CEC Continuing Education Center

DesignNews

IoT Device Prototyping with STMicroelectronics Nucleo Development Boards

Day 5:
Prototyping with the X-NUCLEO-GFX01M1 Graphic Display

Sponsored by

Digi-Key ELECTRONICS

informa markets

Webinar Logistics

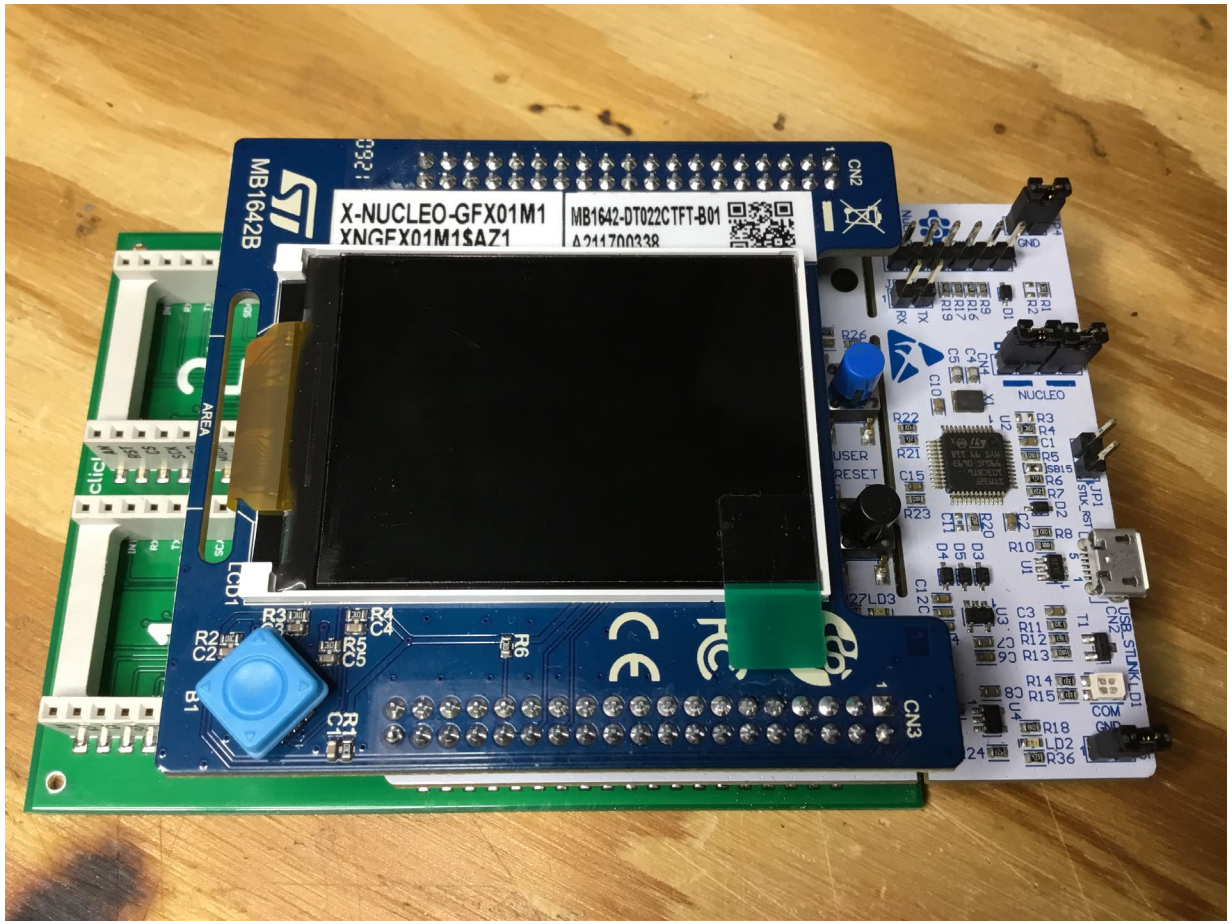
- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.

Fred Eady

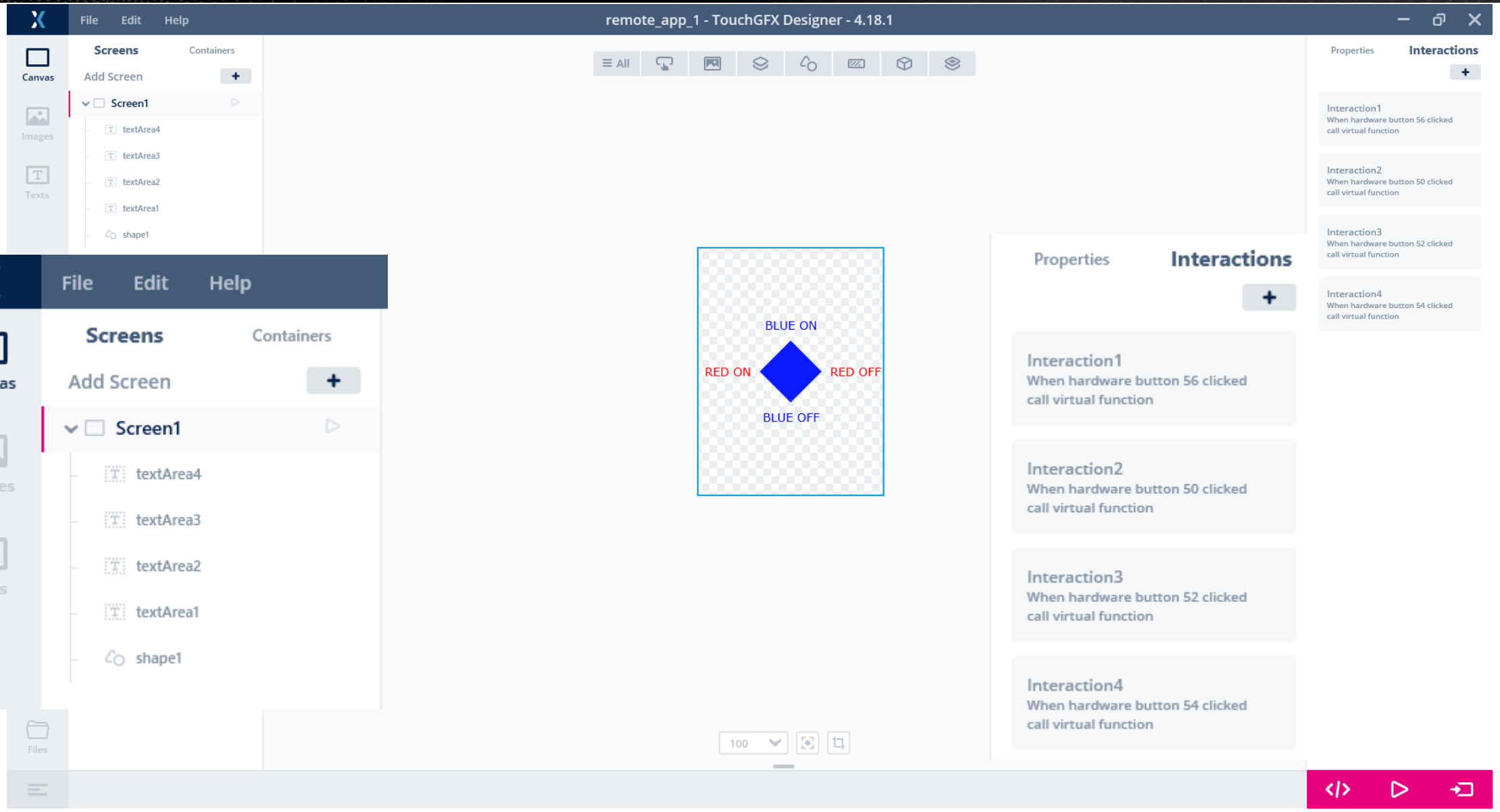
Visit ‘Lecturer Profile’ in your console for more details.

AGENDA

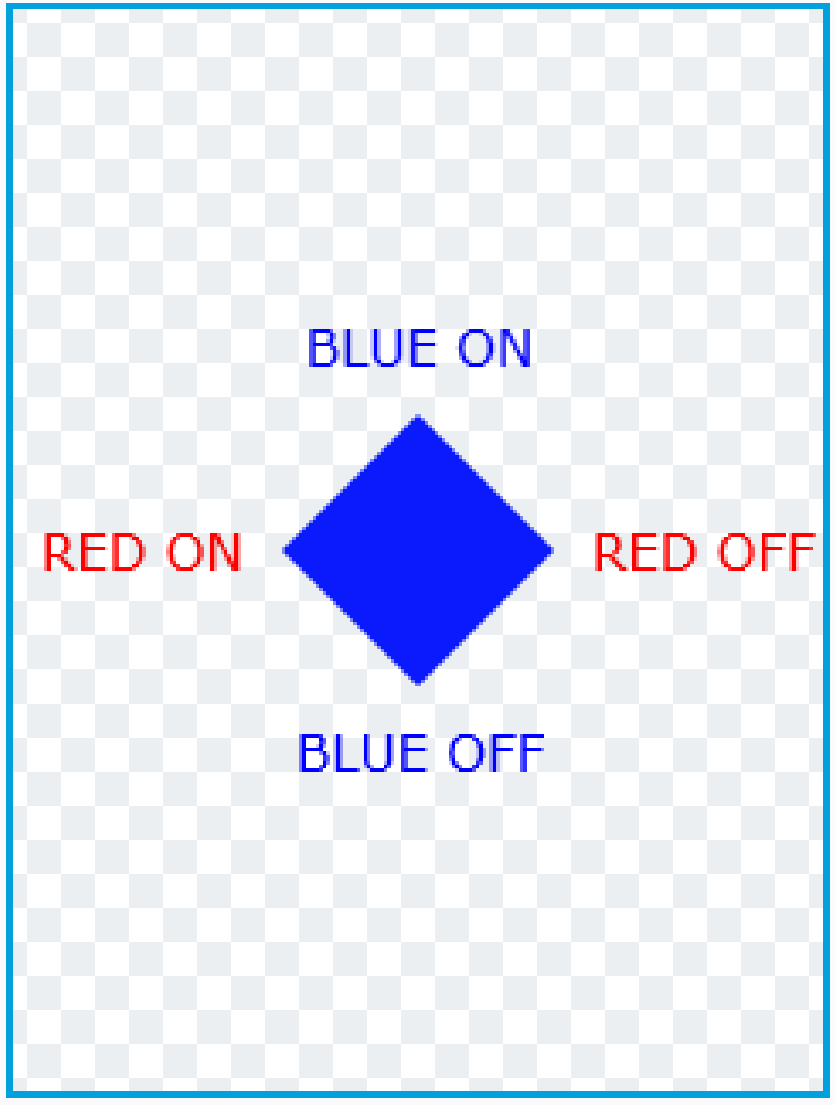
- **Prototype 1: TouchGFX Remote Control Transmitter**
- **Prototype 2: NUCLEO-G070RB Remote Control Receiver**

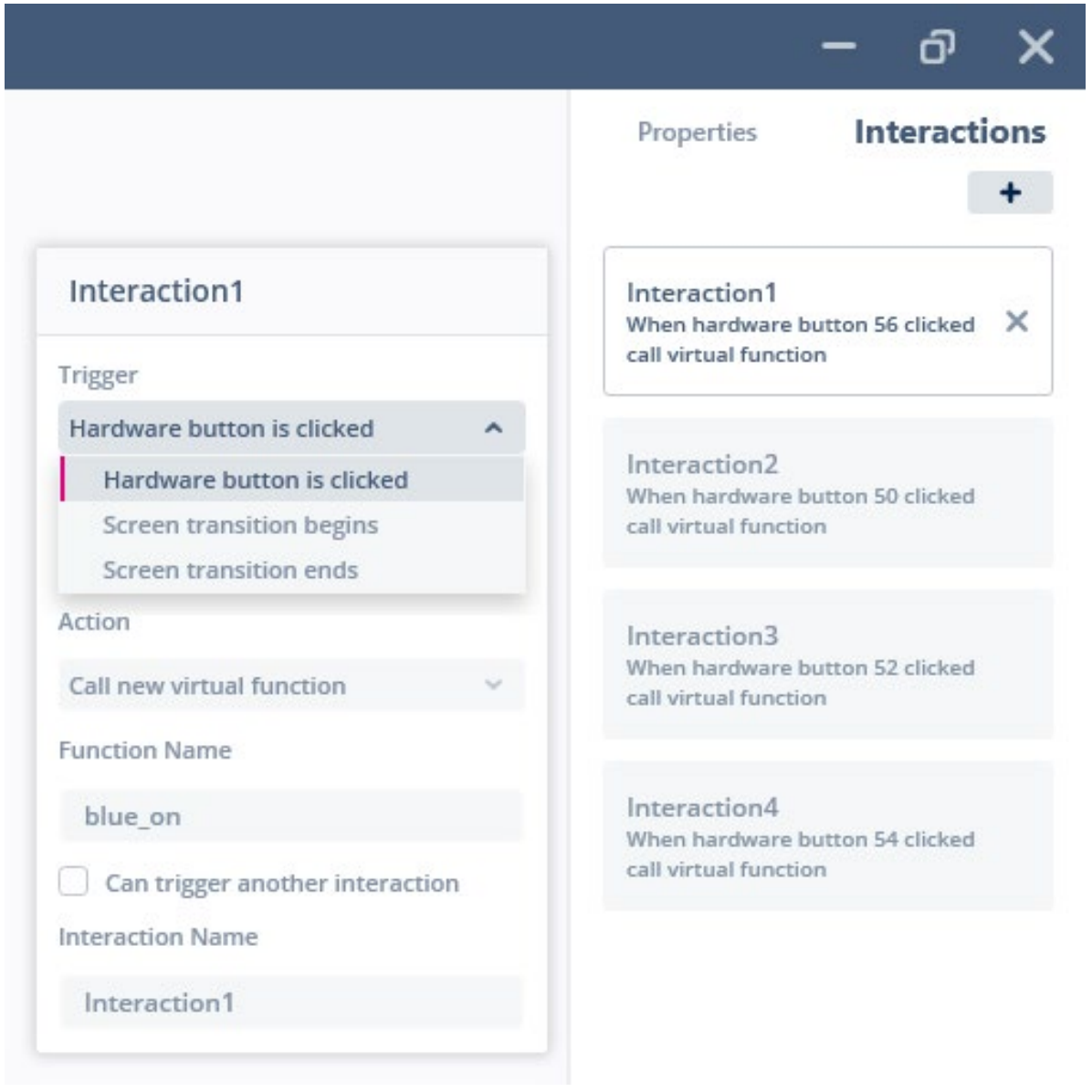


Step 1: Create the TouchGFX Project

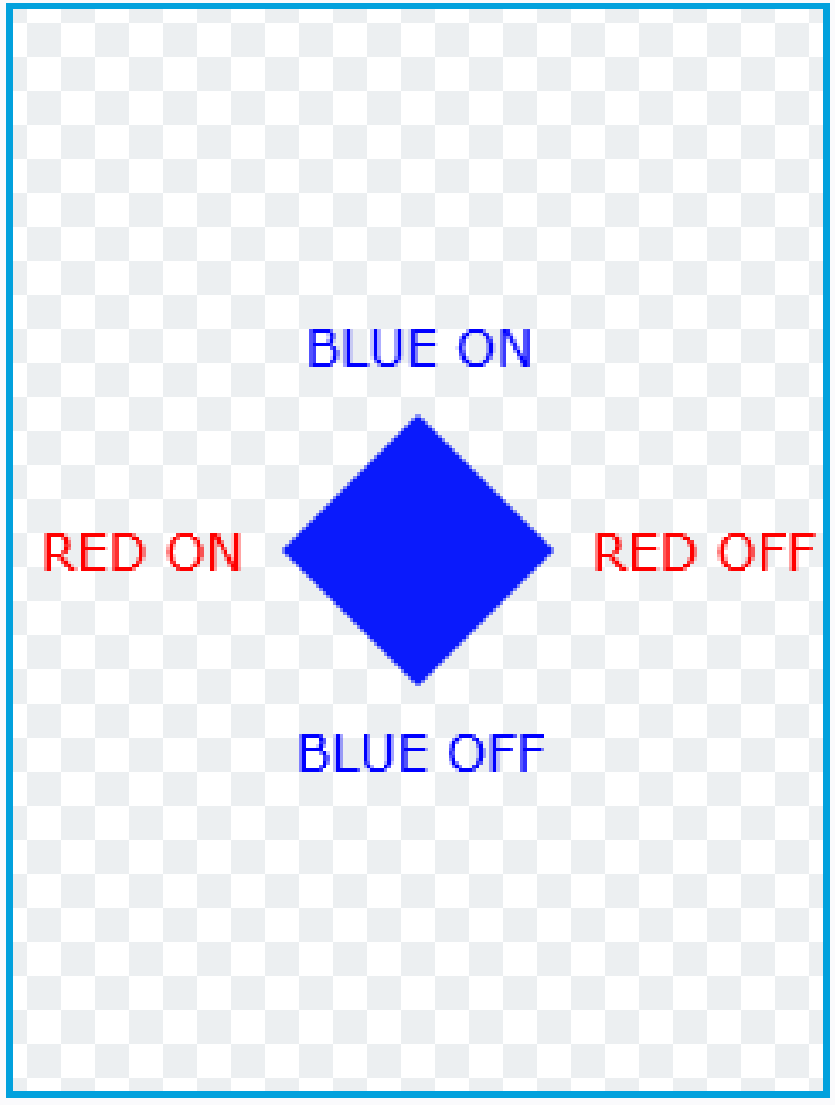


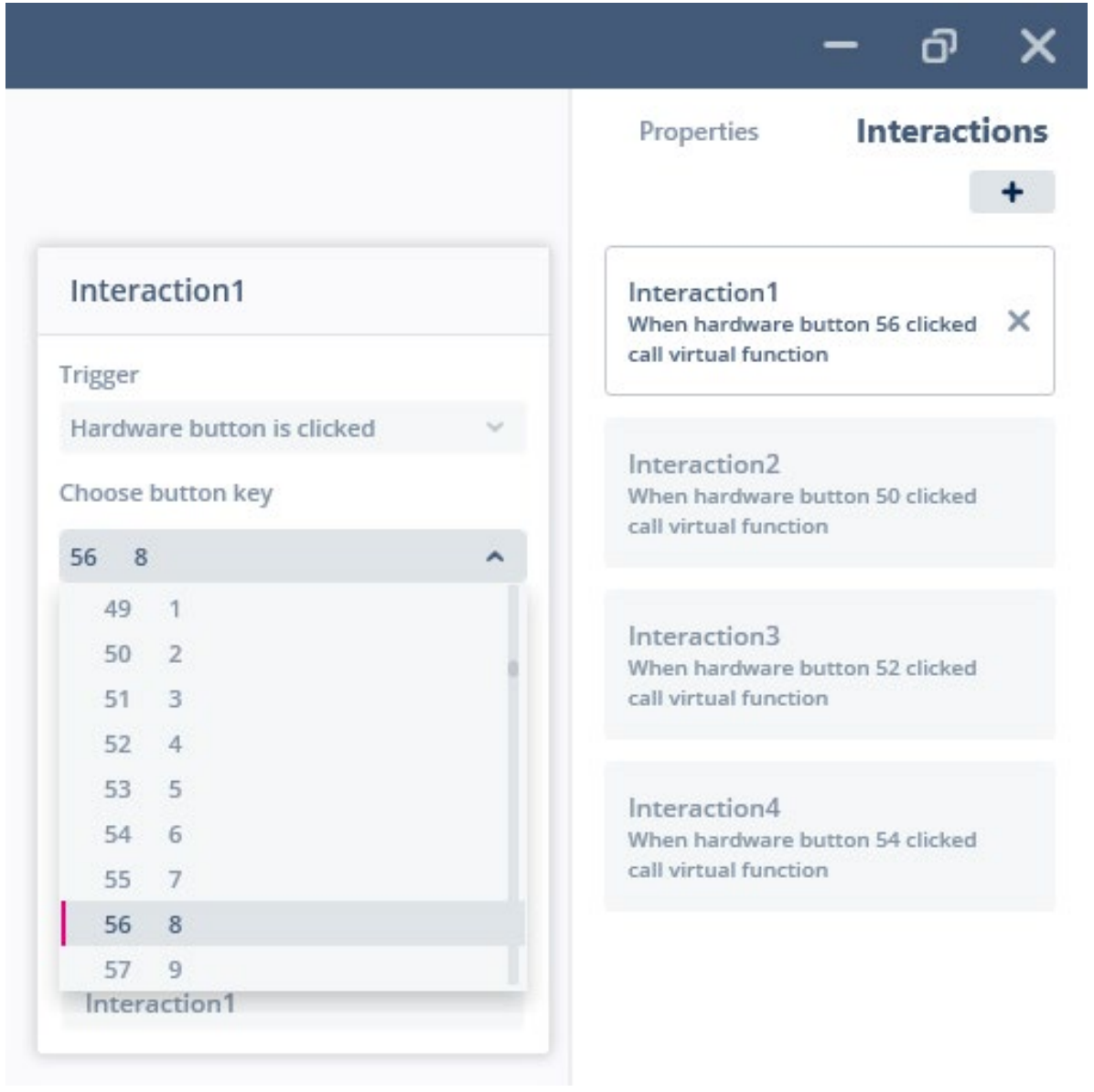
Step 1: Create the TouchGFX Project



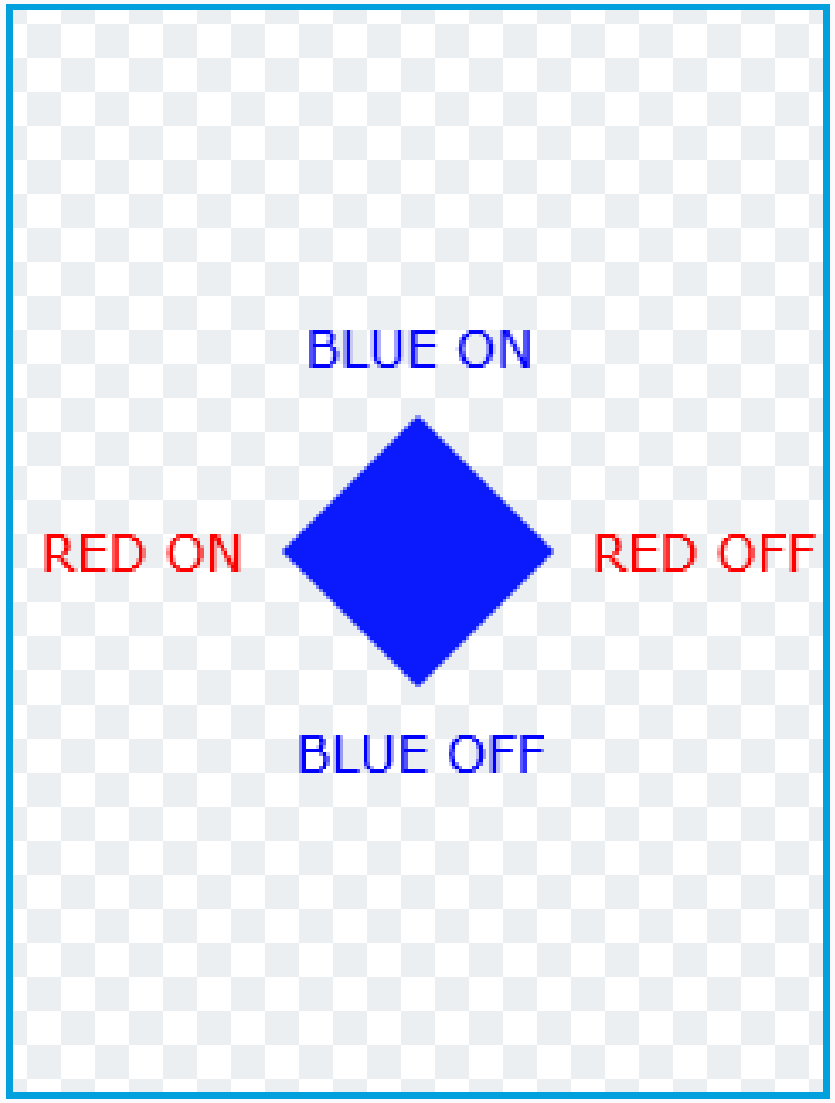


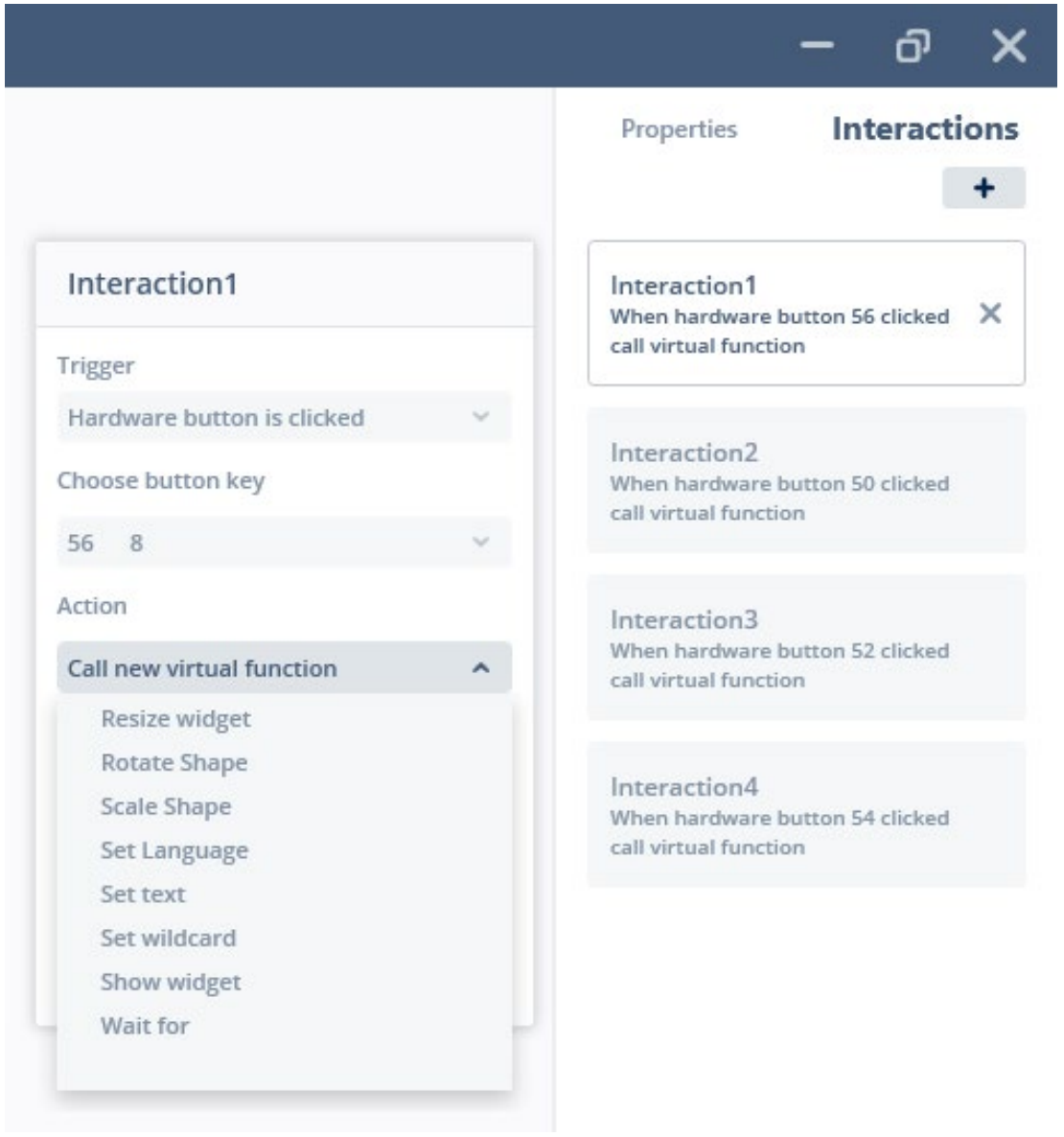
Step 1: Create the TouchGFX Project



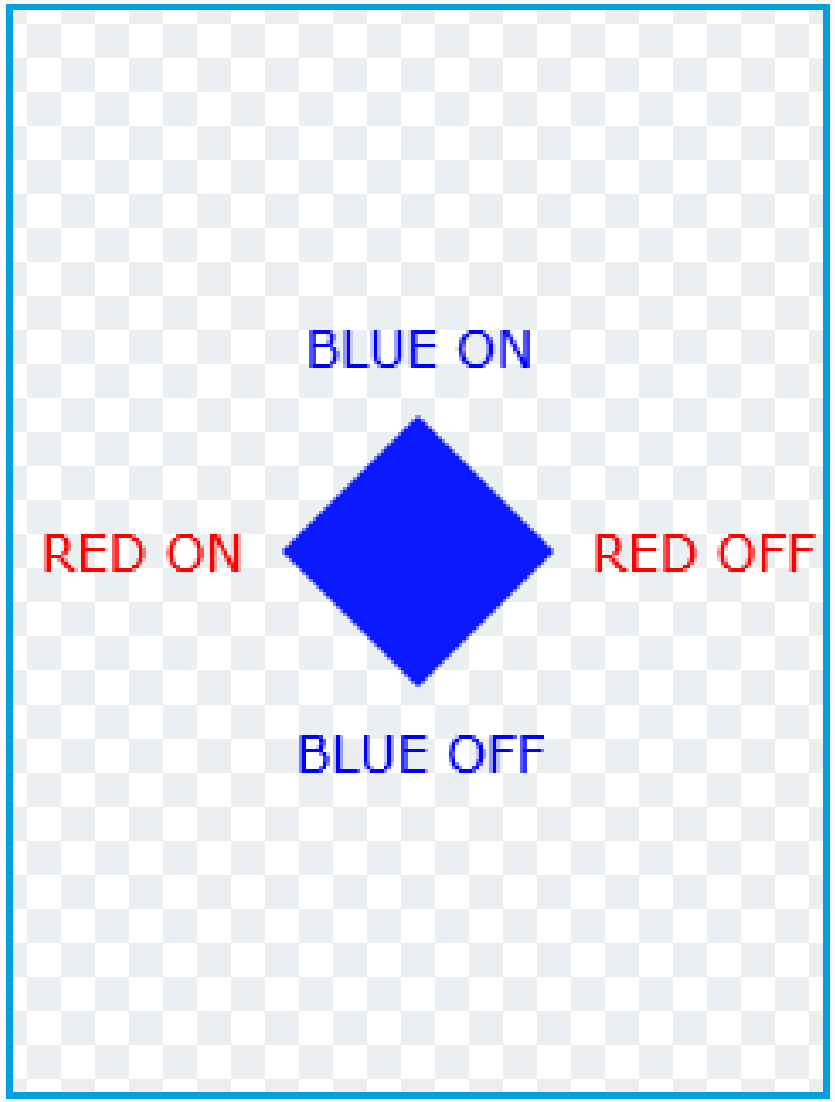


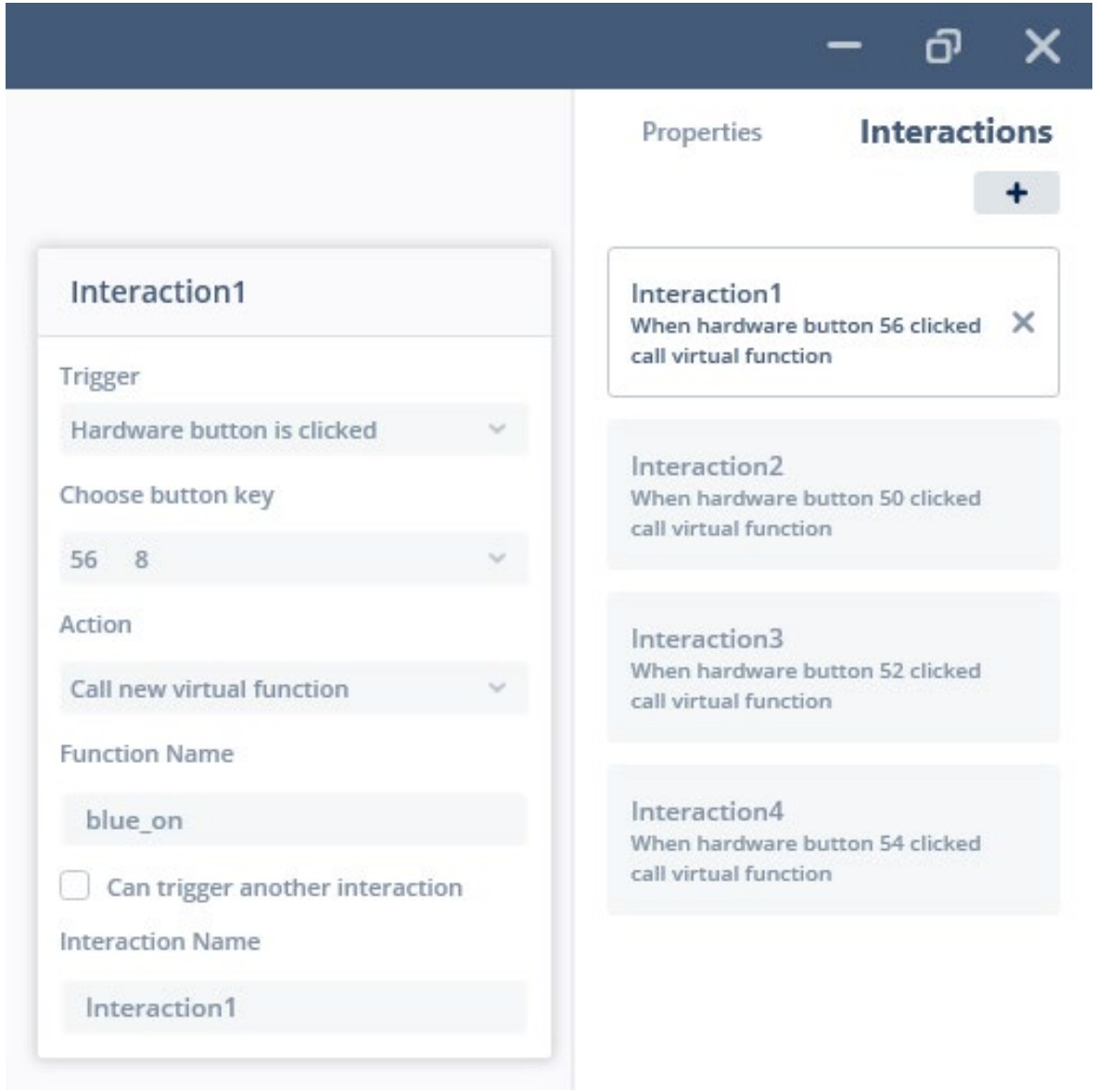
Step 1: Create the TouchGFX Project



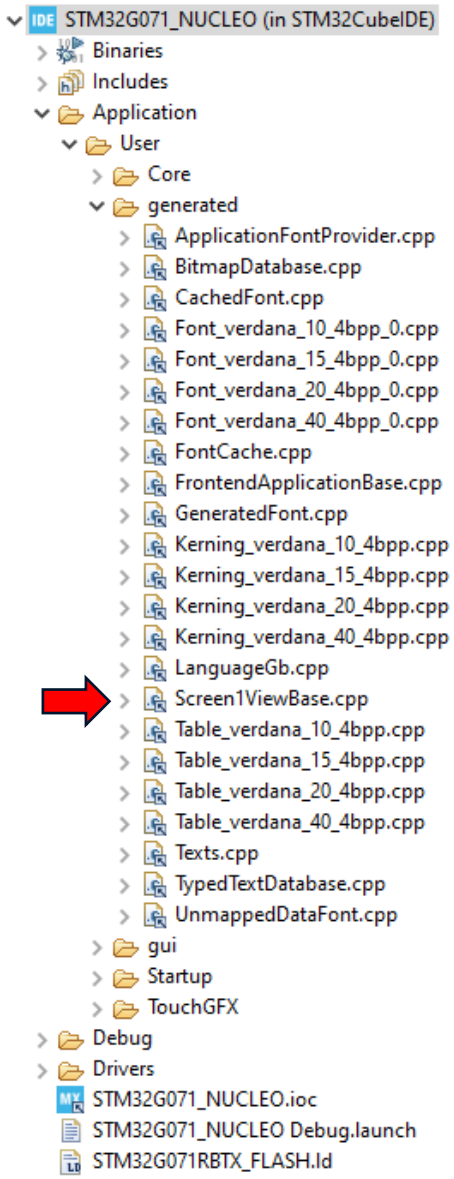


Step 1: Create the TouchGFX Project

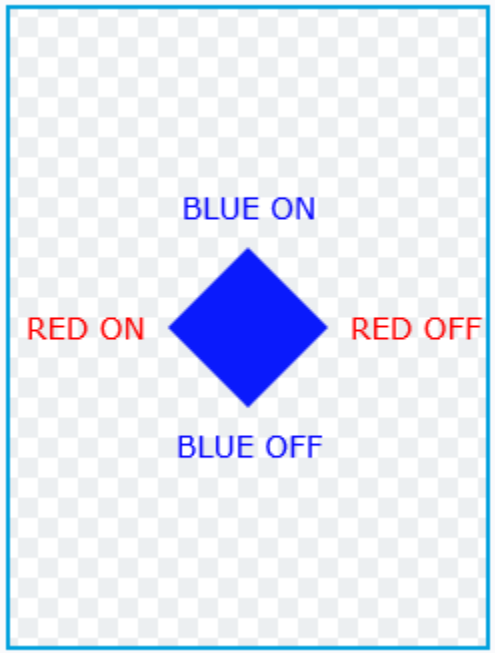




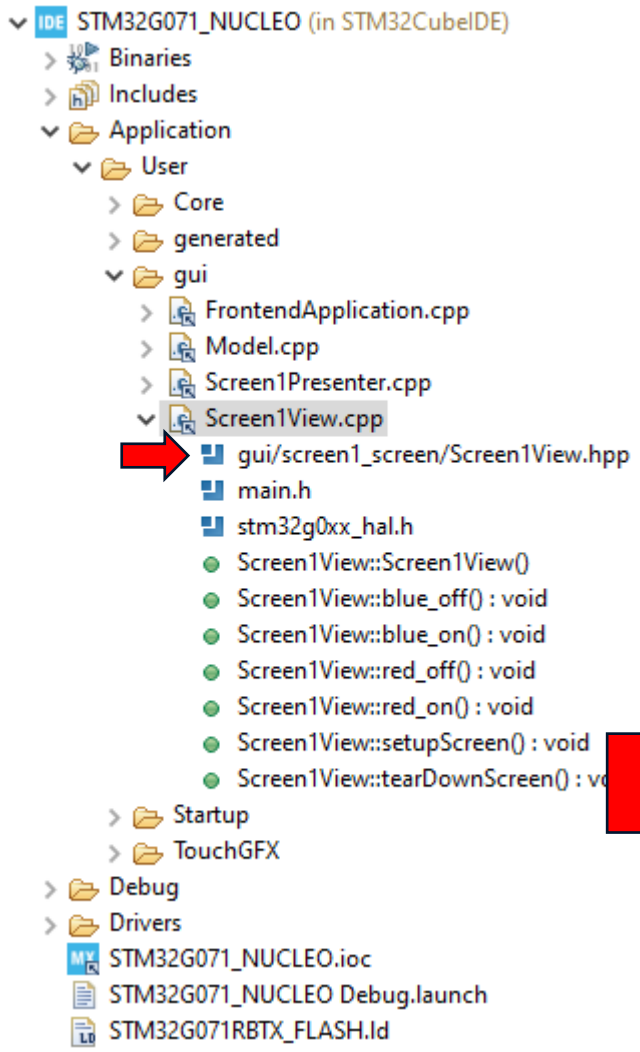
Step 1: Create the TouchGFX Project Code




```
Screen1ViewBase::Screen1ViewBase()
{

    touchgfx::CanvasWidgetRenderer::setupBuffer(canvasBuffer, CANVAS_BUFFER_SIZE);

    __background.setPosition(0, 0, 240, 320);
    __background.setColor(touchgfx::Color::getColorFromRGB(0, 0, 0));

    shape1.setPosition(80, 120, 80, 80);
    shape1.setOrigin(0.000f, 0.000f);
    shape1.setScale(1.000f, 1.000f);
    shape1.setAngle(0.000f);
    shape1Painter.setColor(touchgfx::Color::getColorFromRGB(10, 26, 252));
    shape1.setPainter(shape1Painter);
    const touchgfx::AbstractShape::ShapePoint<float> shape1Points[4] = { { 40.000f, 0.000f }, { 80.000f, 40.000f }, { 40.000f, 80.000f }, { 0.000f, 40.000f } };
    shape1.setShape(shape1Points);

    textArea1.setXY(87, 91);
    textArea1.setColor(touchgfx::Color::getColorFromRGB(0, 0, 255));
    textArea1.setLinespacing(0);
    textArea1.setTypedText(touchgfx::TypedText(T___SINGLEUSE_HCVI));

    textArea2.setXY(84, 210);
    textArea2.setColor(touchgfx::Color::getColorFromRGB(0, 0, 255));
    textArea2.setLinespacing(0);
    textArea2.setTypedText(touchgfx::TypedText(T___SINGLEUSE_MZ1W));

    textArea3.setXY(9, 151);
    textArea3.setColor(touchgfx::Color::getColorFromRGB(255, 0, 0));
    textArea3.setLinespacing(0);
    textArea3.setTypedText(touchgfx::TypedText(T___SINGLEUSE_HLDK));

    textArea4.setXY(171, 151);
    textArea4.setColor(touchgfx::Color::getColorFromRGB(255, 0, 0));
    textArea4.setLinespacing(0);
    textArea4.setTypedText(touchgfx::TypedText(T___SINGLEUSE_RL54));

    add(__background);
    add(shape1);
    add(textArea1);
    add(textArea2);
    add(textArea3);
    add(textArea4);
```

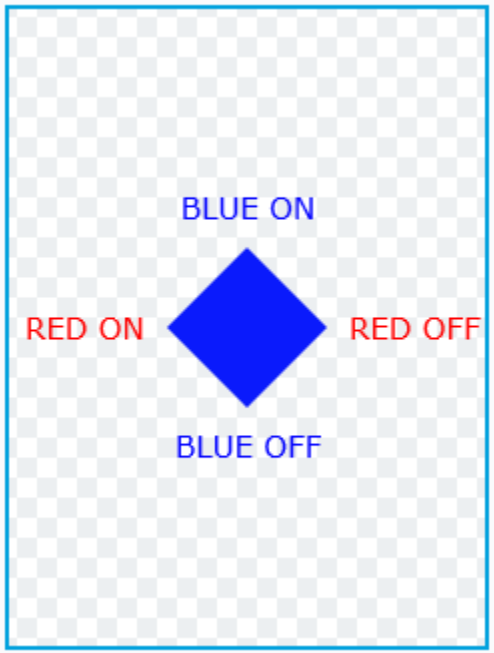


Step 1: Create the TouchGFX Project Code



Add This Code

```
#ifndef SCREEN1VIEW_HPP
#define SCREEN1VIEW_HPP

#include
<gui_generated/screen1_screen/Screen1ViewBase.hpp>
#include <gui/screen1_screen/Screen1Presenter.hpp>

class Screen1View : public Screen1ViewBase
{
public:
    Screen1View();
    virtual ~Screen1View() {}
    virtual void setupScreen();
    virtual void tearDownScreen();
    virtual void blue_on();
    virtual void blue_off();
    virtual void red_on();
    virtual void red_off();
protected:
};

#endif // SCREEN1VIEW_HPP
```



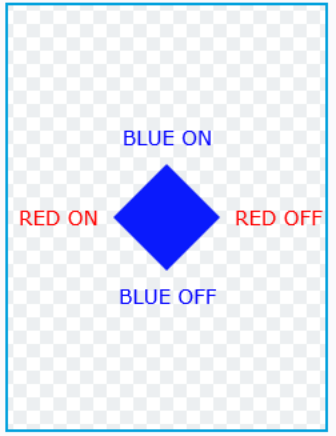
Step 1: Create the TouchGFX Project Code

- STM32G071_NUCLEO (in STM32CubeIDE)
  - Binaries
  - Includes
  - Application
    - User
      - Core
      - generated
      - gui
        - FrontendApplication.cpp
        - Model.cpp
        - Screen1Presenter.cpp
        - Screen1View.cpp
          - gui/screen1_screen/Screen1View.hpp
          - main.h
          - stm32g0xx_hal.h
          - Screen1View::Screen1View()
          - Screen1View::blue_off() : void
          - Screen1View::blue_on() : void
          - Screen1View::red_off() : void
          - Screen1View::red_on() : void
          - Screen1View::setupScreen() : void
          - Screen1View::tearDownScreen() : void
      - Startup
      - TouchGFX
  - Debug
  - Drivers
  - STM32G071_NUCLEO.ioc
  - STM32G071_NUCLEO Debug.launch
  - STM32G071RBTX_FLASH.ld

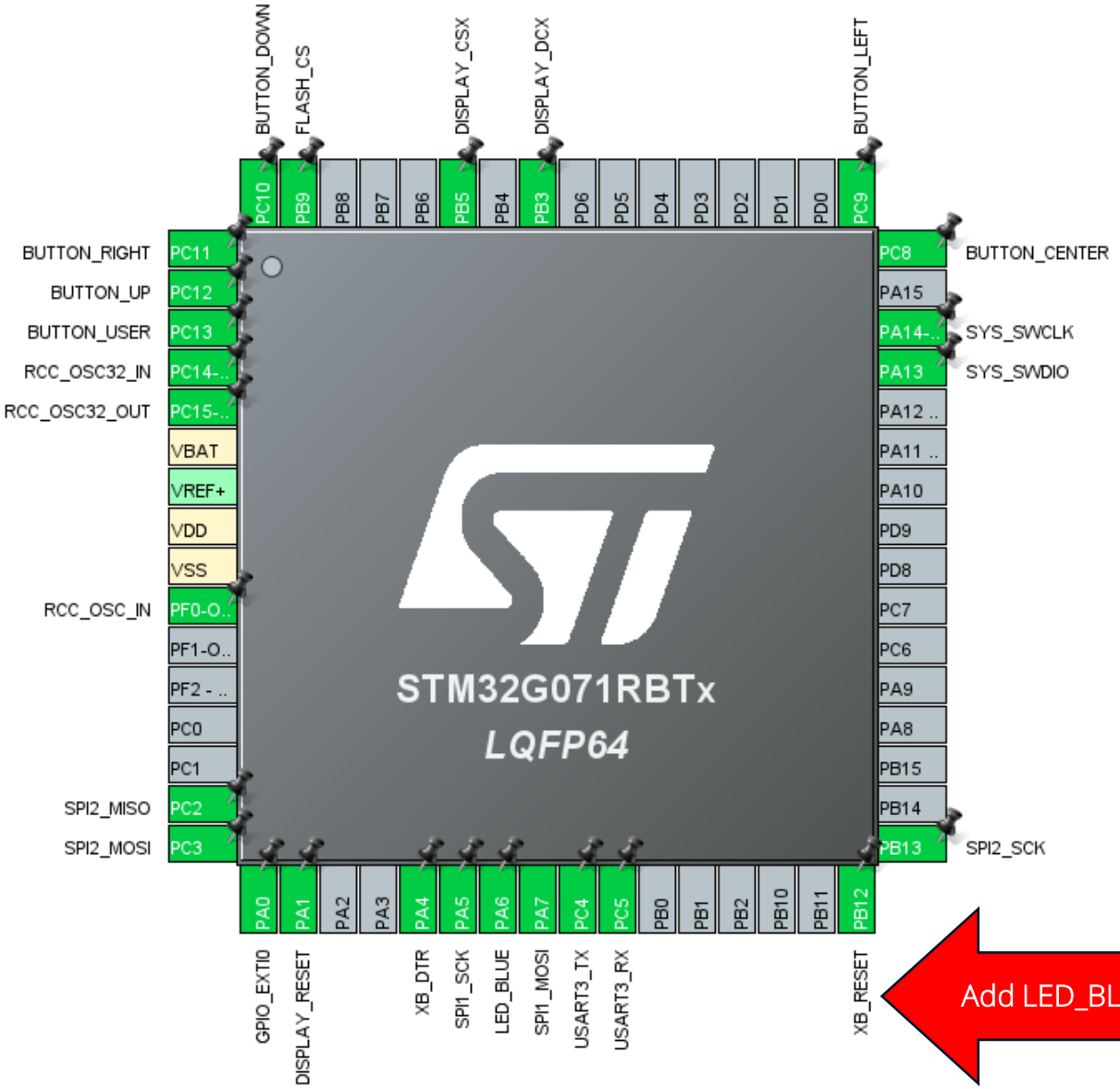
Add This Code

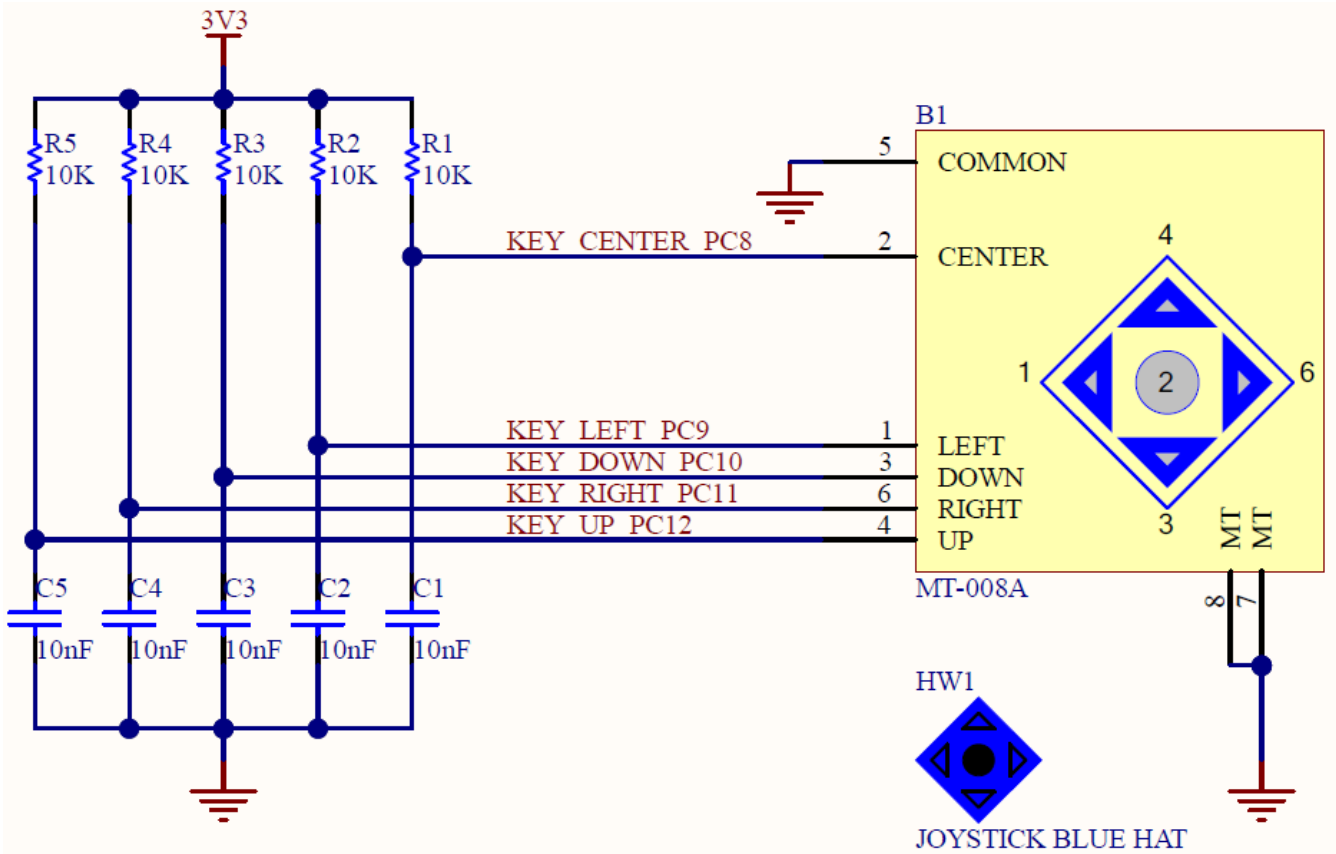
```
#include <gui/screen1_screen/Screen1View.hpp>
#include "main.h"
#include "stm32g0xx_hal.h"
extern UART_HandleTypeDef huart3;

Screen1View::Screen1View()
{

}
void Screen1View::setupScreen()
{
    Screen1ViewBase::setupScreen();
}
void Screen1View::tearDownScreen()
{
    Screen1ViewBase::tearDownScreen();
}

void Screen1View::blue_on()
{
  uint8_t txBite = 56;
  HAL_UART_Transmit(&huart3,&txBite,1,0xFFFF);
}

void Screen1View::blue_off()
{
  uint8_t txBite = 50;
  HAL_UART_Transmit(&huart3,&txBite,1,0xFFFF);
}

void Screen1View::red_on()
{
  uint8_t txBite = 52;
  HAL_UART_Transmit(&huart3,&txBite,1,0xFFFF);
}

void Screen1View::red_off()
{
  uint8_t txBite = 54;
  HAL_UART_Transmit(&huart3,&txBite,1,0xFFFF);
}
```

Key	Code
Left	'4'
Right	'6'
Up	'8'
Down	'2'
Center	'5'
Blue User Button	'0'



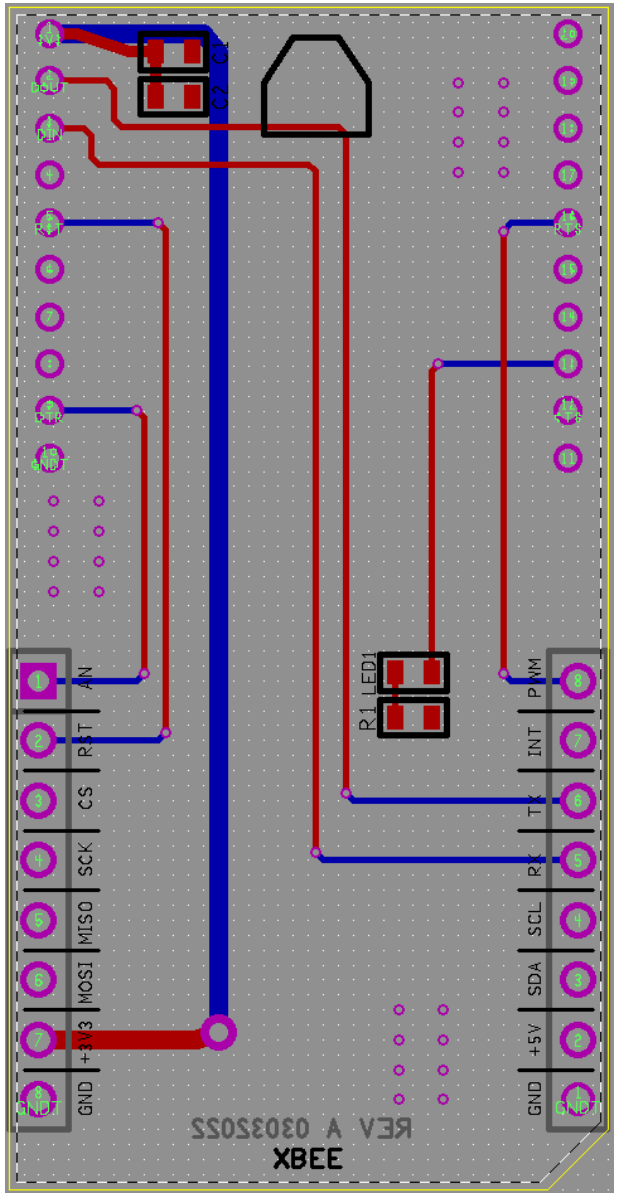
Step 2: Configure the TouchGFX Project Hardware



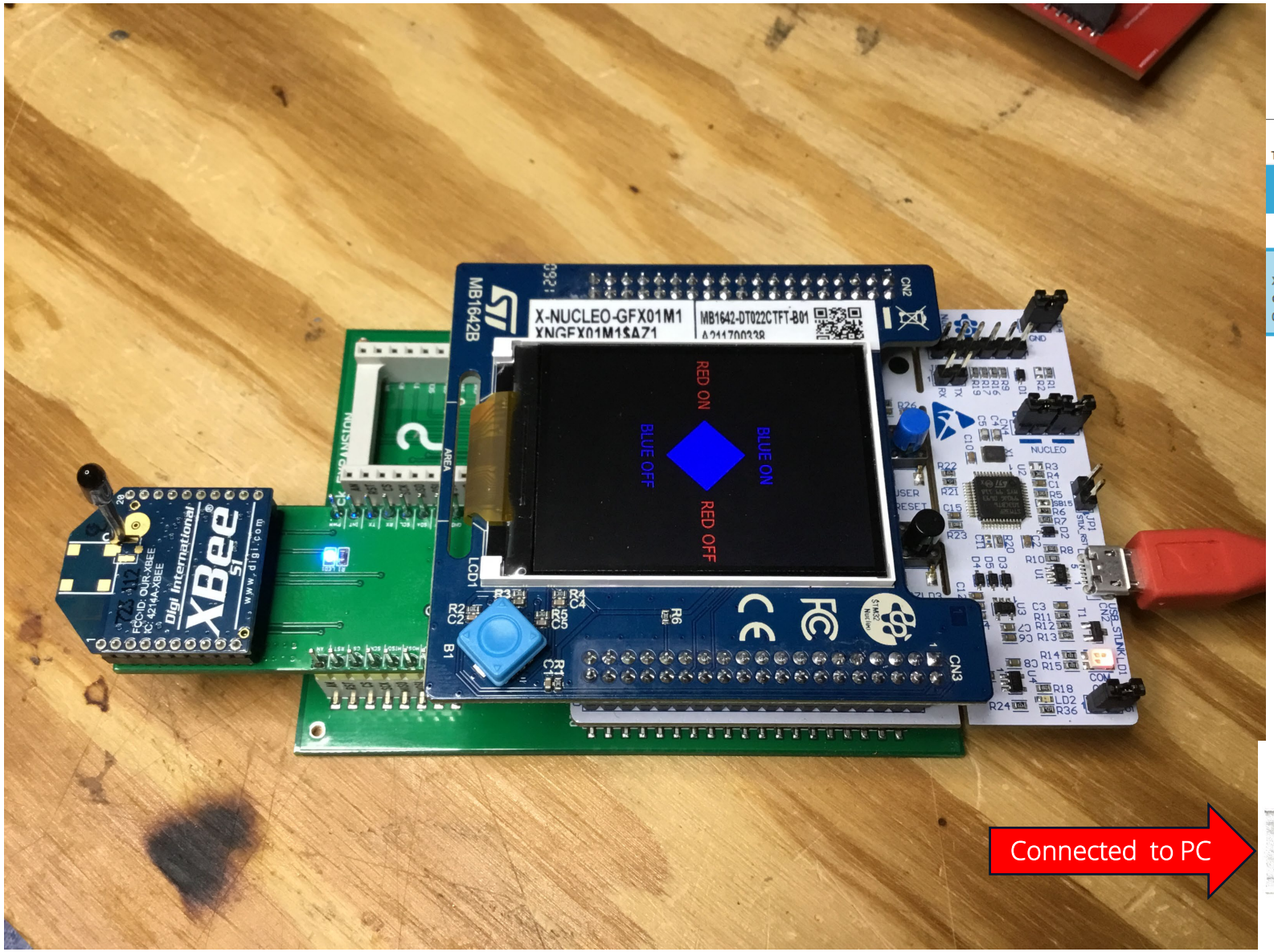


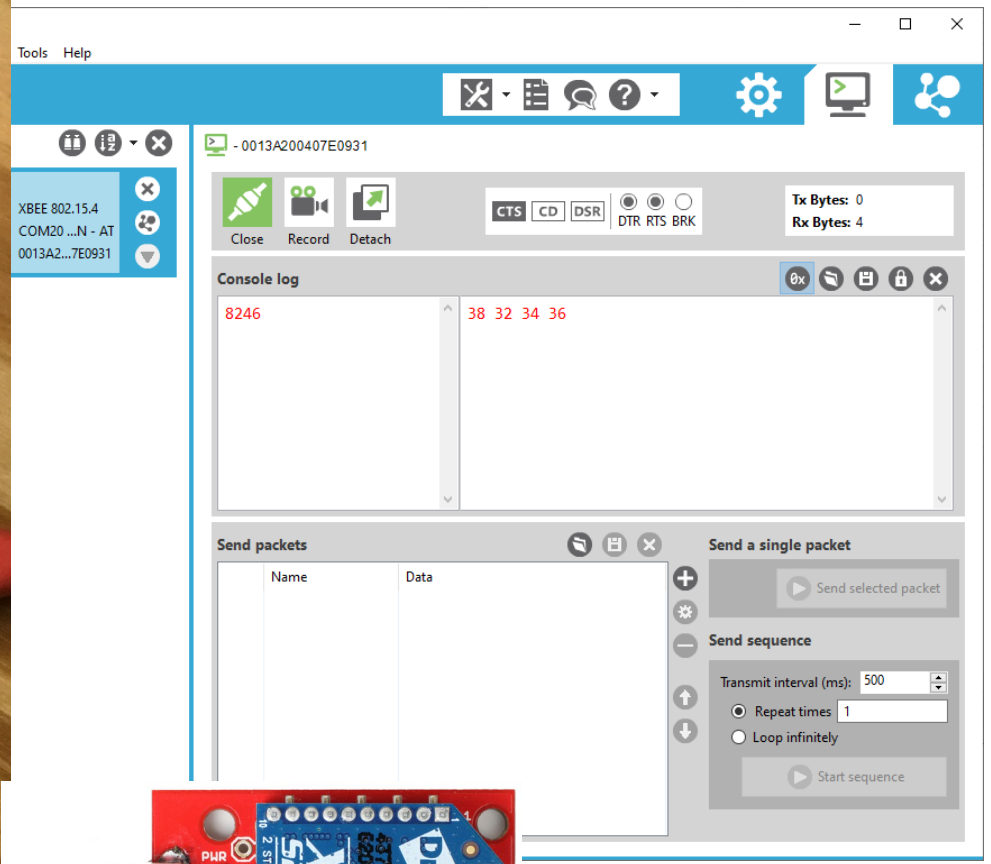
Add LED_BLUE, USART3, XB_RESET and XB_DTR

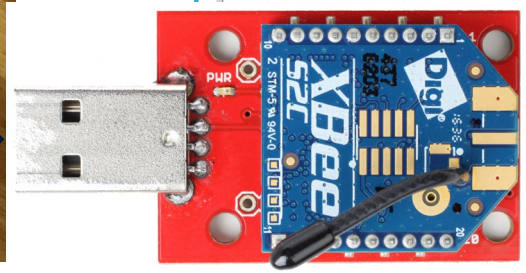
Step 3: Design and Construct the XBee Transmitter



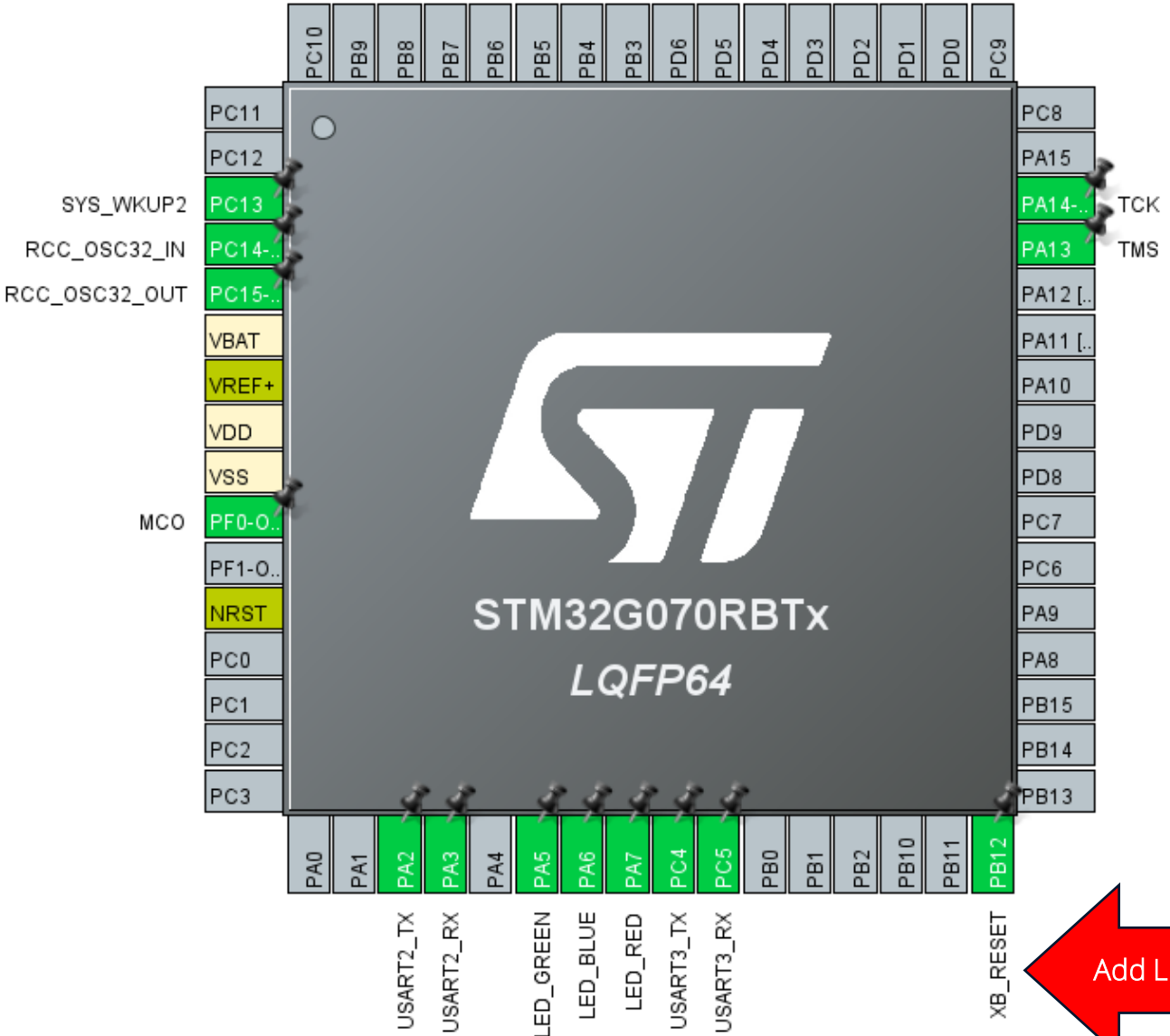
Step 4: Test the XBee Transmitter

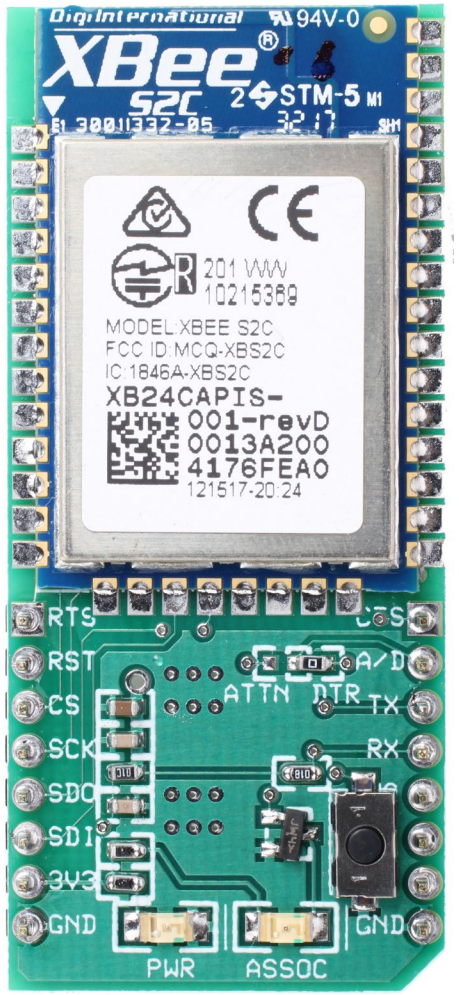






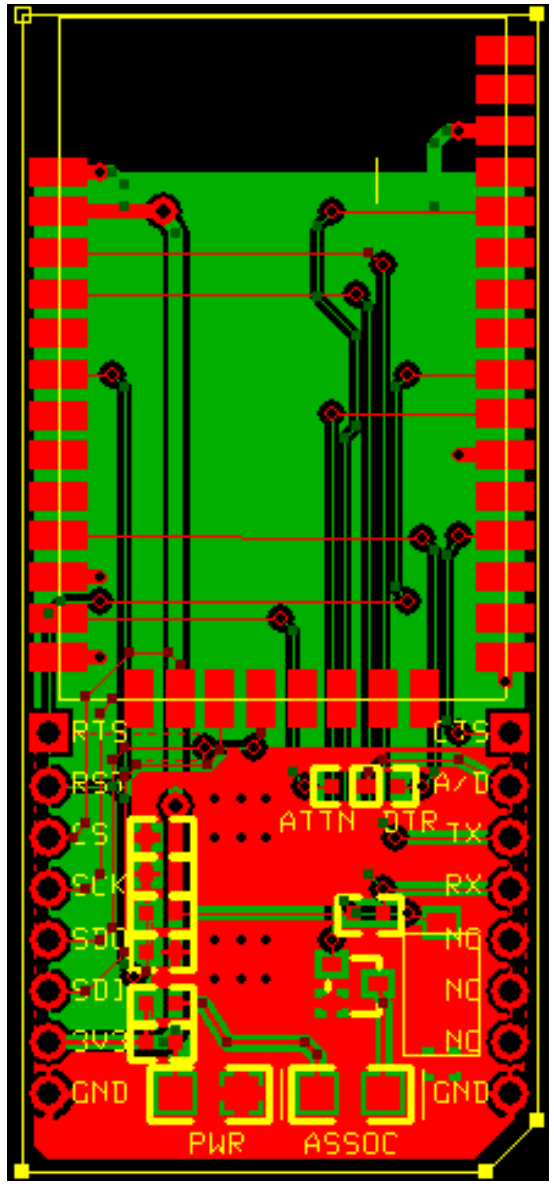
Step 5: Configure the XBee Receiver Hardware

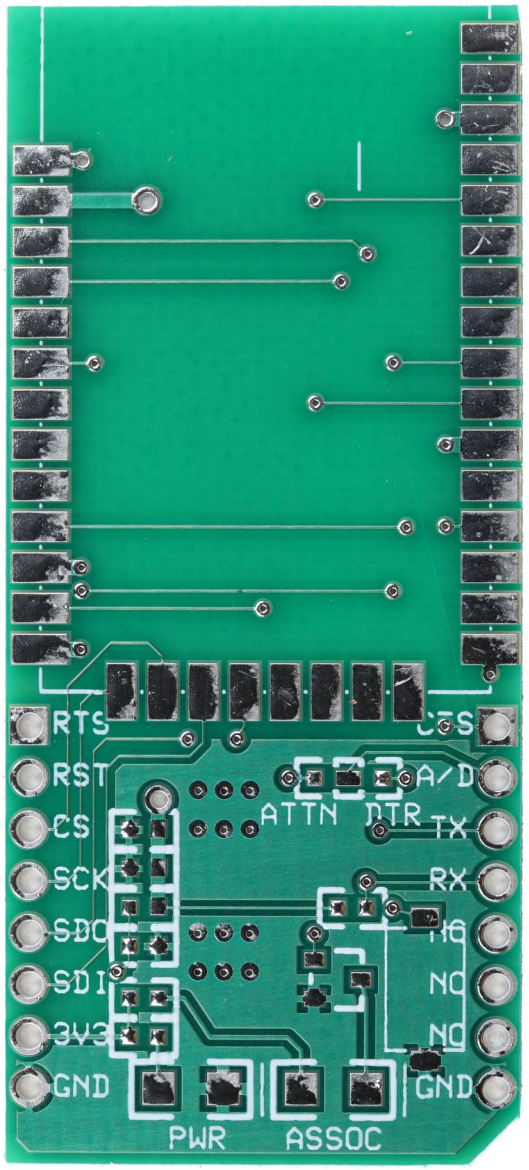


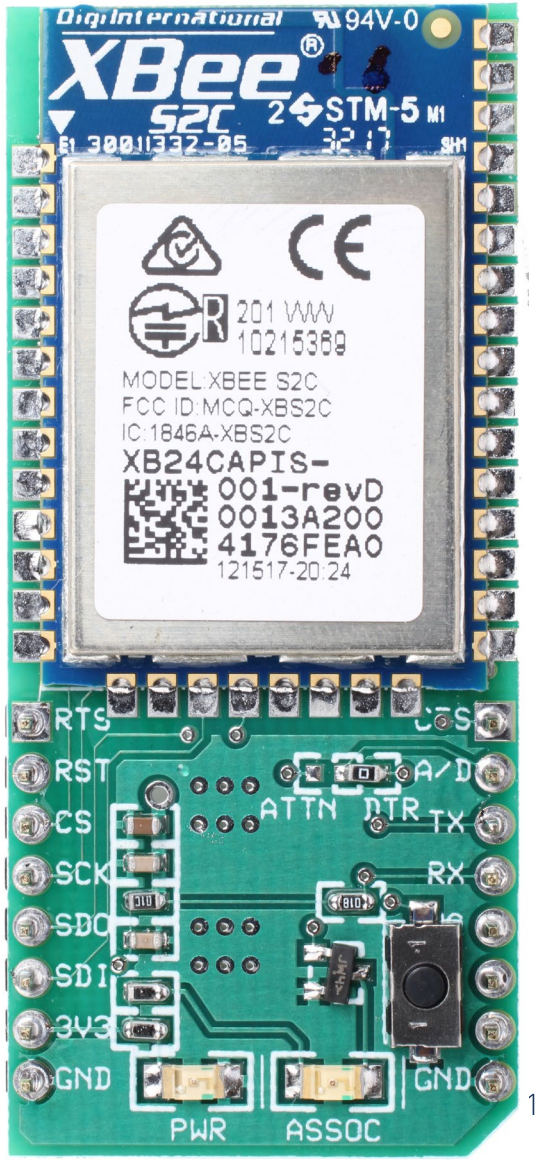


Add LED_BLUE, LED_RED, USART3 and XB_RESET

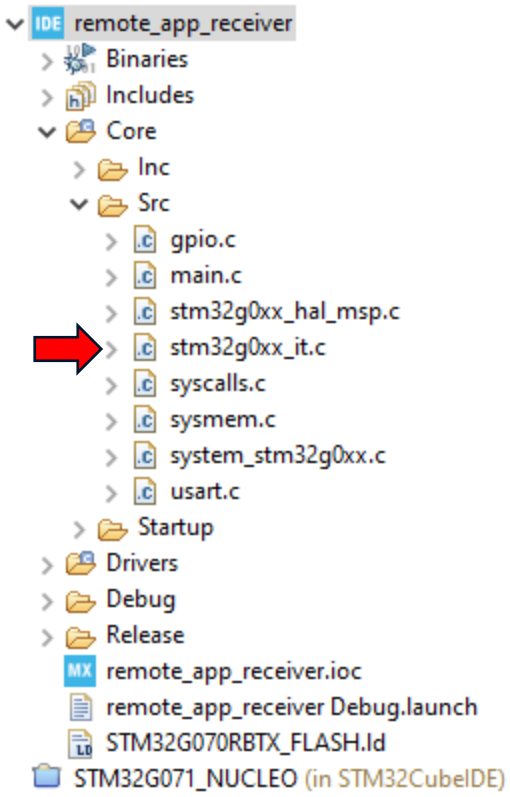
Step 6: Design and Construct the XBee Receiver



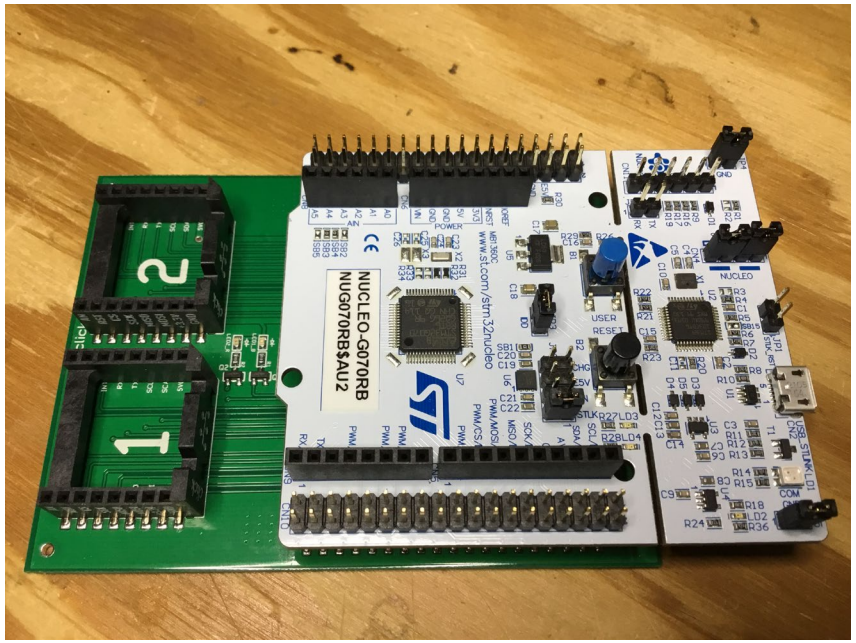




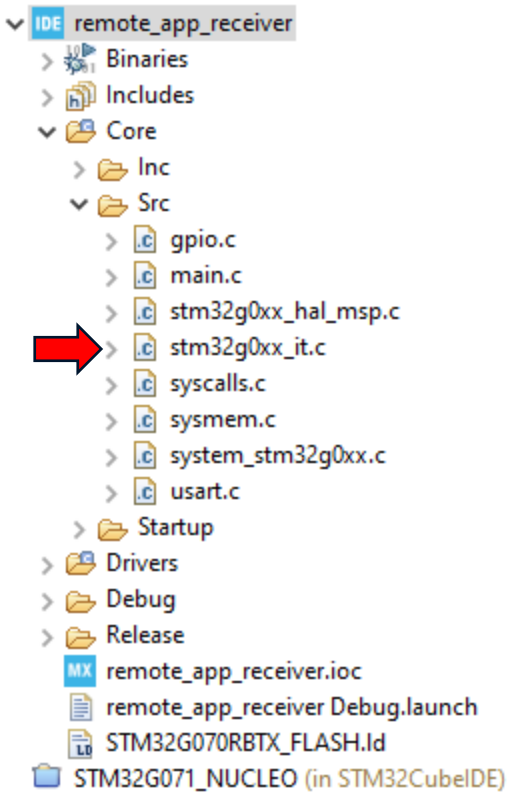
Step 7: Code the XBee Receiver Application



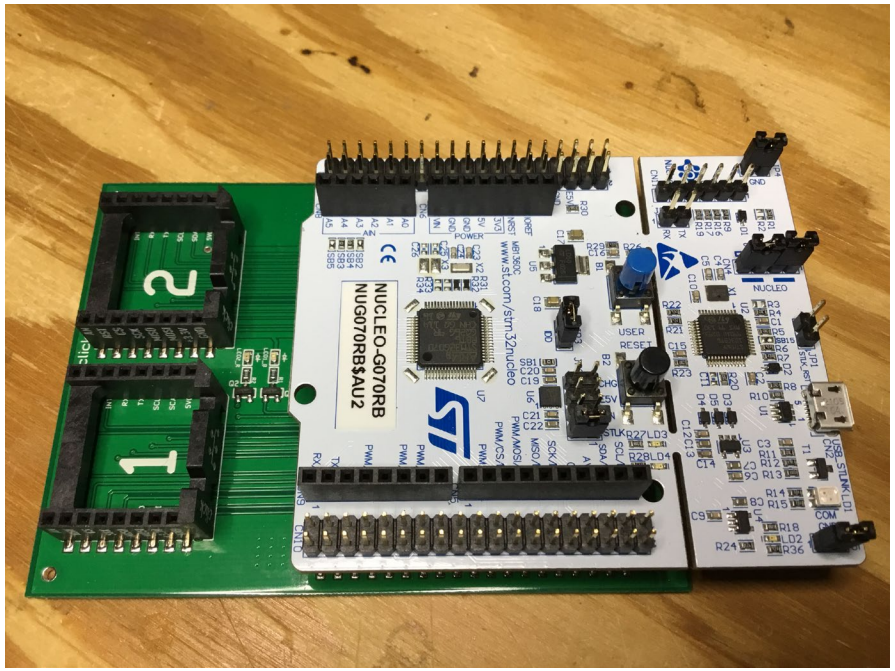

```
extern UART_HandleTypeDef huart3;
#define USART3_RX_BUFFER_SIZE128
#define USART3_RX_BUFFER_MASK (
USART3_RX_BUFFER_SIZE - 1 )
extern uint8_t USART3_RxHead;
extern uint8_t USART3_RxTail;
extern uint8_t
USART3_RxBuffer[USART3_RX_BUFFER_SIZE];
extern uint8_t
USART3_RxBuf[USART3_RX_BUFFER_SIZE];
```



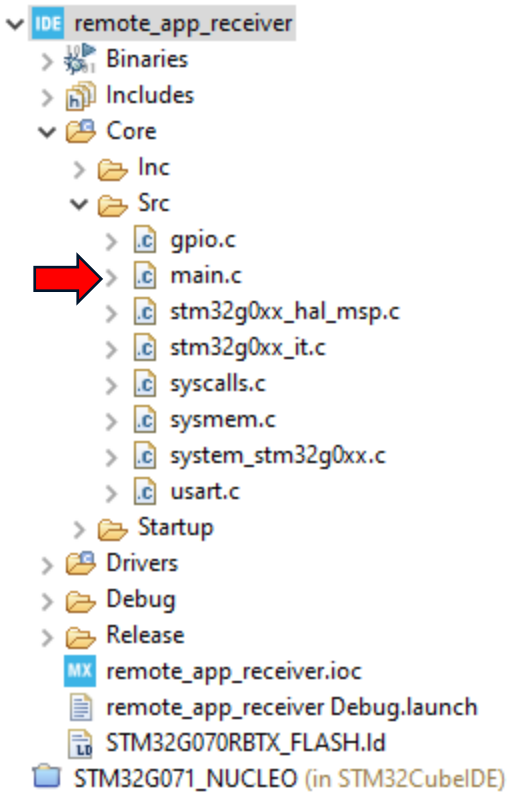
Step 7: Code the XBee Receiver Application



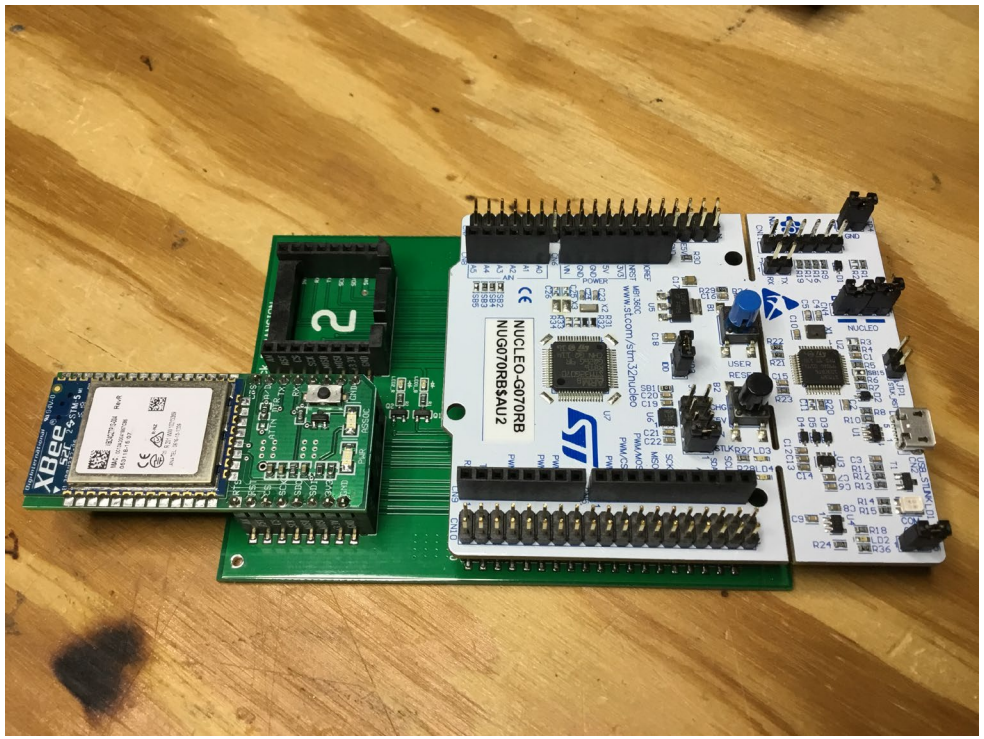

```
void USART3_4_IRQHandler(void)
{
  /* USER CODE BEGIN USART3_4_IRQn 0 */
  uint32_t isrflags   = READ_REG(huart3.Instance->ISR);
  uint32_t cr1its     = READ_REG(huart3.Instance->CR1);
  uint32_t cr3its     = READ_REG(huart3.Instance->CR3);

  uint32_t errorflags;
  uint16_t data;
  uint8_t tmphead;

  uint16_t  dataMask = 0x00FF;

  /* If no error occurs */
  errorflags = (isrflags & (uint32_t)(USART_ISR_PE | USART_ISR_FE | USART_ISR_ORE | USART_ISR_NE | USART_ISR_RTOF));
  if (errorflags == 0U)
  {
    // UART in mode Receiver -------------------------------------------------
    //if(((isrflags & USART_ISR_RXNE) != 0U) && ((cr1its & USART_CR1_RXNEIE) != 0U))
    if (((isrflags & USART_ISR_RXNE_RXFNE) != 0U)
            && (((cr1its & USART_CR1_RXNEIE_RXFNEIE) != 0U)
                || ((cr3its & USART_CR3_RXFTIE) != 0U)))
    {
      // read byte from UART
      data = (uint16_t)READ_REG(USART3->RDR);

      // calculate buffer index
      tmphead = ( USART3_RxHead + 1 ) & USART3_RX_BUFFER_MASK;
      // store new index
      USART3_RxHead = tmphead;

      if ( tmphead == USART3_RxTail )
      {
        // ERROR! Receive buffer overflow
      }
      // store received data in ring buffer
      USART3_RxBuf[tmphead] = (uint8_t)(dataMask & data);
    }
  }
  else
  {
    __HAL_UART_CLEAR_PEFLAG(&huart3);
    __HAL_UART_CLEAR_FEFLAG(&huart3);
    __HAL_UART_CLEAR_OREFLAG(&huart3);
    __HAL_UART_CLEAR_NEFLAG(&huart3);
  }
```



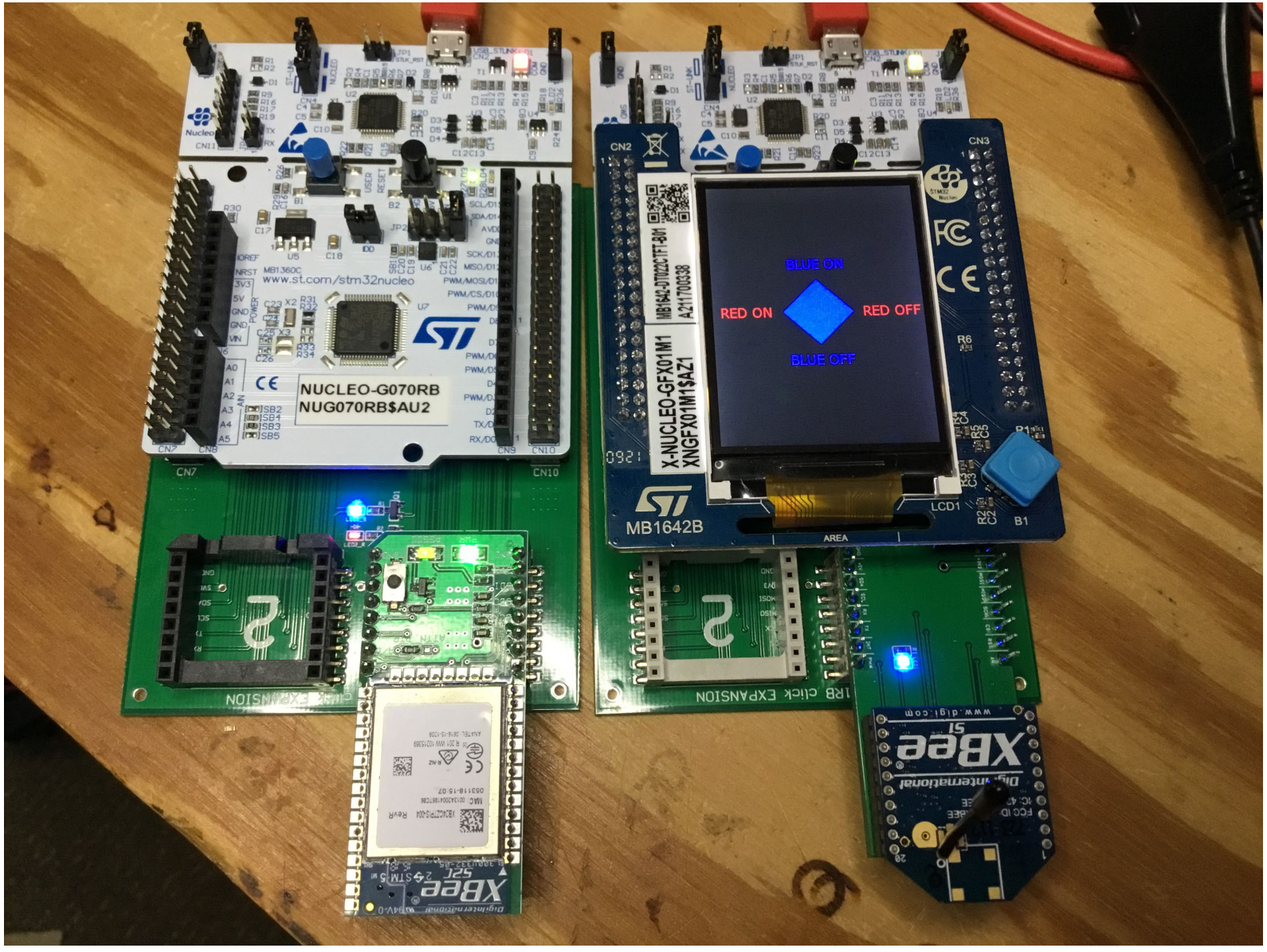
Step 7: Code the XBee Receiver Application




```c
int main(void)
{
  HAL_Init();
  SystemClock_Config();
  MX_GPIO_Init();
  MX_USART2_UART_Init();
  MX_USART3_UART_Init();
  /* USER CODE BEGIN 2 */
  HAL_GPIO_WritePin(XB_RESET_GPIO_Port, XB_RESET_Pin, GPIO_PIN_SET);
  USART3_RxHead = 0x00;
  USART3_RxTail = 0x00;
  __HAL_UART_ENABLE_IT(&huart3,UART_IT_RXNE);
  /* USER CODE END 2 */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    if(CharInRing())
    {
      HAL_Delay(100);
      rxBuf[0] = readring();
      switch(rxBuf[0])
      {
        case 56: //up blue on
          HAL_GPIO_WritePin(LED_BLUE_GPIO_Port, LED_BLUE_Pin, GPIO_PIN_SET);
        break;
        case 50: //down blue off
          HAL_GPIO_WritePin(LED_BLUE_GPIO_Port, LED_BLUE_Pin, GPIO_PIN_RESET);
        break;
        case 52: //left red on
          HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, GPIO_PIN_SET);
        break;
        case 54: //right red off
          HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, GPIO_PIN_RESET);
        break;
      }
    }
  }
```



Step 8: Fire It All Up!



Thank you for attending!!!

Please consider the resources below:

- Today's Project Download Package
- STM32MP1 TouchGFX
- NUCLEO-G070RB User Manual

To get today's Project Download Package please send an email request to:
therealfredeady@gmail.com

LET'S EAT!!

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

Digi-Key ELECTRONICS

informa markets