
CAN Program Examples

1. Introduction

This Application Note provides to customers C and Assembler program examples for UART.

These examples are developed for the different configuration modes of this feature.

1.1 References

Atmel 8051 Microcontrollers Hardware Manual.



**8051
Microcontrollers**

Application Note

4347A–CAN–06/04



2. C Example

2.1 Program

```
/**
 * @file $RCSfile: can.c,v $
 *
 * Copyright (c) 2004 Atmel.
 *
 * Please read file license.txt for copyright notice.
 *
 * @brief This file is an example to use can networking.
 *
 * This file can be parsed by Doxygen for automatic documentation
 * generation.
 * Put here the functional description of this file within the software
 * architecture of your program.
 *
 * @version $Revision: 1.0 $ $Name: $
 */

/* @section INCLUDES */
#include "t89c51cc01.h"

/* baud rate and bit timing parameters */
#define BRP_500k 0x00
#define SJW_500k 0x00
#define PRS_500k 0x02
#define PHS1_500k 0x03
#define PHS2_500k 0x03

unsigned char num_channel, num_data;

/**
 * FUNCTION_PURPOSE: This file set up Can at 500Kbauds with channel 0 id 0x123
 * in reception
 * and channel 1 id 0x001 in emission.
 * FUNCTION_INPUTS: void
 * FUNCTION_OUTPUTS: void
 */
void main(void)
{
    CANGCON |= MSK_CANGCON_GRES; /* reset CAN */
    /* reset all mailboxes */
    for (num_channel = 0; num_channel < 15; num_channel++)
    {
        CANPAGE = num_channel << 4;
        CANCONCH = CH_DISABLE;
        CANSTCH = 0;
        CANIDT1 = 0;
        CANIDT2 = 0;
    }
}
```

```

CANIDT3  = 0;
CANIDT4  = 0;
CANIDM1  = 0;
CANIDM2  = 0;
CANIDM3  = 0;
CANIDM4  = 0;
for (num_data = 0; num_data < 8; num_data++) CANMSG = 0;
}
/* setup bit timing */
CANBT1   = BRP_500k << 1;           /* BRP=0x00; */
CANBT2   &= ~0x60;                  /* reset SJW */
CANBT2   |= SJW_500k << 5;          /* SJW=0x00; */
CANBT2   &= ~0x0E;                  /* reset PRS */
CANBT2   |= PRS_500k << 1;          /* PRS=0x02; */
CANBT3   &= ~0x70;                  /* reset PHS2 */
CANBT3   |= PHS2_500k << 4;         /* PHS2=0x03; */
CANBT3   &= ~0x0E;                  /* reset PHS1 */
CANBT3   |= PHS1_500k << 1;         /* PHS1=0x03 */
CANGCON  |= MSK_CANGCON_ENA;        /* start CAN */

/* Channel 0 init */
CANPAGE = (0 << 4);                 /* CHNB=0x00; select channel 0 */
CANSTCH = 0x00;                     /* reset channel status */
CANCONCH = CH_DISABLE;              /* reset control and dlc register */

/* Channel 0: identifier = 11bits. CANIDT=0x123 */
CANIDT1 = 0x24;
CANIDT2 &= ~0x80;
CANIDT2 |= 0x60;

/* Channel 0: mask = 11bits. 0x7F0 */
CANIDM1 = 0xFE;
CANIDM2 &= ~0xE0;
CANIDM4 = 0;

/* Channel 0 configuration */
CANIDT4 &= ~0x04;                   /* clear bit rtr in CANIDT4. */
CANCONCH |= DLC_MAX;                /* Reception 8 bytes. */
CANCONCH |= CH_RxENA;               /* Reception enabled without
buffer. */

/* Channel 1 init */
CANPAGE = (1 << 4);                 /* CHNB=0x01; select channel 1 */
CANSTCH = 0x00;                     /* reset channel status */
CANCONCH = CH_DISABLE;              /* reset control and dlc register */

/* Channel 1: identifier = 11bits. CANIDT=0x001 */
CANIDT1 = 0x80;
CANIDT2 &= ~0xC0;
CANIDT2 |= 0x20;

```

```

/* interrupt configuration */
CANIE2|=0x01;                                /* IECH0=1 */
CANGIE |= MSK_CANGIE_ENTX;                    /* Can_Tx IT enable */
CANGIE |= MSK_CANGIE_ENRX;                    /* Can_Rx IT enable */
ECAN = 1;                                     /* CAN IT enable */
EA = 1;                                       /* all IT enable */

while(1);                                    /* endless */
}

/**
 * FUNCTION_PURPOSE: can interrupt. echo receive data on channel 0 reception.
 * Reception id between 0x120 and 0x12F.
 * FUNCTION_INPUTS: P4.1(RxDC) can input
 * FUNCTION_OUTPUTS: P4.0(TxDC) can output
 */
can_it(void) interrupt 7
{
char save_canpage;
char i;                                       /* can_data index */
char can_data[8];

save_canpage = CANPAGE;                     /* save current context */

/* echo receive data on channel 0 reception */
CANPAGE = (0 << 4);                          /* CHNB=0x00; select channel 0 */
if(CANSTCH==MSK_CANSTCH_RxOk)
{
    for (i=0; i<8; i++) can_data[i] = CANMSG; /* save receive data */
    CANPAGE = (1 << 4);                        /* CHNB=0x00; select channel 1 */
    /* Channel 1 configuration */
    CANCONCH = CH_DISABLE;                     /* reset channel 1 configuration */
    /*
    for (i=0; i<8; i++) CANMSG = can_data[i]; /* load saved data */
    CANCONCH |= DLC_MAX;                       /* transmit 8 bytes */
    CANCONCH |= CH_TxENA;                      /* emission enabled */
    CANEN2 |= (1 << 1);                       /* channel 1 enable */
    CANSTCH=0x00;                             /* reset channel 1 status */
    */
}
CANPAGE = (0 << 4);                          /* CHNB=0x00; select channel 0 */
CANCONCH = CH_DISABLE;                       /* reset channel 0 configuration */
CANCONCH |= DLC_MAX;                         /* receive 8 bytes */
CANCONCH |= CH_RxENA;                        /* reception enable */
CANEN2 |= (1 << 0);                          /* channel 0 enable */
CANSTCH=0x00;                               /* reset channel 0 status */

CANPAGE= save_canpage;                       /* restore saved context */
CANGIT = 0x00;                              /* reset all flags */
}

```

2.2 SFR Register Definition

```
/*H*****
**
* NAME: T89C51CC01.h
*-----
-
* PURPOSE: include file for KEIL
*****
*/
#ifndef _T89C51CC01_H_

#define _T89C51CC01_H_

#define Sfr(x, y)  sfr x = y
#define Sbit(x, y, z)  sbit x = y^z
#define Sfr16(x,y) sfr16 x = y

/*-----*/
/* Include file for 8051 SFR Definitions */
/*-----*/

/* BYTE Register */
Sfr (P0 , 0x80);
Sfr (P1 , 0x90);

Sbit (P1_7, 0x90, 7);
Sbit (P1_6, 0x90, 6);
Sbit (P1_5, 0x90, 5);
Sbit (P1_4, 0x90, 4);
Sbit (P1_3, 0x90, 3);
Sbit (P1_2, 0x90, 2);
Sbit (P1_1, 0x90, 1);
Sbit (P1_0, 0x90, 0);

Sfr (P2 , 0xA0);

Sbit (P2_7 , 0xA0, 7);
Sbit (P2_6 , 0xA0, 6);
Sbit (P2_5 , 0xA0, 5);
Sbit (P2_4 , 0xA0, 4);
Sbit (P2_3 , 0xA0, 3);
Sbit (P2_2 , 0xA0, 2);
Sbit (P2_1 , 0xA0, 1);
Sbit (P2_0 , 0xA0, 0);

Sfr (P3 , 0xB0);

Sbit (P3_7 , 0xB0, 7);
Sbit (P3_6 , 0xB0, 6);
```

```

Sbit (P3_5 , 0xB0, 5);
Sbit (P3_4 , 0xB0, 4);
Sbit (P3_3 , 0xB0, 3);
Sbit (P3_2 , 0xB0, 2);
Sbit (P3_1 , 0xB0, 1);
Sbit (P3_0 , 0xB0, 0);

Sbit (RD , 0xB0, 7);
Sbit (WR , 0xB0, 6);
Sbit (T1 , 0xB0, 5);
Sbit (T0 , 0xB0, 4);
Sbit (INT1, 0xB0, 3);
Sbit (INT0, 0xB0, 2);
Sbit (TXD , 0xB0, 1);
Sbit (RXD , 0xB0, 0);

Sfr (P4 , 0xC0);

Sfr (PSW , 0xD0);

Sbit (CY , 0xD0, 7);
Sbit (AC , 0xD0, 6);
Sbit (F0 , 0xD0, 5);
Sbit (RS1 , 0xD0, 4);
Sbit (RS0 , 0xD0, 3);
Sbit (OV , 0xD0, 2);
Sbit (UD , 0xD0, 1);
Sbit (P , 0xD0, 0);

Sfr (ACC , 0xE0);
Sfr (B , 0xF0);
Sfr (SP , 0x81);
Sfr (DPL , 0x82);
Sfr (DPH , 0x83);

Sfr (PCON , 0x87);
Sfr (CKCON , 0x8F);

/*----- TIMERS registers -----*/
Sfr (TCON , 0x88);
Sbit (TF1 , 0x88, 7);
Sbit (TR1 , 0x88, 6);
Sbit (TF0 , 0x88, 5);
Sbit (TR0 , 0x88, 4);
Sbit (IE1 , 0x88, 3);
Sbit (IT1 , 0x88, 2);
Sbit (IE0 , 0x88, 1);
Sbit (IT0 , 0x88, 0);

Sfr (TMOD , 0x89);

```

```

Sfr (T2CON , 0xC8);
Sbit (TF2 , 0xC8, 7);
Sbit (EXF2 , 0xC8, 6);
Sbit (RCLK , 0xC8, 5);
Sbit (TCLK , 0xC8, 4);
Sbit (EXEN2 , 0xC8, 3);
Sbit (TR2 , 0xC8, 2);
Sbit (C_T2 , 0xC8, 1);
Sbit (CP_RL2, 0xC8, 0);

```

```

Sfr (T2MOD , 0xC9);
Sfr (TL0 , 0x8A);
Sfr (TL1 , 0x8B);
Sfr (TL2 , 0xCC);
Sfr (TH0 , 0x8C);
Sfr (TH1 , 0x8D);
Sfr (TH2 , 0xCD);
Sfr (RCAP2L , 0xCA);
Sfr (RCAP2H , 0xCB);
Sfr (WDTRST , 0xA6);
Sfr (WDTPRG , 0xA7);

```

```

/*----- UART registers -----*/

```

```

Sfr (SCON , 0x98);
Sbit (SM0 , 0x98, 7);
Sbit (FE , 0x98, 7);
Sbit (SM1 , 0x98, 6);
Sbit (SM2 , 0x98, 5);
Sbit (REN , 0x98, 4);
Sbit (TB8 , 0x98, 3);
Sbit (RB8 , 0x98, 2);
Sbit (TI , 0x98, 1);
Sbit (RI , 0x98, 0);

```

```

Sfr (SBUF , 0x99);
Sfr (SADEN , 0xB9);
Sfr (SADDR , 0xA9);

```

```

/*----- ADC registers -----*/

```

```

Sfr (ADCLK , 0xF2);
Sfr (ADCON , 0xF3);
#define MSK_ADCON_PSIDLE 0x40
#define MSK_ADCON_ADEN 0x20
#define MSK_ADCON_ADEOC 0x10
#define MSK_ADCON_ADSST 0x08
#define MSK_ADCON_SCH 0x07
Sfr (ADDL , 0xF4);
#define MSK_ADDL_UTILS 0x03

```

```

Sfr (ADDH , 0xF5);
Sfr (ADCF , 0xF6);

/*----- FLASH EEPROM registers -----*/
Sfr (FCON , 0xD1);
#define MSK_FCON_FBUSY 0x01
#define MSK_FCON_FMOD 0x06
#define MSK_FCON_FPS 0x08
#define MSK_FCON_FPL 0xF0
Sfr (EECON , 0xD2);
#define MSK_EECON_EEBUSY 0x01
#define MSK_EECON_EEE 0x02
#define MSK_EECON_EEPL 0xF0
Sfr (AUXR , 0x8E);
#define MSK_AUXR_M0 0x20
Sfr (AUXR1 , 0xA2);
#define MSK_AUXR1_ENBOOT 0x20
/*----- IT registers -----*/
Sfr (IPL1 , 0xF8);
Sfr (IPH1 , 0xF7);
Sfr (IEN0 , 0xA8);
Sfr (IPL0 , 0xB8);
Sfr (IPH0 , 0xB7);
Sfr (IEN1 , 0xE8);

/* IEN0 */
Sbit (EA , 0xA8, 7);
Sbit (EC , 0xA8, 6);
Sbit (ET2 , 0xA8, 5);
Sbit (ES , 0xA8, 4);
Sbit (ET1 , 0xA8, 3);
Sbit (EX1 , 0xA8, 2);
Sbit (ET0 , 0xA8, 1);
Sbit (EX0 , 0xA8, 0);

/* IEN1 */
Sbit (ETIM , 0xE8, 2);
Sbit (EADC , 0xE8, 1);
Sbit (ECAN , 0xE8, 0);

/*----- PCA registers -----*/
Sfr (CCON , 0xD8);
Sbit(CF , 0xD8, 7);
Sbit(CR , 0xD8, 6);
Sbit(CCF4, 0xD8, 4);
Sbit(CCF3, 0xD8, 3);
Sbit(CCF2, 0xD8, 2);
Sbit(CCF1, 0xD8, 1);
Sbit(CCF0, 0xD8, 0);

```



```

Sfr (CMOD , 0xD9);
Sfr (CH , 0xF9);
Sfr (CL , 0xE9);
Sfr (CCAP0H , 0xFA);
Sfr (CCAP0L , 0xEA);
Sfr (CCAPM0 , 0xDA);
Sfr (CCAP1H , 0xFB);
Sfr (CCAP1L , 0xEB);
Sfr (CCAPM1 , 0xDB);
Sfr (CCAP2H , 0xFC);
Sfr (CCAP2L , 0xEC);
Sfr (CCAPM2 , 0xDC);
Sfr (CCAP3H , 0xFD);
Sfr (CCAP3L , 0xED);
Sfr (CCAPM3 , 0xDD);
Sfr (CCAP4H , 0xFE);
Sfr (CCAP4L , 0xEE);
Sfr (CCAPM4 , 0xDE);

/*----- CAN registers -----*/
Sfr (CANGIT , 0x9B);
#define MSK_CANGIT_CANIT0x80
#define MSK_CANGIT_OVRTIM      0x20
#define MSK_CANGIT_OVRBUF0x10
#define MSK_CANGIT_SERG0x08
#define MSK_CANGIT_CERG0x04
#define MSK_CANGIT_FERG0x02
#define MSK_CANGIT_AERG0x01

Sfr (CANTEC , 0x9C);
Sfr (CANREC , 0x9D);
Sfr (CANTCON , 0xA1);
Sfr (CANMSG , 0xA3);
Sfr (CANTTCL , 0xA4);
Sfr (CANTTCH , 0xA5);
Sfr (CANGSTA , 0xAA);
#define MSK_CANGSTA_OVFG0x40
#define MSK_CANGSTA_TBSY0x10
#define MSK_CANGSTA_RBSY0x08
#define MSK_CANGSTA_ENFG0x04
#define MSK_CANGSTA_BOFF0x02
#define MSK_CANGSTA_ERRP0x01

Sfr (CANGCON , 0xAB);
#define MSK_CANGCON_ABRQ      0x80
#define MSK_CANGCON_OVRQ      0x40
#define MSK_CANGCON_TTC      0x20
#define MSK_CANGCON_SYNCTTC      0x10
#define TTC_EOF      0x10
#define TTC_SOF      0x00

```

```

#define MSK_CANGCON_AUTBAUD      0x08
#define MSK_CANGCON_ENA    0x02
#define MSK_CANGCON_GRES 0x01

Sfr (CANTIML , 0xAC);
Sfr (CANTIMH , 0xAD);
Sfr (CANSTMP_L , 0xAE);
Sfr (CANSTMP_H , 0xAF);
Sfr (CANPAGE , 0xB1);
Sfr (CANSTCH , 0xB2);
#define MSK_CANSTCH_DLCW    0x80
#define MSK_CANSTCH_TxOk    0x40
#define MSK_CANSTCH_RxOk    0x20
#define MSK_CANSTCH_BERR    0x10
#define MSK_CANSTCH_SERR    0x08
#define MSK_CANSTCH_CERR    0x04
#define MSK_CANSTCH_FERR    0x02
#define MSK_CANSTCH_AERR    0x01

Sfr (CANCONCH , 0xB3);
#define MSK_CANCONCH_IDE    0x10
#define MSK_CANCONCH_DLC    0x0F
#define MSK_CANCONCH_CONF    0xC0
#define DLC_MAX      8
#define CH_DISABLE    0x00
#define CH_RxENA      0x80
#define CH_TxENA      0x40
#define CH_RxBENA      0xC0

Sfr (CANBT1 , 0xB4);
#define CAN_PRESCALER_MIN    0
#define CAN_PRESCALER_MAX    63

Sfr (CANBT2 , 0xB5);
#define MSK_CANBT2_SJW    0x60
#define MSK_CANBT2_PRS    0x0E
#define CAN_SJW_MIN    0
#define CAN_SJW_MAX    3
#define CAN_PRS_MIN    0
#define CAN_PRS_MAX    7

Sfr (CANBT3 , 0xB6);
#define MSK_CANBT3_PHS2    0x70
#define MSK_CANBT3_PHS1    0x0E
#define CAN_PHS2_MIN    0
#define CAN_PHS2_MAX    7
#define CAN_PHS1_MIN    0
#define CAN_PHS1_MAX    7

```

```

Sfr (CANSIT1 , 0xBA);
Sfr (CANSIT2 , 0xBB);
Sfr (CANIDT1 , 0xBC);
Sfr (CANIDT2 , 0xBD);
Sfr (CANIDT3 , 0xBE);
Sfr (CANIDT4 , 0xBF);
#define MSK_CANIDT4_RTRTAG 0x04

Sfr (CANGIE , 0xC1);
#define MSK_CANGIE_ENRX      0x20
#define MSK_CANGIE_ENTX     0x10
#define MSK_CANGIE_ENERCH   0x08
#define MSK_CANGIE_ENBUF    0x04
#define MSK_CANGIE_ENERG    0x02

Sfr (CANIE1 , 0xC2);
Sfr (CANIE2 , 0xC3);
Sfr (CANIDM1 , 0xC4);
Sfr (CANIDM2 , 0xC5);
Sfr (CANIDM3 , 0xC6);
Sfr (CANIDM4 , 0xC7);
#define MSK_CANIDM4_RTRMSK 0x04
#define MSK_CANIDM4_IDEMSK 0x01

Sfr (CANEN1 , 0xCE);
Sfr (CANEN2 , 0xCF);

#endif

```

3. Assembly 51 Examples

3.1 Program

```

$INCLUDE    (t89c51cc01.INC)

num_channel DATA 10H
num_data DATA 11H
data0 DATA 12H
data1 DATA 13H
data2 DATA 14H
data3 DATA 15H
data4 DATA 16H
data5 DATA 17H
data6 DATA 18H
data7 DATA 19H

; /* baud rate and bit timing parameters */
#define BRP_500k  00
#define SJW_500k   00
#define PRS_500k   02
#define PHS1_500k  03
#define PHS2_500k  03

org 000h
ljmp begin

org 3Bh
ljmp can_it

; /**
; * FUNCTION_PURPOSE: This file set up Can at 500Kbauds with channel 0 id
0x123 in reception
; * and channel 1 id 0x001 in emission.
; * FUNCTION_INPUTS: void
; * FUNCTION_OUTPUTS: void
; */

org 0100h
begin:
ORL CANGCON,#01h;                                ; /* reset CAN */

; /* reset all mailboxes */
MOV     num_channel,#00h
reset_mailbox:
    MOV     A,num_channel                        ; /* load accumulator with channel
value */
    SWAP     A
    ANL     A,#0F0h
    MOV     CANPAGE,A                            ; /* CHNB=num_channel; */
    MOV     CANCONCH,#00h                        ; /* channel disable */

```

```

MOV      CANSTCH,#00h
MOV      CANIDT1,#00h
MOV      CANIDT2,#00h
MOV      CANIDT3,#00h
MOV      CANIDT4,#00h
MOV      CANIDM1,#00h
MOV      CANIDM2,#00h
MOV      CANIDM3,#00h
MOV      CANIDM4,#00h
MOV      num_data,#00h
reset_data:
    MOV      CANMSG,#00h
    INC      num_data
    MOV      A,num_data
    CJNE     A,#08h,reset_data
    INC      num_channel
    MOV      A,num_channel
    CJNE     A,#0Fh,reset_mailbox

; /* setup bit timing */
MOV      A,#BRP_500k
RL       A
MOV      CANBT1,A
ANL      CANBT2,#9Fh                ; /* reset SJW */
MOV      A,#SJW_500k
SWAP     A
RL       A
ANL      A,#0E0h
ORL      CANBT2,A                ; /* SJW=0x00; */
ANL      CANBT2,#0F1h            ; /* reset PRS */
MOV      A,#PRS_500k
RL       A
ORL      CANBT2,A                ; /* PRS=0x02; */
ANL      CANBT3,#8Fh            ; /* reset PHS2 */
MOV      A,#PHS2_500k
SWAP     A
ANL      A,#0F0h
ORL      CANBT3,A                ; /* PHS2=0x03; */
ANL      CANBT3,#0F1h            ; /* reset PHS1 */
MOV      A,#PHS1_500k
RL       A
ORL      CANBT3,A                ; /* PHS1=0x03 */
ORL      CANGCON,#02h            ; /* start CAN */
; /* Channel 0 init */
MOV      A,#00h                ; /* load accumulator with channel value */
; /*
SWAP     A
ANL      A,#0F0h
MOV      CANPAGE,A                ; /* CHNB=0x00; select channel 0 */
MOV      CANSTCH,#00h            ; /* reset channel status */

```

```

MOV      CANCONCH,#00h                /* reset control and dlc register */
/* Channel 0: identifier = 11bits. CANIDT=0x123 */
MOV      CANIDT1,#24h
ANL      CANIDT2,#07Fh
ORL      CANIDT2,#060h
/* Channel 0: mask = 11bits. 0x7F0 */
MOV      CANIDM1,#0FEh
ANL      CANIDM2,#1Fh
MOV      CANIDM4,#00h
/* Channel 0 configuration */
ANL      CANIDT4,#0FBh                /* clear bit rtr in CANIDT4. */
ORL      CANCONCH,#08h                /* Reception 8 bytes.*/
ORL      CANCONCH,#80h                /* Reception enabled without buffer.*/
/* Channel 1 init */
MOV      A,#01h                      /* load accumulator with channel value
*/
SWAP     A
ANL      A,#0F0h
MOV      CANPAGE,A                  /* CHNB=0x01; select channel 1 */
MOV      CANSTCH,#00h                /* reset channel status */
MOV      CANCONCH,#00h                /* reset control and dlc register */
/* Channel 1: identifier = 11bits. CANIDT=0x001 */
MOV      CANIDT1,#80h
ANL      CANIDT2,#3Fh
ORL      CANIDT2,#20h
/* interrupt configuration */
ORL      CANIE2,#01h                /* IECH0=1 */
ORL      CANGIE,#10h                /* Can_Tx IT enable */
ORL      CANGIE,#20h                /* Can_Rx IT enable */
SETB     ECAN                        /* CAN IT enable */
SETB     EA                          /* all IT enable */
JMP      $                          /* endless */

/**
 * FUNCTION_PURPOSE: can interrupt. echo receive data on channel 0 reception.
 * Reception id between 0x120 and 0x12F.
 * FUNCTION_INPUTS: P4.1(RxDC) can input
 * FUNCTION_OUTPUTS: P4.0(TxDC) can output
 */
can_it:
MOV      R7,CANPAGE                  /* save current context */
/* set channel */
MOV      A,#00h                      /* load accumulator with channel value
*/
SWAP     A
ANL      A,#0F0h
MOV      CANPAGE,A                  /* CHNB=0x00; select channel 0 */
/* echo receive data on channel 0 reception */
MOV      A,CANSTCH
CJNE     A,#20h,end_if_rxok

```

```

        /* save receive data */
        MOV     data0,CANMSG
        MOV     data1,CANMSG
        MOV     data2,CANMSG
        MOV     data3,CANMSG
        MOV     data4,CANMSG
        MOV     data5,CANMSG
        MOV     data6,CANMSG
        MOV     data7,CANMSG
        /* set channel */
        MOV     A,#01h                      /* load accumulator with channel value
*/
        SWAP    A
        ANL     A,#0F0h
        MOV     CANPAGE,A                  /* CHNB=0x00; select channel 1 */
        /* Channel 1 configuration */
        MOV     CANCONCH,#00h;             /* reset channel 1 configuration */
        /* load saved data */
        MOV     CANMSG,data0
        MOV     CANMSG,data1
        MOV     CANMSG,data2
        MOV     CANMSG,data3
        MOV     CANMSG,data4
        MOV     CANMSG,data5
        MOV     CANMSG,data6
        MOV     CANMSG,data7

        ORL     CANCONCH,#08h              /* transmit 8 bytes */
        ORL     CANCONCH,#40h              /* emission enabled */
        ORL     CANEN2,#02h                /* channel 1 enable */
        MOV     CANSTCH,#00h               /* reset channel 1 status */

end_if_rxok:

        /* set channel */
        MOV     A,#00h                      /* load accumulator with channel value
*/
        SWAP    A
        ANL     A,#0F0h
        MOV     CANPAGE,A                  /* CHNB=0x00; select channel 0 */
        MOV     CANCONCH,#00h              /* reset channel 0 configuration */
        ORL     CANCONCH,#08h              /* receive 8 bytes */
        ORL     CANCONCH,#80h              /* reception enable */
        ORL     CANEN2,#01h                /* channel 0 enable */
        MOV     CANSTCH,#00h              /* reset channel 0 status */
        MOV     CANPAGE,R7                 /* restore saved context */
        MOV     CANGIT,#00h                /* reset all flags */

        RETI
end

```

3.2 SFR Register Definition

```

; *INC*****
**
; NAME: 89C51CC01.inc
;-----
-
; PURPOSE: for Keil
;*****
**

;-----
; Include file for 8051 SFR Definitions
;-----

; BYTE Register
P0      DATA    80H
P1      DATA    90H
P2      DATA    0A0H

P3      DATA    0B0H
RD      BIT      0B7H
WR      BIT      0B6H
T1      BIT      0B5H
T0      BIT      0B4H
INT1    BIT      0B3H
INT0    BIT      0B2H
TXD     BIT      0B1H
RXD     BIT      0B0H

P4      DATA    0C0H

PSW     DATA    0D0H
CY      BIT      0D7H
AC      BIT      0D6H
F0      BIT      0D5H
RS1     BIT      0D4H
RS0     BIT      0D3H
OV      BIT      0D2H
P       BIT      0D0H

ACC     DATA    0E0H
B       DATA    0F0H
SP      DATA    81H
DPL     DATA    82H
DPH     DATA    83H
PCON    DATA    87H
CKCON   DATA    8FH

;----- TIMERS registers -----
TCON    DATA    88H

```


TF1	BIT	8FH
TR1	BIT	8EH
TF0	BIT	8DH
TR0	BIT	8CH
IE1	BIT	8BH
IT1	BIT	8AH
IE0	BIT	89H
IT0	BIT	88H

TMOD	DATA	89H
------	------	-----

T2CON	DATA	0C8H
TF2	BIT	0CFH
EXF2	BIT	0CEH
RCLK	BIT	0CDH
TCLK	BIT	0CCH
EXEN2	BIT	0CBH
TR2	BIT	0CAH
C_T2	BIT	0C9H
CP_RL2	BIT	0C8H

T2MOD	DATA	0C9H
TL0	DATA	8AH
TL1	DATA	8BH
TL2	DATA	0CCH
TH0	DATA	8CH
TH1	DATA	8DH
TH2	DATA	0CDH
RCAP2L	DATA	0CAH
RCAP2H	DATA	0CBH
WDTRST	DATA	0A6H
WDTPRG	DATA	0A7H

;----- UART registers -----

SCON	DATA	98H
SM0	BIT	9FH
FE	BIT	9FH
SM1	BIT	9EH
SM2	BIT	9DH
REN	BIT	9CH
TB8	BIT	9BH
RB8	BIT	9AH
TI	BIT	99H
RI	BIT	98H

SBUF	DATA	99H
SADEN	DATA	0B9H
SADDR	DATA	0A9H

;----- ADC registers -----

```
ADCLK DATA0F2H
ADCON DATA0F3H
ADDL DATA0F4H
ADDH DATA0F5H
ADCF DATA0F6H
```

```
;----- FLASH EEPROM registers -----
```

```
FPGACON DATA0F1H
FCON DATA0D1H
EECON DATA0D2H
AUXR DATA8EH
AUXR1 DATA0A2H
```

```
;----- IT registers -----
```

```
IPL1 DATA0F8H
IPH1 DATA0F7H
IEN0 DATA0A8H
IPL0 DATA0B8H
IPH0 DATA0B7H
IEN1 DATA0E8H
```

```
; IEN0
```

```
EA BIT 0AFH
EC BIT 0AEH
ET2 BIT 0ADH
ES BIT 0ACH
ET1 BIT 0ABH
EX1 BIT 0AAH
ET0 BIT 0A9H
EX0 BIT 0A8H
```

```
; IEN1
```

```
ETIM BIT 0EAH
EADC BIT 0E9H
ECAN BIT 0E8H
```

```
;----- PCA registers -----
```

```
CCON DATA0D8H
CF BIT 0DFH
CR BIT 0DEH
CCF4BIT0D4H
CCF3BIT0D3H
CCF2BIT0D2H
CCF1BIT0D1H
CCF0BIT0D0H
```

```
CMOD DATA0D9H
CH DATA0F9H
CL DATA0E9H
CCAP0H DATA0FAH
```

```
CCAP0L DATA0EAH
CCAPM0 DATA0DAH
CCAP1H DATA0FBH
CCAP1L DATA0EBH
CCAPM1 DATA0DBH
CCAP2H DATA0FCH
CCAP2L DATA0ECH
CCAPM2 DATA0DCH
CCAP3H DATA0FDH
CCAP3L DATA0EDH
CCAPM3 DATA0DDH
CCAP4H DATA0FEH
CCAP4L DATA0EEH
CCAPM4 DATA0DEH
```

```
;----- CAN registers -----
```

```
CANGIT DATA 09BH
CANTEC DATA 09CH
CANREC DATA 09DH
CANTCON DATA 0A1H
CANMSG DATA 0A3H
CANTTCL DATA 0A4H
CANTTCH DATA 0A5H
CANGSTA DATA 0AAH
CANGCON DATA 0ABH
CANTIML DATA 0ACH
CANTIMH DATA 0ADH
CANSTMPL DATA 0AEH
CANSTMPH DATA 0AFH
CANPAGE DATA 0B1H
CANSTCH DATA 0B2H
CANCONCH DATA 0B3H
CANBT1 DATA 0B4H
CANBT2 DATA 0B5H
CANBT3 DATA 0B6H
CANSIT1 DATA 0BAH
CANSIT2 DATA 0BBH
CANIDT1 DATA 0BCH
CANIDT2 DATA 0BDH
CANIDT3 DATA 0BEH
CANIDT4 DATA 0BFH
CANGIE DATA 0C1H
CANIE1 DATA 0C2H
CANIE2 DATA 0C3H
CANIDM1 DATA 0C4H
CANIDM2 DATA 0C5H
CANIDM3 DATA 0C6H
CANIDM4 DATA 0C7H
CANEN1 DATA 0CEH
CANEN2 DATA 0CFH
```



Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe

Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia

Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan

9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory

2325 Orchard Parkway
San Jose, CA 95131
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers

2325 Orchard Parkway
San Jose, CA 95131
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards

Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive

Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

e-mail

literature@atmel.com

Web Site

<http://www.atmel.com>

Disclaimer: Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

©Atmel Corporation 2004. All rights reserved. Atmel, the Atmel logo, and combinations thereof are registered trademarks of Atmel Corporation or its subsidiaries. Windows® Windows 98™, Windows XP™, and Windows 2000™ are trademarks and/or registered trademark of Microsoft Corporation. Other terms and product names in this document may be the trademarks of others.



Printed on recycled paper.