

CAN-Part 프로그램 소스 및 설명

1. 초기화

```

#pragma model(kc)
#pragma interrupt (InterruptCan = 7)
#include <kc_funcs.h>
#include <kc_sfrs.h>
#include "Can_Vnr.h"

/* for can commnuication headfile */

#define CONTROL_REG          0x1c00      // CONTRL REGISTER
#define COMMAND_REG          0x1c01      // COMMAND REGISTER
#define STATUS_REG           0x1c02      // STATUS REGISTER
#define INTERRUPT_REG        0x1c03      // INTERRUPT REGISTER
#define ACCEPTANCE_CODE_REG  0x1c04      // ACCEPTANCE CODE REGISTER
#define ACCEPTANCE_MASK_REG  0x1c05      // ACCEPTANCE MASK REGISTER
#define BUS_TIMING_0_REG     0x1c06      // BUS TIMING 0 REGISTER
#define BUS_TIMING_1_REG     0x1c07      // BUS TIMING 1 REGISTER
#define OUTPUT_CONTROL_REG   0x1c08      // OUTPUT CONTROL REGISTER

#define TRANS_ID_0            0x1c0a      // TRANSMIT DATA IDENTIFIER
#define TRANS_DATA_LENGTH     0x1c0b      // TRANSMIT DATA DATA LENGTH
#define TRANS_DATA_BYTE_1     0x1c0c      // TRANSMIT DATA BYTE #1
#define TRANS_DATA_BYTE_2     0x1c0d      // TRANSMIT DATA BYTE #2
#define TRANS_DATA_BYTE_3     0x1c0e      // TRANSMIT DATA BYTE #3
#define TRANS_DATA_BYTE_4     0x1c0f      // TRANSMIT DATA BYTE #4
#define TRANS_DATA_BYTE_5     0x1c10      // TRANSMIT DATA BYTE #5
#define TRANS_DATA_BYTE_6     0x1c11      // TRANSMIT DATA BYTE #6
#define TRANS_DATA_BYTE_7     0x1c12      // TRANSMIT DATA BYTE #7
#define TRANS_DATA_BYTE_8     0x1c13      // TRANSMIT DATA BYTE #8

#define RECEIVE_ID_0          0x1c14      // RECEIVE DATA IDENTIFIER
#define RECEIVE_DATA_LENGTH   0x1c15      // RECEIVE DATA DATA LENGTH
#define RECEIVE_DATA_BYTE_1   0x1c16      // RECEIVE DATA BYTE #1
#define RECEIVE_DATA_BYTE_2   0x1c17      // RECEIVE DATA BYTE #1

```

```

#define RECEIVE_DATA_BYTE_3      0x1c18      // RECEIVE DATA BYTE #1
#define RECEIVE_DATA_BYTE_4      0x1c19      // RECEIVE DATA BYTE #1
#define RECEIVE_DATA_BYTE_5      0x1c1a      // RECEIVE DATA BYTE #1
#define RECEIVE_DATA_BYTE_6      0x1c1b      // RECEIVE DATA BYTE #1
#define RECEIVE_DATA_BYTE_7      0x1c1c      // RECEIVE DATA BYTE #1
#define RECEIVE_DATA_BYTE_8      0x1c1d      // RECEIVE DATA BYTE #1

#define CLOCK_DIVIDER_REG        0xe01f      // CLOCK DIVIDER

// Control register define
#define CAN_RESET_REQUEST        0x01
#define INIT_CAN_CONTROL        0x42      // 0100 0010 receive interrupt enable

// Command register define
#define CANREFRESH_CMD          0x0e
#define RECEIVE_CMD              0x04
#define TRANSMIT_CMD            0x01

// Status register define
#define CAN_BUS_OFF              0x80      // HIGH : CAN bus OFF, LOW : CAN bus ON
#define ERROR_STATUS            0x40      // HIGH : error counted, LOW : communication ok
#define BUS_BUSY                0x30      // HIGH : CAN bus busy, LOW : CAN bus idle
#define TRANSMIT_COMPLETE        0x08      // HIGH : transmit complete, LOW : incomplete
#define TRANSMIT_BUF_RELEASE     0x04      // HIGH : released buffer, LOW : can not access
                                         buffer
#define DATA_OVERRUN           0x02      // HIGH : CAN data overrun, LOW : no overrun
#define RECEIVE_BUF_FULL        0x01      // HIGH : buffer full, LOW : buffer empty

// Interrupt register define
#define OVERRUN_INT              0x08
#define ERROR_INT                0x04
#define ANY_ERROR_INT           0x0C
#define TRANSMIT_INT             0x02
#define RECEIVE_INT              0x01

// CAN bus timing register define
#define BAUD0                    0x80
#define BAUD1                    0x23      // 1M baud rate

```

```

#define BAUD0                0x81
#define BAUD1                0x23        // 500k baud rate
#define BAUD1                0xb2        // 1011 0010 500k baud rate

// Output control register define
#define OUTPUT_CONTROL        0xfa
#define R_BUF_SIZE            200        /* CAN_RECEIVE_BUF_SIZE */
unsigned char ReBuf[R_BUF_SIZE][10];
int            ReBufBegin;
int            ReBufEnd;

unsigned char CanDataError;

```

2. CAN 통신 읽기 Function

- ReadCan Function : CAN Controller에 있는 특정 번지 데이터를 읽어서 값을 전송함.
- CANGet Function : CAN-Bus에서 취득된 데이터들(1~8Byte)을 배열 형태로 불러오는 기능을 수행함.

```

/*****
    CAN Communication Function Define
    *****/
unsigned char ReadCan(unsigned int address)
{
    unsigned char *addr_ptr;
    unsigned char candata;

    addr_ptr = (unsigned char *) address;
    candata = *addr_ptr;
    return candata;
}

void CanGet(unsigned char *TempData)
{
    char i;

    if(ReBufBegin == ReBufEnd){
        *(TempData+0) = 0;
        return;
    }

```

```

}

if(++ReBufBegin > R_BUF_SIZE-1)          /* make buffer appear */
    ReBufBegin = 0;                      /* circular. */
for(i=0 ; i < 10 ; i++) *(TempData+i) = ReBuf[ReBufBegin][i];
}

```

3. CAN 통신 쓰기 Function

- WriteCan Function : CAN Controller의 Resister를 변경할 때 사용함.
- CanPut Function : CAN-Bus를 통해 데이터를 송신할 때 사용함.

```

void WriteCan(unsigned int address, unsigned char data)
{
    unsigned char *addr_ptr;

    addr_ptr = (unsigned char *) address;
    *addr_ptr = data;
}

void CanPut(unsigned char * TransData)
{
    unsigned char i, dummy;
    unsigned char can_status, count;
    int DataError, RepeatCount;

    DataError = 0;
    can_status = ReadCan(STATUS_REG);
    if(can_status & ERROR_STATUS) WriteCan(COMMAND_REG,CANREFRESH_CMD);
    while (can_status&BUS_BUSY) can_status = ReadCan(STATUS_REG);

    DataError = 0;
    WriteCan(TRANS_ID_0,      *(TransData+0));
    WriteCan(TRANS_DATA_LENGTH,*(TransData+1));
    count = *(TransData+1) & 0x0f;
    for(i=0 ; i < count ; i++) WriteCan( TRANS_DATA_BYTE_1+i , *(TransData+i+2) );
    WriteCan(COMMAND_REG, TRANSMIT_CMD);
}

```

4. CAN Controller 초기화 및 Reset 함수

```

void InitCan(unsigned char ACCEPTANCE_CODE,unsigned char ACCEPTANCE_MASK)
{
    unsigned char wsr_image = wsr;
    wsr = 0;
    disable();
    int_mask &= 0x7f;
    enable();

    WriteCan(CONTROL_REG,      CAN_RESET_REQUEST);
    WriteCan(ACCEPTANCE_CODE_REG,  ACCEPTANCE_CODE);
    WriteCan(ACCEPTANCE_MASK_REG,  ACCEPTANCE_MASK);
    WriteCan(BUS_TIMING_0_REG,   BAUD0);
    WriteCan(BUS_TIMING_1_REG,   BAUD1);
    WriteCan(OUTPUT_CONTROL_REG,  OUTPUT_CONTROL);
    ReadCan(INTERRUPT_REG);
    WriteCan(CONTROL_REG,      INIT_CAN_CONTROL);
    WriteCan(COMMAND_REG,      CANREFRESH_CMD);

    CanDataError = 0;
    ReBufBegin = 0;
    ReBufEnd = 0;
    disable();
    int_mask |= 0x80;
    enable();
    wsr = wsr_image;
}

void ResetCan(void)
{
    unsigned int wreg;
    unsigned char i,j;

    ioport2 &= 0xBF;
    for(i=0;i<100;i++) for(j=0;j<10;j++);
    ioport2 |= 0x40;
}

```

5. CAN Interrupt & CAN 통신 상태 Check 함수

```

void InterruptCan(void)
{
    unsigned char i,dummy;
    unsigned char IntTemp;
    unsigned char count;
    disable();
    int_mask &= 0x7f;
    enable();
    IntTemp = ReadCan(INTERRUPT_REG);
    if(IntTemp & RECEIVE_INT){
        if(ReBufEnd+1==ReBufBegin || (ReBufEnd==R_BUF_SIZE-1 && !ReBufBegin)) {
            if(CanDataError != 0xff ) CanDataError++;    /* overrun error code count */
        }
        else {
            if(++ReBufEnd > R_BUF_SIZE - 1) ReBufEnd=0;
            ReBuf[ReBufEnd][0] = ReadCan(RECEIVE_ID_0);
            ReBuf[ReBufEnd][1] = ReadCan(RECEIVE_DATA_LENGTH);
            count = ReBuf[ReBufEnd][1] & 0x0f;
            for(i = 0 ; i < count ; i++){
                ReBuf[ReBufEnd][i+2]=ReadCan(RECEIVE_DATA_BYTE_1+i);
            }
        }
        WriteCan(COMMAND_REG, RECEIVE_CMD);
    }
    if(IntTemp & ANY_ERROR_INT) if(CanDataError != 0xff) CanDataError++;
    disable();
    int_mask |= 0x80;
    enable();
}

unsigned int CanCheck(void)
{
    unsigned char tmp;
    tmp = CanDataError;
    CanDataError = 0;
    return(tmp);
}

```