



DesignNews

IoT Device Prototyping with STMicroelectronics Nucleo Development Boards

Day 2:

Prototyping a NUCLEO-U575ZI-Q Inertial Measurement Unit

Sponsored by



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.

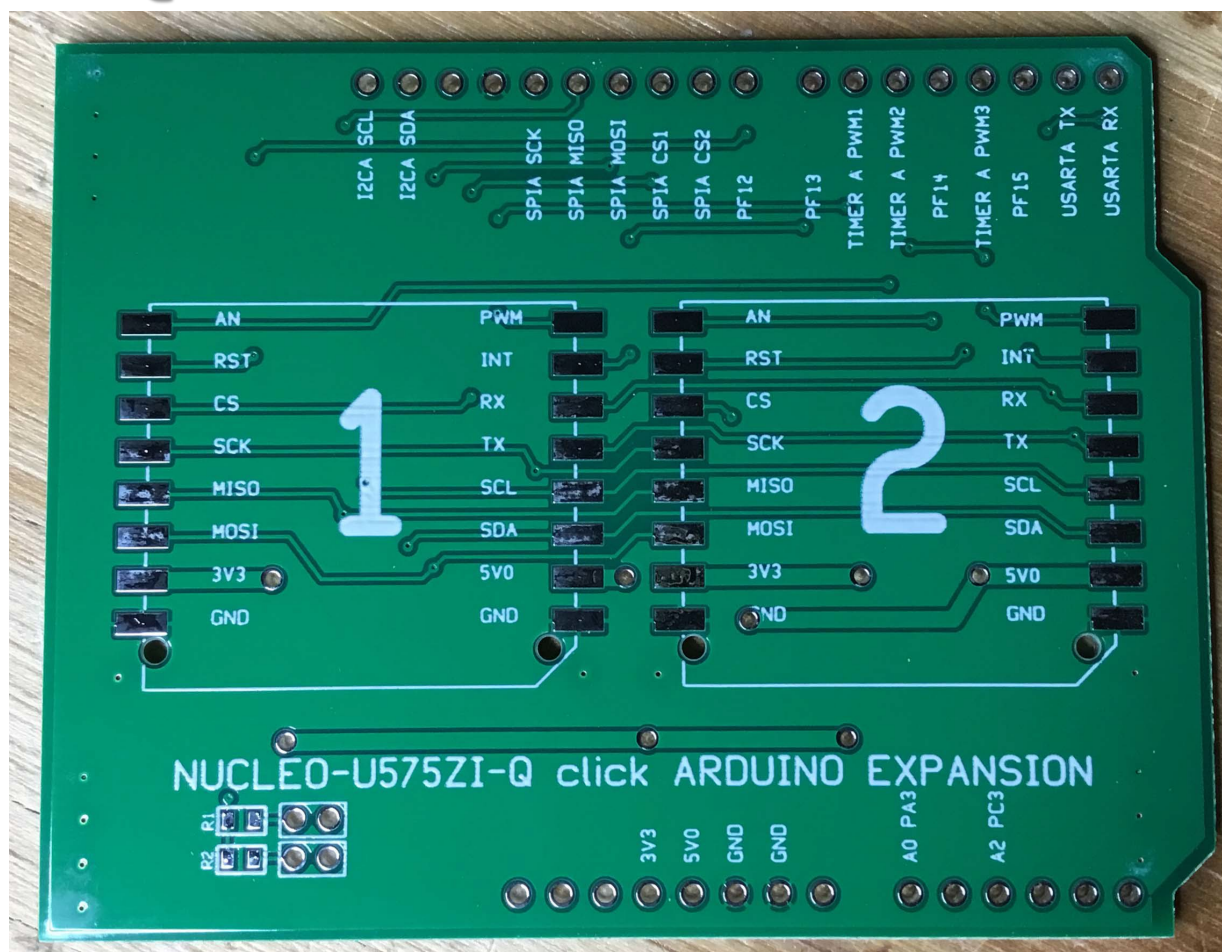


Fred Eady

Visit 'Lecturer Profile' in your console for more details.

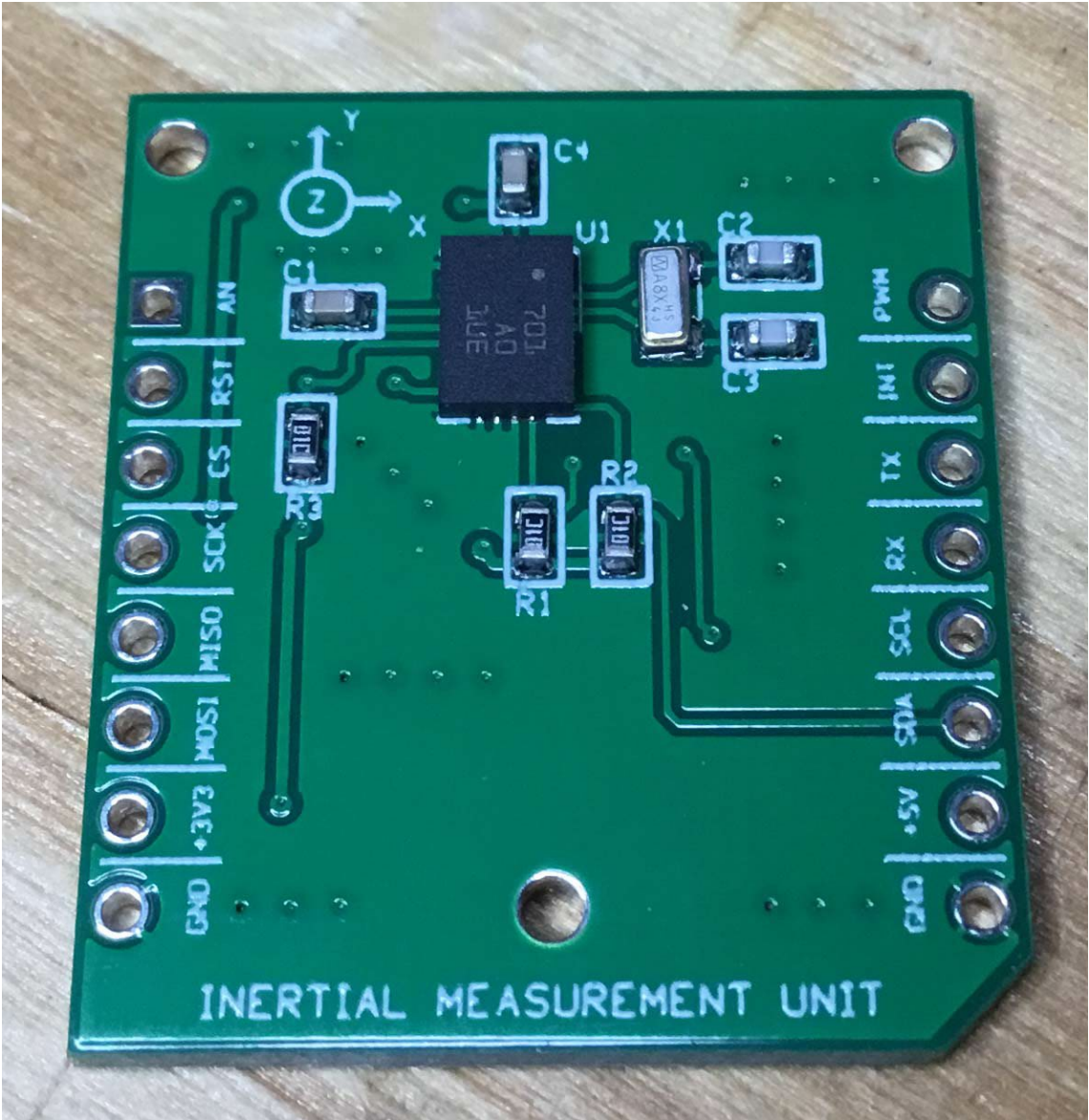
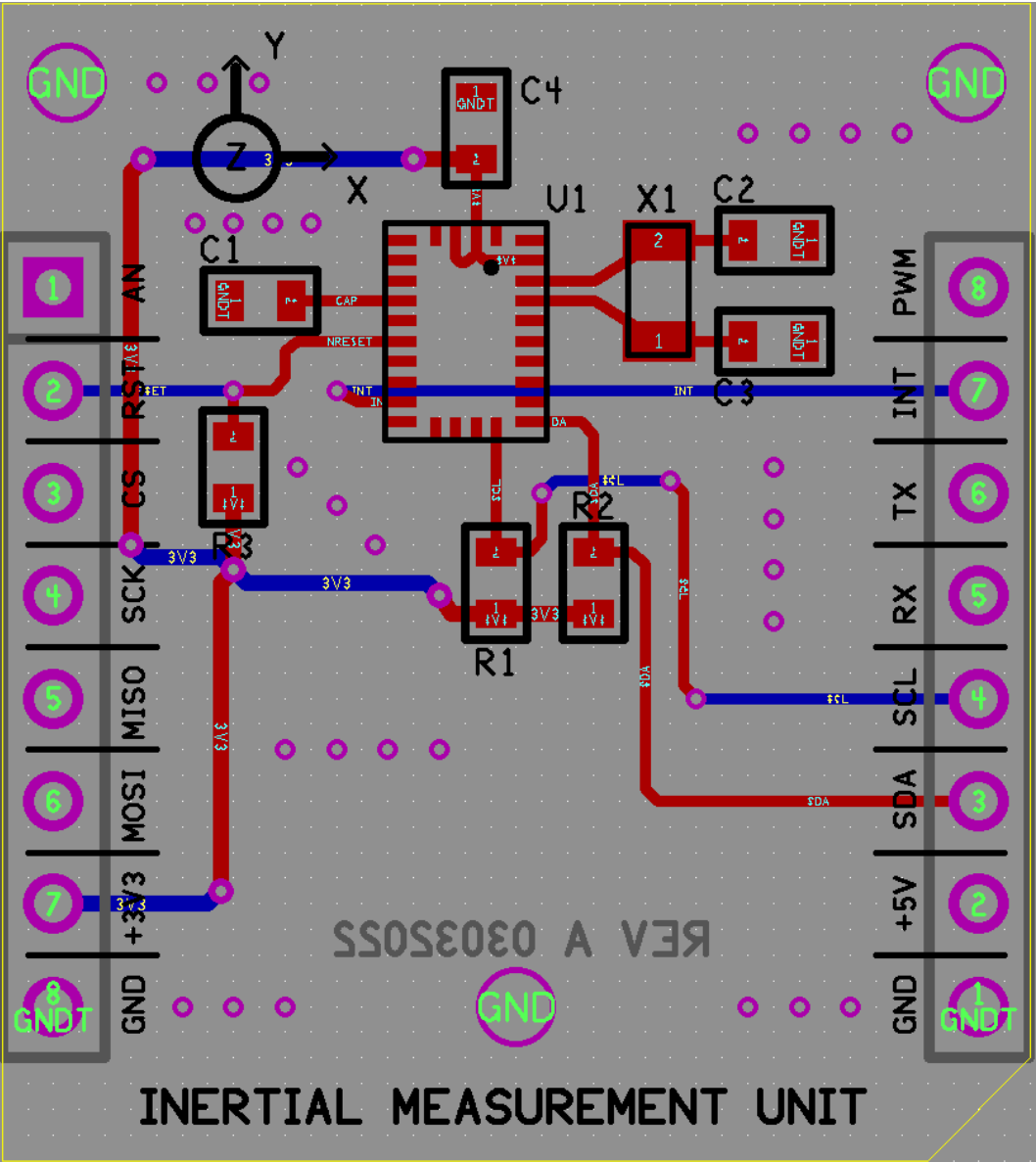
AGENDA

- **Inertial Measurement Unit click**
- **NUCLEO-U575ZI-Q click ARDUINO EXPANSION**
- **IMU Driver**

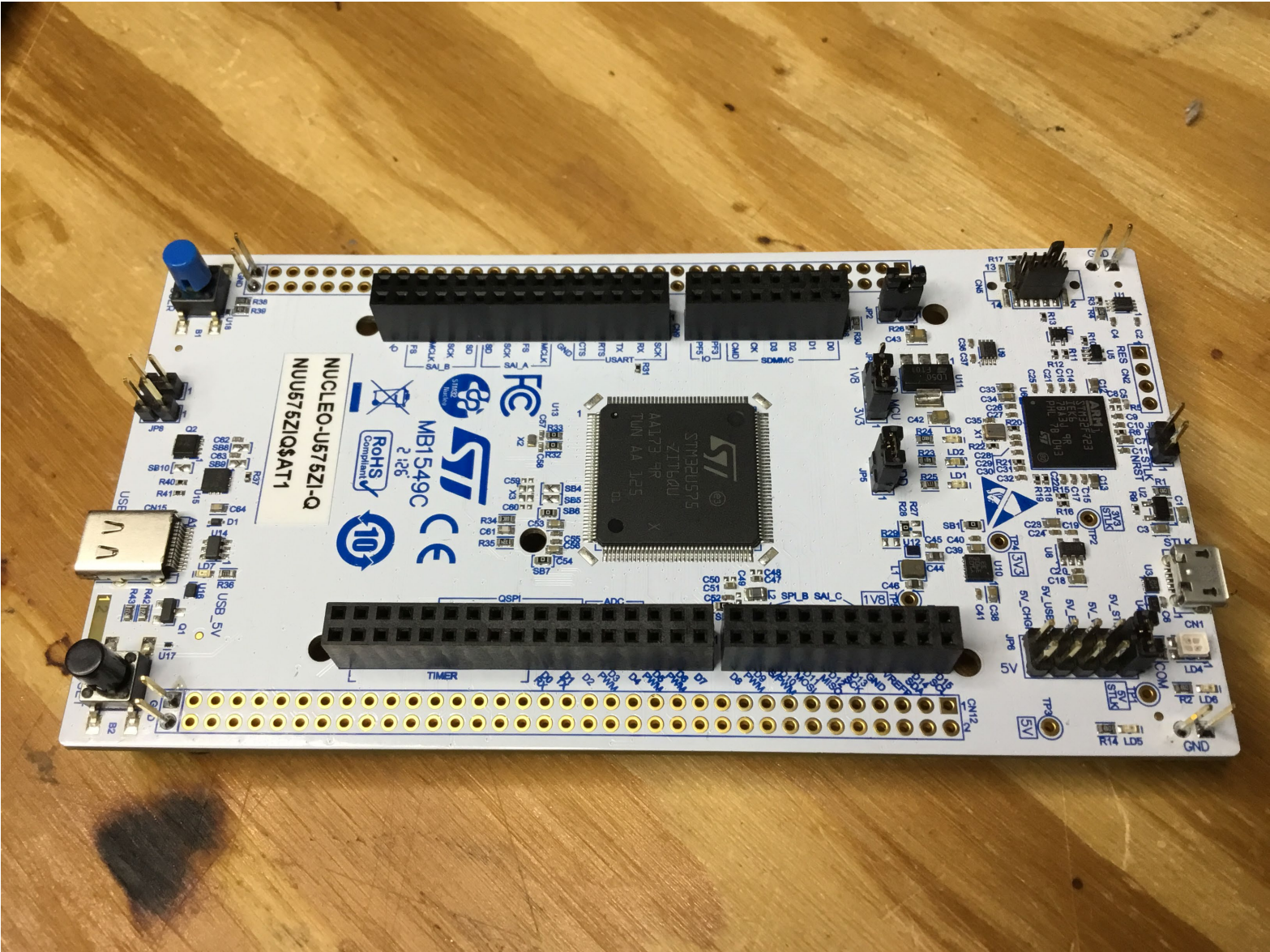


Inertial Measurement Unit

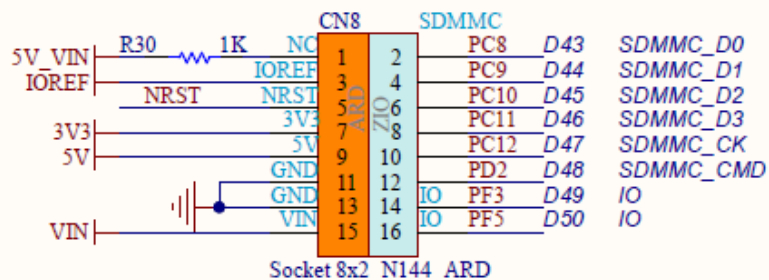
click



NUCLEO-U575ZI-Q click ARDUINO EXPANSION

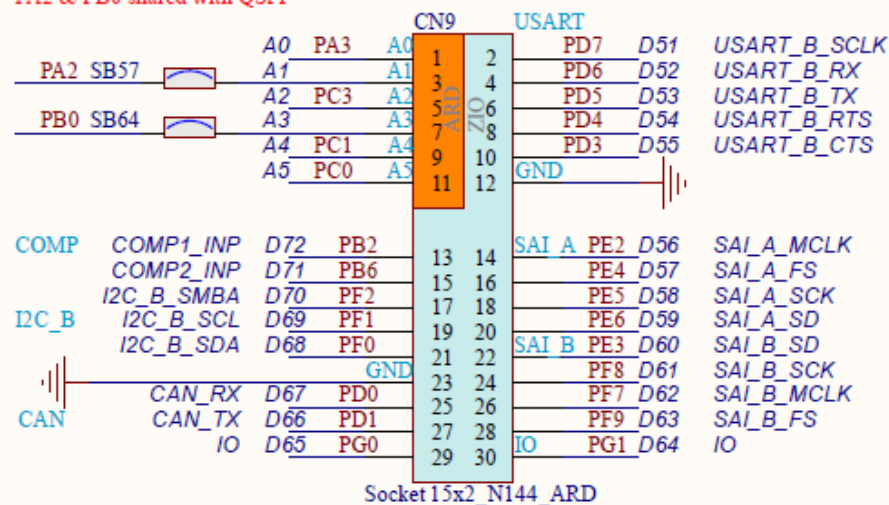


Arduino Uno compatible



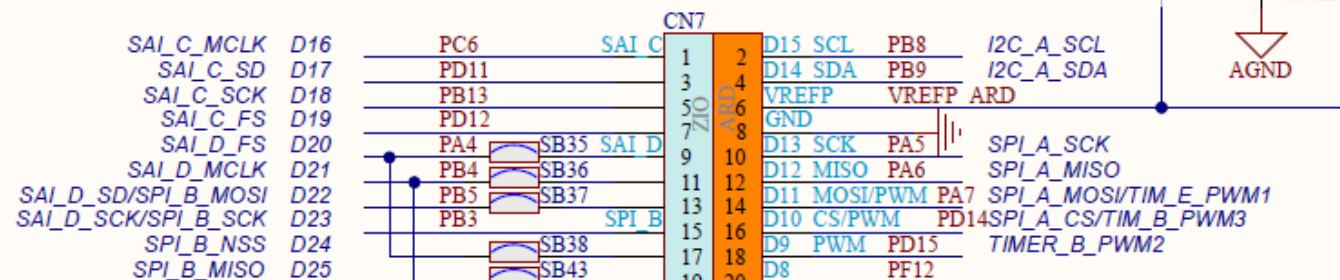
Socket 8x2 N144 ARD

PA2 & PB0 shared with OSPI



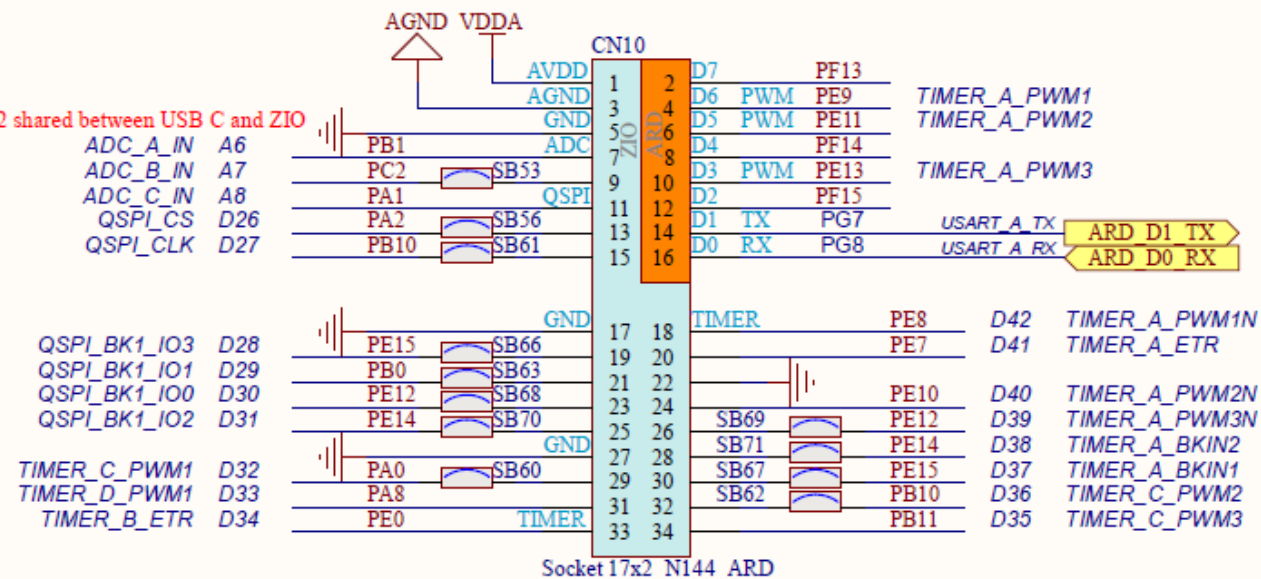
Socket 15x2 N144 ARD

Arduino Uno compatible

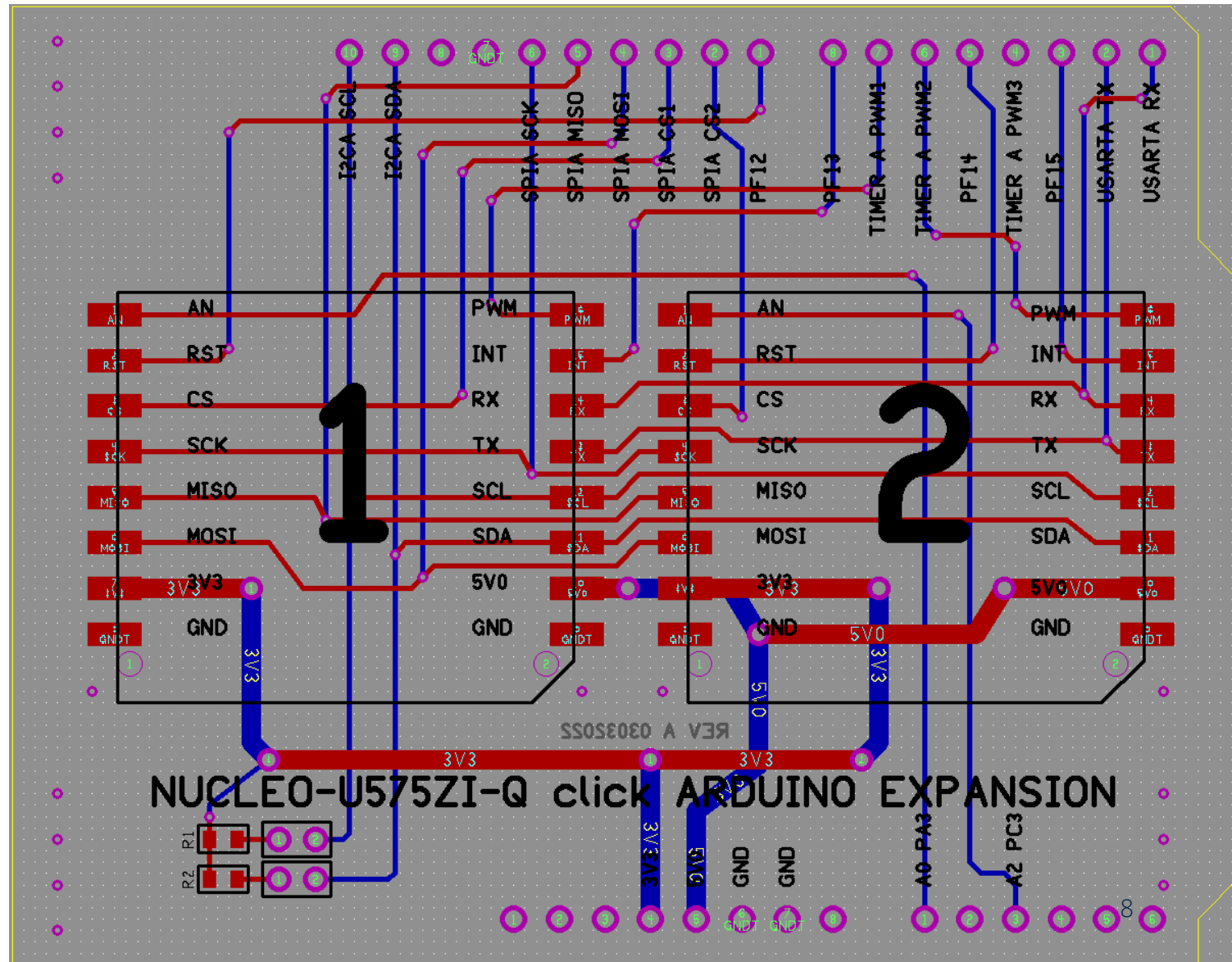
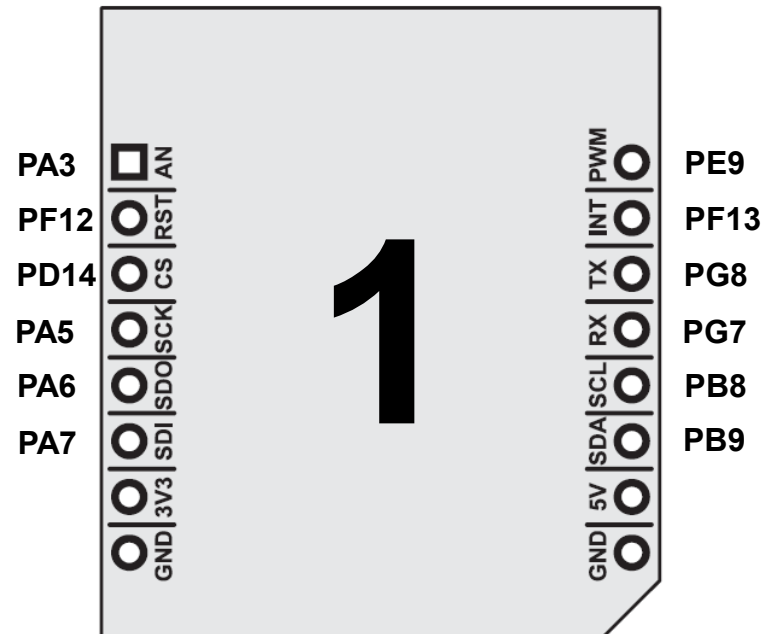


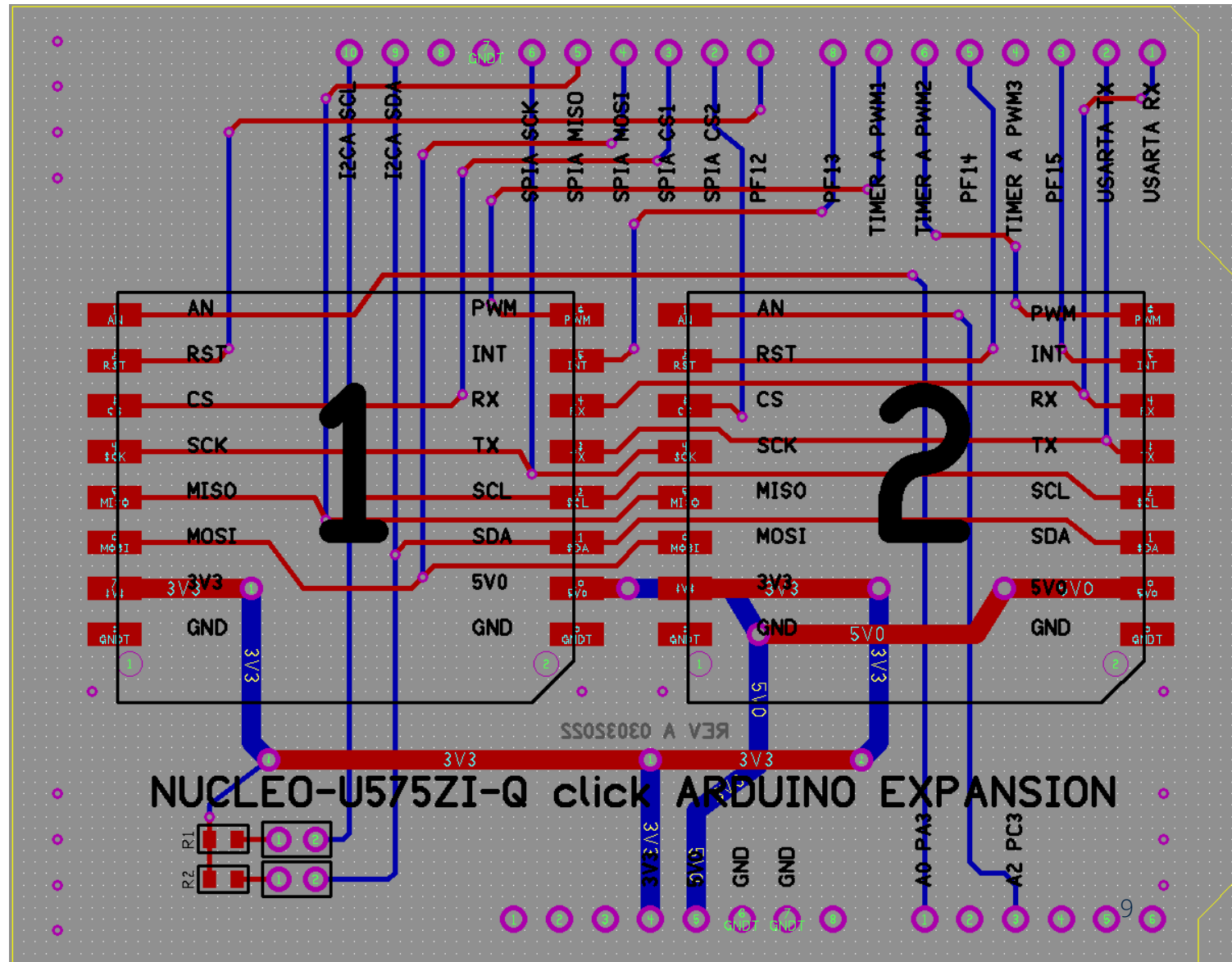
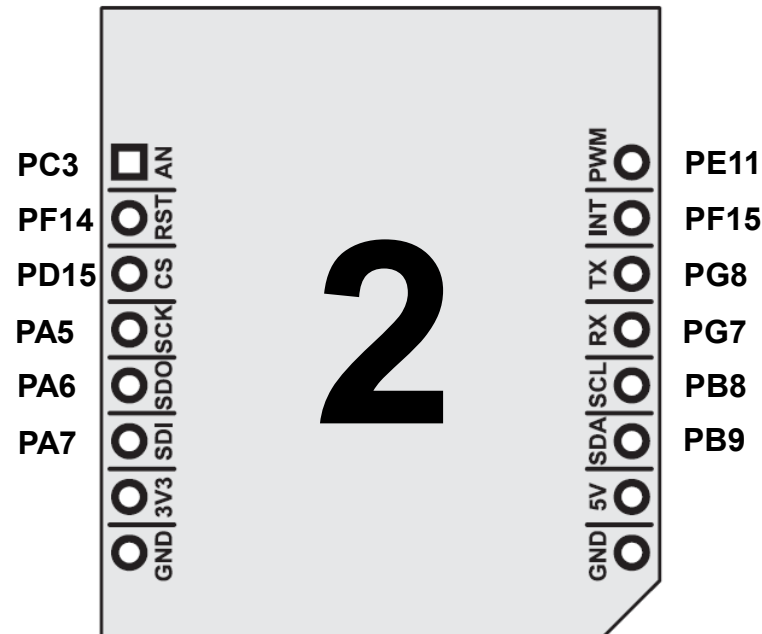
Socket 10x2 N144 ARD

PC2 shared between USB C and ZIO

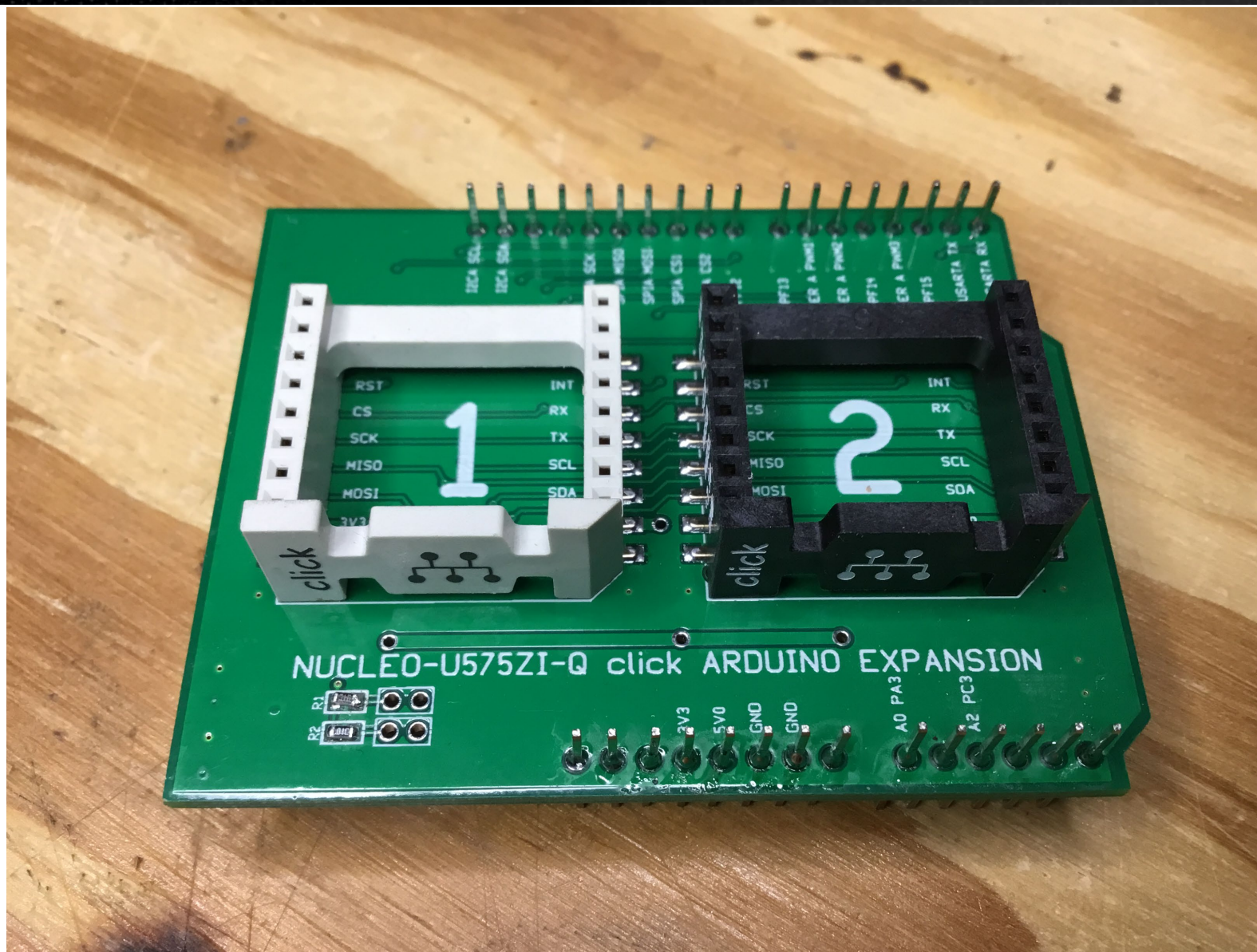


Socket 17x2 N144 ARD

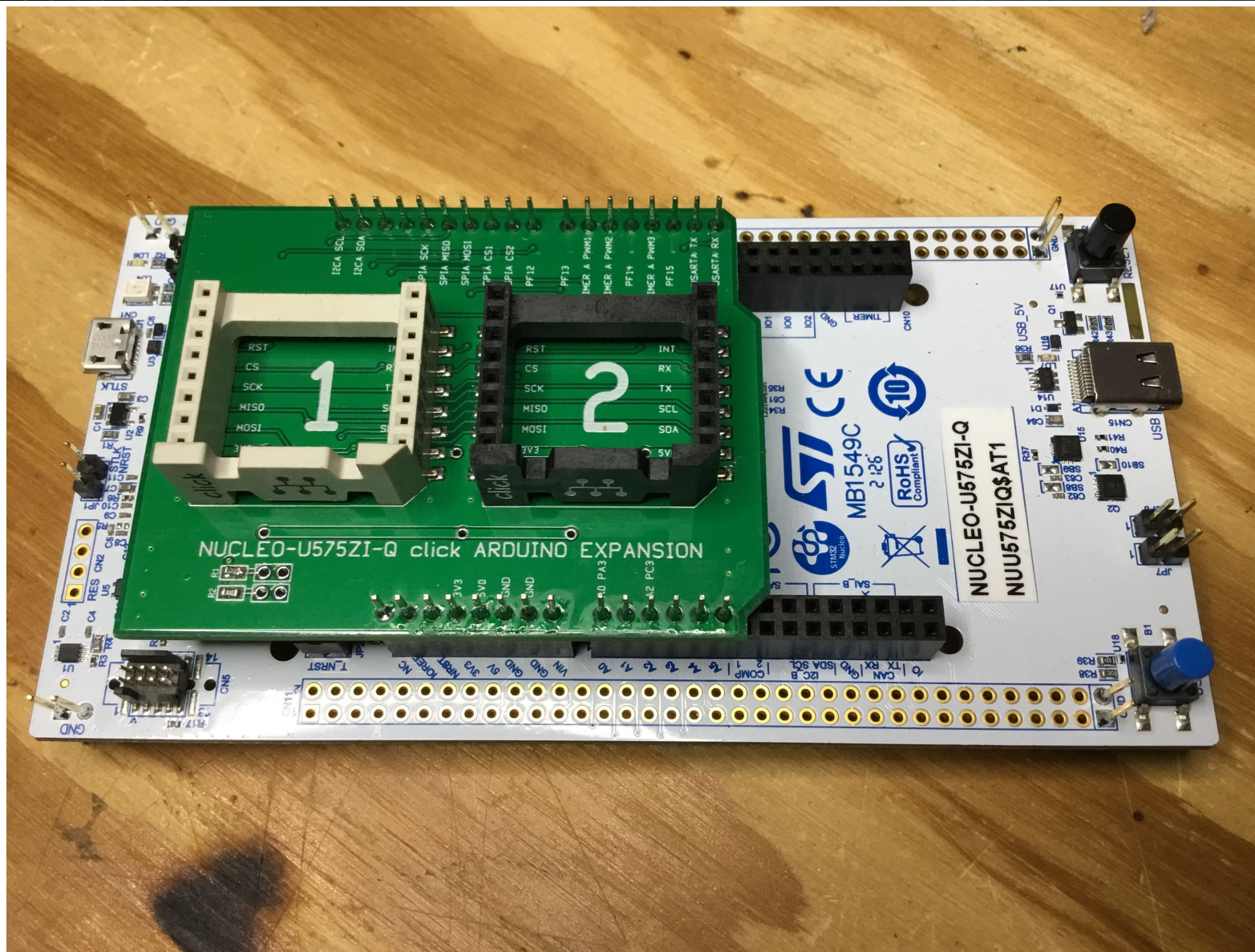
NUCLEO-U575ZI-Q click ARDUINO EXPANSION

NUCLEO-U575ZI-Q click ARDUINO EXPANSION

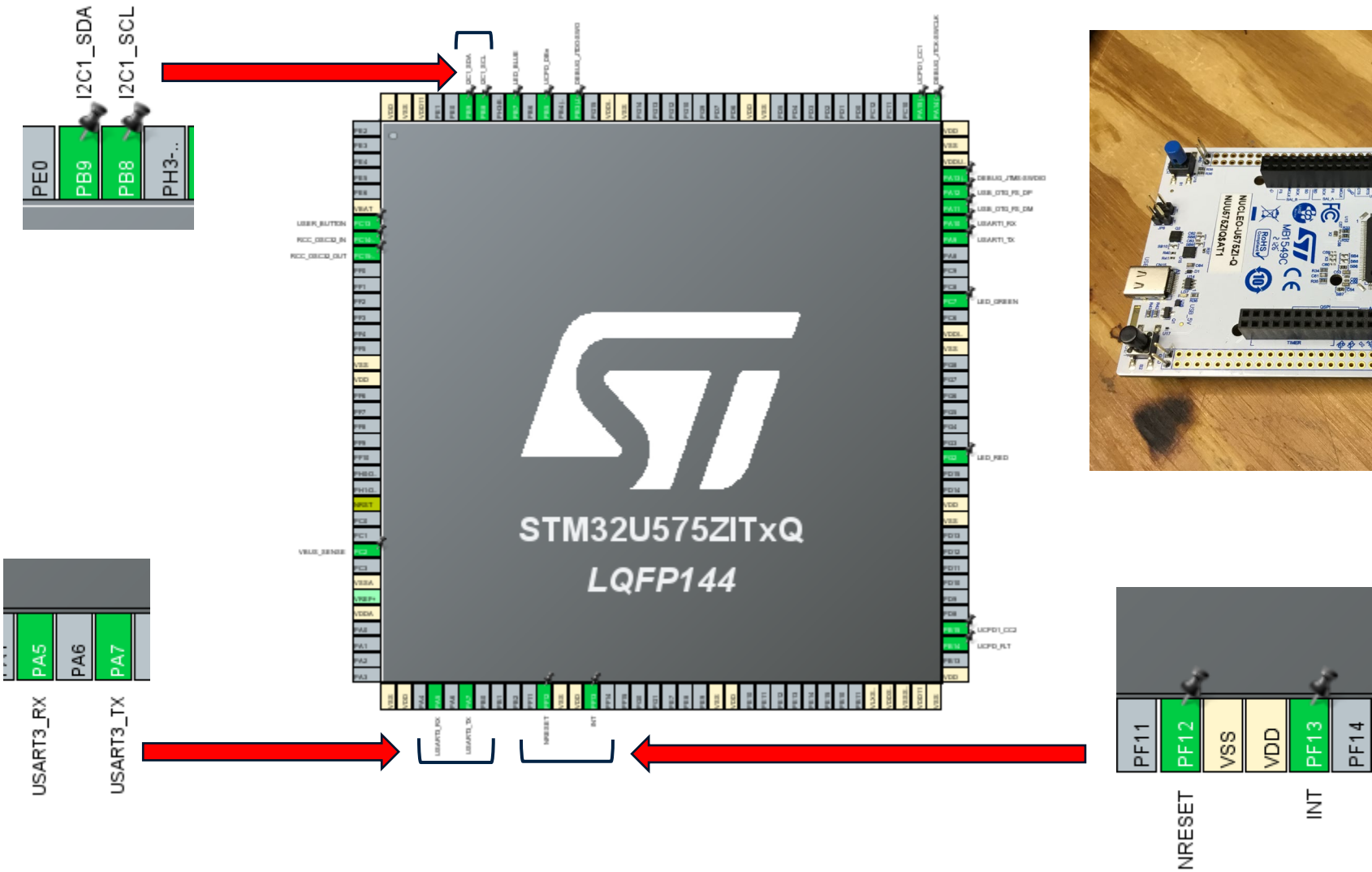
NUCLEO-U575ZI-Q click ARDUINO EXPANSION

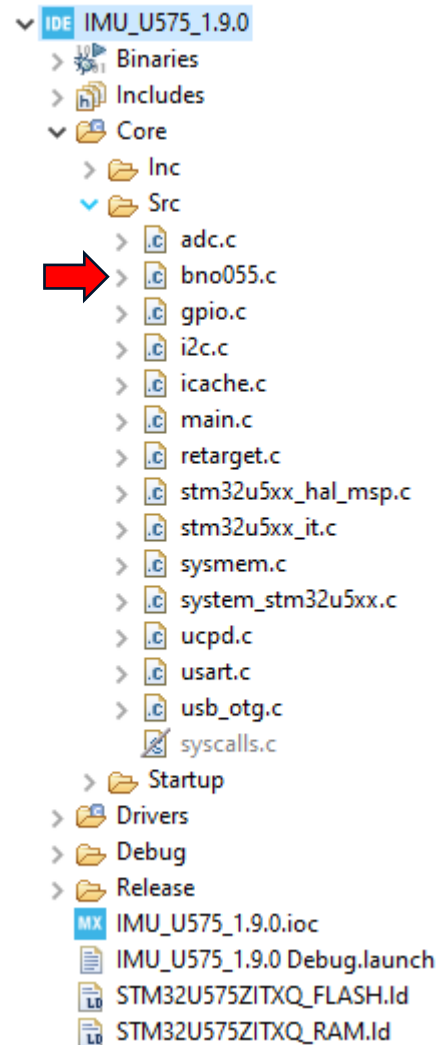


NUCLEO-U575ZI-Q click ARDUINO EXPANSION



IMU Driver: Hardware Configuration

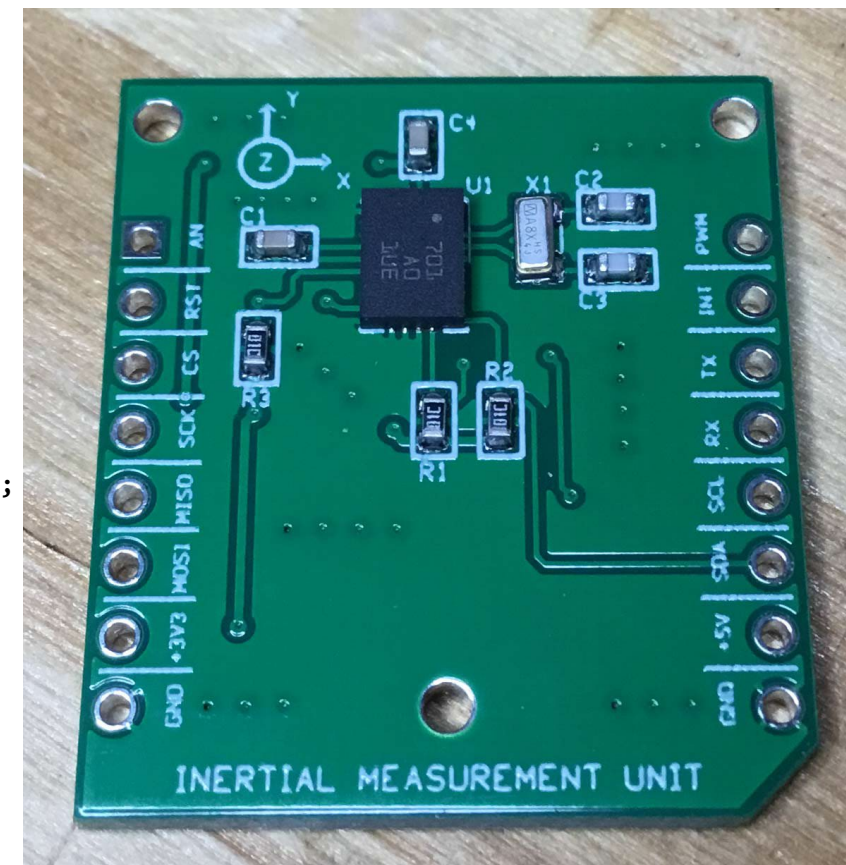


IMU Driver: bno055.c

```
void bno055_writeData(uint8_t reg, uint8_t data)
{
    uint8_t txdata[2] = {reg, data};

    HAL_I2C_Master_Transmit(&hi2c1, BNO055_I2C_ADDR << 1, txdata, sizeof(txdata), 10);
}

void bno055_readData(uint8_t reg, uint8_t *data, uint8_t len)
{
    HAL_I2C_Master_Transmit(&hi2c1, BNO055_I2C_ADDR << 1, &reg, 1, 100);
    HAL_I2C_Master_Receive(&hi2c1, BNO055_I2C_ADDR << 1, data, len, 100);
}
```



IMU Driver: i2c.c



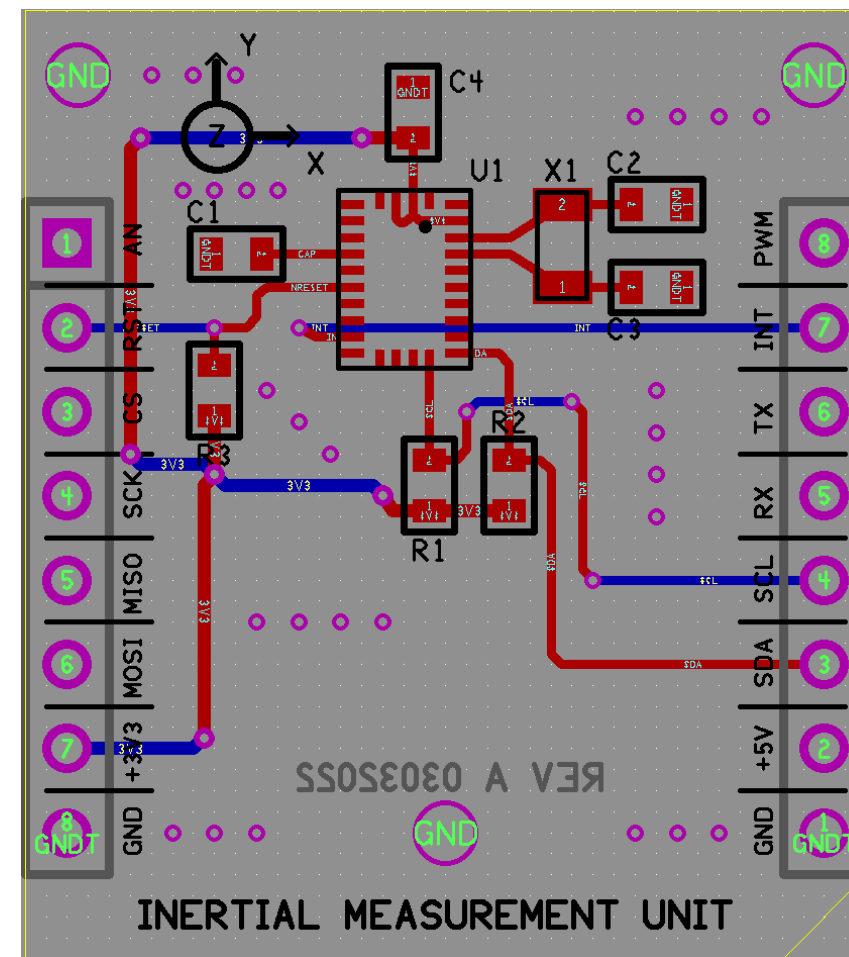
```
void MX_I2C1_Init(void)
{

    /* USER CODE BEGIN I2C1_Init 0 */

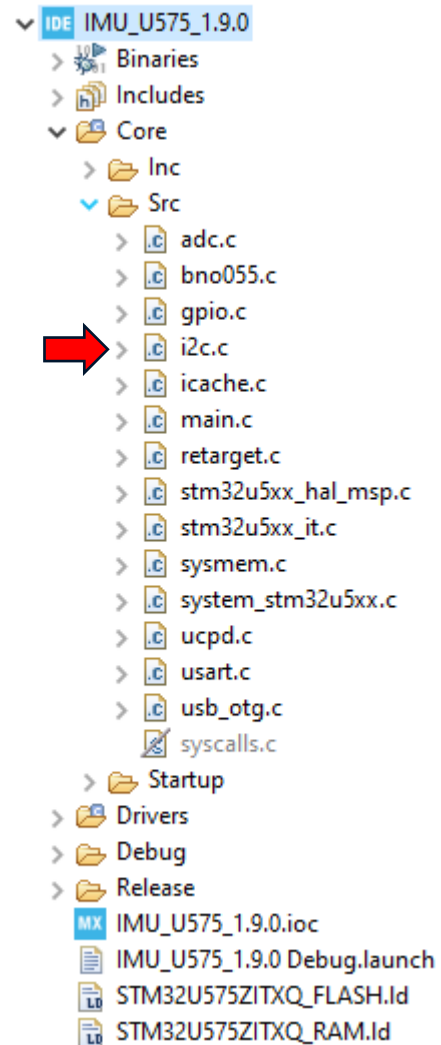
    /* USER CODE END I2C1_Init 0 */

    /* USER CODE BEGIN I2C1_Init 1 */

    /* USER CODE END I2C1_Init 1 */
    hi2c1.Instance = I2C1;
    hi2c1.Init.Timing = 0x30909DEC;
    hi2c1.Init.OwnAddress1 = 0;
    hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
    hi2c1.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
    hi2c1.Init.OwnAddress2 = 0;
    hi2c1.Init.OwnAddress2Masks = I2C_OA2_NOMASK;
    hi2c1.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
    hi2c1.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
    if (HAL_I2C_Init(&hi2c1) != HAL_OK)
    {
        Error_Handler();
    }
}
```



IMU Driver: i2c.c



```
void HAL_I2C_MspInit(I2C_HandleTypeDef* i2cHandle)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};
    RCC_PeriphCLKInitTypeDef PeriphClkInit = {0};
    if(i2cHandle->Instance==I2C1)
    {
        /* USER CODE BEGIN I2C1_MspInit 0 */

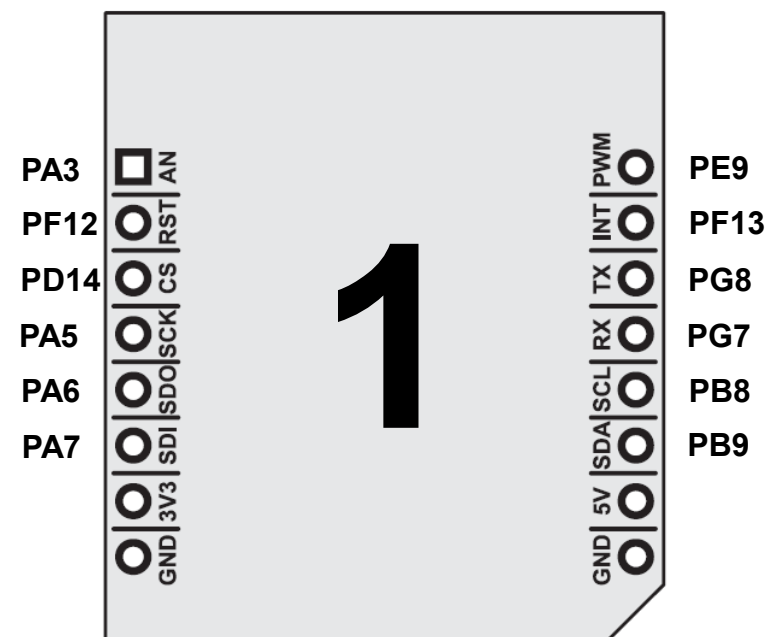
        /* USER CODE END I2C1_MspInit 0 */

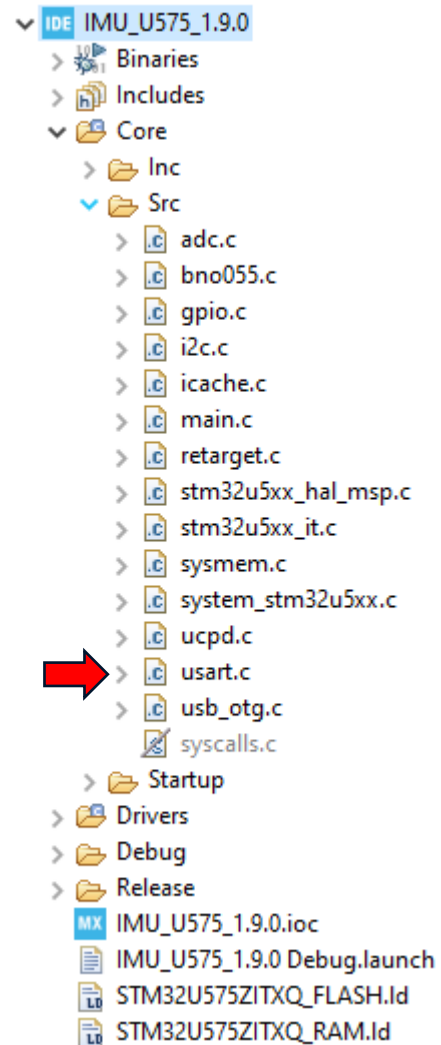
        /** Initializes the peripherals clock
        */
        PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_I2C1;
        PeriphClkInit.I2c1ClockSelection = RCC_I2C1CLKSOURCE_PCLK1;
        if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
        {
            Error_Handler();
        }

        __HAL_RCC_GPIOB_CLK_ENABLE();
        /**I2C1 GPIO Configuration
        PB8      -----> I2C1_SCL
        PB9      -----> I2C1_SDA
        */
        GPIO_InitStruct.Pin = GPIO_PIN_8|GPIO_PIN_9;
        GPIO_InitStruct.Mode = GPIO_MODE_AF_OD;
        GPIO_InitStruct.Pull = GPIO_NOPULL;
        GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
        GPIO_InitStruct.Alternate = GPIO_AF4_I2C1;
        HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

        /* I2C1 clock enable */
        __HAL_RCC_I2C1_CLK_ENABLE();
        /* USER CODE BEGIN I2C1_MspInit 1 */

        /* USER CODE END I2C1_MspInit 1 */
    }
}
```



IMU Driver: **usart.c**

```

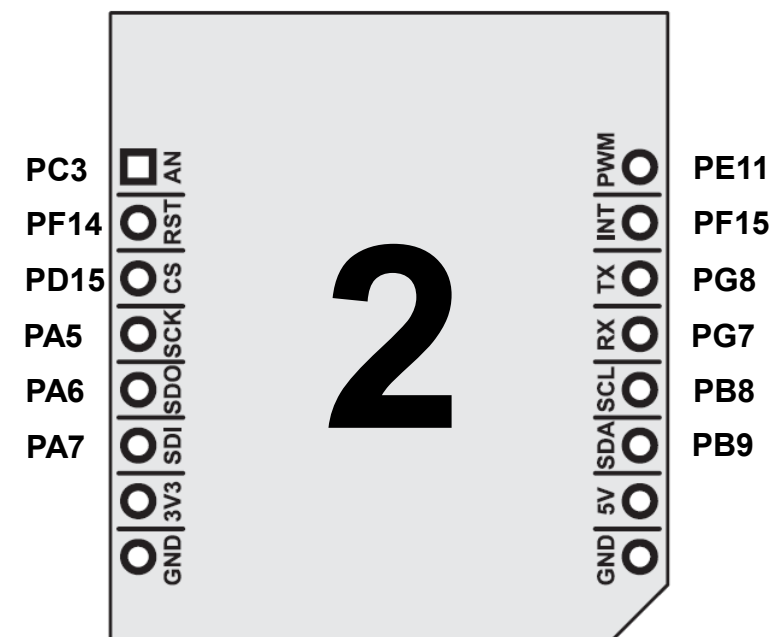
void MX_USART3_UART_Init(void)
{
    /* USER CODE BEGIN USART3_Init 0 */

    /* USER CODE END USART3_Init 0 */

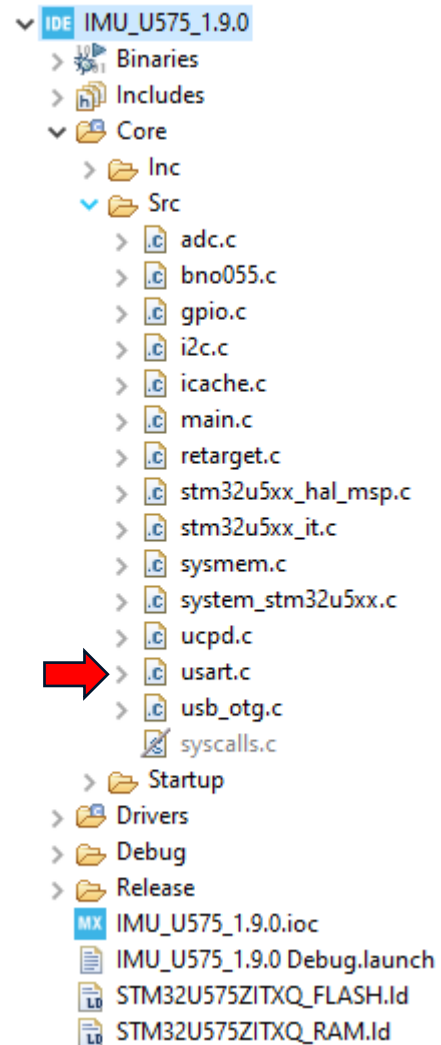
    /* USER CODE BEGIN USART3_Init 1 */

    /* USER CODE END USART3_Init 1 */
    huart3.Instance = USART3;
    huart3.Init.BaudRate = 115200;
    huart3.Init.WordLength = UART_WORDLENGTH_8B;
    huart3.Init.StopBits = UART_STOPBITS_1;
    huart3.Init.Parity = UART_PARITY_NONE;
    huart3.Init.Mode = UART_MODE_TX_RX;
    huart3.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart3.Init.OverSampling = UART_OVERSAMPLING_16;
    huart3.Init.OneBitSampling = UART_ONE_BIT_SAMPLE_DISABLE;
    huart3.Init.ClockPrescaler = UART_PRESCALER_DIV1;
    huart3.AdvancedInit.AdvFeatureInit = UART_ADVFEATURE_NO_INIT;
    if (HAL_UART_Init(&huart3) != HAL_OK)
    {
        Error_Handler();
    }
    if (HAL_UARTEx_SetTxFifoThreshold(&huart3, UART_TXFIFO_THRESHOLD_1_8) != HAL_OK)
    {
        Error_Handler();
    }
    if (HAL_UARTEx_SetRxFifoThreshold(&huart3, UART_RXFIFO_THRESHOLD_1_8) != HAL_OK)
    {
        Error_Handler();
    }
    if (HAL_UARTEx_DisableFifoMode(&huart3) != HAL_OK)
    {
        Error_Handler();
    }
}

```



IMU Driver: usart.c



```

else if(uartHandle->Instance==USART3)
{
    /* USER CODE BEGIN USART3_MspInit 0 */

    /* USER CODE END USART3_MspInit 0 */

    /** Initializes the peripherals clock
    */
    PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_USART3;
    PeriphClkInit.Usart3ClockSelection = RCC_USART3CLKSOURCE_PCLK1;
    if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
    {
        Error_Handler();
    }

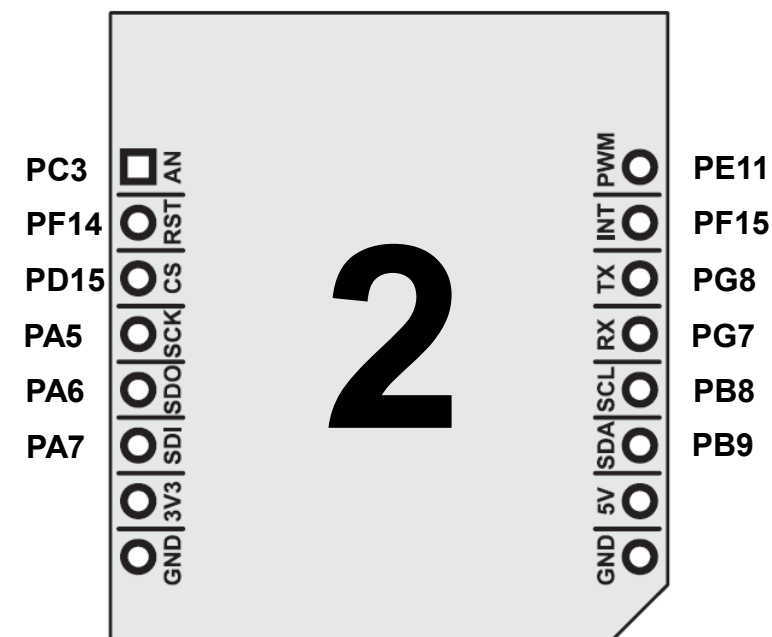
    /* USART3 clock enable */
    __HAL_RCC_USART3_CLK_ENABLE();

    __HAL_RCC_GPIOA_CLK_ENABLE();
    /**USART3 GPIO Configuration
    PA5      -----> USART3_RX
    PA7      -----> USART3_TX
    */
    GPIO_InitStruct.Pin = GPIO_PIN_5|GPIO_PIN_7;
    GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    GPIO_InitStruct.Alternate = GPIO_AF7_USART3;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

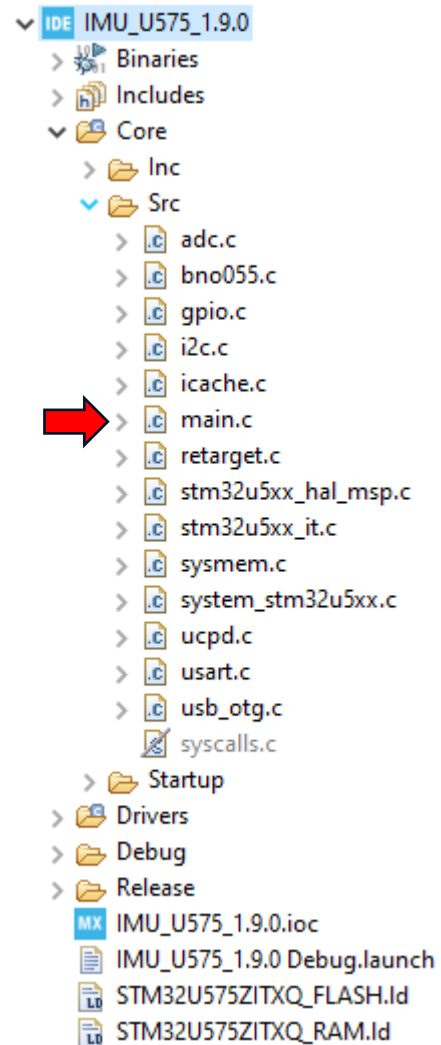
    /* USART3 interrupt Init */
    HAL_NVIC_SetPriority(USART3_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(USART3_IRQn);
    /* USER CODE BEGIN USART3_MspInit 1 */

    /* USER CODE END USART3_MspInit 1 */
}
}

```

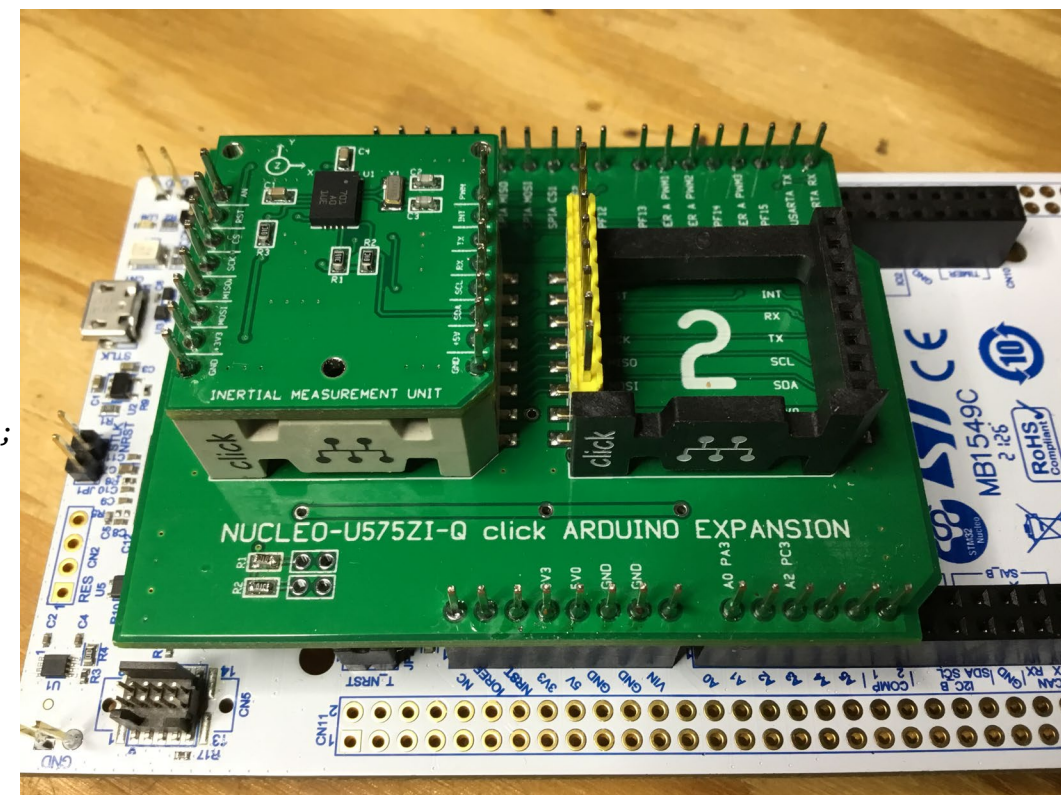


IMU Driver: main.c

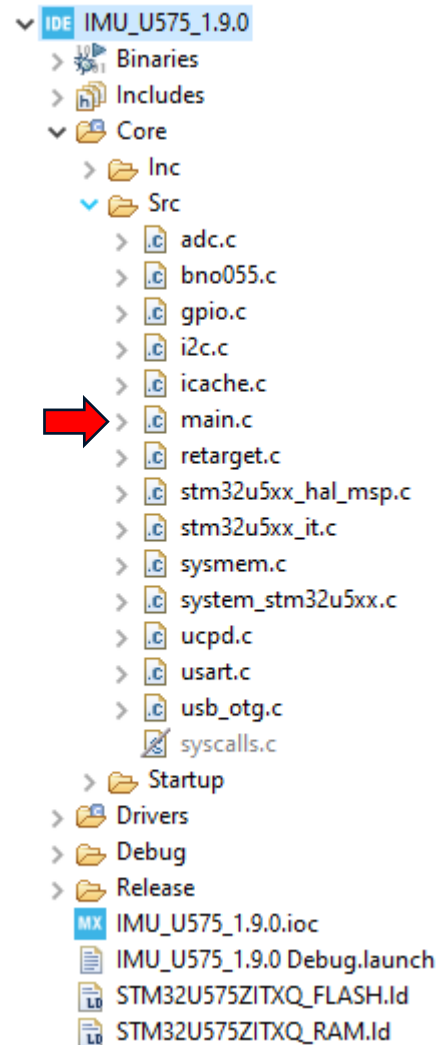


```
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    SystemPower_Config();
    MX_GPIO_Init();
    MX_ADC1_Init();
    MX_UCPD1_Init();
    MX_USART1_UART_Init();
    MX_USB_OTG_FS_PCD_Init();
    MX_I2C1_Init();
    MX_ICACHE_Init();
    MX_USART3_UART_Init();
    RetargetInit(&huart3);
    HAL_GPIO_WritePin(NRESET_GPIO_Port, NRESET_Pin, GPIO_PIN_SET);
    bno055_assignI2C(&hi2c1);
    bno055_setup();
    bno055_setOperationModeND0F();
    bno055Calibrate();

    while (1)
    {
        v = bno055_getVectorEuler();
        HAL_Delay(1000);
        printf("x = %.2f y = %.2f z = %.2f\r\n",v.x,v.y,v.z);
    }
}
```



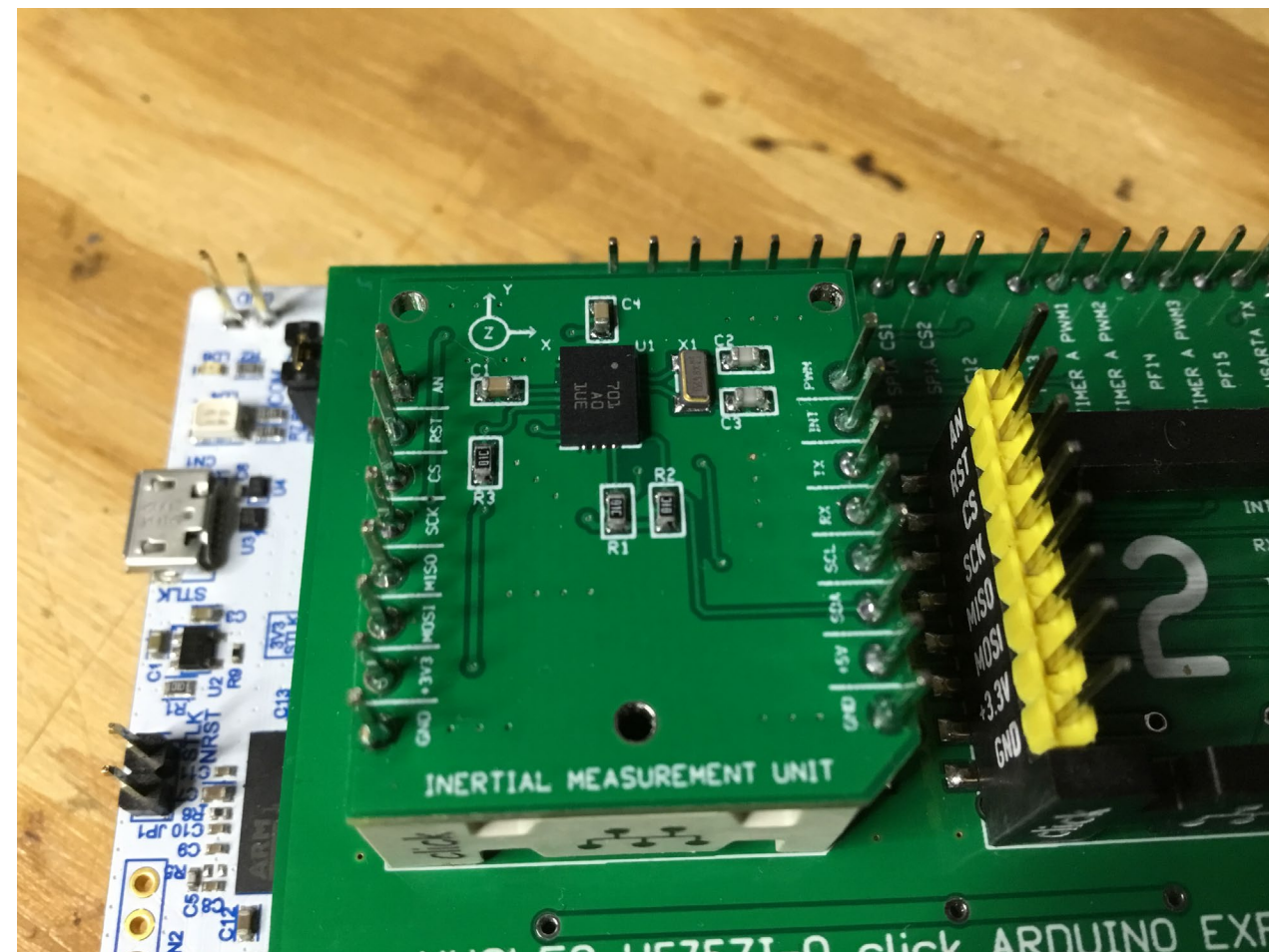
IMU Driver: main.c



```
void bno055Calibrate(void)
{
    printf("Calibrate the IMU\r\n");
    do{
        calState = bno055_getCalibrationState();

        if(sysCalibrated == 0x00)
        {
            if(calState.prevsys != calState.sys)
            {
                calState.prevsys = calState.sys;
                if(calState.sys == 0x03)
                {
                    sysCalibrated = 0x01;
                    printf("System Calibrated\r\n");
                }
            }
        }

        if(gyroCalibrated == 0x00)
        {
            if(calState.prevgyro != calState.gyro)
            {
                if(magCalibrated == 0x00)
                {
                    if(calState.prevmag != calState.mag)
                    {
                        if(accelCalibrated == 0x00)
                        {
                            if(calState.prevaccel != calState.accel)
                            {
                                while(sysCalibrated == 0 || gyroCalibrated == 0 || magCalibrated == 0 || accelCalibrated == 0);
                            }
                        }
                    }
                }
            }
        }
    } while(1);
}
```



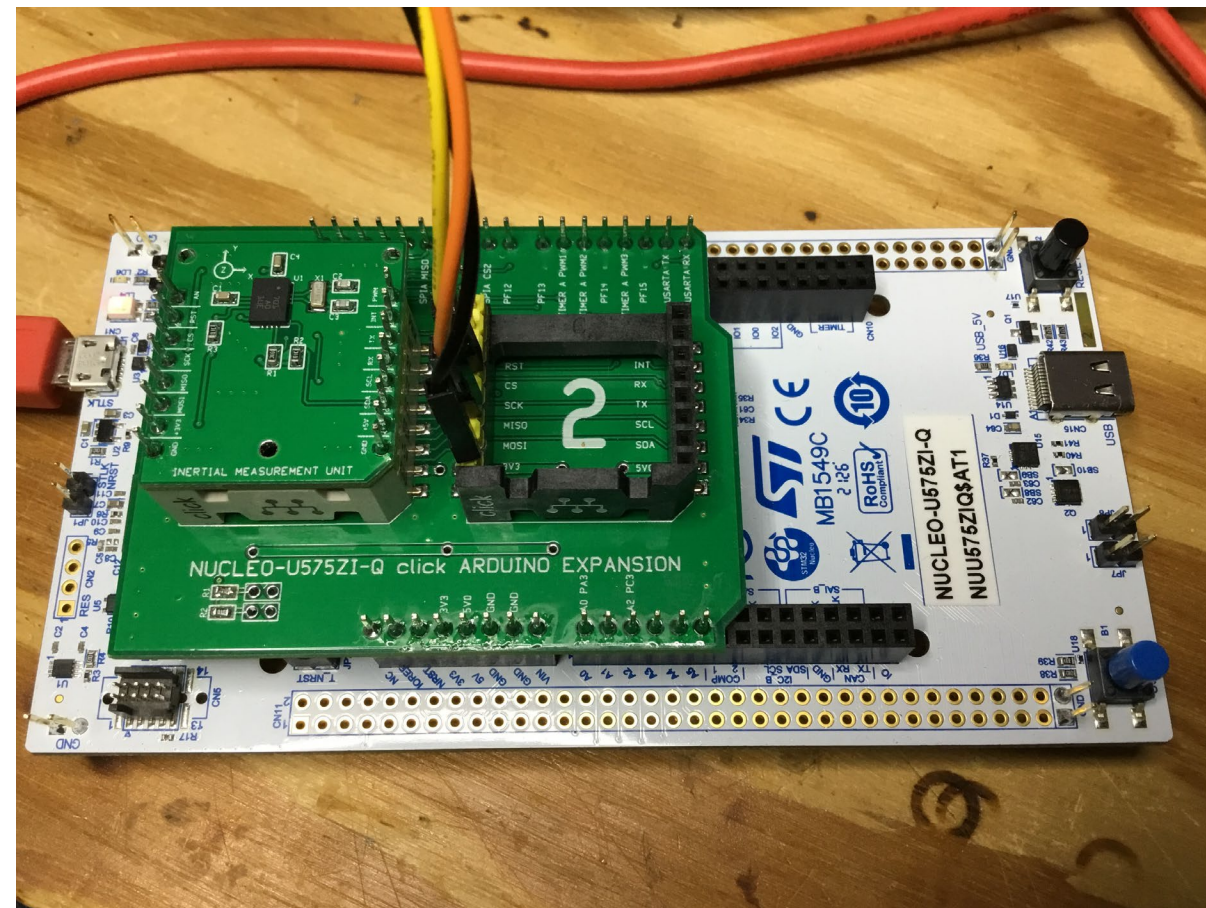
IMU Driver: retarget.c



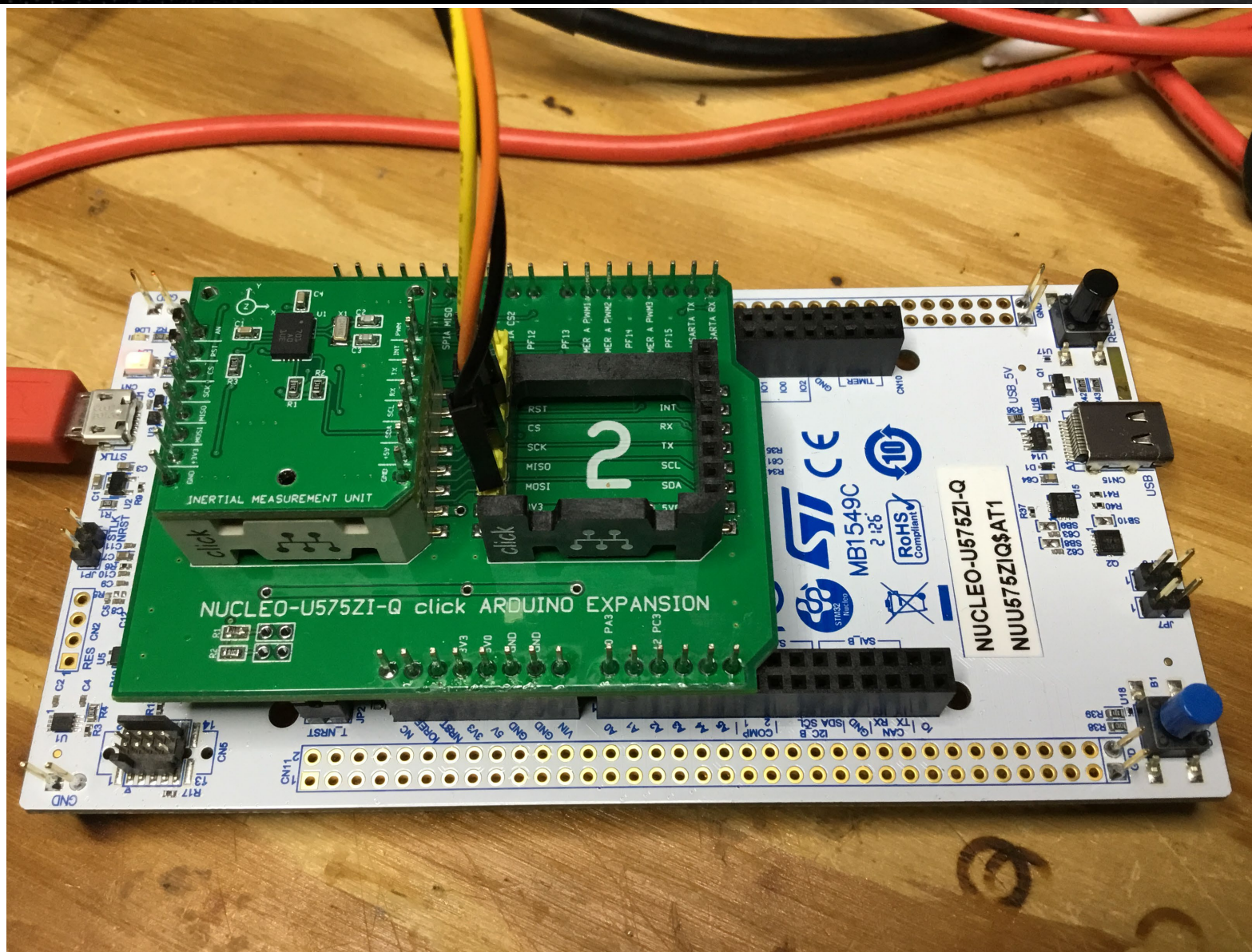
PC3		AN
PF14		RST
PD15		CS
PA5		SCK
PA6		SDO
PA7		SDI
		3V3
		GND

PC3
PF14
PD15
PA5
PA6
PA7

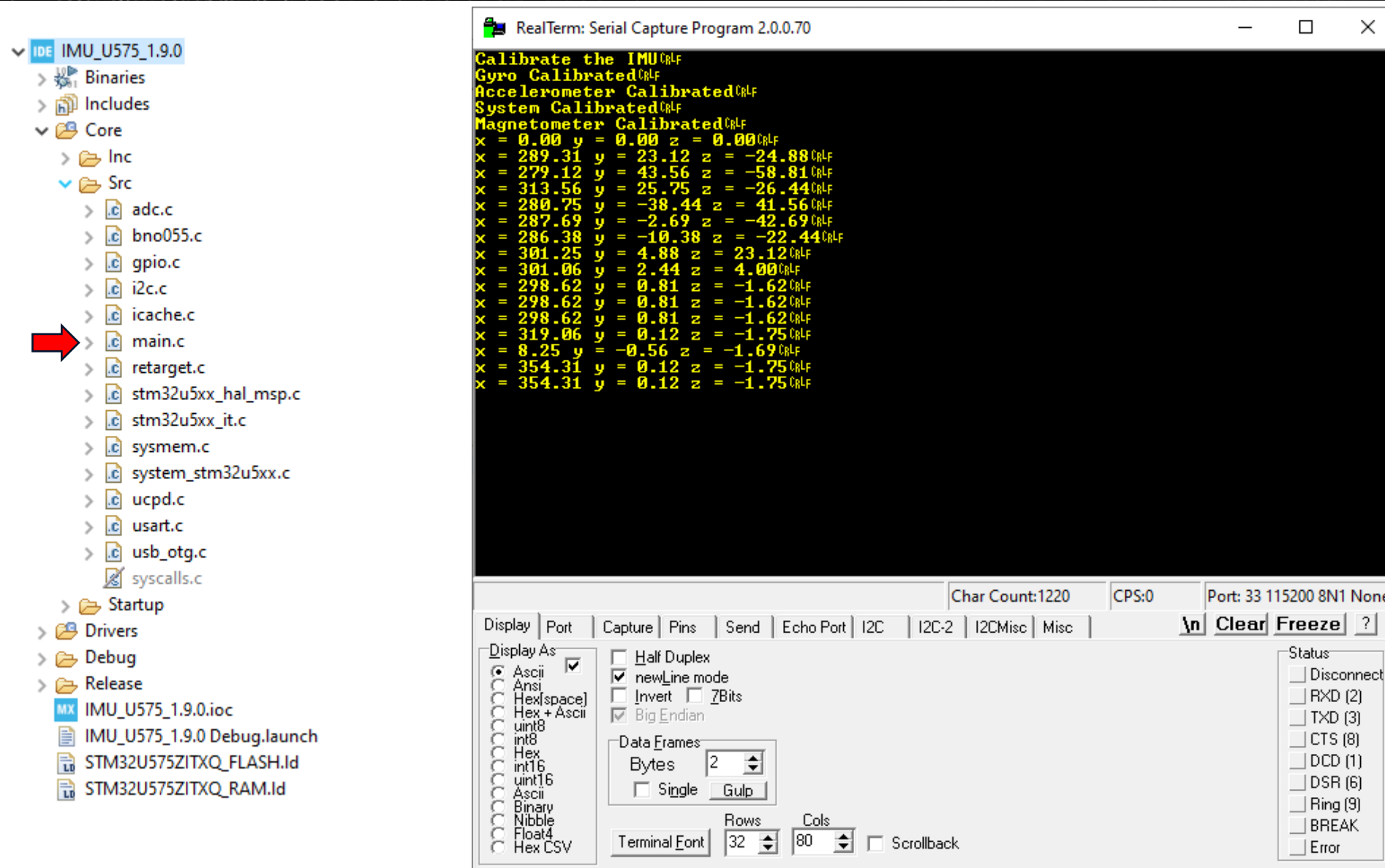
PE11
PF15
PG8
PG7
PB8
PB9



IMU Driver: Fire It Up!!



IMU Driver: Fire It Up!!



The screenshot shows an IDE window on the left with a project tree for **IMU_U575_1.9.0**. A red arrow points to the **main.c** file under the **Src** folder. The main components of the project are:

- Binaries
- Includes
- Core
 - Inc
 - Src
 - adc.c
 - bn055.c
 - gpio.c
 - i2c.c
 - icache.c
 - main.c
 - retarget.c
 - stm32u5xx_hal_msp.c
 - stm32u5xx_it.c
 - system.c
 - system_stm32u5xx.c
 - ucpd.c
 - usart.c
 - usb_otg.c
 - syscalls.c
- Startup
- Drivers
- Debug
- Release
- MX
 - IMU_U575_1.9.0.ioc
 - IMU_U575_1.9.0 Debug.launch
 - STM32U575ZITXQ_FLASH.ld
 - STM32U575ZITXQ_RAM.ld

The RealTerm: Serial Capture Program 2.0.0.70 window on the right displays the following output:

```
Calibrate the IMU CRLF
Gyro CalibratedCRLF
Accelerometer CalibratedCRLF
System CalibratedCRLF
Magnetometer CalibratedCRLF
x = 0.00 y = 0.00 z = 0.00CRLF
x = 289.31 y = 23.12 z = -24.88CRLF
x = 279.12 y = 43.56 z = -58.81CRLF
x = 313.56 y = 25.75 z = -26.44CRLF
x = 280.75 y = -38.44 z = 41.56CRLF
x = 287.69 y = -2.69 z = -42.69CRLF
x = 286.38 y = -10.38 z = -22.44CRLF
x = 301.25 y = 4.88 z = 23.12CRLF
x = 301.06 y = 2.44 z = 4.00CRLF
x = 298.62 y = 0.81 z = -1.62CRLF
x = 298.62 y = 0.81 z = -1.62CRLF
x = 298.62 y = 0.81 z = -1.62CRLF
x = 319.06 y = 0.12 z = -1.75CRLF
x = 8.25 y = -0.56 z = -1.69CRLF
x = 354.31 y = 0.12 z = -1.75CRLF
x = 354.31 y = 0.12 z = -1.75CRLF
```

The RealTerm window also shows a status bar with the following information:

- Char Count: 1220
- CPS: 0
- Port: 33 115200 8N1 None
- Buttons: **Clear**, **Freeze**, **?**
- Display As: ☒ Ascii, ☐ Ansi, ☐ Hex(space), ☐ Hex + Ascii, ☐ uint8, ☐ int8, ☐ Hex, ☐ int16, ☐ uint16, ☐ Ascii, ☐ Binary, ☐ Nibble, ☐ Float4, ☐ Hex CSV
- Half Duplex: ☐ ☒ newLine mode
- Invert: ☐ ☐ ZBits
- Big Endian: ☒ ☐ Little Endian
- Data Frames: Bytes: 2, ☐ Single, ☒ Gulp
- Rows: 32, Cols: 80, ☐ Scrollback
- Status: ☐ Disconnect, ☐ RXD (2), ☐ TXD (3), ☐ CTS (8), ☐ DCD (1), ☐ DSR (6), ☐ Ring (9), ☐ BREAK, ☐ Error

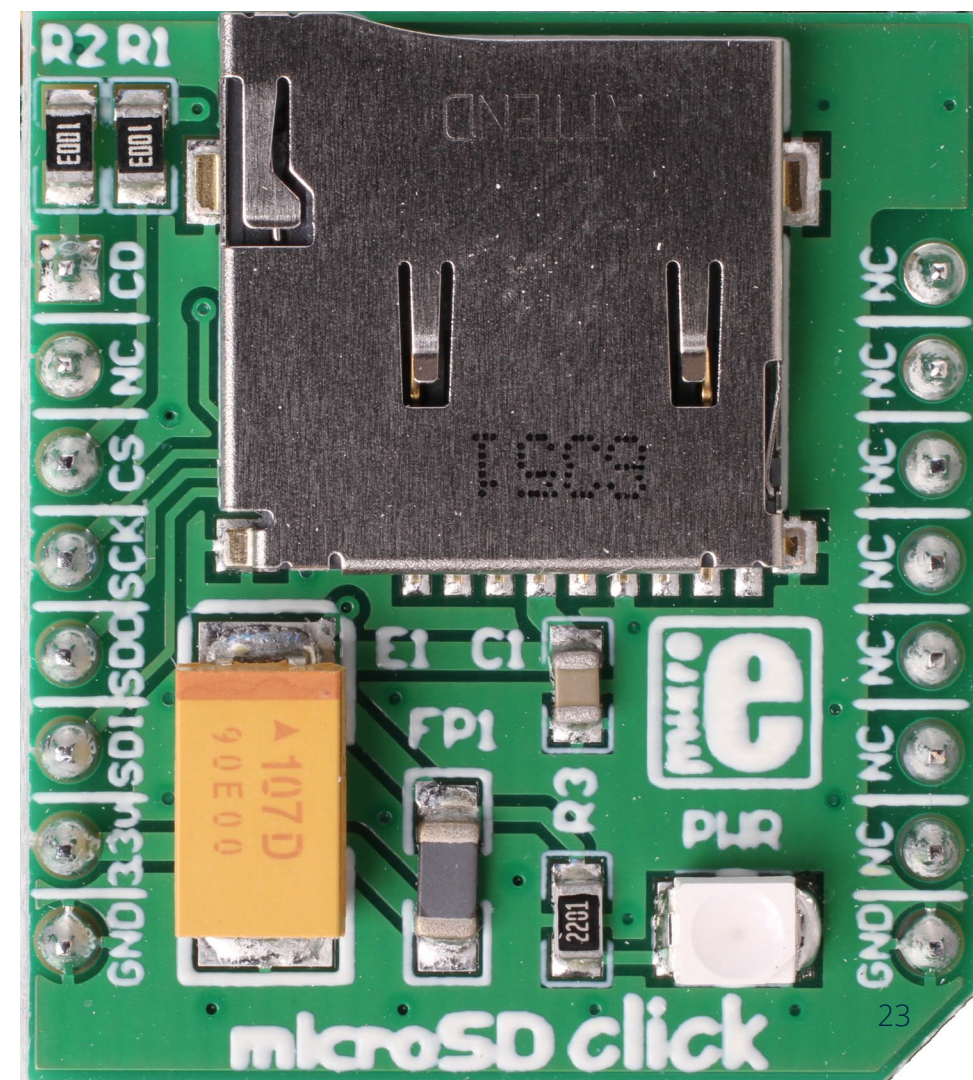
Thank you for attending!!!

Please consider the resources below:

- **Today's Project Download Package**
- **BN0055 Data Sheet**
- **NUCLEO-U575ZI-Q User Manual / Data Sheet**

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

MORE TO COME..





DesignNews

Thank You

Sponsored by

