MIKBUG拡張版資料　　2018/04/30
By Old_Ordinary

外部から使用可能なサブルーチン　　色付け部分が変更・拡張部分です

| | |
|---|---|
| MIKBUG | Motorola版(オリジナル) |
| MIKBUG2 | Motorola版(参考用) |
| MIKBUG_V | vintagechips版 |
| MIKBUG_O | Old_Ordinary版 |
| FLCTSR | Flow Control Service Routine |

| 名前 (版) | 番地 MIKBUG | MIKBUG2 | MIKBUG_V | MIKBUG_O | FLCTSR | 機能・仕様 |
|---|---|---|---|---|---|---|
| ACIRXF | ----- | ----- | ----- | $E19F | $E242 | RXバッファの受信有無チェック(C=0：無、C=1：有) |
| INCH8 | ----- | ----- | ----- | $E1A4 | $E200 | １文字入力、ACA：8bit文字 |
| INEEE | $E1AC | $F966 | $E1AC | $E1AC | ----- | １文字入力、ACA：7bit文字(bit7=0) |
| (CVUPSW) | ----- | ----- | ----- | $1F15 | $1F15 | UPPER CASE変換制御($00：変換有、$00以外：変換無) |
| (OUTSW) | ----- | $A016 | Echo固定 | $1F16 | $1F16 | エコー制御($00：エコー有、$00以外：エコー無) |
| INITSW | ----- | ----- | ----- | $E1E7 | ----- | OUTSW、CVUPSW 初期化 |
| OUTEEE | $E1D1 | $F959 | $E1D1 | $E1D1 | | １文字出力、ACA：文字 |
| PDATA1 | $E07E | $F87E | $E07E | $E07E | | 文字列出力、IX：文字列先頭番地、ターミネータ：$04 |
| OUT2H | $E0BF | $F8BF | $E0BF | $E0BF | | OUTPUT 2 HEX CHAR |
| OUT2HS | $E0CA | $F8CA | $E0CA | $E0CA | | OUTPUT 2 HEX CHAR + SPACE |
| OUT4HS | $E0C8 | $F8C8 | $E0C8 | $E0C8 | | OUTPUT 4 HEX CHAR + SPACE |
| OUTS | $E0CC | $F8CC | $E0CC | $E0CC | | OUTPUT SPACE |
| OUTHL | $E067 | $F867 | $E067 | $E067 | | OUT HEX LEFT BCD DIGIT |
| OUTHR | $E06B | $F86B | $E06B | $E06B | | OUT HEX RIGHT BCD DIGIT |
| | | | | | | 廃止理由 |
| 廃止 SAV | $E1A5 | ----- | $E1A5*1 | ----- | | PIAを使わない(MIKBUG内で呼ぶ所が無い)(*1)消し忘れ |
| 廃止 DEL | $E1EF | ----- | ----- | ----- | | PIAを使わない(MIKBUG内で呼ぶ所が無い) |

MIKBUGに戻るためのエントリーポイント

| | | | | | | |
|---|---|---|---|---|---|---|
| START | $E0D0 | $F97B | $E0D0 | $E0D0 | | ACIA初期化後 CONTRLに戻る |
| 8N1_NIRQ | ----- | ($F99C) | $E0DE | $E0DE | | ACIA設定(8N1 NON-INTERRUPT)後 CONTRLに戻る |
| CONTRL | $E0E3 | $F9BA | $E0E3 | $E0E3 | | スタックの設定($1F42)後 CONTRLに戻る |
| SFE | $E113 | $F80A | $E113 | $E113 | | SWI 割込処理ルーチン、レジスタ表示後 CONTRLに戻る |
| LOAD19 | $E040 | (?---) | $E040 | $E040 | | ″?″ を出力後 CONTRLに戻る |

ワークエリア

| | | | | | | |
|---|---|---|---|---|---|---|
| RAM | $A000-7F | $A000-7F | $1F00-7F | $1F00-7F | $1F00-7F | モニターワーク |
| BUFFER | | | | | $1FC0-FF | バッファ |
| IOV | $A000 | | $1F00 | $1F00 | $1F00 | |
| XTEMP | $A012 | | $1F12*2 | -----*2 | $1F12 | (*2)SAV、DEL を廃止したため使わない |
| BEGA | $A002 | | $1F02 | $1F02 | | |
| ENDA | $A004 | | $1F04 | $1F04 | | |
| CHRCNT | | | | | $1F14 | |
| CVUPSW | | | | $1F15 | $1F15 | |
| OUTSW | | $1F16 | | $1F16 | $1F16 | |
| BPTW | | | | | $1F17 | |
| BPTR | | | | | $1F19 | |
| スタック設定 | $A042 | (????) | $1F42 | $1F42 | | |
| STARTV | $A048 | | $1F48 | $1F48 | | |
| TARGTV | | | | | $1F4A | |

ハードウェア

| | | | | | | |
|---|---|---|---|---|---|---|
| PIAS | $8005 | | | | | ソフトウェアシリアル |
| PIAD | $8004 | | | | | ソフトウェアシリアル |
| ACIACS | ($8008) | $8008 | $8018 | $8018 | $8018 | |
| ACIADA | ($8009) | $8009 | $8019 | $8019 | $8019 | RXD、TXD（コンソール） |

変更・追加部分の詳細

ACIRXF　RXバッファの受信有無チェック
　　A, B, IX は保存されます
　　戻りコード： C(キャリー)　C=0：無、C=1：有
　　チェック後すぐに戻ります

INCH8　　１文字入力　　8bit 文字
　　B, IX は保存されます
　　入力した文字は A に入る　　A：8bitデータ
　　文字が入力されるまで待ちます

INEEE　　１文字入力　　7bit 文字(bit7=0)
　　B, IX は保存されます
　　入力した文字は A に入る　　A：7bitデータ
　　文字が入力されるまで待ちます

変更点 OUTSW($1F16)が $00であればエコーする、$00以外であればエコーしない(MIKBUG2.0互換)
　　CVUPSW($1F15)が $00であれば英小文字を英大文字に変換する、$00以外であれば変換しない
　　初期化時に、OUTSW、CVUPSW を $00にしている

　　bit7(Parity)は外していますが、Rubout($FF)の読み捨ては止めました
　　理由は、テレタイプ対応のパディングキャラクタを外したり、PTR・PTPの制御を外した時点で
　　テレタイプ端末を使用しない前提であり、紙テープデータのRubout($FF)は読み込まないため
　　Rubout($FF)は、テレタイプで紙テープを打った時に、間違えてしまった文字を消すため
　　8bit全て鑽孔(さんこう)したコードであり、紙テープを使わなければ不要である

INITSW　OUTSW、CVUPSW 初期化
　　OUTSW、CVUPSW に $00 をセットする
　　A, B, IX は保存されます
　　CVUPSW($1F15)　　UPPER CASE変換制御($00：変換有、$00以外：変換無)
　　OUTSW($1F16)　　エコー制御($00：エコー有、$00以外：エコー無)

テレタイプ端末（電動機械式タイプライター）
https://ja.wikipedia.org/wiki/%E3%83%86%E3%83%AC%E3%82%BF%E3%82%A4%E3%83%97%E7%AB%AF%E6%9C%AB
紙テープリーダーとパンチを備えた ASR Teletype Model 33 は、コンピュータ用端末として使用可能
1963年 ： 低コスト化された全機械式の代表的な機種。大文字のみのASCIIコード仕様
その頃のコンピュータ端末の標準となった。

MIKBUGの環境

MIKBUGは、ASR Teletype Model 33 を接続する様に設計されている
電動機械式タイプライターに紙テープの鑽孔装置(PTP)と読取装置(PTR)が付属している
印刷速度は毎秒10文字で印刷できるのは大文字のみ　通信は ASCIIコード、110bps
PTPと PTRは制御文字で起動停止できる構造になっている(DC1:PTR ON、DC2:PTP ON、DC3:PTR OFF、DC4:PTP OFF)

１．テレタイプ・紙テープとの決別
今や、テレタイプは写真でしか見ることができません
マイコンの端末としては、パソコンで動かすターミナルソフトが一般的です
例えば、フリーソフトの Tera Termでは、
PTPの代わりに、Log機能が使えますし、PTRの代わりには Send File機能が使えます
通信速度も格段に速くなっています(ACIAで 115200bpsも可能)

MIKBUGのターミナルソフト向けの改造点
(１)紙テープ装置を制御する　制御コード DC1($11), DC2($12), DC3($13), DC4($14) の廃止
(２)印刷機のキャリッジ(活字ユニット)を左端に戻す CR(Carriage Return)の時に
　時間稼ぎのために送る文字(Padding character：NULやRuboutなど印刷されない文字) の廃止
(３)編集した紙テープ用の Rubout($FF)文字読み捨ての廃止
　テレタイプで紙テープを打った時に、間違えてしまった文字を消すために
　8bit全て鑽孔(さんこう)したコードであり、紙テープを使わなければ不要である

２．ソフトウェアシリアル変換(PIA)から UART(ACIA)へ
MIKBUGは、なぜか PIAを使ってソフトウェアシリアル変換を行っています
　利点は、ハードウェア(タイマーの時間設定)で通信速度を変えられる
　エコーの ON/OFF をハードウェアの設定で簡単に変えられる　などが有りますが
　欠点として、半二重通信しかできない
　通信速度が速くできない（私の実験では CPU:1MHzで 1200bpsが限界でした）

| |
|---|
| MIKBUG(ソフトウェア)にはエコーバック機能が有りません<br>ソフトウェアシリアル変換なので、エコーバック出来ないのが実情です<br>半二重通信なので、エコーバック送信している間に送られてくる文字は受信できません<br>エコーバックは、ハードウェアで行います<br>方法は、RXDと TXDのラインを（スイッチを介して）接続するだけです<br>半二重通信なので、送信中に自分の TXDを受信することは有りません |

MIKBUG以降のモニタでは、UART(ACIA)を使った通信が一般的になっています
そして、例外なく、エコーバックの有無を制御する機能（エコーフラグ）が追加されています
ソフトウェアシリアル変換(PIA)は処理が複雑ですが、ACIAでは処理が簡単になります

MIKBUGの ACIAへの改造点
(１)ACIAのコントロールレジスタの初期化や設定
(２)ACIAのデータレジスタへの読み書き

ハードウェアの改造点
(１)ACIAクロックの生成方法
　通信速度の 16倍の周波数(周期)のクロックが必要（一般的な UARTと同じです）

３．機能追加
(１)エコー制御をソフトウェア制御できる様にする
(２)英小文字を英大文字と同等に扱える様にする(ソフトウェア制御)
(３)外部から使える、ブレークチェック用のサブルーチンを準備する
(４)8bitデータの入力サブルーチンを準備する
(５)ROM化したアプリケーションに制御を移すコマンドを追加する(任意)