

Advanced Topics

- 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

Joe George, Northeast Digital Field Applications
Texas Instruments
Americas Sales and Marketing

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)

The screenshot shows the Code Composer Studio (CCS) interface for an MSP432 project. The Project Explorer on the left shows the file structure for 'Blink_Joe'. The main editor displays the source code for 'main.cpp', which includes hardware requirements, LED definitions, and a loop routine. The Disassembly window shows the assembly code for the 'loop()' function. The Console window at the bottom shows the output of the debugger, including a warning about breakpoint detection.

```
7 Hardware Required:
8 * LaunchPad with an LED
9
10 This example code is in the public domain.
11 */
12
13 // most launchpads have a red LED
14 #define LED_RED_LED
15
16 //see pins_energia.h for more LED definitions
17 //define LED_GREEN_LED
18
19 // the setup routine runs once when you press reset:
20 void setup() {
21 // initialize the digital pin as an output.
22 pinMode(LED, OUTPUT);
23 }
24
25 // the loop routine runs over and over again forever:
26 void loop() {
27 digitalWrite(LED, HIGH); // turn the LED on (HIGH is the vo
28 delay(1000); // wait for a second
29 digitalWrite(LED, LOW); // turn the LED off by making the v
30 delay(1000); // wait for a second
31 }
32
```

Name	Value	Description
PC	0x000005F4	Program Counter (Core
SP	0x20002AB0	General Purpose Regis
LR	0x00000623	General Purpose Regis
xPSR	0x41000000	Stores the status of inte

```
loop
loop():
000005f4: B508      push
27      digitalWrite(LED, HIGH);
000005f6: 2101      movs
000005f8: 2048      movs
000005fa: F000F965 b1
28      delay(1000);
000005fe: F44F707A mov.w
00000602: F000F9BF b1
29      digitalWrite(LED, LOW);
00000606: 2048      movs
00000608: 2100      movs
0000060a: F000F95D b1
30      delay(1000);
0000060e: F44F707A mov.w
31
00000612: E8BD4008 pop.w
30      delay(1000);
00000616: F00089B5 b.w
44
the_task(unsigned int, unsig
0000061a: B510      push
0000061c: 450C      movs
```

Console Output:

```
MSP432P401R.cxml
CORTEX_M4_0: GEL Output: Memory Map Initialization Complete
CORTEX_M4_0: GEL Output: Halting Watchdog Timer
CORTEX_M4_0: WARNING : On MSP432P401R hitting a breakpoint cannot be detected by the debugger when the device is
Click the pause button during debug to check if the device is held at the breakpoint.
```

The screenshot shows the Energia IDE interface. The main editor displays the source code for 'Blink_Joe.ino'. The terminal window at the bottom shows the upload progress, including the loading of ELF symbols and sorting of symbols.

```
//see pins_energia.h for more LED definitions
//define LED_GREEN_LED

// the setup routine runs once when you press reset:
void setup() {
// initialize the digital pin as an output.
pinMode(LED, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
digitalWrite(LED, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000); // wait for a second
digitalWrite(LED, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}

Done uploading
loading ELF symbols: 93%
sorting and removing duplicate symbols: 100%
Success
```

The screenshot shows a forum post from the TI E2E Forum. The post discusses the issue of debugging MSP432 boards with CCS and provides a workaround for the breakpoint detection warning.

Wed, Jun 14 2017 1:59 PM

Robert Wessels

In reply to J. Martin:

Currently there are two MSP432 packages in the [Energia](#) board manager. One is called "[Energia MSP432 boards](#)" and the other is called "[Energia EMT RED MSP432 boards](#)". The "[Energia MSP432 boards](#)" is the legacy packages that supports both RED and BLACK boards. The "[Energia EMT RED MSP432 boards](#)", which only supports the RED boards as the name implies currently has some you have this package installed, then the work-around is to uninstall the "[Energia EMT RED MSP432 boards](#)" and install the "[Energia MSP432 boards](#)".

Debug Energia with CCS (Assembly)



MSP432_SimpleLinkSDK - CCS Debug - C:\vimages\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

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 0x000006C
 - 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 to this)

Registers

Name	Value	Description
PC	0x0000467E	Program Counter
SP	0x20002740	General Purpose Register 15
LR	0x00005B03	General Purpose Register 14
xSPR	0x41000000	Stores the status of the processor
R0	0x40004C03	General Purpose Register 0
R1	0x40000201	General Purpose Register 1
R2	0x000000B4	General Purpose Register 2
R3	0x20001E88	General Purpose Register 3
R4	0x20001190	General Purpose Register 4
R5	0x20001E88	General Purpose Register 5
R6	0x000000B4	General Purpose Register 6
R7	0x007F7F7F	General Purpose Register 7
R8	0x00000032	General Purpose Register 8
R9	0xFFFFFFFF	General Purpose Register 9
R10	0xFFFFFFFF	General Purpose Register 10

Disassembly

```
loop
0000466e: 88A6      ldrh   r6, [r4, #4]
1284      pinMode(RED_LED, OUTPUT);
00004670: 204B      movs   r0, #0x4b
00004672: 2101      movs   r1, #1
00004674: F00F9E0  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: 2700      movs   r7, #0xd
0000469e: 3F01      subs   r7, #1
000046a0: F47FAFFD bne.w  #0x469e
000046a4: F00B800B b.w    #0x46be
000046a8: 7001      strb   r1, [r0]
000046aa: 2700      movs   r7, #0xd
```

Console

```
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
```