

P I C を利用した L 3 G D 2 0 ジャイロセンサーの基本動作の確立

金野茂男

1. はじめに

とある目的での利用を考え、秋月電商で、「L 3 G D 2 0 使用 3 軸ジャイロセンサーモジュール」(価格 7 5 0 円) を購入した。モジュールに添付の使用案内書は、表裏の 2 頁のみ。ダウンロードできる L 3 G D 2 0 のテクニカルノートは英文で 4 4 頁にわたる。ネットで、このセンサーと P I C を連携させた製作例を探してみたが、役に立つような例は見つけることができなかつた。そのため、添付されていた「不十分な」使用案内書と、やっぴいな英文のテクニカルノートを参考にして、漸く、P I C とジャイロセンサーを連携させることができた。又、装置の動作を正しく確認する必要性のこともあり、取得したデータ等のモニター用として 2 0 文字× 4 行の L C D も付属させた。

P I C とジャイロセンサーの合体はできたのであるが、ジャイロ及び P I C から出力されるデータはデジタルデータである。付属させた L C D で、これらの数値列をモニターすることもできるようにはした。が、センサーからの角速度の値は、2 進数 1 6 ビットの数値列である。これでは、視認しにくい。その対応策として、アナログ電圧値に変換する回路も付加すると、装置の動作の確認、及び、より広い使用性があると考え、デジタル・アナログ (D A) 変換器を追加することとした。写真 1 に、「一応」完成した装置の、試験動作中における外観を示している。

当面、本システムの基本動作の確立を行った。ジャイロセンサーから、x、y、z の 3 軸周りの回転角速度のデータを読み出し、必要ならば、各軸周りの、符号付き 2 進数 1 6 ビットデータを 2 0 文字× 4 行の L C D に表示する。しかし、1 と 0 の羅列である 1 6 ビットデータは、「人間的」には見難い。このデータを、必要ならば、符号付き 5 桁 1 0 進数に変換したデータも L C D に表示できるようにした。また、D A 変換器を付加したので、ジャイロセンサーの出力する角速度を、正負のアナログ電圧として、出力できるようにもした。これも必要ならば、この出力電圧をオシロスコープなどでモニターすれば良い。

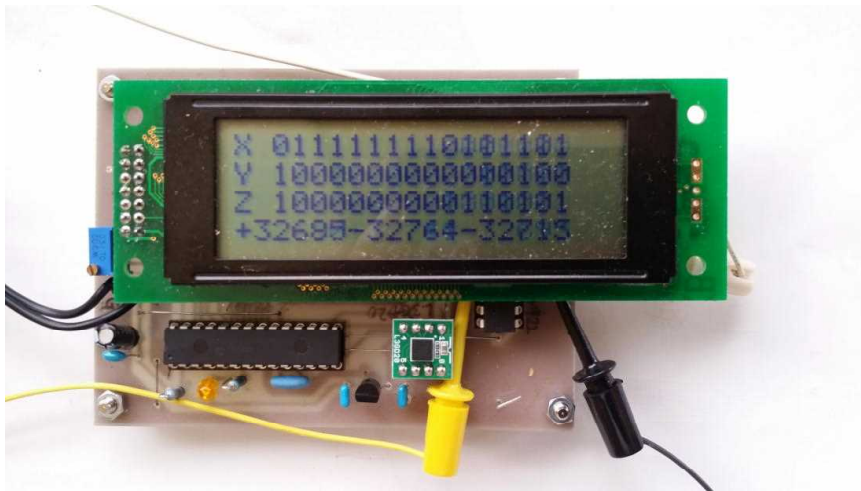


写真 1 試作品の外観。

今後の応用のことも考え、正負絶対値 2 進数 1 6 ビットデータ (なを、L 3 G D 2 0 からは、角速度の負値は 2 の補数形式で送り出される) の加減算ルーチン、及び符号付き 5 桁 1 0 進数の加減算ルーチンも書き上げた。早すぎる感もあるが、この時点で、仕事の成果を公表することにした。ジャイロセンサー L 3 G D 2 0 を、及び 1 2 ビット D A 変換素子である M C P 4 9 2 2 を P I C で駆動したいと考えている読者に役立てば幸いである。

なを、ジャイロセンサーだけ (= L C D 表示をしない) のサンプリングの繰り返しならば、1 k H z 以上でのサンプリングが可能である。が、L C D 全面にデータを表示すると、L C D 駆動には時間がかかり、サンプリング周波数は 1 0 H z 以下となる。しかし、提供するアセンブラプログラムに、それなりの改良、或いは修正を加えれば、少しは早くなるう。

写真1に試験中の試作品の外観を示している。上半分が20文字×4行のLCD, 左下がPIC, 中央下がジャイロセンサー、その右上に、LCDにほぼ隠されているDA変換器。LCDには、ジャイロセンサーから得られた角速度の値である、x軸、y軸、z軸の16ビット・デジタルデータが表示されている。DA変換用に変換した値である。ジャイロセンサーは平衡状態にある。即ち、出力は電圧レベルで0Vなので、x、y、z軸とも、16ビットの中間値当たりの値を表示し続けている。最下行には、これら16ビットデータを10進数に変換した値が表示されている。なを、この画面で、表示されている±符号は無視すること。静止状態、即ち回転していないので、センサーからのデジタル出力は、16ビットの中間値1000000000000000当たりであり、10進数表記では32768当たりにあることが視認できよう。数値列の下位ビットにはバラツキが顕著であることがわかる。主にノイズである。

2. 設計・製作

図1に試作回路のブロック図を示している。使用するジャイロセンサーは3軸のL3GD20, ワンチップマイコンはPIC18F252, LCDは4行×20文字のSC2004CS*B, DAはシリアル入力・12ビットのMCP4922である。

ジャイロセンサーのデジタル出力データは正負符号付き16ビットなので、それに対応したDA変換器としては、16ビットで且つ正負電圧駆動の物が宜しいのであるが、多分に高価である。制作中に、ジャイロからの16ビット出力数値をLCDで視認してみた。下位の4ビット～6ビットあたりまでは、ノイズレベルでありそうであった。このことは、写真1で現認できる。従って、無理に16ビットDA変換器は必要はないと判断した。つまり、下位4ビットはDA変換には不要と判断した。また、入手が容易であり、且つ安価なこともあり、正電圧駆動の12ビットDA変換器を使用することにした。このDA変換器からの出力は正電圧である。ジャイロからの正負の出力電圧に対応していない。そのため、DA変換器の後段に、クランプ回路を付加し、正負のアナログ電圧が出力されるようにした。

MCP4922の通信インターフェースはSPIモードである。従って、PICもL3GD20もSPIモードで使用することにした。なを、PICとL3GD20の通信インターフェースはI2Cモードにも対応している。

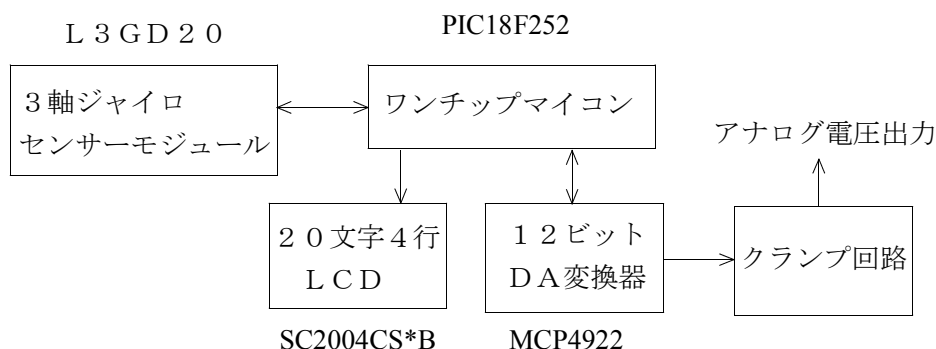


図1 試作品のブロック図

図2が回路図である。まず、電源電圧について。L3GD20の電源電圧は2.4V～3.6V、PIC18F252の電源電圧は2.0V～5.5V、LCDであるSC2004CS*Bの電源電圧は5V、MCP4922の電源電圧は2.7V～5.5V。従ってまず、LCDとPICには5Vを供給し、L3GD20とMCP4922には3.3Vを供給することにした。PICとL3GD20とMCP4922の間のロジック電圧に差が生ずるが、何の不調も無く動作することは確認した。OPアンプ使用のクランプ回路用には±5Vの電源を供給する。

以上を考慮して、主電源として、+9V出力AC/DCアダプタを使用し、2電圧出力モジュールであるMCW03-05D05で±5Vを供給することとした。この+5Vから、3端子レギュレータで+3.3Vを得ている。

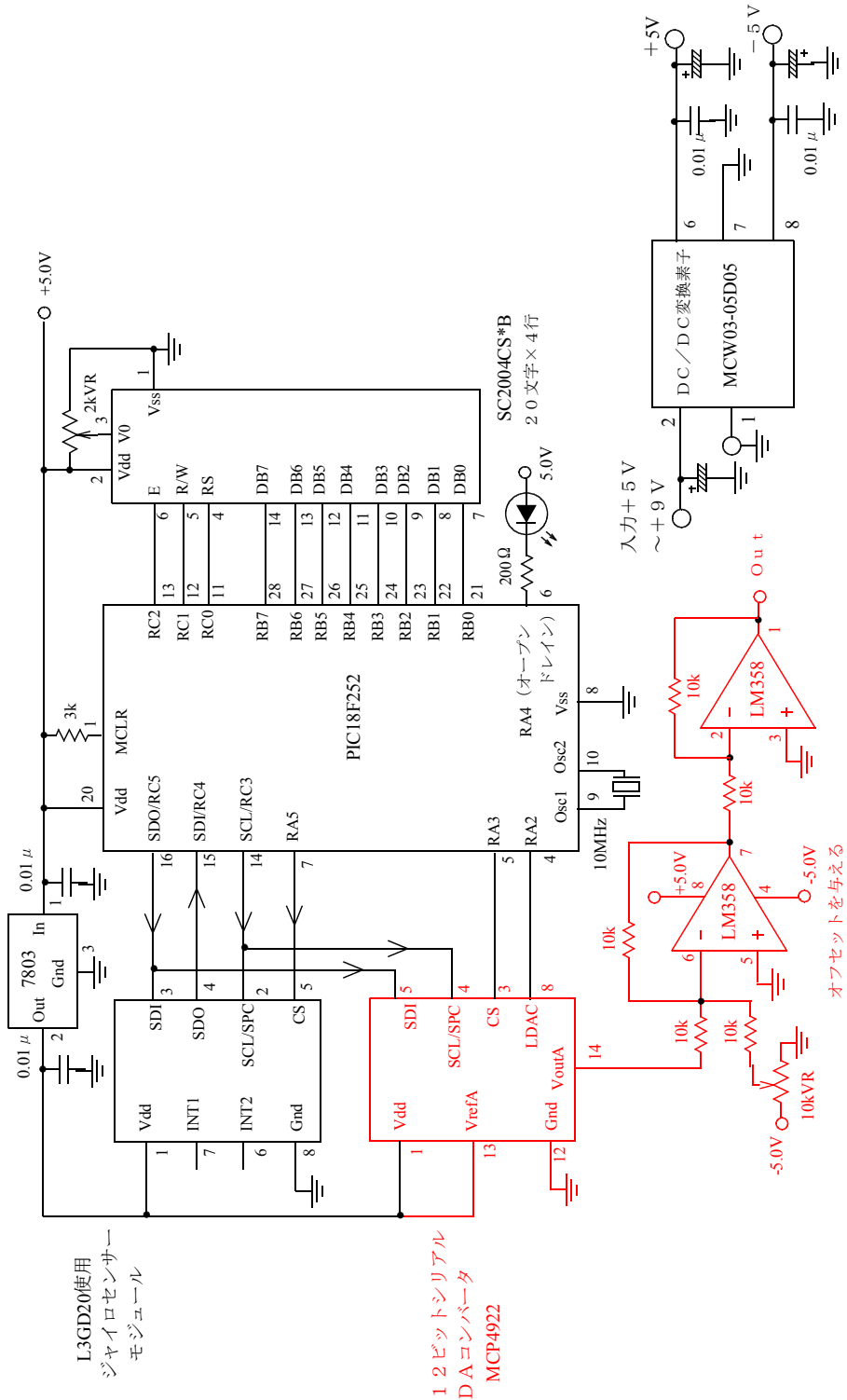


図2 試作品の回路図

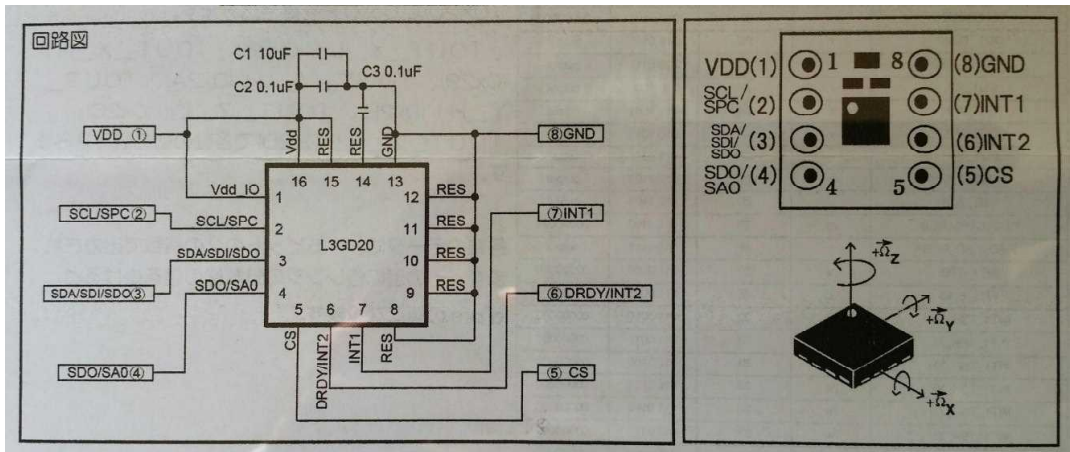


図3 ジャイロセンサーL3GD20とそのモジュール図

図3には、購入時ジャイロセンサーに付属していた使用書の一部を複製して掲載している。ジャイロセンサー自体は4mm×4mm×1mmの大きさである。写真1を参照。本体は表面実装型であるが、DIP基板に取り付けられており、使用は容易である。

ジャイロからの角速度データは16ビットであるが、DA変換器の変換ビットは12ビット。4ビット足りないのが気になる。しかし、ジャイロの試験運転において、16ビットデータと言いながら、その下位4ビットから6ビットあたりまでは、雑音成分が大きいことに気がついた。と言うことは、得られる16ビットデータの下位4ビットは捨て去り、残りの12ビットで事足りると判断した。この判断以外に、16ビットのDA変換器はそう安くはない、し簡単に入手もできない。かつ、正負の出力をしてくれる16ビット変換器は目が飛び出るほどの値段でもあろう。本システムで使用することとした、12ビットDA変換器であるMCP4922は安価で、簡単に入手することができたのである。秋月電子通商で¥200円。

DA変換において、デジタル入力12ビットデータ[0000 00000000]はアナログ出力電圧約0.0V、[1111 11111111]は約3.3Vに対応している。ジャイロからは符号付きの16ビットデータが出力されるので、アナログ出力電圧も、ジャイロの正負に対応させたい。12ビットデジタルデータの間値は[1000 00000000]である。これをアナログ出力電圧の0.0Vとすれば、[0000 00000000]から[0111 11111111]の範囲を負電圧出力に、[1000 00000001]から[1111 11111111]の範囲を正電圧に対応させることができる。DA変換器からの出力をクランプ回路に送り、これを実現している。図2中のOPアンプの近傍にある10kVRが、このクランプ電圧調整用の可変抵抗である。DA変換器へのデジタルデータを[1000 00000000]としておき、その時のOutでの電圧が0.0Vとなるように、10kVRを調整することになる。

なを、本回路での調整部分は、この10kVRとLCDの輝度調整用の可変抵抗2kVRだけである。

図4に、PICに書き込んだアセンブラプログラムのフローチャートを示す。以下のような流れである。

(1) 電源を入れると、LCD画面に「**Kinnno Labo**」、「Jairo Sencer」が表示される。この画面が出れば、プログラムのインストールは正しく行われ、ハード回路も一応正常に動作していることを確認できる。

(2) PICはジャイロとDA変換器とのSPI通信の確立を行った後、ジャイロを能動状態とする。

(3) ある程度のウォーミングアップを行う。

(4) PICはジャイロから3軸のデータを順次取り込む。

(5) ジャイロのオフセット値を確定するために、3軸のデータをある程度の回数取り込みを繰り返し、その平均値を各軸のオフセット値として確定する。

(6) これ以降、ジャイロから得られる「現値」に、オフセット値を処置して、「真値」を求めることを繰り返す。必要ならば、「真値」を積算し、変位角度に比例した値も得ることができる。現在は、短時間内であるが。この積算値については後述する。

(D A出力選択可)は、その時点におけるデジタルデータをD A変換し、アナログ電圧に変換した出力を得ることができることを意味している。また、要所要所で、LCDにデータ列を表示することもできるようにしている。ただ、LCD動作には時間がかかる。LCD動作をするかどうかは、その命令文を実行行とするか、注釈行にするかで簡単に選択することができるようにしている。

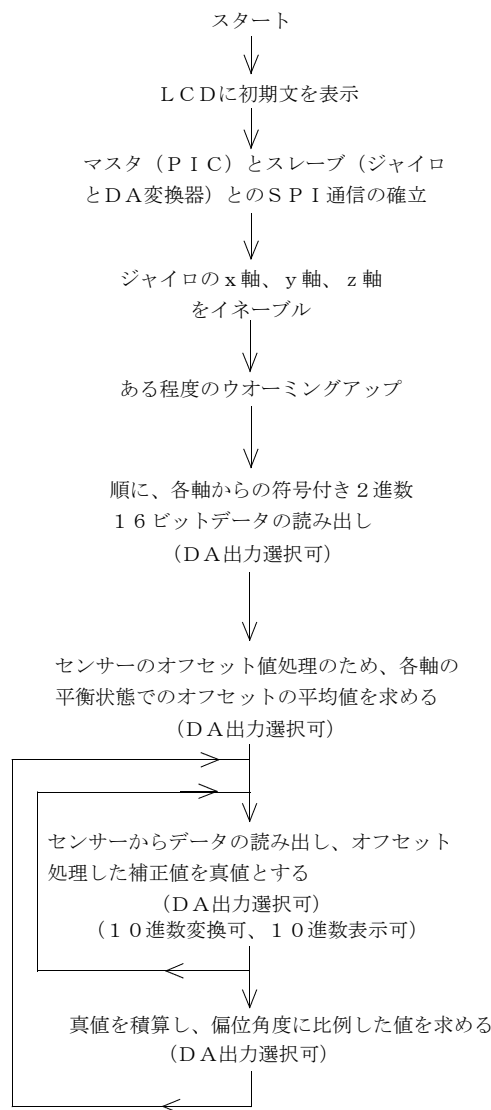


図4 PICに書き込んだアセンブラプログラムのフローチャート

現在、以上のような単純な流れで動作する。また、幾つかの応用ルーチンも書いている。

(1) ジャイロからの符号付き2進数16ビットデータを、符号付き10進数5桁に変換する。

(2) 符号付き5桁をLCDに表示する。写真1で、4行目でこの10進数数値を視認できよう。

(3) 符号付き5桁10進数を加算及び減算をする。

(4) 符号付き2進数16ビットを、絶対値2進数16ビットとする。DA変換器は12ビットデータ変換の物なので、この16ビットデータの下位4ビットを捨て、12ビットを右詰とし、上位4ビットにDA変換器のための制御ビットを付加する。そして、DA変換器に出力する。図5中で簡単に解説をしている。何かくどいようであるが、用は、DA変換器は入力ポートでシリアル16ビット列を受け入れる。その列の中で、制御ビットの4ビットを16ビットの上位ビットとし、残りの12ビットにデータを配当している。この事については、DA変換器の仕様書を参考すると良い。

取得データをDA変換のための処理

正值の時は、最左端ビットを1、

負値の時は、最左端ビットを0

とし、16ビットで絶対値化



16ビットデータの内、下位4ビットを捨て、
上位12ビットを2バイトに、右詰とする。

12ビットDA変換器を使用するためである。



DAに送り出す2バイトデータの内、
左端の4ビット分はDAのための制御ビット
なので必要なデータとする。

図5 DA変換器ための処理手順。詳細はMCP4922のテクニカルノートを参照すること。

ジャイロからの16ビットデータは、負数は2の補数表記である。これは、見難い上に、現プログラムでは、結構処理に面倒である。そのため、ジャイロから得られるこの負数の2の補数表記を、直ぐに符号付き絶対値表記に変換した。最左端の16ビット目を「1」とした。正数の場合は16ビット目はそのまま「0」である。

アセンブラプログラムは、添付しているが、その主要部分を図6に複写しておく。図4のフローチャートと対比すると良い。注釈文を理解すれば、各サブルーチンを実行サブルーチンに、或いは、注釈サブルーチンに自在に変更することで、希望する動作を選択できよう。

3. 調整・特性

本システムのハード的な調整箇所は記述したとおり、LCDの輝度調整用の可変抵抗と、クランプ電圧調整用の可変抵抗の2箇所だけである。DA変換を必要としなければ、クランプ回路は必要ないので、そのための可変抵抗も必要なく、OPアンプも不用となる。本システムが正常に動作をすることが確認できたならば、場合によっては、LCDも不用とすることができる。それぞれである。

ハード回路を製作し、PICにプログラムをインストールして、電源を入れると、LCD画面に初期メッセージが表示される。それ以降の動作は、図6に示しているアセンブラプログラムの主要部分で、種々に設定することができる。とにかく、ジャイロからデータを取り込み、処理をし、必要ならばLCDにデータを表示させたり、DAから出力電圧を得ることができる。プログラムに記載している注釈を読み込み、希望する動作を選択すれば良い。が、LCDの動作には時間がかかる。LCDを使用する場合には、そのことを承知しておく必要がある。それに対して、DA変換には、殆ど時間がかからない。しかし、DAからの出力電圧をモニターするにはオシロスコープなどが必要である。

LCDとPICの電源電圧は仕様に従い+5V、ジャイロとDA変換器の電源電圧は仕様に従って+3.3Vにした。回路図を見れば、+3.3V用3端子レギュレータからこの+3.3Vが供給されていることがわらう。が、回路の結線を完了すると、この+3.3Vは+4V当たりまで上昇していることに気がついた。PICのI/O端子は、これら

ジャイロとDA変換器に接続している。PICから、このI/Oポートを経由して、+5Vの電圧がかかることになる。それによる影響であろうと思うが、その原因は理解できなかった。しかし、このような状態でも、回路は何の問題もなく正常に動作し続けるので、「よし」とした。

```

;ジャイロセンサーとのSPI通信の確立
    call jairo_comm_start ;
;センサーのX軸、Y軸、Z軸をイネーブルとする
    call xyz_enable ;
;ウォーミングアップ
    movlw 60
    movwf warmup_counter ;
warmup_next30
    btg LED_monitor ;
    call wait_1sec
    decfsz warmup_counter,1 ;
    goto warmup_next30 ;
;平衡状態としているセンサーからX軸、Y軸、Z軸のデータを多数回読み出し、平均値を出し、
    call offset_value_get ;オフセット値を決める
;call offset_value_disp ;算出したオフセット値をLCDに表示
    goto test_jump9
;センサーからX軸、Y軸、Z軸のデータを読み出すことをエンドレスで行う。
comm_repeat_1
    btg LED_monitor ;
    call home_out ;カーソルホーム
    call read_data_repeat ;2進数符号付き16ビットのデータをジャイロから読み出す
    call conv_comp_abs_xyz ;x、y、zの補数16ビットデータを符号付き絶対値16ビットに変換する
;call disp_xyz_16bit ;変換し符号付き絶対値16ビットをLCDに表示する
    call offset_manage ;得られたジャイロの現値に、オフセット処理をして、真値(≒補正值)とする
;call real_value_disp ;真値をLCDに表示する
    call sum_degree ;角速度の総和と、1軸だけのアナログ電圧出力
    call wait_10msec ;サンプリング時間間隔を設定
    goto comm_repeat_1

test_jump9
;センサーからX軸、Y軸、Z軸の角速度データを読み出すことをエンドレスで行う。
;データをLCD表示するか、DA電圧出力するかは選択可、その他も選択可
comm_repeat_2
    btg LED_monitor ;
    call home_out ;カーソルホーム
    call read_data_repeat ;2進数符号付き16ビットのデータをジャイロから読み出す
;call disp_xyz_16bit ;x、y、z軸の16ビットデータをLCDに表示する
    call da_data_creat ;2の補数表示を、DA変換用に絶対値表示とする
    call DA_out_xyz ;絶対値化した各軸のデータをDA変換し、Voutに出力
;call DA_out_sample ;DA変換器での出力例
;call conv_16bit_5dec_LCD_disp ;各軸の2の補数2進数符号付き16ビットデータを
; ;符号付き5桁10進数に変換し、LCDに表示。

;call offset_value_get ;オフセット値を求める
;call sum_degree ;角速度の総和と、1軸だけのアナログ電圧出力
    call wait_10msec ;サンプリング時間間隔を設定
    goto comm_repeat_2

```

図6 アセンブラプログラムの主要部分

ジャイロからの3軸の2進数2の補数16ビットデジタルデータ、及び、それらの10進数5桁を20文字×4行のLCDに表示するには、結構時間がかかる。1回のサンプリングデータを表示しきるのに150msもかかっている。動作検査などには、LCDでのデータ表示は必要である。正常に動作することを確認したならば、LCDでの表示を飛ばすのも良いであろう。LCDでの表示を止めれば、今のところ、大凡1ms毎にジャイロのデータをサンプリングし続けることができている。つまり1kHzでのサンプリング動作が可能である。

DA出力は、現プログラムではx軸のみとなっている。y軸、z軸について出力したければ、現プログラムではそれらの関する箇所は注釈行としている。従って、x軸に関する

箇所を注釈行とし、y 軸、或いは z 軸に関する箇所を、実行行とすればよい。実は、使用している D A 変換器 M C P 4 9 2 2 は 2 チャンネルの素子である。そのうちの 1 チャンネルしか、使用していない。必要ならば、2 ちゃん得るも使用できるようにプログラムを修正すればよい。従って、x 軸、y 軸、z 軸の内の 2 軸を選択し、D A 変換器からの出力を 2 出力とすることができる。できれば、3 軸を同時に、出力できるようにすればよい気がするが、その場合には、3 チャンネルのオシロスコープや、レコーダーが必要となる。

5. 終わりに

本システムが正常に動作することが確認できれば、これを基礎として、L C D, D A 変換器、クランプ回路が不用なシステムとすることができる。そうすれば、本システムは極めてコンパクトなシステムとなる。P I C 1 8 F 2 5 2 も表面実装型の相当品に置き換えることもできよう。

ジャイロから得られるデータは、回転を受けたジャイロの 3 軸周りの角速度である。角速度なので、これを時間積算すれば、ジャイロの偏移した角度が、計算式上では、得られるはずである。 $\Delta \omega_i = i$ 番目の角速度、 $\tau =$ 測定時間間隔、 $\Delta \theta =$ 偏移角度として、

$$\Delta \theta = \text{SUM} (\Delta \omega_i \cdot \tau)$$

しかし、これはオフセット、及びドリフトに敏感である。オフセットは極限まで 0 値にしておく必要があり、ドリフトも極限までゼロとする必要がある。そうでなければ、これらの値も時間積算されてしまうからである。

角速度を積算するサブルーチンを書いて、その D A 出力を検討し続けている。現在以下のような状況である。

(1) オフセット処理は必須である。ジャイロからは平衡状態でも、0 V ではない、オフセット電圧が出力されている。このオフセット値の処理をしないと、得られる「偏移角度」のドリフトは非常に大きくなる。

オフセット処理を行った時には、サンプリング周波数にも依存するが、得られる「偏移角度」のドリフトは、結構小さくなる。

なを、オフセット値は何回かのサンプリングで得られるデータを x 軸、屋軸、z 軸各々毎に積算し、それらの平均値を各軸のオフセット値としている。

(2) 本システムは、ウォーミングアップを十分にしておき、ある程度の時間内無いならば、偏移角度の測定には使用できそうである。長時間での使用を考えるならば、工夫が必要である。

2 0 1 5 年 4 月