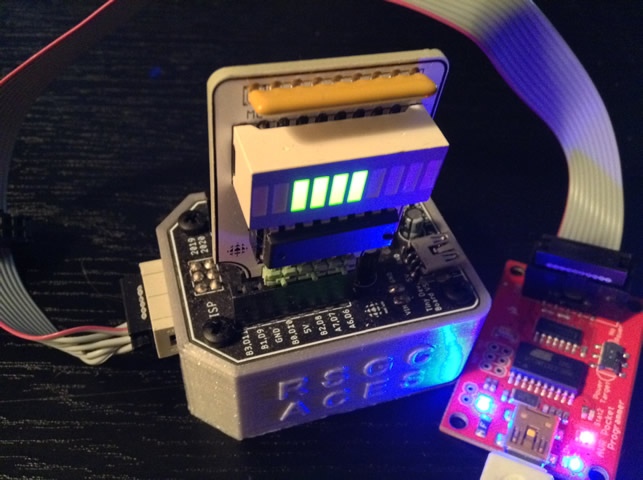
# **ATMEL Studio 7 (Variables.asm)**

;PROJECT :Shiftout (Solution)

;PURPOSE :ICS4U RSGC ACES Project (link below)

;AUTHOR :C. D'Arcy

;DATE :2020 04 19

;DEVICE :Dolgin Development Platform

;MCU :ATtiny84

;COURSE :ICS4U

;STATUS :Working

;REFERENCE :<https://mail.rsgc.on.ca/~cdarcy/Datasheets/doc0856.pdf>

;NOTES :[http://darcy.rsgc.on.ca/ACES/TEI4M/1920/Tasks.html#shiftout](http://darcy.rsgc.on.ca/ACES/TEI4M/1920/Tasks.html%23shiftout)

.def util = r16 ;utility register

.def value = r17 ;holds the value to be displayed

.def mask = r18 ;byte with one set bit to act as a mask

.def dir = r19 ;shift direction: LSBFIRST:0, MSBFIRST:1

.equ OE = PA6 ;595 output enable pin

.equ LATCH = PA7 ;595 latch pin

.equ DATA = PB2 ;595 data pin

.equ CLOCK = PB0 ;595 clock pin

.equ n = 0xF0 ;constant to be presented in binary on the bargraph

.equ LSBFIRST= 0 ;same familiar constants from Arduino days

.equ MSBFIRST= 1 ;ditto

; CODE Segment (default)

.cseg ;locate for Code Segment (FLASH)

; \*\*\*\*\* INTERRUPT VECTOR TABLE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.org 0x0000 ;start of Interrupt Vector Table (IVT) aka. Jump Table

rjmp reset ;lowest interrupt address == highest priority!

; \*\*\*\*\* START OF CODE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.org 0x0100 ;well clear of IVT

reset: ;PC jumps to here (start of code) on reset interrupt

ldi util,1<<LATCH|1<<OE ;create an OR'ed flag set for output pins

out DDRA,util ;set 'em

ldi util,1<<DATA|1<<CLOCK ;create an OR'ed flag set for output pins

out DDRB,util ;set 'em

;ready to shift n, simply request shift order: either LSBFIRST or MSBFIRST

ldi dir,MSBFIRST ;change this manually (M or L)

rcall shiftout ;invoke the shiftout function

wait: ;done!

**rjmp wait ;hold...finished**

shiftout: ;shifts constant n into MBv3

ldi mask,0x80 ;assume order is MSBFIRST

sbrs dir,0 ;if bit 0 is set, it's MSBFIRST

ldi mask,0x01 ;OK, it's LSBFIRST so redefine the mask

cbi PORTA,LATCH ;pull LATCH pin LOW

again:

cbi PORTB,CLOCK ;pull CLOCK pin LOW

ldi value,n ;reload the value to be presented

and value,mask ;mask off the target bit

breq lo ;was it 0?

sbi PORTB,DATA ;no, so pull DATA pin HIGH

rjmp clockit ;ready to clock the 1

lo: cbi PORTB,DATA ;else, it was a 0, so pull DATA pin LOW

clockit:

sbi PORTB,CLOCK ;pull CLOCK pin HIGH

; hmmm, must decide what direction to shift the mask...

sbrs dir,0 ;if bit 0 is set, it's MSBFIRST

rjmp shiftLeft ;OK, it's LSBFIRST

lsr mask ;MSBFIRST so shift the mask right

brne again ;repeat if there are still more bits to stuff in

rjmp done ;we're done, so only one more thing to do

shiftLeft:

lsl mask ;LSBFIRST, so shift the mask left

brne again ;repeat if there are still more bits to stuff in

done:

sbi PORTA,LATCH ;pull LATCH pin HIGH to present internal

ret ;finished, return.

# Arduino IDE (Variables.S)

;PROJECT    :Shiftout (Solution)

;PURPOSE    :Response to 2019-2020 ICS4U Project (link below)

;AUTHOR     :C. D'Arcy

;DATE       :2020 04 19

;DEVICE     :Dolgin Development Platform

;MCU        :ATtiny84

;COURSE     :ICS4U

;STATUS     :Working

// REFERENCE:<http://darcy.rsgc.on.ca/ACES/TEI4M/Assembly/AVR8AssemblyLanguage.html>

// REFERENCE:[http://darcy.rsgc.on.ca/ACES/TEI4M/1920/Tasks.html#shiftout](http://darcy.rsgc.on.ca/ACES/TEI4M/1920/Tasks.html%23shiftout)

#include  <avr/io.h>            //required to reference ports by name (-0x20 offset)

#define util   r16              //improve readability with aliases

#define value  r17              //alternate register

#define mask   r18              //byte with one set bit to act as a mask

#define dir    r19              //shift direction: LSBFIRST:0, MSBFIRST:1

.equ    OFFSET,   0x20          ;avr-as requires offset address for PORTS

.equ    PRTA,     PORTA-OFFSET  ;0x3B – 0x20 = 0x1B

.equ    DRA,      DDRA-OFFSET   ;0x3A – 0x20 = 0x1A

.equ    PRTB,     PORTB-OFFSET  ;0x38 – 0x20 = 0x1A

.equ    DRB,      DDRB-OFFSET   ;0x37 – 0x20 = 0x17

.equ    OE,       PA6           ;595 output enable pin

.equ    LATCH,    PA7           ;595 latch pin

.equ    DATA,     PB2           ;595 data pin

.equ    CLOCK,    PB0           ;595 clock pin

.equ    n,        0xF0          ;constant to be presented in binary on the bargraph

.equ    LSBFIRST, 0             ;same familiar constants from Arduino days

.equ    MSBFIRST, 1             ;ditto

; CODE Segment (default)

.text                           ;locate for Code Segment (FLASH)

; \*\*\*\*\* INTERRUPT VECTOR TABLE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.org    0x0000                  ;start of Interrupt Vector Table (IVT) aka. Jump Table

   rjmp     main               ;lowest interrupt address == highest priority!

; \*\*\*\*\* START OF CODE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.org    0x0100                  ;well clear of IVT

.global main

main:

  ldi     util,1<<LATCH|1<<OE  ;create an OR'ed flag set for output pins

  out     DRA,util             ;set 'em

  ldi     util,1<<DATA|1<<CLOCK;create an OR'ed flag set for output pins

  out     DRB,util             ;set 'em

;ready to shift n, simply request shift order: either LSBFIRST or MSBFIRST

  ldi     dir,MSBFIRST         ;change this manually (M or L)

  rcall   shiftout             ;invoke the shiftout function

wait:                           ;done.

**rjmp    wait                 ;hold...finished**!

shiftout:                       ;shifts constant n into MBv3

  ldi     mask,0x80            ;assume order is MSBFIRST

  sbrs    dir,0                ;if bit 0 is set, it's MSBFIRST

  ldi     mask,0x01            ;OK, it's LSBFIRST so redefine the mask

  cbi     PRTA,LATCH           ;pull LATCH pin LOW

again:

  cbi     PRTB,CLOCK           ;pull CLOCK pin LOW

  ldi     value,n              ;reload the value to be presented

  and     value,mask           ;mask off the target bit

  breq    lo                   ;was it 0?

  sbi     PRTB,DATA            ;no, so pull DATA pin HIGH

  rjmp    clockit              ;ready to clock the 1

lo:

  cbi     PRTB,DATA            ;else, it was a 0, so pull DATA pin LOW

clockit:

  sbi     PRTB,CLOCK           ;pull CLOCK pin HIGH

;hmmm, must decide what direction to shift the mask...

  sbrs    dir,0                ;if bit 0 is set, it's MSBFIRST

  rjmp    shiftLeft            ;OK, it's LSBFIRST

  lsr     mask                 ;MSBFIRST so shift the mask right

  brne    again                ;repeat if there are still more bits to stuff in

  rjmp    done                 ;we're done, so only one more thing to do

shiftLeft:

  lsl     mask                 ;LSBFIRST, so shift the mask left

  brne    again                ;repeat if there are still more bits to stuff in

done:

  sbi     PRTA,LATCH           ;pull LATCH pin HIGH to present 595's internal latches on output pins

  ret                          ;finished, return.