UNIVERSITI SAINS MALAYSIA

Peperiksaan Semester Pertama
Sidang Akademik 2011/2012

Januari 2012

EEM 352 – REKABENTUK MEKATRONIK II

Masa : 2 Jam

Sila pastikan bahawa kertas peperiksaan ini mengandungi **TUJUH** muka surat beserta Lampiran **TIGA** muka surat bercetak sebelum anda memulakan peperiksaan ini.

Kertas soalan ini mengandungi **EMPAT** soalan.

Jawab **TIGA** soalan.

Mulakan jawapan anda untuk setiap soalan pada muka surat yang baru.

Agihan markah bagi setiap soalan diberikan di sudut sebelah kanan soalan berkenaan.

Jawab semua soalan dalam Bahasa Malaysia atau Bahasa Inggeris atau kombinasi kedua-duanya.

[Sekiranya terdapat sebarang percanggahan pada soalan peperiksaan, versi Bahasa Inggeris hendaklah diguna pakai].

"In the event of any discrepancies, the English version shall be used".

1. Anda dikehendaki untuk merekabentuk robot pendaki tali untuk satu pertandingan. Jarak perjalanan yang diperlukan bagi robot ialah 177" seperti ditunjukkan dalam Rajah 1. Jumlah berat robot keseluruhan ialah 1 kg. Sasaran masa untuk menamatkan pendakian ialah 4 saat.

You are required to design a rope climbing robot for a competition. The distance that the robot has to travel is 177" as shown in Figure 1. The total weight of the robot is 1 kg. The target time to complete the climb is 4 seconds.

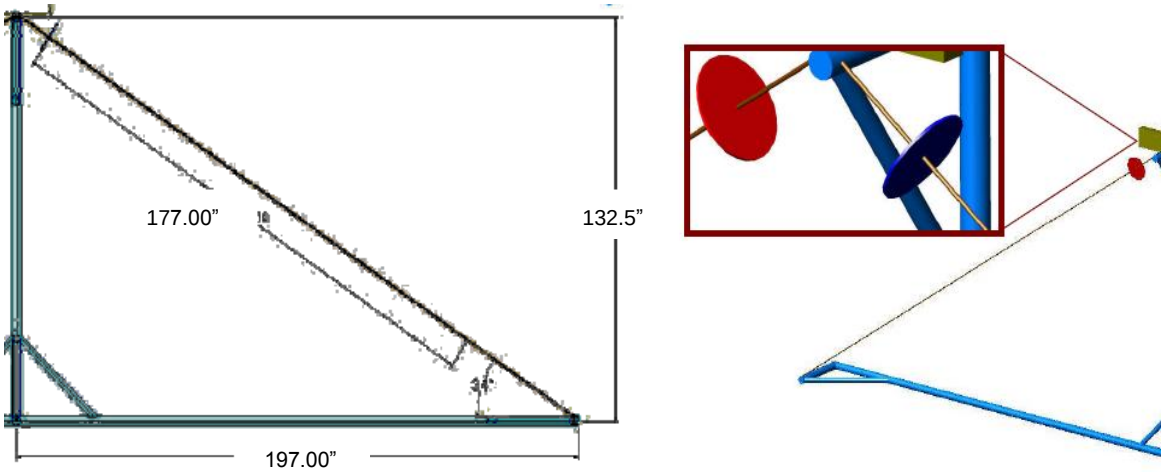
Syarat Pertandingan:
Competition Tasks:

Syarat 1 - Robot mesti mendaki tali sejauh 4.5 meters (177").

Rule 1 - The robot shall climb the rope for 4.5 meters (177").

Syarat 2 - Robot mesti berhenti mendaki apabila menyentuh plat bulat.

Rule 2 - The robot shall identify and stop the climbing actions when the robot touches the round plate.



Rajah 1
Figure 1

Bincangkan secara lengkap isu-isu berikut:-
Discuss in detail the following issues:-

(i) Analisa statik dan dinamik.
Static and dynamic analyse.

(20 markah/*marks*)

(ii) Mekanisma dan rekabentuk robot.
Mechanism and robot design.

(30 markah/*marks*)

(iii) Motor, penderia, mikropengawal, skematik lengkap dan kod program.
Motors*, sensors, microcontroller, detailed schematic and program code.*

(50 markah/*marks*)

2. (a) Terangkan penyediaan satu pin mikropengawal PIC sebagai masukan, keluaran dan 'tarik-naik lemah'. Lukis gambarajah blok bagi pin tersebut.

Explain the setting up of a pin of a PIC microcontroller as input, output and 'weak pull-up'. Draw the block diagram of the pin.

(20 markah/*marks*)

(b) Lukiskan litar antaramuka mikropengawal bagi 4 peranti masukan dan 6 peranti keluaran.

Draw the microcontroller interface circuits for 4 input devices and 6 output devices.

(20 markah/*marks*)

(c) Terangkan sebab-sebab mikropengawal digunakan untuk rekabentuk sistem mekatronik.

Explain the reasons for using microcontroller for mechatronic system design.

(20 markah/*marks*)

(d) (i) Lukis skematik untuk mengawal satu servo menggunakan mikropengawal, PIC16F84.

Draw the schematic to control a servo using microcontroller, PIC16F84.

(5 markah/*marks*)

(ii) Tulis satu program untuk mengawal servo tersebut dengan 40% kitar tugas menggunakan bahasa perhimpunan.

Write a program to control the servo with 40% duty cycle using assembly language.

(20 markah/*marks*)

(iii) Tulis satu program untuk mengawal servo tersebut dengan 40% kitar tugas menggunakan bahasa PIC Basic Pro.

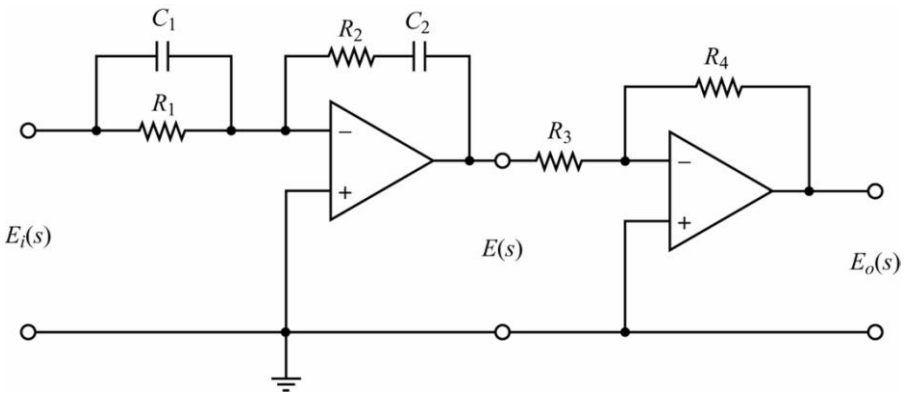
Write a program to control the servo with 40% duty cycle using PIC Basic Pro language.

(15 markah/*marks*)

3. (a) Pertimbangkan pengawal PID elektronik dalam Rajah 2. Tentukan nilai-nilai R_1, R_2, R_3, R_4, C_1 dan C_2 pengawal tersebut apabila rangkap pindah $G_c(s)=\frac{E_o(s)}{E_i(s)}$ adalah bersamaan dengan

Consider the electronic PID controller shown in Figure 2. Determine the values of R_1, R_2, R_3, R_4, C_1 and C_2 of the controller such that the transfer function $G_c(s)=\frac{E_o(s)}{E_i(s)}$ *is*

$$G_c(s) = 39.42\left(1 + \frac{1}{3.077s} + 0.7692s\right)$$
$$= 30.3215\frac{(s+0.65)^2}{s}$$



Rajah 2
Figure 2

(60 markah/*marks*)

(b) Terangkan dengan jelas langkah kaedah untuk
Explain the step by step method of

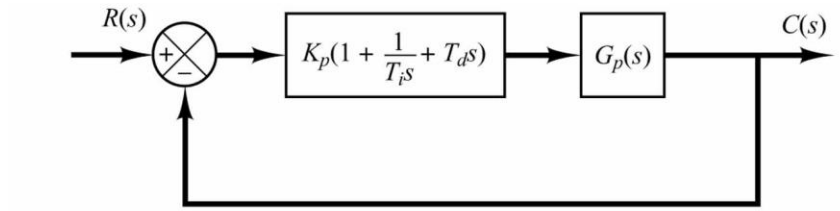
(i) Talaan PID gelung-buka Ziegler - Nichols
Ziegler – Nichols open-loop PID tuning

(ii) Talaan PID Cohen - Coon PID
Cohen – Coon PID tuning

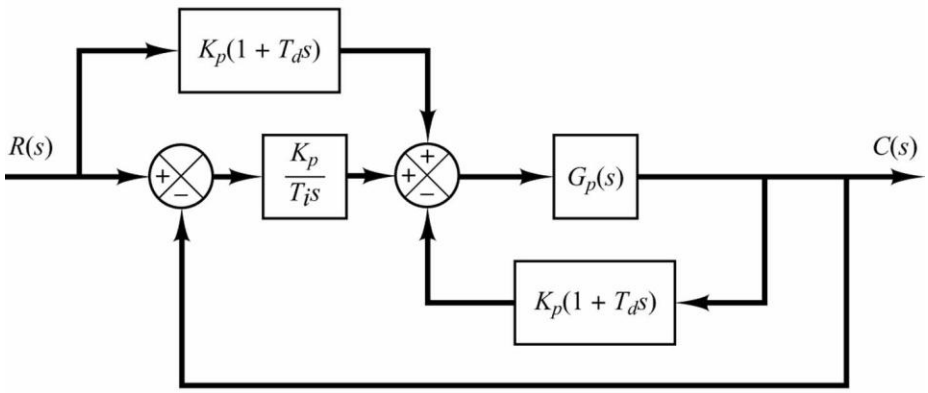
(40 markah/*marks*)

4. (a) Tunjukkan bahawa sistem terkawal PID yang ditunjukkan dalam Rajah 3(a) adalah bersamaan dengan sistem I-PD dikawal dengan kawalan suap depan yang ditunjukkan dalam Rajah 3(b).

Show that the PID-controlled system shown in Figure 3(a) is equivalent to the I-PD-controlled system with feedforward control shown in Figure 3(b).



(a)



(b)

Rajah 3
Figure 3

(50 markah/*marks*)

…7/-

(b) Dengan menggunakan penguat kendalian, tentukan rekabentuk untuk pengawal PID, $G_c(s) = \frac{(s+55.92)(s+0.5)}{s}$. Anda dibekalkan dengan pemuat-pemuat bernilai 0.1 µF dan 5.59 µF, beserta beberapa perintang.

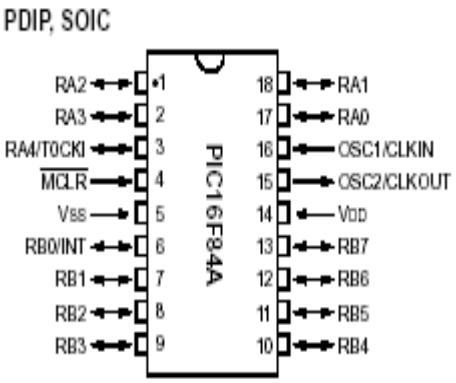
Design the PID controller of $G_c(s) = \frac{(s+55.92)(s+0.5)}{s}$ *by using operational amplifiers. You are provided with 0.1* µF *and 5.59* µF *capacitors, with several resistors.*

(50 markah/*marks*)

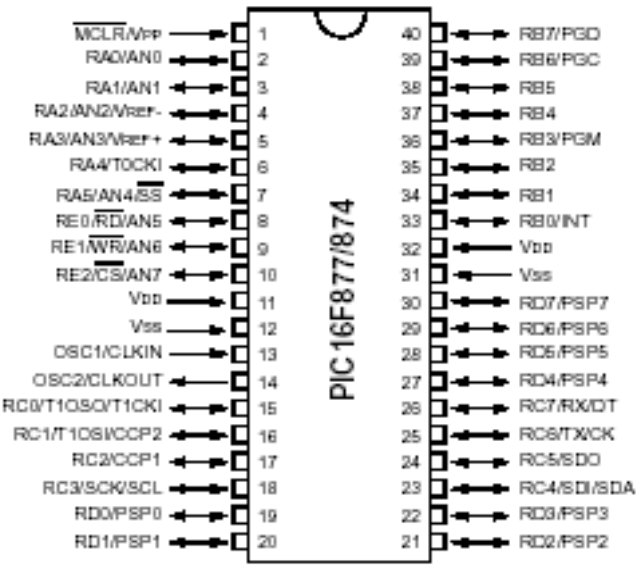
-oooOooo-

Lampiran 1

Appendix 1



PIC16F84



PIC16F877

Lampiran 2

Appendix 2

Table 7.5 PicBasic Pro statement summary

Statement	Description
@ assembly statement	Insert one line of assembly language code
ADCIN channel, var	Read the on-chip analog to digital converter (if there is one)
ASM . . . ENDASM	Insert an assembly language code section consisting of one or more statements
BRANCH index, [label1{, label2, . . .}]	Computed goto that jumps to a label based on index
BRANCHL index, [label1{, label2, . . .}]	Branch to a label that can be outside of the current page of code memory (for PICs with more than 2 k of program ROM)
BUTTON pin, down_state, auto_repeat_delay, auto_repeat_rate, countdown_variable, action_state, label	Read the state of a pin and perform debounce (by use of a delay) and autorepeat (if used within a loop)
CALL assembly_label	Call an assembly language subroutine
CLEAR	Zero all variables
CLEARWDT	Clear the watch-dog timer
COUNT pin, period, var	Count the number of pulses occurring on a pin during a period
DATA {@ location,} constant1{, constant2, . . .}	Define initial contents of the on-chip EEPROM (same as the EEPROM statement)
DEBUG item1{, item2, . . .}	Asynchronous serial output to a pin at a fixed baud rate
DEBUGIN {timeout, label,} [item1{,{item2, . . .}]	Asynchronous serial input from a pin at a fixed baud rate
DISABLE	Disable ON INTERRUPT and ON DEBUG processing
DISABLE DEBUG	Disable ON DEBUG processing
DISABLE INTERRUPT	Disable ON INTERRUPT processing
DTMFOUT pin, {on_ms, off_ms,} [tone1{, tone2, . . .}]	Produce touch tones on a pin
{EEPROM {@ location,} constant1{, constant2, . . .}}	Define initial contents of on-chip EEPROM (same as the DATA statement)
ENABLE	Enable ON INTERRUPT and ON DEBUG processing
ENABLE DEBUG	Enable ON DEBUG processing
ENABLE INTERRUPT	Enable ON INTERRUPT processing
END	Stop execution and enter low power mode
FOR count = start TO end {STEP {-} inc} {body statements} NEXT {count}	Repeatedly execute statements as count goes from start to end in fixed increment
FREQOUT pin, on_ms, freq1{, freq2}	Produce up to two frequencies on a pin
GOSUB label	Call a PicBasic subroutine at the specified label
GOTO label	Continue execution at the specified label
HIGH pin	Make pin output high
HSERIN {parity_label,} {time_out, label,} [item1{, item2, . . .}]	Hardware asynchronous serial input (if there is a hardware serial port)
HSEROUT [item1{, item2, . . .}]	Hardware asynchronous serial output (if there is a hardware serial port)
I2CREAD data_pin, clock_pin, control,{ address,} [var1{, var2, . . .}]{, label}	Read bytes from an external I^2C serial EEPROM device
I2CWRITE data_pin, clock_pin, control,{ address,} [var1{, var2, . . .}]{, label}	Write bytes to an external I^2C serial EEPROM device
IF log_comp THEN label	Conditionally jump to a label
IF log_comp THEN true_statements ELSE false_statements ENDIF	Conditional execution of statements
INPUT pin	Make pin an input
LCDIN {address,} [var1{, var2, . . .}]	Read RAM on a liquid crystal display (LCD)
LCDOUT item1{, item2, . . .}	Display characters on LCD
{LET} var = value	Assignment statement (assigns a value to a variable)

Lampiran 3

Appendix 3

Statement	Description
LOOKDOWN value, [const1{, const2, . . .}], var	Search constant table for a value
LOOKDOWN2 value, {test} [value1{, value2, . . .}], var	Search constant/variable table for a value
LOOKUP index, [const1{, const2, . . .}], var	Fetch constant value from a table
LOOKUP2 index, [value1{, value2, . . .}], var	Fetch constant/variable value from a table
LOW pin	Make pin output low
NAP period	Power down processor for a selected period of time
ON DEBUG GOTO label	Execute PicBasic debug subroutine at label after every statement if debug is enabled
ON INTERRUPT GOTO label	Execute PicBasic subroutine at label when an interrupt is detected
OUTPUT pin	Make pin an output
PAUSE period	Delay a given number of milliseconds
PAUSEUS period	Delay a given number of microseconds
{PEEK address, var}	Read byte from a register
{POKE address, var}	Write byte to a register
POT pin, scale, var	Read resistance of a potentiometer, or other variable resistance device, connected to a pin with a series capacitor to ground
PULSIN pin, state, var	Measure the width of a pulse on a pin
PULSOUT pin, period	Generate a pulse on a pin
PWM pin, duty, cycles	Output a pulse width modulated (PWM) pulse train to pin
RANDOM var	Generate a pseudo-random number
RCTIME pin, state, var	Measure pulse width on a pin
READ address, var	Read a byte from on-chip EEPROM
READCODE address, var	Read a word from code memory
RESUME {label}	Continue execution after interrupt handling
RETURN	Continue execution at the statement following last executed GOSUB
REVERSE pin	Make output pin an input or an input pin an output
SERIN pin, mode,{ timeout, label,} {[qual1, qual2, . . .],}{ item1{, item2, . . .}}	Asynchronous serial input (Basic Stamp 1 style)
SERIN2 data_pin{\flow_pin}, mode, {parity_label,} {timeout, label,} [item1{, item2, . . .}]	Asynchronous serial input (Basic Stamp 2 style)
SEROUT pin, mode, [item1{, item2, . . .}]	Asynchronous serial output (Basic Stamp 1 style)
SEROUT2 data_pin{\flow_pin}, mode, {pace,} {timeout, label,} [item1{, item2, . . .}]	Asynchronous serial output (Basic Stamp 2 style)
SHIFTIN data_pin, clock_pin, mode, [var1{\bits1} {, var2{\bits2}, . . .}]	Synchronous serial input
SHIFTOUT data_pin, clock_pin, mode, [var1{\bits1} {, var2{\bits2}, . . .}]	Synchronous serial output
SLEEP period	Power down the processor for a given number of seconds
SOUND pin, [note1, duration1{, note2, duration2, . . .}]	Generate a tone or white noise on a specified pin
STOP	Stop program execution
SWAP var1, var2	Exchange the values of two variables
TOGGLE pin	Change the state of an output pin
WHILE logical_comp statements WEND	Execute code while condition is true
WRITE address, value	Write a byte to on-chip EEPROM
WRITECODE address, value	Write a word to code memory
XIN data_pin, zero_pin, {timeout, label,} [var1{, var2, . . .}]]	Receive data from an external X-10 type device
XOUT data_pin, zero_pin, [house_code1\key_code1{\repeat1}{, house_code2\key_code2{\repeat2, . . .}]	Send data to an external X-10 type device

PicBasic Pro commands