

USB出力9軸IMU センサモジュール 仕様書

1.2版
2015年11月2日
株式会社アールティ

マニュアル改定歴

改定日	バージョン	変更内容	担当
2015/11/02	1.2版	ROS対応についての記述を追加 Linux環境でのサンプルに関する記述を追加	高橋
2015/10/21	1.1版	センサ出力値のバイアスについての記述を追加	高橋
2015/06/09	1.0版	内容の修正	高橋
2015/06/08	0.92版	サンプルプログラムの更新	高橋
2015/06/06	0.91版	図の追加	高橋
2015/05/27	0.9版	内容修正	高橋
2015/05/25	0.8版	内容追加	高橋
2015/05/10	0.7版	内容追加	高橋
2015/04/23	0.6版	誤字脱字修正	高橋
2015/04/23	0.5版	注意事項追加	中川範
2015/04/22	0.4版	内容追加	高橋
2015/04/22	0.3版	内容追加, 修正	中川範
2015/04/22	0.2版	内容追加	高橋
2015/03/10	0.1版	作成	重田

ファームウェア改定歴

改定日	バージョン	変更内容	担当
2015/6/09	4	ファームウェアを作成	高橋

目次

- [1.注意事項](#)
- [2. 概要](#)
 - [内容物](#)
 - [オプション品\(別売\)](#)
- [3.使用環境](#)
- [4.仕様](#)
 - [製品仕様](#)
 - [通信仕様](#)
 - [通信プロトコル](#)
 - [センサデータの物理量への変換式](#)
 - [センサ出力値のバイアス](#)
- [5.動作確認](#)
 - [準備するもの](#)
 - [ドライバインストール手順](#)
 - [サンプルプログラムの動作確認](#)
- [6.開発環境の構築](#)
- [7.LPCXpressoのサンプルプログラムについて](#)
 - [サンプルプログラムのインポート方法](#)
 - [UARTの通信ボーレートの変更方法](#)
 - [USB接続のタイムアウト時間の変更方法](#)
- [8.Linuxサンプルプログラムについて](#)
- [9.ファームウェアの書き込み](#)
- [10.図面](#)
- [11.お問い合わせ](#)

1. 注意事項

本製品をご使用頂く前に本マニュアルを熟読下さい。使用者および周囲の人に対する安全のため、内容をよく理解してから製品をお使い下さい。本製品をご使用したことによる、損害・損失について弊社は一切補償できません。また、本製品は民生用です。測定データの絶対的な信頼性の保証はできません。

2. 概要

本製品はInvenSense社(インベンセンス)の9軸センサ(3軸加速度, 3軸ジャイロ, 3軸コンパス)MPU9150を使用したセンサです。本モジュール上には9軸センサとマイコン(LPC1343)が実装されており、センサデータの取得とデータの処理を一枚の基板で実現できます。

USB出力9軸IMUセンサモジュール(以下、9軸センサモジュール)はロボット、マルチコプタ、ゲーム・アミューズメント、モーションセンサ、ポータブルナビゲーション、ラジコン、模型ヘリ、ヘッドマウントディスプレイなどの製品の開発段階での使用を想定して設計されています。インターフェースはUSBとUARTがあり、すでにマイコン側にファームウェアが書き込まれているので、接続するだけですぐにセンサ値を取得することができます。また、マイコンのプログラムを自分で書き換えることでボード側にてレンジの変更やフィルタ等の作成も可能です。

USBケーブル(A to microB)を用いてPCと接続した際にはVirtual COM Portとして認識されます。アプリケーション側からは通常のシリアルポートでのアクセスとまったく同様に使用することが可能です。出力されるデータは3軸加速度, 3軸地磁気, 3軸角速度, 温度になります。表1にデータの測定レンジと分解能を示します。また、ROS(Robot Operating System)にも対応しており、ROS用パッケージも公開されています。

	加速度	角速度	地磁気
測定レンジ	±16[g]	±2000[deg/sec]	±1200[μT]
分解能	16ビット	16ビット	13ビット

表1.データの測定レンジと分解能

内容物

- USB出力9軸IMUセンサモジュール 1台
- マニュアル 1枚

サンプルプログラム

- Processing用プログラム sketch_9axisIMU_test.pde
- Firmware RT-USB-9AXIS-00_20150608.bin
- LPCXpresso用サンプルプロジェクト
- Linux用サンプルプログラム usb9axis_test.c

オプション品（別売）

- USBケーブル(A to microB)
- USBシリアル変換モジュール

ROSパッケージ

パッケージ本体:rt_usb_9axisimu_driver

GitHubリポジトリ: https://github.com/rt-net/rt_usb_9axisimu_driver

説明wiki: https://github.com/rt-net/rt_usb_9axisimu_driver/wiki

3. 使用環境

OS: Windows XP SP2以上 / Vista / 7 (32/64bit)
 CPU: 800MHz以上の32bit(x86) or 64bit(x64)のプロセッサ
 Memory: 512MB以上
 Storage: 500MB以上
 USB: USB2.0 1ポート

参考: 一部のLinux, Mac環境にて動作を確認.

動作確認済みOSは以下になります.

Rasbian(Raspberry Pi 2), Ubuntu14.04, Mac OS X Marvericks 10.9.4

4. 仕様

製品仕様

- サイズ: 30mm×30mm×4.4mm
- 重量: 4g
- センサ: MPU9150
- USBバスパワー動作, (基板上に3.3V出力のレギュレータが実装されており, 3.6Vから 6.5Vまでの入力電圧に対応. この範囲以外の電圧は入力しないで下さい. レギュレータの故障につながります.)
- マイコン: LPC1343 72MHz 3.3V動作 センサ値取得用ファームウェア書き込み済み
- ローパスフィルタ 100Hz以上カット
- 各センサの測定レンジはMPU9150の最大レンジに設定

加速度, ジャイロ, 地磁気センサの出力の軸の取り方は以下のようになります. ジャイロの角速度は軸の方向に沿って反時計回りが正方向です. (右ねじを回して進む向きを軸の方向としたときに, ネジを進める回転方向が正)

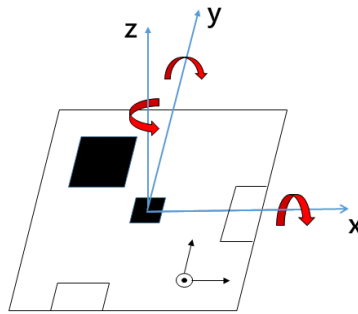


図1.ファームウェアの加速度, 角速度, 地磁気の正方向

注)本製品のファームウェアでは地磁気センサの軸の取り方がMPU9150と異なります。

モジュールとの接続方法

本モジュールはUSB, もしくはUARTでのデータ出力が可能です。

USB接続

USB接続の際はUSBケーブルでPCと本モジュールを接続するか, USBの4ピンコネクタとPCを接続してください。USB接続が確立されると基板上のLEDが低速で点滅します。また, USB接続の際にはUART端子からもセンサデータが出力されます。(USBケーブルを給電のみの目的で接続しUART端子からの出力を用いることが可能)

UART接続

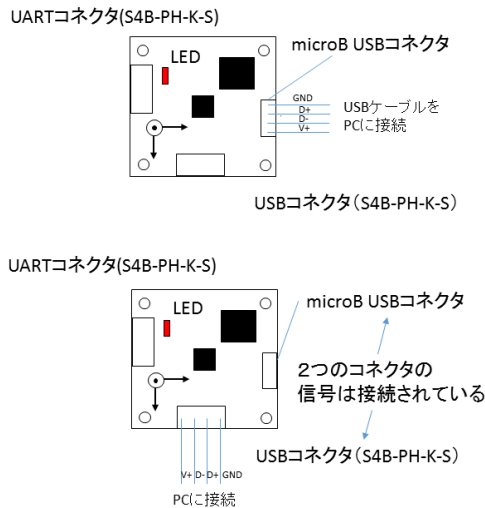
UART接続を使用するときはUART端子のGND, V+に3.6Vから6.5Vの外部電源を接続します。モジュール上のlpc1343マイコンは3.3V動作なのでレギュレータを通して3.3Vの電源がマイコンに供給されます。このときTX, RXの信号は3.3V系であることに注意してください。また, UART接続の際は基板上のLEDが高速で点滅します。

注1) UART端子とUSB端子の+VとGNDは基板上で結線されています。そのためUART端子から給電をした際にはUSB端子から給電をしないでください。USBポートまたは本モジュールの破損につながります。

注2)最初に書き込まれているファームウェアでは電源供給時にUSB接続を5秒間待ち, この間にUSBの接続が確立された場合はUSBおよびUARTの出力がされます。UARTのみの出力は接続してから5秒間にUSBの接続が確立されなかったときに始まります。USB接続を待つ時間の変更については7章に記述があるので参考にしてください。

モジュールとの接続例

USB接続時の配線例



UARTのみでの使用時の配線例

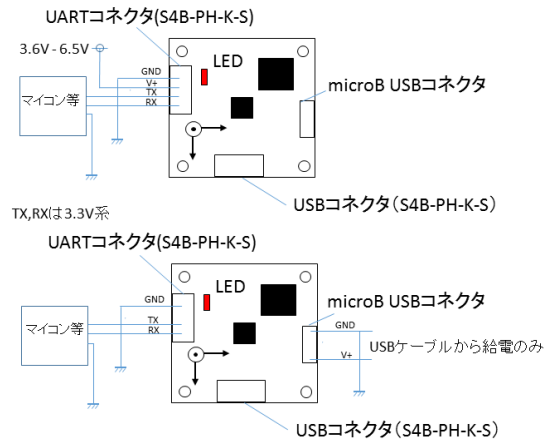
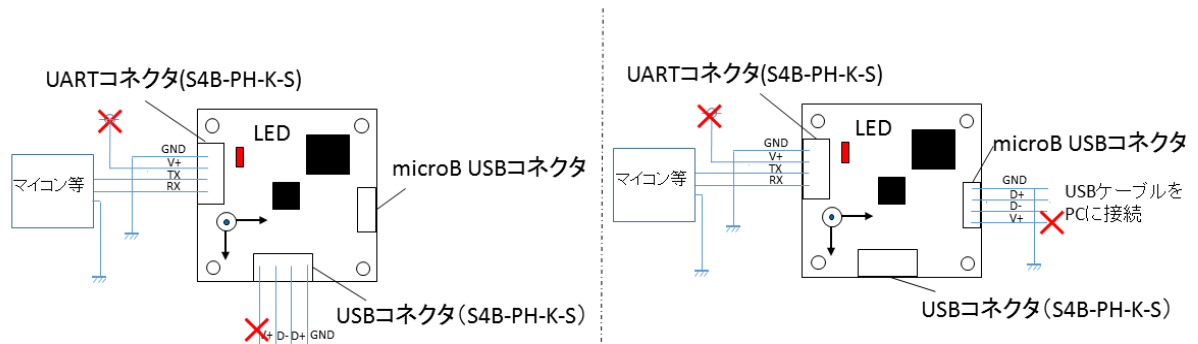


図2.

図2. モジュールとの接続図

接続時の注意



3つのコネクタのGNDとV+は基板上で結線されているため、2つのコネクタから異なる電圧の入力をしないでください。故障の原因となります。

図3. 禁止されている接続

ファームウェアの動作の流れ

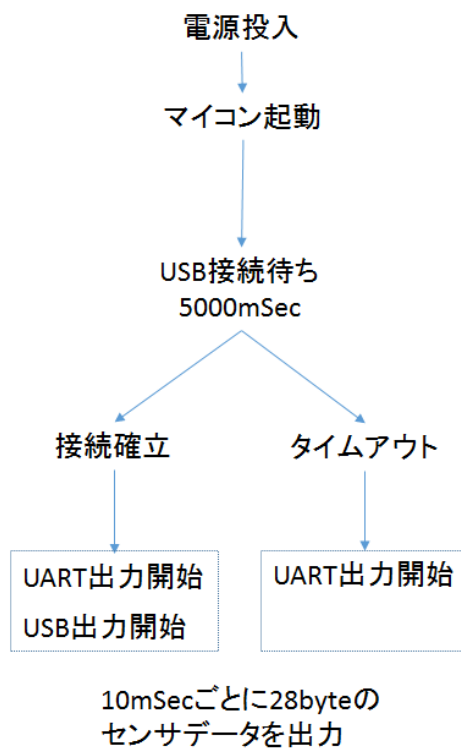


図4. ファームウェアの動作の流れ

通信仕様

USBケーブルでPCと接続した際にはVirtual COM Portとして認識されます。アプリケーション側からは通常のシリアルポートでのアクセスとまったく同様に使用することが可能です。出力されるデータは3軸加速度, 3軸地磁気, 3軸角速度, センサ温度になります。データ送信調歩同期式シリアル通信で行われます。モジュールからは10msec毎に28byteのセンサデータが送信され続けます。

- USB接続時：CDCクラスを使用して通信
- ボーレート：57600 bps
- データ：8bit
- パリティ：なし
- ストップビット：1bit
- フロー制御：なし
- データ送信周期：100Hz

通信プロトコル

センサ測定値の通信データは以下のような割り当てになります。0byte目から6byte目までの出力は常に固定なので、受信データの先頭を見つけるのに用いてください。7byte目のタイ

ムスタンプはデータが送信されるたびに0,1,2, ... ,255となり, 255の次は再び0になります。
通信データの抜けを確認するために使用してください。

Byte	内容	Byte	内容
0	0xff	14	AD値 (TEMP)下位8ビット
1	0xff	15	AD値 (TEMP)上位8ビット
2	0x52(ASCIIコードのR)	16	AD値 (GYRO X)下位8ビット
3	0x54(ASCIIコードのT)	17	AD値 (GYRO X)上位8ビット
4	0x39(製品識別子下位8bit)	18	AD値 (GYRO Y)下位8ビット
5	0x41(製品識別子上位8bit)	19	AD値 (GYRO Y)上位8ビット
6	ファームウェアバージョン	20	AD値 (GYRO Z)下位8ビット
7	タイムスタンプ(0x00 - 0xff)	21	AD値 (GYRO Z)上位8ビット
8	AD値 (ACC X)下位8ビット	22	AD値 (MAG X)下位8ビット
9	AD値 (ACC X)上位8ビット	23	AD値 (MAG X)上位8ビット
10	AD値 (ACC Y)下位8ビット	24	AD値 (MAG Y)下位8ビット
11	AD値 (ACC Y)上位8ビット	25	AD値 (MAG Y)上位8ビット
12	AD値 (ACC Z)下位8ビット	26	AD値 (MAG Z)下位8ビット
13	AD値 (ACC Z)上位8ビット	27	AD値 (MAG Z)上位8ビット

表2. 送信データの割り当て

センサデータの物理量への変換式

本モジュールのファームウェアでは, 各センサデータは16bit符号付整数で出力されます。受信データは上位, 下位8bitに分かれているので, 結合して16bit符号付整数データにしてください。

受信データの結合の例

concatenation [32bit 符号付き整数]

data_H [8bit符号なし整数]

data_L [8bit符号なし整数]

concatenation = data_L + (data_H<<8)

if(concatenation >= 32767) concatenation = concatenation - 65535

加速度

加速度センサ値 : acc [16bit 符号付整数]

計算式 : acc / 2048 [g]

ジャイロセンサ

ジャイロセンサ値 : ω [16bit符号付整数]

計算式: $(\omega) / 16.4$ [deg/sec]

地磁気センサ

地磁気センサ値 : mag [16bit符号付整数]

計算式 : $\text{mag} * 0.3$ [μT]

温度センサ

温度センサ値 : temp [16bit符号付整数]

計算式 : $\text{temp} / 340 + 35$ [$^{\circ}\text{C}$]

センサ出力値のバイアス

9軸センサモジュールから出力されるデータにはバイアス(センサの出力値の原点ずれ)が存在します。これはMEMS9軸センサ特有の現象です。バイアスを補正することより精度の高い測定データが得られます。

以下にジャイロセンサと加速度センサの簡易的なバイアスキャリブレーション方法を示します。ただし、この方法はあくまでも簡易的な方法なので注意してください。

1) ジャイロセンサ

センサを完全に静止させた状態でジャイロセンサ値を100-1000回程取得し平均をとります。これを ω_{ref} とします。これ以後のジャイロセンサの測定値 ω から ω_{ref} をオフセットしてください。

ジャイロリファレンス値 : ω_{ref} [16bit符号付整数]

ジャイロセンサ値 : ω [16bit符号付整数]

計算式: $(\omega - \omega_{\text{ref}}) / 16.4$ [deg/sec]

2) 加速度センサ

基本的にはジャイロセンサと同様な手順でキャリブレーションが可能ですが、加速度センサの場合静止状態で重力加速度を検出しますので、z軸のみ調整方法が変わります。

センサ基板のz軸が重力の方向と平行になるように置き、完全に静止させます。

静止状態で加速度センサ値を100-1000回程取得し平均を取ります。これを各軸 acc_ref_x , acc_ref_y , acc_ref_z とします。これ以後の加速度センサの測定値を acc_x , acc_y , acc_z とします。

加速度リファレンス値: acc_x , acc_y , acc_z [16bit符号付整数]

加速度センサ値: acc_x , acc_y , acc_z [16bit符号付整数]

計算式 acc_x : $(\text{acc_x} - \text{acc_ref_x}) / 2048$ [g]

計算式 $acc_y: (acc_y - acc_ref_y) / 2048[g]$
計算式 $acc_z: (acc_z - (acc_ref_z - 1)) / 2048 [g]$
(測定場所の重力の大きさが1gと仮定)

5. 動作確認

準備するもの

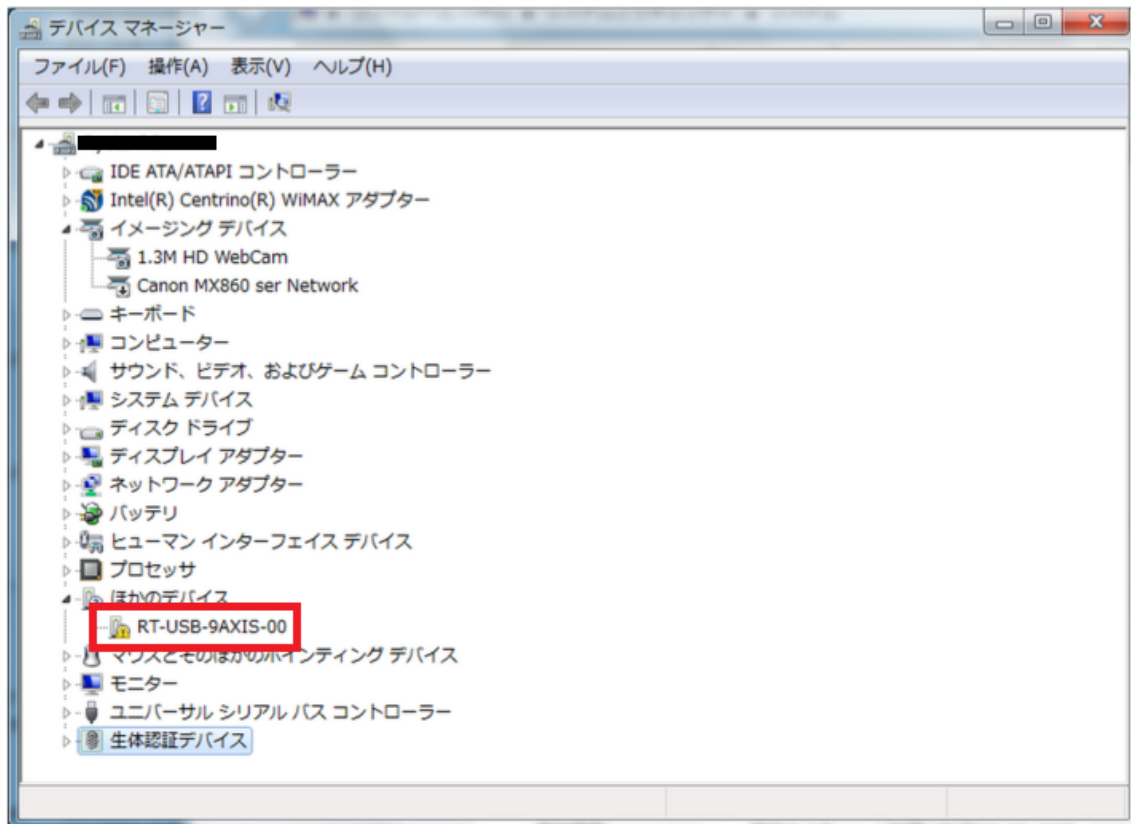
- 9軸センサモジュール
- USBケーブル(A to microB)

ドライバインストール手順

9軸センサモジュールがパソコンと通信を行うためには、対応したドライバのインストールが必要です。

1. 9軸センサモジュールと接続

9軸センサモジュールをPCにUSBで接続します。その状態でデバイスマネージャーを開いてください。すると、RT-USB-9AXIS-00というデバイスがあるので(ない場合はモジュールを接続したときに新しく表示されたデバイス)選択します。



2. ドライバのダウンロード

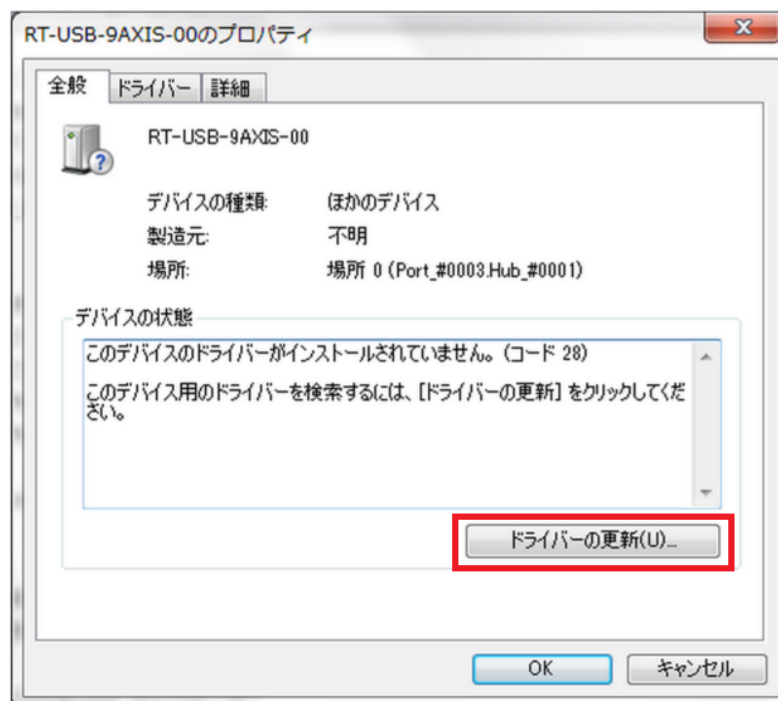
以下の、アールティロボットショップのダウンロードページにアクセスして下さい。

<http://www.rt-shop.jp/download/RT-IMU9/>

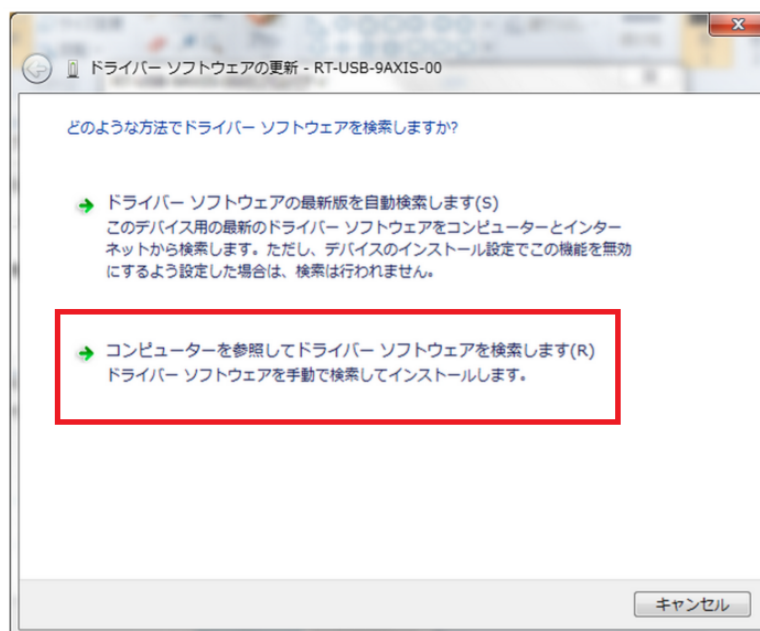
「RT-USB-9AXIS-00.zip」をダウンロードして、解凍して下さい。

3. ドライバの選択

デバイスドライバーがインストールされていないと表示がでます。次にドライバーの更新をクリックします。



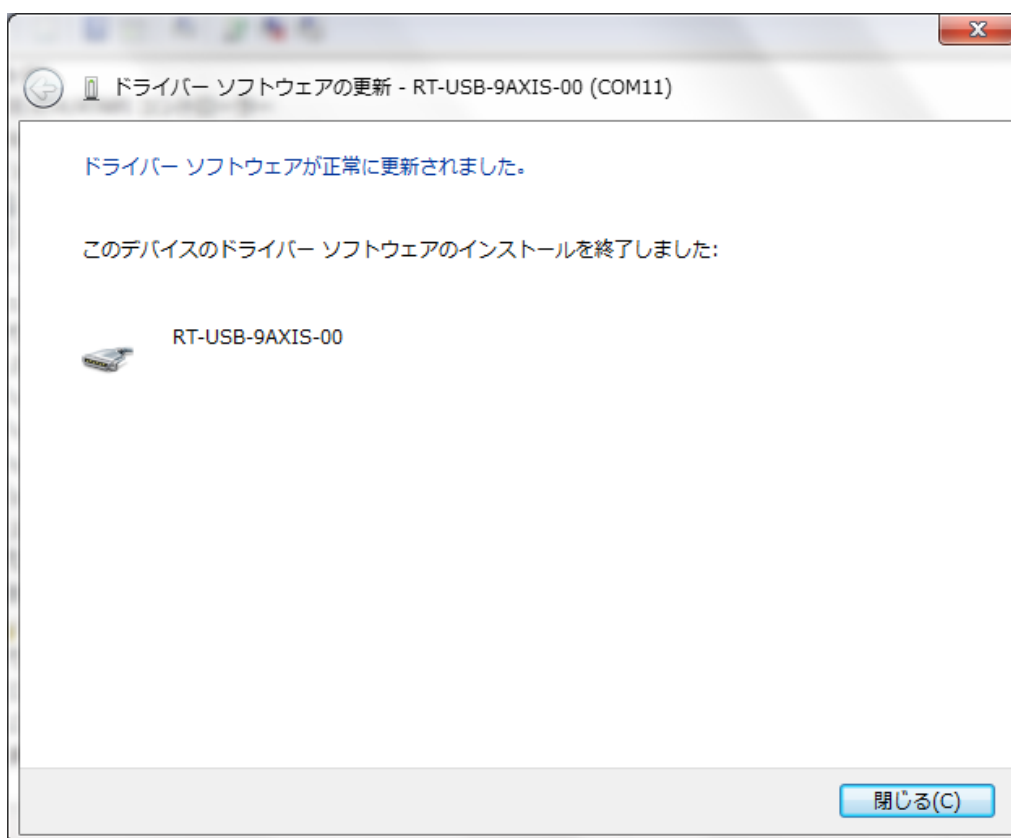
ダウンロードしたドライバを使用するため、「コンピュータを参照してドライバーソフトウェアを検索します」を選択します。



解凍したフォルダの中のdriverフォルダを選択し、インストールを開始します。



以下のような表示が出れば、インストール完了となります。



サンプルプログラムの動作確認

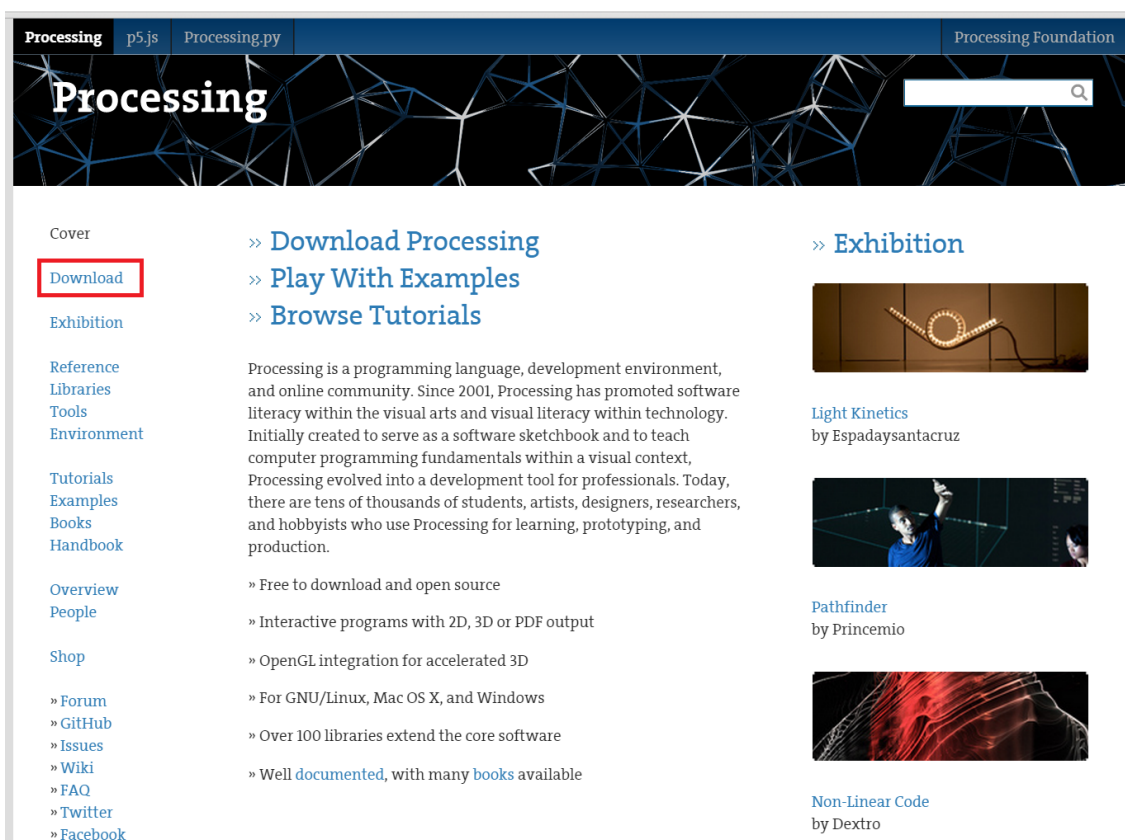
アールティロボットショップダウンロードサイトにあるサンプルソフトウェアを用いて、9軸センサモジュールの動作確認を行うことができます。使用するにはJavaがインストールされている必要がありますのでご注意ください。

Javaは次のHPからダウンロードできます。 <http://java.com/ja/download/>

サンプルプログラムの動作確認はProcessingというプログラミング言語を用いて行います。ProcessingはC言語風の文法で図形の描画等の視覚的表現が手軽に実現できます。

1. Processingのダウンロード

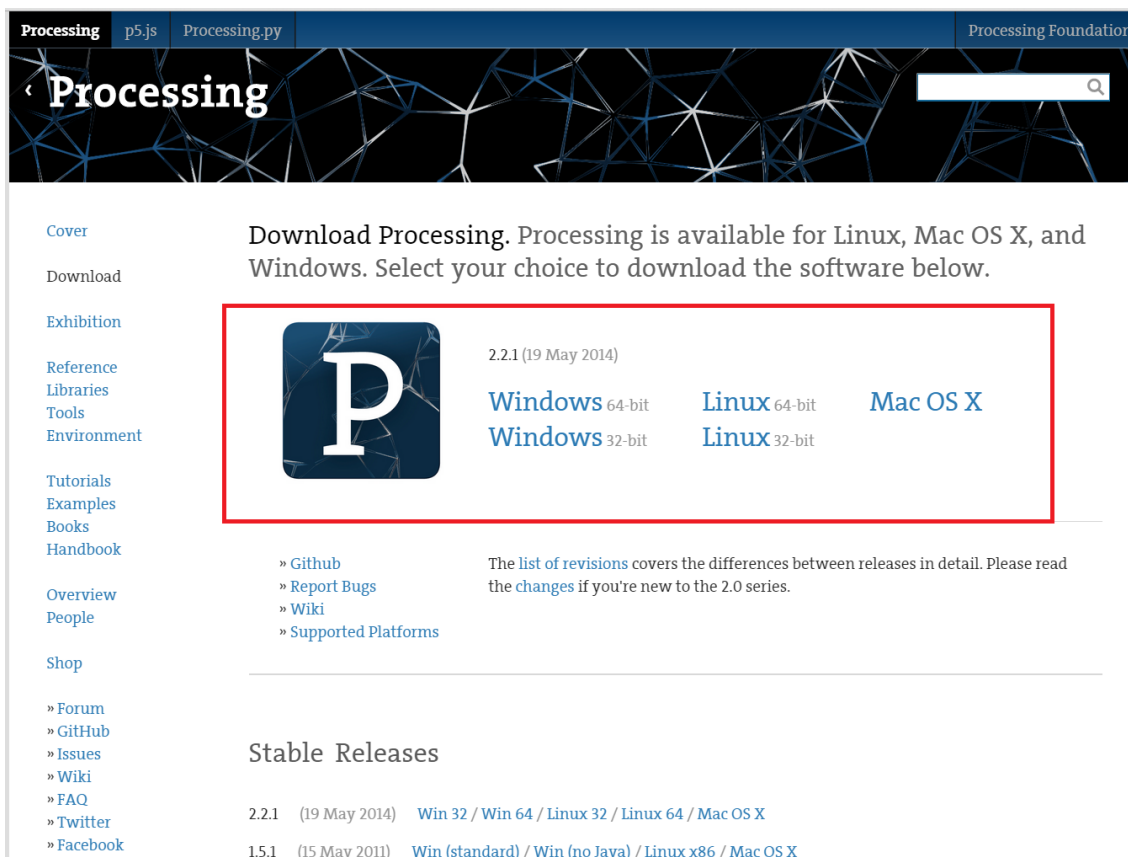
次のアドレス<https://processing.org/> からProcessingのHPに飛び画面左のDownloadをクリックします。



自分のPCの環境に合わせソフトウェアをダウンロードしてください。

また、Processingのバージョンが1.xと2.xでは仕様が異なりサンプルプログラムが動作しないためバージョン2.xをダウンロードしてください。

動作確認済みのバージョンは2.2.1(19 May 2014)



The screenshot shows the Processing website's download page. The header includes links for Processing, p5.js, Processing.py, and Processing Foundation. The main content area features a large 'Download Processing' button and links for Windows, Linux, and Mac OS X. A sidebar on the left contains various navigation links. The footer includes links to GitHub, Report Bugs, Wiki, and Supported Platforms.

ダウンロードが終わったら展開し、processing.exeを起動します。

2. プログラムのダウンロード

アールティロボットショップページから「RT-USB-9AXIS-00.zip」ファイルをダウンロードします

・ <http://www.rt-shop.jp/download/RT-IMU9/>

ダウンロードが終わったら展開します。

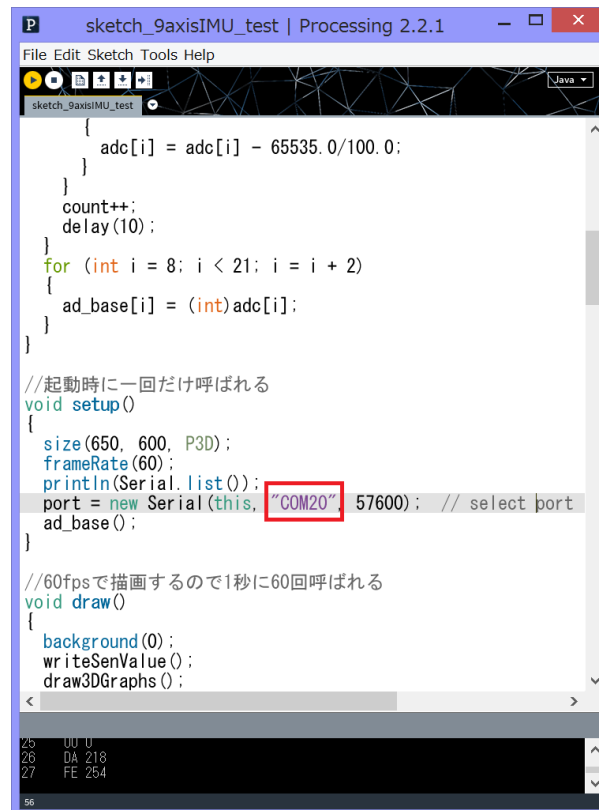
3. 実行する

Processingの方にもどり、「File」→「Open...」から、先ほどダウンロードしたプログラムの中にあるsketch_9axisIMU_test.pdeを選択します。デバイスマネージャにて9軸センサモジュールが接続されているポートを確認したのち、プログラム内の以下の部分を変更してください。

また、サンプルプログラムのコメントが文字化けする場合はProcessingの上部の

File >> Preferences >> Editor and Console font

と進みフォントを日本語表示できるもの(MS ゴシック等)に変更してください。



```

{
  adc[i] = adc[i] - 65535.0/100.0;
}
count++;
delay(10);
}
for (int i = 8; i < 21; i = i + 2)
{
  ad_base[i] = (int)adc[i];
}
}

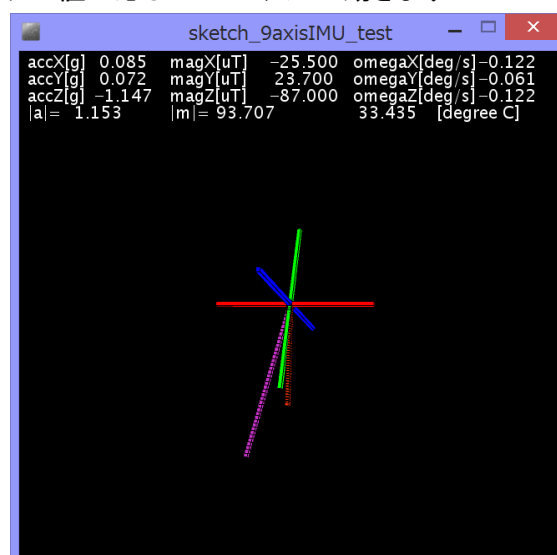
//起動時に一回だけ呼ばれる
void setup()
{
  size(650, 600, P3D);
  frameRate(60);
  println(Serial.list());
  port = new Serial(this, "COM20", 57600); // select port
  ad_base();
}

//60fpsで描画するので1秒に60回呼ばれる
void draw()
{
  background(0);
  writeSenValue();
  draw3DGraphs();
}

```

「COM*」の番号を設定したのち、左上の再生ボタン(三角のボタン)をクリックしてください。
(Linux, Macの環境の場合、COM*ではなく/dev/tty*)最初にセンサ値の原点取得を行うため。
始めは9軸センサモジュールを動かさないでください。

3Dモデルが表示されたら、9軸センサモジュールを好きな方向に傾けてみてください。加速度センサと地磁気センサの値に応じて3Dモデルが動きます。



6. 開発環境の構築

9軸センサモジュールに搭載されているマイコンには、あらかじめファームウェアが書き込まれていますが、ユーザーによるプログラムの書き換えも可能です。開発環境としては、NXP社から提供されている統合開発環境(IDE)LPCXpressoを推奨しています。

<http://www.lpcware.com/lpcxpresso>

LPCXpressoのアクティベーションを行うと、256KBまでのコードサイズで開発が可能となります。また、有料版のProバージョンを購入すると、サイズ制限はなくなります。

1. ダウンロード

以下のURLよりLPCXpressoのインストーラーをダウンロードします。

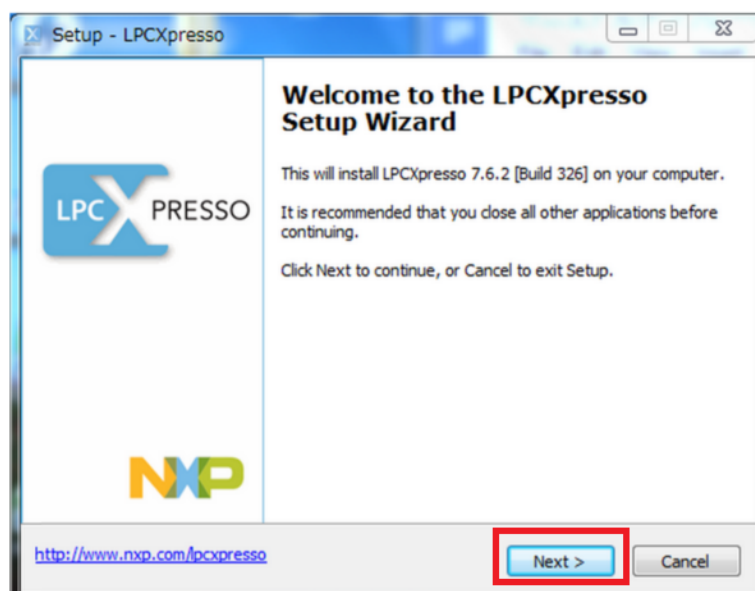
<http://www.lpcware.com/lpcxpresso/download>

ご自身のOSにあったものを選択してください。

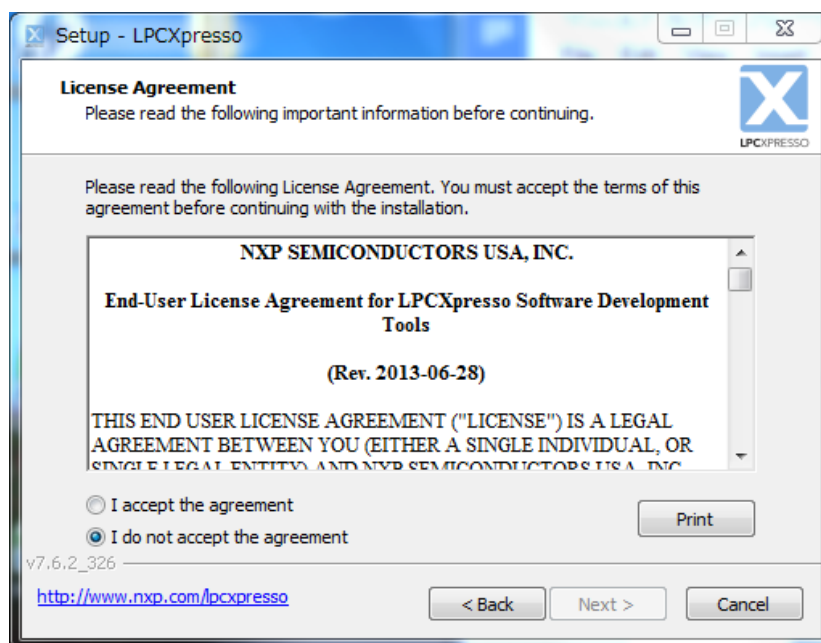
2. インストール

ダウンロードが完了したら、インストーラーをダブルクリックし実行します。

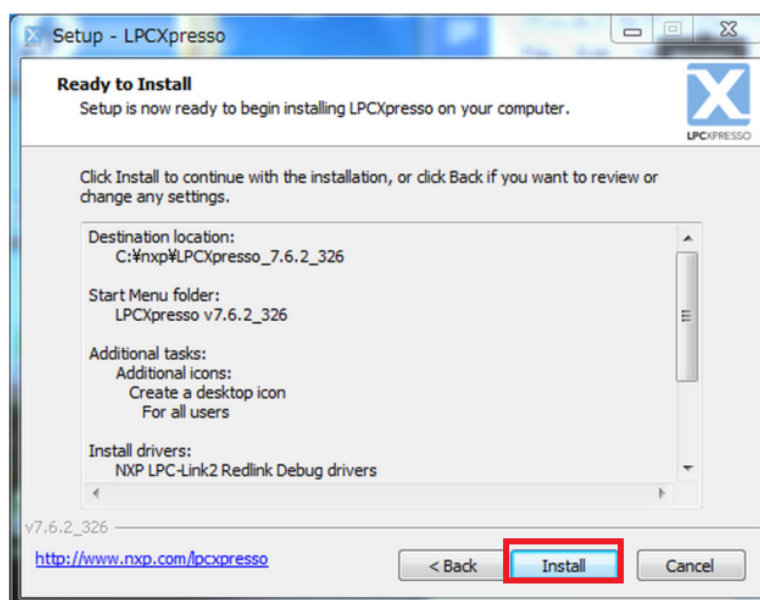
「Next >」をクリックします。



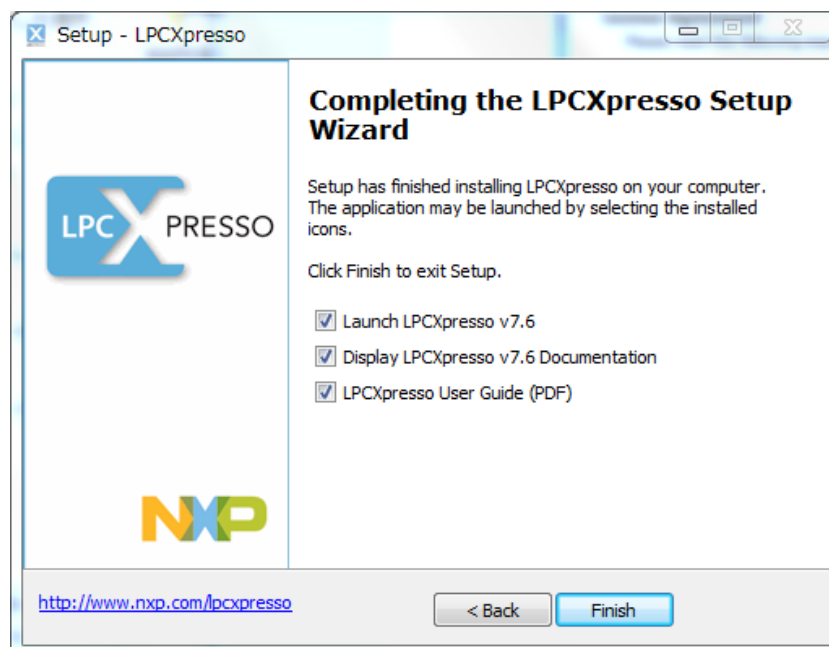
ライセンスの同意が求められますので、説明を読み同意するならば、「I accept the agreement」にチェックを入れて、「Next >」をクリックします。これ以降は変更が必要な点はありませんので、インストール開始まで「Next >」を選択し続けてください。



インストール直前まで設定が完了したら、「Install」をクリックしインストールします。

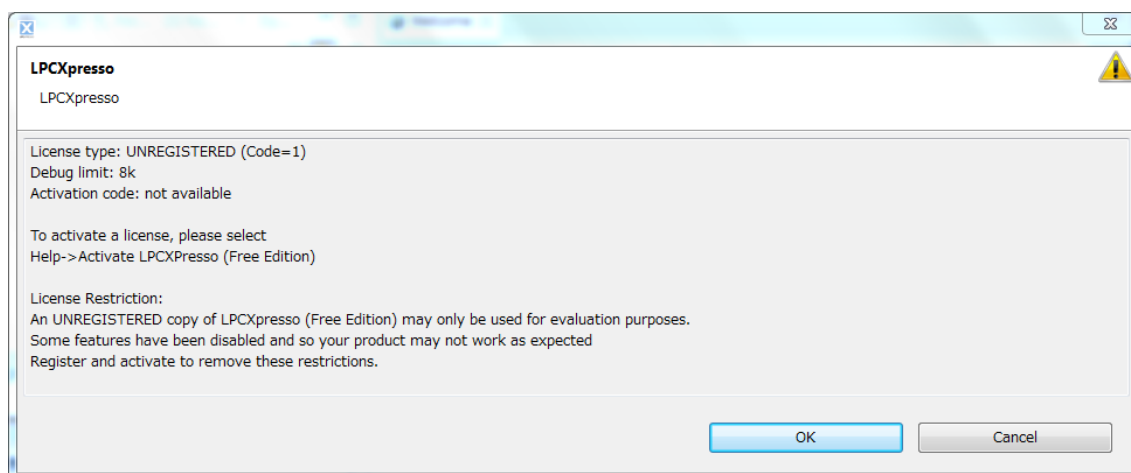


インストールが完了したら、「Finish」をクリックしインストーラを終了します。一番上のチェックボックスにチェックが入っていれば、LPCXpressoが自動で起動します。



3. アクティベーション

アクティベーションを行っていない状態では、以下のように8KBまでのサイズ制限が掛かっています。



このサイズ制限を広げるには、LPCware.comにてアカウントを作成し、アクティベーションに必要なシリアルコードを取得する必要があります。

7. LPCXpressoのサンプルプログラムについて

LPCXpressoのサンプルプログラムは「RT-USB-9AXIS-00.zip」内の「LPCXpresso Sample Program」にあります。このファイルをLPCXpressoにインポートして使用することが可能です。本章ではサンプルプログラムのインポート方法および、サンプルプログラムの一部を変更して、UARTの通信ボーレートとUSB接続のタイムアウト時間を変更する方法を説明します。

1. サンプルプログラムのインポート方法

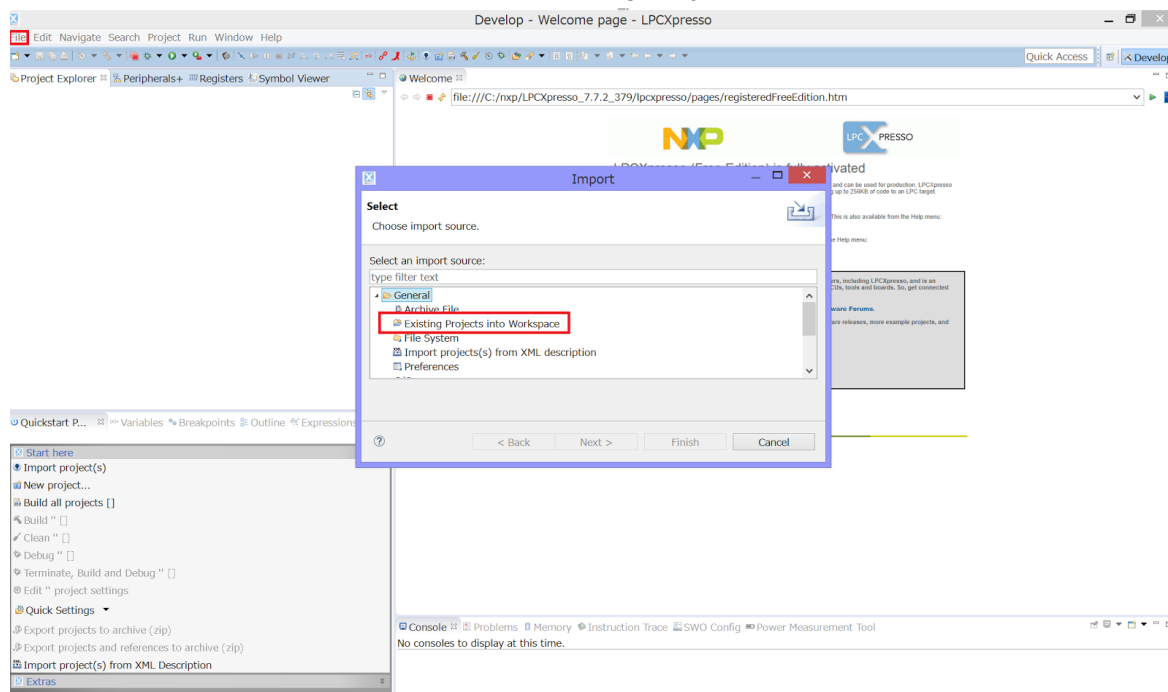
サンプルプログラムは「LPCXpresso Sample Program」内にある以下の2プロジェクトです。

- RT-USB-9AXIS-00_20150608
- CMSISv1p30_LPC13xx

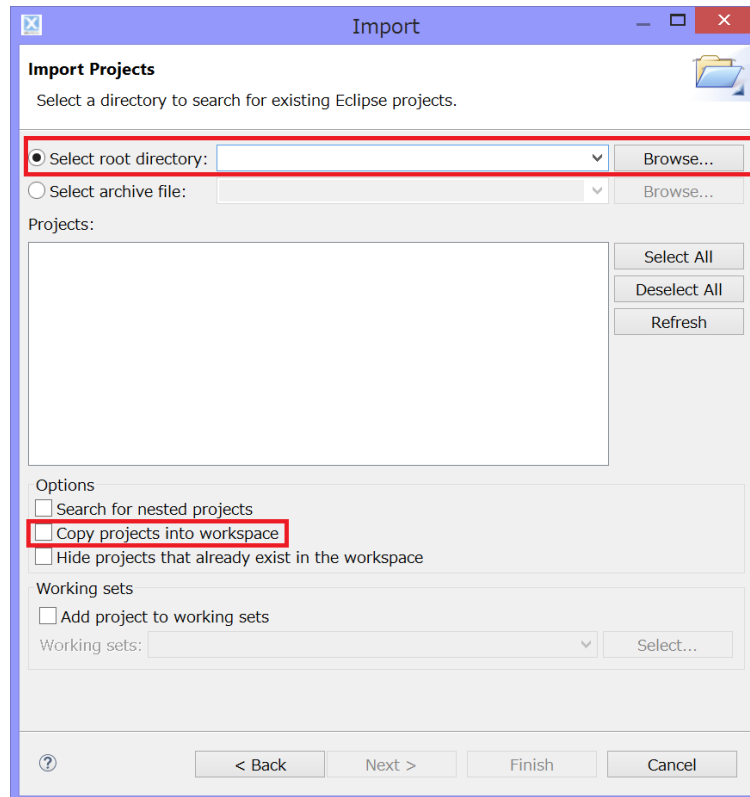
プログラム本体は「RT-USB-9AXIS-00_20150608」ですが、「CMSISv1p30_LPC13xx」にあるライブラリファイルを読み込むため両方のプロジェクトがインポートされている必要があるので注意してください。

サンプルプログラムのインポートの手順は以下のようになります。

- LPCXpressoを起動
- File>>Import>>General>>Existing Projects into Workspaceを選択



- Select root directory>>Browse でインポートしたいディレクトリを選択
- LPCXpressoのworkspace内にプロジェクトのコピーを作成する場合は OptionsのCopy projects into workspaceにチェックを入れる
- Finishをクリック



- 2つのプロジェクトをインポートした状態でctrl + B (build all)を押し、エラーが出ずコンパイルが通ればインポート完了です。

2. UARTの通信ボーレート変更方法

main.c内のUARTInit()内部を変更。デフォルトでは57600bps。
下図の赤枠内

3. USB接続のタイムアウト時間変更方法

main.c内のmain()内部を変更。デフォルトでは5000msecの待ち時間。
下図の青枠内

```

int main (void)
{
    volatile unsigned long i = 0;

    wait1msec(200);

    LPC_SYSCON->SYSAHBCLKCTRL |= ((1<<16) | (1<<14)); //EN IOCON, EN USBREG

    Init_IO();           // IO Initialization
    LED(1);

    USBIOClkConfig();
    Init_Timer32_0();    // TIMER32_0 Initialization
    Init_Timer32_1();    // TIMER32_1 Initialization
    VCOM_Init();         // VCOM Initialization
    USB_Init();          // USB Initialization
    UARTInit(57600);

    USB_Connect(TRUE);  // USB Connect
    while (!USB_Configuration) {
        if(i >= 5000) {
            while (!I2CInit((uint32_t)I2CMaster)); // wait until I2C is configured
            wait1usec(1000);
            initMPU6050();
            initAK8975();
            wait1usec(1000);
            LPC_TMR32B0->TCR = 1; // start timer0 interrupt
            USB_Connect(FALSE);
            while(1);
        }
        wait1usec(1000);
        i++;
    }; // wait until USB is configured

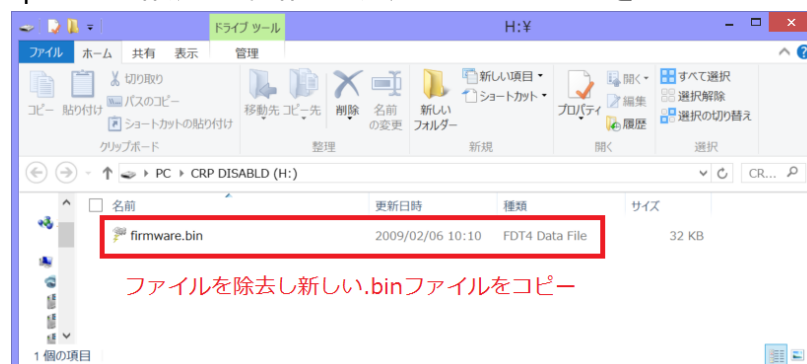
    while (!I2CInit((uint32_t)I2CMaster)); // wait until I2C is configured
    wait1usec(1000);
    initMPU6050();
    initAK8975();
    wait1usec(1000);
}

```

8. ファームウェアの書き込み

初期状態では9軸センサモジュール上のLPC1343に既に9軸センサの出力を取得するためのファームウェアが書き込まれています。ここでは自作プログラムの書き込み方法を説明します。

1. 9軸センサモジュール上のタクトスイッチを押したままUSBケーブルを接続
このとき、モジュール上のLEDが点灯します。
2. タクトスイッチから手を離します。
3. ブートローダーの起動まで待機
4. もともののfirmware.binを削除
5. LPCXpressoで作成した自作プログラムの.binファイルをコピー



以上でファームウェアの書き込みは完了です。

9. Linuxサンプルプログラムについて

Linux環境でのサンプルプログラムについて解説します。動作環境としてはUbuntuを想定します。サンプルプログラムはLinux Sample Program内のusb9axis_test.cになります。

実行手順

1. USBケーブルでセンサモジュールとPCを接続
2. /dev配下のttyから始まるファイルで、センサモジュールを接続して新しく表示されたファイルを記憶
/dev/ttyACM0で認識されることが多い
(Mac環境などでは他の名前の場合もあり)
コマンド: `ls /dev/tty*`
3. usb9axis_test.c内のstatic char serial_port[128]という箇所を編集
static char serial_port[128]="手順2で記憶したデバイスファイルのパス"
static char serial_port[128="/dev/ttyACM0"
4. gccでusb9axis_test.cをコンパイル
usb9axis_test.cが/hogeというPathにあるとして
コマンド: `gcc /hoge/usb9axis_test.c`
a.outというファイルができます。
5. 手順4でできた実行ファイルを管理者権限で実行
コマンド: `sudo /hoge/a.out`

実行結果

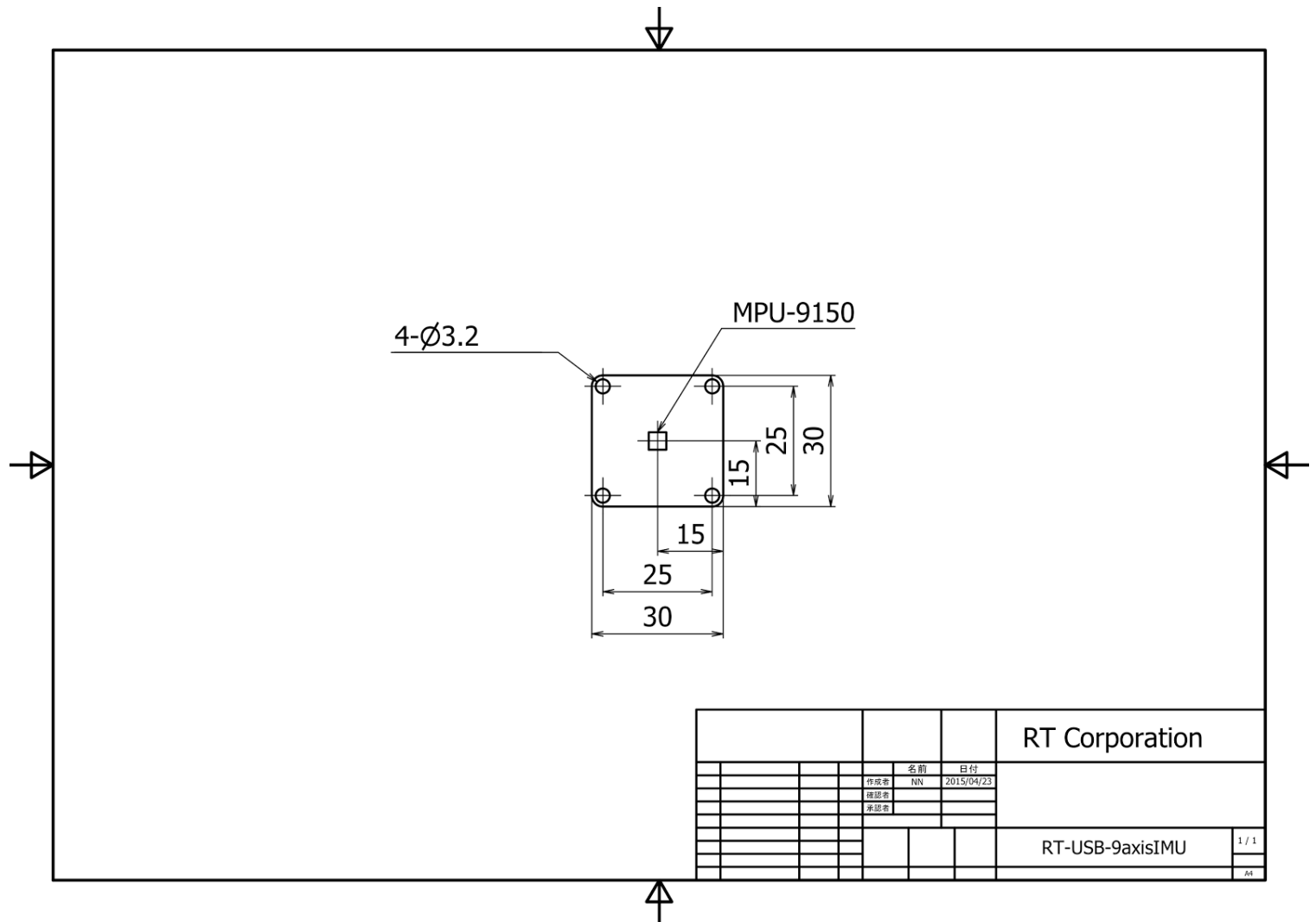
9軸センサモジュールから出力される28byteのデータを各byteずつ画面上の表示し続けます。プログラムはCtrl + cで終了させてください。

```

10|e8 ||
11|2 ||
12|bf ||
13|fa ||
14|4f ||
15|f6 ||
16|f5 ||
17|0 ||
18|72 ||
19|fe ||
20|4 ||
21|ff ||
22|1 ||
23|0 ||
24|87 ||
25|ff ||
26|ad ||
27|ff ||
====len=28 byte====
10|ff ||
11|ff ||
12|52 ||
13|54 ||
14|39 ||
15|41 ||
16|4 ||
17|e0 ||
18|85 ||
19|4 ||
20|43 ||
21|3 ||
22|95 ||
23|fe ||
24|58 ||
25|f6 ||
26|18 ||
27|1 ||
28|ae ||
29|fe ||
30|dc ||
31|fe ||
32|fe ||
33|ff ||
34|8a ||
35|ff ||
36|ae ||
37|ff ||
turtlebot@turtlebot-ThinkPad-X201s:~/takahashi/c_test$

```


10. 図面



11. お問い合わせ

カスタム等も有料にて承っておりますので、お気軽にお問い合わせ下さい。
If you have any inquiries upon this product, please contact us at the following.

RT Corporation 株式会社アールティ

住所: 〒101-0021 東京都千代田区外神田3-2-13山口ビル3F

Address: 3F, 3-2-13 Sotokanda, Chiyodaku 101-0021, Tokyo, Japan

TEL +81-3-6666-2566 FAX +81-3-5809-5738

E-mail: shop@rt-net.jp

Open: 11:00a.m.- 18:00p.m. (JST+9)

Close: weekend, national holiday, summer vacation, new year

Copyright

All the company and product names in this document are trademarks or registered trademarks of their respective companies.

All the documents, photos, and illustrations are copyrighted and protected by the copyright law of Japan and overseas. All the contents in this document are not allowed to be uploaded to any public or local area networks such as the Internet without permission from RT Corporation.