CEC Continuing Education Center

DesignNews

IoT Device Prototyping with Microchip Curiosity Development Boards

Day 2:

Curiosity Firmware Development Using the Microchip XC8 Compiler

Sponsored by

Digi-Key ELECTRONICS

informa markets

Webinar Logistics

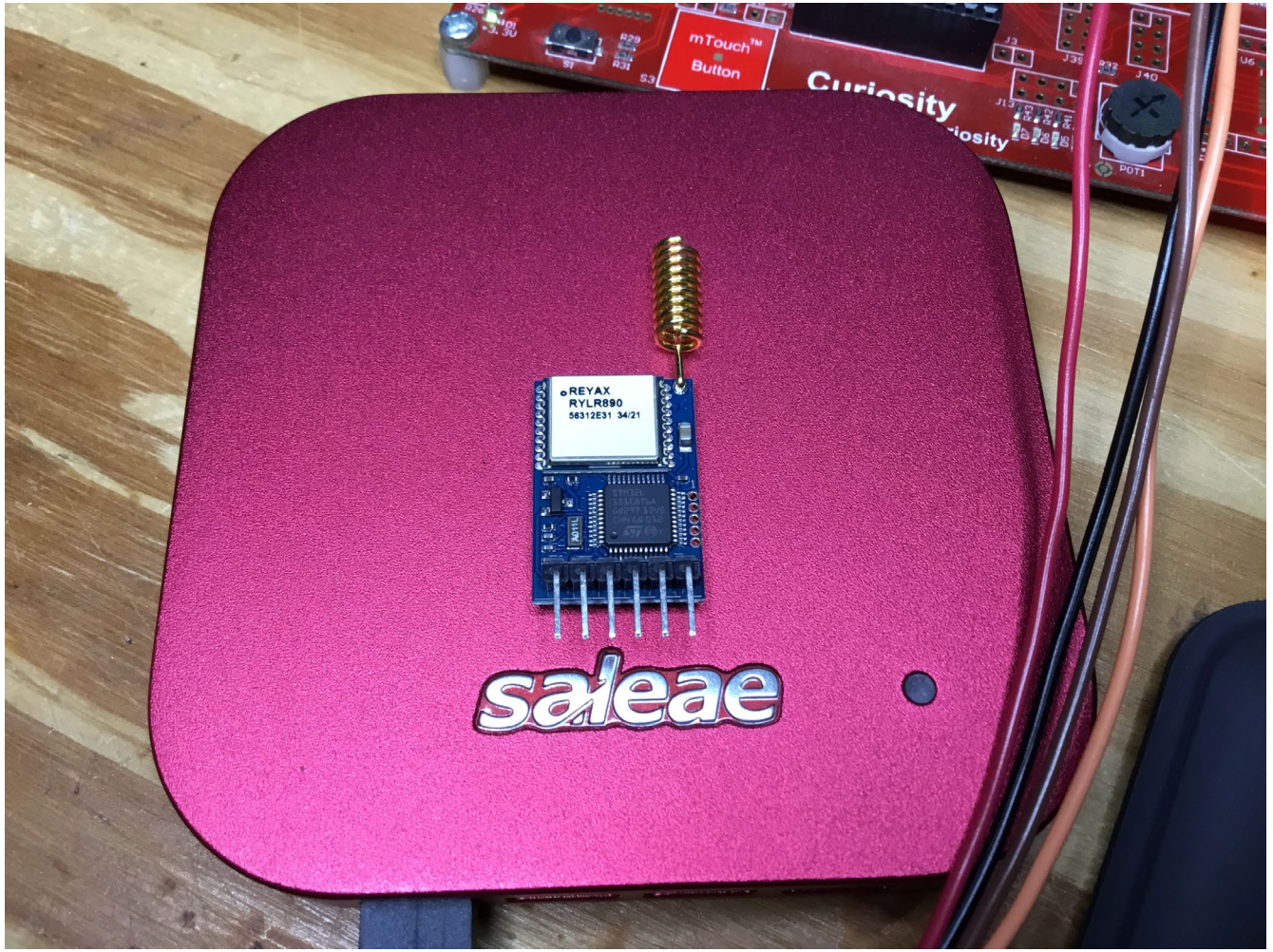
- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Fred Eady

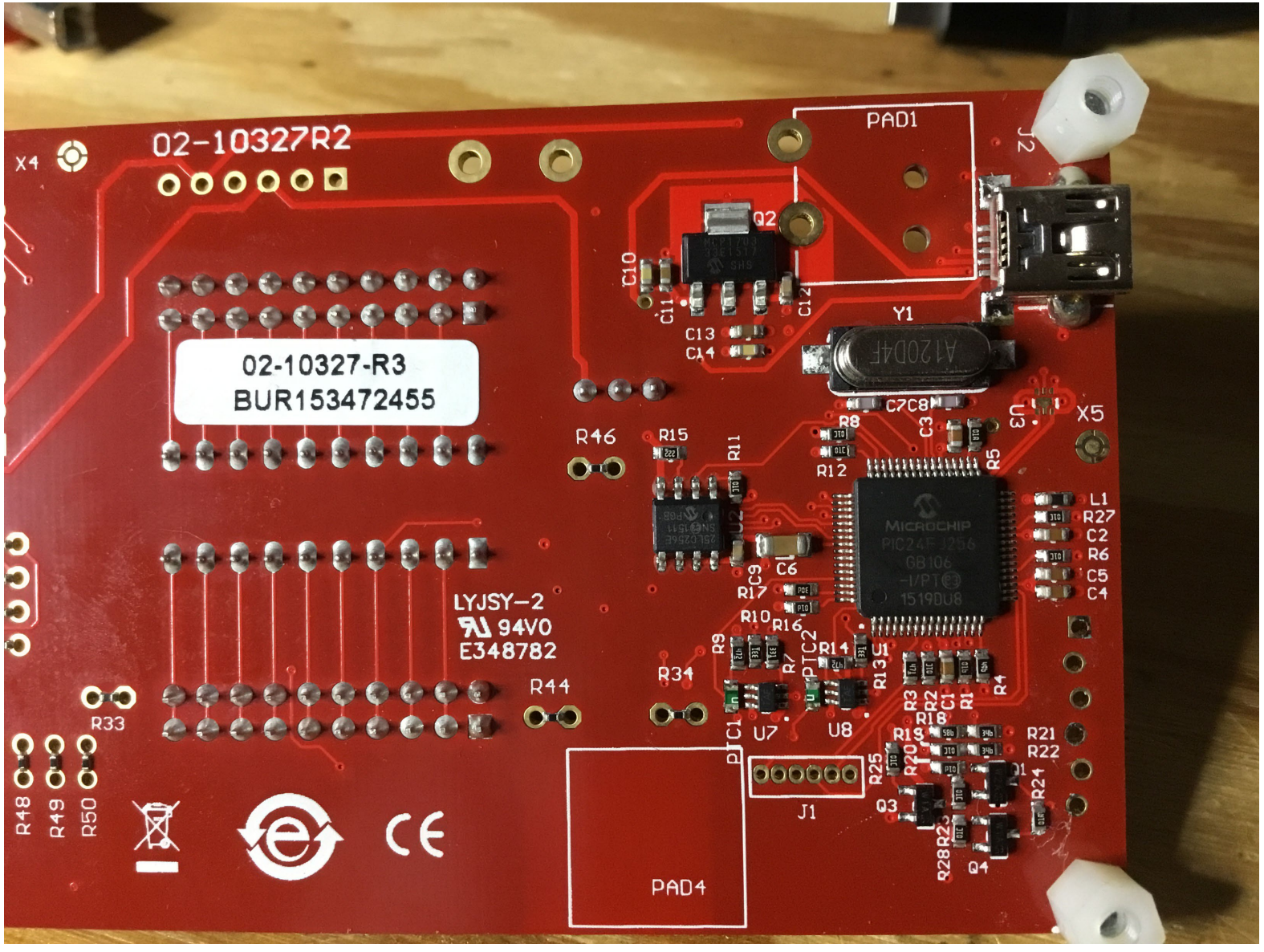
Visit ‘Lecturer Profile’ in your console for more details.

AGENDA

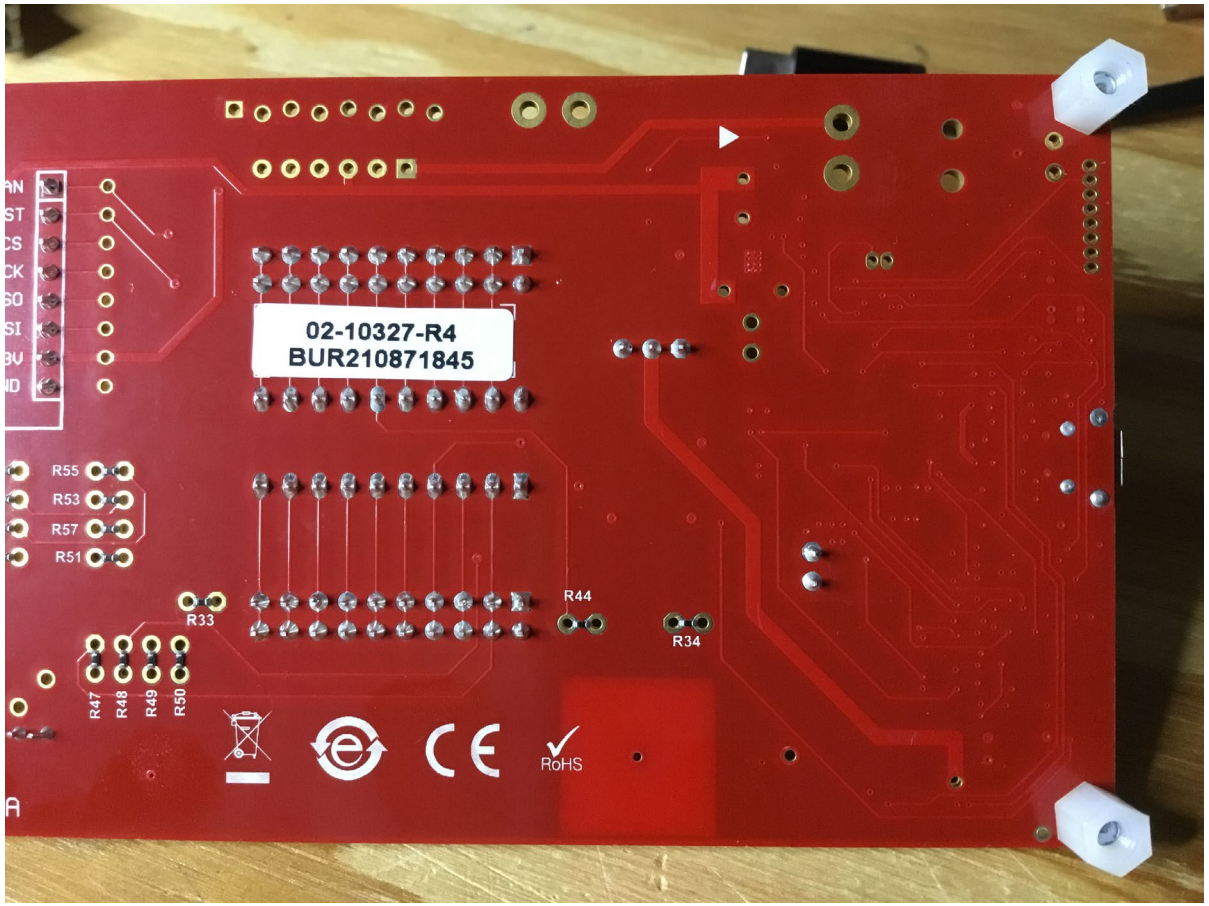
- Coding a PIC16F1619 REYAX RYLR890 Driver
- Coding a PIC16F18446 REYAX RYLR890 Driver

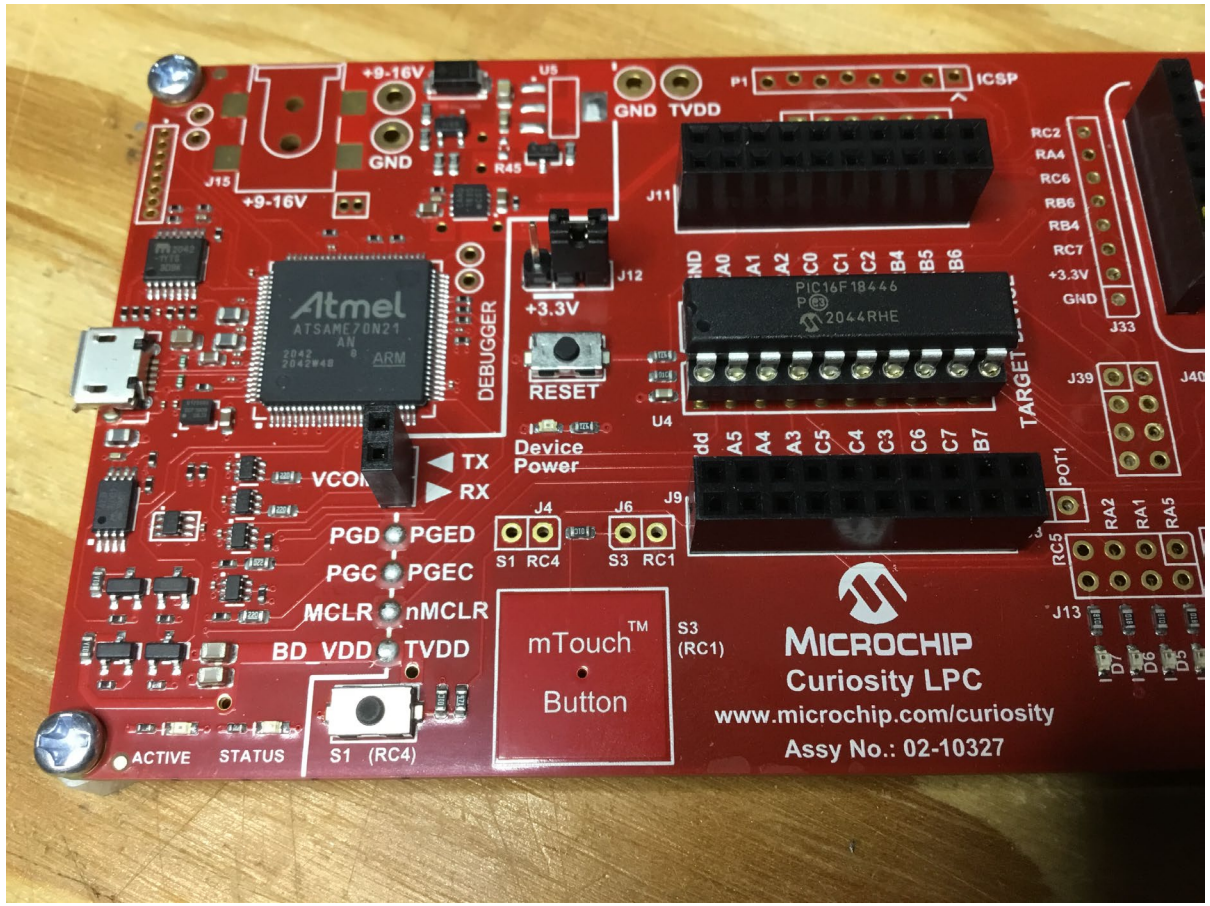


Legacy Curiosity

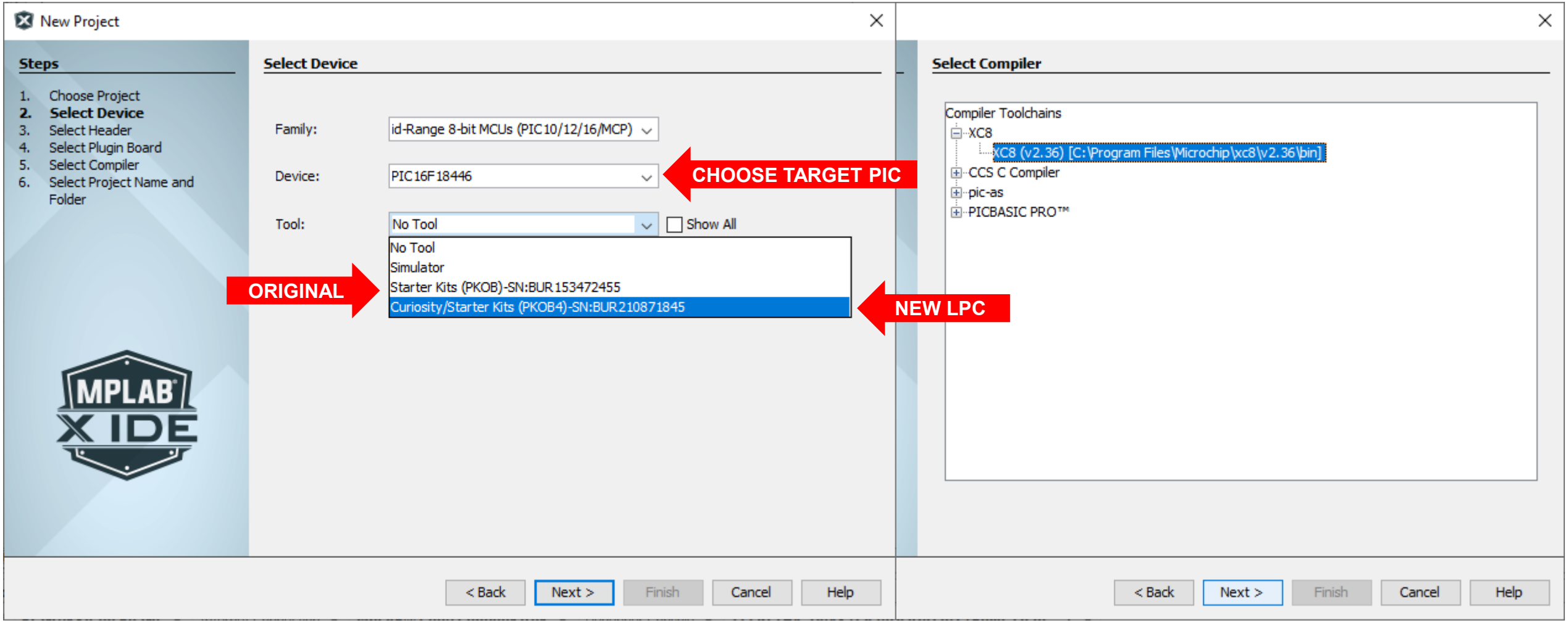


Curiosity LPC

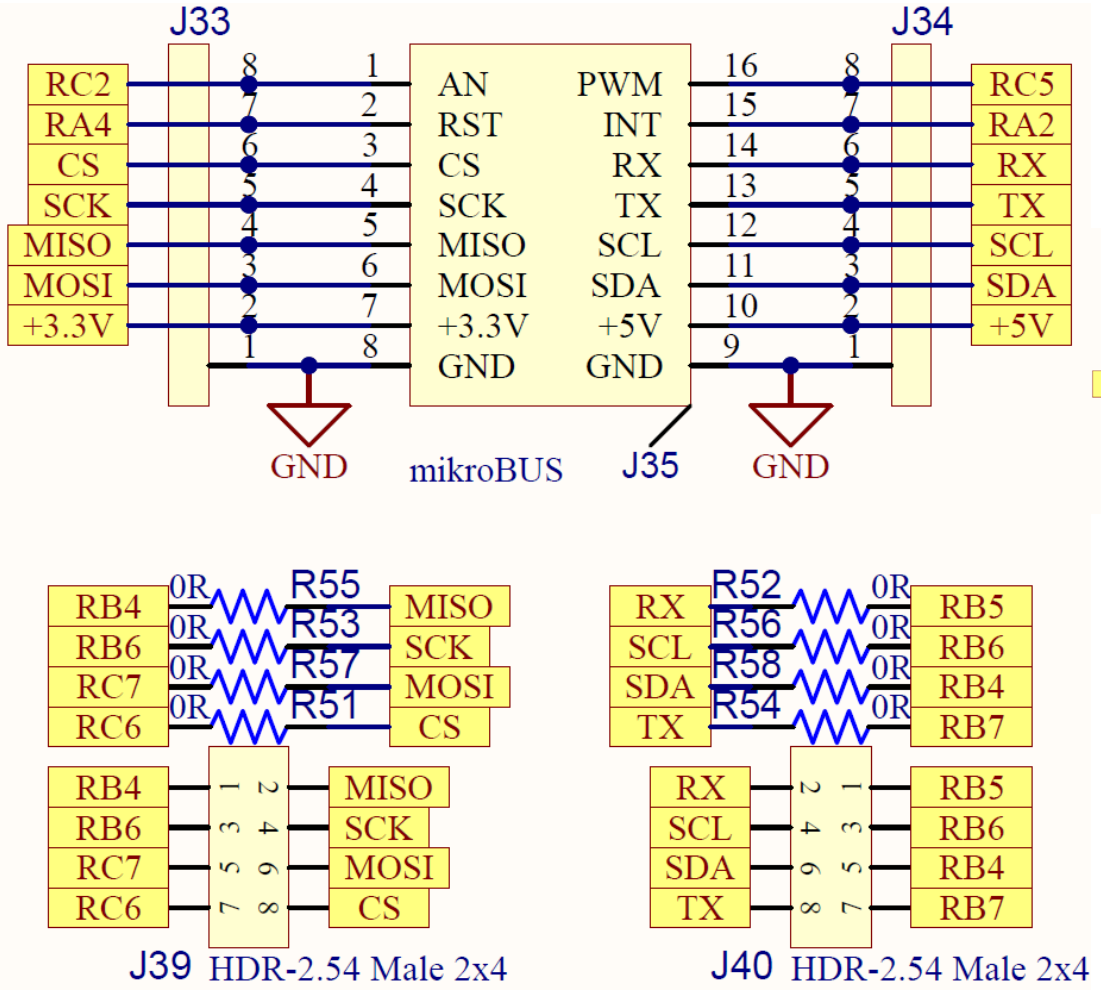


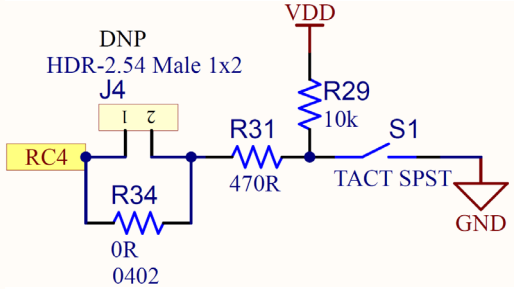


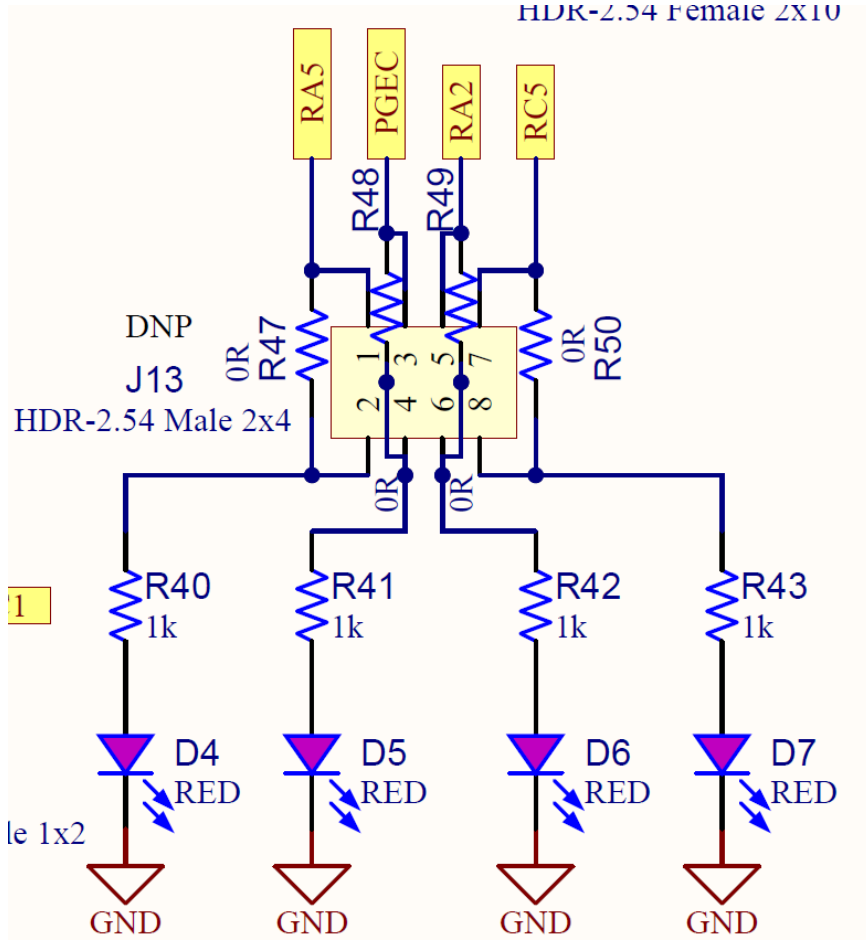
Create the PIC16F1619 MPLAB X Project



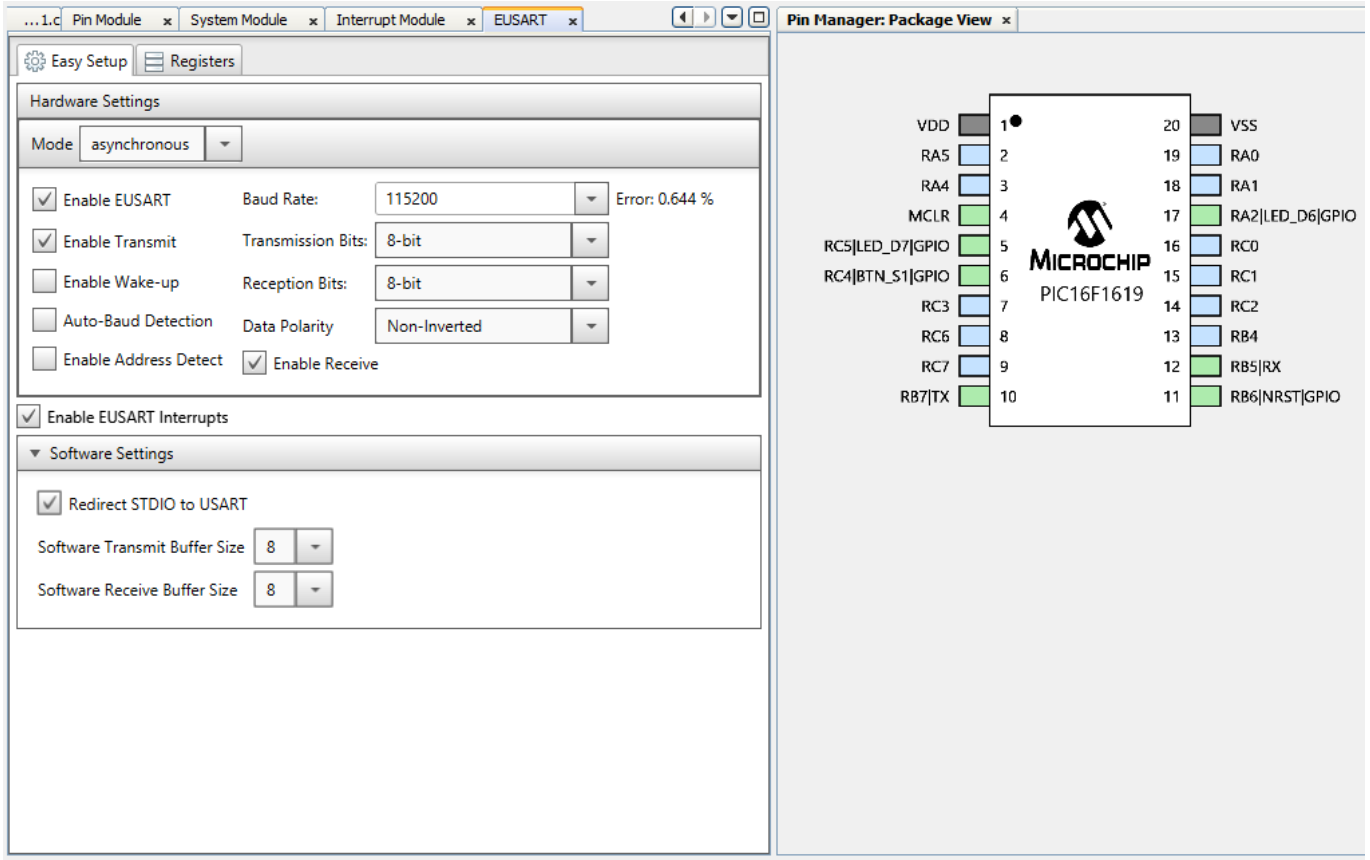
Curiosity Pin and LED Layouts



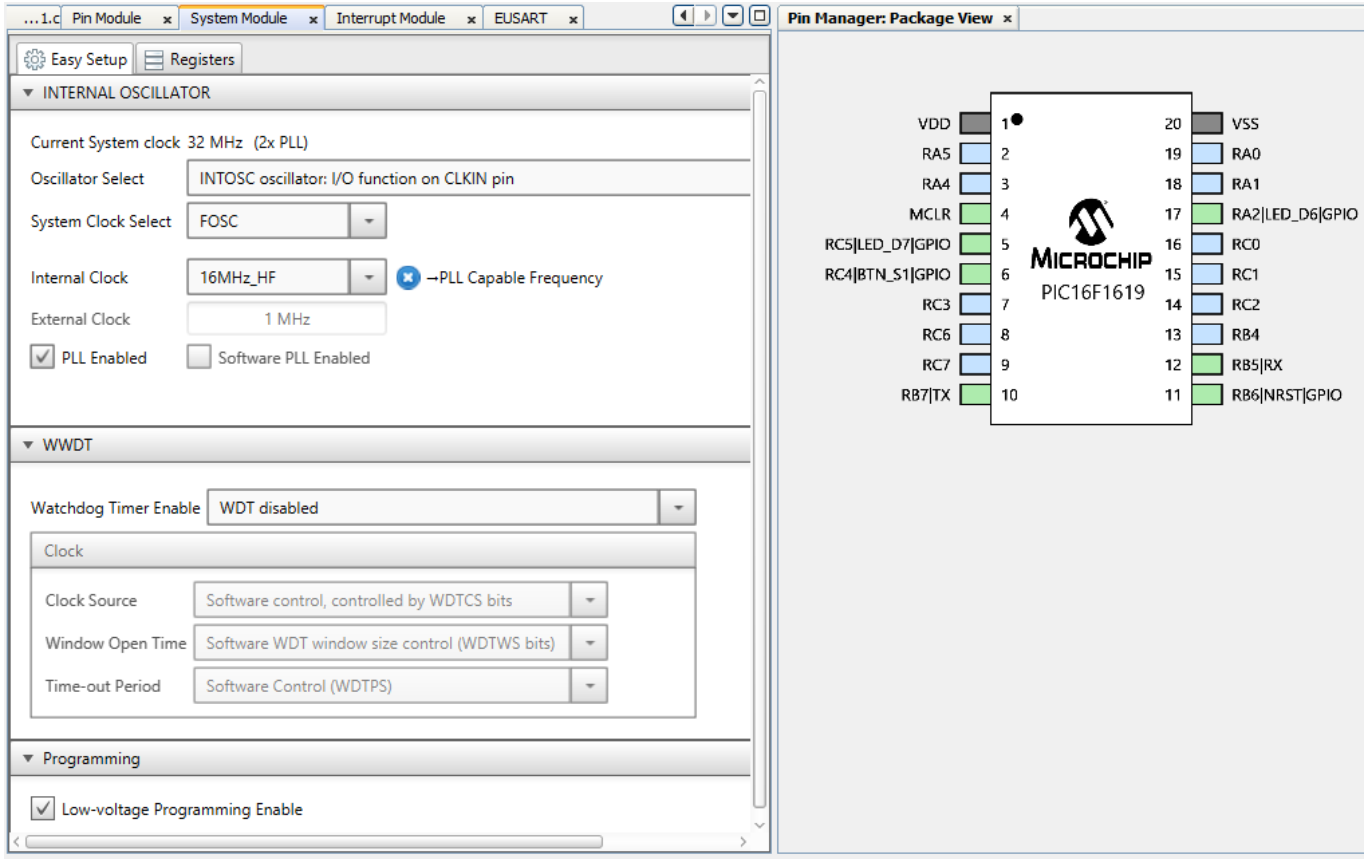




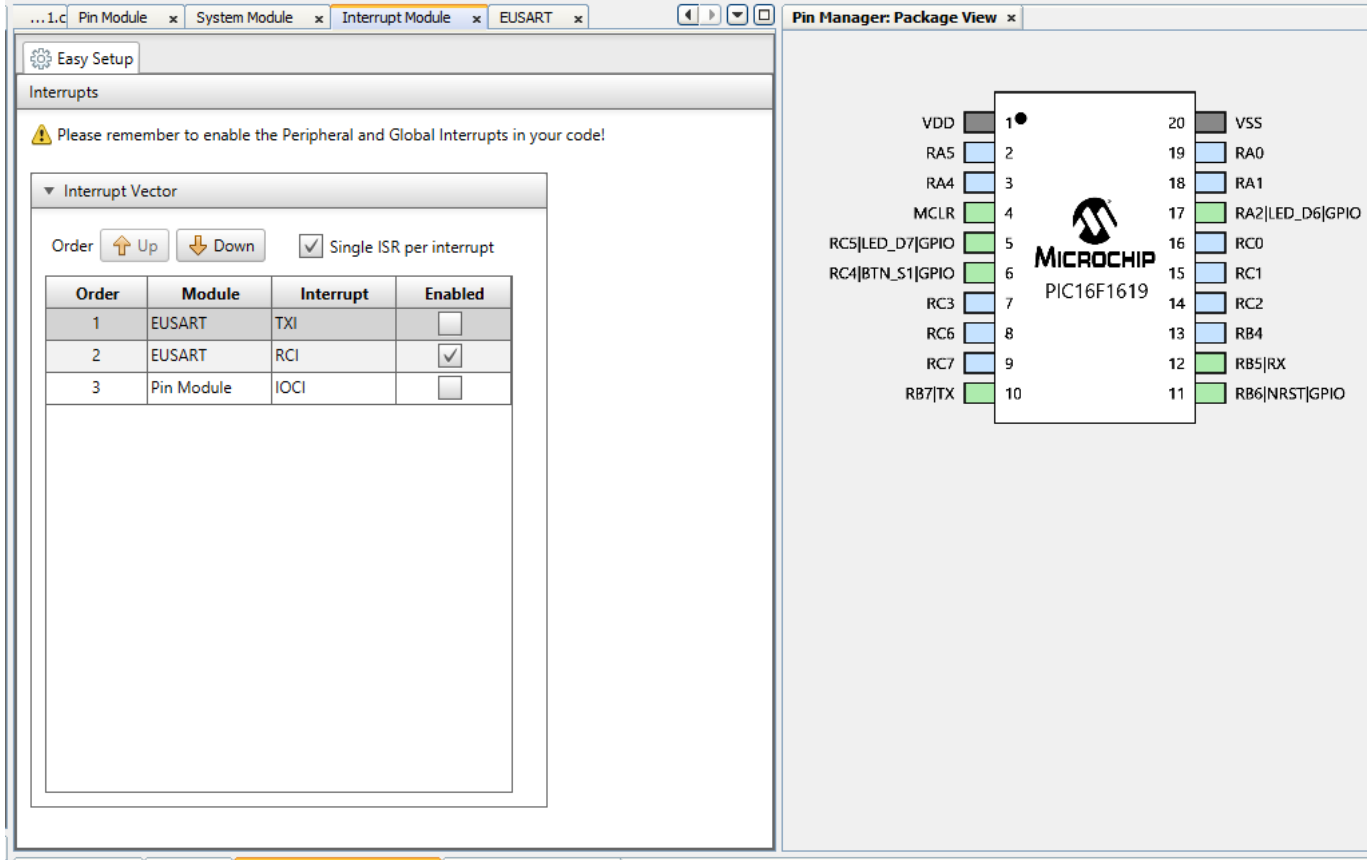
MCC Configuration – EUSART



MCC Configuration – Clock/Programming Mode



MCC Configuration – EUSART Interrupts



MCC-Generated Project Tree

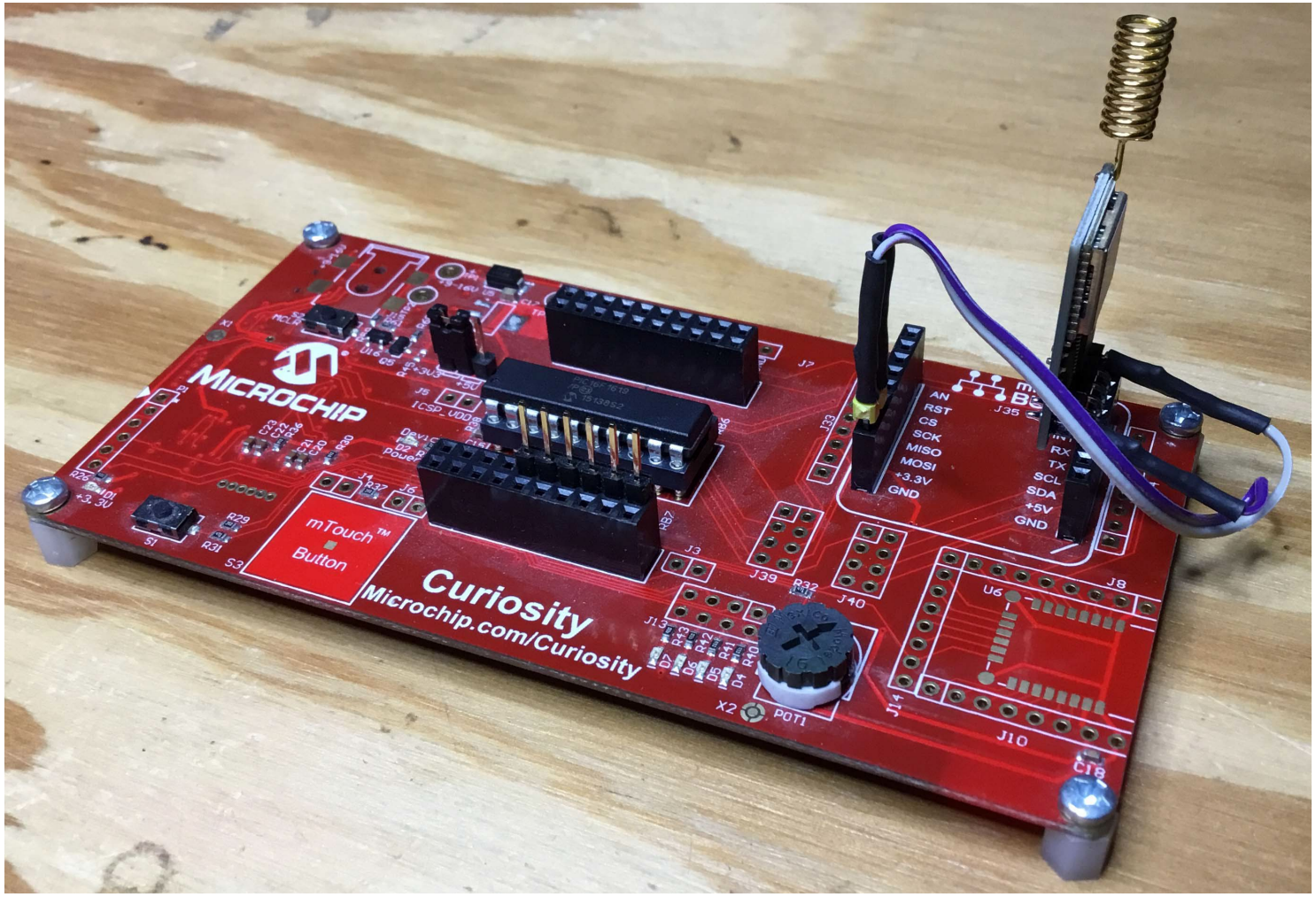
MPLAB X IDE v6.00 - cecREYAX_Day2 : default

File Edit View Navigate Source Refactor Production Debug Team Toc

default

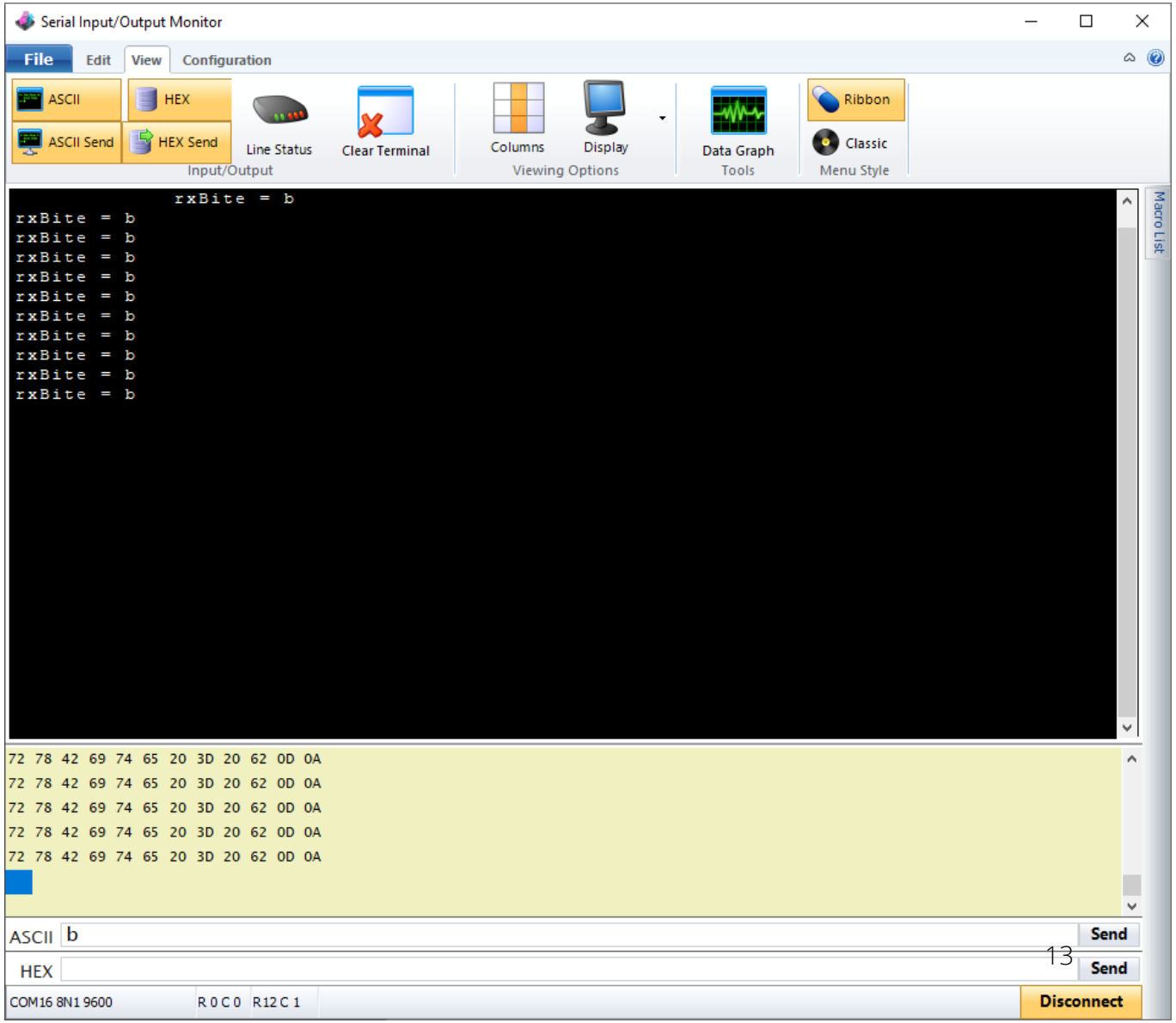
Projects × | Files | Services | Resource Management [MCC]

- cecREYAX_Day2
  - Header Files
    - MCC Generated Files
      - device_config.h
      - eusart.h
      - interrupt_manager.h
      - mcc.h
      - pin_manager.h
  - Important Files
  - Linker Files
  - Source Files
    - main.c
    - MCC Generated Files
      - device_config.c
      - eusart.c
      - interrupt_manager.c
      - mcc.c
      - pin_manager.c
  - Libraries
  - Loadables



EUSART/EUSART Interrupt Handler Verification

```
#include "mcc_generated_files/mcc.h"
uint8_t rxBite;
void main(void)
{
    // initialize the device
    SYSTEM_Initialize();

    // When using interrupts, you need to set the Global and Peripheral Interrupt Enable bits
    // Use the following macros to:

    // Enable the Global Interrupts
    INTERRUPT_GlobalInterruptEnable();

    // Enable the Peripheral Interrupts
    INTERRUPT_PeripheralInterruptEnable();

    // Disable the Global Interrupts
    //INTERRUPT_GlobalInterruptDisable();

    // Disable the Peripheral Interrupts
    //INTERRUPT_PeripheralInterruptDisable();

    while (1)
    {
        do{
            rxBite = EUSART_Read();
            printf("rxBite = %c\r\n",rxBite);
        }while(EUSART_is_rx_ready());
    }
}
```



EUSART Support Code

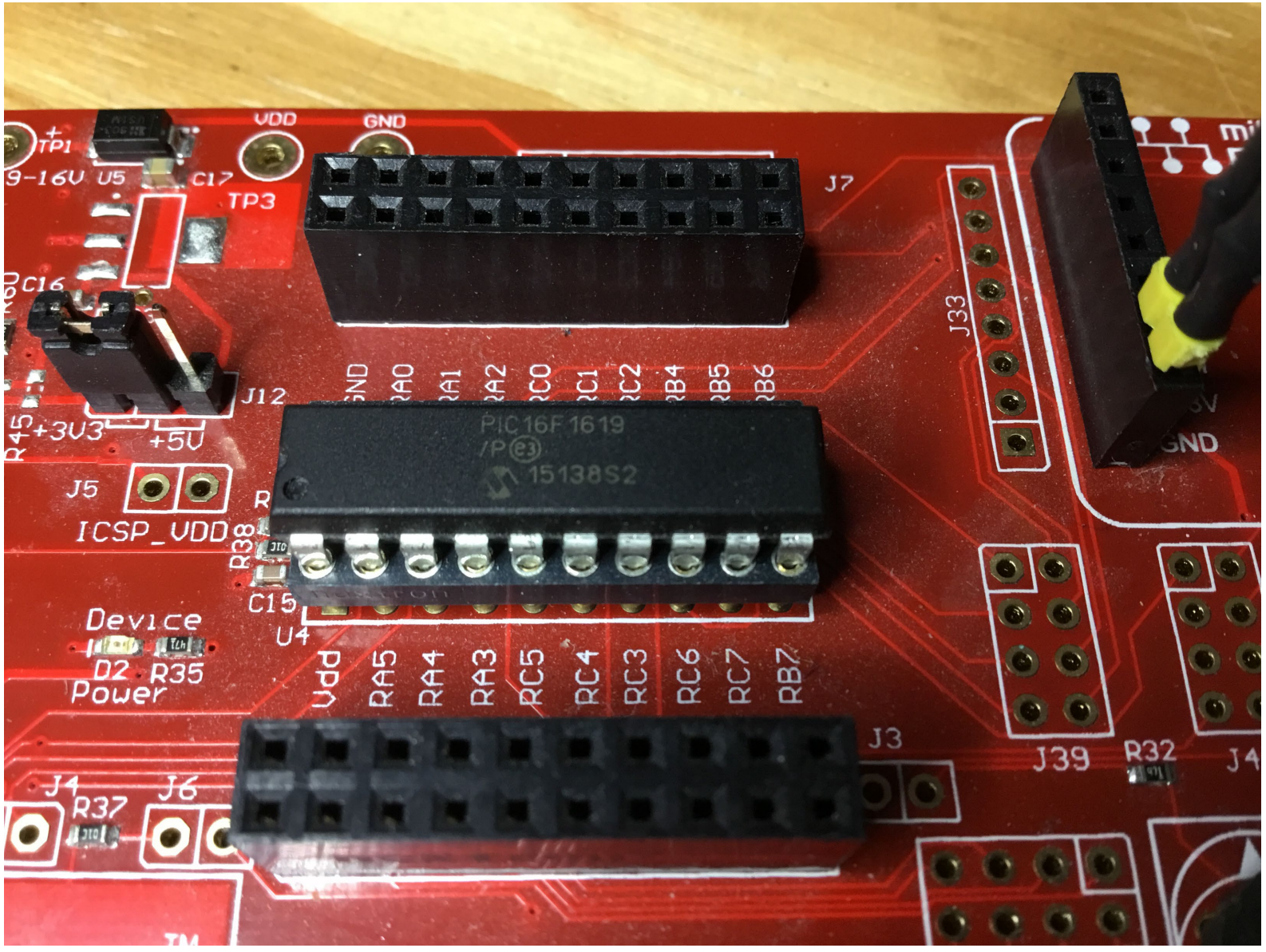
```
#include "mcc_generated_files/mcc.h“

uint8_t rxBuf[64];
uint8_t indx;
uint8_t loraState;
uint8_t txBite;

#define EUSART_RX_BUFFER_SIZE 64
extern volatile uint8_t
eusartRxBuffer[EUSART_RX_BUFFER_SIZE];

enum
{
    sWakeUp,
    sTx,
    sRx
};

void main(void)
{
SYSTEM_Initialize();

INTERRUPT_GlobalInterruptEnable();
INTERRUPT_PeripheralInterruptEnable();

loraState = sWakeUp;
```

eusart.c

```
#define EUSART_TX_BUFFER_SIZE 64
#define EUSART_RX_BUFFER_SIZE 64

/**
  Section: Global Variables
*/

volatile uint8_t eusartRxHead = 0;
volatile uint8_t eusartRxTail = 0;
volatile uint8_t
eusartRxBuffer[EUSART_RX_BUFFER_SIZE];
volatile eusart_status_t
eusartRxStatusBuffer[EUSART_RX_BUFFER_SIZE];
volatile uint8_t eusartRxCount;
volatile eusart_status_t eusartRxLastError;
```

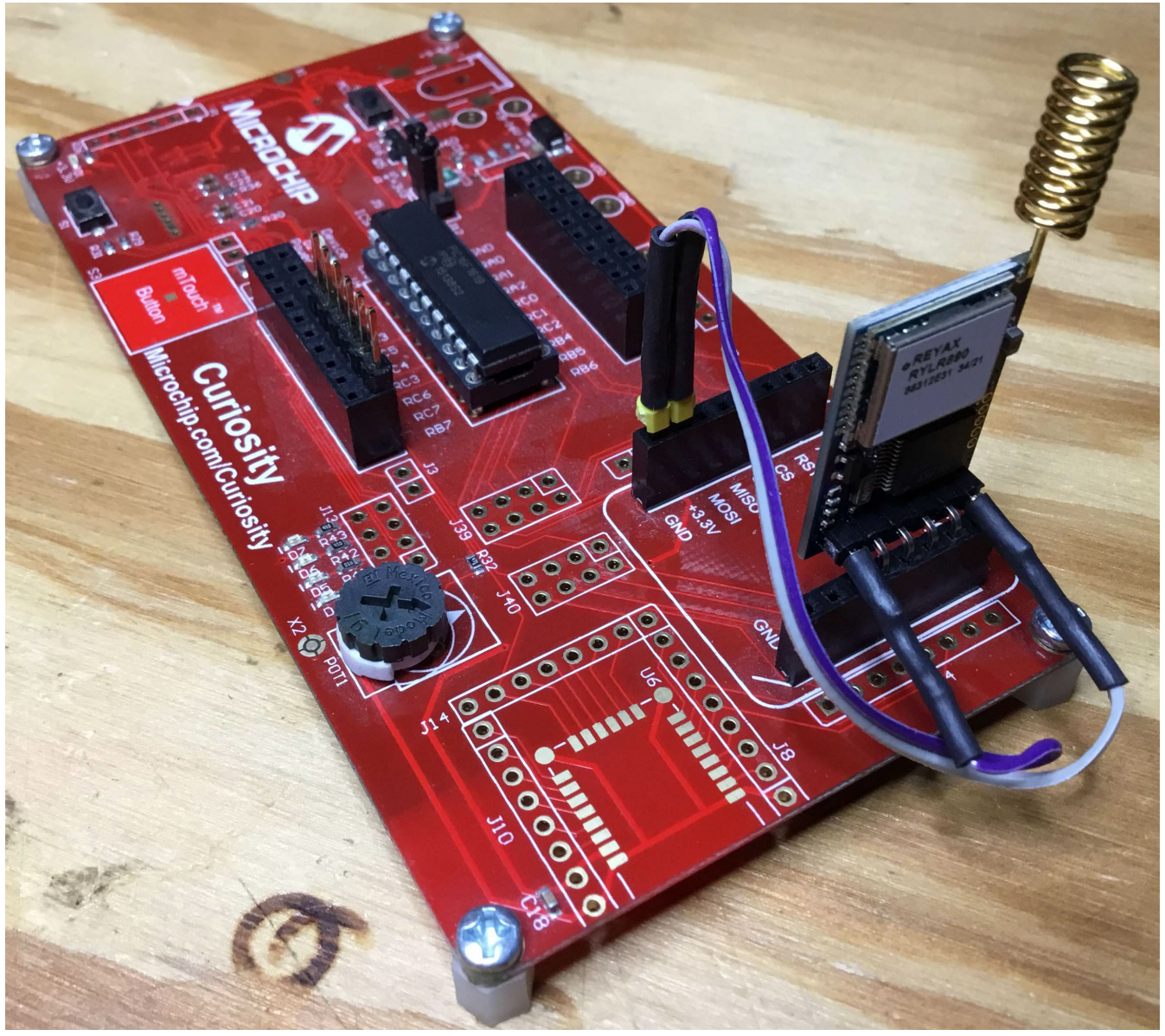
sWakeUp State

```
while (1)
   {
      //* State = sWakeUp
      if(loraState == sWakeUp)
      {
         indx = 0;

         NRST_SetLow();
         __delay_ms(10);
         NRST_SetHigh();

         while(EUSART_is_rx_ready() == false);
         __delay_ms(100);
         do{
            rxBuf[indx++] = EUSART_Read();
         }while(EUSART_is_rx_ready());

         indx = 0;
         memset(rxBuf,0x00,sizeof(rxBuf));
         printf("AT\r\n");

         while(EUSART_is_rx_ready() == false);
         __delay_ms(100);
         do{
            rxBuf[indx++] = EUSART_Read();
         }while(EUSART_is_rx_ready());

         indx -= 3;
         if(rxBuf[indx] == 'K')
         {
            loraState = sTx;
         }
      }
```

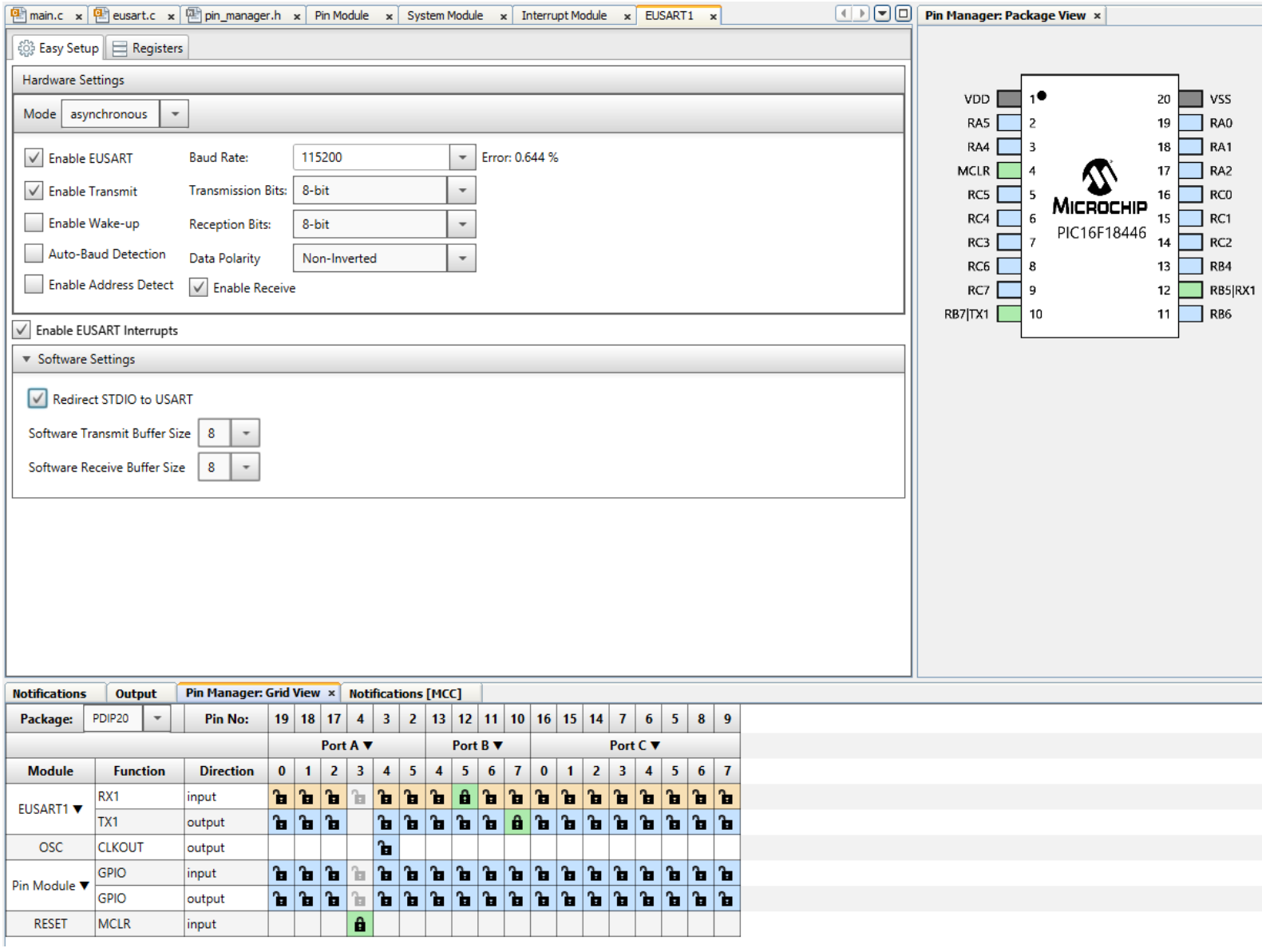


sTx State

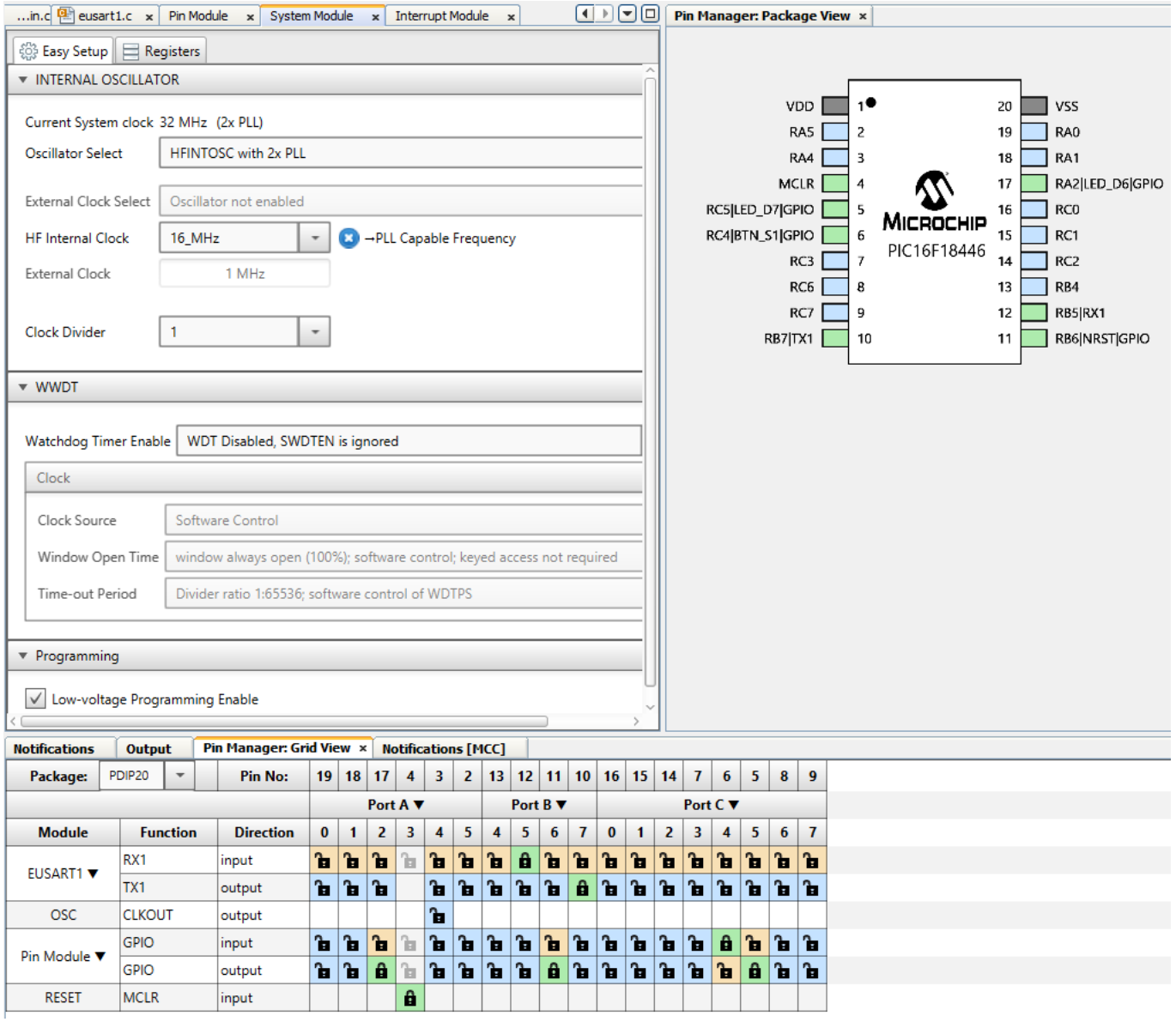
```
//* State = sTx
        if(loraState == sTx)
        {
            txBite = 0x42;
            printf("AT+SEND=0,1,%c\r\n",txBite);
            __delay_ms(1000);
            LED_D6_Toggle();
        }
    }//while(1))
}//main
```



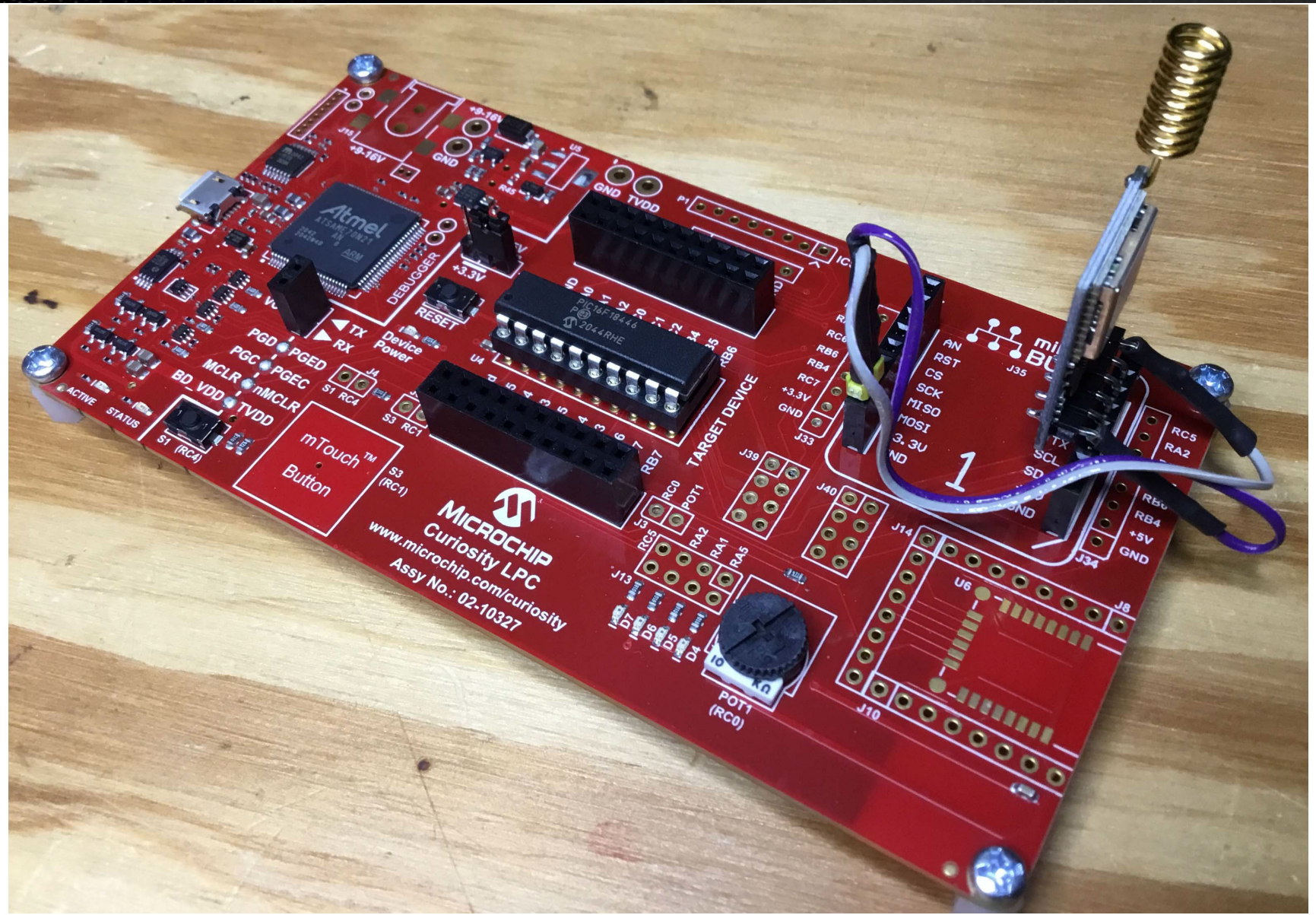
MCC Configuration – EUSART



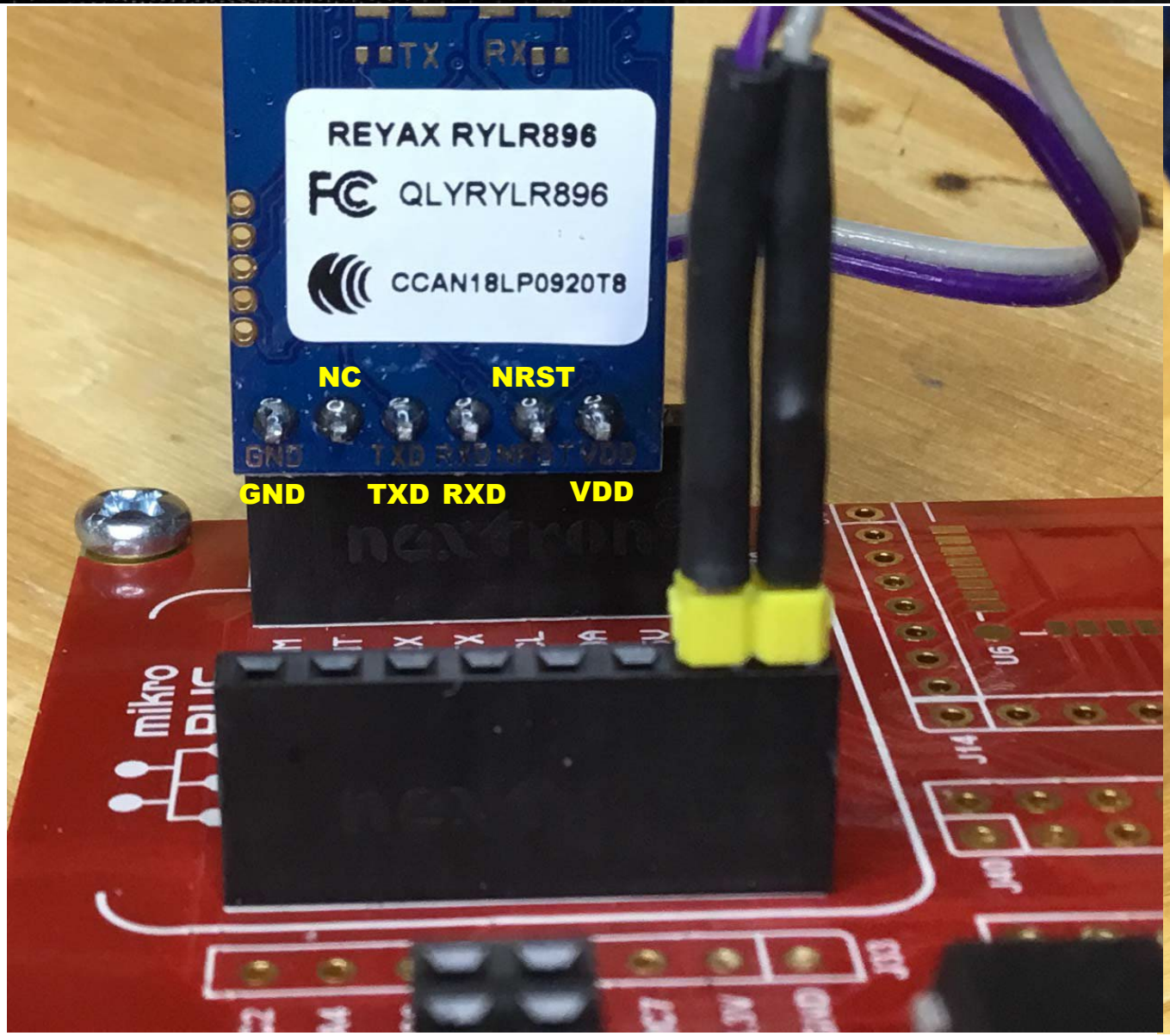
MCC Configuration – PIC16F18446

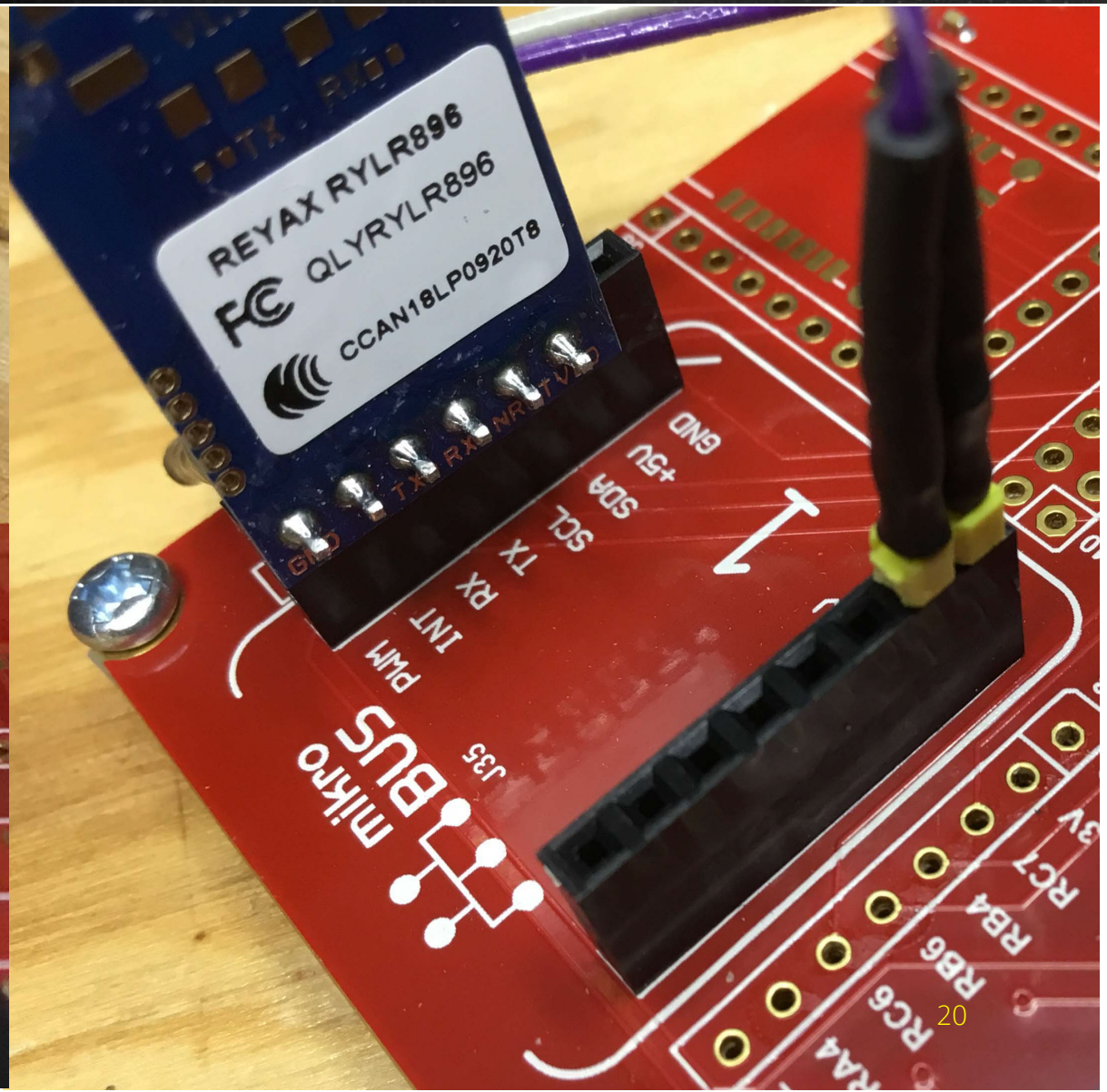


MCC Configuration – PIC16F18446

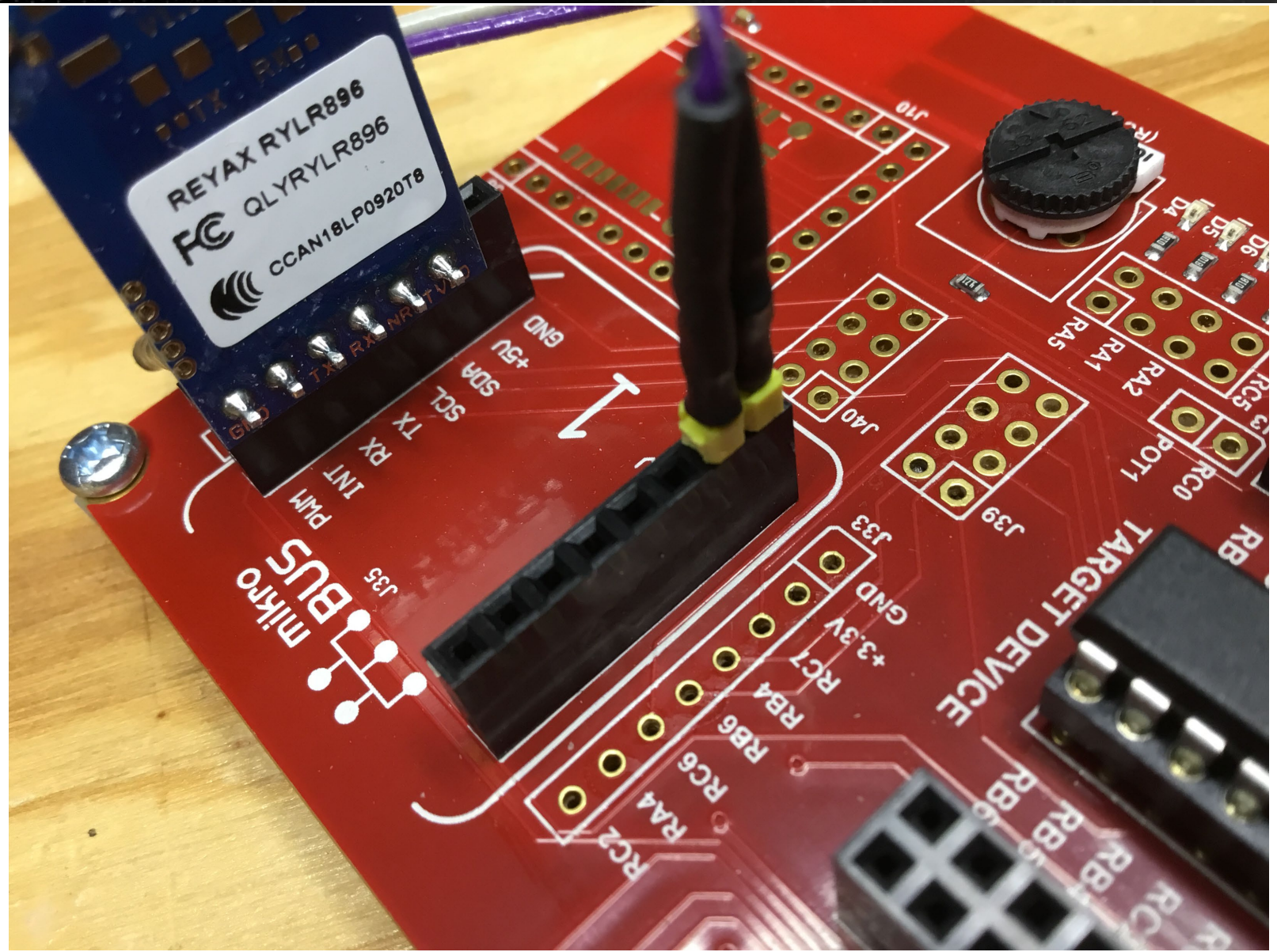


MCC Configuration – PIC16F18446

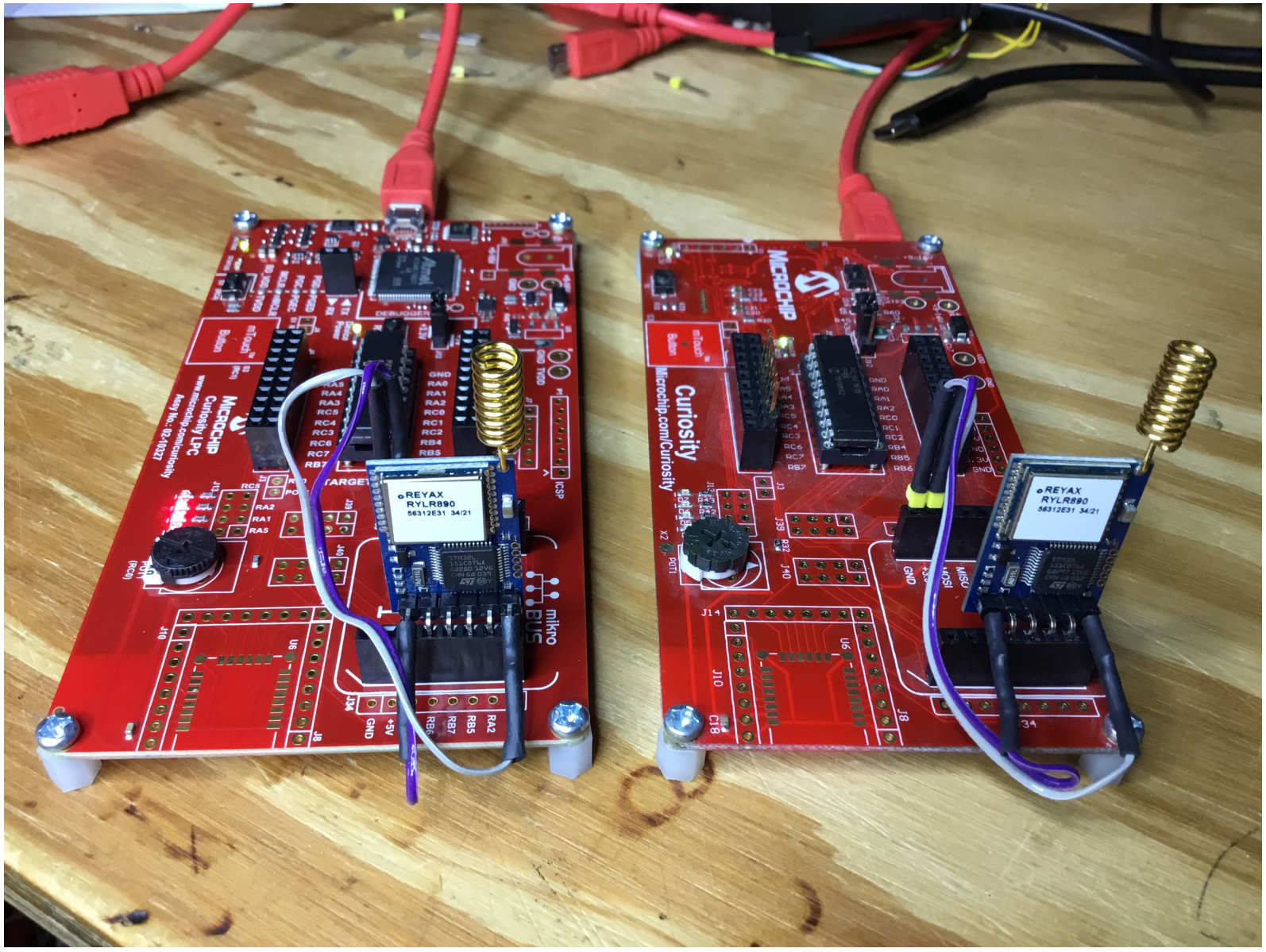




MCC Configuration – PIC16F18446



PIC16F18446 – PIC16F1619

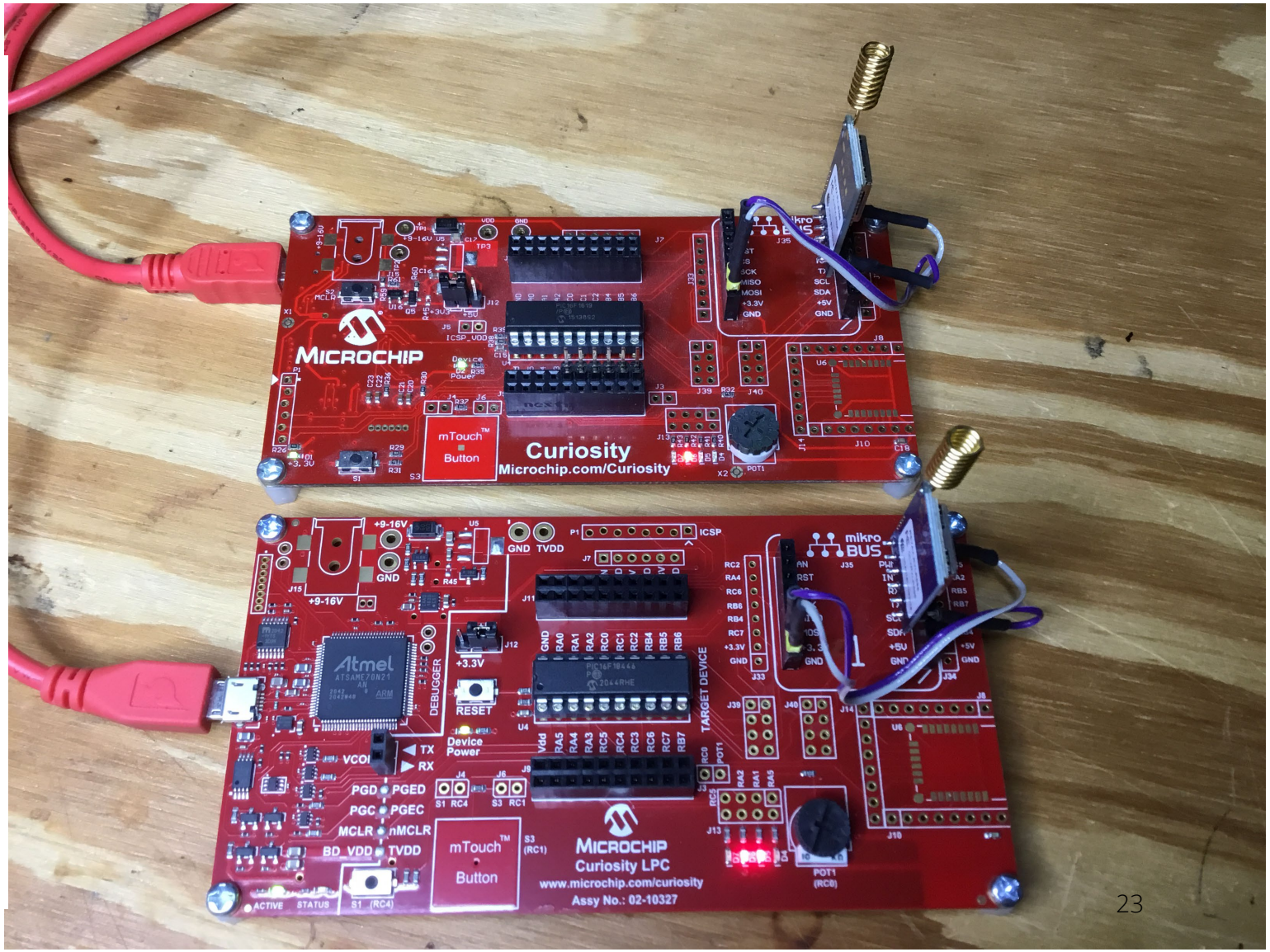


11. **AT+SEND** Send data to the appointment Address

Syntax	Response
AT+SEND=<Address>,<Payload Length>,<Data> <Address>0~65535, When the <Address> is 0, it will send data to all address (From 0 to 65535.) <Payload Length>Maximum 240bytes <Data>ASCII Format Example : Send HELLO string to the Address 50, AT+SEND=50,5,HELLO	+OK
AT+SEND?	+SEND=50,5,HELLO

12. +RCV Show the received data

Syntax	Response
+RCV=<Address>,<Length>,<Data>,<RSSI>,<SNR>, <Address>Transmitter Address ID <Length>Data Length <Data>Data <RSSI> Received Signal Strength Indicator <SNR> Signal-to-noise ratio	
Example: Module received the ID Address 50 send 5 bytes data, Content is HELLO string ,RSSI is -99dBm, SNR is 40, It will show as below: +RCV=50,5,HELLO,-99,40	



MCC Configuration – PIC16F18446

main.c | eusart1.c

Source | History

```
111             //* sRx
112             if(loraState == sRx)
113             {
114                 if(EUSART1_is_rx_ready());
115                 {
116                     indx = 0;
117                     __delay_ms(100);
118                     do{
119                         rxBuf[indx++] = EUSART1_Read();
120                     }while(EUSART1_is_rx_ready());
121                 }
122
123                 if(indx)
124                 {
125                     __delay_ms(100);
126                     LED_D6_Toggle();
127                 }
128             }
129         }//while(1))
130     }
131     /**
132      End of File
133     */
```

Variables

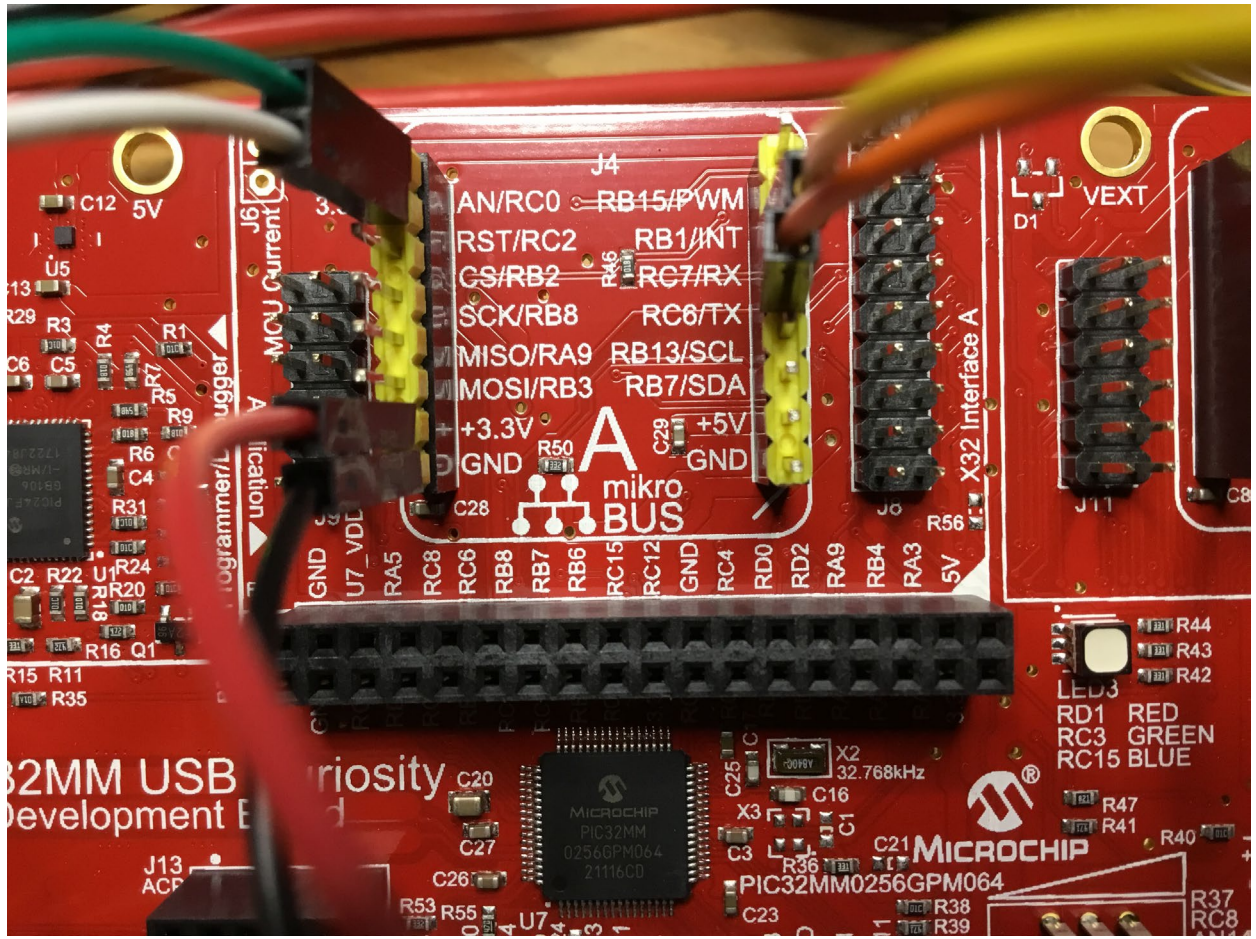
Name	Type	Address	Value
rxBuf; file:C:\Users\Public\cecCuriosi	unsigned char[64]	0x120	"RCV=0,1,B,-48,42\r\n\\0\\0
rxBuf[0]	unsigned char	0x120	'R'; 0x52
rxBuf[1]	unsigned char	0x121	'C'; 0x43
rxBuf[2]	unsigned char	0x122	'V'; 0x56
rxBuf[3]	unsigned char	0x123	'='; 0x3d
rxBuf[4]	unsigned char	0x124	'0'; 0x30
rxBuf[5]	unsigned char	0x125	','; 0x2c
rxBuf[6]	unsigned char	0x126	'1'; 0x31
rxBuf[7]	unsigned char	0x127	','; 0x2c
rxBuf[8]	unsigned char	0x128	'B'; 0x42
rxBuf[9]	unsigned char	0x129	','; 0x2c
rxBuf[10]	unsigned char	0x12A	'-'; 0x2d
rxBuf[11]	unsigned char	0x12B	'4'; 0x34
rxBuf[12]	unsigned char	0x12C	'8'; 0x38
rxBuf[13]	unsigned char	0x12D	','; 0x2c
rxBuf[14]	unsigned char	0x12E	'4'; 0x34
rxBuf[15]	unsigned char	0x12F	'2'; 0x32
rxBuf[16]	unsigned char	0x130	\r; 0xd
rxBuf[17]	unsigned char	0x131	\n; 0xa
rxBuf[18]	unsigned char	0x132	NUL; 0x0
rxBuf[19]	unsigned char	0x133	NUL; 0x0
rxBuf[20]	unsigned char	0x134	NUL; 0x0
rxBuf[21]	unsigned char	0x135	NUL; 0x0
rxBuf[22]	unsigned char	0x136	NUL; 0x0
rxBuf[23]	unsigned char	0x137	NUL; 0x0
rxBuf[24]	unsigned char	0x138	NUL; 0x0

MORE TO COME..

Thank you for attending!!!

Please consider the resources below:

- microchip.com
- REYAX LoRa AT COMMAND GUIDE
- lemosint.com



CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

Digi-Key ELECTRONICS

informa markets