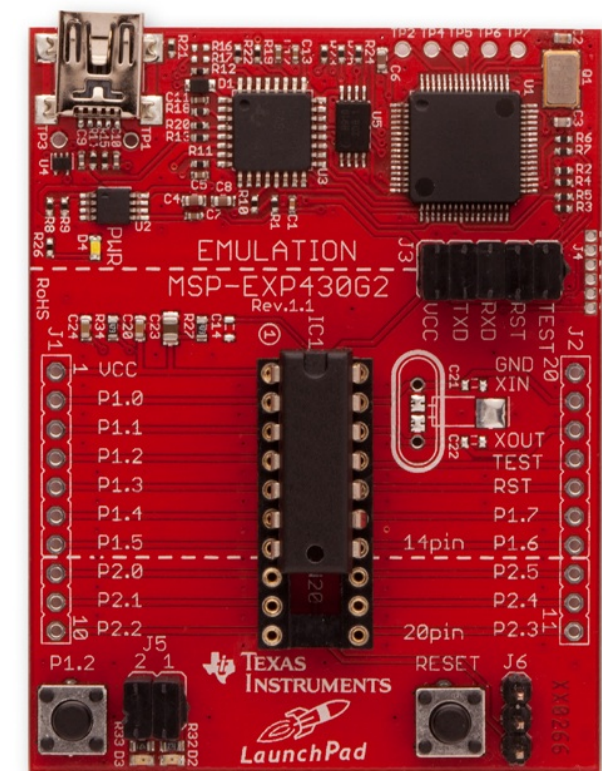
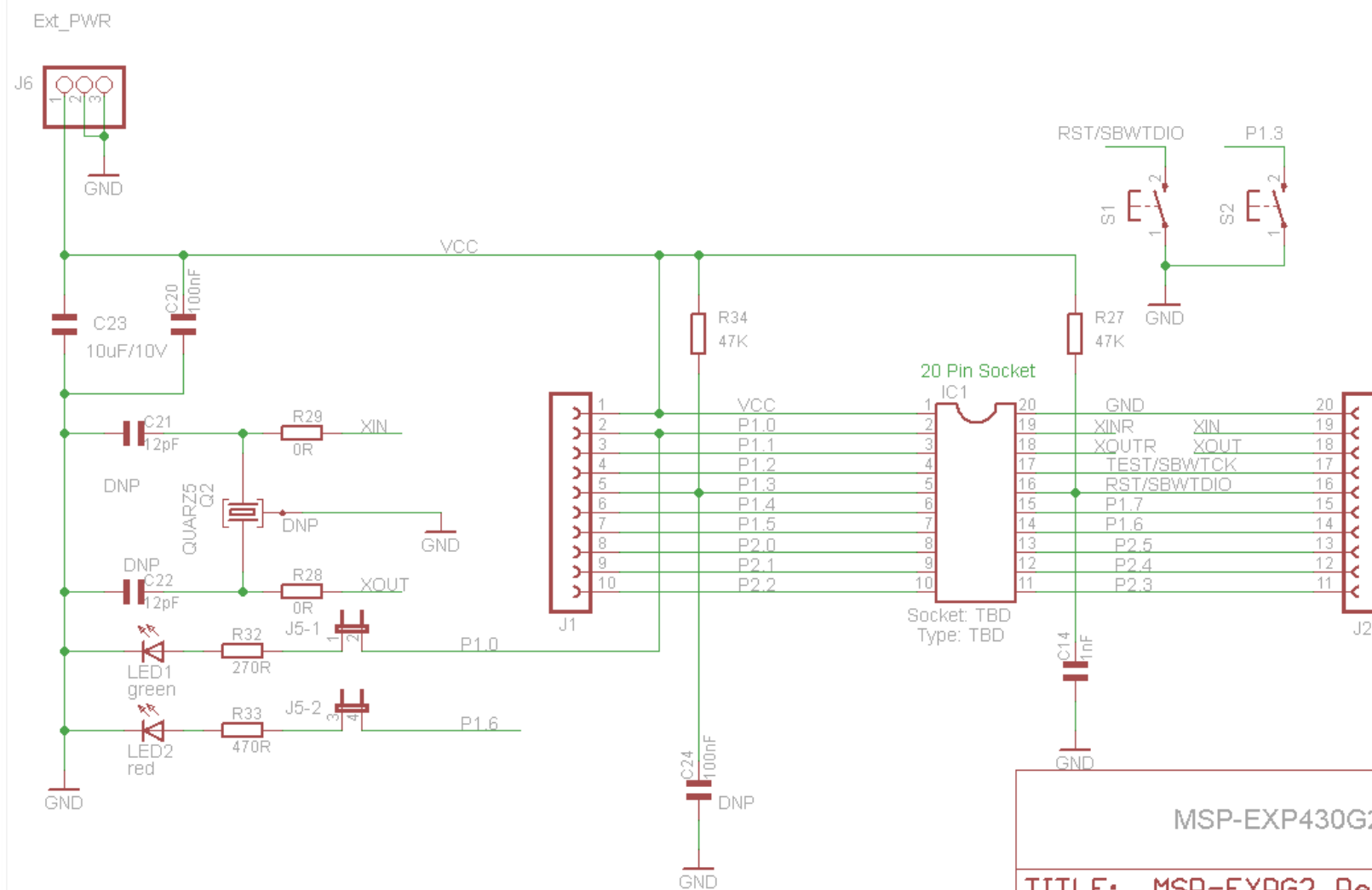


# MICROPROCESSADORES E MICROCONTROLADORES



# MSP430 LAUNCHPAD



MSP-EXP430G2 TARGET SOCKET

TITLE: MSP-EXPG2\_Rev.1.3

Document Number:

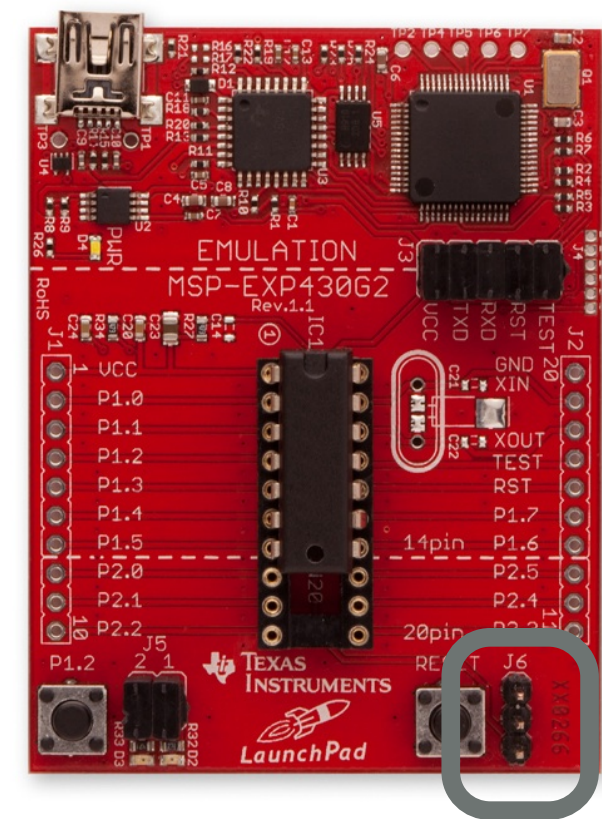
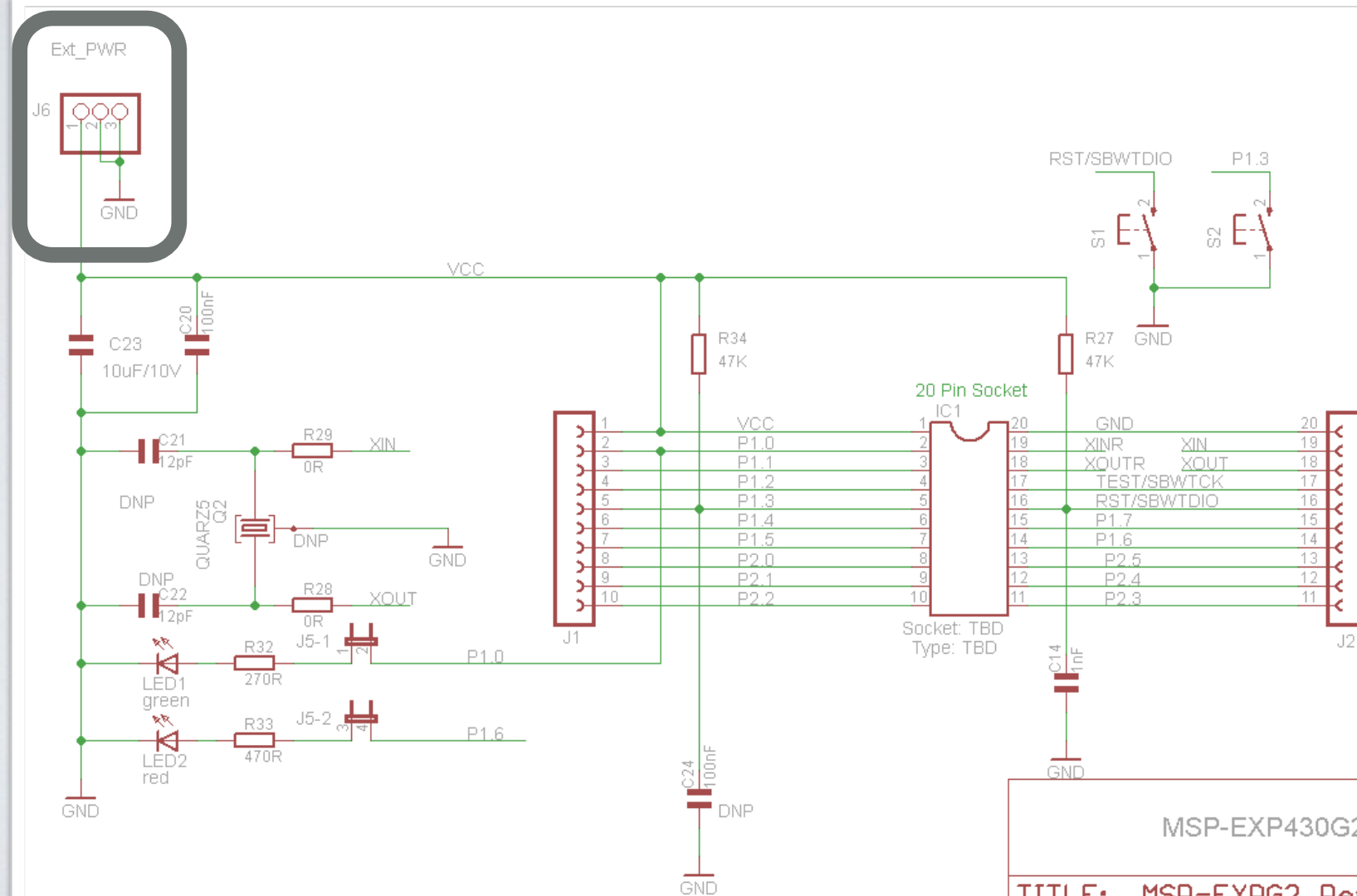
REV:  
1.3

Date: 7/8/2010 10:08:43 PM

Sheet: 3/3



# MSP430 LAUNCHPAD



MSP-EXP430G2 TARGET SOCKET

TITLE: MSP-EXP430G2\_Rev.1.3

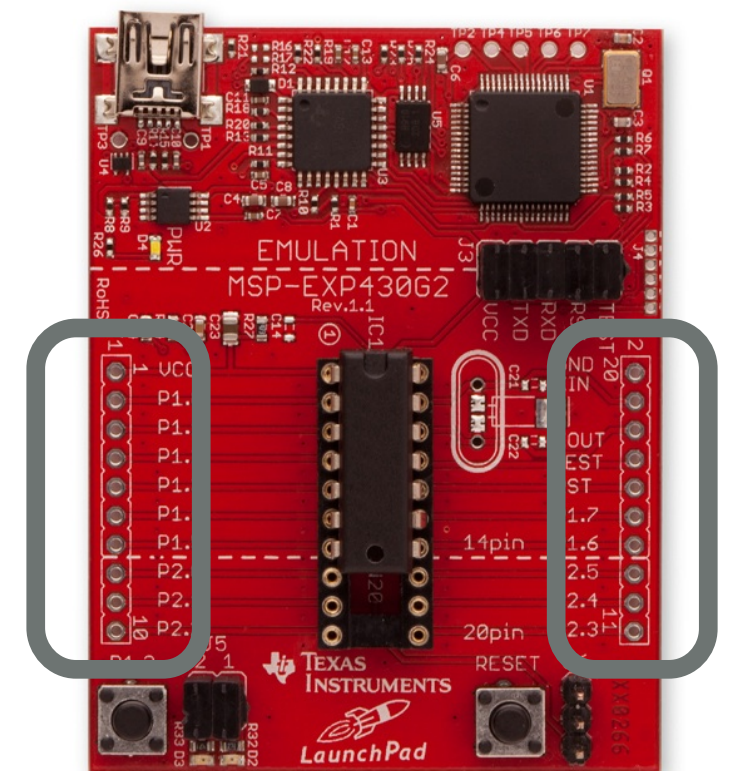
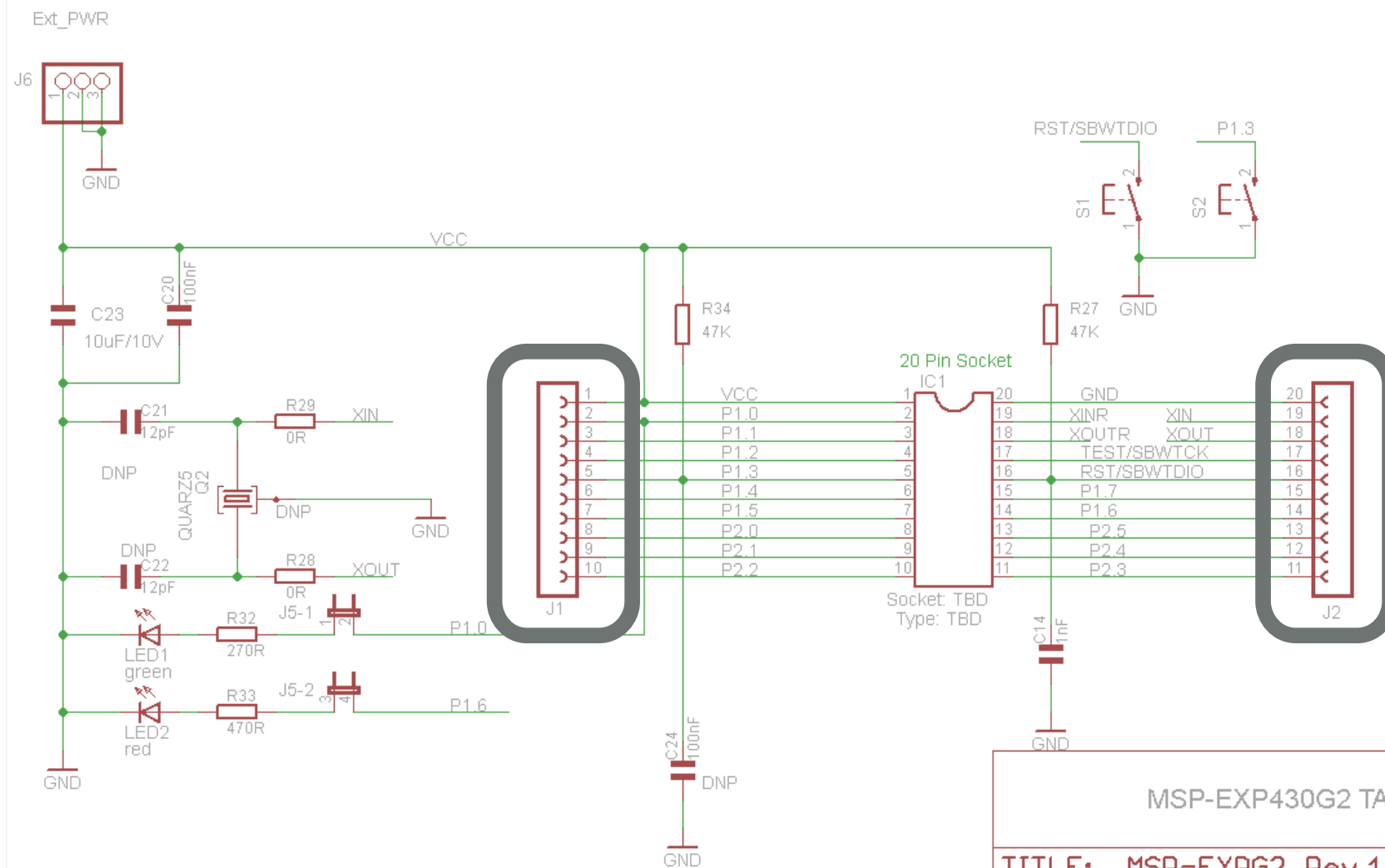
Document Number:

REV:  
1.3

Date: 7/8/2010 10:08:43 PM

Sheet: 3/3

# MSP430 LAUNCHPAD



MSP-EXP430G2 TARGET SOCKET

TITLE: MSP-EXP430G2\_Rev.1.3

Document Number:

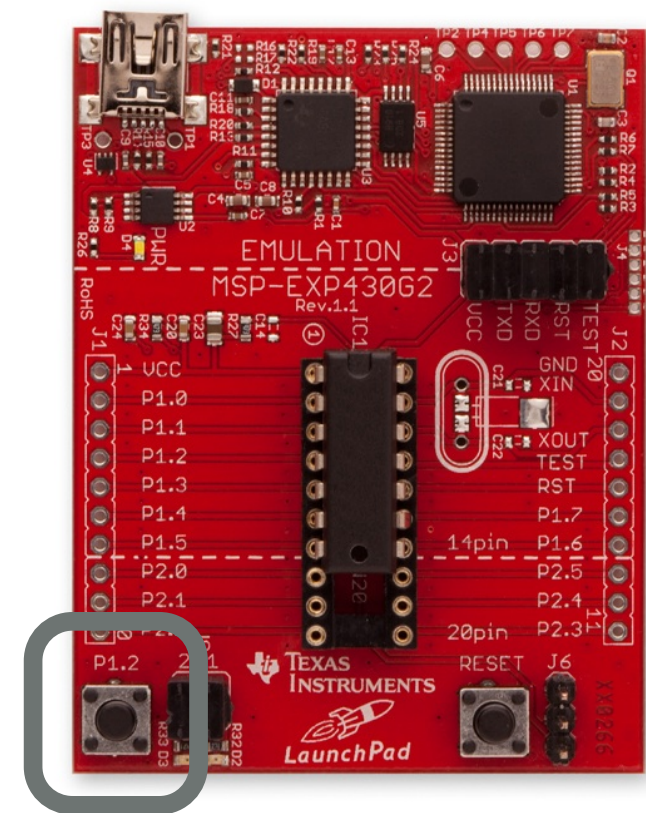
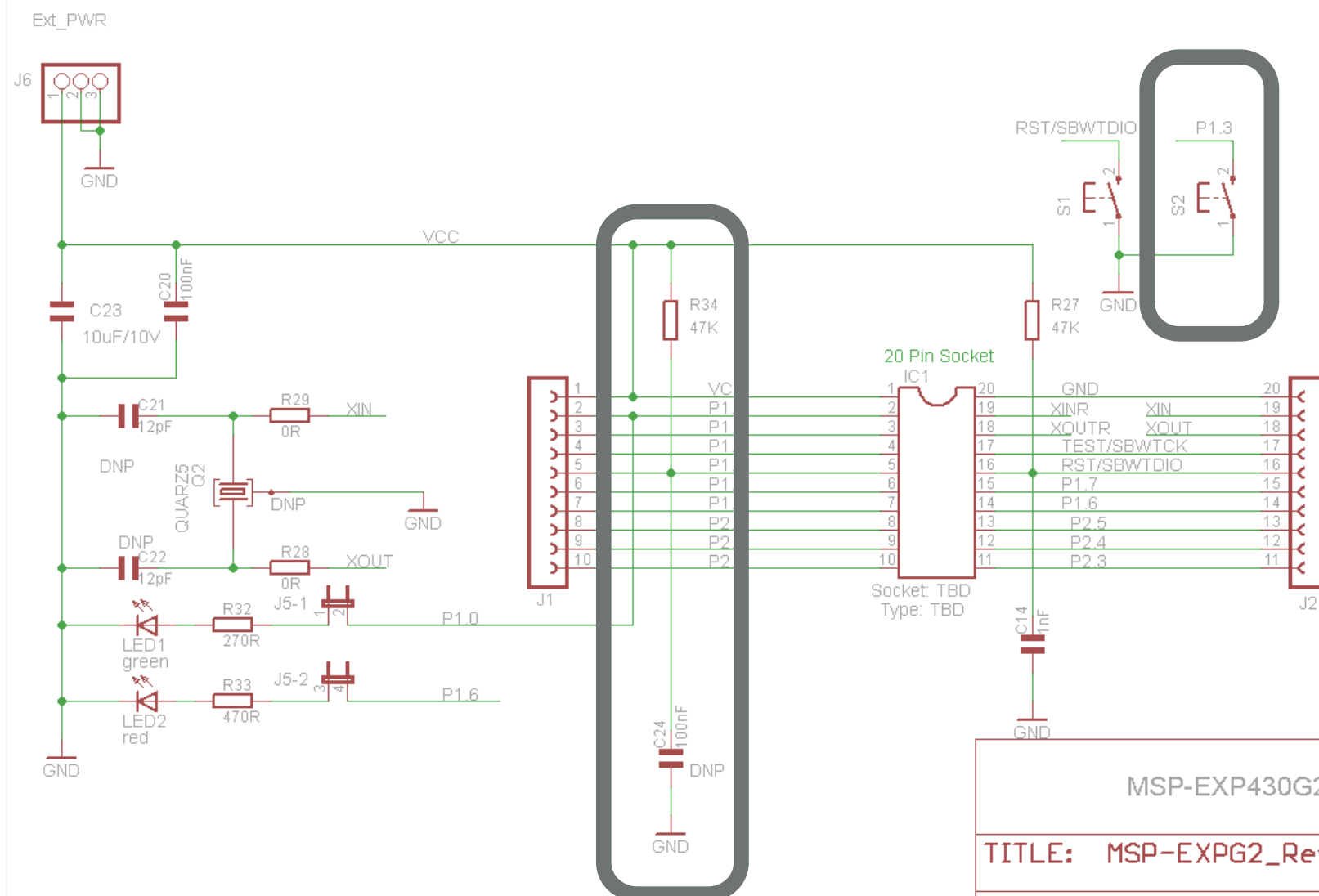
REV:  
1.3

Date: 7/8/2010 10:08:43 PM

Sheet: 3/3



# MSP430 LAUNCHPAD



MSP-EXP430G2 TARGET SOCKET

TITLE: MSP-EXP430G2\_Rev.1.3

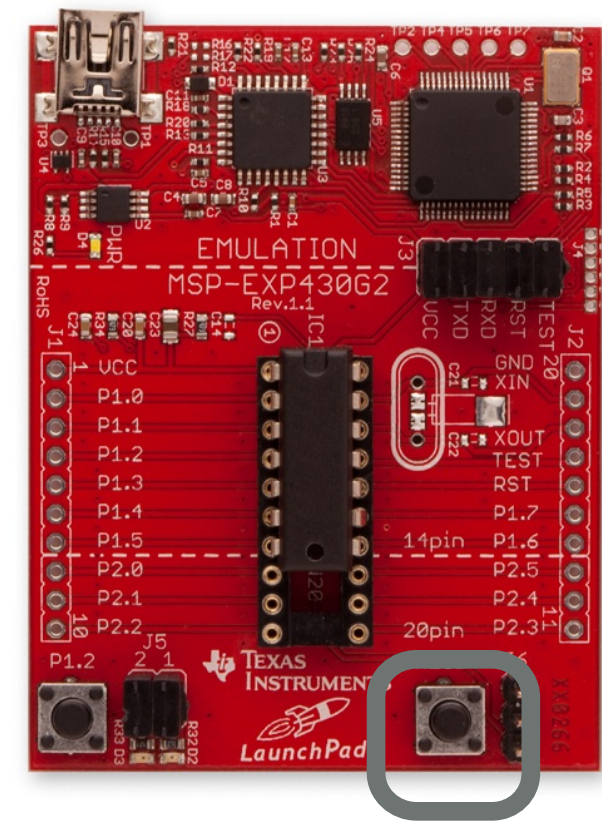
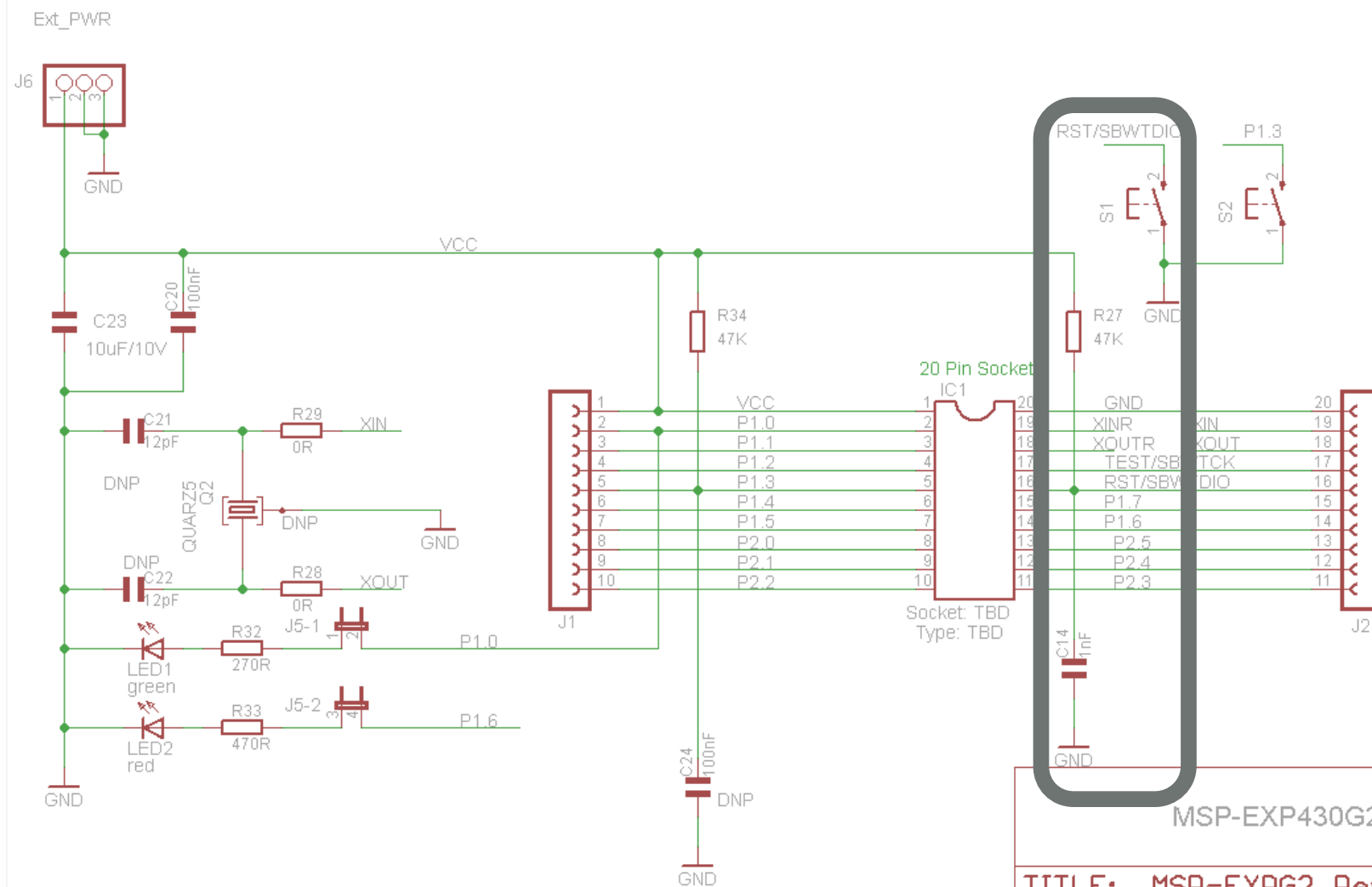
Document Number:

REV:  
1.3

Date: 7/8/2010 10:08:43 PM

Sheet: 3/3

# MSP430 LAUNCHPAD



MSP-EXP430G2 TARGET SOCKET

TITLE: MSP-EXP430G2\_Rev.1.3

Document Number:

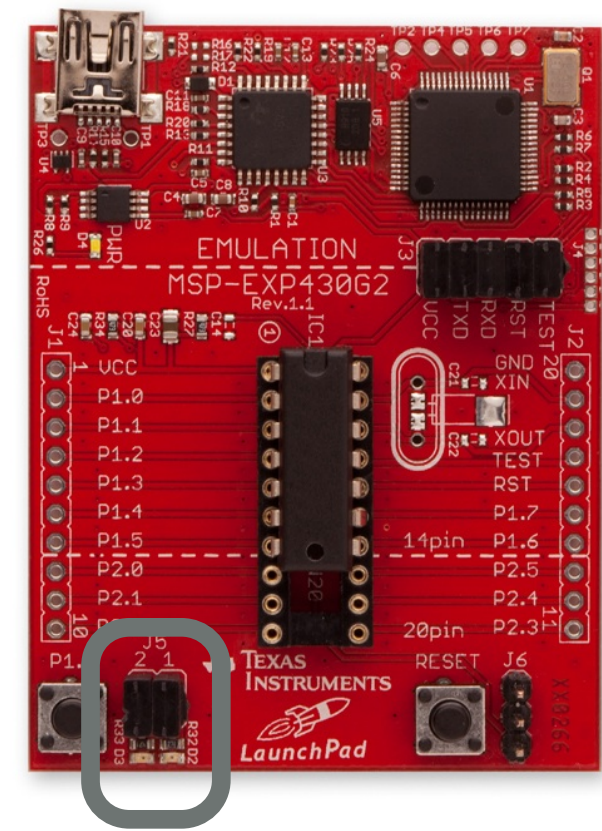
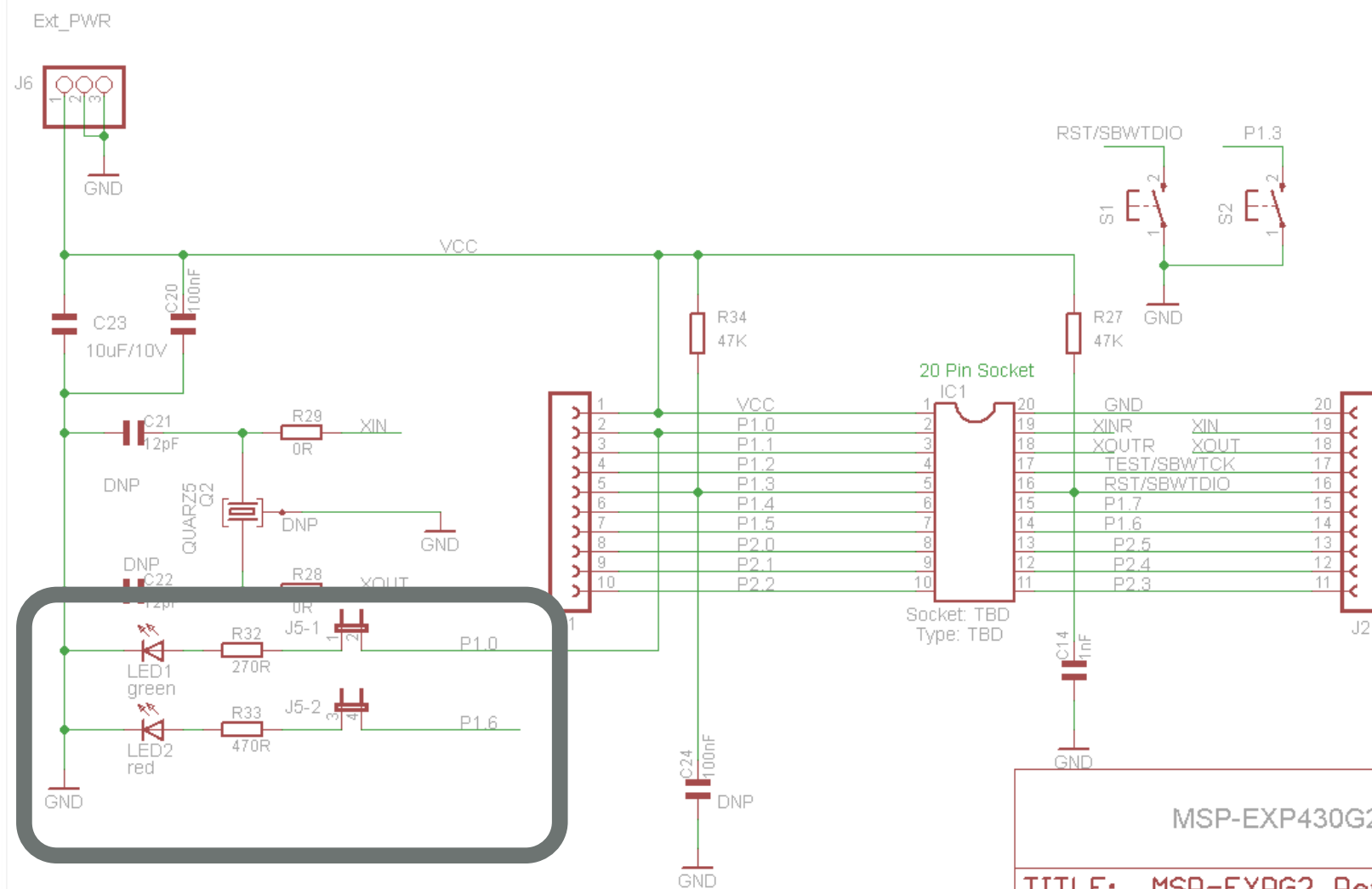
REV:  
1.3

Date: 7/8/2010 10:08:43 PM

Sheet: 3/3



# MSP430 LAUNCHPAD



MSP-EXP430G2 TARGET SOCKET

TITLE: MSP-EXP430G2\_Rev.1.3

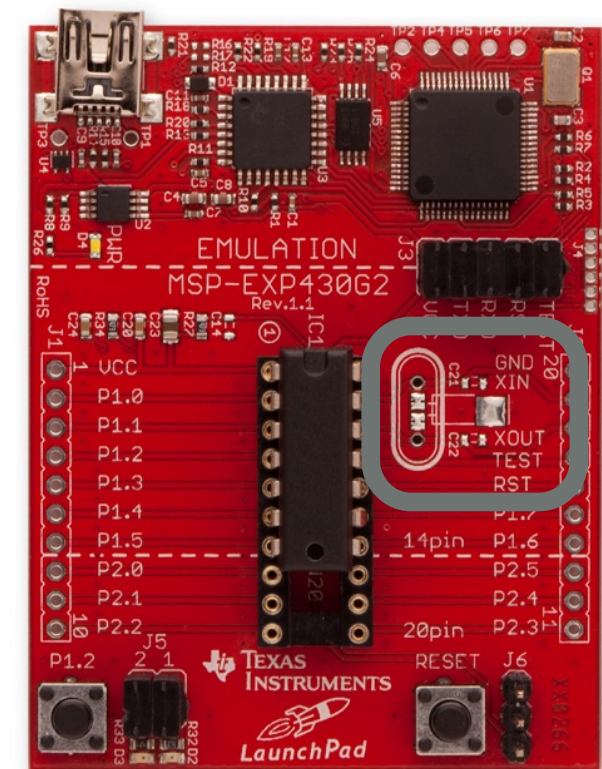
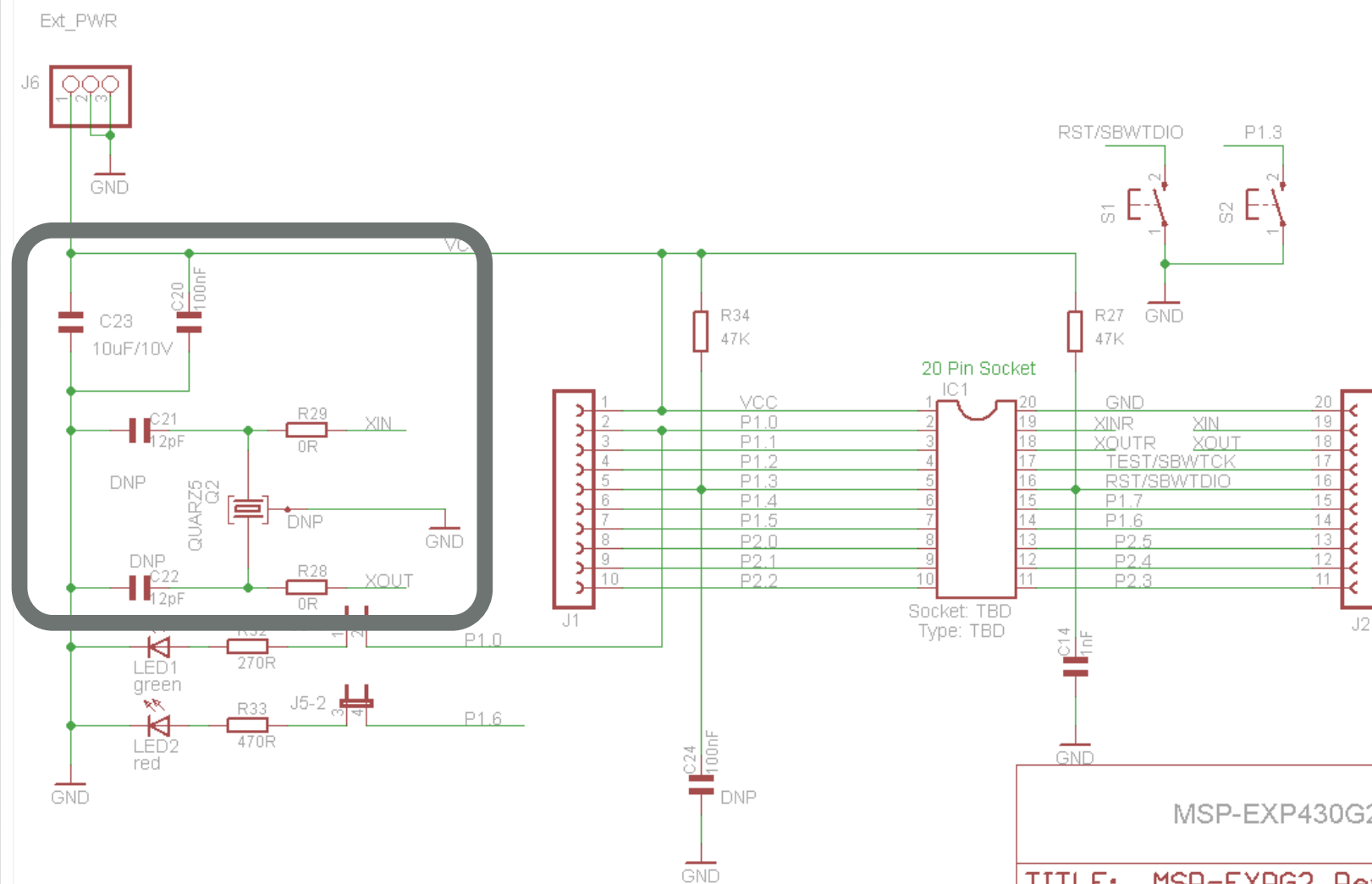
Document Number:

REV:  
1.3

Date: 7/8/2010 10:08:43 PM

Sheet: 3/3

# MSP430 LAUNCHPAD



MSP-EXP430G2 TARGET SOCKET

TITLE: MSP-EXP430G2\_Rev.1.3

Document Number:

REV:  
1.3

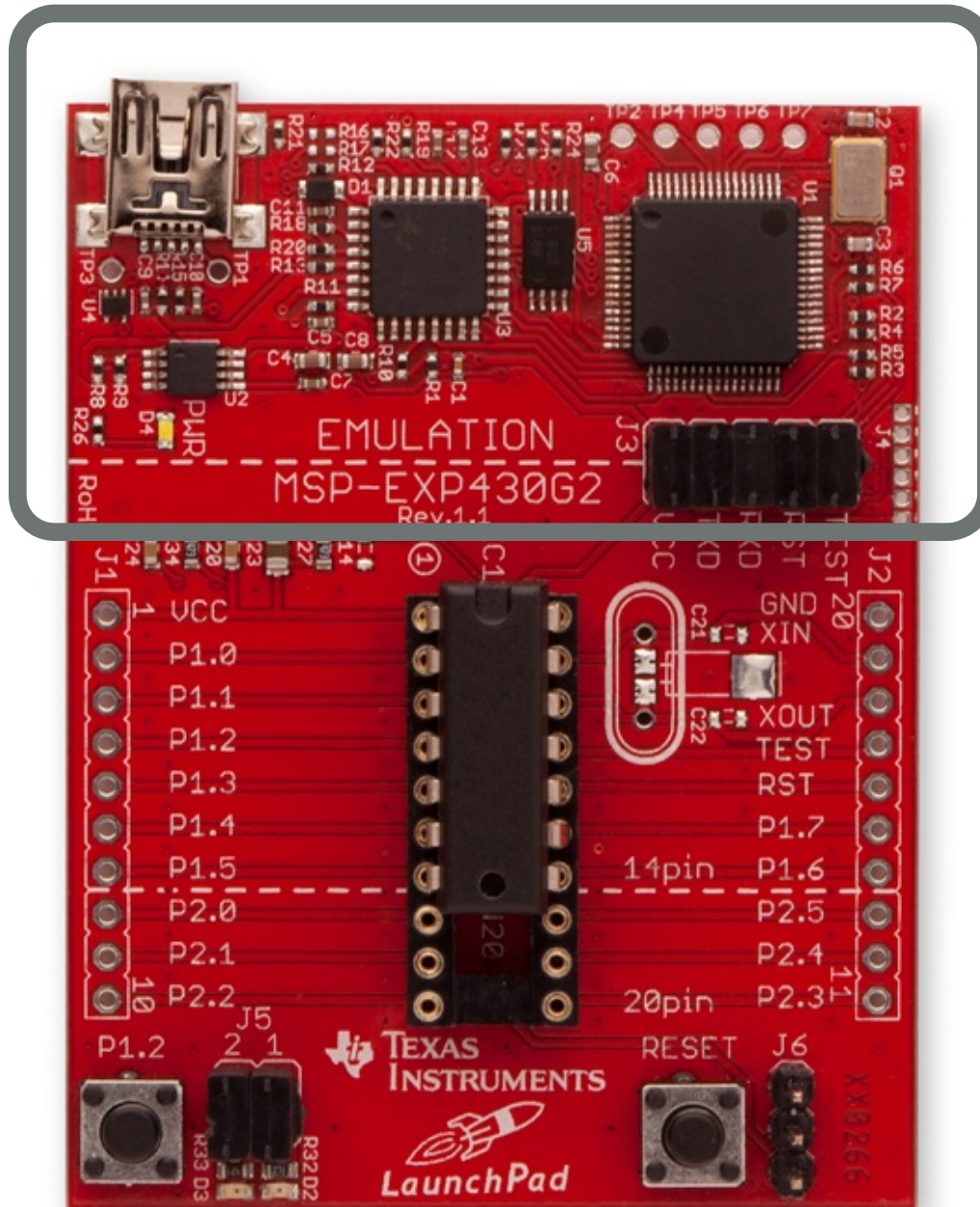
Date: 7/8/2010 10:08:43 PM

Sheet: 3/3



# MSP430 LAUNCHPAD

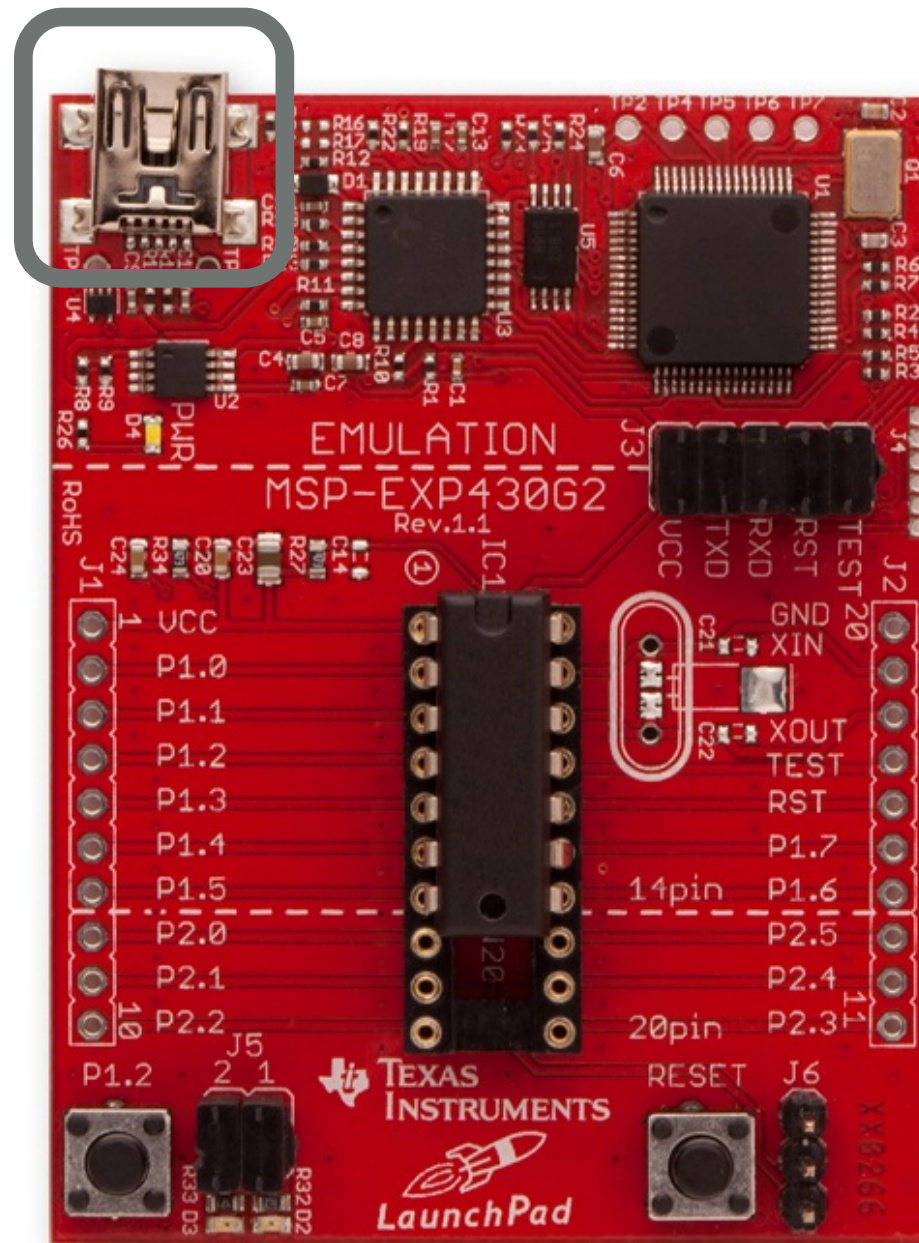
Sistema de  
programação  
para o MSP430





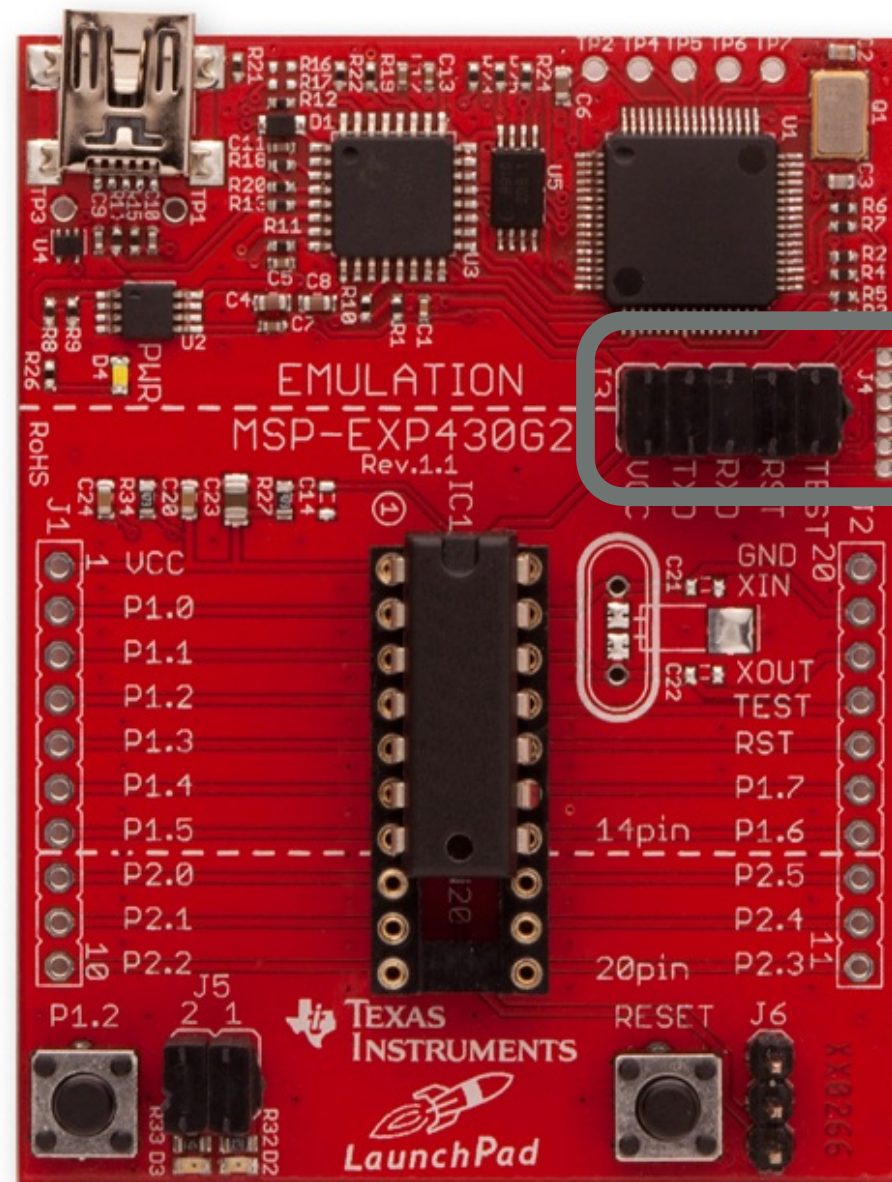
# MSP430 LAUNCHPAD

Conexão USB para  
a programação do  
MSP430





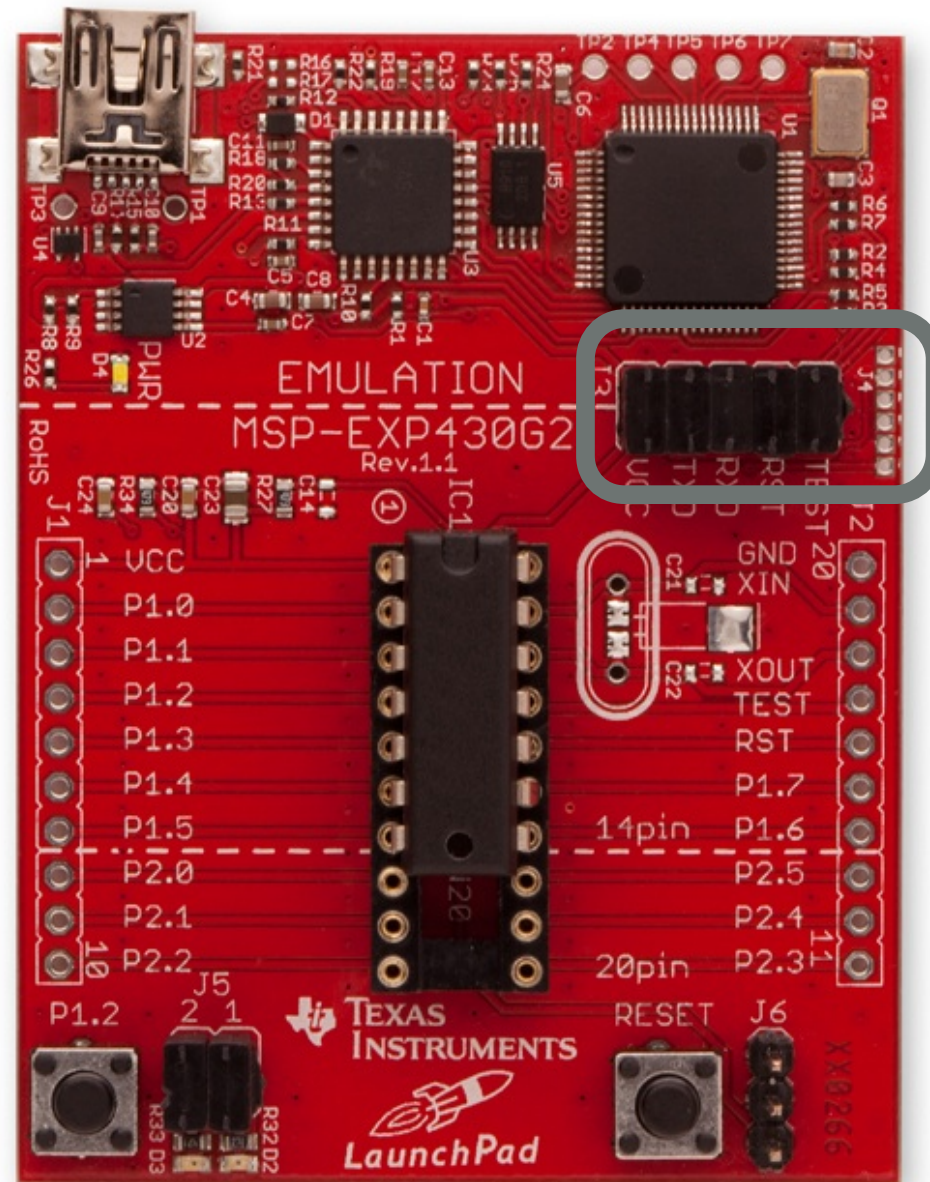
# MSP430 LAUNCHPAD



Soquetes para  
conectar sistema  
de programação



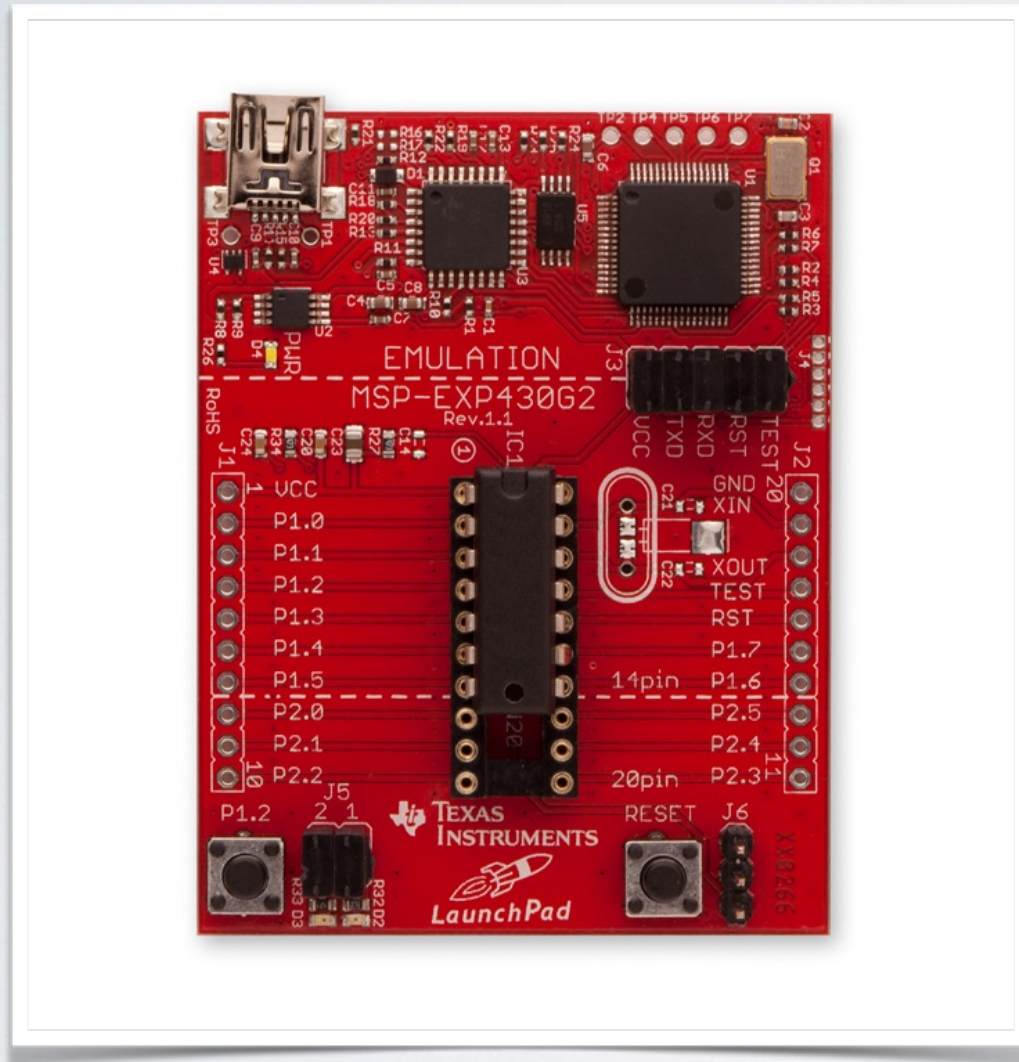
# MSP430 LAUNCHPAD



Podem ser desconectados e ligados a uma protoboard, para a programação de outro MSP430



# MSP430 LAUNCHPAD



```
//Hello world
```

```
#include <msp430g2553.h>
```

```
void main(void)
```

```
{
```

```
    WDTCTL = WDTPW | WDTHOLD;
```

```
    P1DIR = 0x41;
```

```
    P1OUT = 0x41;
```

```
    for(;;)
```

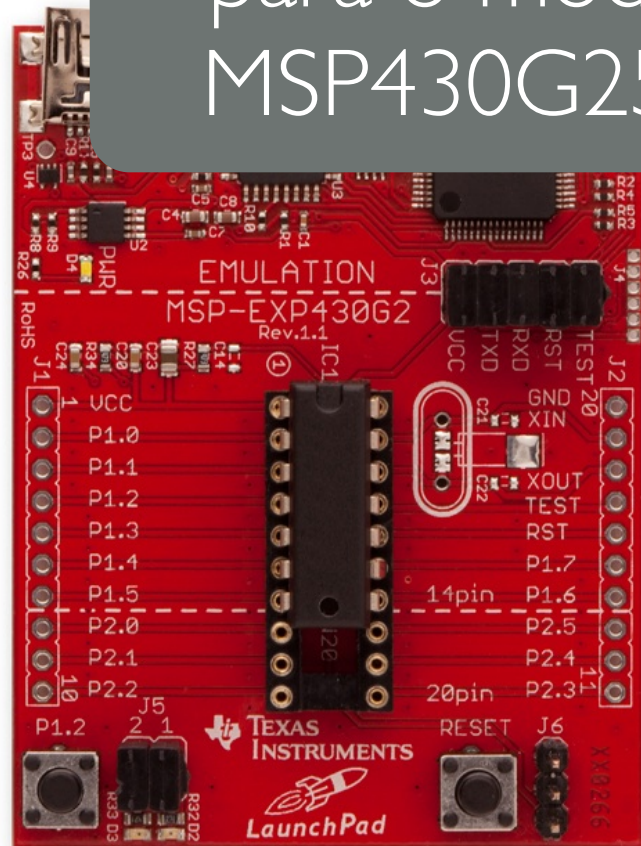
```
    {
```

```
    }
```

```
}
```

# MSP430 LAUNCHPAD

Header específico  
para o modelo  
MSP430G2553



```
//Hello world
```

```
#include <msp430g2553.h>
```

```
void main(void)
```

```
{
```

```
    WDTCTL = WDTPW | WDTHOLD;
```

```
    P1DIR = 0x41;
```

```
    P1OUT = 0x41;
```

```
    for(;;)
```

```
    {
```

```
    }
```

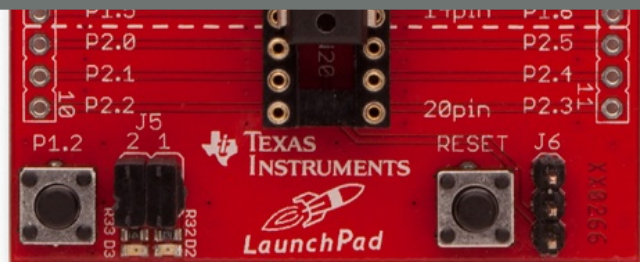
```
}
```



# MSP430 LAUNCHPAD



Parar o Watchdog Timer  
(mais detalhes na aula  
sobre Timers)



```
//Hello world
```

```
#include <msp430g2553.h>
```

```
void main(void)
```

```
{
```

```
    WDTCTL = WDTPW | WDTHOLD;
```

```
    P1DIR = 0x41;
```

```
    P1OUT = 0x41;
```

```
    for(;;)
```

```
    {
```

```
    }
```

```
}
```

# MSP430 LAUNCHPAD



Configurar os pinos  
0 e 6 como saída  
(0x41 = 0b01000001),  
que estão conectados  
aos dois LEDs do  
Launchpad

```
//Hello world
```

```
#include <msp430g2553.h>
```

```
void main(void)
```

```
{
```

```
    WDTCTL = WDTPW | WDTHOLD;
```

```
    P1DIR = 0x41;
```

```
    P1OUT = 0x41;
```

```
    for(;;)
```

```
    {
```

```
    }
```

```
}
```



# MSP430 LAUNCHPAD



Levar os pinos  
0 e 6 para Vcc  
(0x41 = 0b01000001),  
ligando os dois LEDs

```
//Hello world
```

```
#include <msp430g2553.h>
```

```
void main(void)
```

```
{
```

```
    WDTCTL = WDTPW | WDTHOLD;
```

```
    P1DIR = 0x41;
```

```
    P1OUT = 0x41;
```

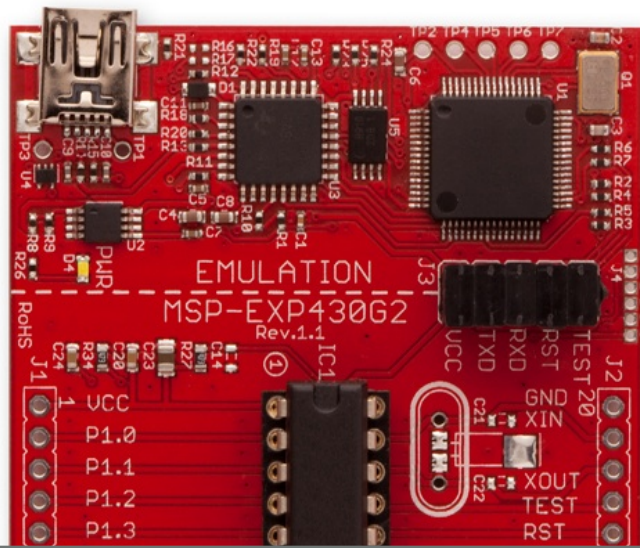
```
    for(;;)
```

```
    {
```

```
    }
```

```
}
```

# MSP430 LAUNCHPAD



Loop infinito -  
o MSP430 não tem um sistema operacional

```
//Hello world
```

```
#include <msp430g2553.h>
```

```
void main(void)
```

```
{
```

```
    WDTCTL = WDTPW | WDTHOLD;
```

```
    P1DIR = 0x41;
```

```
    P1OUT = 0x41;
```

```
    for(;;)
```

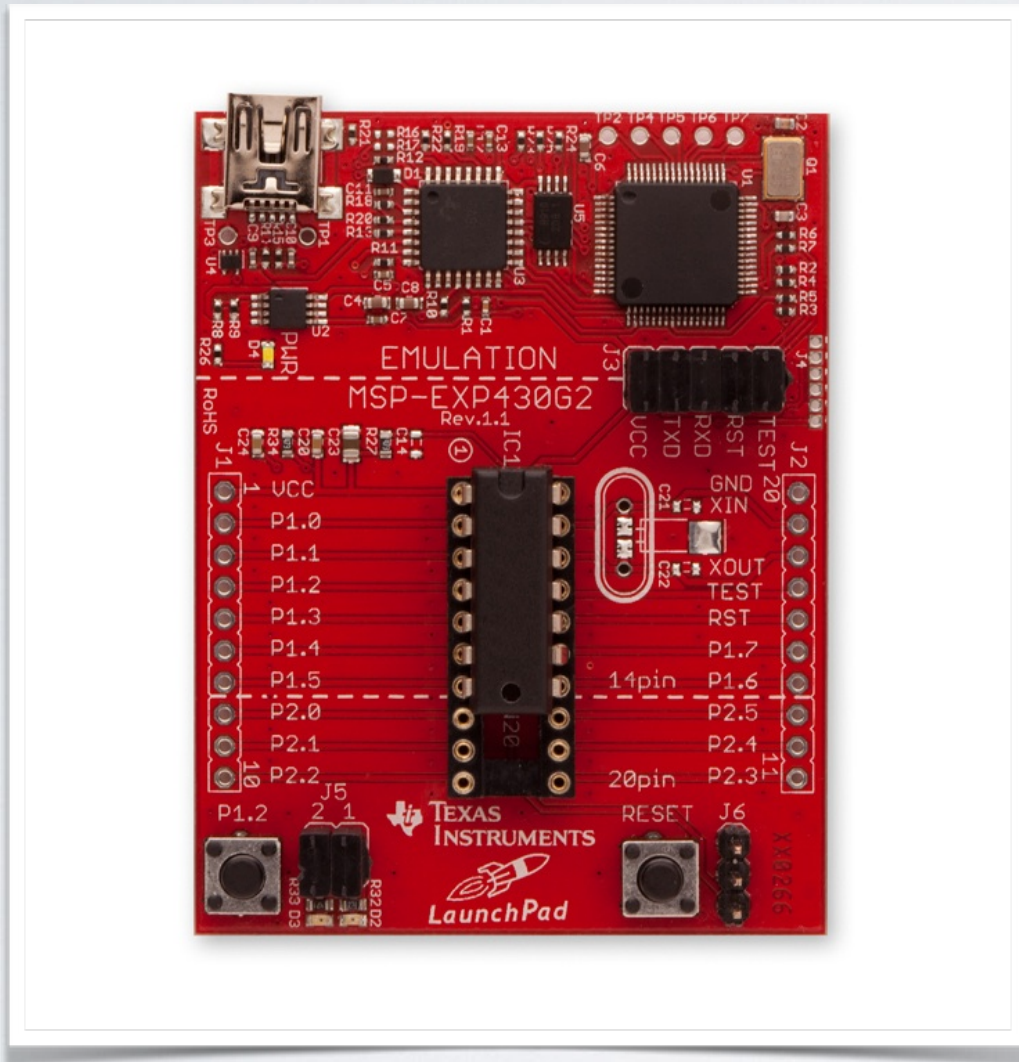
```
    {
```

```
    }
```

```
}
```



# MSP430 LAUNCHPAD



```
//Hello world
```

```
#include <msp430g2553.h>
```

```
void main(void)
```

```
{
```

```
    WDTCTL = WDTPW | WDTHOLD;
```

```
    P1DIR = BIT6 + BIT0;
```

```
    P1OUT = BIT6 + BIT0;
```

```
    for(;;)
```

```
    {
```

```
    }
```

```
}
```

# MSP430 LAUNCHPAD



Forma alternativa de  
setar bits:

BIT0 = 0b000000001

BIT6 = 0b010000000

(definidos no header)

```
//Hello world
```

```
#include <msp430g2553.h>
```

```
void main(void)
```

```
{
```

```
    WDTCTL = WDTPW | WDTHOLD;
```

```
    P1DIR = BIT6 + BIT0;
```

```
    P1OUT = BIT6 + BIT0;
```

```
    for(;;)
```

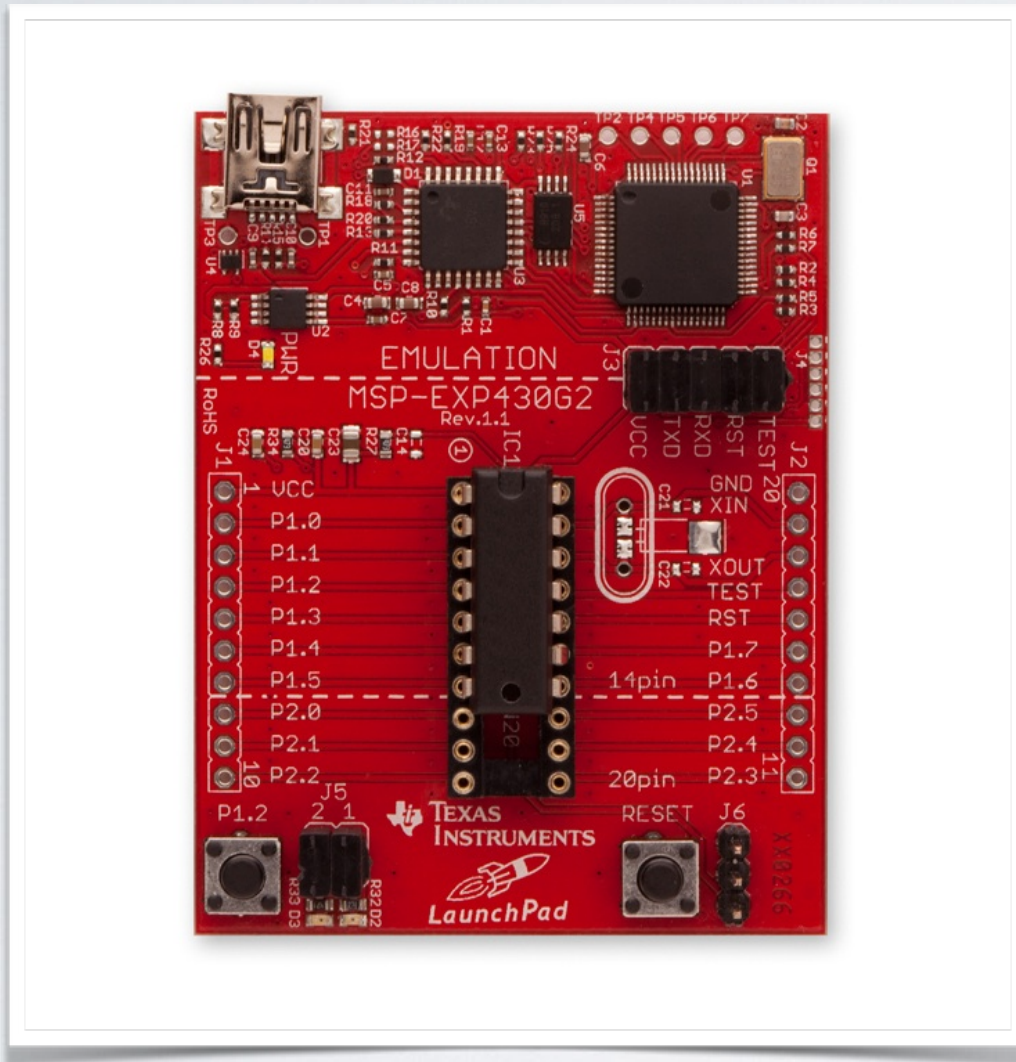
```
    {
```

```
    }
```

```
}
```



# MSP430 LAUNCHPAD



```
;Hello world em assembly  
#include <msp430g2553.h>
```

```
ORG 0xC000
```

```
Reset:
```

```
    mov.w  #WDTPW | WDTHOLD,  
&WDTCCTL
```

```
    mov.b  #0 | 000000 | b, P1OUT
```

```
    mov.b  #0 | 000000 | b, P1DIR
```

```
Loop:
```

```
    jmp    Loop
```

```
;
```

```
ORG 0xFFFF
```

```
DW Reset
```

```
END
```

# MSP430 LAUNCHPAD

Comentários

```
;Hello world em assembly
```

```
#include <msp430g2553.h>
```

```
ORG 0xC000
```

```
Reset:
```

```
    mov.w  #WDTPW | WDTHOLD,  
&WDTCTL
```

```
    mov.b  #0 | 000000 | b, P1OUT
```

```
    mov.b  #0 | 000000 | b, P1DIR
```

```
Loop:
```

```
    jmp    Loop
```

```
;-----
```

```
ORG 0xFFFF
```

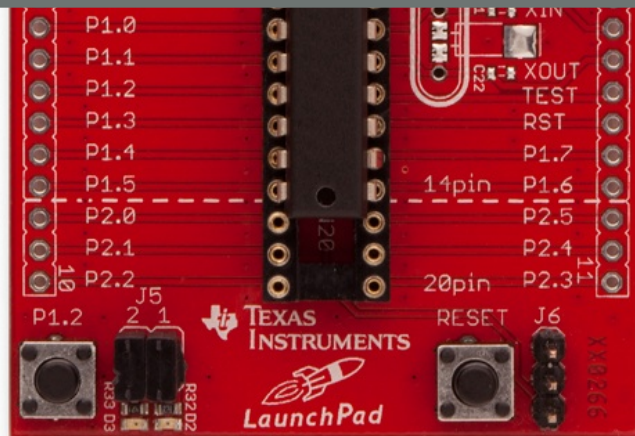
```
DW Reset
```

```
END
```



# MSP430 LAUNCHPAD

MSP430G2553 tem  
16kB de memória flash,  
que começa no  
endereço 0xC000  
(agora, a posição inicial  
do nosso código)



```
;Hello world em assembly  
#include <msp430g2553.h>
```

```
ORG 0xC000
```

Reset:

```
    mov.w  #WDTPW | WDTHOLD,  
&WDTCCTL
```

```
    mov.b  #0 | 000000 | b, P1OUT
```

```
    mov.b  #0 | 000000 | b, P1DIR
```

Loop:

```
    jmp    Loop
```

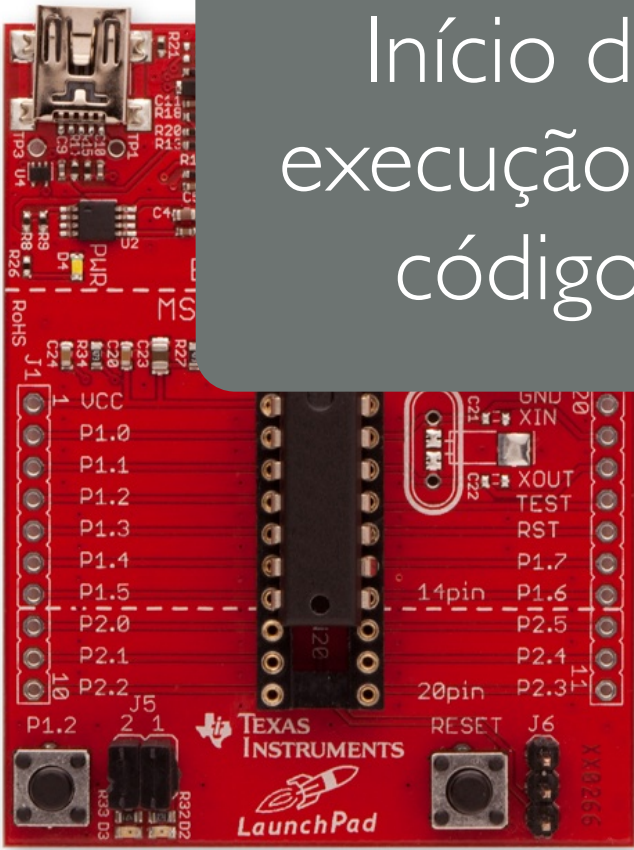
```
;
```

```
ORG 0xFFFF
```

```
DW Reset
```

```
END
```

# MSP430 LAUNCHPAD



Início da  
execução do  
código

```
;Hello world em assembly  
#include <msp430g2553.h>
```

```
ORG 0xC000
```

```
Reset:
```

```
mov.w #WDTPW | WDTHOLD,  
&WDTCCTL
```

```
mov.b #0 | 000000 | b, P1OUT
```

```
mov.b #0 | 000000 | b, P1DIR
```

```
Loop:
```

```
jmp Loop
```

```
;-----
```

```
ORG 0xFFFF
```

```
DW Reset
```

```
END
```



# MSP430 LAUNCHPAD

Mesma lógica do  
programa em C

```
;Hello world em assembly
```

```
#include <msp430g2553.h>
```

```
ORG 0xC000
```

```
Reset:
```

```
    mov.w    #WDTPW | WDTHOLD,  
    &WDTCTL
```

```
    mov.b    #0 | 000000 | b, P1OUT
```

```
    mov.b    #0 | 000000 | b, P1DIR
```

```
Loop:
```

```
    jmp      Loop
```

```
ORG 0xFFFF
```

```
DW Reset
```

```
END
```

# MSP430 LAUNCHPAD



Endereço do vetor  
de RESET do  
MSP430G2553

```
;Hello world em assembly  
#include <msp430g2553.h>
```

```
ORG 0xC000
```

Reset:

```
mov.w #WDTPW | WDTHOLD,  
&WDTCCTL
```

```
mov.b #0 | 000000 | b, P1OUT
```

```
mov.b #0 | 000000 | b, P1DIR
```

Loop:

```
jmp Loop
```

---

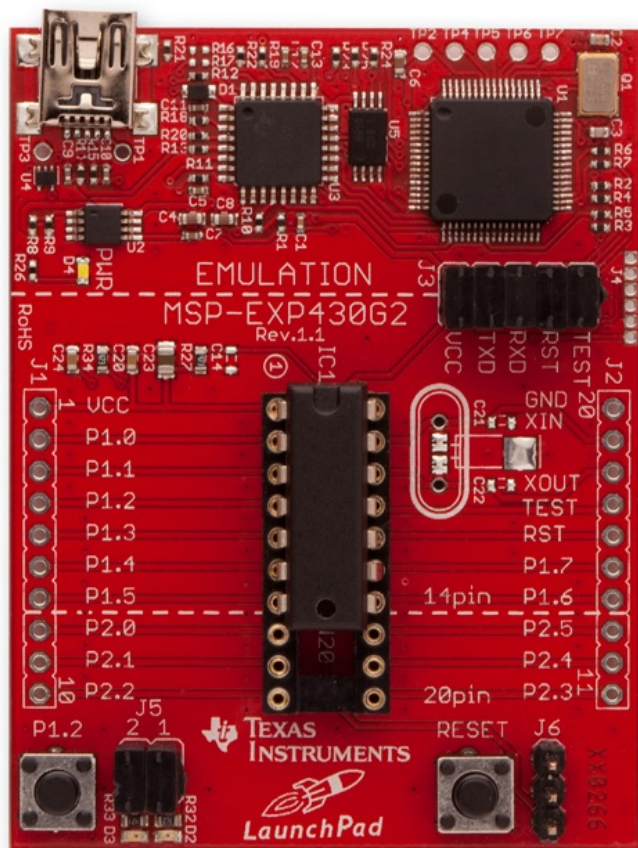
```
ORG 0xFFFF
```

```
DW Reset
```

```
END
```



# MSP430 LAUNCHPAD



Endereço para onde  
aponta o vetor de RESET

```
;Hello world em assembly  
#include <msp430g2553.h>
```

```
ORG 0xC000
```

Reset:

```
mov.w #WDTPW | WDTHOLD,  
&WDTCCTL
```

```
mov.b #0 | 000000 | b, P1OUT
```

```
mov.b #0 | 000000 | b, P1DIR
```

Loop:

```
jmp Loop
```

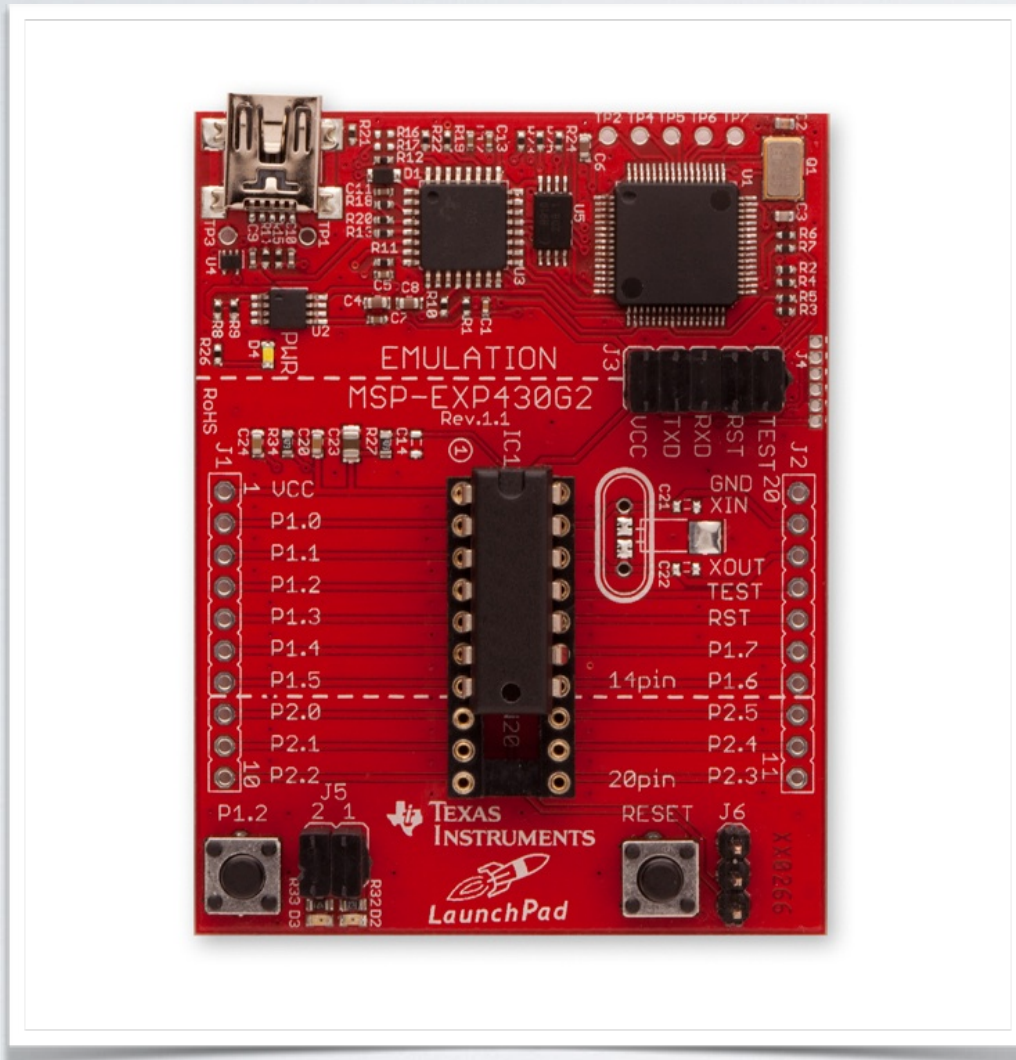
---

```
ORG 0xFFFF
```

```
DW Reset
```

```
END
```

# MSP430 LAUNCHPAD



```
;Hello world em assembly  
#include <msp430g2553.h>
```

```
ORG 0xC000
```

```
Reset:
```

```
    mov.w  #WDTPW | WDTHOLD,  
&WDTCCTL
```

```
    mov.b  #0 | 000000 | b, P1OUT
```

```
    mov.b  #0 | 000000 | b, P1DIR
```

```
Loop:
```

```
    jmp    Loop
```

```
;
```

```
ORG 0xFFFF
```

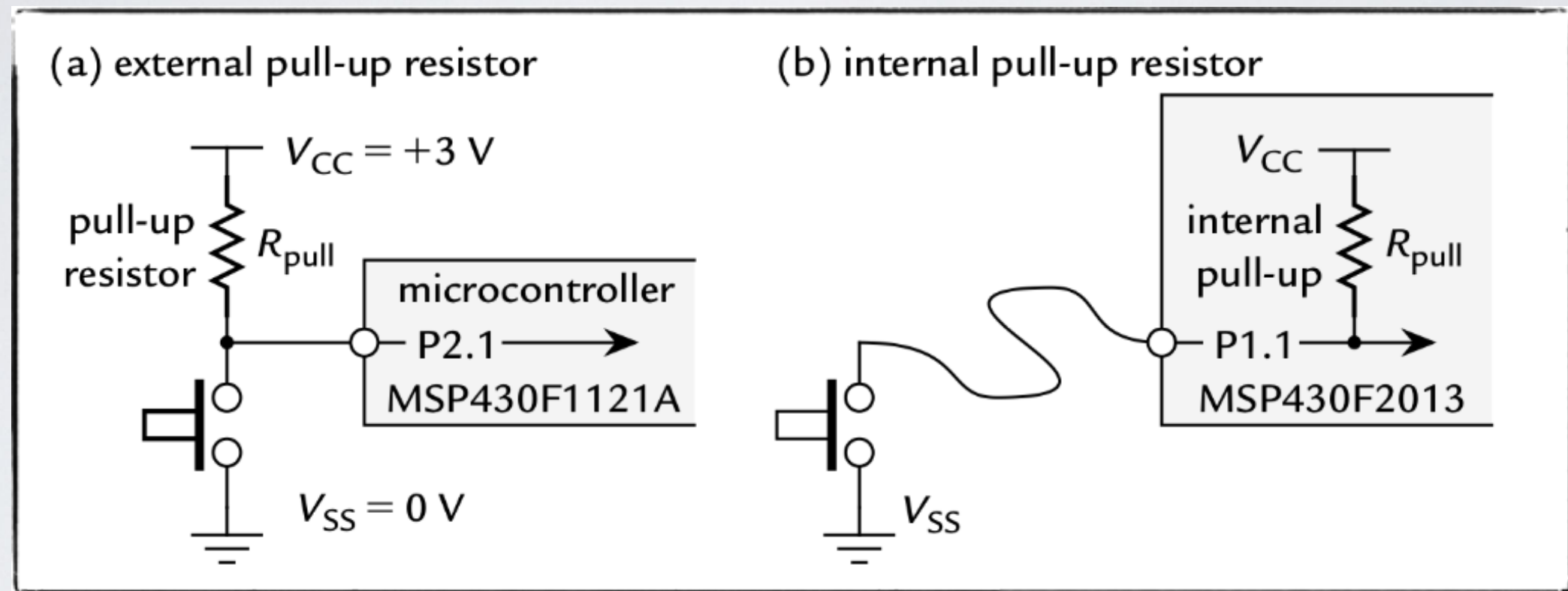
```
DW Reset
```

```
END
```

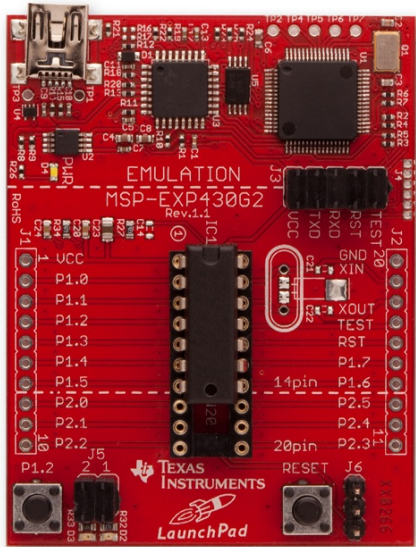
FIM



# MSP430 LAUNCHPAD

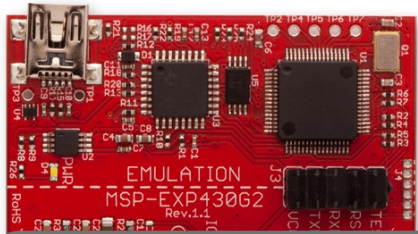


Conexões mais comuns para o uso de botões



```
//Ligar LEDS enquanto o
//botão estiver pressionado
#include <msp430g2553.h>
#define BTN BIT2
#define LED1 BIT0
#define LED2 BIT6
void main(void){
    WDTCTL = WDTPW | WDTHOLD;
    P1OUT = 0;
    P1DIR = LED1 + LED2;
    for(;;)
    {
        if(P1IN & BTN == 0)
            P1OUT |= LED1 + LED2;
        else
            P1OUT &= ~(LED1 + LED2);
    }
}
```



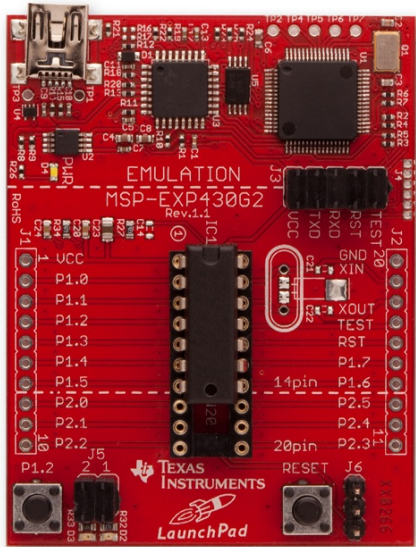


Definições para  
clarear o código

```
//Ligar LEDS enquanto o  
//botão estiver pressionado  
#include <msp430g2553.h>
```

```
#define BTN BIT2  
#define LED1 BIT0  
#define LED2 BIT6
```

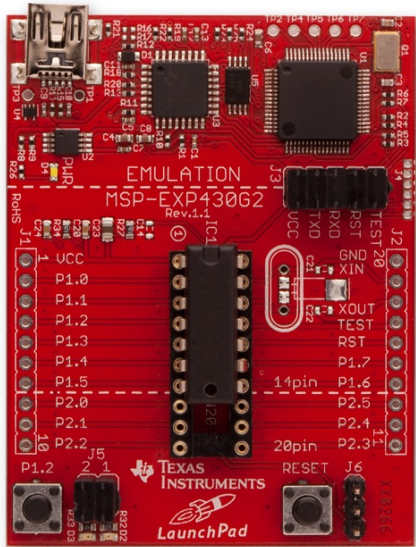
```
void main(void){  
    WDTCTL = WDTPW | WDTHOLD;  
    P1OUT = 0;  
    P1DIR = LED1 + LED2;  
    for(;;)  
    {  
        if(P1IN & BTN == 0)  
            P1OUT |= LED1 + LED2;  
        else  
            P1OUT &= ~(LED1 + LED2);  
    }  
}
```



Começar com  
os LEDs desligados

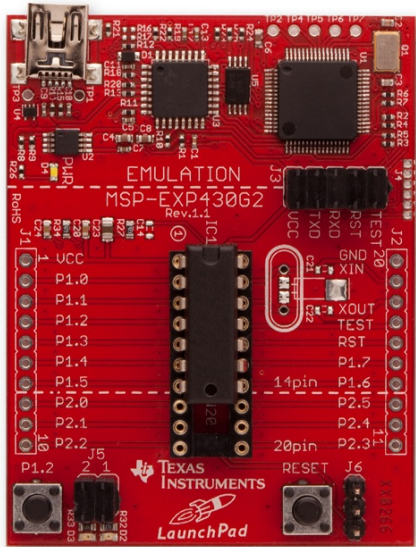
```
//Ligar LEDs enquanto o
//botão estiver pressionado
#include <msp430g2553.h>
#define BTN BIT2
#define LED1 BIT0
#define LED2 BIT6
void main(void){
    WDTCTL = WDTPW | WDTHOLD;
    P1OUT = 0;
    P1DIR = LED1 + LED2;
    for(;;)
    {
        if(P1IN & BTN == 0)
            P1OUT |= LED1 + LED2;
        else
            P1OUT &= ~(LED1 + LED2);
    }
}
```





LED1 e LED2  
como saídas,  
BTN como entrada

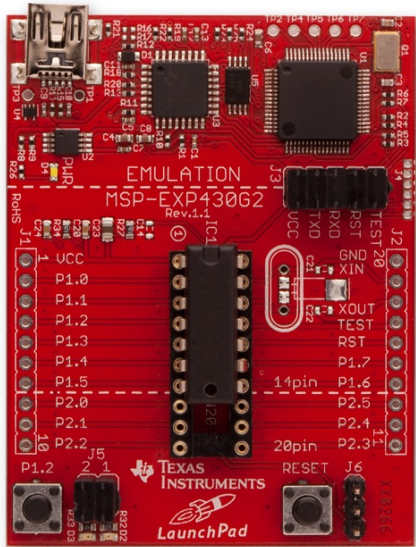
```
//Ligar LEDS enquanto o
//botão estiver pressionado
#include <msp430g2553.h>
#define BTN BIT2
#define LED1 BIT0
#define LED2 BIT6
void main(void){
    WDTCTL = WDTPW | WDTHOLD;
    P1OUT = 0;
    P1DIR = LED1 + LED2;
    for(;;)
    {
        if(P1IN & BTN == 0)
            P1OUT |= LED1 + LED2;
        else
            P1OUT &= ~(LED1 + LED2);
    }
}
```



Ligar os LEDs enquanto o botão estiver pressionado

```
//Ligar LEDs enquanto o
//botão estiver pressionado
#include <msp430g2553.h>
#define BTN BIT2
#define LED1 BIT0
#define LED2 BIT6
void main(void){
    WDTCTL = WDTPW | WDTHOLD;
    P1OUT = 0;
    P1DIR = LED1 + LED2;
    for(;;)
    {
        if(P1IN & BTN == 0)
            P1OUT |= LED1 + LED2;
        else
            P1OUT &= ~(LED1 + LED2);
    }
}
```





```
//Ligar LEDS enquanto o  
//botão estiver pressionado
```

```
#include <msp430g2553.h>
```

```
#define BTN BIT2
```

```
#define LED1 BIT0
```

```
#define LED2 BIT6
```

```
void main(void){
```

```
    WDTCTL = WDTPW | WDTHOLD;
```

```
    P1OUT = 0;
```

```
    P1DIR = LED1 + LED2;
```

```
    for(;;)
```

```
    {
```

```
        if(P1IN & BTN == 0)
```

```
            P1OUT |= LED1 + LED2;
```

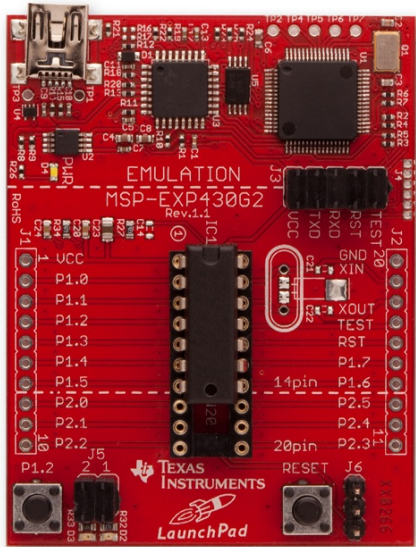
```
        else
```

```
            P1OUT &= ~(LED1 + LED2);
```

```
    }
```

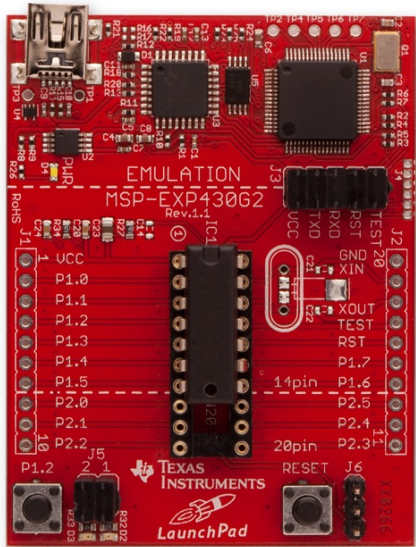
```
}
```

Caso contrário,  
apagar os LEDs



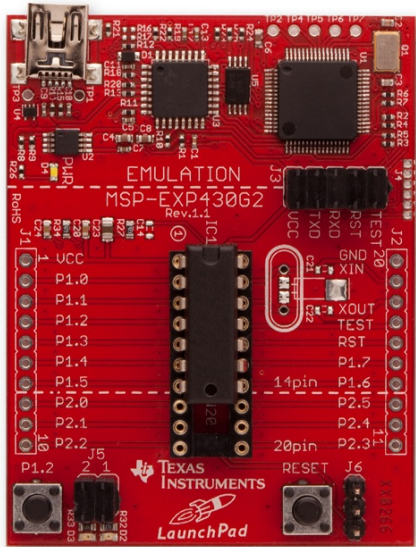
```
//Ligar LEDs enquanto o
//botão estiver pressionado
#include <msp430g2553.h>
#define BTN BIT2
#define LED1 BIT0
#define LED2 BIT6
void main(void){
    WDTCTL = WDTPW | WDTHOLD;
    P1OUT = 0;
    P1DIR = LED1 + LED2;
    for(;;)
    {
        while((P1IN & BTN) != 0){
            P1OUT |= LED1 + LED2;
            while((P1IN & BTN) == 0){
                P1OUT &= ~(LED1 + LED2);
            }
        }
    }
}
```





Código fica preso neste  
laço enquanto o botão  
não é pressionado

```
//Ligar LEDS enquanto o  
//botão estiver pressionado  
#include <msp430g2553.h>  
#define BTN BIT2  
#define LED1 BIT0  
#define LED2 BIT6  
void main(void){  
    WDTCTL = WDTPW | WDTHOLD;  
    P1OUT = 0;  
    P1DIR = LED1 + LED2;  
    for(;;)  
    {  
        while((P1IN & BTN) != 0){  
            P1OUT |= LED1 + LED2;  
            while((P1IN & BTN) == 0){  
                P1OUT &= ~(LED1 + LED2);  
            }  
        }  
    }  
}
```



Quando o botão é pressionado, o código sai do laço e liga os LEDs

```
//Ligar LEDS enquanto o
//botão estiver pressionado
```

```
#include <msp430g2553.h>
```

```
#define BTN BIT2
```

```
#define LED1 BIT0
```

```
#define LED2 BIT6
```

```
void main(void){
```

```
    WDTCTL = WDTPW | WDTHOLD;
```

```
    P1OUT = 0;
```

```
    P1DIR = LED1 + LED2;
```

```
    for(;;)
```

```
    {
```

```
        while((P1IN & BTN) != 0){}
```

```
        P1OUT |= LED1 + LED2;
```

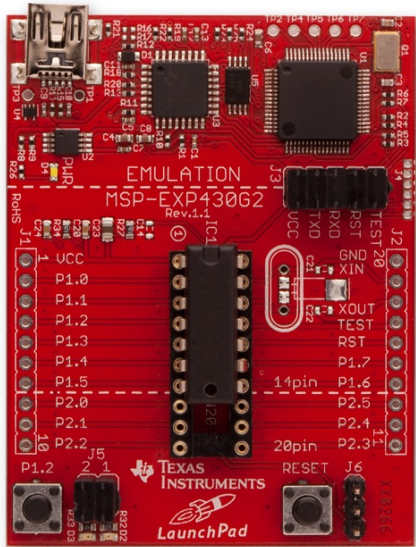
```
        while((P1IN & BTN) == 0){}
```

```
        P1OUT &= ~(LED1 + LED2);
```

```
    }
```

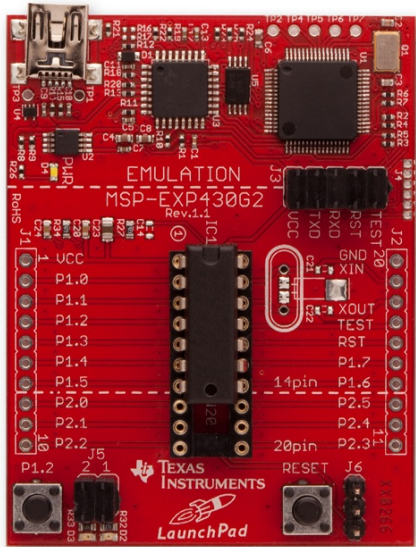
```
}
```





Código fica preso neste  
laço enquanto o botão é  
pressionado

```
//Ligar LEDs enquanto o
//botão estiver pressionado
#include <msp430g2553.h>
#define BTN BIT2
#define LED1 BIT0
#define LED2 BIT6
void main(void){
    WDTCTL = WDTPW | WDTHOLD;
    P1OUT = 0;
    P1DIR = LED1 + LED2;
    for(;;)
    {
        while((P1IN & BTN) != 0){
            P1OUT |= LED1 + LED2;
            while((P1IN & BTN) == 0){
                P1OUT &= ~(LED1 + LED2);
            }
        }
    }
}
```



```
//Ligar LEDS enquanto o  
//botão estiver pressionado  
#include <msp430g2553.h>  
#define BTN BIT2  
#define LED1 BIT0  
#define LED2 BIT6
```

```
void main(void){  
    WDTCTL = WDTPW | WDTHOLD;  
    P1OUT = 0;  
    P1DIR = LED1 + LED2;  
    for(;;)  
    {  
        while((P1IN & BTN) != 0){  
            P1OUT |= LED1 + LED2;  
            while((P1IN & BTN) == 0){  
                P1OUT &= ~(LED1 + LED2);  
            }  
        }  
    }  
}
```

Quando o botão é solto,  
o código sai do laço e  
desliga os LEDs