# **ATMEL Studio 7 (main.asm)**

;PROJECT :**ExternalInterrupt84**

;PURPOSE :1st Look at Interrupts: **External Interrupt 0**

;AUTHOR :C. D'Arcy

;DATE :**2020 05 13**

;DEVICE :Dolgin Development Platform **plus Universal Shield**

;MCU :ATtiny84

;STATUS :Working

;REFERENCE :[tn84def.inc](https://github.com/DarkSector/AVR/blob/master/asm/include/tn84def.inc)

.cseg ;optional

.org 0x0000 ;start of vector jump table

 rjmp reset ;lower the address => higher the priority!

.org EXT\_INT0addr ;External Interrupt (see end tn84def.inc)

 rjmp EXT\_INT0ISR ;our ISR (can call it anything)

.def util = r16 ;

.org INT\_VECTORS\_SIZE ;just beyond the IVT

reset: ;PC jumps to here on reset interrupt...

 sbi DDRA,PA0 ;set digital pin 0 for output

 cbi DDRB,PB2 ;guarantee INT0 (PB2) declared for input

 sbi PORTB,PB2 ;set pullup resistor on INT0 (PB2)

 rcall Int0Setup ;configure the required registers to respond to Ext. Int. 0 (PB2)

 sei ;enable Global Interrupt System (I flag in SREG)

hold:

 rjmp hold ;hold here a press the button to toggle the LED on PA0

Int0Setup:

 ldi util,1<<ISC01 ;prepare the Interrupt Sense Control for FALLING edge

 out MCUCR,util ;set it

 ldi util,1<<INT0 ;prepare to respond to Ext. Int. 0

 out GIMSK,util ;set it

 ret

EXT\_INT0ISR:

 sbi PINA,PA0 ;remember this? Fastest way to toggle a PORT bit :)

 reti ;return from interrupt

# Arduino IDE (ExternalInterrupt0.S)

**;PROJECT    :ExternalInterrupt84**

**;PURPOSE    :1st Look at Interrupts: External Interrupt 0**

**;AUTHOR     :C. D'Arcy**

**;DATE       :2020 05 13**

**;DEVICE     :Dolgin Development Platform plus Universal Shield**

**;MCU        :ATtiny84**

**;STATUS     :Working**

**;REFERENCE  :**[**http://darcy.rsgc.on.ca/ACES/TEI4M/iotnx4.h**](http://darcy.rsgc.on.ca/ACES/TEI4M/iotnx4.h)

#include  "avr/io.h"      //pulls in the master set of predefines

.section .text            ;optional

.global main              ;expose this code to the compiler

.org    0x0000            ;start of vector jump table

 rjmp main                ;lower the address => higher the priority!

#define util  r16         //readability is enhanced through the use of aliases for GP Registers

.org    \_VECTORS\_SIZE     ;set Location Counter just beyond IVT

main:                     ;PC jumps to here on reset interrupt...

 sbi  DDRA-0x20,PA0       ;set digital pin 0 for output

 cbi  DDRB-0x20,PB2       ;guarantee INT0 (PB2) declared for input

 sbi  PORTB-0x20,PB2      ;set pullup resistor on INT0 (PB2)

 rcall  Int0Setup         ;configure the required registers to respond to Ext. Int. 0 (PB2)

 sei                      ;enable Global Interrupt System (I flag in SREG)

hold:

 rjmp hold                ;hold here a press the button to toggle the LED on PA0

Int0Setup: ;expose this code to the compiler

 ldi  util,1<<ISC01     ;prepare the Interrupt Sense Control for FALLING edge

 out  MCUCR-0x20,util   ;set it!

 ldi  util,1<<INT0      ;prepare to respond to Ext. Int. 0

 out  GIMSK-0x20,util   ;set it!

 ret

.global INT0\_vect

INT0\_vect:

 sbi  PINA-0x20,PA0     ;remember this? Fastest way to toggle a PORT bit :)

 reti                   ;return from interrupt