

マイクロコンピュータのデバッグ — (リモートモニタ) の開発 —

機械金属部 中嶋 勝也

1. 緒言

リモートモニタは、岡山県工業技術センターの平松氏によって開発された、おもにシングルボード・コンピュータを対象とした、デバッグ・システムである。

一方当工業技術センターでは、昭和60年度から昭和63年度にかけて、技術パイオニア養成事業先端技術研修においてマイクロコンピュータ技術課程を実施してきており、教材として6809MPU用のリモートモニタを開発したので報告する。

2. リモートモニタの概要

2.1 特徴

デバッグ・ツールとしてのリモート・モニタは、次のような特徴を持っている。

- ① ICE などのデバッグ・ツールは高価であり、複数台を設置するのが難しい。これに対して、このデバッグは、パソコンとソフトウェアで構成されているため、非常に安価である。
- ② パソコンで使用する汎用OS（ここでは、MS-DOS）上で動作するため、エディタやアセンブラと同一の環境上でデバッグが行なえる。
- ③ ホスト側のプログラムが高級言語で記述されているため、マン・マシン・インターフェースの改良や拡張が、ユーザ自身で、比較的簡単に行なえる。
- ④ ホスト側のプログラムの一部を修正し、またターゲット側の数百バイトのプログラムを開発しさえすれば、新しいCPU へも即座に対応できる。
- ⑤ デバッグしようとするプログラムはすべてターゲット・システム上で動作するため、ICE に起因するタイミングなどの問題がない。

これらの点から、研修におけるデバッグ・ツールとしては、最も適したものといえる。

2.2 構成

リモートモニタをハードウェアからみると、①ユーザ・インターフェースを実現するホスト・コンピュータ、②デバッグのターゲットとなるシングルボード・コンピュータ、③ホストとターゲットを接続する通信インターフェースの3つの部分からなる。一方ソフトウェアの点からみると、ホスト側とターゲット側の2つのプログラムから構成されている。

ホスト・プログラムは、デバッグに必要な情報を、ユーザに分かりやすい形で提供することが主な役目であり、ユーザが入力したデバッグ・コマンドを解釈し、実行する。そのために必要となるターゲット側との情報のやりとりは、プリミティブ・コマンドとよばれるコマンドを、ホストがターゲットに発行することによって行われる。

一方ターゲット・プログラムは、ホストが送信したプリミティブ・コマンドを解釈し、実行す

るためのものである。

このようにリモートモニタでは、デバッグに必要な機能をホストとターゲットに分割しており、またデバッグに必要な情報の入出力を、単純化されたプリミティブ・コマンドで行っているため、拡張性や移植性がすぐれたものとなっている。

3. ターゲットの構成

3.1 モニタ・プログラムの機能と構成

ターゲット側には、ホストから送られてくるプリミティブ・コマンドを解釈・実行するための、小規模なモニタ・プログラムが必要である。

ユーザがターゲットのデバッグ作業を行うために必要な機能は、メモリ及びI/O 空間の内容が参照・変更できること、レジスタの内容が参照・変更できること、及びターゲット側のプログラムが実行できること、に分けることができる。プリミティブ・コマンドは、これらの機能をホストとターゲット間で実現するためのものであり、今回開発したリモートモニタでは、次に示す9 つのコマンドで構成した。

- ① R コマンド：ターゲット・システムのメモリ内容を、ホスト・コンピュータに転送する。
- ② W コマンド：ホスト・コンピュータから転送されてくるデータを、ターゲット・システムのメモリに格納する。
- ③ I コマンド：ターゲット・システムのI/O アドレスの内容を、ホスト・コンピュータに転送する。(80 系CPU)
- ④ O コマンド：ホスト・コンピュータから転送されてくるデータを、ターゲット・システムのI/O アドレスに格納する。(80 系CPU)
- ⑤ F コマンド：ホスト・コンピュータから転送されてくるデータで、ターゲット・システムのメモリを埋める。
- ⑥ X コマンド：ターゲット・システムのレジスタ保管エリアの全内容を、ホスト・コンピュータに転送する。
- ⑦ Y コマンド：ホスト・コンピュータから転送されてくる全レジスタの内容を、ターゲット・システムのレジスタ保管エリアに格納する。
- ⑧ S コマンド：保管されているプログラム・カウンタからのプログラムを1 命令実行する。
- ⑨ G コマンド：保管されているプログラム・カウンタからのプログラムを実行する。

これらのコマンドの実行手順を、図3-1 から図3-9 に示す。

これらのコマンドは、使用されているCPU やハードウェア構成に無関係であり、また単純な機能であるため、開発が容易であり、数百バイトのサイズで構成することができた。

また、ホストとの通信にはRS-232C ンリアル・インターフェースを使用しているが、転送の速度が遅いため、大量のデータ転送を伴うデバッグ・コマンドでは、レスポンスが悪くなってしまう。これを向上させるために、メモリフィル(F コマンド)とステップ実行(S コマンド)の2 種類のコマンドは、他のコマンドで実現可能であるが、別コマンドとしている。さらに、メモリ内

容の転送 (R, W コマンド) は、当初1 バイトごとに行っていたが、指定されたバイト数をブロック転送するよう変更したことより、かなりの高速化を図ることができた。

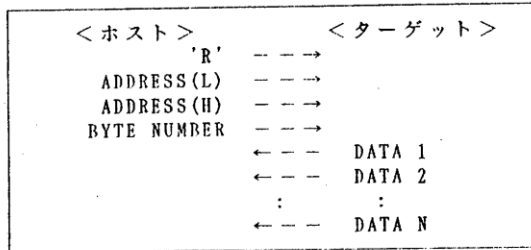


図 3-1 Rコマンドの通信手順

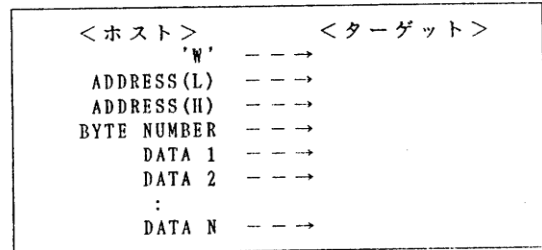


図 3-2 Wコマンドの通信手順

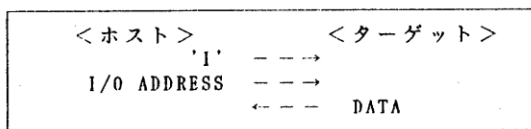


図 3-3 Iコマンドの通信手順

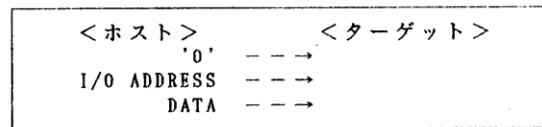


図 3-4 Oコマンドの通信手順

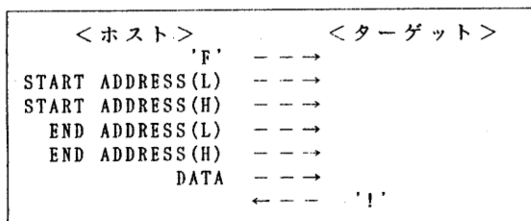


図 3-5 Fコマンドの通信手順

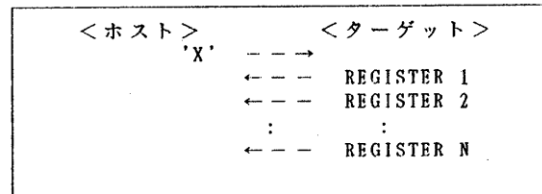


図 3-6 Xコマンドの通信手順

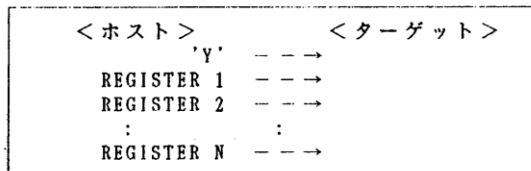


図 3-7 Yコマンドの通信手順

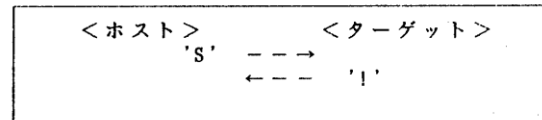


図 3-8 Sコマンドの通信手順



図 3-9 Gコマンドの通信手順

ひとつのCPUに対するリモートモニタが開発されると、同じCPUを使用している別のターゲットにリモートモニタを移植することは、容易に行うことができる。

ホストとターゲット間は、プリミティブ・コマンドによる通信だけが行われている。さらにこのプリミティブ・コマンドは、ターゲットのハードウェア構成とは無関係であるため、ホスト側のプログラムを変更する必要がない。

一方ターゲット側は、プリミティブ・コマンドが共通であるため、ほとんどの部分は修正する必要がなく、通信とRAMとROMのロケーションに関する部分を変更するだけで、別ターゲットへの移植作業が終了する。

今回開発したリモートモニタにおいては、この移植作業をさらに容易にするために上記の2つの部分をインクルード・ファイルとして分割した(図3-10)。これにより、ターゲット側のモニタ・プログラムの中から変更する部分を捜し出す必要がないため、移植作業を単純化することができた。

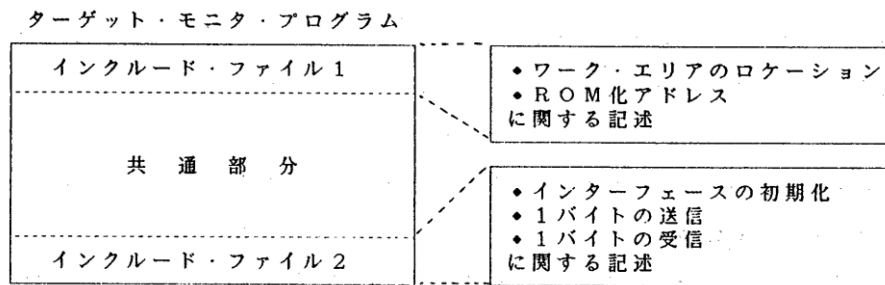


図3-10 ターゲットのモニタ・プログラムの構成

4. ホスト・プログラムの構成

4.1 ホスト・プログラムの機能

ホスト・プログラムは、ユーザが入力したデバッグ・コマンドを解釈・実行する。

今回開発したリモートモニタに用意されているデバッグ・コマンドを以下に示す

- | | |
|-------------|--------------------------------|
| (1) E)ecute | : MS-DOSのコマンドを実行する |
| (2) @) | : キーボード・マクロ命令の定義 |
| (3) -) | : キーボード・マクロ命令の実行 |
| (4) W)indow | : 表示ウィンドウを切り替える |
| (5) M)emory | : メモリの内容を参照, 変更する |
| (6) C)opy | : 指定した範囲のメモリの内容を, 別のアドレスにコピーする |
| (7) F)ill | : 指定した範囲のメモリの内容を, 同じデータで埋める |
| (8) D)ump | : 指定した範囲のメモリの内容を, 16進表示する |
| (9) L)ist | : 指定した範囲のメモリの内容を, 逆アセンブル表示する |

- (10) A)ssemble : 入力されたプログラムをアセンブルし、メモリ上に展開する
- (11) R)ead : ディスク上のHEXファイルを、メモリに展開する (ダウンロード)
- (12) V)erify : ディスク上のHEXファイルの内容と、メモリの内容とを比較する
- (13) H)exsave : 指定した範囲のメモリの内容を、HEX形式でディスクに格納する (アップロード)
- (14) S)tack level : トレース時のスタックのレベルを設定する
- (15) T)race : プログラムを指定した行数だけ実行する
- (16) B)reak Point : ブレーク・ポイントの設定, 解除を行う
- (17) G)o : 指定したアドレスからプログラムを実行する
- (18) eX)amine reg : レジスタの内容を参照, 変更する
- (19) Q)uit : リモートモニタを終了し, MS-DOS に戻る
- (20) ?) : コマンド・メニューを表示する

これらのうち, M・C・F・D・L・A・R・V・H の各コマンドは, ターゲットのメモリ内容に対する操作であり, プリミティブ・コマンドのRとWがあれば, 実行が可能である。つまりホスト・プログラムは, ユーザが入力したこれらのコマンドを, 必要なプリミティブ・コマンドに分解してターゲットと情報の交換を行い, ユーザの見やすい形にデータを変換し表示する作業を行っている。

4.2 ホスト・プログラムの構造

ホスト・プログラムはC 言語で記述しているが, 1 つのCPU に対するリモートモニタは, 次の3つのファイルで構成し, 分割コンパイルによって実行ファイルを生成する方法をとった。

1) RM09. C

このファイルでは, ターゲットのCPU に依存するコマンドの解釈・実行に関する記述を行っている。デバッグ・コマンドのうちCPU に依存する命令は, レジスタの参照, 変更とアセンブル及び逆アセンブルと, 全命令の一部である。したがって, 一つのファイルにまとめたことにより, 他のCPUへの移植作業はこのファイルに限定することができる。

今回の開発において, このファイルを3 種類 (6309, 6301/03, Z80) 作成したが, 他のファイルを変更すること無しに3 種類のリモートモニタが開発できた。

2) RMMAIN. C

このファイルでは, CPU に依存しないコマンドに関する記述を行っている。中身はさらに, 入力されたコマンドとパラメータの解析部と, コマンドの実行部 (プリミティブ・コマンドへの分解) で構成される。

3) RMPARTS. C

コマンドの解釈, 実行や, プリミティブ・コマンドの発行を行うための小規模関数群で構成される。

4.3 ユーザ・インターフェース

リモートモニタは、研修で用いるデバッガとして最適であるばかりでなく、通常のシステム開発にも十分に利用できる。そこで、コマンドの入力方法についても工夫した。

通常、ひとつのコマンドにはし、くつかのパラメータが必要であるが、すべてのコマンドとそれに必要なパラメータを覚えることは、研修などで初めて使用する場合には、困難をとまなう。そこでこのリモートモニタでは、コマンド及びパラメータの入力方法として、2つのモードを用意した。

ひとつは、コマンドの1文字だけを入力する方法で、必要なパラメータは、ホストとの対話によって入力することができるものである。使用の初期の段階ではこの方法を用いる。また、コマンドとパラメータへの理解が進んだ段階では、コマンドとパラメータを一行で記述し、入力する方法を用いることができる。

5. マルチウィンドウ表示

マルチウィンドウ表示は、表示できるデータが増加することや、情報を分離して表示できることから、ユーザ・インターフェースの向上に役立つ表示システムである。今回のリモートモニタの開発にあたっては、隠れたウィンドウに対する書込みができないなど、本格的なものではないが、表示のマルチウィンドウ化のためのライブラリを作成した。

このライブラリに含まれる関数を、以下に示す

- (1) wn-to-con : 画面の切り替え (ウィンドウ→コンソール) ・
- (2) con-to-wn : 画面の切り替え (コンソール→ウィンドウ)
- (3) w-reset : 全ウィンドウの消去
- (4) w-cls : カレント・ウィンドウの消去
- (5) w-open : 新しいウィンドウのオープン
- (6) w-change : カレント・ウィンドウの変更
- (7) w-current : カレント・ウィンドウの番号を得る
- (8) w-tmpopen : テンポラリ・ウィンドウのオープン
- (9) w-tmpclose : テンポラリ・ウィンドウのクローズ
- (10) w-tcolor : 文字の色を設定する
- (11) w-locate : カーソルの移動
- (12) w-position : カーソル位置を得る
- (13) w-chkbb : キーボードの検査
- (14) w-putchar : 1文字出力
- (15) w-puts : 文字列出力
- (16) w-printf : 書式付出力
- (17) W-getchar : 1文字入力
- (18) W-gets : 文字列入力

これらの関数は、ポーランド社製TURBO C コンパイラに付属する、テキストビデオ関数を使って記述した。このライブラリを用いたマルチウィンドウ表示例を、図5-1 に示す。

memory	execution	register
2000: 8611		PC :2004
2010: 0101	LDAA #11	ACCA:11
2020: 0101	PC A B IX SP (HINZVC)	ACCB:00
2030: 0101	2002 11 00 0000 7FBC 11110000	IX :0000
2040: 0101		SSP :7FBC
2050: 0101	NOP	CCR :F0
2060: 0101	PC A B IX SP (HINZVC)	1
2070: 0101	2003 11 00 0000 7FBC 11110000	1
2080: 0101		H-1
2090: 0101	NOP	I-1
20A0: 0101	PC A B IX SP (HINZVC)	N-0
20B0: 0101	2004 11 00 0000 7FBC 11110000	Z-0
20C0: 0101		V-0
20D0: 0101		C-0
20E0: 0101		
20F0: 8633		
command		
D: rm>?		
D: rm>d2000,100		
D: rm>x		
D: rm>t3		
D: rm>		

図5-1 6301リモートモニタのマルチウィンドウ表示例

6. 結言

今回開発したリモートモニタは、3年間にわたった研修を実施しながら改良を加えてきたものであり、現在いくつかの企業や研修・訓練機関においても使用されている。

今後もさらに有効な機能の追加や、リアルタイム・モニタへの組み込み等の開発を進める予定である。

開発にあたっては、岡山県工業技術センターの平松氏、中小企業大学校の谷氏及び、東京都立工業技術センターの坂巻氏よりご指導をいただきました。深謝いたします。