

Name **SOLUTION**

CM Section

ME430 - Mechatronics

Examination I

December 17th, 2019

Problem	Score	
1	/ 20	
2	/ 18	
3	/ 15	
4	/ 15	
5	/ 12	
6	code	check off
	/ 18	/2
Total	/100	

You may use only:

- Your computer
- Any paper notes (such as notes on the videos) you brought to the exam, so long as those notes were written by you or your lab partner.
- Any electronic notes or code residing on your local (C:) hard drive, so long as those notes/code were written by you or your lab partner.
- The course website. (This is the only approved use of the internet for this exam.)
- A calculator (optional).
- Moodle for code submission.

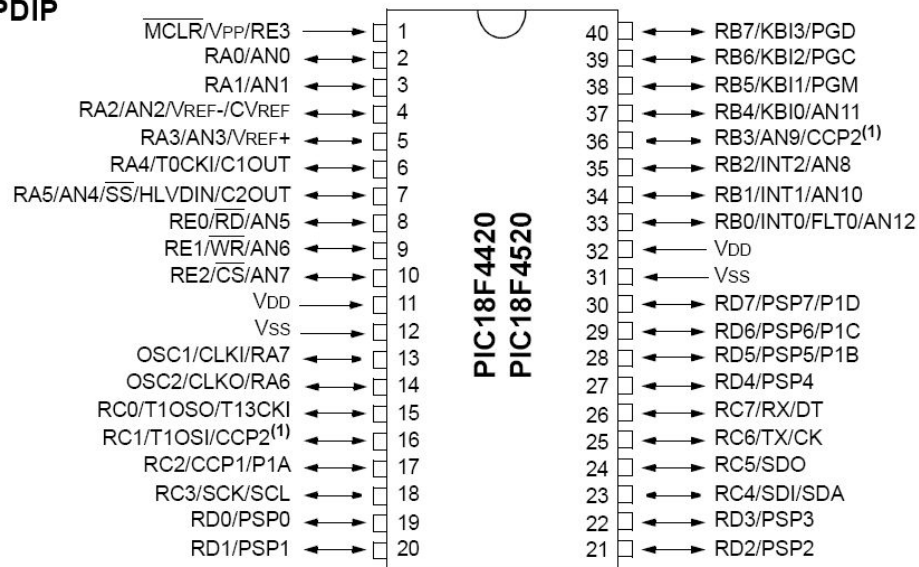
Anything not specifically allowed is prohibited. In particular, you may not use notes or code written by someone outside your lab group.

In this exam MPLAB may only be used for the final problem!

Reference

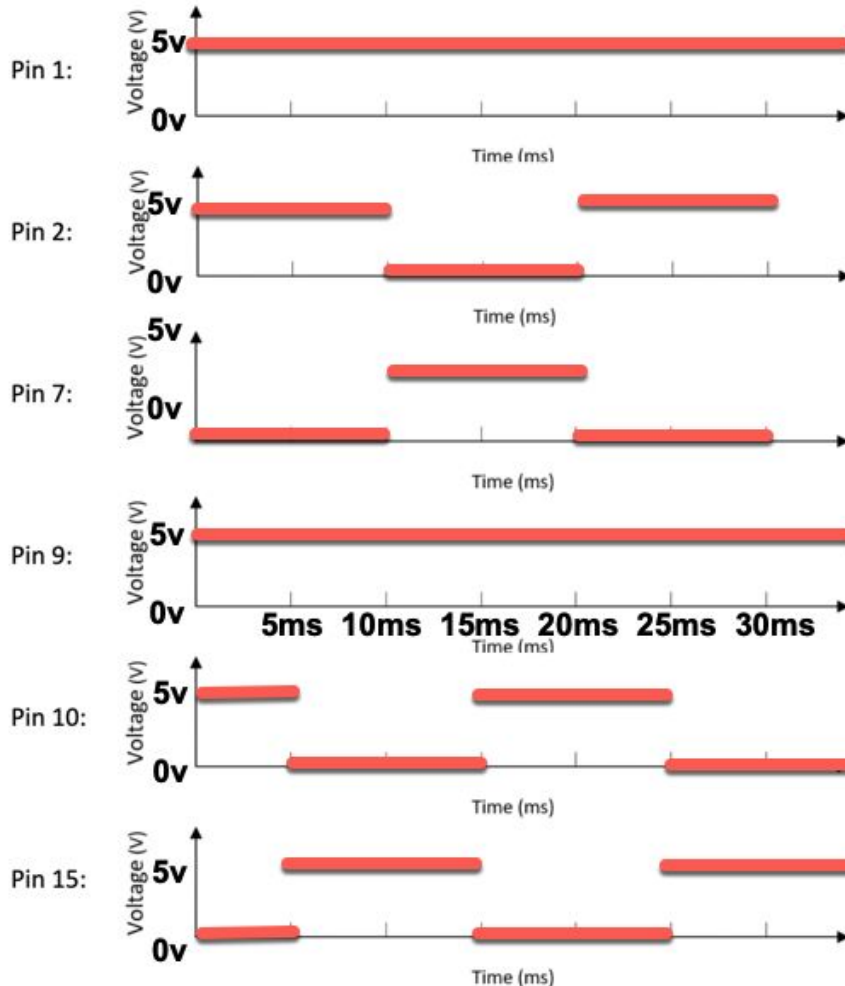
This diagram can also be found on the course website. It is copied here as a reference.

40-pin PDIP

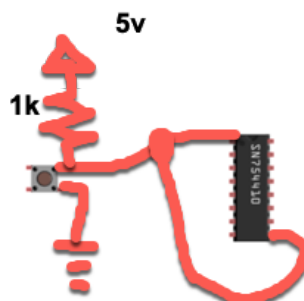


Problem 1 – Stepper Motor Control

You would like to drive a stepper motor using an SN754410 H-bridge chip. This is the same motor driver chip that you used in Lab 2. The stepper motor has a stride angle of 1.8° (200 steps per revolution) and you would like it to run at 1 revolution/second. Sketch the applied voltage on each of the chip pins that would make the motor take **six steps** at the desired speed. Make sure to put appropriate values on all axes.



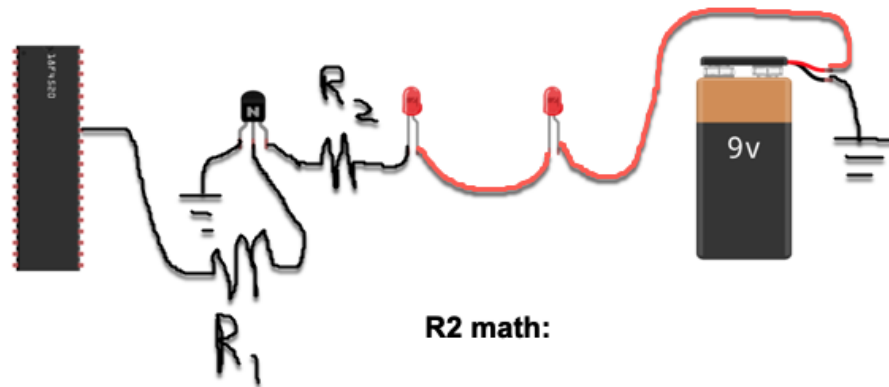
Suppose you are using the SN754410 chip to continuously run your stepper motor. Draw a circuit diagram below that uses a single switch to stop rotation of the motor *when the switch is pressed*. Draw any necessary connections between your switch circuit and the chip. You do not need to include any other connections to the chip.



Problem 2 – Lighting LEDs

You would like to use the PIC microcontroller to turn on two identical LEDs at the same time in series. Draw the connections between the components shown below so that both LEDs can be turned on using a digital output from the PIC on pin RB0 and a single BJT (type NPN). You want these LEDs to be very bright, so run them at their maximum rated current of 50mA using a 9V battery power supply. These LEDs each have a forward voltage drop of 1.9V. Add resistor symbols wherever a resistor is needed. For each resistor that you add, indicate its value using sizes available in the E24 series. Show all of the work for your resistor calculations.

Don't worry about any PIC connections other than the connection to pin RB0.



R1 math:
 $V = iR$
 $5 = (.050 / 10) R1$
 $R1 = 5/.005$
 $R1 = 1k$

**which is in the E24
 so we're done with R1**

R2 math:

$$V = iR$$

$$9 - 1.9 - 1.9 = 0.050 * R2$$

$$R2 = 5.2 / 0.05$$

$$R2 = 104 \text{ ohms}$$

R2 real:
 $R2 = 100 \text{ ohms}$

Note R2 real could be 110 ohms or 100 ohms. Don't mind either. 104 is closer, but 110 is probably better since it will stay below the limit.³²

Problem 3 – Number Systems

Make the conversions necessary to fill in the following table given the declaration of the variable 'ME' in the first column.

Variable Declaration	Decimal	Binary	Hex	ASCII
char ME = 33;		0b 0010 0001	0x21	'!'
char ME = -17;		0b 1110 1111	0xEF	
char ME = 0b1100 1010;	-54		0xCA	
char ME = 0b01000000 + 90;	-102	0b 1001 1010	0x9A	
unsigned char ME = 0xF2;	242	0b 1111 0010		

Note: the spaces in the binary were added to make it more readable, but in code don't leave spaces in the binary.

Problem 4 – Syntax Debugging

Consider the code shown below. This code contains **many** syntax errors and won't compile. Circle each mistake and indicate how it should be fixed. Modify only issues that will cause the code not to run--do not change the functionality.

```
#include <stdio.h>

#define MASS = 500
#define LIMIT = 100

char force = 10;
char time_array = {1, 3, 5, 7};
int i;

void main(void) {
    for (i = 7, i < MASS, i++) {
        if (i * 10 < LIMIT) {
            fprintf("Result = %d\n", time_array[i % 4])
        }
    }
    while (1) {
    }
```

```
#include <stdio.h>
```

```
#define MASS = 500
#define LIMIT = 100
```

Omit the = sign

```
char force = 10;
char time_array = {1, 3, 5, 7};
int i;
```

time_array[]

```
void main(void) {
    for (i = 7, i < MASS, i++) {
        if (i * 10 < LIMIT) {
            fprintf("Result = %d\n", time_array[i % 4])
        }
    }
    while (1) {
    }
```

missing semicolon

printf

Missing closing bracket

If the syntax errors in this code were fixed, what would the printed output in the UART window be?

Output:

```
Result = 7
Result = 1
Result = 3
```


Problem 5 – C programming

Write a small C program below to display even numbers on the LEDs from 0 to 14 **using a loop**. and **then stop** (14 will still be showing when you stop). Use a `Delay10KTCYx(50)` ; between each step. *Hint: **PORTC = 5**; would turn on RC0 and RC2 (others off) and would be a valid way to show the value 5 on the LEDs, i.e. you don't have to say `PORTC = 0b0101`, simply `PORTC = 5` is the exact same.*

You may not use MPLABx for this problem. This is a by hand coding exercise.

You may of course look at notes and other programs, but you may not type anything into MPLABx (or any other program on your computer). Write the necessary code in the blank below.

```
#include <p18f4520.h>
#include <delays.h>

char i = 0;    // (or int is fine)

// Assume all the #pragma stuff is done correctly just not shown
void main (void) {
    ADCON1 = 0b00001111;    // Sets all the pins to digital
    // You write the code to do everything else that is needed...

    TRISC = 0x00;    // All outputs

    for (i = 0; i < 15; i = i + 2) {
        PORTC = i;
        Delay10KTCYx(50);
    }

    while(1) {
    }

}
```


Problem 6 – C Programming

Create a new project in MPLABx that uses the Simulator. Starting from template.c, create a file called “lastname_firstname_Exam1.c”.

In that file, write a program that will loop over an array and print certain totals for each item in the array. For example if you had this array:

```
#define NUM_HOUSES 8
int christmasPresents[] = {0, 10, 3, 0, 0, 7, 35, 0};
```

Then you would print:

```
Ho! Ho! Ho!
Houses: 1  Coal: 1  Presents: 0
Ho! Ho! Ho!
Houses: 2  Coal: 1  Presents: 10
Houses: 3  Coal: 1  Presents: 13
Houses: 4  Coal: 2  Presents: 13
Houses: 5  Coal: 3  Presents: 13
Ho! Ho! Ho!
Houses: 6  Coal: 3  Presents: 20
Houses: 7  Coal: 3  Presents: 55
Houses: 8  Coal: 4  Presents: 55
```

Requirements:

Santa tracks the number of houses, number of houses that received no presents (got coal), and the total number of presents delivered. After each house is visited a summary line is printed as shown above. Additionally any time that the **total** number of presents delivered is **even** Santa lets out a “Ho! Ho! Ho!”. His “Ho! Ho! Ho!” is printed before the gift summary line. For example, notice that before the 0, 10, and 20, which are all even, there is a “Ho! Ho! Ho!” printed.

(continued on next page for important details)

For your demo, to check off your solution, don't use the previously shown array, since it's output is known (you can use it for testing only). Instead use the array below for your final code / demo:

```
#define NUM_HOUSES 9  
int christmasPresents[] = {5, 5, 30, 0, 0, 7, 0, 7, 1000};
```

The format of your output in the UART1 window should follow the example output exactly. Recall that you learned how to **enable the UART1 window** in the day 4 lecture.

You are required to use **a for loop** to solve this problem and will receive no credit for hard-coded solutions. Assume that we could later test your code with a different array, so make sure that your program would work for other array values and NUM_HOUSES lengths (note, the two will always match as appropriate).

When you complete this problem and have it running, call your instructor over to check it off on the front sheet.

Regardless of whether you finish or not, submit your code into the Moodle dropbox (just the lastname_firstname.c file).

```
#include <stdio.h>

//#define NUM_HOUSES 8
//int christmasPresents[] = {0, 10, 3, 0, 0, 7, 35, 0};

#define NUM_HOUSES 9
int christmasPresents[] = {5, 5, 30, 0, 0, 7, 0, 7, 1000};

int totalPresents = 0;
int coalHouses = 0;

void main(void) {
    int i = 0;
    for (i = 0; i < NUM_HOUSES; i++) {
        int presents = christmasPresents[i];
        if (presents == 0) {
            coalHouses++;
        }
        totalPresents += presents;
        if (totalPresents % 2 == 0) {
            printf("Ho! Ho! Ho!\n");
        }
        printf("Houses: %d Coal: %d Presents: %d\n", i+1, coalHouses,
totalPresents);
    }

    while (1) {
        // This area loops forever
    }
}
```