CMPE 242 Spring 2021
Midterm Exam Version A

First Name: _____ Last Name: _____
Student ID: _____ Email:_____ (Total 30 points)

Honor Code: All students agree, individually and collectively, that they will not give or receive unpermitted aid during examinations. Engaging in such unpermitted actions is considered cheating. By my signature, I affirm that I have adhered to the spirit of the Honor Code.

Signature:_____Date:_____

Please Read: instructions for Zoon based online exam.
1. this is an one hour exam, starting from 3:00 and last till 4:00;
2. use cellphone to scan your papers, save them as pdf file, then you must convert it zip file;
3. Submission must be made in the zip file format on line before 4:15, late submission will make the paper disqualified; After 4:15 no submission will be accepted (based on the time stamp of the submission). Thank you and stay safe**.**

This Midterm Exam consists of 3 Questions, total 30 points. The Questions start next page.

QUESTION 1 (5 points) Using ARM-11 or your selected target platform, answer the following design questions:

1.1 (5 pts) Design input and output testing circuit by first drawing circuit schematics and then calculate resistors value to realize the above required input/output testing functions? assume LED current = 10 mA, and V_led = 1.2 V.

QUESTION 2 (15 points) Design a PID (proportion, integral, and derivative) controller by answering the following questions:

2.0 (1 pt) Draw a block diagram of a PID controller?

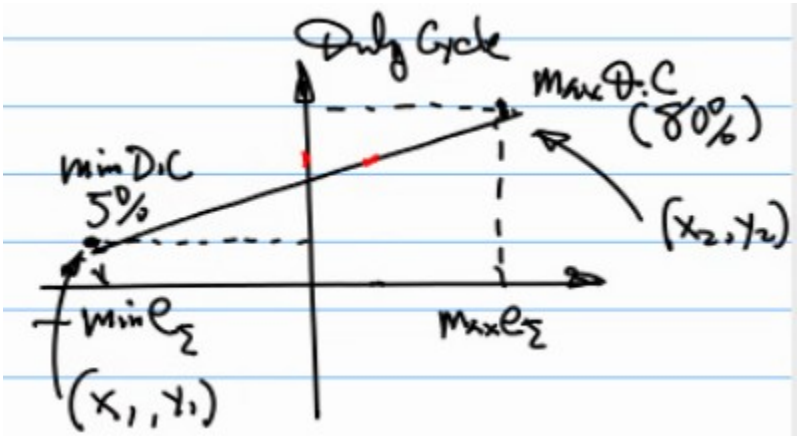
2.1 (2 pts) Explain the limitation of the forward difference (FD) and backward difference (BD) kernels, derive central difference (CD) kernel and explain its key feature?

2.2 (3 pts) Explain what can Laplace of Gaussian (LoG) kernel offer in terms of improving the performance of CD kernel? Find LoG(0), LoG(1) and Log(-1)? Assum sigma = 1 for simplicity (LoG(x,y) formula is given in the Appendix).

2.3 (4 pts) In a given PID controller in the Appendix, e(t) is measured and tabulated below, compute (a) the derivatives of error e(t) by convolution using the CD kernel, and the integral of error e(t) by counting the history of the error going back one time step.

2.4 (5 pts) Based on the PID Controller's e(t)_{sigma} (Proportion, derivative, and integral error at time t), find the control action, e.g., a duty cycle (DC) of the PWM signal for time t=2. Note use the following Figure for your calculation. In this figure, the min e_{sigma} is -500, and max e_{sigma} is 500, the min duty cycle is 5%, the max duty cycle is 80%.

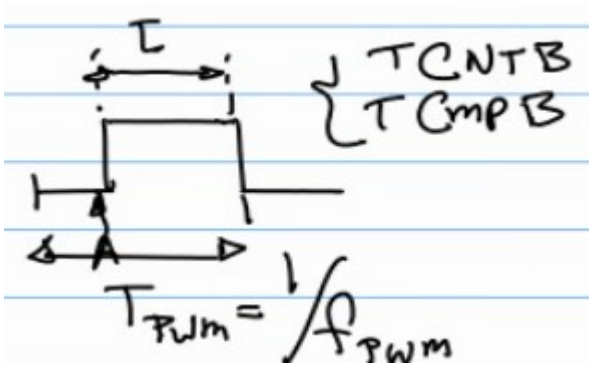
t=0 e(0)=1.3	t=1 e(t)=4.5	t=2 e(t)=10	t=3 e(t)=8	t=4 e(t)=3	t=5 e(t)=-1



QUESTION 3 (10 Points) Design PID controller for stepper motor control, by answering the following questions:

3.1 (3 pts) Design by drawing schematics for stepper motor control with your target embedded board (an external connector), a stepper motor drive board, and NEMA 14 stepper motor. Be sure to provide all the signal connections with label for each of them, and use "arrow" to indicate the signal flow direction (suppose PWM and GPP are employed in this design);

3.2 (2 pts) Suppose in you design, f_{pwm} = 10KHz, CLK_p (peripheral clock) = 25 MHz, PWM duty cycle has to be set to 60% as in figure below. In ARM11, the count N for TCNTB and the count M for TCMPB special purpose registers have to calculated, find N and M?



3.3 (3 pts) Now modify the PWM driver code (see code listing in Appendix), explain in details (bitwise) what do the following 4 lines of code do?

(a) what does line 61 do? Explain in bitwise details what does line 62 do?

(b) what does line 63 do, explain bitwise details?

(c) what do line 83 and 84 do?

```
61      tmp = readl(S3C64XX_GPFCON);
62      tmp &= ~(0x3U << 28);
63      tmp |=  (0x2U << 28);
64      writel(tmp, S3C64XX_GPFCON);
…
83      tcnt  = (pclk/50/16)/freq;
84      tduty  = tcnt*duty;
```

3.4 (2 pts) Describe by drawing a flow chart to describe the process of writing/modifying a device driver function? (Starting from (a) Kernel Source OS distribution, (b) Cross Compiler Tool Chain distribution, and (c) CPU datasheet, to the completion of kernel image with the device driver, be sure to include the explanation of $make menuconfig and Kconf .

Appendix 2 SPR for GPECON

Reference: CPU Datasheet, 10.4 REGISTER DESCRIPTION

GPECON	0x7F008080	Configuration Register
GPEDAT	0x7F008084	Data Register
GPEPUD	0x7F008088	Pull up/down Register

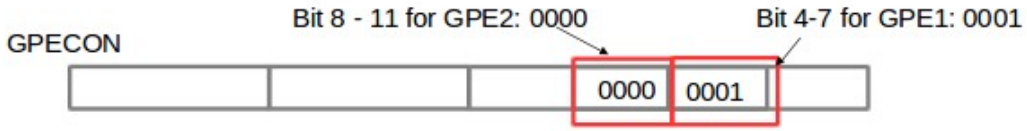
GPECON
(Compiler used name: GPECON)

Example: GPE Init & Config,

Set GPE1 output, "1"
GPE2 input, "0"

So the binary pattern is 0x10

GPE1	[7:4]	0000 = Input	0001 = Output
		0010 = PCM EXTCLK[1]	0011 = I2S CDCLK[1]
		0100 = AC97 RESETn	0101 = Reserved
		0110 = Reserved	0111 = Reserved
GPE2	[11:8]	0000 = Input	0001 = Output
		0010 = PCM FSYNC[1]	0011 = I2S LRCLK[1]
		0100 = AC97 SYNC	0101 = Reserved
		0110 = Reserved	0111 = Reserved



There are five control registers including GPECON, GPEDAT, GPEPUD GPECONSLP and GPEPUDSLP in the Port E Control Registers.

Register	Address	R/W	Description	Reset Value
GPECON	0x7F008080	R/W	Port E Configuration Register	0x00
GPEDAT	0x7F008084	R/W	Port E Data Register	Undefined
GPEPUD	0x7F008088	R/W	Port E Pull-up/down Register	0x00000155
GPECONSLP	0x7F00808C	R/W	Port E Sleep mode Configuration Register	0x0
GPEPUDSLP	0x7F008090	R/W	Port E Sleep mode Pull-up/down Register	0x0

GPDCON	Bit	Description		Initial State
GPE0	[3:0]	0000 = Input	0001 = Output	0000
		0010 = PCM SCLK[1]	0011 = I2S CLK[1]	
		0100 = AC97 BITCLK	0101 = Reserved	
		0110 = Reserved	0111 = Reserved	
GPE1	[7:4]	0000 = Input	0001 = Output	0000
		0010 = PCM EXTCLK[1]	0011 = I2S CDCLK[1]	
		0100 = AC97 RESETn	0101 = Reserved	
		0110 = Reserved	0111 = Reserved	
GPE2	[11:8]	0000 = Input	0001 = Output	0000
		0010 = PCM FSYNC[1]	0011 = I2S LRCLK[1]	
		0100 = AC97 SYNC	0101 = Reserved	
		0110 = Reserved	0111 = Reserved	
GPE3	[15:12]	0000 = Input	0001 = Output	0000
		0010 = PCM SIN[1]	0011 = I2S DI[1]	
		0100 = AC97 SDI	0101 = Reserved	
		0110 = Reserved	0111 = Reserved	
GPE4	[19:16]	0000 = Input	0001 = Output	0000
		0010 = PCM SOUT[1]	0011 = I2S DO[1]	
		0100 = AC97 SDO	0101 = Reserved	
		0110 = Reserved	0111 = Reserved	

GPEDAT	Bit	Description
GPE[4:0]	[4:0]	When the port is configured as input port, the corresponding bit is the pin state. When the port is configured as output port, the pin state is the same as the corresponding bit. When the port is configured as functional pin, the undefined value will be read.

GPEPUD	Bit	Description
GPE[n]	[2n+1:2n] n = 0~4	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved

GPESLPCON	Bit	Description	Initial State
GPE[n]	[2n+1:2n] n = 0~4	00 = output 0 01 = output 1 10 = input 11 = Previous State	00

GPEPUDSLP	Bit	Description
GPE[n]	[2n+1:2n] n = 0~4	00 = pull-up/down disabled 01 = pull-down enabled 10 = pull-up enabled 11 = Reserved.

APPENDIX PWM Driver Function (partial list)

```
static void PWM_Set_Freq( unsigned long freq, duty )  //Harry: add "duty"
{
        unsigned long tcon;
        unsigned long tcnt;
        unsigned long tduty;  //Harry
        unsigned long tcfg1;
        unsigned long tcfg0;

        struct clk *clk_p;
        unsigned long pclk;

        unsigned tmp;

        tmp = readl(S3C64XX_GPFCON);
        tmp &= ~(0x3U << 28);
        tmp |=  (0x2U << 28);
        writel(tmp, S3C64XX_GPFCON);

        tcon = __raw_readl(S3C_TCON);
        tcfg1 = __raw_readl(S3C_TCFG1);
```

```
tcfg0 = __raw_readl(S3C_TCFG0);

//prescaler = 50
tcfg0 &= ~S3C_TCFG_PRESCALER0_MASK;
tcfg0 |= (50 - 1);

//mux = 1/16
tcfg1 &= ~S3C_TCFG1_MUX0_MASK;
tcfg1 |= S3C_TCFG1_MUX0_DIV16;

__raw_writel(tcfg1, S3C_TCFG1);
__raw_writel(tcfg0, S3C_TCFG0);

clk_p = clk_get(NULL, "pclk");
pclk  = clk_get_rate(clk_p);
tcnt  = (pclk/50/16)/freq;
tduty  = tcnt*duty;      //Harry: duty is x% of the period, duty cycle
```

(the above for the reference, cut from here)

APPENDIX GPFCON

GPFCON	Bit	Description	Initial State
GPF0	[1:0]	00 = Input 01 = Output 10 = CAMIF CLK 11 = External Interrupt Group 4[0]	00
GPF1	[3:2]	00 = Input 01 = Output 10 = CAMIF HREF 11 = External Interrupt Group 4[1]	00
GPF2	[5:4]	00 = Input 01 = Output 10 = CAMIF PCLK 11 = External Interrupt Group 4[2]	00
...			
GPF12	[25:24]	00 = Input 01 = Output 10 = CAMIF YDATA[7] 11 = External Interrupt Group 4[12]	00
GPF13	[27:26]	00 = Input 01 = Output 10 = PWM ECLK 11 = External Interrupt Group 4[13]	00
GPF14	[29:28]	00 = Input 01 = Output 10 = PWM TOUT[0] 11 = CLKOUT[0]	00
GPF15	[31:30]	00 = Input 01 = Output 10 = PWM TOUT[1] 11 = Reserved	00

APPENDIX LoG(x,y)

$$\frac{\partial^2}{\partial x^2} G(x,y) = -\frac{1}{\sqrt{2\pi}\sigma^3} e^{-\frac{x^2+y^2}{2\sigma^2}} + \frac{x^2}{\sqrt{2\pi}\sigma^5} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \ldots (4)$$

(END)