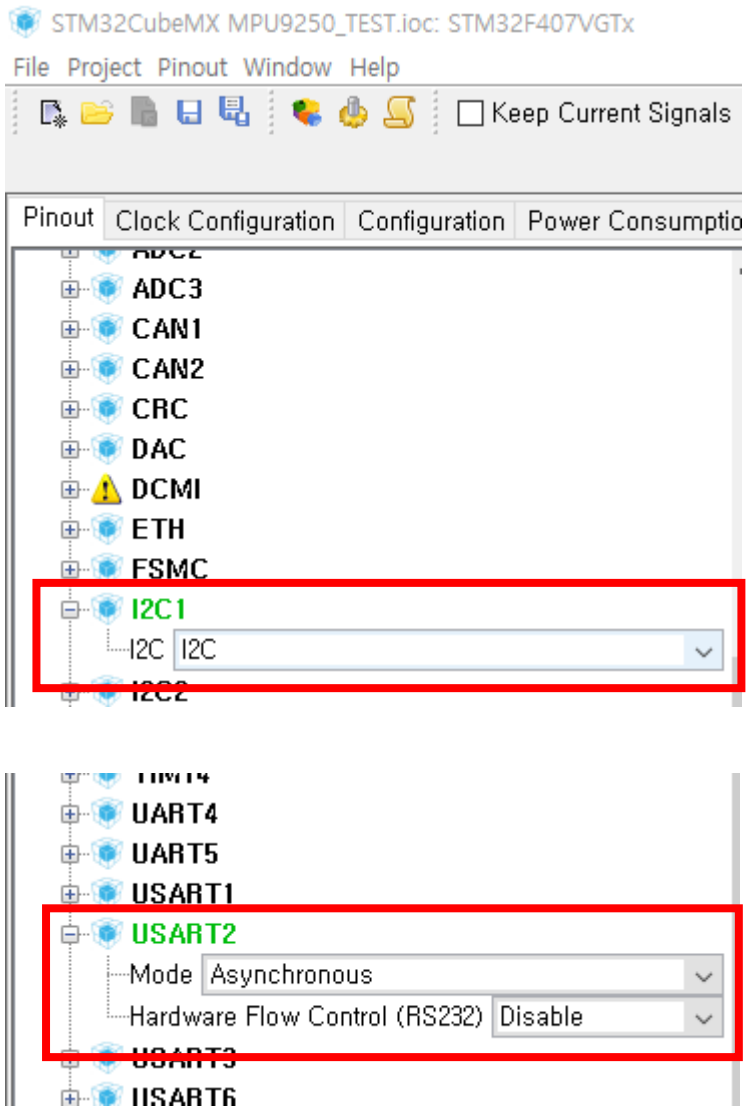
TI DSP, MCU, Xilinx Zynq FPGA
프로그래밍 전문가 과정

Control MPU9250 with I2C

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

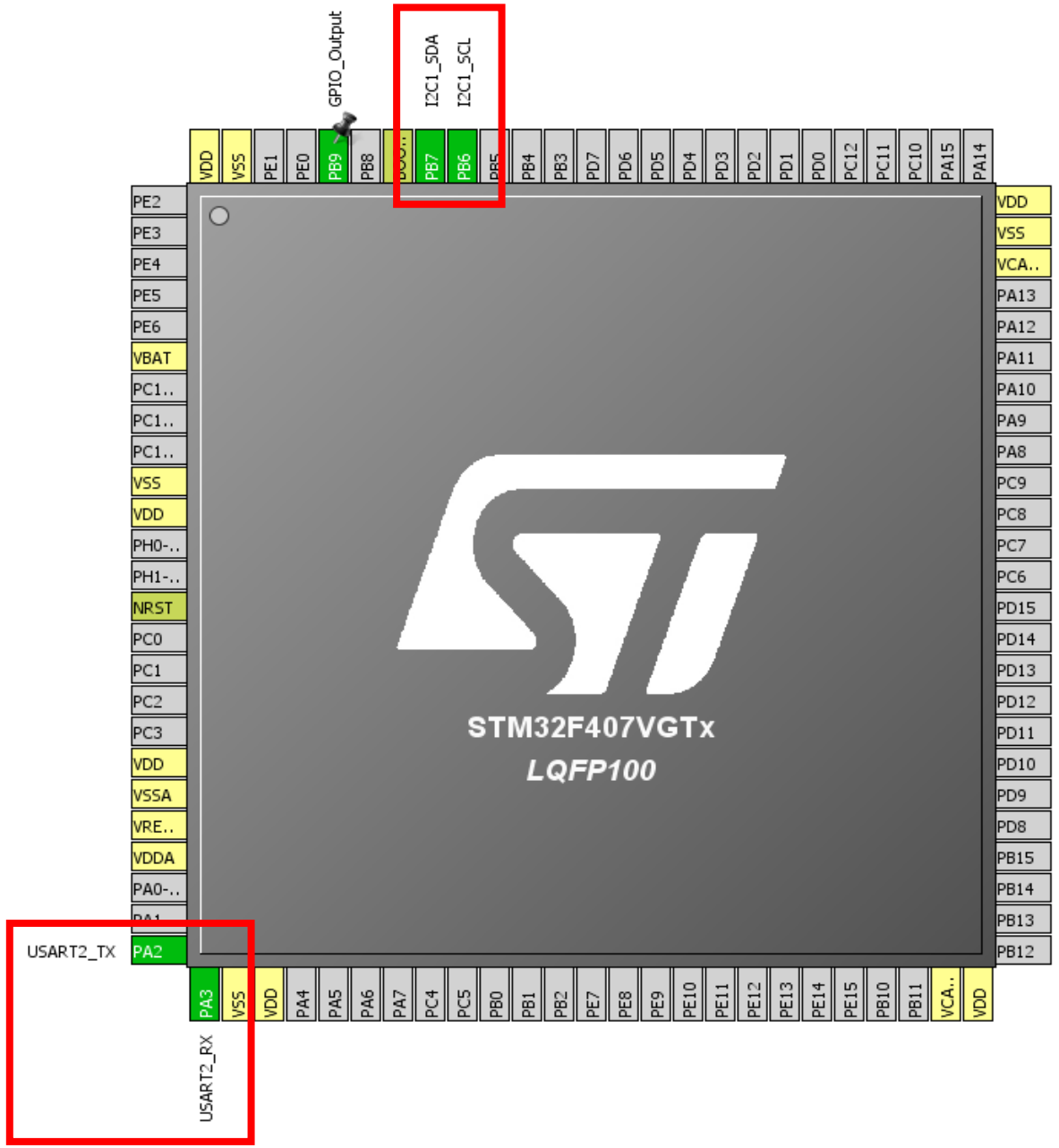
학생 – 안상재
sangjae2015@naver.com

1. CubeMX 설정

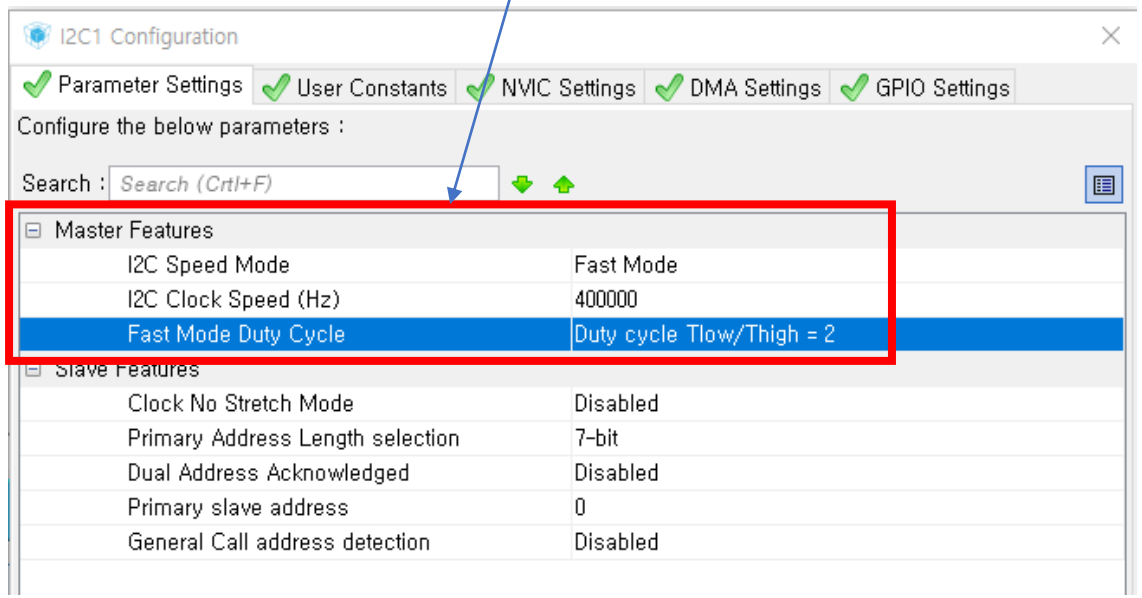


I2C1 enable

USART2 enable
=> 센서 데이터 값 출력



MCU를 Master로 설정



2. 소스 코드

WHO_AM_I 레지스터를 읽으면 0x71이 나와야 함!

```
whoami = readByte(MPU9250_ADDRESS, WHO_AM_I_MPU9250);
sprintf(str, "str = 0x%x\r\n", whoami);
HAL_UART_Transmit(&huart2, str, 30, 10);
```

WHO_AM_I에서 0x71이 나오면 정상

MPU9250의 모든 레지스터를 디폴트 값으로 셋팅

레지스터 calibrate

```
if(whoami == 0x71)
{
    resetMPU9250();
    calibrateMPU9250(gyroBias, accelBias); // Calibrate gyro and accelerometers, load biases in bias registers

    sprintf(str, "x gyro bias = %f\n\r", gyroBias[0]);
    HAL_UART_Transmit(&huart2, str, 50, 10);
    sprintf(str, "y gyro bias = %f\n\r", gyroBias[1]);
    HAL_UART_Transmit(&huart2, str, 50, 10);
    sprintf(str, "z gyro bias = %f\n\r", gyroBias[2]);
    HAL_UART_Transmit(&huart2, str, 50, 10);

    HAL_UART_Transmit(&huart2, "\r\n", 2, 10);
    sprintf(str, "x accel bias = %f\n\r", accelBias[0]);
    HAL_UART_Transmit(&huart2, str, 50, 10);
    sprintf(str, "y accel bias = %f\n\r", accelBias[1]);
    HAL_UART_Transmit(&huart2, str, 50, 10);
    sprintf(str, "z accel bias = %f\n\r", accelBias[2]);
    HAL_UART_Transmit(&huart2, str, 50, 10);

    HAL_UART_Transmit(&huart2, "\r\n\n", 5, 10);
```

MPU9250 레지스터 셋팅

```
initMPU9250();
```

```
initAK8963(magCalibration);
```

지자계 레지스터 셋팅

```
if(Mscale == 0)
{
    sprintf(str, "Magnetometer resolution = 14  bits\n\r");
    HAL_UART_Transmit(&huart2, str, 50, 10);
}
if(Mscale == 1)
{
    sprintf(str, "Magnetometer resolution = 16  bits\n\r");
    HAL_UART_Transmit(&huart2, str, 50, 10);
}
if(Mmode == 2)
{
    sprintf(str, "Magnetometer ODR = 8 Hz\n\r");
    HAL_UART_Transmit(&huart2, str, 50, 10);
}
if(Mmode == 6)
{
    sprintf(str, "Magnetometer ODR = 100 Hz\n\r");
    HAL_UART_Transmit(&huart2, str, 50, 10);
}
```

민감도 설정

```
getAres();    // get accelerometer sensitivity
getGres();    // get gyro sensitivity
getMres();    // get magnetometer sensitivity

sprintf(str, "Accelerometer sensitivity is %f LSB/g \r\n", 1.0f/aRes);
HAL_UART_Transmit(&huart2, str, 50, 10);

sprintf(str, "Gyroscope sensitivity is %f LSB/deg/s \n\r", 1.0f/gRes);
HAL_UART_Transmit(&huart2, str, 50, 10);

sprintf(str, "Magnetometer sensitivity is %f LSB/G \n\r", 1.0f/mRes);
HAL_UART_Transmit(&huart2, str, 50, 10);

magbias[0] = +470.; // User environmental x-axis correction in milliGauss, should be automatically calculated
magbias[1] = +120.; // User environmental x-axis correction in milliGauss
magbias[2] = +125.; // User environmental x-axis correction in milliGauss
```

지자계 값 보정

raw 데이터 read

```
  while (1)
  {
/* USER CODE END WHILE */
/* USER CODE BEGIN 3 */
      if(readByte(MPU9250_ADDRESS, INT_STATUS) & 0x01) // On interrupt, check if data ready interrupt
      {
        readAccelData(accelCount);  // Read the x/y/z adc values
        // Now we'll calculate the accleration value into actual g's
        ax = (float)accelCount[0]*aRes - accelBias[0];  // get actual g value, this depends on scale being set
        ay = (float)accelCount[1]*aRes - accelBias[1];
        az = (float)accelCount[2]*aRes - accelBias[2];

        readGyroData(gyroCount);  // Read the x/y/z adc values
        // Calculate the gyro value into actual degrees per second
        gx = (float)gyroCount[0]*gRes - gyroBias[0];  // get actual gyro value, this depends on scale being set
        gy = (float)gyroCount[1]*gRes - gyroBias[1];
        gz = (float)gyroCount[2]*gRes - gyroBias[2];

        readMagData(magCount);  // Read the x/y/z adc values
        // Calculate the magnetometer values in milliGauss
        // Include factory calibration per data sheet and user environmental corrections
        mx = (float)magCount[0]*mRes*magCalibration[0] - magbias[0];  // get actual magnetometer value, this depends on scale being set
        my = (float)magCount[1]*mRes*magCalibration[1] - magbias[1];
        mz = (float)magCount[2]*mRes*magCalibration[2] - magbias[2];
      }

      MadgwickQuaternionUpdate(ax, ay, az, gx*PI/180.0f, gy*PI/180.0f, gz*PI/180.0f,  my,  mx, mz);
```

raw 데이터 준비 완료

데이터 정규화 과정

온도 데이터 read

```
tempCount = readTempData();  // Read the adc values
temperature = ((float) tempCount) / 333.87f + 21.0f; // Temperature in degrees Centigrade

 sprintf(str1, "ax = %f   ay = %f   az = %f  mg", 1000*ax, 1000*ay, 1000*az);
 HAL_UART_Transmit(&huart2, str1, sizeof(str1), 10);
 HAL_UART_Transmit(&huart2, "\r\n", 2, 10);

 sprintf(str2, "gx = %f   gy = %f   gz = %f  deg/s", gx, gy, gz);
 HAL_UART_Transmit(&huart2, str2, sizeof(str2), 10);
 HAL_UART_Transmit(&huart2, "\r\n", 2, 10);

 sprintf(str3, "mx = %f  my = %f  mz = %f mG", mx, my, mz);
 HAL_UART_Transmit(&huart2, str3, sizeof(str3), 10);
 HAL_UART_Transmit(&huart2, "\r\n", 2, 10);

sprintf(str4, "temperature = %f  C\r\n\n", temperature);
HAL_UART_Transmit(&huart2, str4, sizeof(str4), 10);

HAL_Delay(1000);
```

3. 결과 화면