
EXAMPLE 3

Description:

Calling a function with parameters from inline assembler. This example shows how to use IAP programming interface for NXP controllers.

Notes:

IAR code is not fully finished and verified yet. **Preliminary - do not use.**

GCC code example:

```
static void iapCall (void) __attribute ((naked));
static void iapCall (void)
{
  register void *pFP0 asm("r0") = iapCommands;
  register void *pFP1 asm("r1") = iapResults;

  asm volatile(" bx  %[iapLocation]"
               :
               : "r" (pFP0), "r" (pFP1), [iapLocation] "r" (IAP_LOCATION) );
}
```

GCC assembler output:

```
           .align     2
           .type      iapCall, %function
iapCall:
.LFB4:
           .loc 1 267 0
           @ Naked Function: prologue and epilogue provided by programmer.
           @ args = 0, pretend = 0, frame = 0
           @ frame_needed = 1, uses_anonymous_args = 0
           .loc 1 268 0
           ldr        r0, .L17
           .loc 1 269 0
           ldr        r1, .L17+4
           .loc 1 271 0
           mvn        r3, #-2147483634
            bx  r3
           .loc 1 274 0
.L18:
           .align     2
.L17:
           .word      iapCommands
           .word      iapResults
.LFE4:
           .size      iapCall, .-iapCall
```

Acknowledgment: Ivan Paulik @ PASAT, Bratislava

Equivalent IAR C code:

```
unsigned int iapCommands [5];
unsigned int iapResults [2];
#define IAP_LOCATION 0x7ffffff1

typedef void (__thumb *IAP)(void*, void*);
const static IAP iap_entry = (IAP)IAP_LOCATION; // MCU flash firmware interface
function.

void iapCall (void)
{
        iap_entry(iapCommands, iapResults);
}
```

IAR assembler output:

```
   \                                 In section .bss, align 4
     60          unsigned int iapCommands [5];
   \                     iapCommands:
   \   00000000                      DS8 20
     61          unsigned int iapResults [2];
   \                     iapResults:
   \   00000014                      DS8 8
     62          #define IAP_LOCATION 0x7ffffff1
     63
     64          typedef void (__thumb *IAP)(void*, void*);
     65          const static IAP iap_entry = (IAP)IAP_LOCATION; // MCU flash
firmware interface function.
     66

   \                                 In section .text, align 4, keep-with-next
     67          void iapCall (void)
     68          {
   \                     iapCall:
   \   00000000   80B5                PUSH     {R7,LR}
     69                  iap_entry(iapCommands, iapResults);
   \   00000002   0348                LDR      R0,??iapCall_0   ;; iapCommands
   \   00000004   0100                MOVS     R1,R0
   \   00000006   1431                ADDS     R1,R1,#+20
   \   00000008   024A                LDR      R2,??iapCall_0+0x4  ;; 0x7ffffff1
   \   0000000A   ........            BL       __iar_via_R2
     70          }
   \   0000000E   01BD                POP      {R0,PC}          ;; return
   \                     ??iapCall_0:
   \   00000010   ........            DC32     iapCommands
   \   00000014   F1FFFF7F            DC32     0x7ffffff1
```

Acknowledgment: Ivan Paulik @ PASAT, Bratislava