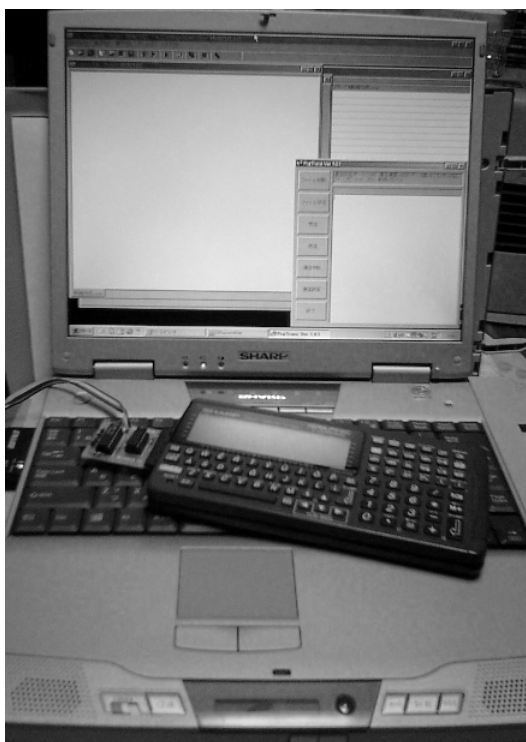


- SHARP 製 ポケットコンピュータ PC-G850 対象 -

ぽけこんのほん 3



特集

「PC-G850 のクロス開発環境の構築」

著：単色ドット絵向上委員会 (G67)

PC-G850 のクロス開発環境の構築

SHARP ポケットコンピュータ PC-G850 のユーザが使用可能なメモリ容量は 30179 バイトです。標準の環境で様々なプログラムを開発しているとすぐにメモリを食いつぶしてしまいます。また、マシン語プログラムの開発には内部メモリの内容が破壊されることもしばしばです。そこで「ポケコン外の物に保存する」方法を取るわけです。外部で開発してポケコンに転送する、という手段も取ることが出来ます。ここではどのようにしてそれらを実現しているかを紹介していきます。各種ソフトのマニュアルの補足的な内容になります。

その前に。

外部(ここではパソコン)とやり取りさせるために必要な物として「シリアル通信ケーブル」がなくてはなりません。シャープからは「CE-T800」という名称で発売されていますが、9800 円くらいだと思います。高いです。なので、ネット上で見つかる回路図を元に自作するか、以下のサイトからキット/完成品を通販して組み立てると良いでしょう。

有限会社ユウピット電子 商品名 : SIO (2940 円)

<http://www.info-north.net/youpit/>

私は、このキットを組み立てて使用しています。5 年以上使ってますが、故障なしです。Windows 用の通信ソフト「PcgTrans」が付いてくるのが魅力的です。通信ソフトが XP で動くのかは試してませんが・・・

有限会社高松製作所 商品名 PC-E200 シリーズ用 パソコン接続ケーブル(3000 円)

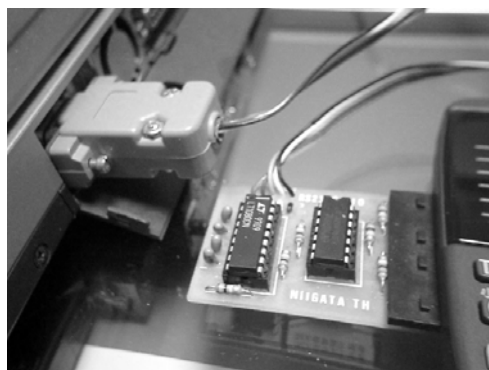
<http://tss.ram.ne.jp/>

こちらの会社からは PC-E500 用のメモリカードやクロックアップキットを購入したことがあるのですが、G850 でも使用できる(G850 は E200 の上位互換機)ケーブルも販売しているようです。

さて、次から紹介していきます。

<例 1 : SIO(同等品も可)+PcgTrans>

ユウピット電子さんから購入したシリアル通信ケーブルキットに付属していたソフト「PcgTrans」を使用してポケコンとパソコン間でデータのやり取りをします。



(←左) パソコンと SIO-ポケコンはこんな感じで使ってます
(↑中央) パソコン背面シリアル端子と SIO-ポケコンとの接続例
(→右) PcgTrans メイン画面

この組み合わせで可能になるのは以下になります。

- ・ポケコンからパソコンへのテキストデータ送付
- ・パソコンからポケコンへのテキストデータ送付
- ・ポケコンからパソコンへのマシン語データ(インテルヘキサ形式)の送付
- ・パソコンからポケコンへのマシン語データ(インテルヘキサ形式)の送付

何でも出来るような気がしますが、開発時にいろいろ制限がかかります。例えば、BASIC 言語で開発している場合、テキストデータから BASIC データに変換するため(逆もあり)、約 15000 バイトがプログラムに要求できる最大サイズになります。マシン語開発を内臓のアセンブラを用いて行くと、経験則ですが、3000 バイトのプログラムを作成するのに 20000 バイトくらいのテキストデータが必要になるのでメモリをたくさん使ったマシン語プログラムが組めません(もちろん、誰かが作ったメモリいっぱいマシン語データをロードするのは出来ますが)。めいいっぱいメモリを使ったプログラムを保存したいときはもう 1 台同じポケコンを用意して 11 ピン端子をクロス結線をつないで CSAVE で保存したり(G850V は BSAVE?)、カセットテープインターフェースを用いて保存すると良いのですが、私は「外部で開発すること」に主眼を置いているためここでは考えません。マシン語をメモリにめいいっぱい展開する開発方法は後の例 2 で紹介します。

ソフト(PcgTrans)をインストールし、ポケコンとパソコンを SIO でつなぎます。パソコン側にシリアルポートがない時は USB からシリアルに変換するケーブルとか使ってください(高いけど)。

まずは通信設定をします。

基本的に PcgTrans と PC-G850 の通信設定が同じであれば良いです。

私は通信速度を最高速(9600bps)にするだけで他はいじっていません。この速度にしているのは単純に早く転送が済むからです。別に設定が同じになっていれば 4800bps とか 1200bps とかでもかまいません。しかし、通信速度が速いとエラーが出ることもあります。2400bps 以下は経験上安定して使えます。

また、PcgTrans の説明書でマシン語とテキストを送るときはそれぞれ違う設定をしなくてはならないように書かれていますが、実際はどちらも同じで大丈夫です。

通信設定：PcgTrans

「通信設定ボタン」を押すと通信設定画面が出てきます。

以下のように設定してください。

ポート番号 : 1 (各環境に依存)
通信速度 : 9600
データ長 : 8
パリティ : なし
フロー制御 : RS/CS

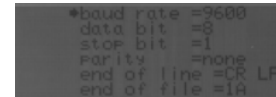
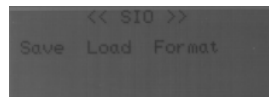
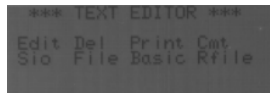


通信設定：PC-G850

TEXT ボタンを押す→Sio 選択(「S」を押す)→Format 選択(「F」を押す)

左右キーと ENTER キーで値を以下のように調整します。

baud rate : 9600
data bit : 8
stop bit : 1
parity : none
end of line : CR LF
end of file : 1A
line number : yes
flow : RS/CS



まずはポケコンで打ち込んだ BASIC プログラムをパソコンに転送してみましょう。

(C や ASMBL を使う方は Basic と Text の変換のところをコンパイル/アセンブルに置き換えてください。)

以下のようなプログラム例(HELLO WORLD!!)を PRO モードで打ち込みます。

```
10 CLS
20 LOCATE 0,0
30 PRINT "HELLO WORLD!!"
```

典型的入門プログラム「ハローワールド」です。

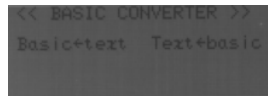
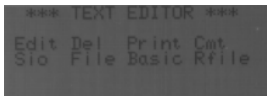
RUNMODE で実行(RUN)するとは以下のように表示されます。



「HELLO WORLD!!」と表示されました。
当たり前ですが・・・

テキスト形式に変換しないと送ることができないため、BASIC のデータをテキストに変換します。

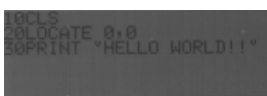
まず、TEXT キーを押してテキストモードにした後、Basic を選択(B キーを押す)します。



BASIC CONVERTER 画面が表示されます。

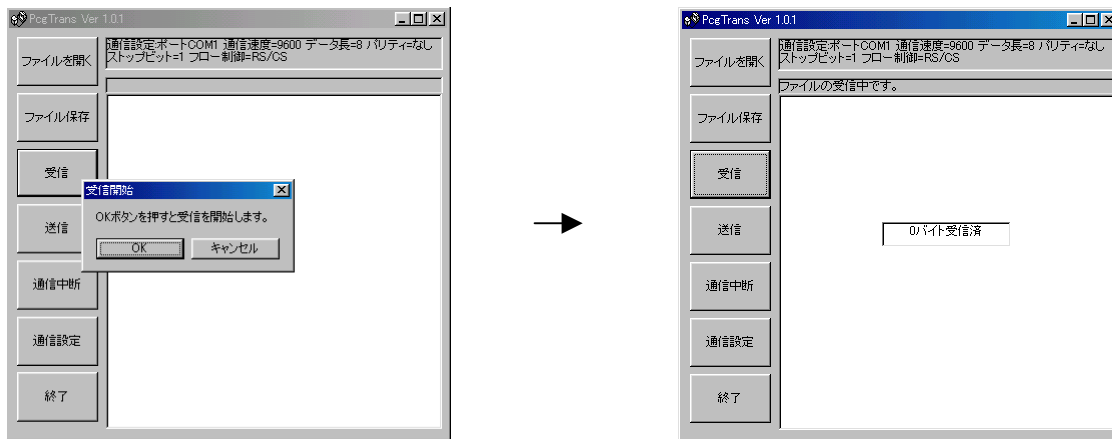
次に Text←basic を選択(T キーを押す)すると PRO モードで入力した物がテキストエリアにコンバートされます。テキストエリアに既に何か入っていると「TEXT DELETE OK?(Y)」と出るので Y キーを押してください。

TEXT ボタンを押した後 Edit を選択(E キーを押す)して TEXT EDITOR を立ち上げ、↓キーを押すと先ほど BASIC で作ったプログラムがテキストに変換されたことを確認できます。

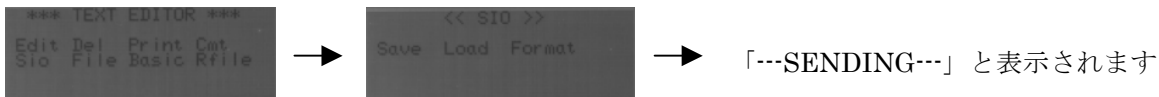


行番号と命令文の間のスペースがなくなっていますが、PRO モードではスペースがあるように見えるだけで、実際のデータはスペースがないためです。

転送するものが出来たら受ける側(パソコンの PcgTrans)で受信準備をします。「受信」ボタンを押して出てくる「OK を押すと受信を開始します」というメッセージダイアログの OK ボタンを押してください。進行状況を表示するバーが出てきて受信の準備が整います。



受信側の準備が出来たので、送信側であるポケコンからデータを送信します。
TEXT キーを押した後 **Sio** を選択(**S** キーを押す)して **Sio** 画面を立ち上げます。
 ここで **Save** を選択(**S** キーを押す)とポケコン-パソコン間で通信が始まります。



通信に成功すると、パソコン側の PCGTRANS 画面にポケコンで作成したデータが表示されます。



後はファイル保存ボタンからデータの保存が出来ます。
 (拡張子が付かないので自分で.txtとか.basとか付けてください)

ちなみに、ポケコンの通信設定の「line number」を **no** に設定
 すると行番号なしで送られます。このあたりのことは **G850** の
 マニュアルを参照してください。

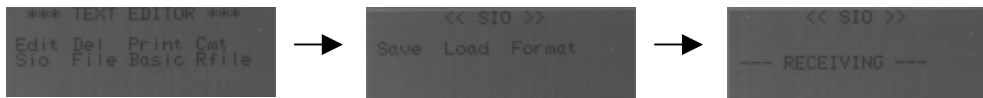
これで、ポケコン内の **BASIC** のデータはパソコンに送って保存
 することが出来るようになりました。

今度はパソコンからポケコンに **BASIC** のプログラムを転送し、実行してみましよう。
 先ほどやり取りしたデータの Print 文で表示した「HELLO WORLD!」を変更して送ることにします。

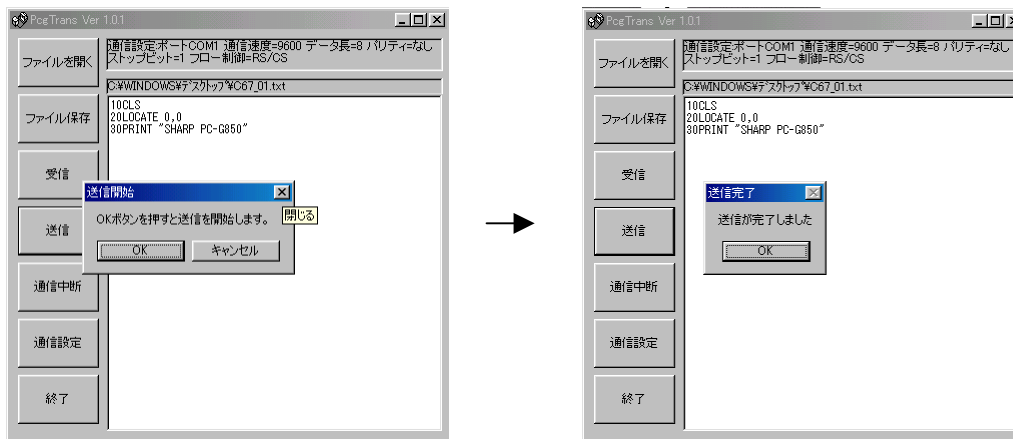


PcgTrans 上で「HELLO WORLD!!」→「SHARP PC-G850」
 のように変更しました。

送るデータが出来たので、ポケコン側の受信の準備をします。
TEXT ボタンを押した後 **Sio** を選択(**S** キーを押す)して **Sio** 画面を立ち上げます。
 ここで **Load** を選択(**L** キーを押す)とポケコン側で受信準備をします。

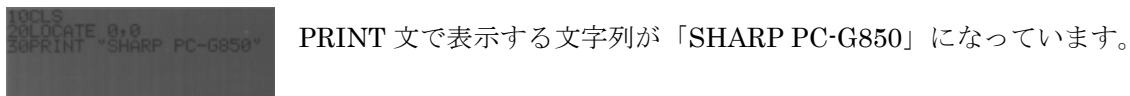


ポケコン側に「--- RECEIVING ---」と表示されたら PcgTrans からデータを送信します。「送信」ボタンを押して出てくる「OK を押すと送信を開始します」というメッセージダイアログの OK ボタンを押してください。パソコン-ポケコン間で通信が始まります。

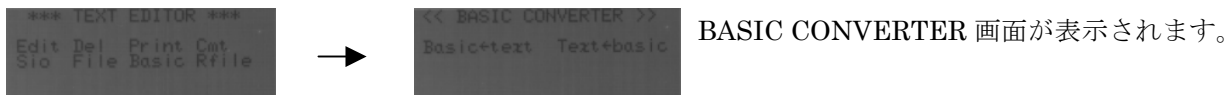


注：ここで通信に失敗する場合がありますが、大概是最終行に改行が入っていないのが原因です。パソコン側でデータを作成してポケコンに送るときはよくあります・・・

通信が完了したらデータを確認しましょう。TEXT ボタンを押した後 Edit を選択(E キーを押す)して TEXT EDITOR を立ち上げ、↓キーを押すと先ほど送られてきたテキストデータを確認できます。

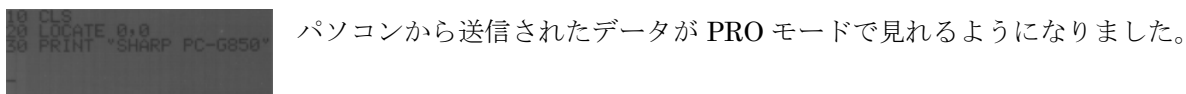


このデータを Basic で使えるように変換します。まず、TEXT キーを押してテキストモードにした後、Basic を選択(B キーを押す)します。



次に Basic← Text を選択(B キーを押す)するとテキストが BASIC のデータにコンバートされます。テキストエリアに既に何か入っていると「BASIC DELETE OK?(Y)」と出るので Y キーを押してください

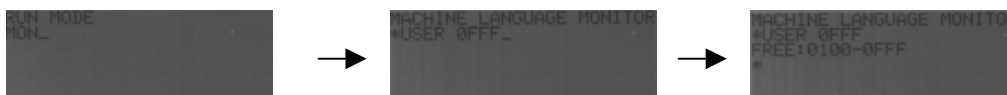
BASIC キーを 2 回押して PRO モードにして「LIST」と打ち込んで変換されたデータを確認しましょう。



BASIC キーを押して RUN モードにした後、「RUN」して実行すると「SHARP PC-G850」と表示されます。



次にマシン語データ(インテルヘキサ形式)のやり取りをしてみましょう。データを作るためにまずはテキストエディタでマシン語プログラムを作成し、メモリ上に配置します。まずはマシン語領域を確保します。今回は大した物は作らないので、適当に 0100H から 0FFFH をマシン語領域とします。RUN モードから MON と打ち込み、マシン語モニタを起動した後、USER 0FFFH と打ち込んでください。



(注： この時 Basic のプログラム等があるとデータが破壊されます。データを退避する等してください。マシン語のデータを扱う前は ALLRESET するのが最も望ましいんですが・・・)

TEXT ボタンを押した後 Edit を選択(E キーを押す)して TEXT EDITOR を立ち上げ、以下のプログラムを入力してください。わたくしの G850 用マシン語 HELLO WORLD!!プログラムです。ラベルと行番号の間に空白を入れない等、注意して打ち込んでください。

```
10 ORG 0100H
20 LD DE,0000H
30 LD B,0DH
40 LD HL,DAT
50 CALL 0BFF1H
60 RET
70DAT:
80 DB 'HELLO WORLD!!'
```

一応解説しておきます。

ORG 命令で開始番地を 100H に設定。

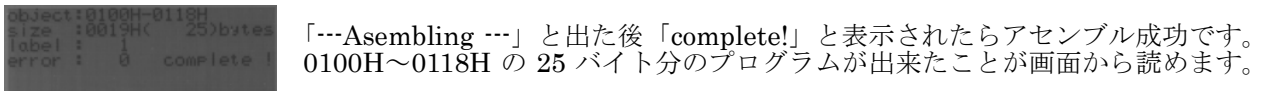
文字列表示のために IOCS(Input Output Control System の略。シャープが公開しているマシン語ルーチンのこと)の文字列表示「0 BFF1H」をコールします。引数は D レジスタに行、E レジスタに列、B レジスタに表示する文字の数、HL レジスタに表示したい文字の格納先アドレスとなっております。今回は 0 行 0 列目から 13 文字、ラベル DAT からのデータを表示、となっております。

RET でマシン語モニタに戻ります。最外周の RET のため、モニタに戻るのです。

入力し終わったらアセンブルします。SHIFT+BASIC でアセンブラメニューを起動し、Assembler を選択し(A キーを押す)、て Z80 アセンブラを起動してください。



テキストエリアにあるテキストをアセンブルします。Asm を選択し(A キーを押す)アセンブルしてください。



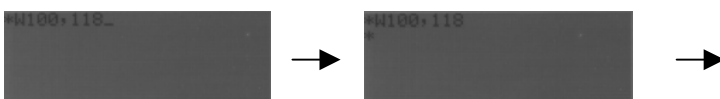
アセンブルエラーが出たときは G850 のマニュアルや Z80 のリファレンスを見てエラーを修正してください。

アセンブルが正常終了したときに CLS キーを押すとすぐにマシン語モニタに飛びます。マシン語モニタに移ったら、今回のプログラムには CLS ルーチンを含んでいないので CLS キーを押して画面をクリアしてください。画面をクリアしたら G100 と打ち込んでアセンブルしたものを実行してみましょう。



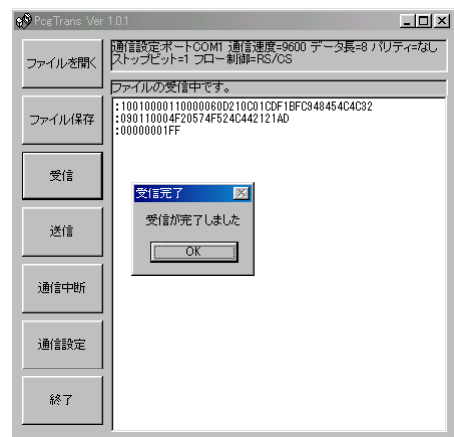
実行の確認が出来たらポケコンからパソコン側にマシン語データ(インテルヘキサ形式)を送りましょう。まずは以前と同じくパソコン側を受信できる状態にします。PcgTrans の「受信」ボタンを押して出てくる「OK を押すと受信を開始します」というメッセージダイアログの OK ボタンを押してください。(ここでは画像は省略します。BASIC のデータのやり取りを参照してください。)

マシン語モニタから W コマンドを使ってデータ送信します。引数はアセンブルしたときに出了使用メモリ領域の数値を指定します。なので「W100,118」と実行します。



実行すると通信が開始され、PcgTrans の画面に指定したメモリの内容がインテルヘキサ形式でデータが表示されます。(ポケコンから W コマンドを使うとインテルヘキサ形式で送られるのです。)データは以下のように表示されます。

```
:10010000110000060D210C01CDF1BFC948454C4C32
:090110004F20574F524C442121AD
:00000001FF
```



後はファイル保存ボタンからデータの保存が出来ます。(拡張子が付かないので自分で.hexとか.asmとか付けてください)

インテルヘキサ形式が何を表しているかは下記の Web ページが参考になります。

PastelMagic 内 HEXファイルフォーマット

<http://www.pastelmagic.com/tips/hexform/hexform.html>

すぐに理解できるとおもいます。

さて、今度はパソコンからポケコンにインテルヘキサ形式のデータを送ってみましょう。

すごく特殊なやり方になりますが、先ほどのデータの文字列格納場所のデータを「HELLO WORLD!!」→「SHARP PC-G850」のように変更します。「HELLO WORLD!!」を ASCII コードに直すと、「48454C4C4F20574F524C442121」となります。この部分を「SHARP PC-G850」を ASCII コードに直した「53484152502050432D47383530」と置き換えます。文字とコードの対応は ASCII コード表を見てください。インテルヘキサフォーマットに照らし合わせると、1行目の最後の方と2行目の途中からが文字列データになっています。この部分を書き換えます。また、行の最後はチェックサムとして、行のラスト1バイトを除いた全てのデータを足し合わせたものの下位16ビットの2の補数となっているため、計算して書き換えます。

文字列データを入れかえると

:10010000110000060D210C01CDF1BFC95348415232

:09011000502050432D47383530AD

:00000001FF

・・・となります。そして1行目と2行目のチェックサムを計算しなおして書き換えます。

$10 + 01 + 00 + 00 + 11 + 00 + 00 + 06 + 0D + 21 + 0C + 01 + CD + F1 + BF + C9 + 53 + 48 + 41 + 52 = 4D7$

下位16ビットは29。2の補数なので、1を引いてビット反転します。

$D7 - 1 = D6 = 11010110$ 反転します → $00101001 = 29$

$09 + 01 + 10 + 00 + 50 + 20 + 50 + 43 + 2D + 47 + 38 + 35 + 30 = 22E$

下位16ビットは2E。2の補数なので、1を引いてビット反転します。

$2E - 1 = 2D = 00101101$ 反転します → $11010010 = D2$

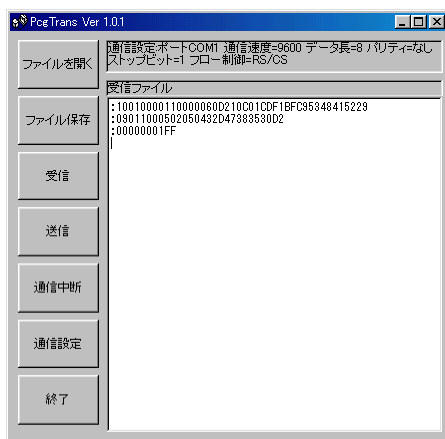
チェックサムを書きかえると

:10010000110000060D210C01CDF1BFC9 5348415229

:09011000502050432D47383530D2

:00000001FF

・・・となります。これを今度はポケコンに送信します。



あー。特殊過ぎるので、よく分からない人は何も考えずに変えるか、そのまま変えないで次に進んでください。

こんな修正の仕方、普通はしません。

なんというか、表示されるものの意味が分かっていると親しみやすいかな、と思ったのでやっただけです。

ポケコンをデータが受信できる状態にします。RUN モードから MON を実行しマシン語モニタを立ち上げた後、R100 と実行します。するとポケコン側が通信待ちの状態になります(BUSY が点灯します)。引数の100は0100H番地からプログラムをセットするという意味です。ちなみに、通信待ち状態はBREAKで復帰出来ます。ここでPcgTransからデータを送信します。「送信」ボタンを押して出てくる「OKを押すと送信を開始します」というメッセージダイアログのOKボタンを押してください。パソコン-ポケコン間で通信が始まります。(PcgTrans側の画像は省略します)



通信に成功すると、INFO:0100-118 と格納アドレスが知らされます

注：この時ポケコン側で「I/O DEVICE ERROR」が出る時は通信速度を下げると出なくなるようです。2400bps以下だとよいと思います。エラー原因は不明ですが、外部からのノイズとかでチェックサムエラーでも起こしているのかも。ポケコンからの送信時は大丈夫なのになあ・・・

通信が完了したら CLS キーを押してから G100 を実行し、送られたデータの確認をしましょう。D コマンドでメモリの内容が変わっていることを確認しても良いです。(マシン語モニタ上で D100 と実行すると 0100 番地からの内容を表示する)



G コマンドで実行して確認
先ほど、パソコン側で直した修正が反映され、「SHARP PC-G850」と表示されています。



D コマンドで文字列データの変更を確認

ここまででマシン語データのやり取りができるようになりました。

<例 2 : SIO(同等品も可)+Z80assembler for Windows98/Me>

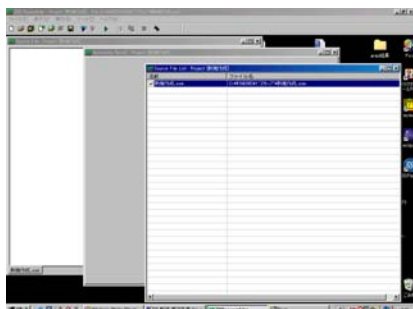
Z80assembler for Windows98/Me はフリーソフトです。Vector のソフトライブラリからダウンロードできます。Vector では「Z80CPU Assembler & Simulator」という名前ですが、ソフトのバージョン情報見ると「Z80assembler for Windows98/Me」となっているので、こちらを使用します。逆？

Vector

<http://www.vector.co.jp/>

Z80assembler for Windows98/Me (Z80CPU Assembler & Simulator)

<http://www.vector.co.jp/soft/win95/prog/se232600.html>



98/Me 用とありますが、XP でも動くと思います。作者の方は 2000 までは動くと言っています。

また、このソフトは別にポケコン用というわけではなく汎用 Z80 アセンブルプログラムです。

起動すると

← のように複数ウィンドウが立ちあがります。

特色として以下のようなことが挙げられます。

- ・「行番号がない」マシン語プログラムをインテルヘキサ形式にアセンブルできる
- ・当たり前だが、コメントに日本語が使える。便利！
- ・ラベルの文字制限がない。楽。
- ・シリアル通信でインテルヘキサ形式のデータを送ることが出来る(受信は不可)
- ・ORG を 0100H 以前を指定することが出来る(ORG 0080H のように)
- ・使用可能メモリをまるまる使ったマシン語プログラムが組める！

行番号、というのはポケコン上のアセンブラで長いプログラムを作る上でひとつの障害となっていました。アセンブラのプログラムリストでは 1 行 1 命令が基本で、しかもひとつの機能を実現するために多くの命令を使います。そのため、なにか変更を加えようとするとき「行番号を追加すること」に労力を使わなくてはならず、全然使えませんでした。しばらくは、パソコン上で行番号なしでプログラムを組み、転送時に行番号を付加できるテキストエディタで行番号を追加していたりしていましたが、これでも面倒くさいです。これが解消されるというのは非常に素晴らしいことです。

このソフトはインテルヘキサ形式でシリアルポートからデータを送信できます。便利です。そのため、この機能を使うと G850 のメモリ 30179 バイトをマシン語で使い尽くすことも可能です。また、内臓アセンブラの ORG 命令では 0100H 以降でしか指定してアセンブルすることしか出来ませんが、このソフトを使うと未使用領域 0040H~00FFH にプログラムを配置することも可能です。

注：未使用領域 0040H~00FFH は未使用の領域なので自由に使うことが出来ます。例えば BASIC 上からデータの保存に使用したり、USER 命令で確保しなくても使えるマシン語領域として自由に使えます。192 バイトは大きいです。ただし、未使用とはいってもシステム使われることがあります。0066H には Z80 のノンマスクブルインタラプトが検出されたとき用の命令(ED 45 = RETN)が入っていますが、つぶしても問題ありません。007FH は電源投入時に必ず「FFH」が入ってしまいます。なので、この領域を使うときには命令を書かないようにするとか、定数 FFH として扱う等気をつけなくてはなりません。0080H~0085H は内臓アセンブラを使うと書きこまれます。しかし、外部でアセンブルするので消しても問題ありません。内臓アセンブラを使っている時でも問題ないと思いますが。

あと、このソフトは私が使い始めて間もないため詳細な機能とかはさっぱり分からないので、基本的な部分だけ触れたいと思います。

Z80assembler をインストールして立ち上げたら、まずは通信設定を済ませてしまいましょう。ポケコンの設定は例1を参照してください。

通信設定：Z80assembler

ツール→設定→通信 で通信設定画面が出てきます。ポケコンの設定と合わせるのと同じです。以下のように設定してください。

通信速度 : 9600
 データビット : 8
 ストップビット : 1
 ターミネータ : CR + LF
 パリティ : なし
 ポート : Com1 (各環境に依存)
 フロー : RTS/CTS



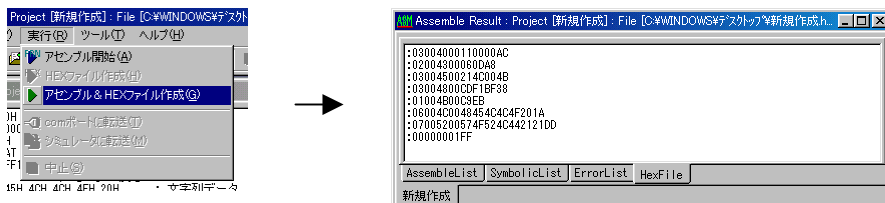
では、今回の例プログラムを示します。「Source File」とあるタブに以下を打ち込んでください。コメントは打ち込まなくても結構です。

```

ORG 0040H      ; 40Hから配置できる！
LD DE,0000H   ; x0, y0にセット
LD B,0DH     ; 13文字表示
LD HL,DAT    ; 文字列先頭アドレス
CALL 0BFF1H  ; IOCS文字列表示のコール
RET          ; モニタに戻る
DAT:         DB 48H,45H,4CH,4CH,4FH,20H ; 文字列データ
             DB 57H,4FH,52H,4CH,44H,21H,21H
    
```

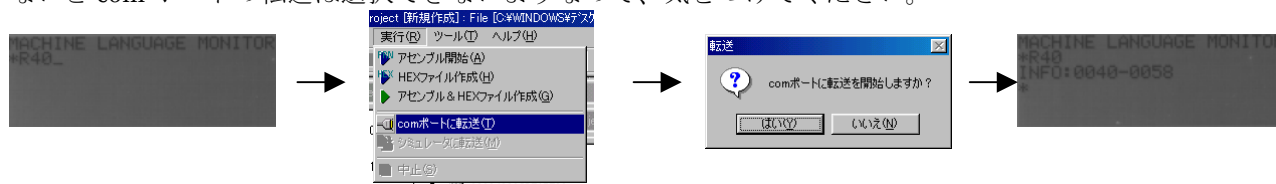
・・・ずいぶん様変わりしました。入力がある程度自由になったので、日本語コメントを入れ、TAB を多用しています。ORG で指定しているのは 0040H ですが、ここからだとマシン語モニタで USER 命令を使って領域確保しなくても格納できます。DB 命令の部分が文字列ではなく、文字コードを指定していますが、このアセンブラには文字列をアセンブル時に文字コードに変換してくれないのでしょうかなく文字コードを入れています。また、ラベルと命令が同一の行にないとダメだったので(ラベルのみの行は認めない)ラベルの後に命令を入れています。

入力し終わったら、実行 → アセンブル&HEX ファイル作成 を選択し、アセンブルを開始します。



成功すると「AssembleResult」とあるタブにインテルヘキサ形式のリスト等が作成されます。エラーのときは同ウィンドウの ErrorList タブにエラー情報が出るのでそれを参考に直してください。

それでは、ポケコンにこのリストを送付しましょう。ポケコン側をマシン語モニタから R40 として 0040H から情報を受け取るようにさせ、Z80assembler 側では 実行→com ポートに転送を選択して、「com ポートに転送を開始しますか」と出たら「はい」を選択します。AssembleResult タブにフォーカスが当たっていないと com ポートの転送は選択できないようなので、気をつけてください。



補足：この時、0040H～0058H に展開され、0040～00FFH の間に収まっているため USER でマシン語領域を確保していなくても転送に成功します。子ネタなんですけど。

通信が完了したら CLS キーを押してから G40 を実行し、送られたデータの確認をしましょう。



これで、マシン語をパソコンで開発してポケコンで実行する、ということができるようになりました。

<最後に>

さて、ここまでで 2 例紹介しましたがどうだったでしょうか。「当たり前のことを何をいまさら」と言う方もいらっしゃるかもしれませんが、「おお、そうなのか、やってみよう！」という方もいらっしゃると思います。

画像を多用し、例プログラムとか用意してやりましたが、効果はどのようなだろう。少なくともマシン語で HELLO WORLD を書いて公開しているのは私くらいだと思うけど。一般的な Z80 マシン語の演習するときって IOCS 分からないとやりようがないんだよね・・・

今までポケコンの本を 2 冊出しましたが、出していて不安だったのが「買っていった人はマシン語とかわかるのか？」というものでした。2004 年、そう、ポケコンジャーナルが休刊してからもう 8 年たちます。そのためポケコンの情報は Web 上で手に入れるしかありませんが、はたして、「どのようにしてポケコンのプログラムをパソコンに保存する」ことを詳細に書いているページがあったのでしょうか。あったらここには書かないし、私も苦勞していない・・・です。確かに、ケーブルの回路図はありますがそれだけ。パソコンでアセンブルしてくださいとか書いている人も居ますが、「どうやって？」といった感じです。そもそも、マシン語や通信のことを扱ったりするページが少なすぎです。一般常識なのかもしれませんが、2ch の掲示板でその手のことを聞く人が居たりするので、分からない人にはとことん分からない世界なのだと思います。

保留していたネタとして「LINUX ザウルスで PC-G850 のマシン語アセンブルと転送をしよう」というものがありました。これは次回かなーと。通信は CF→シリアルのアダプタを購入し、cu というソフトを利用すれば出来そうです。アセンブルに関しては Perl で Z80 アセンブラを作った方が居たので何とかなりそうです(バイナリ形式 to インテルヘキサ形式のスクリプトを書く必要がありそうですが)。

今回の記事が誰かの役に立つことを願います・・・

<あなうめ：漫画の中のポケコン番外編>

ぼけこんのほん1で「漫画の中のポケコン」という記事を書きました。その中で「ああっ女神さまっ」の劇中でポケコンが登場していることを書きました。(5巻と6巻にちらっと、15巻に解説)



それで、最近「ああっ女神さまっ COLLECTION」というオフィシャルガイドブック(左)が出たわけですが、気になったのがポケコンは載っているのかなーという点でした。

ポケコン
高級言語 BASIC の使えるプログラム電卓。シャープはポケットコンピュータと呼んでいた。略してポケコン。
中でも PC-1401 は科学技術計算用に関数を強化した機種だが、もう 2~3 世代旧式になっている(15巻に解説あり)

おおっ・・・ありました！よかった・・・
メカ辞典(P271)に載ってます。

ポケコン シャープ PC-1401

高級言語 BASIC の使えるプログラム電卓。シャープはポケットコンピュータと呼んでいた。略してポケコン。

中でも PC-1401 は科学技術計算用に関数を強化した機種だが、もう 2~3 世代旧式になっている(15巻に解説あり)

・・・とのこと。どの辺が 2~3 世代旧式に、なんだろう。ポケコン的に、だろうなあ。現代パソコンから見ると 5 世代以上旧式だもんなあ・・・

なにはともあれ、よかったよかった。

～ あとがき ～

今回のネタはありきたりだったと思うけれども、昔こんな情報があったら、という気持ちで書きました。ただし、正規のマニュアル読まないのを書いたので当たり前の事項がたくさん含まれていると思います。

今回の作業日数は2日でした。怠けすぎました。次回から気をつけます・・・年末のゲームラッシュに負けた・・・

ああ、今思いついたけど、SIOの説明書でマシン語とテキストの通信設定が違うのはパリティ付与とかしでエラーを防止するためかもしれない。理由が書いてないからどうでもいいけど。

前回のコミケで一番印象に残っているのは、私のスペースをちらっと見て「G かよ」と言った人がいたことでしょうか。



いや、他の機種も持っているんですけどね・・・

PC-U6000

(友人の大学の先輩の置き土産をもらった)

VX-4

(友人の妹の教材(白石工業高校)を譲り受け)

PC-G820

(倒産した会社で使われていたのをもらった)

全部学校教育用だー。しかも、もらったのばかり。

他の機種は初級者レベルなのでネタに出来ません。すいません。

さあ、夏に向けてゲーム作るぞ！っと。・・・作れたらいいなあ。

2004年12月26日 あきひ(首謀者&執筆者)

奥付

誌名 : 「ぼけこんのほん3」
制作 : 単色ドット絵向上委員会(あきひ)
印刷 : OFFICE 24 仙台店
(24時間営業の文房具屋さんで便利この上ない)
発行日 : 2004年12月28日 初版

HP : 「秋日和」
<http://ha9.seikyou.ne.jp/home/akihi/>
メアド : akihi@ma9.seikyou.ne.jp

※本当はこの本の内容は無断転載とかして欲しくないんですが、こんなの転載しても得する人は少ないので「著作権は放棄しないが転載自由」とします。
あくまでもこの本の目的は「ネタが少ないポケコン界にネタを突っ込んでみる」のが目的です。
内容をちょこっと変えてさも自分がやったように書いても、俺が笑うだけです。害はないです。たぶん。
私も私で「私の80%はものまねで形成されています」状態なので人のことをツッコめないのですが。

でも、好き。大好き。

…ポケコンジャーナル誌の投稿で一番印象に残ったもの

2004年 冬 単色ドット絵向上委員会