|  |  |
| --- | --- |
| Main Program (main.c) | Interrupt Service Routine (isr.s) |
| unsigned int counter ;  extern void task();  extern void SysTick\_Init();  int main(void) {  counter = 0;  SysTick\_Init();  // Wait for counter  **while(counter != 10);**  // Continue the tasks  while(1);  } | AREA ISR, CODE, READONLY  IMPORT counter  ENTRY  SysTick\_Handler PROC  EXPORT SysTick\_Handler  LDR r1,=counter  LDR r0,[r1] ; load counter  ADD r0,r0,#1 ; counter++  STR r0,[r1] ; save counter  BX LR ; exit  ENDP  END |

|  |  |  |
| --- | --- | --- |
| Unoptimized code | Optimized code | Optimized code if counter is volatile |
| \_\_main PROC  LDR r1,=counter  MOV r0,#0  STR r0,[r1]  NOP  BL enable\_SysTick  wait LDR r1,=counter  LDR r0,[r1]  CMP r0,#10  BNE wait  NOP  stop B stop  ENDP | \_\_main PROC  LDR r1,=counter  MOV r0,#0  STR r0,[r1]  BL enable\_SysTick  stop B stop  ENDP | \_\_main PROC  LDR r1,=counter  MOV r0,#0  STR r0,[r1]  BL enable\_SysTick  wait LDR r0,[r1]  CMP r0,#10  BNE wait  stop B stop  ENDP |