CEC Continuing Education Center

DesignNews

IoT Device Prototyping with Microchip Curiosity Development Boards

Day 3:

Constructing a 32-bit Wi-Fi Node Using the PIC32MM USB Curiosity Development Board

Sponsored by

Digi-Key ELECTRONICS

informa markets

Webinar Logistics

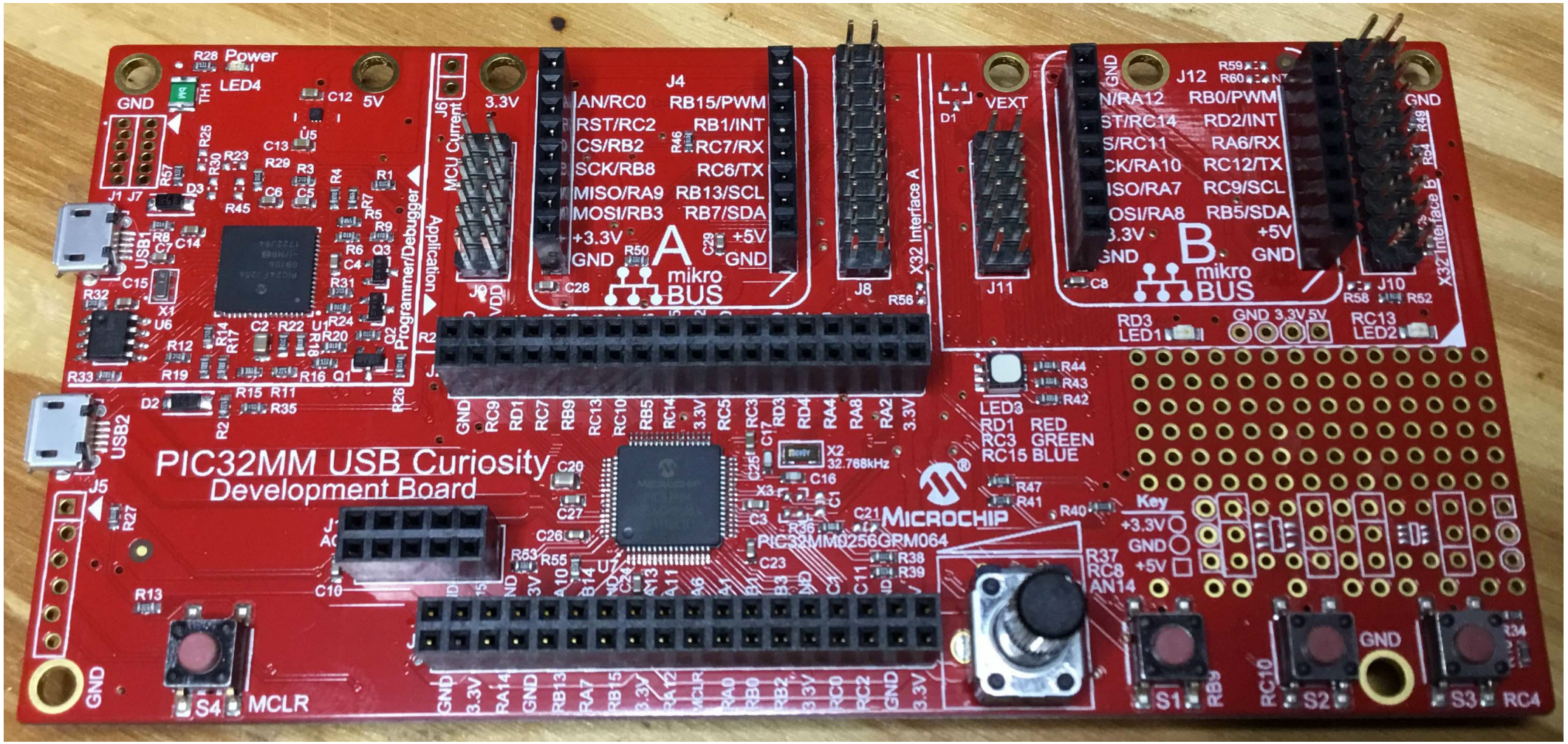
- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Fred Eady

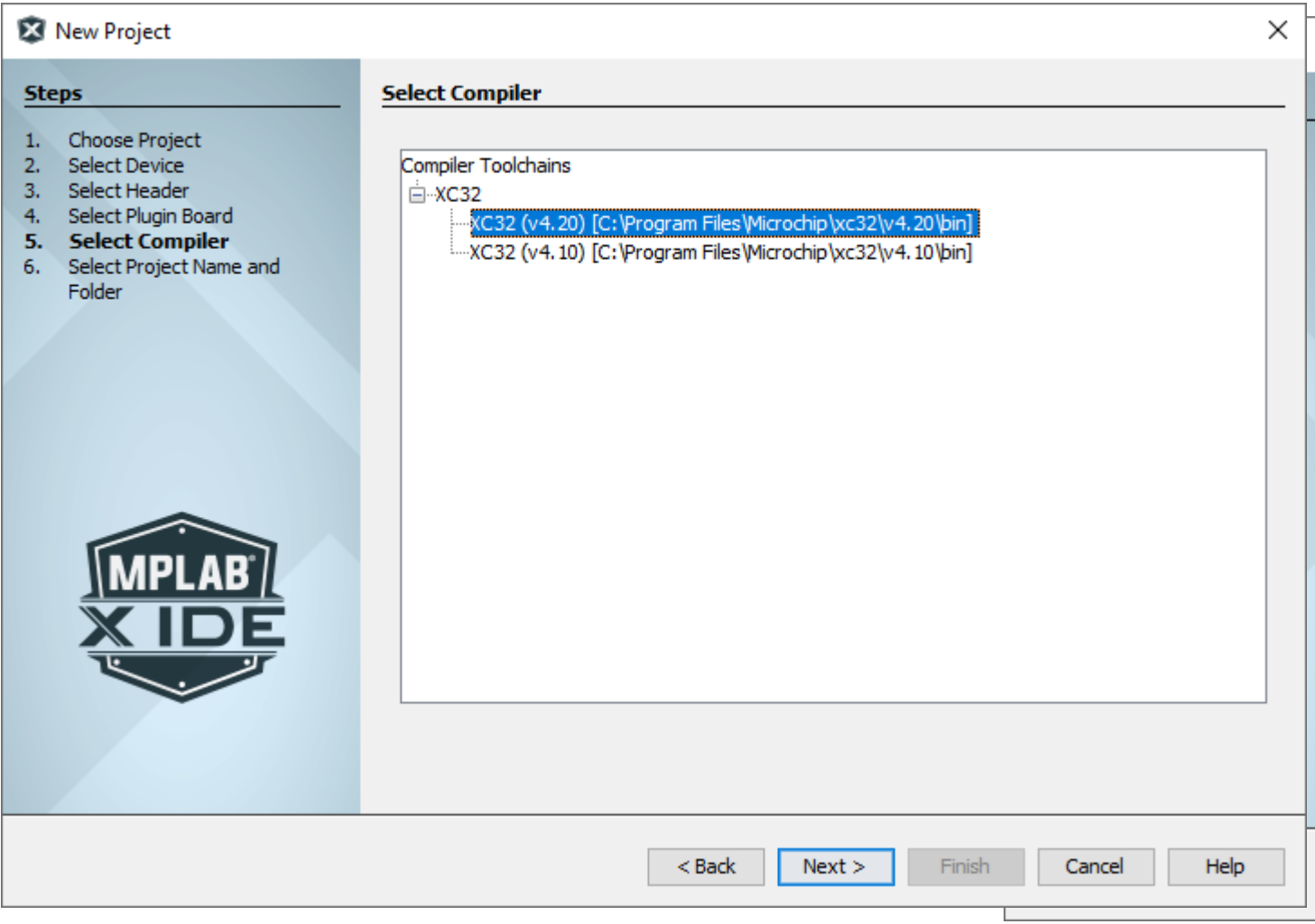
Visit ‘Lecturer Profile’ in your console for more details.

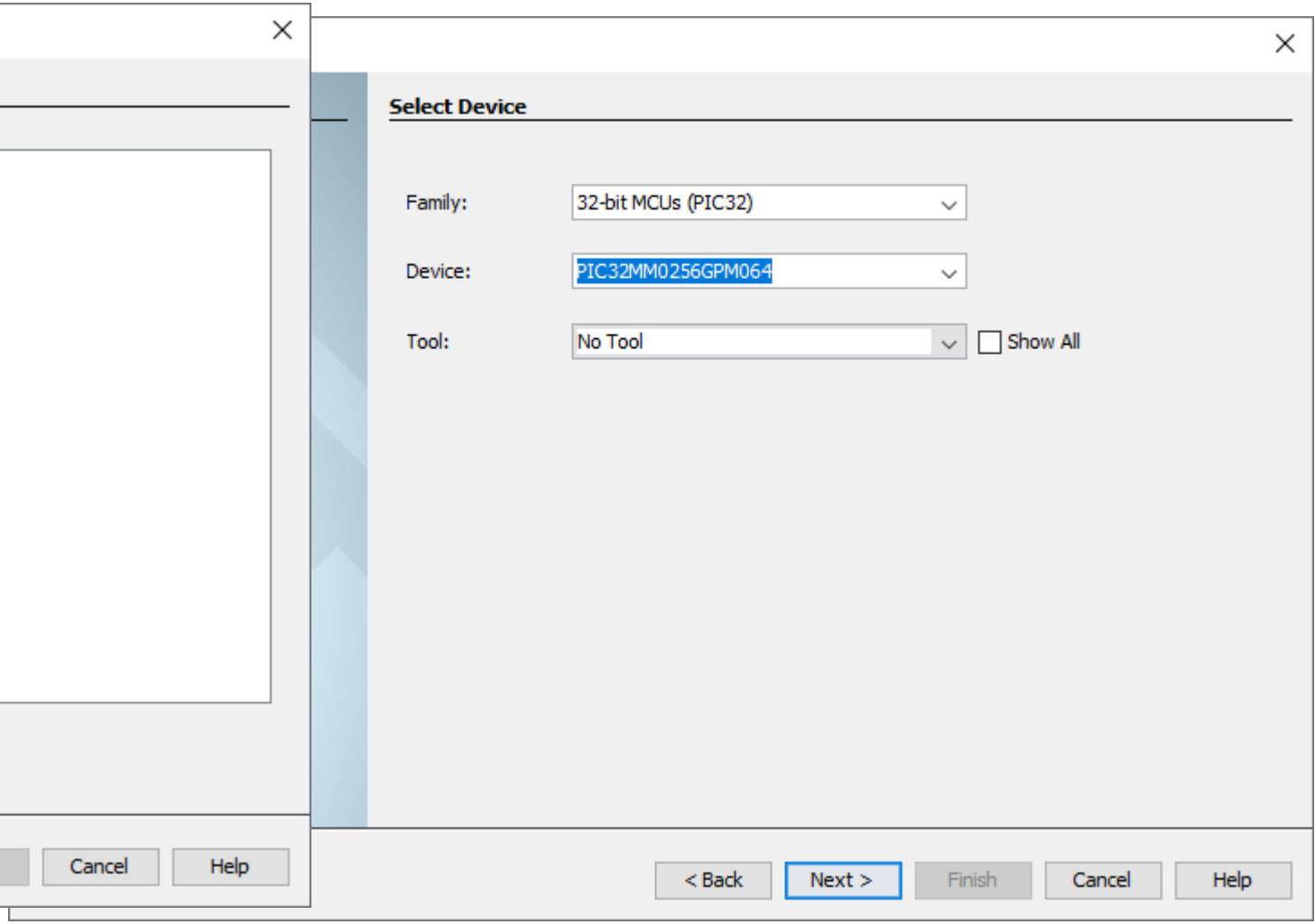
AGENDA

- **Coding a PIC32MM0256GPM064 WizFi360 Driver**

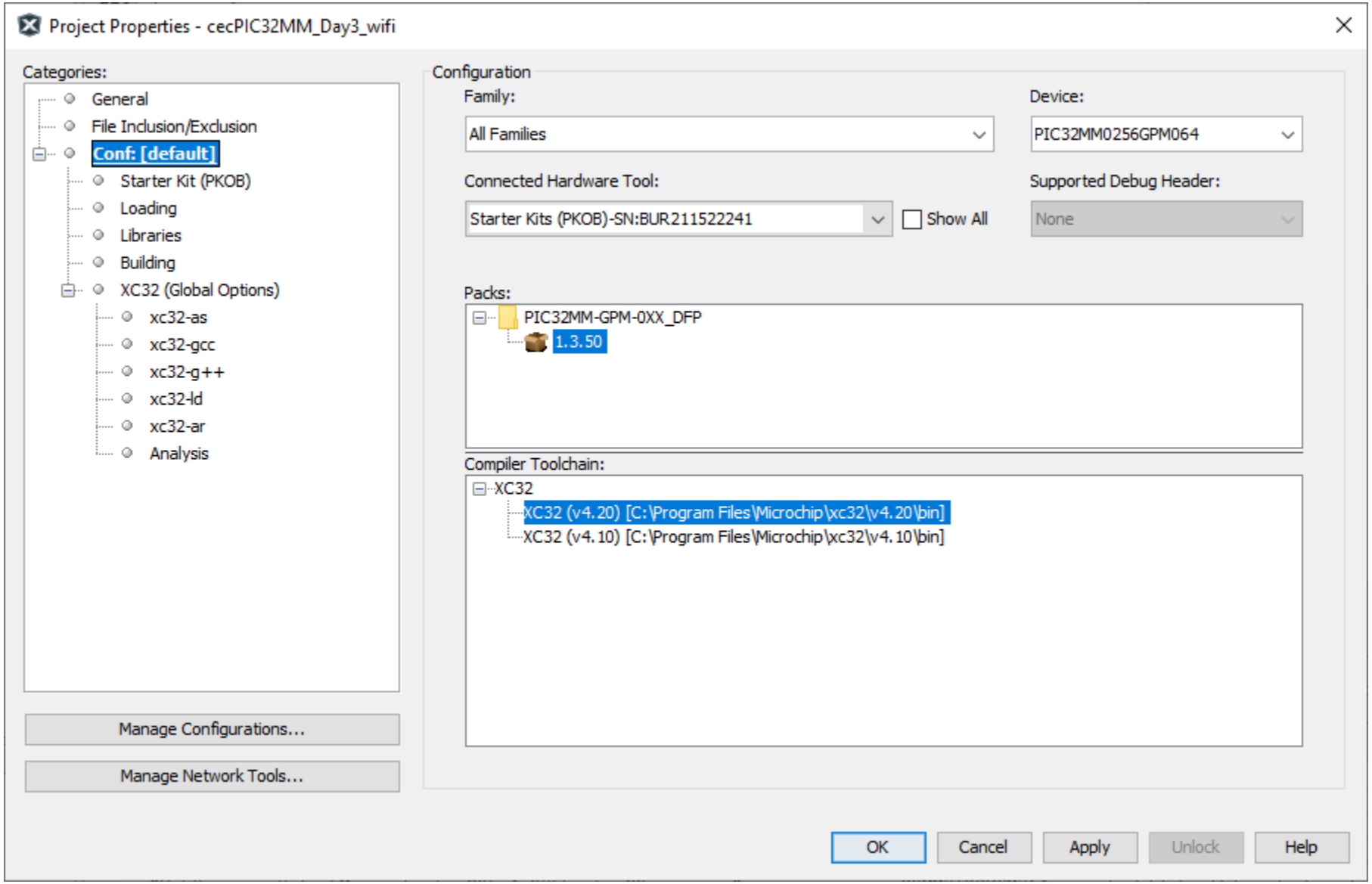


Create the PIC32MM0256GPM064 MPLAB X Project

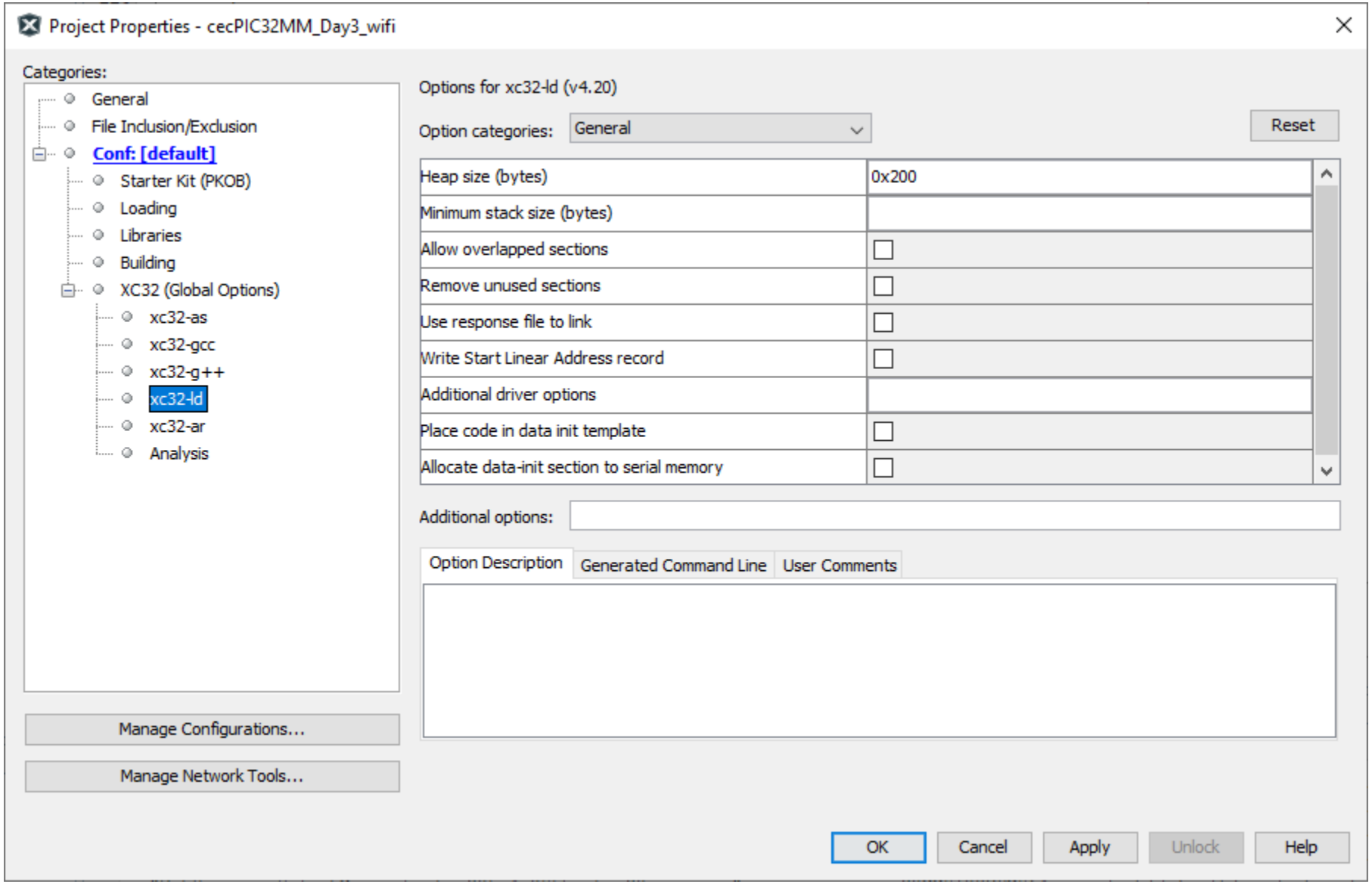




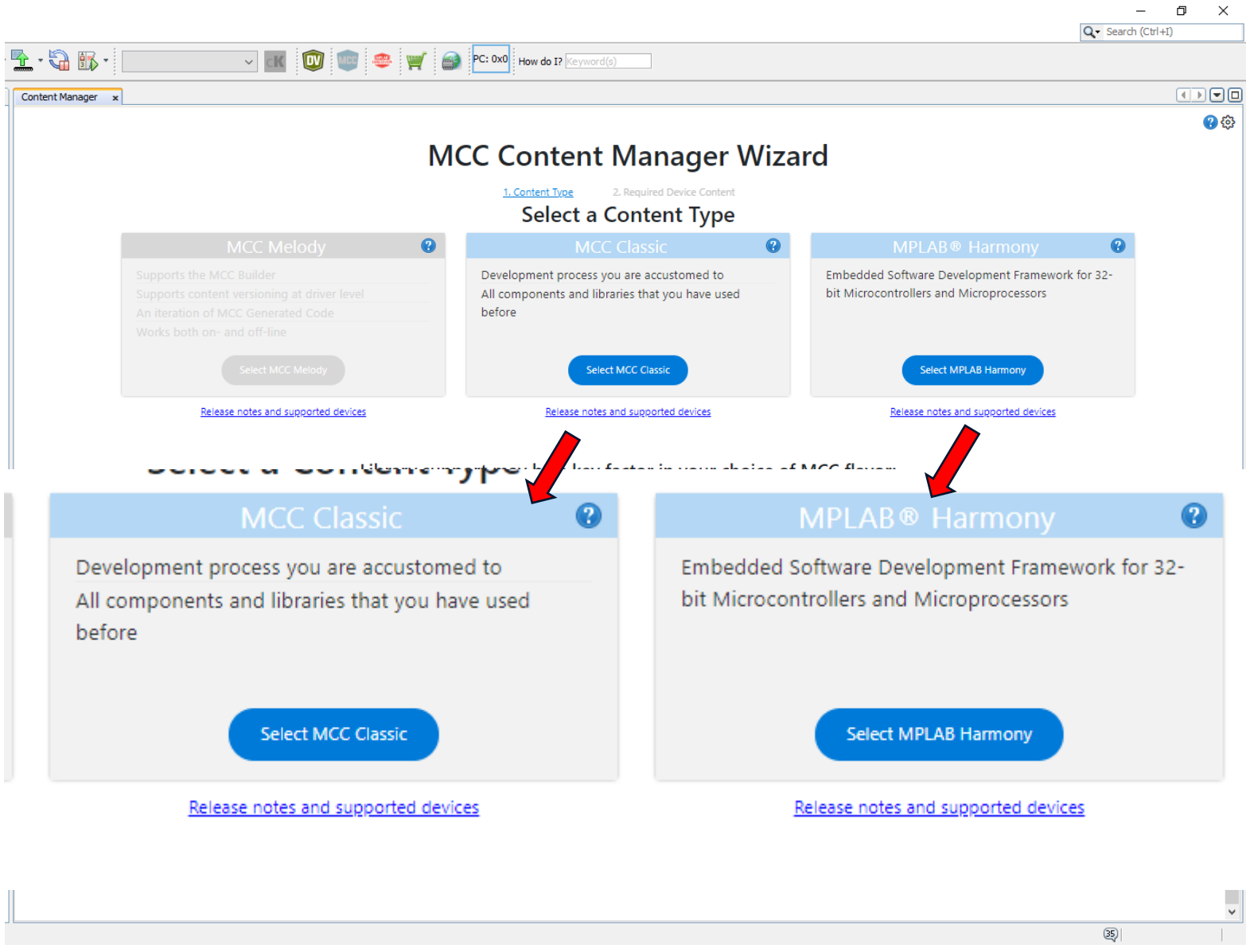
Create the PIC32MM0256GPM064 MPLAB X Project



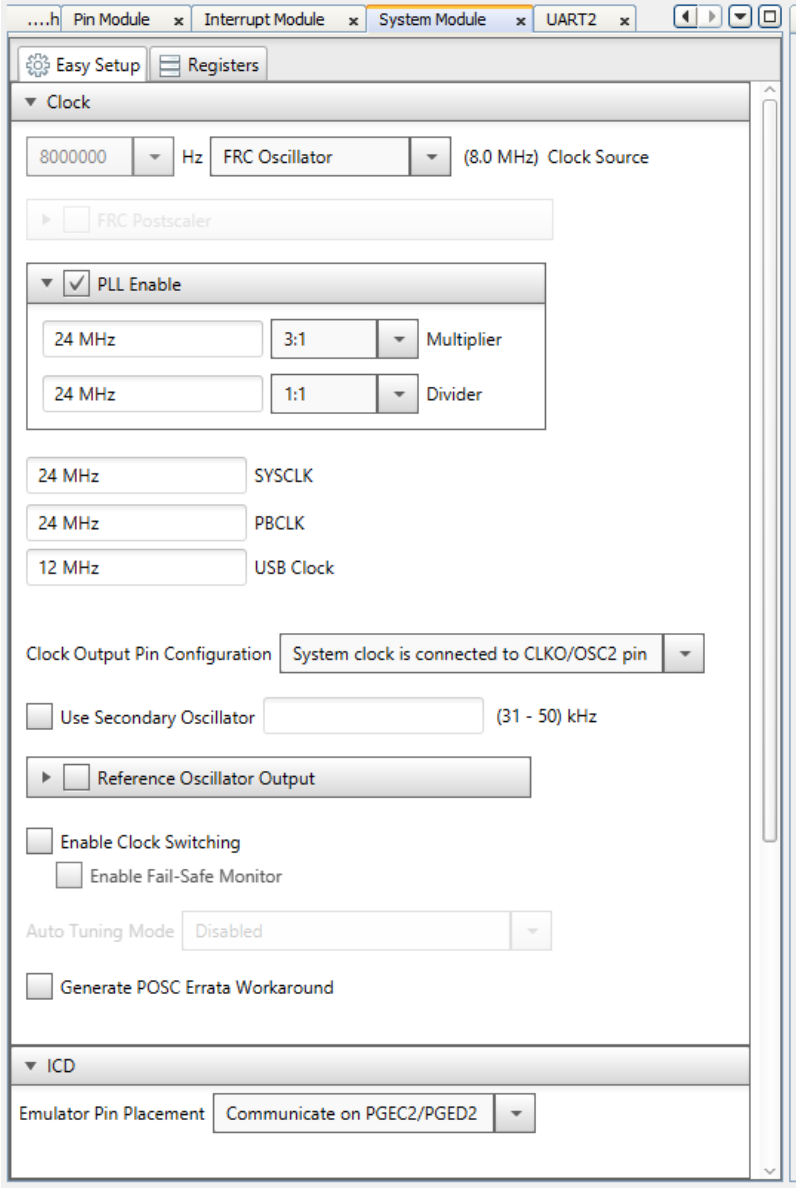
Define the Heap Size

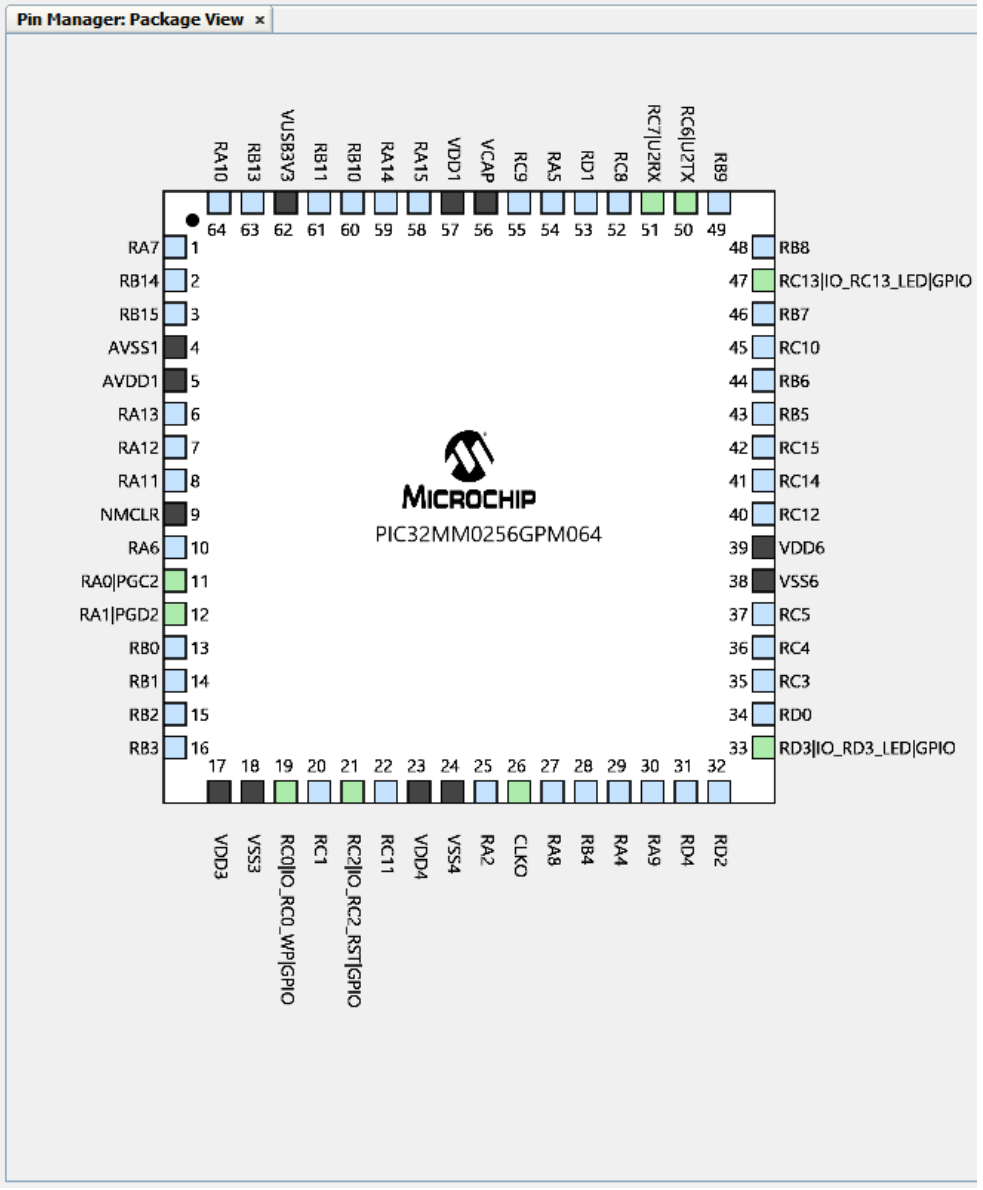


MCC Content Manager Selections



MCC Configuration – System





MCC Configuration – EUSART

main.c | uart2.c | uart2.h | Pin Module | Interrupt Module | System Module | UART2

Easy Setup | Registers

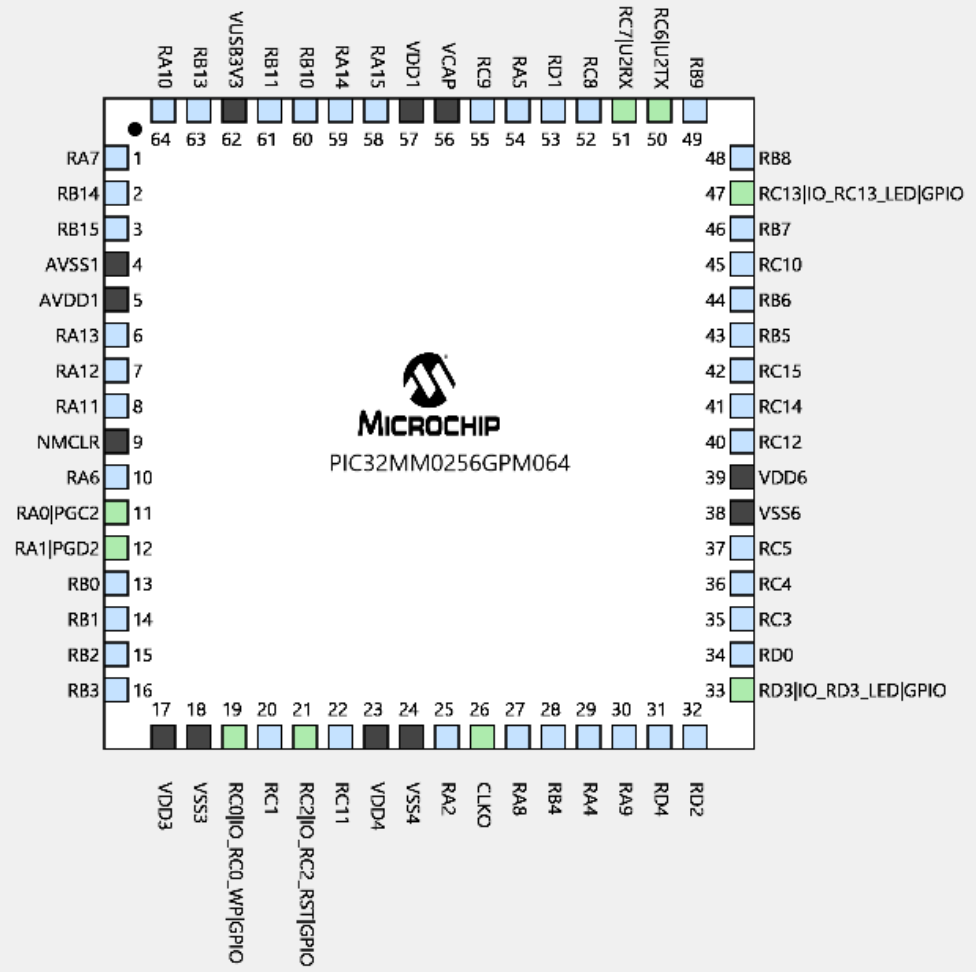
Hardware Settings

- [x] Enable UART
- Clock Source: PBCLK
- Baud Rate: 115200 — Error Rate = 0.160
- Parity: None
- Data Bits: 8
- Stop Bits: 1
- Flow Control: None
- [x] Enable UART Interrupts

Software Settings

- [x] Redirect Printf to UART
- Software Transmit Buffer Size: 0x8 Bytes
- Software Receive Buffer Size: 0x8 Bytes

Pin Manager: Package View



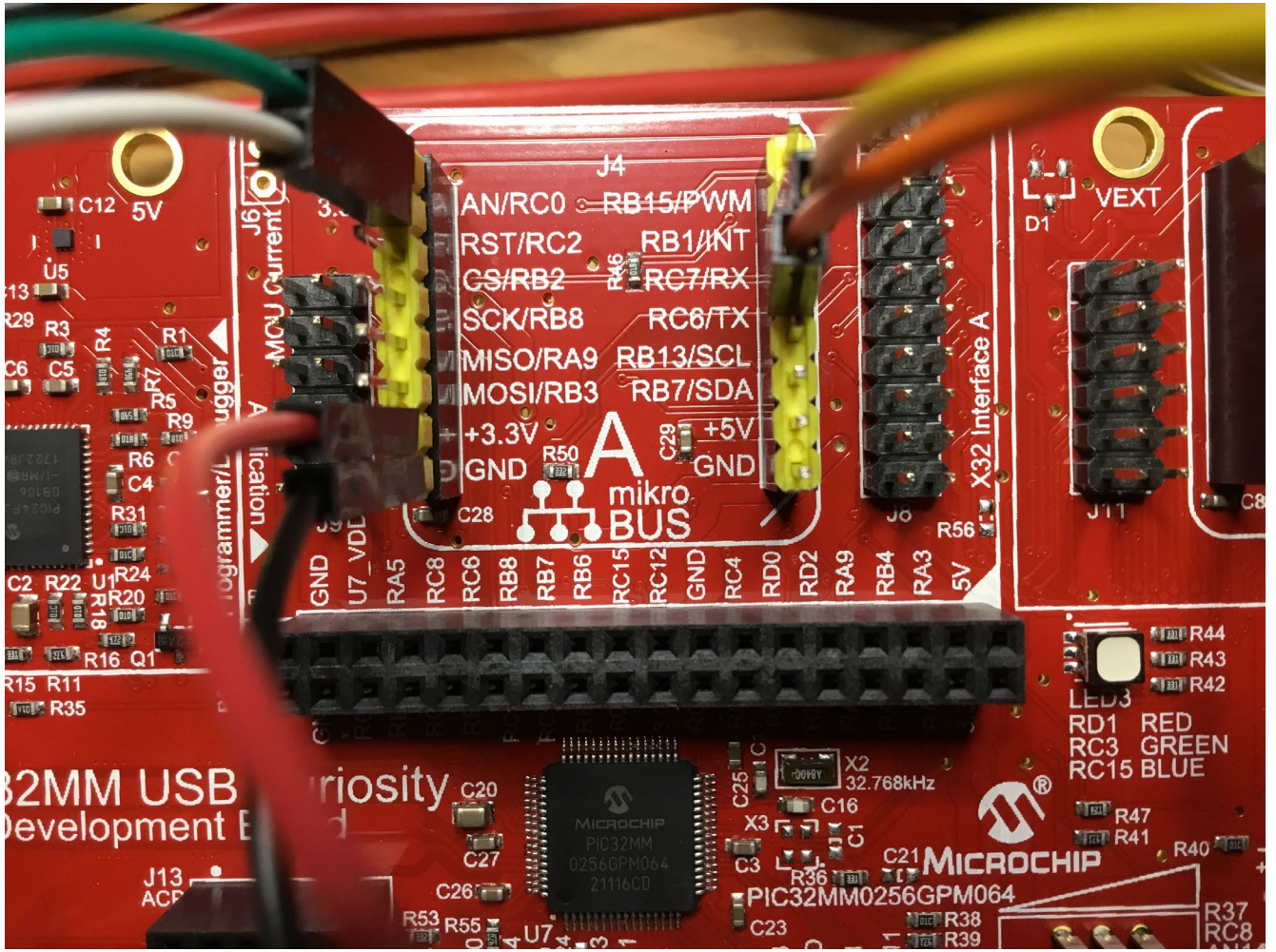
MCC Configuration – I/O Pins

Notifications | Output | Pin Manager: Grid View × | Notifications [MCC]

Package: TQFP64	Pin No:	11	12	25	26	29	54	10	1	27	30	64	8	7	6	59	58	13	14	15	16	28	43	44	46	48	49	60	61	63	2	3	19	20	21	35	36	37	50	51	52	55	45	22	40	47	41	42	34	53	32	33	31
		Port A ▼																Port B ▼															Port C ▼																			Port D ▼				
Module	**Function**	**Direction**	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7	8	9	10	11	13	14	15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1	2	3	4
Clock ▼	CLKI	input																																																					
	CLKO	output																																																					
	OSCI	input																																																				
	OSCO	output																																																				
	REFCLKI	input																																																			
	REFCLKO	output																																																			
	SOSCI	input																																																			
	SOSCO	output																																																			
ICD ▼	PGCx	input																																																			
	PGDx	input																																																				
Pin Module ▼	GPIO	input																																																			
	GPIO	output																																																			
UART2 ▼	U2CTS	input																																																			
	U2RTS	output																																																			
	U2RX	input																																																			
	U2TX	output																																																			

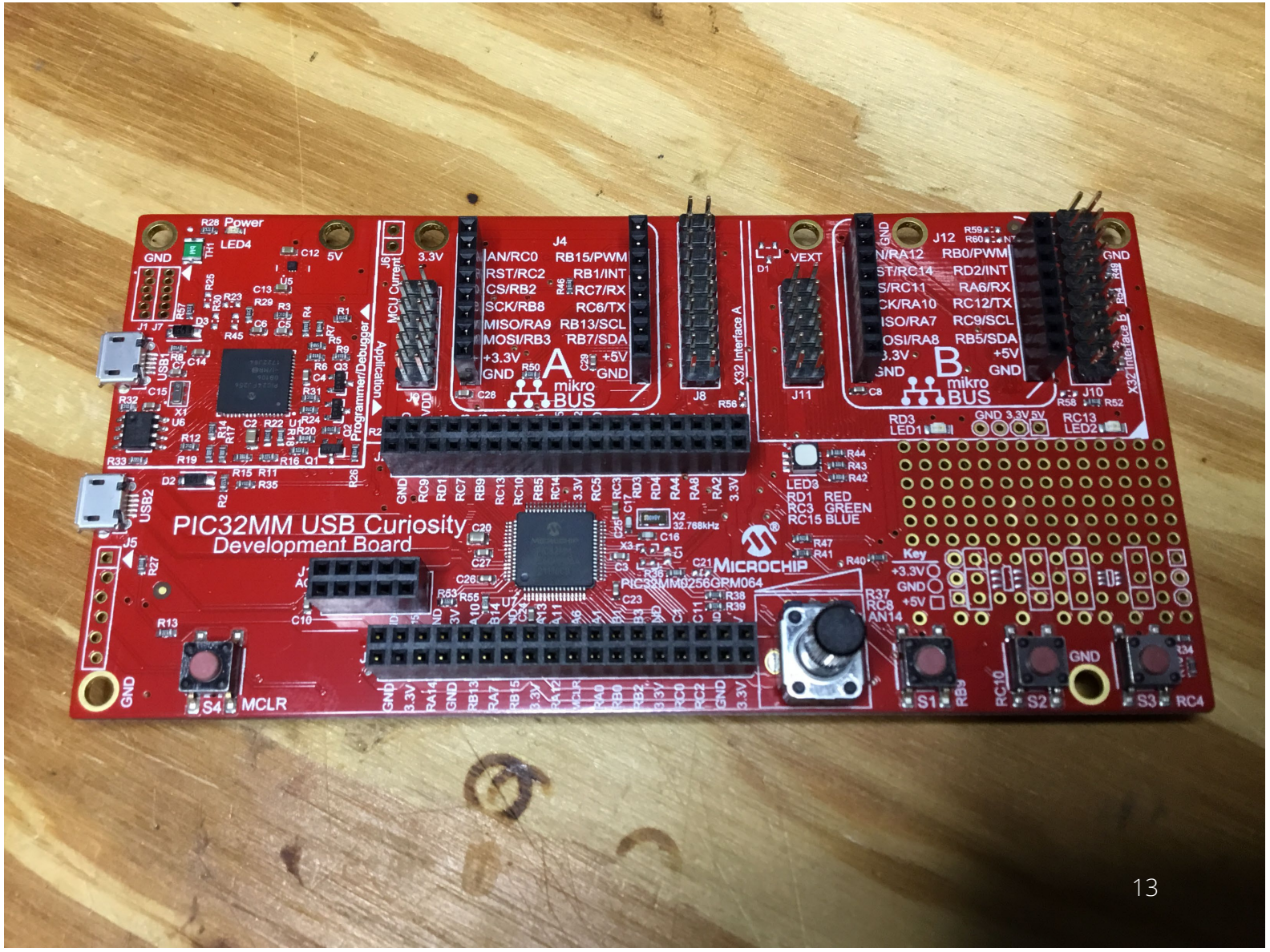
EUSART User Function

```
//* From usart2.h
void UART2_ResetRxQueue(void);

//* From usart2.c
void UART2_ResetRxQueue(void)
{
    IEC1bits.U2RXIE = 0;    //disable Rx interrupt
    rxHead = rxQueue;       //rxHead to top of rxQueue
    rxTail = rxQueue;       //rxTail to top of rxQueue
    rxOverflowed = false;   //reset overflow flag
    IEC1bits.U2RXIE = 1;    //enable Rx interrupt
}
```



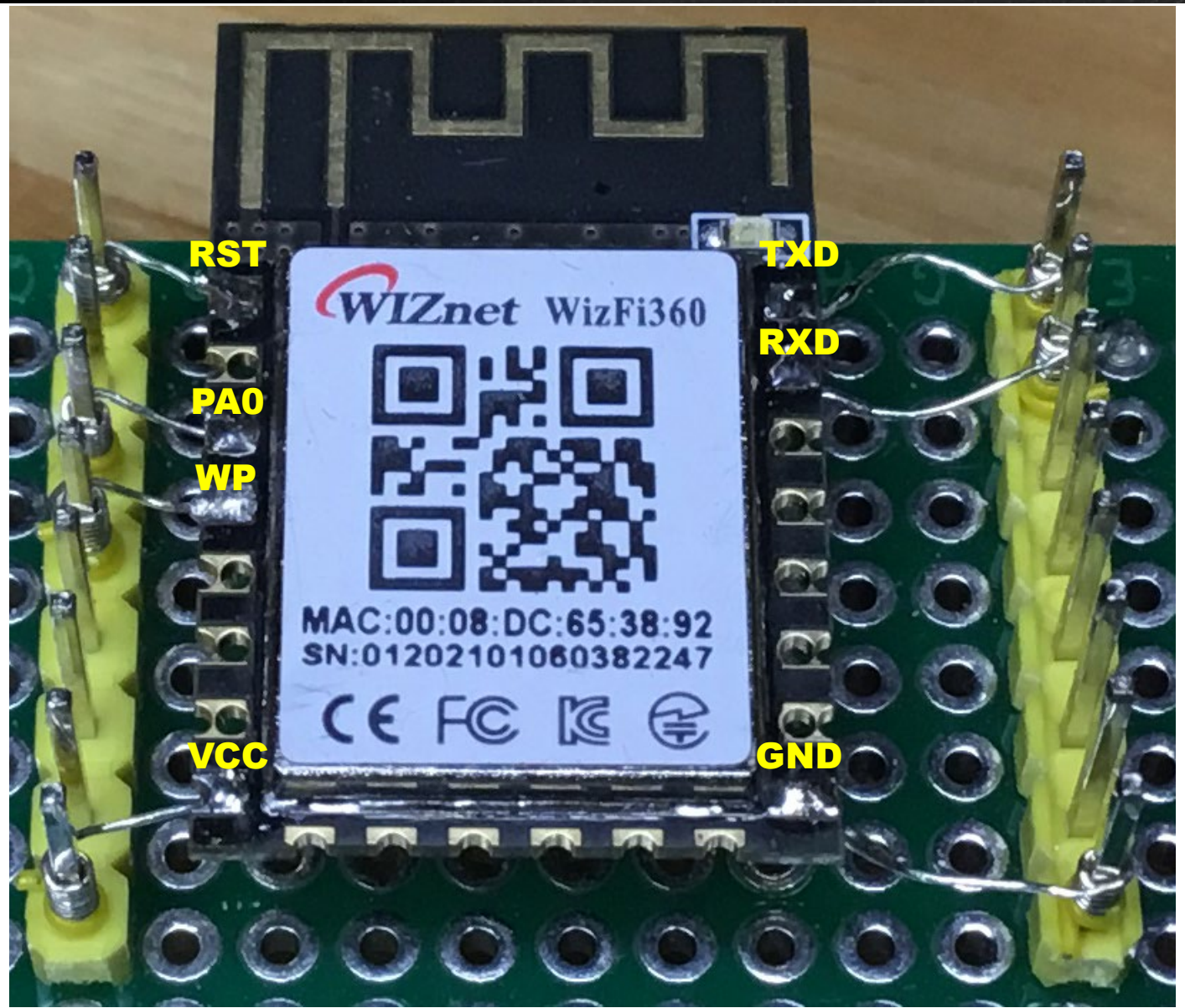
PIC32MM Delay Functions

```
#define SYS_FREQ 24000000
//************************************************
//* DELAY FUNCTIONS
//************************************************
void delay_us(unsigned int us)
{
    // Convert microseconds into clock ticks
    // Core Timer updates every 2 ticks
    us *= SYS_FREQ / 1000000 / 2;
    // Set Core Timer count to 0
    _CP0_SET_COUNT(0);
    // Wait until Core Timer count reaches the
    //us number we calculated earlier
    while (us > _CP0_GET_COUNT());
}
void delay_ms(int ms)
{
    delay_us(ms * 1000);
}
```



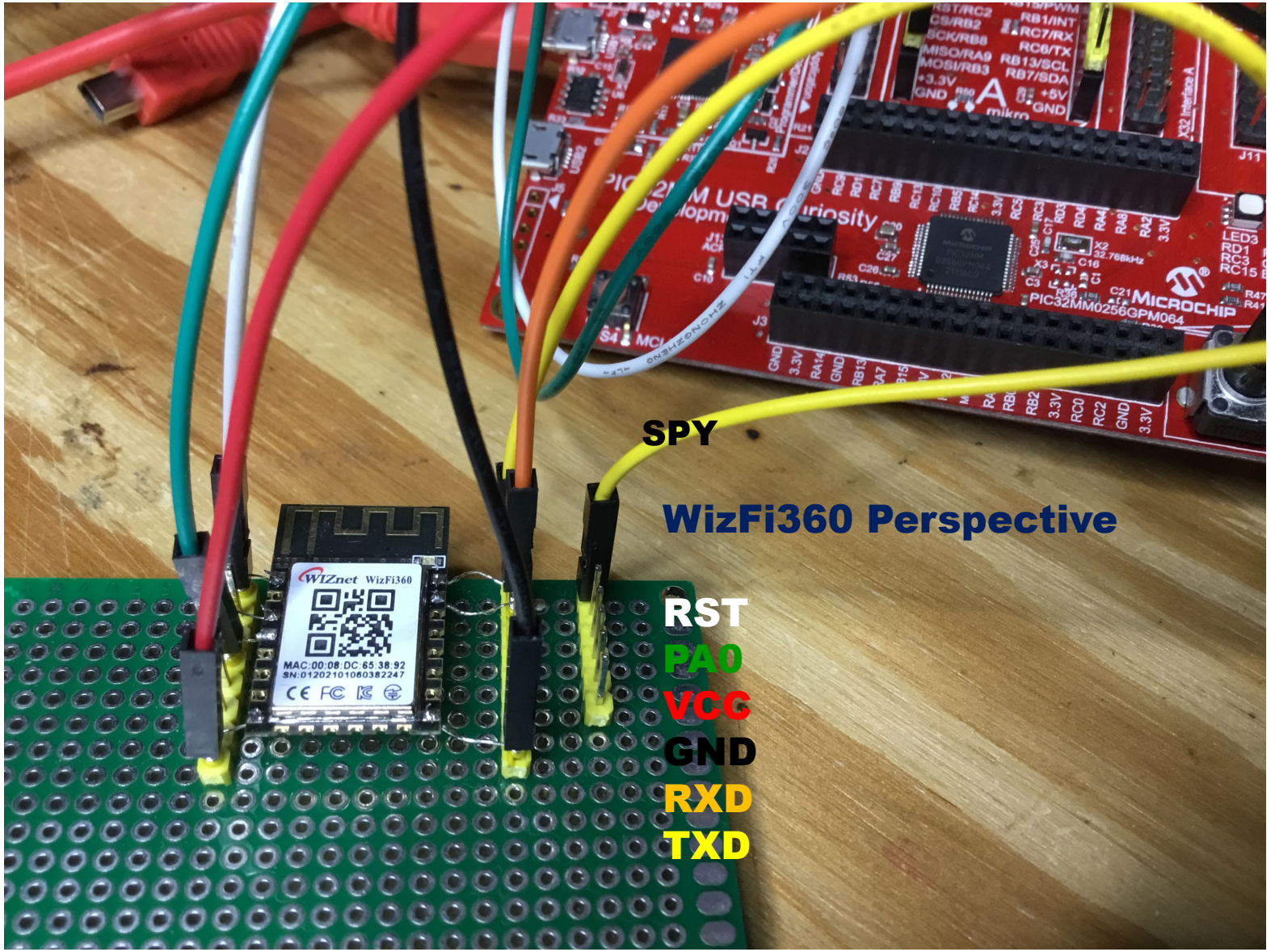
End of Command Function

```
//*****************************************
//* CHECK FOR OK
//*****************************************
uint8_t chk_atok(void)
{
    rc = 0;
    IO_RD3_LED_SetLow();
    while(!UART2_IsRxReady());
    delay_ms(100);

    do{
        rxBuf[indx++] = UART2_Read();
    }while(UART2_IsRxReady());

    indx -= 3;
    if(rxBuf[indx] == 'K')
    {
        rc = 1;
    }
    return rc;
}
```



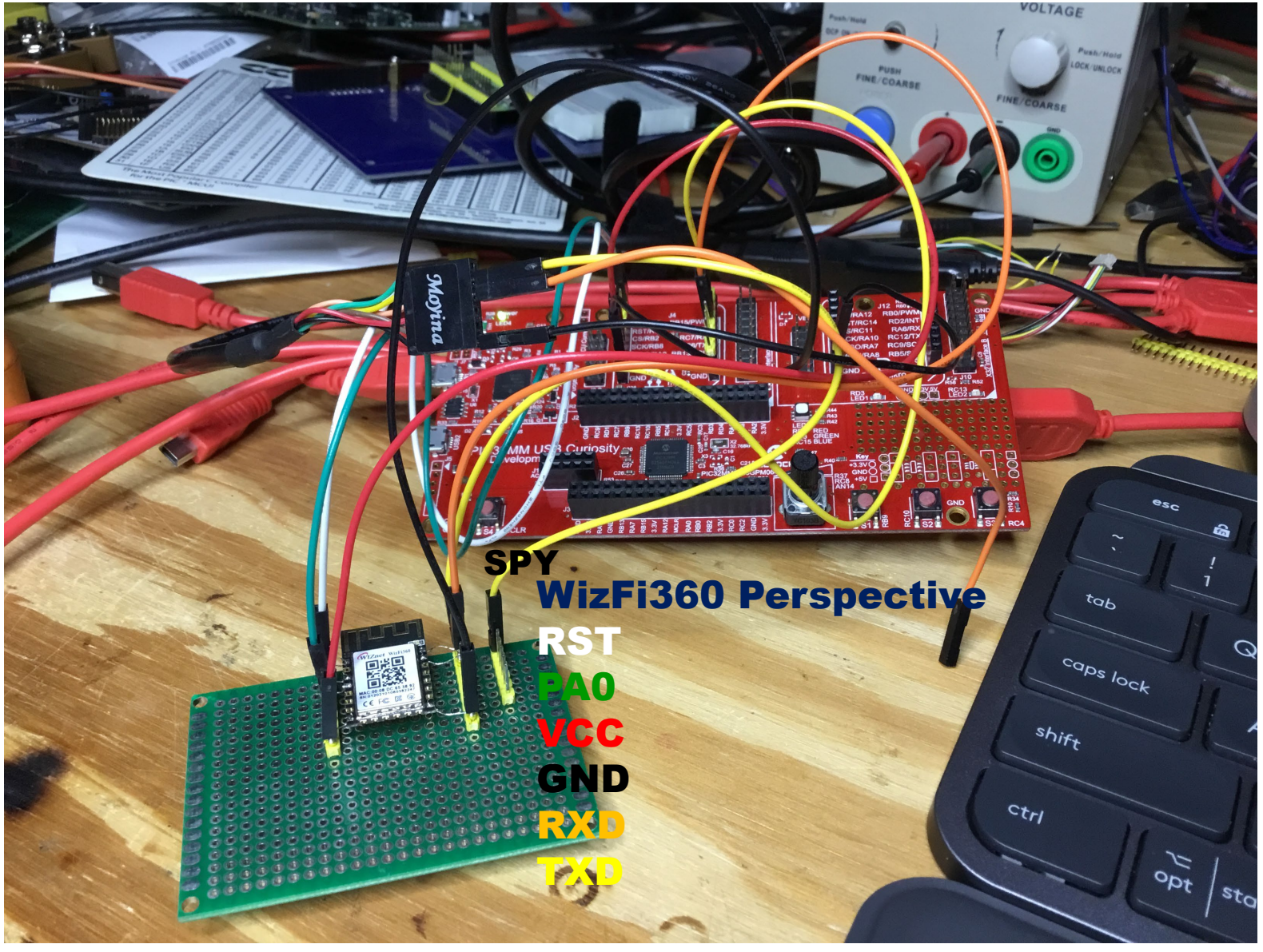
Station Mode Function

```
//*****************************************
//* SET STATION MODE
//*****************************************
uint8_t setStationMode(void)
{
    rc = 0;
    UART2_ResetRxQueue();
    indx = 0;

    printf("AT+CWMODE_CUR=1\r\n");

    if(chk_atok())
    {
        rc = 1;
    }
    return rc;
}
```



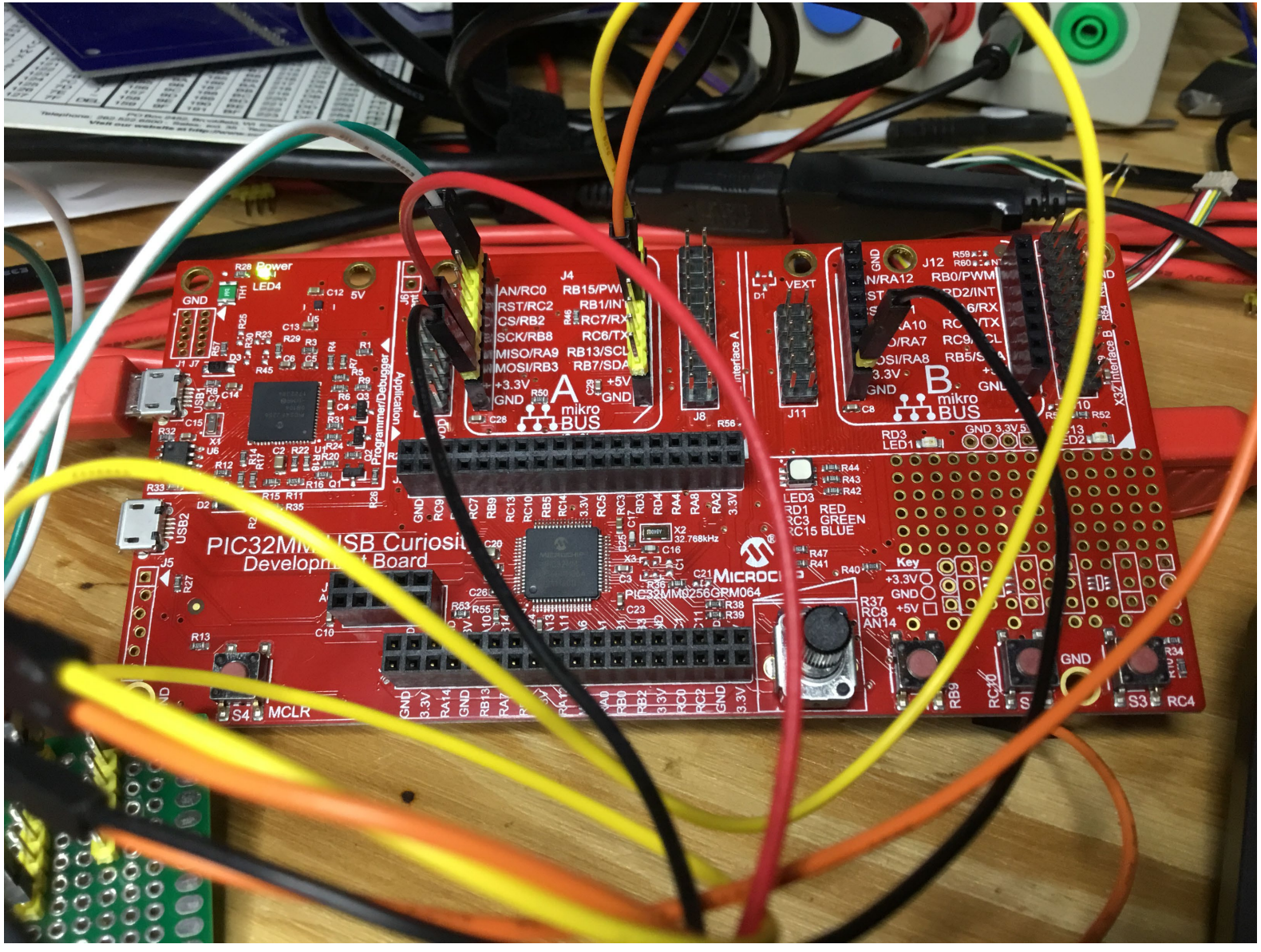
Single Connection Mode Function

```
//*********************************************
//* SET SINGLE CONNECTION MODE
//*********************************************
uint8_t setSingleConnectionMode(void)
{
    rc = 0;
    UART2_ResetRxQueue();
    indx = 0;

    printf("AT+CIPMUX=0\r\n");

    if(chk_atok())
    {
      rc = 1;
    }
    return rc;
}
```



DHCP Enable Function

```
//*****************************************
//* SET DHCP ENABLE
//*****************************************
uint8_t setDhcpEnable(void)
{
    rc = 0;
    UART2_ResetRxQueue();
    indx = 0;

    printf("AT+CWDHCP_CUR=1,1\r\n");
    if(chk_atok())

    {
      rc = 1;
    }
    return rc;
}
```

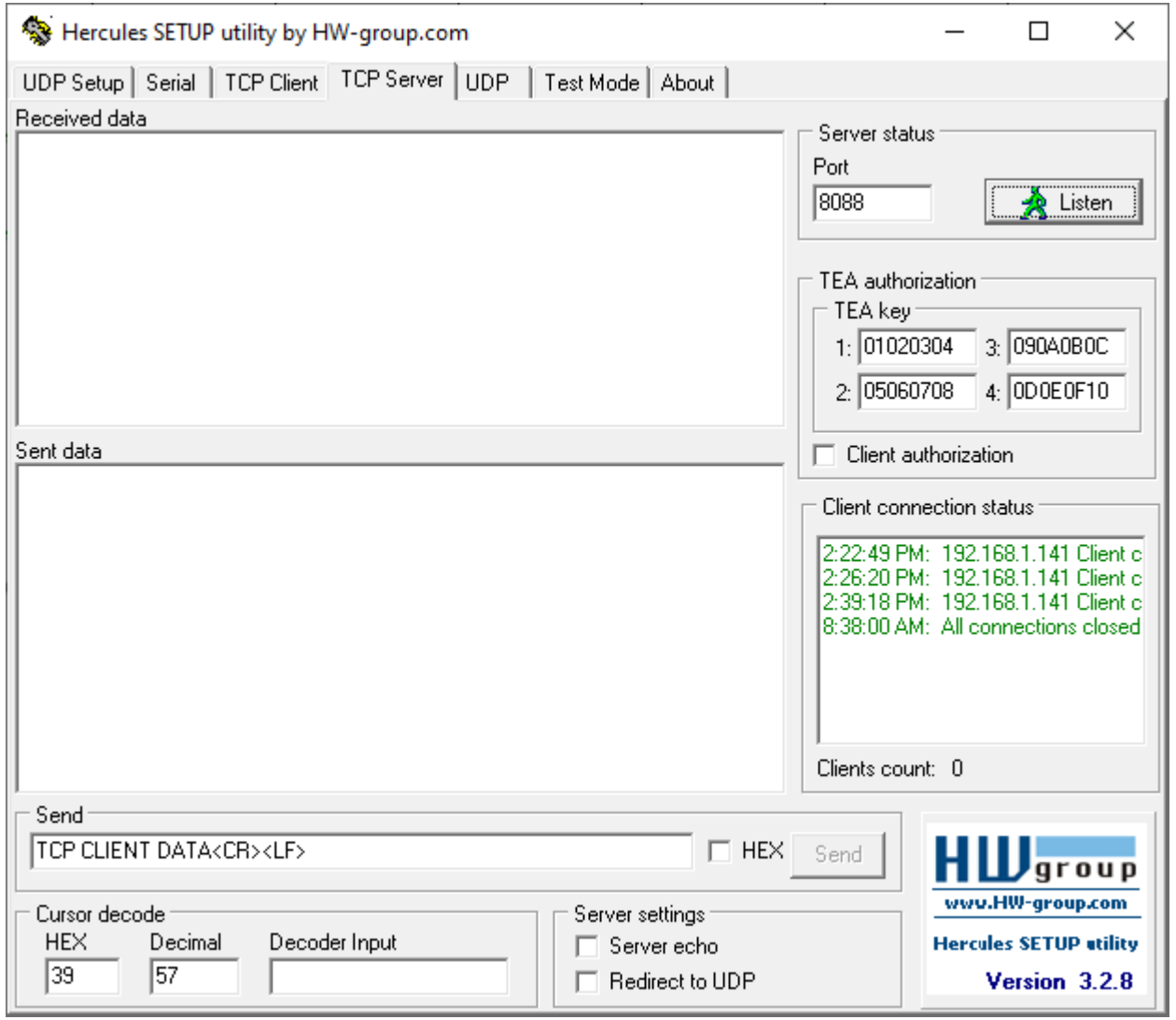


Connect to the Access Point Function

```
//*****************************************
//* CONNECT TO THE AP
//*****************************************
uint8_t connect2AP(void)
{
    uint8_t tmr;
    rc = 0;
    UART2_ResetRxQueue();
    indx = 0;

    printf("AT+CWJAP_CUR=\"edtpnet2\",\“password\"\r\n");
    for(tmr=0;tmr<11;tmr++)
    {
        delay_ms(1000);
    }

    if(chk_atok())
    {
      rc = 1;
    }
    return rc;
}
```

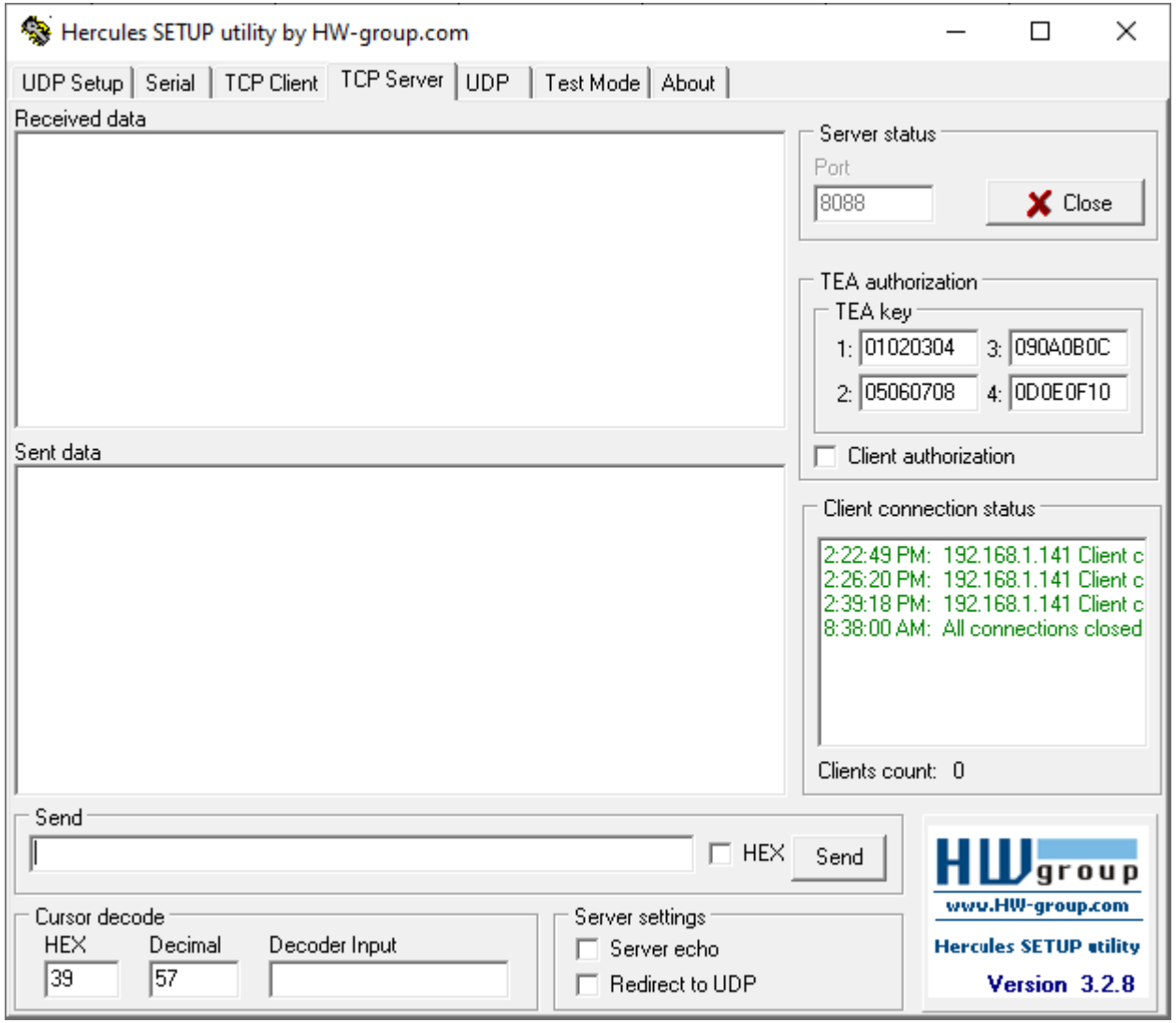
Connect to the Server Function

```
//*****************************************************************************
//* CONNECT TO THE SERVER
//*****************************************************************************
uint8_t connect2Server(void)
{
    rc = 0;
    UART2_ResetRxQueue();
    indx = 0;

    printf("AT+CIPSTART=\"TCP\",\"192.168.1.235\",8088\r\n");
    delay_ms(2000);

    if(chk_atok())
    {
      rc = 1;
    }
    return rc;
}
```



Send to the Server Function

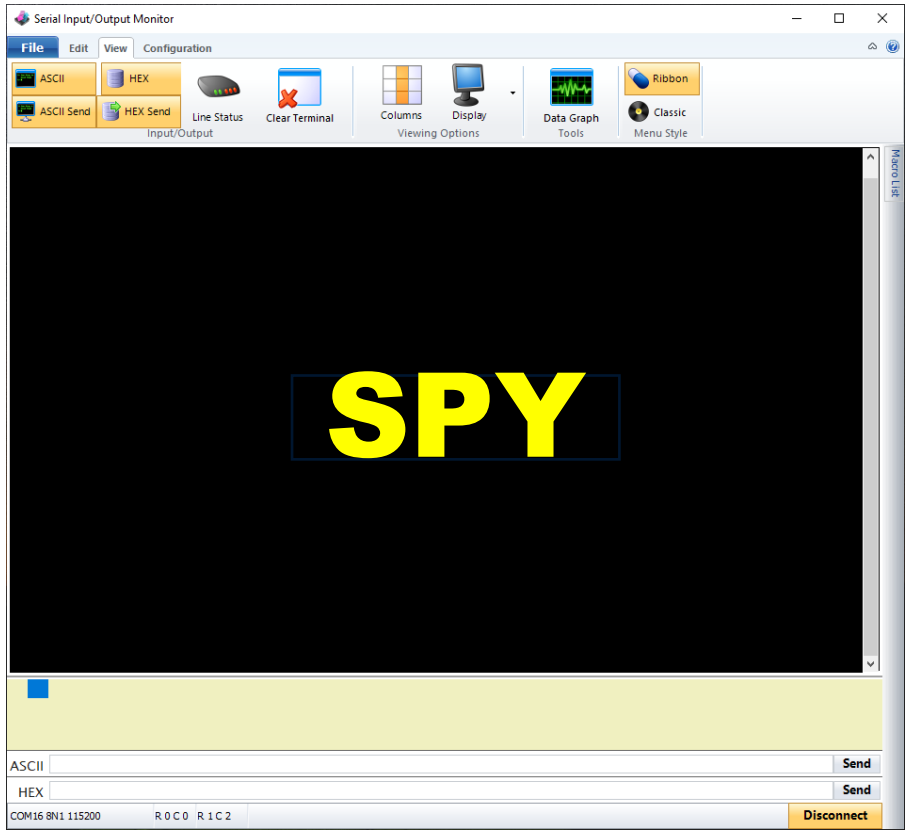
```
//*****************************************************************************
//* SEND DATA TO THE SERVER
//*****************************************************************************
uint8_t send2Server(void)
{
    uint8_t bite = 0x30;
    rc = 0;
    UART2_ResetRxQueue();
    indx = 0;
    printf("AT+CIPSEND=10\r\n");
    delay_ms(1000);
    do{
      if(UART2_IsRxReady())
      {
        rxBuf[indx++] = UART2_Read();
      }
    }while( UART2_IsRxReady());
    if(rxBuf[indx-2] == '>')
    {
      indx = 0;
      for(scratch8=0;scratch8<10;scratch8++)
      {
        printf("%c",bite++);
      }
      delay_ms(1000);
      do{
        if( UART2_IsRxReady())
        {
          rxBuf[indx++] = UART2_Read();
        }
      }while( UART2_IsRxReady());
      indx -= 3;
      if(rxBuf[indx] == 'K')
      {
        rc = 1;
      }
      return rc;
    }
}
```



Main Function

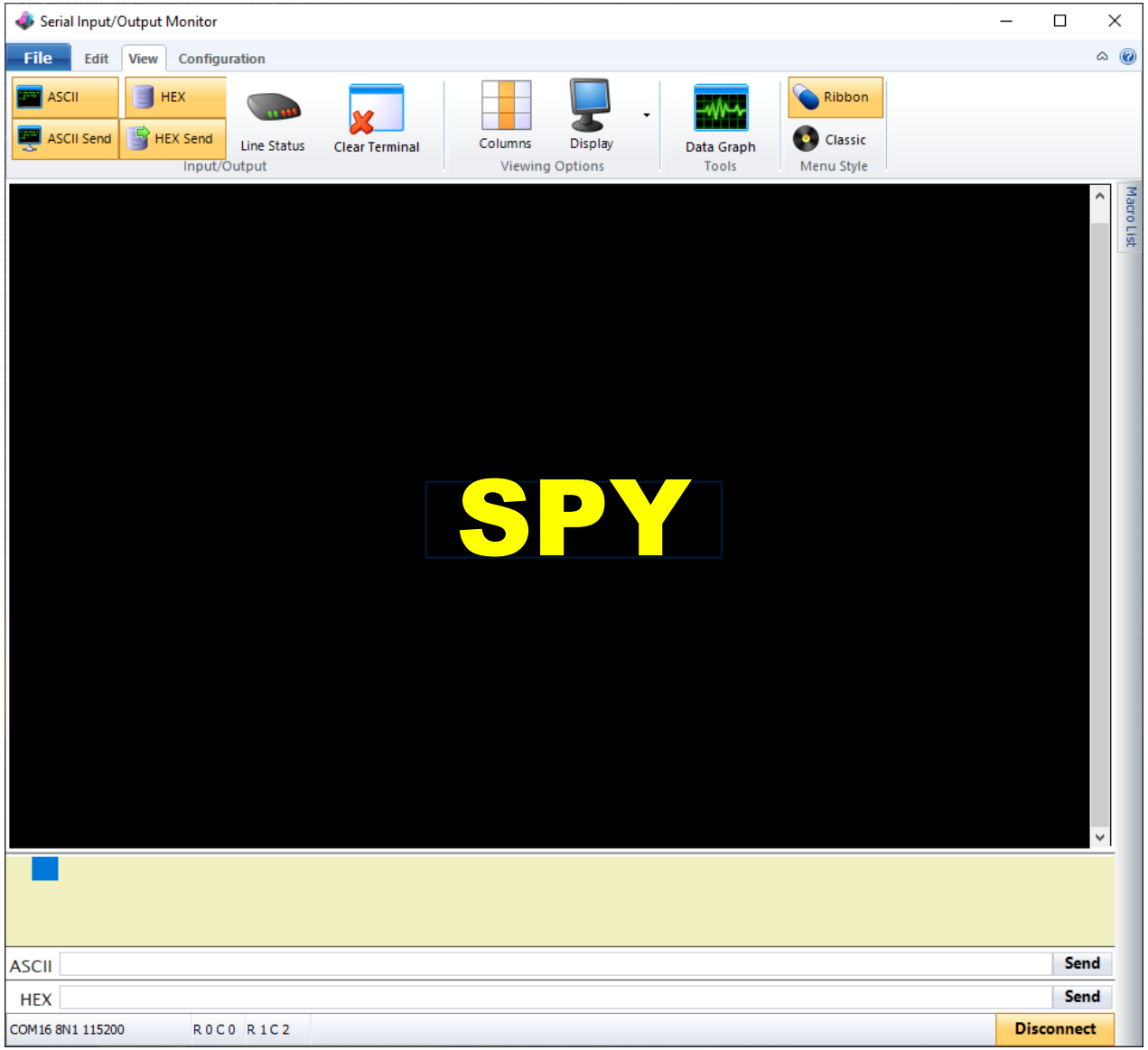
```
//****************************************
//* MAIN FUNCTION
//****************************************
int main()
{
    // initialize the device
    SYSTEM_Initialize();

    okat = 0x00;
    indx = 0x00;
    IO_RC2_RST_SetLow();
    IO_RC0_WP_SetLow();
    delay_ms(100);
    IO_RC2_RST_SetHigh();
    delay_ms(100);
    while(!UART2_IsRxReady());
    delay_ms(100);
    do{
        rxBuf[indx++] = UART2_Read();
    }while(UART2_IsRxReady());
```

```
//****************************************
//* MAIN FUNCTION
//****************************************
do{
        indx = 0;
        memset(rxBuf,0x00,sizeof(rxBuf));

        printf("AT\r\n");
        while(!UART2_IsRxReady());
        delay_ms(100);
        do{
            rxBuf[indx++] = UART2_Read();
        }while(UART2_IsRxReady());

        indx -= 3;
        if(rxBuf[indx] == 'K')
        {
            okat = 1;
        }
    }while(okat == 0);
```

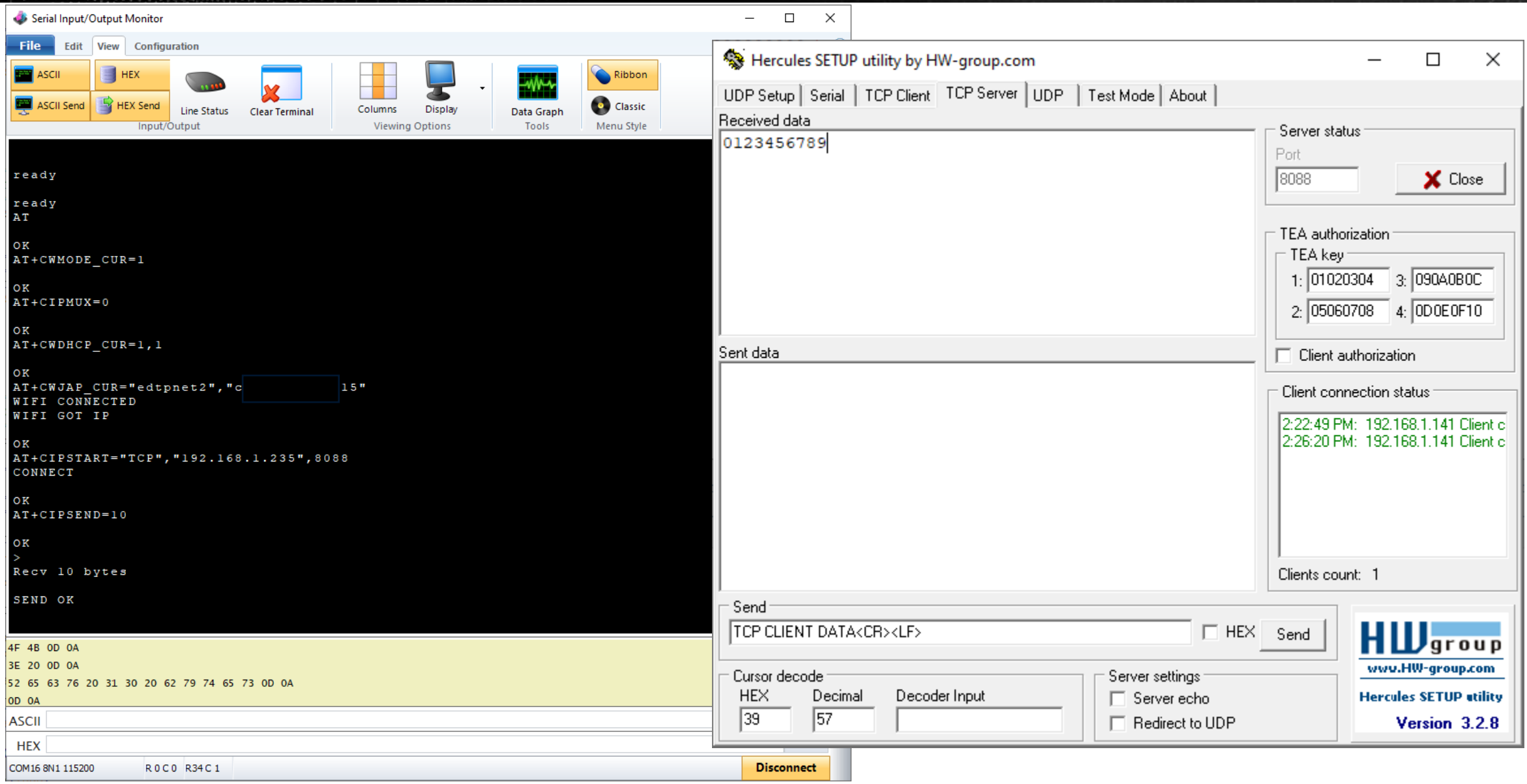


Main Function

```
if(!setStationMode())
{
  do{
    delay_ms(200);
    IO_RD3_LED_Toggle();
  }while(1);
}
if(!setSingleConnectionMode())
{
}
if(!setDhcpEnable())
{
}
if(!connect2AP())
{
}
if(!connect2Server())
{
}
if(!send2Server())
{
}
```



Execute the Main Function

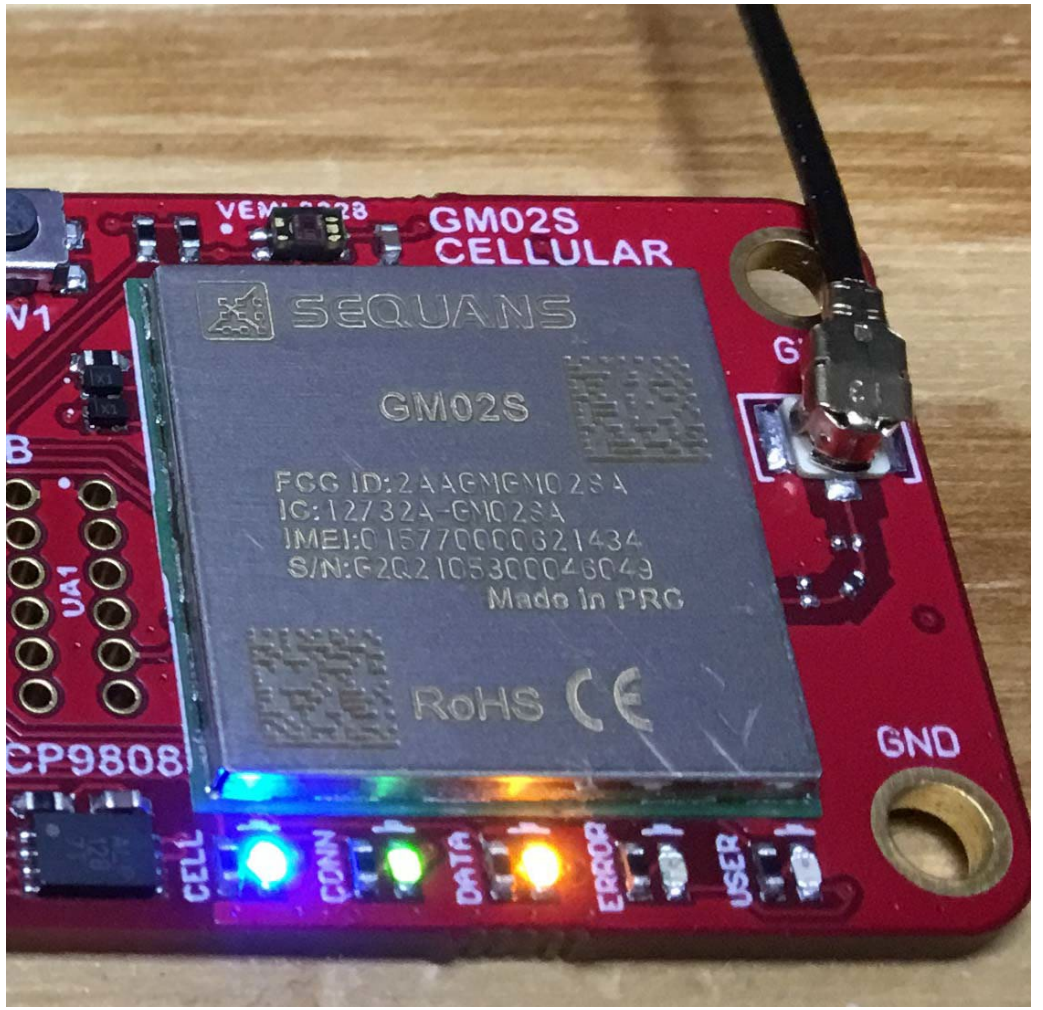


MORE TO COME..

Thank you for attending!!!

Please consider the resources below:

- microchip.com
- wiznet.io
- WizFi360 User Manual



CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

Digi-Key ELECTRONICS

informa markets