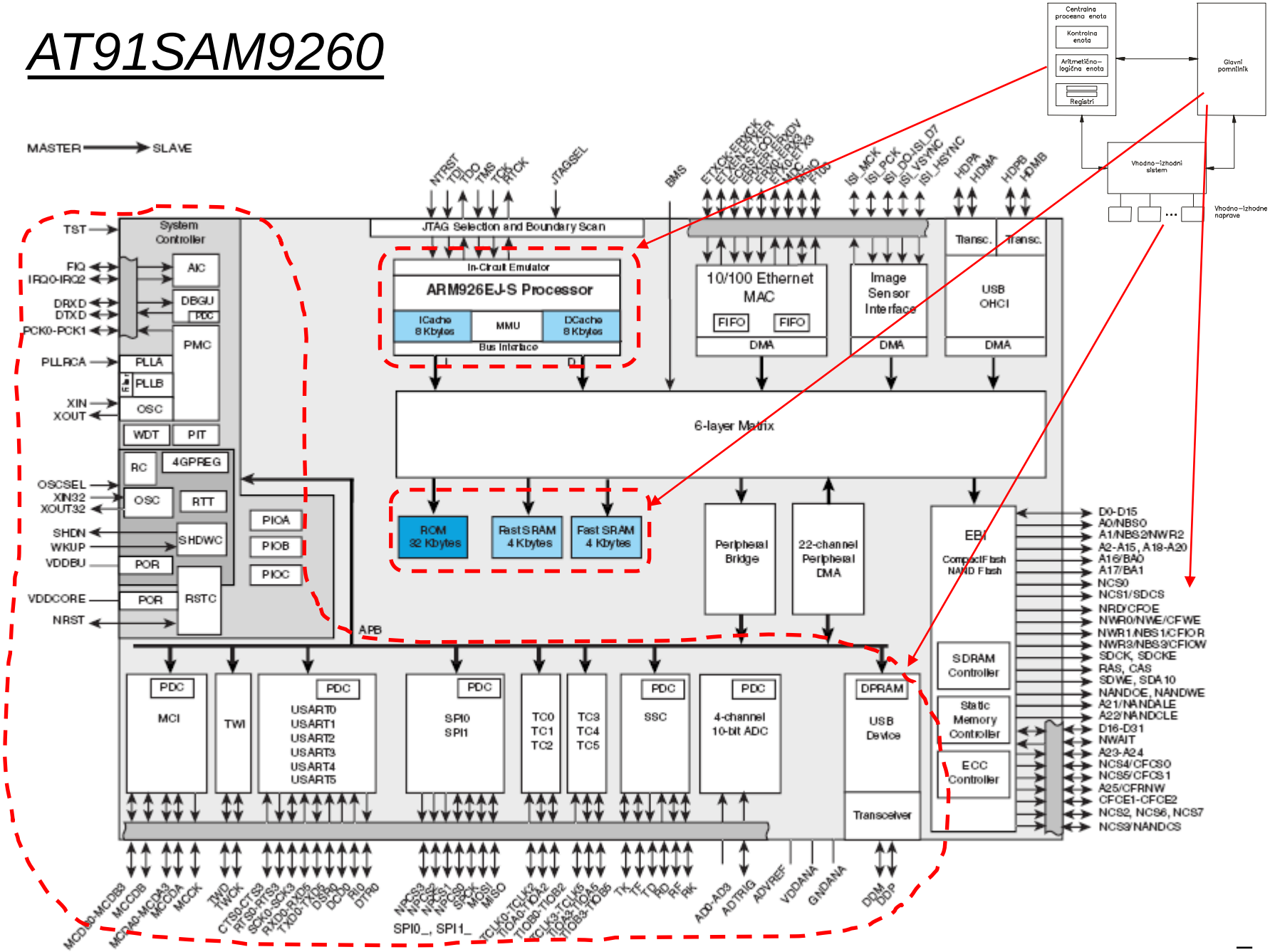
ARM

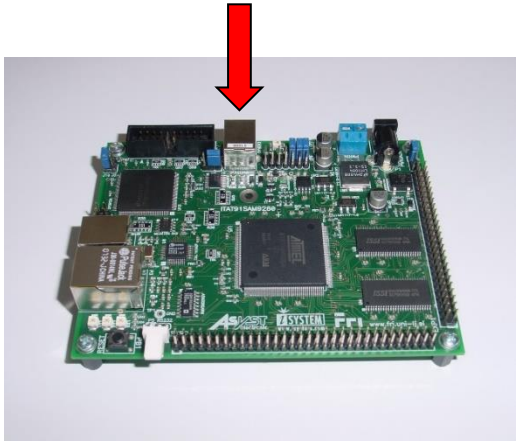
Project for FRI-SMS embedded system
(informative, additional content)

winIDEA

AT91SAM9260



Work on FRI-SMS development system

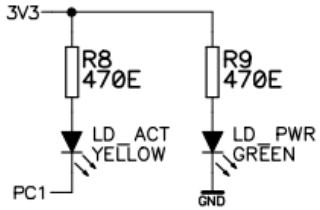


Connection :

- **USB** connection on longer side, **green LED is on**

Published workspace for FRI-SMS (e-classroom) :

- ***System's initialization*** (informative content) :
  - Set the frequency of the clock signal (higher means greater consumption!)
  - Enable caches
  - Initialization of stack (SP – Stack Pointer)



- ***Adding program code (start.s):***
  - *data/operands:*
    - added after `/*constants*/` mark, finish with .align
  - *program* :
    - added after `/* enter your code here */` mark
    - endless loop at the end
    - Subprograms after endless loop

System‘s initialization – initial start

intvec.s :

```
.text
.code 32

.global _start
 B _start            /* RESET INTERRUPT */
 B _error            /* UNDEFINED INSTRUCTION INTERRUPT */
 B _error            /* SOFTWARE INTERRUPT */
 B _error            /* ABORT (PREFETCH) INTERRUPT */
 B _error            /* ABORT (DATA) INTERRUPT */
 B _error            /* RESERVED */
 B _error           /* IRQ INTERRUPT */
 B _error          /* FIQ INTERRUPT */


.end
```

Operating mode, stack

start.s :

```
_start:

/* select system mode
   CPSR[4:0]      Mode
11111      System
*/

   mrs r0, cpsr
   bic r0, r0, #0x1F    /* clear mode flags */
   orr r0, r0, #0xDF    /* set supervisor mode (0b11111) + DISABLE IRQ, FIQ
                                                        I=F=1, T(humb)=0 */
   msr cpsr, r0

   /* init stack */
   ldr sp,_Lstack_end
```

System Clock

start.s :

```
  /* setup system clocks */
   ldr r1, =PMC_BASE

   ldr r0, = 0x0F01
   str r0, [r1,#CKGR_MOR]      /* Main Osc. enable, 15*8 cycles for osc. setup */

 osc_lp:
   ldr r0, [r1,#PMC_SR]        /* Main oscillator is stabilized ?          */
   tst r0, #0x01
   beq osc_lp

   mov r0, #0x01
   str r0, [r1,#PMC_MCKR]      /* CSS: Master Clock Selection - Main Clock*/

   ldr r0, =0x2000bf00 | ( 124 << 16) | 12   /* 18,432 MHz * 125 / 12 = 192 MHz*/
   str r0, [r1,#CKGR_PLLAR]    /* Clock Generator PLL A Register           */

 pll_lp:
   ldr r0, [r1,#PMC_SR]        /* PLL A is locked ?                        */
   tst r0, #0x02
   beq pll_lp
```

System Clock II + enable caches

start.s :

```
   /* MCK = PCK/4 */
   ldr r0, =0x0202
   str r0, [r1,#PMC_MCKR]      /* CSS = PLLA, MCK=PCK/4                    */

mck_lp:
   ldr r0, [r1,#PMC_SR]        /* MCKRDY: Master Clock Status ?            */
   tst r0, #0x08
   beq mck_lp

   /* Enable caches */
   mrc p15, 0, r0, c1, c0, 0   /* Move to ARM register from coprocessor */
   orr r0, r0, #(0x1 <<12)
   orr r0, r0, #(0x1 <<2)
   mcr p15, 0, r0, c1, c0, 0   /* Move to coprocessor from ARM registers*/
```

Main program

start.s :

```
.global _main
/* main program */
_main:
    …
/* user code here */
LOOP:
    Switch on LED=1 and/or BUZZ=1
        bl WRITEOUT

    „0.5 second loop1“

    Switch off LED=0 and/or BUZZ=0
        bl WRITEOUT

    „0.5 second loop1“

    b LOOP
/* end user code */
```