



shaping tomorrow with you

# Fujitsu New-8FX MCU

## UART Bootloader

This sample bootloader (*[UART\\_bootloader.wsp](#)*) is designed for the purpose of implementing in-application Programming (IAP). In-application programming refers to the performing of erase and write operation on the on-chip flash memory, as directed by the end-user application code.

This bootloader sample code is developed using Fujitsu Softune IDE and debugged using MB95F264K starterkit MB2146-420-E.

In addition, a sample user program (*[User\\_Prog.wsp](#)*) is also provided. Using the bootloader, user can flash this user program into the MCU.

The PC terminal program used for communicating with bootloader is REALTERM. It can be downloaded free of charge at <http://realterm.sourceforge.net/>

# Bootloader Features

- **Uses UART for RS-232 connection.**
- **Resides in address 0xB000 – 0xBFFF.**
- **Utilizing the dual operation flash feature, flash writing/erasing routines are executed from flash to erase/flash new code in 0xC000 – 0xFFFF.**
- **CR trimming data in NVR (none volatile register) area are saved before flash erase. Likewise, reset vector data (0xB000) at address 0xFFFFE is saved and restored after erase. This is to ensure the reset location is always 0xB000.**
- **Bootloader check condition is the checking of S1 button press within 3 seconds of power on. If S1 pressed, go to boot mode, if not pressed, go to user code at 0xC000.**
- **This sample code demonstrates LED lighting sequence of LED4-3-2-4-3-2...**

# Bootloader Features

- Routine used to jump to user code location of 0xC000.

```
#pragma asm
MOVW    A, #0xC000
JMP @A          // Jump to firmware start address
#pragma endasm
```

Main.c

- **#pragma section** command that is used to assign address 0xC000.

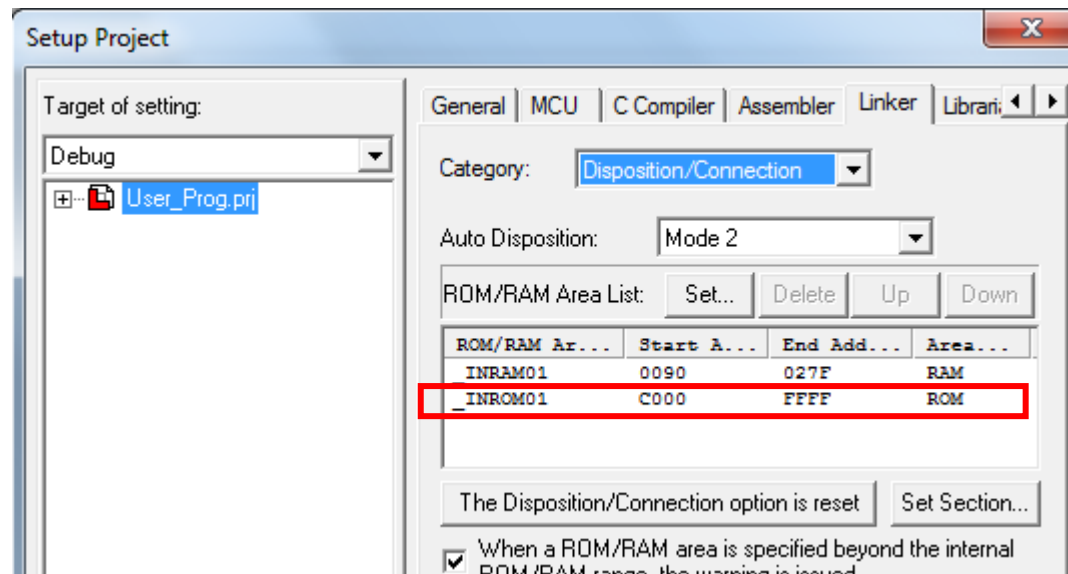
```
#pragma section CODE=user_main, attr=CODE, locate=0xC000

/*----- SUB ROUTINES -----*/
void user_main(){
    while(1)
    {
        LED_seq432();
    }
}
```

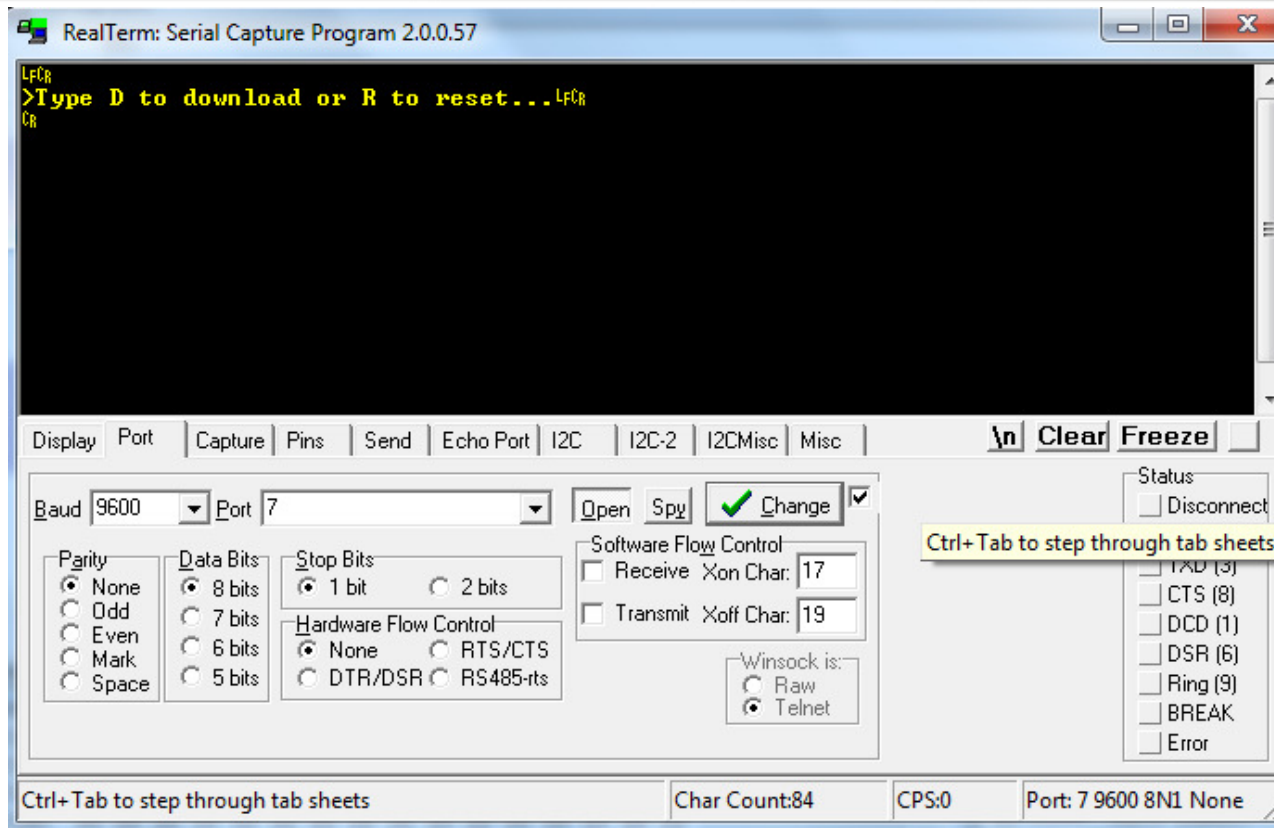
User\_main.c

# Sample Code Features

- Sample user application without bootloader (*User\_Prog.wsp*) with code residing from 0xC000 – 0xFFFF is provided.
- It demonstrates the LED lighting sequence of LED2-3-4-2-3-4...
- Code does not include 0xB000 – 0xBFFF area as this is reserved for bootloader code.

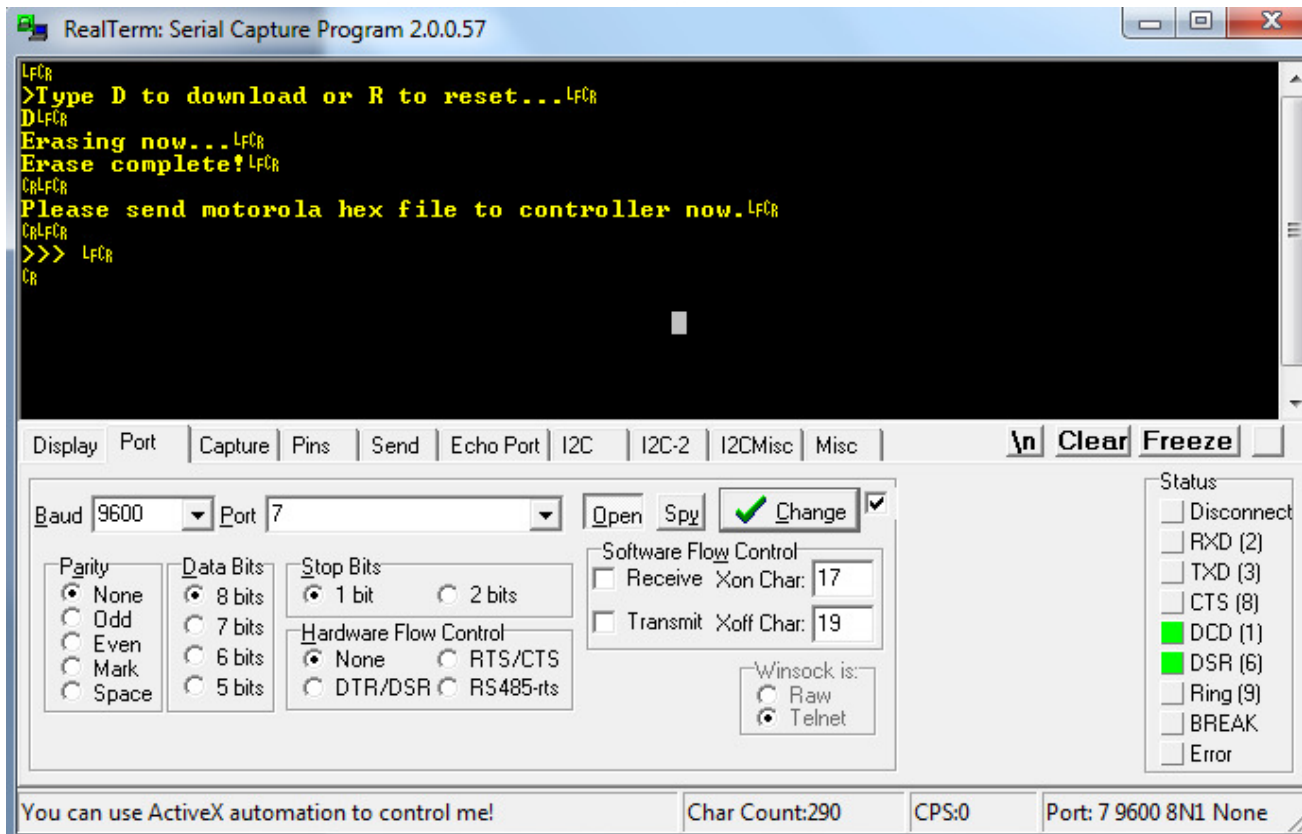


# Program Use (1)



- **Make sure setting of baud rate 9600bps, 8 data bit, 1 stop bit and no parity. After all settings, click on ‘Change’ button to effect changes.**
- **After power on, press S1 button within 3sec to enter bootloader mode. Message is output to prompt user to type in ‘D’ to proceed.**
- **If S1 is not pressed, observe LED sequence of LED4-3-2-4-3-2.**

# Program Use (2)

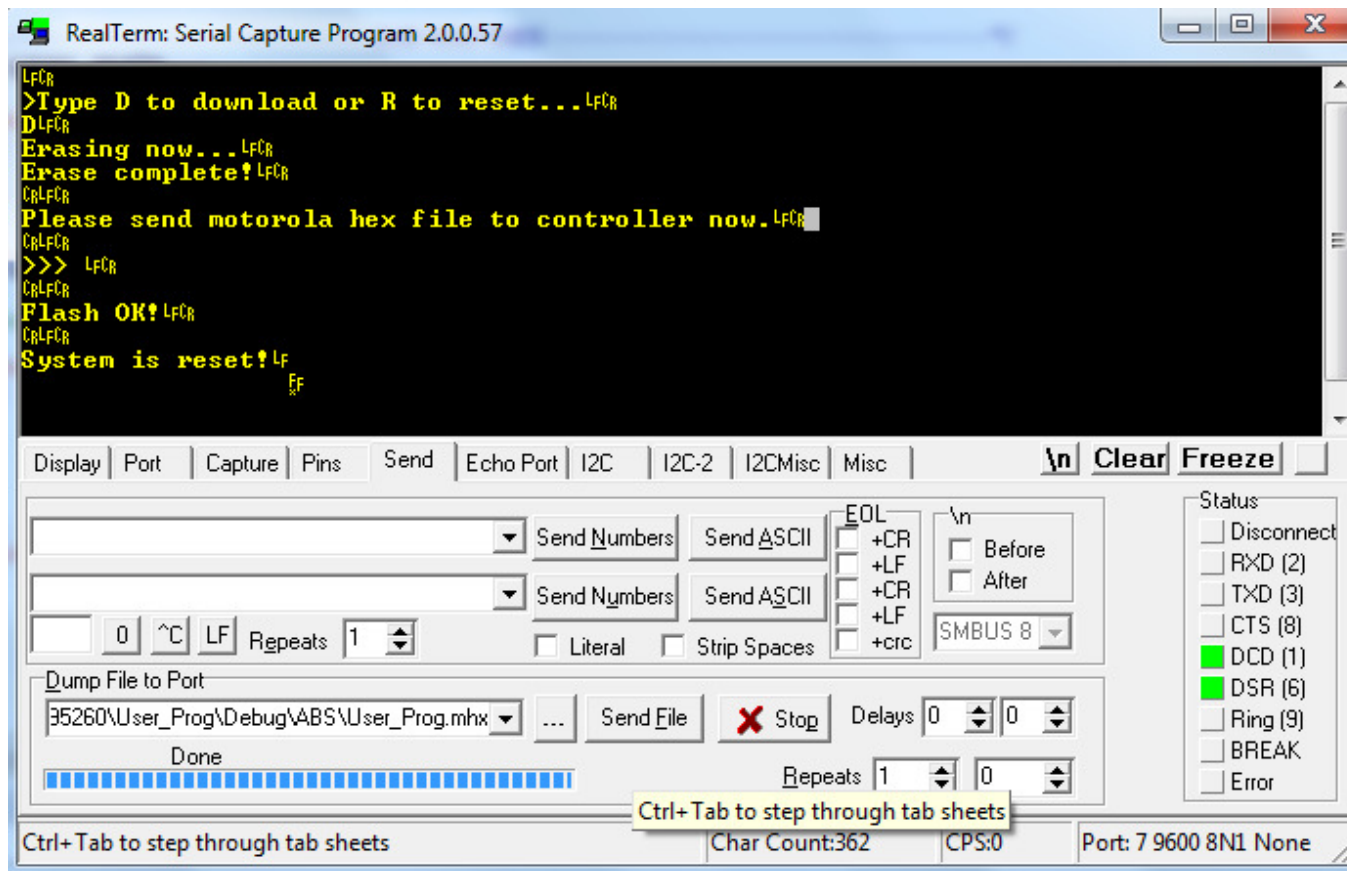


- After 'D' and ENTER are pressed, program proceeds to erase flash area 0xC000 – 0xFFFF. When erase finishes, it prompts user to send motorola hex file.

- User can go to 'Send' tag and select motorola hex file to send to MCU for flash writing. After selecting hex file (.mhx file extension by default in Softune), click 'Send File' to proceed.



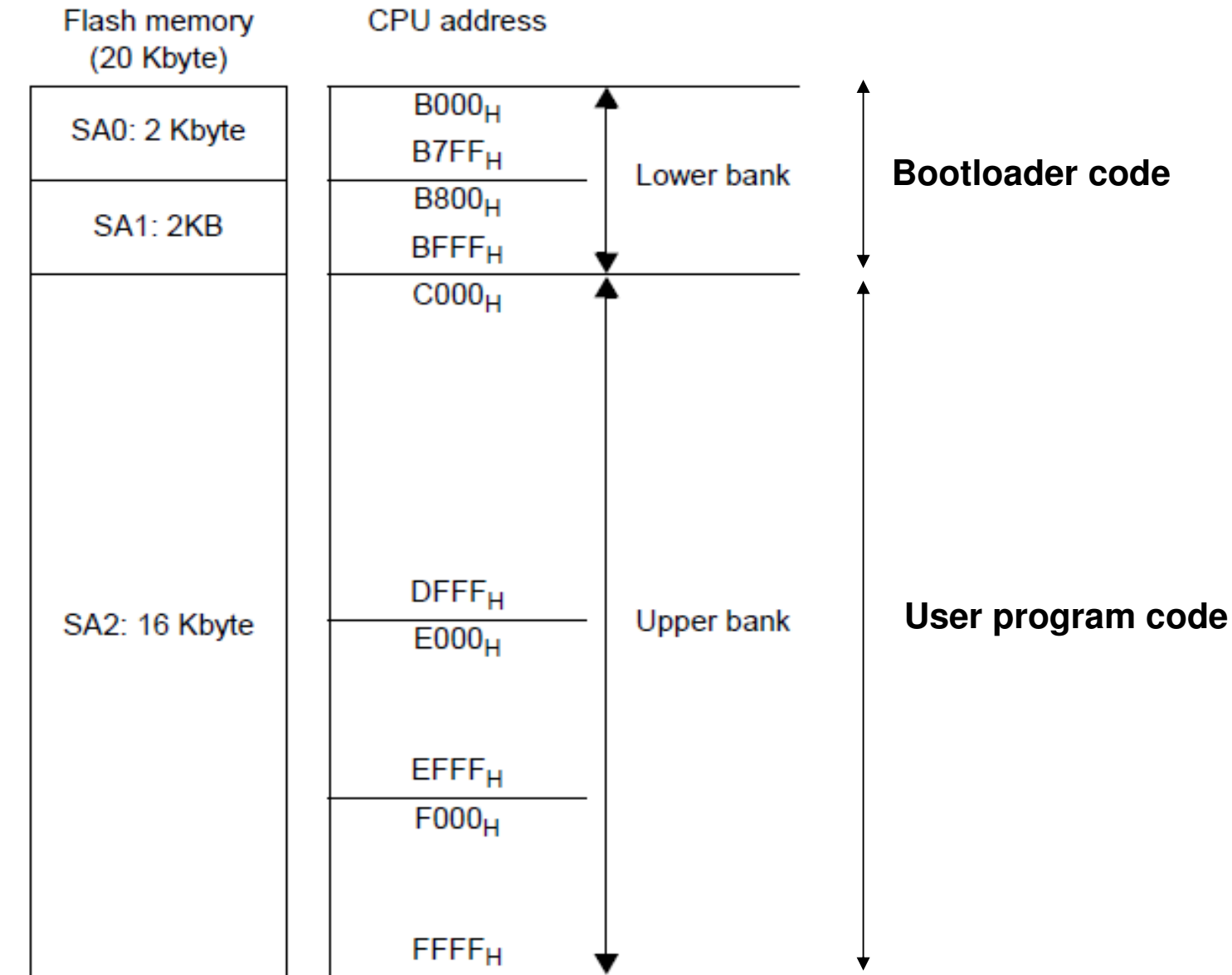
# Program Use (3)




- After motorola hex file has been sent, data is written into MCU on-chip flash. If writing is successful, 'Flash OK' is displayed and MCU does software reset.
- Observe that LED light up sequence is changed to LED2-3-4-2-3-4.



# Flash Memory Configuration (MB95F264K)





FUJITSU

shaping tomorrow with you