CEC Continuing Education Center

DesignNews

IoT Device Prototyping with Microchip Curiosity Development Boards

Day 5:

Coding the PIC32CM MC Curiosity Nano Evaluation Kit Using Harmony v3

Sponsored by

Digi-Key ELECTRONICS

informa markets

Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

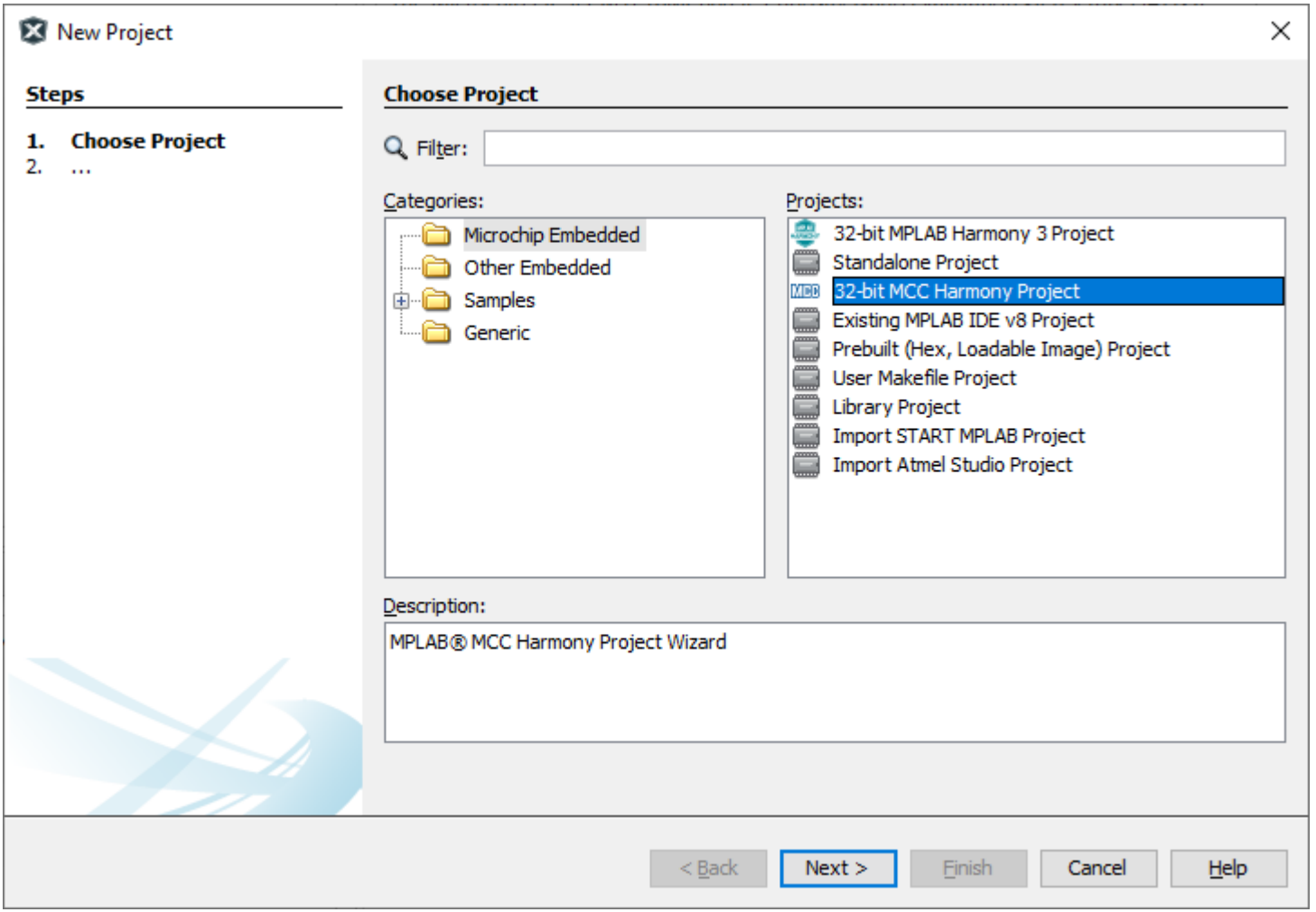
Fred Eady

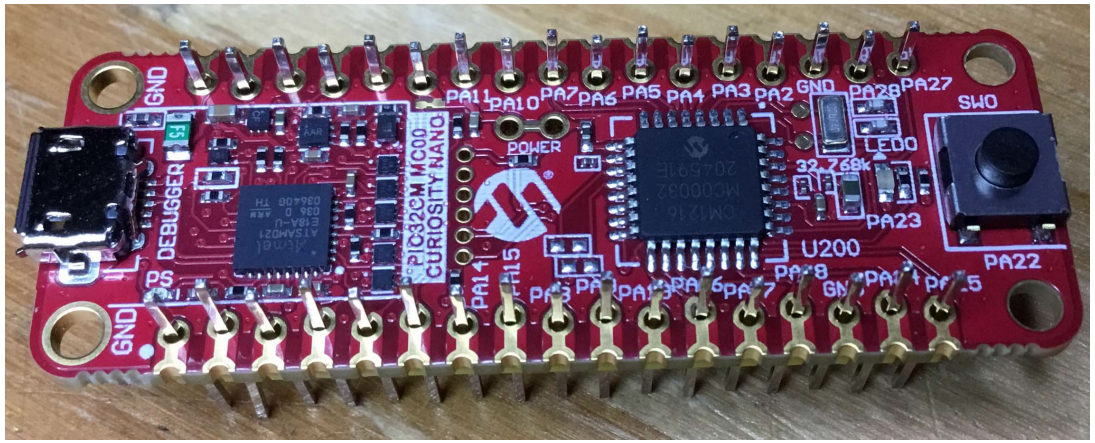
Visit ‘Lecturer Profile’ in your console for more details.

AGENDA

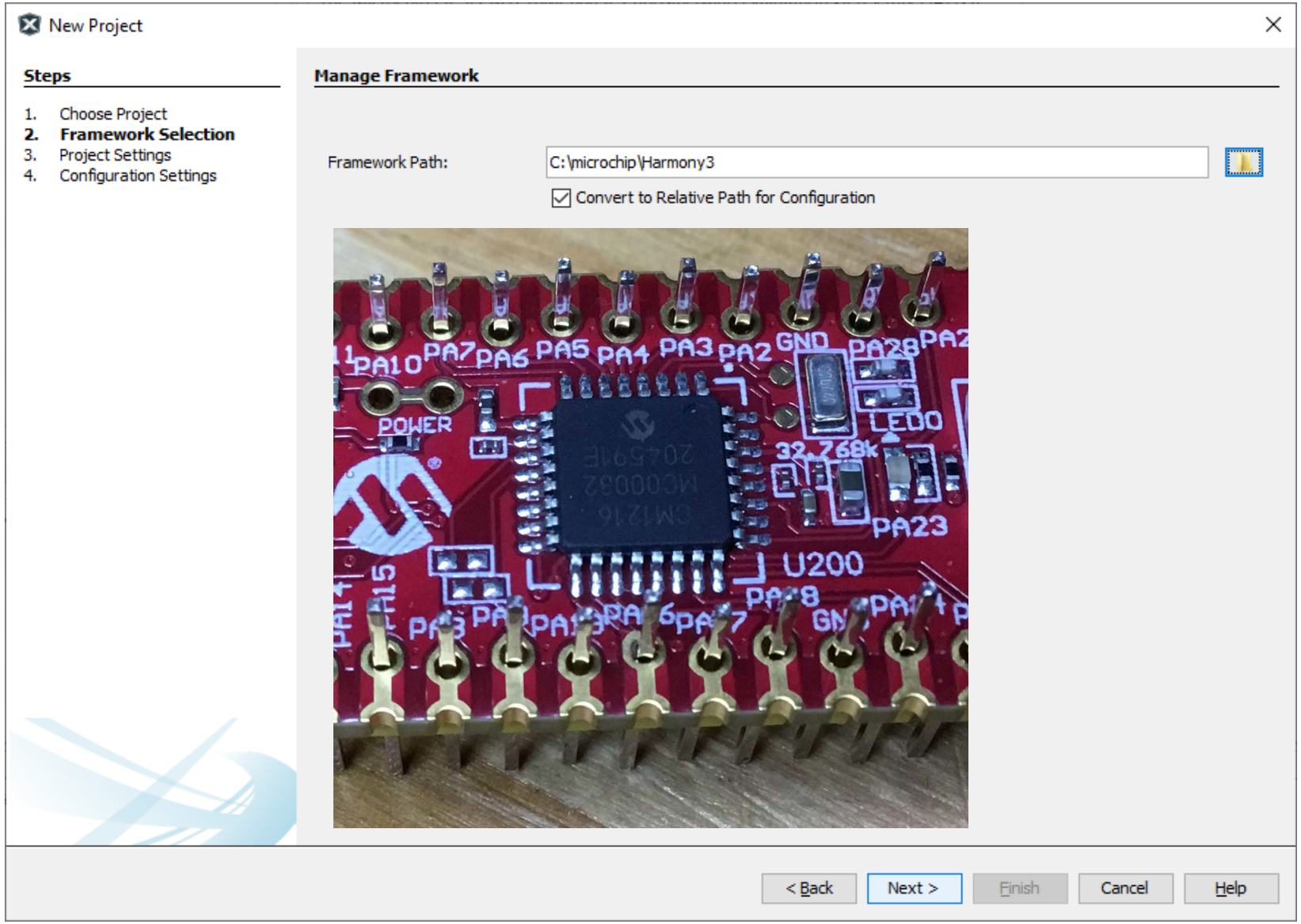
- Blink an LED the Harmony v3 Way
- I SPI
- Coding a Harmony v3-Generated USB CDC Port

Create a 32-bit MCC Harmony Project

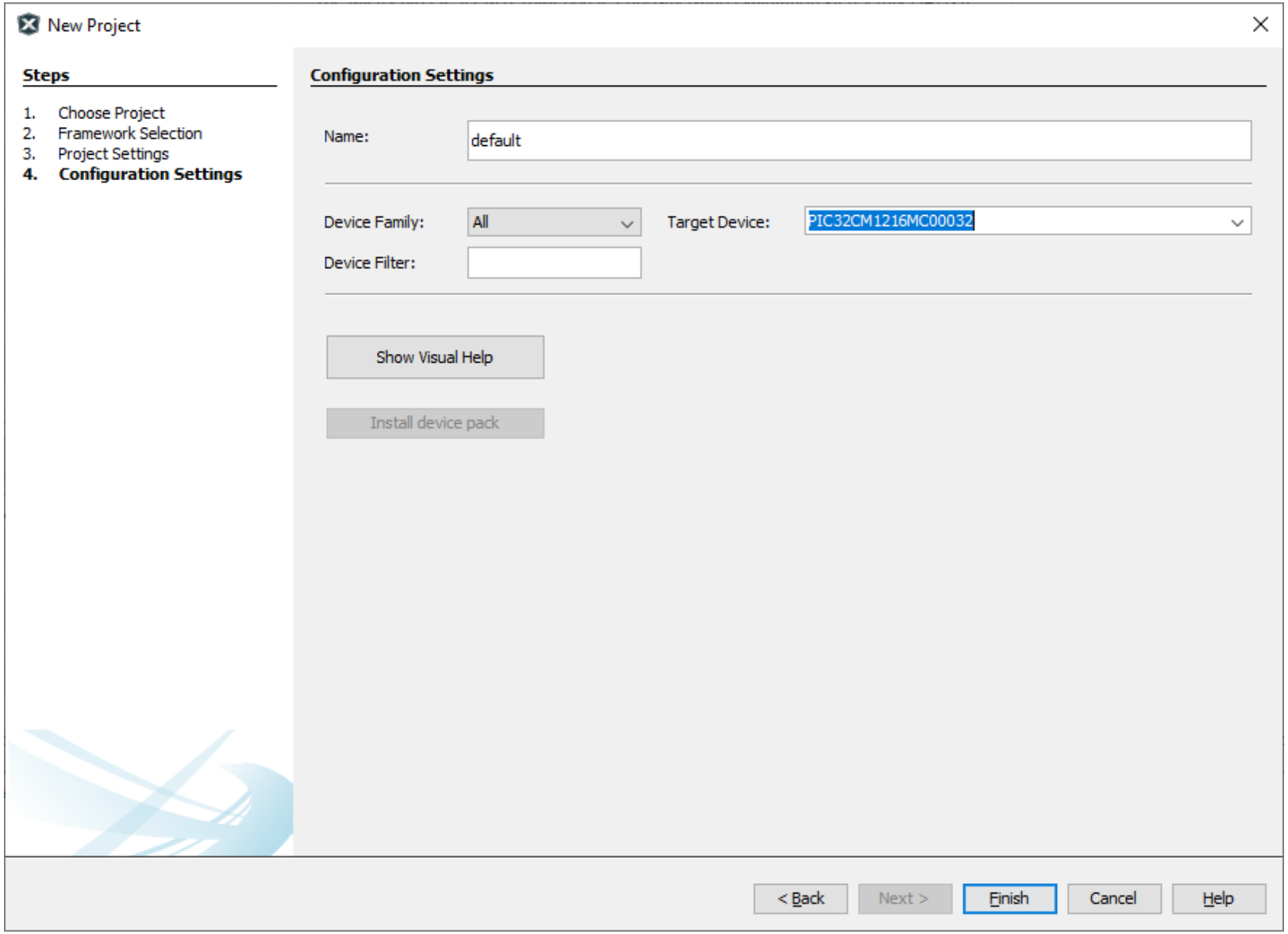


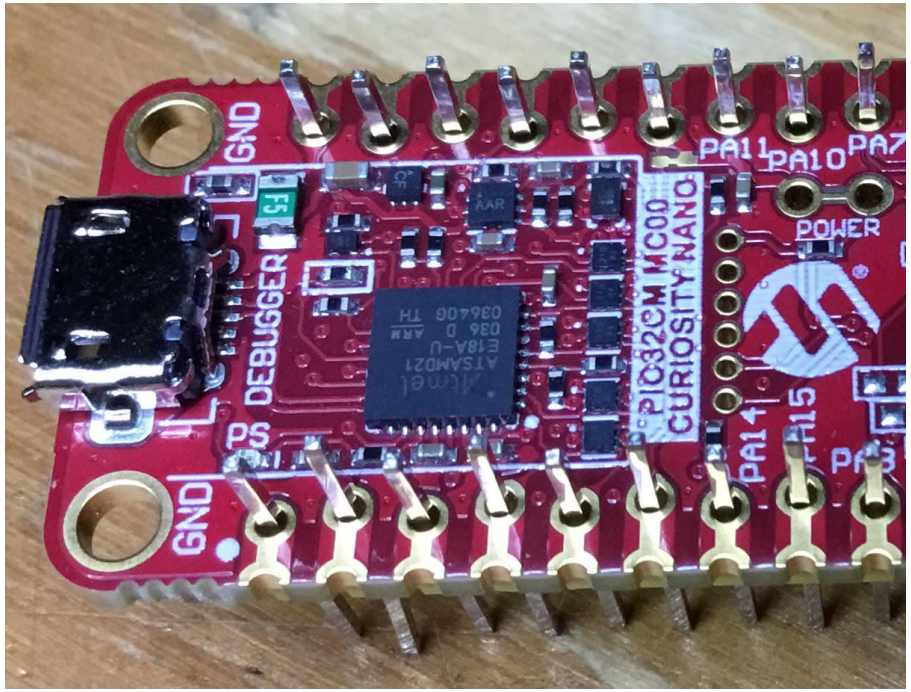


Define the Framework Path

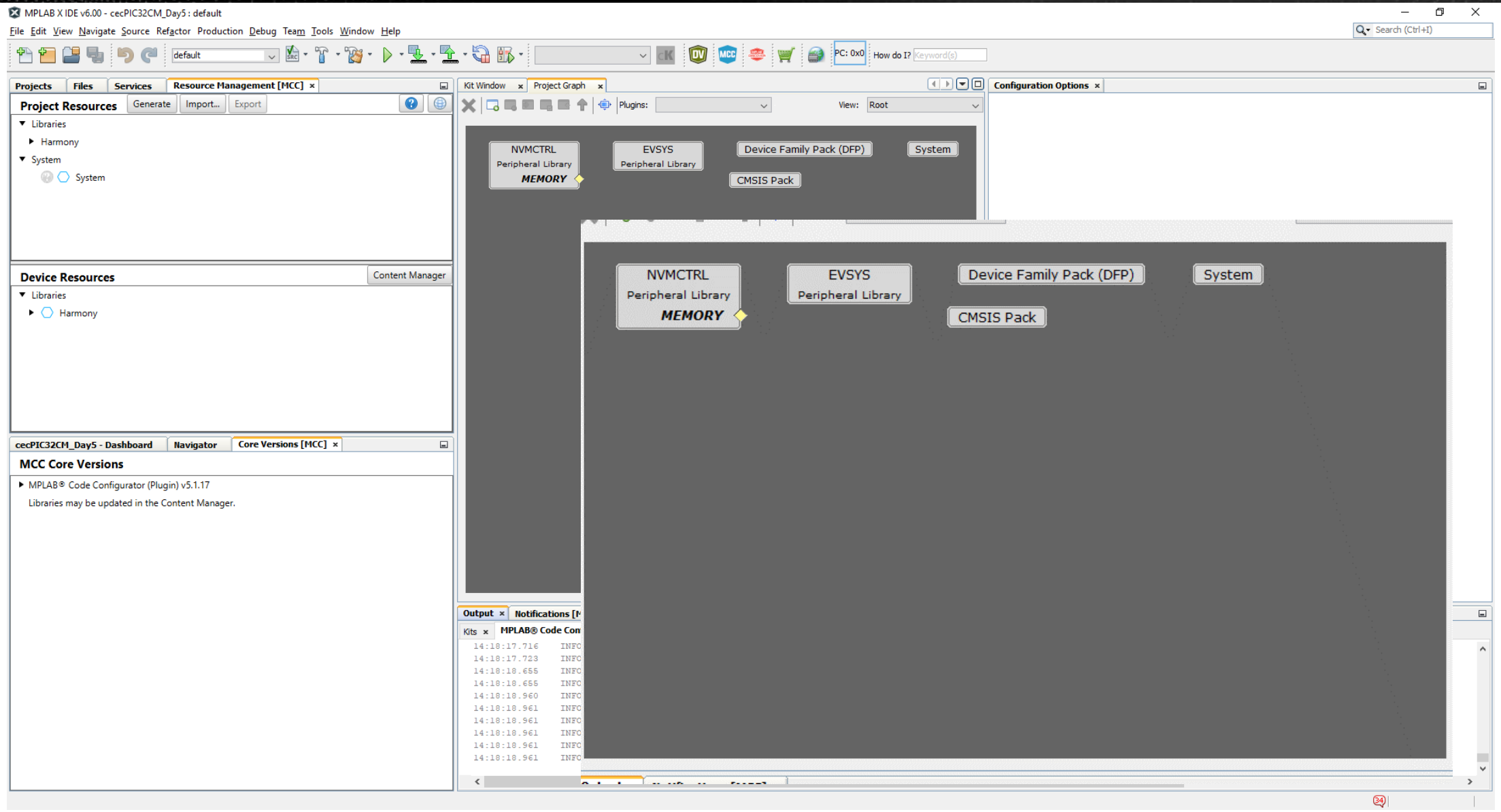


Specify the PIC32CM1216MC00032

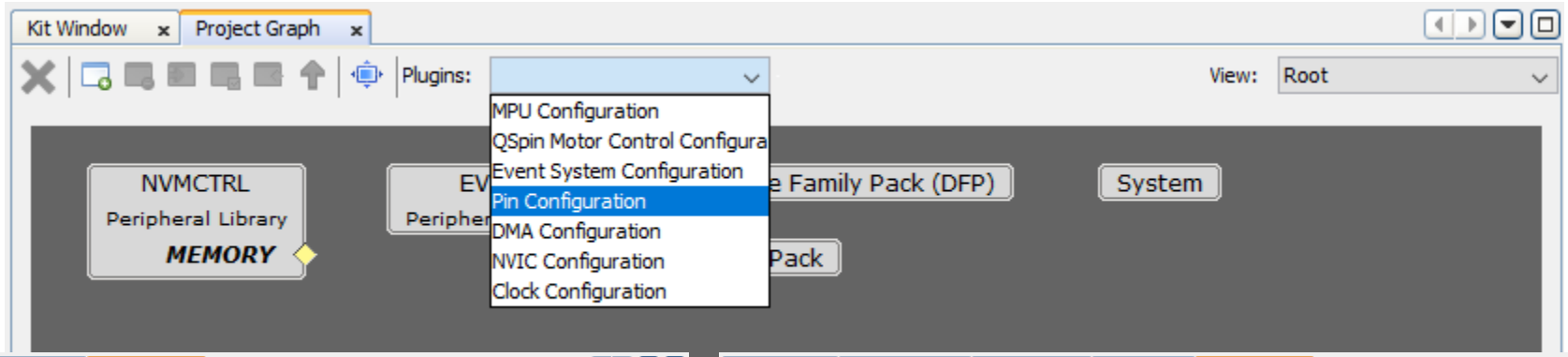


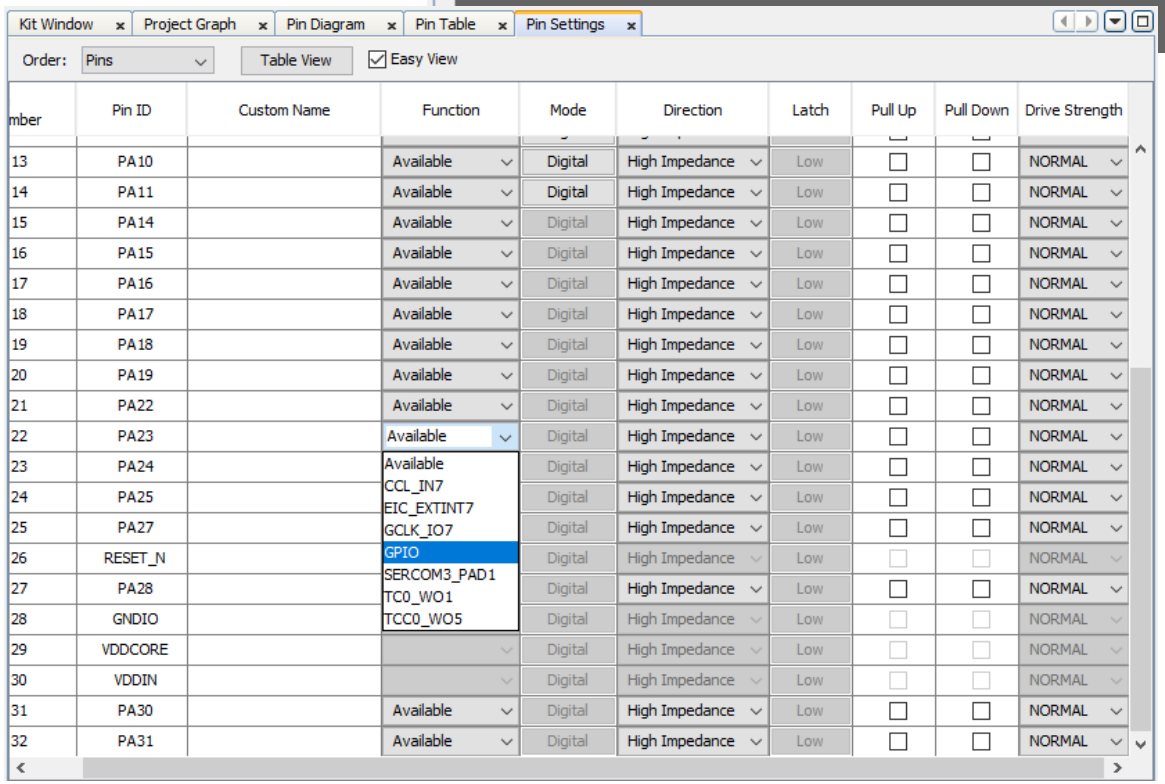


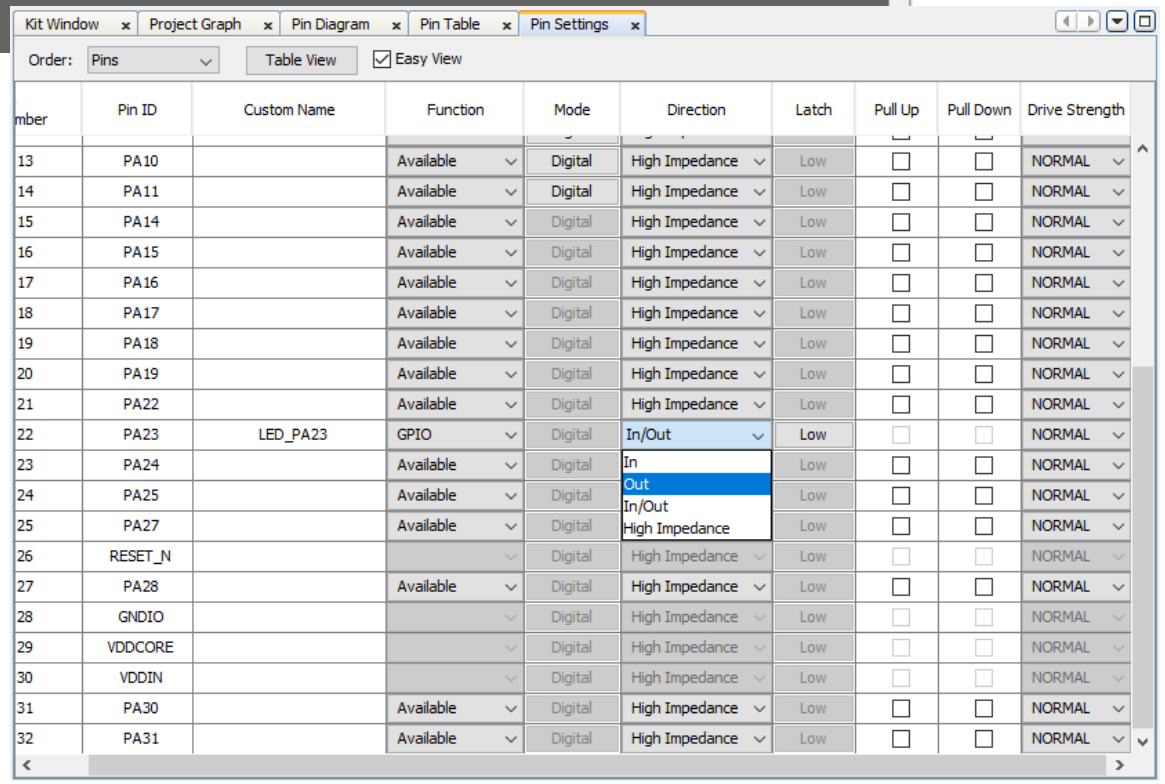
Project Graph



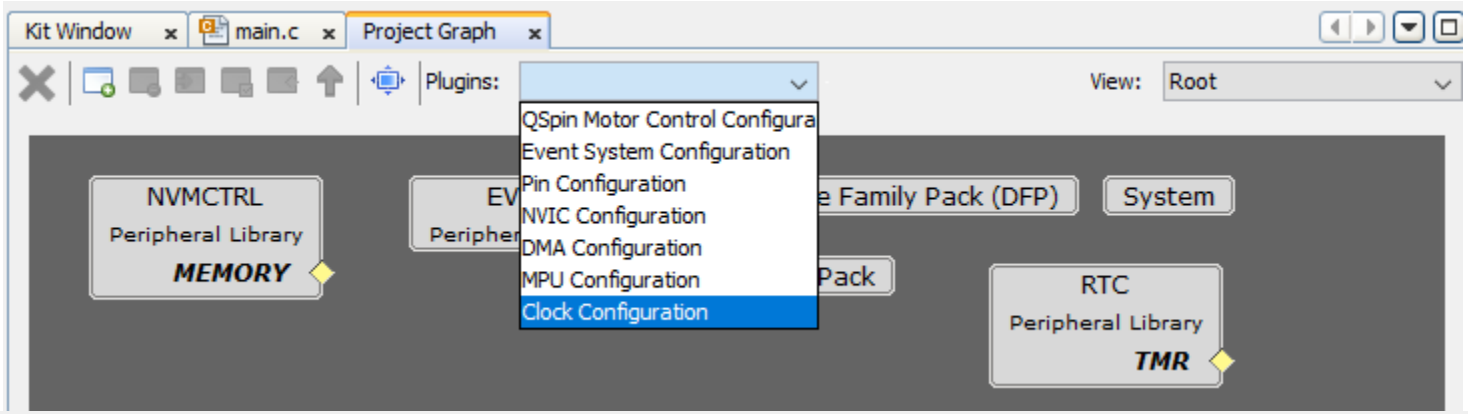
Add the LED to the Project

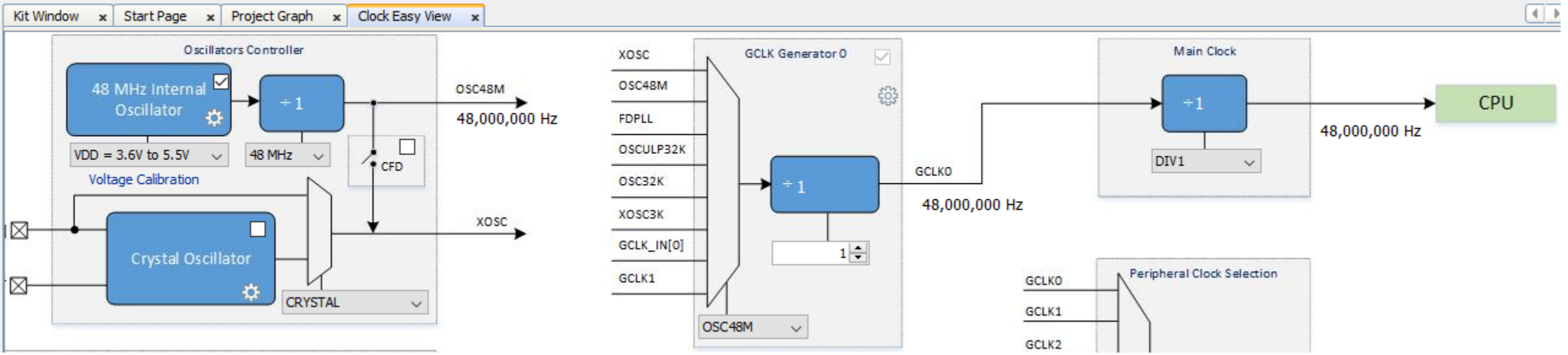




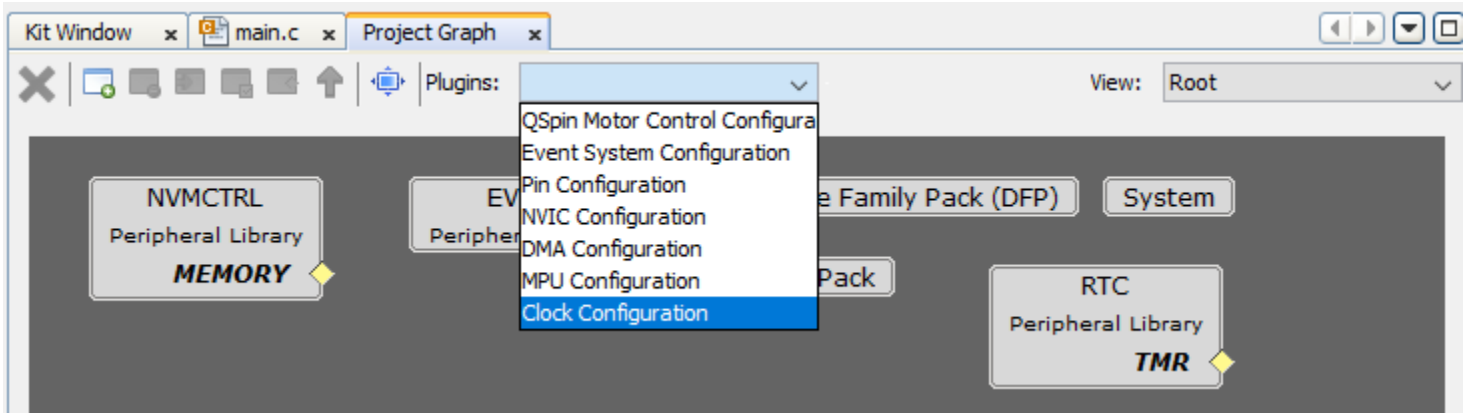


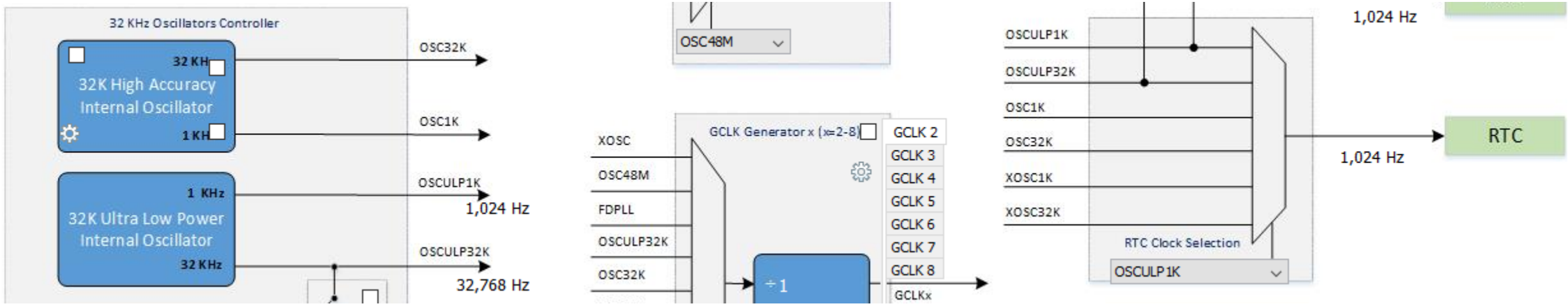
Check the Main Clock Configuration and Add the RTC





Check the RTC Clock Configuration

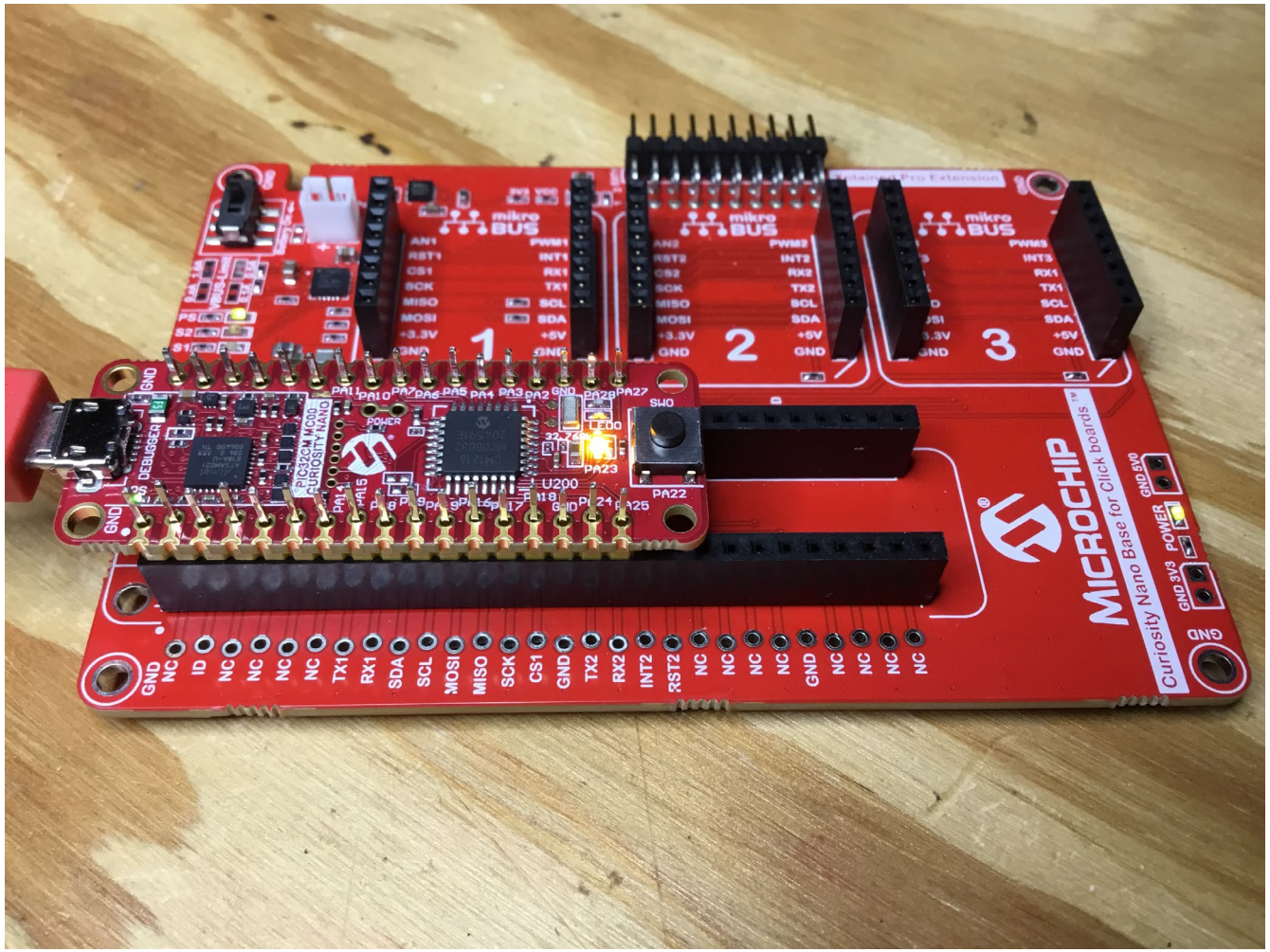




Configure the RTC Interrupt

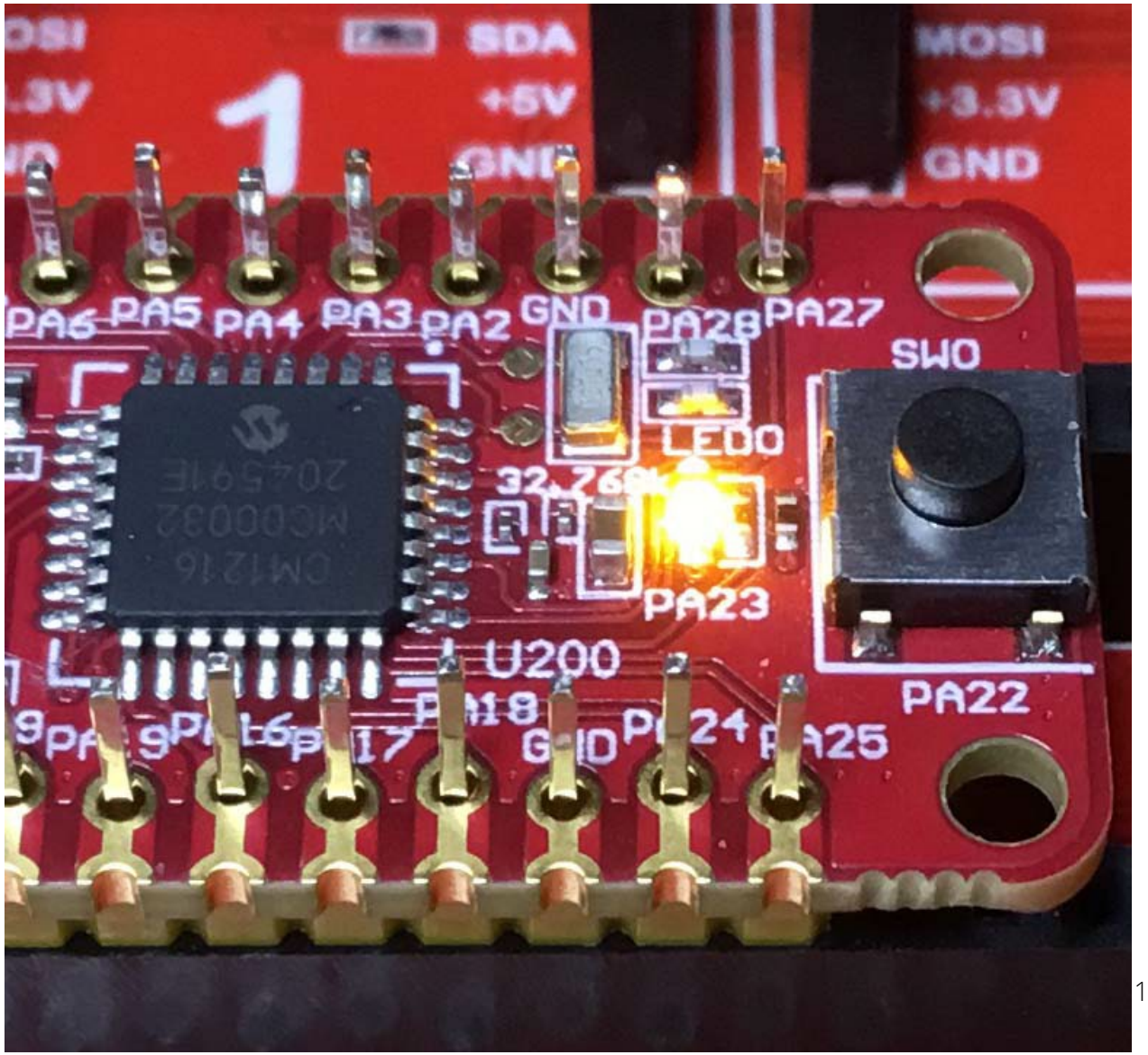
Configuration Options ×

- RTC
  - Hardware Settings
    - Generate Frequency Correction API ☐
    - RTC Count Sync Enable ☑
    - RTC Operation Mode: 32-bit Counter with Single 32-bit Compare
      - RTC MODE 0 Configuration
        - Enable Interrupts ? ☑
          - Periodic Interval 0 Interrupt Enable ☐
          - Periodic Interval 1 Interrupt Enable ☐
          - Periodic Interval 2 Interrupt Enable ☐
          - Periodic Interval 3 Interrupt Enable ☐
          - Periodic Interval 4 Interrupt Enable ☐
          - Periodic Interval 5 Interrupt Enable ☐
          - Periodic Interval 6 Interrupt Enable ☐
          - Periodic Interval 7 Interrupt Enable ☐
          - Compare 0 Interrupt Enable ☑
          - Overflow Interrupt Enable ☐
        - RTC Prescaler: DIV1
        - Compare Value: 0x 200
        - Clear on compare Match ☑
        - RTC EVENTS configuration

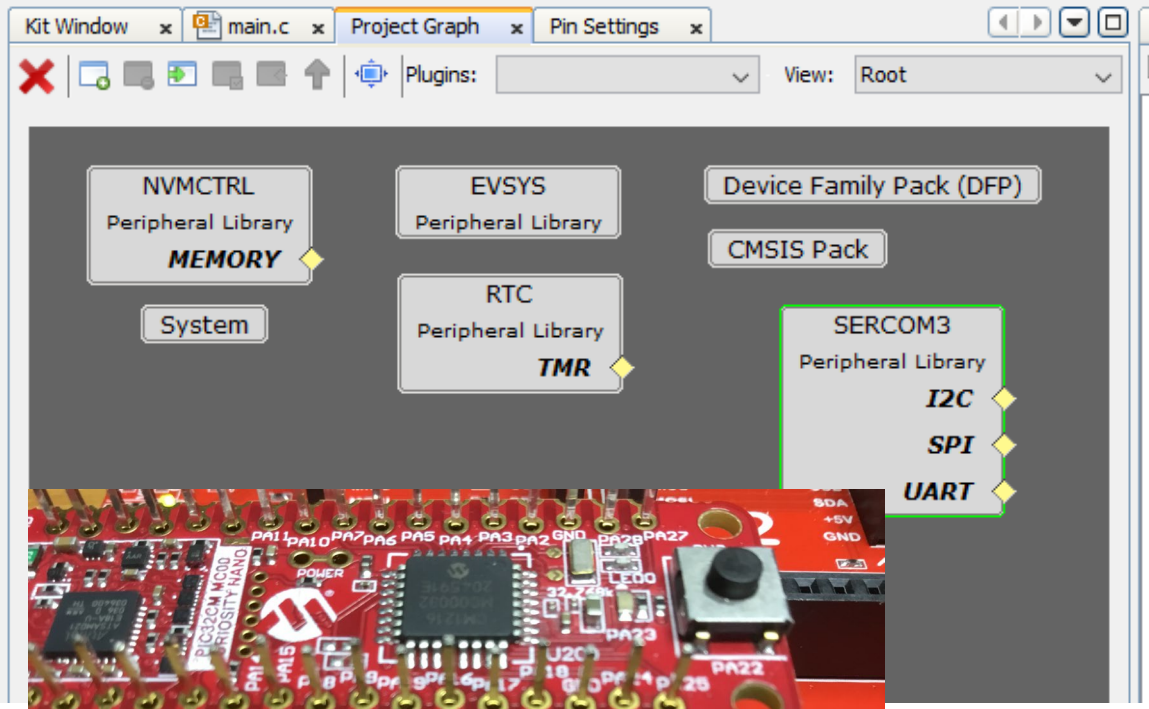


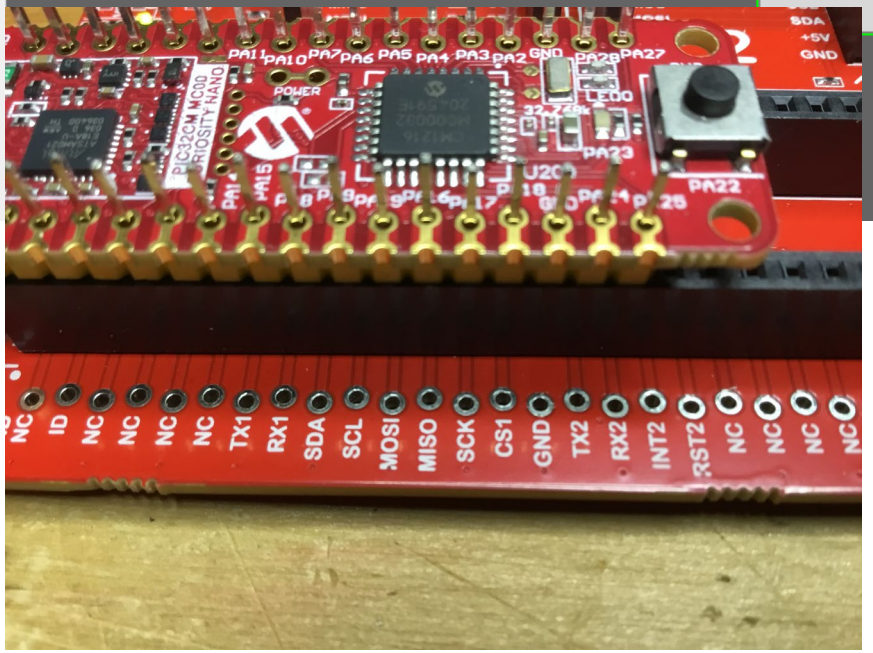
Generate and Run the Code

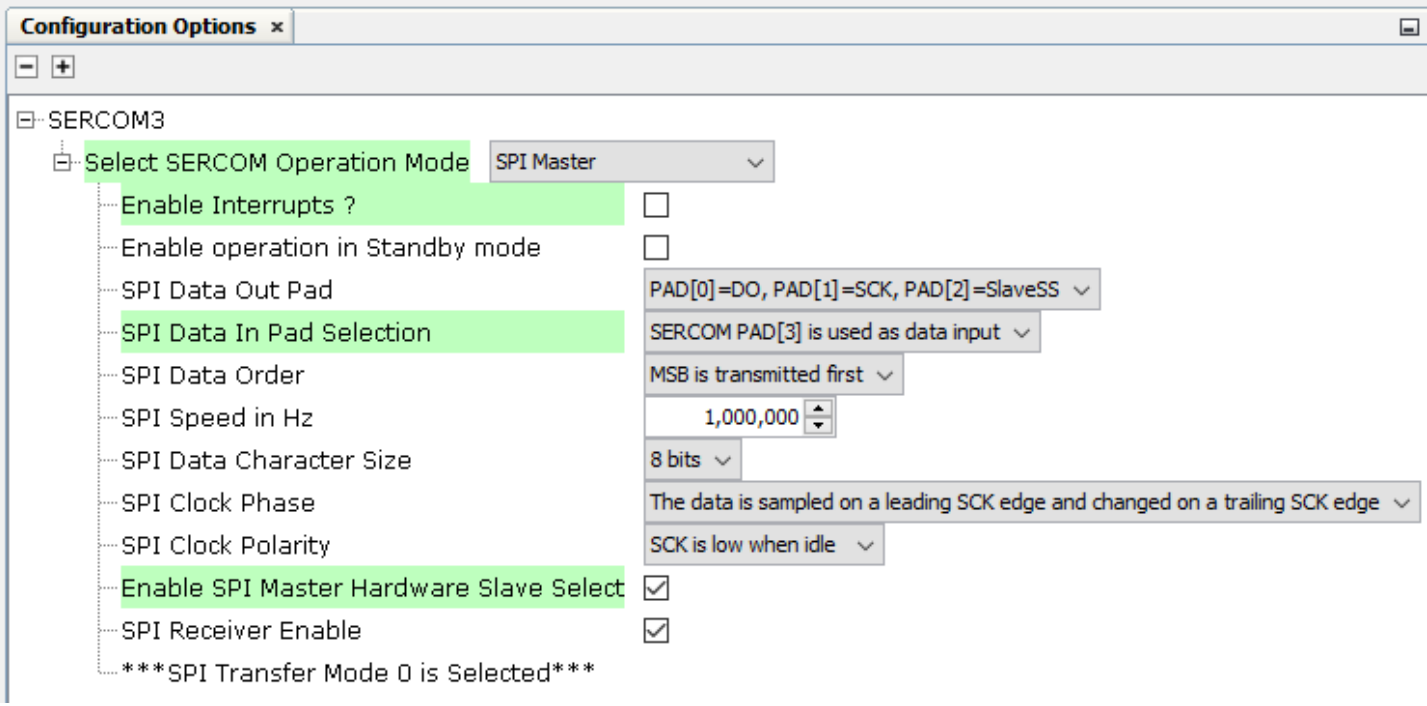
```
#include <stddef.h>                 // Defines NULL
#include <stdbool.h>                // Defines true
#include <stdlib.h>                 // Defines EXIT_FAILURE
#include "definitions.h"            // SYS function prototypes

static volatile bool rtcTick= false;
static void rtcEventHandler (RTC_TIMER32_INT_MASK intCause, uintptr_t context)
{
    if (intCause & RTC_MODE0_INTENSET_CMP0_Msk)
    {
        rtcTick    = true;
    }
}

int main ( void )
{
    /* Initialize all modules */
    SYS_Initialize ( NULL );
    RTC_Timer32CallbackRegister(rtcEventHandler, 0);
    RTC_Timer32Start();
    while ( true )
    {
        if (rtcTick == true)
        {
            rtcTick= false;
            LED_PA23_Toggle();
        }
    }

    /* Execution should not come here during normal operation */

    return ( EXIT_FAILURE );
}
```



Add the SPI Peripheral







Add the SPI Peripheral and SPI Application Code

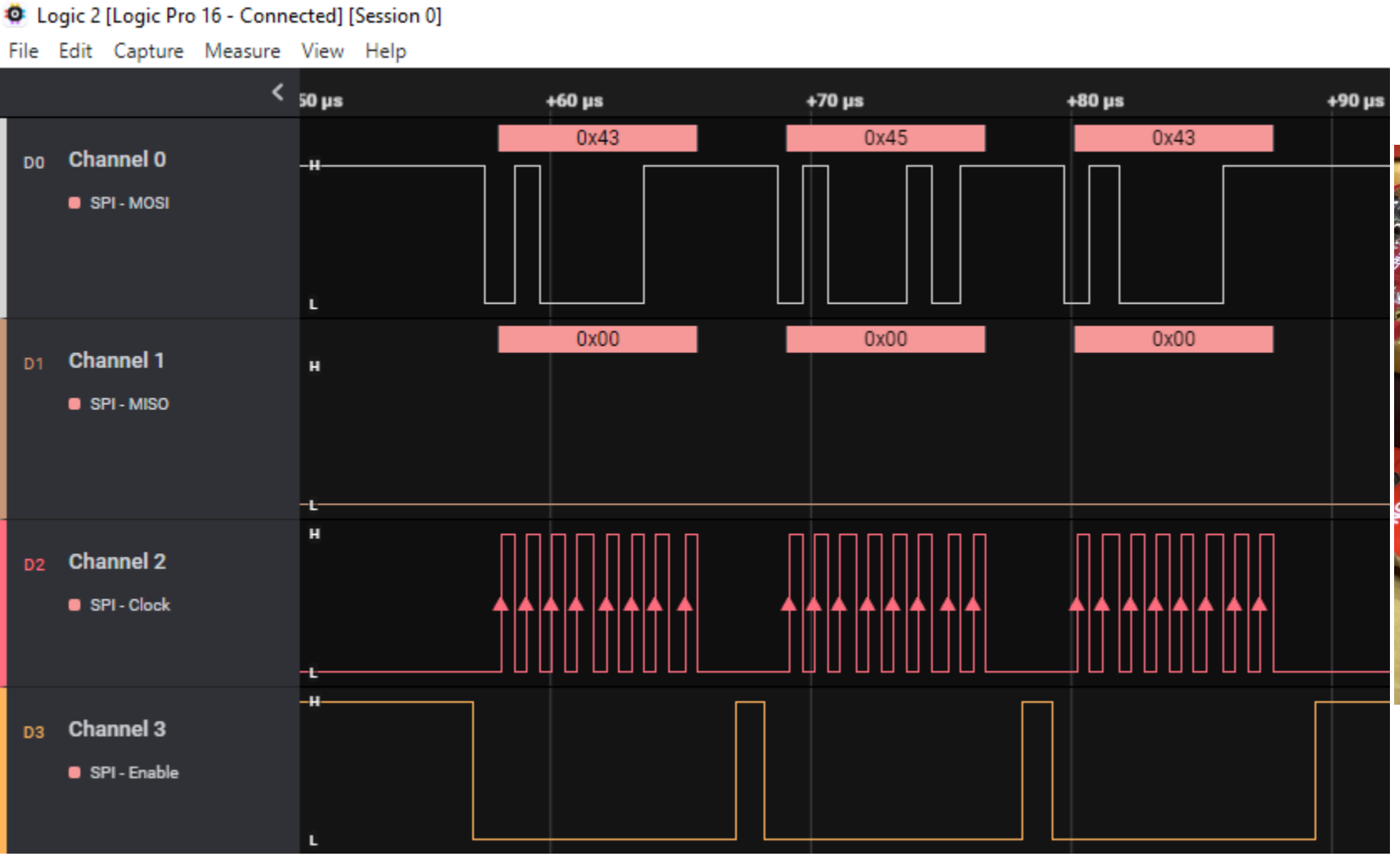
Kit Window | main.c | Project Graph | Pin Settings

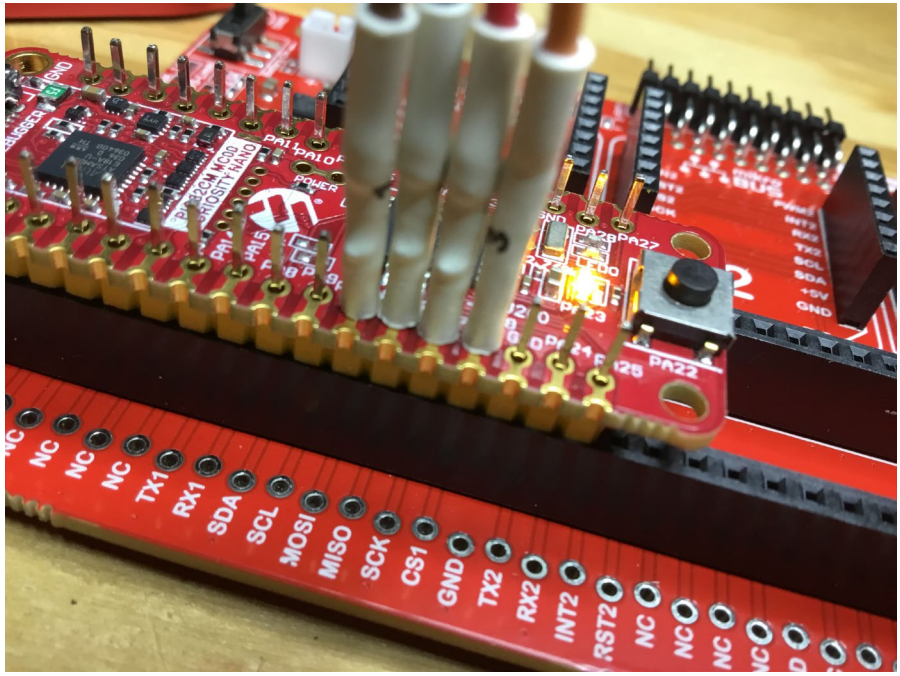
Order: Pins | Table View | Easy View

Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength
4	PA03		Available	Analog	High Impedance	Low			NORMAL
5	PA04		Available	Analog	High Impedance	Low			NORMAL
6	PA05		Available	Analog	High Impedance	Low			NORMAL
7	PA06		Available	Analog	High Impedance	Low			NORMAL
8	PA07		Available	Analog	High Impedance	Low			NORMAL
9	VDDANA			Digital	High Impedance	Low			NORMAL
10	GNDIO			Digital	High Impedance	Low			NORMAL
11	PA08		Available	Analog	High Impedance	Low			NORMAL
12	PA09		Available	Analog	High Impedance	Low			NORMAL
13	PA10		Available	Analog	High Impedance	Low			NORMAL
14	PA11		Available	Analog	High Impedance	Low			NORMAL
15	PA14		Available	Digital	High Impedance	Low			NORMAL
16	PA15		Available	Digital	High Impedance	Low			NORMAL
17	PA16		SERCOM3_PAD0	Digital	High Impedance	n/a			NORMAL
18	PA17		SERCOM3_PAD1	Digital	High Impedance	n/a			NORMAL
19	PA18		SERCOM3_PAD2	Digital	High Impedance	n/a			NORMAL
20	PA19		SERCOM3_PAD3	Digital	High Impedance	n/a			NORMAL
21	PA22		Available	Digital	High Impedance	Low			NORMAL
22	PA23	LED_PA23	GPIO	Digital	Out	Low			NORMAL
23	PA24		Available	Digital	High Impedance	Low			NORMAL
24	PA25		Available	Digital	High Impedance	Low			NORMAL

```
uint8_t spiTxBuf[8];
int main ( void )
{
    /* Initialize all modules */
    SYS_Initialize ( NULL );
    spiTxBuf[0] = 0x43;
    spiTxBuf[1] = 0x45;
    spiTxBuf[2] = 0x43;
    RTC_Timer32CallbackRegister(rtcEventHandler, 0);
    RTC_Timer32Start();
    while ( true )
    {
        if (rtcTick == true)
        {
            rtcTick= false;
            SERCOM3_SPI_Write(spiTxBuf, 0x03 );
            LED_PA23_Toggle();
        }
    }
```

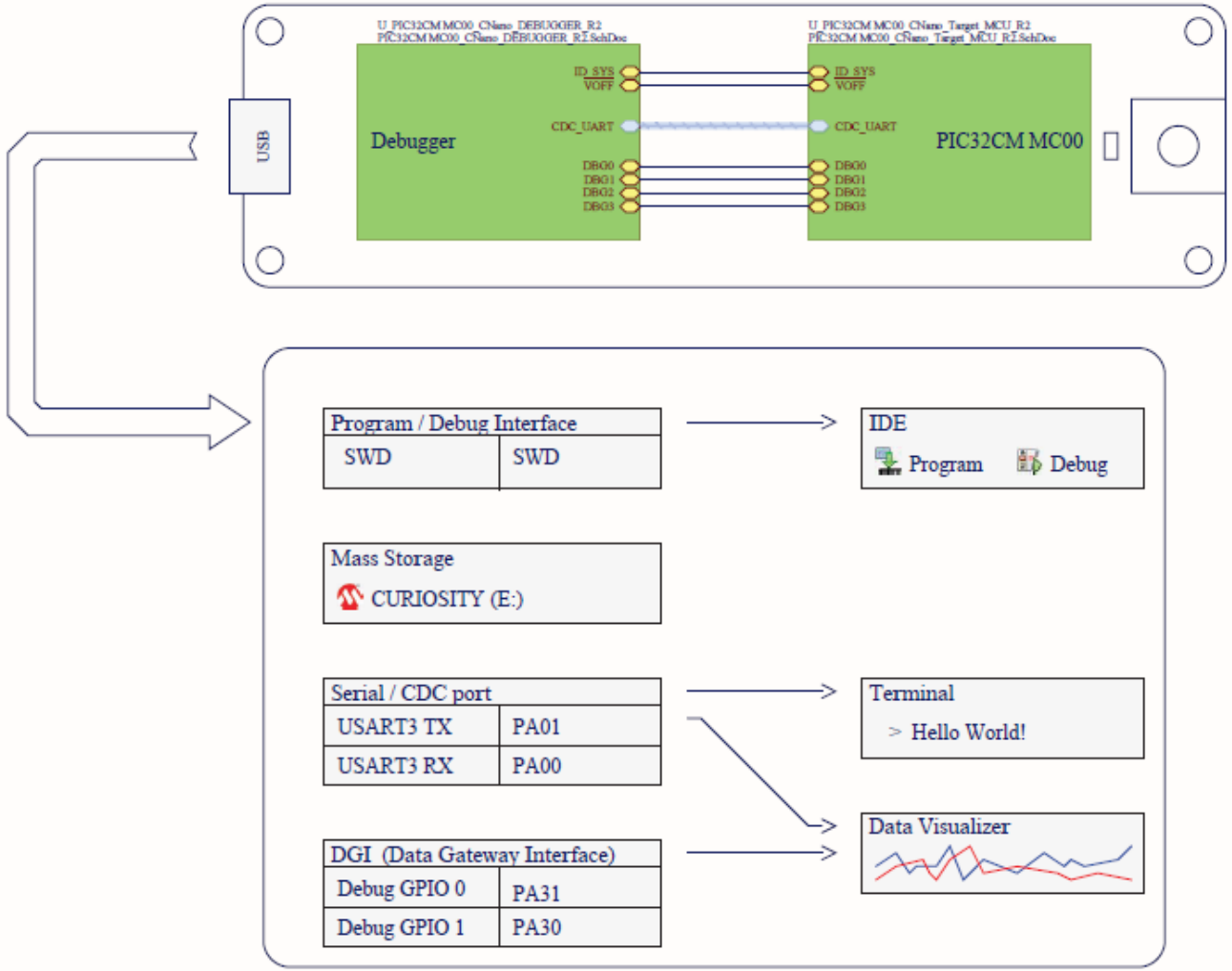
Hook Up and Execute the SPI Peripheral Code



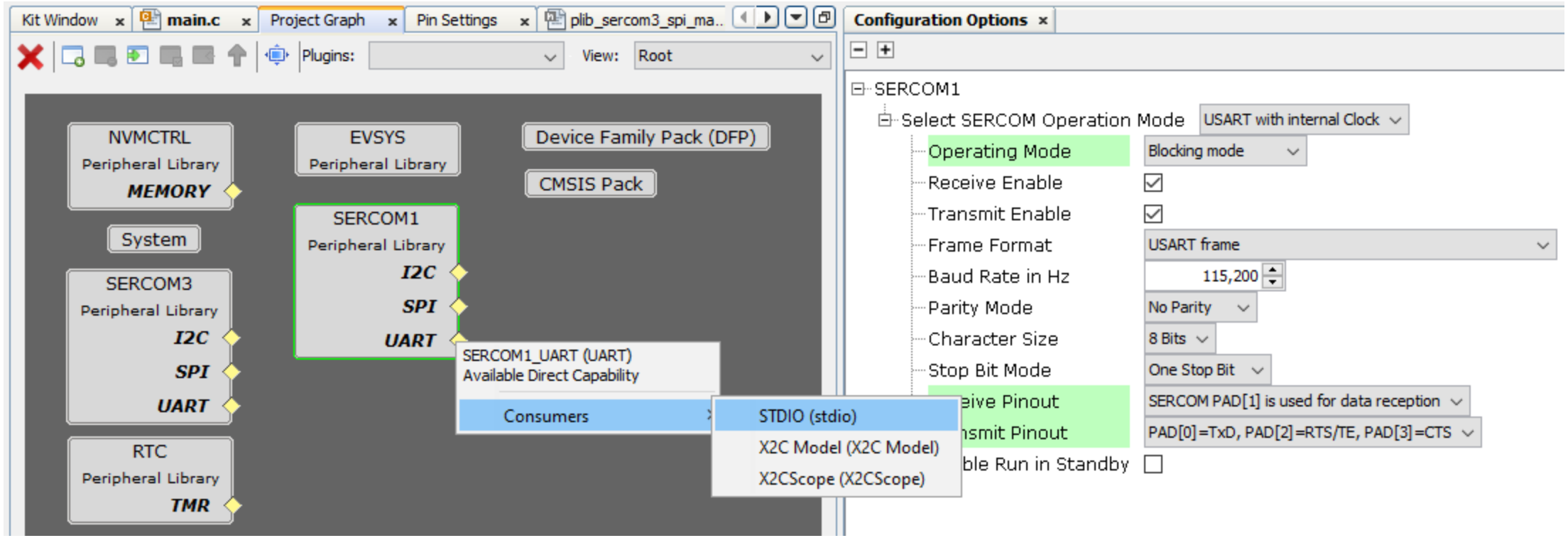


Hook Up and Execute the SPI Peripheral Code

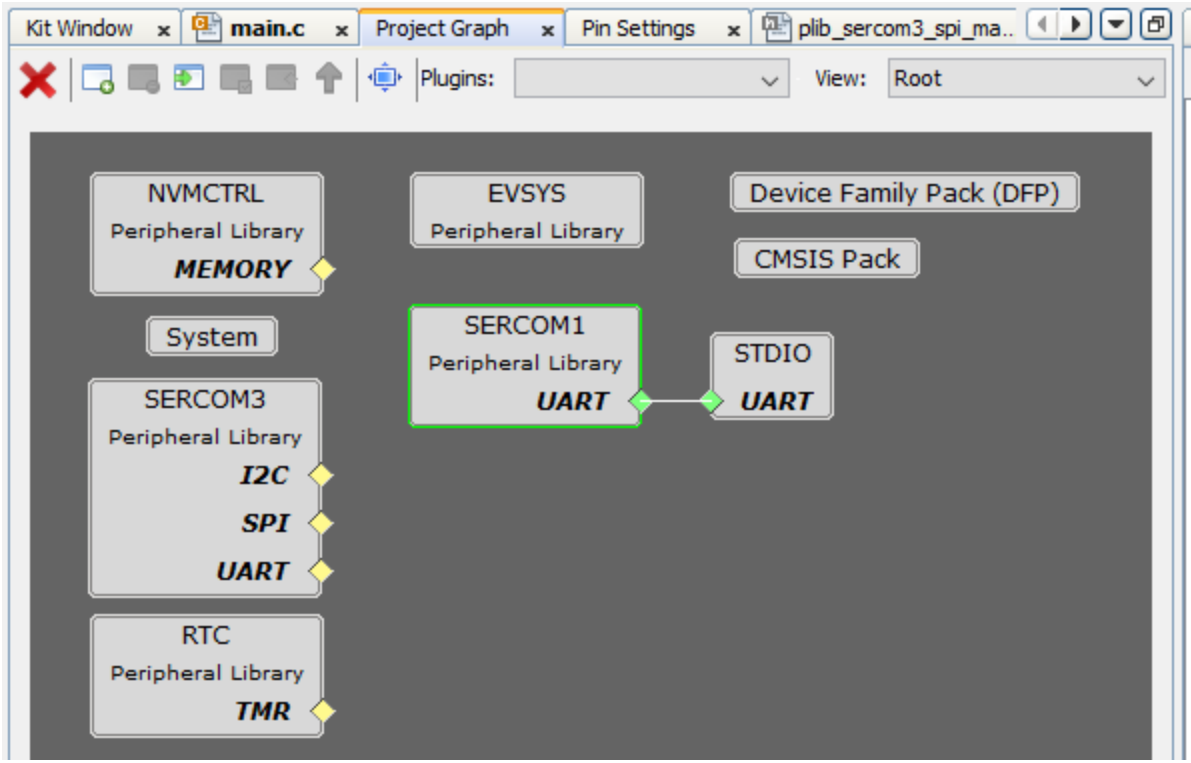
PIC32CM MC00 Curiosity Nano

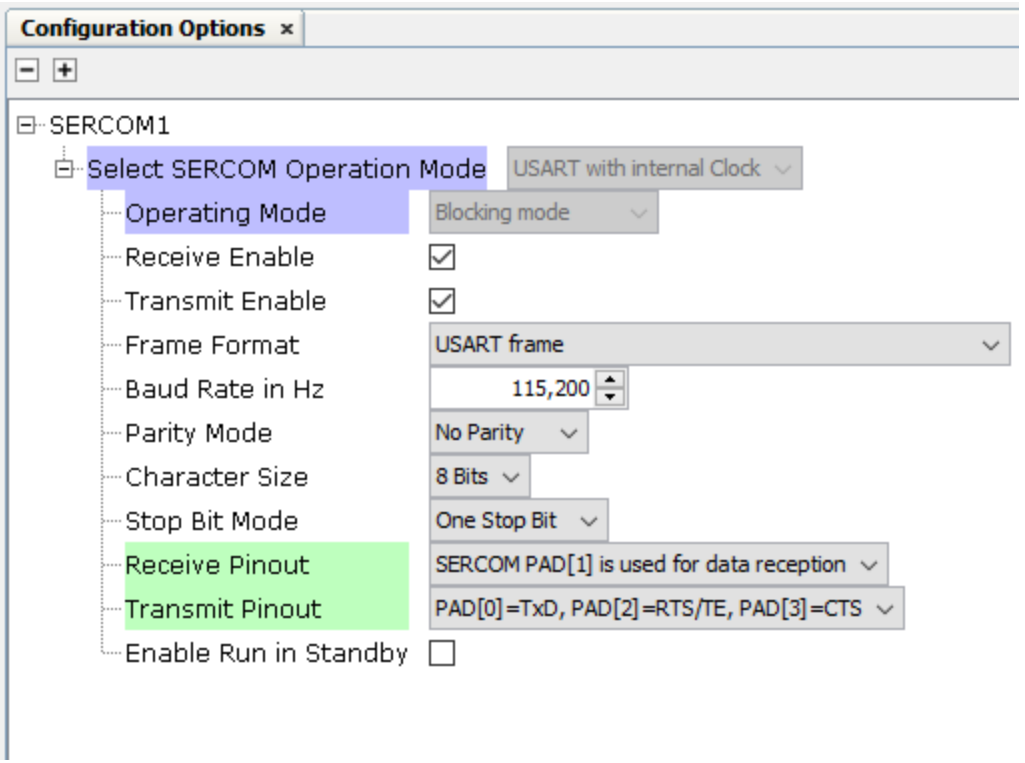


Add UART Peripheral



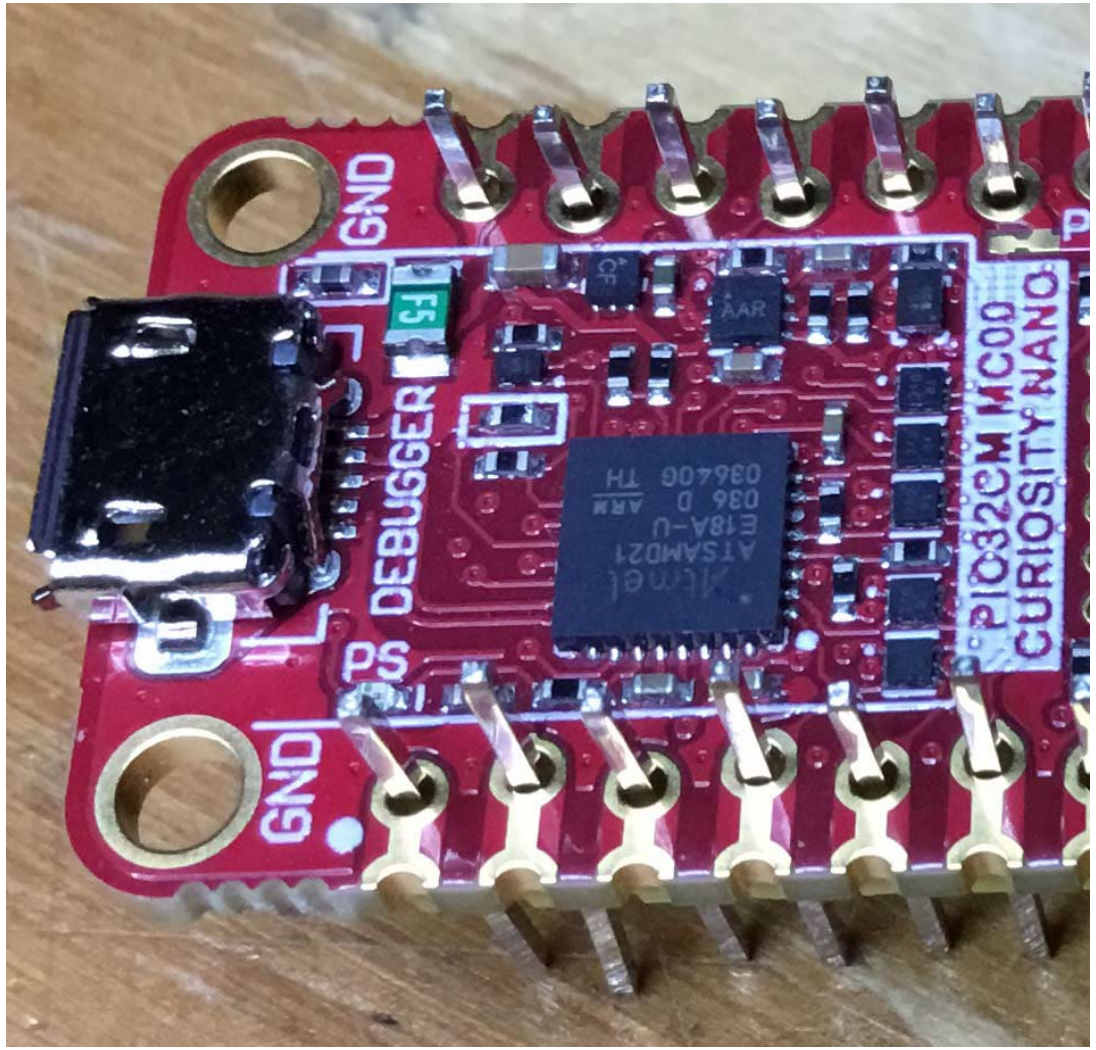
Attach UART Peripheral to STDIO



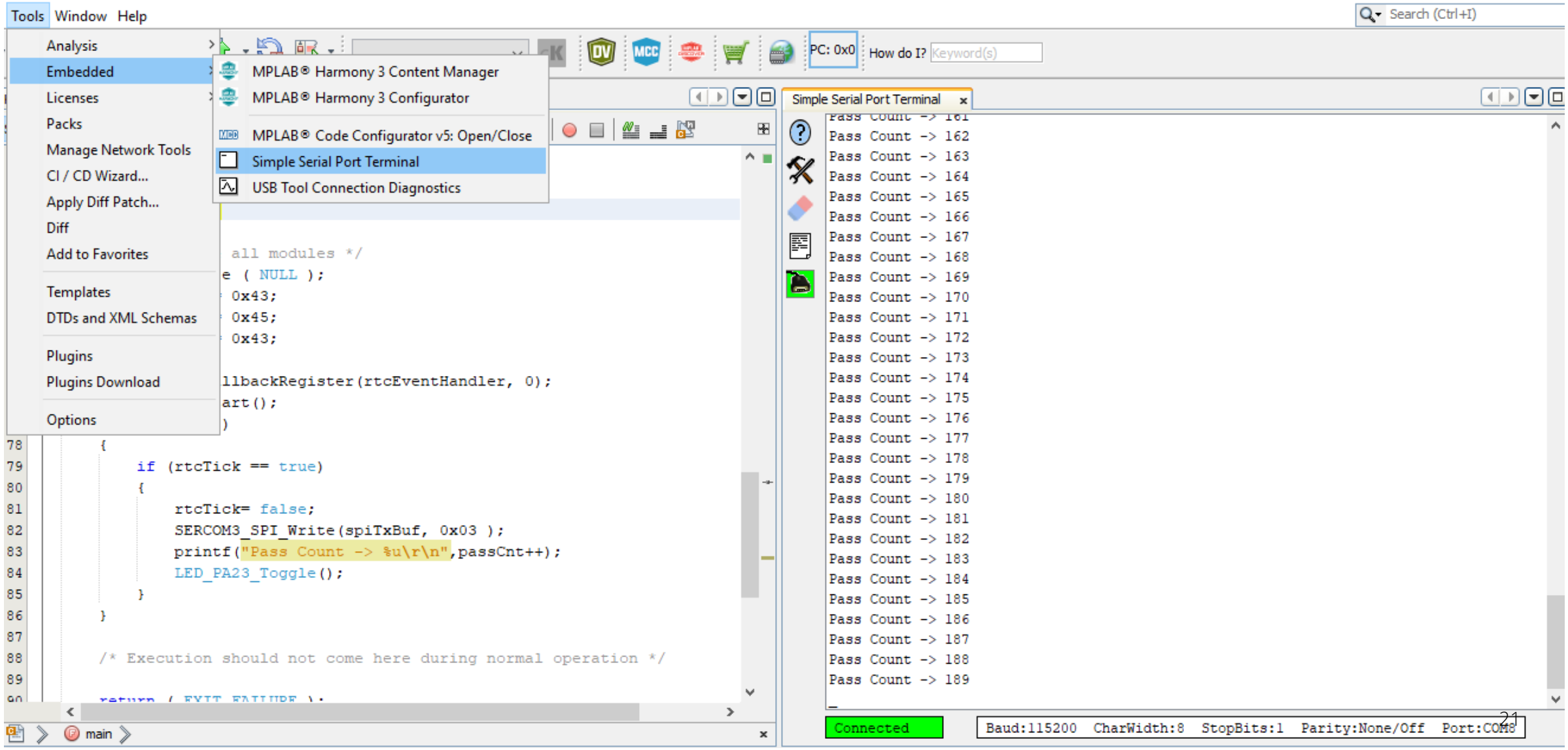


Add Pass Count Code

```
uint8_t spiTxBuf[8];
uint16_t passCnt;
int main ( void )
{
    /* Initialize all modules */
    SYS_Initialize ( NULL );
    spiTxBuf[0] = 0x43;
    spiTxBuf[1] = 0x45;
    spiTxBuf[2] = 0x43;
    passCnt = 0;
    RTC_Timer32CallbackRegister(rtcEventHandler, 0);
    RTC_Timer32Start();
    while ( true )
    {
        if (rtcTick == true)
        {
            rtcTick= false;
            SERCOM3_SPI_Write(spiTxBuf, 0x03 );
            printf("Pass Count -> %u\r\n",passCnt++);
            LED_PA23_Toggle();
        }
    }
```



Activate Serial Port Terminal and Execute the App

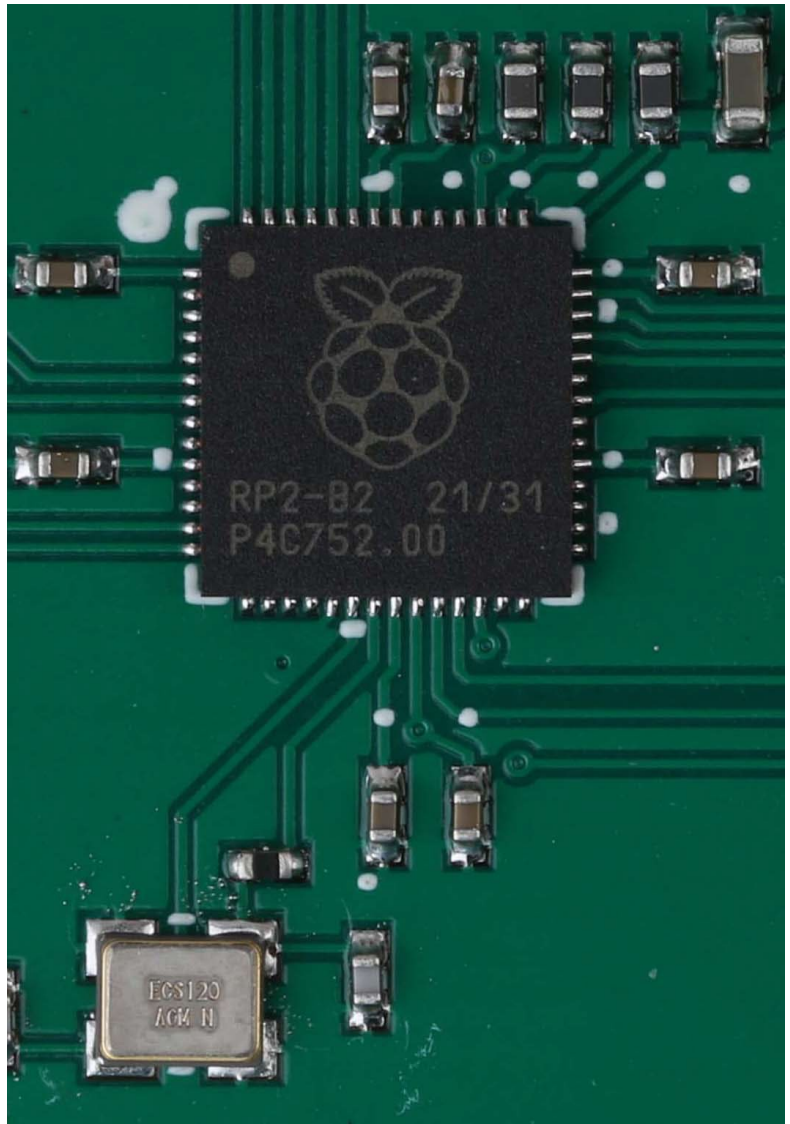


Thank you for attending!!!

Please consider the resources below:

- **microchip.com**

MORE TO COME..



CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

Digi-Key ELECTRONICS

informa markets