Lab 2 notes:

Input Terminal

Switch on=closed circuit= Input grounded

Switch off=open circuit= input is Vcc

Output Terminal

Switch on=Input grounded, if I copy grounded 0 to output, output = 0. Voltage across Diode

Switch off=input VCC, if I copy Vcc input to output, output =1. Voltage across diode =0

Part B Code

.include "ATxmega128A1Udef.inc"

.list

.org 0x0000

rjmp MAIN

.dseg

.equ outputall=0xFF ;set all for output

.equ switch=0x00 ;set all for input

.cseg

.org 0x200 ;where we will start the program

MAIN:

ldi r16, switch ;load inputs(0x00) to r16

sts PORTA\_DIRCLR, r16 ;set Port A to be input

ldi r17, outputall ;load outputs (0xFF) to r17

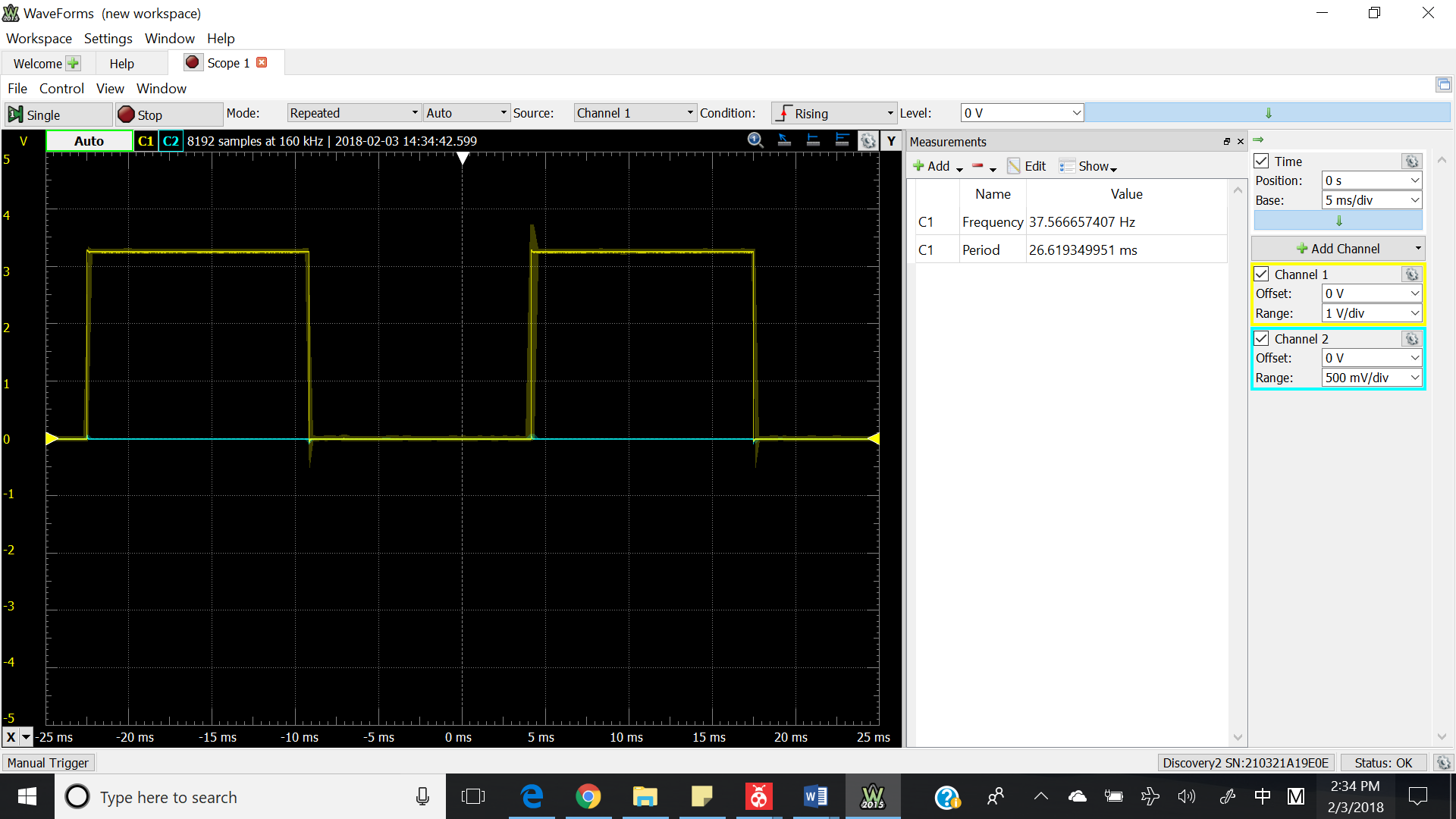
sts PORTC\_DIRSET, r17 ;set Port C to be output

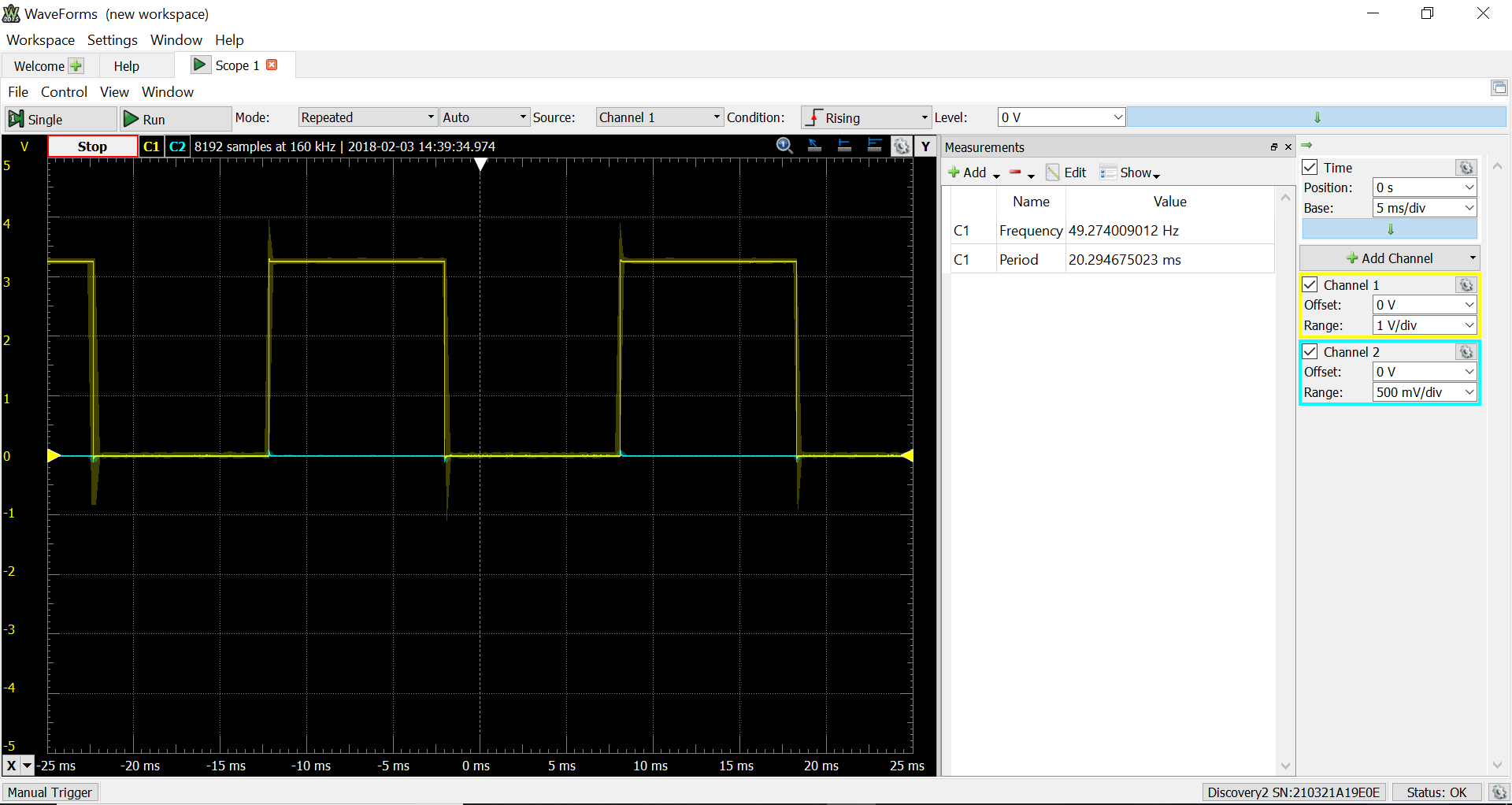
LOOP:

lds r16, PORTA\_IN ;load value at input to r16

sts PORTC\_OUT, r16 ;input to output

rjmp LOOP ;infinite loop





Part C code with just the 10ms delay

.include "ATxmega128A1Udef.inc"

.list

.org 0x00

rjmp MAIN

.equ stack\_init=0x3FFF

MAIN:

ldi YL, low(stack\_init)

out CPU\_SPL, YL

ldi YL, high(stack\_init)

out CPU\_SPH, YL ;initialize high byte of stack pointer

ldi r16, 0x80 ;set last LED as output

sts PORTC\_DIRSET, r16

LOOP:

ldi r17, 0x80

sts PORTC\_OUTTGL, r17

rcall Delay\_10ms

rjmp LOOP

Delay\_10ms:

push r16

push r17

ldi r17, 15

START:

ldi r16, 0xFF

HI:

cpi r16,0

breq SECOND

dec r16

rjmp HI

SECOND:

cpi r17, 0

breq rdone

dec r17

rjmp START

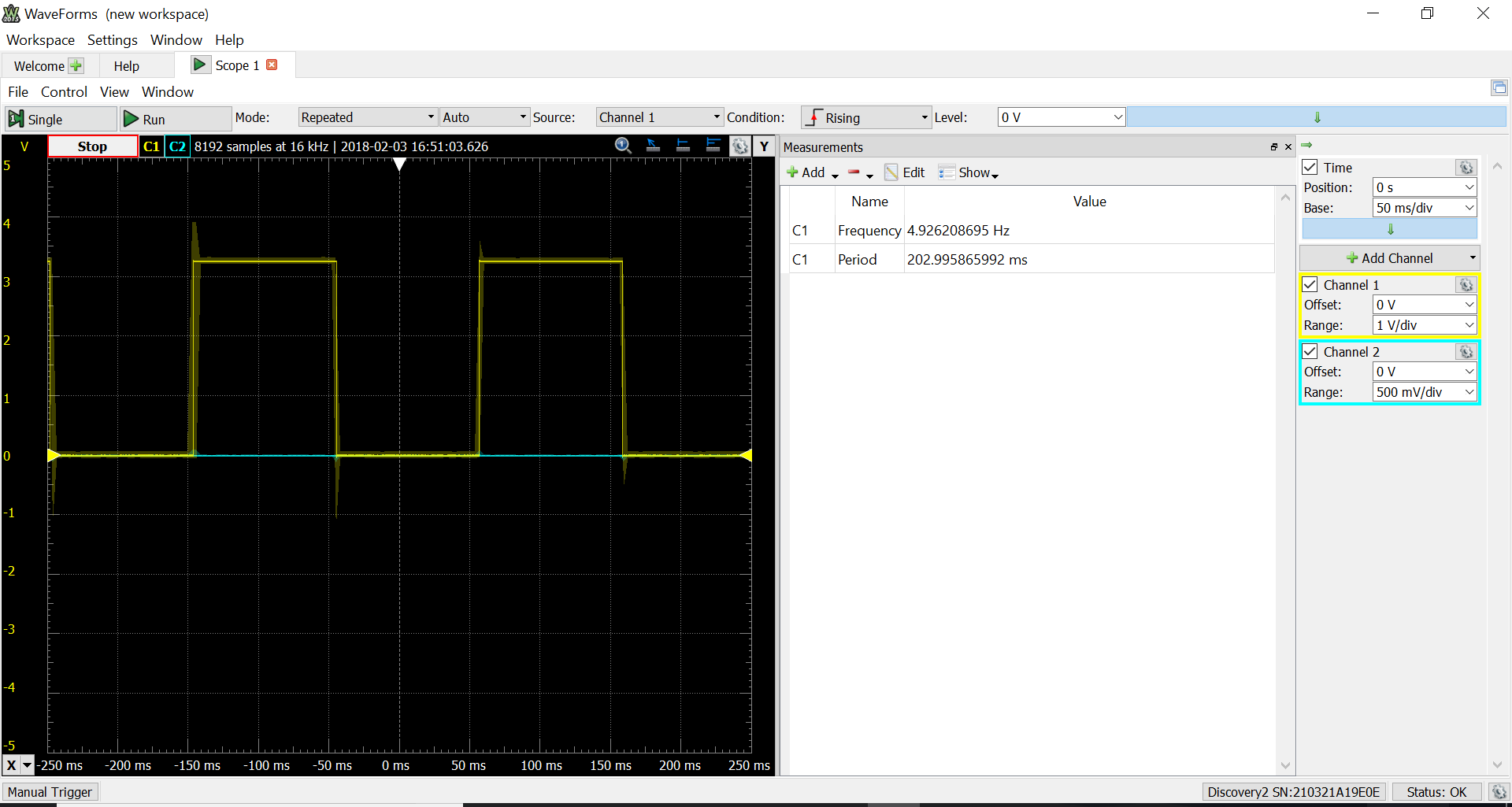
rdone:

pop r17

pop r16

ret

Part C subroutine with multiple of 100ms delays



.include "ATxmega128A1Udef.inc"

.list

.org 0x00

rjmp MAIN

.equ stack\_init=0x3FFF ;initialize stack pointer

MAIN:

ldi YL, low(stack\_init) ;initialize low byte of stack pointer

out CPU\_SPL, YL

ldi YL, high(stack\_init) ;initialize hgih byte of stack pointer

out CPU\_SPH, YL

ldi r16, 0x80 ;set last LED as output

sts PORTC\_DIRSET, r16

LOOP:

ldi r17, 0x80 ;toggle last LED of PORTC

sts PORTC\_OUTTGL, r17

rcall Delay\_100ms ;call delay 100ms subroutine

rjmp LOOP

Delay\_100ms:

push r20

ldi r20, 9

REPEAT:

rcall Delay\_10ms

cpi r20, 0

breq DONE

dec r20

rjmp REPEAT

DONE:

pop r20

ret

Delay\_10ms: ;delay 10ms subroutine

push r16 ;push

push r17

ldi r17, 15 ;do this loop 15 times. will be obvious later in the code

START:

ldi r16, 0xFF ;some value to take away running time

HI:

cpi r16,0

breq SECOND ;code to take away running time

dec r16

rjmp HI

SECOND:

cpi r17, 0

breq rdone ;code to take away running time

dec r17

rjmp START

rdone: ;by the time the code gets back here.

pop r17 ;should be a 10ms delay

pop r16

ret ; return to main routine

PART D

.include "ATxmega128A1Udef.inc"

.list

.org 0x00 ;

rjmp MAIN

.equ stack\_init=0x3FFF ;initialize stack pointer

.org 0x200

Table: .db 0b10000001, 0b01000010, 0b00100100, 0b00011000,0b00100100, 0b01000010,0b10000001,0b00000000

.org 0x300

MAIN:

ldi YL, low(stack\_init) ;initialize low byte of stack pointer

out CPU\_SPL, YL

ldi YL, high(stack\_init) ;initialize hgih byte of stack pointer

out CPU\_SPH, YL

ldi r17, 0xFF ;to turn off all active low LED later

ldi r18, 0x00 ;to turn off all active high stuff

ldi r22, 0b00001100 ;set S1 and S2 as input

sts PORTF\_DIRCLR, r22

ldi r22, 0b00110000 ;bit 5 is red, bit 6 is green

sts PORTD\_DIRSET, r22

sts PORTD\_OUTSET, r17 ;to turn off the LED for now

ldi r16, 0xFF

sts PORTC\_DIRSET, r16 ;set Port C to be output

LEDLOOP:

ldi ZL, low(Table << 1)

ldi ZH, high(Table << 1)

ldi r20, 8

LEDSWITCH:

lpm r16, Z+ ;first and last LED first

sts PORTC\_OUTSET, r17

sts PORTC\_OUTCLR, r16

rcall Delay\_100ms

lds r23, PORTF\_IN ;if statement code

bst r23, 3 ;if pressed. store bit 3 of r23 into T-flag

brtc PRESS ;if pressed. voltage is 0

dec r20

cpi r20, 0

breq LEDLOOP

rjmp LEDSWITCH

PRESS:

bst r16, 4 ;store bit 4 of r16 into T-flag

brts GREEN

brtc RED

GREEN:

ldi r23, 0b00100000 ;low true

sts PORTD\_OUTCLR, r23 ;turn on GREEN

rjmp RESET

RED:

ldi r23, 0b00010000 ;low true

sts PORTD\_OUTCLR, r23 ;turn on LED

rjmp RESET

RESET:

lds r23, PORTF\_IN ;check if user have pressed S1 to reset the game. if not. infinite until he resets it

bst r23, 2

brtc OFF

brts RESET

OFF:

sts PORTD\_OUTSET, r17 ;to turn off all active low LED

rjmp LEDLOOP

Delay\_100ms:

push r20

ldi r20, 9 ;suppose to be 9

REPEAT:

rcall Delay\_10ms

cpi r20, 0

breq DONE

dec r20

rjmp REPEAT

DONE:

pop r20

ret

Delay\_10ms: ;delay 10ms subroutine

push r16 ;push

push r17

ldi r17, 15 ;do this loop 15 times. will be obvious later in the code

START:

ldi r16, 0xFF ;some value to take away running time

HI:

cpi r16,0

breq SECOND ;code to take away running time

dec r16

rjmp HI

SECOND:

cpi r17, 0

breq rdone ;code to take away running time

dec r17

rjmp START

rdone: ;by the time the code gets back here.

pop r17 ;should be a 10ms delay

pop r16

ret ; return to main routine

Part D Screenshot

