MPU6050+Madgwich4 のプログラム

2022.06.12 M. shimojo

MPU6050 による姿勢の検出を行うプログラムメモです。少しでも皆様の参考になれば幸いです。

- STM32 NUCLEO F446RE を用いて、MPU6050 から加速度と角速度の値を検出する。
- 検出データは TeraTerm を使いデスクトップ PC へ転送する。
- 描画プログラムは、Processing を用いた。
- 加速度と角速度から姿勢の検出には Madgwick Filter を用いた。
- STM32 NUCLEO F446RE のプログラム開発には、STM32CubeIDE を利用した。
- MPU6050 の制御プログラムは数多くある。ただし、STM32 での実装は多くない。簡単に利用するなら Arduino 方式を用いた方が簡便で良い。
  - 私はジャンクボックスの CPU から速度の速いものを選び出した結果 F446RE となった。結果として、STM32CubeIDE を利用したことで、Arduino 方式より細かな制御ができるので良かった。(Arduino は隔靴掻痒←個人の感想)
- 以下に既存のプログラムから改変した点を記す。

✓ MPU6050 の姿勢の検出（サンプリング）周期の計測プログラムを入れた。
  - サンプリング周期は、早すぎると MPU が対応できないため、内部 Delay を入れて調節した

✓ Madgwick Filter へのサンプリング周期の受渡（Madgwick.c の一部改変含む）プログラムを入れた。
  - サンプリング周期の値は、実際の姿勢の検出周期＋Delay であり、その値を受け渡した（←少し手抜きをしている）
  - サンプリング周期の値は Madgwick Filter の演算処理で大変重要である

✓ 青ボタンによる割込み処理で Gyro のオフセットの補正を行うプログラムを入れた。
  - Gyro からの角速度の積分値である角度は、安価な Gyro は時間変動（ドリフト）が大きい。例えば MPU6050 では、時間あたり 2 桁程の角度(Degree)変化した。このため青ボタンを押すと、割込み処理により Gyro のオフセットの補正を行うプログラムを入れた

✓ 描画ソフト processing へのデータ出力を行うプログラムを入れた。

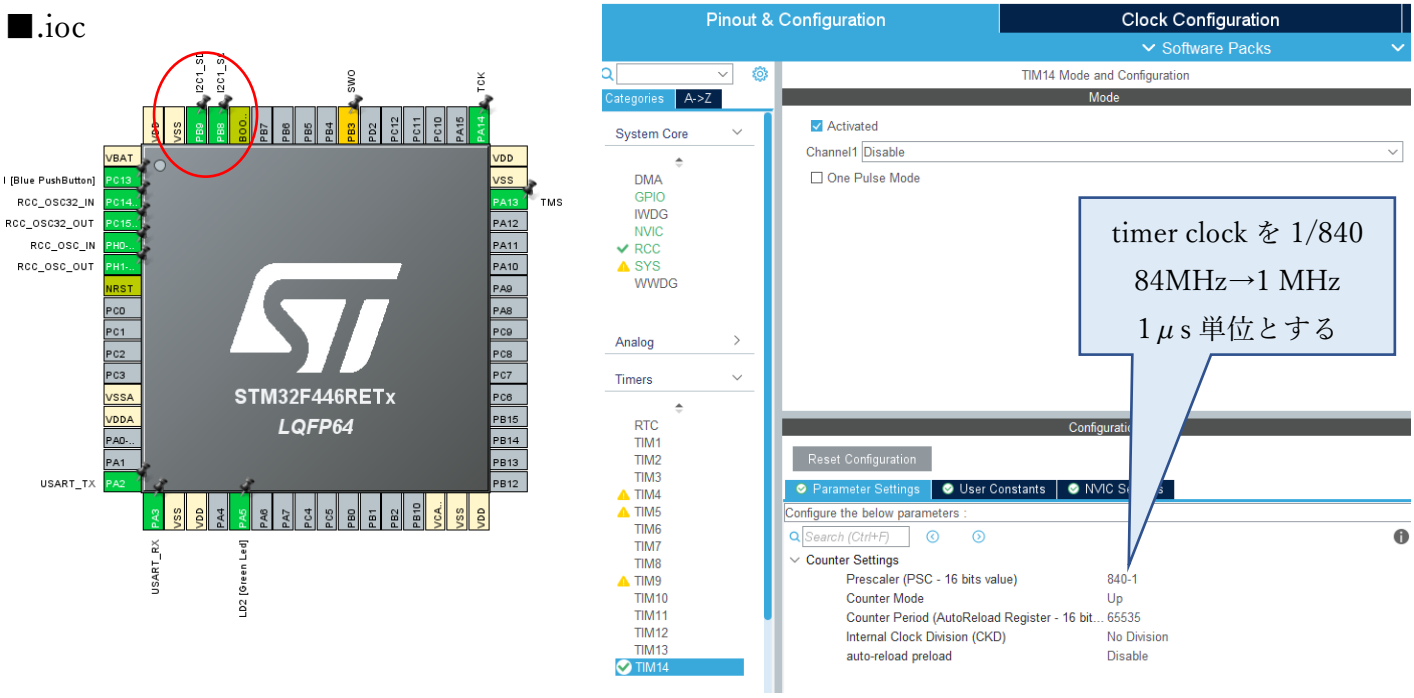
✓ Processing のプログラムは自作した。なお Madgwick Filter から出力されるオイラー角は特異姿勢が存在するようだ。Quaternion をそのまま利用する方が良いか？（後の姿勢検出では Quaternion を利用した）

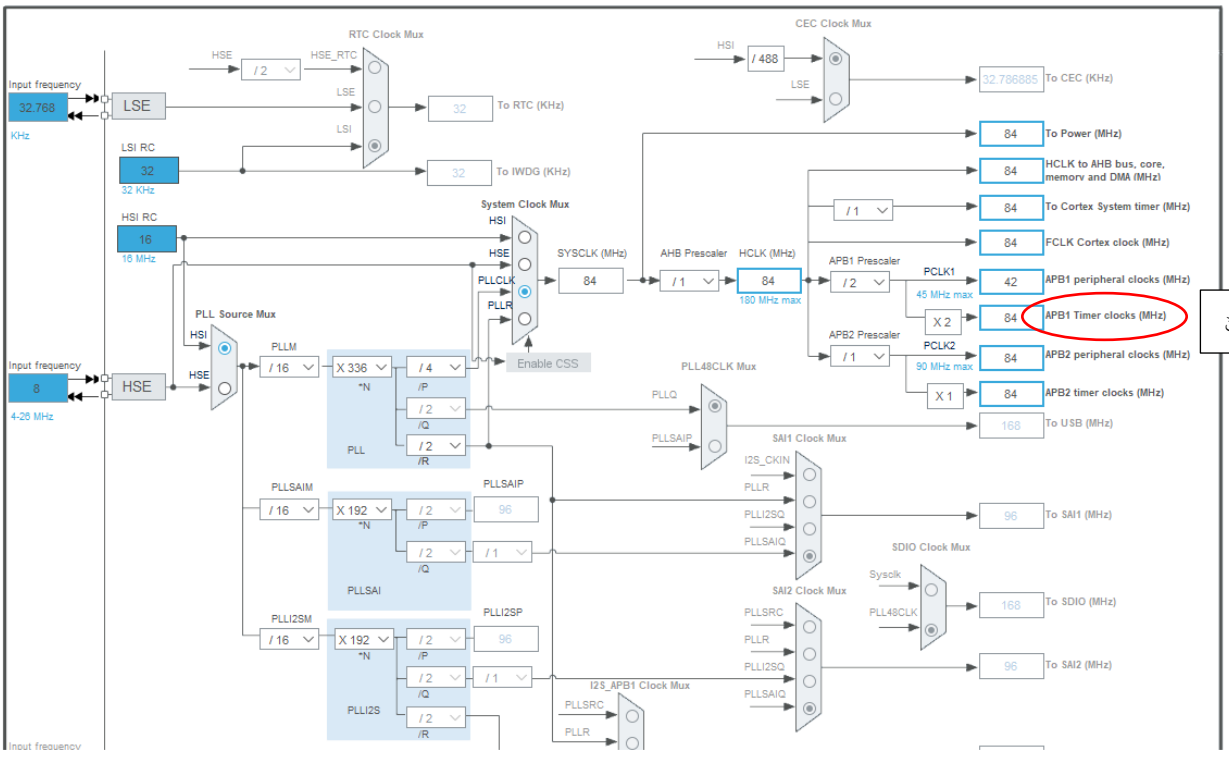
後記：プログラムの作成後、数週間経過するとプログラムの詳細が分からなくなった。年齢のためか（現在 71 歳)、いや昔からそうだった記憶がある。このため以下のプログラムに対する問い合わせは回答出来ません。あしからず。

（ファイルの場所：D:¥shimo¥Documents¥STM32workspace¥IMU2¥Madgwick4）

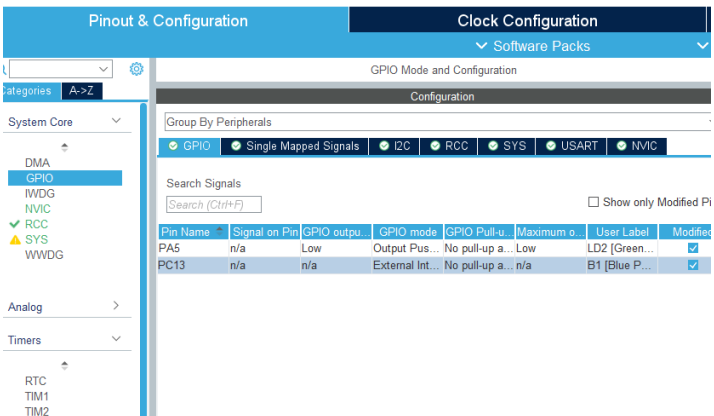
■STM32 NUCLEO F446RE で動作、開発は STM32 CUBE IDE を利用

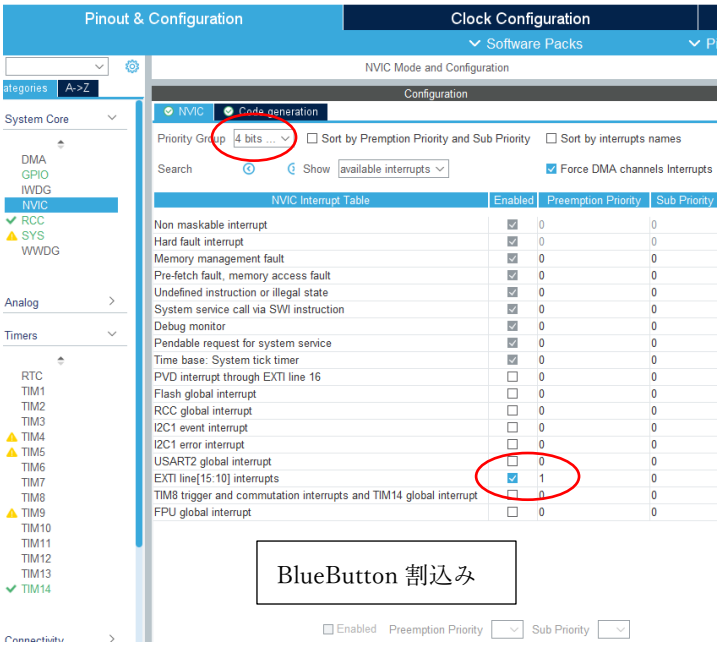
■.ioc

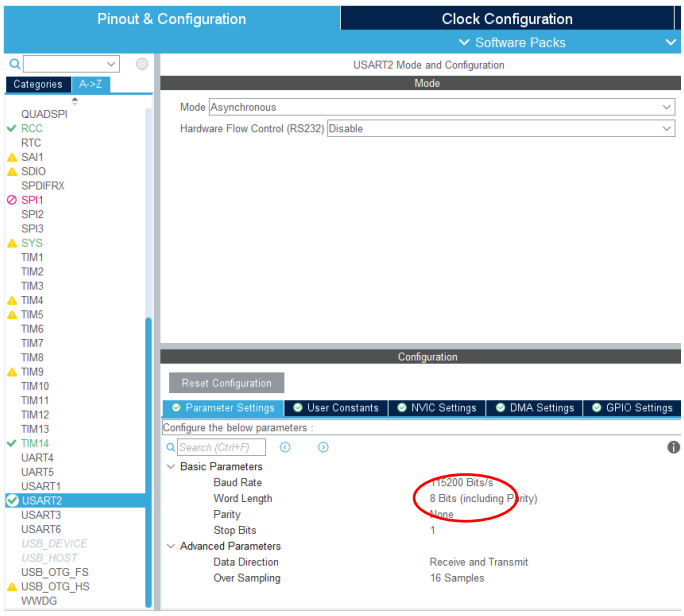


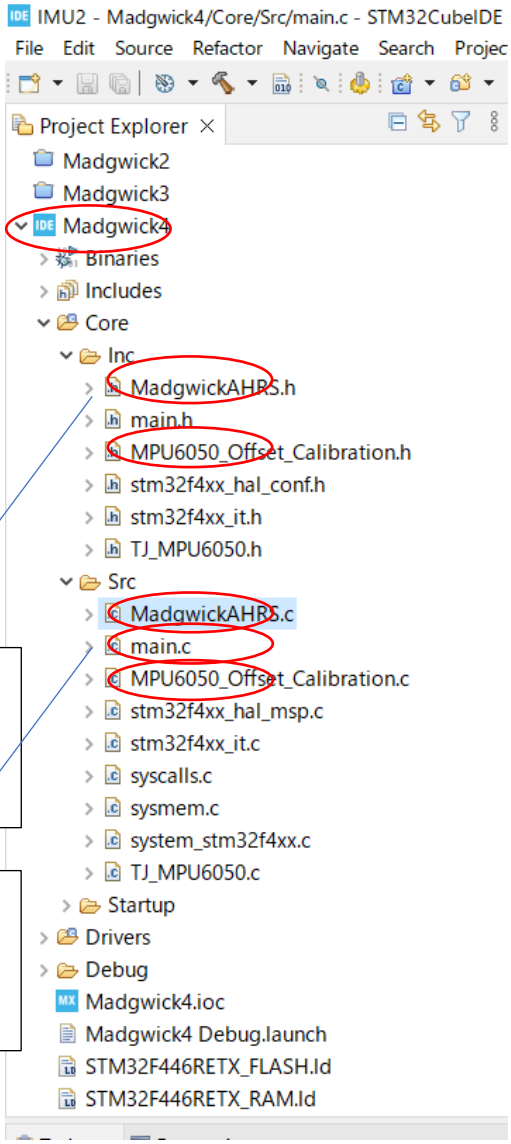


これの 1/840









プログラムの挿入方法。(**main.c 以外**)

① (win10)explore でプログラムが存在するフォルダを開き、MadgwickAHRS.h を右クリック→copy

② (CubeIDE)ProjectExplore のフォルダ(Core->Inc)で、右クリック→paste

プログラムの挿入方法。(**main.c**)

① main.c を貼り付け(paste)で行うと、**コンパイラエラーとなる。**

② このため、USER CODE BEGIN 1， 2， 3， 4， 5で個別にプログラムコードを paste していく。

■プログラムを作る

① MPU6050_offset_Calibration.h
② MPU6050_offset_Calibration.c
③ Madgwick.h　　一部改変（①のため）
④ Madgwick.c　　一部改変（①のため）
⑤ main.c

（1） MPU6050_offset_Calibration.h

maic.c から Call

ポインタ引数

```
// header file
void MPU6050_Offset_Calibration(int *Ax_bias, int *Ay_bias, int *Az_bias, int *Gx_bias, int *Gy_bias, int
*Gz_bias );
//
```

（2） MPU6050_offset_Calibration.c

printf を使う時に入れた

10 回程度計測しその平均をとる

ポインタを使い値の入出力を行う

必要、include "TJ_MPU6050.h"も必要
これに気付くのに時間がかかった！

ローカル変数とした
大意無し

データ構造を構築する情報が必要ですよね

平均をとる

```
/*
* MPU6050 offset calibration. shimojo
*  拙いプログラムのため、ただの参考にしてください。間違いが多くあります。
*/

//Header files
#include <stdio.h>
#include "MPU6050_Offset_Calibration.h"
#include "TJ_MPU6050.h"
#define Cal_num   10 //オフセットを校正するための計測回数


//=============================================================
// Functions
//-------------------------------------------------------------
// Offset Caliburation

void MPU6050_Offset_Calibration(int *Ax_bias, int *Ay_bias, int *Az_bias, int *Gx_bias, int *Gy_bias, int *Gz_bias )
{
      RawData_Def myAccelRaw, myGyroRaw; // add
      //ScaledData_Def myAccelScaled, myGyroScaled; // add

    int Ax,Ay,Az;//Raw data
    int Gx,Gy,Gz;//Raw data
    int Ax_av, Ay_av, Az_av=0;
    int Gx_av, Gy_av, Gz_av=0;
    int i=0;

    Ax_av=0; Ay_av=0; Az_av=0;
    Gx_av=0; Gy_av=0; Gz_av=0;

// Get initial measure data
    while (i< Cal_num )
    {
        MPU6050_Get_Accel_RawData(&myAccelRaw);//この命令で加速度とジャイロを取得
        MPU6050_Get_Gyro_RawData(&myGyroRaw);//上の命令で取得したデータを渡すだけ

        Ax=myAccelRaw.x-*Ax_bias;   Ay=myAccelRaw.y-*Ay_bias; Az=myAccelRaw.z-*Az_bias;
        Gx=myGyroRaw.x-*Gx_bias;   Gy=myGyroRaw.y-*Gy_bias; Gz=myGyroRaw.z-*Gz_bias;
      Ax_av+=Ax; Ay_av+=Ay; Az_av+=Az;
      Gx_av+=Gx; Gy_av+=Gy; Gz_av+=Gz;
      i+=1;
      HAL_Delay(10); //delay 10ms
    }
    Ax_av=(float)Ax_av/Cal_num; Ay_av=(float)Ay_av/Cal_num;Az_av=(float)Az_av/Cal_num;
    Gx_av=(float)Gx_av/Cal_num; Gy_av=(float)Gy_av/Cal_num;Gz_av=(float)Gz_av/Cal_num;

    // calculate average of offset value & Correct for GyroData only.
```

```
  // *Ax_bias+=(float)Ax_av; *Ay_bias+=(float)Ay_av;*Az_bias+=(float)Az_av;
    *Gx_bias+=(float)Gx_av; *Gy_bias+=(float)Gy_av;*Gz_bias+=(float)Gz_av;
}
```

ジャイロ角速度のみ戻す

なぜ、ジャイロ角速度のみオフセットの校正値を戻すのか？

① MPU6050 で姿勢検出を行ったところ、時間経過に従って姿勢がずれていく。例えば、z 軸を上面としたとき、z 軸周りの姿勢回転が増加する現象が観測された。これは z 軸周りの角速度のオフセットがあると、z 軸周りの回転があることになり、その積分として z 軸周りの回転が発生することになる。このため、ジャイロからの角速度出力オフセットは出来る限りゼロとすべきである。

② 翻って、加速度成分の姿勢変動の寄与は少ない。加速度成分は、重力が方向を検出することで姿勢を検出する。この場合、軸方向成分では重力場方向の値が大きい。また姿勢推定アルゴリズムでは、加速度の X,Y,Z の相対的関係より姿勢を検出（正規化して利用する）しているため、温度などの一様なドリフトの影響は少なくなる。

③ Blue-Button を押し、Flag を割込みにより設定して、MPU6050_offset_Calibration を行うようにしている。これはあくまでもジャイロ角速度のオフセット値の変更であり、ボタンを押した姿勢をオリジナルの姿勢とする分けではない。

（3） Madgwick.h 一部改変

```
//extern volatile float sampling_Freqency;      // sampling frequency shimojo
//---------------------------
---
// Function declarations
void MadgwickSetSampling(float samFreq);  //shimojo

void MadgwickAHRSupdate(float gx, float gy, float gz, float ax, float ay, float az, float mx,
float my, float mz);
void MadgwickAHRSupdateIMU(float gx, float gy, float gz, float ax, float ay, float az);

```

ここで関数を宣言した。コンパイラに予め指示する必要がある

（4） Madgwick.c 一部改変

```
// Variable definitions

volatile float beta = betaDef;                              // 2 * proportional gain (Kp)
volatile float q0 = 1.0f, q1 = 0.0f, q2 = 0.0f, q3 = 0.0f;  // quaternion of sensor frame relative to
auxiliary frame

volatile float sampleFreq = 100.0f;     // sampling frequency shimojo ？？？
//--------------------------------------------------------------------------------
```

必要か不明

```
// AHRS algorithm update

#if 1
//The sampling frequency is set in the main program.(by shimojo)
void MadgwickSetSampling(float samFreq){
    //float sampleFreq;
    sampleFreq=samFreq;
    }
//shimojo
#endif
```

不要；コンパイルオプション

新たに付け足した関数

main からの引数（値のみ引き渡す）

不要；コンパイルオプション

Madgwick Filter ではサンプリング周期に重要な意味がある。

その逆数のΔtがフィルタアルゴリズムで利用される。Δtは積分として扱われるので、その値の大きさが、結果の大きさに反映される。すなわちΔtが2倍になれは、フィルタ出力（quaternion）も2倍になる。

```
// Fast inverse square-root
// See: http://en.wikipedia.org/wiki/Fast_inverse_square_root

float invSqrt(float x) {
    float halfx = 0.5f * x;
    float y = x;
    long i = *(long*)&y;
    i = 0x5f3759df - (i>>1);
    y = *(float*)&i;
    y = y * (1.5f - (halfx * y * y));//1st iteration shimojo
    y = y * (1.5f - (halfx * y * y));//2nd iteration shimojo
    return y;
}
```

inverse square root の近似式。2nd iteration まで記入した。これは無視可能とのこと（実行しなくてもよい）

（5）main.c

①CODE BEGIN Includes

```
/* USER CODE BEGIN Includes */
#include <stdio.h> //add shimojo
#include <stdint.h> //add shimojo
#include "TJ_MPU6050.h"
#include "MadgwickAHRS.h"
#include "MPU6050_Offset_Calibration.h"
#include "math.h"
/* USER CODE END Includes */
```

不要

新たに挿入

```
/* USER CODE BEGIN 0 */

/* Gloval values */
const float PI=3.14159265358979;
extern volatile float beta; // add extern
extern volatile float q0, q1, q2, q3; // add extern
uint8_t data[5];
float Yaw_m,Pitch_m,Roll_m;

//It is made for sampling period measurement. Its value is observed by DEBUGGER(shimojo)
uint16_t timer_val; // timer start value
uint16_t timer_val2; // sampling period

/* USER CODE END 0 */
```

グローバル変数とした

CPU 処理経過時間計測用

②USER CODE BEGIN 1

```
int main(void)
{
  /* USER CODE BEGIN 1 */
  MPU_ConfigTypeDef myMpuConfig; // ジャイロセンサの特性を決めるパラメータを set
  /* USER CODE END 1 */
```

③USER CODE BEGIN 2

```
/* USER CODE BEGIN 2 */

  RawData_Def myAccelRaw, myGyroRaw; // add
  //ScaledData_Def myAccelScaled, myGyroScaled; // add

   //printf("Gyro Start Madgwick3 ¥r¥n");
```

データ構造。今回生データを基に自前で処理

スケーリングは、自前でおこなうので不要

```
    int Ax,Ay,Az;//Raw data
    int Gx,Gy,Gz;//Raw data
    float Axf,Ayf,Azf;//Scaled data
    float Gxf,Gyf,Gzf;//Scaled data

    // Offset average
    int Ax_bias,Ay_bias,Az_bias;//bias
    int Gx_bias,Gy_bias,Gz_bias;//bias
    Ax_bias=671;Ay_bias=-150;Az_bias=1099; // measured data by experiment by sh
    Gx_bias=-193;Gy_bias=86;Gz_bias=-50;// measured data by experiment by shimo
    float AccelMetricScaling;
    //Here the gravity unit. Since acceleration is normalized. (shimojo)
    AccelMetricScaling=(2000.0f/32768.0f);   //2000.0f for AFS_SEL_2g
    float GyroMetricScaling;
  // Gyro's Angular velocity uses radians. (shimojo)
    GyroMetricScaling= (250.0f/32768.0f)*(PI/180.0f);      //250.0f for FS_SEL_250
   // GyroMetricScaling= (500.0f/32768.0f);      //500.0f for FS_SEL_500

#if 1
    /* The sampling frequency is calculated by the following formula
     * sampling frequency=1000/(5+delay(ms))
     * exsample 50=1000/(5+15)
     * CPU processing time=5ms, delay=15ms
     */
    MadgwickSetSampling(66.7f); // Setsampling Frequency (Added by shimojo)
#endif

    //timer is created for sampling period measurement. Its value is observed by DEBUGGER(shimojo)
    // Start timer
     HAL_TIM_Base_Start(&htim14); //

    //1. initialize the MPU6050 module and i2c
      MPU6050_Init(&hi2c1);

    //2.   configure accel and gyro parameters

      myMpuConfig.Accel_Full_Scale=AFS_SEL_2g   ;
      myMpuConfig.ClockSource = Internal_8MHz;
      //myMpuConfig.CONFIG_DLPF =   DLPF_21A_20G_Hz;// Delay 8.5ms
      myMpuConfig.CONFIG_DLPF = DLPF_184A_188G_Hz;// Delay 2.0ms
     // myMpuConfig.CONFIG_DLPF = DLPF_260A_256G_Hz;//
      myMpuConfig.Gyro_Full_Scale =FS_SEL_250;
      //myMpuConfig.Gyro_Full_Scale = FS_SEL_500;
      myMpuConfig.Sleep_Mode_Bit = 0;   //1: sleep mode, 0: normal mode
      MPU6050_Config(&myMpuConfig);

      Ax_bias=671;Ay_bias=-150;Az_bias=1099; // measured data by experiment by shimojo
      Gx_bias=-193;Gy_bias=86;Gz_bias=-50;// measured data by experiment by shimojo

      //Here the gravity unit. Since acceleration is normalized. (shimojo)
      AccelMetricScaling=(2000.0f/32768.0f);   //2000.0f for AFS_SEL_2g

    // Gyro's Angular velocity uses radians. (shimojo)
      GyroMetricScaling= (250.0f/32768.0f)*(PI/180.0f);      //250.0f for FS_SEL_500
   // GyroMetricScaling= (500.0f/32768.0f);      //500.0f for FS_SEL_500

  // Calculate offset value. Compensate only for Gyro offset data.
    //Because,Attitude drift is due to gyro angular velocity offset. (shimojo)

      MPU6050_Offset_Calibration(&Ax_bias, &Ay_bias, &Az_bias,   &Gx_bias, &Gy_bias, &Gz_bias );

  /* USER CODE END 2 */
```

グローバル変数とした。ただ、MPU6050_offset_Calibration.c ではローカル変数としてますね。不統一ですね。また extern が無い、上の記述と相違。extern は必要/不要？？

キャリブレーションは事前に行った。これは重要
① 加速度は、x,y,z 軸を各々重力場方向と正逆配置して中間点をゼロとした
② 角速度は、静止状態で測定、オフセットを求めた
③ LowPass フィルタ、感度設定で値が若干変わる

角速度は radian/s とする

不要、コンパイラオプション

サンプリング周期の MadgwickFilter への通知は本質的に必要。オリジナルでは Madgwick.c 内の#define で指定している。これは面倒くさいので、プログラムを下条が改変した。

サンプリング周期を Magdwick へ渡す

CPU 処理計測用タイマを enable

MPU6050 の各種設定を行う。ローパスフィルタは、ノイズと遅延時間のトレードオフ。感度も同様。

オフセット値は事前に計測した。特に加速度センサは、x,y,z 各軸に対して重力場を反転させ、計測し、中立点を設定した。あと温度変動は行っていない。必要かも？？

ジャイロの角速度はラジアン/秒とする。
加速度は、正規化し相対的な値として利用するので単位は問わない。

“&” アドレスを渡す

④ USER CODE BEGIN 3

```
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */
```

```
    /* USER CODE BEGIN 3 */

  //if (Cal_num==1)     {   printf("%u us¥r¥n",timer_val2);Cal_num=0;   }
   timer_val = __HAL_TIM_GET_COUNTER(&htim14);//Start timer
    //Get Raw data
        MPU6050_Get_Accel_RawData(&myAccelRaw);//この命令で加速度とジャイロを取得
        MPU6050_Get_Gyro_RawData(&myGyroRaw);//上の命令で取得したデータを渡すだけ
        Ax=myAccelRaw.x-Ax_bias;   Ay=myAccelRaw.y-Ay_bias; Az=myAccelRaw.z-Az_bias;
        Gx=myGyroRaw.x-Gx_bias;   Gy=myGyroRaw.y-Gy_bias; Gz=myGyroRaw.z-Gz_bias;

        Axf=(float)Ax*(AccelMetricScaling);Ayf=(float)Ay*(AccelMetricScaling);Azf=(float)Az*(AccelMetricScaling);// gravity unit
        Gxf=(float)Gx*(GyroMetricScaling);Gyf=(float)Gy*(GyroMetricScaling);Gzf=(float)Gz*(GyroMetricScaling);// radian/s
        //Gxf=0.0f;Gyf=0.0f;Gzf=0.0f;// radian/s

        /*STM32CubeIDE の printf で float が使えない、対処方法   */
        /*
          プロジェクトを右 click, properties,
          C/C++ Build→Settings→MCU Settings,
          Use float with printf from newlib-nano (-u _printf_float)に check
         */

        //Madgwick Filter. Sampling frequency must be set in MadgwickAHRS.c (shimojo))
            MadgwickAHRSupdateIMU(Gxf,Gyf,Gzf,Axf,Ayf,Azf);

      HAL_Delay(10); //delay 5ms
      //Roll(X-axis),   Pitch(Y-axis), Yaw(Z-axis)
      Roll_m=(180.0f / PI)*atan2f(q0*q1 + q2*q3, 0.5f - q1*q1 - q2*q2);
      Pitch_m =(180.0f / PI)*asinf(-2.0f * (q1*q3 - q0*q2));
      Yaw_m=(180.0f / PI)*atan2f(q1*q2 + q0*q3, 0.5f - q2*q2 - q3*q3);

#if 1
       //Drawing graph by Processing
      printf ("%f",Roll_m); printf (",");//
    printf ("%f",Pitch_m); printf (",");//
      printf ("%f",Yaw_m); printf ("   ¥r¥n");//
       //get time elaasped
       timer_val2 = __HAL_TIM_GET_COUNTER(&htim14) - timer_val;
#endif

#if 0
        //MPU6050 measurd data quaternion
        printf ("%f",q0); printf (",");//
        printf ("%f",q1); printf (",");//
        printf ("%f",q2); printf (",");//
        printf ("%f",q3);   printf ("   ¥r¥n");//
#endif

#if 0
        //MPU6050 measurd data
        printf ("%f",Axf); printf (",");//
        printf ("%f",Ayf); printf (",");//
        printf ("%f",Azf); printf (",");//
        printf ("%f",Gxf); printf (",");//
        printf ("%f",Gyf); printf (",");//
        printf ("%f",Gzf);   printf ("   ¥r¥n");//
#endif

  }
  /* USER CODE END 3 */
```

CPU 処理計測　Start!

STM32CubeIDE の printf で%f は不可！
① プロジェクトを右 click, properties,
② C/C++ Build→Settings→MCU Settings,
③ Use float with printf from newlib-nano (-u _printf_float) に check

Madgwick Filter

ジャイロの角速度はラジアン/秒とする

サンプリング周期を決める遅延（重要）
遅延 0 だと Madgwick Filter の値が？？

Quaternion から Roll,Pitch,Yaw への変換
特異姿勢が存在するようだ。
Quaternion をそのまま利用する方が良いか？

CPU 処理計測　ここまでの経過時間

Quaternion 出力。processing のための format

コンパイラする/しないを指示

測定値出力

⑤ USER CODE BEGIN 4

```
/* USER CODE BEGIN 4 */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) //blueBotton の割込みルーチン
{
  if (GPIO_Pin == GPIO_PIN_13)
  {
    Pushed = 1;
  }
}
// printf 用コー?.ST-Link より Teraterm に出力す?,COM4 経由かな
int _write(int file, char *ptr, int len)
{
  HAL_UART_Transmit(&huart2,(uint8_t *)ptr,len,10);
  return len;
}
/* USER CODE END 4 */
```

BlueButton 用割込み処理

printf 処理

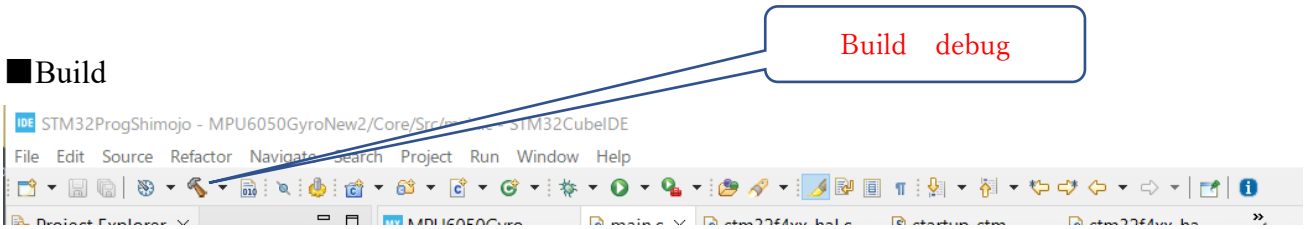
printf 用処理.　ST-Link より Teraterm に出力する
今回の例では、Wnidows10 の COM4 経由となっていた。

以上は基本設定終了

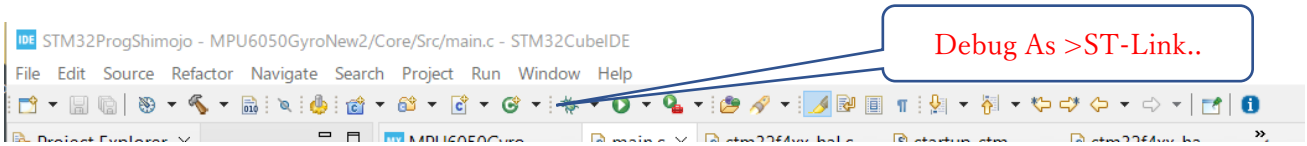
##

■以下は付録

■Build

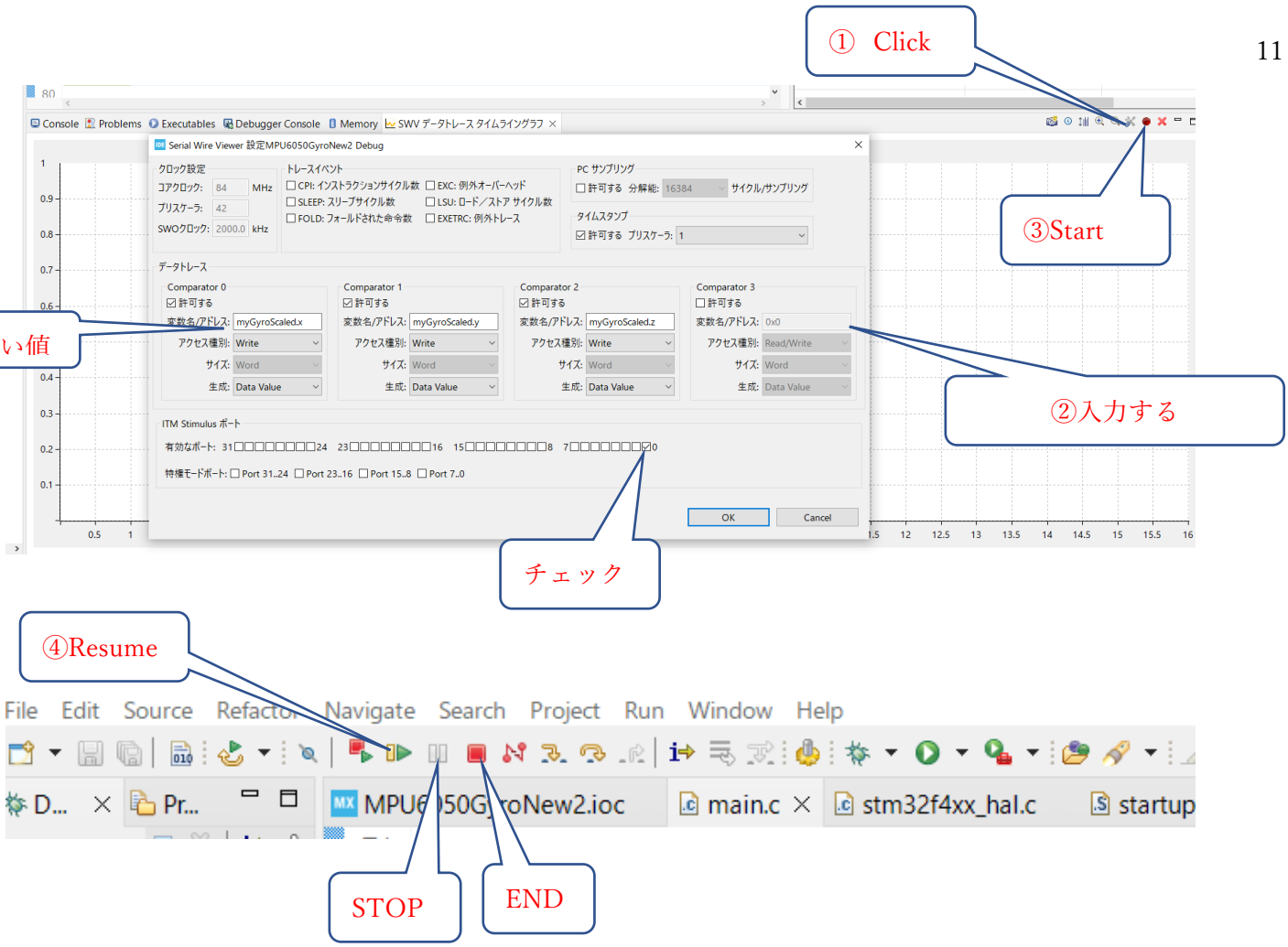


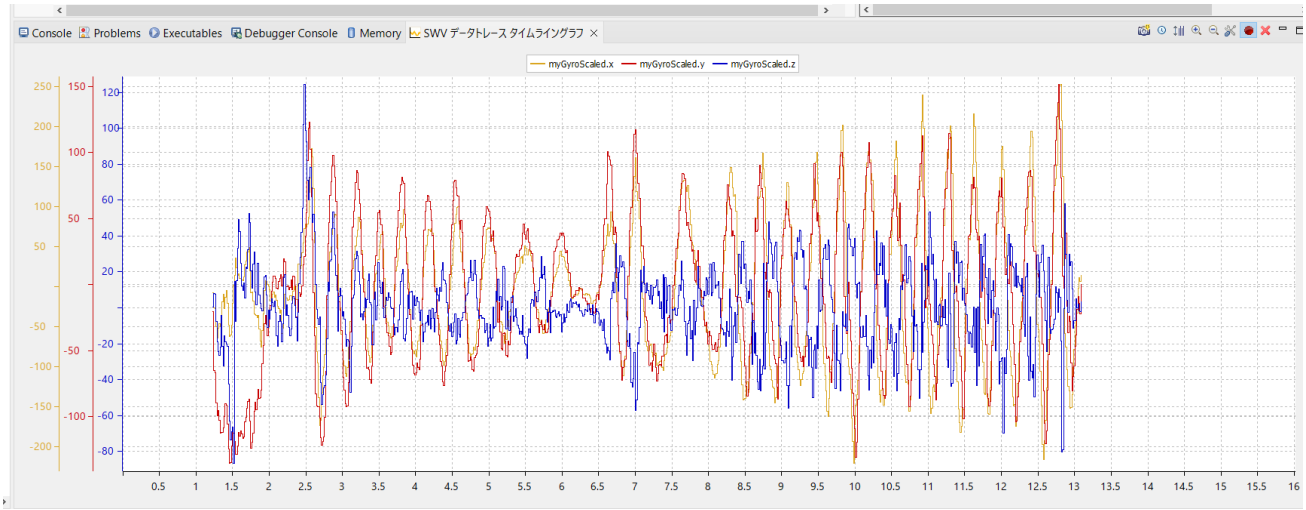
■Debug



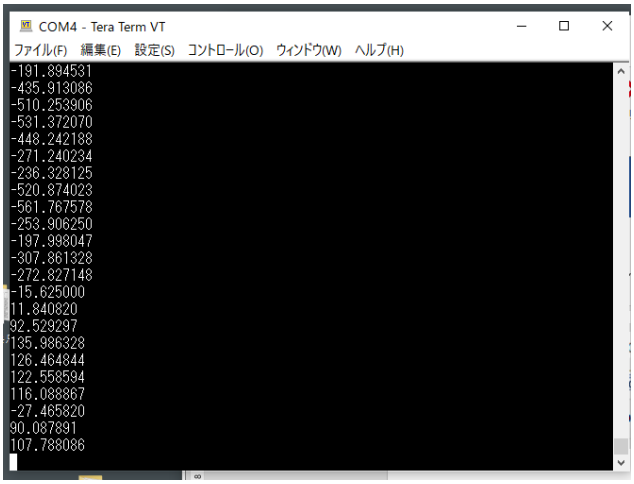
■測定データは、Debug mode で表示可能。
Debug>debug configuration>デバッガ＞特定の ST-Link..をチェック。探すをクリック
シリアルワイヤビューアを有効、Core Clock 84.0（自身の設定値）Debug

○デバックモードに入る
コマンドラインの LiveExpression をクリック、myAccelScaled（例えば）を追加
Resume をクリックすると、myAccelScaled.x、myAccelScaled.y、myAccelScaled.z
が出力される。





■TeraTerm 起動しているデータ出力



■描画プログラム processing でグラフを描く

https://processing.org/

◯MPU6050GyroNew2 の main.c に出力コードを入れる。

```
printf ("%f",myAccelScaled.x); printf (",");//
printf ("%f",myAccelScaled.y); printf (","); //
printf ("%f",myAccelScaled.z); printf (","); //
printf ("%f",myGyroScaled.x); printf (",");//
printf ("%f",myGyroScaled.y); printf (",");//
printf ("%f",myGyroScaled.z); printf ("    ¥r¥n");//

//HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5); // Green LED Blink
HAL_Delay(100); //delay 10ms
```

◯processing のプログラム。一部パラメータを変えた

https://shizenkarasuzon.hatenablog.com/entry/2019/02/16/163954

TeraTerm で COM4 の共存は不可。TeraTerm を終了する

```
import processing.serial.*;

static final int GYRO = 1;
static final int ACCEL = 2;
static final int MAG = 3;
static final int ROTATION = 4;

Serial myPort;

float gx, gy, gz;
float ax, ay, az;

graphMonitor GyroGraph;
graphMonitor AccelGraph;
graphMonitor MagGraph;
graphMonitor RotateGraph;
```

```
void setup(){
  size(1200, 800);

  myPort = new Serial(this, "COM4", 115200);
 //「COM4」とは、Arduino が接続されている COM ポート番号をさします。
 // エラーが出る場合は、デバイスマネージャ等から Arduino の COM ポート番号を調べて
書き換えてください。

  frameRate(30);
  smooth();
  myPort.bufferUntil('¥n');
  GyroGraph   = new graphMonitor("gyro_Rawdata",  100, 100,  900,
250);
  AccelGraph  = new graphMonitor("accel_Rawdata", 100, 400, 900, 250);
}

void draw(){
  background(0);
  //text(str(gx),100,100);
  GyroGraph.graphDraw(gx, gy, gz);
  AccelGraph.graphDraw(ax, ay, az);
}

void serialEvent(Serial myPort){
  String myString = myPort.readStringUntil('¥n');

  if (myString != null) {
    myString = trim(myString);

    float sensors[] = float(split(myString, ','));
    if (sensors.length > 5) {
        ax = sensors[0];
        ay = sensors[1];
        az = sensors[2];
        gx = sensors[3];
        gy = sensors[4];
        gz = sensors[5];
    }
  }
}
class graphMonitor {
    String TITLE;
```

```
    int X_POSITION, Y_POSITION;
    int X_LENGTH, Y_LENGTH;
    float [] y1, y2, y3;
    float maxRange;
    graphMonitor(String _TITLE, int _X_POSITION, int _Y_POSITION, int
_X_LENGTH, int _Y_LENGTH) {
      TITLE = _TITLE;
      X_POSITION = _X_POSITION;
      Y_POSITION = _Y_POSITION;
      X_LENGTH   = _X_LENGTH;
      Y_LENGTH   = _Y_LENGTH;
      y1 = new float[X_LENGTH];
      y2 = new float[X_LENGTH];
      y3 = new float[X_LENGTH];
      for (int i = 0; i < X_LENGTH; i++) {
        y1[i] = 0;
        y2[i] = 0;
        y3[i] = 0;
      }
    }

    void graphDraw(float _y1, float _y2, float _y3) {
      y1[X_LENGTH - 1] = _y1;
      y2[X_LENGTH - 1] = _y2;
      y3[X_LENGTH - 1] = _y3;
      for (int i = 0; i < X_LENGTH - 1; i++) {
        y1[i] = y1[i + 1];
        y2[i] = y2[i + 1];
        y3[i] = y3[i + 1];
      }
      maxRange = 1;
      for (int i = 0; i < X_LENGTH - 1; i++) {
        maxRange = (abs(y1[i]) > maxRange ? abs(y1[i]) : maxRange);
        maxRange = (abs(y2[i]) > maxRange ? abs(y2[i]) : maxRange);
        maxRange = (abs(y3[i]) > maxRange ? abs(y3[i]) : maxRange);
      }

      pushMatrix();

      translate(X_POSITION, Y_POSITION);
      fill(240);
      stroke(130);
```

```
      strokeWeight(1);
      rect(0, 0, X_LENGTH, Y_LENGTH);
      line(0, Y_LENGTH / 2, X_LENGTH, Y_LENGTH / 2);

      textSize(25);
      fill(255);
      textAlign(LEFT, BOTTOM);
      text(TITLE, 20, -5);
      textSize(22);
      textAlign(RIGHT);
      text(0, -5, Y_LENGTH / 2 + 7);
      text(nf(maxRange, 0, 1), -5, 18);
      text(nf(-1 * maxRange, 0, 1), -5, Y_LENGTH);

      translate(0, Y_LENGTH / 2);
      scale(1, -1);
      strokeWeight(1);
      for (int i = 0; i < X_LENGTH - 1; i++) {
        stroke(255, 0, 0);
        line(i, y1[i] * (Y_LENGTH / 2) / maxRange, i + 1, y1[i + 1] *
(Y_LENGTH / 2) / maxRange);
        stroke(255, 0, 255);
        line(i, y2[i] * (Y_LENGTH / 2) / maxRange, i + 1, y2[i + 1] *
(Y_LENGTH / 2) / maxRange);
        stroke(0, 0, 0);
        line(i, y3[i] * (Y_LENGTH / 2) / maxRange, i + 1, y3[i + 1] *
(Y_LENGTH / 2) / maxRange);
      }
      popMatrix();
    }
}
```

以上描画プログラム