



EEL 4744

Menu

- Wonders of the include file
- Bit Masks
- Bit Positions
- Group Configurations



See examples on
web-site:

[ATxmega128aludef.inc](#)

University of Florida, EEL 4744 – File 00
© Dr. Eric M. Schwartz

1

1



EEL 4744



[ATxmega128aludef.inc](#)

Wonders of the Include File: Equates, Bit Masks, and Bit Positions

- Equates:
 - > XL, XH, YL, YH, ZL, ZH defined as r26-r31, near end of include file
 - > See SRAM_END, RAMEND, SRAM_SIZE, etc.
 - > CPU registers, e.g., CPU_RAMPZ, CPU_SREG, CPU_SPL
 - > PORTs, e.g., PORTF_DIRSET and PORTF_OUTSET
- Bit masks (`_bm`)
 - > Allow a user to change only specific bits without knowing the bit positions, e.g., TC0_CLKSEL2_bm
 - `.equ TC0_CLKSEL2_bm = (1<<2) ; Clock Selection bit 2 mask`
- Bit positions (`_bp`)
 - > Allow the programmer to change specific bits, e.g., TC0_CLKSEL2_bp
 - `.equ TC0_CLKSEL2_bp = 2 ; Clock Selection bit 2 position`

University of Florida, EEL 4744 – File 00
© Dr. Eric M. Schwartz

2

2



EEL 4744 Bit Mask & Bit Position

- Bit masks (`_bm`) allow a user to change only specific bits without needing to know the actual bit positions

- Ex:

```
.equ CPU_C_bm = 0x01 (= 0b0000 0001)
.equ CPU_Z_bm = 0x02 (= 0b0000 0010) = (1<<1)
.equ CPU_I_bm = 0x80 (= 0b1000 0000) = (1<<7)
```

Space are not allowed in programs, but can use `_`; ex 0b0101_1010

- Bit positions (`_bp`) allows change to specific bits

- Ex:

```
.equ CPU_C_bp = 0
.equ CPU_Z_bp = 1
.equ CPU_I_bp = 7
```

ATxmega128a1udef.inc

Status Register (SREG)

See doc8331, section 3.14.9

I	T	H	S	V	N	Z	C
7	6	5	4	3	2	1	0

University of Florida, EEL 4744 – File 00
© Dr. Eric M. Schwartz

3

3



EEL 4744

Example Using BP

- Since the RED_LED is on PortD bit 4 (PD4) and the GREEN_LED on PD5, could use the following to turn on the Red and turn off the Green (active-low) LEDs

```
.equ RED_LED_BP = 4
.equ GREEN_LED_BP = 5
...
ldi r16, (1 << RED_LED_BP)
sts PORTD_OUTCLR, r16 ;turn on red LED
ldi r16, (1 << GREEN_LED_BP)
sts PORTD_OUTSET, r16 ;turn off green LED
```

University of Florida, EEL 4744 – File 00
© Dr. Eric M. Schwartz

4

4



EEL 4744 XMEGA TC Clock Sources

- Group Configurations (`_gc`) allow the programmer to change how a particular system works without remember bit positions, e.g.,
`TC_CLKSEL_DIV4_gc`

doc8331:
Table 14-3

ATxmega128aludef.inc

```
; Clock Selection
.equ TC_CLKSEL_OFF_gc = (0x00<<0) ; Timer Off
.equ TC_CLKSEL_DIV1_gc = (0x01<<0) ; System Clock
.equ TC_CLKSEL_DIV2_gc = (0x02<<0) ; System Clock / 2
.equ TC_CLKSEL_DIV4_gc = (0x03<<0) ; System Clock / 4
.equ TC_CLKSEL_DIV8_gc = (0x04<<0) ; System Clock / 8
.equ TC_CLKSEL_DIV64_gc = (0x05<<0) ; System Clock / 64
.equ TC_CLKSEL_DIV256_gc = (0x06<<0) ; System Clock / 256
.equ TC_CLKSEL_DIV1024_gc = (0x07<<0) ; System Clock / 1024
```

CLKSEL [3..0]	Group Config	Description (PS=Prescaler)
0000	Off	None (T/C OFF)
0001	DIV1	PS: Clk
0010	DIV2	PS: Clk/2
0011	DIV4	PS: Clk/4
0100	DIV8	PS: Clk/8
0101	DIV64	PS: Clk/64
0110	DIV256	PS: Clk/256
0111	DIV1024	PS: Clk/1024

TCp_x_CTRLA (p=C,D,E, or F; x=0 or 2)

CTRLA



University of Florida, EEL 4744 – File 00
© Dr. Eric M. Schwartz

5

5



EEL 4744 Group Configuration

- Ex: For asynchronous UART on PC0, 6-bits, odd parity, 2 stop bits

> The following is in the include file

```
.equ USART_CMODE_ASYNCHRONOUS_gc = (0x00<<6)
; Asynchronous Mode
.equ USART_CHSIZE_6BIT_gc = (0x01<<0)
; Character size: 6 bit
.equ USART_PMODE_ODD_gc = (0x03<<4) ATxmega128aludef.inc
; Odd Parity
```

> Add the following for the stop bits (for some reason, **NOT** in the include file)

```
.equ USART_SBMODE_2BIT_gc = (0x01<<3)
.equ USART_SBMODE_1BIT_gc = (0x00<<3)
```

> Then use below to finish the initialization

```
ldi r16, USART_CMODE_ASYNCHRONOUS_gc | USART_CHSIZE_6BIT_gc |
USART_PMODE_ODD_gc | USART_SBMODE_2BIT_gc
sts USARTC0_CTRLA, r16
```

.equ USART_SBMODE_bp = 3 ; Stop Bit Mode bit position

USARTC0_CTRLA

CTRLA



University of Florida, EEL 4744 – File 00
© Dr. Eric M. Schwartz

6

6



EEL 4744

The End!

University of Florida, EEL 4744 – File 00
© Dr. Eric M. Schwartz

7

7