

Energia dla zestawu startowego MSP430 LaunchPad Value Line

MSP430 to rodzina mikrokontrolerów o bardzo małym poborze mocy zaprojektowana i produkowana przez firmę Texas Instruments. Najtańszy i bardzo popularny zestaw startowy MSP430 LaunchPad Value Line Development kit (MSP-EXP430G2) może być uzupełniony o darmowe środowisko programowe *Energia* zgodne z *Arduino*. Wtedy w bardzo łatwy sposób można zacząć pracować z programami przeznaczonymi do wykonywania w czasie rzeczywistym.

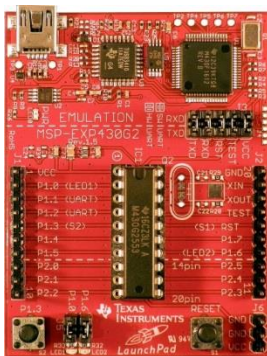
Zestaw startowy MSP430 LaunchPad Value Line

Zestaw startowy MSP430 LaunchPad Value Line Development kit (MSP-EXP430G2) jest kompletną płytką do tworzenia i programowania systemów z procesorem MSP430G2553 (lub MSP430G2231) z rodziny MSP430 [3]. Więcej informacji o rodzinie MSP430 jak i o programowaniu i narzędziach sprzętowych można znaleźć na stronie Getting Started [11].

Płytkę zestawu składa się z dwóch części: górna to programator/emulator a w dolnej jest układ procesorowy z wyprowadzeniami [1].

Górną część płytki (rys.1), aż do linii złącza JP3 zajmuje układ programatora/emulatora sprzętowego. Jest tam zamontowane gniazdko standardu USB Mini-B oraz dioda LED sygnalizacji zasilania. Układ scalony TUSB3410 pracuje jako konwerter USB-JTAG/UART. Układ udostępnia na dołączonym komputerze dwa porty: jeden do programowania/emulacji i drugi (MSP430 Application UART) jako wirtualny port VCOM do komunikacji.

W dolnej części płytki jest układ procesorowy z wyprowadzeniami oraz przyciski i diody LED. Wszystkie wyprowadzenia układu procesorowego są udostępnione na dwu złączach J1 i J2 po bokach płytki drukowanej zestawu. W 20 pozycyjną podstawkę można włożyć 14 wyprowadzeniowe procesory serii MSP320G22x1 lub 20 wyprowadzeniowe procesory serii MSP430G25x3.



Rys. 1 Zestaw startowy MSP430 LaunchPad Value Line [3]

Czerwona dioda LED2 jest dołączona do wyprowadzenia P1.6 procesora. Zielona dioda LED1 jest dołączona do wyprowadzenia P1.0 procesora.

Przycisk użytkownika S2 jest dołączony do wyprowadzenia P1.3 procesora. Przycisk RESET S1 jest dołączony do wyprowadzenia RST/SBWTIO procesora.

Program demonstracyjny

Po pierwszym dołączeniu zestaw startowy kablem USB do wolnego portu komputera PC zostaną zainstalowane sterowniki systemu Windows. Sterowniki są automatycznie pobierane z sieci Internet. Można je również pobrać i zainstalować ze strony TI [7].

Port USB dostarcza do zasilania do płytki i zaczyna pracować przykładowy program demonstracyjny wpisany do pamięci Flash procesora [4]. Naprzemiennie świecą się diody zielona LED1 i czerwona LED2.

Dodatkowo program przykładowy wysyła informacje poprzez port szeregowy układu procesorowego typu UART. Aby je wykorzystać należy najpierw zidentyfikować numer wirtualnego portu COM. W tym celu należy kliknąć prawym klawiszem myszy na *Mój komputer* (np. w menu Start). Wybrać *Właściwości* a następnie *Sprzęt* oraz *Menedżer urządzeń*. Na liście *Porty (COM i LPT)* należy znaleźć port o nazwie *MSP430 Application UART (COMxx)*, gdzie xx jest numerem.

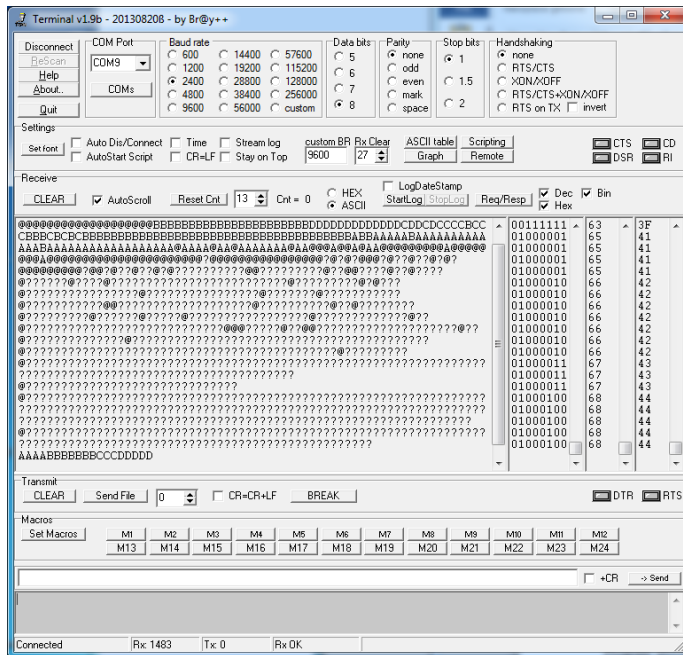
Program przykładowy był uruchamiany z obsługą komunikacji na PC poprzez dedykowany program. Jednak próba jego uruchomienia nie powiodła się. Brakuje jakiś elementów środowiska Java.

Dlatego dobrym pomysłem jest użycie darmowego programu Terminal (by Br@y+) (do pobrania darmowo ze strony [12]). Po uruchomieniu programu Terminal kliknij na przycisk *ReScan* a następnie w polu *COM Port* wybierz z listy poprawny numer portu COM. Ustaw parametry transmisji. Poprawna praca wymaga ustawienia parametrów komunikacji 2400 8N1. Kliknij na przycisk *Connect* (rys.2). Na razie w oknie *Receive* programu Terminal jest pusto.

Po przyknięciu (i przytrzymaniu) na przycisk S2 (lewy) zestawu startowego program przechodzi do pomiaru wewnętrznej temperatury układu procesorowego.

Mierzona jest aktualna temperatura wewnętrzna układu procesorowego i zapisywana jako temperatura odniesienia Tref. Program przechodzi do trybu cyklicznego pomiaru temperatury bieżącej Tcur.

Jasność świecenia diod LED jest regulowana w zależności od wartości różnicy temperatury bieżącej i odniesienia: Tcur-Tref. Gdy bieżąca temperatura jest większa od referencyjnej to świeci dioda czerwona, gdy jest niższa to świeci dioda zielona.



Rys.2 Transmisja pomiarów temperatury

Początkowo wysyłana jest wartość binarna odpowiadająca znakowi '?' w kodzie ASCII. W miarę zmiany temperatury wysyłane są znaki o wyższej lub niższej pozycji w tabeli kodu.

W celu poprawienia czytelności stanu transmisji należy zaznaczyć (kliknąć) pola *Dec*, *Hex* oraz *Bin*. Wtedy w głównym oknie *Receive* pokazywana jest reprezentacja ASCII transmitowanych słów a po prawej stronie pokazywana jest ich reprezentacja binarna, dziesiętna i Hex.

Projekt z kodem źródłowym program demonstracyjny (MSP-EXP430G2-LaunchPad) dla środowiska CCS oraz IAR jest dostępny na stronie producenta w pakiecie programowym *MSP-EXP430G2 Software Examples* [5]. Jest również dostępny pełny pakiet oprogramowania dla procesorów MSP430 o nazwie *MSP430ware* instalowany razem ze środowiskiem CCS lub jako osobny folder [6].

Środowisko programowe Energia

Energia jest otwartym środowiskiem programowym (dostępnym pod adresem [9]) przeznaczonym dla zestawów startowych LaunchPad firmy Texas Instruments i procesorów rodzin MSP430, TM4C, C2000, and CC3200.

Energia bazuje na oryginalnej platformie Wiring i Arduino z zastosowaniem kompilatora MSPGCC. Arduino to platforma programistyczna dla systemów wbudowanych oparta na prostym projekcie przeznaczonym dla mikrokontrolerów montowanych w pojedynczym obwodzie drukowanym, z wbudowaną obsługą wejścia/wyjścia oraz standaryzowanym językiem programowania.

Energia realizuje filozofię nauki poprzez działanie i umożliwia realizowanie tego bezpośrednio z użyciem modułów sprzętowych [2]. Jest dostępna prezentacja prostego warsztatu dla środowiska Energia z zestawem MSP-EXP430G2-LaunchPad [8] oraz bardzo ciekawy projekt z LED [13].

Instalowanie

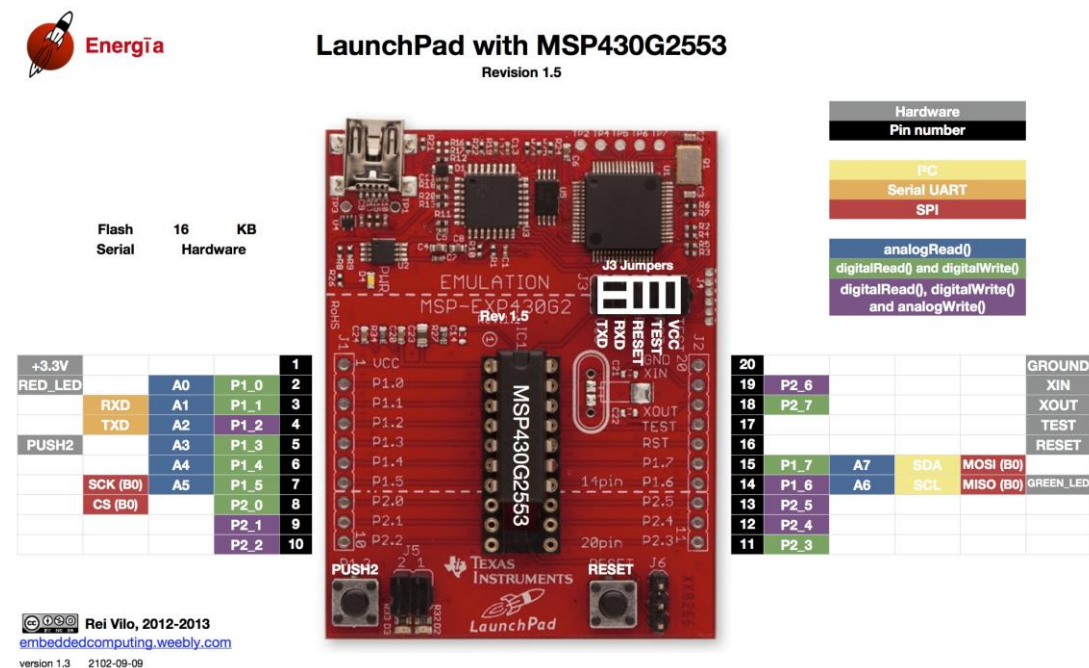
Z podstrony [/guide/windows/](#) pobierz aktualne sterowniki dla odpowiednich zestawów startowych. Dla zestawu MSP-EXP430G2 LaunchPad oraz dla systemu Windows jest to plik EZ430-UART.zip (ok. 600 kB). Po rozpakowaniu należy uruchomić program DPinst.exe dla Windows 32 bit lub DPinst64.exe dla Windows 64 bit. Jeśli na komputerze jest zainstalowane środowisko CCS z obsługą procesorów MSP430 to nie trzeba instalować dodatkowych sterowników.

Z podstrony strony [/download/](#) pobierz plik najnowszej wersji środowiska Energia Obecnie (11.2014) jest to dla systemu Windows plik energia-0101E0013-windows.zip wersji *Energia 0101E0013* (09/05/2014). Po rozpakowaniu należy folder \energia-0101E0013 umieścić w dowolnym miejscu.

Wydaje się, że najlepiej umieścić go w ścieżce C:\ti gdzie jest standardowo instalowane oprogramowanie narzędziowe firmy Texas Instruments.

Projekty zdefiniowane w środowisku Energia można również importować do środowiska CCS.

Na stronie [10] zamieszczone są informacje o rozmieszczeniu i użyciu wyprowadzeń procesora na zestawie startowym MSP430 LaunchPad (MSP-EXP430G2). Rozmieszczenie wyprowadzeń z podziałem na funkcje do nich przypisane jest pokazane na rys.3. Udostępnione są też opisy dla innych zestawów startowych i modułów rozszerzeń.



Rys. 3 Rozmieszczenie wyprowadzeń procesora [10]

Programowanie

Projekty (sketches) środowiska Energia bazują na języku programowania C/C++ w wariacie zdefiniowanym przez Wiring. Opis elementów języka programowania jest zamieszczony na podstronie [/reference/](#). Elementów języka jest zaskakująco niewiele. Siła środowiska tkwi w bardzo rozbudowanych bibliotekach opisanych na podstronie [/guide/libraries/](#). Środowisko jest bardzo intensywnie rozbudowywane, ostatnio o obsługę procesorów rodziny C2000.

Struktura programu jest stała i składa się z dwóch obowiązkowych funkcji.

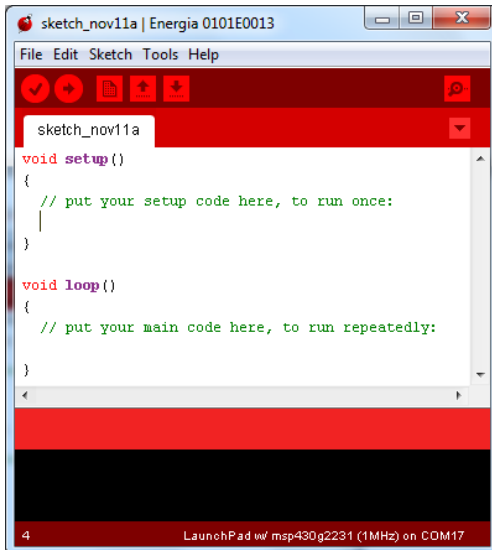
Funkcja setup() jest jeden raz wywoływana po rozpoczęciu wykonywania programu. W niej wykonywana jest inicjalizacja zmiennych i bibliotek, definiowanie trybu wyprowadzeń itd.

Następnie wywoływana jest funkcja loop(). Funkcja ta jest wykonywana w pętli nieskończonej.

Funkcja zawiera podstawowy kod obsługi zestawu sprzętowego.

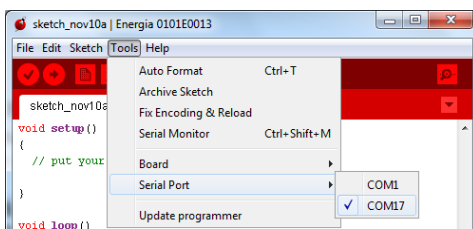
Pierwszy program w środowisku Energia

1. Dołącz zestaw startowy kablem USB do wolnego portu komputera PC. Oczekaj aż zostaną zainstalowane sterowniki systemu Windows.
2. Wystartuj program energia.exe. Wyświetlane jest okno środowiska Energia z pustym projektem (rys.4).



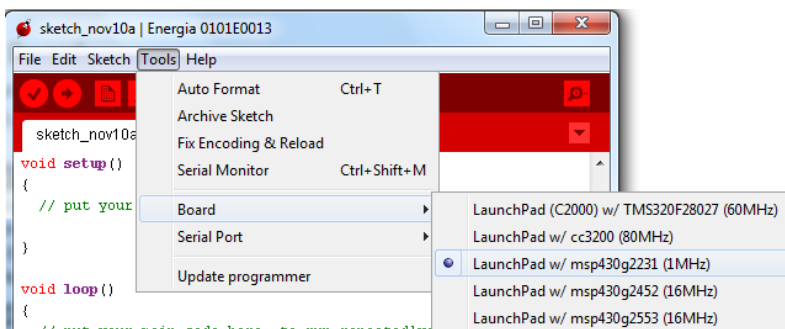
Rys. 4 Pierwsze okno środowiska Energia

3. Z menu File → Preferences otwórz okno Preferences. W polu *Sketchbook location* zmień ścieżkę do foldera roboczego projektów c:\Temp.
4. Z menu Tools → Serial Port wybierz port komunikacyjny przypisany do zestawu (rys.5). Typowo ma on najwyższy numer. Sprawdzenie numeru portu można wykonać w sposób omówiony przy opisie programu demonstracyjnego.



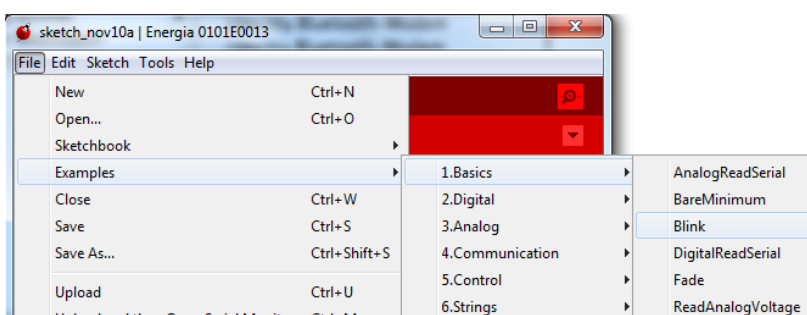
Rys. 5 Wybór portu komunikacyjnego

5. Z menu Tools → Board wybierz odpowiedni zestaw LanuchPad (rys.6).



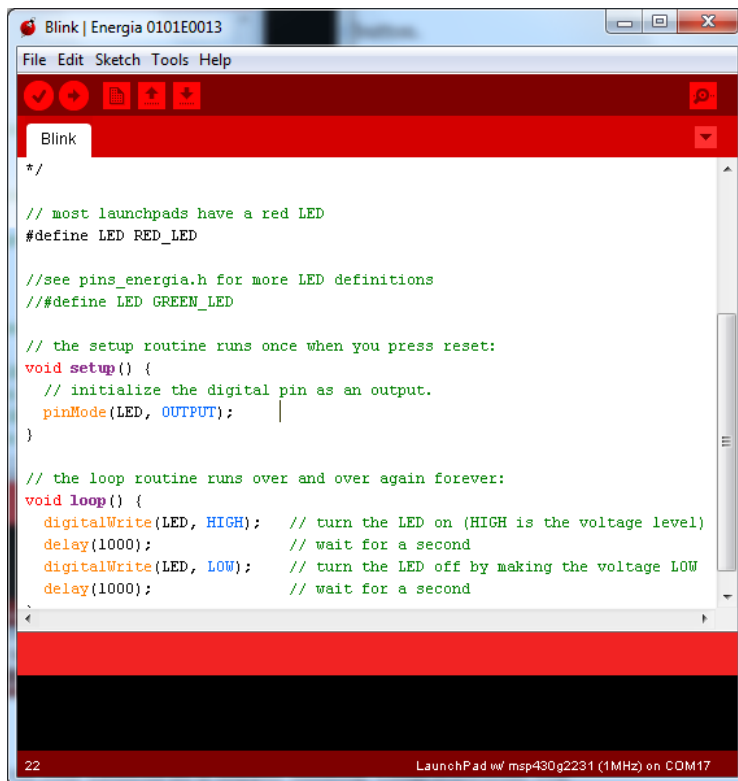
Rys. 6 Wybór zestawu LanuchPad

6. Z menu File → Examples wybierz odpowiedni projekt przykładowy (rys.7). Na początek dobrym projektem jest *Blink* z grupy *1.Basics*.



Rys. 7 Wybór projektu przykładowego

Otwierane jest kolejne okno aplikacji Energia z nazwą projektu (Blink) na pasku tytułu (rys.8).



Rys. 8 Okno projektu *Blink*.

Poniżej paska menu znajduje się pasek narzędzi. Ikony na pasku narzędzi służą do wykonywania podstawowych operacji:



Verify/Compile – Wykonaj kompilację projektu i sprawdź błędy. Tą samą operację można wywołać z menu Sketch → Verify/Compile.



Upload – Wykonaj kompilację projektu i załadowanie (upload) kodu do dołączonego modułu sprzętowego. Moduł sprzętowy jest resetowany i kod jest ładowany. Następnie sygnalizowane jest powodzenie operacji lub podawany jest rodzaj błędu.



New – Utwórz nowy projekt (sketch)



Open – Otwierana jest lista aktywnych projektów. Po wybraniu projektu jest on ładowany do bieżącego okna.



Save – Zapisz bieżący projekt.



Serial Monitor – Otwórz okno komunikacji szeregowej dotyczącej wybranego wcześniej portu VCOM. Tą samą operację można wywołać z menu Tools → Serial Monitor.

Poniżej paska narzędzi znajduje się duże okno edycji. Dalej znajduje się czerwony pasek stanu. Na nim wyświetlane są aktualne informacje np. o pomyślnym zakończeniu kompilacji („Done compiling”).

Kolejne okno debugowania zawiera informacje dotyczące wykonywanych operacji. Po zakończeniu kompilacji pokazywany jest rozmiar wygenerowanego kodu i rozmiar dostępnej pamięci. Informacje o błędach wyświetlane są w kolorze czerwonym.

Na samym dole znajduje się pasek stanu dołączonego sprzętu na którym wyświetlana jest nazwa modułu sprzętowego i nazwa udostępnionego portu komunikacyjnego.

7. Wykonaj kompilację projektu i załadowanie (upload) kodu (rys.9).

Bezpośrednio po załadowaniu kodu jest on uruchamiany. Zielona dioda LED błyska powoli.

8. Zapisz aktualny projekt do pliku. Z menu *File* → *Save as* otwórz okno *Save sketch folder as* i wpisz zmienioną nazwę pliku.

```

Done uploading.
[C:\ti\energia-0101E0013\hardware\tools\msp430\bin\msp430-objcopy, -O, ihex, -R, .eeprom,
C:\Users\hkw\AppData\Local\Temp\build5789903031320445833.tmp\Blink_01.cpp.elf,
C:\Users\hkw\AppData\Local\Temp\build5789903031320445833.tmp\Blink_01.cpp.hex]
Binary sketch size: 783 bytes (of a 2 048 byte maximum)
02
Erasing...
Programming...
Done, 784 bytes total

```

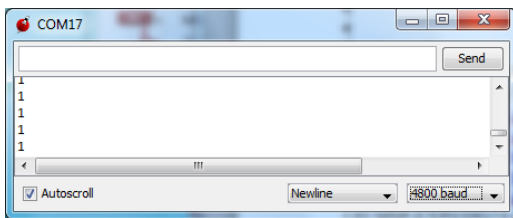
Rys. 9 Okno debugowania projektu *Blink* Po załadowaniu kodu.

9. Teraz można swobodnie modyfikować program. Np. zamiast diody zielonej zastosuj diodę czerwoną zestawu startowego.

Inne projekty przykładowe

Spróbuj uruchomić projekt przykładowy *Fade* (ściemnianie).

10. Załaduj program przykładowy *Fade*. Wykonaj kompilację projektu, załadowanie i uruchomienie kodu. Zaobserwuj sposób świecenia diody LED.
11. Załaduj program przykładowy *DigitalReadSerial*. Wykonaj kompilację projektu, załadowanie i uruchomienie kodu.
12. Otwórz okno komunikacji szeregowej środowiska Energia (np. z menu *Tools* → *Serial Monitor*).



Rys. 10 Okno komunikacji dla projektu *DigitalReadSerial*.

13. Ustaw szybkość transmisji odpowiednią dla używanego modułu procesorowego: 9600 baud dla MSP430G2553 lub 4800 baud dla MSP430G2231 (rys.10).
14. Przyciśnij przycisk użytkownika S2 i zobacz zmianę wyświetlanego stanu cyfrowego z 1 na 0.

Problemy debugowania

Nie powiodła się próba uruchomienia projektu *DigitalReadSerial* w środowisku Energia.

Niestety, środowisko Energia do debugowania udostępnia tylko wydruki kontrolne. Dla tego przypadku jest to zupełnie nieprzydatne. Błąd występuje w bibliotece obsługującej właśnie wysyłanie znaków do komputera PC.

Jednak najnowsza wersja środowiska Code Composer Studio 6.0.1 umożliwia importowanie projektów środowiska Energia. W ten sposób powstaje normalny projekt CCS i możliwe jest jego pełne debugowanie.

Bibliografia

- [1] Ergooszczędne MSP430, Nowa seria G2xxx, [Elektronika Praktyczna](#), 9/2010
- [2] Energia - "Arduino" dla Launchpada, [Elektronika Praktyczna](#), 12/2012
- [3] [MSP-EXP430G2 LaunchPad Evaluation Kit User's Guide \(Rev. E\)](#) 06 Mar 2014 SLAU318E
- [4] [MSP-EXP430G2 LaunchPad Quick Start Guide \(Rev. A\)](#) 19 Dec 2011 SLAC432A
- [5] [MSP-EXP430G2 Software Examples \(Rev. E\)](#) ZIP 20 Jun 2014 SLAC435E
- [6] <http://www.ti.com/tool/msp430ware> MSP430ware 1.95.00.32 Standalone
- [7] [MSP-EXP430G2 LaunchPad Driver](#) 23 May 2013 SLAC524
- [8] MSP-EXP430G2 LaunchPad Workshop, LaunchPadSimpleProject.pdf [link](#)
- [9] Energia <http://energia.nu/>
- [10] Guide to the MSP430 LaunchPad (MSP-EXP430G2) http://energia.nu/pin-maps/guide_msp430g2launchpad/
- [11] MSP Microcontrollers, Getting Started http://www.ti.com/lscs/ti/microcontrollers_16-bit_32-bit/msp/getting-started.page
- [12] Terminal, com port development tool <https://sites.google.com/site/terminalbpp/>
- [13] Blinking LED on MSP430 LaunchPad- Just in Time for Halloween http://e2e.ti.com/blogs_/b/designproject/archive/2012/10/12/blinking-led-on-msp430-launchpad-just-in-time-for-halloween.aspx