

Register	Value
Core	
R0	0x20002B20
R1	0x20002F20
R2	0x20002F20
R3	0x20002F20
R4	0x00000000
R5	0x20002AC0
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x0800C620
R11	0x00000000
R12	0x01010101
R13 (SP)	0x20003720
R14 (LR)	0x0800022B
R15 (PC)	0x0800A588
xPSR	0x61000000
N	0
Z	1
C	1
V	0
Q	0
T	1
IT	Disabled
ISR	0
Banked	
System	
BASEPRI	0x00
PRIMASK	0
FAULTMASK	0
CONTROL	0x00
Internal	
Mode	Thread
Privilege	Privileged

```

* Description :-
  At this stage the microcontroller clock setting is already configured,
  this is done through SystemInit() function which is called from startup
  file (startup_stm3211xx_xx.s) before to branch to application main.
  To reconfigure the default setting of SystemInit() function, refer to
  system_stm3211xx.c file
  *****/
int main(void)
{
  // Maintain below order due to interdependency each other module
  pwr_vInit();

  FLS_vInit();
  TMR_vStart();
  SPI_vInit();
  LCD_vInit();
  I2C_vInit();

  RTC_vInit();
  FFS_vInit();
  KEY_vInit();
  USB_vInit();

  ADC_vInit();
  SCR_vInit(); // should be called after key and adc init only

  pwr_vStart();

  while(1)
  {
    TMR_vHandleStages();
  }
  /* end of main Function */

  /*****/
  * Function :- static void pwr_vInit(void)
  *****/

```

Name	Location/Value	Type
Call Stack + Locals	Watch 1 Watch 2	

Registers

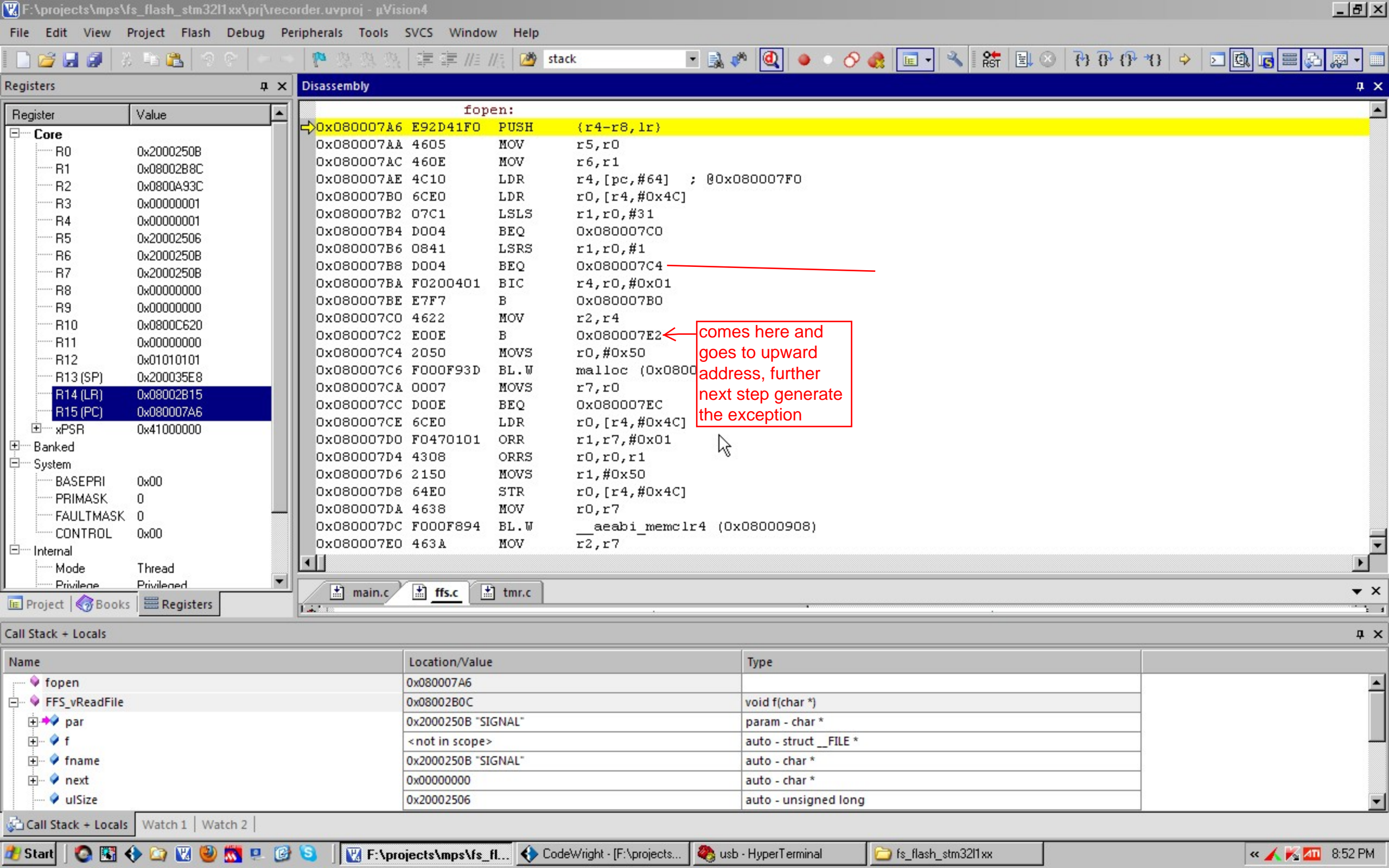
Register	Value
Core	
R0	0x2000250B
R1	0x20003630
R2	0x0800A93C
R3	0x00000001
R4	0x00000001
R5	0x20002506
R6	0x2000250B
R7	0x2000250B
R8	0x00000000
R9	0x00000000
R10	0x0800C620
R11	0x00000000
R12	0x01010101
R13 (SP)	0x200035E8
R14 (LR)	0x0800A43B
R15 (PC)	0x08002B0C
xPSR	0x41000000
Banked	
System	
BASEPRI	0x00
PRIMASK	0
FAULTMASK	0
CONTROL	0x00
Internal	
Mode	Thread
Privilege	Privileged

Disassembly

```

0x08002AFE F7FDFF2A BL.W   strncpy (0x08000956)
317:   USB_vTxDtString((const char *)aubText);
350:   fclose(f); /* close the input file when done */
0x08002B02 A801   ADD    r0,sp,#0x04
0x08002B04 F002FC18 BL.W   USB_vTxDtString (0x08005338)
351: } // end of FFS_vReadFile
0x08002B08 B013   ADD    sp,sp,#0x4C
0x08002B0A BDF0   POP    (r4-r7,pc)
321:   f = fopen(fname, "r"); /* open the file for reading */
322:
0x08002B0C A11F   ADR    r1,(pc)+4 ; @0x08002B8C
0x08002B0E 4630   MOV    r0,r6
0x08002B10 F7FDFF49 BL.W   fopen (0x080007A6)
0x08002B14 4604   MOV    r4,r0
323:   if(f == NULL)
324:   {
0x08002B16 B95C   CBNZ  r4,0x08002B30
325:   KEY_vMSGDisplay2(KEY_FILE_NOT_FOUND);
0x08002B18 2011   MOVS  r0,#0x11
0x08002B1A F000FC87 BL.W   KEY_vMSGDisplay2(0x0800342C)
326:   strncpy(aubText,"File not found!\n\r", sizeof(aubText));
0x08002B1E 2244   MOVS  r2,#0x44
0x08002B20 A11B   ADR    r1,(pc)+4 ; @0x08002B90
0x08002B22 A801   ADD    r0,sp,#0x04
0x08002B24 F7FDFF17 BL.W   strncpy (0x08000956)
327:   USB_vTxDtString((const char *)aubText);
    
```

Name	Location/Value	Type
FFS_vReadFile	0x08002B0C	void f(char *)
par	0x2000250B "SIGNAL"	param - char *
f	<not in scope>	auto - struct __FILE *
fname	0x2000250B "SIGNAL"	auto - char *
next	0x00000000	auto - char *
ulSize	0x20002506	auto - unsigned long
aubText	0x200035EC "y...\n\r"	auto - char[68]



Registers

Register	Value
Core	
R0	0x00002A21
R1	0x00001510
R2	0x0800A93C
R3	0x00000001
R4	0x00002A20
R5	0x2000250B
R6	0x08002B8C
R7	0x2000250B
R8	0x00000000
R9	0x00000000
R10	0x0800C620
R11	0x00000000
R12	0x01010101
R13 (SP)	0x20003580
R14 (LR)	0xFFFFFFFF9
R15 (PC)	0x08002E70
xPSR	0x21000003
Banked	
System	
BASEPRI	0x00
PRIMASK	0
FAULTMASK	0
CONTROL	0x00
Internal	
Mode	Handler
Privilege	Privileged

Disassembly

```

72:  hard_fault_handler_c();
73:  while (1)
74:  {
75:  }
76:  }
77:
78:  /**
79:   * @brief This function handles Memory Manage exception.
80:   * @param None
81:   * @retval None
82:   */
83:  void MemManage_Handler(void)
84:  {
85:   /* Go to infinite loop when Memory Manage exception occurs */
0x08002E74 BF00      NOP
0x08002E76 E7FE      B          0x08002E76
78:  {
0x08002E78 B510      PUSH     {r4,lr}
0x08002E7A 4604      MOV      r4,r0
79:   I2C1->DR = adr;
0x08002E7C 4804      LDR      r0,[pc,#16] ; @0x08002E90
0x08002E7E 8004      STRH     r4,[r0,#0x00]
80:   while (!(I2C_sr() & 0x0002));
0x08002E80 BF00      NOP
0x08002E82 F000F8E3 BL.W     I2C_sr (0x0800304C)
    
```

Fault Reports

Memory Manage Faults

MM_FAULT_ADDR: 0x00002A6C

MM_FAULT_STAT: 0x00

IACCVIOL MUNSTKERR

DACCVIOL MSTKERR

MMARVALID

Bus Faults

BUS_FAULT_ADDR: 0x00002A6C

BUS_FAULT_STAT: 0x82

IBUSERR UNSTKERR

PRECISERR STKERR

IMPRECISERR BFARVALID

Usage Faults

USG_FAULT_STAT: 0x0000

UNDEFINSTR NOCP

INVSTATE UNALIGNED

INVPC DIVBYZERO

Hard Faults

HARD_FAULT_STAT: 0x40000000

VECTTBL DEBUGEVT

FORCED

Debug Faults

DBG_FAULT_STAT: 0x00000003

HALTED VCATCH

BKPT EXTERNAL

DWTTRAP

Call Stack + Locals

Name	Location/Value	Type
HardFault_Handler	0x08002E70	void f()
fopen	0x080007A6	
FFS_vReadFile	0x08002B0C	void f(char *)
par	0x2000250B "SIGNAL"	param - char *
f	<not in scope>	auto - struct __FILE *
fname	0x2000250B "SIGNAL"	auto - char *
next	0x00000000	auto - char *