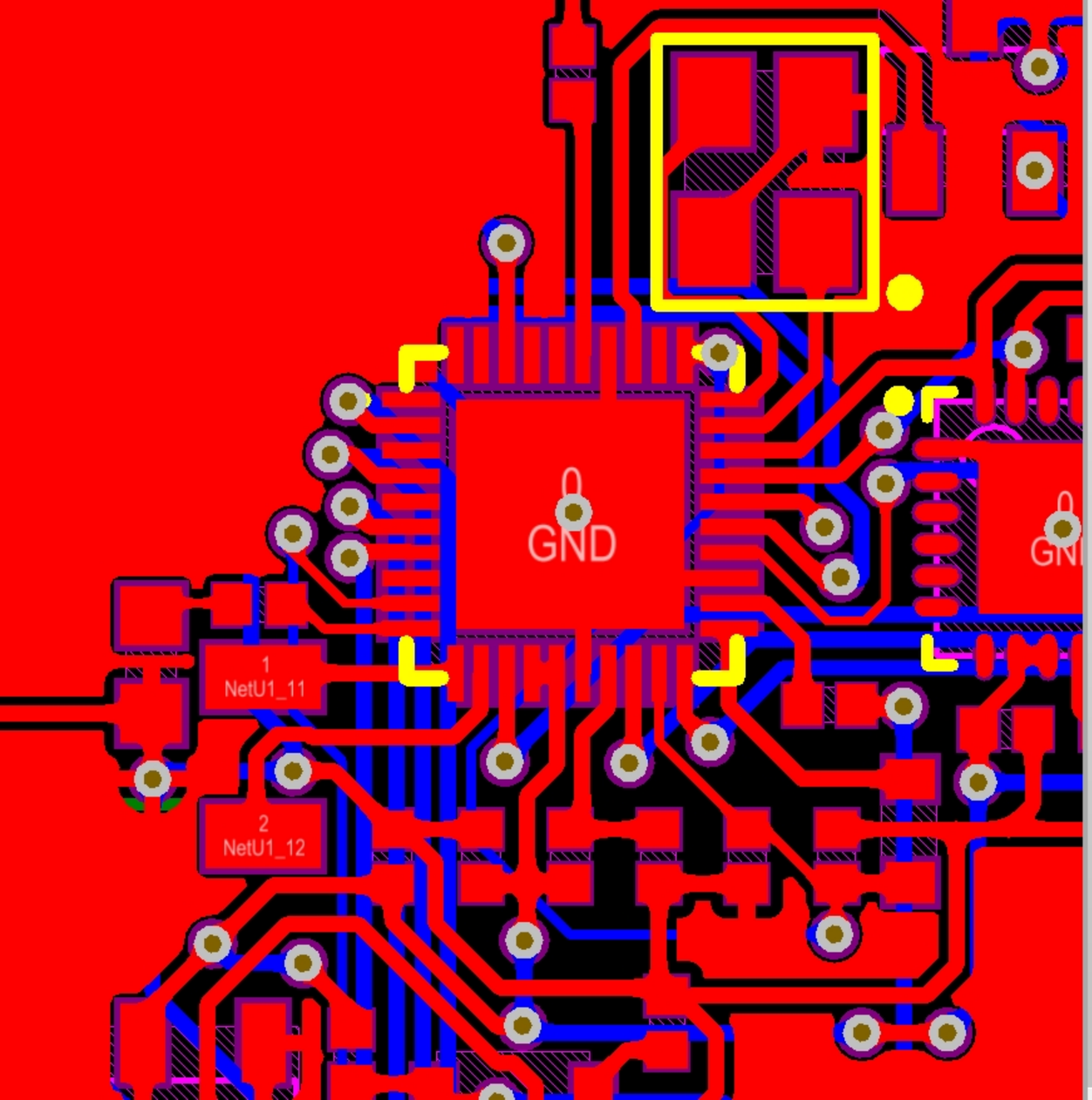
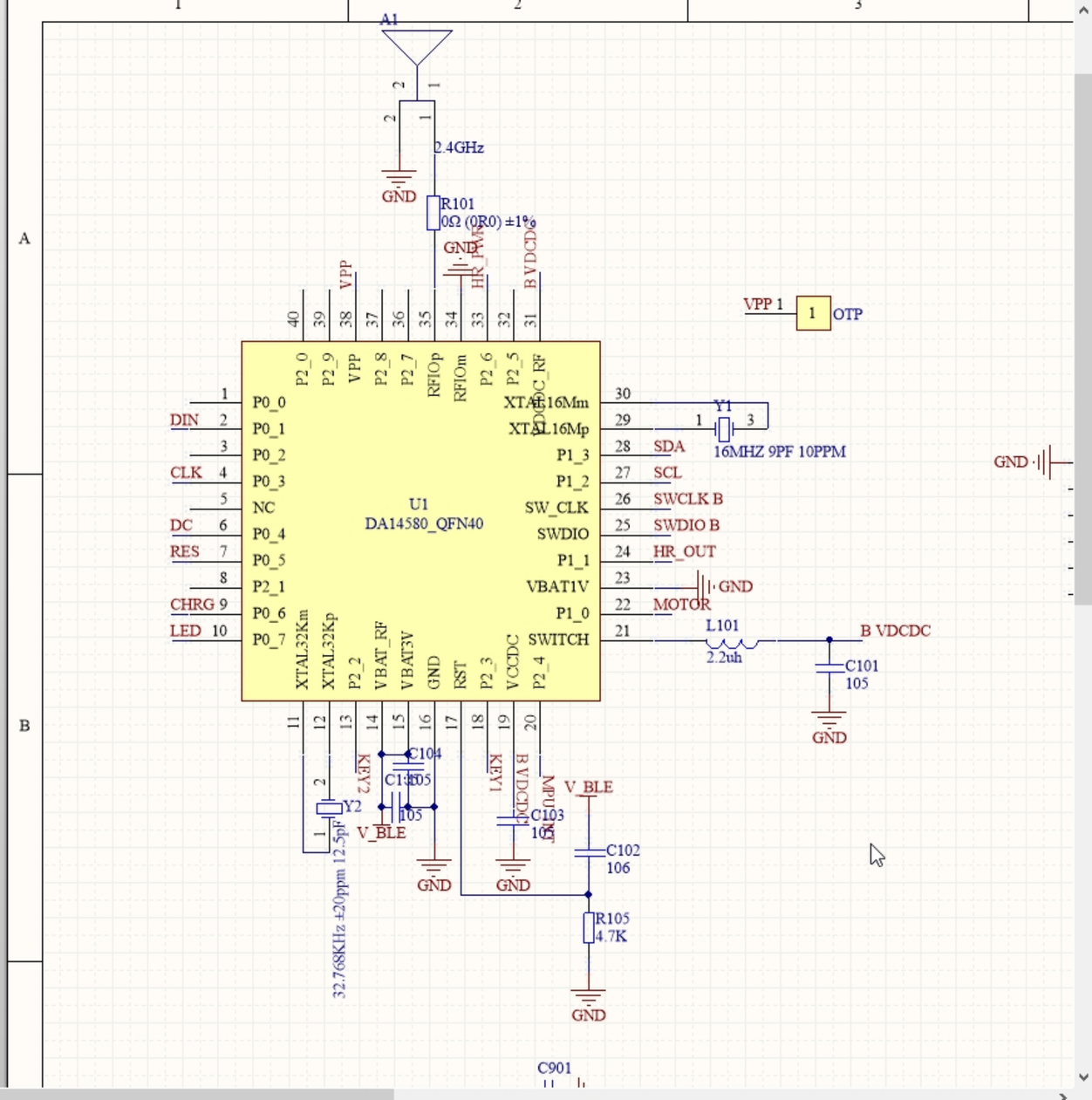
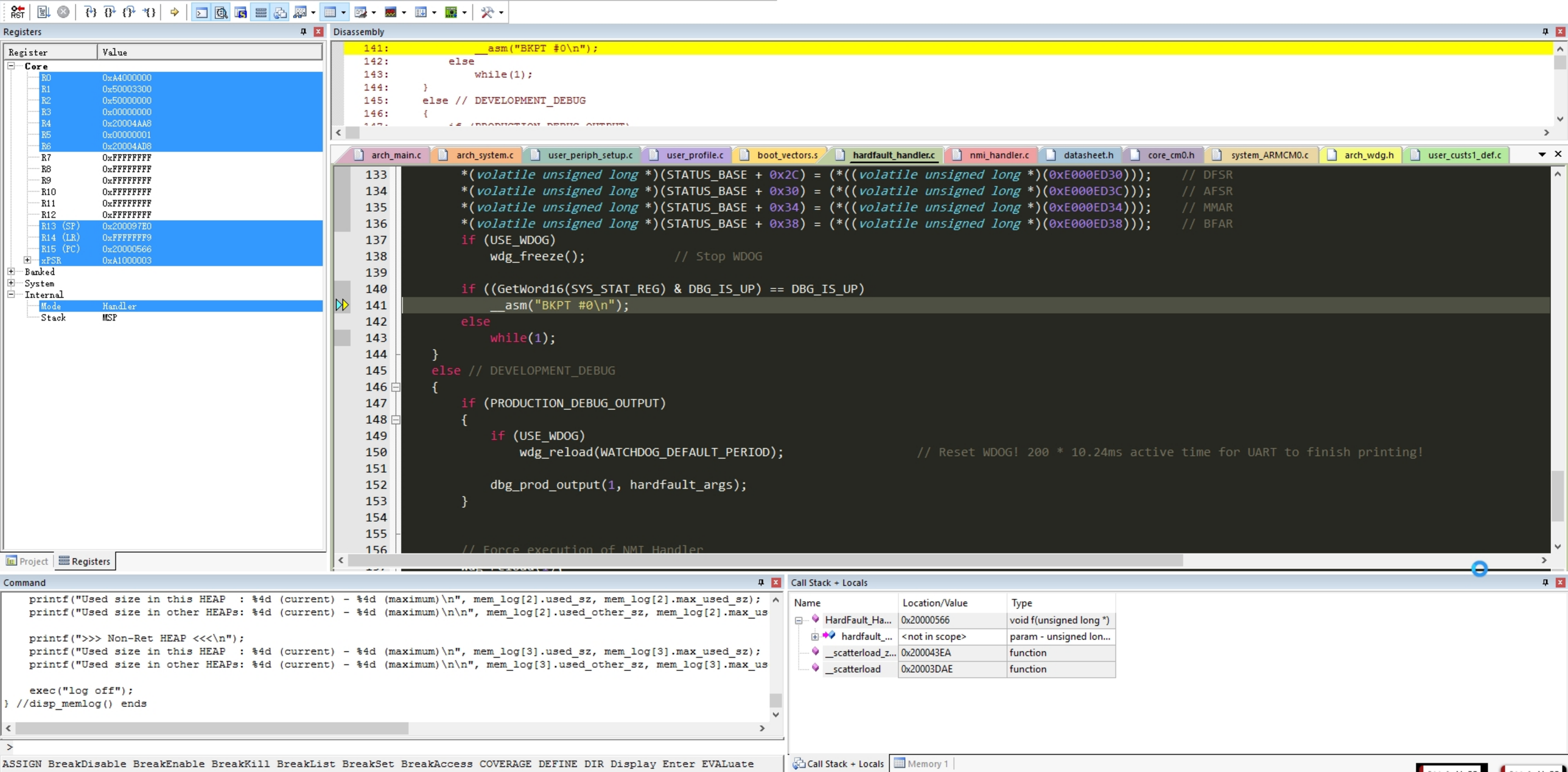

GND
1
NetU1_11
2
NetU1_12


A1
2.4GHz
GND
R101
0Ω (0R0) ±1%
GND
HR_PWM
B VDCDC
VPP
VPP 1
1
OTP
U1
DA14580_QFN40
P0_0
P0_1
P0_2
P0_3
NC
P0_4
P0_5
P2_1
P0_6
P0_7
DIN
CLK
DC
RES
CHRG
LED
P2_0
P2_9
VPP
P2_8
P2_7
RFIOp
RFIOm
P2_6
P2_5
VDCDC_RF
XTAL16Mm
XTAL16Mp
P1_3
P1_2
SW_CLK
SWDIO
P1_1
VBAT1V
P1_0
SWITCH
SDA
SCL
SWCLK B
SWDIO B
HR_OUT
MOTOR
Y1
16MHZ 9PF 10PPM
L101
2.2uh
C101
105
XTAL32Km
XTAL32Kp
P2_2
VBAT_RF
VBAT3V
GND
RST
P2_3
VCCDC
P2_4
C104
C105
Y2
32.768KHz ±20ppm 12.5pF
V_BLE
C103
C102
106
R105
4.7K
C901


Registers
Register
Value
Core
R0 0xA4000000
R1 0x50003300
R2 0x50000000
R3 0x00000000
R4 0x20004AA8
R5 0x00000001
R6 0x20004AD8
R7 0xFFFFFFFF
R8 0xFFFFFFFF
R9 0xFFFFFFFF
R10 0xFFFFFFFF
R11 0xFFFFFFFF
R12 0xFFFFFFFF
R13 (SP) 0x200097E0
R14 (LR) 0xFFFFFFF9
R15 (PC) 0x20000566
xPSR 0xA1000003
Banked
System
Internal
Mode Handler
Stack MSP
Project
Registers
Disassembly
141: __asm("BKPT #0\n");
142: else
143: while(1);
144: }
145: else // DEVELOPMENT_DEBUG
146: {
arch_main.c
arch_system.c
user_periph_setup.c
user_profile.c
boot_vectors.s
hardfault_handler.c
nmi_handler.c
datasheet.h
core_cm0.h
system_ARMCM0.c
arch_wdg.h
user_custs1_def.c
133 *(volatile unsigned long *)(STATUS_BASE + 0x2C) = (*((volatile unsigned long *)(0xE000ED30))); // DFSR
134 *(volatile unsigned long *)(STATUS_BASE + 0x30) = (*((volatile unsigned long *)(0xE000ED3C))); // AFSR
135 *(volatile unsigned long *)(STATUS_BASE + 0x34) = (*((volatile unsigned long *)(0xE000ED34))); // MMAR
136 *(volatile unsigned long *)(STATUS_BASE + 0x38) = (*((volatile unsigned long *)(0xE000ED38))); // BFAR
137 if (USE_WDOG)
138 wdg_freeze(); // Stop WDOG
139
140 if ((GetWord16(SYS_STAT_REG) & DBG_IS_UP) == DBG_IS_UP)
141 __asm("BKPT #0\n");
142 else
143 while(1);
144 }
145 else // DEVELOPMENT_DEBUG
146 {
147 if (PRODUCTION_DEBUG_OUTPUT)
148 {
149 if (USE_WDOG)
150 wdg_reload(WATCHDOG_DEFAULT_PERIOD); // Reset WDOG! 200 * 10.24ms active time for UART to finish printing!
151
152 dbg_prod_output(1, hardfault_args);
153 }
154
155
156 // Force execution of NMI Handler
Command
printf("Used size in this HEAP : %4d (current) - %4d (maximum)\n", mem_log[2].used_sz, mem_log[2].max_used_sz);
printf("Used size in other HEAPs: %4d (current) - %4d (maximum)\n\n", mem_log[2].used_other_sz, mem_log[2].max_us
printf(">>> Non-Ret HEAP <<<\n");
printf("Used size in this HEAP : %4d (current) - %4d (maximum)\n", mem_log[3].used_sz, mem_log[3].max_used_sz);
printf("Used size in other HEAPs: %4d (current) - %4d (maximum)\n\n", mem_log[3].used_other_sz, mem_log[3].max_us
exec("log off");
} //disp_memlog() ends
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter EVALuate
Call Stack + Locals
Name
Location/Value
Type
HardFault_Ha... 0x20000566 void f(unsigned long *)
hardfault_... <not in scope> param - unsigned lon...
__scatterload_z... 0x200043EA function
__scatterload 0x20003DAE function
Call Stack + Locals
Memory 1