ワンチップマイコンで構成するシーケンス制御機器の開発

木嶋　祐太*　本多　章作*

Development of a sequence controller using an one-chip microcomputer

KIJIMA Yuta* and HONDA Syosaku*

抄　　録

PLC 言語（シーケンス制御用プログラム）を PIC マイコン用のアセンブラ言語に変換するプログラムを開発した。プログラミングのしやすさ，処理速度の観点から，STL（SFC を基にして作成される PLC 言語）を変換できるようにした。開発の結果，STL で PIC マイコンを動かすことができるようになった。本研究は，シーケンス制御を組み込みシステムに統合することを見据えている。

1．緒　　言

マイクロプロセッサの高性能化に伴い，より高度な機能をソフトウエア開発で実現できるようになっている。FA システムの分野においては，高速応答を必要とする制御機能や大量データ処理機能が組み込みソフト開発の対象となっており，ソフトウエア開発の適用範囲が拡大している。

シーケンス制御は多くの現場で用いられている。現場技術者はそのノウハウをもっているので，シーケンス制御には保守管理がしやすいというメリットがある。シーケンス制御は，PLC が使用されることが多く，PLC 言語によるプログラミングが主流となっている。

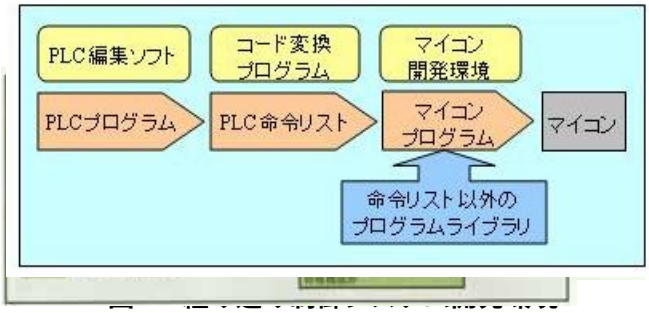
シーケンス制御を組み込みソフトで開発した場合，現場技術者が PLC 言語でプログラムを保守管理できなくなり，利便性が損なわれる。組み込みシステムにおいて，プログラムを PLC 言語で記述できる環境が構築できれば，現場の保守性を損なわずに他の制御機能と統合した組み込みシステムを構築できる。

本研究では，このような開発環境を構築することを目的として，PLC 言語ソースプログラムをマイコンで動作させる方法を検討し，開発した。

2．組み込み制御システム開発環境

本研究では STL をマイコン言語に変換するプログラムを作成したが，図 1 のような組み込み制御システム開発環境を想定した場合，変換プログラムはその一部として使用できる（図 1 のコード変換プログラム）。

制御システム開発者は，シーケンス制御プログラム（図 1 の PLC プログラム）を SFC とラダー図で記述する。それ以外の特殊制御機能やそれらの実行管理を統括するプログラム（図 1 の命令リスト以外のプログラムライブラリ）は C 言語などの構造化プログラム言語で記述する。それらは，マイコン開発環境でロードモジュール化されマイコンボードに書き込まれる。



* 上越技術支援センター

システム利用者（現場技術者等）は，必要に応じて SFC とラダー図を書き換え，制御システム開発者から提供されたマイコン開発環境でプログラムを再構築してマイコンに上書きできる。

3. マイコンによるシーケンス制御の実現法

3.1 PLC の動作

一般的に PLC の駆動方式はサイクリックスキャン方式，入出力方法はリフレッシュ（一括書き換え）方式が採用されることが多い。サイクリックスキャン方式では，図 2 のように入力要素読み込み，命令リストによる演算，出力要素書き出しを 1 サイクルとして順次繰り返される。

3.2 シーケンス制御の高速化

自動制御では，処理の遅れが致命的な誤動作を引き起こす可能性があるため，マルチタスク環境が想定される組み込みシステムでは，専用プロセッサを使用する PLC とは異なる独自の高速処理アルゴリズムを考案する必要がある。シーケンス制御の高速化には次の方法が考えられる。

(1) マイコン言語にアセンブリ言語を使用する。

シーケンス制御の命令リストはアセンブリ言語と同じくニーモニックを規定したもので，共に論理演算命令をもつため，命令リストからアセンブリ言語に変換することで冗長性が少なくなる。

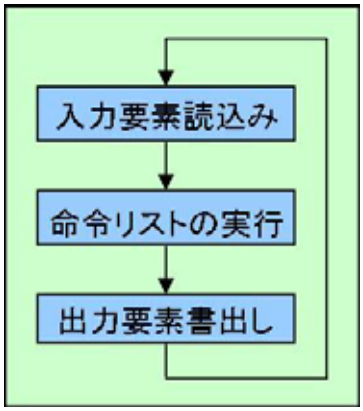


図 2 PLC のシーケンス制御フロー

(2)シーケンス制御に SFC を使用する。

SFC は図 3 のような図である。PLC では，この図を用いてシーケンス制御をプログラミングすることができる。

図中の四角がステップをあらわしている。それぞれのステップ毎にラダー命令が書かれ，そのステップの時だけそのラダー命令は有効になる。ステップとステップの間に条件が書かれていて，その条件が ON になると次（上から下へ）のステップに進む。

SFC では，有効になっているステップ以外は無視できるため，その分高速化が図られる。

(3)シーケンス制御の駆動をイベントドリブン型とする。

要素（入力要素，出力要素，タイマ要素，内部リレー要素など）変更時だけシーケンス制御命令をスキャンし，余分なスキャンサイクルを省くことで高速化を実現する。それぞれの状態に出てくる要素のみ変更を監視し，変更があった時だけ命令リストの実行，出力要素書き出し（図 2 参照）を行う。

4. コード変換プログラムの試作と評価

4.1 コード変換プログラム試作

STL を PIC マイコン用アセンブラ言語に変換するプログラムを作成した。試作にあたっては，三菱電機の PLC 言語を使用した [1)]。

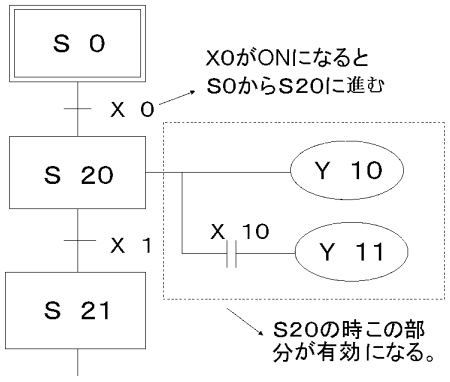


図 3 SFC

4.1.1　STL

STL は，SFC を PLC で実現するための PLC 言語である。図 4 が図 3 の SFC から作成される STL である。

4.1.2　STL の実現

STL をマイコン上で実現する方法は，図 4 のフロー図の形で行う。存在するステップすべてにビットを割り当てて，ON ならそのステップである，OFF ならそのステップで無い，とする。そのビットすべてを監視し，ビットが ON の場合，そのステップでのラダー命令を処理するサブルーチンを呼ぶ。

4.1.3　ラダー命令リスト処理

ここでは，ラダー命令リストをアセンブラに変換する方法を説明する。この方法で作成され

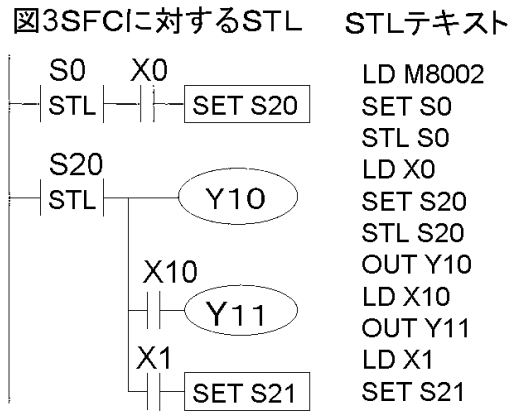


図 4　STL

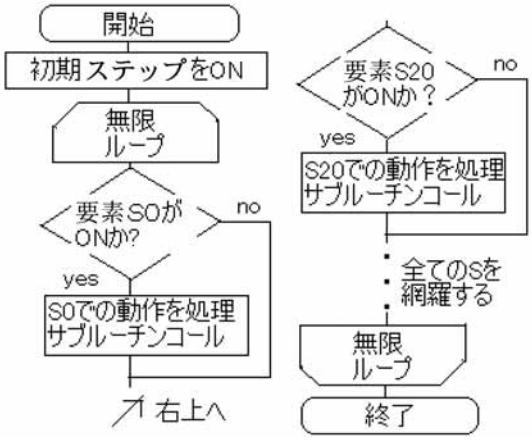


図 5　STL 処理フロー

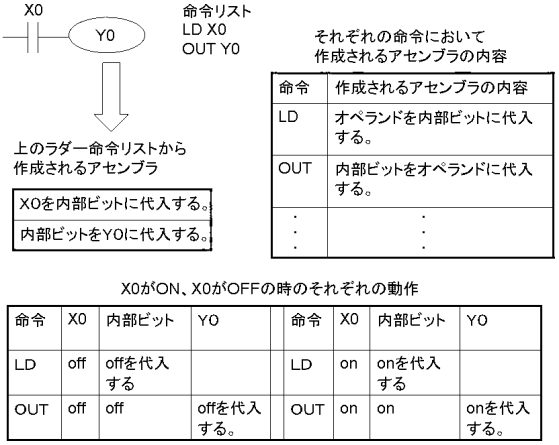


図 6　ラダー命令リスト処理

たアセンブラは，図 5 のサブルーチンコールで呼ばれるサブルーチンとなる。

ラダー命令リストの処理は，1 ビットの内部ビットを用いて行う。ラダー命令一つにつき，オペランド（もしくは内部ビット）を読んで，その値に応じて内部ビット（もしくはオペランドが指す要素）を変更する，という内容のアセンブラを作成する（図 6）。

4.2　変換プログラムの評価

4.2.1　動作検証

作成した変換プログラムで STL をアセンブラ言語に変換した。そのアセンブラ言語を PIC マイコンに書き込み動作させたところ，STL の意図と同じ動作をした。

4.2.2　性能評価

マイコンに書き込むプログラムのサイズは，基本プログラムが 1K バイトで，STL10 行につき 100 バイト増えた。

動作スピードを表 1 に示す。その動作スピードで，SFC を使用しなかった場合，PLC を使用した場合と動作スピードを比較してみた（表 2）。比較プログラムは，ステップが 50 でそのステップ内に AND，LD，OUT 命令がそれぞれ 10 個でてくるプログラムとした。SFC を使用したことで PLC に近い処理スピードになった。PIC より性能の良いマイコンを使えばさらに高速にすることができる。

表 1　動作スピード

処理内容	処理にかかる時間
入力要素読み込み処理	5μ秒
STL処理(図5)	1μ秒×ステップの数
各ステップでの処理 (図6)	命令の種類・量によって変わる AND命令　1μ秒 LD命令　1μ秒
出力要素書き出し処理	30μ秒

表 2　処理時間の比較

	PICで実施		PLCで実施※
	SFC未使用	SFC使用	
1サイクルの処理時間	1585μ秒	115μ秒	70μ秒

※PLCは、1μ秒で2命令処理すると想定

5．結　　言

(1) STL をマイコン言語に変換するプログラムを開発した。

(2) シーケンス制御を組み込みシステムに統合する環境が想定できた。

(3) 動作スピードは，工夫しだいで PLC に近づけられることが確認できた。

参考文献

1)浅野哲正，“ 図解シーケンサ百科 ”，オーム社，1990．