**CG1112 Engineering Principles and Practices**

**Week 4 Studio 2 – PWM**

**Answer Book**

**Student Number: A0214561M**

**Name: Zhuang Jianning**

**Pre-start Check (Check Fail: -5 mark penalty)**

**Did you name your file AxxxxxxY.docx, where AxxxxxxY is your student number? -2 marks if NO.**

**Did you fill in your student number and name above? -3 marks if NO.**

Question 1 (1 mark)

Diagram

Description automatically generatedGraphical user interface, diagram, text

Description automatically generated

Question 2 (1 mark)

Period: 2.04ms

High-Time: 0.200ms

Low-Time: 1.84ms

Duty-Cycle: 9.8%

Question 3 (1 mark)

The duty cycle is determined by the value stored in the OCR0A register

Duty-Cycle = OCR0A/255 \* 100%

= 25/255 \* 100

= 9.80%

This matches the value from the waveform

Question 4 (1 mark)

We only want to keep the LED turned on at a fixed voltage determined by the duty cycle, hence there is no need to change any variables or the duty cycle(OCR0A) when the Output Compare interrupt is enabled.

Question 5 (1 mark)

TCCR0A = 0b11000001;

Set COM0A to 11 such that we set OC0A on compare match when up-counting and clear OC0A on compare match when down-counting, which generates a complement of the current waveform.

This is because in the Phase Correct PWM mode, the timer counts from 0 to 255 and then back down to 0. When we set COM0A to 11, the output turns ON as the timer hits the output compare register value 25 on the way up, and turns back OFF as the timer hits the output compare register value 25 on the way down. Hence OC0A is ON from 25 to 255 and back down to 25 while OC0A is OFF from 25 to 0 back up to 25, which is the complement of before when we set COM0A to 10 as OC0A is OFF from 25 to 255 and back down to 25 while OC0A is ON from 25 to 0 back up to 25.

Question 6 (1 mark)

Graphical user interface

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidence

Question 7 (2 marks)

#define PIN7 (1 << 7)

#define PIN6 (1 << 6)

#define PIN5 (1 << 5)

#define PIN4 (1 << 4)

#define PIN3 (1 << 3)

#define PIN2 (1 << 2)

#define PIN1 (1 << 1)

#define PIN0 (1 << 0)

static volatile int fade = 0; //0 to fade in and 1 to fade out

void setup() {

TCNT0 = 0;

TCCR0A = 0b10000001; // Set OCOM0A to 10 and WGM to 01

TIMSK0 |= 0b10; // Enable Int for Output Compare Match

OCR0A = 25;

TCCR0B = 0b00000011; // Set clk source to clk/64

//Set PORTD Pin 6 (Arduino Pin 6) as Output

DDRD |= PIN6;

sei();

}

ISR(TIMER0\_COMPA\_vect)

{

if (OCR0A == 255){

fade = 1;

}

if (OCR0A == 0){

fade = 0;

}

if (fade == 1){

OCR0A--;

}

else{

OCR0A++;

}

}

void loop() {

while (1)

{

}

}

I declared a global variable fade such that 0 represents the LED fading in and 1 represents the LED fading out.

static volatile int fade = 0; //0 to fade in and 1 to fade out

The PWM duty cycle is increased or decreased based on the value of fade at the end of each PWM signal and the fading in/out effect is flipped when OCR0A reaches the max/min value.

Question 8 (1 mark)

TCCR0B = 0b00000101;

By setting clk source to clk/1024, the frequency of PWM is smaller, hence the TCNT0 register increments at a slower frequency so interrupts to change OCR0A are enabled at a slower rate resulting in a slower fading effect.

Question 9 (1 mark)

When right\_motor\_forward() is called, COM0A is set to 10 while COM0B is set to 00 (OC0B is disconnected). A PWM signal will be generated to AIN2 while AIN1 is driven LOW, hence the wheel spins in the forward direction. When right\_motor\_reverse() is called, COM0A is set to 00 (OC0A is disconnected) while COM0B is set to 10. A PWM signal will be generated to AIN1 while AIN2 is driven LOW, hence the wheel spins in the reverse direction.

**For TA Use:**

**Studio Marks: \_\_\_\_\_\_\_\_\_\_\_ / 10**

**PENALTIES: \_\_\_\_\_\_\_\_\_\_\_\_/-5**

**Total: \_\_\_\_\_\_\_\_\_\_\_\_/ 10**