

```

int nvm_flash()
{
    if (CUST_EnterWriteMode(0, SECTOR_0 | SECTOR_1 | SECTOR_2 | SECTOR_3 | SECTOR_4 ) != 0 ) return -1;

    if (CUST_WriteSector(0,0,Sector0) != 0 ) return -1;
    if (CUST_WriteSector(0,1,Sector1) != 0 ) return -1;
    if (CUST_WriteSector(0,2,Sector2) != 0 ) return -1;
    if (CUST_WriteSector(0,3,Sector3) != 0 ) return -1;
    if (CUST_WriteSector(0,4,Sector4) != 0 ) return -1;

    if (CUST_ExitTestMode(0) != 0 ) return -1;
}

int nvm_read(unsigned char * pSectorsOut, int SectorsLength)
{
    unsigned char Sectors[5][8];
    if (CUST_EnterReadMode(0) != 0 ) return -1;
    if (CUST_ReadSector(0, 0, &Sectors[0][0]) != 0 ) return -1;
    if (CUST_ReadSector(0, 1, &Sectors[1][0]) != 0 ) return -1;
    if (CUST_ReadSector(0, 2, &Sectors[2][0]) != 0 ) return -1;
    if (CUST_ReadSector(0, 3, &Sectors[3][0]) != 0 ) return -1;
    if (CUST_ReadSector(0, 4, &Sectors[4][0]) != 0 ) return -1;
    if (CUST_ExitTestMode(0) != 0 ) return -1;

    unsigned char * pSectors = &Sectors[0][0];
    for(int i=0; i< NVM_SIZE; i++)
    { pSectorsOut[i] = pSectors[i]; }
}

/*
FTP Registers
o FTP_CUST_PWR (0x9E b(7), ftp_cust_pwr_i in RTL); power for FTP
o FTP_CUST_RST_N (0x9E b(6), ftp_cust_reset_n_i in RTL); reset for FTP
o FTP_CUST_REQ (0x9E b(4), ftp_cust_req_i in RTL); request bit for FTP operation
o FTP_CUST_SECT (0x9F (2:0), ftp_cust_sect1_i in RTL); for customer to select between sector 0 to 4 for read/write
operations (functions as lowest address bit to FTP, remainders are zeroed out)
o FTP_CUST_SER_MASK[4:0] (0x9F b(7:4), ftp_cust_ser_i in RTL); customer Sector Erase Register; controls erase of
sector 0 (00001), sector 1 (00010), sector 2 (00100), sector 3 (01000), sector 4 (10000) ) or all (11111).
o FTP_CUST_OPCODE_MASK[2:0] (0x9F b(2:0), ftp_cust_op3_i in RTL). Selects opcode sent to
FTP. Customer Opcodes are:
o 000 = Read sector
o 001 = Write Program Load register (PL) with data to be written to sector 0 or 1
o 010 = Write Sector Erase Register (SER) with data reflected by state of FTP_CUST_SER_MASK[4:0]
o 011 = Read Program Load register (PL)
o 100 = Read SER;
o 101 = Erase sector 0 to 4 (depending upon the mask value which has been programmed to SER)
o 110 = Program sector 0 to 4 (depending on FTP_CUST_SECT1)
o 111 = Soft program sector 0 to 4 (depending upon the value which has been programmed to SER)*/

int CUST_EnterWriteMode(uint8_t Port,unsigned char ErasedSector)
{
    unsigned char Buffer[2];

    Buffer[0]=FTP_CUST_PASSWORD; /* Set Password*/
    if ( I2C_Write_USB_PD(Port,FTP_CUST_PASSWORD_REG,Buffer,1) != HAL_OK )return -1;

    Buffer[0]= 0 ; /* this register must be NULL for Partial Erase feature */
    if ( I2C_Write_USB_PD(Port,RW_BUFFER,Buffer,1) != HAL_OK )return -1;

    {
        //NVM Power-up Sequence
        //After STUSB start-up sequence, the NVM is powered off.

        Buffer[0]= 0; /* NVM internal controller reset */
        if ( I2C_Write_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK ) return -1;

        Buffer[0]= FTP_CUST_PWR | FTP_CUST_RST_N; /* Set PWR and RST_N bits */
        if ( I2C_Write_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK ) return -1;
    }

    Buffer[0]=((ErasedSector << 3) & FTP_CUST_SER_MASK) | ( WRITE_SER & FTP_CUST_OPCODE_MASK) ; /* Load 0xF1 to
erase all sectors of FTP and Write SER Opcode */
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_1,Buffer,1) != HAL_OK )return -1; /* Set Write SER Opcode */

    Buffer[0]=FTP_CUST_PWR | FTP_CUST_RST_N | FTP_CUST_REQ ;
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1; /* Load Write SER Opcode */

    do
    {
        if ( I2C_Read_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1; /* Wait for execution */
    }
}

```

```

    }
    while(Buffer[0] & FTP_CUST_REQ);

    Buffer[0]= SOFT_PROG_SECTOR & FTP_CUST_OPCODE_MASK ;
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_1,Buffer,1) != HAL_OK )return -1; /* Set Soft Prog Opcode */

    Buffer[0]=FTP_CUST_PWR | FTP_CUST_RST_N | FTP_CUST_REQ ;
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1; /* Load Soft Prog Opcode */

    do
    {
        if ( I2C_Read_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1; /* Wait for execution */
    }
    while(Buffer[0] & FTP_CUST_REQ);

    Buffer[0]= ERASE_SECTOR & FTP_CUST_OPCODE_MASK ;
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_1,Buffer,1) != HAL_OK )return -1; /* Set Erase Sectors Opcode */

    Buffer[0]=FTP_CUST_PWR | FTP_CUST_RST_N | FTP_CUST_REQ ;
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1; /* Load Erase Sectors Opcode */

    do
    {
        if ( I2C_Read_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1; /* Wait for execution */
    }
    while(Buffer[0] & FTP_CUST_REQ);

    return 0;
}

int CUST_EnterReadMode(uint8_t Port)
{
    unsigned char Buffer[2];

    Buffer[0]=FTP_CUST_PASSWORD; /* Set Password*/
    if ( I2C_Write_USB_PD(Port,FTP_CUST_PASSWORD_REG,Buffer,1) != HAL_OK )return -1;

    {
        //NVM Power-up Sequence
        //After STUSB start-up sequence, the NVM is powered off.

        Buffer[0]= 0; /* NVM internal controller reset */
        if ( I2C_Write_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK ) return -1;

        Buffer[0]= FTP_CUST_PWR | FTP_CUST_RST_N; /* Set PWR and RST_N bits */
        if ( I2C_Write_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK ) return -1;
    }

    return 0 ;
}

int CUST_ReadSector(uint8_t Port,char SectorNum, unsigned char *SectorData)
{
    unsigned char Buffer[2];

    Buffer[0]= FTP_CUST_PWR | FTP_CUST_RST_N ; /* Set PWR and RST_N bits */
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1;

    Buffer[0]= (READ & FTP_CUST_OPCODE_MASK);
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_1,Buffer,1) != HAL_OK )return -1; /* Set Read Sectors Opcode */

    Buffer[0]= (SectorNum & FTP_CUST_SECT) |FTP_CUST_PWR |FTP_CUST_RST_N | FTP_CUST_REQ;
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1; /* Load Read Sectors Opcode */

    do
    {
        if ( I2C_Read_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1; /* Wait for execution */
    }
    while(Buffer[0] & FTP_CUST_REQ); //The FTP_CUST_REQ is cleared by NVM controller when the operation is
    finished.

    if ( I2C_Read_USB_PD(Port,RW_BUFFER,&SectorData[0],8) != HAL_OK )return -1; /* Sectors Data are available in
    RW-BUFFER @ 0x53 */

    Buffer[0]= 0 ; /* NVM internal controller reset */
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1;

    return 0;
}

```

```

int CUST_WriteSector(uint8_t Port,char SectorNum, unsigned char *SectorData)
{
    unsigned char Buffer[2];

    //Write the 64-bit data to be written in the sector
    if ( I2C_Write_USB_PD(Port,RW_BUFFER,SectorData,8) != HAL_OK )return -1;

    Buffer[0]=FTP_CUST_PWR | FTP_CUST_RST_N; /*Set PWR and RST_N bits*/
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1;

    //NVM Program Load Register to write with the 64-bit data to be written in sector
    Buffer[0]= (WRITE_PL & FTP_CUST_OPCODE_MASK); /*Set Write to PL Opcode*/
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_1,Buffer,1) != HAL_OK )return -1;

    Buffer[0]=FTP_CUST_PWR |FTP_CUST_RST_N | FTP_CUST_REQ; /* Load Write to PL Sectors Opcode */
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1;

    do
    {
        if ( I2C_Read_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1; /* Wait for execution */
    }
    while(Buffer[0] & FTP_CUST_REQ) ; //FTP_CUST_REQ clear by NVM controller

    //NVM "Word Program" operation to write the Program Load Register in the sector to be written
    Buffer[0]= (PROG_SECTOR & FTP_CUST_OPCODE_MASK);
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_1,Buffer,1) != HAL_OK )return -1; /*Set Prog Sectors Opcode*/

    Buffer[0]=(SectorNum & FTP_CUST_SECT) |FTP_CUST_PWR |FTP_CUST_RST_N | FTP_CUST_REQ;
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1; /* Load Prog Sectors Opcode */

    do
    {
        if ( I2C_Read_USB_PD(Port,FTP_CTRL_0,Buffer,1) != HAL_OK )return -1; /* Wait for execution */
    }
    while(Buffer[0] & FTP_CUST_REQ); //FTP_CUST_REQ clear by NVM controller

    return 0;
}

int CUST_ExitTestMode(uint8_t Port)
{
    unsigned char Buffer[2];

    Buffer[0]= FTP_CUST_RST_N;
    Buffer[1]= 0x00; /* clear registers */
    if ( I2C_Write_USB_PD(Port,FTP_CTRL_0,Buffer,2) != HAL_OK )return -1;

    Buffer[0]= 0x00;
    if ( I2C_Write_USB_PD(Port,FTP_CUST_PASSWORD_REG,Buffer,1) != HAL_OK )return -1; /* Clear Password */

    return 0 ;
}

```