CPE301 – SPRING 2019

Design Assignment 2A

Student Name: Biruk Gebremeskel

Student #: 5002212730

Student Email: gebreb1@unlv.nevada.edu

Primary Github address: https://github.com/gebreb1

Directory: DA2A

Design Assignment 2A:

The goal of the assignment is use GPIO and delays:

1. Design a delay subroutine to generate a waveform on PORTB.2 with 60% DC and 0.725 sec period.

2. Connect a switch to PORTC.2 (active high - turn on the pull up transistor) to poll for an event to turn on the led at PORTB.2 for 1.250 sec after the event.

Submission: The following are required for successful completion of the design assignment:

a. AVR ASM code that has been compiled and working for all tasks. Verify the period and duty cycle of the waveforms in simulation and emulation.

b. AVR C code that has been compiled and working for all tasks. Verify the period and duty cycle of the waveforms in simulation and emulation.

c. The C code should be well documented with explanation of every instruction.

d. A word document that contains the code with comments, complete schematics, that includes the AVR, components connected on the breadboard and LED should be included. Follow the template provided.

e. A snapshot of the board with connected components and a video of the complete LED bar blink sequence should be recorded and uploaded to Youtube and the line to be provided for each task.

f. The git directory should have DA2\DA2T1, DA2\DA2T2, … folders, with one doc file and video link file.

**Answer**

* 1. **Assembly Code**

.org 0

LDI R16, 32

SBI DDRB, 5 ;output from Port5

LDI R17, 0

OUT PORTB, R17 ;initializing PortB to 0

LDI R20, 5

STS TCCR1B, R20 ;setting 1024 a prescaler

begin:

RCALL delay ;calling the delay function

EOR R17, R16 ;xor 32 with 0

OUT PORTB, R17 ;output the value from the previous operation

RJMP begin ;jump to begin

delay:

LDS R29, TCNT1H ;get the high value of the counter

LDS R28, TCNT1L ;get the low value of the counter

CPI R28, 0xF3 ;compare low value with F3

BRSH body ;branch if same or higher

RJMP delay ;jump to delay

body:

CPI R29, 0x00 ;compare high value to 0

BRSH done ;branch if same or higher

RJMP delay ;jump to delay

done:

LDI R20, 0x00

STS TCNT1H, R20 ;reset TCNT1H

LDI R20, 0x00

STS TCNT1L, R20 ;reset TCNT1L

RET

* 1. **C-code**

int main()

{

DDRB = 0b00000100;

PORTB = 0x00;

TCCR1B = 5; // set prescaler to 1024

while(1) {

if(TCNT1 == 0x0D47){

PORTB ^= 0b00000100;

TCNT1 = 0x00;

}

}

}

# Where, 0.725 \* 60% = 0.435

# TCNT = 8Mhz\*(0.425)/(1024) = 3399 = 0xD47

1. 1. **Assembly Code**

.ORG 0

MAIN:

LDI R16, 32

SBI DDRB, 5 ;set PORTB 5 to output

LDI R17, 0

LDI R18, 0

OUT DDRC, R18 ;set DDRC to input

LDI R20, 13

STS TCCR1B, R20 ;set prescaler to 1024

IN R20, PIND ;get input values

ANDI R20, 0b00000010 ;bitmask input value

CPI R20, 0b00000010 ;check if button was pressed

BRNE MAIN

begin:

RCALL delay ;call delay function

EOR R17, R16 ;xor 32 with 0

OUT PORTB, R17 ;output to PORTB 5

RJMP begin

delay:

LDS R29, TCNT1H ;get the upper half of counter

LDS R28, TCNT1L ;get lower half of counter

CPI R28, 0xF3 ;check if TCNT1L is 0xF3

BRSH body ;branch if same or higher

RJMP delay ;jump to delay

body:

CPI R29, 0x00 ;check if TCNT1H is 0x00

BRSH done ;branch if same or higher

RJMP delay ;jump to delay

done:

LDI R20, 0x00

STS TCNT1H, R20 ;reset TCNT1H to 0

LDI R20, 0x00

STS TCNT1L, R20 ;reset TCNT1L to 0

RET

* 1. **C-code**

int main(void)

{

DDRC = 0x00; //set DDRC as input

DDRB = 0xFF; //set DDRB as output

while(1)

{

if((PINC & 0b00000100) == 0b00000100) //check if the button was pressed

{

*\_delay\_ms*(1000); //delay of 1000ms

PORTB = 0XFF; //set PORTB 1 to output

}

}

}

“This assignment submission is my own, original work”.

Biruk Gebremeskel