// PROJECT :BicolorBlink (BicolorBlink.S)

// PURPOSE :ICS4U-E Introduction to Standalone Assembly within the Arduino IDE

// COURSE :ICS4U-E

// AUTHOR :C. D'Arcy

// DATE :2025 01 10

// MCU :328P

// STATUS :Working

#include <avr/io.h> // in turn includes 328p-specific predefines (iom328p.h)

#define DDR DDRB-0x20 //avr-as requires **relative** addressing

#define PORT PORTB-0x20 //ditto

#define DATA 0xAA //data value to be presented

#define NUM 8 //number of bits to examine

.global setup ;must be visible from outside this file

setup: ;one-time call from main.cpp

 sbi DDR,0 ;set PORTB0 for output

 sbi DDR,1 ;set PORTB1 for output

 **ret** ;exit from function

.global loop ;must be visible from outside this 'file'

loop: ;repeated invocation from main.cpp

 ldi r16,DATA ;place the value on the table

 ldi r17,NUM ;set the LCV (loop control variable)

next:

 lsl r16 ;logical shift left to load the C flag of the SREG

 brcs doGrn ;if it's 1, show green

 rcall red ;otherwise its 0, so show red

 rjmp admire ;branch to observe/confirm

doGrn:

 rcall green ;show green

admire:

 rcall delay1s ;pausing for 1s

 dec r17 ;one less iteration

 brne next ;if we're not finished, repeat

 rcall LEDOff ;we're done, so turn both colours off and pause for 3s…

 rcall delay1s ;pause for 1s

 rcall delay1s ;pause for 1s

 rcall delay1s ;pause for 1s

 **ret** ;exit from function

red:

 sbi PORT,0 ;aka digitalWrite(?,HIGH)

 cbi PORT,1 ;aka digitalWrite(?,LOW)

 **ret** ;exit from function

green:

 sbi PORT,1 ;aka digitalWrite(?,HIGH)

 cbi PORT,0 ;aka digitalWrite(?,LOW)

 **ret** ;exit from function

LEDOff:

 cbi PORT,0 ;aka digitalWrite(?,LOW)

 cbi PORT,1 ;aka digitalWrite(?,LOW)

 **ret** ;exit from function

delay1s:

; Assembly code auto-generated

; by utility from Bret Mulvey

; Delay 16 000 000 cycles

; 1s at 16.0 MHz

 ldi r18, 82

 ldi r19, 43

 ldi r20, 0

L1:

 dec r20

 brne L1

 dec r19

 brne L1

 dec r18

 brne L1

 lpm

 nop

 **ret**