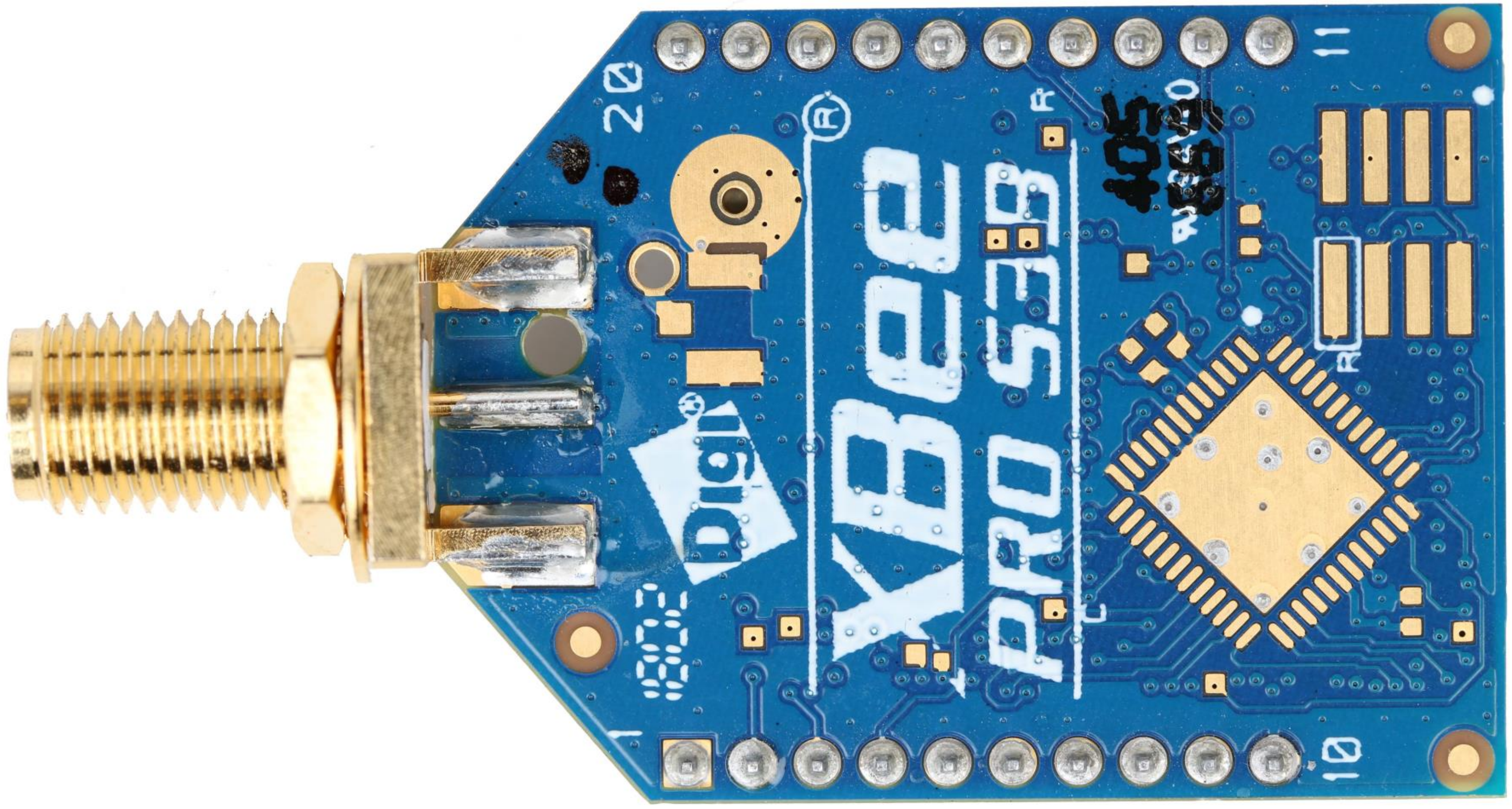
XBee Radio Modules



Building a Battery-Optimized High Power XBee Node

January 29, 2020

Fred Eady

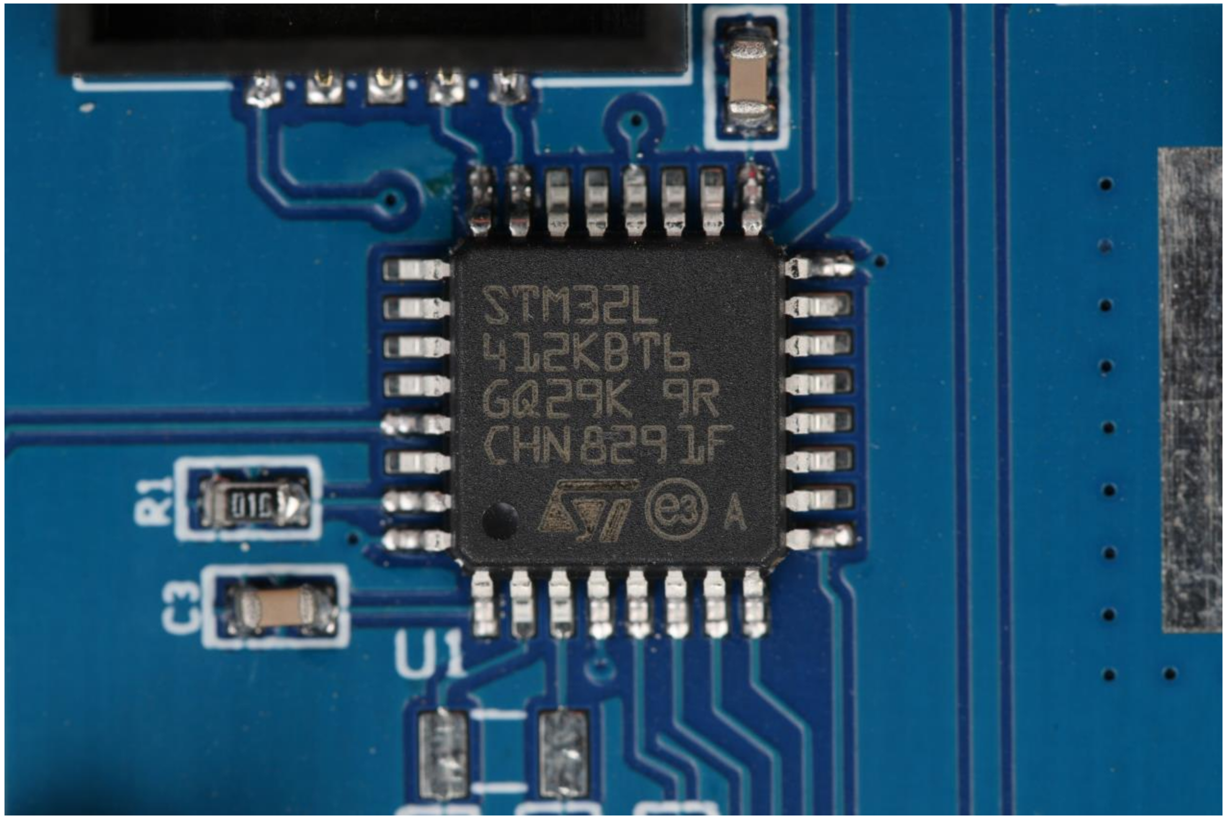
Presented by:

Information Classification: General

XBee Radio Modules

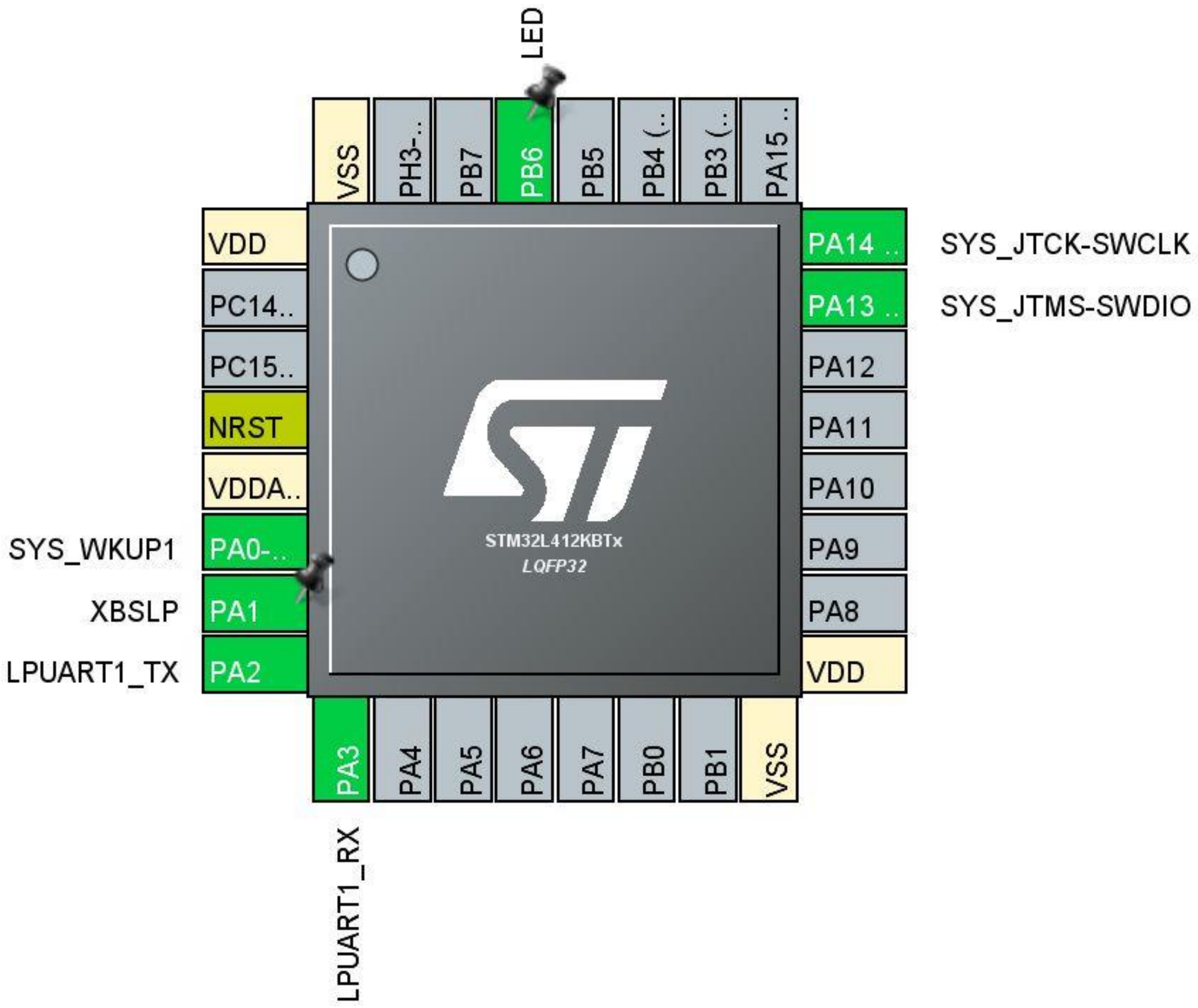
AGENDA

- XBee Hardware Design
- XBee Application Code
- Day 3 Summary



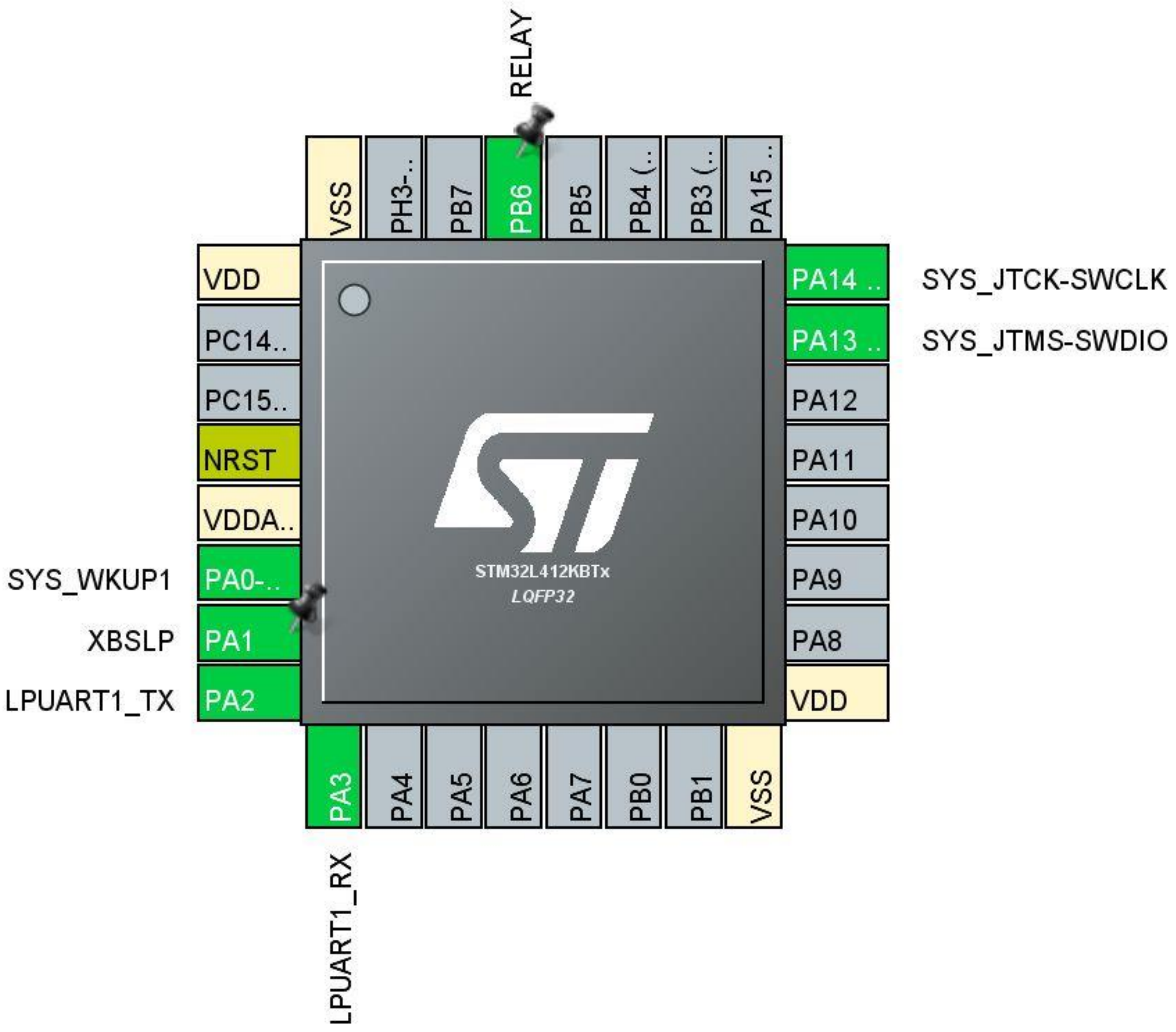
XBee Radio Modules

XBee Hardware Design – Transmitter Variant



XBee Radio Modules

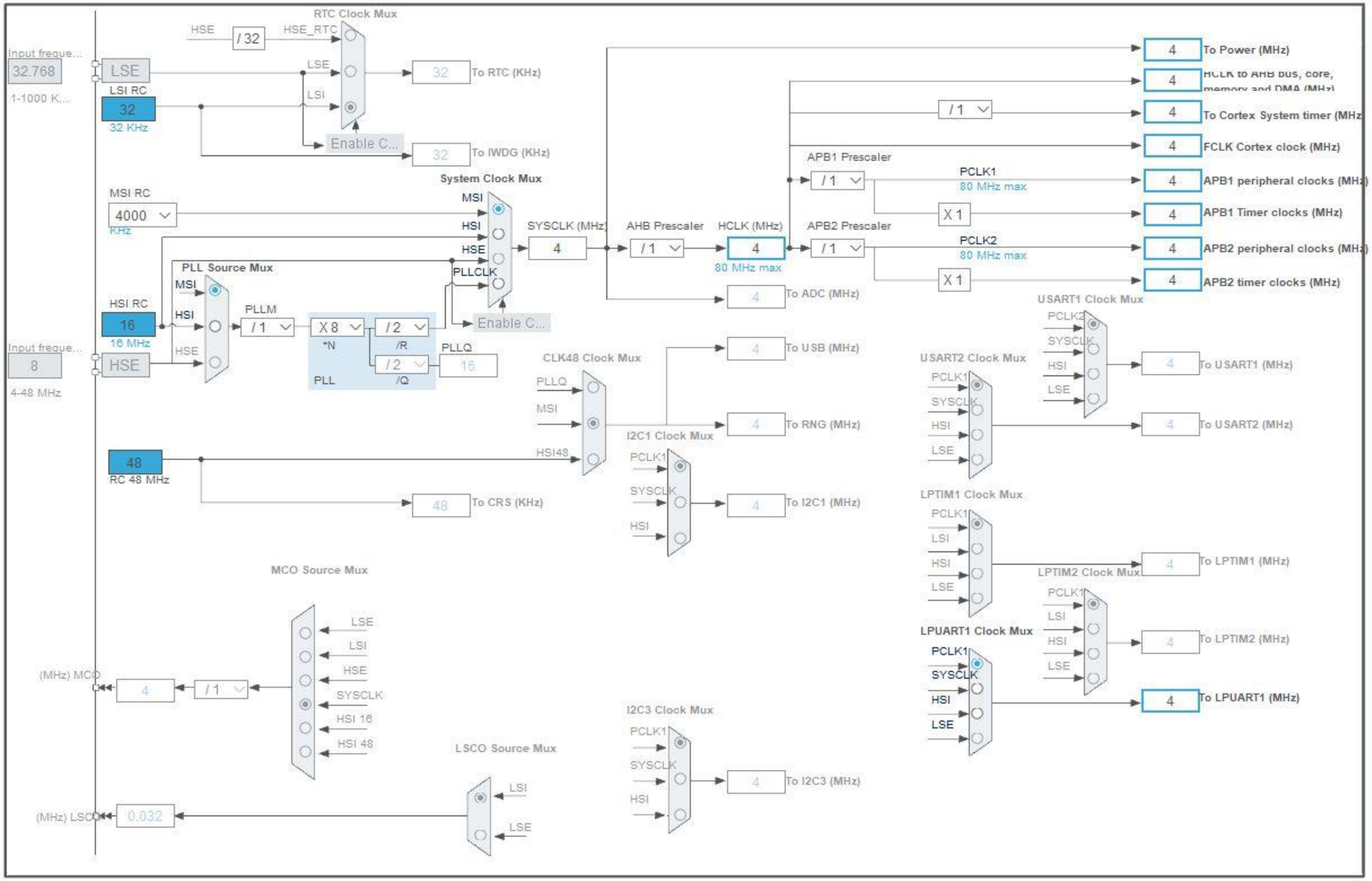
XBee Hardware Design – Receiver Variant



Presented by:

XBee Radio Modules

XBee Hardware Design – Transmitter/Receiver



Presented by:

DesignNews

Information Classification: General

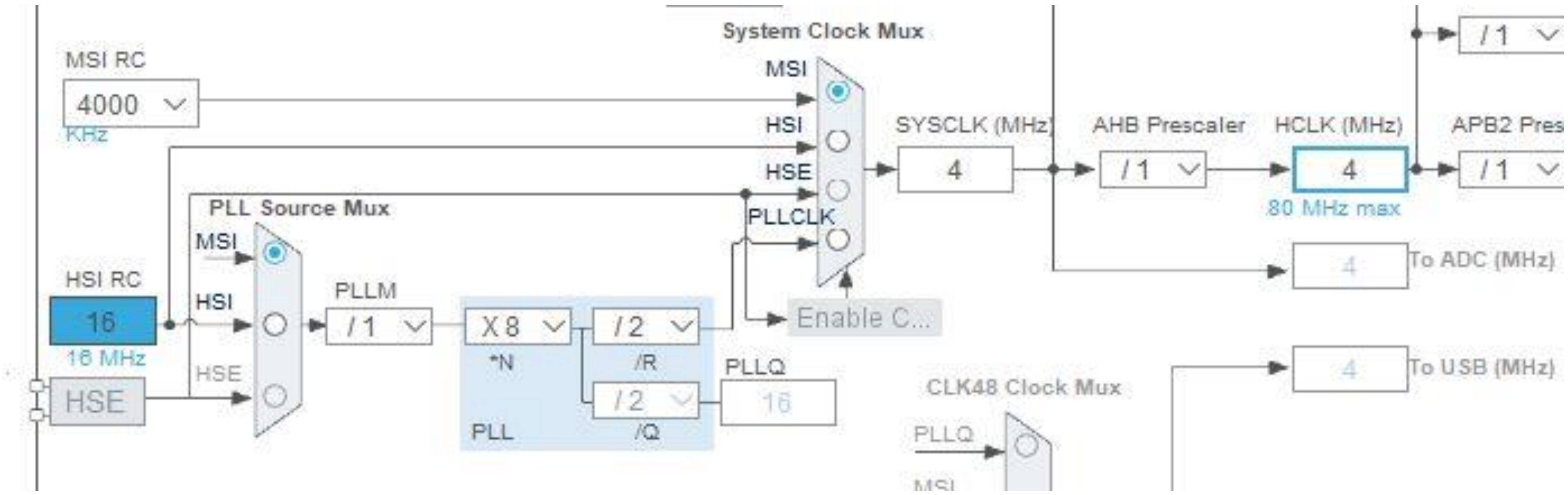
STM32 Cube

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

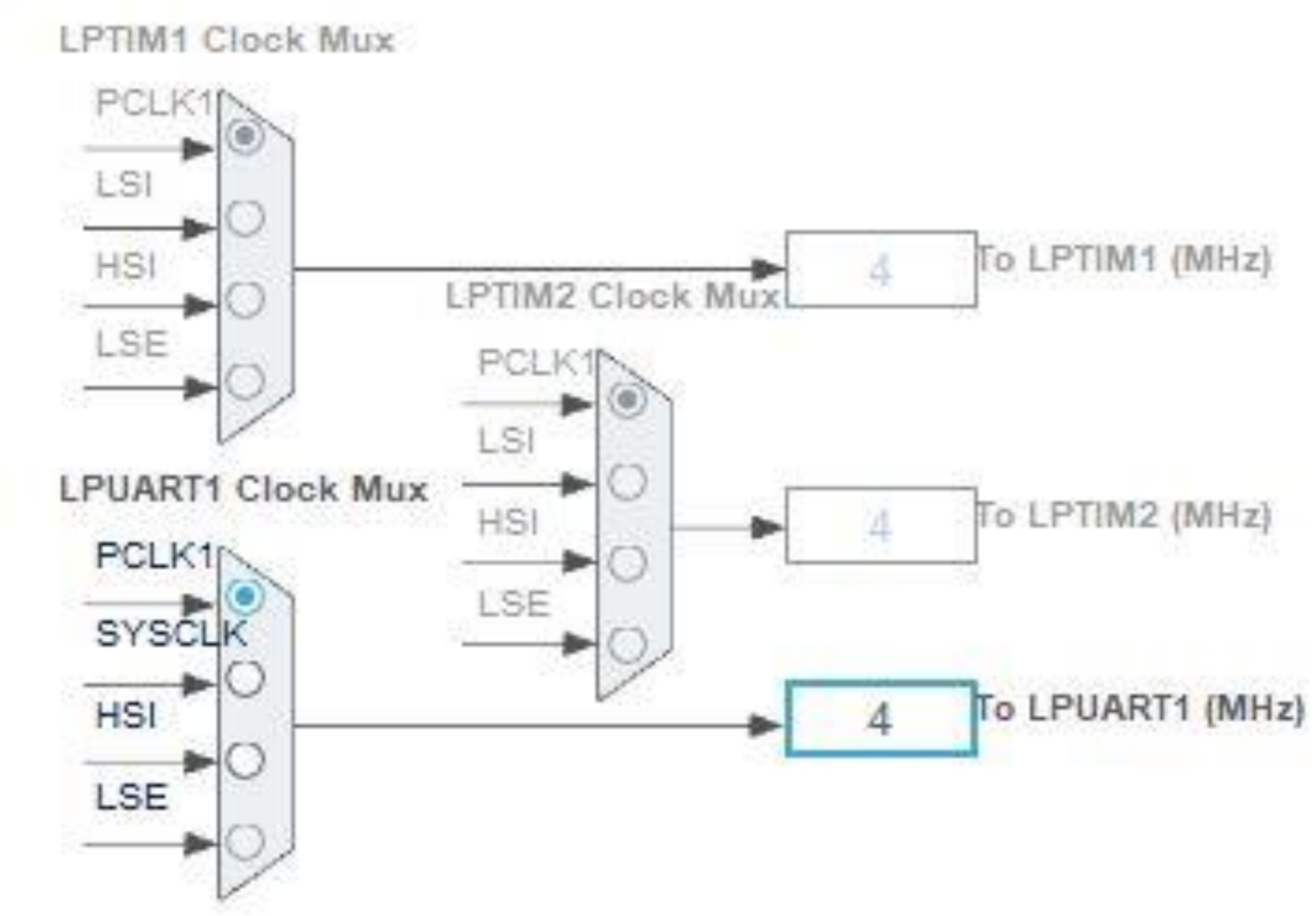
XBee Radio Modules

XBee Hardware Design – Transmitter/Receiver



XBee Radio Modules

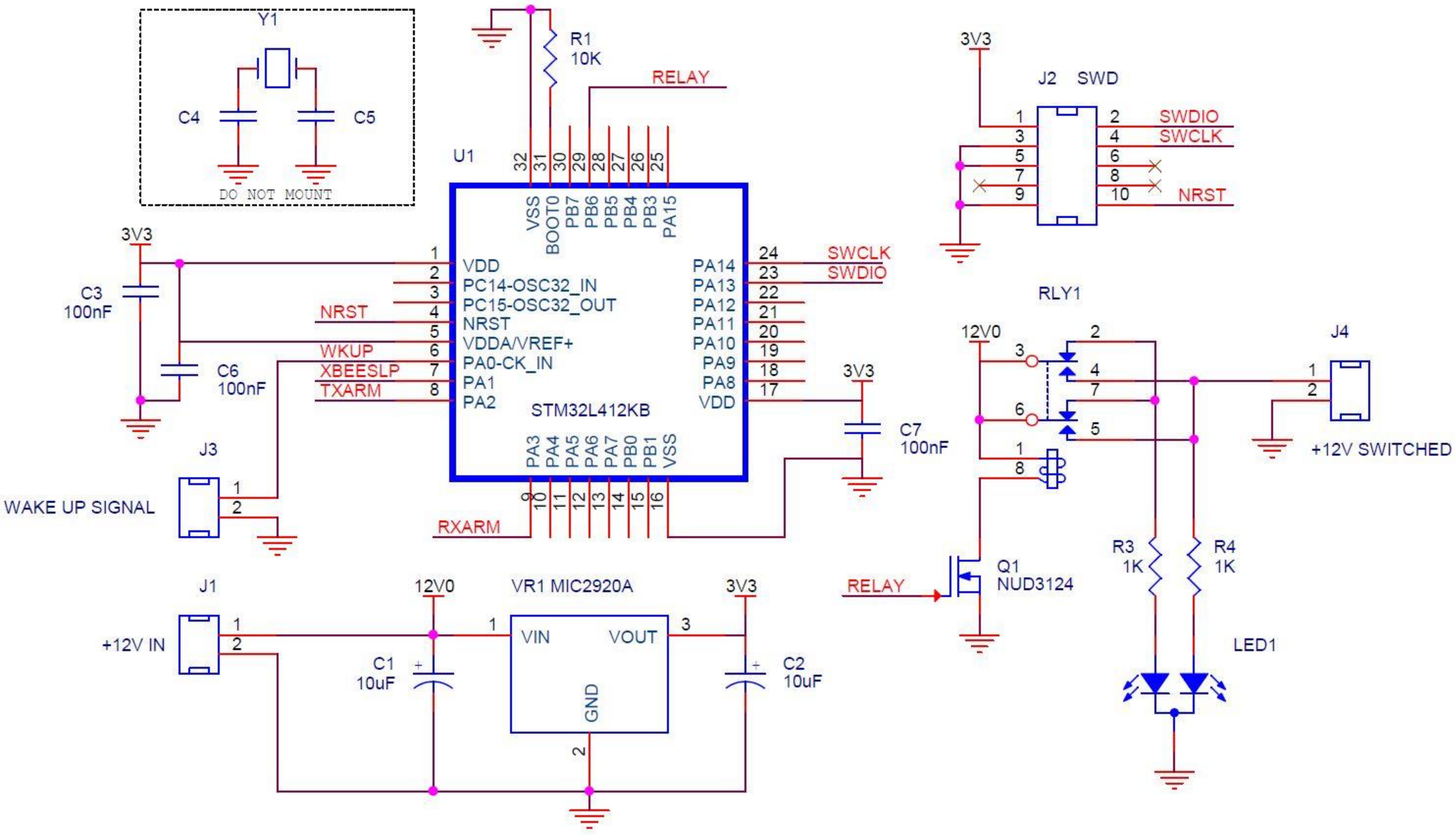
XBee Hardware Design – Transmitter/Receiver



Presented by:

XBee Radio Modules

XBee Hardware Design – Transmitter/Receiver

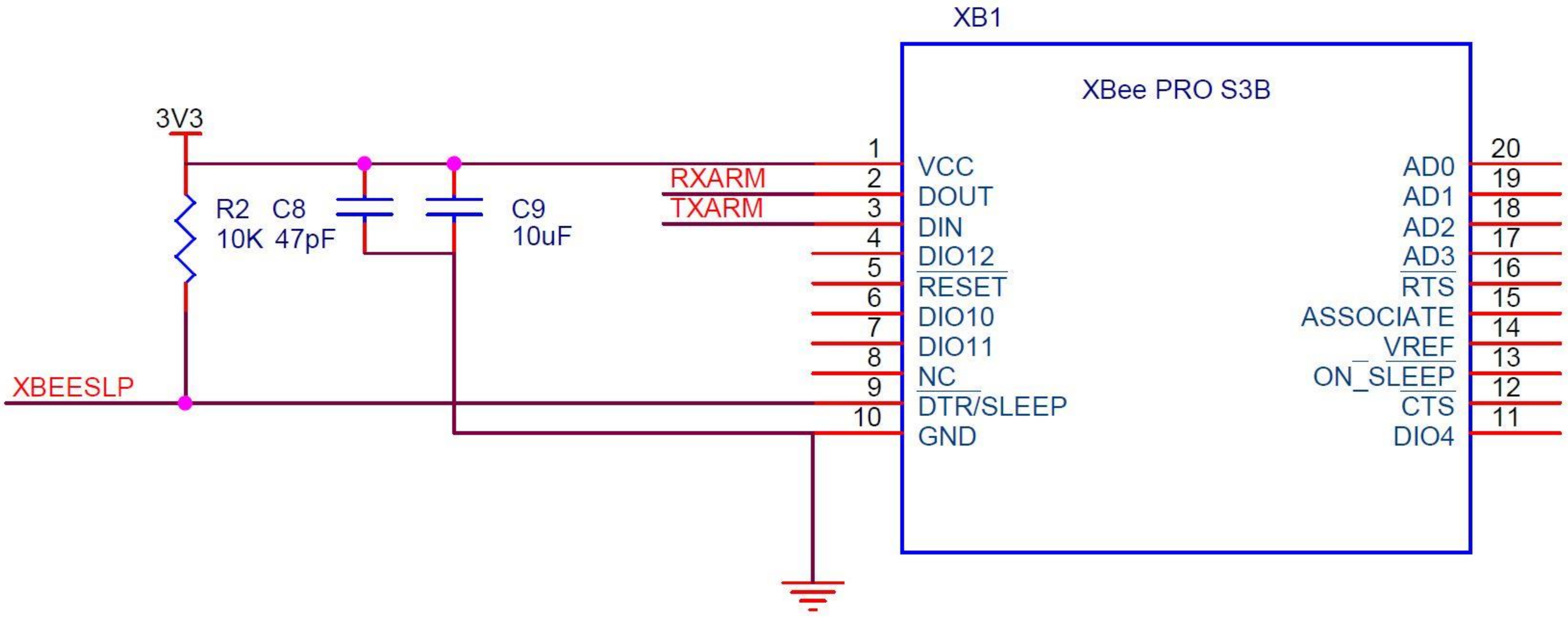


8

Presented by:

XBee Radio Modules

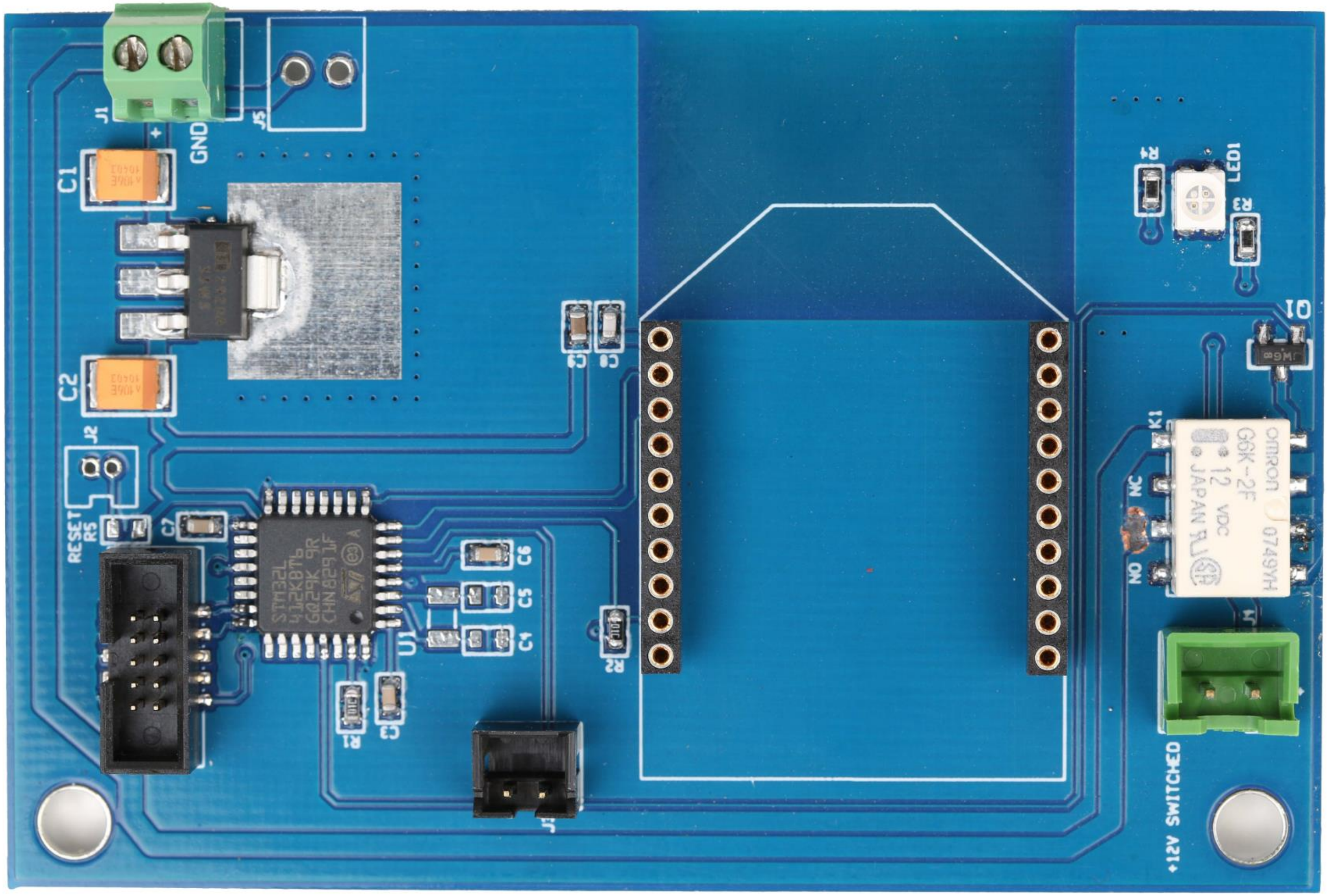
XBee Hardware Design – Transmitter/Receiver

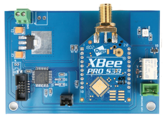


Presented by:

XBee Radio Modules

XBee Hardware Design – Transmitter/Receiver

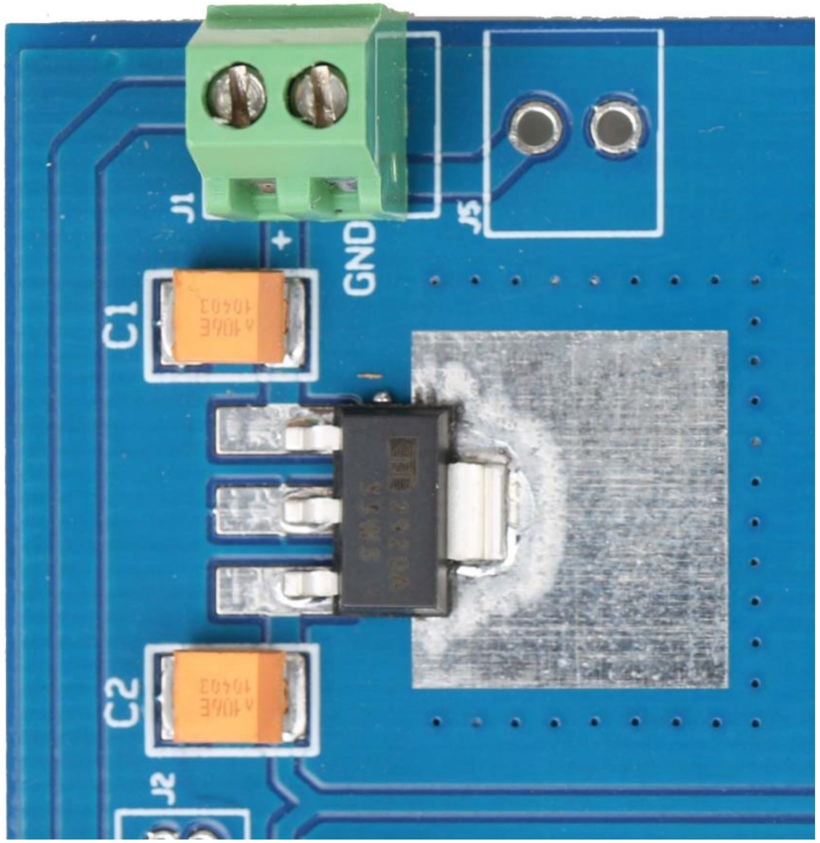




Presented by:

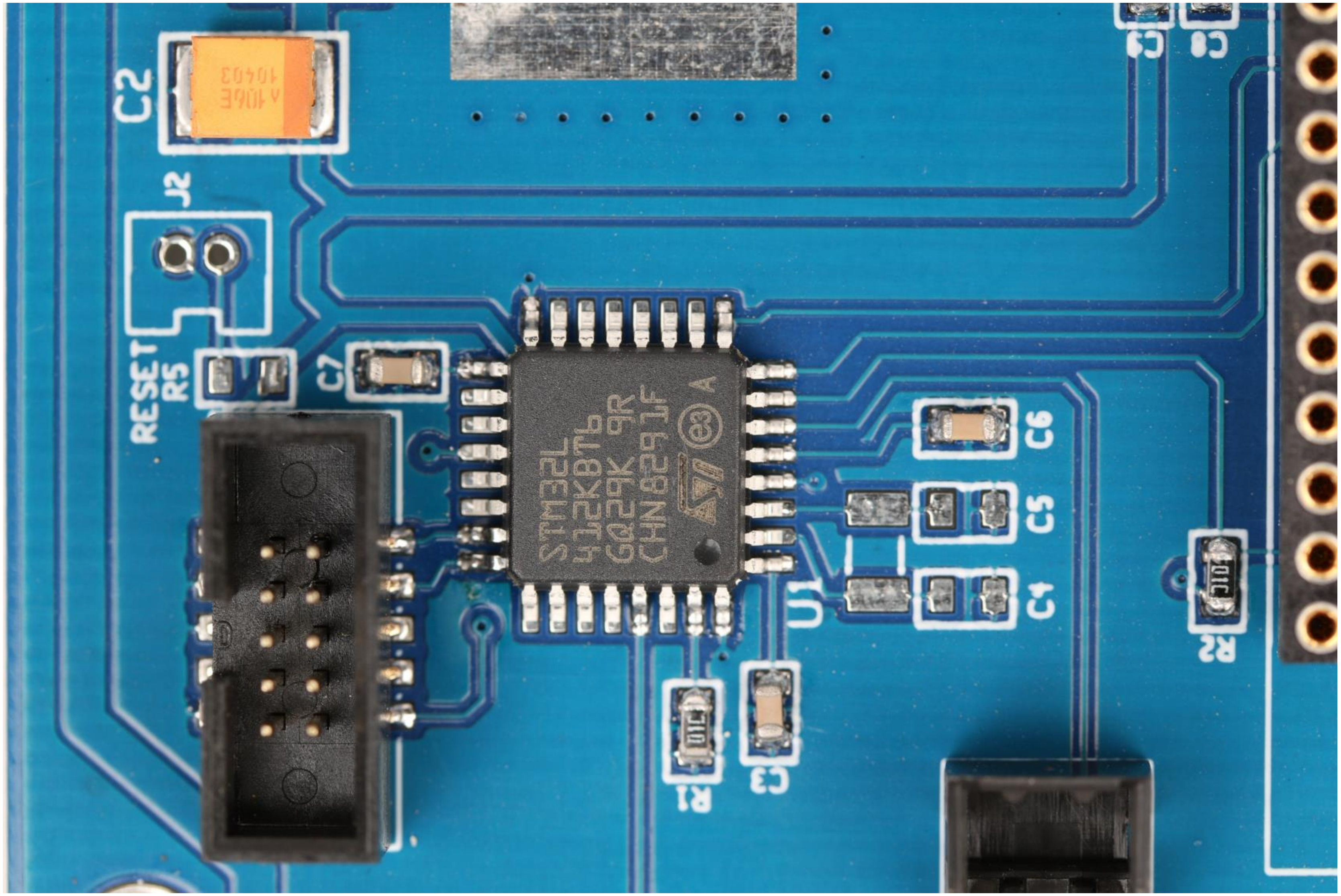
XBee Radio Modules

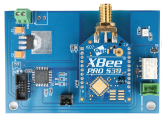
XBee Hardware Design – Transmitter/Receiver



XBee Radio Modules

XBee Hardware Design – Transmitter/Receiver

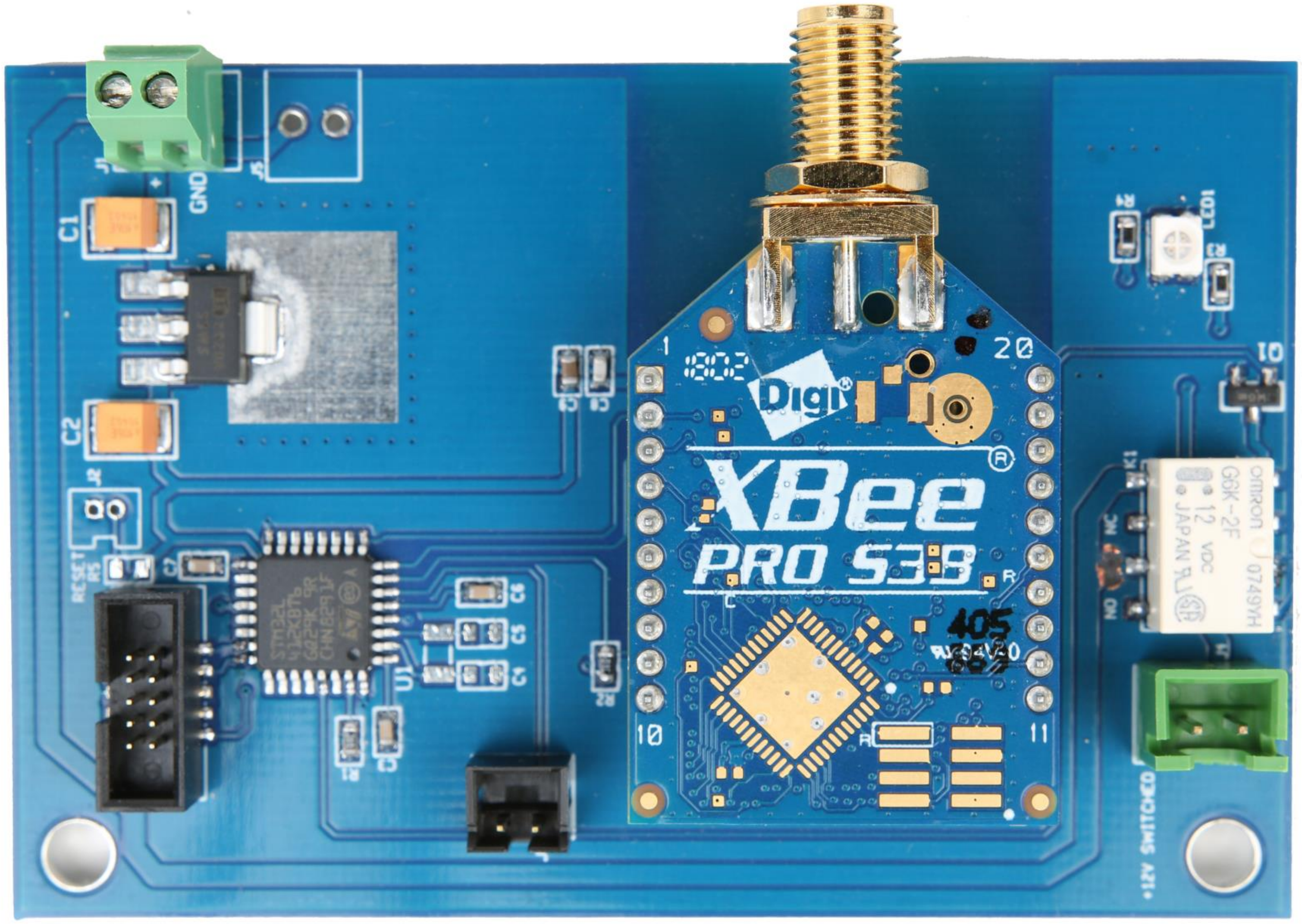




Presented by:

XBee Radio Modules

XBee Hardware Design – Transmitter/Receiver



Presented by:

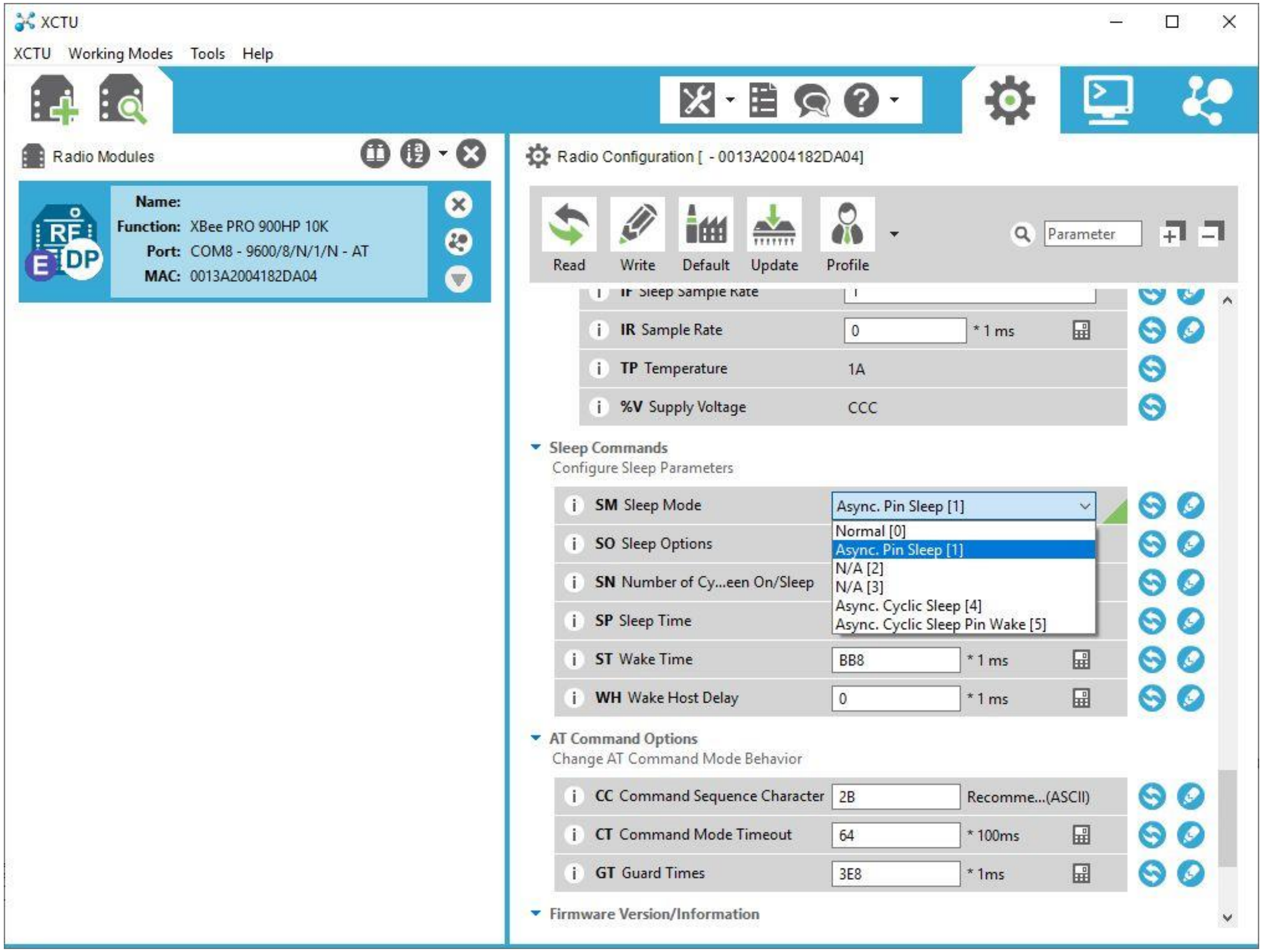
DesignNews
Information Classification: General

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

XBee Radio Modules

XBee Hardware Design – Transmitter


XCTU
XCTU Working Modes Tools Help
Radio Modules
Radio Configuration [- 0013A2004182DA04]
Read Write Default Update Profile
Parameter

Name:
Function: XBee PRO 900HP 10K
Port: COM8 - 9600/8/N/1/N - AT
MAC: 0013A2004182DA04

IF Sleep Sample Rate	1	
IR Sample Rate	0	* 1 ms
TP Temperature	1A	
%V Supply Voltage	CCC	

Sleep Commands
Configure Sleep Parameters

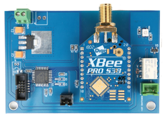
SM Sleep Mode	Async. Pin Sleep [1]	
SO Sleep Options		
SN Number of Cy...een On/Sleep		
SP Sleep Time		
ST Wake Time	BB8	* 1 ms
WH Wake Host Delay	0	* 1 ms

Normal [0]
Async. Pin Sleep [1]
N/A [2]
N/A [3]
Async. Cyclic Sleep [4]
Async. Cyclic Sleep Pin Wake [5]

AT Command Options
Change AT Command Mode Behavior

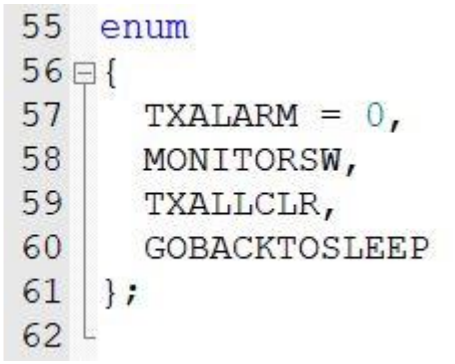
CC Command Sequence Character	2B	Recomme...(ASCII)
CT Command Mode Timeout	64	* 100ms
GT Guard Times	3E8	* 1ms

Firmware Version/Information

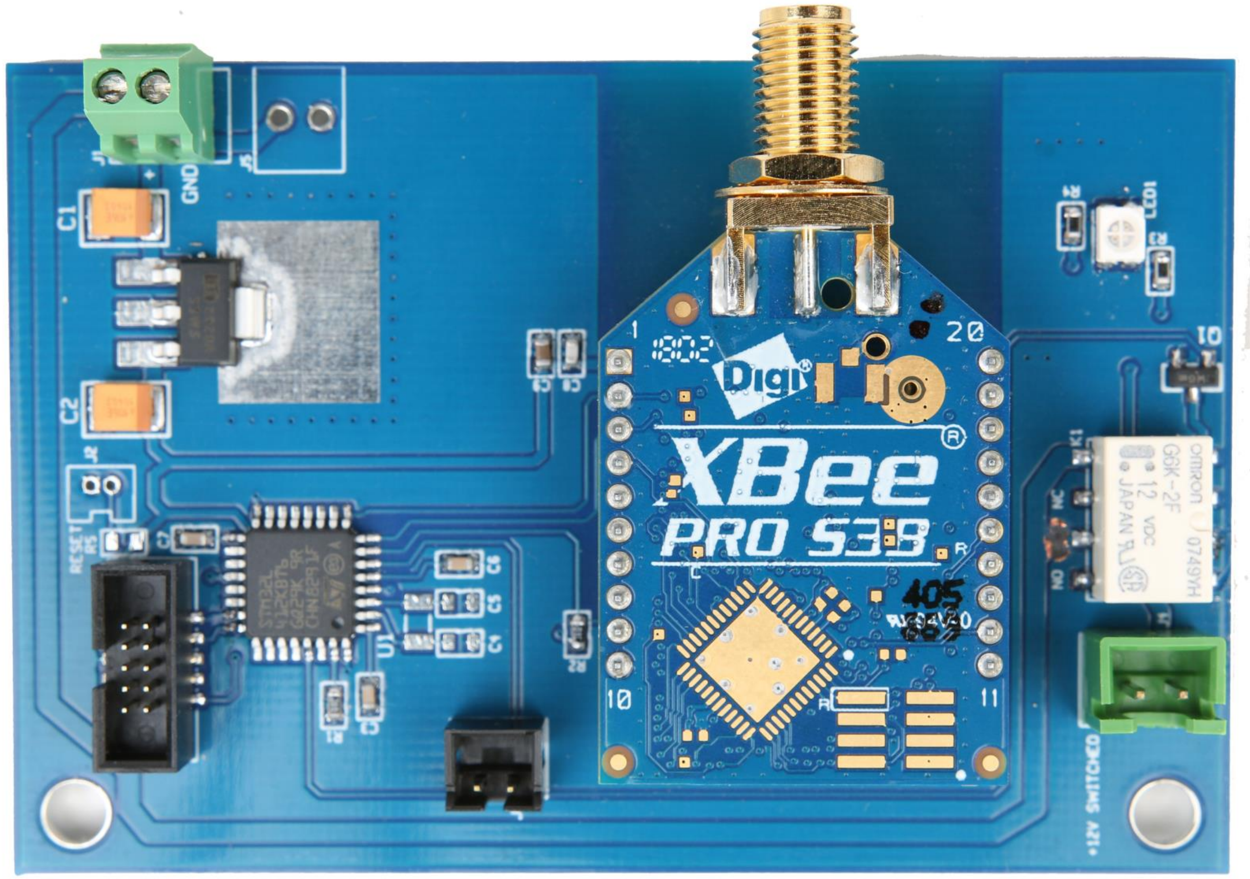


XBee Radio Modules

XBee Application Code

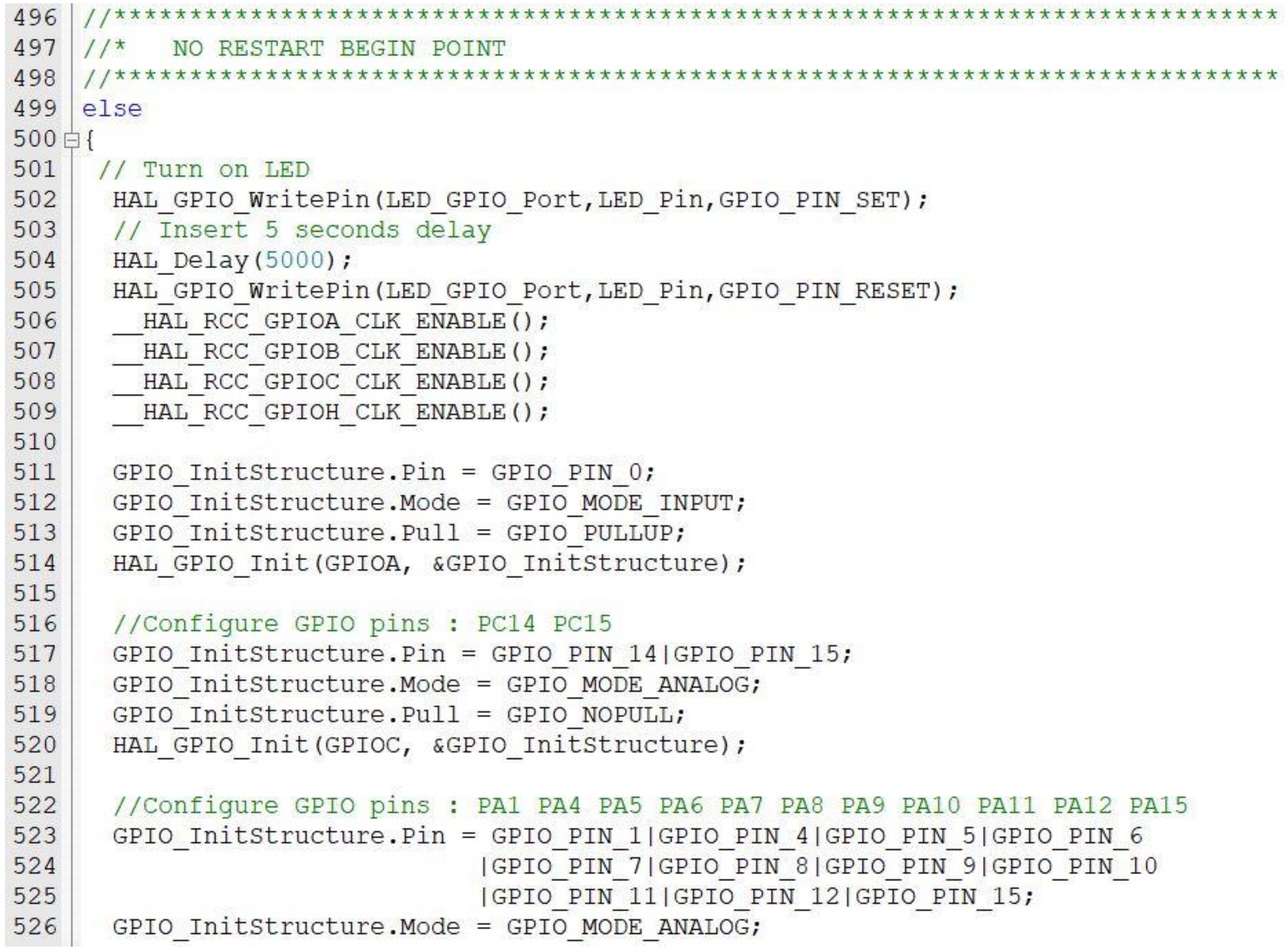


```
enum
{
   TXALARM = 0,
   MONITORSW,
   TXALLCLR,
   GOBACKTOSLEEP
};
```



XBee Radio Modules

XBee Application Code – GPIO Initialization

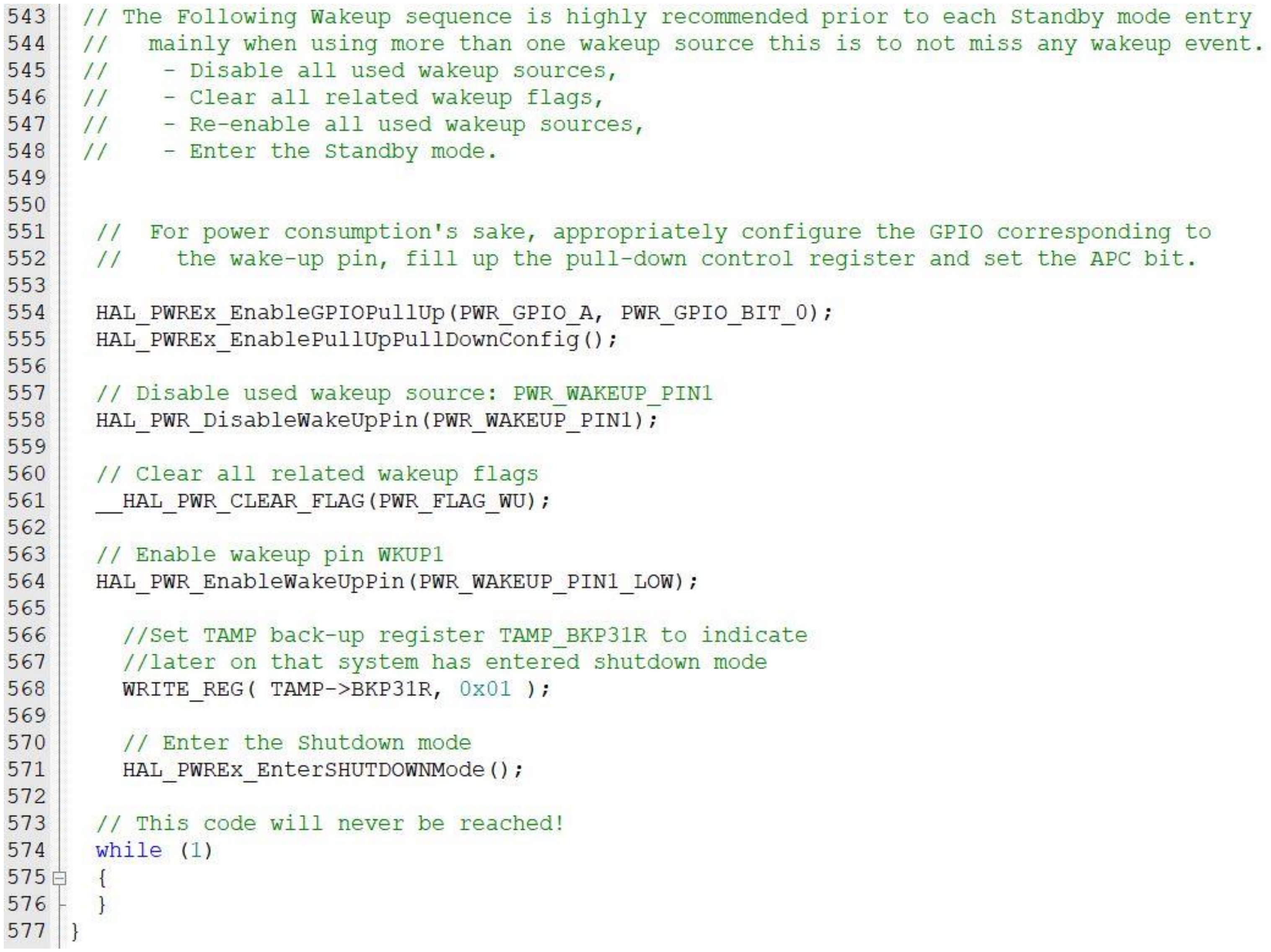


```
//*********************************************************************
//*   NO RESTART BEGIN POINT
//*********************************************************************
else
{
 // Turn on LED
  HAL_GPIO_WritePin(LED_GPIO_Port,LED_Pin,GPIO_PIN_SET);
  // Insert 5 seconds delay
  HAL_Delay(5000);
  HAL_GPIO_WritePin(LED_GPIO_Port,LED_Pin,GPIO_PIN_RESET);
  __HAL_RCC_GPIOA_CLK_ENABLE();
  __HAL_RCC_GPIOB_CLK_ENABLE();
  __HAL_RCC_GPIOC_CLK_ENABLE();
  __HAL_RCC_GPIOH_CLK_ENABLE();

  GPIO_InitStructure.Pin = GPIO_PIN_0;
  GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
  GPIO_InitStructure.Pull = GPIO_PULLUP;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);

  //Configure GPIO pins : PC14 PC15
  GPIO_InitStructure.Pin = GPIO_PIN_14|GPIO_PIN_15;
  GPIO_InitStructure.Mode = GPIO_MODE_ANALOG;
  GPIO_InitStructure.Pull = GPIO_NOPULL;
  HAL_GPIO_Init(GPIOC, &GPIO_InitStructure);

  //Configure GPIO pins : PA1 PA4 PA5 PA6 PA7 PA8 PA9 PA10 PA11 PA12 PA15
  GPIO_InitStructure.Pin = GPIO_PIN_1|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6
                        |GPIO_PIN_7|GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10
                        |GPIO_PIN_11|GPIO_PIN_12|GPIO_PIN_15;
  GPIO_InitStructure.Mode = GPIO_MODE_ANALOG;
```

Presented by:

XBee Radio Modules

XBee Application Code – Power Down

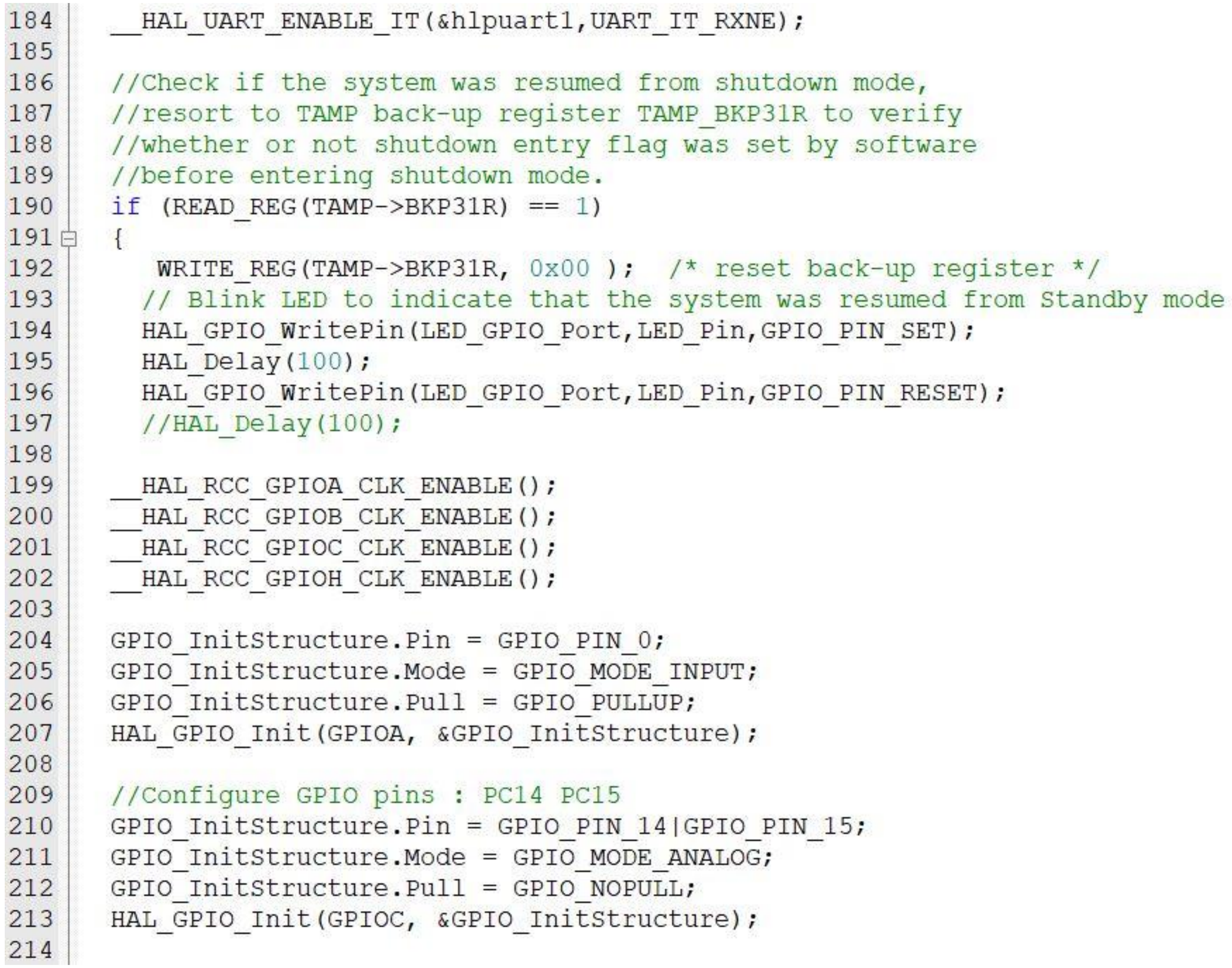


```
 // The Following Wakeup sequence is highly recommended prior to each Standby mode entry
 //   mainly when using more than one wakeup source this is to not miss any wakeup event.
 //    - Disable all used wakeup sources,
 //    - Clear all related wakeup flags,
 //    - Re-enable all used wakeup sources,
 //    - Enter the Standby mode.


  //  For power consumption's sake, appropriately configure the GPIO corresponding to
  //    the wake-up pin, fill up the pull-down control register and set the APC bit.

  HAL_PWREx_EnableGPIOPullUp(PWR_GPIO_A, PWR_GPIO_BIT_0);
  HAL_PWREx_EnablePullUpPullDownConfig();

  // Disable used wakeup source: PWR_WAKEUP_PIN1
  HAL_PWR_DisableWakeUpPin(PWR_WAKEUP_PIN1);

  // Clear all related wakeup flags
  __HAL_PWR_CLEAR_FLAG(PWR_FLAG_WU);

  // Enable wakeup pin WKUP1
  HAL_PWR_EnableWakeUpPin(PWR_WAKEUP_PIN1_LOW);

    //Set TAMP back-up register TAMP_BKP31R to indicate
    //later on that system has entered shutdown mode
    WRITE_REG( TAMP->BKP31R, 0x01 );

    // Enter the Shutdown mode
    HAL_PWREx_EnterSHUTDOWNMode();

  // This code will never be reached!
  while (1)
  {
  }
}
```

Presented by:

XBee Radio Modules

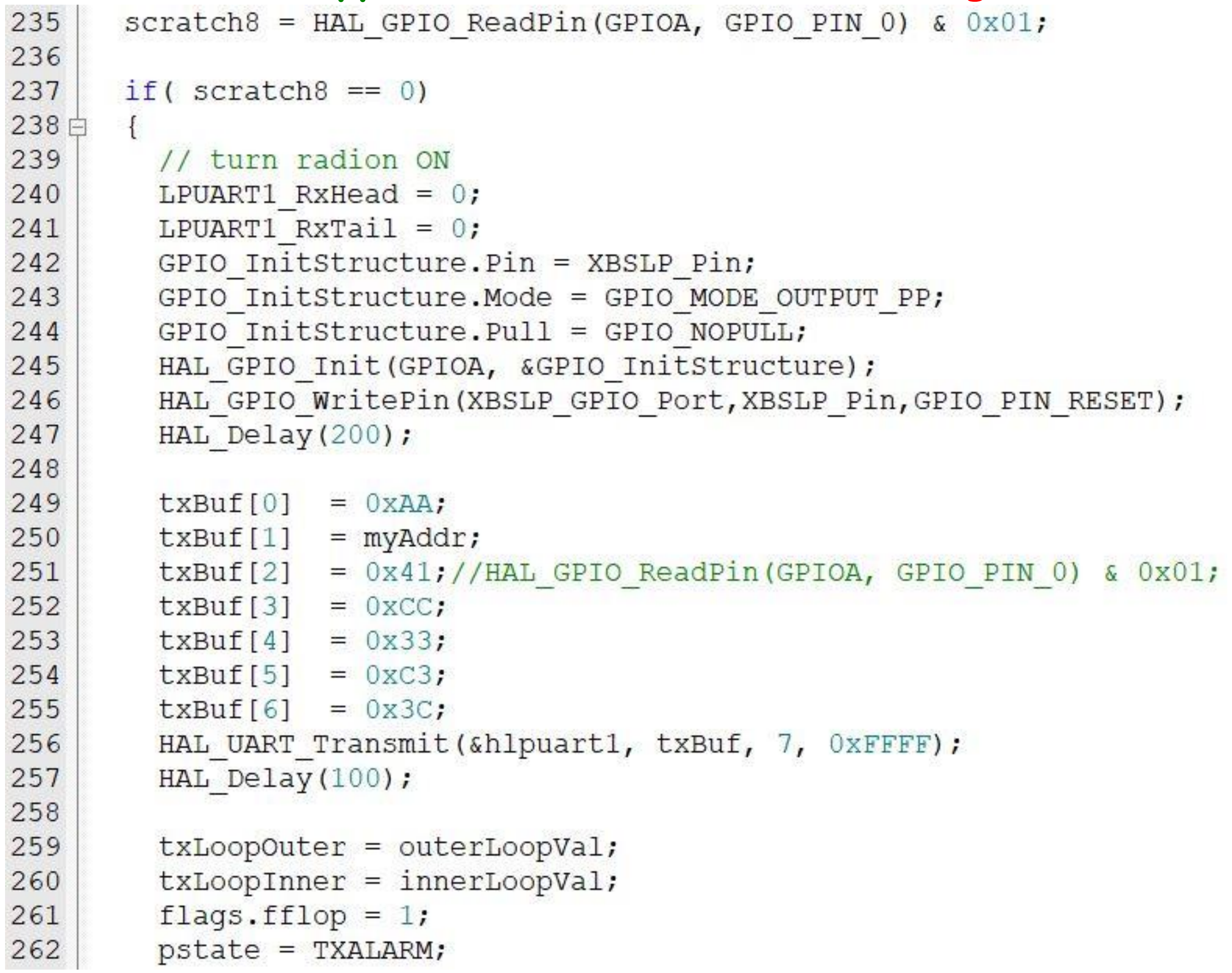
XBee Application Code – Wake Up



```
__HAL_UART_ENABLE_IT(&hlpuart1,UART_IT_RXNE);

//Check if the system was resumed from shutdown mode,
//resort to TAMP back-up register TAMP_BKP31R to verify
//whether or not shutdown entry flag was set by software
//before entering shutdown mode.
if (READ_REG(TAMP->BKP31R) == 1)
{
   WRITE_REG(TAMP->BKP31R, 0x00 );  /* reset back-up register */
  // Blink LED to indicate that the system was resumed from Standby mode
  HAL_GPIO_WritePin(LED_GPIO_Port,LED_Pin,GPIO_PIN_SET);
  HAL_Delay(100);
  HAL_GPIO_WritePin(LED_GPIO_Port,LED_Pin,GPIO_PIN_RESET);
  //HAL_Delay(100);

__HAL_RCC_GPIOA_CLK_ENABLE();
__HAL_RCC_GPIOB_CLK_ENABLE();
__HAL_RCC_GPIOC_CLK_ENABLE();
__HAL_RCC_GPIOH_CLK_ENABLE();

GPIO_InitStructure.Pin = GPIO_PIN_0;
GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
GPIO_InitStructure.Pull = GPIO_PULLUP;
HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);

//Configure GPIO pins : PC14 PC15
GPIO_InitStructure.Pin = GPIO_PIN_14|GPIO_PIN_15;
GPIO_InitStructure.Mode = GPIO_MODE_ANALOG;
GPIO_InitStructure.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOC, &GPIO_InitStructure);
```

XBee Radio Modules

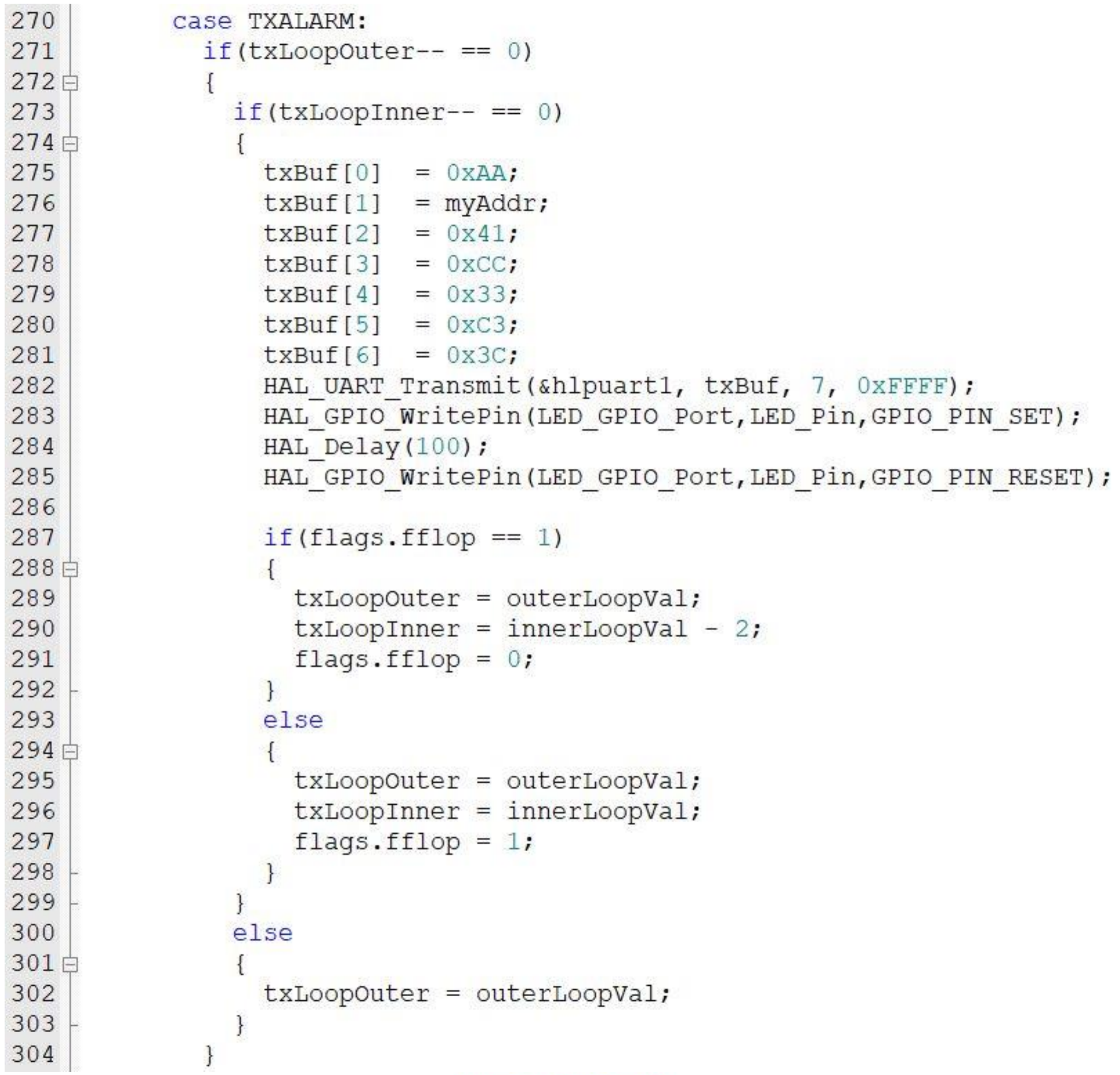
XBee Application Code – Send Alarm Signal



```
scratch8 = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) & 0x01;

if( scratch8 == 0)
{
  // turn radion ON
  LPUART1_RxHead = 0;
  LPUART1_RxTail = 0;
  GPIO_InitStructure.Pin = XBSLP_Pin;
  GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStructure.Pull = GPIO_NOPULL;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);
  HAL_GPIO_WritePin(XBSLP_GPIO_Port,XBSLP_Pin,GPIO_PIN_RESET);
  HAL_Delay(200);

  txBuf[0]  = 0xAA;
  txBuf[1]  = myAddr;
  txBuf[2]  = 0x41;//HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) & 0x01;
  txBuf[3]  = 0xCC;
  txBuf[4]  = 0x33;
  txBuf[5]  = 0xC3;
  txBuf[6]  = 0x3C;
  HAL_UART_Transmit(&hlpuart1, txBuf, 7, 0xFFFF);
  HAL_Delay(100);

  txLoopOuter = outerLoopVal;
  txLoopInner = innerLoopVal;
  flags.fflop = 1;
  pstate = TXALARM;
```

XBee Radio Modules

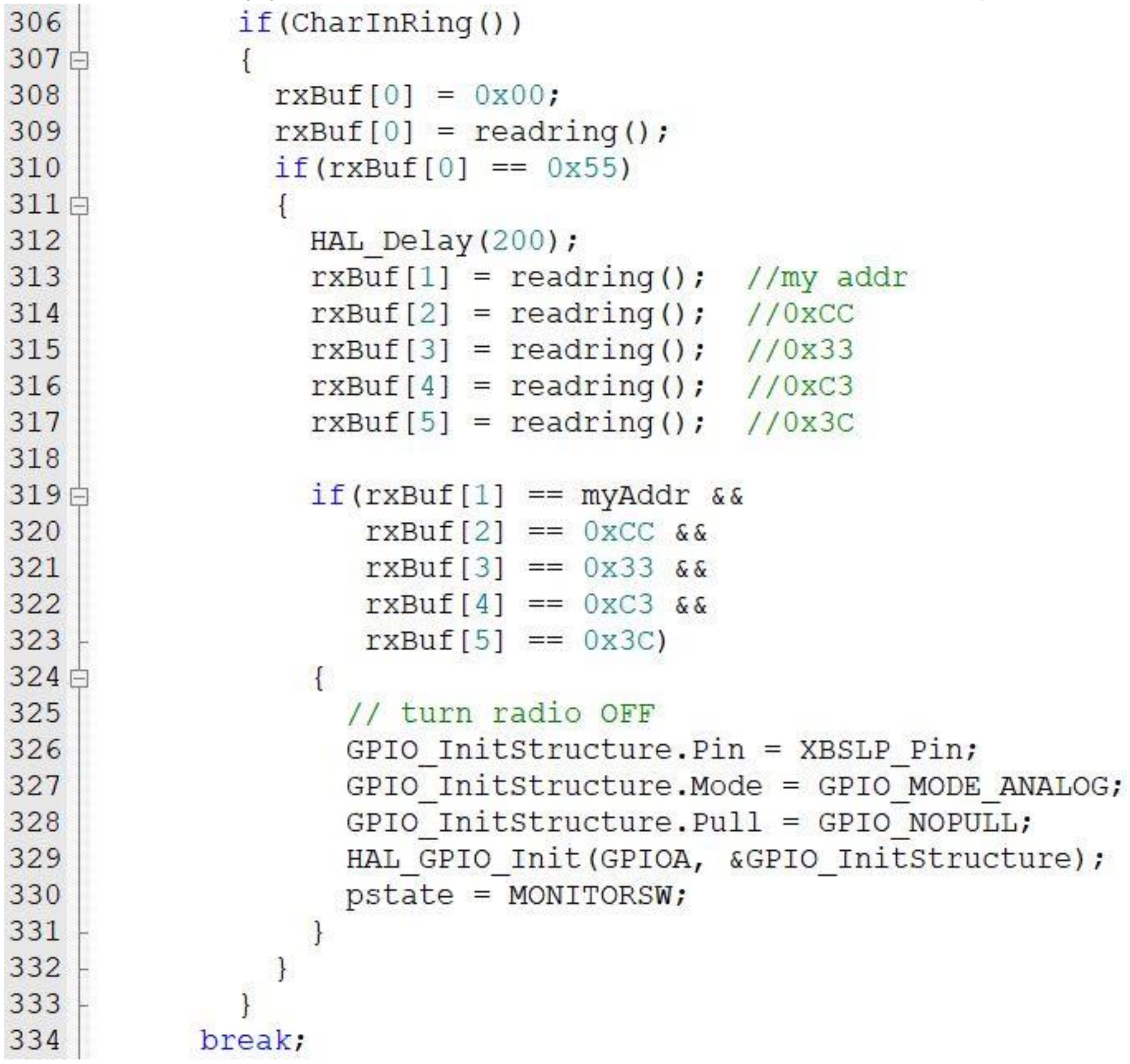
XBee Application Code – Send Alarm Until Acknowledged



```
        case TXALARM:
          if(txLoopOuter-- == 0)
          {
            if(txLoopInner-- == 0)
            {
              txBuf[0]  = 0xAA;
              txBuf[1]  = myAddr;
              txBuf[2]  = 0x41;
              txBuf[3]  = 0xCC;
              txBuf[4]  = 0x33;
              txBuf[5]  = 0xC3;
              txBuf[6]  = 0x3C;
              HAL_UART_Transmit(&hlpuart1, txBuf, 7, 0xFFFF);
              HAL_GPIO_WritePin(LED_GPIO_Port,LED_Pin,GPIO_PIN_SET);
              HAL_Delay(100);
              HAL_GPIO_WritePin(LED_GPIO_Port,LED_Pin,GPIO_PIN_RESET);

              if(flags.fflop == 1)
              {
                txLoopOuter = outerLoopVal;
                txLoopInner = innerLoopVal - 2;
                flags.fflop = 0;
              }
              else
              {
                txLoopOuter = outerLoopVal;
                txLoopInner = innerLoopVal;
                flags.fflop = 1;
              }
            }
            else
            {
              txLoopOuter = outerLoopVal;
            }
          }
```

XBee Radio Modules

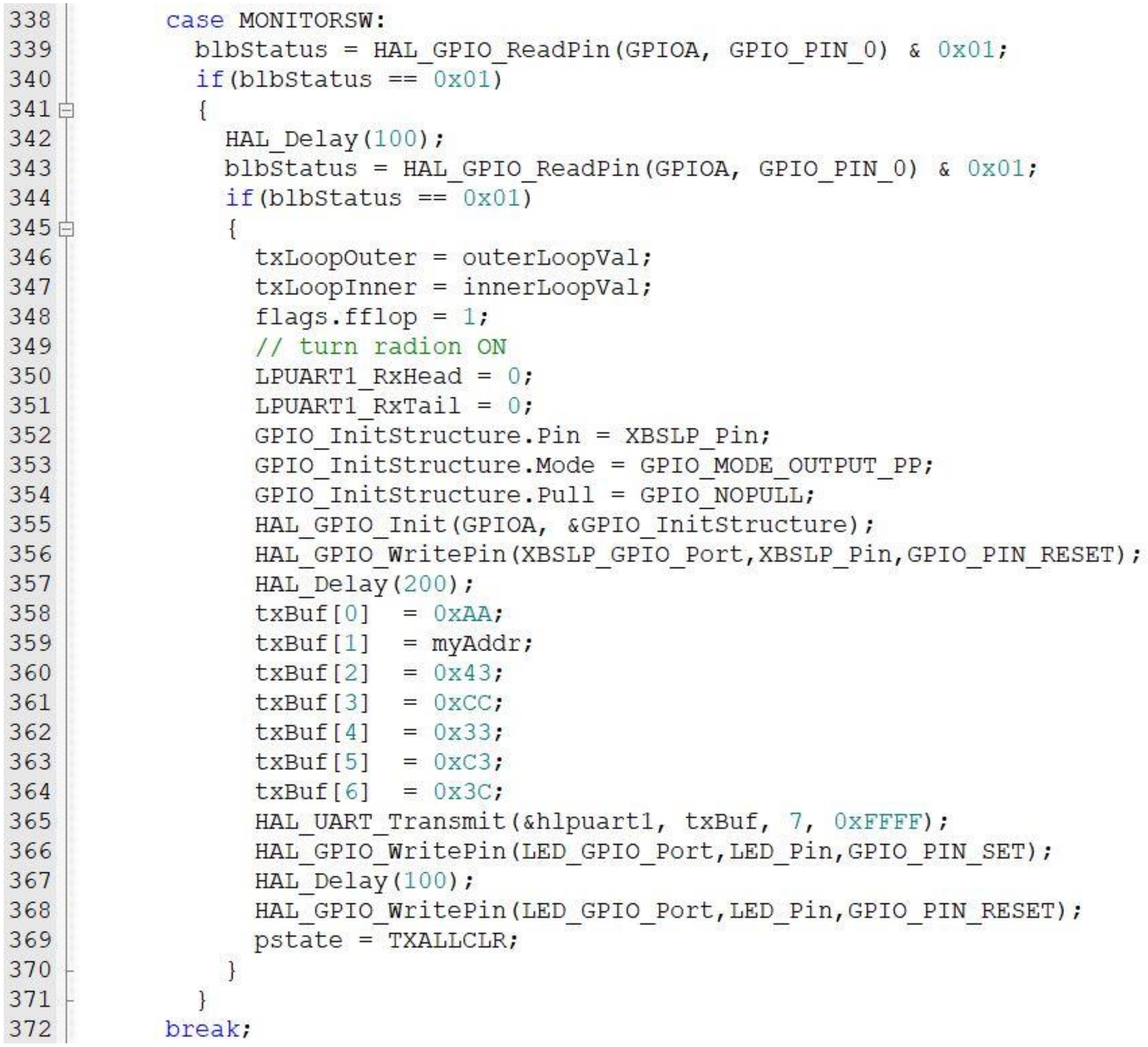
XBee Application Code – Alarm Acknowledged



```
        if(CharInRing())
        {
          rxBuf[0] = 0x00;
          rxBuf[0] = readring();
          if(rxBuf[0] == 0x55)
          {
            HAL_Delay(200);
            rxBuf[1] = readring();  //my addr
            rxBuf[2] = readring();  //0xCC
            rxBuf[3] = readring();  //0x33
            rxBuf[4] = readring();  //0xC3
            rxBuf[5] = readring();  //0x3C

            if(rxBuf[1] == myAddr &&
               rxBuf[2] == 0xCC &&
               rxBuf[3] == 0x33 &&
               rxBuf[4] == 0xC3 &&
               rxBuf[5] == 0x3C)
            {
              // turn radio OFF
              GPIO_InitStructure.Pin = XBSLP_Pin;
              GPIO_InitStructure.Mode = GPIO_MODE_ANALOG;
              GPIO_InitStructure.Pull = GPIO_NOPULL;
              HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);
              pstate = MONITORSW;
            }
          }
        }
      break;
```

XBee Radio Modules

XBee Application Code – Monitor Alarm Sensor

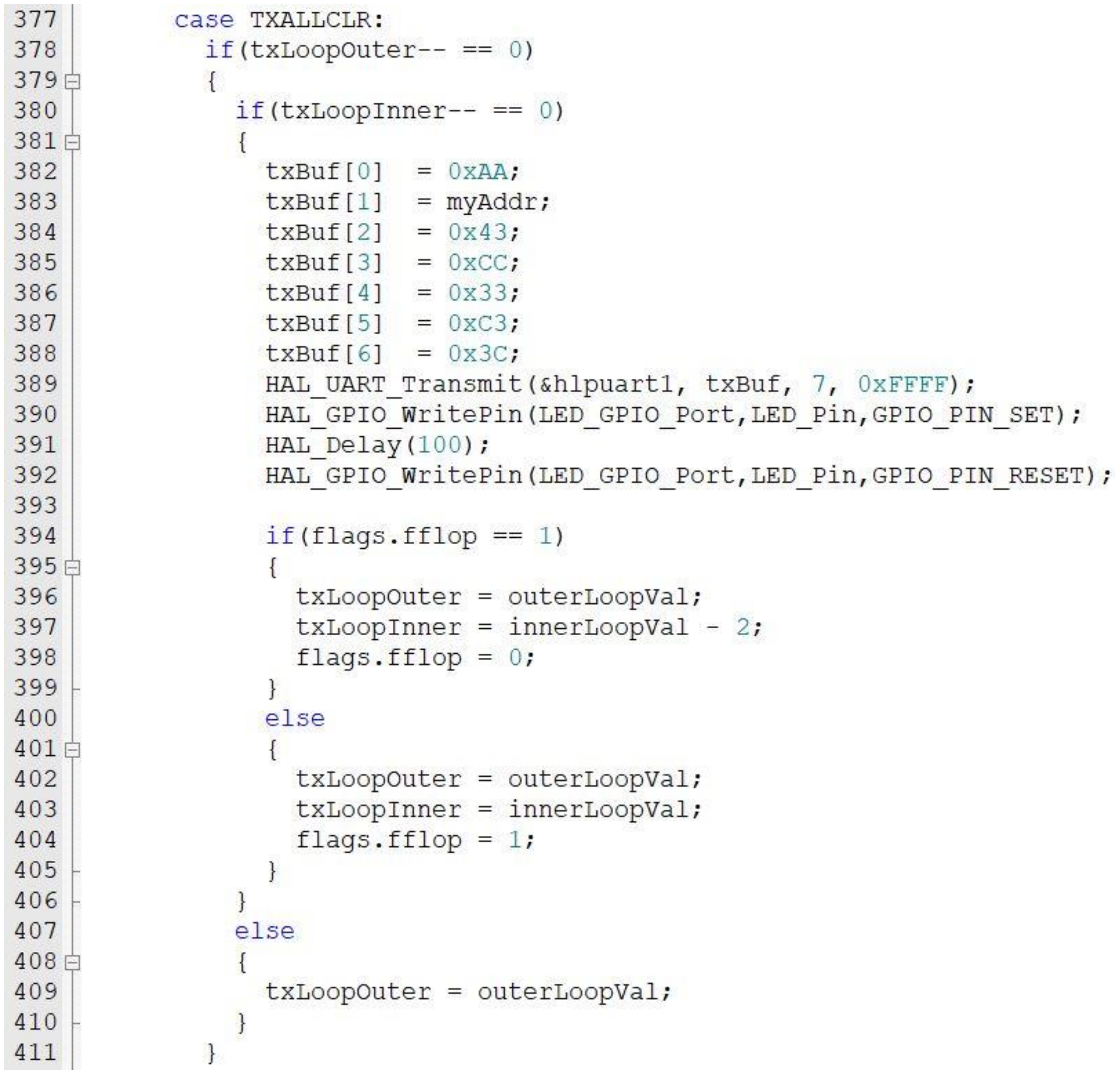


```
case MONITORSW:
  blbStatus = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) & 0x01;
  if(blbStatus == 0x01)
  {
    HAL_Delay(100);
    blbStatus = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0) & 0x01;
    if(blbStatus == 0x01)
    {
      txLoopOuter = outerLoopVal;
      txLoopInner = innerLoopVal;
      flags.fflop = 1;
      // turn radion ON
      LPUART1_RxHead = 0;
      LPUART1_RxTail = 0;
      GPIO_InitStructure.Pin = XBSLP_Pin;
      GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
      GPIO_InitStructure.Pull = GPIO_NOPULL;
      HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);
      HAL_GPIO_WritePin(XBSLP_GPIO_Port,XBSLP_Pin,GPIO_PIN_RESET);
      HAL_Delay(200);
      txBuf[0]  = 0xAA;
      txBuf[1]  = myAddr;
      txBuf[2]  = 0x43;
      txBuf[3]  = 0xCC;
      txBuf[4]  = 0x33;
      txBuf[5]  = 0xC3;
      txBuf[6]  = 0x3C;
      HAL_UART_Transmit(&hlpuart1, txBuf, 7, 0xFFFF);
      HAL_GPIO_WritePin(LED_GPIO_Port,LED_Pin,GPIO_PIN_SET);
      HAL_Delay(100);
      HAL_GPIO_WritePin(LED_GPIO_Port,LED_Pin,GPIO_PIN_RESET);
      pstate = TXALLCLR;
    }
  }
break;
```

Presented by:

XBee Radio Modules

XBee Application Code – Send All Clear



```
case TXALLCLR:
 if(txLoopOuter-- == 0)
 {
  if(txLoopInner-- == 0)
  {
   txBuf[0]  = 0xAA;
   txBuf[1]  = myAddr;
   txBuf[2]  = 0x43;
   txBuf[3]  = 0xCC;
   txBuf[4]  = 0x33;
   txBuf[5]  = 0xC3;
   txBuf[6]  = 0x3C;
   HAL_UART_Transmit(&hlpuart1, txBuf, 7, 0xFFFF);
   HAL_GPIO_WritePin(LED_GPIO_Port,LED_Pin,GPIO_PIN_SET);
   HAL_Delay(100);
   HAL_GPIO_WritePin(LED_GPIO_Port,LED_Pin,GPIO_PIN_RESET);

   if(flags.fflop == 1)
   {
    txLoopOuter = outerLoopVal;
    txLoopInner = innerLoopVal - 2;
    flags.fflop = 0;
   }
   else
   {
    txLoopOuter = outerLoopVal;
    txLoopInner = innerLoopVal;
    flags.fflop = 1;
   }
  }
  else
  {
   txLoopOuter = outerLoopVal;
  }
 }
```

Presented by:

DesignNews
Information Classification: General

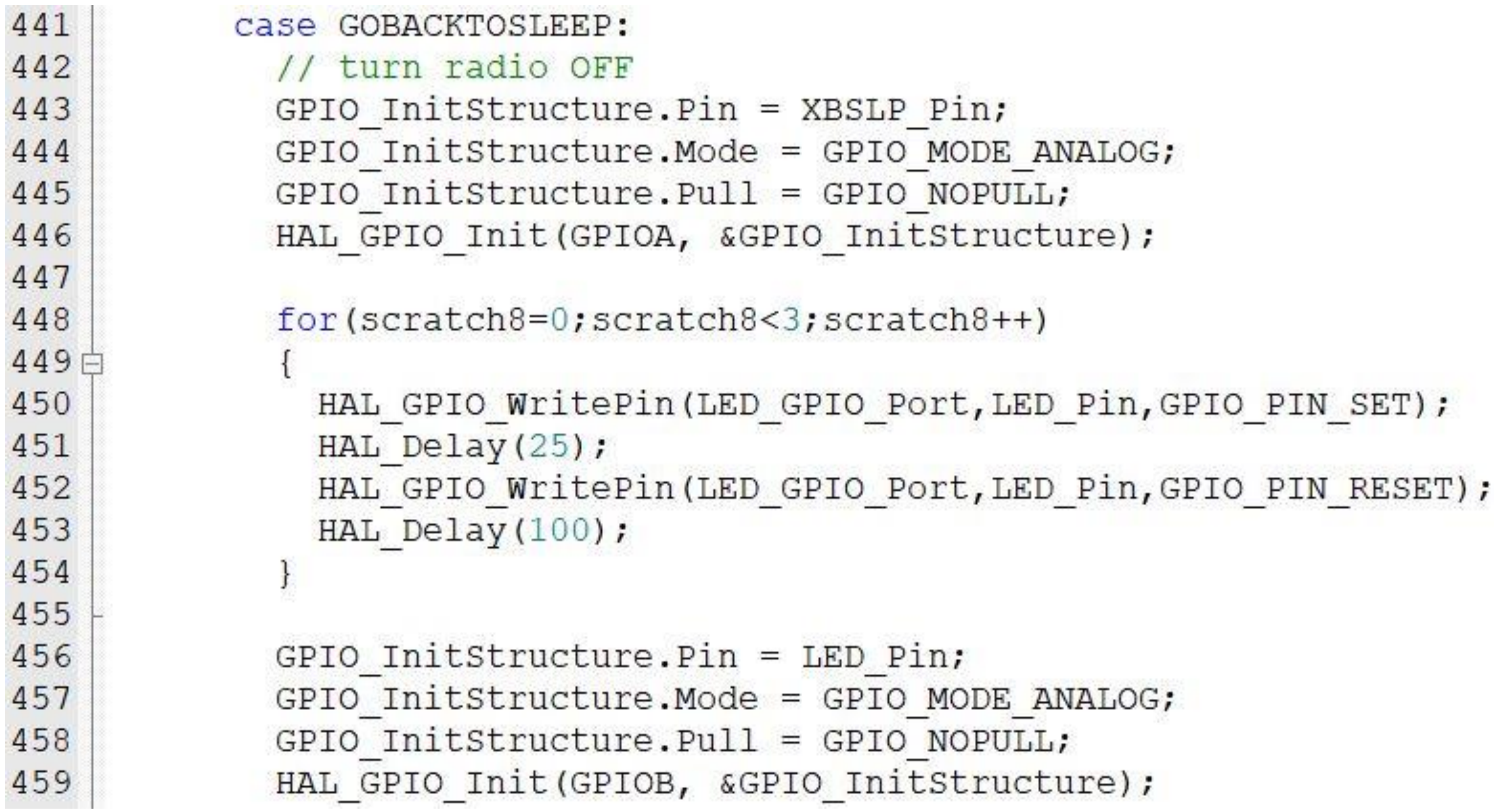
ARMKEIL Microcontroller Tools

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

XBee Radio Modules

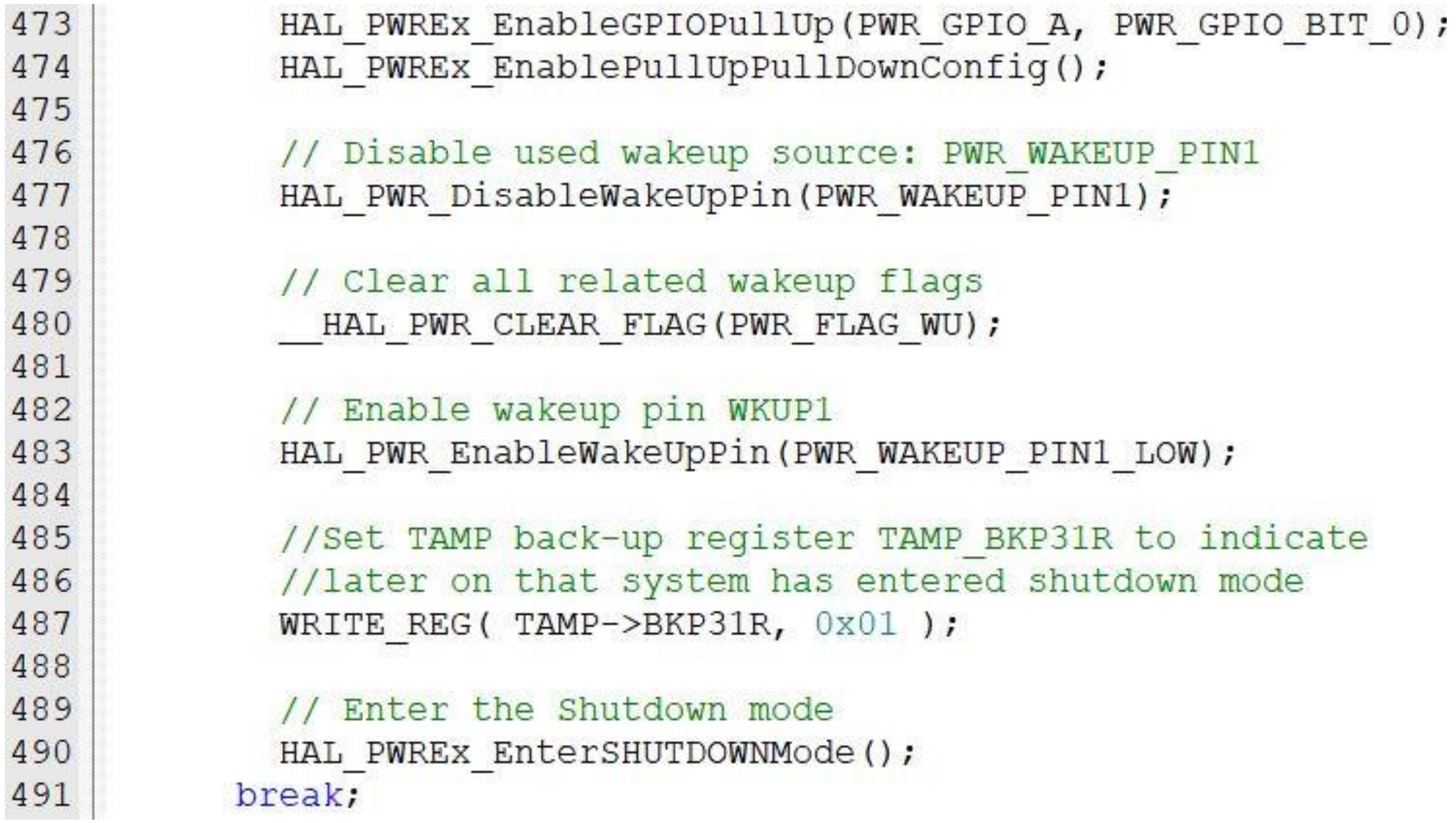
XBee Application Code – All Clear Acknowledged – Go Back To Sleep



```
        case GOBACKTOSLEEP:
          // turn radio OFF
          GPIO_InitStructure.Pin = XBSLP_Pin;
          GPIO_InitStructure.Mode = GPIO_MODE_ANALOG;
          GPIO_InitStructure.Pull = GPIO_NOPULL;
          HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);

          for(scratch8=0;scratch8<3;scratch8++)
          {
            HAL_GPIO_WritePin(LED_GPIO_Port,LED_Pin,GPIO_PIN_SET);
            HAL_Delay(25);
            HAL_GPIO_WritePin(LED_GPIO_Port,LED_Pin,GPIO_PIN_RESET);
            HAL_Delay(100);
          }

          GPIO_InitStructure.Pin = LED_Pin;
          GPIO_InitStructure.Mode = GPIO_MODE_ANALOG;
          GPIO_InitStructure.Pull = GPIO_NOPULL;
          HAL_GPIO_Init(GPIOB, &GPIO_InitStructure);
```

XBee Radio Modules

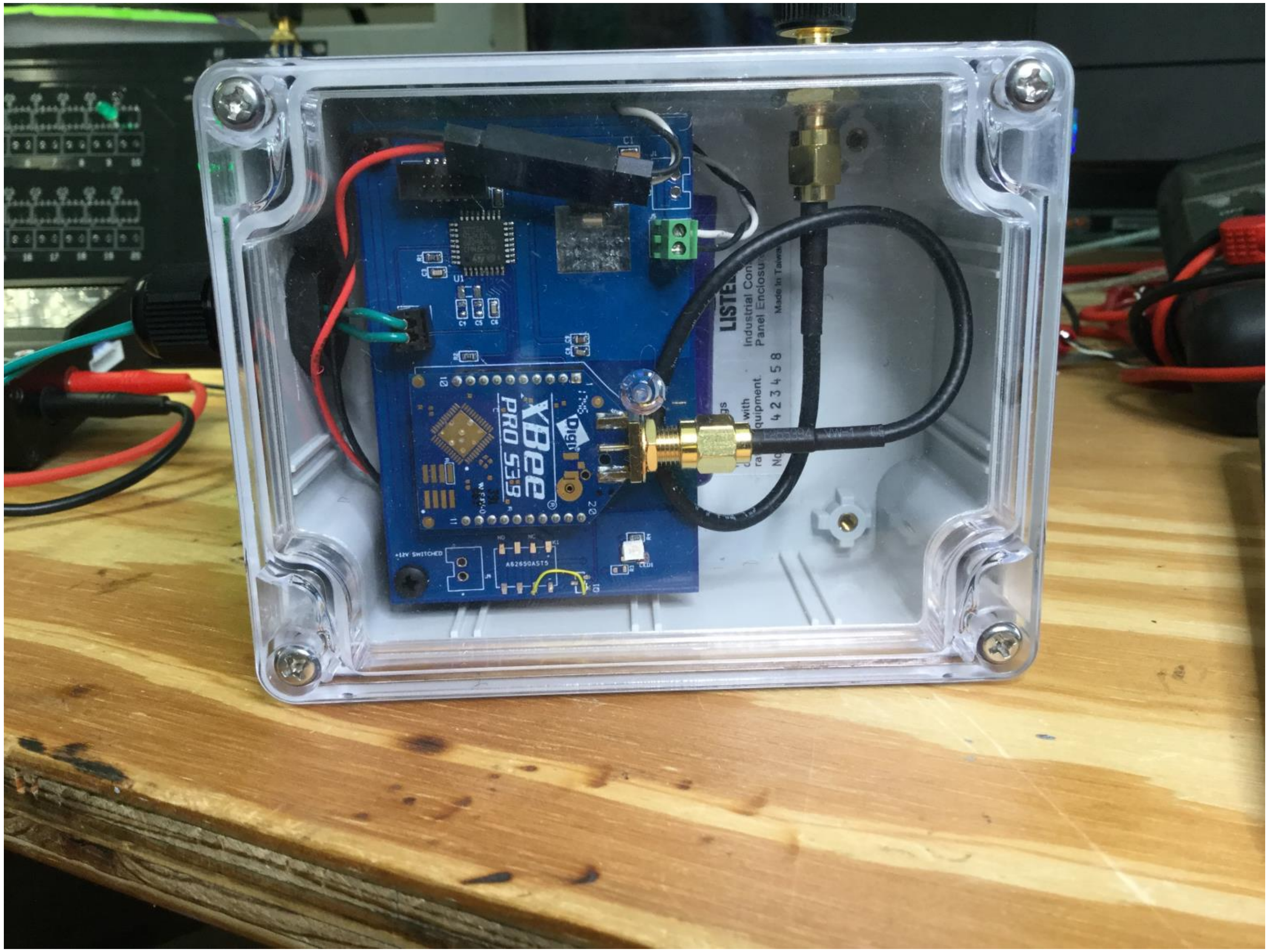
XBee Application Code – All Clear Acknowledged – Go Back To Sleep



```
      HAL_PWREx_EnableGPIOPullUp(PWR_GPIO_A, PWR_GPIO_BIT_0);
      HAL_PWREx_EnablePullUpPullDownConfig();

      // Disable used wakeup source: PWR_WAKEUP_PIN1
      HAL_PWR_DisableWakeUpPin(PWR_WAKEUP_PIN1);

      // Clear all related wakeup flags
      __HAL_PWR_CLEAR_FLAG(PWR_FLAG_WU);

      // Enable wakeup pin WKUP1
      HAL_PWR_EnableWakeUpPin(PWR_WAKEUP_PIN1_LOW);

      //Set TAMP back-up register TAMP_BKP31R to indicate
      //later on that system has entered shutdown mode
      WRITE_REG( TAMP->BKP31R, 0x01 );

      // Enter the Shutdown mode
      HAL_PWREx_EnterSHUTDOWNMode();
    break;
```

XBee Radio Modules

Day 3 Summary



Presented by: