短期連載

ARMプロセッサを使用した ロボット制御システムの製作

システムの記事　ARMの記事

第3回　GPSモジュールの値をパソコンに取り込む

操田浩之

本誌2008年5月号に付属したARMプロセッサ基板の応用例を紹介する連載である．第3回目の今回は，ロボットに搭載したGPSモジュールのデータを，ZigBeeモジュール経由でパソコンに取り込む方法について解説する．　（編集部）

第1回（本誌2008年6月号，pp.123-132）で紹介したロボット制御システムは，Rコントローラ，GPSモジュールを含むセンサ・モジュール，Rコントローラとパソコンを無線でつなぐZigBeeモジュール，パソコン用制御・プログラムで構成されています．

第1回ではRS-485を使ったサーボモータの制御について説明しました．第2回はDMA（Direct Memory Access）を利用して，センサ・モジュールから連続して出力されてくるアナログ・データを効率良くメモリに取り込む方法について説明しました．

第3回となる今回は，ロボットに搭載するGPSモジュールのデータをパソコンに取り込みます（**写真1**）．さらにZigBeeモジュールを介したパソコンとRコントローラとの通信について説明します．

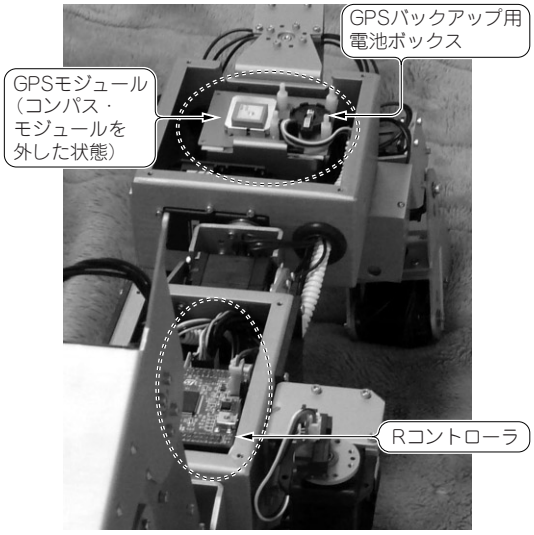

写真1　ロボットに搭載されたGPSモジュールとRコントローラ

1　GPSモジュールのデータをRコントローラに取り込む

● GPSモジュールとRコントローラを接続

初めにGPSモジュールとRコントローラとの接続について説明します．GPSモジュール（ポジションの「GPS-52」）のからのシリアル・データはASCII形式の文字列で出力されます．通信速度は，初期状態で9600bpsです．必要なときにデータを読み出すのではなく，1秒周期で常にデータが送られてきます．

送られてくるデータのフォーマットは，$GPで始まるメッセージIDとフィールド，チェックサム，ターミネータ（CR + LF）となっていて，それぞれカンマ「,」で区切られます．フィールドはメッセージIDによって異なる内容に分けられ，同じくカンマで区切られています．

メッセージIDの種類は$GPGGA，$GPGSA，$GPGSV，$GPRMC，$GPVTG，$GPZDAの6種類です．今回は

Keyword　GPSモジュール，Rコントローラ，XBee，ZigBeeモジュール，ZigBee-USB変換モジュール，Visual C#

$GPGGA だけを使用しました．$GPGGA には世界協定時，緯度，北緯/南緯，経度，東経/西経，測位状態，使用衛星数などの情報が含まれています．

GPS モジュールには時間などの受信データをバックアップするためのボタン電池を接続できるので，コンパス・モジュールの下に設置しています．

● DMA を利用してデータを繰り返し保存する

送られてきたデータは DMA を利用して 512 バイトの配列変数 GPS_DATA[] に保存しました．512 バイトを超えるデータについては再度，GPS_DATA[] の先頭番地から保存し，これを繰り返します．DMA を利用した処理イメージを**図1**に示します．

GPS モジュールのデータが欲しいときには，この 512 バイトの文字列の中から '$' で始まる文字列を検索し，今回使用する $GPGGA という緯度や経度の情報が含まれる 1 行分を取り出します．

センサ情報の A-D 変換値の保存と似ていますが，センサ情報の場合は配列の同じ位置には同じセンサからのデータが保存されるので，1 周期遅れたデータであってもあまり問題になりません．GPS モジュールの場合は，出力するデータ長が固定ではないためデータが不連続になってしまいます．

例えば配列変数の容量が少なければ，DMA が行うデータ更新を追い越してデータを読んでしまう可能性があります．データ更新の周期が短ければ，読み出すタイミングと更新のタイミングが難しく，配列を大きくするなどの対策が必要です．今回の実験では，512 バイトあればほぼ $GPGGA の行を正しく読み出せました．更新周期が 1 秒と長く，通信速度も 9600bps と比較的遅いことが良かったのだと思います．念のために 1 行分のデータの大きさを調べて，極端に短いものや長いものは使用しないようにしました．DMA 転送完了割り込みなどを併用してデータの書き込み完了アドレスなどを管理する方法もあります．

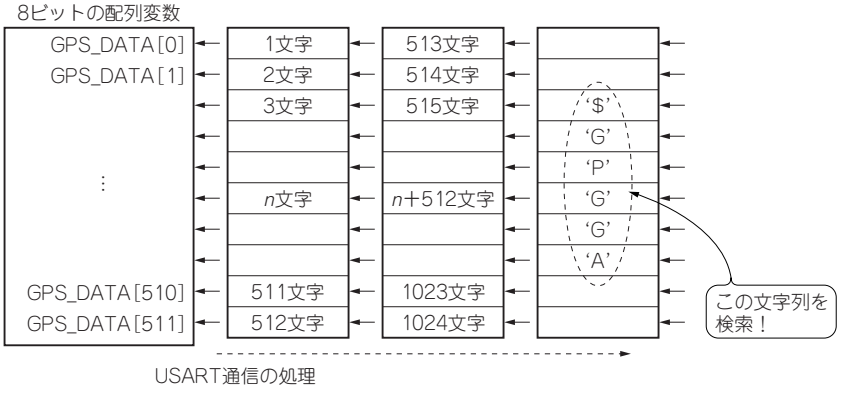

図1
DMA を利用して GPS モジュールからのデータを繰り返し保存する処理
GPS モジュールのデータが欲しいときには，この 512 バイトの文字列の中から '$' で始まる文字列を検索し，今回使用する $GPGGA という緯度や経度の情報が含まれる 1 行分を取り出す．

リスト1　DMA 転送に関係する初期化と実行

```
void dma_Init(void)
{  RCC->AHBENR |= 0x00000001;// clock for DMA
   DMA_Channel1->CMAR  = (unsigned long)&ADC_DATA;
   DMA_Channel1->CPAR  = (unsigned long)&(ADC1->DR);
   DMA_Channel1->CNDTR = 10;// transmit 10 word
   DMA_Channel1->CCR   = 0x000015A1;//0b0000 0000 0000 0000 0001 0101 1010 0001
   DMA_Channel5->CMAR  = (unsigned long)&GPS_DATA;
   DMA_Channel5->CPAR  = (unsigned long)&(USART1->DR);
   DMA_Channel5->CNDTR = 512;// transmit 512 byte
   DMA_Channel5->CCR   = 0x000020A1;//0b0000 0000 0000 0000 0010 0000 1010 0001
}

void com_Init(void)
{  unsigned long i;
   for(i = 0; i < 512; i++)
   {  GPS_DATA[i] = '?';
   }
   USART1->CR3  |=  0x00000040;//0b0000 0000 0000 0000 0000 0000 0100 0000 USART1 DMA enable
}
```

A-D変換用DMAチャネル1の設定
（前回説明分）
なお，コメント部の0bはバイナリ・データを表すものとした

USART用DMAチャネル5の設定
- 送信データ：512バイト
- アドレス更新：サーキュラ・モード
- 優先度：高
- 転送先：GPS_DATA[]

GPS用シリアル・ポートの配列変数を初期化した後，DMAを開始する