

# Pushing Your Arduino Project to the Next Level with Texas Instruments

Discovering the Energia Ecosystem



**ESIEE**  
PARIS



*May 17, 2018*

# Agenda

- **Part 1 – Micro-Controller Boards**

- Arduino and Energia
- Example: Install Energia

- **Part 2 – LaunchPad and Energia**

- The LaunchPad Family
- Discover the MSP432 LaunchPad
- Example: Blink a LED
- Example: Port the Code to Another LaunchPad

- **Part 3 – BoosterPack and Libraries**

- The BoosterPack Family
- Example: Install the Educational BoosterPack
- Example: Display Rainbow Colours
- Example: Read Temperature

- **Conclusion – The Energia Ecosystem**

© Rei Vilo – Presentation at ESIEE – May 17, 2018

*All product names, logos, trademarks and brands cited herein are the property of their respective owners.*

*All company, product and service names used in this presentation are for identification purposes only.*

*Use of these names, logos, and brands does not imply endorsement.*

# Who Am I?

- Screen name: Rei Vilo
- Consultant
- Founding member of the Energia project
- Contribute with tools and libraries
- Focus on industrial robots, smart sensors, displays, and IoT
- Website: Embedded Computing  
[embeddedcomputing.weebly.com](http://embeddedcomputing.weebly.com)

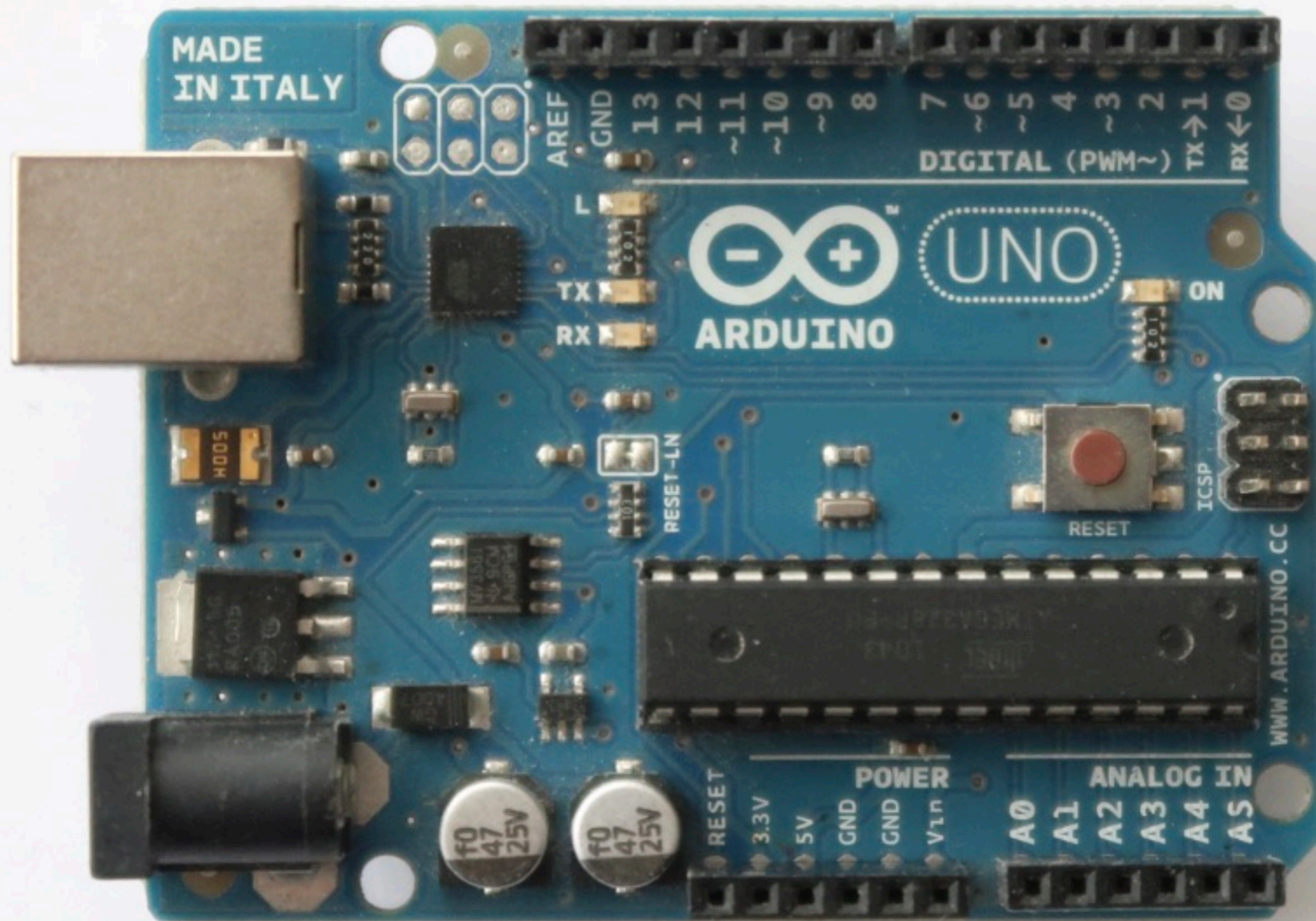


# Agenda

- **Part 1 — Micro-Controller Boards**
  - What Is Arduino?
  - LaunchPad *vs.* Arduino
  - Official and Third-Party IDEs
  - Install Energia

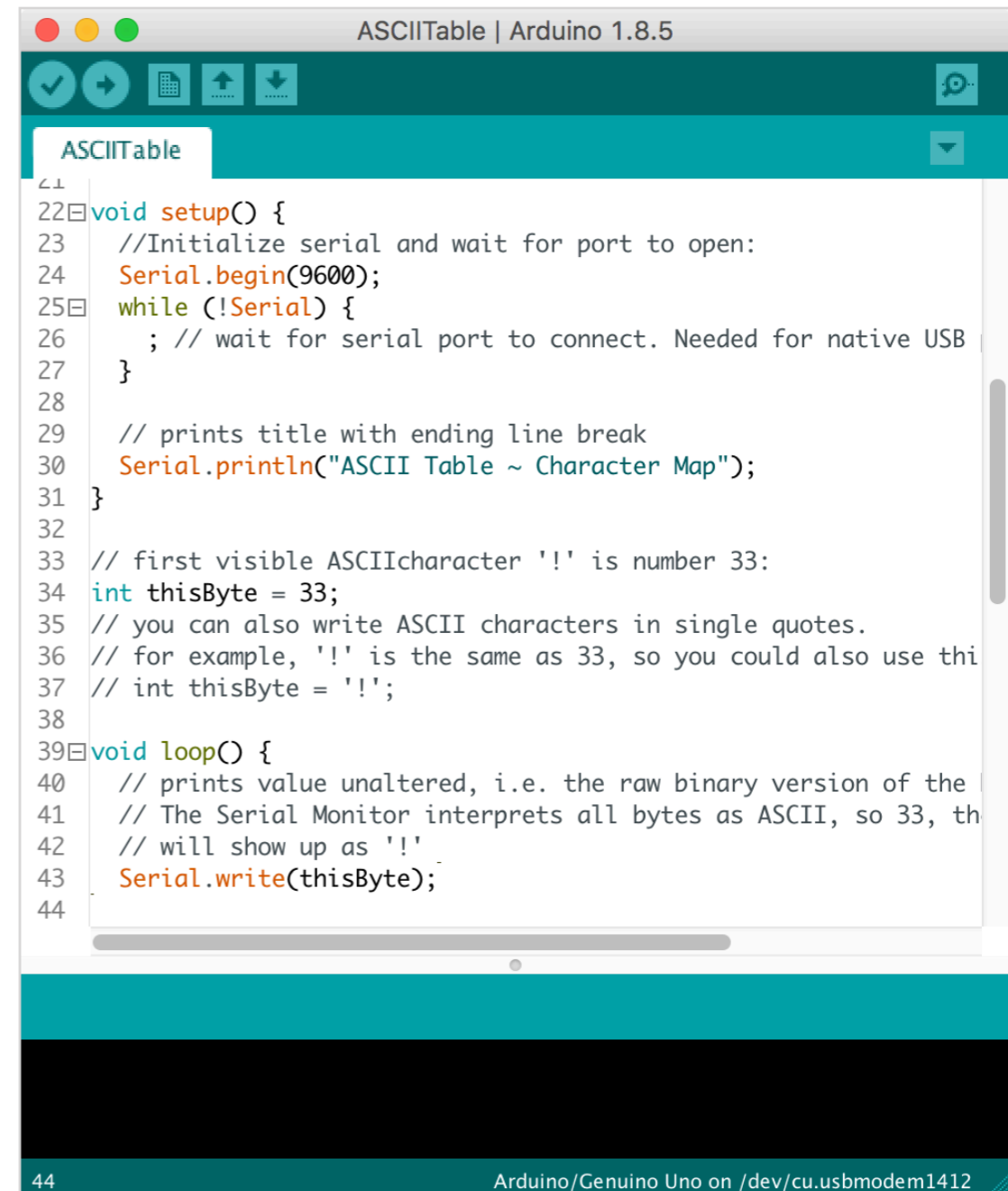


# What Is Arduino?



# What Is Arduino?

- Framework
- Open-source
- Based on C / C++
- Relies on objects
- Derived from Hernando Barragán's master thesis [Wiring - Prototyping Physical Interaction Design \(2003\)](#), with Massimo Banzi as supervisor

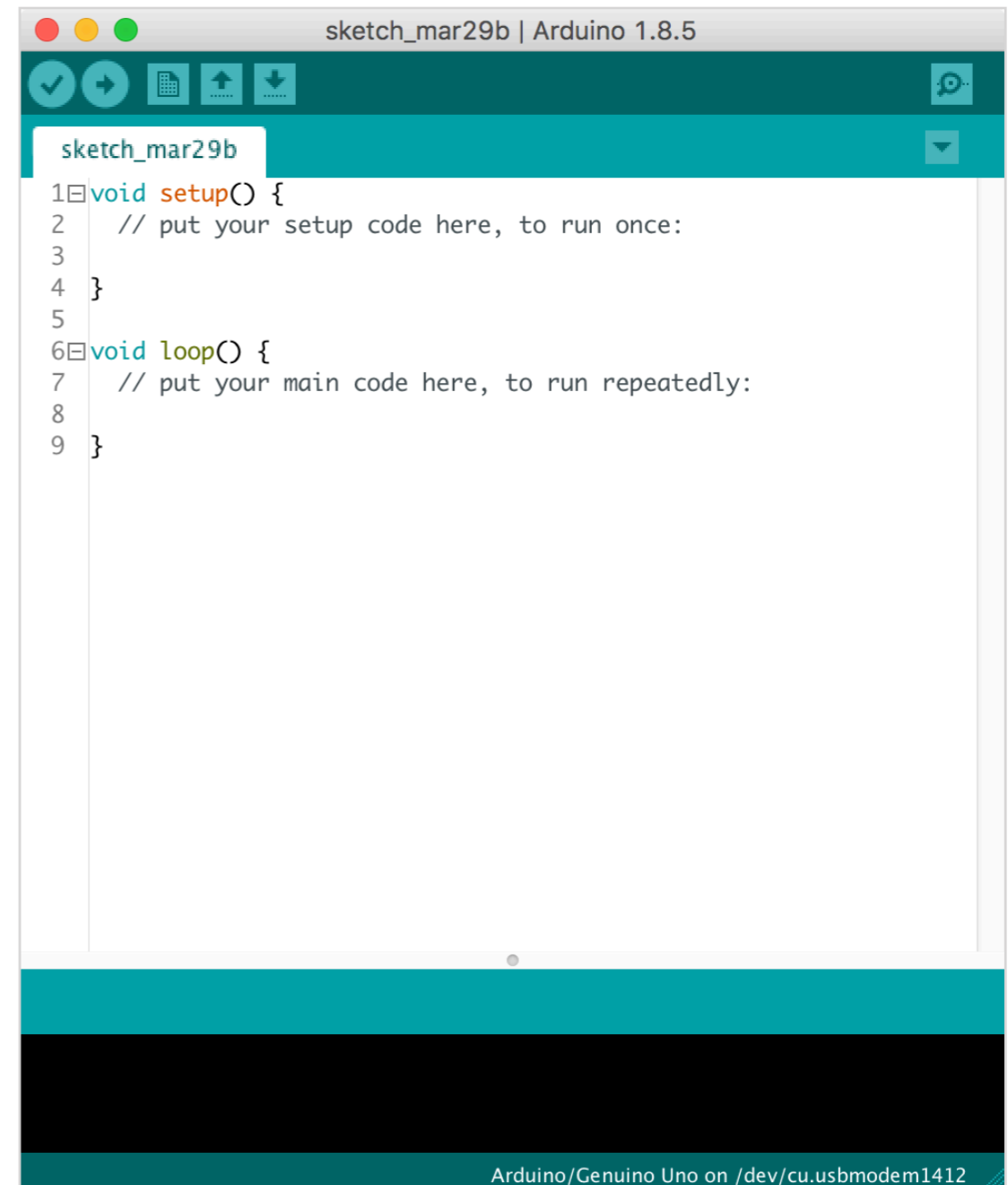


```
ASCIITable | Arduino 1.8.5
ASCIITable
22 void setup() {
23   //Initialize serial and wait for port to open:
24   Serial.begin(9600);
25   while (!Serial) {
26     ; // wait for serial port to connect. Needed for native USB
27   }
28
29   // prints title with ending line break
30   Serial.println("ASCII Table ~ Character Map");
31 }
32
33 // first visible ASCII character '!' is number 33:
34 int thisByte = 33;
35 // you can also write ASCII characters in single quotes.
36 // for example, '!' is the same as 33, so you could also use thi
37 // int thisByte = '!';
38
39 void loop() {
40   // prints value unaltered, i.e. the raw binary version of the
41   // The Serial Monitor interprets all bytes as ASCII, so 33, th
42   // will show up as '!'
43   Serial.write(thisByte);
44 }
```

44 Arduino/Genuino Uno on /dev/cu.usbmodem1412

# What Is Arduino?

- IDE or Integrated Development Environment
- Based on Processing
- Minimalistic design
- Edit, build, upload
- Serial console, no debugger
- GCC tool-chain



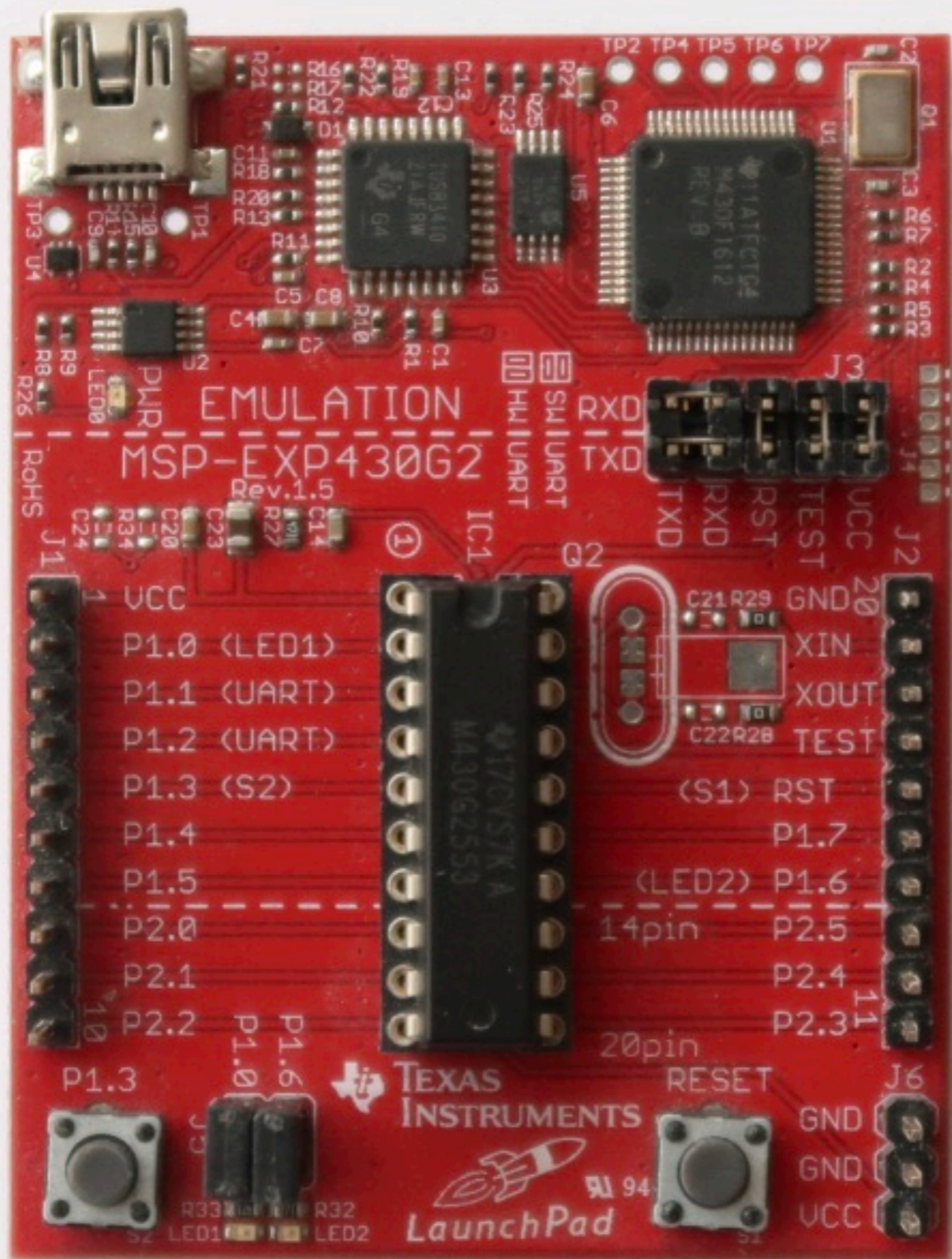
```
sketch_mar29b | Arduino 1.8.5
sketch_mar29b
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Arduino/Genuino Uno on /dev/cu.usbmodem1412

# What Is Arduino?

- Hardware
- Software
- Tools
- Community
- Generic name
- Arduino Uno
- Arduino framework
- Arduino IDE
- Forum and contributed libraries
- Open-source hardware and software

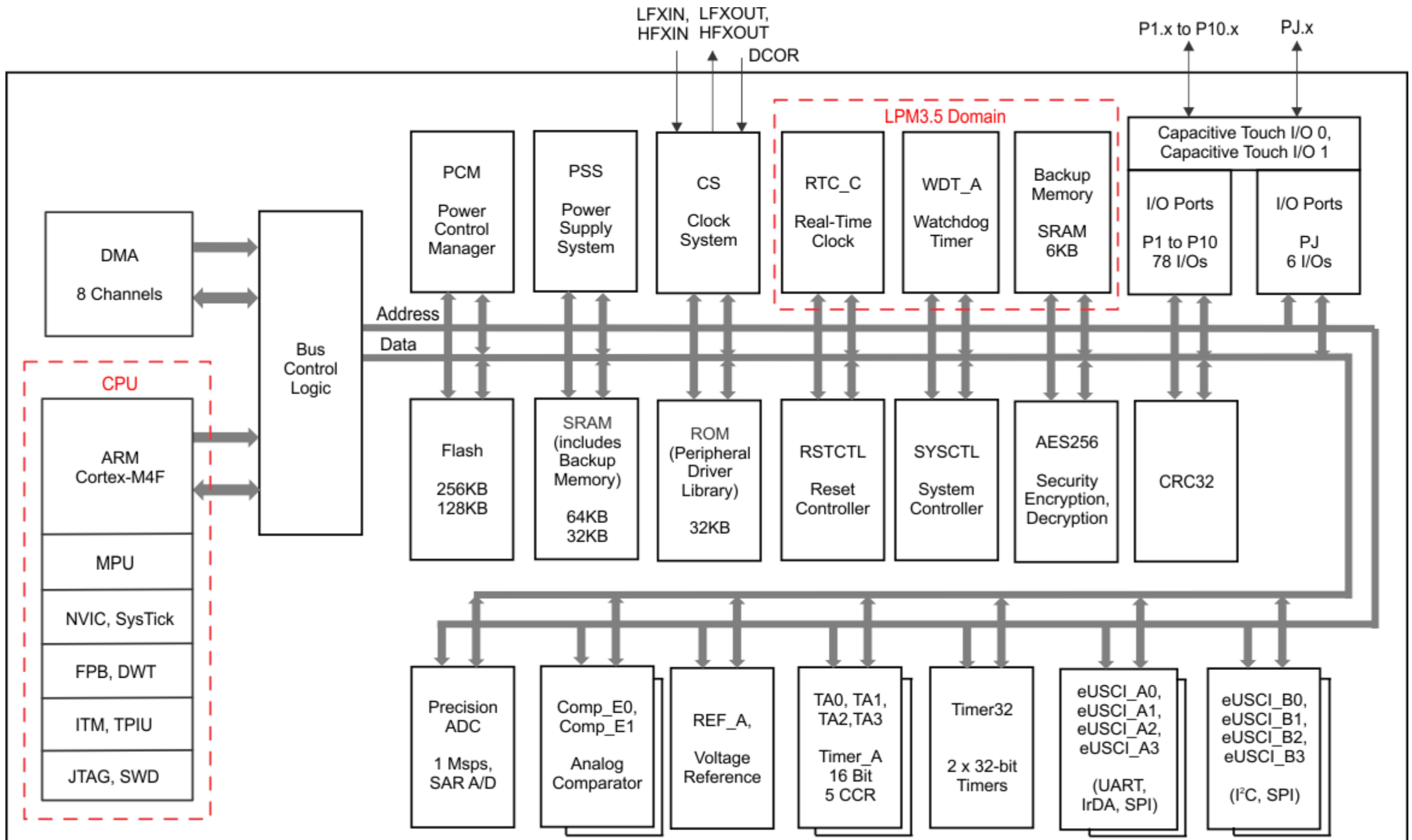




# MCU *vs.* MPU

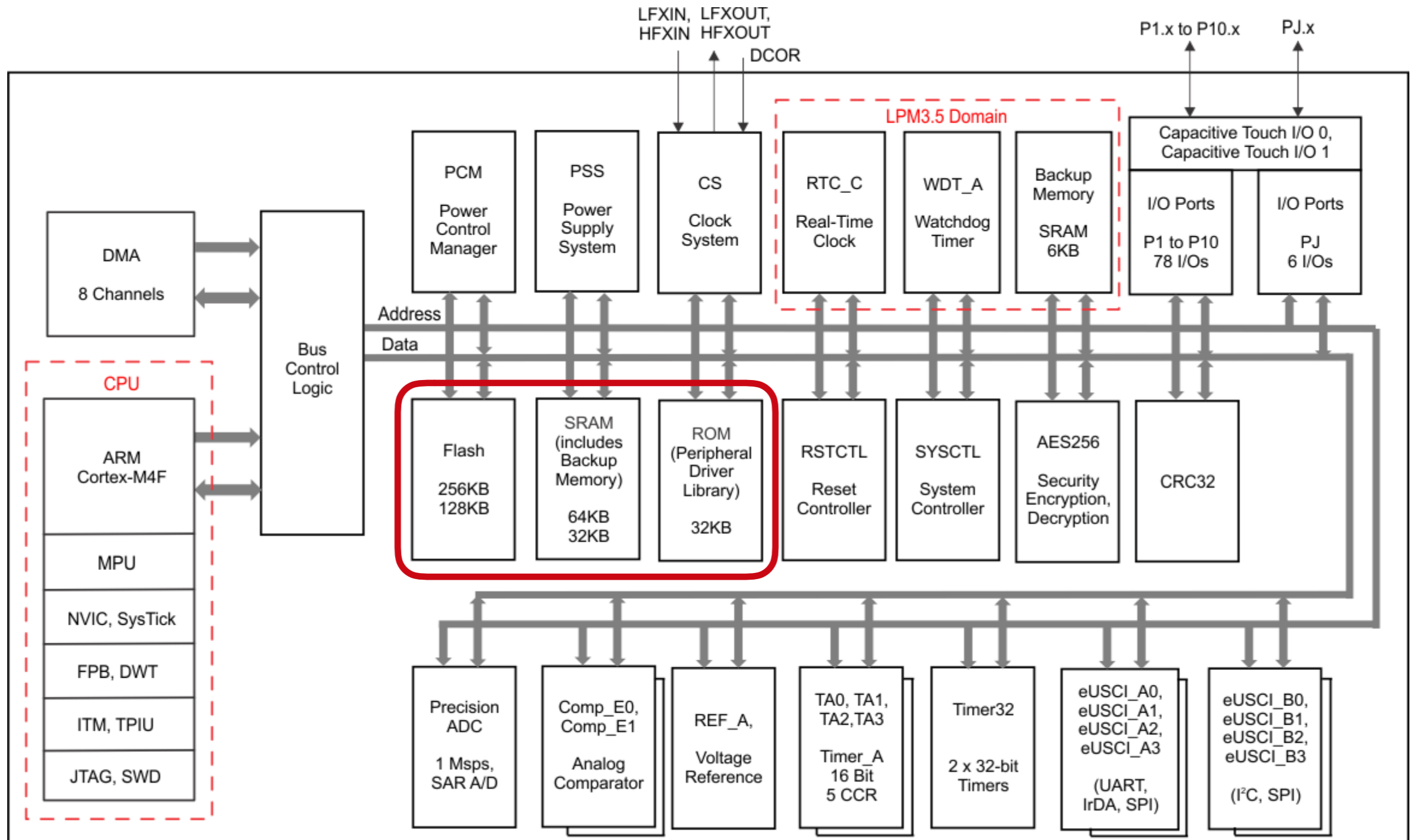
- MCU *aka.* Micro-Controller Unit
- MPU *aka.* Micro-Processor Unit

# Inside the MSP432





# Inside the MSP432





# Perspective

		IBM PC	LaunchPad MSP430G2	LaunchPad MSP432
<b>Year</b>		1981	2010	2015
<b>Processor</b>		Intel 8088	MSP430G2553	MSP432P401R
<b>Architecture</b>	<b>bits</b>	8	16	32
<b>Speed</b>	<b>MHz</b>	4,77	16	48
<b>RAM</b>	<b>kB</b>	16	0,5	64
<b>ROM</b>	<b>kB</b>	40	0	32
<b>Mass Storage</b>	<b>kB</b>	Dual 160~360 5¼" floppy disks	16	256
<b>Price</b>	<b>USD</b>	1 600 ~ 3 000	4,30	12,99

# Energia

- Framework and IDE specific for Texas Instruments boards
- Initiated in January 2012 by Robert Wessels



- Forked from Arduino
- Open-source software

A screenshot of the Energia IDE interface. The window title is "sketch\_mar29a | Energia 1.8.6E19". The code editor shows the following code:

```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }
```

The IDE has a dark red theme. At the bottom, a status bar displays "LaunchPad (Stellaris) w/ lm4f120 (80MHz) on /dev/cu.usbmodemM4321001".

# LaunchPad vs. Arduino

## Shared Features

- IDE based on Processing
- Framework based on Wiring
- Underlying C / C++
- Tool-chain GCC
- Open-source hardware and software

# LaunchPad vs. Arduino

## Shared Features

- IDE based on Processing
- Framework based on Wiring
- Underlying C / C++
- Tool-chain GCC
- Open-source hardware and software

## LaunchPad Features

- Boards supported by Texas Instruments
- All boards with hardware debugger
- Energia libraries based on TI-DriverLib
- RTOS extension with Energia Multi-Tasking based on TI-RTOS
- Three official IDEs

# Official IDEs



- **Energia IDE**  
*Cross-platform*  
[energia.nu](http://energia.nu)



- **Code Composer Studio**  
*Based on Eclipse*  
*Cross-platform*  
[ti.com/ccs](http://ti.com/ccs)



- **CCS Cloud**  
*Chrome add-on*  
[dev.ti.com](http://dev.ti.com)

# Official IDEs



- **Energia IDE**  
*Cross-platform*  
[energia.nu](http://energia.nu)



- **Code Composer Studio**  
*Based on Eclipse*  
*Cross-platform*  
[ti.com/ccs](http://ti.com/ccs)



- **CCS Cloud**  
*Chrome add-on*  
[dev.ti.com](http://dev.ti.com)



# Third-Party IDEs



- **Visual Micro**  
*for Visual Studio*  
*Windows only*  
[visualmicro.com](http://visualmicro.com)



- **embedXcode**  
*for Xcode*  
*macOS only*  
[embedXcode.com](http://embedXcode.com)

# Third-Party IDEs



- **Visual Micro**  
*for Visual Studio*  
*Windows only*  
[visualmicro.com](http://visualmicro.com)



- **embedXcode**  
*for Xcode*  
*macOS only*  
[embedXcode.com](http://embedXcode.com)

# Which Code Do You Prefer?

- Same application
  - Initialise an analog input
  - Read the analog input
- Same MCU
  - Cortex-M4F  
TM4C123GH6PM
- Three options
  - Registers
  - ROM-ware
  - Energia

# Which Code Do You Prefer?

```
95 // Define variables and constants
96 uint16_t result = 0;
97
98 // Add setup code
99 void setup()
100 {
101     // Declare analog input PE2/A1
102     SYSCTL_RCGC2_R |= 0x00000010;
103     GPIO_PORTE_DIR_R &= ~0b00000100;
104     GPIO_PORTE_AFSEL_R |= 0b00000100;
105     GPIO_PORTE_DEN_R &= ~ 0b00000100;
106     GPIO_PORTE_AMSEL_R |= 0b00000100;
107     SYSCTL_RCGC0_R |= 0x00010000;
108     SYSCTL_RCGC0_R &= ~0x00000300;
109     ADC0_SSPRI_R = 0x0123;
110     ADC0_ACTSS_R &= ~0x0008;
111     ADC0_EMUX_R &= ~0xf000;
112     ADC0_SSMUX3_R &= ~0x000f;
113     ADC0_SSMUX3_R += 1;
114     ADC0_SSCTL3_R = 0x0006;
115     ADC0_ACTSS_R |= 0x0008;
116 }
117
118 // Add loop code
119 void loop()
120 {
121     // Read value form PE2/A1
122     ADC0_PSSI_R = 0x0008;
123     while ((ADC0_RIS_R & 0x08) == 0);
124     result = ADC0_SSFIF03_R & 0xff;
125     ADC0_ISC_R = 0x0008;
126 }
```

## Option 1: Registers

# Which Code Do You Prefer?

```
95 // Define variables and constants
96 uint16_t result = 0;
97
98 // Add setup code
99 void setup()
100 {
101     uint8_t bit = digitalPinToBitMask(28); // PE2 = 28
102     uint8_t port = digitalPinToPort(28);
103     uint32_t portBase = (uint32_t) portBASERegister(port);
104
105     ROM_GPIOPinTypeGPIOInput(portBase, bit);
106 }
107
108 // Add loop code
109 void loop()
110 {
111     // Read value form PE2/A1 = 28
112     uint8_t port = digitalPinToPort(28);
113     uint16_t value[1];
114     uint32_t channel = digitalPinToADCIn(28);
115     ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0);
116     if (channel != ADC_CTL_TS)
117     {
118         ROM_GPIOPinTypeADC((uint32_t) portBASERegister(port), digitalPinToBitMask(pin));
119     }
120     ROM_ADCSequenceConfigure(ADC0_BASE, 3, ADC_TRIGGER_PROCESSOR, 0);
121     ROM_ADCSequenceStepConfigure(ADC0_BASE, 3, 0, channel | ADC_CTL_IE | ADC_CTL_END);
122     ROM_ADCSequenceEnable(ADC0_BASE, 3);
123
124     ROM_ADCIntClear(ADC0_BASE, 3);
125     ROM_ADCProcessorTrigger(ADC0_BASE, 3);
126     while (!ROM_ADCIntStatus(ADC0_BASE, 3, false));
127     ROM_ADCIntClear(ADC0_BASE, 3);
128     ROM_ADCSequenceDataGet(ADC0_BASE, 3, (unsigned long*) value);
129
130     result = mapResolution(value[0], 12, 12);
131 }
```

## Option 2: ROM-ware

# Which Code Do You Prefer?

```
95 // Define variables and constants
96 uint16_t result = 0;
97
98 // Add setup code
99 void setup()
100 {
101     pinMode(28, INPUT); // PE2 = 28
102 }
103
104 // Add loop code
105 void loop()
106 {
107     // Read value form PE2/A1 = 28
108     result = analogRead(28);
109 }
110
```

## Option 3: Energia

# Why Using Energia?

- Higher level of abstraction
  - MCU-independent code
  - Portability across all supported LaunchPads
- Non-exclusive
  - Access to underlying layers
- Robust
  - Industrial-grade libraries and tools
- Rapid prototyping

# Install Energia

- Go to [energia.nu](http://energia.nu)
- Select your operating system
- Download and install
- If needed, install additional drivers required by your operating system





# Tips and Tricks



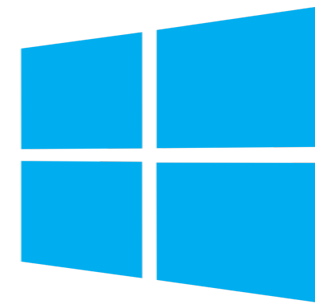
## Linux

- Check files permissions
- [energia.nu/guide/guide\\_linux](http://energia.nu/guide/guide_linux)



## macOS


- Nothing!
- [energia.nu/guide/guide\\_macosx](http://energia.nu/guide/guide_macosx)

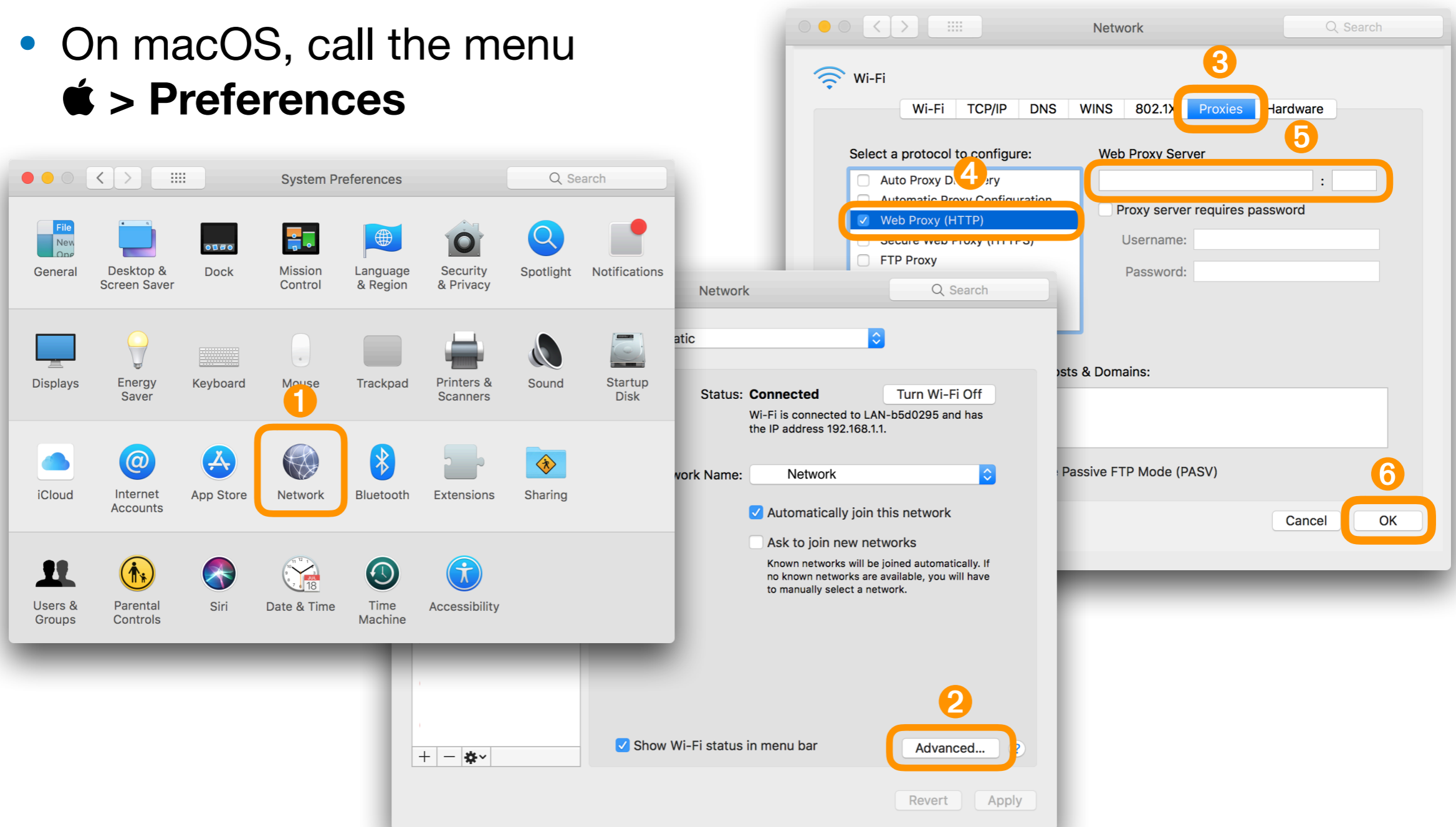


## Windows

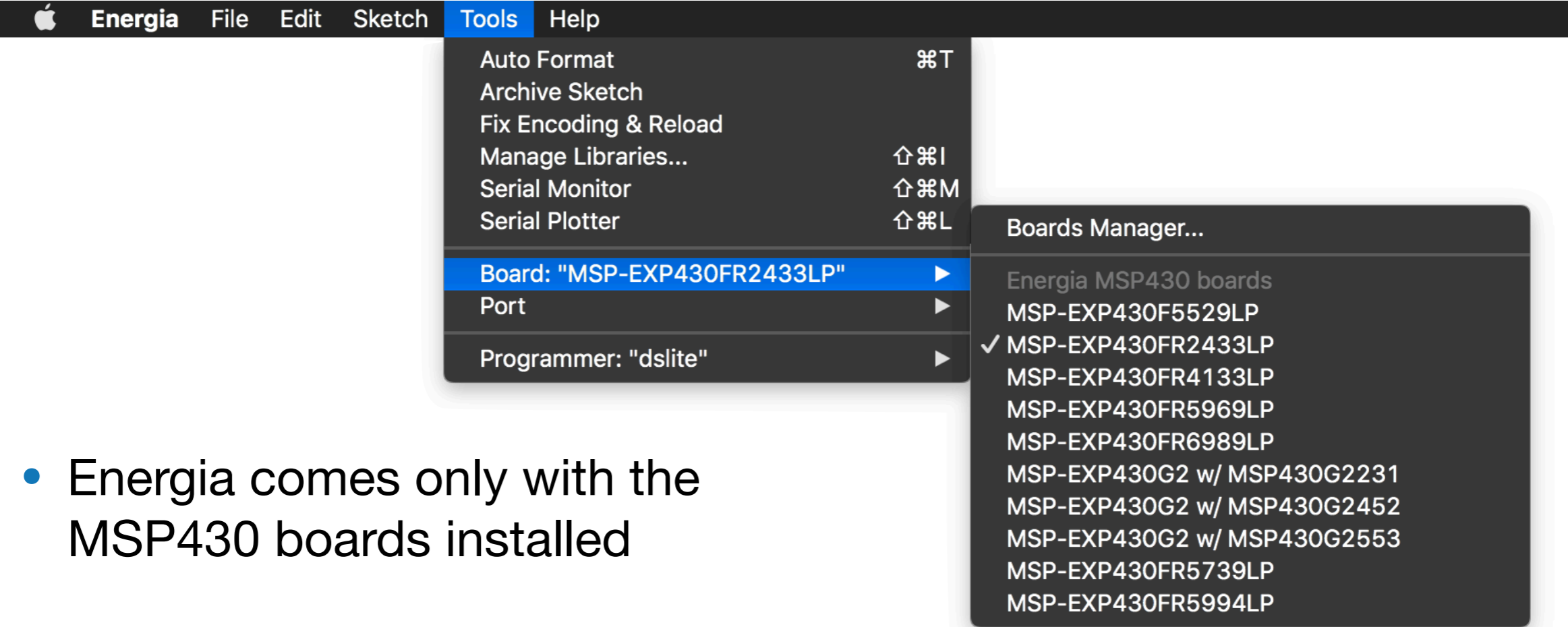
- No space in path, use a root folder
- [energia.nu/guide/guide\\_windows](http://energia.nu/guide/guide_windows)

# Proxy Configuration

- On macOS, call the menu  > **Preferences**



# Select a Board

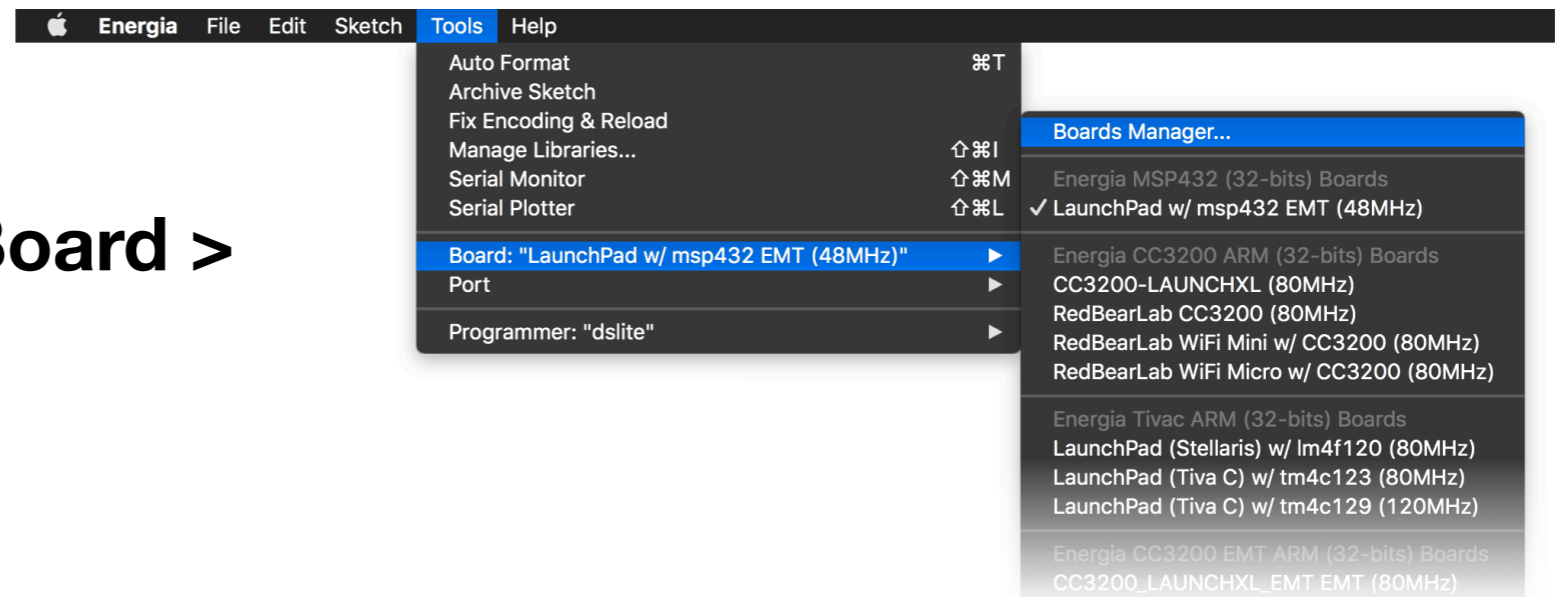


The screenshot shows the Energia IDE interface. The 'Tools' menu is open, and the 'Board' option is selected, which has opened the 'Boards Manager...' dialog box. The 'Boards Manager...' dialog box displays a list of installed boards under the heading 'Energia MSP430 boards'. The board 'MSP-EXP430FR2433LP' is checked, indicating it is the selected board.

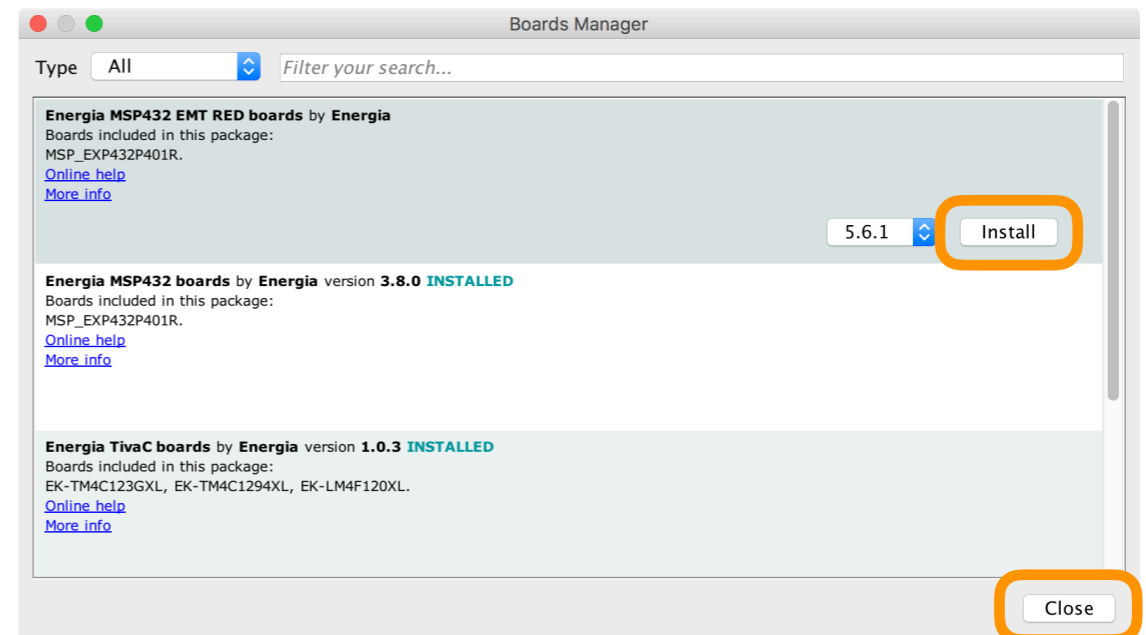
- Energia comes only with the MSP430 boards installed

# Install Other Boards

- Call menu **Tools > Board > Boards Manager**



- Select MSP432 EMT Red
- Click **Install**
- Check [energia.nu](http://energia.nu) and install additional drivers for your operating system



# Install Specific Drivers



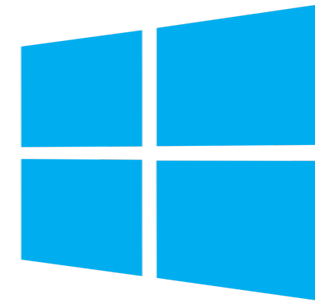
## Linux

- No drivers, but udev rules
- [energia.nu/guide/guide\\_linux](http://energia.nu/guide/guide_linux)



## macOS

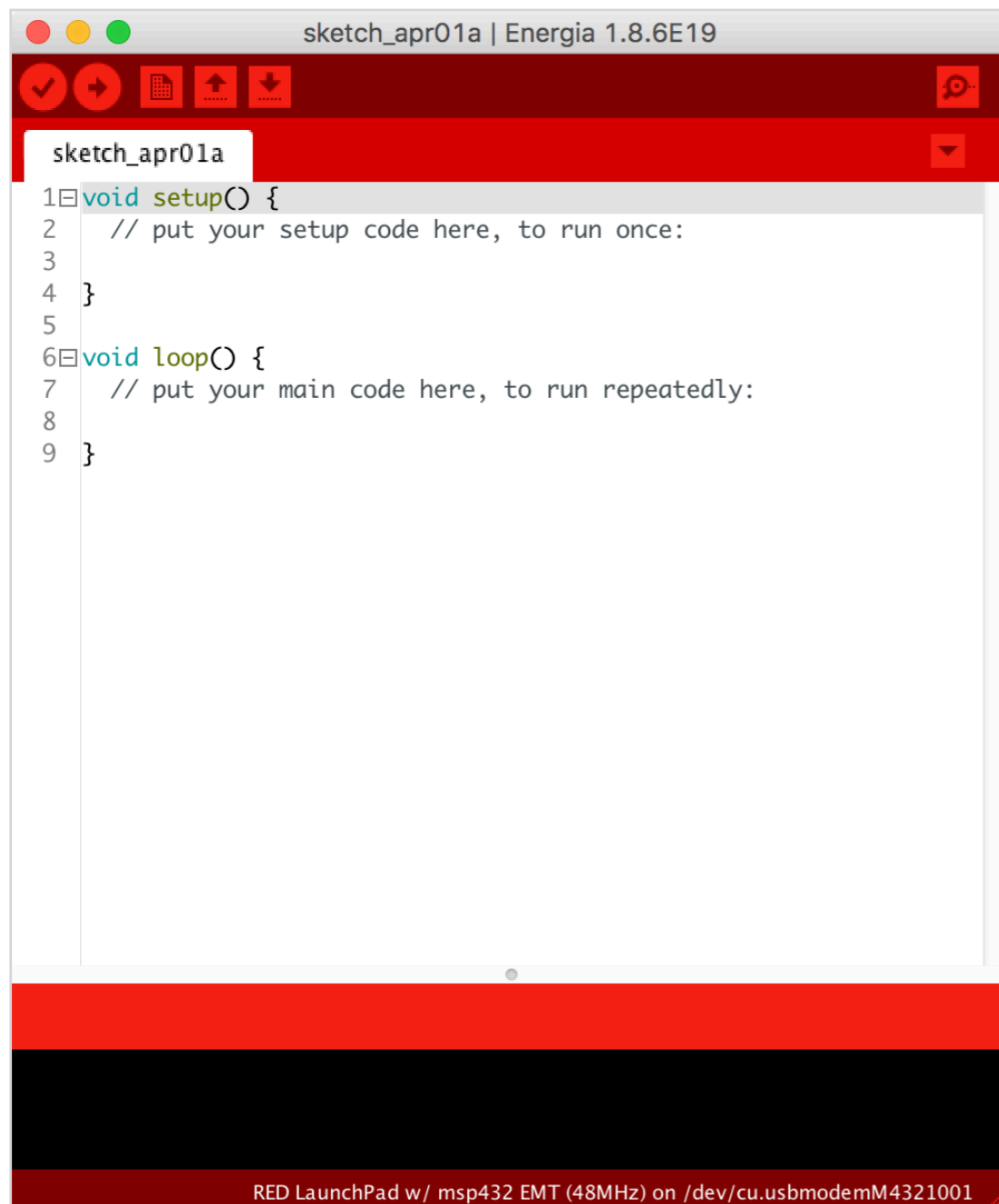
- No drivers are needed
- [energia.nu/guide/guide\\_macosx](http://energia.nu/guide/guide_macosx)



## Windows

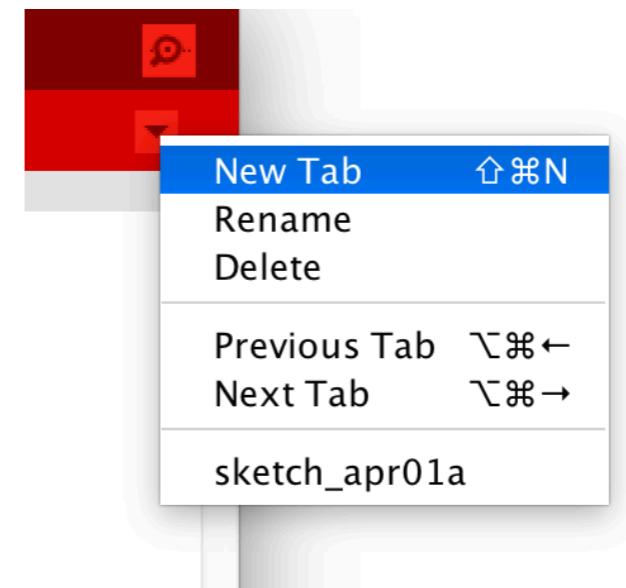
- Install XDS110 and CDC drivers
- [energia.nu/guide/guide\\_windows](http://energia.nu/guide/guide_windows)

# Energia Interface

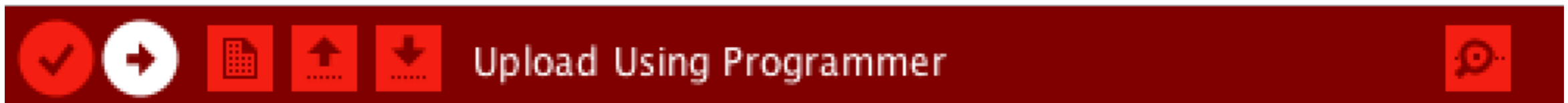


```
sketch_apr01a | Energia 1.8.6E19  
sketch_apr01a  
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }
```

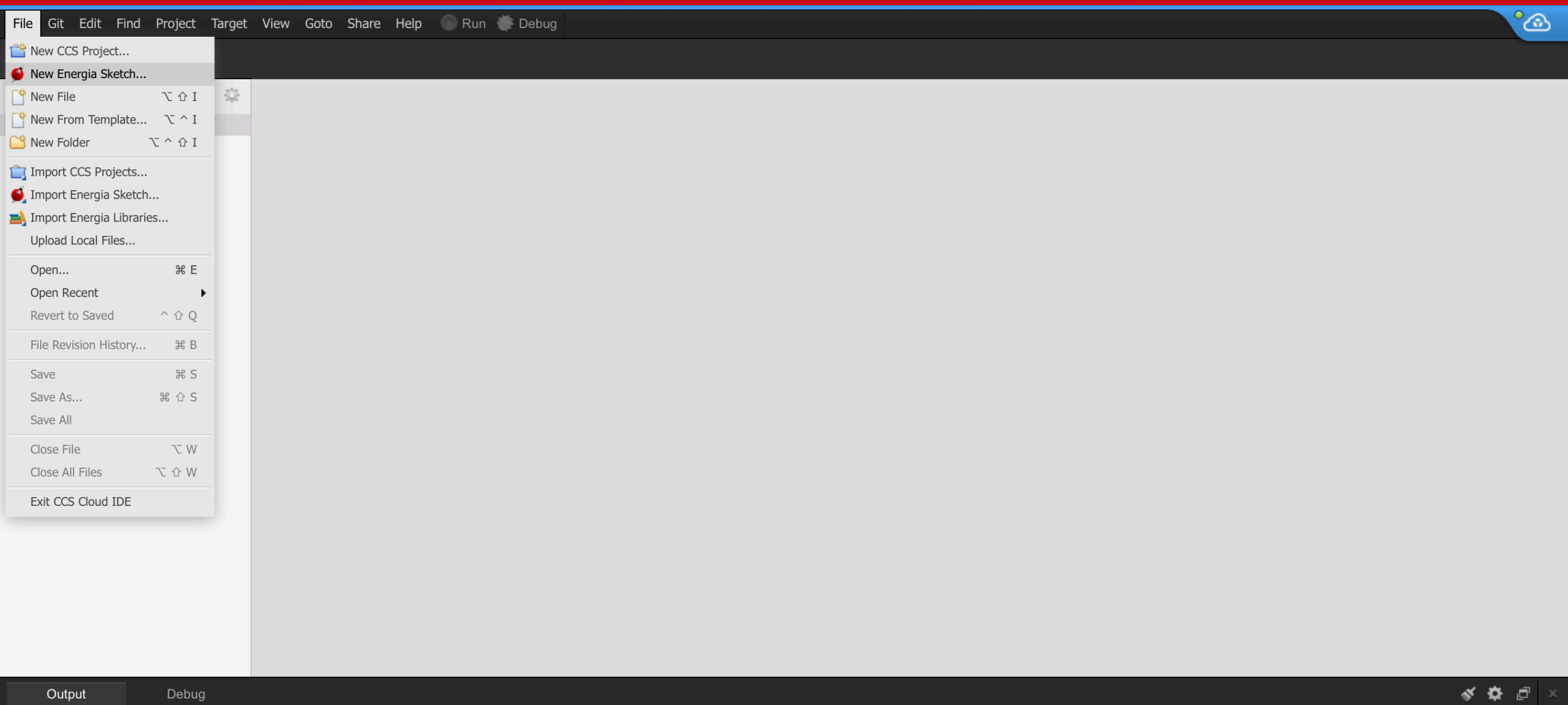
RED LaunchPad w/ msp432 EMT (48MHz) on /dev/cu.usbmodemM4321001



# Energia Tool-Bar



# CCS Cloud Interface





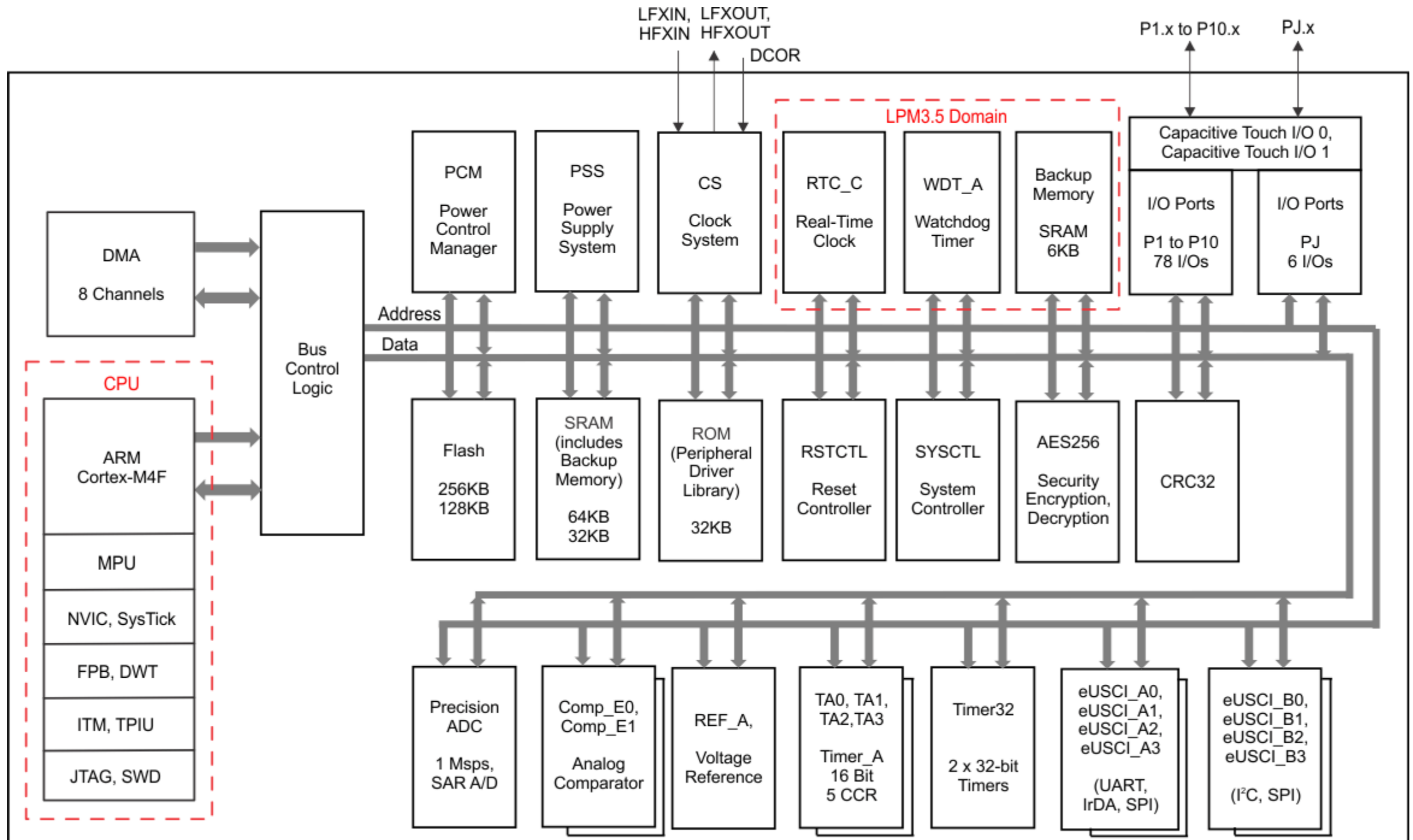
# Agenda

- **Part 2 — LaunchPad and Energia**
  - The LaunchPad Family
  - Discover the MSP432 LaunchPad
  - Example: Blink a LED
  - Example: Port the code to another LaunchPad

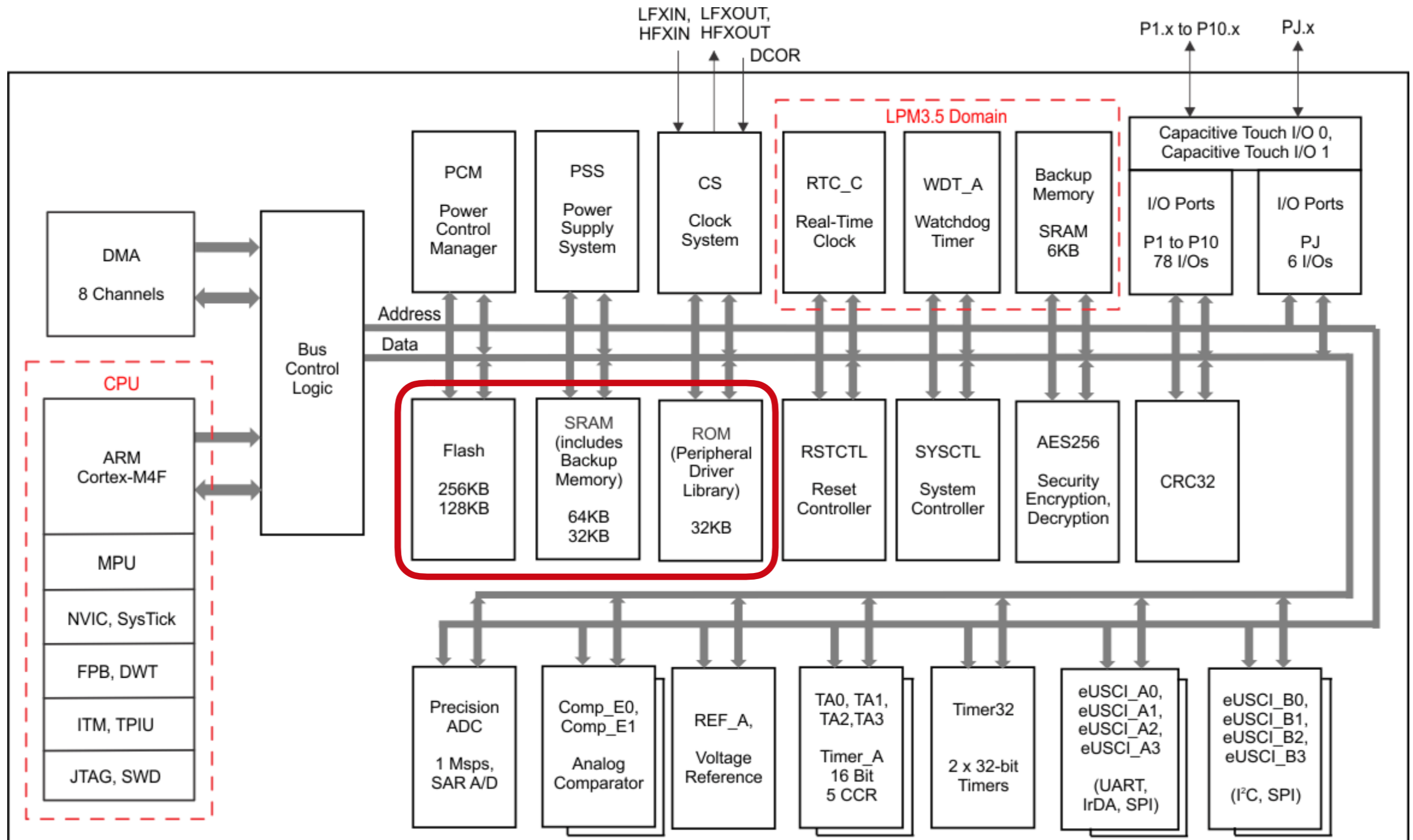
# MCU *vs.* MPU

- MCU *aka.* Micro-Controller Unit
- MPU *aka.* Micro-Processor Unit

# Inside the MSP432



# Inside the MSP432





# Perspective

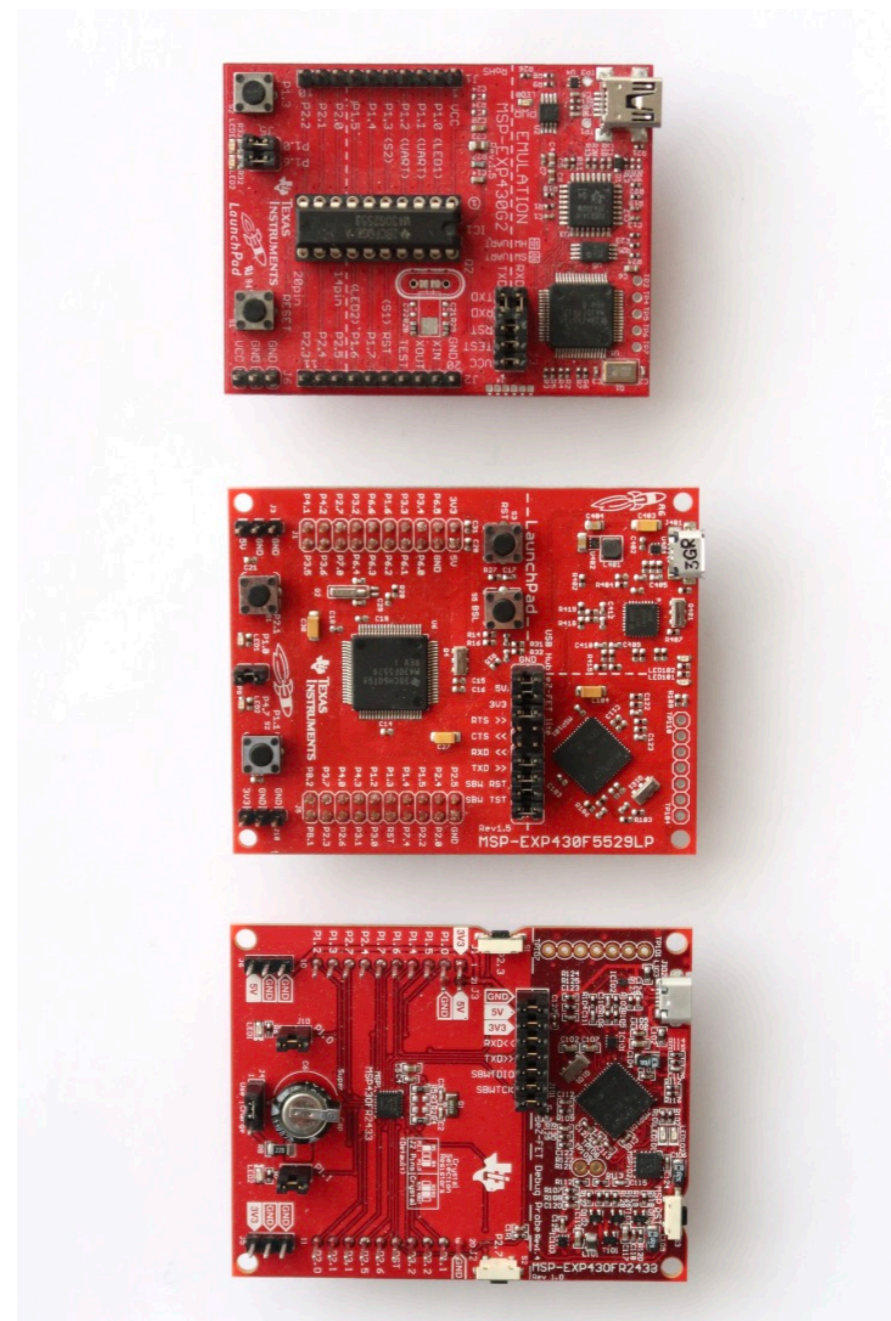
		IBM PC	LaunchPad MSP430G2	LaunchPad MSP432
<b>Year</b>		1981	2010	2015
<b>Processor</b>		Intel 8088	MSP430G2553	MSP432P401R
<b>Architecture</b>	<b>bits</b>	8	16	32
<b>Speed</b>	<b>MHz</b>	4,77	16	48
<b>RAM</b>	<b>kB</b>	16	0,5	64
<b>ROM</b>	<b>kB</b>	40	0	32
<b>Mass Storage</b>	<b>kB</b>	Dual 160~360 5¼" floppy disks	16	256
<b>Price</b>	<b>USD</b>	1 600 ~ 3 000	4,30	12,99



# The LaunchPad Family

## MSP430 Supported Boards

- **MSP430G2553**
- **MSP430F5529**
- **MSP430FRxxxx**
  - FRAM memory

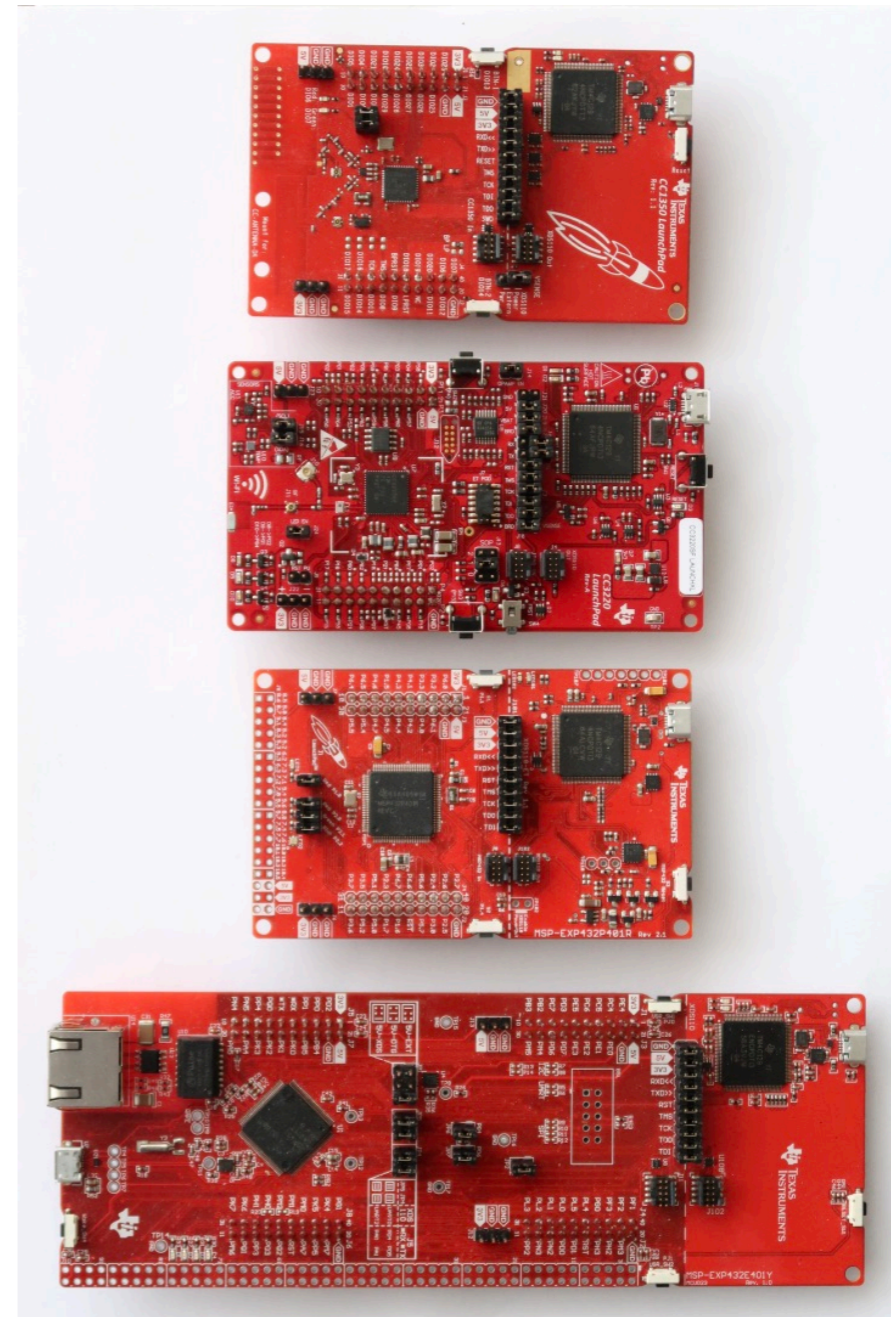




# The LaunchPad Family

## SimpleLink Supported Boards

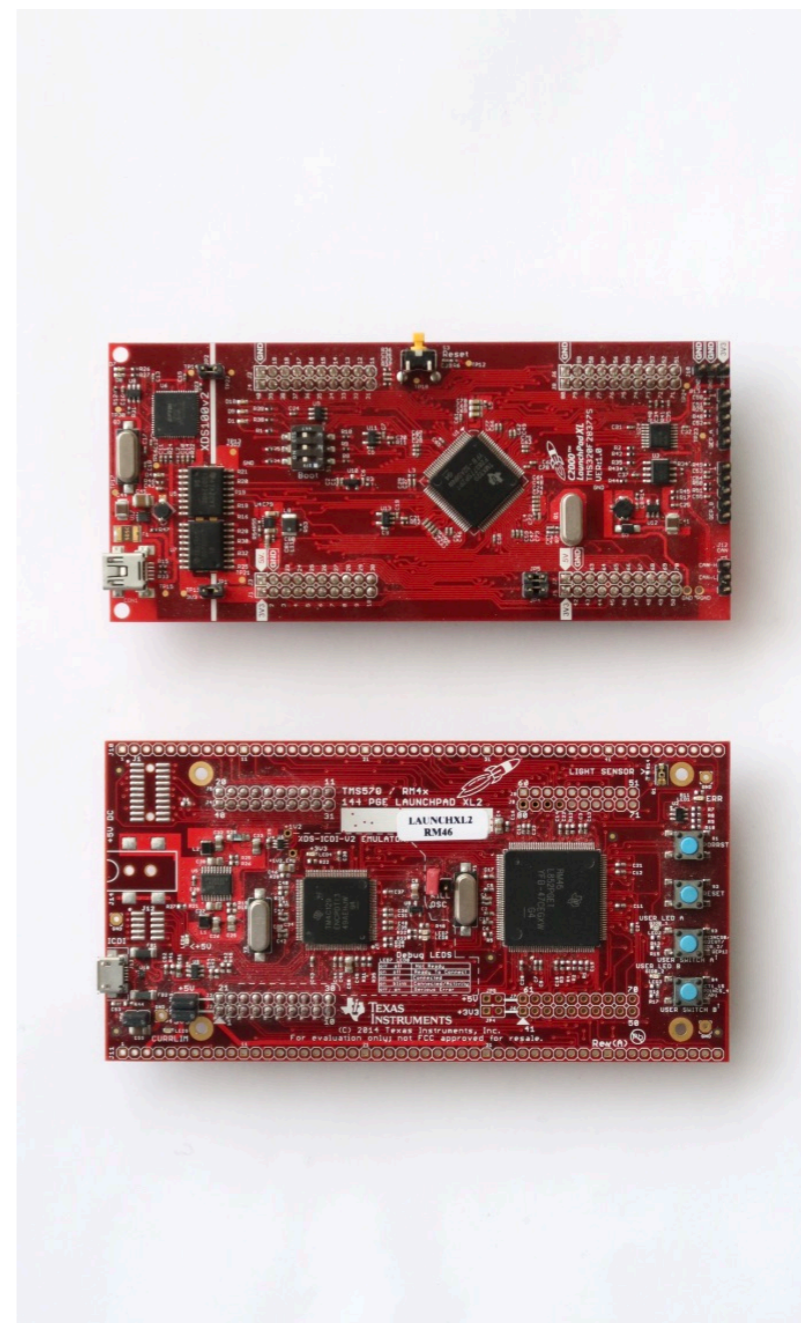
- Cortex-M3 **CC13x0** sub-1 GHz
- Cortex-M3 **CC32x0** WiFi
- Cortex-M4F **MSP432P401R**
- Cortex-M4F **MSP432E401Y** Ethernet
- *Cortex-M3 **CC2650** Bluetooth not supported due to license restrictions*



# Other LaunchPad Boards

## Not Supported by Energia

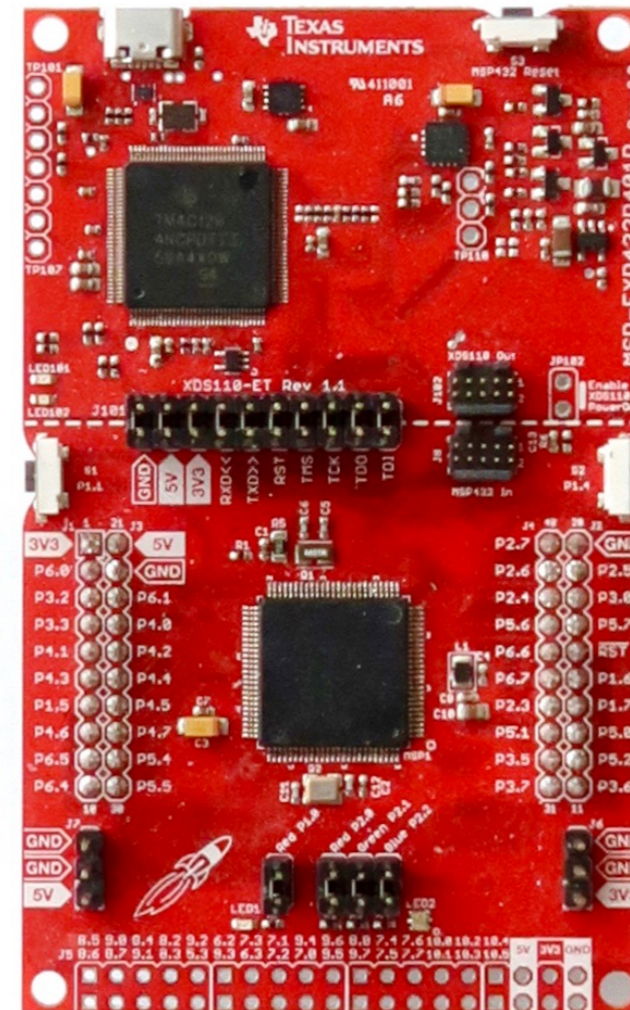
- **C2000** — *up to Energia 17*
  - 32-bit MCUs
  - Real-time process control
- **Hercules**
  - 32-bit ARM Cortex-R4F and -R5F MCUs
  - Safety critical applications





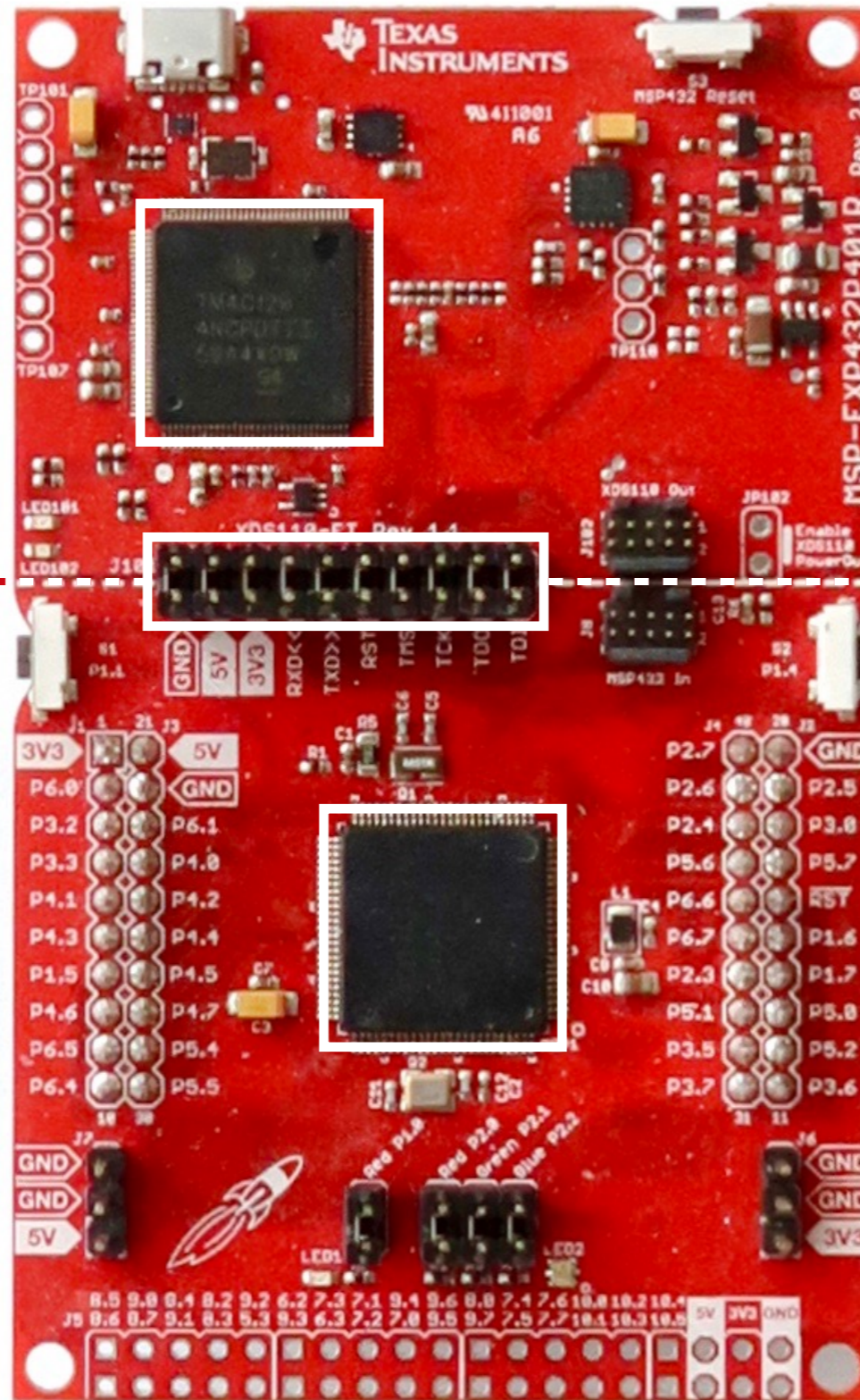
# MSP432 LaunchPad

- MSP432P401R MCU 48 MHz 32-bit ARM Cortex-M4F with FPU
  - Power consumption: 80  $\mu$ A / MHz active and 660 nA RTC standby operation
  - Digital: AES256 Accelerator, CRC, DMA, 32-bit Hardware Multiplier
  - Memory: 256 KB Flash, 64 KB RAM
  - Timers: 4 x 16-bit and 2 x 32-bit
  - Ports: Up to 4 I<sup>2</sup>C, 8 SPI, 4 UART
- 40-pin BoosterPack connector
- XDS-110ET emulator with EnergyTrace+
- Less than USD13



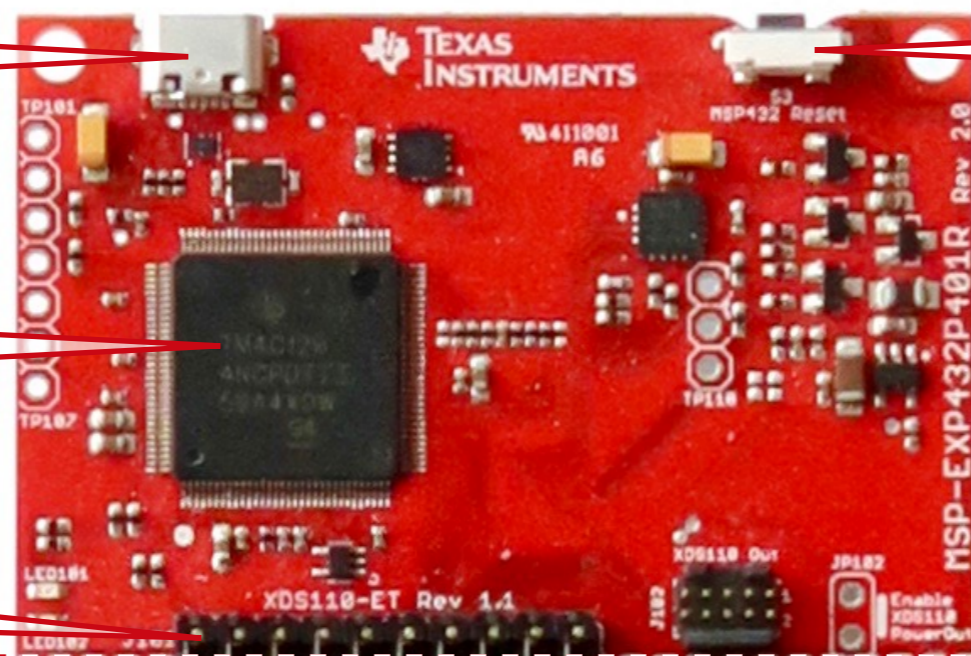


**Programmer  
Debugger**



**MCU**



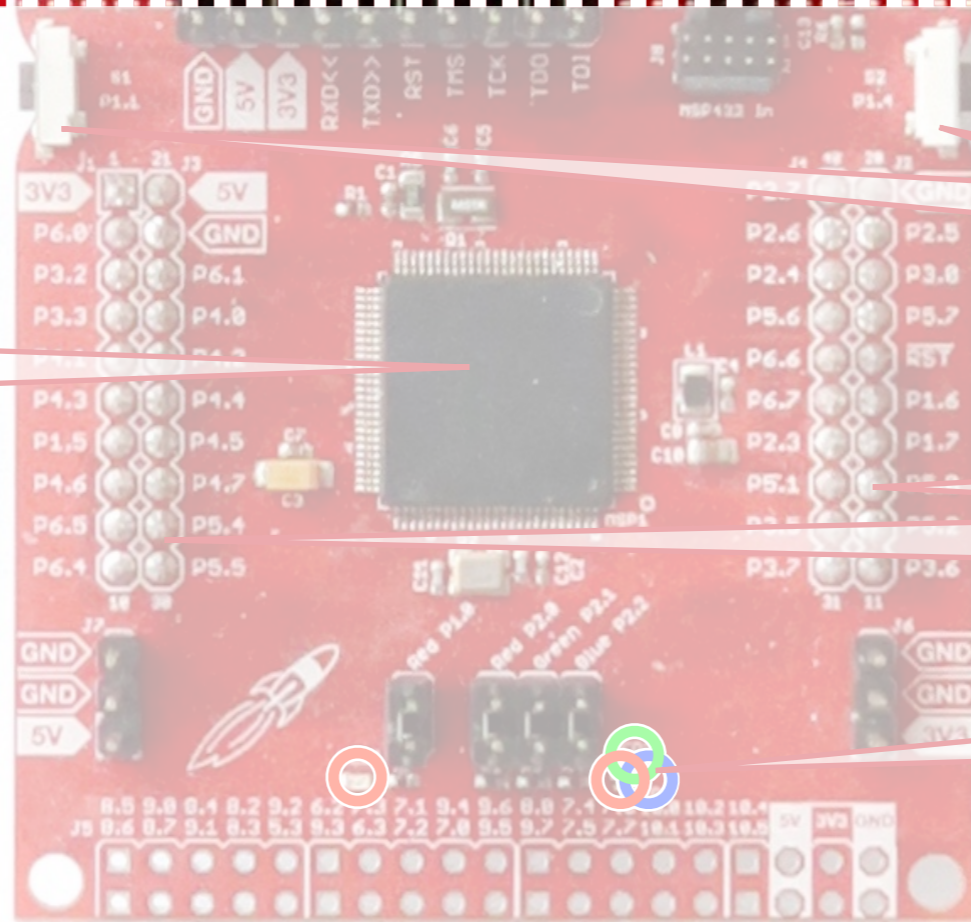


USB

Reset

Programmer Debugger

Isolation Jumpers



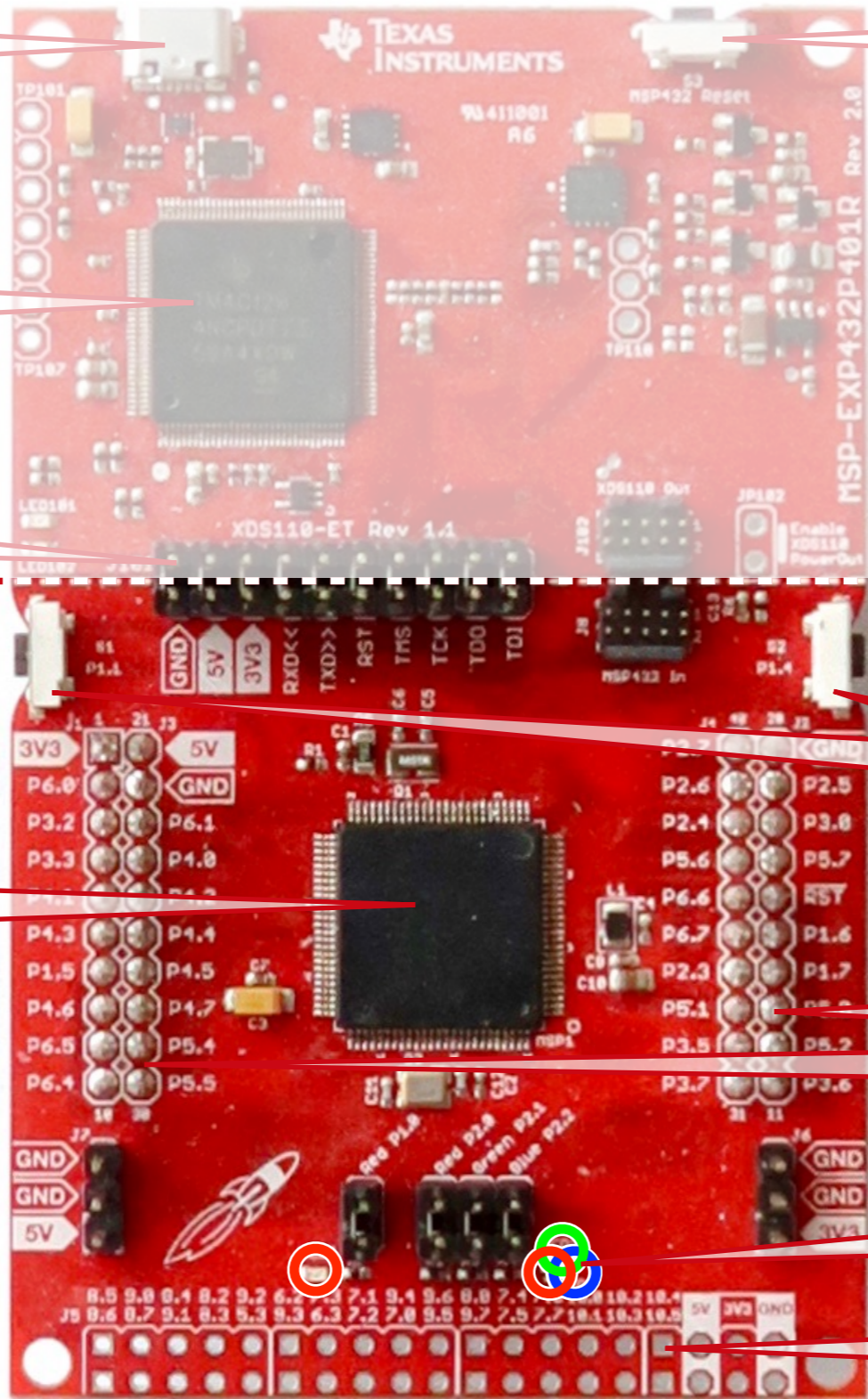
MCU

User Buttons

BoosterPack Connectors

RGB LED





USB

Reset

Programmer Debugger

Isolation Jumpers

MCU

User Buttons

BoosterPack Connectors

RGB LED

More GPIOs

# A Sketch with Two Functions

- A program is called a sketch
- Two main functions:
  - `setup()`, to run once
  - `loop()`, to run repeatedly

```
98 // Add setup code, to run once
99 void setup()
100 {
101
102 }
103
104 // Add loop code, to run repeatedly
105 void loop()
106 {
107
108 }
```



# Hidden File and Include

- The standard `main.cpp` source file is hidden
- It contains the statement `#include "Energia.h"` to include the Energia framework
- and the `main()` function which calls `setup()` and `loop()`

```
10 #include <Energia.h>
11
12 int main(void)
13 {
14     init();
15
16     setup();
17
18     for (;;) {
19         loop();
20         if (serialEventRun) serialEventRun();
21     }
22
23     return 0;
24 }
25
26
27 #include "BlinkRedLED.ino"
```

# The RED\_LED GPIO

- RED\_LED is the name of the red LED available on all the LaunchPads

- For MSP432,  
P2\_0: port 2 bit 0

```
// MSP_EXP432P401R  
// 75 - P2.0 RED_LED  
static const uint8_t RED_LED = 75;
```

- For MSP430FR2433,  
P1\_0: port 1 bit 0

```
// MSP430FR2433  
static const uint8_t P1_0 = 2;  
static const uint8_t RED_LED = 2;
```

- Selecting the board provides its definition
- Read the data-sheet at [ti.com](http://ti.com)!
- Check the pins maps at [energia.nu/pin-maps](http://energia.nu/pin-maps)

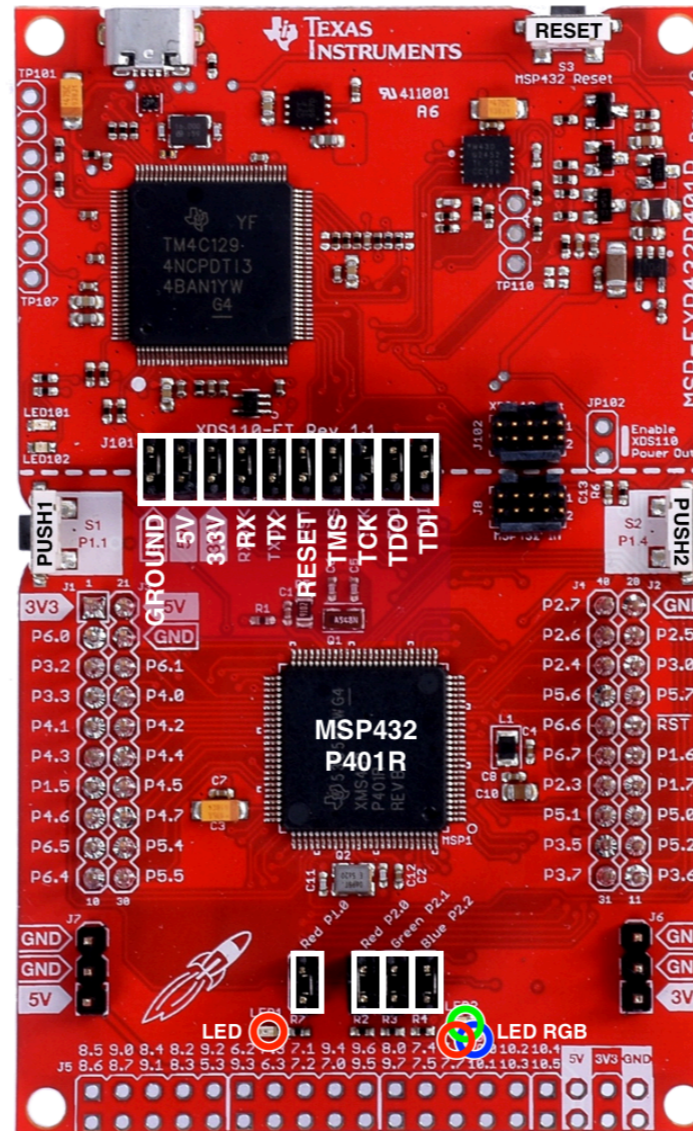
# LaunchPad with MSP432P401R

Revision 2.0

Flash 256 KB  
 SRAM 64 KB

Serial	hardware
ADC	14 bits
Use pins numbers only!	
Default I <sup>2</sup> C = (1)	
I <sup>2</sup> C (1) master only	

Hardware
Pin number
Other pin number
I <sup>2</sup> C
Serial UART
SPI
analogRead()
digitalRead() and digitalWrite()
digitalRead(), digitalWrite() and analogWrite()



		+3.3V		J1	J3		+5V
		A15	P6_0	1	21		GROUND
RX (1)			P3_2	2	22	P6_1	A14
TX (1)			P3_3	3	23	P4_0	A13
		A12	P4_1	4	24	P4_2	A11
		A10	P4_3	5	25	P4_4	A9
SCK			P1_5	6	26	P4_5	A8
		A7	P4_6	7	27	P4_7	A6
	SCL (1)		P6_5	8	28	P5_4	A1
	SDA (1)		P6_4	9	29	P5_5	A0

GROUND
GROUND
+5V

73	P1_1	PUSH1
74	P1_4	PUSH2
75	P2_0	RED_LED
76	P2_1	GREEN_LED
77	P2_2	BLUE_LED
78	P1_0	YELLOW_LED

		J4	J2		GROUND		
		P2_7	40	20			
		P2_6	39	19	P2_5		
		P2_4	38	18	P3_0		CS
		P5_6	37	17	P5_7		
		P6_6	36	16		RESET	
		P6_7	35	15	P1_6		SDA (0) MOSI
		P2_3	34	14	P1_7		SCL (0) MISO
	A4	P5_1	33	13	P5_0	A5	
		P3_5	32	12	P5_2	A3	
		P3_7	31	11	P3_6		

GROUND
GROUND
+3.3V

J5	41	56
	57	72



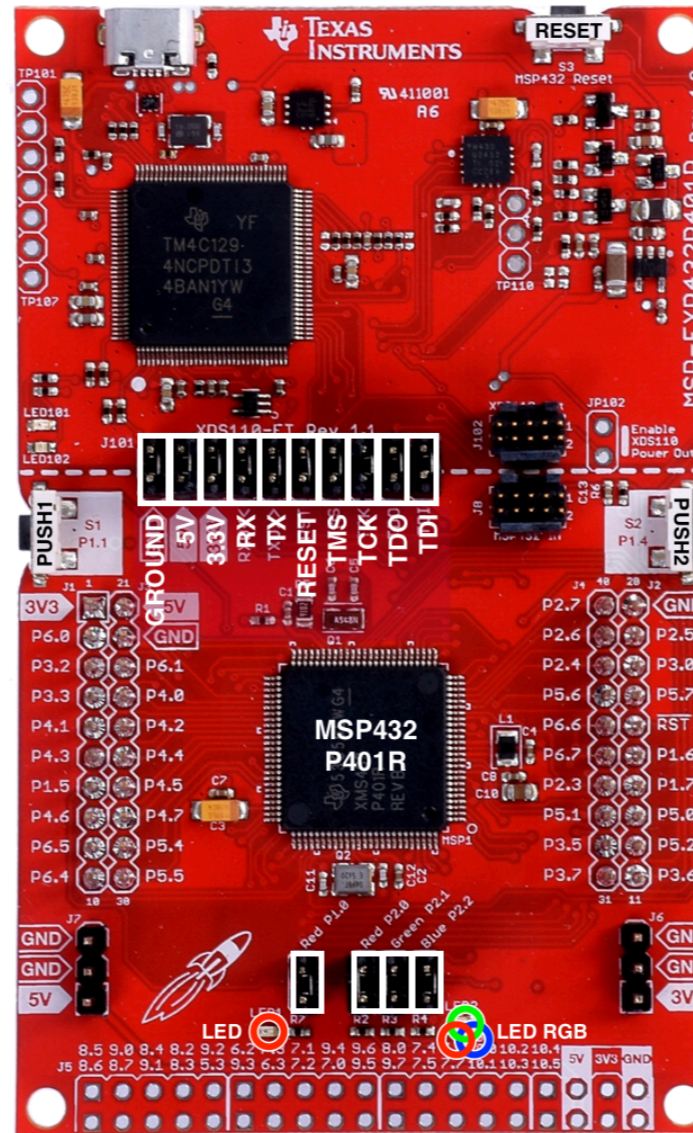
# LaunchPad with MSP432P401R

Revision 2.0

Flash 256 KB  
SRAM 64 KB

Serial	hardware
ADC	14 bits
Use pins numbers only!	
Default I <sup>2</sup> C = (1)	
I <sup>2</sup> C (1) master only	

Hardware
Pin number
Other pin number
I <sup>2</sup> C
Serial UART
SPI
analogRead()
digitalRead() and digitalWrite()
digitalRead(), digitalWrite() and analogWrite()



		+3.3V		J1	J3			+5V
		A15	P6_0	1	21			GROUND
RX (1)			P3_2	2	22			
TX (1)			P3_3	3	23	P6_1	A14	
			P3_4	4	24	P4_0	A13	
		A12	P4_1	5	25	P4_2	A11	
		A10	P4_3	6	26	P4_4	A9	
SCK			P1_5	7	27	P4_5	A8	
		A7	P4_6	8	28	P4_7	A6	
	SCL (1)		P6_5	9	29	P5_4	A1	
	SDA (1)		P6_4	10	30	P5_5	A0	

			J4	J2				
		P2_7	40	20		GROUND		
		P2_6	39	19	P2_5			
		P2_4	38	18	P3_0			CS
		P5_6	37	17	P5_7			
		P6_6	36	16		RESET		
		P6_7	35	15	P1_6		SDA (0)	MOSI
		P2_3	34	14	P1_7		SCL (0)	MISO
	A4	P5_1	33	13	P5_0	A5		
		P3_5	32	12	P5_2	A3		
		P3_7	31	11	P3_6			

75 P2\_0 RED\_LED

76	P1_1	PUSH1
74	P1_2	PUSH2
75	P2_0	RED_LED
76	P2_1	GREEN_LED
77	P2_2	BLUE_LED
78	P1_0	YELLOW_LED

J5	41	56
	57	72



# Using GPIOs

- Initialise a GPIO with `pinMode()`
- Use a GPIO with
  - `digitalWrite()` to write,
  - `digitalRead()` to read
- See the documentation at [energia.nu/reference](http://energia.nu/reference)
- Always use the pin numbers, not the pin names (port, bit)



Home Download Guide Reference Blog Store Getting Help IRC Energia Projects Events

Reference [Language](#) | [Libraries](#) | [Pin Maps](#) | [Comparison](#) | [Changes](#)

## 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`
- ...

### Variables

#### Constants

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

### Functions

#### Digital I/O

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

#### Analog I/O

- `analogReference()`
- ...

# The setup() Function

- `pinMode()` sets the direction of the pin, here OUTPUT
- All functions are documented at [energia.nu/reference](http://energia.nu/reference)
- Always use the pin numbers, not the pin names (port, bit)

```
98 // Add setup code
99 void setup()
100 {
101     pinMode(RED_LED, OUTPUT);
102 }
```

# The loop() Function

- digitalWrite() manages the GPIO when used as output:
  - LOW to clear
  - HIGH to set
- delay() waits for a period set in ms
- All functions are documented at [energia.nu/reference](http://energia.nu/reference)

```
104 // Add loop code
105 void loop()
106 {
107     digitalWrite(RED_LED, HIGH);
108     delay(100);
109     digitalWrite(RED_LED, LOW);
110     delay(1000);
111 }
```



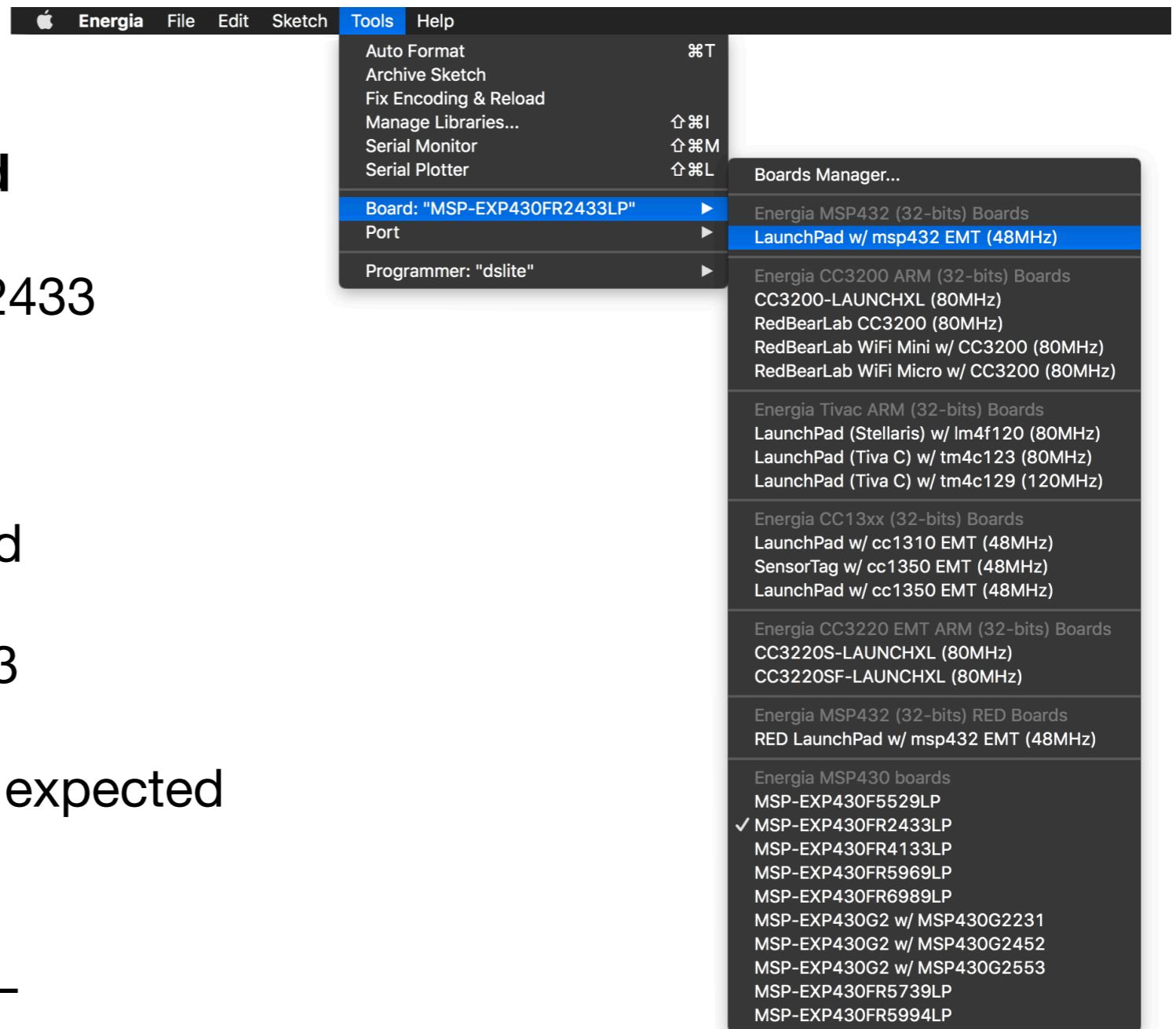
# Final Sketch

- The blinking LED is the *Hello World!* for embedded systems

```
98 // Add setup code
99 void setup()
100 {
101     pinMode(RED_LED, OUTPUT);
102 }
103
104 // Add loop code
105 void loop()
106 {
107     digitalWrite(RED_LED, HIGH);
108     delay(100);
109     digitalWrite(RED_LED, LOW);
110     delay(1000);
111 }
```

# Target Another LaunchPad

- Swap boards
- Call menu **Tools > Board**
- For users of MSP430FR2433
  - Select MSP432 Red
- For users of MSP432 Red
  - Select MSP430FR2433
- Check the LED blinks as expected
- This is called Hardware Abstraction Layer, or HAL





# LaunchPad with MSP430FR2433

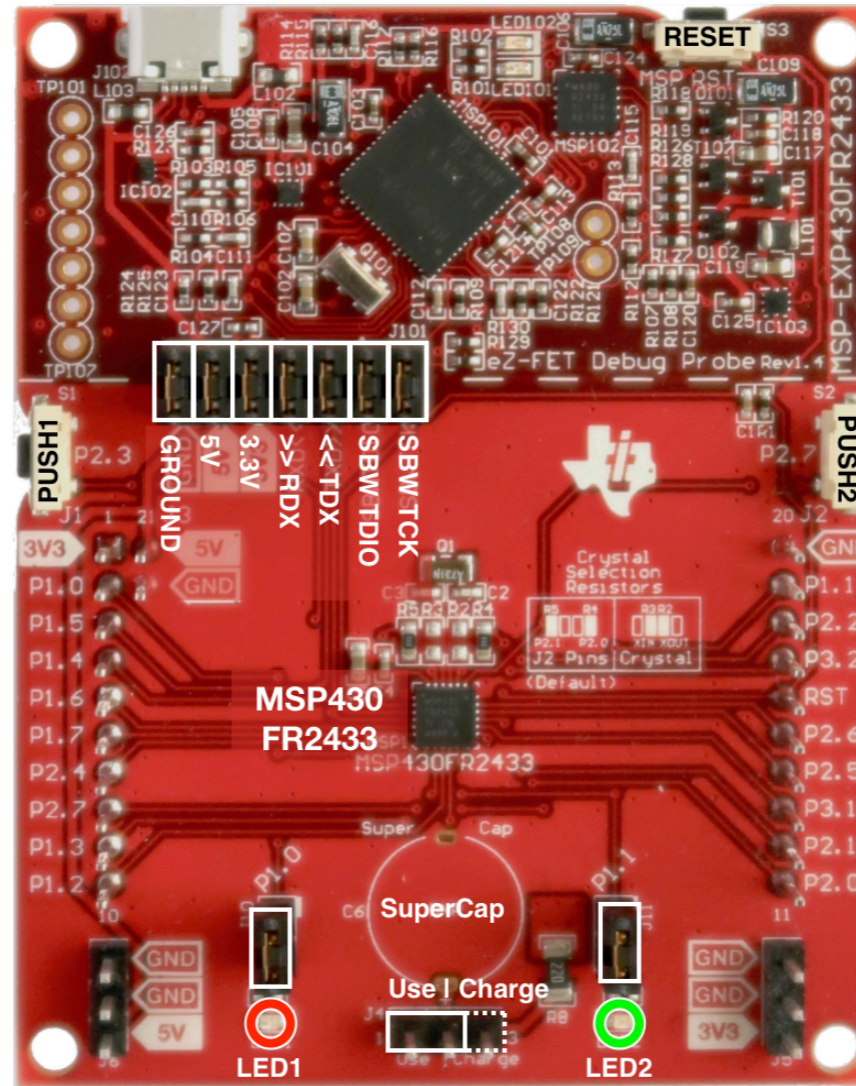
Revision 1.0

FRAM 16 KB  
SRAM 4 KB

Serial hardware  
ADC 10 bits  
Use pins numbers only!  
Default I<sup>2</sup>C = (0)  
Software I<sup>2</sup>C (1) master only

+3.3V				1	+5V
RED_LED		A0	P1_0	2	GROUND
	RXD	A5	P1_5	3	
	TXD	A4	P1_4	4	
		A6	P1_6	5	
		A7	P1_7	6	
	SCK		P2_4	7	
PUSH2			P2_7	8	
	SCL (0)	A3	P1_3	9	
	SDA (0)	A2	P1_2	10	

GROUND
GROUND
+5V



Hardware
Pin number
Other Pins

I <sup>2</sup> C
Serial UART
SPI

analogRead()
digitalRead() and digitalWrite()
digitalRead(), digitalWrite() and analogWrite()

20				GROUND
19	P1_1	A1		GREEN_LED
18	P2_2			
17	P3_2			
16				RESET
15	P2_6	SDA (1)	MOSI	
14	P2_5	SCL (1)	MISO	
13	P3_1			
12	P2_1			
11	P2_0			

GROUND
GROUND
+3.3V

21	P2_3			PUSH1
		A12		TEMP





# LaunchPad with MSP430FR2433

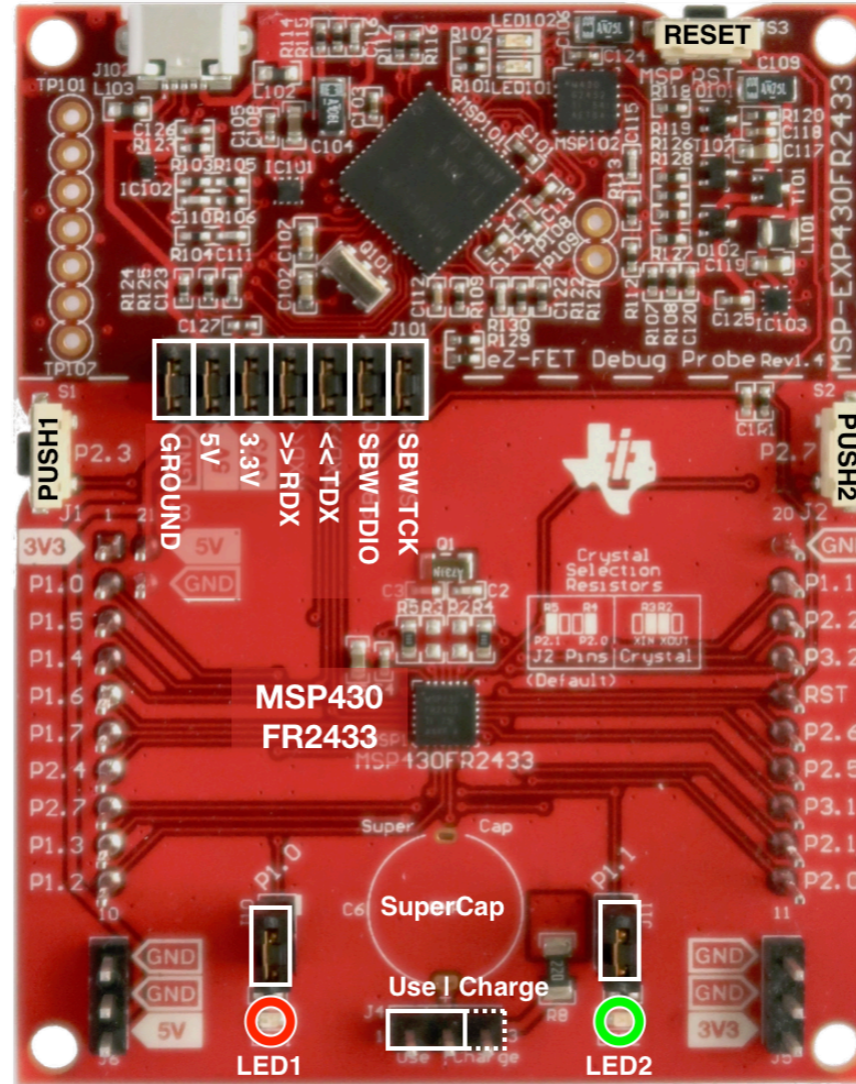
Revision 1.0

FRAM 16 KB  
SRAM 4 KB

- Serial hardware
- ADC 10 bits
- Use pins numbers only!
- Default I<sup>2</sup>C = (0)
- Software I<sup>2</sup>C (1) master only

	RED_LED	A0	P1_0	2	
P1.3				1	+5V
	RED_LED	A0	P1_0	2	GROUND
	HXD	A5	P1_5	3	
	TXD	A4	P1_4	4	
		A6	P1_6	5	
		A7	P1_7	6	
	SCK		P2_4	7	
PUSH2			P2_7	8	
	SCL (0)	A3	P1_3	9	
	SDA (0)	A2	P1_2	10	

GROUND
GROUND
+5V



Hardware
Pin number
Other Pins

I <sup>2</sup> C
Serial UART
SPI

analogRead()
digitalRead() and digitalWrite()
digitalRead(), digitalWrite() and analogWrite()

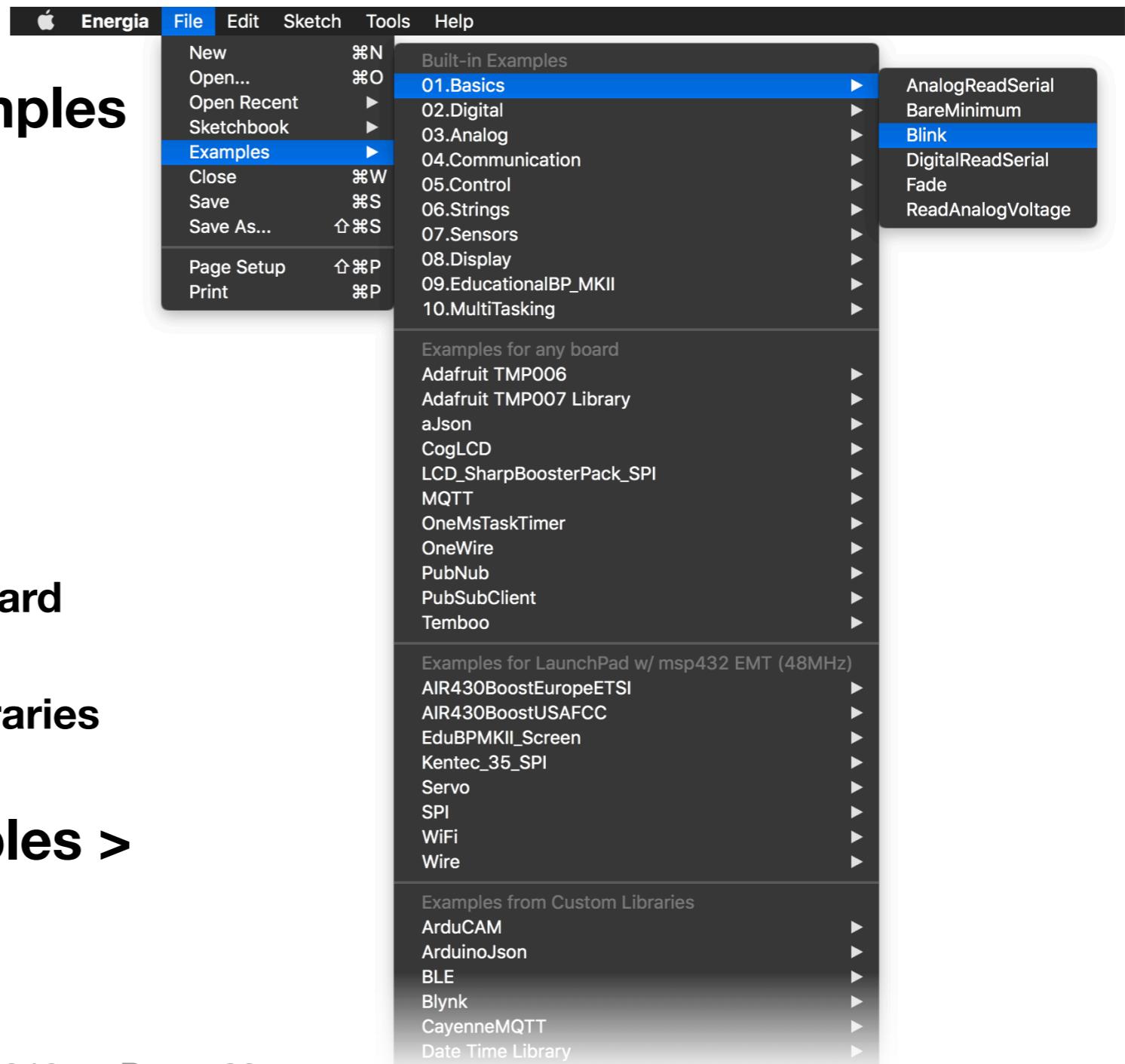
20				GROUND
19	P1_1	A1		GREEN_LED
18	P2_2			
17	P3_2			
16				RESET
15	P2_6	SDA (1)	MOSI	
14	P2_5	SCL (1)	MISO	
13	P3_1			
12	P2_1			
11	P2_0			

GROUND
GROUND
+3.3V

21	P2_3			PUSH1
	A12			TEMP

# Play with the Examples

- Call menu **File > Examples**
- Four options
  - **Built-in Examples**
  - **Examples for Any Board**
  - **Examples for Selected Board**
  - **Examples for Custom Libraries**
- Select **Built-In Examples > 01.Basics > Blink**

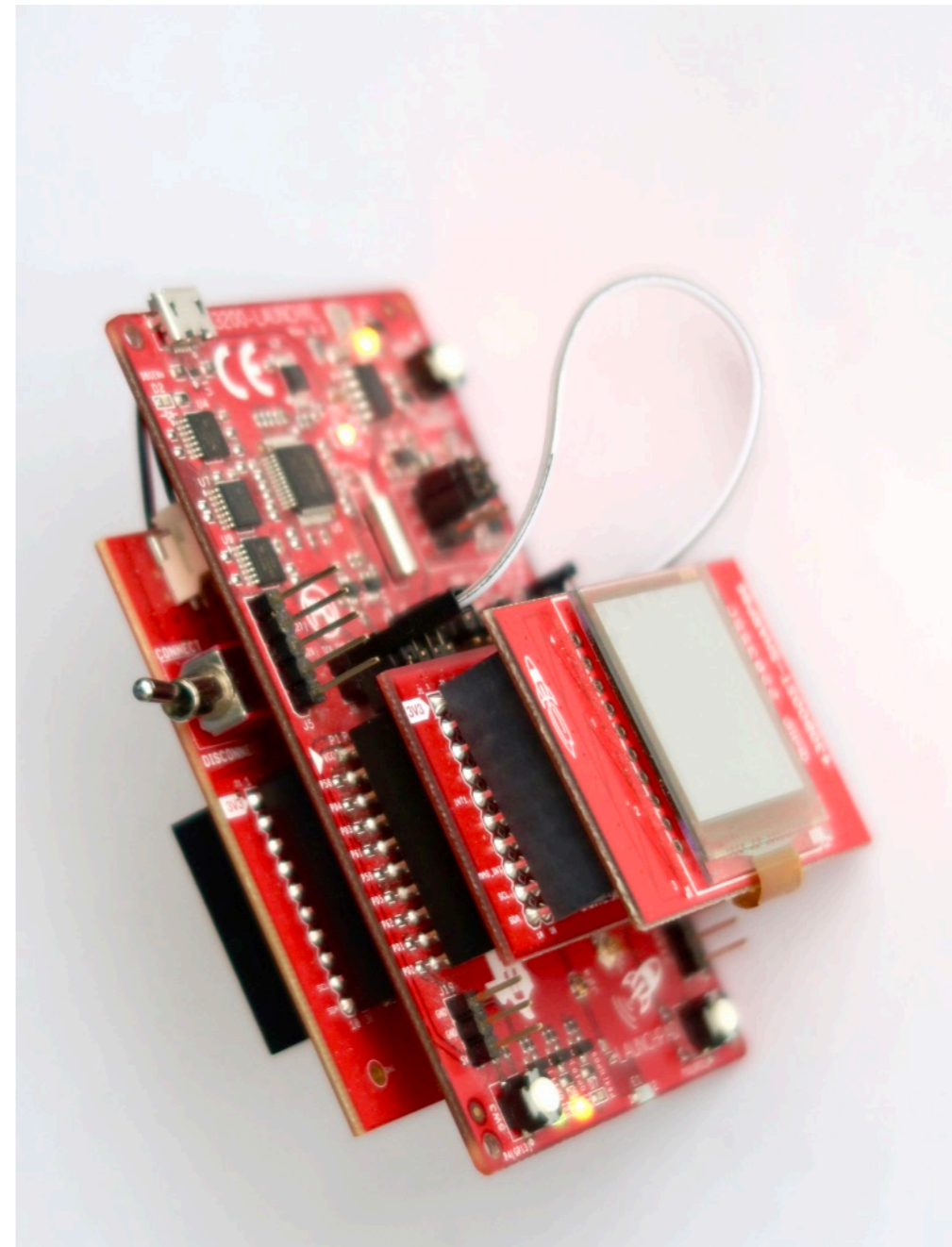


# Agenda

- **Part 3 — BoosterPack and Libraries**
  - Expand with BoosterPack
  - Drive Devices through Libraries
  - Example: Install the Educational BoosterPack
  - Example: Display Rainbow Colours
  - Example: Read the Temperature

# Expand with BoosterPack

- Hardware complement for LaunchPad
- Standard 20 or 40 pins
- To be stacked onto the LaunchPad
- Large range from Texas Instruments and third-party providers







# BoosterPack Standard

as per Dec 30, 2013

Hardware
Pin number
Other pin

I <sup>2</sup> C
Serial UART
SPI

analogRead()
digitalRead() and digitalWrite()
digitalRead(), digitalWrite() and analogWrite()

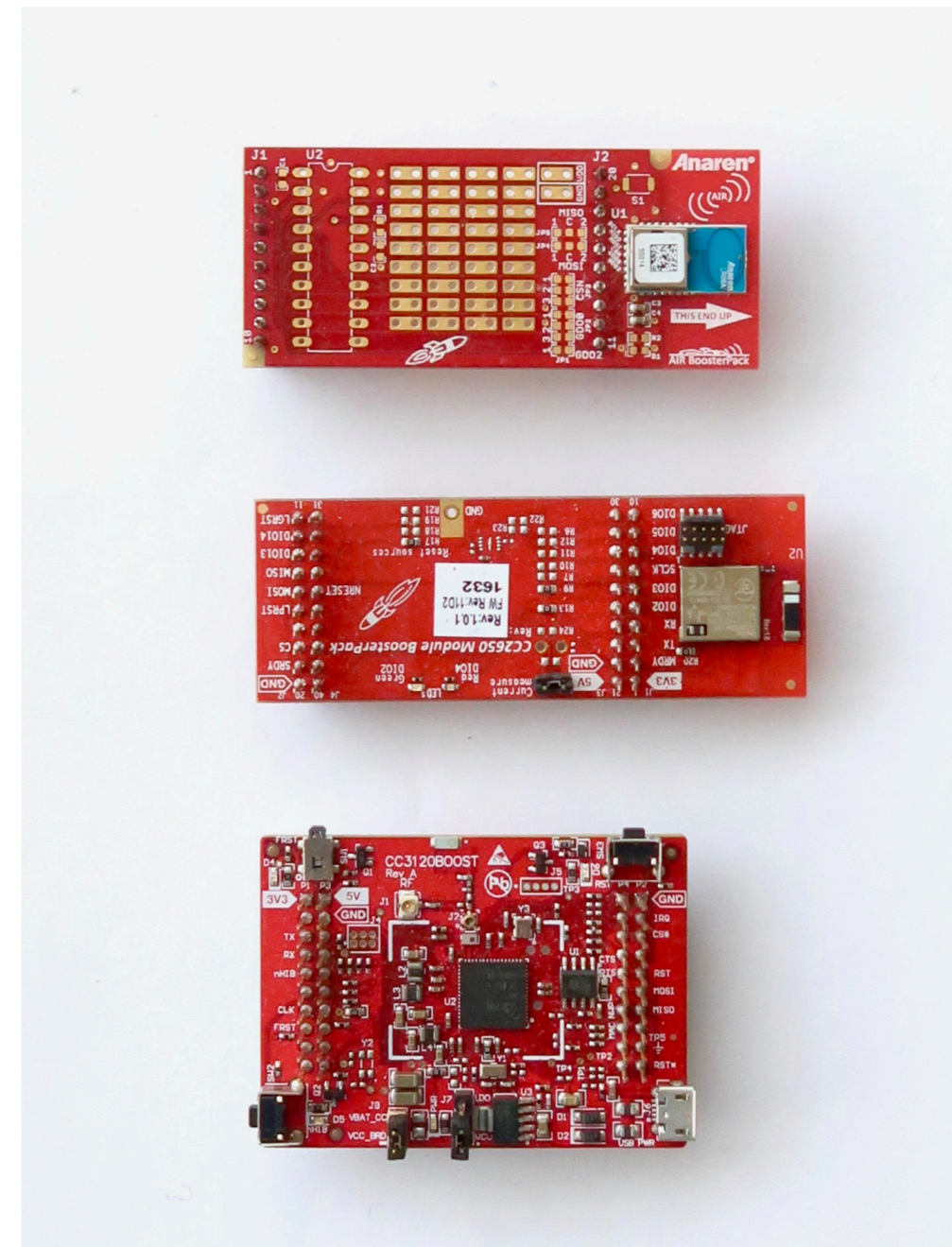
J1	BoosterPack		J3
1	+3.3 V	+5 V	21
2		GROUND	22
3	RX		23
4	TX		24
5			25
6			26
7	SCK		27
8			28
9	SCL		29
10	SDA		30

J4	BoosterPack		J2
40		GROUND	20
39			19
38		CS RF	18
37			17
36		RESET	16
35		MOSI	15
34		MISO	14
33		CS Display	13
32		CS Other	12
31			11

# The BoosterPack Family

## Radios

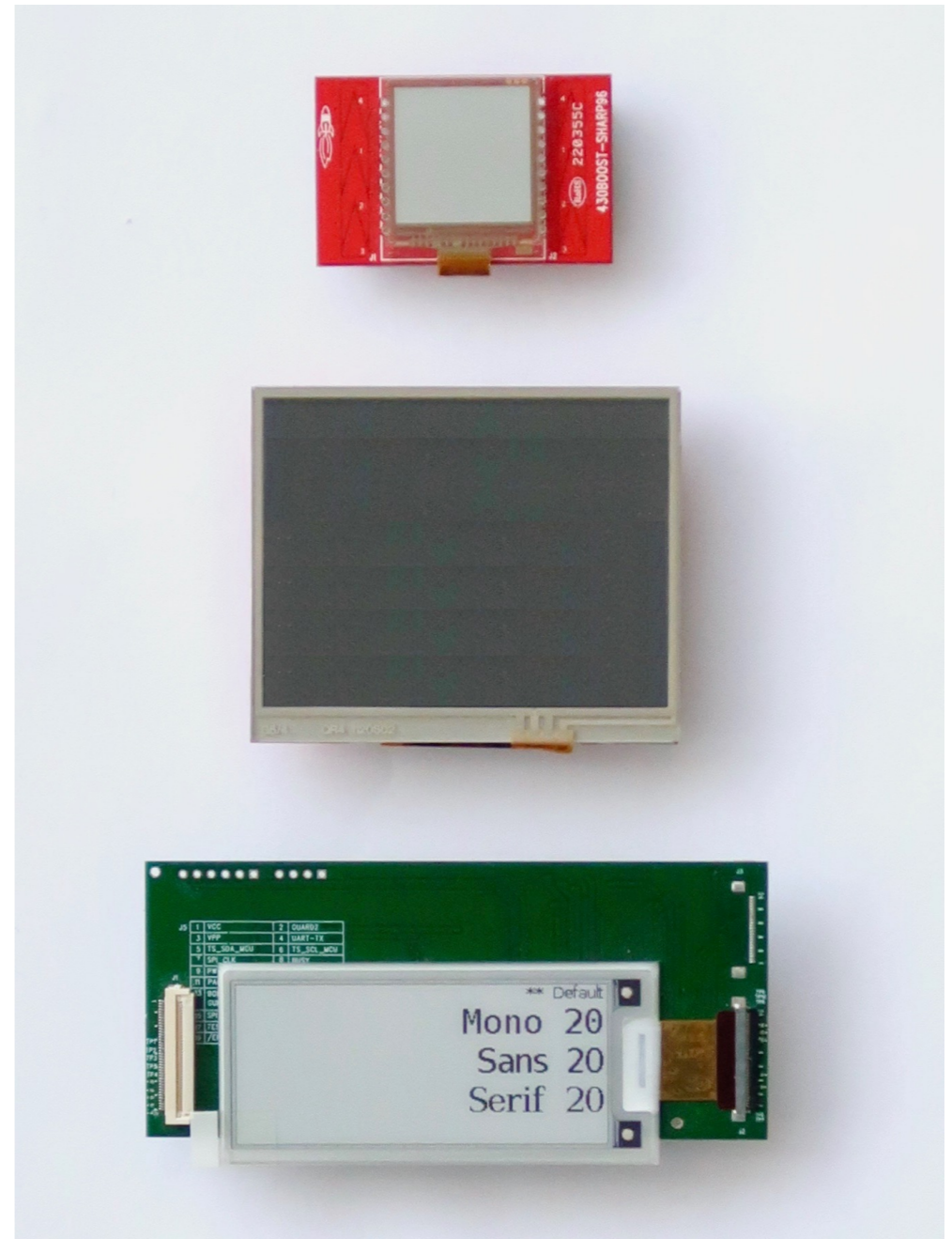
- Anaren sub-1 GHz CC110L
- Bluetooth CC2650
- WiFi CC31x0



# The BoosterPack Family

## Displays

- Sharp Memory Display
- Kentec 3.5" screen
- Pervasive Displays e-ink screen

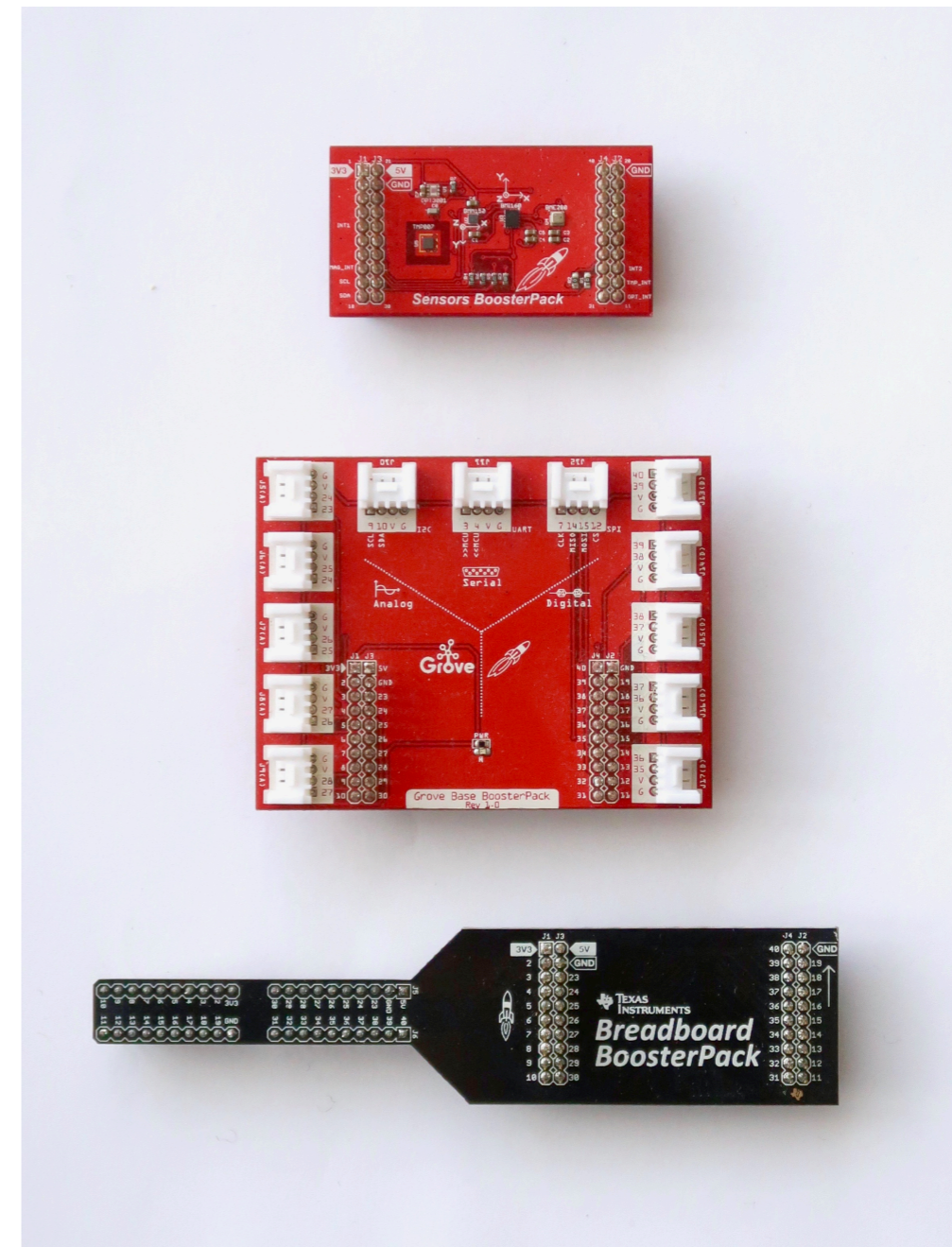




# The BoosterPack Family

## Multi-Sensors

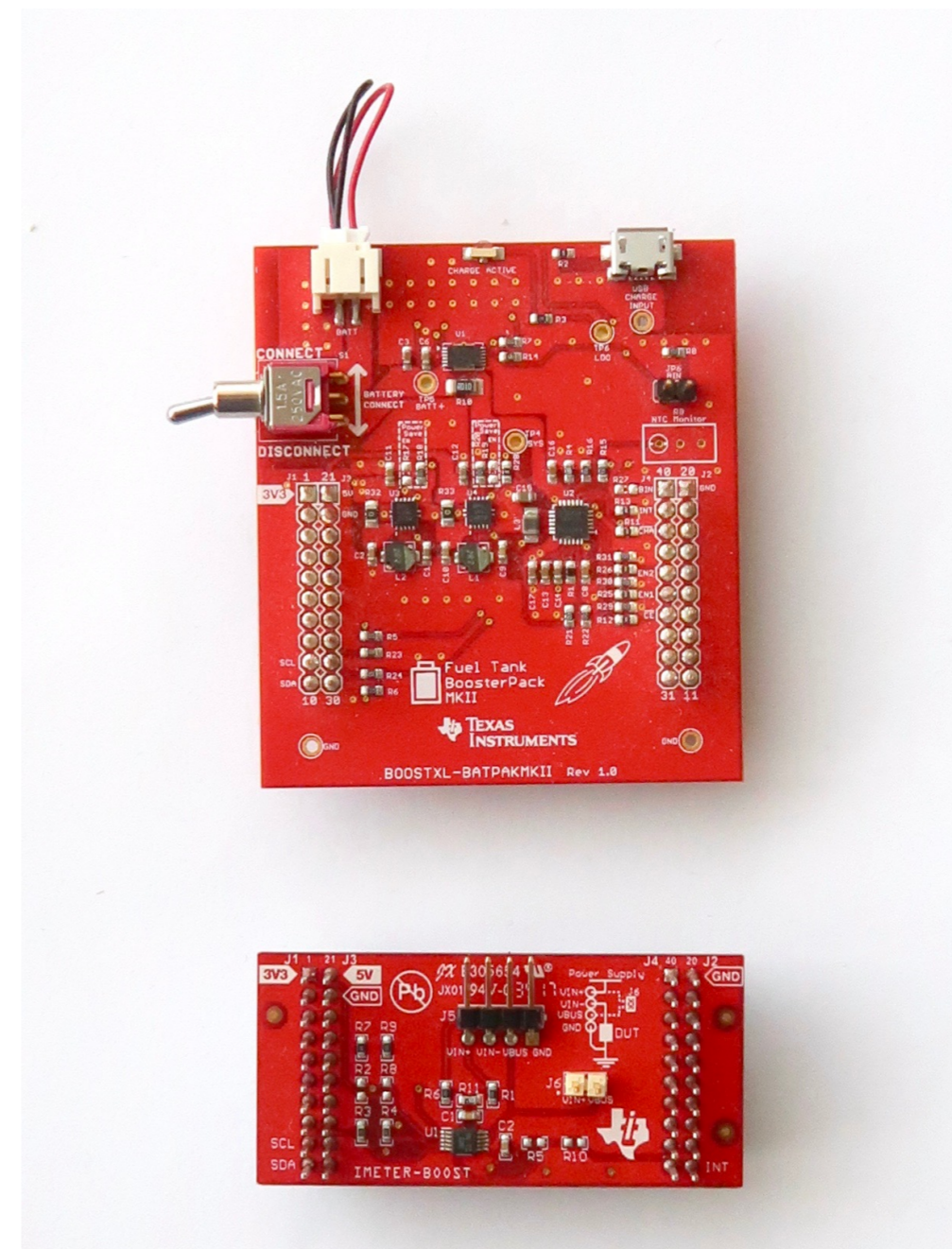
- Sensors
  - Weather and IMU
- Grove
- Breadboard
- Educational Mark-II



# The BoosterPack Family

## Power

- Fuel Tank Mark-II
- INA226 Volt-Amp-Watt-meter

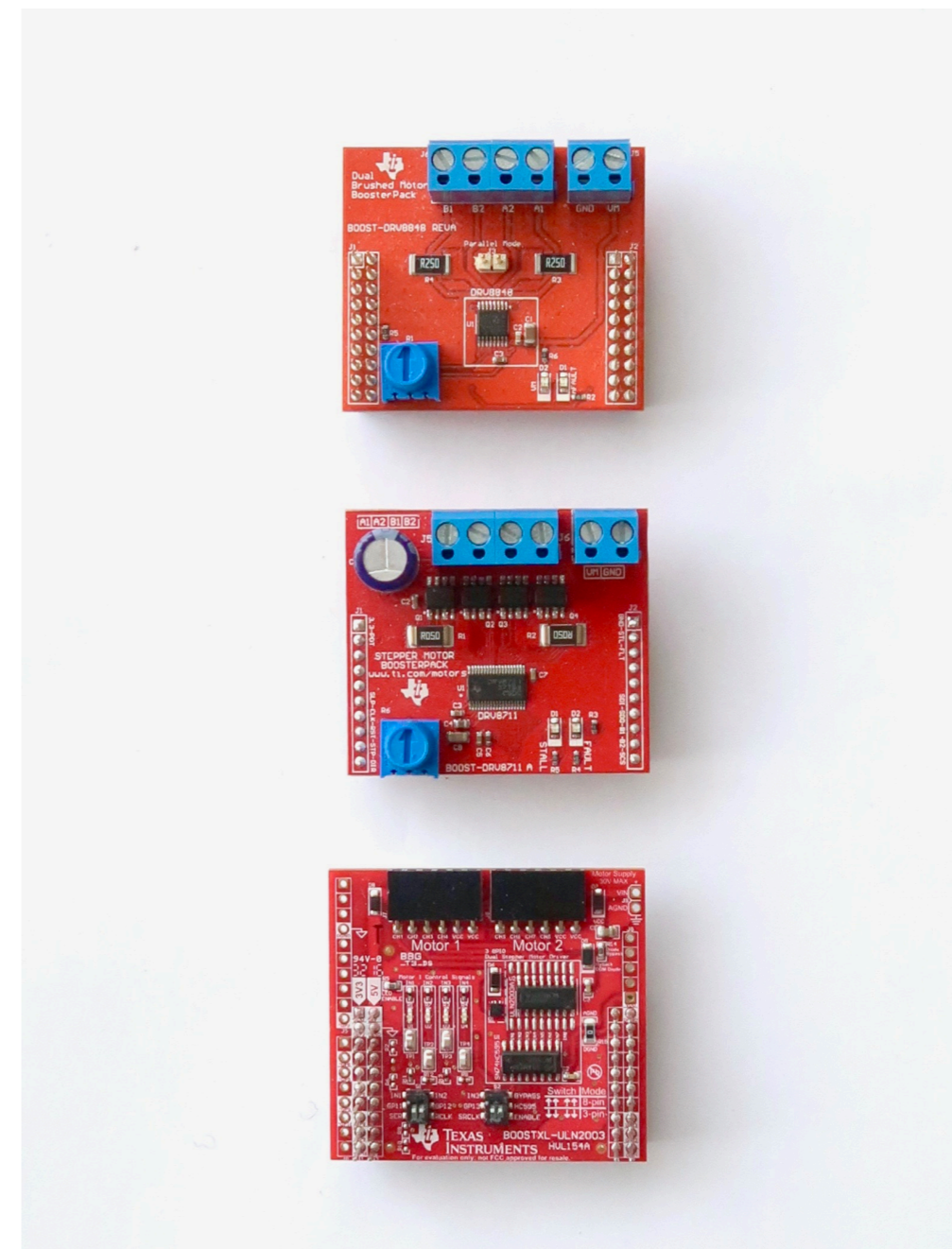




# The BoosterPack Family

## Motor

- DC-motor
- Stepper motor



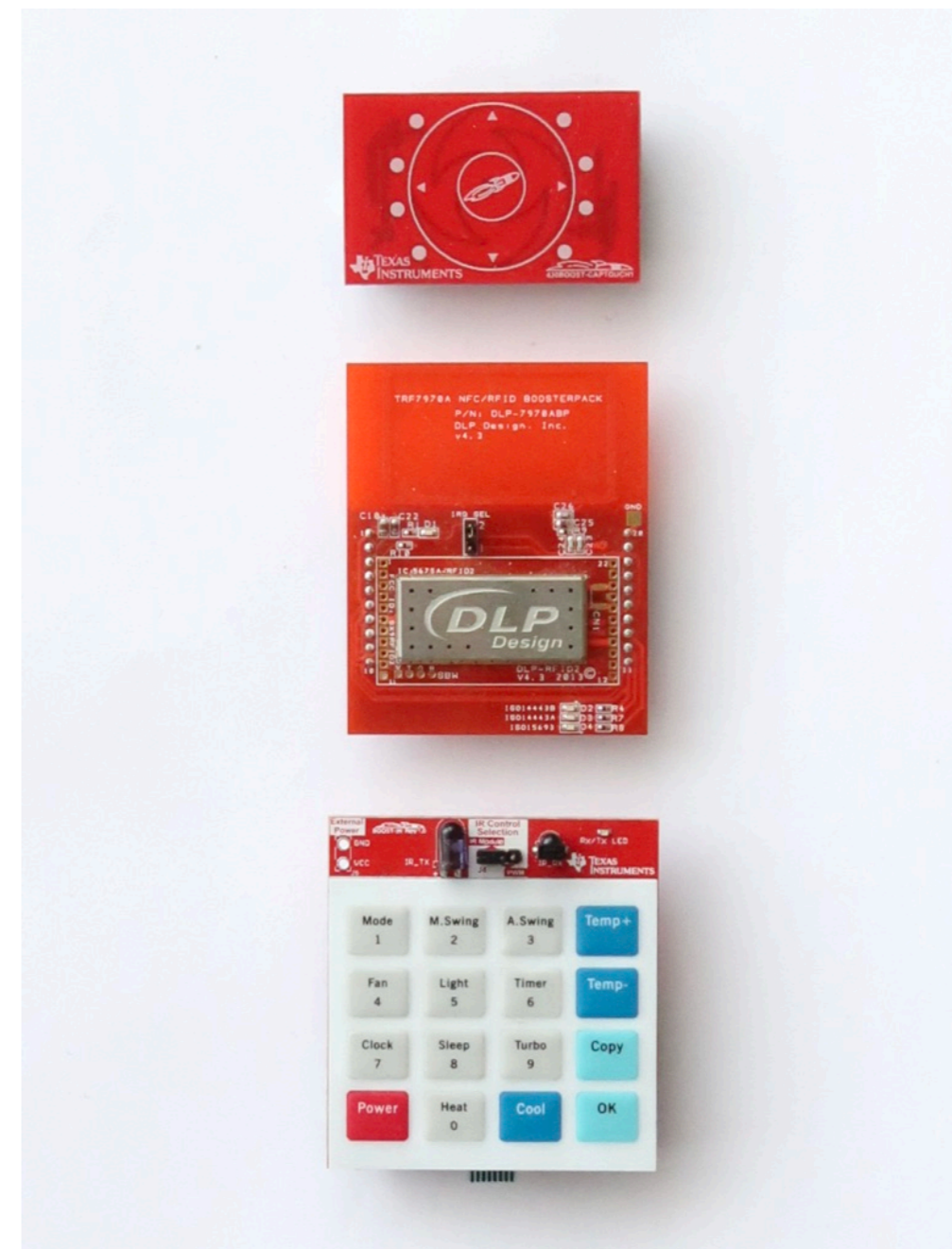
# The BoosterPack Family

## Other interfaces

- Touch
- RFID
- Infra-Red Remote

## And many more...

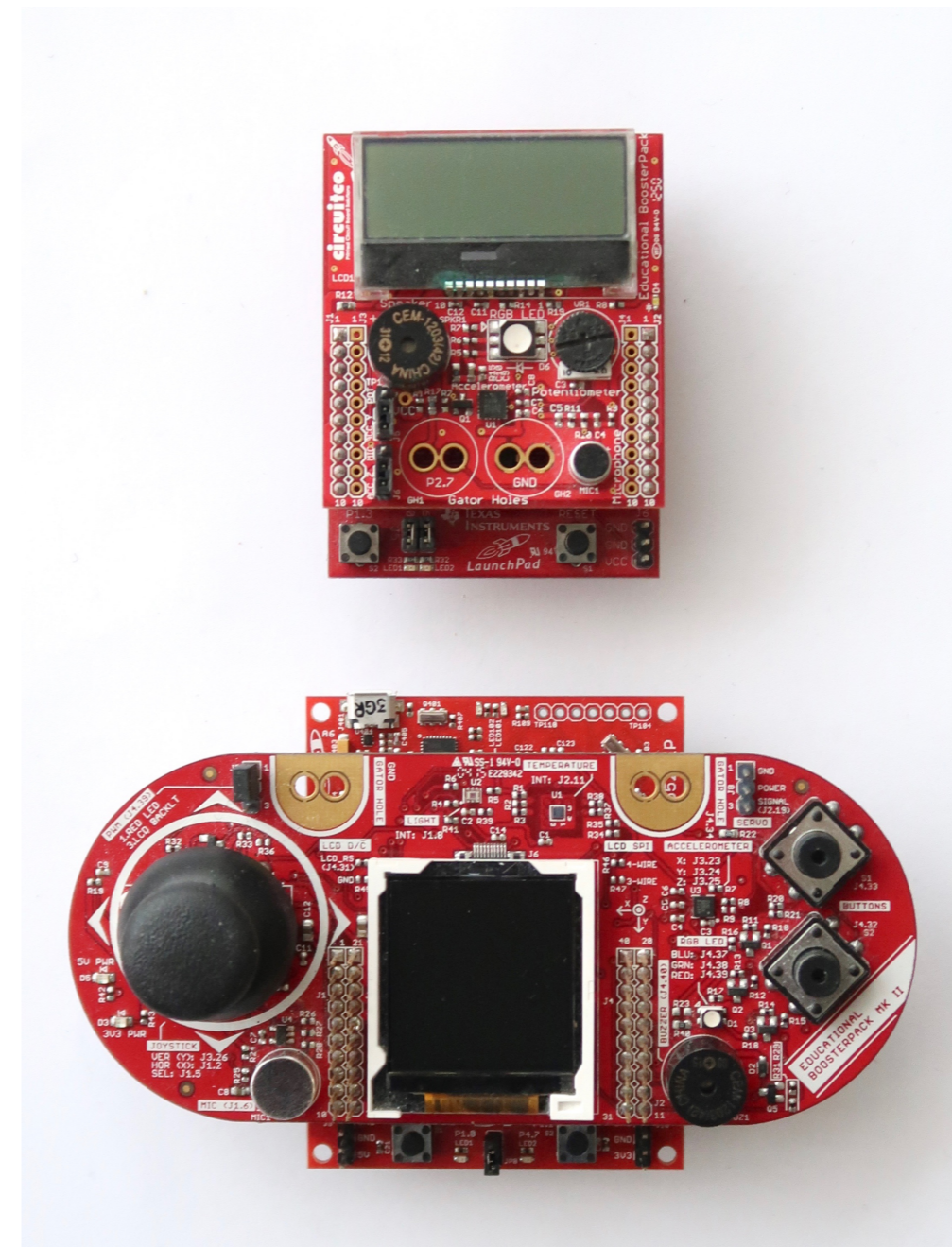
- Build your own!





# Educational BoosterPack Mark-II

- Second generation keeps:
  - Red-Green-Blue LED
  - 3-axis analog accelerometer
  - Buzzer
  - Microphone
- ...and adds:
  - 16-bit colour 128x128 LCD screen
  - Joystick and two buttons
  - Temperature sensor
  - Light sensor
  - Output for servo motor



# Know the BoosterPack

- Read the data-sheet at [ti.com](http://ti.com)!
- Check the pins map
  - Go to [energia.nu/pin-maps](http://energia.nu/pin-maps)
  - Select the BoosterPack
- Is the BoosterPack compatible with my LaunchPad?
  - Check at [dev.ti.com](http://dev.ti.com)



User's Guide  
SLAU599–August 2015

## BOOSTXL-EDUMKII Educational BoosterPack™ Mark II Plug-in Module

The [BOOSTXL-EDUMKII](#) BoosterPack™ (see [Figure 1](#)) is an easy-to-use plug-in module that offers a high level of integration for developers to quickly add to LaunchPad™ designs. Various analog and digital inputs/outputs are at your disposal including an analog joystick, environmental and motion sensors, RGB LED, microphone, buzzer, color LCD display, and more.

This BoosterPack was developed with Energia in mind. Energia is an open-source community-developed coding environment, which is supported by a robust framework of intuitive APIs and easy-to-use software libraries for rapid firmware development. TI recommends Energia v12 or later. Learn more about Energia at [www.energia.nu](http://www.energia.nu).

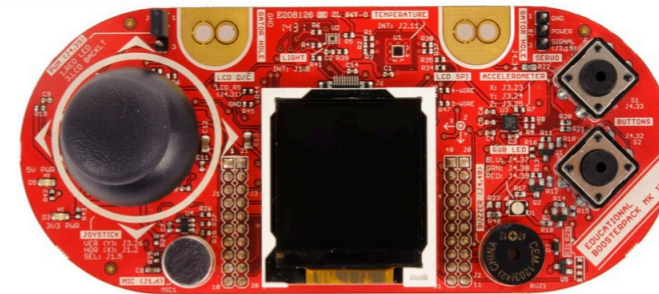


Figure 1. BOOSTXL-EDUMKII BoosterPack

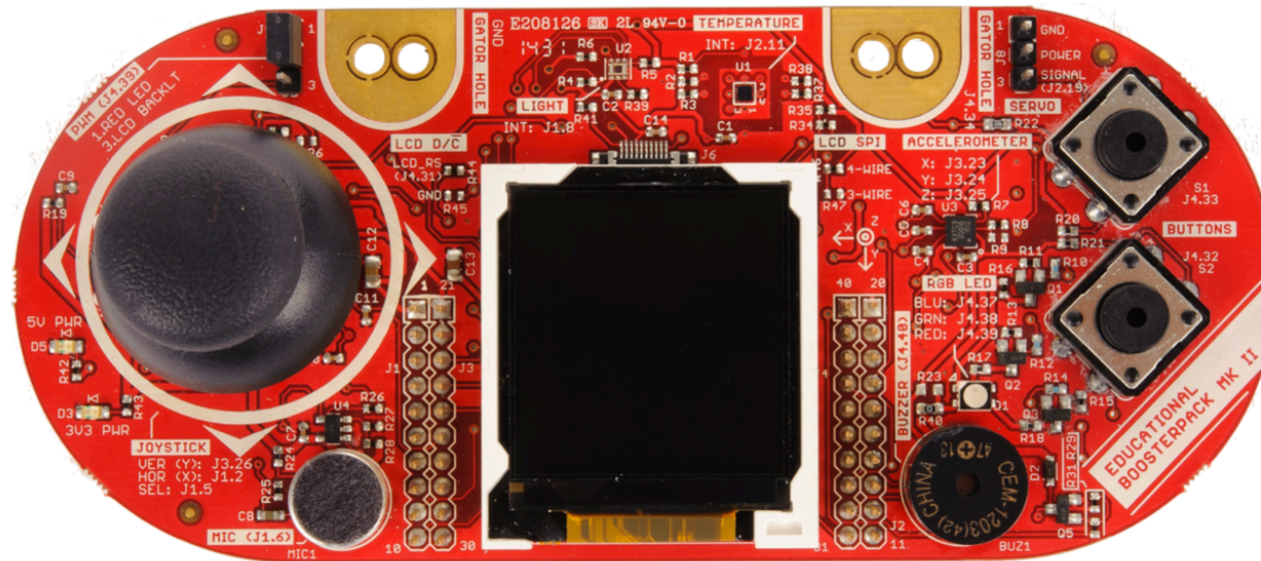
BoosterPack, LaunchPad, Code Composer Studio, E2E are trademarks of Texas Instruments.  
Keil is a trademark of ARM Limited.  
uVision is a registered trademark of ARM Limited.  
IAR Embedded Workbench is a registered trademark of IAR Systems AB.  
All other trademarks are the property of their respective owners.

SLAU599–August 2015  
[Submit Documentation Feedback](#)

BOOSTXL-EDUMKII Educational BoosterPack™ Mark II Plug-in Module  
Copyright © 2015, Texas Instruments Incorporated

# Educational BoosterPack MKII

Sensor		I <sup>2</sup> C
TMP006	thermometer	0x40
OPT3001	luxmeter	0x48



Hardware
Pin number
Other pin
I <sup>2</sup> C
Serial UART
SPI
analogRead()
digitalRead() and digitalWrite()
digitalRead(), digitalWrite() and analogWrite()

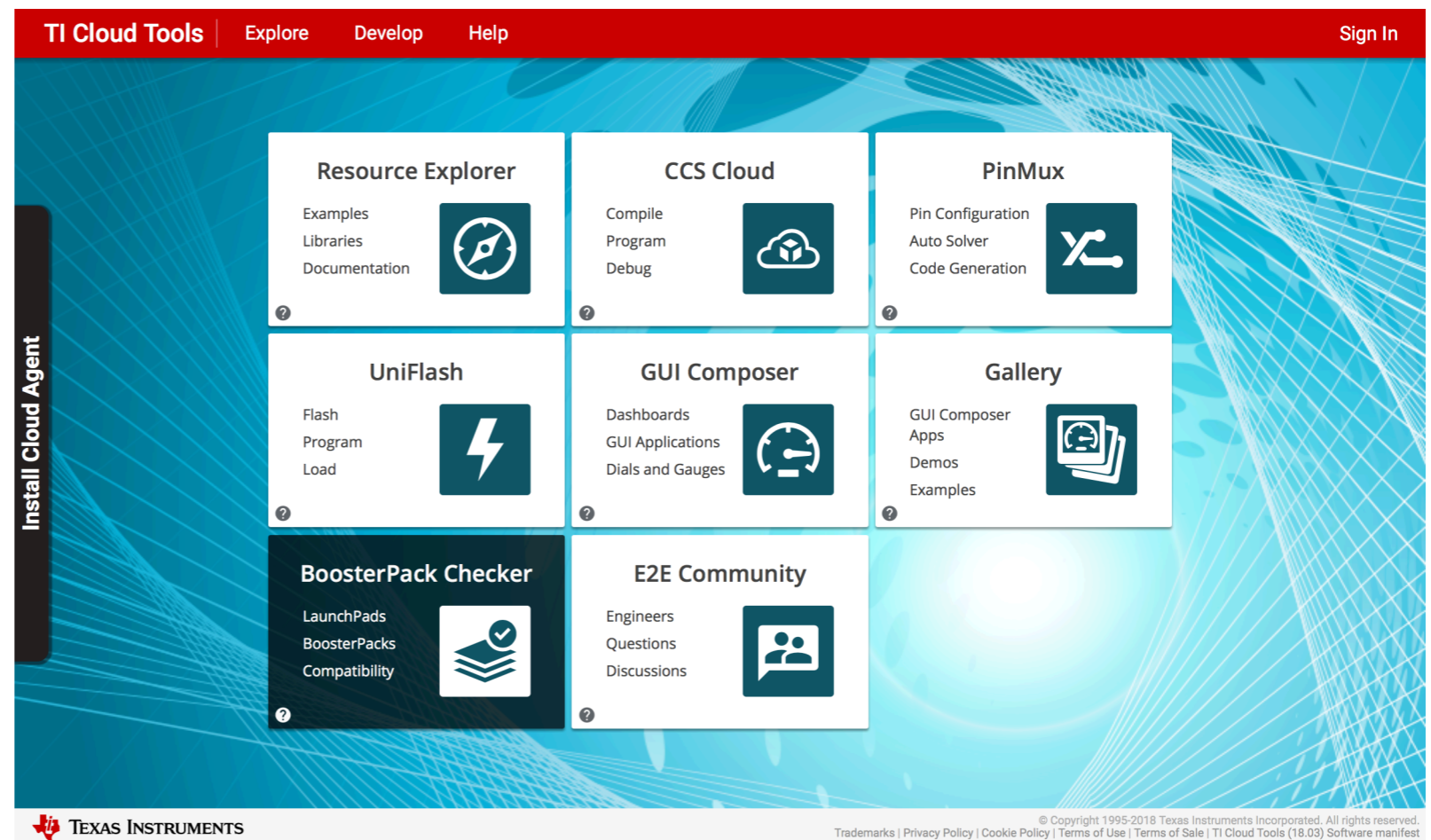
J1	BoosterPack		J3
1	+3.3 V	+5 V	21
2	Joystick X	GROUND	22
3		Acc X	23
4		Acc Y	24
5	Joystick OK	Acc Z	25
6	Microphone	Joystick Y	26
7	LCD SCK		27
8	Light IR		28
9	SCL		29
10	SDA		30

J4	BoosterPack		J2
40	Buzzer	GROUND	20
39	LED red+LCD	Servo PWM	19
38	LED green		18
37	LED blue	LCD RESET	17
36			16
35		LCD MOSI	15
34	Gator		14
33	Button 1	LCD CS	13
32	Button 2		12
31	LCD R/S	Temp IRQ	11



# Check the Compatibility

- Go to [dev.ti.com](http://dev.ti.com)
- Select BoosterPack Checker



# Check the Compatibility







- Select the LaunchPad

Home Library

LaunchPads BoosterPacks

Keyword:

Category: All

	<b>CC3200-LAUNCHXL</b> SimpleLink Wi-Fi CC3200 LaunchPad Connected   <a href="#">More Info</a>
	<b>EK-TM4C123GXL</b> Tiva™ C Series LaunchPad Evaluation Kit Connected   <a href="#">More Info</a>
	<b>EK-TM4C1294XL</b> TM4C1294 Connected LaunchPad Connected   <a href="#">More Info</a>
	<b>LAUNCHXL-CC1310</b> CC1310 LaunchPad Connected   <a href="#">More Info</a>
	<b>LAUNCHXL-CC1350</b> CC1350 LaunchPad Connected   <a href="#">More Info</a>
	<b>LAUNCHXL-CC2640R2</b> CC2640R2F LaunchPad Connected   <a href="#">More Info</a>
	<b>LAUNCHXL-CC2650</b> SimpleLink CC2650 Wireless MCU LaunchPad Connected   <a href="#">More Info</a>
	<b>LAUNCHXL-F28027</b> C2000 Piccolo LaunchPad C2000   <a href="#">More Info</a>
	<b>LAUNCHXL-F28027F</b> InstaSPIN-FOC™ enabled C2000 Piccolo LaunchPad

# Check the Compatibility

- Select the BoosterPack
- Check the compatibility

The screenshot shows the TI E2E website interface. At the top, there are navigation tabs for 'Home' and 'Library'. Below these, there are two sub-tabs: 'LaunchPads' and 'BoosterPacks', with 'BoosterPacks' being the active tab. A search bar contains the keyword 'edu'. Below the search bar, there are filters for 'Category' and 'Maker', both set to 'All'. A checkbox labeled 'Show Community Boards' is checked. The search results are divided into two sections: 'Compatible BoosterPacks' (indicated by a green '2' in a box) and 'Incompatible BoosterPacks' (indicated by a red '1' in a box). The 'Compatible' section lists two items: 'Educational BoosterPack MKI' by CircuitCo and 'Educational BoosterPack MKII' by Texas Instruments. The 'Incompatible' section lists one item: 'EduBase ARM Trainer' by EduBase. At the bottom of the search results, there is a button labeled '+ Add My BoosterPack'.

Compatible? **YES**

Reason: Selected combination is **Fully Compatible!** [\[Show Notes\]](#)

Buy Now

Cloud Tools

Share My Combo

My Selections | [All](#) | [LP Only](#)

BP1



Educational BoosterPack MKII

TI REX

[Buy Now](#)

[More Info](#)

[Remove](#)

+

LP



MSP-EXP432P401R

TI REX

[Buy Now](#)

[More Info](#)

[Remove](#)

▼ Connector #1 (40 Pins)

● 1	● 21	● 40	● 20
● 2	● 22	● 39	● 19
● 3	● 23	● 38	● 18
● 4	● 24	● 37	● 17
● 5	● 25	● 36	● 16
● 6	● 26	● 35	● 15
● 7	● 27	● 34	● 14
● 8	● 28	● 33	● 13
● 9	● 29	● 32	● 12
● 10	● 30	● 31	● 11

Pin Summary

Pin: **#10** > Function: **i2csda** > Instance: **UCB1SDA** > GPIO: **P6.4**

Pin Status

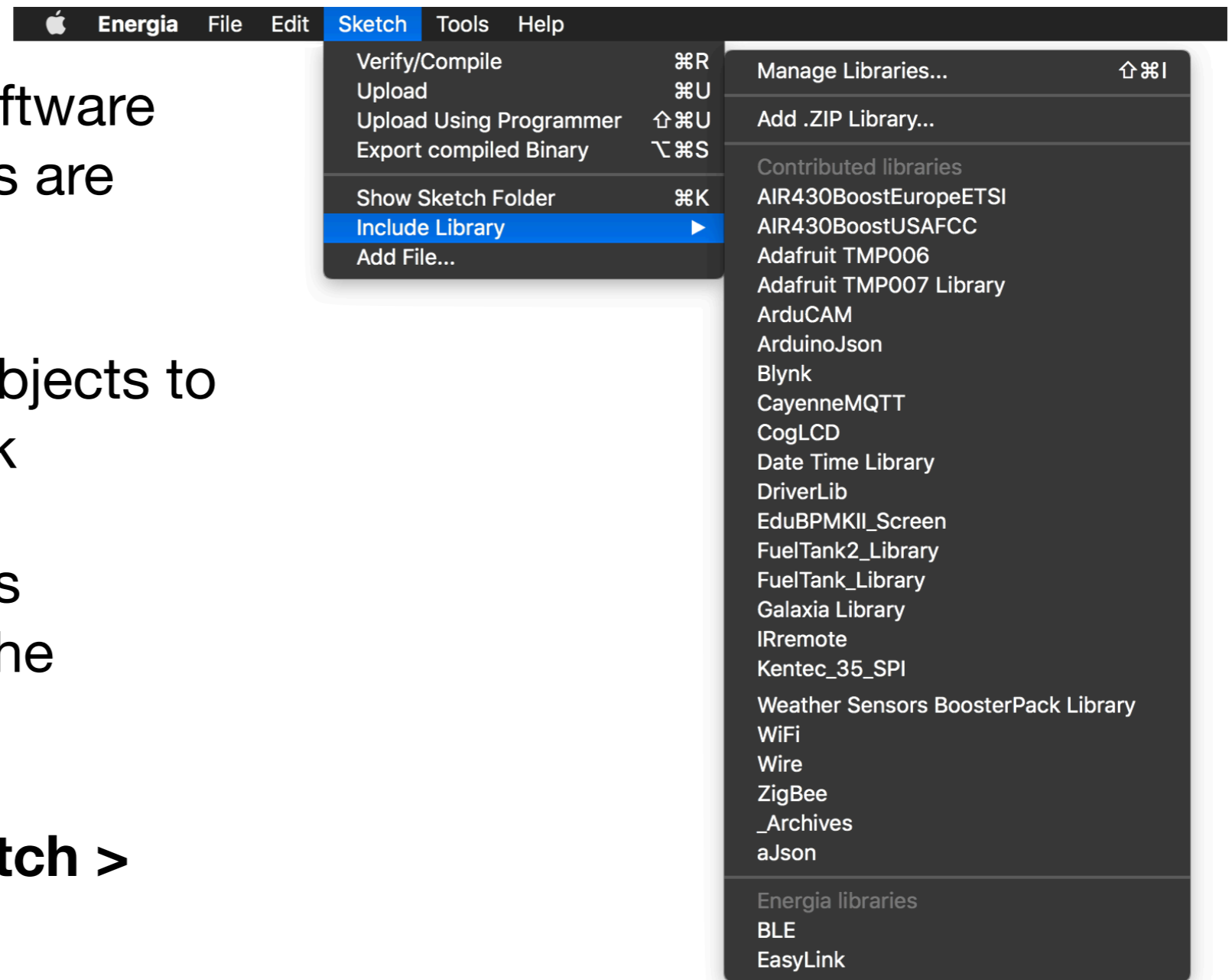
**InUse** **Shareable** - used by BOOSTXL-EDUMKII

Pin Usage

● [BOOSTXL-EDUMKII] > Function: **i2csda** > Instance: **Ambient light and temperature sensor, I2C** > Shareable: **true**

# Libraries

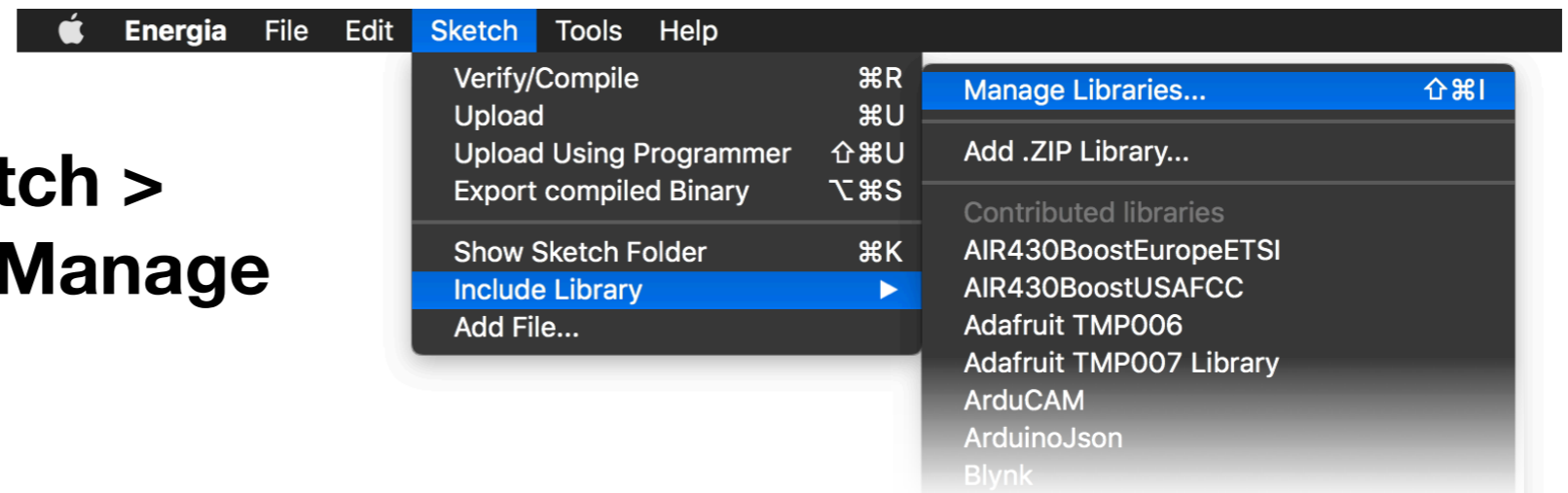
- Libraries are for software what BoosterPacks are for hardware
- The libraries add objects to the core framework
- Identify the libraries corresponding to the BoosterPack
- Call the menu **Sketch > Include Library**



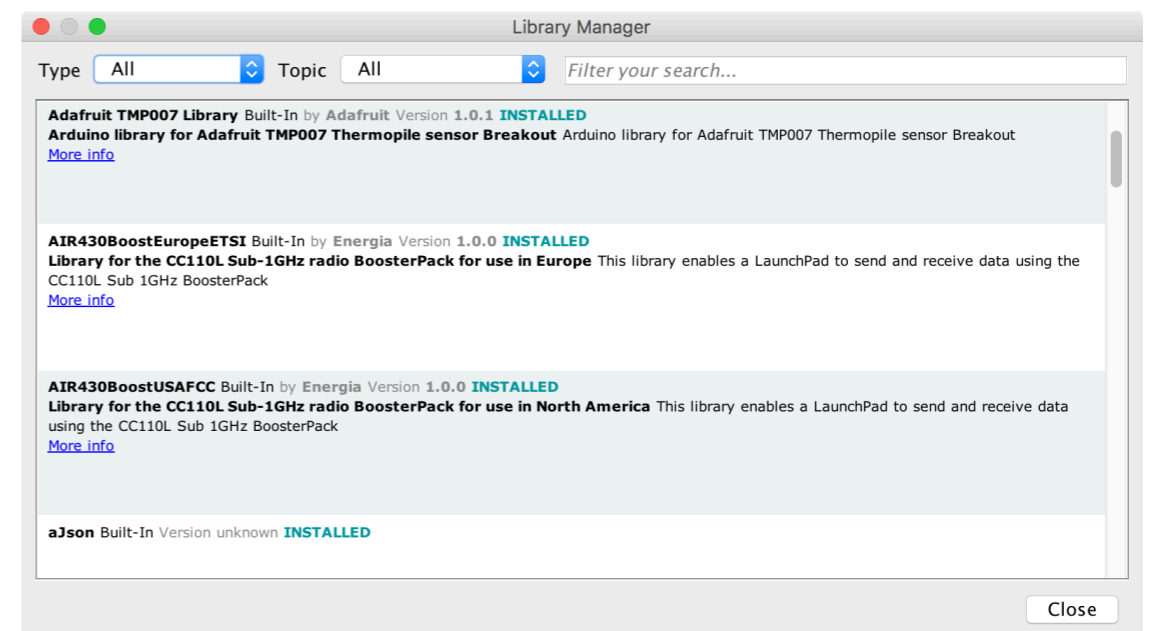


# Add a Library

- Call the menu **Sketch > Include Library > Manage Libraries...**



- Select the library and click **Install**



# Use a Library

- On the code, a library needs to be included
- At the very beginning of the program
  - Add an `#include` statement,
  - With the name of the library, between `<>` or `""`,
  - And no final `;` !

```
78  
79 // Include libraries  
80 #include "Wire.h"  
81
```

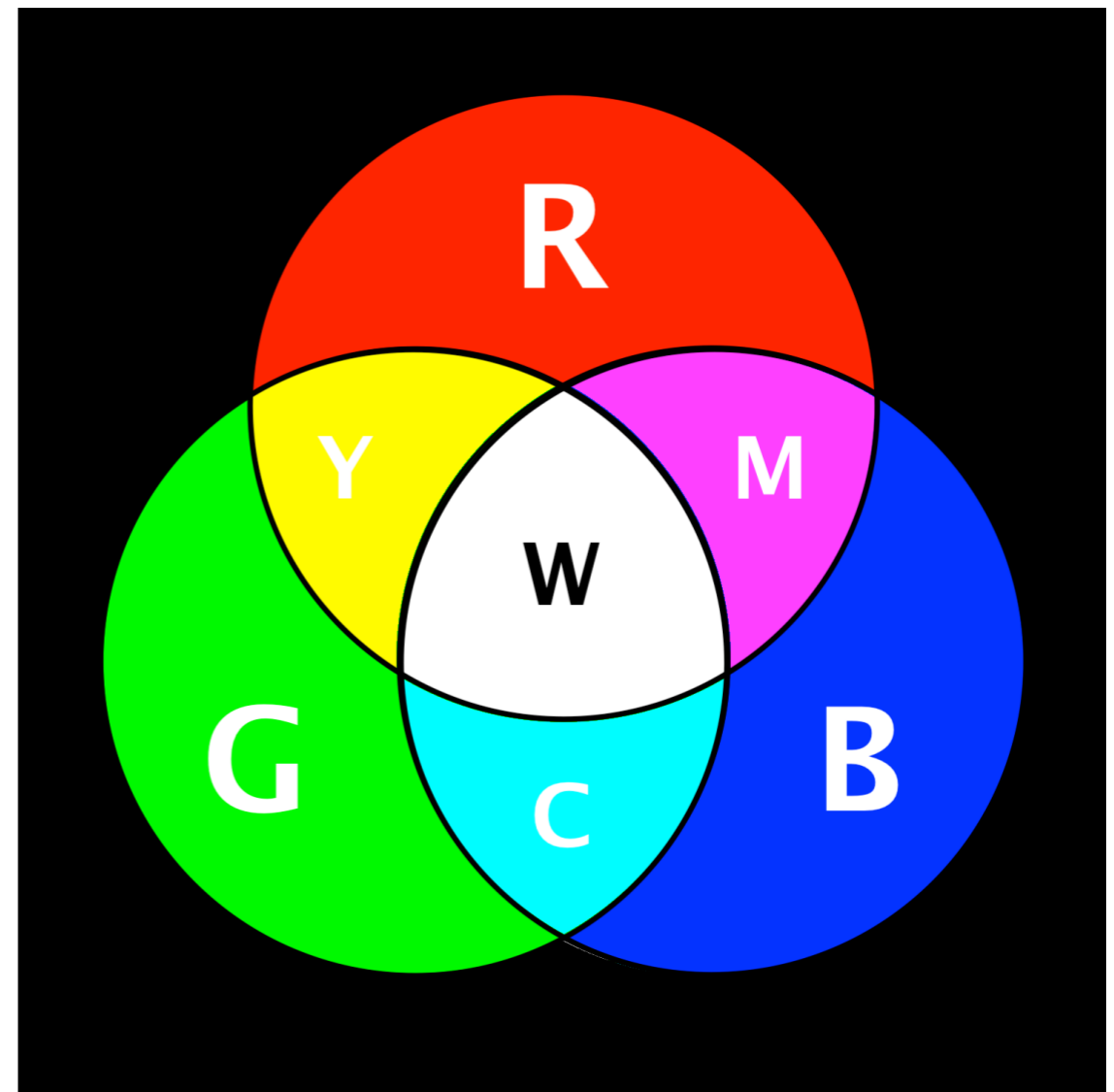
# Install the BoosterPack

- Identify the +3.3V mark
- Plug the BoosterPack onto the LaunchPad



# Display Rainbow Colours

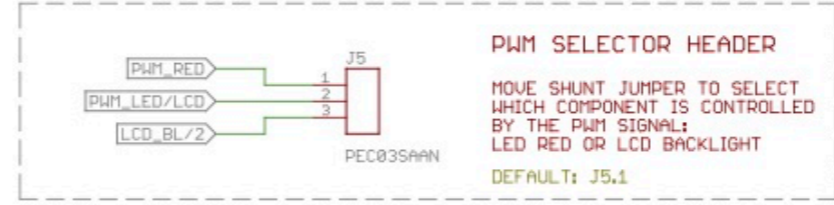
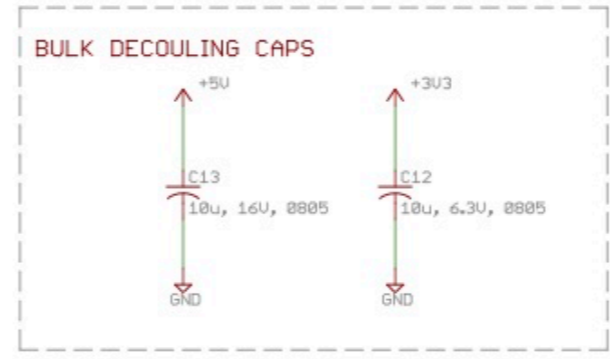
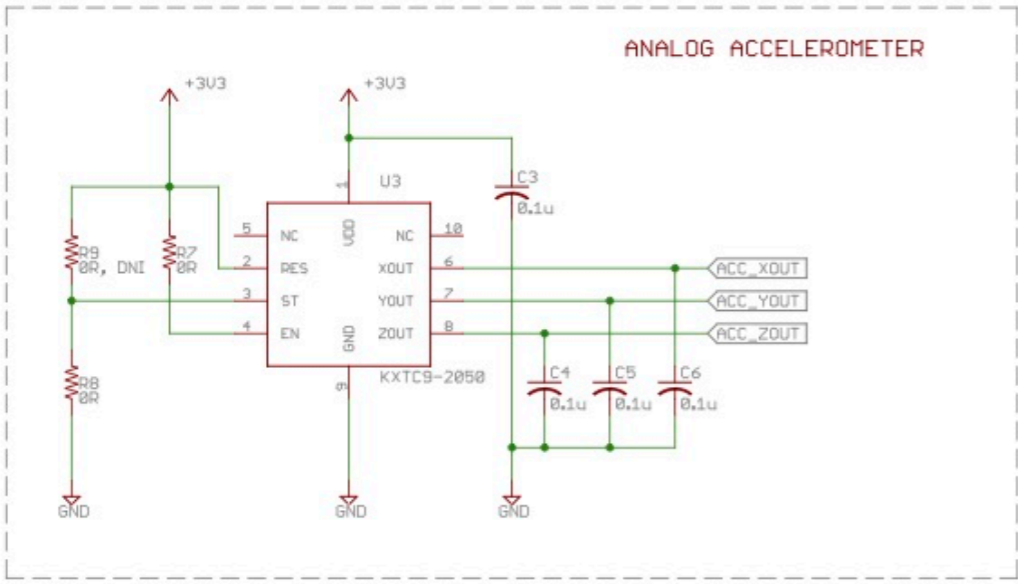
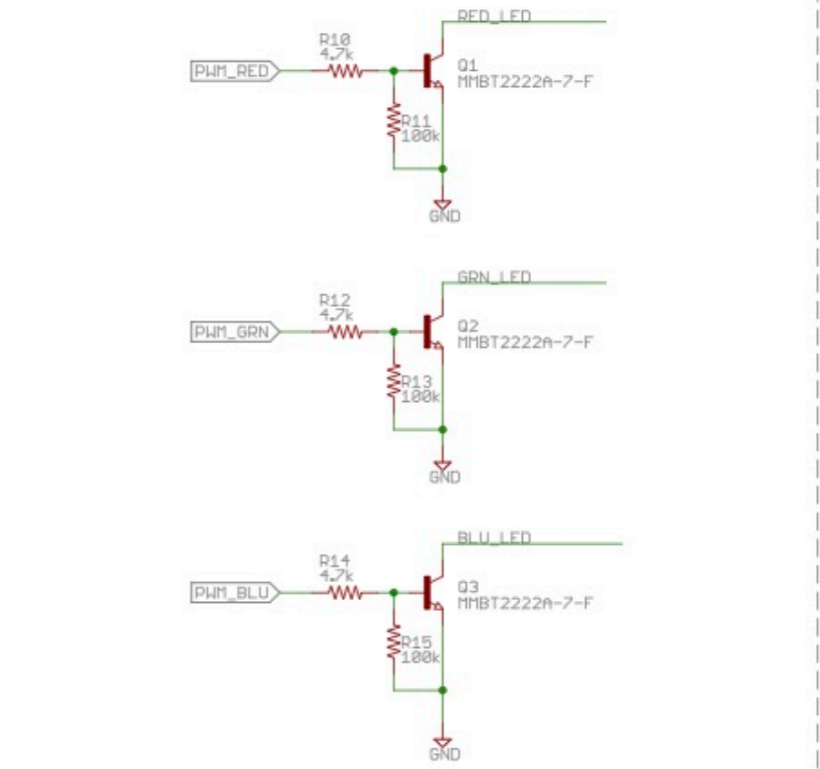
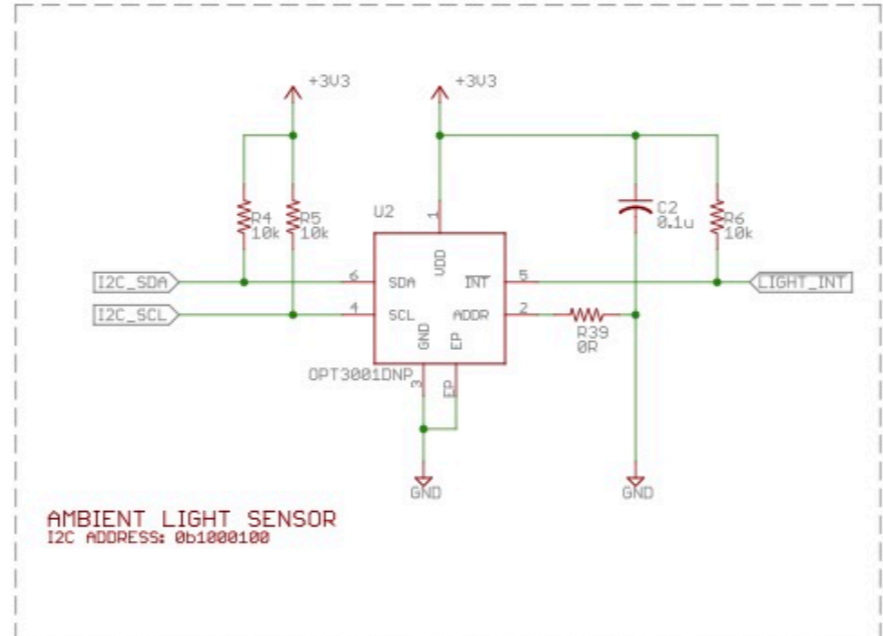
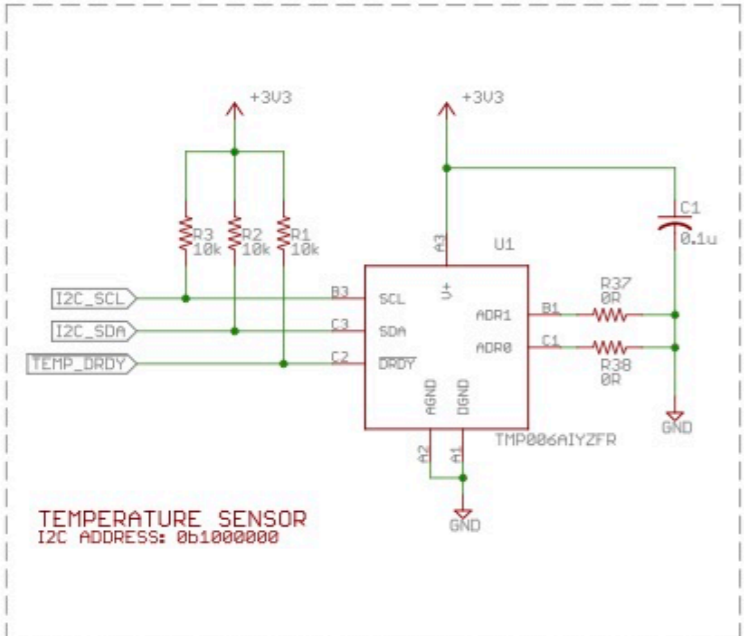
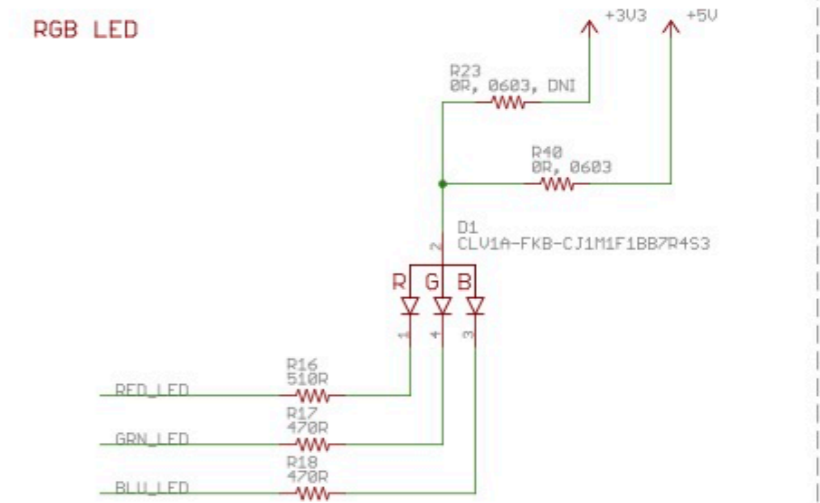
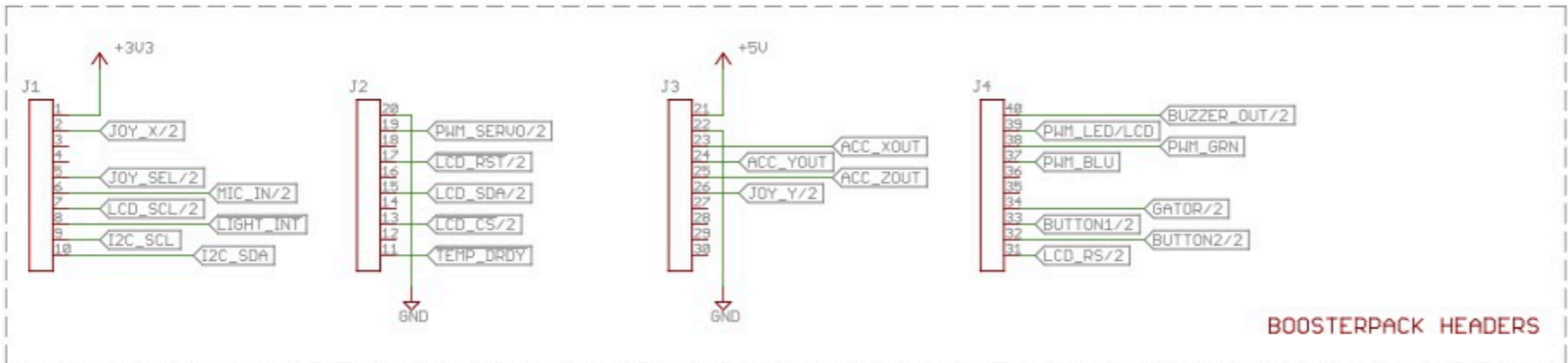
- Display the rainbow colours using the RGB LED
- Hints
  - Find the pins
  - Use PWM
  - Search the reference and examples for `analogWrite()`



# Search Documents

- What documents do we have?
  - Schematics
  - Pins map
  - Data-sheet of the sensors
  - Energia reference
  - Energia examples





TITLE: EDUCATIONAL BOOSTERPACK MK II	
Document Number:	REV: 1.0
Prepared by: Mike Stein	
Date: 7/6/2015 7:22:38 PM	Sheet: 1/2

# Display Rainbow Colours

- Pins are:
  - 39: red LED + LCD
  - 38: green LED
  - 37: blue LED
- Name each pin for later reuse
- Configure the pins

J4	BoosterPack		J2
40	Buzzer	GROUND	20
39	LED red+LCD	Servo PWM	19
38	LED green		18
37	LED blue	LCD RESET	17
36			16
35		LCD MOSI	15
34	Gator		14
33	Button 1	LCD CS	13
32	Button 2		12
31	LCD R/S	Temp IRQ	11

# Display Rainbow Colours

- Use either...
  - `#define redLED 39`
- ...or...
  - `const uint8_t redLED = 39;`
- In the `setup()` function, initialise the pins

```
85 // Define variables and constants
86 #define redLED 39
87 #define greenLED 38
88 #define blueLED 37
89
```

# Display Rainbow Colours

- In the `setup()` function, initialise the pins
  - with `pinMode()` function
- Test the pins separately
  - with `digitalWrite()` function
- Does the red LED work?

```
90 // Add setup code
91 void setup()
92 {
93     pinMode(redLED, OUTPUT);
94     pinMode(greenLED, OUTPUT);
95     pinMode(blueLED, OUTPUT);
96 }
```

# Display Rainbow Colours

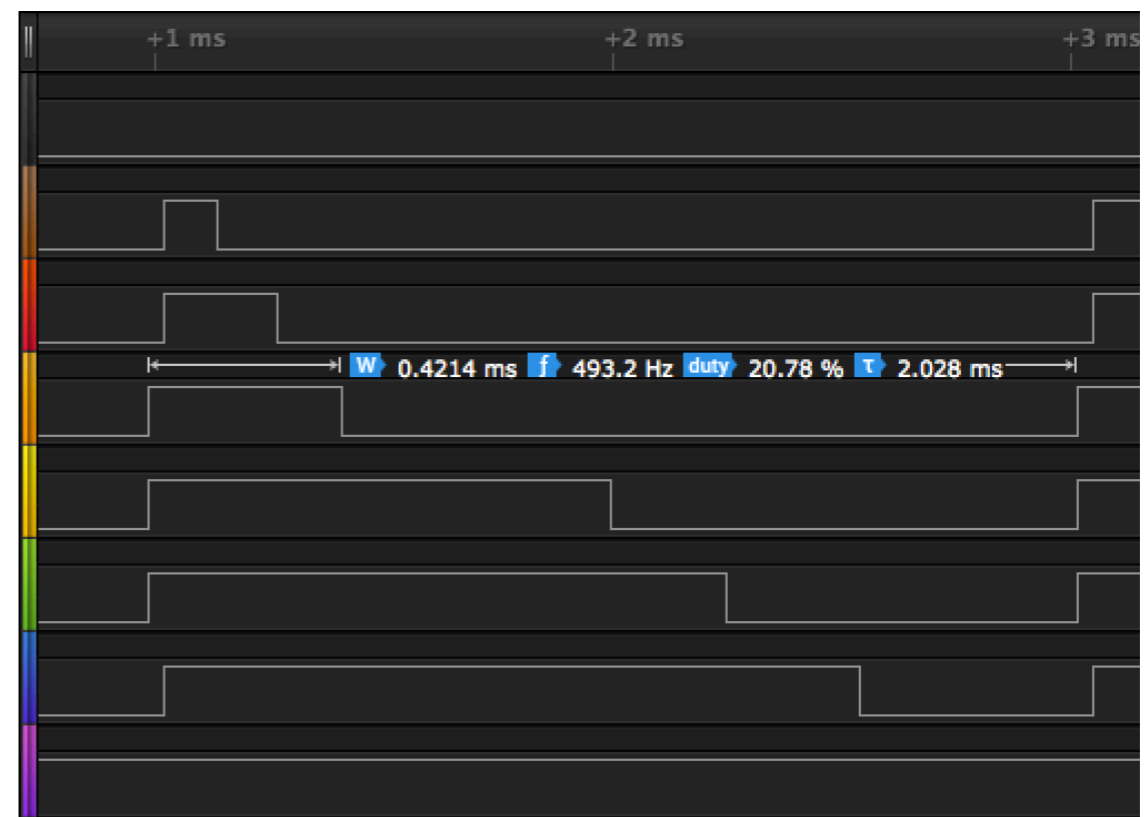
- Does the red LED work?
- If not, check J5 jumper is in position 1. RED LED!





# PWM

- PWM or Pulse Width Modulation
- Duty cycle = period high / total period
- Energia function `analogWrite(pin, value)`
  - `pin`: = pin number
  - `value`: 0..255 for 0..100%



# Display Rainbow Colours

- To get random value for each LED, use `random(min, max)`
- Send the values to the pin with the `analogWrite()` function

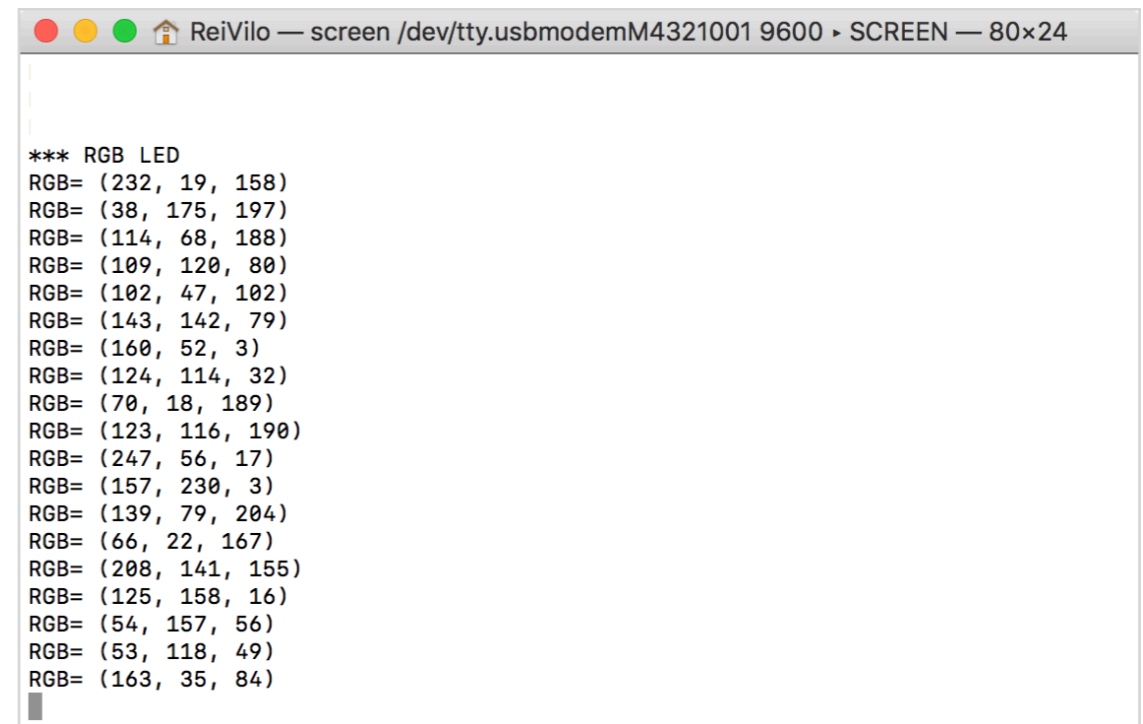
```
101 // Add loop code
102 void loop()
103 {
104     uint8_t redValue = random(0, 255);
105     uint8_t greenValue = random(0, 255);
106     uint8_t blueValue = random(0, 255);
107
108     analogWrite(redLED, redValue);
109     analogWrite(greenLED, greenValue);
110     analogWrite(blueLED, blueValue);
111
112     delay(500);
113 }
```

# Display Rainbow Colours

- Finally, print the values to the console
- Hints:
  - Search the reference and use the examples for `Serial`
  - Two steps: initialisation and use

# Display Rainbow Colours

- Open and check the console
- 24-bit depth colours or 16,777,216 different colours!

A terminal window titled 'ReiVilo — screen /dev/tty.usbmodemM4321001 9600 ▸ SCREEN — 80x24'. The terminal displays a list of RGB values for a rainbow sequence. The text is as follows:

```
*** RGB LED
RGB= (232, 19, 158)
RGB= (38, 175, 197)
RGB= (114, 68, 188)
RGB= (109, 120, 80)
RGB= (102, 47, 102)
RGB= (143, 142, 79)
RGB= (160, 52, 3)
RGB= (124, 114, 32)
RGB= (70, 18, 189)
RGB= (123, 116, 190)
RGB= (247, 56, 17)
RGB= (157, 230, 3)
RGB= (139, 79, 204)
RGB= (66, 22, 167)
RGB= (208, 141, 155)
RGB= (125, 158, 16)
RGB= (54, 157, 56)
RGB= (53, 118, 49)
RGB= (163, 35, 84)
```

# Display Rainbow Colours

- The resulting program

```
85 // Define variables and constants
86 #define redLED 39
87 #define greenLED 38
88 #define blueLED 37
89
90 // Add setup code
91 void setup()
92 {
93     Serial.begin(9600);
94     Serial.println("*** RGB LED");
95
96     pinMode(redLED, OUTPUT);
97     pinMode(greenLED, OUTPUT);
98     pinMode(blueLED, OUTPUT);
99 }
100
101 // Add loop code
102 void loop()
103 {
104     uint8_t redValue = random(0, 255);
105     uint8_t greenValue = random(0, 255);
106     uint8_t blueValue = random(0, 255);
107
108     analogWrite(redLED, redValue);
109     analogWrite(greenLED, greenValue);
110     analogWrite(blueLED, blueValue);
111
112     Serial.print("RGB= (");
113     Serial.print(redValue, DEC);
114     Serial.print(", ");
115     Serial.print(greenValue, DEC);
116     Serial.print(", ");
117     Serial.print(blueValue, DEC);
118     Serial.println(")");
119
120     delay(500);
121 }
```



# Available Help

## Texas Instruments Website [ti.com](http://ti.com)

- Data-sheet of the BoosterPack
- Schematics
- Data-sheet of the sensors
- Compatibility with LaunchPad
- Examples
- Forum at [e2e.ti.com](http://e2e.ti.com)

## Energia Website [energia.nu](http://energia.nu)

- Pins map
- Language reference
- Libraries
- Examples
- Forum at [43oh.com](http://43oh.com)

# Agenda

- **Conclusion – The Energia Ecosystem**
  - To be continued...
  - IDEs
  - General Resources
  - Examples
  - Social

# To Be Continued...

- Use Other Ports and Buses
  - SPI, I<sup>2</sup>C, Slave I<sup>2</sup>C
- Develop Your Own Library
- Debug an Energia project with CCS Cloud
- Low Power Mode
  - Wake-Up, EnergyTrace
- Build Your Own BoosterPack
- IoT
  - sub-1 GHz, Bluetooth, WiFi, Ethernet
- RTOS Extension with Energia Multi-Tasking
  - RTOS Components
  - Galaxia Library
- What Will You Make?

# IDEs

## Official IDEs

- **Energia IDE**  
*cross-platform*  
[energia.nu](http://energia.nu)
- **Code Composer Studio**  
*based on Eclipse*  
*cross-platform*  
[ti.com/ccs](http://ti.com/ccs)
- **CCS Cloud**  
*Chrome add-on*  
[dev.ti.com](http://dev.ti.com)



## Third-Party IDEs

- **Visual Micro**  
*for Visual Studio*  
*Windows only*  
[visualmicro.com](http://visualmicro.com)
- **embedXcode**  
*for Xcode*  
*macOS only*  
[embedXcode.com](http://embedXcode.com)



# IDEs

## Official IDEs

- **Energia IDE**  
*cross-platform*  
[energia.nu](http://energia.nu)



- **Code Composer Studio**  
*based on Eclipse*  
*cross-platform*  
[ti.com/ccs](http://ti.com/ccs)



- **CCS Cloud**  
*Chrome add-on*  
[dev.ti.com](http://dev.ti.com)



## Third-Party IDEs

- **Visual Micro**  
*for Visual Studio*  
*Windows only*  
[visualmicro.com](http://visualmicro.com)



- **embedXcode**  
*for Xcode*  
*macOS only*  
[embedXcode.com](http://embedXcode.com)





# General Resources

## Texas Instruments



- Website [ti.com](http://ti.com)
- LaunchPad and BoosterPack boards [ti.com/launchpad](http://ti.com/launchpad)
- Online tools [dev.ti.com](http://dev.ti.com)
- Resource Explorer [dev.ti.com/tirex](http://dev.ti.com/tirex)
- Compatibility Checker [dev.ti.com/bpchecker](http://dev.ti.com/bpchecker)

## Energia



- Website [energia.nu](http://energia.nu)
- Getting Started [energia.nu/guide](http://energia.nu/guide)
- Reference [energia.nu/reference](http://energia.nu/reference)
- Pins Maps [energia.nu/pins-maps](http://energia.nu/pins-maps)
- GitHub [github.com/energia](http://github.com/energia)

# General Resources

## Texas Instruments



- Website [ti.com](http://ti.com)
- LaunchPad and BoosterPack boards [ti.com/launchpad](http://ti.com/launchpad)
- Online tools [dev.ti.com](http://dev.ti.com)
- Resource Explorer [dev.ti.com/tirex](http://dev.ti.com/tirex)
- Compatibility Checker [dev.ti.com/bpchecker](http://dev.ti.com/bpchecker)

## Energia



- Website [energia.nu](http://energia.nu)
- Getting Started [energia.nu/guide](http://energia.nu/guide)
- Reference [energia.nu/reference](http://energia.nu/reference)
- Pins Maps [energia.nu/pins-maps](http://energia.nu/pins-maps)
- GitHub [github.com/energia](http://github.com/energia)

# dev.ti.com

## Resource Explorer

Examples  
Libraries  
Documentation



## CCS Cloud

Compile  
Program  
Debug



## PinMux

Pin Configuration  
Auto Solver  
Code Generation



## UniFlash

Flash  
Program  
Load



## GUI Composer

Dashboards  
GUI Applications  
Dials and Gauges



## Gallery

GUI Composer  
Apps  
Demos  
Examples



## BoosterPack Checker

LaunchPads  
BoosterPacks  
Compatibility



## E2E Community

Engineers  
Questions  
Discussions



No Device Detected



# dev.ti.com

## Resource Explorer

Examples  
Libraries  
Documentation



## CCS Cloud

Compile  
Program  
Debug



## PinMux

Pin Configuration  
Auto Solver  
Code Generation



## UniFlash

Flash  
Program  
Load



## GUI Composer

Dashboards  
GUI Applications  
Dials and Gauges



## Gallery

GUI Composer  
Apps  
Demos  
Examples



## BoosterPack Checker

LaunchPads  
BoosterPacks  
Compatibility



## E2E Community

Engineers  
Questions  
Discussions



No Device Detected

# energia.nu



## Energia

Follow us on twitter  
[@energiaproject](#)

[Home](#) [Download](#) [Guide](#) [Reference](#) [Blog](#) [Store](#) [Getting Help](#) [IRC](#) [Energia Projects](#) [Events](#) [BYOB](#) [FAQs](#) [Contact Us](#)

## Prototyping Software to Make Things Easy

**11/20/17 MSP-EXP430FR2433 support added through board manager**

The new MSP-EXP430FR2433 is a “replacement” for the value line MSP-EXP430G2. Support for the MSP-EXP430FR2433 has been added to the MSP430 core and made available through the board manager in Energia.

The pinmap for the MSP-EXP430FR2433 is available here: [MSP-EXP430FR2433 pinmap](#)

**So, what is this all about then?**

Energia is an open-source electronics prototyping platform started by Robert Wessels in January of 2012 with the goal to bring the Wiring and Arduino framework to the Texas Instruments MSP430 based LaunchPad. The Energia IDE is cross platform and supported on Mac OS, Windows, and Linux. Energia uses the mspgcc compiler by Peter Bigot and is based on

**[GETTING STARTED GUIDE](#)**

**[Official 430h Energia Forum](#)**

**[Energia Source Code](#)**

**[Energia GitHub Wiki](#)**

**[Energia API References](#)**

**[Energia Libraries](#)**

**[LaunchPad Pin Mapping and Board Setup Instructions:](#)**



# Examples



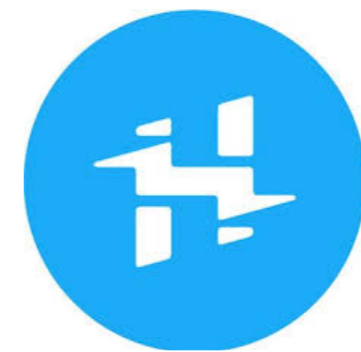
## Texas Instruments

- Gallery  
[dev.ti.com/gallery](http://dev.ti.com/gallery)



## Energia

- Menu  
**File > Examples**



## Third-party

- Hackster  
[hackster.io/  
texasinstruments](http://hackster.io/texasinstruments)

# Social



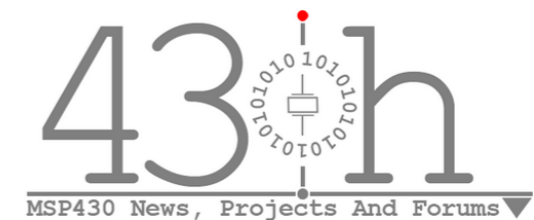
## Social

- [@TXInstruments](https://twitter.com/TXInstruments)
- [@energiaproject](https://twitter.com/energiaproject)



## Forum

- E2E forum  
[e2e.ti.com](https://e2e.ti.com)



## Third-Party

- 43oh!  
[43oh.com](https://43oh.com)

**What Will You Make?**

# Pushing Your Arduino Project to the Next Level with Texas Instruments

Discovering the Energia Ecosystem



**ESIEE**  
PARIS



*May 17, 2018*

