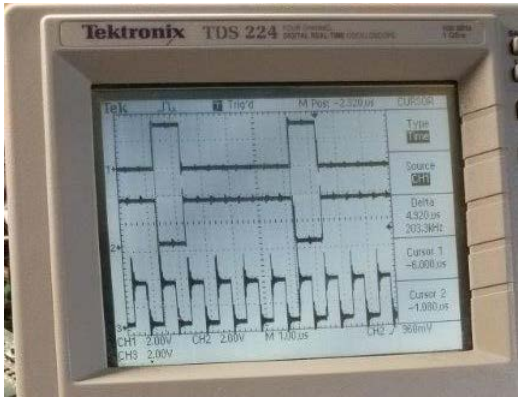

- http://www.energia.nu/pinmaps/msp-exp430g2/

# Demo - GPIO

– Step-by-step Functionality (Demo) - PFC (Power Factor Correction) Controller

- GP Output (“Blinky” is just General Purpose Input) – Energia Blink (square wave)
  - P1.0 ->RED_LED
    - »static const uint8_t P1_0 = 2;
    - »static const uint8_t RED_LED = 2;
  - const int ledPin =  RED_LED;  // the number of the LED pin

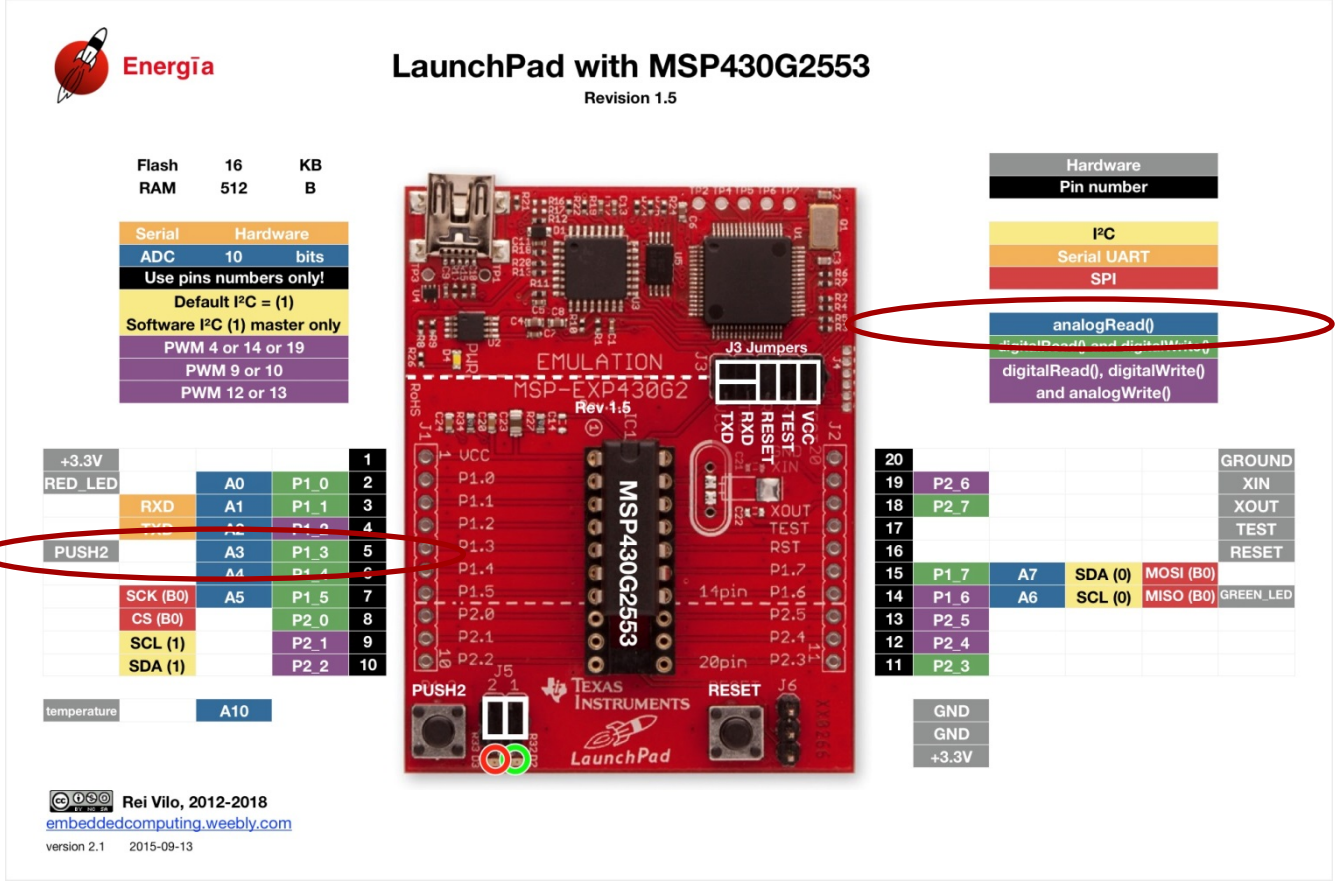



Square Waves with Reference Clock

# Example Pin Map – Analog A/D

- Who doesn't need to read an A/D?



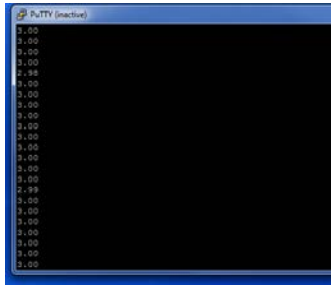

- http://www.energia.nu/pinmaps/msp-exp430g2/

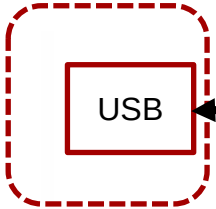
# Demo – Read A/D

– Step-by-step Functionality (Demo)

• Read A/D – Energia ReadAnalogVoltage (level)

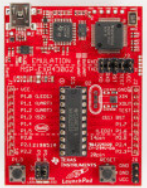
– (A3) P1.3





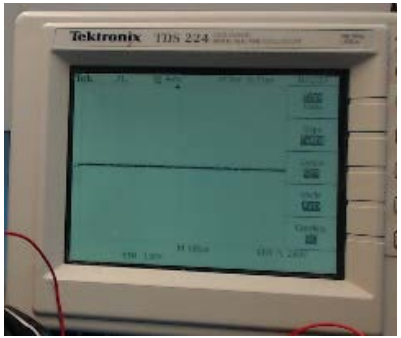

PC


MSP430G2553



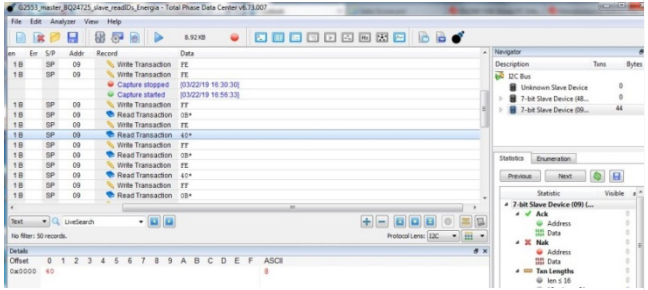
Energia


A3=VCC-3.3V

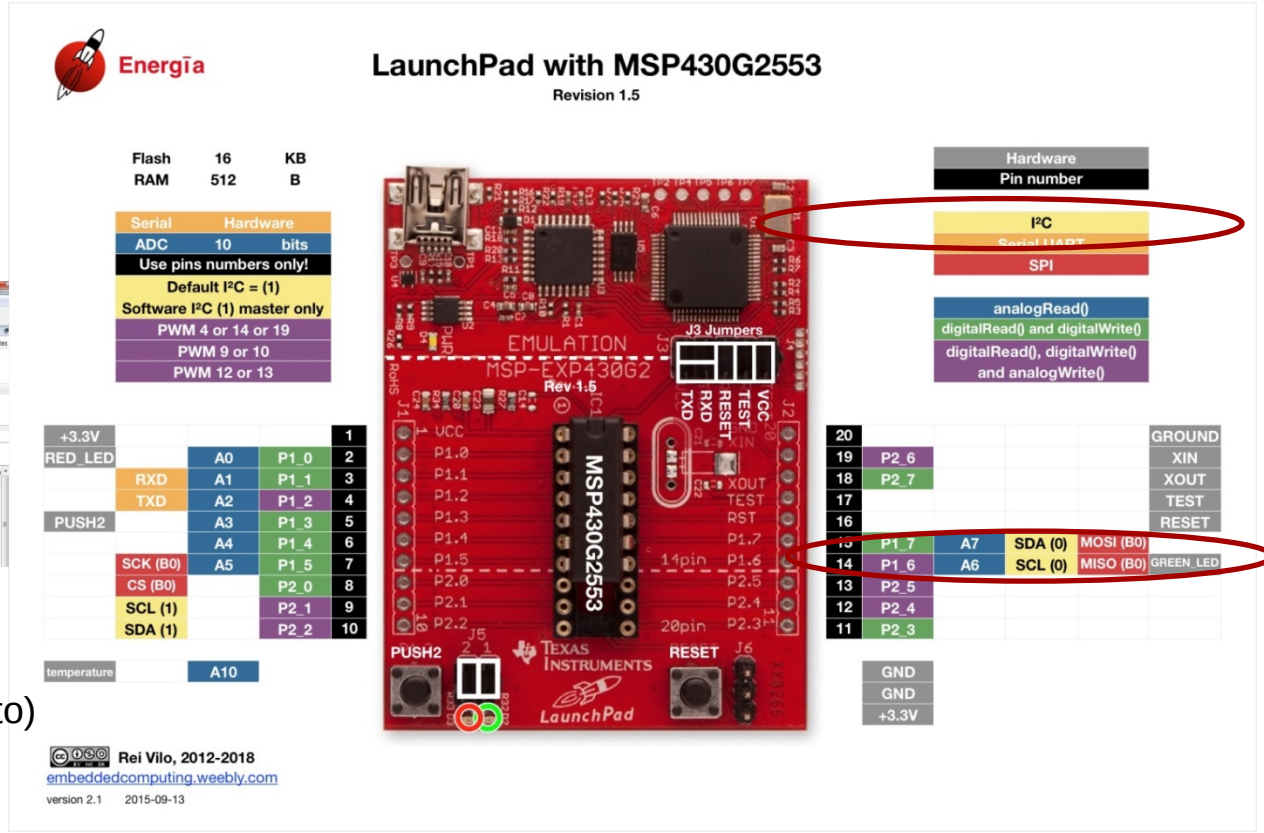



A3=GND=0V

# Example Pin Map – I2C/SMBus



- Gotta send some I2C commands? (GUI tools nice for eval but not proto)

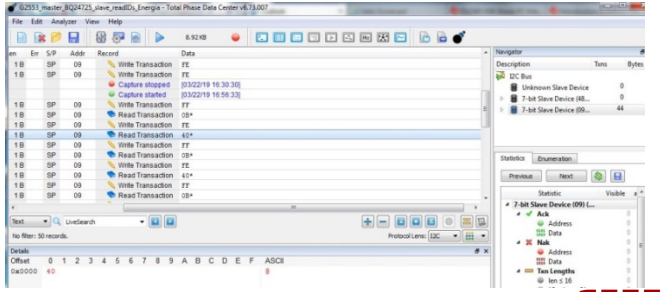


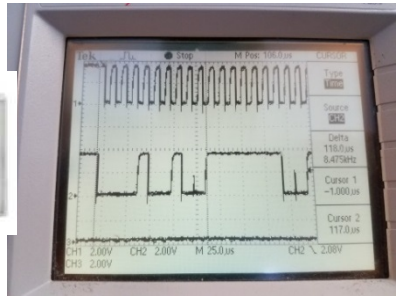

- http://www.energia.nu/pinmaps/msp-exp430g2/

# Demo – I2C

– Step-by-step Functionality (Demo) - SMBus (I2C) setup for Battery Charger

- I2C (Wire Write/Read I2C ) – Energia Wire master_writer (with "Wire Slave Receiver“ BQ24725) and I2C Bus Analyzer





Single I2C Write

~20V DC

USB

PC

USB

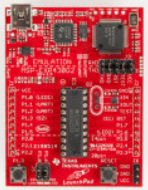
MSP430

I2C (USCI)

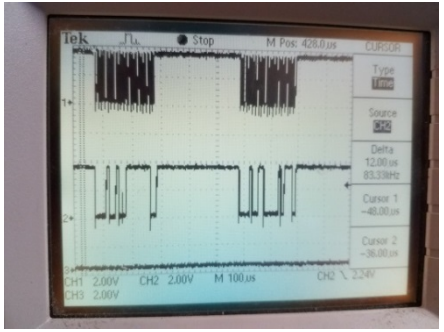
SMBus

BQ24725

MSP430G2553



BQ24725AEVM



I2C Write-Read

# Analog EVMs with ti.com links – NO Wiring

- Interface of Analog EVM's
  - – Building Automation Sensors BoosterPack Module
    - http://www.ti.com/tool/BOOSTXL-SENSORS
    - http://www.ti.com/tool/BOOSTXL-BASSENSORS
  - – DRV8323RH BoosterPack and MSP430F5529LP code – InstaSpin
  - – Fast Prototyping 24bit Sigma-Delta ADC with Energia (TI ADS1220): TIDA-00647

- MSP-EXP430G2 Based Design kits & evaluation modules
  http://www.ti.com/product/MSP430G2553/toolssoftware

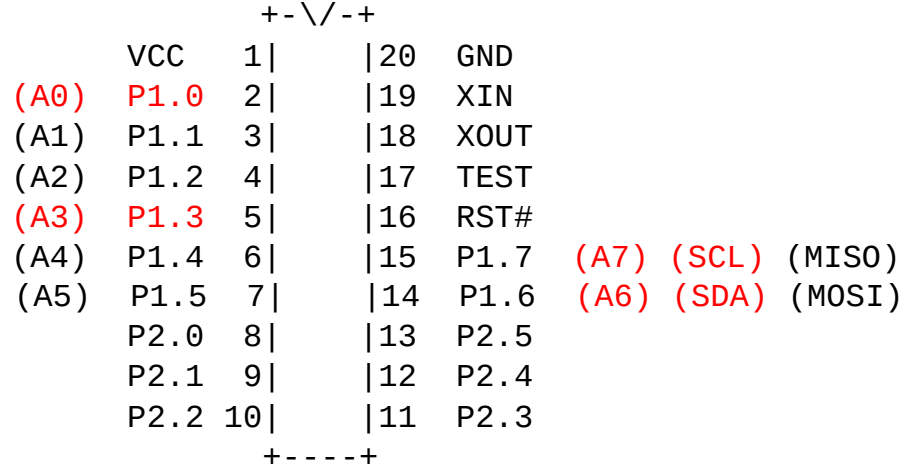
# MSP-EXP430G2 Based Design kits & EVMs

- MSP-EXP430G2 Based Design kits & evaluation modules Part #Name
  - BOOST-IR Infrared (IR) BoosterPack Plug-in Module
  - 430BOOST-ADS1118 ADS1118 BoosterPack
  - 430BOOST-CC110L CC110L RF BoosterPack
  - DRV8873SEVM10A H-bridge motor driver with hardware interface and integrated current sensing EVM
  - DRV8873S-Q1EVM Automotive 10A H-Bridge Motor Driver with SPI and Integrated Current Sensing EVM
  - DRV8308EVM DRV8308 Evaluation Module for Three-phase Brushless Motor Pre-driver
  - DRV8313EVM DRV8313 Evaluation Module
  - DRV8701EVM DRV8701 Full Bridge Brushed DC Motor Gate Driver Evaluation Module
  - DRV8702D-Q1EVM Automotive Half-Bridge Brushed DC Motor Gate Driver Evaluation Module D
  - DRV8703D-Q1EVM DRV8703D-Q1 Automotive Half-Bridge Brushed DC Motor Gate Driver With SPI EVM
  - DRV8847EVM DRV8847 Dual H-Bridge Motor Driver Evaluation Module
  - DRV8847SEVM DRV8847S Dual H-Bridge Motor Driver Evaluation Module
  - DRV8850EVM DRV8850EVM - EVMfor Low-Voltage H-Bridge Motor Driver with LDO Voltage Regulator
  - DRV8912-Q1EVM DRV8912-Q1 multi h-bridges motor driver evaluation module
  - DLP-7970ABP NFC Transceiver Booster Pack

# Demo – Pin Definition Reference

– energia-0101E0014\hardware\msp430\variants\launchpad\pins_energia.h

```
– // Pin names based on the silkscreen
– //
– static const uint8_t P1_0 = 2;
– static const uint8_t P1_1 = 3;
– static const uint8_t P1_2 = 4;
– static const uint8_t P1_3 = 5;
– static const uint8_t P1_4 = 6;
– static const uint8_t P1_5 = 7;
– static const uint8_t P2_0 = 8;
– static const uint8_t P2_1 = 9;
– static const uint8_t P2_2 = 10;
– static const uint8_t P2_3 = 11;
– static const uint8_t P2_4 = 12;
– static const uint8_t P2_5 = 13;
– static const uint8_t P1_6 = 14;
– static const uint8_t P1_7 = 15;
– static const uint8_t P2_7 = 18;
– static const uint8_t P2_6 = 19;
```




```
// Pin names based on the silkscreen
```

# Prototyping Functions

- Step-by-step Functionality (Demos)
  - GP Output (GPIO – General Purpose I/O)
  - Read A/D
  - I2C/SMBus (Wire)
    - *Analog EVM/Boosterpack's*
    - *PinMap Housekeeping*
- Optional WiFi
  - STA (station)
  - AP (access point)
- Optional Energia
- UI
  - Button (GP Input - GPIO, add debounce)
  - LCD Display ("Hello")
  - Music
- UI - Serial Interface (i.e. Putty for  echo "Hello World")

# Demo – WiFi STA https://energia.nu/guide/libraries/wifi/

- Step-by-step Functionality (Demo)
  - WiFi (Wireless UI and Cloud with HTTP Server and/or local Access Point)
    - Energia SimpleWebServerWiFi (HTTP server as WiFi STA) – Blink LED

```
You're connected to the network
Waiting for an ip address

IP Address obtained
SSID: TP-LINK_33C4BE
IP Address: 192.168.1.100
signal strength (RSSI):-40 dBm
To see this page in action, open a browser to
http://192.168.1.100
Starting webserver on port 80
Webserver started!
```

```
new client
GET /L HTTP/1.1
Host: 192.168.1.100
Upgrade-Insecure-Requests: 1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*
;q=0.8
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 11_2_5
like Mac OS X) AppleWebKit/604.5.6 (KHTML, like Gecko)
Version/11.0 Mobile/15D60 Safari/604.1
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: keep-alive
client disonnected
```

CC3220 SimpleLink Wi-Fi (WiFi STA)

# Demo – WiFi AP (Access Point)

https://energia.nu/guide/libraries/wifi/

– Step-by-step Functionality (Demo)

• WiFi (Wireless UI and Cloud with HTTP Server and/or local Access Point)

– Energia APWatchConnectDisconnect (WiFi Access Point) – AP Network

```
Setting up Access Point named: MyEnergiaAPJoe
AP uses WPA and password is: password

AP active.
Client connected! All clients:
Client #0 at IP address = 192.168.1.2, MAC = F8:27:93:44:7B:03
Client disconnected.
Client connected! All clients:
Client #0 at IP address = 192.168.1.2, MAC = F8:27:93:44:7B:03
Client disconnected.
Client connected! All clients:
Client #0 at IP address = 192.168.1.3, MAC = 08:C5:E1:E3:BB:93
Client disconnected.
```

CC3200 SimpleLink Wi-Fi (WiFi AP)

CC3220 SimpleLink SDK Wi-Fi (WiFi AP)

# Prototyping Functions

- Step-by-step Functionality (Demos)
  - GP Output (GPIO – General Purpose I/O)
  - Read A/D
  - I2C/SMBus (Wire)
    - *Analog EVM/Boosterpack's*
    - *PinMap Housekeeping*
- Optional WiFi
  - STA (station)
  - AP (access point)
- Optional Energia
- UI
  - Button (GP Input - GPIO, add debounce)
  - LCD Display ("Hello")
  - Music
- UI - Serial Interface (i.e. Putty for  echo "Hello World")

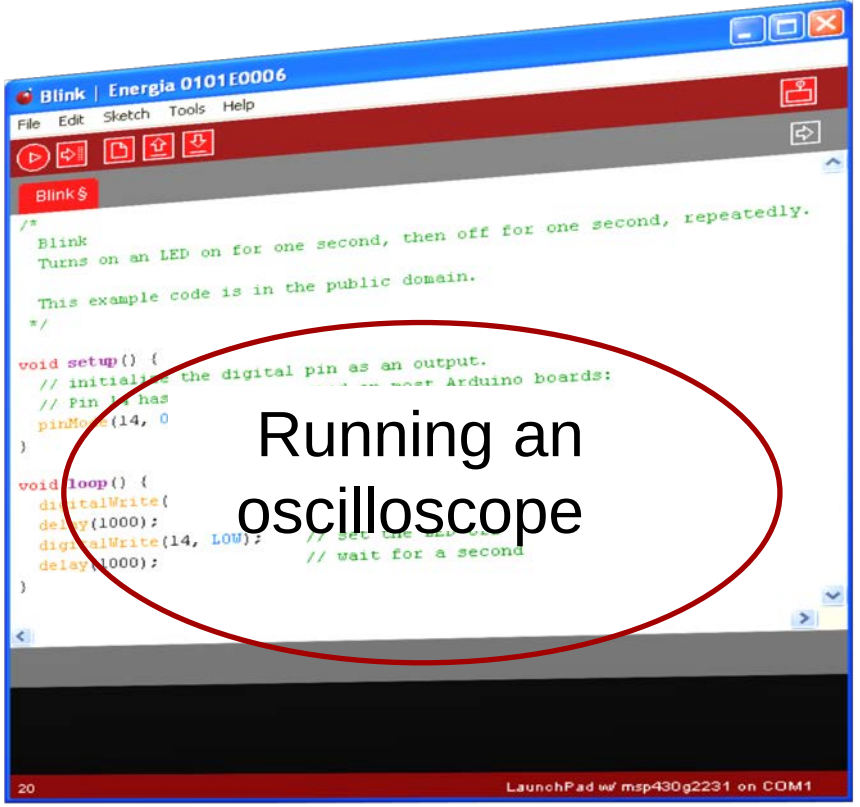
# And Now

Energīa

A simplified programming environment developed by the community (GitHub)

- Open-source electronics prototyping platform for the Texas Instruments LaunchPad (and Boosterpacks)
- Simplify coding to high level functions (Layer over C++)
- Based on the Wiring language
- Brief Energia History and Install http://www.ti.com/tool/ENERGIA
- Getting Started
- Now featuring debugging with integration in CCS

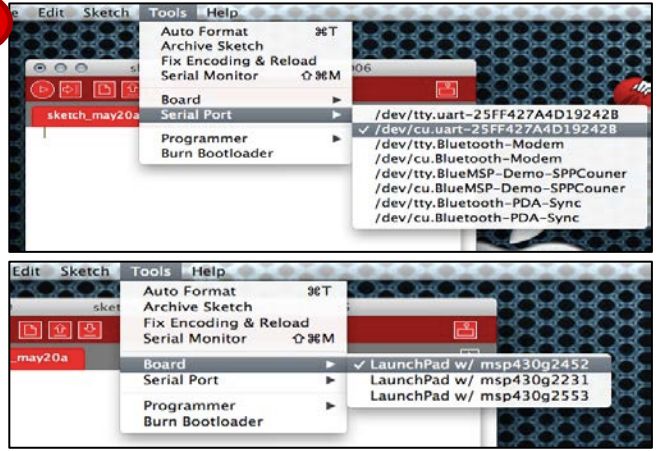
Learn more @ www.energia.nu



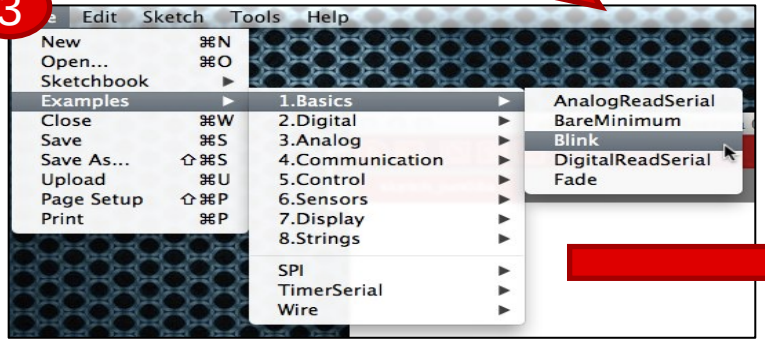
# Blinky Demo

1

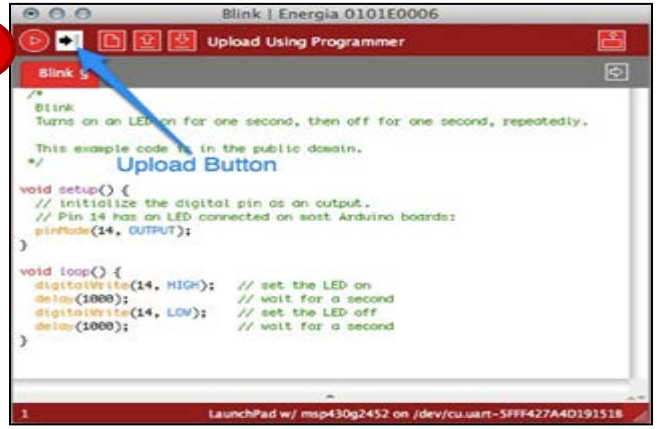

2




3




4




NOTE:

- On **Windows**, ports will be listed as COMXXX port
- On **Linux**, ports will be listed as ttyACM port
- On **Mac OS X** make sure to select the /dev/cu.uart-XXXX NOT the /dev/tty.uart-XXX

Is the green LED blinking? Congratulations! You just successfully uploaded your first sketch.

# Look at the function (digitalwrite) in Code

http://energia.nu/reference/digitalwrite/

Example

```
int ledPin = 14;                    // LED connected to digital pi

void setup()
{
  pinMode(ledPin, OUTPUT);          // sets the digital pin as outp
}

void loop()
{
  digitalWrite(ledPin, HIGH);       // sets the LED on
  delay(1000);                      // waits for a second
  digitalWrite(ledPin, LOW);        // sets the LED off
  delay(1000);                      // waits for a second
}
```

Sets pin 14 to HIGH, makes a one-second-long delay, and sets the pin back to LOW.

# Look at the Library Reference

## Language Reference

Energia programs can be divided in three main parts: *structure*, *values* (variables and constants), and *functions*.

### Structure

#### Program Structure

- setup()
- loop()

#### Control Structures

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

### Variables

#### Constants

- HIGH | LOW
- INPUT | OUTPUT
- INPUT_PULLUP
- INPUT_PULLDOWN
- true | false
- integer constants
- floating point constants

#### Data Types

- void
- boolean
- char
- unsigned char
- byte
- int
- unsigned int

### Functions

#### Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

#### Analog I/O

- analogReference()
- analogRead()
- analogWrite() - *PWM*

#### Advanced I/O

- tone()
- noTone()
- shiftOut()
- shiftIn()
- pulseIn()

# Look at the Library Reference

## Language Reference

Energia programs can be divided in three main parts: *structure*, *values* (variables and constants), and *functions*.

### Structure

#### Program Structure

- setup()
- loop()

#### Control Structures

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

### Variables

#### Constants

- HIGH | LOW
- INPUT | OUTPUT
- INPUT_PULLUP
- INPUT_PULLDOWN
- true | false
- integer constants
- floating point constants

#### Data Types

- void
- boolean
- char
- unsigned char
- byte
- int
- unsigned int

### Functions

#### Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

#### Analog I/O

- analogReference()
- analogRead()
- analogWrite() - *PWM*

#### Advanced I/O

- tone()
- noTone()
- shiftOut()
- shiftIn()
- pulseIn()

# Prototyping Functions

– Step-by-step Functionality (Demos)
  - GP Output (GPIO – General Purpose I/O)
  - Read A/D
  - I2C (Wire)

– Optional WiFi
  - STA (station)
  - AP (access point)

– Optional Energia

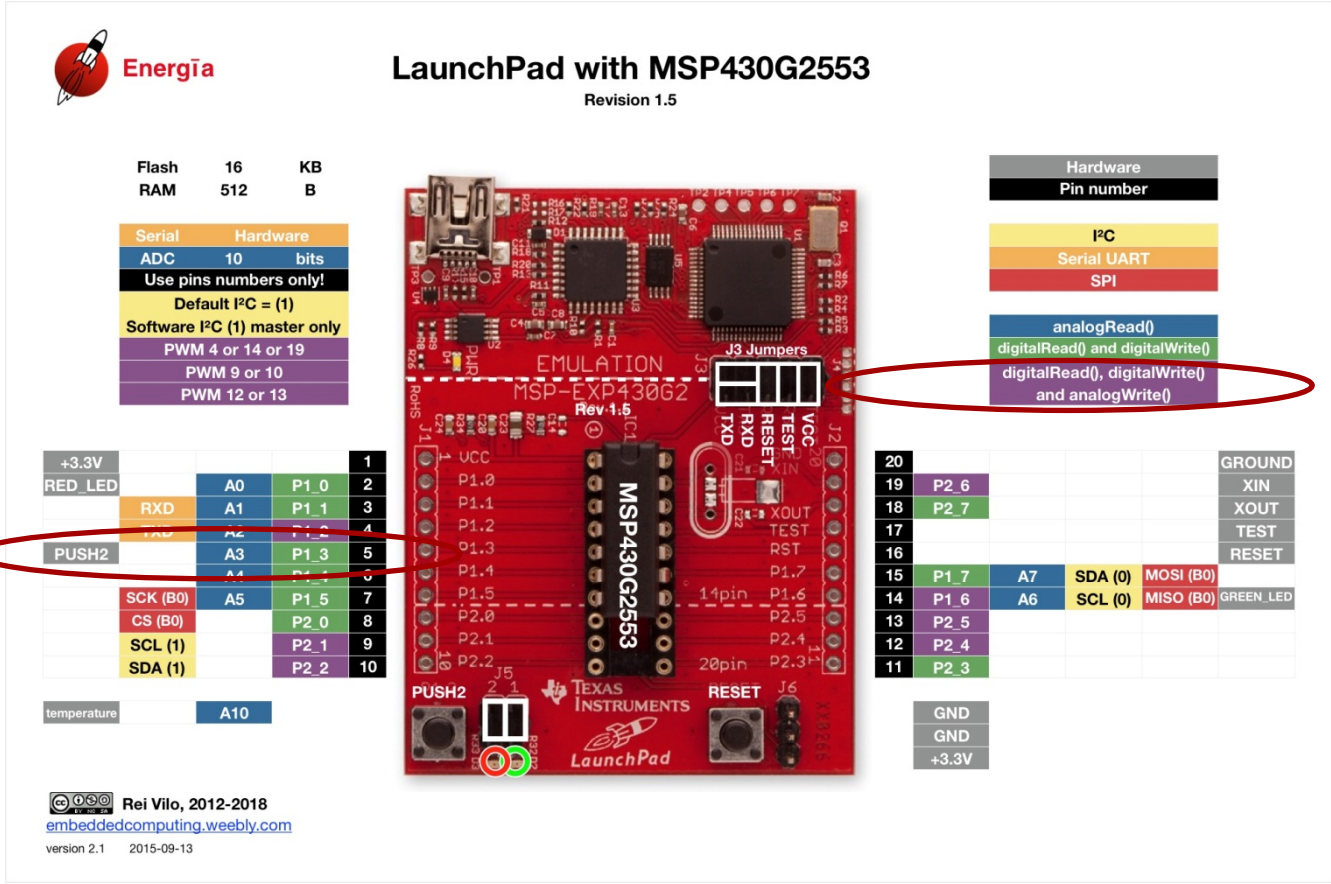
– UI
  - Button (GP Input - GPIO, add debounce)
  - LCD Display (“Hello”)
  - Music

– UI - Serial Interface (i.e. Putty for echo “Hello World”)

# Example Pin Map – Digital I/O

- Button for standalone User Interface

GPIO Input




- http://www.energia.nu/pinmaps/msp-exp430g2/

# Demo - Button

– Step-by-step Functionality (Demo)

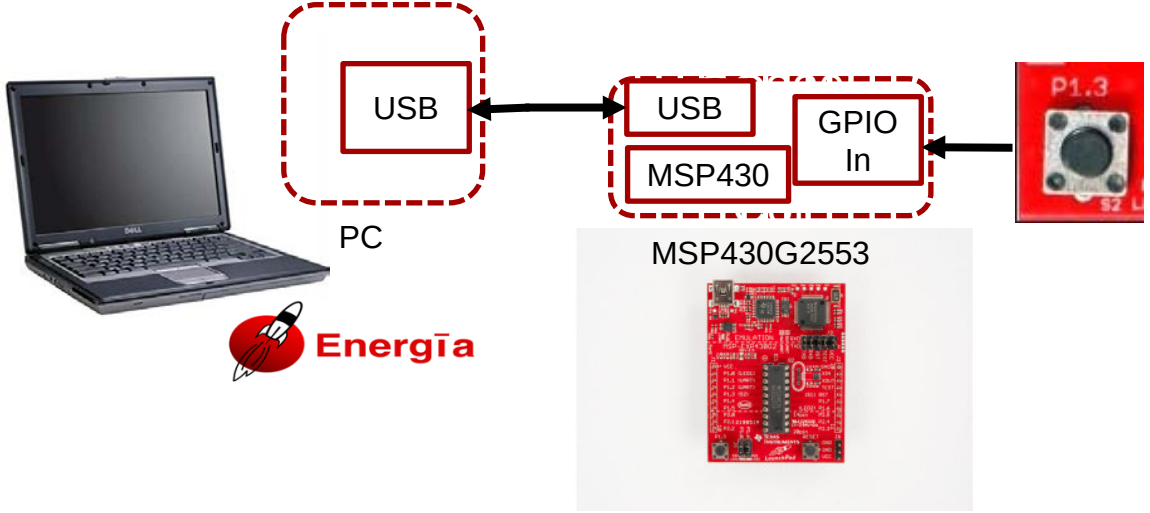
• GP Input (Sample digital signal w General Purpose Input) – Energia Button (edge)

– P1.3 -> PUSH2

»static const uint8_t PUSH2 = 5;

»static const uint8_t P1_3 = 5;

– const int buttonPin = PUSH2;  // the number of the pushbutton pin

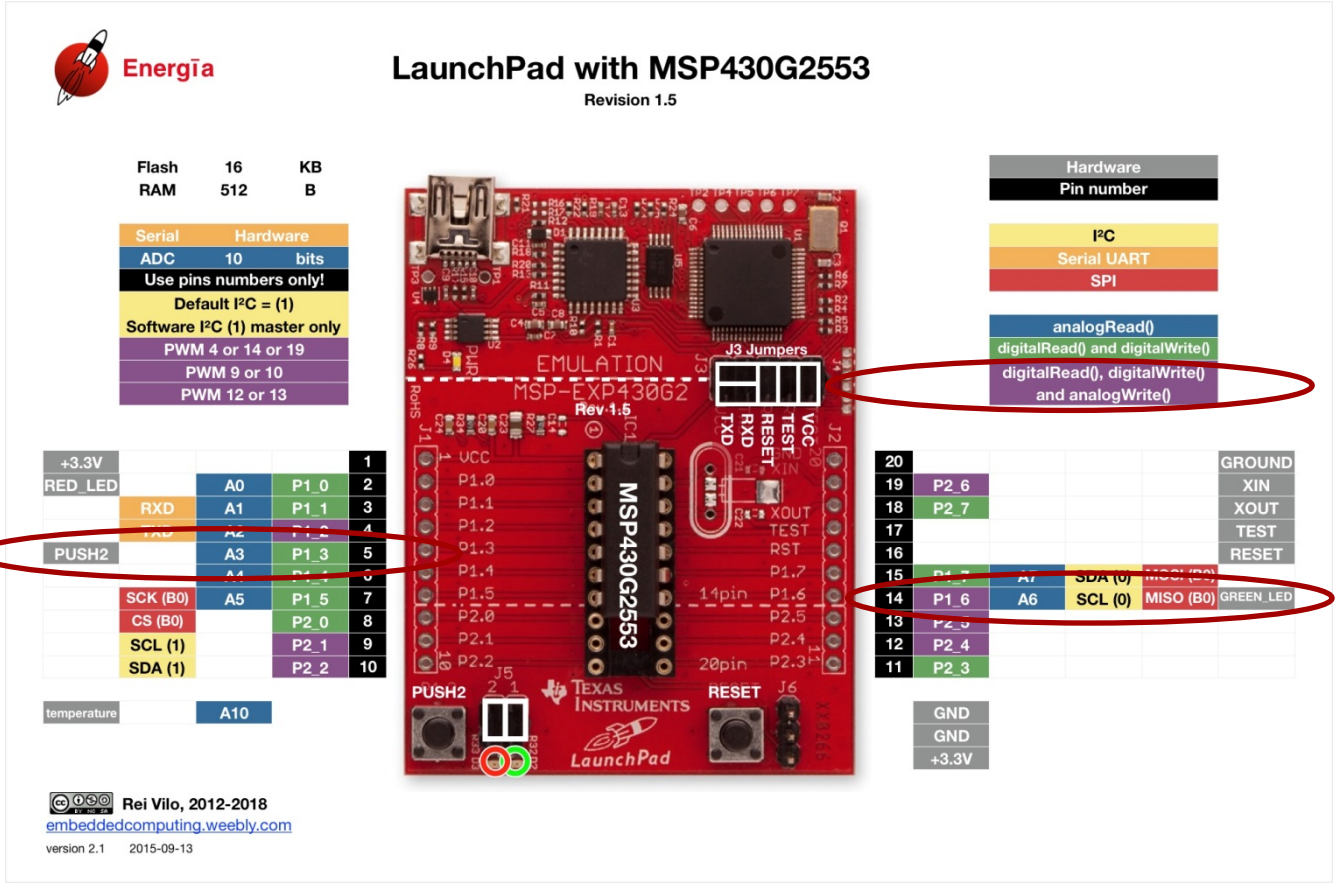




GPIO Input from button

# Example Pin Map – Digital I/O

- Debounce with GPIO Output

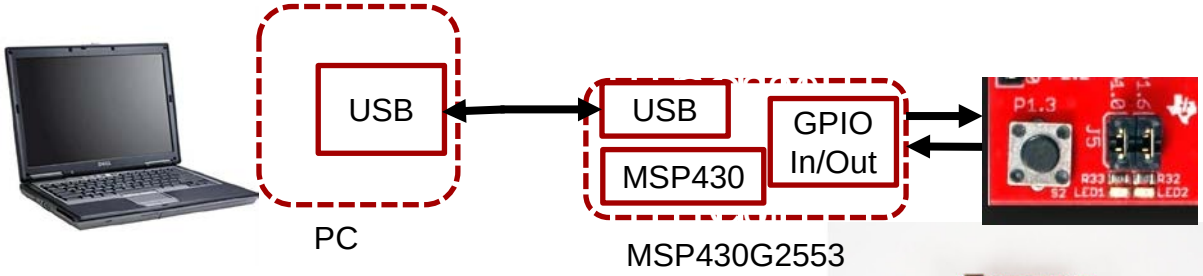



- http://www.energia.nu/pinmaps/msp-exp430g2/

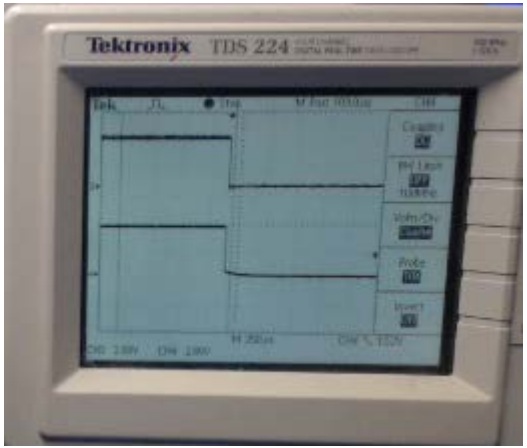
# Demo – Button (Add Debounce)

– Step-by-step Functionality (Demo)

- GP Input (Sample digital signal w General Purpose Input) – Energia Button (edge)
  - P1.3 -> PUSH2
    - »static const uint8_t PUSH2 = 5;
    - »static const uint8_t P1_3 = 5;
  - const int buttonPin = PUSH2;  // the number of the pushbutton pin




  - P1.6 ->GREEN_LED
    - »static const uint8_t GREEN_LED = 14;
    - »static const uint8_t P1_6 = 14
  - const int ledPin =  GREEN_LED




Debounced GPIO Input from button

# Demo – LCD Display

**https://github.com/energia/Energia/tree/master/hardware/msp430/libraries/LCD_Launchpad**

– Step-by-step Functionality (Demo)

• LCD Display (User display interface) – Energia LCD_Launchpad_demo (“Hello”)

***1 Liquid Crystal Display (LCD)***

The MSP430FR4133 LaunchPad features an on-board LCD (see Figure 8). This LCD is driven by the internal LCD driver on the MSP430FR4133 device.

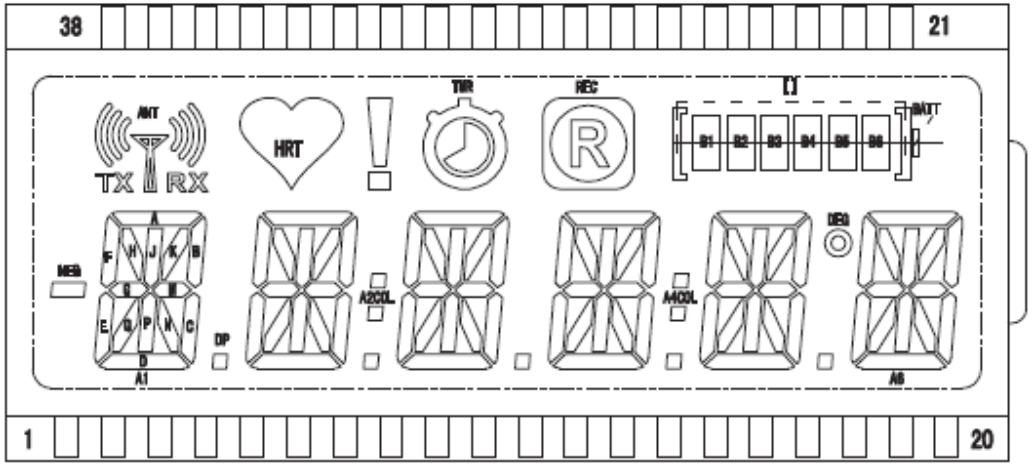


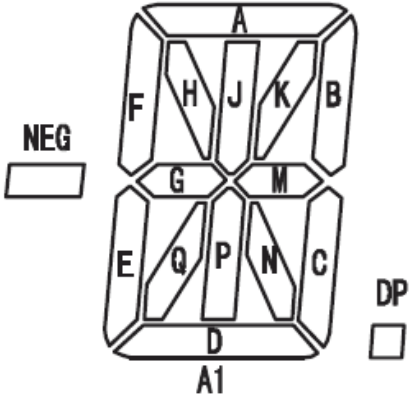

**Figure 8. LCD Segment Layout**

MSP430FR4133 LCD

# Using the Display

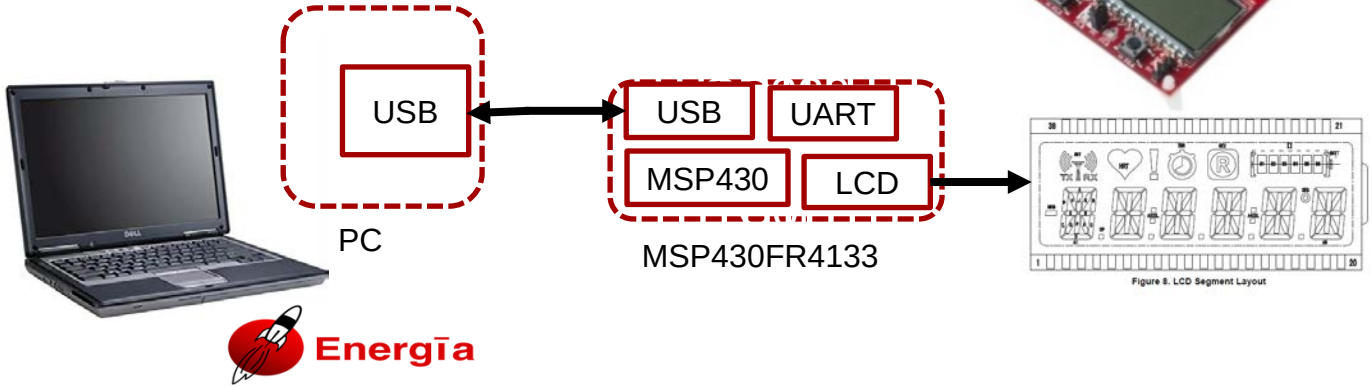
- In the LCD examples, the const char digit array is included. Each of the hexidecimal values refer to which semgents must be turned on to display each respective digit.
- The first byte refers to segments ABCDEFGM for bits [7:0] respectively.
- The most important aspect of muxing this information is this is all done automatically. The user only needs to tell the MSP430 which segments to turn on similar to this array.

```
const char digit[10] =
{
    0xFC, // "0"
    0x60, // "1"
    0xDB, // "2"
    0xF3, // "3"
    0x67, // "4"
    0xB7, // "5"
    0xBF, // "6"
    0xE4, // "7"
    0xFF, // "8"
    0xF7  // "9"
};
```



# Demo - LCD Display

– Step-by-step Functionality (Demo)
  - User Interface (Display) `// initialize LCD`
    - `myLCD.init();`
    - `myLCD.clear();`
    - `myLCD.displayText("Hello");`



# Demo Music – beep

https://github.com/Zolertia/Energia/tree/master/build/shared/examples/9.EducationalBP_MKII/BuzzerImperialMarchTune

– Step-by-step Functionality (Demo)

- Music(Why not?Live a little) – Energia BuzzerImperialMarchTune (yes, Star Wars)
- `beep(note_a, 500);`
- `  beep(note_a, 500);`
- `  beep(note_a, 500);`
- `  beep(note_ff, 350);`
- `  beep(note_cH, 150);`
- `  beep(note_a, 500);`

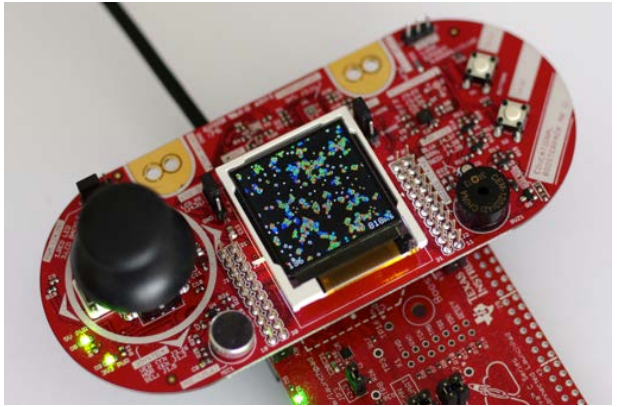
– MSP432P401R Launchpad with

– Educational BoosterPack MKII

– Built-In Examples

»TI Training with MSP432

»YouTube TI video series



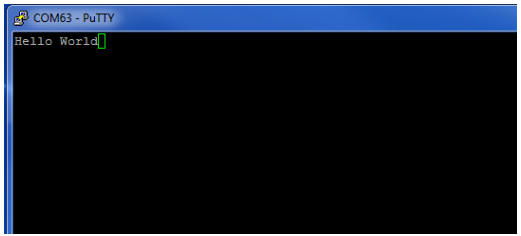
# Prototyping Functions

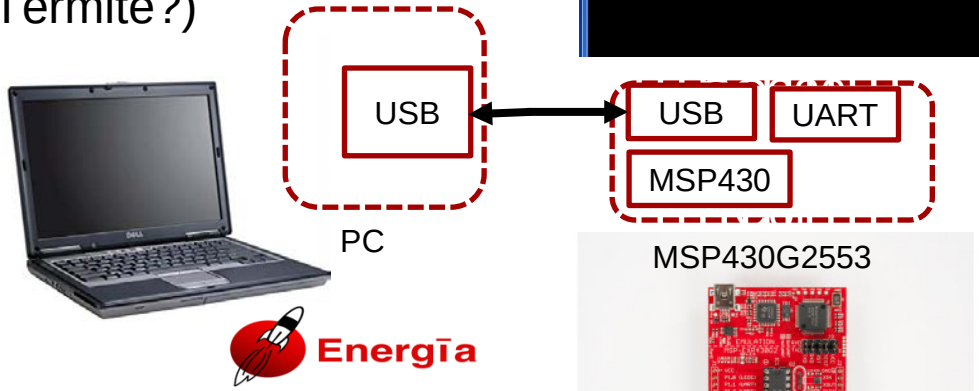
- Step-by-step Functionality (Demos)
  - GP Output (GPIO – General Purpose I/O)
  - Read A/D
  - I2C (Wire)
- Optional WiFi
  - STA (station)
  - AP (access point)
- Optional Energia
- UI
  - GP Input (Button, add debounce)
  - LCD Display (“Hello”)
  - Music
- UI - Serial Interface (i.e. Putty for echo “Hello World”)

# Demo – Serial Interface

– Step-by-step Functionality (Demo)

- Serial Interface (Command Input and Echo) – Energia Serial Event (echo “Hello World”)
  - `// initialize serial:`
  - `Serial.begin(9600);`
  - Putty (Termite?)



# Prototyping Functions Summary with ti.com links

– Step-by-step Functionality (Demo)
  - GP Output (“Blinky” is just General Purpose Input) – Energia Blink
  - Read A/D – Energia ReadAnalogVoltage
  - I2C W/R – Energia Wire Library – Wire write read
  - WiFi (Wireless UI and Cloud with HTTP Server and/or local Access Point)
    – CC3220SF Launchpad and Pin Map
    – (Cloud) Energia (WiFi) Library Examples:
      »WiFi Accessible WebServer (external AP) SimpleWebServerWiFi
      » Make Local AP (own SSID) APWatchConnectDisconnect
  - Button (GP Input, Sample a digital signal with GPIO) – Energia Button

# Prototyping Functions Summary with ti.com links - 2

- LCD Display (User display interface, "Hello") – Energia LCD_Launchpad_demo
  - Display Hardware (MSP430FR4133 Launchpad) and Pin Map
  - Display "Datasheet" (p. 31)
- Music(Why not? Live a little)–Energia BuzzerImperialMarchTune (yes, Star Wars)
  - MSP432P401R Launchpad with
  - Educational BoosterPack MKII
  - Built-In Examples
    - »YouTube TI video series
    - »TI Training with MSP432
- User Interface (Command Input, echo "Hello World") – Energia Serial Event

# Agenda

- Fundamentals (mostly for Analog)
  - Implementing  necessary prototyping functions such clocks/GPIO, Read A/D, I2C/SMBus, etc.
  - Seamless interface of various Analog EVM's for system "proof of concept"
  - Standalone UI - Button (GP Input - GPIO), LCD Display ("Hello"), Music, Serial Interface (Putty)

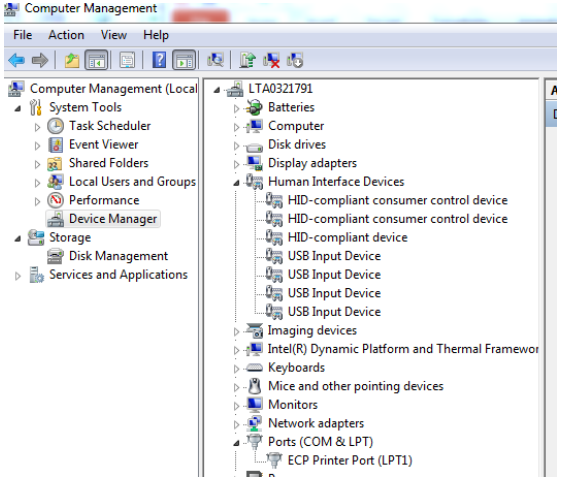
- More UI (i.e. GUI Advanced Comm Tab basically Putty/Serial I/F) - Lessons learned from home networking (if you can setup the WiFi in your house, you can prototype with a few steps)

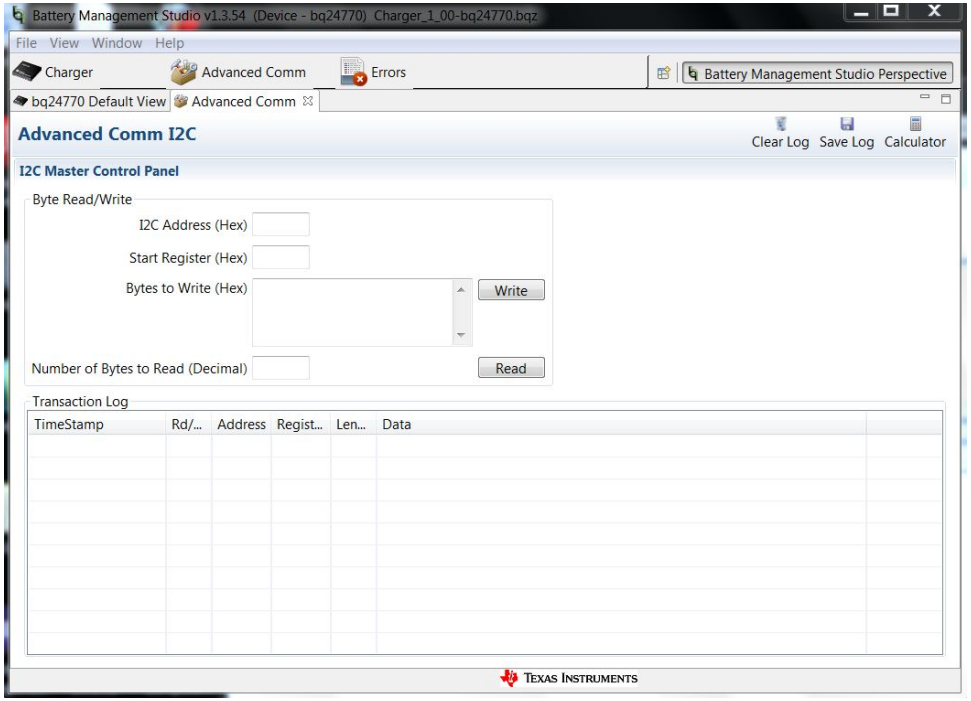
- EP - Embedded prototyping (mostly for Digital)
  - Wired  and Wireless Control
  - Use of TI Cloud Computing Tools for prototype

- Advanced Topics
- Conclusion Demos (Simple and Complex)

# More UI (Revisit serial interface)

- Lessons learned from Analog GUIs
  - Spoiled from using GUI (WiFi Router)
  - Can use Advanced Comm Tab or similar
  - OK to type a few steps in Putty  if needed for development?
    - HID/COM x on PC



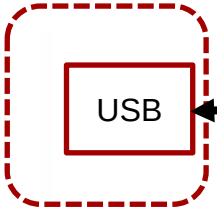

```
Available commands: (Need LOG)
>>I2C_Write 0x12 0x1 0xAB
```

# Demo – Revisit Serial I/F (Count on LCD Display)

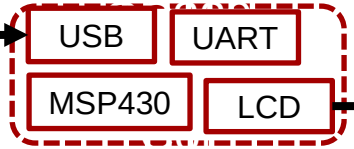
– Step-by-step Functionality (Demo)

- User Interface (Display) – Energia Serial Event (“Hello LCD”)
  - `//initialize LCD`
  - `myLCD.init();`
  - `myLCD.clear();`
  - `myLCD.display;`
  - Putty (Termite) and COMx

```
Hello World
ABCDE - Alpha
12345 - Numeric
Enter a number from 1-10
Number entered, counting up from 7
Counting 78910
Enter a number from 1-10
```




PC




MSP430FR4133

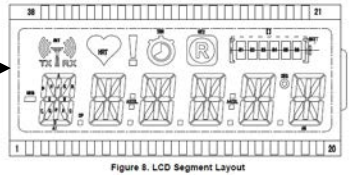


Figure 8. LCD Segment Layout

# Revisit serial interface – Setting up WiFi example

- Smartphone/PC/Router – click on SSID -> Spoiled from using GUI
- Serial Interface - OK to type a few steps in Putty if needed for development?

```
help

================================================================================

Available commands:

help                scan                setpolicy           wlanconnect

wlan_ap_start       wlandisconnect      ping                send

recv                createfilter        enablefilter        disablefilter

deletefilter        enablewowlan        mdnsadvertise       mdnsquery

radiotool           p2pstart            p2pstop             clear

================================================================================
user@CC3220:scan -n 10
-------------------------------------------------------------------------------------------
    |              SSID              |       BSSID       | RSSI  | Ch | Hidden | Security |
-------------------------------------------------------------------------------------------
 1  | TP-LINK_33C4BE                   | 20:25:64:f5:a9:b8 | -70   | 1  | NO     | WPA/WPA2 |
 2  | 5TH AVE Secure                 | ac:86:74:ad:1e:03 | -86   | 1  | NO     | WPA2     |
 3  | halekoa75                      | ac:a3:1e:f9:11:c0 | -59   | 11 | NO     | WPA2     |
 4  | externalhotspot84              | ac:a3:1e:f9:11:c1 | -59   | 11 | NO     | WPA2     |
 5  | net4guest                      | ac:a3:1e:f9:11:c2 | -59   | 11 | NO     | WPA2     |
 -------------------------------------------------------------------------------------------scan -n 1

user@CC3220: wlanconnect –s TP-LINK_33C4BE

[WLAN EVENT] STA Connected to the AP: TP-LINK_33C4BE , BSSID: f8:d1:11:33:c4:be

[NETAPP EVENT] IP set to: IPv4=192.168.1.100 , Gateway=192.168.1.1
```

CC3220 SimpleLink SDK Wi-Fi (WiFi STA)

Simplink SDK Example: Network Terminal

# Agenda

- Fundamentals (mostly for AFA)
  - Implementing  necessary prototyping functions such clocks/GPIO, Read A/D, I2C, etc.
  - Seamless interface of various Analog EVM's for customer "proof of concept"
  - Standalone UI - Button (GP Input - GPIO), LCD Display ("Hello"), Music, Serial Interface (Putty)

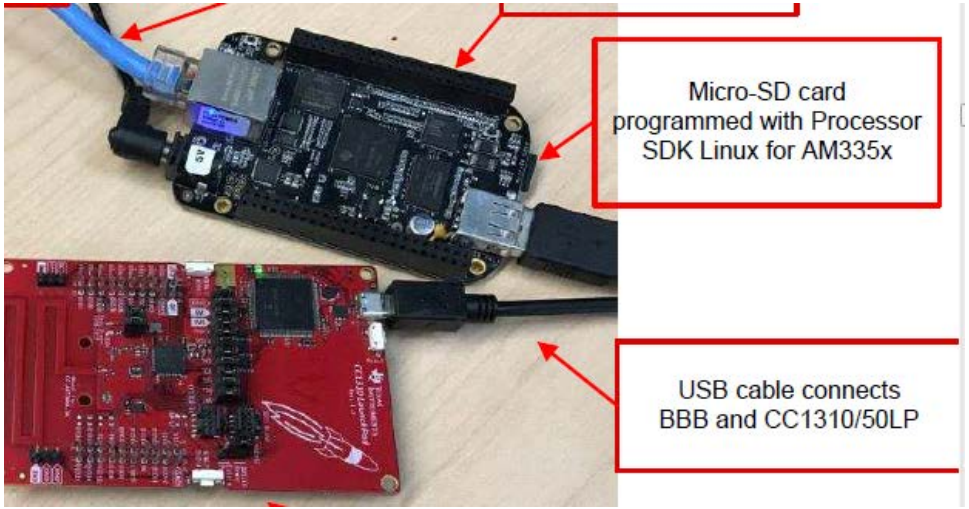
- More UI (i.e. GUI Advanced Comm Tab basically Putty/Serial I/F) - Lessons learned from home networking (if you can setup the WiFi in your house, you can prototype with a few steps)

- EP - Embedded prototyping (mostly for DFA)
  - Wired  and Wireless Control
  - Use of TI Cloud Computing Tools for prototype

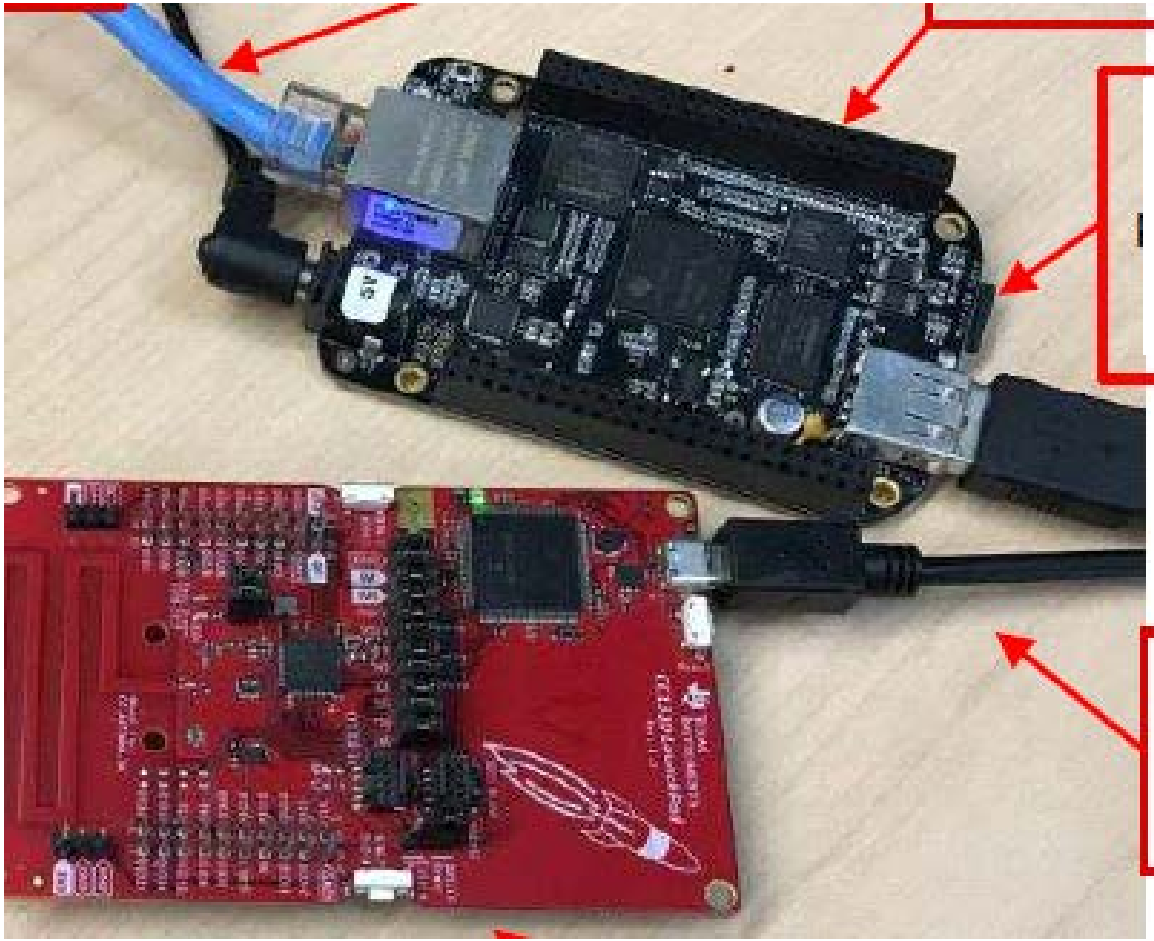
- Advanced Topics
- Conclusion Demos (Simple and Complex)

# Embedded (App Processor/MPU) Prototyping

- Beaglebone to Launchpads like p. 10 http://www.ti.com/lit/ug/tiduci9c/tiduci9c.pdf
- Linux expertise and programming not necessary. Just type on a command line. Can even use a Windows PC): http://beagleboard.org/getting-started#step2




- Can use a Windows PC to Get Started.
- Connect to BBB/BBGW using:
  - Putty SSH connection:
  - Or even BBGW over WiFi:



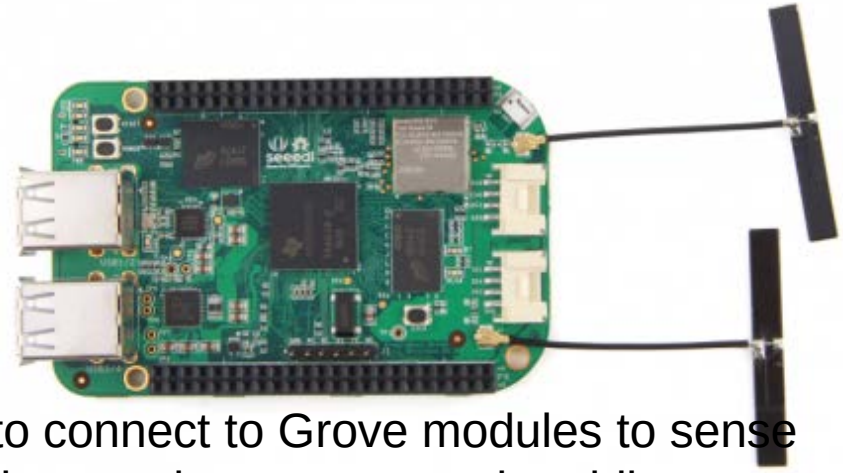
On-board eMMC programmed with Debian Linux for AM335x (more like Ubuntu) for prototyping

- BeagleBone Green Wireless (or Beaglebone Black if you want HDMI output/Gnome Desktop and don't mind adding a USB hub and/or Wilink8 Cape)

USB cable connects BBB and CC1310/50LP

# Embedded (AP/MPU) prototyping - BBGW

- The SeeedStudio BeagleBone Green Wireless, like the BeagleBoard.org BeagleBone Black, runs on a Texas Instruments Sitara AM335x processor.



- The fundamental differences are:
  - USB host with 4-port hub
  - Two Grove connectors that make it easy to connect to Grove modules to sense orientation, location, distance, water, touch, sound, temperature, humidity, barometric pressure, heart rate, finger print and much more
  - WiFi 802.11 b/g/n 2.4GHz (optional)
  - Replaced 5V barrel and miniUSB with a single microUSB connectionboards
  - Removal of the microHDMI video output to save cost
- http://beagleboard.org/green-wireless

# Embedded (App Processor/MPU) Prototyping

- ttyACMx is just COMx from Windows (use Teraterm or Putty):
  http://www.ti.com/tool/tidep0084 or p. 15
  http://www.ti.com/lit/ug/tiduci9c/tiduci9c.pdf

*ls -l /dev/ttyACM**

at the BeagleBone Black console. There are two ttyACM devices that correspond to the serial ports from the CC13x0 or CC13x2 LaunchPad Development Kit (similar to Figure 8).

```
root@am437x-evm:~# ls -l /dev/ttyACM*
crw-rw----    1 root     dialout   166,   0 Oct 24 16:52 /dev/ttyACM0
crw-rw----    1 root     dialout   166,   1 Oct 24 16:52 /dev/ttyACM1
root@am437x-evm:~#
```

**Figure 8. /dev/ttyACM0 Device Check**

# Embedded prototyping - Wired control

- Wired PC Control of Embedded Application Processor from Beaglebone command line

```
debian@beaglebone:~$ screen /dev/ttyACM0 115200
```

- Also (Need `sudo apt-get install picocom` )

```
debian@beaglebone:~$ sudo picocom -b 115200 /dev/ttyACM0 )
```

```
Hello World
ABCDE - Alpha
12345 - Numeric
Enter a number from 1-10
Number entered, counting up from 7
Counting 78910
Enter a number from 1-10
```

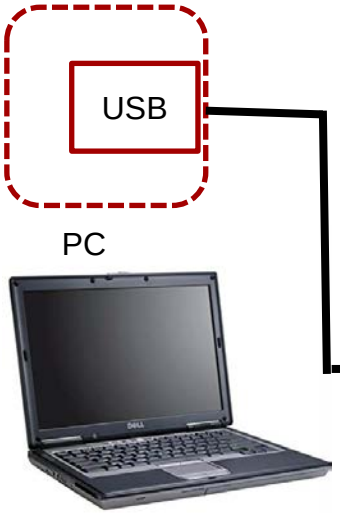
Firewall version:

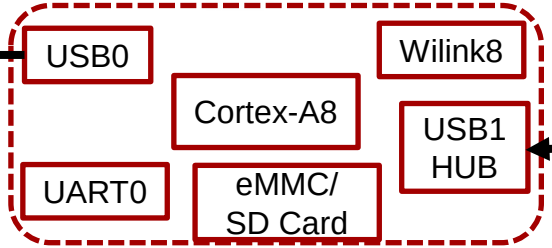
```
sudo bash -c 'http_proxy="http://yourproxyserver.yourcompany.com:80/" apt-get update'
sudo bash -c 'http_proxy="http://yourproxyserver.yourcompany.com:80/" apt-get install picocom'
```

# Embedded prototype demo Wired Control of LCD

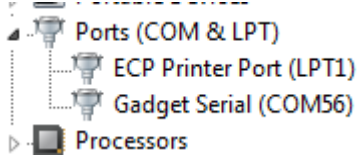
beagleboard.org/getting-started#step3

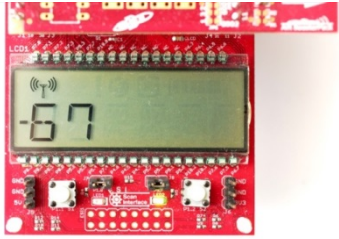
| IP Address | Connection Type | Operating System(s) | Status |
|---|---|---|---|
| 192.168.7.2 | USB | Windows | Active |
| 192.168.6.2 | USB | Mac OS X, Linux | Inactive |



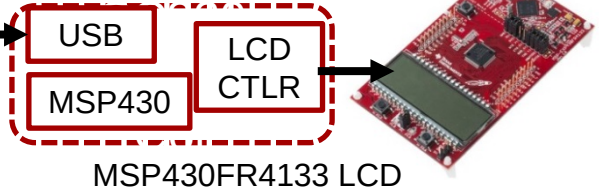



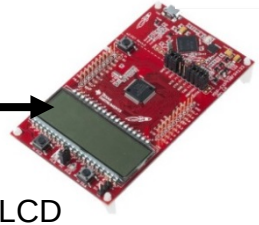

BB Green Wireless AM3358
(WiFi STA)



Energia



MSP430FR4133 LCD

# Agenda

- Fundamentals (mostly for Analog)
  - Implementing  necessary prototyping functions such clocks/GPIO, Read A/D, I2C, etc.
  - Seamless interface of various Analog EVM's for customer "proof of concept"
  - Standalone UI - Button (GP Input - GPIO), LCD Display ("Hello"), Music, Serial Interface (Putty)

- More UI (i.e. GUI Advanced Comm Tab basically Putty/Serial I/F) - Lessons learned from home networking (if you can setup the WiFi in your house, you can prototype with a few steps)

- EP - Embedded prototyping (mostly for Digital)
  - Wired  and Wireless Control
  - Use of TI Cloud Computing Tools for prototype

- Advanced Topics
- Conclusion Demos (Simple and Complex)

# Revisit Setting up WiFi example -> Enable Wireless Control

- Wireless Control of prototype (Impress your friends with no PC, just phone or tablet) using CC3220SF Launchpad and Code Composer Example Network Terminal to setup a soft Access Point (AP):

```
========================================================================

Available commands:

help                scan                setpolicy           wlanconnect

wlan_ap_start       wlandisconnect      ping                send

recv                createfilter        enablefilter        disablefilter

deletefilter        enablewowlan        mdnsadvertise       mdnsquery

radiotool           p2pstart            p2pstop             clear

========================================================================
user@CC3220:scan -n 10
-------------------------------------------------------------------------------------------
    |               SSID                |       BSSID       | RSSI  | Ch | Hidden | Security |
-------------------------------------------------------------------------------------------
 1  | CBCI-3418-2.4                     | 20:25:64:f5:a9:b8 | -70   | 1  | NO     | WPA/WPA2 |
 2  | 5TH AVE Secure                    | ac:86:74:ad:1e:03 | -86   | 1  | NO     | WPA2     |
 3  | halekoa75                         | ac:a3:1e:f9:11:c0 | -59   | 11 | NO     | WPA2     |
 4  | externalhotspot84                 | ac:a3:1e:f9:11:c1 | -59   | 11 | NO     | WPA2     |
 5  | net4guest                         | ac:a3:1e:f9:11:c2 | -59   | 11 | NO     | WPA2     |
 -------------------------------------------------------------------------------------------
user@CC3220:

wlan_ap_start -s CC3220_joe
```

CC3220 SimpleLink SDK Wi-Fi (WiFi AP)

# Embedded prototype demo Wireless Control of LCD

Smartphone/iPAD
(WiFi **STA**)


Termius App SSH

USB Power

USB0

Cortex-A8

Wilink8

USB1 HUB

UART0

eMMC/ SD Card

BB Green Wireless AM3358
(WiFi **STA**)

USB

RF

Cortex-M4

CC3220 SimpleLink SDK Wi-Fi
(WiFi **AP**)


Energia

USB

MSP430

LCD CTLR

MSP430FR4133 LCD

# Agenda

- Fundamentals (mostly for Analog)
  - Implementing  necessary prototyping functions such clocks/GPIO, Read A/D, I2C, etc.
  - Seamless interface of various Analog EVM's for customer "proof of concept"
  - Standalone UI - Button (GP Input - GPIO), LCD Display ("Hello"), Music, Serial Interface (Putty)

- More UI (i.e. GUI Advanced Comm Tab basically Putty/Serial I/F) - Lessons learned from home networking (if you can setup the WiFi in your house, you can prototype with a few steps)

- EP - Embedded prototyping (mostly for Digital)
  - Wired  and Wireless Control
  - Use of TI Cloud Computing Tools for prototype

- Advanced Topics
- Conclusion Demos (Simple and Complex)

# TI Cloud Computing with ti.com links

- TI Cloud Computing https://dev.ti.com/
- TI Resource Explorer (TIREX)
  - Energia Cloud Examples (Files, Docs)
    - Built-In - Blink, ReadAnalog Voltage, etc. EducationalBP_MKII
    - Library – Adafruit_TMP007, Wire, WiFiMKII
- Build and Run Example?
  - Usually Flash Image with Uniflash Cloud - Need **Image Creator** (Security)
  - TI Resource Explorer (TIREX) imports to Code Composer Studio (CCS Cloud)
- Demo(s)
  - Energia Cloud for WiFiMKII APWatchConnectDisconnect)
  - CCS Cloud for Network Terminal

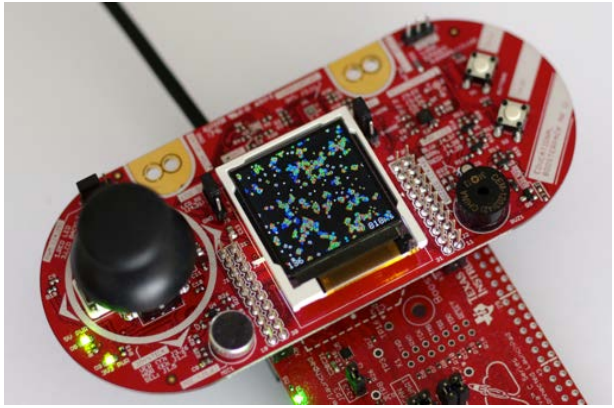
CC3220SF SimpleLink SDK Wi-Fi (WiFi **AP**)

# Advanced Topics

- Energia Libraries
  - Educational BoosterPack MKII (Built-In Energia18)
  - Neopixel/WS2811 Driver (Copy library into Energia18)
  - Seeed Grove Starter Kit (Add from GitHub)
  - TI BLE for CC2650 Boosterpack  (Add from GitHub)
- Layered Energia Code
- Single-step Debug of Energia using CCS (even assembly language)
- Porting AdaFruit Neopixel to MSP432
- Allows Energia<->CCS fluid transitions
- USB (Native)  UART (MSP430F5529 Launchpad)  Energia USBSerialExample
- Multiblink(Use of TI-RTOS) Energia Multitasking (Red+Green+Blue = White LED)

# Educational BoosterPack MKII Libraries

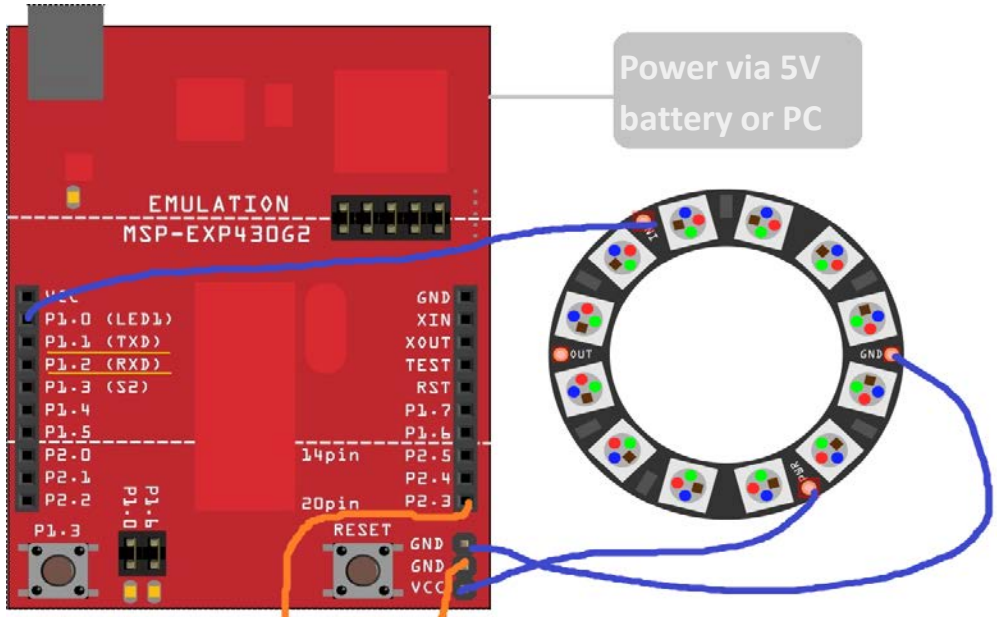
- Educational BoosterPack MKII (Built-In Energia18)
  - Built-In Examples
    - »YouTube TI video series
    - »TI Training with MSP432



- Educational BoosterPack MKII – LCD Joystick:
  - http://dev.ti.com/tirex
  - C:\Energia\energia-1.6.10E18-windows\energia-1.6.10E18\examples\09.EducationalBP_MKII\LCD_Joystick

# Neopixel Libraries

- Neopixel/WS2811 Driver (Copy library into Energia18)
  - GitHub)



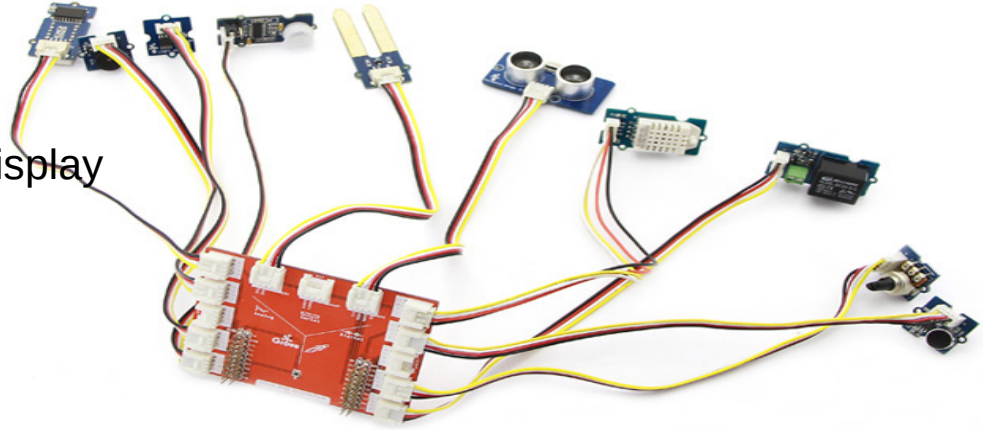

| MSP430G2553 Launchpad | Neopixel LED Ring/Strip |
|---|---|
| P1.0 | IN |
| VCC | PWR |
| GND | GND |

  - Neopixel from https://github.com/ILAMtitan/WS2811Driver to:
    - C:\Energia\energia-1.6.10E18-windows\energia-1.6.10E18\libraries\WS2811Driver

# Grove Starter Kit for Launchpad by SEEED

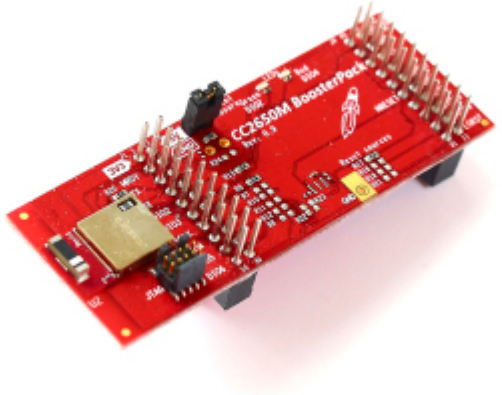
https://www.seeedstudio.com/Grove-Starter-Kit-for-LaunchPad-p-2178.html

- 1x Grove Base Boosterpack
- Digital Grove Modules
  - 1x Grove Relay
  - 1x Grove Buzzer
  - 1x Grove 4 Digit Seven Segment Display
  - 1x Grove PIR Sensor
  - 1x Grove Ultrasonic Range Finder
- Analog Grove Modules
  - 1x Grove Light Sensor
  - 1x Grove Rotary Angle Sensor (Potentiometer)
  - 1x Grove Sound Sensor
  - 1x Grove Moisture Sensor
  - 1x Grove Temperature Humidity Sensor



# TI BLE for CC2650 Boosterpack Libraries

– TI BLE for CC2650 Boosterpack (Add from GitHub):
– https://github.com/ti-simplelink/ble_energia
– MSP430F5529 or MSP432P401



# Energia Libraries - detailed

- Use with different method according to instructions
  - Educational BoosterPack MKII http://www.ti.com/tool/BOOSTXL-EDUMKII in c:\energia-1.6.10E18\examples\09.EducationalBP_MKII
  - Neopixel/WS2811 Driver (Adapting Adafruit Neopixel to MSP430G2xx Launchpad) from forum.430oh in C:\energia-1.6.10E18\libraries\WS2811Driver
  - Seeed Grove Starter Kit from GitHub in …\My Documents\Energia\g2553\LaunchPad_Kit-master\LaunchPad_Kit-master\libraries (haven't tried:http://www.energia.nu/guide/tutorials/other/sidekick/
  - TI BLE for CC2650 Boosterpack from https://github.com/ti-simplelink/ble_energia in C:\energia-1.6.10E18\libraries\BLE

- If you wish to use these libraries please copy the folders into your Energia libraries folder. Windows: My Documents\Energia\libraries Mac & Linux: ~/Documents/Energia/Libraries
- Libraries installed in this folder will remain on your computer even when installing new versions of Energia IDE and the IDE will know to look in this folder automatically for new libraries.
- You will need to restart the Energia IDE for your library list to refresh. Once you reopen Energia you can view the example code stored in the library under File > Examples > [library name]

# Advanced Topics

- Energia Libraries
  - Educational BoosterPack MKII (Built-In Energia18)
  - Neopixel/WS2811 Driver (Copy library into Energia18)
  - Seeed Grove Starter Kit (Add from GitHub)
  - TI BLE for CC2650 Boosterpack  (Add from GitHub)
- Layered Energia Code
- Single-step Debug of Energia using CCS (even assembly language)
- Porting AdaFruit Neopixel to MSP432
- Allows Energia<->CCS fluid transitions
- USB (Native)  UART (MSP430F5529 Launchpad)  Energia USBSerialExample
- Multiblink(Use of TI-RTOS) Energia Multitasking (Red+Green+Blue = White LED)

# Layered Code in Energia

## Code Examples (int. temp sensor) – Combine Button with adc10_temp

setup:

```
ADC10CTL1 = INCH_10 + ADC10DIV_3;         // Temp Sensor ADC10CLK/4
  ADC10CTL0 = SREF_1 + ADC10SHT_3 + REFON + ADC10ON + ADC10IE;
  __enable_interrupt();                     // Enable interrupts.
  TACCR0 = 30;                              // Delay to allow Ref to settle
  TACCTL0 |= CCIE;                          // Compare-mode interrupt.
  TACTL = TASSEL_2 | MC_1;
```

loop:

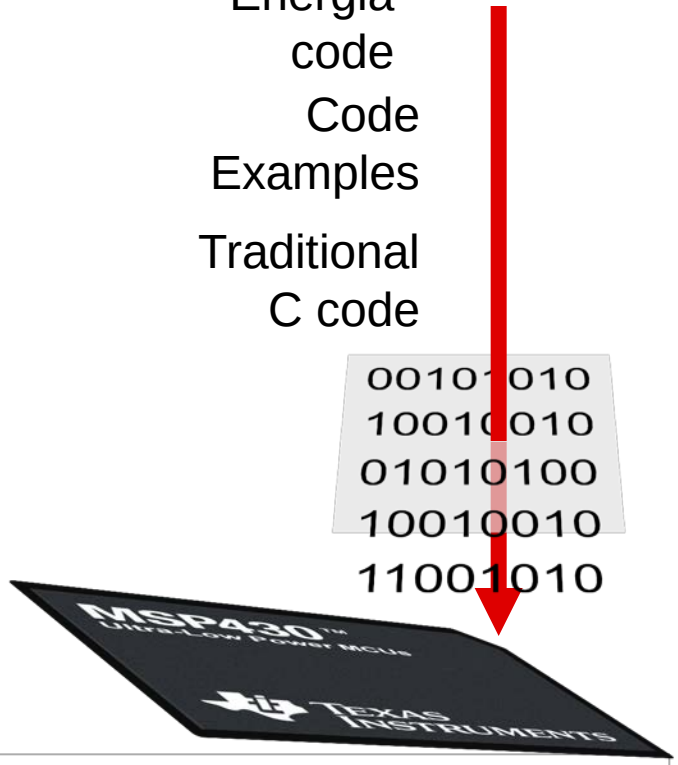
```
   ADC10CTL0 |= ENC + ADC10SC;             // Sampling and conversion start
    __bis_SR_register(CPUOFF + GIE);        // LPM0 with interrupts enabled
    // oF = ((A10/1024)*1500mV)-923mV)*1/1.97mV = A10*761/1024 - 468
    temp = ADC10MEM;
    IntDegF = ((temp - 630) * 761) / 1024;
    Serial.print("Degrees in Farenheit: ");
    Serial.println(IntDegF);
    delay(100); // delay in between reads for stability

   // oC = ((A10/1024)*1500mV)-986mV)*1/3.55mV = A10*423/1024 - 278
        temp = ADC10MEM;
        IntDegC = ((temp - 673) * 423) / 1024;
        Serial.print("Degrees in Centigrade: ");
        Serial.println(IntDegC);
      delay(100); // delay in between reads for stability
```

Energia code

Code Examples

Traditional C code




- Comprehensive parsing allows layering code in Energia
- How about A/D (fixed vs. floating point)?

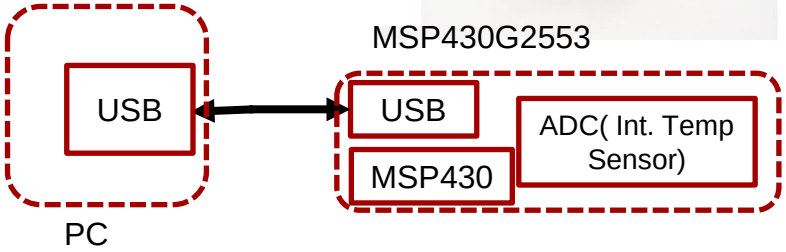
# Demo – Read A/D (Temperature Sensor with Button)

– Step-by-step Functionality (Demo)

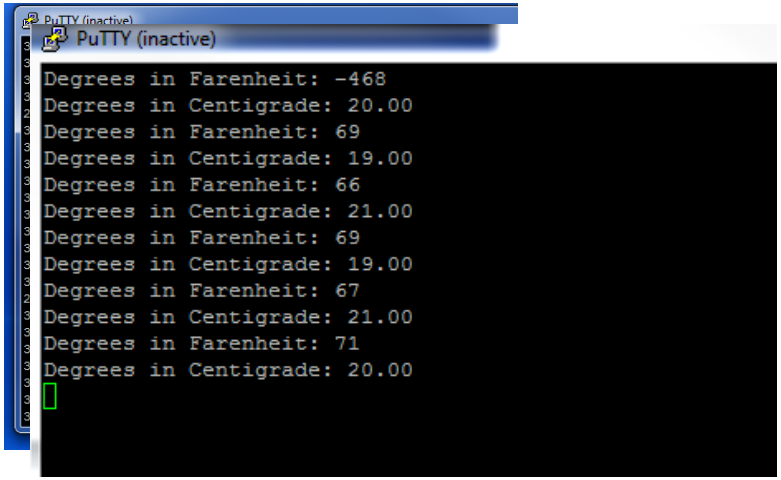
• Read A/D – Was Energia ReadAnalog (level), Now Button (internal temperatature)

– (A3)  P1.3


MSP430G2553




PC




```
// You can compare the size of the code by running the program using int and then running with float
// You will see ~4k bytes for int vs ~6k bytes for float just by changing the datatype, quite astonishing.
  float voltage = sensorValue * (3.0 / 1023.0);
```

# Layered Code in Energia

```
boolean ping() {
  SlPingReport_t report;
  SlPingStartCommand_t pingCommand;
  pingCommand.Ip = SL_IPV4_VAL(192,168,1,1); // destination IP address is 10.1.1.200
  pingCommand.PingSize = 150;                // size of ping, in bytes
  pingCommand.PingIntervalTime = 100;        // delay between pings, in milliseconds
  pingCommand.PingRequestTimeout = 1000;     // timeout for every ping in milliseconds
  pingCommand.TotalNumberOfAttempts = 1;     // max number of ping requests. 0 - forever
  pingCommand.Flags = 0;                     // report only when finished
  sl_NetAppPingStart( &pingCommand, SL_AF_INET, &report, pingRes);
  Serial.print("\nSending ping to....");
  Serial.println(gateway);
```

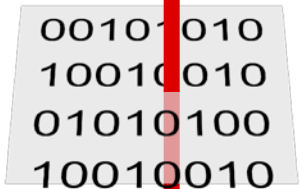
## Simplelink WiFi “Ping”

```
Pinging 192.168.1.1 with 150 bytes of data:
: Reply from 192.168.1.1 Packets sent:1 Packets
   received:1
Approximate round trip times in milli-seconds:
Min =  89 Max = 89 Ave = 44
```

Energia code

Driver Code

Traditional C code



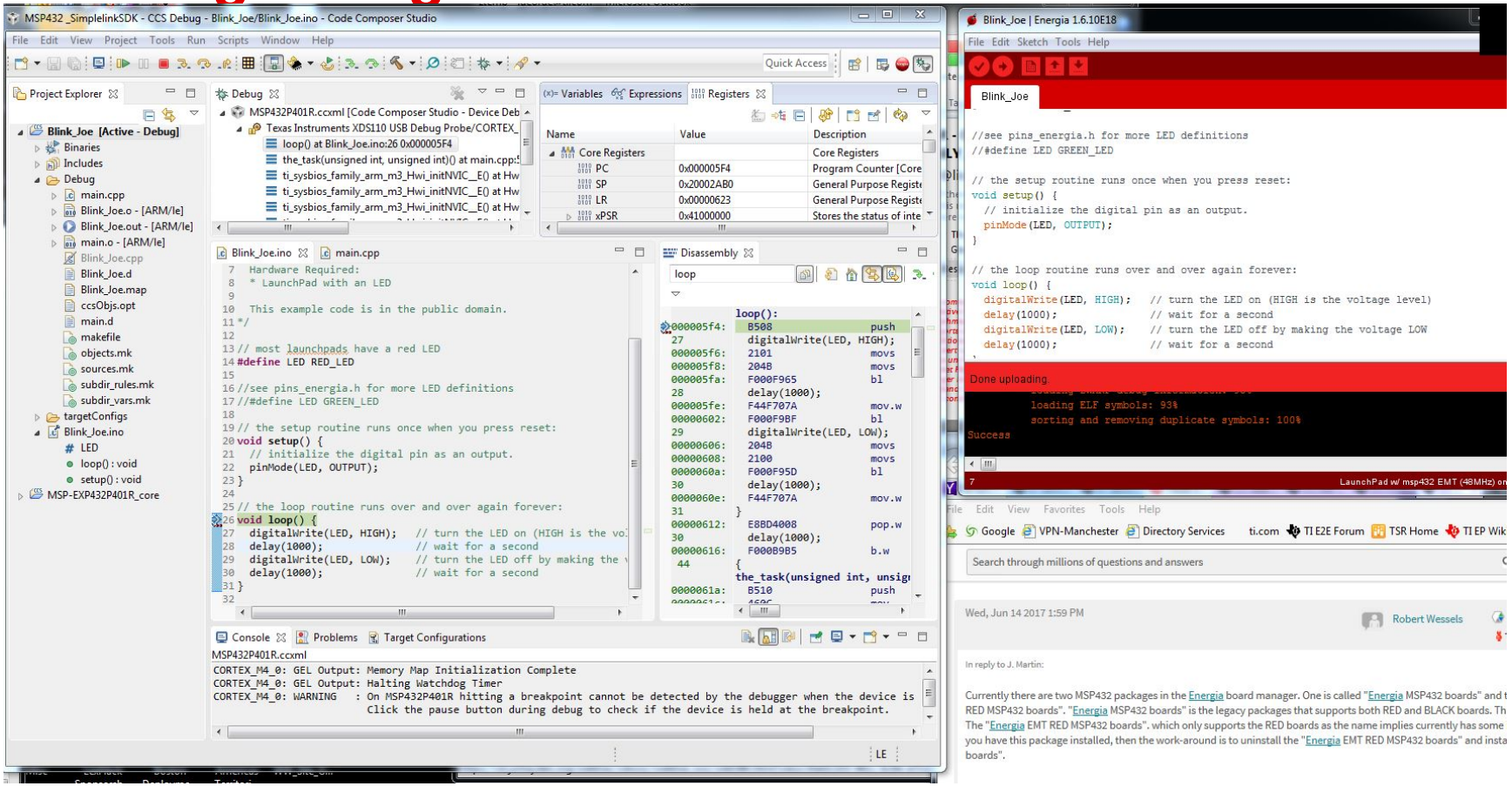


- Comprehensive parsing allows layering code in Energia
- Embed Code Examples, Driverlib or even RF Stack Commands)

# Advanced Topics

- Energia Libraries
  - Educational BoosterPack MKII (Built-In Energia18)
  - Neopixel/WS2811 Driver (Copy library into Energia18)
  - Seeed Grove Starter Kit (Add from GitHub)
  - TI BLE for CC2650 Boosterpack  (Add from GitHub)
- Layered Energia Code
- Porting AdaFruit Neopixel to MSP432 (from MSP430 WS2811 Neopixel Library)
- Single-step Debug of Energia using CCS (even assembly language)
- Allows Energia<->CCS fluid transitions
- USB (Native)  UART (MSP430F5529 Launchpad)  Energia USBSerialExample
- Multiblink(Use of TI-RTOS) Energia Multitasking (Red+Green+Blue = White LED)

# Debug Energia with CCS (Neopixel CCS7_Energia18_MSP432_LP)



# Debug Energia with CCS (Assembly)

MSP432 _SimplelinkSDK - CCS Debug - C:\vmimages\vmwareshare\Energia\energia-1.6.10E18-windows\energia-1.6.10E18\libraries\Adafruit_NeoPixel\Adafruit_NeoPixel.cpp - Code Composer Studio

File Edit View Project Run Tools Scripts Window Help

Quick Access

**Project Explorer**

- Atmo_joe_ccs
- **Blink_Joe [Active - Debug]**
- MSP-EXP432P401R_Adafruit_NeoPixel
  - Includes
  - Debug
  - Adafruit_NeoPixel.cpp
  - esp8266.c
  - examples
- MSP-EXP432P401R_core

**Debug**

- MSP432P401R.ccxml [Code Composer Studio - Device Debugging]
  - Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended)
    - Adafruit_NeoPixel::show(class Adafruit_NeoPixel *)() at Adafruit_NeoPixel.cpp:
    - theaterChase(unsigned long, unsigned char)() at Blink_Joe.ino:108 0x0000066C
    - loop() at Blink_Joe.ino:52 0x0000084E
    - the_task(unsigned int, unsigned int)() at main.cpp:51 0x000008B8
    - ti_sysbios_knl_Task_exit__E() at Task.c:455 0x000022A4
    - ti_sysbios_knl_Task_exit__E() at Task.c:455 0x000022A4 (next frame is identical)

**Variables / Expressions / Breakpoints**

| Identity | Name | Condition | Count | |
|---|---|---|---|---|
| 0x000005ec | Breakpoint | | 0 (0) | Ren |
| 0x000005f4 | Breakpoint | | 0 (0) | Ren |
| 0x00004678 | Breakpoint | | 0 (0) | Remain Halted |
| 0x000046a8 | Breakpoint | | 0 (0) | Remain Halted |
| Adafruit_N | Breakpoint | | 0 (0) | Remain Halted |
| Atmo_Joe.i | Breakpoint | | 0 (0) | Remain Halted |
| Blink_Joe.ir | Breakpoint | | 0 (0) | Remain Halted |

**Adafruit_Ne... / 0x4b8e / main.cpp / wiring_digi...**

```
1272 //                  bitmask = pinMode(75, OUTPUT);
1273 //                    bitmask = PIN_TO_BITMASK(pin);
1274 //                    volatile uint8_t      *reg = PIN_TO_BASEREG(
1275 //                  volatile uint8_t *reg = portSetRegister(pin);
1276                    bitmask = (1 & 0xff);
1277 //                   volatile uint8_t      *reg = 0x40004C02;
1278                    //jg 12/5/17 P2.0 is pin 75. 0x40004C03 RED_L
1279                    //jg 12/5/17 P1.0 is pin 78. 0x40004C02 YELLO
1280                  //  volatile uint8_t      *reg = 0;
1281                   volatile uint32_t      *reg = (volatile uint32
1282                //       volatile uint32_t      *reg = (volatile ui
1283                    uint32_t         num = numBytes;
1284                  pinMode(RED_LED, OUTPUT);
1285 //                   pinMode(YELLOW_LED, OUTPUT);
1286 //jg 12/6/17 change loop counters to subs (before bne) for flag!!
1287 #if 1
1288   asm volatile(
1289     "L%=_begin:"                  "\n\t"
1290     "ldrb   %[pix], [%[p], #0]"     "\n\t"
1291     "lsl    %[pix], #24"            "\n\t"
1292     "movs   %[count], #7"           "\n\t"
1293     "L%=_loop:"               "\n\t"
1294     "lsl    %[pix], #1"             "\n\t"
1295     "bcs    L%=_loop_one"           "\n\t"
1296     "L%=_loop_zero:"
1297     "strb   %[bitmask], [%[reg], #0]"   "\n\t"
```

**Disassembly** — loop

```
0000466e:  88A6        ldrh    r6, [r4, #4]
1284                  pinMode(RED_LED, OUTPUT);
00004670:  204B        movs    r0, #0x4b
00004672:  2101        movs    r1, #1
00004674:  F000F9E0    bl      #0x4a38
1361                  );
00004678:  462B        mov     r3, r5
0000467a:  4632        mov     r2, r6
0000467c:  4824        ldr     r0, [pc, #0x90]
0000467e:  2101        movs    r1, #1
00004680:  781D        ldrb    r5, [r3]
00004682:  EA4F6505    lsl.w   r5, r5, #0x18
00004686:  2607        movs    r6, #7
00004688:  EA4F0545    lsl.w   r5, r5, #1
0000468c:  F080800C    bhs.w   #0x46a8
00004690:  7001        strb    r1, [r0]
00004692:  2704        movs    r7, #4
00004694:  3F01        subs    r7, #1
00004696:  F47FAFFD    bne.w   #0x4694
0000469a:  7101        strb    r1, [r0, #4]
0000469c:  270D        movs    r7, #0xd
0000469e:  3F01        subs    r7, #1
000046a0:  F47FAFFD    bne.w   #0x469e
000046a4:  F000B80B    b.w     #0x46be
000046a8:  7001        strb    r1, [r0]
000046aa:  270D        movs    r7, #0xd
```

**Registers**

| Name | Value | Description |
|---|---|---|
| Core Regis | | Core Registers |
| PC | 0x0000467E | Program Coun... |
| SP | 0x20002740 | General Purpos... |
| LR | 0x00005B03 | General Purpos... |
| xPSR | 0x41000000 | Stores the statu... |
| R0 | 0x40004C03 | General Purpos... |
| R1 | 0x40000201 | General Purpos... |
| R2 | 0x000000B4 | General Purpos... |
| R3 | 0x20001E88 | General Purpos... |
| R4 | 0x20001190 | General Purpos... |
| R5 | 0x20001E88 | General Purpos... |
| R6 | 0x000000B4 | General Purpos... |
| R7 | 0x007F7F7F | General Purpos... |
| R8 | 0x00000032 | General Purpos... |
| R9 | 0xFFFFFFFF | General Purpos... |
| R10 | 0xFFFFFFFF | General Purpos... |

**Console / Problems / Target Configurations**

MSP432P401R.ccxml

```
CORTEX_M4_0: Flash Programmer: Writing 3148 bytes to flash memory 0x00006a98
CORTEX_M4_0: Flash Programmer: Writing 8 bytes to flash memory 0x000076e4
```

LE

# Advanced Topics

- Energia Libraries
  - Educational BoosterPack MKII (Built-In Energia18)
  - Neopixel/WS2811 Driver (Copy library into Energia18)
  - Seeed Grove Starter Kit (Add from GitHub)
  - TI BLE for CC2650 Boosterpack  (Add from GitHub)
- Layered Energia Code
- Porting AdaFruit Neopixel to MSP432
- Single-step Debug of Energia using CCS (even assembly language)
- Allows Energia<->CCS fluid transitions
- USB (Native)  UART (MSP430F5529 Launchpad)  Energia USBSerialExample
- Multiblink(Use of TI-RTOS) Energia Multitasking (Red+Green+Blue = White LED)

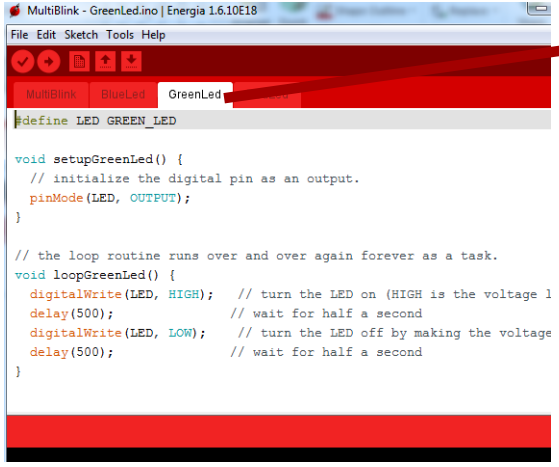
# NativeUSB UART, Energia Multitasking

- With MSP430F5529 LaunchPad (MSP-EXP430F5529LP) and USB Devpack
  - serialevent (UART Echo) – default 9600 baud
  - USBSerialExample (USB UART Echo) – Need USB_config files
  - USBSerialExample2 (UART Echo <-> USB UART i.e. CDC Client Echo)

Need Putty or other Terminal program

MSP430F5529 LP

- Energia Multitasking (just TI RTOS) – MultiBlink with EnergiaMT (TI-RTOS): https://www.youtube.com/watch?v=0f8brpzvdD0




One tab for each LED (Red+Green+Blue = White LED)

MSP432P401r LP

# Agenda

- Fundamentals (mostly for Analog)
  - Implementing  necessary prototyping functions such clocks/GPIO, Read A/D, I2C, etc.
  - Seamless interface of various Analog EVM's for customer "proof of concept"
  - Standalone UI - Button (GP Input - GPIO), LCD Display ("Hello"), Music, Serial Interface (Putty)

- More UI (i.e. GUI Advanced Comm Tab basically Putty/Serial I/F) - Lessons learned from home networking (if you can setup the WiFi in your house, you can prototype with a few steps)

- EP - Embedded prototyping (mostly for Digital)
  - Wired  and Wireless Control
  - Use of TI Cloud Computing Tools for prototype

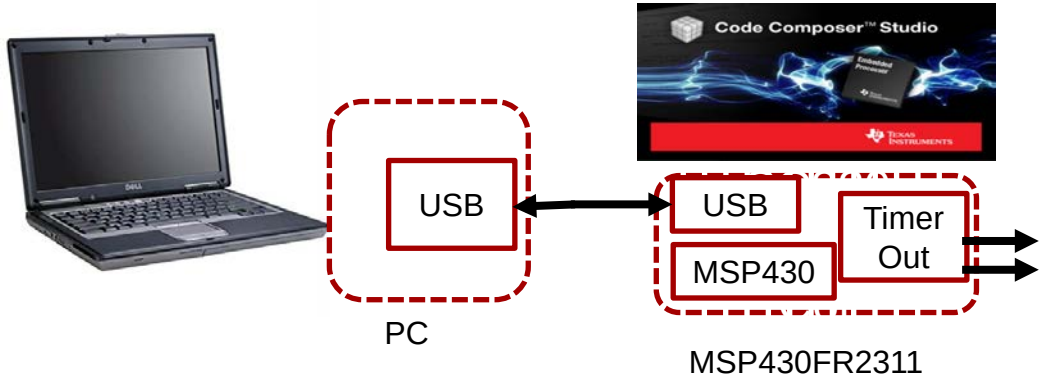
- Advanced Topics
- Conclusion Demos (Simple and Complex)

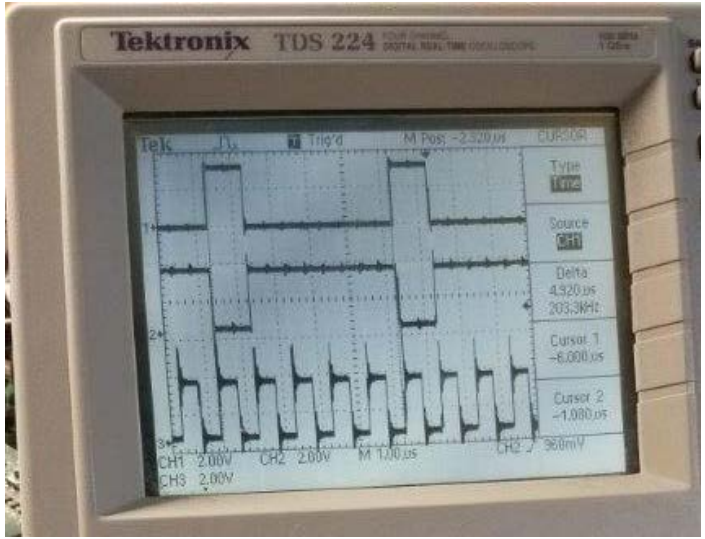
# Conclusion

- Conclusion / Call to Action
  - Keep prototyping with the ideas mentioned here using various tools (i.e TI Cloud Computing)
    - Especially analog EVMs connected to Launchpads (i.e. I2C/SMBus)
    - Influence TI to make rapid prototyping easier (command line interface over GUI?)
  - Have Fun!

- Conclusion Demos
  - (Simple) Drive UCC28070 PFC Controller MSP430 (Revisit Square Wave)
  - (Complex) Neopixel controlled wirelessly from iPad (Fun)

# Simple Demo - Block Diagram



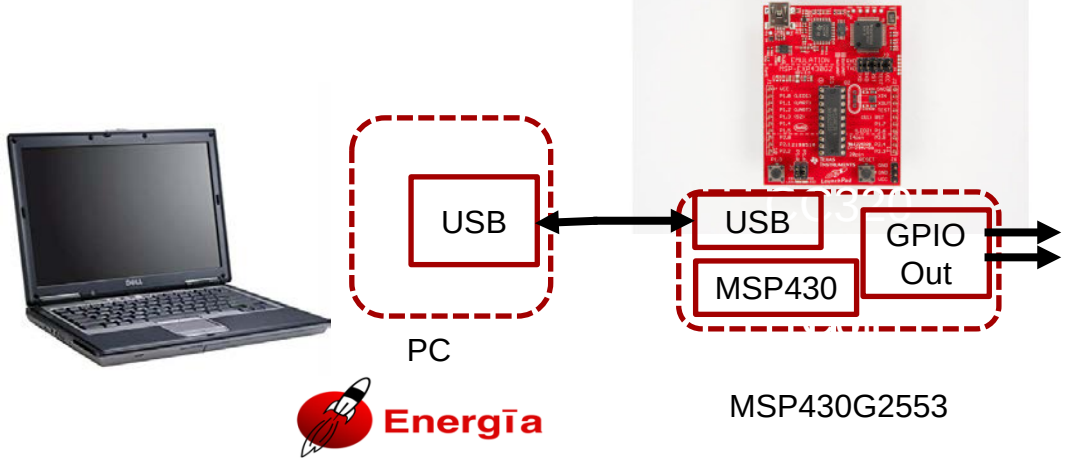

Square Waves (200 kHz) with Reference Clock

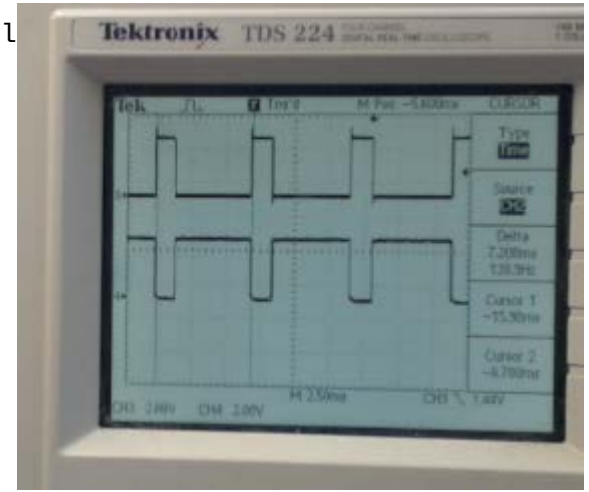
- Revisit Square Wave
- (Simple) Drive UCC28070 PFC Controller with MSP430 Programmable Clock Source + Timer PWM (from MSP 25f25c 25 Functions for $0.25)

# Simple Demo – GPIO – Square waves for PFC

- Step-by-step Functionality (Demo) - PFC (Power Factor Correction) Controller
  - GP Output (“Blinky” is just General Purpose Input) – Energia Blink (square wave)
    - P1.0 ->RED_LED, P1.6 ->GREEN_LED

```
digitalWrite(RED_LED, HIGH);   // turn the LED on (HIGH is the voltage level)
digitalWrite(GREEN_LED, LOW);   // turn the LED off (LOW is the voltage level)
  delay(5);               // wait for 5 milliseconds (long pulse)
digitalWrite(RED_LED, LOW);   // turn the LED on (HIGH is the voltage level)
digitalWrite(GREEN_LED, HIGH);   // turn the LED on (HIGH is the voltage level
  delay(1);               // wait for 1 millisecond (short pulse)
```

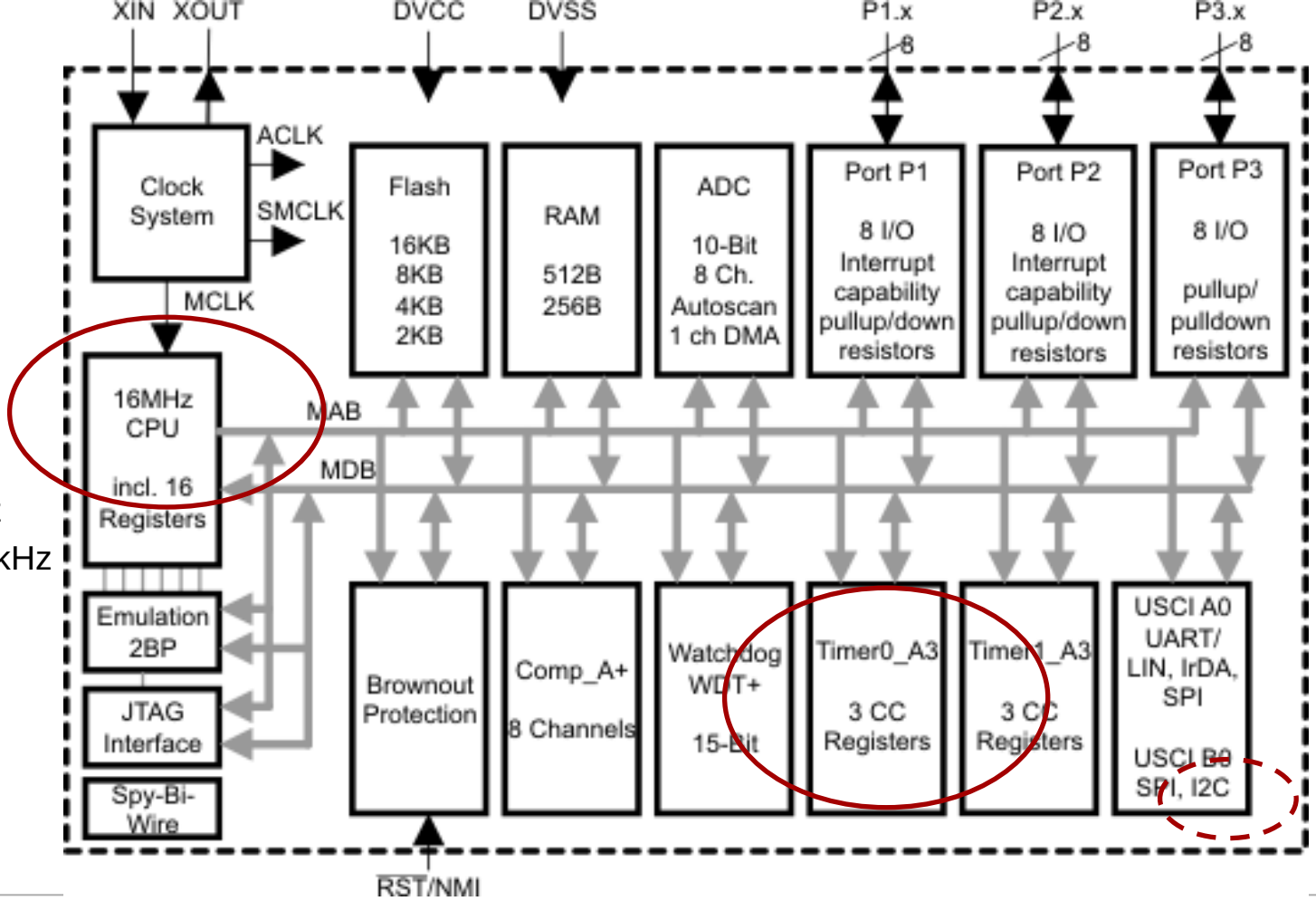



Square Waves with Energia (130 Hz)

# MSP430 G2553 Device

- Clock Speed
  - (I2C 83 kHz)
  - CPU ~130 Hz
  - Timer0 ~100 kHz

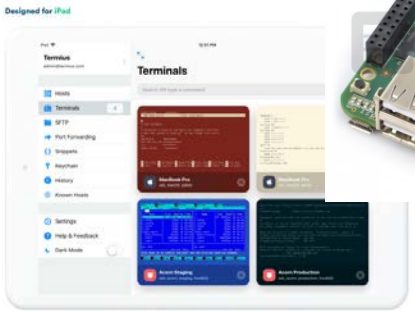


# Complex Demo - Block Diagram

## (Embedded prototype demo - Wireless Control of LCD and Neopixel)

**Add Neopixel controlled wirelessly from iPad**

Smartphone/iPAD
(WiFi **STA**)




Termius App SSH

USB Power

USB0

Wilink8

Cortex-A8

USB1 HUB

UART0

eMMC/ SD Card

BB Green Wireless AM3358
(WiFi **STA**)

Neopixel

GPIO

USB

Cortex-M4

MSP432P SimpleLink

USB

RF

Cortex-M4

CC3220 SimpleLink SDK Wi-Fi
(WiFi **AP**)


Energia

USB

MSP430

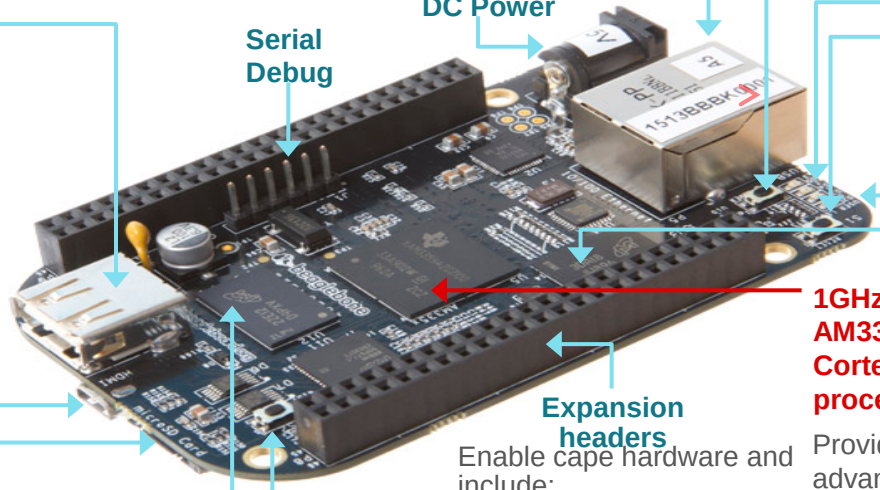
LCD CTLR

MSP430FR4133 LCD

# Bonus Topics

- Beaglebone Black
  - HDMI for Gnome “Desktop” UI (Debian)
  - Remote Desktop i.e. GUI over network (Arago)
- Internet Access
  - Internet USB to Ethernet dongles (Debian)
  - Wilink8 Multirole as Gateway
- Making of Cecil, Stages 1 and 2 picture

# BeagleBone Black
# 1GHz performance ready to use for $45



**10/100 Ethernet**

**USB Host**
Easily connects to almost any everyday device such as mouse or keyboard

**microHDMI**
Connect directly to monitors and TVs

**microSD**
Expansion slot for additional storage

**512MB DDR3**
Faster, lower power RAM for enhanced user-friendly experience

**Serial Debug**

**DC Power**

**Boot Button**

**Expansion headers**
Enable cape hardware and include:
- 65 digital I/O
- 7 analog
- 4 serial
- 2 SPI
- 2 I2C
- 8 PWMs
- 4 timers

**1GHz Sitara AM335x ARM® Cortex™-A8 processor**
Provides a more advanced user interface and up to 150% better performance than ARM11

**Power Button**

**LEDS**

**Reset Button**

**USB Client**
Development interface and directly powers board from PC

**2GB on-board storage using eMMC**
- Pre-loaded with Ångström Linux Distribution
- Accelerates performance
- Frees the microSD slot to be used for additional storage for a less expensive solution than SD cards

Included in price:
- Power supply ~ $10
- USB network cable ~ $3
- 2GB on-board storage $5-$10
- PRU for real-time tasks typically on FPGA ~ $20

# (Remote) Desktop Options

Beaglebone Black – Angstrom GUI on HDMI

USB Power

5V Power (optional)

AM335x

eMMC

HDMI

USB

10/100 PHY

BB-Black

Display


Beaglebone Black – Arago Console (Matrix GUI over network)


USB Power

5V Power (optional)

USB-> UART Cable

AM335x

SD Card

HDMI

USB

10/100 PHY

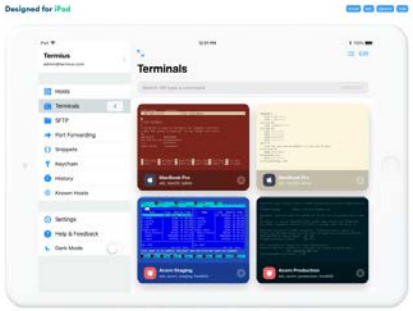
BB-Black

USB WiFi Adapter (D-Link DWA-125)

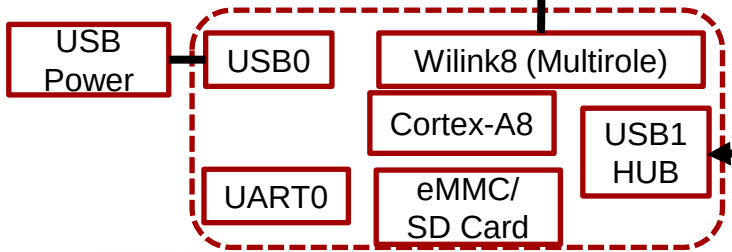
# Embedded prototype – Internet Access

Router/Hotspot

Smartphone/iPAD
(WiFi **STA**)




Termius App SSH

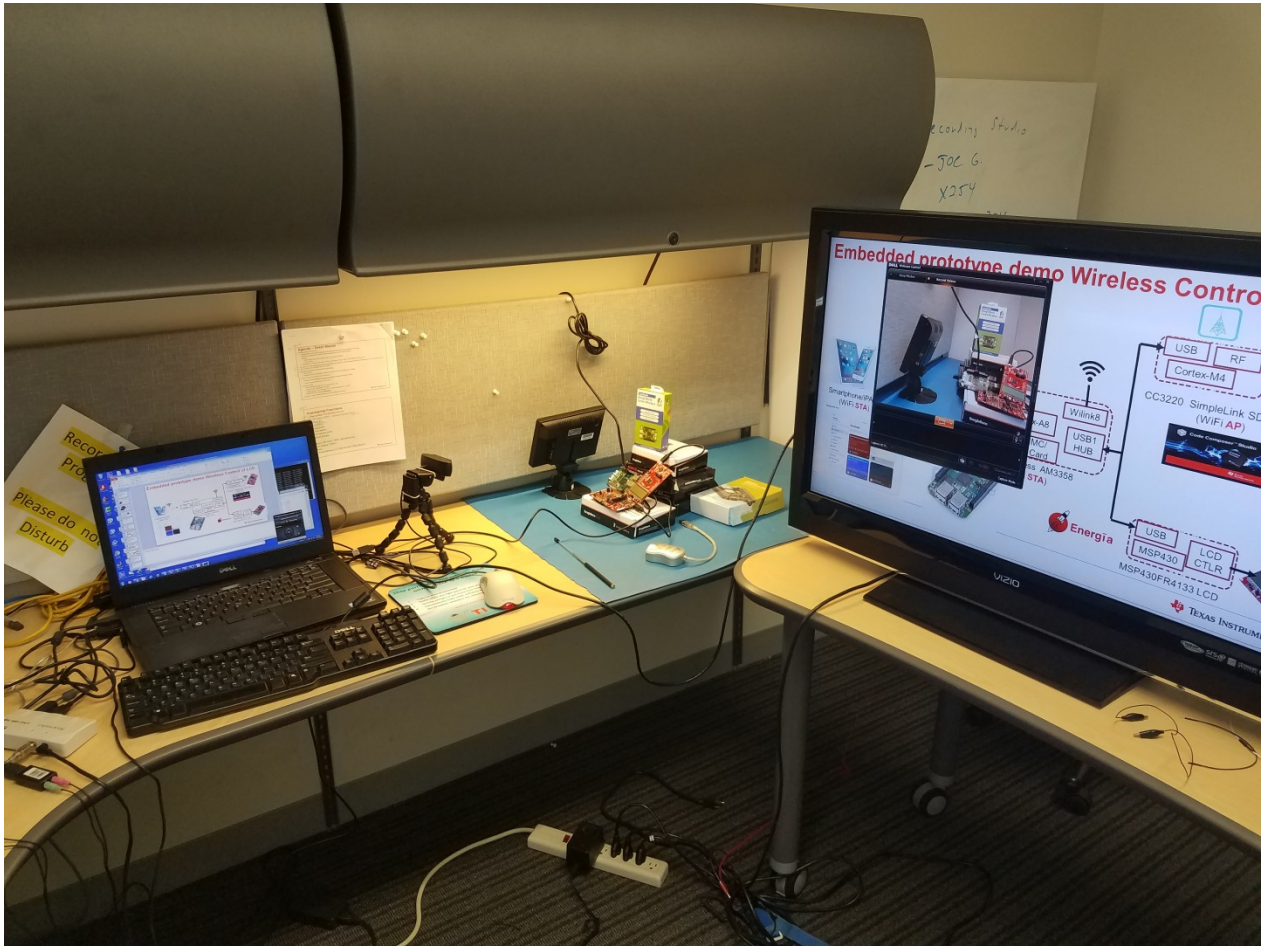



BB Green Wireless AM3358
(WiFi **STA/AP**
MultiRole)

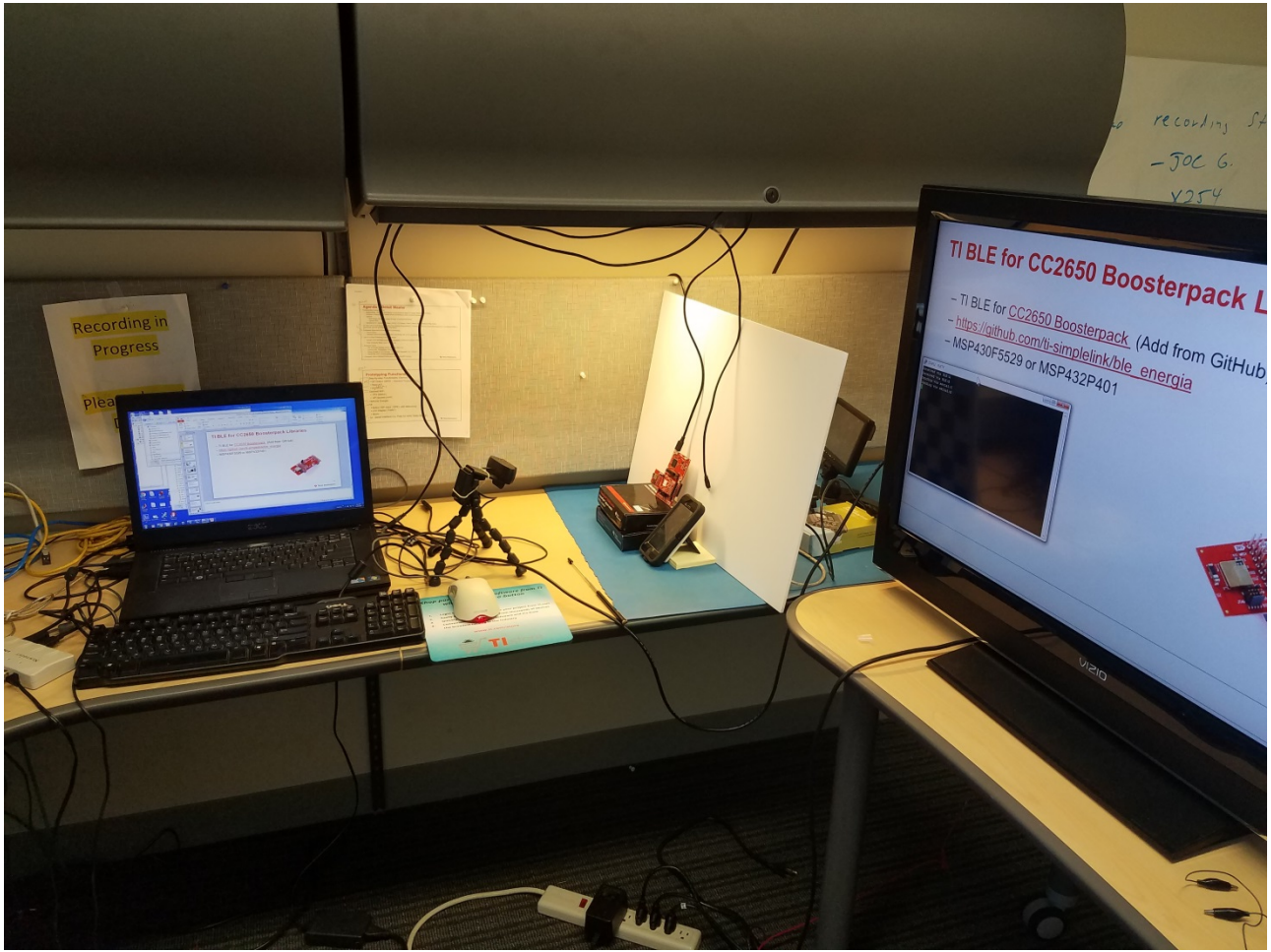
No Simplelink
WiFi?


USB to Ethernet Dongle

# Stage 1 Setup



# Stage 2 Setup



# Backup