

システム仕様 (RL78/G11 での制御編)

第3段の今回は、カムプログラムロボットの制御を行います。

1. システムの概略仕様

CQ 出版の C-First のライントレースカーを参考にして、タミヤの「カムプログラムロボット」の制御を行います。

「模型の組み立て」で示した以下の表の信号を使用します。RL78/G11 のターゲット部では、ボード上のパッドを使用しない場合には、E20B とのインタフェース信号を除くと、使える信号はこれで全てです。

P21/ANI1	フォトリフレクタ	P31/TKB00	AIN2:AENBL
P20/ANI0	フォトリフレクタ	P30/TKB01	BIN2:BENBL
P22/PGAI	PGA 入力 (予約)	P53	AIN1:APHASE
P56/ANI22	フォトリフレクタ	P52	BIN1:BPHASE
P33	フォトリフレクタ制御	P51	MODE (ハイレベル)

最終的な動作としては、

①圧電スピーカー (サウンダー) を音声入力に使用して、手をたたくことで動作開始のトリガに使用するつもりです。このために、RL78/G11 の PGA (プログラマブル・ゲイン・アンプ) を用いて増幅し、その出力をコンパレータに入力します。コンパレータの比較電圧は、内蔵の D/A コンバータの出力を使用し、比較電圧以上なら起動する (スタンバイを解除する)。

RL78/G11 の STOP モード時の動作状況 (下記に示す) を確認すると、D/A コンバータには STOP モードに入る前の状態を保持とあります。コンパレータもデジタルフィルタ未使用時に動作可能、コンパレータも動作可能とあるので、見えそうです。

表 24 - 2 STOP モード時の動作状態

項目	STOPモードの設定			メイン・システム・クロックでCPU動作中のSTOP命令実行時		
	高速オンチップ・オシレータ・クロック (f _{hi})で CPU動作時	中速オンチップ・オシレータ・クロック (f _{mi})で CPU動作時	X1クロック (f _x)で CPU動作時			
システム・クロック	CPUへのクロック供給は停止					
メイン・システム・クロック	f _{hi}	停止				
	f _{mi}	停止	停止	停止		
	f _x	動作可能				
	f _{ex}					
f _l	オプション・バイト(000C0H)のビット0 (WDSTBYON), ビット4 (WDTON)および動作スピー-I (OSMC)のWUTMMCK0ビットにて設定 WUTMMCK0 = 1またはSELLOSC = 1 : 発振 WUTMMCK0 = 0かつSELLOSC = 0かつWDTON = 0 : 停止 WUTMMCK0 = 0かつSELLOSC = 0かつWDTON = 1かつWDSTBYON = 1のとき : 発振 WUTMMCK0 = 0かつSELLOSC = 0かつWDTON = 1かつWDSTBYON = 0のとき : 停止					
CPU	動作停止					
コード・フラッシュ・メモリ						
データ・フラッシュ・メモリ						
RAM						
ポート(ラッチ)	STOPモード設定前の状態を継続					
タイマ・アレイ・ユニット	動作禁止					
タイマKB						
12ビット・インターバル・タイマ	カウント・クロックにf _{clk} 選択時は動作禁止, f _l 選択時は動作可能					
8ビット・インターバル・タイマ	動作可能					
ウォッチドッグ・タイマ	第12章 ウォッチドッグ・タイマ参照					
クロック出力/ブザー出力	カウント・クロックにサブシステム・クロック選択時は動作可能(それ以外は、動作禁止)					
10ビット A/D コンバータ	ウエイク・アップ動作可能(SNOOZEモードへ移行)					
D/A コンバータ	動作可能(STOPモード設定前の状態を継続)					
コンパレータ	動作可能(デジタルフィルタ未使用時のみ)					
プログラマブル・ゲイン・アンプ (PGA)	動作可能					

なお、動作中には振動で誤検出する可能性があるため、あくまで起動用の条件として使うことを考えます。

②フォトリフレクタを用いたライントレースを目標にしますが、キャタピラでの動作なので面白い動作かどうかは不明です。ライントレースのためのLEDの発光も、できるだけ電力消費を抑えるために、必要なタイミングだけにすることにします。

それよりも、信地旋回や超信地旋回の方が面白いかもしれません。ただし、この場合には、どの程度の角度曲がったかを検知する必要があります。正確に検知しないのであれば、時間で設定すればいいかもしれません。

③最終的に動作を停止させるためには、8bitITを16bitで使用して、5分で停止させることにします。

2. 使用するRL78/G11の機能

今回使用する機能は、以下ようになります。これだけの機能を使うことは、殆どないのではないかと思います。

- ・ タイマKBでのPWM出力
- ・ 8bitITによる長時間タイマ（5分）
- ・ A/Dコンバータ
- ・ D/Aコンバータ
- ・ PGA
- ・ コンパレータ
- ・ スタンバイ（STOP）機能
- ・ ポート機能

これ以外に、4MHzの中速オンチップ・オシレータも使ってみようかと思っています。また、RL78の一般的なタイマ（TAU）も何かに利用しようかと思っています。これらのハードウェアを使用することで、CPUでの処理はそれほど必要としなくなることが期待されます。

次は、コード生成を用いたプログラム作成を開始します。最初は、単純なモータ制御からスタートします。