Precision Internal Oscillator (PIOSC)

- 16 MHz ± 3%

Main Oscillator (MOSC) using…

- An external single-ended clock source
- An external crystal

Internal 30 kHz Oscillator

- 30 kHz ± 50%
- Intended for use during Deep-Sleep power-saving modes

Hibernation Module Clock Source

- 32,768Hz crystal
- Intended to provide the system with a real-time clock source

The CPU can be driven by any of the fundamental clocks …

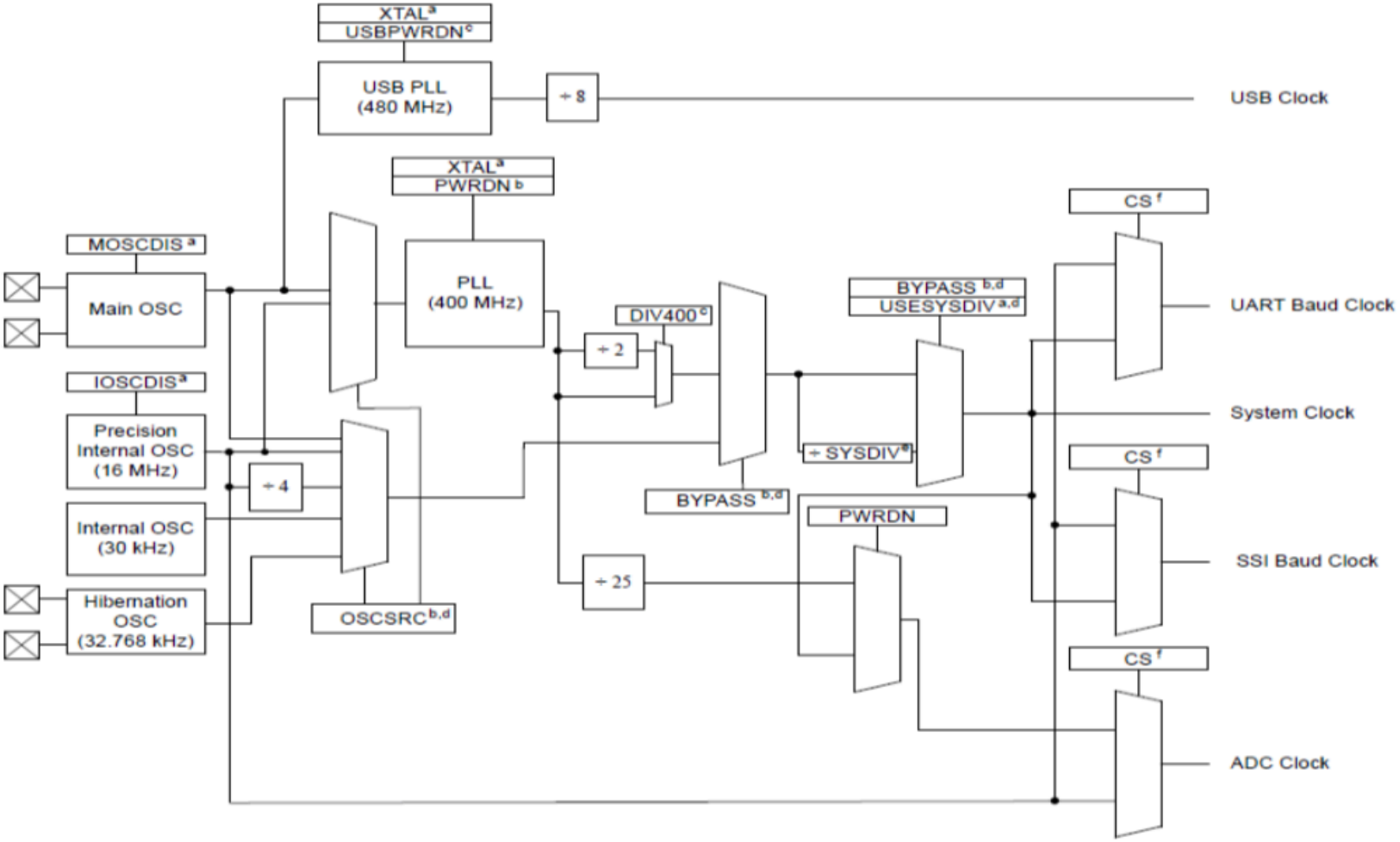
- Internal 16 MHz
- Main
- Internal 30 kHz
- External Real-Time

- Plus -

- The internal PLL (400 MHz)
- The internal 16MHz oscillator divided by four (4MHz ± 3%)

Clock Source	Drive PLL?	Used as SysClk?
Internal 16MHz	Yes	Yes
Internal 16Mhz/4	No	Yes
Main Oscillator	Yes	Yes
Internal 30 kHz	No	Yes
Hibernation Module	No	Yes
PLL	-	Yes

Tiva C Series Clock Tree



- **Any GPIO can be an interrupt:**
  - Edge-triggered on rising, falling or both
  - Level-sensitive on high or low values
- **Can directly initiate an ADC sample sequence or µDMA transfer**
- **Toggle rate up to the CPU clock speed on the Advanced High-Performance Bus. ½ CPU clock speed on the Standard.**
- **5V tolerant in input configuration**
- **Programmable Drive Strength (2, 4, 8mA or 8mA with slew rate control)**
- **Programmable weak pull-up, pull-down, and open drain**
- **Pin state can be retained during Hibernation mode**

Critical Function GPIO Protection

- **Six pins on the device are protected against accidental programming:**
  - PC3,2,1 & 0: JTAG/SWD
  - PD7 & PF0: NMI
- **Any write to the following registers for these pins will not be stored unless the GPIOLOCK register has been unlocked:**
  - GPIO Alternate Function Select register
  - GPIO Pull Up or Pull Down select registers
  - GPIO Digital Enable register
- **The following sequence will unlock the GPIOLOCK register for PF0 using direct register programming:**

```
HWREG(GPIO_PORTF_BASE + GPIO_O_LOCK) = GPIO_LOCK_KEY;
HWREG(GPIO_PORTF_BASE + GPIO_O_CR) |= 0x01;
HWREG(GPIO_PORTF_BASE + GPIO_O_LOCK) = 0;
```

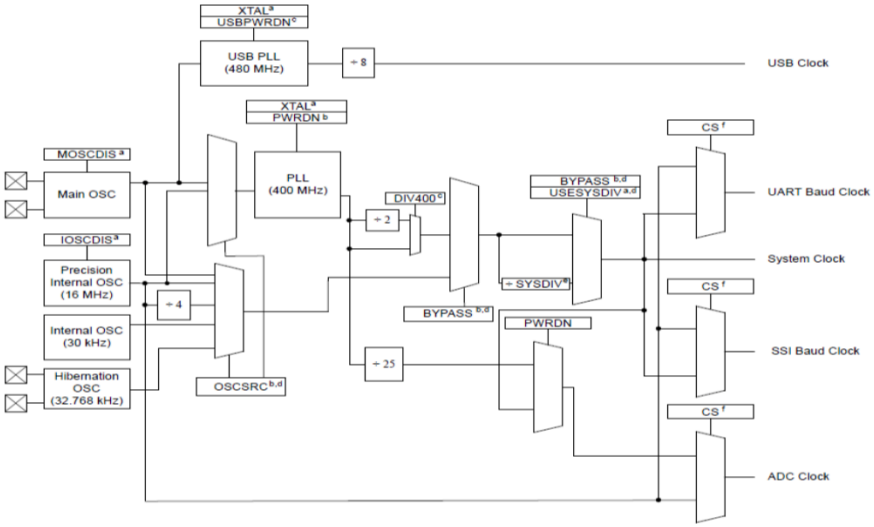
- **Reading the GPIOLOCK register returns it to lock status**

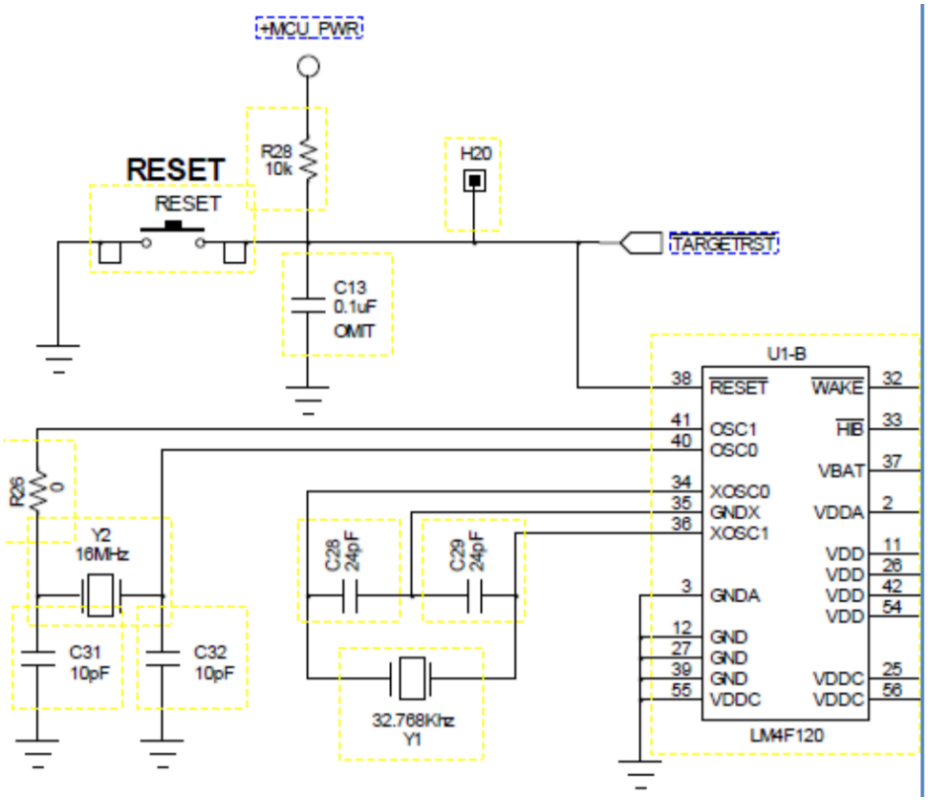
Primo Programma
[richiamo linguaggio C]

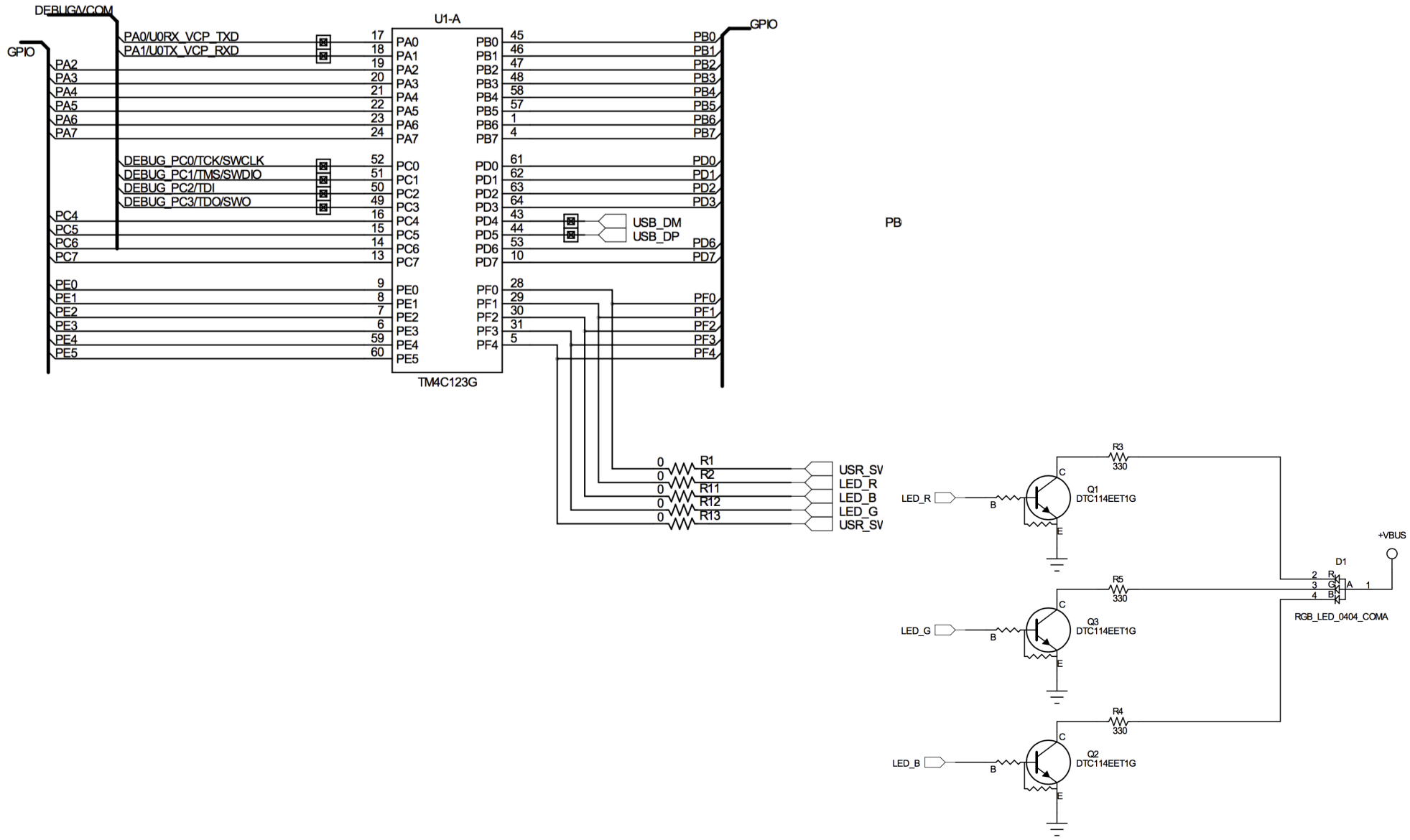
```c
#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/sysctl.h"
#include "driverlib/gpio.h"
uint8_t ui8PinData=2;
```

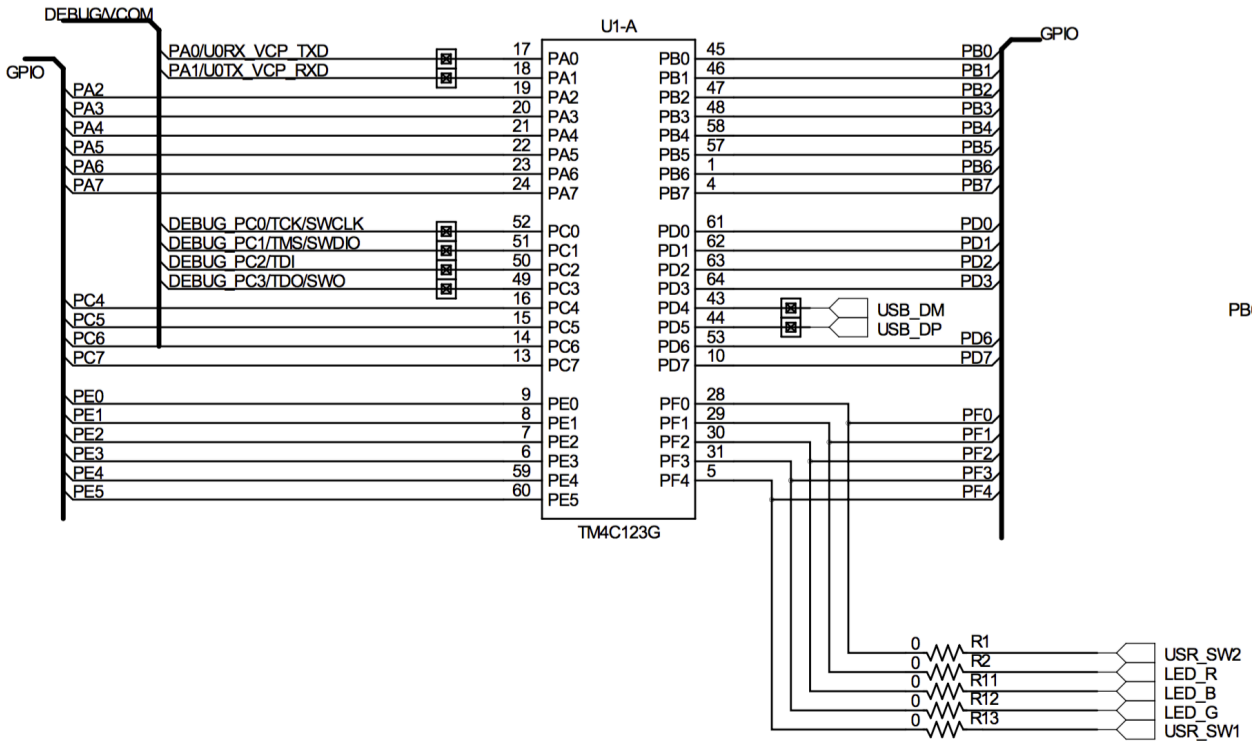
```
int main(void)
{
    SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);
```

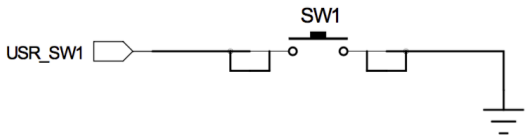
Tiva C Series Clock Tree

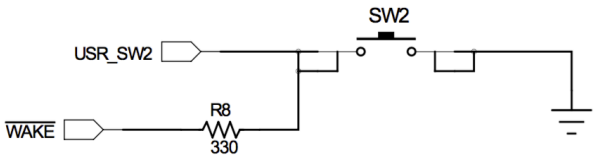





DEBUGA/COM
GPIO
U1-A
PA0/U0RX_VCP_TXD
PA1/U0TX_VCP_RXD
DEBUG_PC0/TCK/SWCLK
DEBUG_PC1/TMS/SWDIO
DEBUG_PC2/TDI
DEBUG_PC3/TDO/SWO
USB_DM
USB_DP
TM4C123G
R1
R2
R11
R12
R13
USR_SW
LED_R
LED_B
LED_G
USR_SW
R3
R5
R4
330
Q1
Q3
Q2
DTC114EET1G
+VBUS
D1
RGB_LED_0404_COMA


DEBUG/COM
GPIO
U1-A
PA0/U0RX_VCP_TXD
PA1/U0TX_VCP_RXD
PA2
PA3
PA4
PA5
PA6
PA7
DEBUG_PC0/TCK/SWCLK
DEBUG_PC1/TMS/SWDIO
DEBUG_PC2/TDI
DEBUG_PC3/TDO/SWO
PC4
PC5
PC6
PC7
PE0
PE1
PE2
PE3
PE4
PE5
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
PF0
PF1
PF2
PF3
PF4
USB_DM
USB_DP
TM4C123G
R1
R2
R11
R12
R13
USR_SW2
LED_R
LED_B
LED_G
USR_SW1


SW1
USR_SW1


SW2
USR_SW2
WAKE
R8
330

```
SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
```

```
GPIO Port A (APB): 0x4000.4000
GPIO Port A (AHB): 0x4005.8000
GPIO Port B (APB): 0x4000.5000
GPIO Port B (AHB): 0x4005.9000
GPIO Port C (APB): 0x4000.6000
GPIO Port C (AHB): 0x4005.A000
GPIO Port D (APB): 0x4000.7000
GPIO Port D (AHB): 0x4005.B000
GPIO Port E (APB): 0x4002.4000
GPIO Port E (AHB): 0x4005.C000
GPIO Port F (APB): 0x4002.5000
GPIO Port F (AHB): 0x4005.D000
```

```
SysCtlClockSet(SYSCTL_SYSDIV_5|SYSCTL_USE_PLL|SYSCTL_XTAL_16MHZ|SYSCTL_OSC_MAIN);
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
```

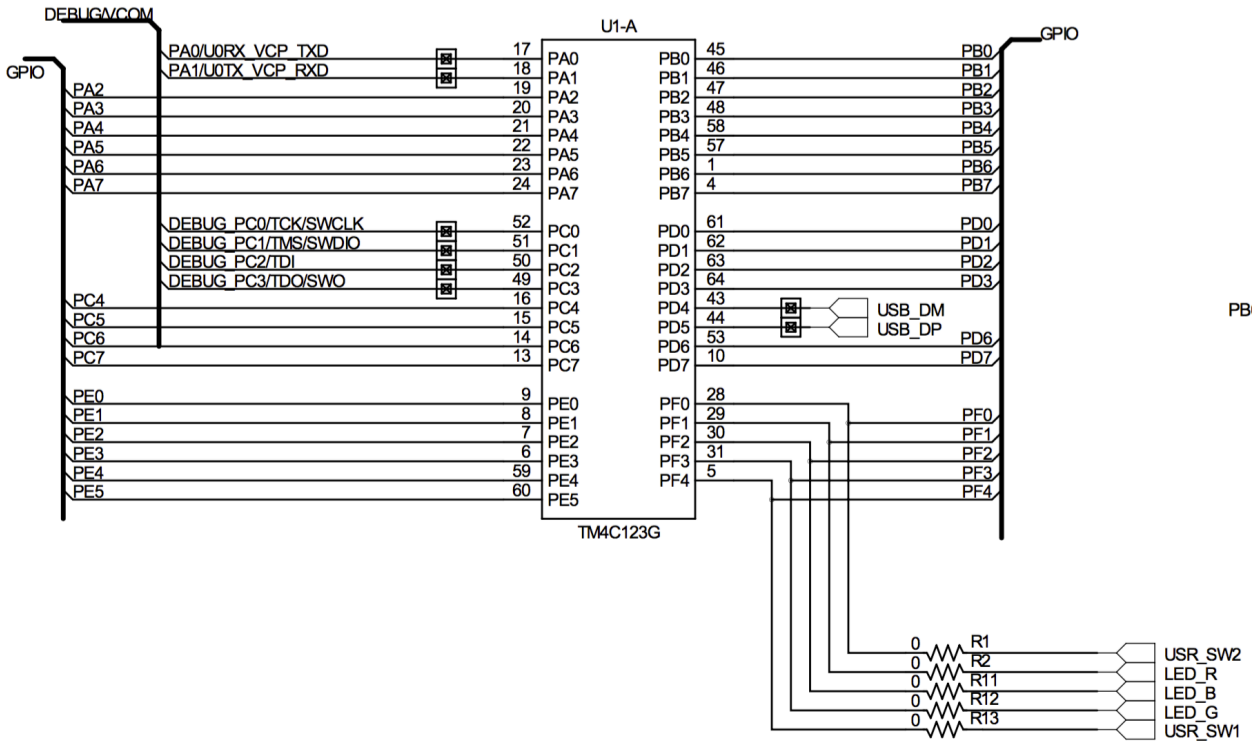
```
GPIO Port A (APB): 0x4000.4000
GPIO Port A (AHB): 0x4005.8000
GPIO Port B (APB): 0x4000.5000
GPIO Port I
GPIO Port (
GPIO Port (
GPIO Port I
GPIO Port I
GPIO Port E (APB): 0x4002.4000
GPIO Port E (AHB): 0x4005.C000
GPIO Port F (APB): 0x4002.5000
GPIO Port F (AHB): 0x4005.D000
```

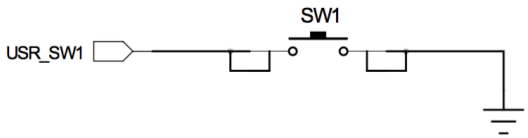
```
#define I2C3_BASE          0x40023000  // I2C3
#define GPIO_PORTE_BASE    0x40024000  // GPIO Port E
#define GPIO_PORTF_BASE    0x40025000  // GPIO Port F
#define GPIO_PORTG_BASE    0x40026000  // GPIO Port G
#define GPIO_PORTH_BASE    0x40027000  // GPIO Port H
```

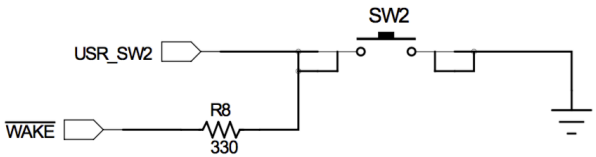
```
GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE,GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3);
```

```
while(1)
  {

    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1| GPIO_PIN_2| GPIO_PIN_3, ui8PinData);
    SysCtlDelay(2000000);

    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3, 0x00);
    SysCtlDelay(2000000);

    if(ui8PinData==8) {ui8PinData=2;} else {ui8PinData=ui8PinData*2;}
  }
return 0;
```

Utilizziamo SW1 e SW2


DEBUG/COM
GPIO
U1-A
PA0/U0RX_VCP_TXD
PA1/U0TX_VCP_RXD
PA2
PA3
PA4
PA5
PA6
PA7
DEBUG_PC0/TCK/SWCLK
DEBUG_PC1/TMS/SWDIO
DEBUG_PC2/TDI
DEBUG_PC3/TDO/SWO
PC4
PC5
PC6
PC7
PE0
PE1
PE2
PE3
PE4
PE5
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
PF0
PF1
PF2
PF3
PF4
USB_DM
USB_DP
TM4C123G
R1
R2
R11
R12
R13
USR_SW2
LED_R
LED_B
LED_G
USR_SW1


SW1
USR_SW1


SW2
USR_SW2
WAKE
R8
330

```c
#include "tm4c123gh6pm.h"

void PortF_Init(void){ volatile unsigned long delay;
    GPIO_PORTF_LOCK_R = 0x4C4F434B;   // 2) unlock GPIO Port F
    GPIO_PORTF_CR_R = 0x1F;           // allow changes to PF4-0
}
```

```c
#include "tm4c123gh6pm.h"

void PortF_Init(void){ volatile unsigned long delay;
    GPIO_PORTF_LOCK_R = 0x4C4F434B;   // 2) unlock GPIO Port F
    GPIO_PORTF_CR_R = 0x1F;           // allow changes to PF4-0
}

```

```c
PortF_Init();

GPIOPinTypeGPIOInput(GPIO_PORTF_BASE, GPIO_PIN_0|GPIO_PIN_4);
GPIOPadConfigSet(GPIO_PORTF_BASE,GPIO_PIN_0|GPIO_PIN_4,GPIO_STRENGTH_2MA,GPIO_PIN_TYPE_STD_WPU);
```

```c
#include "tm4c123gh6pm.h"

void PortF_Init(void){ volatile unsigned long delay;
    GPIO_PORTF_LOCK_R = 0x4C4F434B;   // 2) unlock GPIO Port F
    GPIO_PORTF_CR_R = 0x1F;           // allow changes to PF4-0
}

```

```c
PortF_Init();

GPIOPinTypeGPIOInput(GPIO_PORTF_BASE, GPIO_PIN_0|GPIO_PIN_4);
GPIOPadConfigSet(GPIO_PORTF_BASE,GPIO_PIN_0|GPIO_PIN_4,GPIO_STRENGTH_2MA,GPIO_PIN_TYPE_STD_WPU);
```

```c
iVal4=GPIOPinRead(GPIO_PORTF_BASE,GPIO_PIN_4);
```

J1 Pin	GPIO	Analog Function	On-board Function	Tiva C Series MCU Pin	GPIOPCTL Register Setting										
		GPIO AMSEL			1	2	3	4	5	6	7	8	9	14	15
1.01	3.3 V														
1.02	PB5	AIN11	–	57	–	SSI2Fss	–	M0PWM3	–	–	T1CCP1	CAN0Tx	–	–	–
1.03	PB0	USB0ID	–	45	U1Rx	–	–	–	–	–	T2CCP0	–	–	–	–
1.04	PB1	USB0VBUS	–	46	U1Tx	–	–	–	–	–	T2CCP1	–	–	–	–
1.05	PE4	AIN9	–	59	U5Rx	–	I2C2SCL	M0PWM4	M1PWM2	–	–	CAN0Rx	–	–	–
1.06	PE5	AIN8	–	60	U5Tx	–	I2C2SDA	M0PWM5	M1PWM3	–	–	CAN0Tx	–	–	–
1.07	PB4	AIN10	–	58	–	SSI2Clk	–	M0PWM2	–	–	T1CCP0	CAN0Rx	–	–	–
1.08	PA5	–	–	22	–	SSI0Tx	–	–	–	–	–	–	–	–	–
1.09	PA6	–	–	23	–	–	I2C1SCL	–	M1PWM2	–	–	–	–	–	–
1.10	PA7	–	–	24	–	–	I2C1SDA	–	M1PWM3	–	–	–	–	–	–