SCI
Asynchronies Communication Interface

RS-232

Tsinghua Freescale MCU/DSP Application Center

Asynchronies Serial Communication SCI

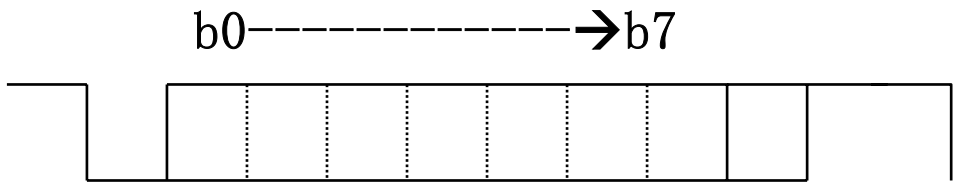
BUAD RATE:

1 Bit / Sec = 1 Buad (Bit Per Second, BPS)

standard rate(very wide bound wades):

110, 300, 600, 1200, 2400, 4800, 9600, 19200,38400…115200

data format:



Start Bit **1**

Data Bits **5, 6, 7, 8**

Parity Bit **Even Odd or None**

Stop Bit **1, 1.5, 2(at least)**

Asynchronies Serial Communication

Use Not Return to Zero (NRZ) code

T & R is synchronized by start bit detection

R Clock = T Clock x 16 (baud rate x 16), slow

3,5,7 clock detect fowling edge of start bit
8,9,10 detect bit's value **(Anti-Noise)**

The maximum speed: bus rate /16

Duplex and Half Duplex

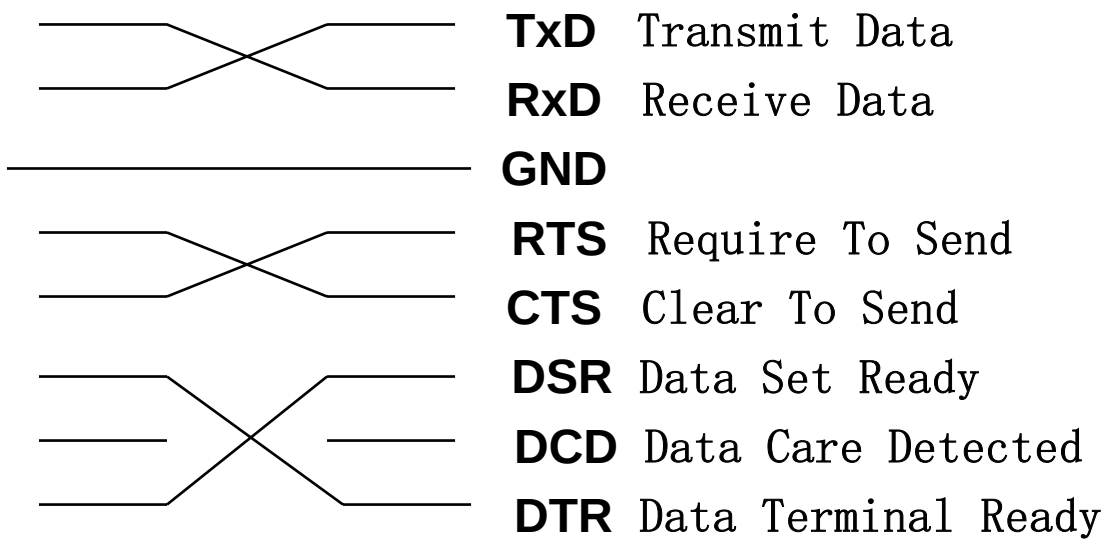
Asynchronies Serial Communication
and RS－232C

- Signals Voltage Levers

  1= -3V ~ - 12V

  0 = +3V ~ + 12V (150 feet)

- Signals definitions **(9** pins **D** type Connectors**)**
  - **DTE / DCE**

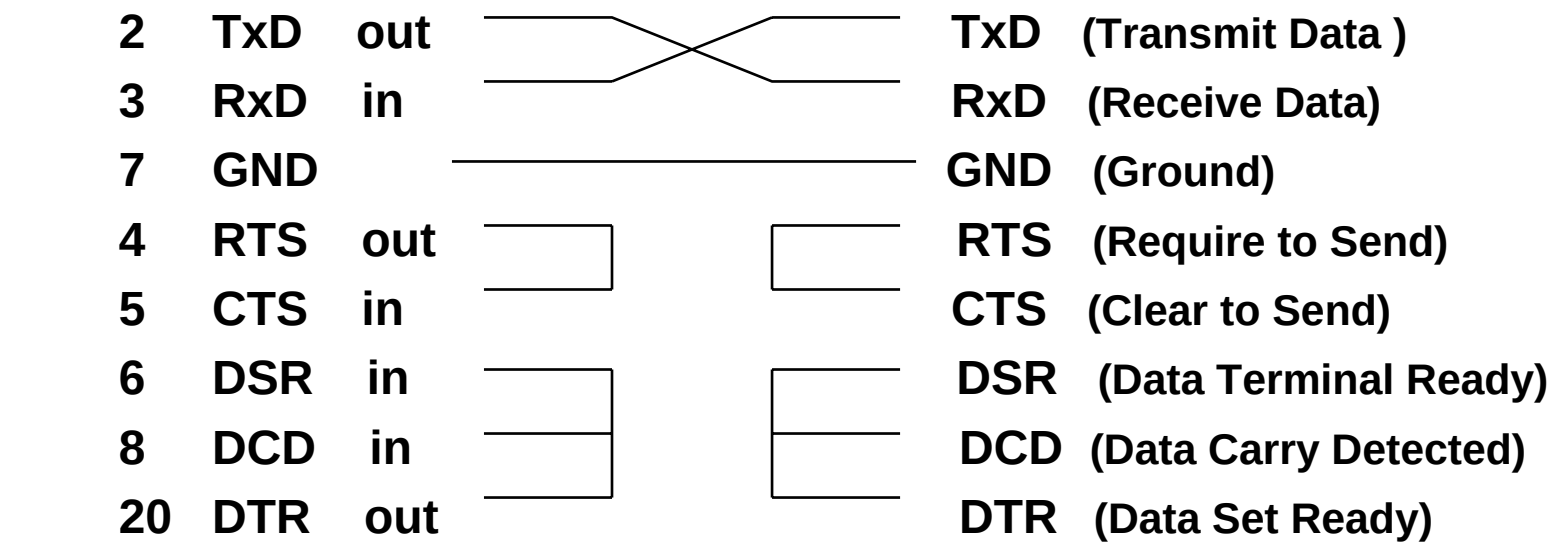


Asynchronies Serial Communication

RS－232C

3 wires connection: **(25** pins **D** type Connector**)**

DTE / DCE

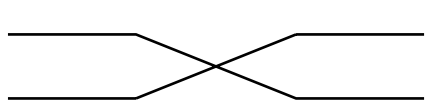


Ready always, or XON/XOFF protocol, ASCII only

Asynchronies Serial Communication

RS－232C

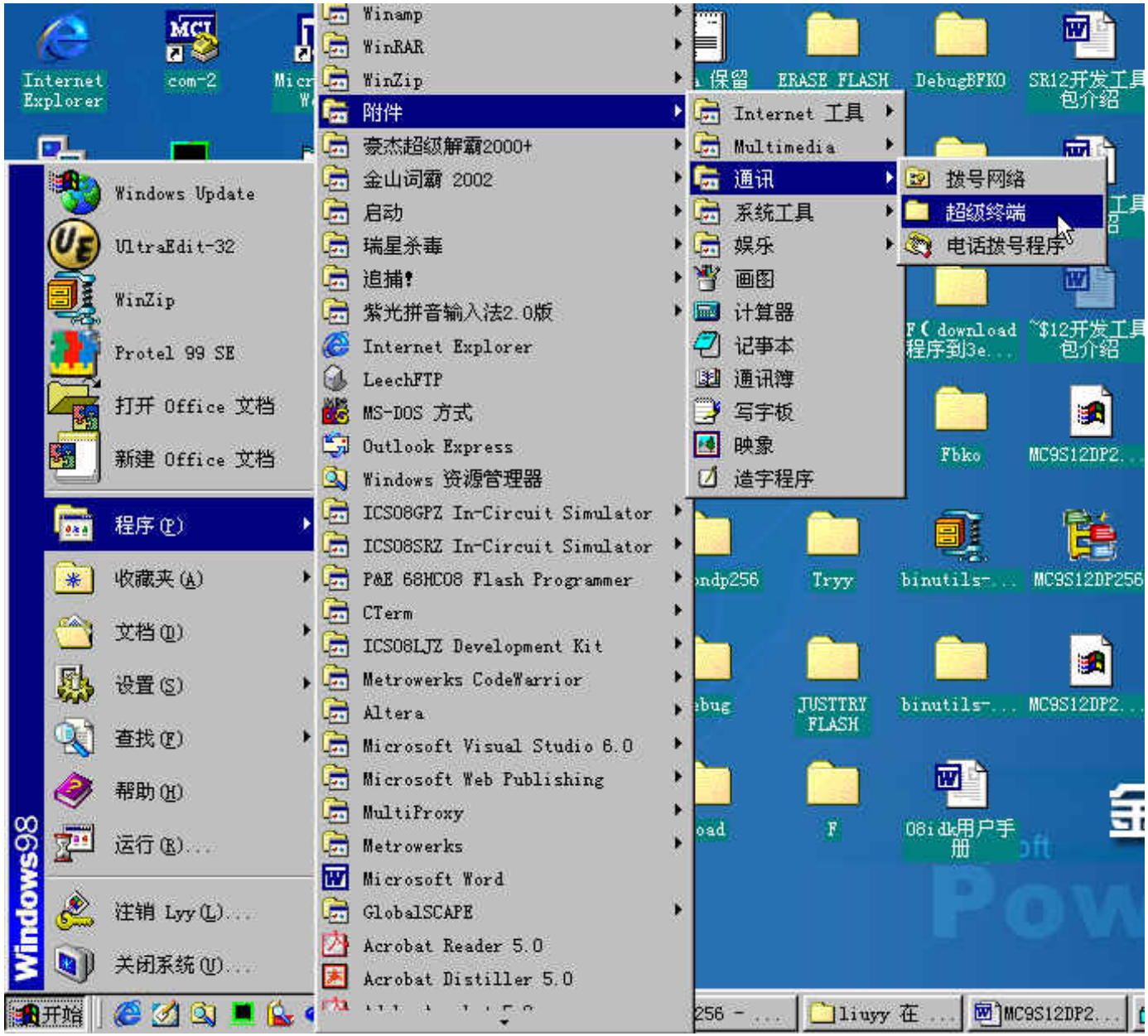
DTE (9 pins connector) in PC



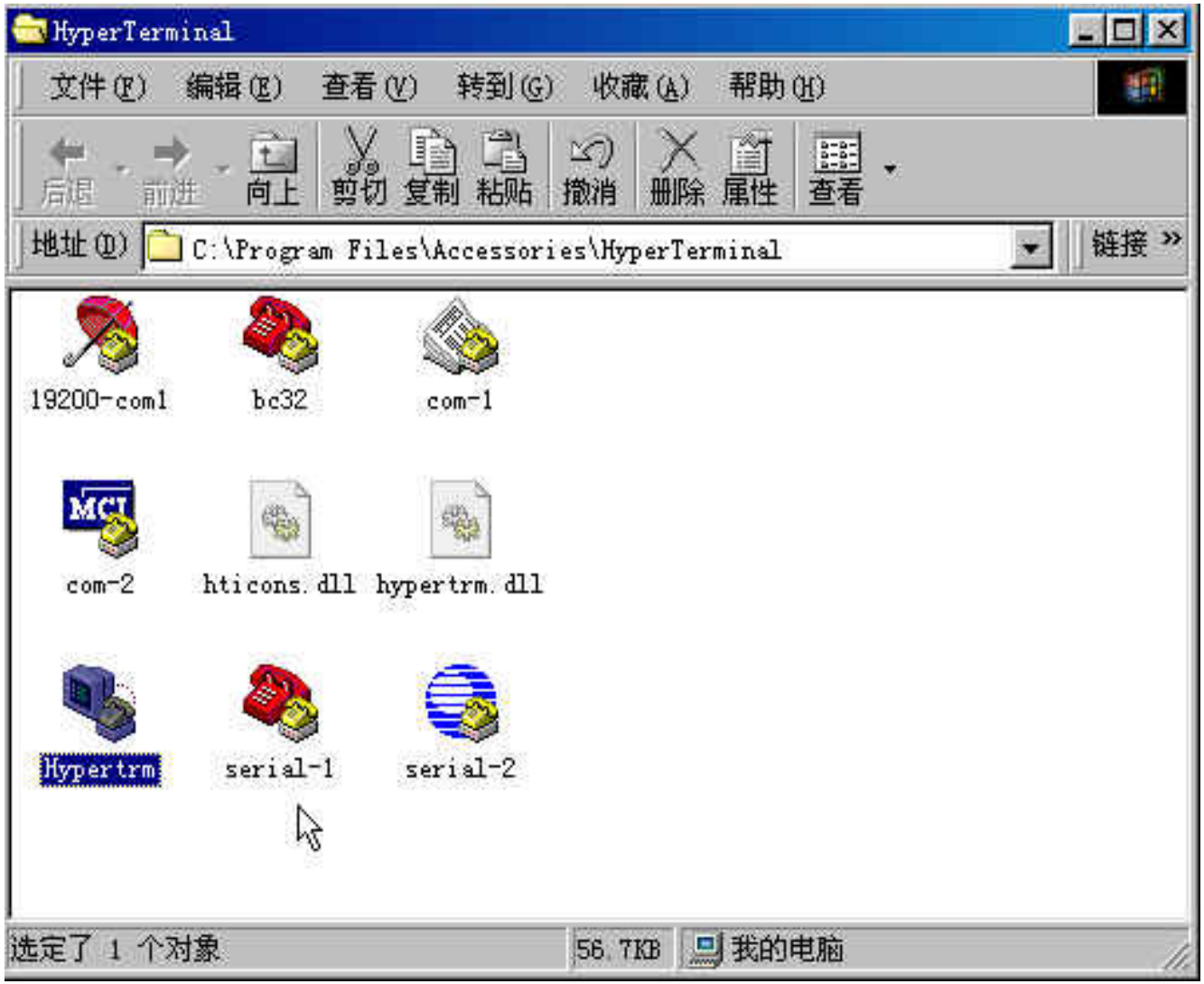
1	protection ground			
2	RxD	in	RxD	(Receive Data)
3	TxD	out	TxD	(Transmit Data)
5	GND		GND	(Ground)
6	DSR	in	DSR	(Data Terminal Ready)
7	RTS	out	RTS	(Require to Send)
8	CTS	in	CTS	(Clear to Send)
9	DCD	in	DCD	(Data Carrier Detected)
4	DTR	out	DTR	(Data Set Ready)

No Handshake, should use XON/XOFF protocol,ASCII only

Talk to PC


Internet Explorer
com-2
Windows Update
UltraEdit-32
WinZip
Protel 99 SE
打开 Office 文档
新建 Office 文档
程序(P)
收藏夹(A)
文档(D)
设置(S)
查找(F)
帮助(H)
运行(R)...
注销 Lyy(L)...
关闭系统(U)...
Windows98
开始
Winamp
WinRAR
WinZip
附件
豪杰超级解霸2000+
金山词霸 2002
启动
瑞星杀毒
追捕!
紫光拼音输入法2.0版
Internet Explorer
LeechFTP
MS-DOS 方式
Outlook Express
Windows 资源管理器
ICS08GPZ In-Circuit Simulator
ICS08SRZ In-Circuit Simulator
P&E 68HC08 Flash Programmer
CTerm
ICS08LJZ Development Kit
Metrowerks CodeWarrior
Altera
Microsoft Visual Studio 6.0
Microsoft Web Publishing
MultiProxy
Metrowerks
Microsoft Word
GlobalSCAPE
Acrobat Reader 5.0
Acrobat Distiller 5.0
Internet 工具
Multimedia
通讯
系统工具
娱乐
画图
计算器
记事本
通讯簿
写字板
映象
造字程序
拨号网络
超级终端
电话拨号程序
保留
ERASE FLASH
DebugBPKO
SR12开发工具包介绍
download 程序到3e...
$12开发工具包介绍
Fbko
MC9S12DP2...
ndp256
Tryy
binutils-...
MC9S12DP256
ebug
JUSTTRY FLASH
oad
F
08idk用户手册
liuyy 在 ...
MC9S12DP2...

Select Hypertrm

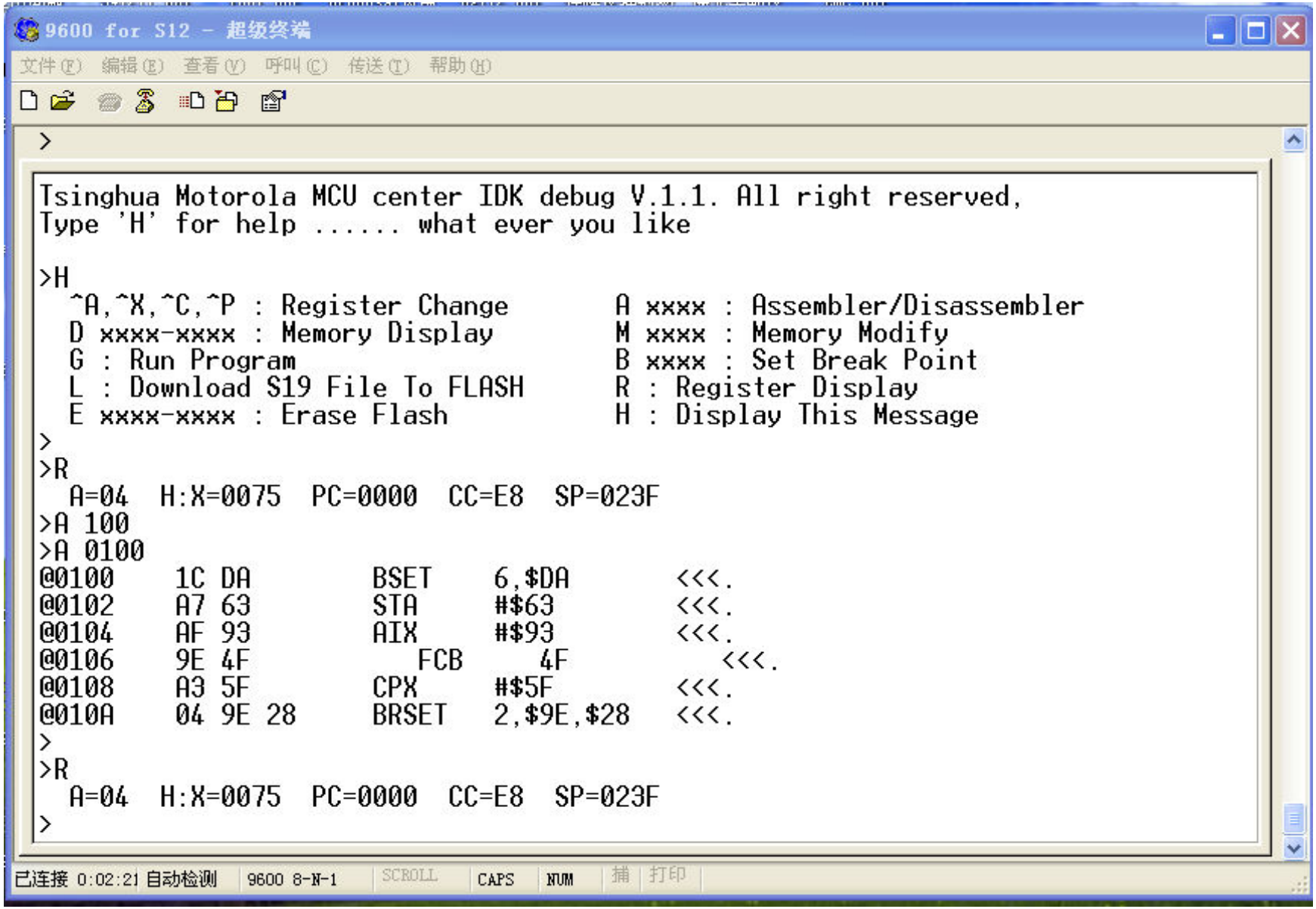


Your Protocol File Name

Select COM1 (or COM2)

Set Communication Protocol

Use SCI Talk to PC



System Initialization

- **Write CONFIG Register to Use Bus Clock as SCI Clock:**

```
CONFIG1 EQU   $1E          ; System configuration register1
CONFIG2 EQU   $1F
*
            LDA   #$01
            STA   CONFIG1      ; Use Bus Clock for SCI
            LDA   #$3D         ; LVI Disabled,
            STA   CONFIG2      ; COP disabled, STOP illegal
```

Bus clock /4 = SCI clock ; see page 244 Technical Data

Initial PLL Let Bus Clock = 2.4576MHz

```
CLR   PCTL         ; $0036 PLL Control Register
MOV   #$01,PCTL    ;SET P=0 E=1,Close PLL, Use CGMXCLK
LDA   #$01         ;SET N=300
STA   PMSH
LDA   #$2c
STA   PMSL
LDA   #$80            ;SET L=128
STA   PMRS
LDA   #$01            ;SET R=1
STA   PMDS
BSET  B_PLLON,PCTL    ;Open   PLL
BSET  B_AUTO,PBWC     ;Set AUTO
BRCLR B_LOCK,PBWC,*   ;Wait Until PLL Stable
BSET  B_BCS,PCTL      ;Use PLL
```

HC08GT60 SCI1 Module Registers

```
SCI1BDH   EQU   $18     ; SCI1 Baud Rate  Reg. H
SCI1BDL   EQU   $19     ; SCI1 Baud Rate  Reg. L
SCI1C1    EQU   $1A     ; SCI1 Control Reg.1
SCI1C2    EQU   $1B     ; SCI1 Control Reg.2
SCI1S1    EQU   $1C     ; SCI1 Status  Reg.1
SCI1S2    EQU   $1D     ; SCI1 Status   Reg.2
SCI1C3    EQU   $1E     ; SCI1 Control  Reg.3
SCI1D     EQU   $1F     ; SCI1 Data     Reg.
```

SCI Initialization

Setup communication protocol:

- **Write to Baud Rate Register($19) Initial the baud rate,**
- **Set Enable SCI bit in SCC1 ($13) Register, Use Default: 8bits, stop bits, Parity Check…**
- **Enable T & R in SCC2**

Diff. Form MCU to MCU

Initial SCI1 Baud Rate Register

SCI1BDH = $0018
SCIB1DL = $0019

0	0	0	BR 12	BR 11	BR 10	BR 9	BR 8	BR 7	BR 6	BR 5	BR 4	BR 3	BR 2	BR 1	BR 0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BR = 0 ~ 8191
Baud Rate = BUSCLK / (16*BR)
If Busclk = 4MHz, for 9600: BR= 26.0416667

```
LDA   #26          ;
STA   SCI1BDL      ; 4M/16/26=9600
```

SCI1 Control Register 1 : SCI1C1

$1A	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
Reset	0	0	0	0	0	0	0	0

LOOPS=0 **NOT Self Loop Mode**

SCISWAI = 0 In Wait mode SCI clock NOT freeze

RSRC=0 **Receiver Source Select in LOOP mode**

M = 0 **8 Bits, NOT 9 Bits**

WAKE=0 **Wake up Mode IS Idle, NOT Address**

ILT=0 **Idle Line Type Counter form Start Bits**

PE =0 **No Parity**

PT =0 **If Parity, Use Even**

Use Default. Not necessary to be initialized

SCI Control Register 2 : SCI1C2

	B7	6	5	4	3	2	1	0
$1B	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK

TIE=0 **Transmit Buffer Empty IRQ Disabled**
TCIE =0 **Transmit Complete IRQ Disabled**
RIE=0 **Receiver IRQ Disabled**
ILIE =0 **Idle Line IRQ Disabled**
TE = 0 **Transmit Disabled**
RE = 0 **Receive Disabled**
RWU =0 **Receiver Is Not in Receiver Wake Up Mode**
SBK = 0 **Transmit Break Bit, Not Break.**

```
BSET 3,SCC2     ; Enable T
BSET 2,SCC2     ; Enable R
```

SCI Initialization

```
* Initial SCI

        LDA  #$02          ; /4
        STA  BAUD          ; 2.4576M/4/4/16=9600
 //       BSET 6,SCC1            ; Enable SCI
        BSET 3,SCC2        ; Enable T
        BSET 2,SCC2        ; Enable R

* Already did in the debug
```

Transmit & Receive with SCI

SCI1S1 States Register ($1C)

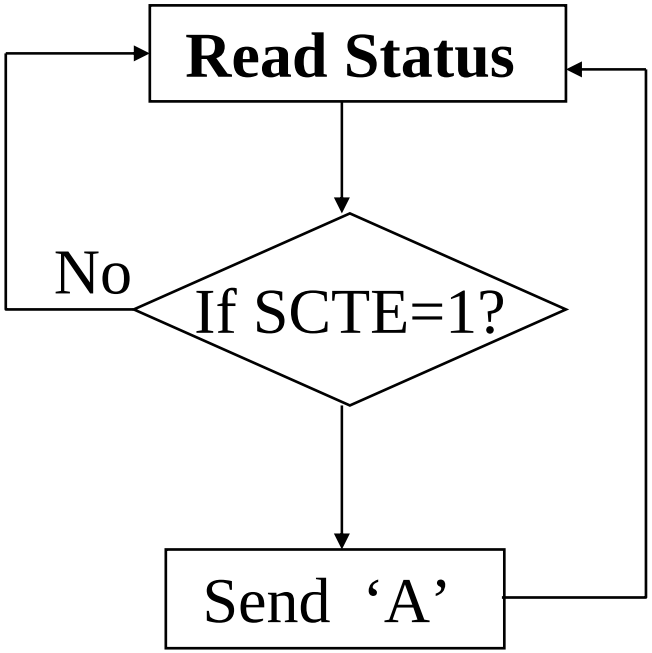
	B7	6	5	4	3	2	1	0
$1C	TDRE	TC	SCRF	NF	OR	NF	FE	PE

TDRE=1, Transmit Data Reg. Empty

RDRF=1, Receive Data Req. Full

SCI1 Data Register ($1F)

Transmit

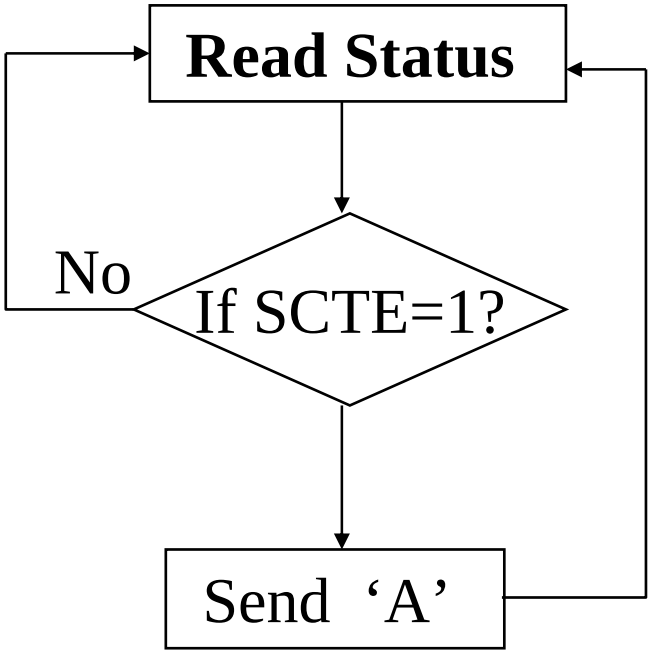



```
*
OUTA    LDA     $1C
        ANDA    #$80
        BEQ     OUTA
        LDA     #'A'
        STA     $1F
        JMP     OUTA
*
```

Assemble by Hand

```
                                ORG     $100
                        *
0100   B6  1C    OUTA      LDA     $1C
0102   A4  80              AND     #$80
0104   27  -6(FA)          BEQ     OUTA
0106   A6  41            LDA     #'A'
0108   B7  1F            STA     $1F
010A   CC 01  00          JMP     OUTA
                        *
```

Transmit




```
SCI1S1    EQU   $1C
SCI1D     EQU   $1F
TDRE:     EQU    7
*
          ORG   $100
*
OUTCH   LDA     #'A'
BUSY    BRCLR   TDRE,SCI1S1,*
        STA     SCID
        BRA     OUTCH
*
```

HIWARE HC08-Assembler

```
Abs. Rel.    Loc     Obj. code    Source line
---- ----    ------ ---------    -----------
   1    1           0000 0016    SCSR     EQU    $1C
   2    2           0000 0018    SCDR     EQU    $1F
   3    3                        *
   4    4                                 ORG    $100
   5    5                        *
   6    6   a000100 A641        OUTCH    LDA    #'A'
   7    7   a000102 0F 1CFD     IFEMPT   BRCLR  7,SCI1S1,*
   8    8   a000105 B71F                 STA    SCI1D
   9    9   a000107 20F7                 BRA    OUTCH
```

Simplest SCI Transmit & Receive

```
;Test RDRF (b_5) bit until it's set
INCH    BRCLR   5,SCSR,INCH
        LDA     SCDR
        RTS
;Test RDRF (b_7) bit until it's set
OUTCH   BRCLR 7,SCSR,*
        STA     SCDR
        RTS
```

```
char Get_char()
  { while (SCSR&0x20);
       return(*SCDR);
  }

Put_Char(char a)

   {while (SCSR&0x80);
       *SCDR = a;
   }
```

Low lever I/O functions for C

100% CPU time (for 9600b/s, ~ 1mS/char)

SCI Interrupts

There are 3 Interrupt sources in SCI:

- **V18 Transmit IRQ(V $FFDA/DB)**
  - **Transmit Register Empty Interrupt**
    - » **Next Char can be written**
  - **Transmit Completed Interrupt**
    - » **Drive one wire to three status or Receive**
- **V17 Receive IRQ (V $FFDC/DD)**
  - **Receiver Register Full Interrupt**
- **V16 Error Interrupt(V $FFDE/DF)**
  - **frame,Parity,Overflow…**
- **Before Enable IRQ Remember to Initial Vectors !**
- **Because of the Monitor, Victors moved !**

Enable the SCI Interrupt

- Write to SCI1C2 Interrupt control Register to enable the IRQ:

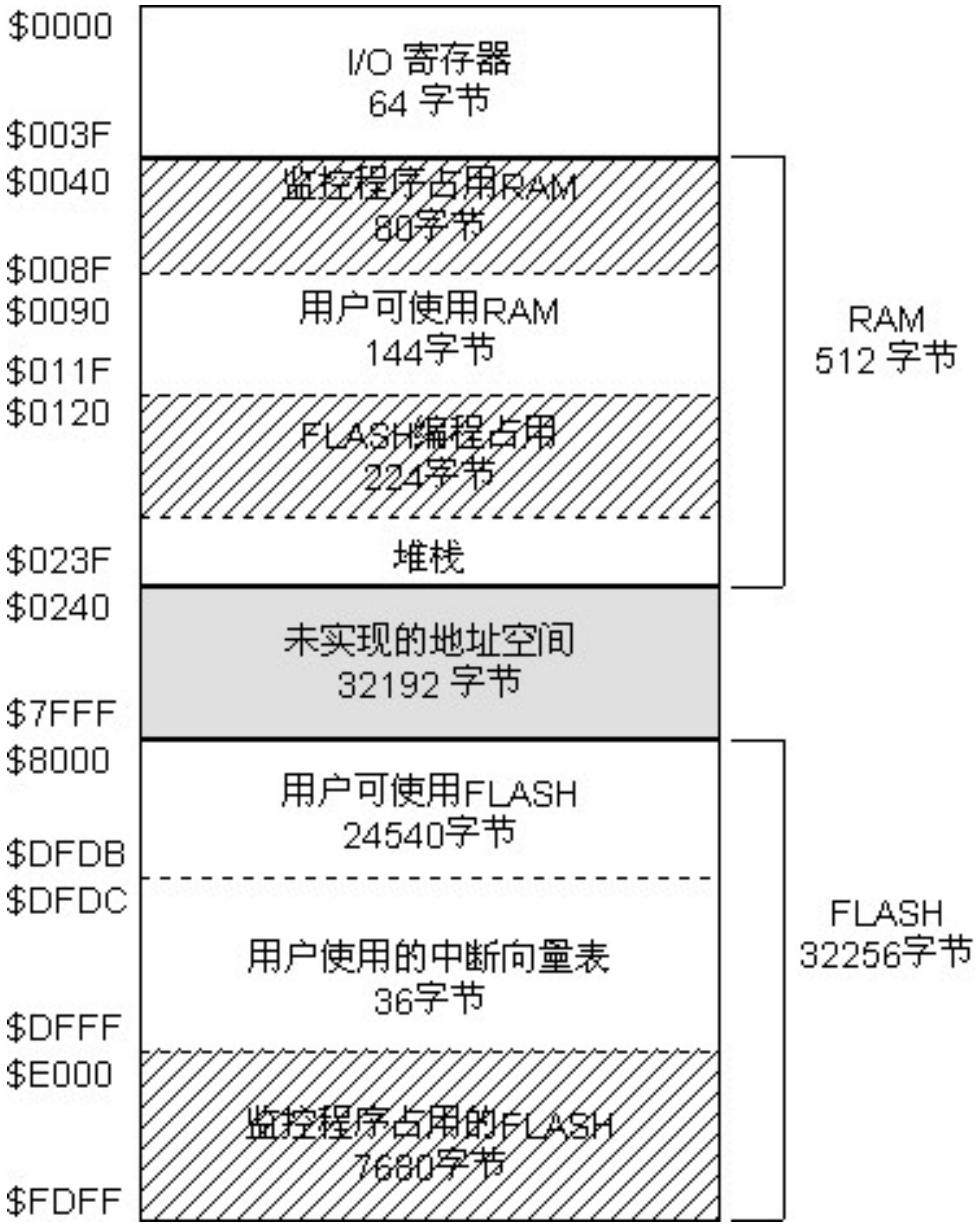
$1B	TIE	TCIE	RIE	IEIL	TE	RE	RWU	SBK

- Initial the Victors before the enable!
- Use CLI to Enable IRQ!

System Interrupt Vector Table

IF18	$FFDA,$FFDB	CSI1 Transmit
IF17	$FFDC ,$FFDD	SCI1 Receive
IF16	$FFDE,$FFDF	SCI1 Error
IF15	$FFED,$FFE1	SPI
IF14	$FFE2,$FFE3	TIM2 Overflow
IF13	$FFE4,$FFE5	TIM2 Channel 4
IF12	$FFE6,$FFE7	TIM2 Channel 3
IF11	$FFE8,$FFE9	TIM2 Channel 2
IF10	$FFEA,$FFEB	TIM2 Channel 1
IF9	$FFEC,$FFED	TIM2 Channel 0
IF8	$FFEE,$FFEF	TIM1 Overflow
IF7	$FFF0,$FFF1	TIM1 Channel 2
IF6	$FFF2,$FFF3	TIM1 Channel 1
IF5	$FFF4,$FFF5	TIM1 Channel 0
IF4	$FFF6,$FFF7	ICG
IF3	$FFF8,$FFF9	LVDF
IF2	$FFFA,$FFFB	IRQ
IF1	$FFFC.$FFFD	SWI
IF0	$FFFE, $FFFF	Reset

User Interrupt Vector Table



User:

$DFDA,$DFDB SCI Transmit

$DFDC,$DFDD SCI Receive

System:

$FFDA,$FFDB SCI Transmit

$FFDC,$FFDD SCI Receive

In the Monitor Program:

```
ORG  VECTORSTART

DW   ISR_TIMB          ;  Time Base
DW   ISR_ADC           ;  ADC is done
DW   ISR_KEY           ;  Keyboard
DW   ISR_SCI_TR        ;  SCI Transmit done   ($FFDA,$FFDB)
DW   ISR_SCI_RX        ;  SCI Received        ($FFDC,$FFDD)
DW   ISR_SCI_ER        ;  SCI Error
DW   ISR_SPI_TR        ;  SPI Transmit
……   ……
DW   ISR_LVD           ; Low Voltage Detected
DW   ISR_IRQ           ;  External IRQ
DW   ISR_SWI           ;  Software IRQ
DW   RESET_INIT        ;  Reset
```

```
VectorStart    EQU    $FFCC    ; is Defined in the file GT60.h
```

ISR_SCI_TR

```
ISR_SCI_TR      LDX   V_SCI_TR
                LDA   V_SCI_TR+1
                CMP   #$FF
                BNE   ISR_SCI_TR_X
                CPX   #$FF
                BNE   ISR_SCI_TR_X
                LDX   MON_DUMMY_VEC
                LDA   MON_DUMMY_VEC+1

ISR_SCI_TR_X    JMP   ISRCOMM
```

V_SCI_TR

```
                ORG  $DFCC
V_RTIF          DW $FFFF
V_IIC           DW  $FFFF
V_ADC           DW  $FFFF
V_KBI           DW  $FFFF
V_SCI2_TR       DW  $FFFF
V_SCI2_RX       DW  $FFFF
V_SCI2_ER       DW  $FFFF
V_SCI1_TR       DW  $FFFF
V_SCI1_RX       DW  $FFFF
V_SCI1_ER       DW  $FFFF
V_SPI_TR        DW  $FFFF
.............   ......  .............
.............   ......  .............
V_PLL           DW  $FFFF
V_IRQ           DW  $FFFF
V_SWI           DW  $FFFF
V_RESET         DW  $8000
```

ISRCOMM

```
ISRCOMM   STX    INTJMP+1
          STA    INTJMP+2
          JMP    INTJMP
```

INTJMP

```
INTJMP   RMB   3

WAIT_FAIL:
          ……    ……
          ……    ……
          LDA   V_RESET
          LDX   V_RESET+1
          STA   INTJMP+1
          STX   INTJMP+2
          LDA   #$CC    ;Opcode of <JMP addr16>
          STA   INTJMP
          JMP   INTJMP
```

ISR for Receiver

```
IRQGETCH  LDA     $1C        ;Clear IRQ
          LDA     $1F
          JSR     OUTCH
          RTI
*
OUTCH     BRCLR   7,SCSR,*
          STA     SCDR
          RTS
*
          ORG     $DFE4
          FDB     IRQGETCH
*
```

Do not Forget Enable IRQ By: CLI

Summery of SCI Receive IRQ

- Initial SCI (Monitor did already for you)
- Write Interrupt service Routine
  - PUSH H, If you use it
  - Clear IRQ Flag
  - Get the Character
  - PULL H If you did Push
  - RTI
- Initial INT Victor Table (User victor Table: $DFE4…)
- Enable SCI Receive Interrupt
- Enable System IRQ in CCR

Motorola S Code

S00D0000746F776572732E616273EA
S123F0009BCCF4190000F30E000000011570001F01A0000F026F01EF020100000580000068
S109F020FFFF00000000E8
S105FFFEF419F0
S123F028A7FECE1054898ACE1055650000931F9EEF028B869EE7019F8B8848594859898A63
S123F04897D6100BC71055D6100AC71054654F8395E601FEA702818789A7FA9E6F049E6F2B
……
……
……
S123F3A8F014F7205395F687EE018AE6019EE706F69EE705AD509EEE02878AE6019EE7044A
S123F3C8F69EE703AD4095F795F687EE018AF68795E60387EE048A86F7956C0326026C028A
S123F3E86C0126017C6D0526026A046A05E60487EE058A65000092D095F687EE018AE60153
S123F408FA26A2A70A819FAB029EE7048B86A90081C6F005A502260DC6F00CB7FEC6F00D0D
S116F428B7FF55FE94CDF331CEF006898ACEF007FD20DEA8
S105F0260000E4
S903F0000C