Name: Dan Kelly

Dordt University Engineering Department

EGR 304, Microprocessor Interfacing — Final Exam, May 4, 2017.
Open book, open notes, take-home.
An Internet-connected computer is required. A calculator is also required.

35/100 = C-

1.) A small local-area computer network (e.g. a network of two or three computers sharing files, printers, etc.—no Internet connection) could be made using USB connections or using Ethernet connections between the computers. Describe some of the challenges and advantages that would be encountered when using a USB connection for networking as opposed to using an Ethernet connection. How are these challenges rooted in the technological differences of the two connection types? (10 points)

7

One of the advantages of having a USB would be that there is going to be less expense when installing a USB connection. You will also have more USB connections as well within your LAN. The challenge with USB is that Ethernet is capability to talk back and forth between the devices on the network whereas USB only goes one way, which would be an advantage to USB as there wouldn't be any types of collisions. The connection speeds of the Ethernet are greater than that of the USB. Another challenge for USB would be that the LAN would have to be very close together as many USB cables aren't as long as Ethernet and wouldn't be able to get to the devices. The challenges are rooted within the differences because they aren't able to make USB as long due to the amount of propagation delay. The signal won't be able to reach the host properly. This is also partly why having Ethernet is better because you are able to transfer more data quicker.

Did not discuss host-client for USB vs. peer-peer for Ethernet

2.) A thirty-foot long asynchronous serial computer interface uses RS-232-style signaling but at "TTL" logic levels meaning 0 volts for logic-1 and idle and mark or 5 volts for logic-0 or space, nominal levels. The interface is in practically continuous use. (There are no significant periods of idle time.) About half of the time the transmission line is sending logic-1 and the other half of the time it is sending logic-0. A low-impedance driver is connected to a 50 Ω transmission line and a 50 Ω matched load is used to prevent reflections. It works fine but battery life is poor due to all the current flowing in the load. Suggest a change or changes to the source and/or load impedances that will improve battery life. (10 points)

1

One change that I would suggest would be to decrease the resistance of the matched load. This will result in having a higher current within the battery allowing it to have better battery life. A decrease in the resistance of the low-impedance driver will also result in a better battery life.

X

3.) A DC motor with a permanent-magnet field, when connected to a 12 V DC source has a no-load speed of 10 000 RPM and a stall (locked rotor) torque (a.k.a. locked rotor torque) of 0.1 Nm. Assume the motor has an ideal linear torque vs. speed characteristic. This motor is connected to a low-impedance 12 V source on the electrical side and to a mechanical load that requires a driving torque that is related to the speed such that

$$T = \sqrt{S}\left(\frac{1}{2500}\ \text{Nm/RPM}^{0.5}\right)$$

Where T = required driving torque (in Nm)
S = load speed (in RPM)

8

Find the speed at which the motor will operate. (10 points)

Linear Torque vs speed equation: $T = 0.1\ \text{Nm} - \left(\frac{0.1\ \text{Nm}}{10{,}000\ \text{RPM}}\right)S$

set them equal

$$0.1\ \text{Nm} - \left(\frac{0.1\ \text{Nm}}{10{,}000\ \text{RPM}}\right)S = \sqrt{S}\left(\frac{1}{2500}\ \text{Nm}/10{,}000^{0.5}\right)$$

$$\left(0.1\ \text{Nm} - \left(\frac{0.1\ \text{Nm}}{10{,}000\ \text{RPM}}\right)S\right)^2 = \left(\sqrt{S}\,(.000004)\right)^2$$

Square them to get rid of $\sqrt{S}$

$$= 1.6\times10^{-11}S$$

$(0.1 - 0.0000015)(0.1 - 0.0000015)$

$0.01 - 0.000000 15 - 0.000000 15 + 1\times10^{-10}S^2$

$$\hookrightarrow 0.01 - 0.0000025 + 1\times10^{-10}S^2 = 1.6\times10^{-11}S$$?

$$0.01 - 2\times10^{-6}S + 1\times10^{-10}S^2 = 0$$ solve for S

$S = 10{,}000$ RPM X

Bad math

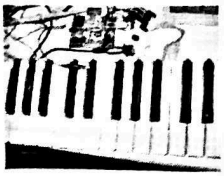
4.) Describe in words or pseudo-code how to use an interrupt driven tic-clock system with a microcontroller to generate a pulse of exactly 10 ms each time a certain subroutine is called. Assume the tic-clock interrupt has a frequency of 10 kHz. (10 points)

When the subroutine is called, you will want to have an interrupt be called causing the program to go to that different method. Within this method, you will have a counter that is going to be measuring every 10ms. The measure of this will be from the top of the rising edge to the bottom of the falling edge to ensure there is no delay. Once completed the program will go back to the state it was in before the subroutine was called. X

1

You must use a task scheduler in order to take advantage of a tic-clock.

5.) Consider creating a keyboard-operated music synthesizer by connecting a keyboard to an Arduino microcontroller. Let the synthesizer have 24 keys (two octaves). Each key will have just one SPST switch. The keyboard has a 48-conductor ribbon cable connected to it, two conductors dedicated to each switch. Could this keyboard be connected directly to an Arduino microcontroller board using no additional parts such as diodes, gates, resistors, or transistors? The Arduino board has 13 digital I/O pins available. If so, describe how and describe any limitations of the connection. If not, give a rationale for your conclusion. (10 points)



I don't think that this is possible without extra parts due to the number of pins that the Arduino board has as well as to not let too much current through the keyboards. With the different keys you might need to use some resistors to make sure too much current isn't being used which would cause things to short out after longer uses. In addition, there are only 13 pins on the Arduino board and 24 keys total. You would be able to use the one pin for grounding all of them but to hook up 24 keys to 12 pins you would have to find a way to split each of the pins which would possibly need the use of a transistor to help get all of the keys working. There would be a lot of current flowing through the whole system as well and would need to find ways of reducing that to ensure the circuit is running properly without any shorts. So I don't think it is possible without any extra parts. X

6.) What is the difference between vectored interrupts and polled interrupts. (10 points)

A vectored interrupt is when the interrupt is called upon from a certain address. The CPU uses the address of the interrupt from memory and executes it. While with polling interrupts, they are letting the CPU know that it is ready to be executed. It will be read until the status of the device changes and then that interrupt is executed. The main difference is that vectored interrupts are called upon whenever it needs to be acknowledged and is more immediate. The polled interrupt will only be executed when there it has a different indication then when last checked and isn't as urgent.

2

Incorrect conception of how interrupts operate.

7.) A microcontroller has three sources of interrupts. Any of these can be eliminated by adding additional hardware. To keep costs down it is desirable to have the microcontroller handle as many of the interrupts as possible. Can the microprocessor handle all the interrupts below? If not, suggest which one (or ones) should be eliminated (requiring additional hardware). Try to eliminate as few interrupts as possible. Justify your answer. (15 points)

Listed from highest priority (on top) to lowest (on bottom)

Source 1) The interrupt service routine (ISR) takes 50 μs to run. Minimum time between interrupts is 100 μs. Prompt response to the interrupt request is not needed as long as the interrupt is serviced before it is requested again.

Source 2) ISR takes 10 μs. Minimum time between interrupts is 500 μs but the maximum allowable latency (time from request to end of execution of ISR) is 100 μs.

Source 3) ISR takes 5 μs. Minimum time between interrupts is 70 μs. Prompt response is not needed as long as the interrupt is serviced before it is requested again.

The longest critical region in the main software is 8 μs and the interrupts are disabled during each ISR. The longest instruction takes 1 μs.

(Please write your answer on the next page.)

(This page is provided for the answer to the problem stated on the previous page.)

Summary of the question: Will interrupts work? Eliminate as few as possible if any need to be eliminated. Justify your answer.

Source 1) ISR takes 50 µs (T_i) to run, 100 µs (T_p) between interrupts.
Source 2) ISR takes 10 µs to run, 500 µs between interrupts, latency < 100 µs req'd.
Source 3) ISR takes 5 µs to run, 70 µs between interrupts
Critical region is 8 µs, interrupts disabled during ISRs, longest instruction is 1 µs.

$\frac{50}{100} + \frac{10}{500} + \frac{5}{70} = .59 < 1.0$ ✓
.5 .02 .07

$N(i,x) = ceil((T_{pi} - T_i)/T_{px})$

$N(1,1) = 1$ $N(2,1) = 5$ ✗ $N(3,1) = 1$
$N(2,2) = 1$ $N(3,2) = 1$
$N(3,3) = 1$

Source 1
$T_{i1} + N(1,1)T_1 < T_{p1}$
50 + 50 < 100
✗ 100 < 100

Source 2
$T_{i2} + N(2,2)T_2 + N(2,1)T_1 < T_{p2}$
50 + (1)(10) + 5(50) < 500
310 < 500 ✗

Source 3
$T_{i3} + N(3,3)T_3 + N(3,2)T_2 + N(3,1)T_1 < T_{p3}$
50 1(5) + 1(50) + 1(50) < 70
✗ 155 < 70

I would eliminate Source 3 Interrupts because it is unreliable the amount of time between does not allow it to completely run. You could possibly get rid of 1 but it is not completly neressary.

8

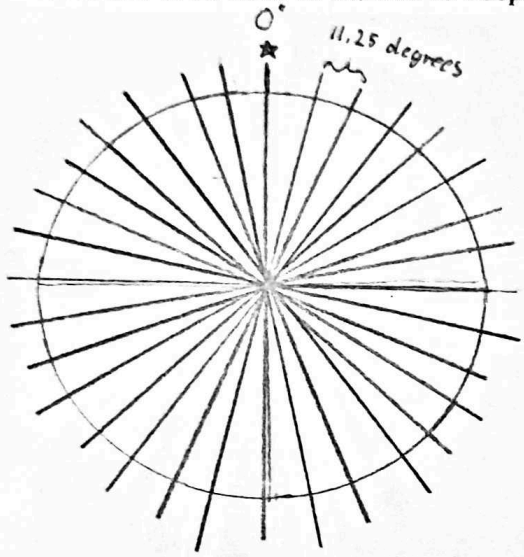
Correct conclusion, but there are substantial errors and little thought given to options to resolve the problem.

8.) Design a rotational position sensor for sensing ***absolute*** position with a resolution of 11.25 degrees (1/32 of a revolution). The rotational position sensor needs to only keep track of one revolution. If the shaft of the sensor should roll past the home position, the position code can roll over too. Each time the rotational position of a disk changes by 1/32 of a revolution, an interrupt should be produced. You may assume that the interrupt will be serviced before the disk moves enough to generate another interrupt request. You may decide for yourself how much logic to do in hardware and how much to do in the ISR. Your design should be relatively easy to manufacture. (15 points)

a.) Show the design of the disk and number and location of the sensor(s).

b.) Show a flow chart (or equivalent pseudo-code) to describe the processing that needs to be done in the ISR to keep track of ***absolute*** position.

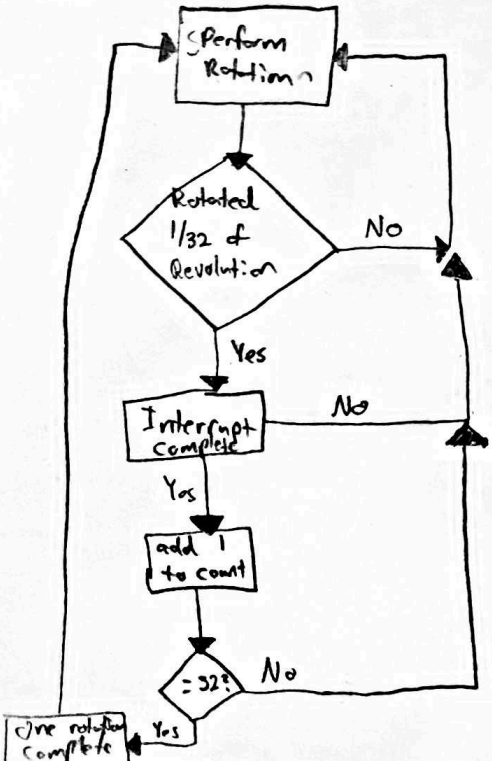
a.)



★ Location of the sensor only need 1 because it will sense when it is making on full revolution.

2

b.)



The disk will spin and once it reaches a 1/32 of a rotation the interrupt will activate and add to a count to know where it is and once it gets to 32 it will have completed one full rotation and know that it has done so. The absolute position will be based on the counted Interrupts

You have described a relative position encoder.

9. The performance of a car's shock absorber needs to be monitored to assist with the design and testing of the shock absorber. The fastest rate at which the shock absorber might effectively move to absorb the energy of bumps is 100 Hz. (100 bumps per second for example.)

The transducer used is a linear analog position sensor. It produces a voltage proportional to the length between the two anchor points of the shock absorber. The maximum voltage is +0.100 V at full extension of the shock absorber and the minimum is –0.100 V at full compression. The transducer has a frequency response up to 10000 Hz. Thus, it is not only is sensitive to the motion of the shock absorber, it is also sensitive to some of the sound of the wheel rolling over the road and conducted into the shock-absorber.

A commercially available analog-to-digital converter is in use. The sample rate of the converter is 1000 Hz. (One sample is taken every millisecond). This analog-to-digital converter has an input range of +10 V to –10 V and provides a 10-bit output in a two's complement format such that +10 V produces the code 01 1111 1111_{TC} which corresponds to $+511_{10}$ and –10 V produces the code 10 0000 0001_{TC} which corresponds to -511_{TC}. This converter has no anti-aliasing filter. The user is expected to add their own anti-aliasing filter.

The user figured that the 10000 Hz frequency response of the transducer and the 1000 Hz sample rate are 100 and 10 times (respectively) more than needed since the user is only interested in motions at 100 Hz or less. Thus, these were perceived as very good specifications. (The user may be naive in this determination.) The ±10 V input range is also perceived as 100 times greater than needed so that should also be very good. (The user may be naive in this determination.) The user is getting poor results from this system.

For an anti-aliasing filter the user has resorted to averaging 8 sequential samples (using digital addition of the sampled data, then dividing by 8) in order to get better results. This has reduced their effective sample rate to 1000/8 = 125 Hz. (Every eight digital samples get averaged to produce just one average sample for the interval.) However, the samples still are not accurate.

Assume that averaging digital samples must be abandoned as it is an incorrect method of addressing the problems here. Assume analog filters and amplifiers are available to any specifications you might like. For a filter, specify if it should pass low or high frequencies and what the cutoff frequency should be. For an amplifier, specify the gain. How can you use filters and amplifiers to improve the results? Explain your choices. (10 points)

For a filter, the frequency should be pass a lower frequency as it says that it should be less than 100Hz and we want it to be less than the 125 Hz that it states. The cut off frequency of the filter should be roughly 80Hz. When dividing by the 8 samples you will get 100Hz and that is the maximum desired. For an amplifier, the gain should be roughly 100V. The amplifier takes the output divided by the input to give you the gain. This results in 10/.1 and gives you the 100V.

Gain is dimensionless (or some say, "volts per volt" or V/V)

5

Name:_____Patrick Munsey_____

Dordt University Engineering Department

EGR 304, Microprocessor Interfacing — Final Exam, May 4, 2017.
Open book, open notes, take-home.
An Internet-connected computer is required. A calculator is also required.

38/100 = C

1.) A small local-area computer network (e.g. a network of two or three computers sharing files, printers, etc.—no Internet connection) could be made using USB connections or using Ethernet connections between the computers. Describe some of the challenges and advantages that would be encountered when using a USB connection for networking as opposed to using an Ethernet connection. How are these challenges rooted in the technological differences of the two connection types? (10 points)

1

-Within ethernet, a proper frame rate is used with results in collisions being detected by all devices on the local-area network. This is an issue within an ethernet system because frames must be long enough to flood the entire network. With this comes the issue that the network isn't idle until the entire frame has reached every device, regardless of its involvement in the system communication. Within this delay, two computers can find the system idle since the whole network has to be flooded for every device to know that it isn't idle. These collisions create garbage within receiving devices and can't be detected by the devices on the network until it reaches a computer that is transmitting where it is then detected. This creates network lag and inefficiency. For USB, enumeration streamlines communication as devices are addressed automatically to what the device is. USB connections can transfer power as well. Larger streams of data can be passed through USB ports if they are configured for bulk transactions. This versatility is also a plus for USB with isochronous, control, and interrupt driven transactions all possible. USBs have grown to increase its capacitor availability with type C connectors have twenty-four of them. This allows for more versatility within the bus's transfer capabilities.

Your answer does not address networking. Issues such as distance and peer-to-peer communication should be addressed.

2.) A thirty-foot long asynchronous serial computer interface uses RS-232-style signaling but at "TTL" logic levels meaning 0 volts for logic-1 and idle and mark or 5 volts for logic-0 or space, nominal levels. The interface is in practically continuous use. (There are no significant periods of idle time.) About half of the time the transmission line is sending logic-1 and the other half of the time it is sending logic-0. A low-impedance driver is connected to a 50 Ω transmission line and a 50 Ω matched load is used to prevent reflections. It works fine but battery life is poor due to all the current flowing in the load. Suggest a change or changes to the source and/or load impedances that will improve battery life. (10 points)

1

-A change that could be made would be lowering the load impedances to offset the higher current flowing through it. This change would reduce the voltage use of the load and maintain the battery live for a longer duration. This should be done with reflections in mind as to not damage the driver or transmission line severely.

K

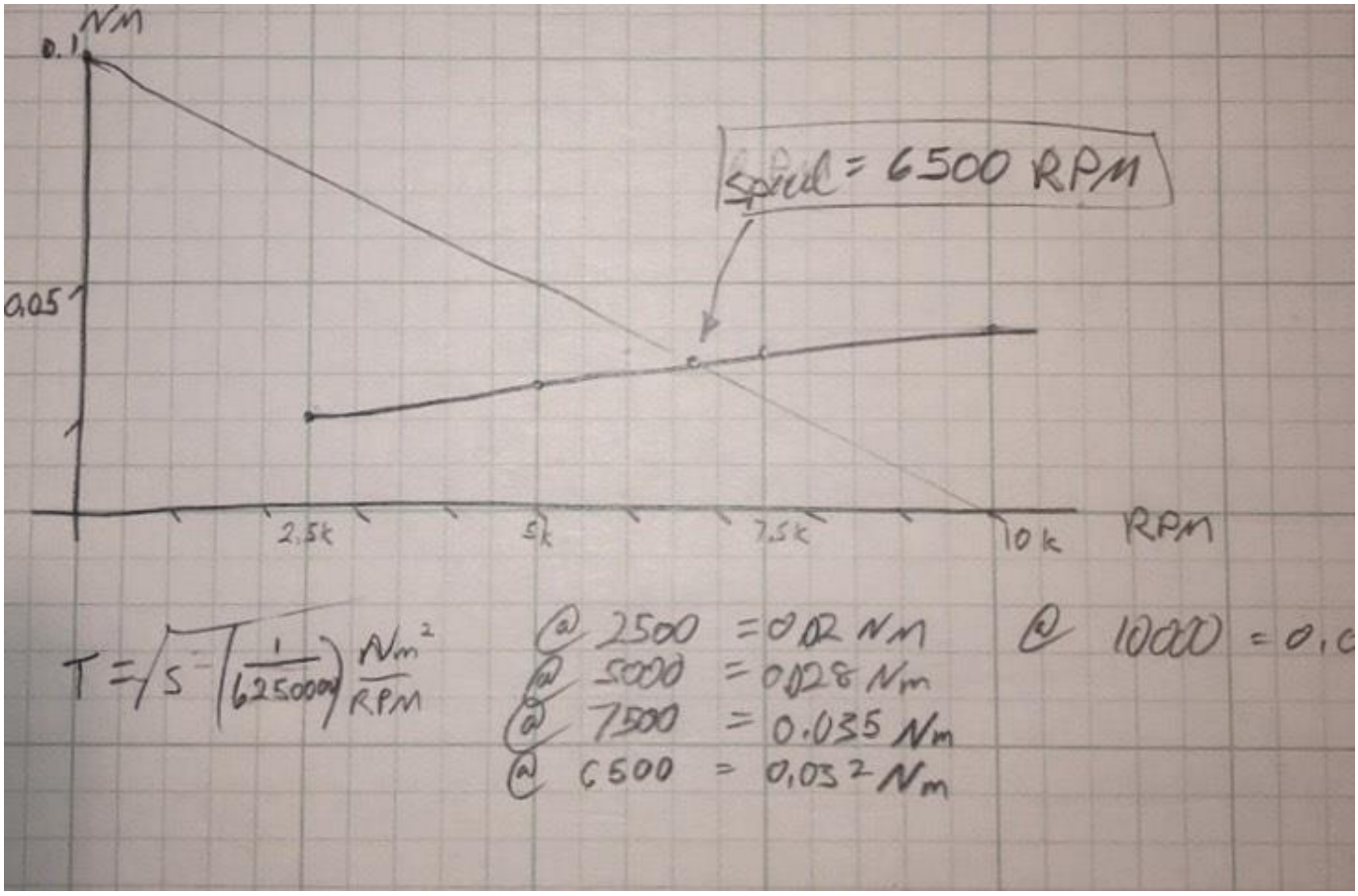
3.) A DC motor with a permanent-magnet field, when connected to a 12 V DC source has a no-load speed of 10 000 RPM and a stall (locked rotor) torque (a.k.a. locked rotor torque) of 0.1 Nm. Assume the motor has an ideal linear torque vs. speed characteristic. This motor is connected to a low-impedance 12 V source on the electrical side and to a mechanical load that requires a driving torque that is related to the speed such that

$$T = \sqrt{S}\left(\frac{1}{2500}\,\mathrm{Nm/RPM^{0.5}}\right)$$

Where T = required driving torque (in Nm)
S = load speed (in RPM)

Find the speed at which the motor will operate. (10 points)

8



Not accurate enough.
Not even accurate to the second significant figure.

4.) Describe in words or pseudo-code how to use an interrupt driven tic-clock system with a microcontroller to generate a pulse of exactly 10 ms each time a certain subroutine is called. Assume the tic-clock interrupt has a frequency of 10 kHz. (10 points)

-When that subroutine is called, the tic-clock interrupt will read it on its next poll. Once the tic-clock reads that the subroutine has been called, the code can then execute the microcontroller to start the pulse and end it exactly 10 ms later.

You must use a task scheduler to take advantage of a tic clock.

5.) Consider creating a keyboard-operated music synthesizer by connecting a keyboard to an Arduino microcontroller. Let the synthesizer have 24 keys (two octaves). Each key will have just one SPST switch. The keyboard has a 48-conductor ribbon cable connected to it, two conductors dedicated to each switch. Could this keyboard be connected directly to an Arduino microcontroller board using no additional parts such as diodes, gates, resistors, or transistors? The Arduino board has 13 digital I/O pins available. If so, describe how and describe any limitations of the connection. If not, give a rationale for your conclusion. (10 points)

-Since the output of the keyboard is a conductor ribbon, we would need a way to attach that to a breadboard. Then connecting those low conductors to ground and the high side to a common node with their octave counterpart. This would create twelve nodes that would then be connected to digital I/O pins on the Arduino. To differentiate between the higher and lower octave, a resistor will be inserted between the low octave's conductor ribbon and the common node of the note. This would result in a lower voltage value(approx. 2.5-3) going into the Arduino when the lower octave is played. Along with some clever coding, the program can use this differential in voltage with "if" statements to play the correct tone.

Digital I/O pins can only read one of two states, high and low.

6.) What is the difference between vectored interrupts and polled interrupts? (10 points)

-Vectored interrupts are like a child tapping on their parent requesting attention. They tell the computer that a request is present at the hardware level while telling the computer what device is giving the request. Polled interrupts require the computer to send a question to each device asking if it the one who needed attention. Like a parent in a car who heard a yell but now has to ask each child individually if it was them who yelled.

10

7.) A microcontroller has three sources of interrupts. Any of these can be eliminated by adding additional hardware. To keep costs down it is desirable to have the microcontroller handle as many of the interrupts as possible. Can the microprocessor handle all the interrupts below? If not, suggest which one (or ones) should be eliminated (requiring additional hardware). Try to eliminate as few interrupts as possible. Justify your answer. (15 points)

Listed from highest priority (on top) to lowest (on bottom)

Source 1) The interrupt service routine (ISR) takes 50 μs to run. Minimum time between interrupts is 100 μs. Prompt response to the interrupt request is not needed as long as the interrupt is serviced before it is requested again.

Source 2) ISR takes 10 μs. Minimum time between interrupts is 500 μs but the maximum allowable latency (time from request to end of execution of ISR) is 100 μs.

Source 3) ISR takes 5 μs. Minimum time between interrupts is 70 μs. Prompt response is not needed as long as the interrupt is serviced before it is requested again.

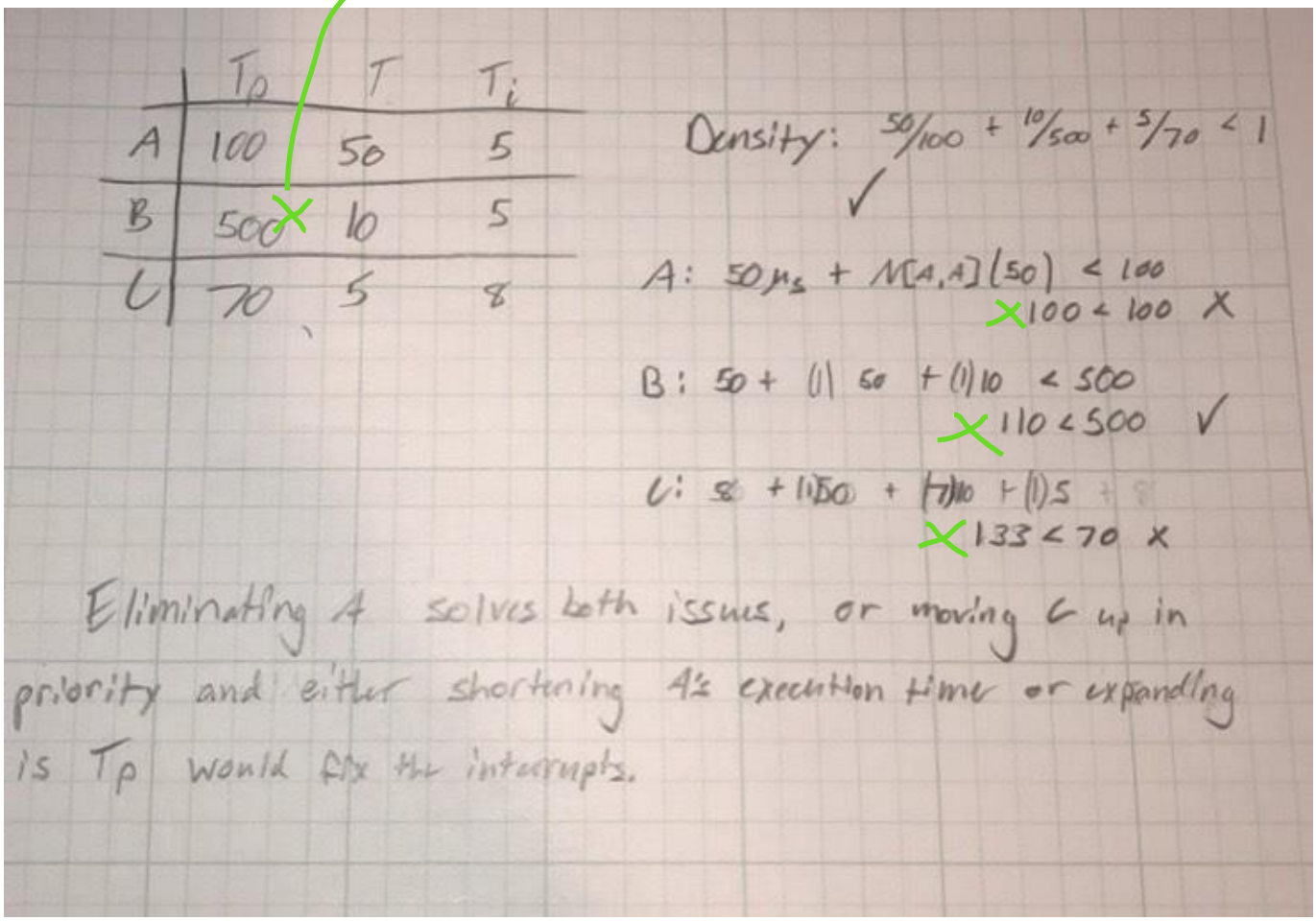
The longest critical region in the main software is 8 μs and the interrupts are disabled during each ISR. The longest instruction takes 1 μs.

(Please write your answer on the next page.)

(This page is provided for the answer to the problem stated on the previous page.)

Summary of the question: Will interrupts work? Eliminate as few as possible if any need to be eliminated. Justify your answer.

Source 1) ISR takes 50 µs to run, 100 µs between interrupts.
Source 2) ISR takes 10 µs to run, 500 µs between interrupts, latency < 100 µs req'd.
Source 3) ISR takes 5 µs to run, 70 µs between interrupts
Critical region is 8 µs, interrupts disabled during ISRs, longest instruction is 1 µs.



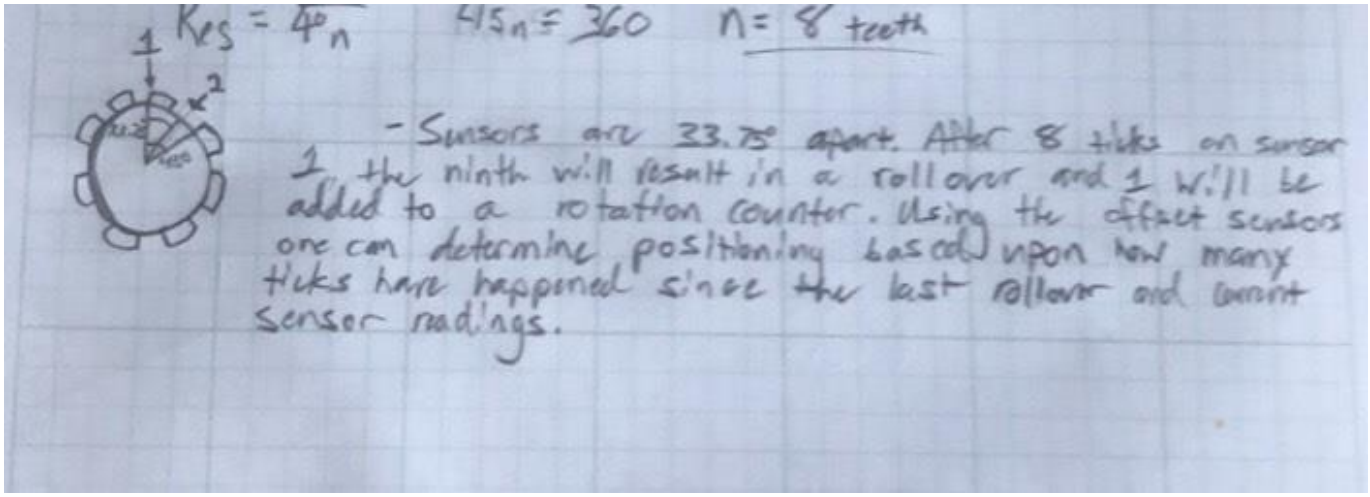
6

Very little of the math is shown, so partial credit is difficult to award.
There are errors in every calculation, but the general conclusion that it will not work as given is correct, rather by happenstance.

8.) Design a rotational position sensor for sensing ***absolute*** position with a resolution of 11.25 degrees (1/32 of a revolution). The rotational position sensor needs to only keep track of one revolution. If the shaft of the sensor should roll past the home position, the position code can roll over too. Each time the rotational position of a disk changes by 1/32 of a revolution, an interrupt should be produced. You may assume that the interrupt will be serviced before the disk moves enough to generate another interrupt request. You may decide for yourself how much logic to do in hardware and how much to do in the ISR. Your design should be relatively easy to manufacture. (15 points)

a.) Show the design of the disk and number and location of the sensor(s).

b.) Show a flow chart (or equivalent pseudo-code) to describe the processing that needs to be done in the ISR to keep track of ***absolute*** position.



You have described a relative position encoder.

3

9. The performance of a car's shock absorber needs to be monitored to assist with the design and testing of the shock absorber. The fastest rate at which the shock absorber might effectively move to absorb the energy of bumps is 100 Hz. (100 bumps per second for example.)

   The transducer used is a linear analog position sensor. It produces a voltage proportional to the length between the two anchor points of the shock absorber. The maximum voltage is +0.100 V at full extension of the shock absorber and the minimum is –0.100 V at full compression. The transducer has a frequency response up to 10000 Hz. Thus, it is not only is sensitive to the motion of the shock absorber, it is also sensitive to some of the sound of the wheel rolling over the road and conducted into the shock-absorber.

   A commercially available analog-to-digital converter is in use. The sample rate of the converter is 1000 Hz. (One sample is taken every millisecond). This analog-to-digital converter has an input range of +10 V to –10 V and provides a 10-bit output in a two's complement format such that +10 V produces the code $01\ 1111\ 1111_{TC}$ which corresponds to $+511_{10}$ and –10 V produces the code $10\ 0000\ 0001_{TC}$ which corresponds to -511_{TC}. This converter has no anti-aliasing filter. The user is expected to add their own anti-aliasing filter.

   The user figured that the 10000 Hz frequency response of the transducer and the 1000 Hz sample rate are 100 and 10 times (respectively) more than needed since the user is only interested in motions at 100 Hz or less. Thus, these were perceived as very good specifications. (The user may be naive in this determination.) The $\pm$10 V input range is also perceived as 100 times greater than needed so that should also be very good. (The user may be naive in this determination.) The user is getting poor results from this system.

   For an anti-aliasing filter the user has resorted to averaging 8 sequential samples (using digital addition of the sampled data, then dividing by 8) in order to get better results. This has reduced their effective sample rate to 1000/8 = 125 Hz. (Every eight digital samples get averaged to produce just one average sample for the interval.) However, the samples still are not accurate.

   Assume that averaging digital samples must be abandoned as it is an incorrect method of addressing the problems here. Assume analog filters and amplifiers are available to any specifications you might like. For a filter, specify if it should pass low or high frequencies and what the cutoff frequency should be. For an amplifier, specify the gain. How can you use filters and amplifiers to improve the results? Explain your choices. (10 points)

   -Filters are used to trim the fat off of incoming signals and their data. This filtering will be the most useful for the noise coming off the road. Even while taking the average of eight points, these high frequency outliers could still do a lot of damage to the user's samples. Because the road noise will be at a higher frequency than the bumps, a low pass filter should be used. Road noise is documented to be around 700-1300 Hz so a cutoff frequency of 400 Hz should be sufficient in cutting out the road noise while still keeping all of the bump energy frequency. For the amplifier, it should be a voltage gain amplifier to use more of that +/- 10V range that his analog to digital converter is utilizing. This converter not having an anti-aliasing filter is the main reason why we need to manipulate the samples with amplifiers and filters.

7

The frequency specified is acceptable, but the reasoning expressed here is poor.
You also need to specify the gain for the amplifier.

Name: Lucas Nelson

Dordt University Engineering Department

EGR 304, Microprocessor Interfacing — Final Exam, May 4, 2017.
Open book, open notes, take-home.
An Internet-connected computer is required. A calculator is also required.

63/100 = B

1.) A small local-area computer network (e.g. a network of two or three computers sharing files, printers, etc.—no Internet connection) could be made using USB connections or using Ethernet connections between the computers. Describe some of the challenges and advantages that would be encountered when using a USB connection for networking as opposed to using an Ethernet connection. How are these challenges rooted in the technological differences of the two connection types? (10 points)

1

One advantage is that USB is getting frequent updates. USB cables are getting connections that have increased speeds and power. USB has more wires for more possibilities. USB cables now offer speeds not reached by ethernet cables. Ethernet cables are very standard so odds are all devices will have the same connection making it easy to get cables. With USB, there is no guarantee you could find cables that could connect two devices or that the cable you find will work. USB allows for good behavior programing. USB allows for active connection check to see if new things are connected. USB just offers more connections to interface with than an ethernet cable.

Your answer does not address networking. Issues such as distance and peer-to-peer communication should be addressed.

2.) A thirty-foot long asynchronous serial computer interface uses RS-232-style signaling but at "TTL" logic levels meaning 0 volts for logic-1 and idle and mark or 5 volts for logic-0 or space, nominal levels. The interface is in practically continuous use. (There are no significant periods of idle time.) About half of the time the transmission line is sending logic-1 and the other half of the time it is sending logic-0. A low-impedance driver is connected to a 50 Ω transmission line and a 50 Ω matched load is used to prevent reflections. It works fine but battery life is poor due to all the current flowing in the load. Suggest a change or changes to the source and/or load impedances that will improve battery life. (10 points)

5

$R_L = Z_C$ at the moment

One change that will improve battery life is to reduce the matched load from 50 Ω. This is preventing reflections, but in the process is pulling more power from the battery. The line would not operate as well if there are reflections, but the battery would last longer. It would be a fine line to try to find where the RS-232 style still runs mildly efficient with reflections.

Reduce toward 0 ohms?

3.) A DC motor with a permanent-magnet field, when connected to a 12 V DC source has a no-load speed of 10 000 RPM and a stall (locked rotor) torque (a.k.a. locked rotor torque) of 0.1 Nm. Assume the motor has an ideal linear torque vs. speed characteristic. This motor is connected to a low-impedance 12 V source on the electrical side and to a mechanical load that requires a driving torque that is related to the speed such that

$$T = \sqrt{S}\left(\frac{1}{2500}\text{Nm/RPM}^{0.5}\right)$$

10

Where T = required driving torque (in Nm)
S = load speed (in RPM)

Find the speed at which the motor will operate. (10 points)

when $S=0$ $T=0.1$
when $S=10{,}000$ $T=0$
> linear relationship → $T = 0.1 - 0.00001S$

$S = (2500 \times T)^2$

$S = (250 - 0.025S)^2$

$0 = 0.000625S^2 - 6.25S - 6.25S - S + 62500$

$0 = 0.000625S^2 - 13.5S + 62500$ → solve for S

↳ $S = 6720.78$ or 14879

Speed is 6,720.78 rpm because 14,879 exceeds a no load speed so with load will be slower

4.) Describe in words or pseudo-code how to use an interrupt driven tic-clock system with a microcontroller to generate a pulse of exactly 10 ms each time a certain subroutine is called. Assume the tic-clock interrupt has a frequency of 10 kHz. (10 points)

You would need to define the interrupt so that when it is called, it lasts 10 ms as a pulse. The main code would run alongside the tick clock and when a sub routine gets called, a pulse that turns on for 10ms or for 100 Hz of the clock. So when the main code hits a certain sub routine, this pulse runs. ✕

1

You must use a task scheduler to take advantage of a tic clock.

5.) Consider creating a keyboard-operated music synthesizer by connecting a keyboard to an Arduino microcontroller. Let the synthesizer have 24 keys (two octaves). Each key will have just one SPST switch. The keyboard has a 48-conductor ribbon cable connected to it, two conductors dedicated to each switch. Could this keyboard be connected directly to an Arduino microcontroller board using no additional parts such as diodes, gates, resistors, or transistors? The Arduino board has 13 digital I/O pins available. If so, describe how and describe any limitations of the connection. If not, give a rationale for your conclusion. (10 points)

No, it could not be connected directly without using gates the Arduino only has 13 I/O ports and no spot for the 48 conductor ribbon cable. These inputs would have to be multiplexed in to provoke the use of each key essentially simultaneously. This is performed by using resistors and transistors. It can be connected but not straight to the board because of the lack of a 48 conductor ribbon cable on the board.

3

6.) What is the difference between vectored interrupts and polled interrupts. (10 points)

A polled interrupt notifies through hardware that an interrupt has been recieved and is waiting to be ran, but there is no specific location attached to where the interrupt came from. Vectored interrupts notify through hardware that an interrupt has been requested and has the location of the interrupt with it.

10

7.) A microcontroller has three sources of interrupts. Any of these can be eliminated by adding additional hardware. To keep costs down it is desirable to have the microcontroller handle as many of the interrupts as possible. Can the microprocessor handle all the interrupts below? If not, suggest which one (or ones) should be eliminated (requiring additional hardware). Try to eliminate as few interrupts as possible. Justify your answer. (15 points)

Listed from highest priority (on top) to lowest (on bottom)

Source 1) The interrupt service routine (ISR) takes 50 μs to run. Minimum time between interrupts is 100 μs. Prompt response to the interrupt request is not needed as long as the interrupt is serviced before it is requested again.

Source 2) ISR takes 10 μs. Minimum time between interrupts is 500 μs but the maximum allowable latency (time from request to end of execution of ISR) is 100 μs.

Source 3) ISR takes 5 μs. Minimum time between interrupts is 70 μs. Prompt response is not needed as long as the interrupt is serviced before it is requested again.

The longest critical region in the main software is 8 μs and the interrupts are disabled during each ISR. The longest instruction takes 1 μs.

(Please write your answer on the next page.)

(This page is provided for the answer to the problem stated on the previous page.)

Summary of the question: Will interrupts work? Eliminate as few as possible if any need to be eliminated. Justify your answer.

Source 1) ISR takes 50 µs to run, 100 µs between interrupts.
Source 2) ISR takes 10 µs to run, 500 µs between interrupts, latency < 100 µs req'd.
Source 3) ISR takes 5 µs to run, 70 µs between interrupts
Critical region is 8 µs, interrupts disabled during ISRs, longest instruction is 1 µs.

I believe this is met because if ISR3 is running will be done in 5ms and then 1 runs, will both be done in 55ms so it will be able to run

Yes, it is met, but your calculations could verify that. By using 500 us as T2P your calculation does not verify the 100 us constraint. Also this causes N(2, 1) to be incorrect.

$T_1 = 50\,\mu s$ $T_{1P} = 100\,\mu s$ $T_{1+} = 10\,\mu s$

$T_2 = 10\,\mu s$ $T_{2P} = 500\,\mu s$ $T_{2+} = 8\,\mu s$

$T_3 = 5\,\mu s$ $T_{3P} = 70\,\mu s$ $T_{3+} = 8\,\mu s$

Interrupt density $\frac{50}{100} + \frac{10}{500} + \frac{5}{70} < 1 \rightarrow \frac{207}{350} < 1$ ✓

$N(1,1) = 1$ $N(2,1) = 5$ $N(2,2) = 1$ $N(3,1) = 1$ $N(3,2) = 1$ $N(3,3) = 1$

$N(2,1) = \frac{T_{2P} - T_2}{T_{1P}} = 5$ $N(3,1) = \frac{T_{3P} - T_3}{T_{1P}} = 1$ $N(3,2) = \frac{T_{3P} - T_3}{T_{2P}} = 1$

15

Interrupt 1: $T_{1+} + N(1,1)T_1 < T_{1P} \rightarrow 10 + 1 \times 50 < 100 \rightarrow 60 < 100$ ✓

Interrupt 2: $T_{2+} + N(2,2)T_2 + N(2,1)T_1 < T_{2P} \rightarrow 8 + 1 \times 10 + 5 \times 50 < 500 \rightarrow 268 < 500$ ✓ ✗

Interrupt 3: $T_{3+} + N(3,3)T_3 + N(3,2)T_2 + N(3,1)T_1 < T_{3P}$

$\hookrightarrow 8 + 1 \times 5 + 1 \times 10 + 1 \times 50 < 70 \rightarrow 73 < 70$ ✗

Interrupt 3 fails to meet the inequality. Thus, the interrupts will not work.

A better solution:
Observe that ISR 1 has margin, thus does not need priority. Demote it to lowest priority and everythng will work.

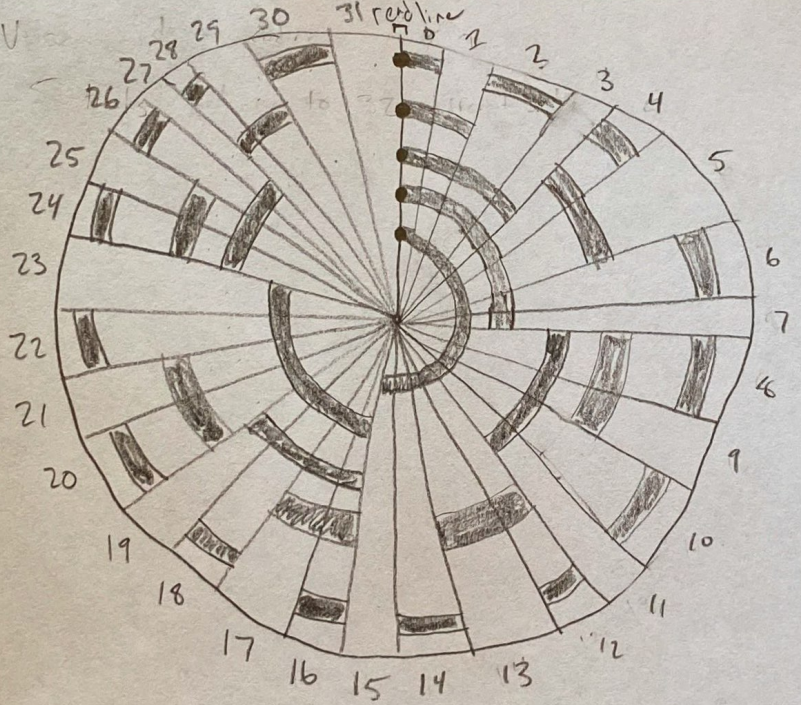
I would eliminate interrupt 3. This would get rid of the interrupt that causes system to bug out and Interrupts 1 and 2 both have some extra time to make up for the lack of interrupt 3. The latency is significantly lower than 100ms as well. Thus, eliminating interrupt 3 would solve the problem.

8.) Design a rotational position sensor for sensing ***absolute*** position with a resolution of 11.25 degrees (1/32 of a revolution). The rotational position sensor needs to only keep track of one revolution. If the shaft of the sensor should roll past the home position, the position code can roll over too. Each time the rotational position of a disk changes by 1/32 of a revolution, an interrupt should be produced. You may assume that the interrupt will be serviced before the disk moves enough to generate another interrupt request. You may decide for yourself how much logic to do in hardware and how much to do in the ISR. Your design should be relatively easy to manufacture. (15 points)

a.) Show the design of the disk and number and location of the sensor(s).

b.) Show a flow chart (or equivalent pseudo-code) to describe the processing that needs to be done in the ISR to keep track of ***absolute*** position.

a)



colored = off or 0
blank = on or 1
dots are sensors

sensors on read line
exactly 5 total sensors

8

This encoder uses binary encoding but does not use V-scan or U-scan sensors, thus it is subject to many problems that we discussed in class.

b)

Flow chart:

each sensor is checked
code is read in as a value
and returns the position

code received	pos
00000	= 0
00001	= 1
00010	= 2
00011	= 3
00100	= 4
00101	= 5
00110	= 6
00111	= 7
01000	= 8
01001	= 9
01010	= 10
01011	= 11
01100	= 12
01101	= 13
01110	= 14

code received	pos
01111	= 15
10000	= 16
10001	= 17
10010	= 18
10011	= 19
10100	= 20
10101	= 21
10110	= 22
10111	= 23
11000	= 24
11001	= 25
11010	= 26
11011	= 27
11100	= 28
11101	= 29
11110	= 30
11111	= 31

9. The performance of a car's shock absorber needs to be monitored to assist with the design and testing of the shock absorber. The fastest rate at which the shock absorber might effectively move to absorb the energy of bumps is 100 Hz. (100 bumps per second for example.)

The transducer used is a linear analog position sensor. It produces a voltage proportional to the length between the two anchor points of the shock absorber. The maximum voltage is +0.100 V at full extension of the shock absorber and the minimum is –0.100 V at full compression. The transducer has a frequency response up to 10000 Hz. Thus, it is not only is sensitive to the motion of the shock absorber, it is also sensitive to some of the sound of the wheel rolling over the road and conducted into the shock-absorber.

A commercially available analog-to-digital converter is in use. The sample rate of the converter is 1000 Hz. (One sample is taken every millisecond). This analog-to-digital converter has an input range of +10 V to –10 V and provides a 10-bit output in a two's complement format such that +10 V produces the code $01\ 1111\ 1111_{TC}$ which corresponds to $+511_{10}$ and –10 V produces the code $10\ 0000\ 0001_{TC}$ which corresponds to -511_{TC}. This converter has no anti-aliasing filter. The user is expected to add their own anti-aliasing filter.

The user figured that the 10000 Hz frequency response of the transducer and the 1000 Hz sample rate are 100 and 10 times (respectively) more than needed since the user is only interested in motions at 100 Hz or less. Thus, these were perceived as very good specifications. (The user may be naive in this determination.) The ±10 V input range is also perceived as 100 times greater than needed so that should also be very good. (The user may be naive in this determination.) The user is getting poor results from this system.

For an anti-aliasing filter the user has resorted to averaging 8 sequential samples (using digital addition of the sampled data, then dividing by 8) in order to get better results. This has reduced their effective sample rate to 1000/8 = 125 Hz. (Every eight digital samples get averaged to produce just one average sample for the interval.) However, the samples still are not accurate.

Assume that averaging digital samples must be abandoned as it is an incorrect method of addressing the problems here. Assume analog filters and amplifiers are available to any specifications you might like. For a filter, specify if it should pass low or high frequencies and what the cutoff frequency should be. For an amplifier, specify the gain. How can you use filters and amplifiers to improve the results? Explain your choices. (10 points)

Anti aliasing filter $f_{nyquist} > 1.1 f_{max}$ $f_{nyquist} > 10 f_{max}$

10

Need to eliminate the noise of just driving

Need to filter analog before it gets digital

Filter should be low pass that has frequency cutoff that nothing above 110 Hz gets through

Amplifier Gain should boost the signal up to more valid ranges of the analog to digital sensor by factor of 100 so that it operates a signal ± 10 V.

Name: Zachary Sanford

Dordt University Engineering Department

EGR 304, Microprocessor Interfacing — Final Exam, May 4, 2017.
Open book, open notes, take-home.
An Internet-connected computer is required. A calculator is also required.

49/100 = C+

1.) A small local-area computer network (e.g. a network of two or three computers sharing files, printers, etc.—no Internet connection) could be made using USB connections or using Ethernet connections between the computers. Describe some of the challenges and advantages that would be encountered when using a USB connection for networking as opposed to using an Ethernet connection. How are these challenges rooted in the technological differences of the two connection types? (10 points)

2

Your answer does not address networking. Issues such as distance and peer-to-peer communication should be addressed.

Ethernet connection - one of the first things required for ethernet is that a frame (synchronous to client-server model) to send information must be long enough to flood the network of computers, otherwise a collision could occur and corrupt the information being sent. This may happen because the synchronous clock for all computers may delay in propagation and end up showing different times for all computers. Ethernet has advatages with hieravchy levels. Using TCP/IP stacking to connect to multiple systems. Signal is stable and consistant.

USB connection - usb has many advantages. The connectors enfore network topography, meaning it is easy to use. Class drivers allow for easy connection between multiple computers and hardware. USB's supply power as well as data. Transfer large files of Mbps over USB fast. A problem can occur when the USB has no way to get the attention of the host, Interrupt transaction. There is a bound on the latency of the devices, this is delayed if there is an interrupt request.

A thirty-foot long asynchronous serial computer interface uses RS-232-style signaling but at "TTL" logic levels meaning 0 volts for logic-1 and idle and mark or 5 volts for logic-0 or space, nominal levels. The interface is in practically continuous use. (There are no significant periods of idle time.) About half of the time the transmission line is sending logic-1 and the other half of the time it is sending logic-0. A low-impedance driver is connected to a 50 Ω transmission line and a 50 Ω matched load is used to prevent reflections. It works fine but battery life is poor due to all the current flowing in the load. Suggest a change or changes to the source and/or load impedances that will improve battery life. (10 points)

Improving batter life means drawing less current, getting rid of parasidic capacity, change voltage use, change wave frequency on which the data is being transfered. Power = Voltage · Current

Clock management - running the clock frequency and increasing frames can cause the code to execute faster and return to low power.

2

Transmission Line - divide the transmission line into sections and have resistors inductors and capacitors at each section, this helps even out the voltage and current loads forced on the line and stablizes the power. The cable should be insulated to reduce parasidic current.

Impedance Matching - the impedance along the 30' long interface will need to match the t-line and load to make sure no reflection occurs. Voltage will decrease when current is negative (discharging) meaning high frequency on t-line

None of these ideas are very effective.

3.) A DC motor with a permanent-magnet field, when connected to a 12 V DC source has a no-load speed of 10 000 RPM and a stall (locked rotor) torque (a.k.a. locked rotor torque) of 0.1 Nm. Assume the motor has an ideal linear torque vs. speed characteristic. This motor is connected to a low-impedance 12 V source on the electrical side and to a mechanical load that requires a driving torque that is related to the speed such that

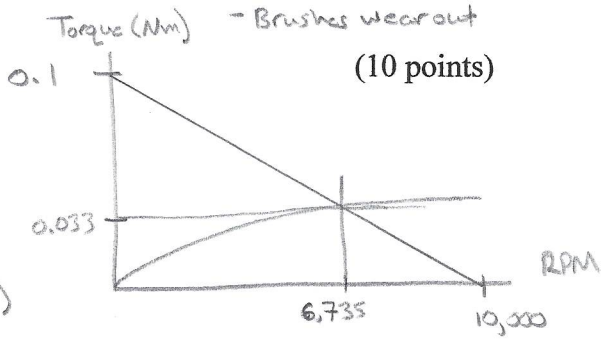
$$T = \sqrt{S}\left(\frac{1}{2500}\,\text{Nm/RPM}^{0.5}\right)$$

Where T = required driving torque (in Nm)
S = load speed (in RPM)

Brushed DC Motor
- No sensors needed
- Easy to control
- Voltage → speed
- Current → torque
- Brushes wear out

10

Find the speed at which the motor will operate. (10 points)



$T_m = 0.1 - 0.00001(S)$

$T_f = \sqrt{S}\left(\frac{1}{2500}\,\text{Nm/RPM}^{0.5}\right)$

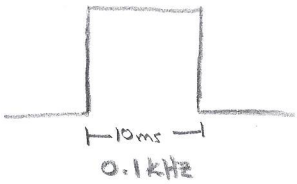
$\sqrt{S}\left(\frac{1}{2500}\frac{\text{Nm}}{\sqrt{\text{RPM}}}\right) = 0.1\,\text{N·m} - [0.1\,\text{N·m}/10{,}000\,\text{RPM}](S)$

square both sides

I used mathcad graph to solve for speed.
The operating speed is close to 6,710 RPM and 6,735 RPM.
The graph is hard to get an exact number.

4.) Describe in words or pseudo-code how to use an interrupt driven tic-clock system with a microcontroller to generate a pulse of exactly 10 ms each time a certain subroutine is called. Assume the tic-clock interrupt has a frequency of 10 kHz. (10 points)

10 kHz = 0.1 ms 10ms = 0.1 kHz



When the subroutine is called an (ISR) interrupt service request is triggered. When the interrupt is serviced it will generate a tick_start() time and keep the voltage on high, or low depending, for a tenth of the frequency or 100 Hz. This can happen backwards as well, keeping the voltage high for 100 Hz, then the end_tick() will trigger and the timer will have reset. 100Hz = 10ms

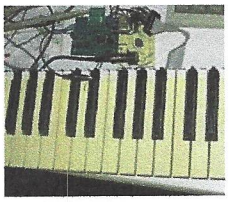
and interrupt

2

Using the internal clock interrupt in the code requires you to use a variable called, global variable "relativeTime", this allows you to update the ISR. Much of the time the tic counter will count up one per kH until 10 has been reached and the interrupt is over.

You must use a task scheduler to take advantage of a tic clock.

5.) Consider creating a keyboard-operated music synthesizer by connecting a keyboard to an Arduino microcontroller. Let the synthesizer have 24 keys (two octaves). Each key will have just one SPST switch. The keyboard has a 48-conductor ribbon cable connected to it, two conductors dedicated to each switch. Could this keyboard be connected directly to an Arduino microcontroller board using no additional parts such as diodes, gates, resistors, or transistors? The Arduino board has 13 digital I/O pins available. If so, describe how and describe any limitations of the connection. If not, give a rationale for your conclusion. (10 points)



Synthesizer def - a instrument producing a wide variety of sounds by generating and combining signals of different frequencies.

SPST switch - Single Pole Single Throw x 24

Question - does the keyboard synthesizer need current and voltage limiters?

Question - can we combine the 48-conductor ribbon cable to limit the number of pins used?

Question - can we combine 24 keys to 12 keys, two keys sharing one pin? That means that we share one pin with two SPST switches.

Idea - 12 pins for 24 keys (two shared), 1 pin for 48-conductor ribbon.

Idea - 12 pins for 48-cond. ribbon (combined), 1 pin for 24 keys (shared)

Idea - 4 pins for 24 keys, 7 pins for cond. ribbon.

Conclusion - No, you could not connect the keyboard synthesizer directly without using resistors & transistors.

Yes, you could do it without diodes & gates.

The reason for this being that there are not enough pin I/O available for every frequency at the same time.

If the pins are being shared by two or more SPST switches or the 48-cond. ribbon, the keys will not all come through. The resistors are needed for the power/current protection of the Arduino.

If this is possible I would be impressed and supprised.

4

6.) What is the difference between vectored interrupts and polled interrupts. (10 points)

Polled interrupts - also known as periodic polling. This interrupt requires a timer interrupt, meaning it needs additional hardware. This interrupt uses a status bit to check I/O device before reading or writing to it. While the I/O device is busy the CPU may work on other threads of code.

Vectored interrupts - similar to interrupt driven, the I/O device has a method in hardware to request I/O service. It tells the computer that handles interrupts an I/O interrupt requests attention from the device at the hardware level.

The difference being how the interrupt is handled and how the interrupts get processed by the I/O interrupt service handler.

7.) A microcontroller has three sources of interrupts. Any of these can be eliminated by adding additional hardware. To keep costs down it is desirable to have the microcontroller handle as many of the interrupts as possible. Can the microprocessor handle all the interrupts below? If not, suggest which one (or ones) should be eliminated (requiring additional hardware). Try to eliminate as few interrupts as possible. Justify your answer. (15 points)

Listed from highest priority (on top) to lowest (on bottom)

Source 1) The interrupt service routine (ISR) takes 50 µs to run. Minimum time between interrupts is 100 µs. Prompt response to the interrupt request is not needed as long as the interrupt is serviced before it is requested again.

Source 2) ISR takes 10 µs. Minimum time between interrupts is 500 µs but the maximum allowable latency (time from request to end of execution of ISR) is 100 µs.

Source 3) ISR takes 5 µs. Minimum time between interrupts is 70 µs. Prompt response is not needed as long as the interrupt is serviced before it is requested again.

The longest critical region in the main software is 8 µs and the interrupts are disabled during each ISR. The longest instruction takes 1 µs.

(Please write your answer on the next page.)

(This page is provided for the answer to the problem stated on the previous page.)

Summary of the question: Will interrupts work? Eliminate as few as possible if any need to be eliminated. Justify your answer.

Source 1) ISR takes 50 µs to run, 100 µs between interrupts.
Source 2) ISR takes 10 µs to run, 500 µs between interrupts, latency < 100 µs req'd.
Source 3) ISR takes 5 µs to run, 70 µs between interrupts
Critical region is 8 µs, interrupts disabled during ISRs, longest instruction is 1 µs.

#1) $T_{p1} = 100\,\mu s$ $T_1 = 50\,\mu s$ $T_{1+} = 50\,\mu s$

#2) $T_{p2} = 500\,\mu s$ ✗ $T_2 = 10\,\mu s$ $T_{2+} = 50\,\mu s$

#3) $T_{p3} = 70\,\mu s$ $T_3 = 5\,\mu s$ $T_{3+} = 1\,\mu s$ ✗

$N(1,1) = 1$ $N(2,1) = 5$ ✗ $N(3,1) = 1^{0.65}$

$N(2,2) = 1$ $N(3,2) = 1^{0.13}$

$N(3,3) = 1$

$N(i,x) = \dfrac{\text{ceil}(T_{pi} - T_i)}{T_{px}}$

Step 1 check density $\dfrac{50\,\mu s}{100\,\mu s} + \dfrac{10}{500} + \dfrac{5}{70} = \dfrac{207}{350}$ OK

Step 2 check max latency

$(i=1)$ $T_{1+} + N(1,1)T_1 < T_{p1} \rightarrow 50\,\mu s + (1)50\,\mu s = 100\,\mu s < 100\,\mu s$ Fail ✗

$(i=2)$ $T_{2+} + N(2,2)T_2 + N(2,1)T_2 \rightarrow 50\,\mu s + (1)10\,\mu s + (5)10\,\mu s = 110\,\mu s < 100\,\mu s$ ✗

Fail

$(i=3)$ $T_{3+} + N(3,3)T_3 + N(3,2)T_3 + N(3,1)T_3 \rightarrow 1\,\mu s + (1)5\,\mu s + (1)5\,\mu s + (1)5\,\mu s$

$16\,\mu s < 70\,\mu s$ ✗

OK

Lots of errors in the set-up of the inequalities. Conclusion is pretty close but you should draw a conclusion instead of just speculating.

Conclusion This microprocessor will not run properlly.

10

Interrupt Source 1) and Source 2) are failures.

The first thing I would do is rearrange the interrupts that would help putting source 3) first priority. If needed I would eliminate Source 2). Then try again and it should work putting Source 3) as highest priority because it is the fastest and most often.

8.) Design a rotational position sensor for sensing ***absolute*** position with a resolution of 11.25 degrees (1/32 of a revolution). The rotational position sensor needs to only keep track of one revolution. If the shaft of the sensor should roll past the home position, the position code can roll over too. Each time the rotational position of a disk changes by 1/32 of a revolution, an interrupt should be produced. You may assume that the interrupt will be serviced before the disk moves enough to generate another interrupt request. You may decide for yourself how much logic to do in hardware and how much to do in the ISR. Your design should be relatively easy to manufacture. (15 points)

a.) Show the design of the disk and number and location of the sensor(s).

b.) Show a flow chart (or equivalent pseudo-code) to describe the processing that needs to be done in the ISR to keep track of ***absolute*** position.

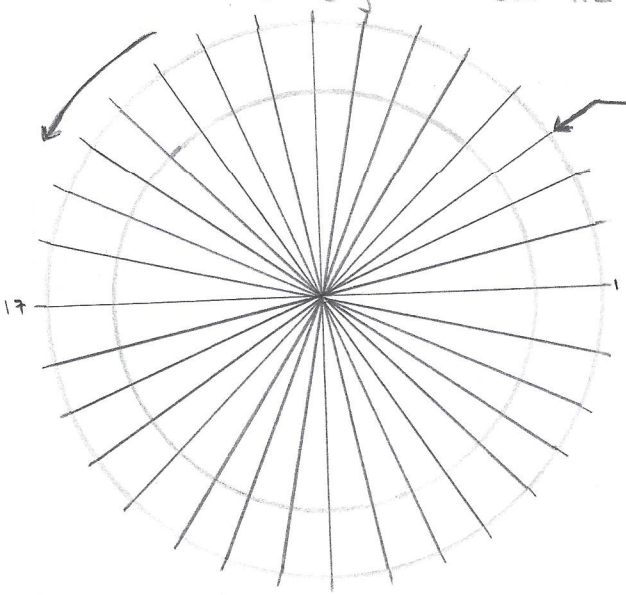
a) If the degree of resolution is 11.25° and this is 1/32nd of 360°. I could have 32 "teeth" on the rotator and have a "rising-edge" tick sensor count up to a max of 31 counts then roll back to a count of zero & and start counting again. The ISR will execute every time the counter changes. as long as: counter = count+1 < 32; the interrupt will continue() execute

else: count = 32;
count = 0

2

I will only need one sensor.
Absolute position sensor gives information on its position within a given scaler or range with out the need for a reference point.

$\frac{360°}{1} \cdot \frac{1}{32^{nd}\ res} = 11.25°/resol.$



Sensor, everytime the sensor sees the tooth/line it counts up.

b) Execute IRS, use a counter

```
define sensor() as input to counter
count = tick_sensor()
count = 0
counter == count + 1
    if  count < 32 :
        continue()
    else  count = 32 :
        reset count
```

This enables the ISR to execute everytime the count changes. → `ISR == count + i ; execute_ISR()`

You have described a relative position encoder.

9. The performance of a car's shock absorber needs to be monitored to assist with the design and testing of the shock absorber. The fastest rate at which the shock absorber might effectively move to absorb the energy of bumps is 100 Hz. (100 bumps per second for example.)

0.1V+ 0.1V- The transducer used is a linear analog position sensor. It produces a voltage proportional to the length between the two anchor points of the shock absorber. The maximum voltage is +0.100 V at full extension of the shock absorber and the minimum is –0.100 V at full compression. The transducer has a frequency response up to 10000 Hz. Thus, it is not only is sensitive to the motion of the shock absorber, it is also sensitive to some of the sound of the wheel rolling over the road and conducted into the shock-absorber. 10000 Hz (response) / 100 Hz rate = 100 responses/rate +

0.2V range 2V range 10 times as much V A commercially available analog-to-digital converter is in use. The sample rate of the converter is 1000 Hz. (One sample is taken every millisecond). This analog-to-digital converter has an input range of +10 V to –10 V and provides a 10-bit output in a two's complement format such that +10 V produces the code 01 1111 1111$_{TC}$ which corresponds to $+511_{10}$ and –10 V produces the code 10 0000 0001$_{TC}$ which corresponds to –511$_{TC}$. This converter has no anti-aliasing filter. The user is expected to add their own anti-aliasing filter. Not included in the sampler. 1000 Hz (sample) / 100 Hz (rate) 10 samples/rates +

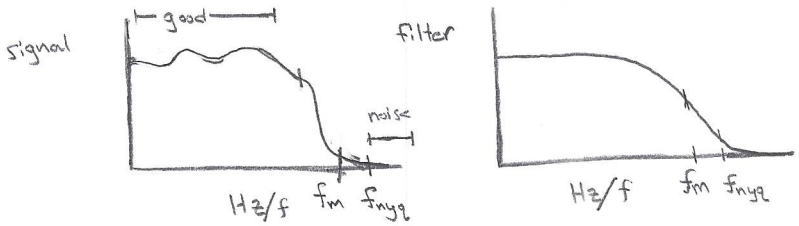
The user figured that the 10000 Hz frequency response of the transducer and the 1000 Hz sample rate are 100 and 10 times (respectively) more than needed since the user is only interested in motions at 100 Hz or less. Thus, these were perceived as very good specifications. (The user may be naive in this determination.) The ±10 V input range is also perceived as 100 times greater than needed so that should also be very good. (The user may be naive in this determination.) The user is getting poor results from this system. Your precision is only as good as your input not output.

For an anti-aliasing filter the user has resorted to averaging 8 sequential samples (using digital addition of the sampled data, then dividing by 8) in order to get better results. This has reduced their effective sample rate to 1000/8 = 125 Hz. (Every eight digital samples get averaged to produce just one average sample for the interval.) However, the samples still are not accurate. ← that's bad!

Assume that averaging digital samples must be abandoned as it is an incorrect method of addressing the problems here. Assume analog filters and amplifiers are available to any specifications you might like. For a filter, specify if it should pass low or high frequencies and what the cutoff frequency should be. For an amplifier, specify the gain. How can you use filters and amplifiers to improve the results? Explain your choices. (10 points)

10 I would first get an Anti-Aliasing Filter and put it right between the Analog signal and Sampler to smooth out the signal and junk. Make sure $f_{nyquist} > f_{max}$ so that all the signal makes it through the anti-aliasing filter. Make $f_{nyquist}$ pass low frequencies to reject noise. $F_o \triangleq f_o/f_s < \frac{1}{2}$ $f_o < f_{nyquist} < f_s/2$

$f_o = 100$ Hz $f_s = 1000$ Hz. cutoff should $f_{nyquist} > 1.1 f_{max}$ ~120 Hz cutoff.

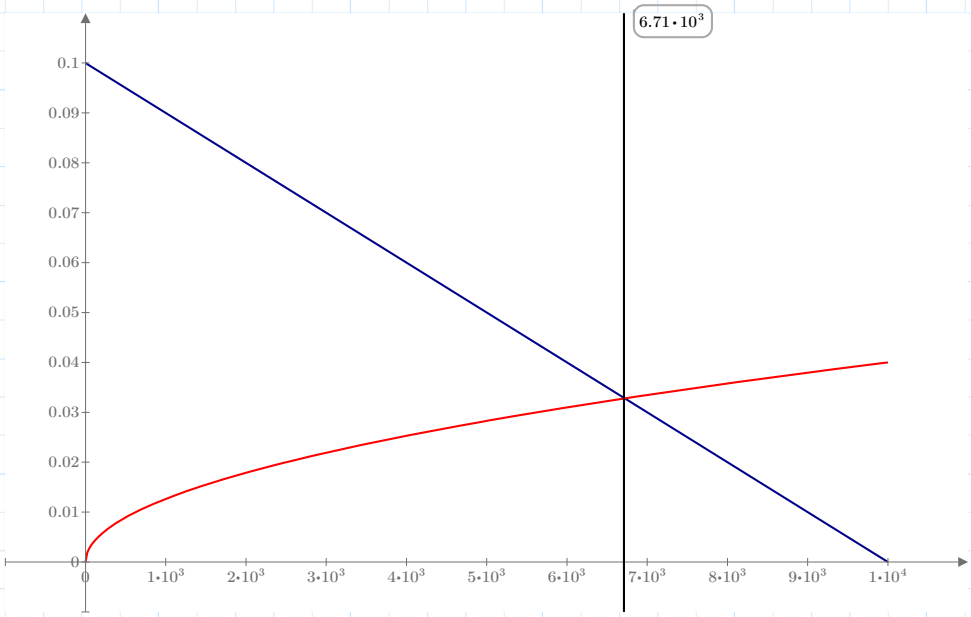


Filters alter the amplitudes and phase characteristic while Amplifiers alter the gain.

The amplifier should be the last piece or right before the Quantizer.

gain $A_{volt} = \frac{V_{output}}{V_{input}}$ $\frac{10V \text{ (converter)}}{0.1V \text{ (shock)}} = 100 V_{gain}$ $20\log_{10}(100) =$ 40 dB Gain

$$T_m(x) := 0.1\ N \cdot m - \left(0.00001\ \frac{N \cdot m}{rpm}\right) \cdot x \cdot rpm \qquad T_f(x) := \sqrt{x \cdot rpm} \cdot \left(\frac{1}{2500} \cdot \frac{N \cdot m}{\sqrt{rpm}}\right)$$



x

$T_m(x)\ \ (J)$

$T_f(x)\ \ (J)$

Name: Shane Tinklenberg

Dordt University Engineering Department

EGR 304, Microprocessor Interfacing — Final Exam, May 4, 2017. 93/100 = A

Open book, open notes, take-home.

An Internet-connected computer is required. A calculator is also required.

1.) A small local-area computer network (e.g. a network of two or three computers sharing files, printers, etc.—no Internet connection) could be made using USB connections or using Ethernet connections between the computers. Describe some of the challenges and advantages that would be encountered when using a USB connection for networking as opposed to using an Ethernet connection. How are these challenges rooted in the technological differences of the two connection types? (10 points)

8

The advantage of using a USB driver would include the ability of the device to sort of "morph" itself into a preexisting class of device as specified by the USB standard such that windows would know how to send it documents very easily. Although both ethernet and USB utilize hot plugging, the use of USB allows for quick and often times automatic installation of drivers without requiring user input. Thus many times devices using USB require no setup at all other than being plugged into the device they are intend to be used on. Furthermore, USB also supports the use of hubs that allows devices to be connected a central location before being routed to the motherboard. This is possible in Ethernet as well, but it less simple. The disadvantages of using USB for a network are rooted in the fact that USB is built on a Master-Slave model that typically necessitates that one device taking control of the exchange of data. Thus, if two computers were plugged into the same USB network there may be some conflict between deciding who exactly was master and slave. This would also create issue if both printers tried to access the same hardware at the same time. Ethernet is more well suited for this type of multiple access style of information exchange. The collision detection features built into ethernet make it much more capable of handling multiple requests for data at the same time.

Distance is another factor that could be considered. USB is more limited in length of wires.
Based on citing two good distinctions (class drivers, host-client vs. peer-to-peer) you get full credit.

2.) A thirty-foot long asynchronous serial computer interface uses RS-232-style signaling but at "TTL" logic levels meaning 0 volts for logic-1 and idle and mark or 5 volts for logic-0 or space, nominal levels. The interface is in practically continuous use. (There are no significant periods of idle time.) About half of the time the transmission line is sending logic-1 and the other half of the time it is sending logic-0. A low-impedance driver is connected to a 50 Ω transmission line and a 50 Ω matched load is used to prevent reflections. It works fine but battery life is poor due to all the current flowing in the load. Suggest a change or changes to the source and/or load impedances that will improve battery life. (10 points)

In order to improve this situation, I would swap out the low impedance driver for a 50-ohm driver such that the source is matched to the transmission line which is matched to the termination.

8

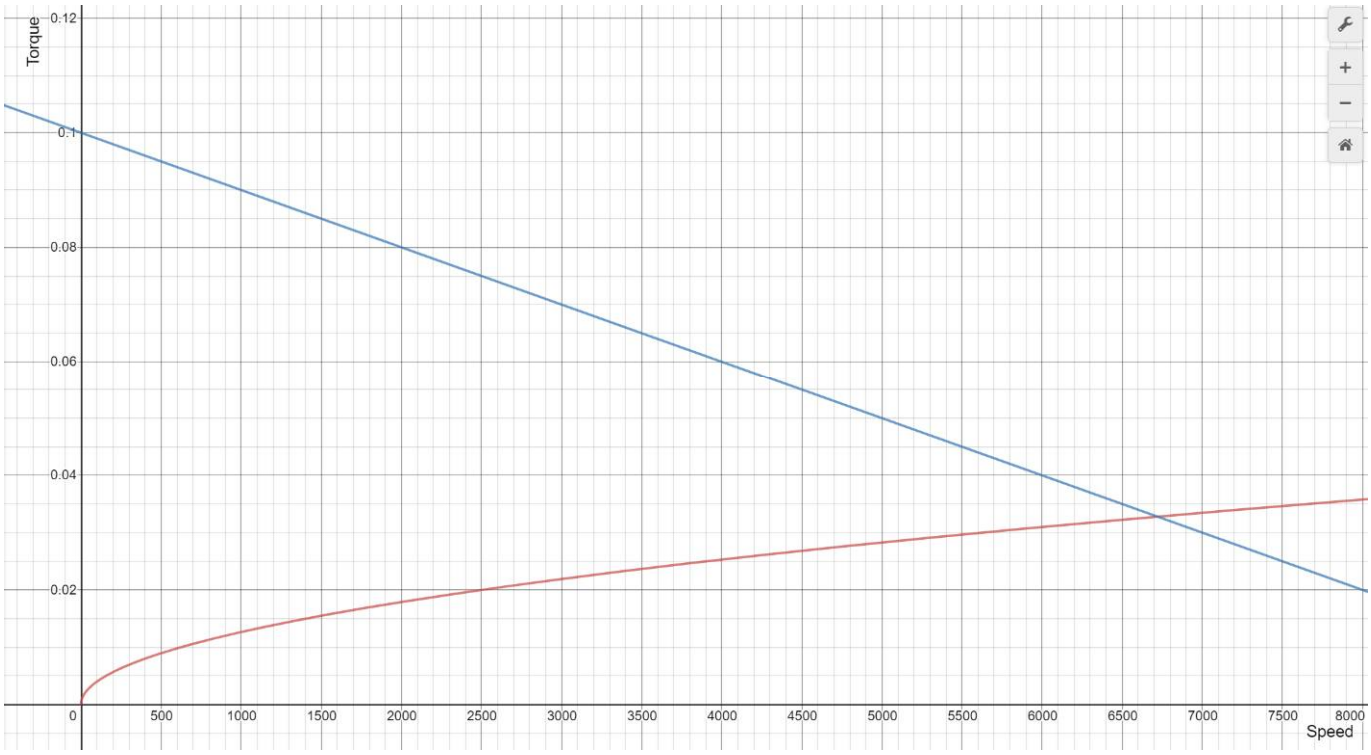
An important further improvement is possible. Raise R_L as much as possible toward an open circuit. Reflections will now be absorbed at the source.

3.) A DC motor with a permanent-magnet field, when connected to a 12 V DC source has a no-load speed of 10 000 RPM and a stall (locked rotor) torque (a.k.a. locked rotor torque) of 0.1 Nm. Assume the motor has an ideal linear torque vs. speed characteristic. This motor is connected to a low-impedance 12 V source on the electrical side and to a mechanical load that requires a driving torque that is related to the speed such that

$$T = \sqrt{S}\left(\frac{1}{2500}\,\mathrm{Nm/RPM}^{0.5}\right)$$

Where T = required driving torque (in Nm)
S = load speed (in RPM)

Find the speed at which the motor will operate. (10 points)



By plotting the curves T1(s) = sqrt(s)/2500 and T2(s) = 0.1 – 0.00001x I was able to solve for the speed by finding the s value of the intersection of the two functions. **The speed at which the motor will operate is ~6720.8 revolutions per minute.**

10

4.) Describe in words or pseudo-code how to use an interrupt driven tic-clock system with a microcontroller to generate a pulse of exactly 10 ms each time a certain subroutine is called. Assume the tic-clock interrupt has a frequency of 10 kHz.

(10 points)

I would create two functions.

I would first attach an interrupt service routine to function two in the setup routine that is triggered by an equal compare value in the system tic clock compare register.

I would then set up the system tic clock compare register to a value of 100 such that after 100 periods of the tic clock the interrupt service routine would be triggered when enabled.

I would then disable the tic clock interrupt.

FunctionOne() – Takes no parameters, called to trigger the 10 ms pulse on.

-- Set the appropriate parallel port register bit high as designated by the pin that should produce the pulse

-- Set the tic clock to zero.

--Enable tic clock interrupt.

--Return to executing whatever routine previously active

FunctionTwo() – Takes no parameters, triggered by the true compare flag of the system tic clock compare register

--Set the appropriate parallel port register bit LOW as designated by the pin that should produce the pulse

--Disables the system tic clock interrupt

10

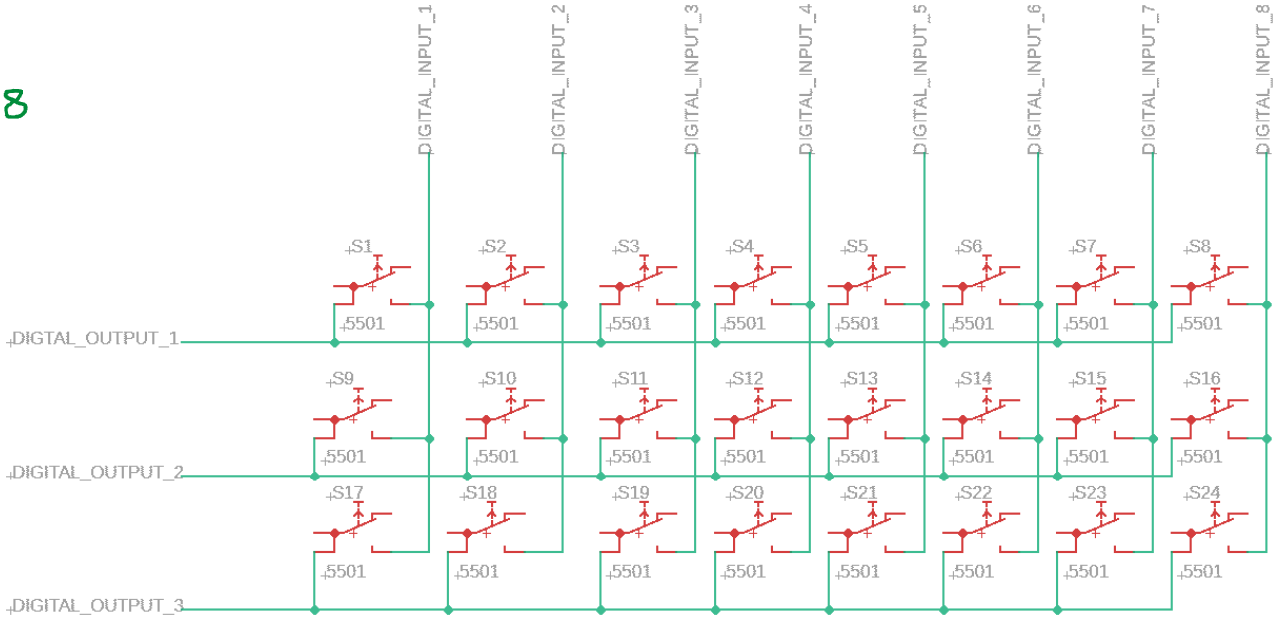
This will work, but it takes total control of the tic-clock interrupt. A better way is to use a task scheduler. Let your first subroutine (or function, method, whatever) schedule the turn-on of the pulse for just a brief moment ahead of the present tic-clock count (time) The task scheduler will then call the second subroutine which will start the pulse and also schedule the end of the pulse. A third subroutine will end the pulse. Any number of other tasks can then also use the task scheduler since the tic-clock interrupt is not being enabled or disabled in service of just one task.

5.) Consider creating a keyboard-operated music synthesizer by connecting a keyboard to an Arduino microcontroller. Let the synthesizer have 24 keys (two octaves). Each key will have just one SPST switch. The keyboard has a 48-conductor ribbon cable connected to it, two conductors dedicated to each switch. Could this keyboard be connected directly to an Arduino microcontroller board using no additional parts such as diodes, gates, resistors, or transistors? The Arduino board has 13 digital I/O pins available. If so, describe how and describe any limitations of the connection. If not, give a rationale for your conclusion. (10 points)

In this scenario it is possible to monitor all twenty-four switches assuming your loop routine can run continuously without any significant interruptions or blocks in the routine. In order to accomplish this task, I would set up three parallel port pins as a digital output using the appropriate data direction register. I would then set up eight parallel port pins as an input using the appropriate data direction register. I would then wire each digital output to eight switches and connect the normally open end of each switch to its own input pin. I would repeat that process for each output pin until I had three rows of eight switches with each row connected to its own output pin and each switch within a row connected to one of the eight input pins I created (See Schematic Below). I would then write a looping routine that would raise one of the three digital outputs high at a time. While each digital output is HIGH, I would then check the input status of each the eight input pins. I would then set the output of that row to LOW and set the next row's output to HIGH and check the input status of all eight input pins. Assuming the loop is not blocked and executes quickly enough, I could determine the state of each button by continually checking the status of each row of eight buttons using only eleven pins.

You need to consider the effect of simultaneous key presses.
To prevent shorts, you need to use open-collector outputs for the "digital output" pins.

8



6.) What is the difference between vectored interrupts and polled interrupts? (10 points)

10

In vectored interrupts the system is aware where the request for an interrupt originated from. In polled interrupts, the system must poll each possible source of interrupts to check the state and determine the source of the interrupt service request and thereby what action is required.

7.) A microcontroller has three sources of interrupts. Any of these can be eliminated by adding additional hardware. To keep costs down it is desirable to have the microcontroller handle as many of the interrupts as possible. Can the microprocessor handle all the interrupts below? If not, suggest which one (or ones) should be eliminated (requiring additional hardware). Try to eliminate as few interrupts as possible. Justify your answer. (15 points)

Listed from highest priority (on top) to lowest (on bottom)

Source 1) The interrupt service routine (ISR) takes 50 μs to run. Minimum time between interrupts is 100 μs. Prompt response to the interrupt request is not needed as long as the interrupt is serviced before it is requested again.

Source 2) ISR takes 10 μs. Minimum time between interrupts is 500 μs but the maximum allowable latency (time from request to end of execution of ISR) is 100 μs.

Source 3) ISR takes 5 μs. Minimum time between interrupts is 70 μs. Prompt response is not needed as long as the interrupt is serviced before it is requested again.

The longest critical region in the main software is 8 μs and the interrupts are disabled during each ISR. The longest instruction takes 1 μs.

(Please write your answer on the next page.)

Summary of the question: Will interrupts work? Eliminate as few as possible if any need to be eliminated. Justify your answer.

Source 1) ISR takes 50 μs to run, 100 μs between interrupts.
Source 2) ISR takes 10 μs to run, 500 μs between interrupts, latency < 100 μs req'd.
Source 3) ISR takes 5 μs to run, 70 μs between interrupts
Critical region is 8 μs, interrupts disabled during ISRs, longest instruction is 1 μs.

T_1 = 50 us T_{P1} = 100 us T_{10} = 10 us

T_2 = 10 us T_{P2} = 500 us T_{2+} = 8 us

T_3 = 5 us T_{P3} = 70 us T_{3+} = 8 us

50/100 + 10/500 + 5/70 = 0.59 < 1.00 Interrupt Density Check Passed

$$N(i,x) = \frac{TPi - Ti}{TPx}$$

N(1,1) = 1

N(2,1) = 5

N(2, 2) = 1

N(3, 1) = 1

N(3, 2) = 1

N(3, 3) = 1

Interrupt One:

10 + 1(50) < 100

60 < 100 Interrupt One Passes

Interrupt Two:

8 + 1(10) + 5(50) < 500

268 < 500 Interrupt Two Passes

15

Interrupt Three:

8 + 1(5) + 1(10) + 1(50) < 70

73 ≮ 70 Interrupt Three Fails

In order to ensure stable operation of the system you could eliminate the use of interrupt three which is causing the system to fail and is the lowest priority interrupt on the system.

A better solution:
Observe that ISR 1 has margin, thus does not need priority.
Demote it to lowest priority and everything will work.

8.) Design a rotational position sensor for sensing ***absolute*** position with a resolution of 11.25 degrees (1/32 of a revolution). The rotational position sensor needs to only keep track of one revolution. If the shaft of the sensor should roll past the home position, the position code can roll over too. Each time the rotational position of a disk changes by 1/32 of a revolution, an interrupt should be produced. You may assume that the interrupt will be serviced before the disk moves enough to generate another interrupt request. You may decide for yourself how much logic to do in hardware and how much to do in the ISR. Your design should be relatively easy to manufacture. (15 points)

a.) Show the design of the disk and number and location of the sensor(s).

b.) Show a flow chart (or equivalent pseudo-code) to describe the processing that needs to be done in the ISR to keep track of ***absolute*** position.

a.

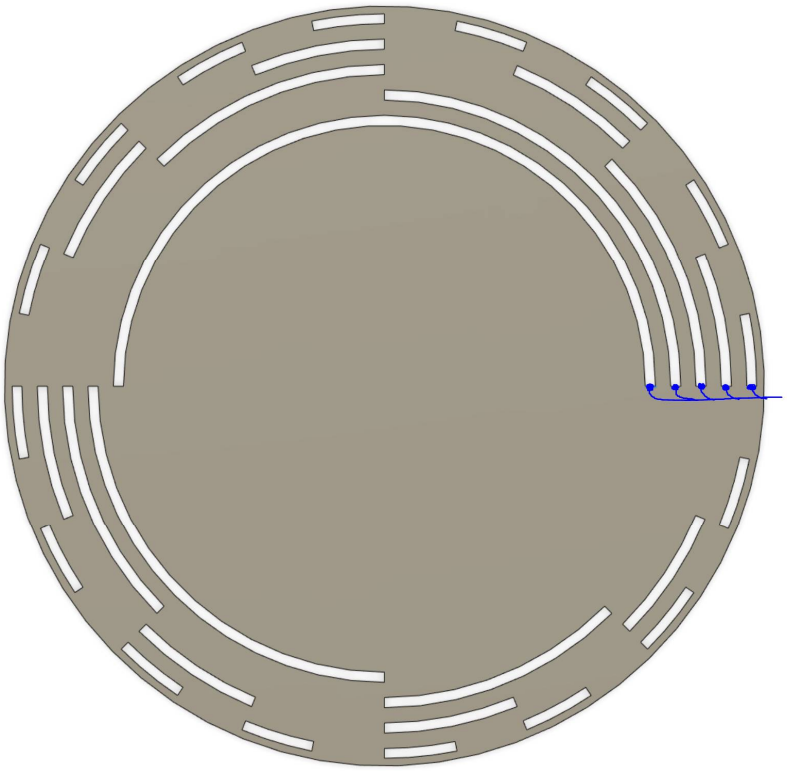


Figure 1.1
In this figure each blue dot represents one of the five stationary optical sensors. The sensors should be wired in such a fashion such that the rightmost sensor represents the least significant bit in the software.

The encoder wheel should be divided into 32 equal arc sections equaling 11.25 degrees each. Each arc should contain five independent sections with the appropriate sections such that each section represents one of five data bits(In this case we will make the encoder optical so each section will be either By using 5 bits of information per 11.25 degree section, it is possible to assign an independent five bit word to each section such that the each section is represented by 00000, 00001, 00010, etc. There should be five optical sensors mounted in line with one another along such that the center of each sensor rests on one of the center of one of the five optical tracks.

b. In order to monitor the position of the I would first attach the input of each optical sensor to pins three through seven on the Arduino with the least significant bit sensor (the rightmost sensor on Figure 1.1) wired to pin three, the second least significant bit wired to pin four, and so on and so forth. I would then attach an interrupt service routine to pin three that is triggered by both rising and falling edges. I would then follow the pseudo-code found on the next page.

Pseudo-Code:

Beginning of Interrupt Service Routine Triggered by rising or falling edge of pin three:

Get the eight-bit word on Port D and store it in a temporary eight-bit unsigned integer.

Bit shift the temporary variable three times such that the least significant sensor bit is now represented by the least significant bit in the temporary variable word.

Set the value of the temporary variable into global volatile eight-bit unsigned integer responsible for holding the current position.

Return to the normal loop routine.

14

As we discussed in class, binary codes have too many issues to use in this application.
This design done with Gray code would work. (Modify the interrupt request to be triggered by the change of any sensor's state to accommodate Gray code.)

9. The performance of a car's shock absorber needs to be monitored to assist with the design and testing of the shock absorber. The fastest rate at which the shock absorber might effectively move to absorb the energy of bumps is 100 Hz. (100 bumps per second for example.)

   The transducer used is a linear analog position sensor. It produces a voltage proportional to the length between the two anchor points of the shock absorber. The maximum voltage is +0.100 V at full extension of the shock absorber and the minimum is –0.100 V at full compression. The transducer has a frequency response up to 10000 Hz. Thus, it is not only is sensitive to the motion of the shock absorber, it is also sensitive to some of the sound of the wheel rolling over the road and conducted into the shock-absorber.

   A commercially available analog-to-digital converter is in use. The sample rate of the converter is 1000 Hz. (One sample is taken every millisecond). This analog-to-digital converter has an input range of +10 V to –10 V and provides a 10-bit output in a two's complement format such that +10 V produces the code 01 1111 1111_{TC} which corresponds to $+511_{10}$ and –10 V produces the code 10 0000 0001_{TC} which corresponds to -511_{TC}. This converter has no anti-aliasing filter. The user is expected to add their own anti-aliasing filter.

   The user figured that the 10000 Hz frequency response of the transducer and the 1000 Hz sample rate are 100 and 10 times (respectively) more than needed since the user is only interested in motions at 100 Hz or less. Thus, these were perceived as very good specifications. (The user may be naive in this determination.) The ±10 V input range is also perceived as 100 times greater than needed so that should also be very good. (The user may be naive in this determination.) The user is getting poor results from this system.

   For an anti-aliasing filter, the user has resorted to averaging 8 sequential samples (using digital addition of the sampled data, then dividing by 8) in order to get better results. This has reduced their effective sample rate to 1000/8 = 125 Hz. (Every eight digital samples get averaged to produce just one average sample for the interval.) However, the samples still are not accurate.

   Assume that averaging digital samples must be abandoned as it is an incorrect method of addressing the problems here. Assume analog filters and amplifiers are available to any specifications you might like. For a filter, specify if it should pass low or high frequencies and what the cutoff frequency should be. For an amplifier, specify the gain. How can you use filters and amplifiers to improve the results? Explain your choices.

(10 points)

For this problem I would first pass the transducer signal through a low pass filter (with as many poles as possible and afforded such that there is a steep cutoff beyond the -3-dB cutoff point.) with a cutoff frequency of approximately 100 Hz. I would then pass the signal through an amplifier with a gain of 100 such that a signal of -0.1 volts is amplified to negative 10 volts and a signal of 0.1 volts is amplified to positive 10 volts.

10

Name: Matthew Van Eps

Dordt University Engineering Department

EGR 304, Microprocessor Interfacing — Final Exam, May 4, 2017. 93/100 = A

Open book, open notes, take-home.

An Internet-connected computer is required. A calculator is also required.

1.) A small local-area computer network (e.g. a network of two or three computers sharing files, printers, etc.—no Internet connection) could be made using USB connections or using Ethernet connections between the computers. Describe some of the challenges and advantages that would be encountered when using a USB connection for networking as opposed to using an Ethernet connection. How are these challenges rooted in the technological differences of the two connection types? (10 points)

There are other things that could be mentioned in your answer, but two good distinctions are good for full credit.

USB use generic class drivers for any hardware. Because some of the class drivers that come with OS are poor quality, USB devices often include replacement drivers, but often offer special features to only one brand of product. This sometimes results in difficulties in installation, but often USB drivers usually install instantaneously. Ethernet drivers typically need to be installed manually which can be a pain. USB is not meant for long distance transfer. It is limited to a max of about 15 feet. It was never designed for long distance connections. In order to span a long way, many hubs will need to be used to amplify the signal. The limitation is due to the twisted pairs that are used to transmit the data. Another limitation and annoyance of USB would be the multiple different types of connectors. One must be sure to choose the correct ones for the devices. An advantage of the USB would be the power that it provides alongside of the twisted pair. USB is a great choice for powering a device along with data. USB connectors also enforce network topology and are very easy to use. Along with ethernet, "hot-plugging" is acceptable. USB will not crash the OS when removed or inserted. USB uses device enumeration to configure the device and receive an address which is very convenient. Ethernet would be advantageous because it can be used over long distances. Ethernet can also be used to connect via a LAN through a hub or switch that allows the computers and devices to communicate with each other. When using USB to connect multiple devices on the same port, if too much data is being transferred through a single port, the devices may compete with each other whereas an ethernet connected system will only be as fast as its slowest device.

10

2.) A thirty-foot long asynchronous serial computer interface uses RS-232-style signaling but at "TTL" logic levels meaning 0 volts for logic-1 and idle and mark or 5 volts for logic-0 or space, nominal levels. The interface is in practically continuous use. (There are no significant periods of idle time.) About half of the time the transmission line is sending logic-1 and the other half of the time it is sending logic-0. A low-impedance driver is connected to a 50 Ω transmission line and a 50 Ω matched load is used to prevent reflections. It works fine but battery life is poor due to all the current flowing in the load. Suggest a change or changes to the source and/or load impedances that will improve battery life. (10 points)

5

Yes

The best way to improve the line would be to match the source to the line impedance and the load impedance. In this case, the ideal source impedance should be increased to 50Ω. They do not need to be at 50 so long as all three are the same value of impedance. This will ensure that there is no reflection along the line at either the source or load, and thus maximum power will be transferred with the least amount of loss. It would be ideal if all of the impedances were purely resistive but in all reality that is hard to achieve. Seeing as the more resistance, the more energy is dissipated, it would be beneficial to lower all three resistances of the load, source, and line.

X

Maximum power is not consistent with extending battery life.

3.) A DC motor with a permanent-magnet field, when connected to a 12 V DC source has a no-load speed of 10 000 RPM and a stall (locked rotor) torque (a.k.a. locked rotor torque) of 0.1 Nm. Assume the motor has an ideal linear torque vs. speed characteristic. This motor is connected to a low-impedance 12 V source on the electrical side and to a mechanical load that requires a driving torque that is related to the speed such that

$$T = \sqrt{S}\left(\frac{1}{2500}\text{Nm/RPM}^{0.5}\right)$$

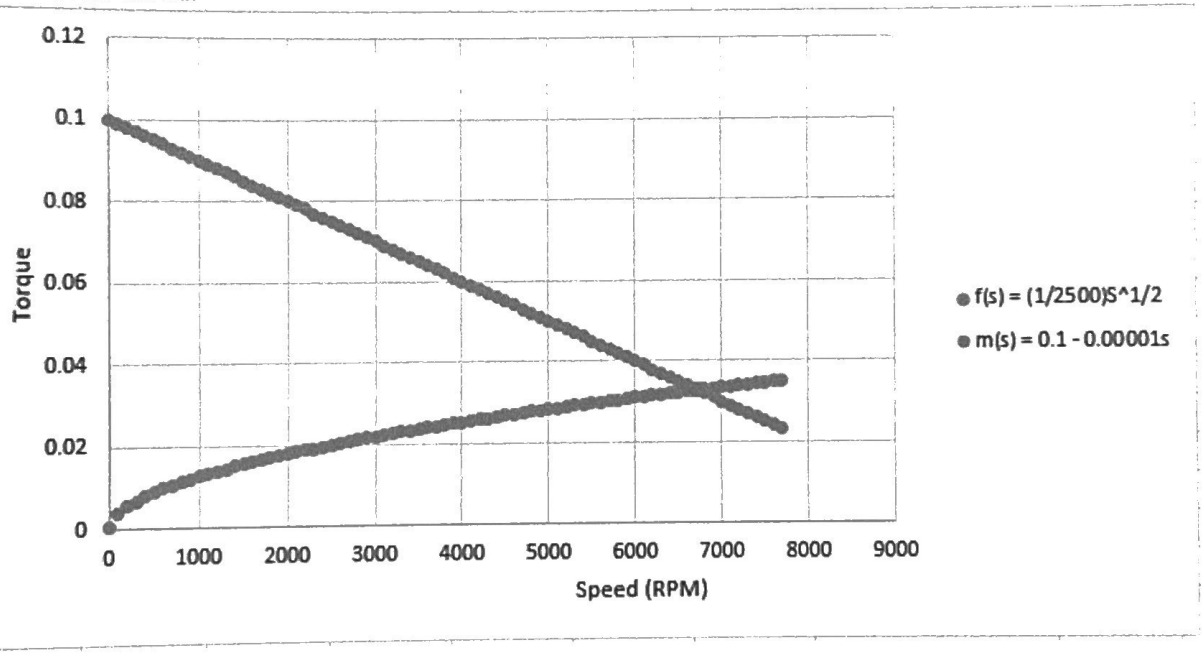
Where T = required driving torque (in Nm)
S = load speed (in RPM)

Find the speed at which the motor will operate. (10 points)

0.1 Nm / 10000 RPM = 0.00001
Tm(s) = 0.1 – 0.00001s
Tf(s) = (1/2500) s^1/2

10

From the Excel graph and by comparing values in the rows of excel, we find that the motor will operate at about 6725 RPM.



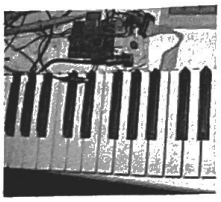
Tm(s)	S(RPM)	Tf(s)
0.033	6700	0.032741
0.03275	6725	0.032802
0.0325	6750	0.032863

4.) Describe in words or pseudo-code how to use an interrupt driven tic-clock system with a microcontroller to generate a pulse of exactly 10 ms each time a certain subroutine is called. Assume the tic-clock interrupt has a frequency of 10 kHz. (10 points)

I would use a timer interrupt scheduler library. I would create a function that turns on a pin, and then schedules a task 10ms after the time that is currently read off of the system timer. Then I could create another function inside that same interrupt. that would be triggered by the timer and that function would turn of the pin.

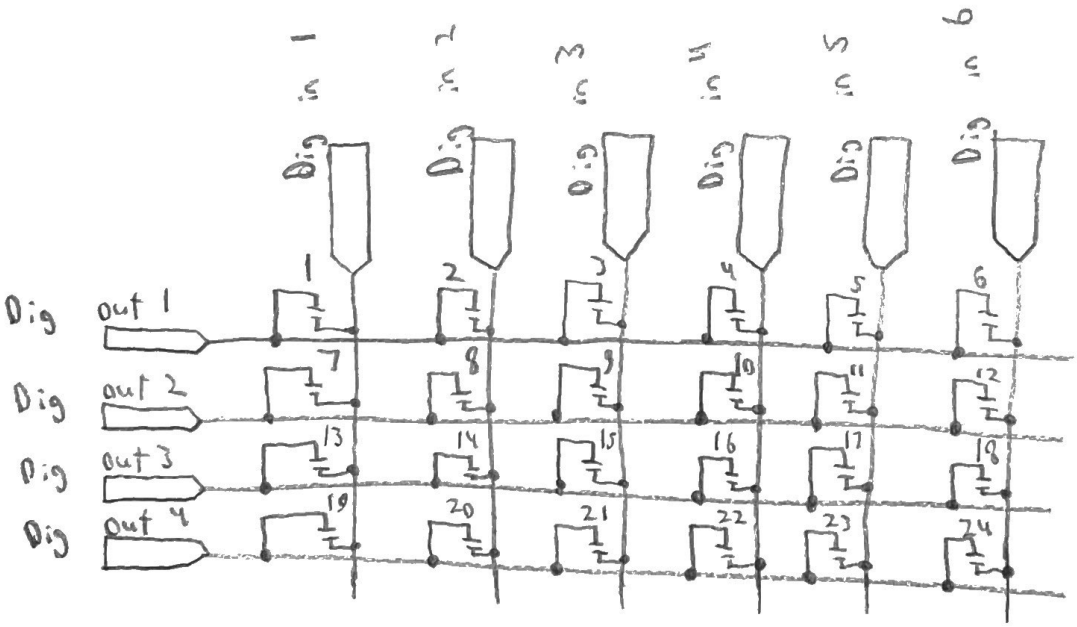
10

5.) Consider creating a keyboard-operated music synthesizer by connecting a keyboard to an Arduino microcontroller. Let the synthesizer have 24 keys (two octaves). Each key will have just one SPST switch. The keyboard has a 48-conductor ribbon cable connected to it, two conductors dedicated to each switch. Could this keyboard be connected directly to an Arduino microcontroller board using no additional parts such as diodes, gates, resistors, or transistors? The Arduino board has 13 digital I/O pins available. If so, describe how and describe any limitations of the connection. If not, give a rationale for your conclusion. (10 points)



Yes it is possible to use an arduino for this project. This assumes the arduino runs fast enough and no outside interrupts. You first start a routine to set the outputs HIGH one at a time. You then write a loop to read each of the inputs before the next output goes HIGH. This way you can tell from the output and input which key is pressed. Continue this loop.

8



The problem of simultaneous key presses should be addressed.

6.) What is the difference between vectored interrupts and polled interrupts? (10 points)

Both interrupts are caused by a hardware device signaling for attention via an interrupt request. A vectored interrupt is an interrupt that tells the computer that handles the interrupts that a request came from an I/O device has been received and also tells which device or where that request came from. It will give an address where the interrupt came from.

A polled interrupt is an interrupt that tells computer a request has been filed but does not say where it came from. As a result, the handler must send out a poll or signal to each device to find out where the request cam from. It is much less efficient that vectored interrupts seeing as it has to send a signal to each device to see which one had the interrupt.

10

7.) A microcontroller has three sources of interrupts. Any of these can be eliminated by adding additional hardware. To keep costs down it is desirable to have the microcontroller handle as many of the interrupts as possible. Can the microprocessor handle all the interrupts below? If not, suggest which one (or ones) should be eliminated (requiring additional hardware). Try to eliminate as few interrupts as possible. Justify your answer. (15 points)

Listed from highest priority (on top) to lowest (on bottom)

Source 1) The interrupt service routine (ISR) takes 50 μs to run. Minimum time between interrupts is 100 μs. Prompt response to the interrupt request is not needed as long as the interrupt is serviced before it is requested again.

Source 2) ISR takes 10 μs. Minimum time between interrupts is 500 μs but the maximum allowable latency (time from request to end of execution of ISR) is 100 μs.

Source 3) ISR takes 5 μs. Minimum time between interrupts is 70 μs. Prompt response is not needed as long as the interrupt is serviced before it is requested again.

The longest critical region in the main software is 8 μs and the interrupts are disabled during each ISR. The longest instruction takes 1 μs.

(Please write your answer on the next page.)

(This page is provided for the answer to the problem stated on the previous page.)

Summary of the question: Will interrupts work? Eliminate as few as possible if any need to be eliminated. Justify your answer.

Source 1) ISR takes 50 µs to run, 100 µs between interrupts.
Source 2) ISR takes 10 µs to run, 500 µs between interrupts, latency < 100 µs req'd.
Source 3) ISR takes 5 µs to run, 70 µs between interrupts
Critical region is 8 µs, interrupts disabled during ISRs, longest instruction is 1 µs.

$T_1 = 50\mu s$ $T_{p1} = 100\mu s$ $T_{1+} = 10\ \mu s$
$T_2 = 10\mu s$ $T_{p2} = 500\mu s$ $T_{2+} = 8\ \mu s$
$T_3 = 5\mu s$ $T_{p3} = 70\mu s$ $T_{3+} = 8\ \mu s$

From notes, T_{i+} is defined as the longest machine instruction in code or the longest lower priority interrupt's T_i or longest critical region

Source 2 requires < 100 µs latency. Source 1 and 2 = 50µs + 5µs = 55 µs to complete so we don't have to worry about the latency.

interrupt density = $\frac{50}{100} + \frac{10}{500} + \frac{5}{70} = 0.59 < 100$ ✓

$N(i,x) = \frac{(T_{pi} - T_i)}{T_{px}}$ $N(1,1) = 1$ $N(2,1) = 5$ ✗ $N(2,2) = 1$ $N(3,1) = 1$ $N(3,2) = 1$ $N(3,3) = 1$

15

(1) $T_{1+} + N(1,1)T_1 < T_{p1}$
$10 + 1(50) < 100$
$60 < 100$ ✓

(2) $T_{2+} + N(2,2)T_2 + N(2,1)T_1 < T_{p2}$
$8 + 1(10) + 5(50) < 500$
$268 < 500$ ✓ ✗

(3) $T_{3+} + N(3,3)T_3 + N(3,2)T_2 + N(3,1)T_1 < T_{p3}$
$8 + 1(5) + 1(10) + 1(50) < 70$
$73 > 70$ ✗

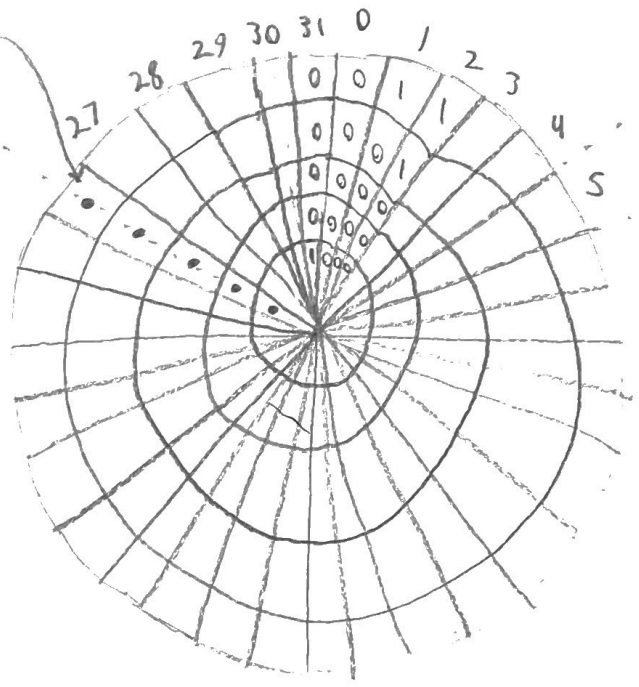
Eliminate the 3rd interrupt seeing as it is the least priority and the other two will allow the microcontroller to work if it is gone.

8.) Design a rotational position sensor for sensing ***absolute*** position with a resolution of 11.25 degrees (1/32 of a revolution). The rotational position sensor needs to only keep track of one revolution. If the shaft of the sensor should roll past the home position, the position code can roll over too. Each time the rotational position of a disk changes by 1/32 of a revolution, an interrupt should be produced. You may assume that the interrupt will be serviced before the disk moves enough to generate another interrupt request. You may decide for yourself how much logic to do in hardware and how much to do in the ISR. Your design should be relatively easy to manufacture. (15 points)

a.) Show the design of the disk and number and location of the sensor(s).

b.) Show a flow chart (or equivalent pseudo-code) to describe the processing that needs to be done in the ISR to keep track of ***absolute*** position.

5 sensors all in a line as shown



32 slices

$2^5 = 32$

5 bits needed for grey code

15

0 = 00000
1 = 00001
2 = 00011
3 = 00010
4 = 00110
5 = 00111
↓
all the way through 31

As one can see, using gray code, only one bit changes at a time so even if one sensor is off a little, the resolution stays fairly accurate. As shown to the left, a large table 0-31 decimal values with corresponding gray code would allow the computer to know the absolute position. The circle is not filled in completely for sake of time and neatness.

9. The performance of a car's shock absorber needs to be monitored to assist with the design and testing of the shock absorber. The fastest rate at which the shock absorber might effectively move to absorb the energy of bumps is 100 Hz. (100 bumps per second for example.)

   The transducer used is a linear analog position sensor. It produces a voltage proportional to the length between the two anchor points of the shock absorber. The maximum voltage is +0.100 V at full extension of the shock absorber and the minimum is –0.100 V at full compression. The transducer has a frequency response up to 10000 Hz. Thus, it is not only is sensitive to the motion of the shock absorber, it is also sensitive to some of the sound of the wheel rolling over the road and conducted into the shock-absorber.

   A commercially available analog-to-digital converter is in use. The sample rate of the converter is 1000 Hz. (One sample is taken every millisecond). This analog-to-digital converter has an input range of +10 V to –10 V and provides a 10-bit output in a two's complement format such that +10 V produces the code $01\ 1111\ 1111_{TC}$ which corresponds to $+511_{10}$ and –10 V produces the code $10\ 0000\ 0001_{TC}$ which corresponds to -511_{TC}. This converter has no anti-aliasing filter. The user is expected to add their own anti-aliasing filter.

   The user figured that the 10000 Hz frequency response of the transducer and the 1000 Hz sample rate are 100 and 10 times (respectively) more than needed since the user is only interested in motions at 100 Hz or less. Thus, these were perceived as very good specifications. (The user may be naive in this determination.) The ±10 V input range is also perceived as 100 times greater than needed so that should also be very good. (The user may be naive in this determination.) The user is getting poor results from this system.

   For an anti-aliasing filter the user has resorted to averaging 8 sequential samples (using digital addition of the sampled data, then dividing by 8) in order to get better results. This has reduced their effective sample rate to 1000/8 = 125 Hz. (Every eight digital samples get averaged to produce just one average sample for the interval.) However, the samples still are not accurate.

   Assume that averaging digital samples must be abandoned as it is an incorrect method of addressing the problems here. Assume analog filters and amplifiers are available to any specifications you might like. For a filter, specify if it should pass low or high frequencies and what the cutoff frequency should be. For an amplifier, specify the gain. How can you use filters and amplifiers to improve the results? Explain your choices. (10 points)

Write, "-0.1 V" (no leading decimals in engineering)

The scope of this is horribly off. You would never use a -10V to 10V converter to sample a -.1V to .1V transducer. A vast majority of the bits in the converter are not even being used so as of right now it is a waste of money. First, I would set the gain to 100 in order to get the -.1V to .1V up to scale of -10V to 10V. This way almost all of the bits will be utilized in the 10-bit convertor. Also, I would set up a low pass anti-aliasing filter to cut off everything over 100 Hz because in the first paragraph it is specified that the fastest rate at which the shock absorber might effectively move is 100Hz. I would use multiple polls of filters in order to make the cutoff as dramatic as possible.

10

Name: Nolan V. G.

Dordt University Engineering Department

EGR 304, Microprocessor Interfacing — Final Exam, May 4, 2017.
Open book, open notes, take-home.
An Internet-connected computer is required. A calculator is also required.

28/100 = D

Your answer does not address networking. Issues such as distance and peer-to-peer communication should be addressed.

1.) A small local-area computer network (e.g. a network of two or three computers sharing files, printers, etc.—no Internet connection) could be made using USB connections or using Ethernet connections between the computers. Describe some of the challenges and advantages that would be encountered when using a USB connection for networking as opposed to using an Ethernet connection. How are these challenges rooted in the technological differences of the two connection types? (10 points)

1

Challenges

Very bland/common signal being sent all internal + therefore can be breached more easily

Advantages

Can plug it in and the work almost automatically doesn't need to deal w/ hectic security + are direct connections

And the tech differences lay in the variability of ethernet + USB as it can have variable receiving speeds while USB holds slower speeds but more immediate connections I think

2.) A thirty-foot long asynchronous serial computer interface uses RS-232-style signaling but at "TTL" logic levels meaning 0 volts for logic-1 and idle and mark or 5 volts for logic-0 or space, nominal levels. The interface is in practically continuous use. (There are no significant periods of idle time.) About half of the time the transmission line is sending logic-1 and the other half of the time it is sending logic-0. A low-impedance driver is connected to a 50 Ω transmission line and a 50 Ω matched load is used to prevent reflections. It works fine but battery life is poor due to all the current flowing in the load. Suggest a change or changes to the source and/or load impedances that will improve battery life. (10 points)

don't know but I'll give my best shot.

2

You could potentially make some parallel connections so as to lower the amperage + lower overall current. Potentially could increase the resistances so as to maintain voltage level while reducing current.

Which resistance(s)?

Vague

3.) A DC motor with a permanent-magnet field, when connected to a 12 V DC source has a no-load speed of 10 000 RPM and a stall (locked rotor) torque (a.k.a. locked rotor torque) of 0.1 Nm. Assume the motor has an ideal linear torque vs. speed characteristic. This motor is connected to a low-impedance 12 V source on the electrical side and to a mechanical load that requires a driving torque that is related to the speed such that

0.0004

$$T = \sqrt{S}\left(\frac{1}{2500}\ \text{Nm/RPM}^{0.5}\right)$$

Where T = required driving torque (in Nm)
S = load speed (in RPM)

Find the speed at which the motor will operate. (10 points)

10

$T_m(S) = 0.1 - 0.00001\,S$

$T_s(S) = 0.0004\,S^{1/2}$

at $S = 6720.8$

they intersect so it would be 6720.8 RPM

4.) Describe in words or pseudo-code how to use an interrupt driven tic-clock system with a microcontroller to generate a pulse of exactly 10 ms each time a certain subroutine is called. Assume the tic-clock interrupt has a frequency of 10 kHz. (10 points)

1

It should trigger on the downshift of the microcontroller and then read a value of 1 for 10 percent of the total recorded frequency cycles (or after 100 interrepts have been checked for).

X

You must use as task scheduler to take advantage of a tic-clock.

5.) Consider creating a keyboard-operated music synthesizer by connecting a keyboard to an Arduino microcontroller. Let the synthesizer have 24 keys (two octaves). Each key will have just one SPST switch. The keyboard has a 48-conductor ribbon cable connected to it, two conductors dedicated to each switch. Could this keyboard be connected directly to an Arduino microcontroller board using no additional parts such as diodes, gates, resistors, or transistors? The Arduino board has 13 digital I/O pins available. If so, describe how and describe any limitations of the connection. If not, give a rationale for your conclusion. (10 points)

I believe it can be as it would be possible to have the 13th pin be on/off + while checking that value you can correspond to which other paired button has been pressed and specify the notes that way. The limitations would be the debouncing and allocating the various types of timing interrupts that would be required.

Scan a matrix.

6.) What is the difference between vectored interrupts and polled interrupts. (10 points)

a vectored interrupt also includes, for lack of another term, a call sign dictating where it came from and with a polled interrupt the computer that received the ping has to inquire as to which terminal sent it the signal in the first place.

10

7.) A microcontroller has three sources of interrupts. Any of these can be eliminated by adding additional hardware. To keep costs down it is desirable to have the microcontroller handle as many of the interrupts as possible. Can the microprocessor handle all the interrupts below? If not, suggest which one (or ones) should be eliminated (requiring additional hardware). Try to eliminate as few interrupts as possible. Justify your answer. (15 points)

Listed from highest priority (on top) to lowest (on bottom)

Source 1) The interrupt service routine (ISR) takes 50 μs to run. Minimum time between interrupts is 100 μs. Prompt response to the interrupt request is not needed as long as the interrupt is serviced before it is requested again.

✓

✓ Source 2) ISR takes 10 μs. Minimum time between interrupts is 500 μs but the maximum allowable latency (time from request to end of execution of ISR) is 100 μs.

Source 3) ISR takes 5 μs. Minimum time between interrupts is 70 μs. Prompt response is not needed as long as the interrupt is serviced before it is requested again.

The longest critical region in the main software is 8 μs and the interrupts are disabled during each ISR. The longest instruction takes 1 μs.

(Please write your answer on the next page.)

(This page is provided for the answer to the problem stated on the previous page.)

Summary of the question: Will interrupts work? Eliminate as few as possible if any need to be eliminated. Justify your answer.

Source 1) ISR takes 50 µs to run, 100 µs between interrupts.
Source 2) ISR takes 10 µs to run, 500 µs between interrupts, latency < 100 µs req'd.
Source 3) ISR takes 5 µs to run, 70 µs between interrupts
Critical region is 8 µs, interrupts disabled during ISRs, longest instruction is 1 µs.

It looks to be an okay system of interrupts as I don't see any place where they will conflict w/ another as the one it gets closest to would be Source 3 and that barely makes it unless it happens w/ the order 3, 1, 2, C and even then it looks like it should be okay.

/

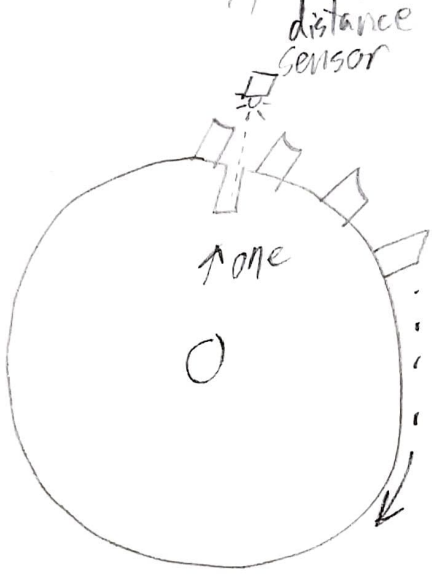
Your answer appears to be founded on speculation. You should show your work as you check interrupt density, and the interrupt interval constraint for each ISR. Had you done that you would see that ISR #3 fails. Then you should propose a fix.

8.) Design a rotational position sensor for sensing ***absolute*** position with a resolution of 11.25 degrees (1/32 of a revolution). The rotational position sensor needs to only keep track of one revolution. If the shaft of the sensor should roll past the home position, the position code can roll over too. Each time the rotational position of a disk changes by 1/32 of a revolution, an interrupt should be produced. You may assume that the interrupt will be serviced before the disk moves enough to generate another interrupt request. You may decide for yourself how much logic to do in hardware and how much to do in the ISR. Your design should be relatively easy to manufacture. (15 points)

a.) Show the design of the disk and number and location of the sensor(s).

b.) Show a flow chart (or equivalent pseudo-code) to describe the processing that needs to be done in the ISR to keep track of ***absolute*** position.

dont think I will have enough time.



will measure each cog change in position & when the value is further that is the home position

You have started describing a relative position sensor.

9. The performance of a car's shock absorber needs to be monitored to assist with the design and testing of the shock absorber. The fastest rate at which the shock absorber might effectively move to absorb the energy of bumps is 100 Hz. (100 bumps per second for example.)

The transducer used is a linear analog position sensor. It produces a voltage proportional to the length between the two anchor points of the shock absorber. The maximum voltage is +0.100 V at full extension of the shock absorber and the minimum is –0.100 V at full compression. The transducer has a frequency response up to 10000 Hz. Thus, it is not only is sensitive to the motion of the shock absorber, it is also sensitive to some of the sound of the wheel rolling over the road and conducted into the shock-absorber.

A commercially available analog-to-digital converter is in use. The sample rate of the converter is 1000 Hz. (One sample is taken every millisecond). This analog-to-digital converter has an input range of +10 V to –10 V and provides a 10-bit output in a two's complement format such that +10 V produces the code 01 1111 1111_{TC} which corresponds to $+511_{10}$ and –10 V produces the code 10 0000 0001_{TC} which corresponds to -511_{TC}. This converter has no anti-aliasing filter. The user is expected to add their own anti-aliasing filter.

The user figured that the 10000 Hz frequency response of the transducer and the 1000 Hz sample rate are 100 and 10 times (respectively) more than needed since the user is only interested in motions at 100 Hz or less. Thus, these were perceived as very good specifications. (The user may be naive in this determination.) The ±10 V input range is also perceived as 100 times greater than needed so that should also be very good. (The user may be naive in this determination.) The user is getting poor results from this system.

For an anti-aliasing filter the user has resorted to averaging 8 sequential samples (using digital addition of the sampled data, then dividing by 8) in order to get better results. This has reduced their effective sample rate to 1000/8 = 125 Hz. (Every eight digital samples get averaged to produce just one average sample for the interval.) However, the samples still are not accurate.

Assume that averaging digital samples must be abandoned as it is an incorrect method of addressing the problems here. Assume analog filters and amplifiers are available to any specifications you might like. For a filter, specify if it should pass low or high frequencies and what the cutoff frequency should be. For an amplifier, specify the gain. How can you use filters and amplifiers to improve the results? Explain your choices. (10 points)

I would personally amplify the ~~compressions~~ extensions while filtering out the compressions since any significant shocks/jolts would inflect a jump up of the frame whereas contractions can be as minor as a person jostling around in the car. I'd also probably make it a bit more tolerant as going down a hill might cause minor extensions and skew data.

/

X

Name: Kyle Waas

Dordt University Engineering Department

EGR 304, Microprocessor Interfacing — Final Exam, May 4, 2017.
Open book, open notes, take-home.
An Internet-connected computer is required. A calculator is also required.

38/100 = C

1.) A small local-area computer network (e.g. a network of two or three computers sharing files, printers, etc.—no Internet connection) could be made using USB connections or using Ethernet connections between the computers. Describe some of the challenges and advantages that would be encountered when using a USB connection for networking as opposed to using an Ethernet connection. How are these challenges rooted in the technological differences of the two connection types? (10 points)

2

Some of the advantages that one could see when using an USB connection would be the enumeration and the device class control. With enumeration the host will detect the USB and immediately start to give it an address. This will in turn let the host know what device it is. Another pro is that the USB has a universal plug but because of this it has device classes. These classes let the host know is exactly on the other end of the USB cable. The disadvantage with this is that it can be hard for manufactures to distinguish their products (a mouse is a mouse). USB also has many different kinds of transactions such as control, interrupt, Isochronous and bulk. There are also no collisions because of the slave-master relationship. This means that a USB device is only allowed to "talk" when the host asks for it to talk and even then, it has an assigned address, so the host knows exactly where it is coming from. This unlike Ethernet where there can be a lot of collisions it has to scrap the transmissions every time.

Your answer does not address networking. Issues such as distance and peer-to-peer communication should be addressed.

2.) A thirty-foot long asynchronous serial computer interface uses RS-232-style signaling but at "TTL" logic levels meaning 0 volts for logic-1 and idle and mark or 5 volts for logic-0 or space, nominal levels. The interface is in practically continuous use. (There are no significant periods of idle time.) About half of the time the transmission line is sending logic-1 and the other half of the time it is sending logic-0. A low-impedance driver is connected to a 50 Ω transmission line and a 50 Ω matched load is used to prevent reflections. It works fine but battery life is poor due to all the current flowing in the load. Suggest a change or changes to the source and/or load impedances that will improve battery life. (10 points)

2

In order to improve the battery life, one would want to lower the internal resistance. So, I would lower the resistance of the matched load.

✗

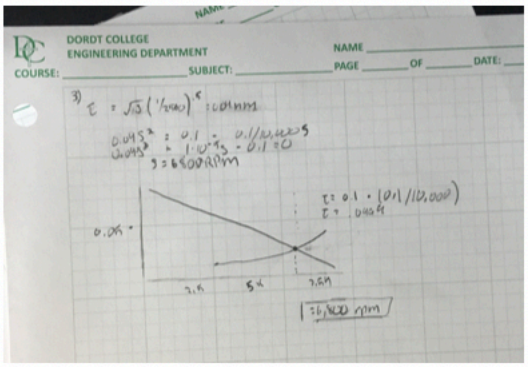
3.) A DC motor with a permanent-magnet field, when connected to a 12 V DC source has a no-load speed of 10 000 RPM and a stall (locked rotor) torque (a.k.a. locked rotor torque) of 0.1 Nm. Assume the motor has an ideal linear torque vs. speed characteristic. This motor is connected to a low-impedance 12 V source on the electrical side and to a mechanical load that requires a driving torque that is related to the speed such that

$$T = \sqrt{S}\left(\frac{1}{2500}\text{Nm/RPM}^{0.5}\right)$$

Where T = required driving torque (in Nm)
S = load speed (in RPM)

Find the speed at which the motor will operate. (10 points)

8



Not accurate enough.
The graph is not properly labeled or shaped.

4.) Describe in words or pseudo-code how to use an interrupt driven tic-clock system with a microcontroller to generate a pulse of exactly 10 ms each time a certain subroutine is called. Assume the tic-clock interrupt has a frequency of 10 kHz. (10 points)

One would write in the loop section of the code how at the time the frequency is 10kHz the tic -clock interrupt will be called. If the subroutine has been called, we can then check that in the tic-clock interrupt, or we can see if the subroutine is called in the interrupt itself. If the subroutine has been called and we are in the interrupt one would set the digital out of the desired pin to High. Delay for 10 ms with delay(10); and then set the digital out of the pin to low again. ✗

1

You must use a task scheduler to take advantage of a tic clock.

5.) Consider creating a keyboard-operated music synthesizer by connecting a keyboard to an Arduino microcontroller. Let the synthesizer have 24 keys (two octaves). Each key will have just one SPST switch. The keyboard has a 48-conductor ribbon cable connected to it, two conductors dedicated to each switch. Could this keyboard be connected directly to an Arduino microcontroller board using no additional parts such as diodes, gates, resistors, or transistors? The Arduino board has 13 digital I/O pins available. If so, describe how and describe any limitations of the connection. If not, give a rationale for your conclusion. (10 points)

I think that you could do it using 12 of the digital I/O pins. If one connects two octaves to each pin. The trouble with this would be how to determine which octave it is. So, you could have a read in as voltage high and one as voltage low when doing a digital read. However, with that comes the trouble of reading in the voltage low and knowing if that octave is being pressed or not. I still think this could fixed with a lot of coding. It would require a lot of testing of the other digital reads. For example if one would think we are receiving a low input then we would first test to make sure there are no high inputs. This could be done using a lot of nested if statements.

5 Vague

6.) What is the difference between vectored interrupts and polled interrupts? (10 points)

The vectored interrupt will tell the I/O interface which device has a request while the poll will tell the I/O interface that something has a request and not what it is. So, the I/O interface will then have to go each device and see if that is the one that is ready.

10

7.) A microcontroller has three sources of interrupts. Any of these can be eliminated by adding additional hardware. To keep costs down it is desirable to have the microcontroller handle as many of the interrupts as possible. Can the microprocessor handle all the interrupts below? If not, suggest which one (or ones) should be eliminated (requiring additional hardware). Try to eliminate as few interrupts as possible. Justify your answer. (15 points)

Listed from highest priority (on top) to lowest (on bottom)

Source 1) The interrupt service routine (ISR) takes 50 μs to run. Minimum time between interrupts is 100 μs. Prompt response to the interrupt request is not needed as long as the interrupt is serviced before it is requested again.

Source 2) ISR takes 10 μs. Minimum time between interrupts is 500 μs but the maximum allowable latency (time from request to end of execution of ISR) is 100 μs.

Source 3) ISR takes 5 μs. Minimum time between interrupts is 70 μs. Prompt response is not needed as long as the interrupt is serviced before it is requested again.

The longest critical region in the main software is 8 μs and the interrupts are disabled during each ISR. The longest instruction takes 1 μs.

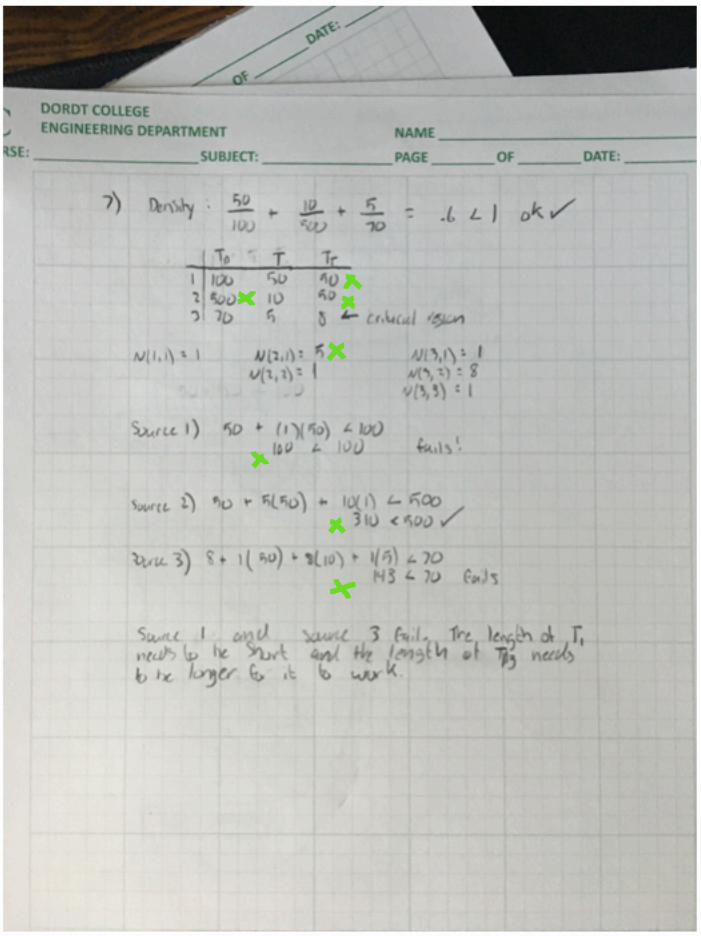
(Please write your answer on the next page.)

(This page is provided for the answer to the problem stated on the previous page.)

Summary of the question: Will interrupts work? Eliminate as few as possible if any need to be eliminated. Justify your answer.

Source 1) ISR takes 50 µs to run, 100 µs between interrupts.
Source 2) ISR takes 10 µs to run, 500 µs between interrupts, latency < 100 µs req'd
Source 3) ISR takes 5 µs to run, 70 µs between interrupts
Critical region is 8 µs, interrupts disabled during ISRs, longest instruction is 1 µs.

7



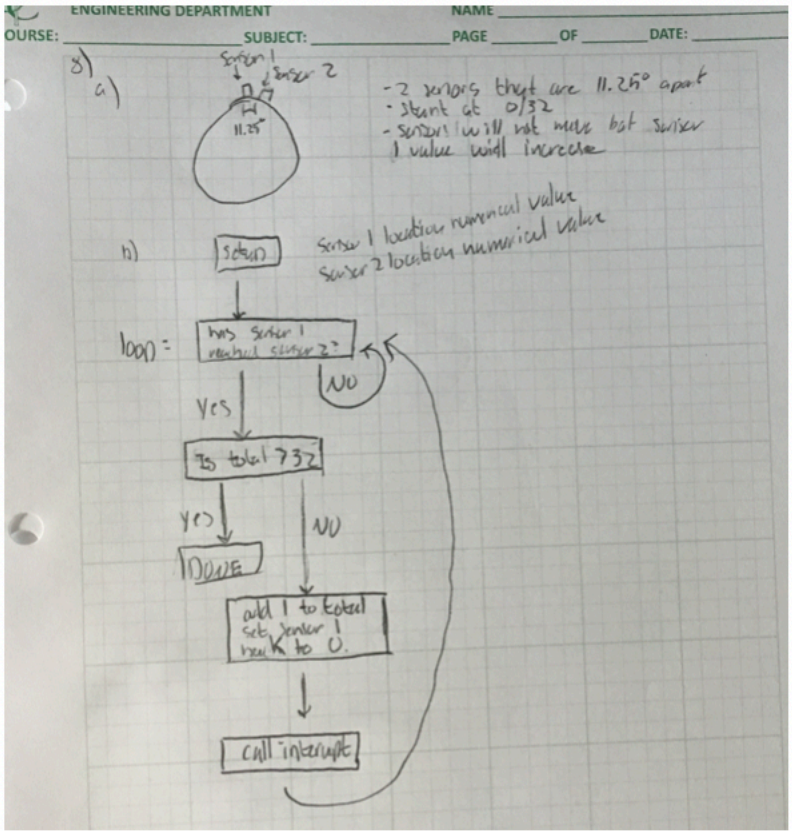
There are quite a few errors in the analysis. However the technique chosen is the one to apply.

8.) Design a rotational position sensor for sensing ***absolute*** position with a resolution of 11.25 degrees (1/32 of a revolution). The rotational position sensor needs to only keep track of one revolution. If the shaft of the sensor should roll past the home position, the position code can roll over too. Each time the rotational position of a disk changes by 1/32 of a revolution, an interrupt should be produced. You may assume that the interrupt will be serviced before the disk moves enough to generate another interrupt request. You may decide for yourself how much logic to do in hardware and how much to do in the ISR. Your design should be relatively easy to manufacture. (15 points)

a.) Show the design of the disk and number and location of the sensor(s).

b.) Show a flow chart (or equivalent pseudo-code) to describe the processing that needs to be done in the ISR to keep track of ***absolute*** position.

2



You have described a relative position encoder.
The number of teeth is unspecified. As drawn it looks wrong.

9. The performance of a car's shock absorber needs to be monitored to assist with the design and testing of the shock absorber. The fastest rate at which the shock absorber might effectively move to absorb the energy of bumps is 100 Hz. (100 bumps per second for example.)

   The transducer used is a linear analog position sensor. It produces a voltage proportional to the length between the two anchor points of the shock absorber. The maximum voltage is +0.100 V at full extension of the shock absorber and the minimum is –0.100 V at full compression. The transducer has a frequency response up to 10000 Hz. Thus, it is not only is sensitive to the motion of the shock absorber, it is also sensitive to some of the sound of the wheel rolling over the road and conducted into the shock-absorber.

   A commercially available analog-to-digital converter is in use. The sample rate of the converter is 1000 Hz. (One sample is taken every millisecond). This analog-to-digital converter has an input range of +10 V to –10 V and provides a 10-bit output in a two's complement format such that +10 V produces the code $01\ 1111\ 1111_{TC}$ which corresponds to $+511_{10}$ and –10 V produces the code $10\ 0000\ 0001_{TC}$ which corresponds to -511_{TC}. This converter has no anti-aliasing filter. The user is expected to add their own anti-aliasing filter.

   The user figured that the 10000 Hz frequency response of the transducer and the 1000 Hz sample rate are 100 and 10 times (respectively) more than needed since the user is only interested in motions at 100 Hz or less. Thus, these were perceived as very good specifications. (The user may be naive in this determination.) The $\pm$10 V input range is also perceived as 100 times greater than needed so that should also be very good. (The user may be naive in this determination.) The user is getting poor results from this system.

   For an anti-aliasing filter the user has resorted to averaging 8 sequential samples (using digital addition of the sampled data, then dividing by 8) in order to get better results. This has reduced their effective sample rate to 1000/8 = 125 Hz. (Every eight digital samples get averaged to produce just one average sample for the interval.) However, the samples still are not accurate.

   Assume that averaging digital samples must be abandoned as it is an incorrect method of addressing the problems here. Assume analog filters and amplifiers are available to any specifications you might like. For a filter, specify if it should pass low or high frequencies and what the cutoff frequency should be. For an amplifier, specify the gain. How can you use filters and amplifiers to improve the results? Explain your choices.

   (10 points)

I would first set the data to maximum specification such as the user will no longer be naïve. So, leave the 100Hz at that and 1000Hz at that. I would do this, so we are testing at the maximum possible scenario and if the samples come back and they are accurate we could then start moving down. As data is coming it is being filtered in some way. As this could possibly be getting rid of data, I would decrease the filters as you never know what that filter is taking out or trying to smooth over. I would also increase the amplifiers to make sure all data is seen. We are working with 100Hz and that can be a lot of bumps per second, so amplifying it would be important, so we do not miss anything. This may also fix the problem with the transducer. If the bumps are amplified there might be a way to make it not sensitive to the sound.

Name: Stefan Walicord

Dordt University Engineering Department

EGR 304, Microprocessor Interfacing — Final Exam, May 4, 2017.
Open book, open notes, take-home.
An Internet-connected computer is required. A calculator is also required.

$\frac{97}{100} = A$

1.) A small local-area computer network (e.g. a network of two or three computers sharing files, printers, etc.—no Internet connection) could be made using USB connections or using Ethernet connections between the computers. Describe some of the challenges and advantages that would be encountered when using a USB connection for networking as opposed to using an Ethernet connection. How are these challenges rooted in the technological differences of the two connection types? (10 points)

USB was designed to connect a peripheral devices to a master PC. Ethernet was created for networking and Peer-to-Peer style communication.

10

Challenges

- USB cables should stay below 16 feet for USB 2.0, and CAT5e for ethernet manages 100 m (328.1 ft).
- USB 2.0 bans A-to-A connectors. Using C-to-C may work, but the PCs will need two USB controllers, one to act as host and one as device. USB is not made for Peer-to-Peer.

Advantages

- USB 3.2 Gen 2x2 offers up to 20Gbps, and the more common USB 2.0 manages 480 mbps, with typical Ethernet systems only hitting 100mbps. USB 4 boasts 40 Gbps.
- USB is designed to send power, while PoE is an afterthought of sorts.

2.) A thirty-foot long asynchronous serial computer interface uses RS-232-style signaling but at "TTL" logic levels meaning 0 volts for logic-1 and idle and mark or 5 volts for logic-0 or space, nominal levels. The interface is in practically continuous use. (There are no significant periods of idle time.) About half of the time the transmission line is sending logic-1 and the other half of the time it is sending logic-0. A low-impedance driver is connected to a 50 Ω transmission line and a 50 Ω matched load is used to prevent reflections. It works fine but battery life is poor due to all the current flowing in the load. Suggest a change or changes to the source and/or load impedances that will improve battery life. (10 points)

8

$\Gamma_L = \frac{Z_L - Z_c}{Z_L + Z_c} = 0 \quad \Gamma_s = \frac{Z_s - Z_c}{Z_s + Z_c} = -1$

$Z_L = 50\,\Omega$
$Z_c = 50\,\Omega$
$Z_s = 0\,\Omega$ apparently

An important further improvement is possible. Raise RL as much as possible. Reflections will now be absorbed at the source.

If Z_s is set to 50Ω (i.e. put a resistor in the source), Γ_s would become 0 (no reflections).

Currently, $V_L = (5\,V) \cdot \frac{50}{50+0} = 5\,V$ (for example)

$V = i \cdot R = 5\,V = i \cdot 50\,\Omega$

$i = 0.1\,A$ Bad

Also, no reflections to worry about!

Now:

$V_L = (5V) \cdot \frac{50}{100} = 2.5\,V > 2$

$2.5\,V = i_{new} 50\,\Omega$

$0.05\,A = i_{new}$

50% improvement!

Still accepted by TTL!

3.) A DC motor with a permanent-magnet field, when connected to a 12 V DC source has a no-load speed of 10 000 RPM and a stall (locked rotor) torque (a.k.a. locked rotor torque) of 0.1 Nm. Assume the motor has an ideal linear torque vs. speed characteristic. This motor is connected to a low-impedance 12 V source on the electrical side and to a mechanical load that requires a driving torque that is related to the speed such that

$$T = \sqrt{S}\left(\frac{1}{2500}\text{Nm/RPM}^{0.5}\right)$$

10

Where T = required driving torque (in Nm)
S = load speed (in RPM)

Find the speed at which the motor will operate. (10 points)

$T = 0.1\,\text{Nm} - 0.00001\,\frac{\text{Nm}}{\text{RPM}} \cdot S$

$0.1 - 0.00001 \cdot S = \frac{\sqrt{S}}{2500}$

S solves to 6720 RPM

4.) Describe in words or pseudo-code how to use an interrupt driven tic-clock system with a microcontroller to generate a pulse of exactly 10 ms each time a certain subroutine is called. Assume the tic-clock interrupt has a frequency of 10 kHz. (10 points)

$10\text{ kHz} = \frac{1}{10{,}000\text{ s}} = 0.0001\,\frac{1}{s}$ so 0.1 ms btwn ticks

9

$\frac{10\text{ ms}}{0.1\text{ ms}} = 100$ ticks of the clock.

When the subroutine (SR) is called, tell the tic-clock to generate an interrupt that counts up each tic of the clock, and immediatly sent the pin/pulse high. Each 0.1 ms, the interrupt iterates up one but until it reaches 100 (assuming we start at 0 when our signal first goes high). When it reaches 100, turn the pulse low, reset the counter, and stop logging.

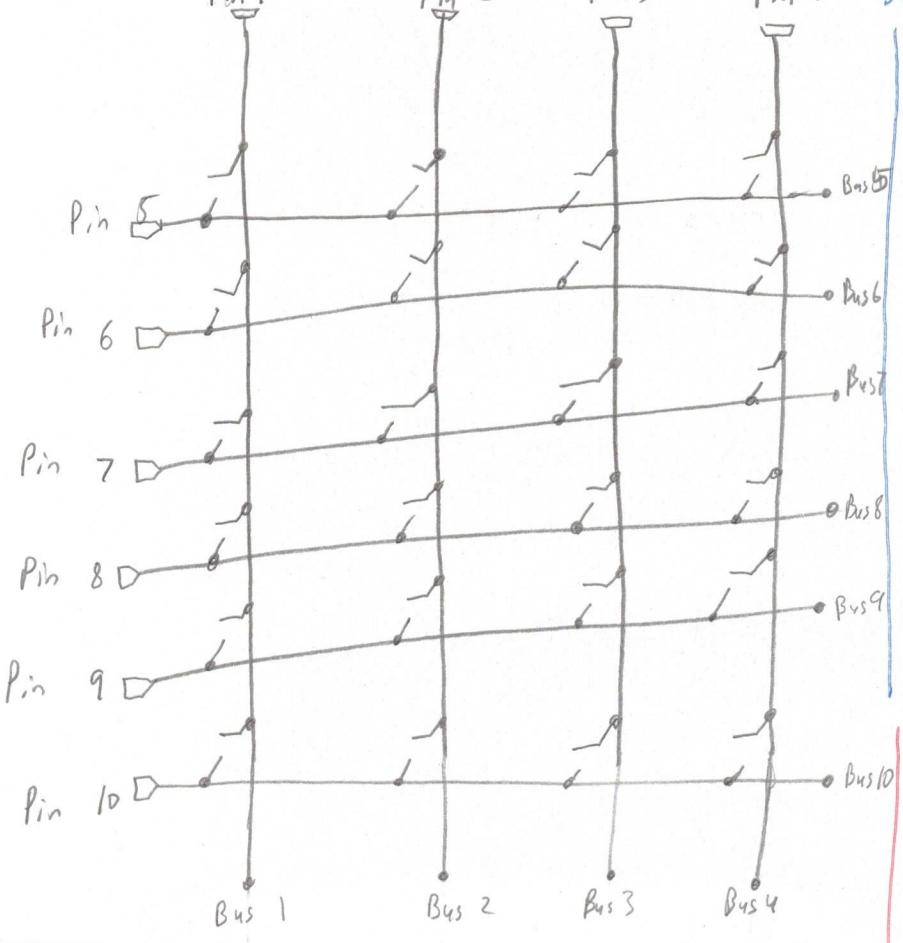
Alternatively, have the clock constantly iterate in modulo 100 and have it turn the signal low when it returns to the point the first interrupt occurs.

The description of what needs to happen is vague.

5.) Consider creating a keyboard-operated music synthesizer by connecting a keyboard to an Arduino microcontroller. Let the synthesizer have 24 keys (two octaves). Each key will have just one SPST switch. The keyboard has a 48-conductor ribbon cable connected to it, two conductors dedicated to each switch. Could this keyboard be connected directly to an Arduino microcontroller board using no additional parts such as diodes, gates, resistors, or transistors? The Arduino board has 13 digital I/O pins available. If so, describe how and describe any limitations of the connection. If not, give a rationale for your conclusion. (10 points)

It can be done, but it'll be, to use a technical term, kinda trash. :) We create a multiplexed matrix!

Introduction



Diagram

Solution

Pins 5-10 are inputs. Pins 1-4 are outputs that default to high z (impedance), aka tristate. An interrupt or loop cycles 1-4 on one at a time. If a key is pressed and the switch closes, the input pins will register it, know the current send pin, and then reconstruct which key was pressed, and then either play or record the note.

Note: you will need to run wires or connectors from the ribbons to the buses, and also connect the pins to the buses, which requires a breadboard & cables.

Bonus: 3 pins are still free.

Problems

Two biggest limitations:

> only one key per bus can be pressed at a time, or wierd current splitting effects can occur at the input pins

> Only one bus is live at a time, so if it doesn't cycle fast enough, notes can be missed

10

I judge this answer to be close enough. Actually, any two keys can be played at once and easily and properly scanned. Sophisticated software can do more than this.

6.) What is the difference between vectored interrupts and polled interrupts? (10 points)

In a polled interrupt, the device tells the computer I/O interface that a device should be serviced, and the computer polls the devices to find out which one and what to do.

Vectored interrupts identify the device sending the interrupt request by sending the vector, which points to a place in memory that identifies the correct interrupt handler.

10

7.) A microcontroller has three sources of interrupts. Any of these can be eliminated by adding additional hardware. To keep costs down it is desirable to have the microcontroller handle as many of the interrupts as possible. Can the microprocessor handle all the interrupts below? If not, suggest which one (or ones) should be eliminated (requiring additional hardware). Try to eliminate as few interrupts as possible. Justify your answer. (15 points)

Listed from highest priority (on top) to lowest (on bottom)

Source 1) The interrupt service routine (ISR) takes 50 μs to run. Minimum time between interrupts is 100 μs. Prompt response to the interrupt request is not needed as long as the interrupt is serviced before it is requested again.

Source 2) ISR takes 10 μs. Minimum time between interrupts is 500 μs but the maximum allowable latency (time from request to end of execution of ISR) is 100 μs.

Source 3) ISR takes 5 μs. Minimum time between interrupts is 70 μs. Prompt response is not needed as long as the interrupt is serviced before it is requested again.

The longest critical region in the main software is 8 μs and the interrupts are disabled during each ISR. The longest instruction takes 1 μs.

(Please write your answer on the next page.)

(This page is provided for the answer to the problem stated on the previous page.)

Summary of the question: Will interrupts work? Eliminate as few as possible if any need to be eliminated. Justify your answer.

Source 1) ISR takes 50 µs to run, 100 µs between interrupts.
Source 2) ISR takes 10 µs to run, 500 µs between interrupts, latency < 100 µs req'd. ? → T_{p2} is 100 µs for reliability
Source 3) ISR takes 5 µs to run, 70 µs between interrupts
Critical region is 8 µs, interrupts disabled during ISRs, longest instruction is 1 µs.

① $T_{p1} = 100\ \mu s$ $T_1 = 50\ \mu s$ $T_{1+} = 10\ \mu s$ $\sum_i \frac{T_i}{T_{pi}} = \frac{50}{100} + \frac{10}{500} + \frac{5}{70} = 0.591$ ✓ Interrupt density ok

② $T_{p2} = 100\ \mu s$ $T_2 = 10\ \mu s$ $T_{2+} = 8\ \mu s$

③ $T_{p3} = 70\ \mu s$ $T_3 = 5\ \mu s$ $T_{3+} = 8\ \mu s$

$$N(i,x) = \left\lceil \frac{T_{pi} - T_i}{T_{px}} \right\rceil$$

Ⓐ $T_{1+} + N(1,1)\cdot T_1 < T_{p1}$ $\quad 10 + \left\lceil \frac{100-51}{100} \right\rceil \cdot 50 < 100$ (the ceiling term = 1)

$60 < 100$ ✓

Ⓑ $T_{2+} + N(2,1)\cdot T_1 + N(2,2)\cdot T_2 < T_{p2}$ $\quad 8 + \left\lceil \frac{100-10}{100} \right\rceil \cdot 50 + \left\lceil \frac{100-10}{100} \right\rceil \cdot 10 < 100$ (each ceiling term = 1)

$68 < 100$ ✓

Ⓒ $T_{3+} + N(3,1)\cdot T_1 + N(3,2)\cdot T_2 + N(3,3)\cdot T_1 < T_{p3}$

$$8 + \left\lceil \frac{70-5}{100} \right\rceil \cdot 50 + \left\lceil \frac{70-5}{100} \right\rceil \cdot 10 + \left\lceil \frac{70-5}{70} \right\rceil \cdot 5 < 70$$

(each ceiling term = 1)

50 too Big; 70 too small

$73 < 73$ ✗

Interval C fails

15

This system does not work.

T_1 is too long and T_{p3} is quite often, so either could be removed and it would work. Likely, removing ISR 2 would work too!

I think ISR 1 should be cut. It happens fairly often and takes a very long time to execute. A quick inspection of the math above shows the vast majority of time is taken up by its 50 µs runtime, and removing that might even allow room for 1 or 2 more short interrupts!

A better solution:
Observe that ISR 1 has margin, thus does not need priority. Demote it to lowest priority and everythng will work.

8.) Design a rotational position sensor for sensing ***absolute*** position with a resolution of 11.25 degrees (1/32 of a revolution). The rotational position sensor needs to only keep track of one revolution. If the shaft of the sensor should roll past the home position, the position code can roll over too. Each time the rotational position of a disk changes by 1/32 of a revolution, an interrupt should be produced. You may assume that the interrupt will be serviced before the disk moves enough to generate another interrupt request. You may decide for yourself how much logic to do in hardware and how much to do in the ISR. Your design should be relatively easy to manufacture. (15 points)

15

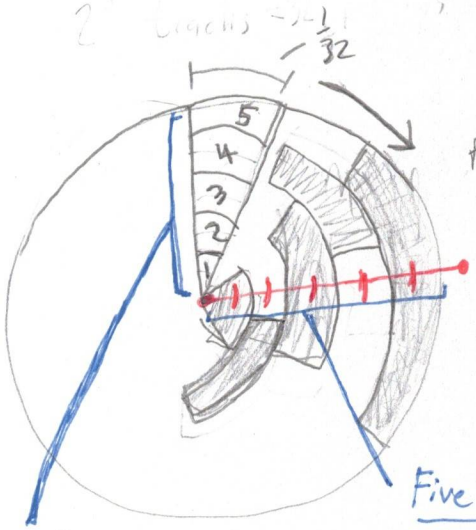
a.) Show the design of the disk and number and location of the sensor(s).

b.) Show a flow chart (or equivalent pseudo-code) to describe the processing that needs to be done in the ISR to keep track of ***absolute*** position.

(a

$\frac{1}{32}$ of a revolution => 32 sections. Design plan: Absolute position greycode optical rotary shaft encoder

$2^5 = 32$ positions, so 5 integers.



$00000_2 \rightarrow 11111_2$ normal code

Positions 0 through 31

$00000_2 - 10000_2$ grey code → so wrong is still close

Note that the numbers of least significance should be inside toward the middle to minimize incorrect readings

Five sensors in a line

Five lines of grey code rotary surfaces

codes shown only to demonstrate.

(b Sensor values can be directly called from a lookup table that contains grey code or a similar tactic,

table

Grey Code	Position
00000	→ 0
00001	→ 1
00011	→ 2
00010	→ 3

etc, you get the idea

We can multiplex the raw inputs to get the correct position value if we do it in hardware.

9. The performance of a car's shock absorber needs to be monitored to assist with the design and testing of the shock absorber. The fastest rate at which the shock absorber might effectively move to absorb the energy of bumps is 100 Hz. (100 bumps per second for example.)

Weird frequencies

The transducer used is a linear analog position sensor. It produces a voltage proportional to the length between the two anchor points of the shock absorber. The maximum voltage is +0.100 V at full extension of the shock absorber and the minimum is –0.100 V at full compression. The transducer has a frequency response up to 10000 Hz. Thus, it is not only is sensitive to the motion of the shock absorber, it is also sensitive to some of the sound of the wheel rolling over the road and conducted into the shock-absorber.

Voltage mismatch

A commercially available analog-to-digital converter is in use. The sample rate of the converter is 1000 Hz. (One sample is taken every millisecond). This analog-to-digital converter has an input range of +10 V to –10 V and provides a 10-bit output in a two's complement format such that +10 V produces the code 01 1111 1111_{TC} which corresponds to $+511_{10}$ and –10 V produces the code 10 0000 0001_{TC} which corresponds to -511_{TC}. This converter has no anti-aliasing filter. The user is expected to add their own anti-aliasing filter.

perhaps a bit Naive

The user figured that the 10000 Hz frequency response of the transducer and the 1000 Hz sample rate are 100 and 10 times (respectively) more than needed since the user is only interested in motions at 100 Hz or less. Thus, these were perceived as very good specifications. (The user may be naive in this determination.) The ±10 V input range is also perceived as 100 times greater than needed so that should also be very good. (The user may be naive in this determination.) The user is getting poor results from this system.

X Bad

For an anti-aliasing filter the user has resorted to averaging 8 sequential samples (using digital addition of the sampled data, then dividing by 8) in order to get better results. This has reduced their effective sample rate to 1000/8 = 125 Hz. (Every eight digital samples get averaged to produce just one average sample for the interval.) However, the samples still are not accurate.

Assume that averaging digital samples must be abandoned as it is an incorrect method of addressing the problems here. Assume analog filters and amplifiers are available to any specifications you might like. For a filter, specify if it should pass low or high frequencies and what the cutoff frequency should be. For an amplifier, specify the gain. How can you use filters and amplifiers to improve the results? Explain your choices. (10 points)

You will need an analog antialiasing filter and a digital filter. X

$f_{nyquist} > 1.1 \cdot 100\ Hz$, so filter out all frequencies over $f_n = 110$ Hz and passes those below

$f_{nyquist} \cdot 2 < f_s = 1000\ Hz$ so that checks out; digital noise should also be filtered too (digitally) to avoid oversampling noise.

Amplify the transducer voltage ~×100 so +0.1 V becomes +10 V and –0.1 V becomes –10 V so that quantization doesn't lose so much potential precision. ×100 may be unnecessary, even ×80 should do (leave a margin on both ends for safety).

10

Name: Ty White

Dordt University Engineering Department
EGR 304, Microprocessor Interfacing — Final Exam, May 4, 2017.
Open book, open notes, take-home.
An Internet-connected computer is required. A calculator is also required.

$\frac{77}{100}$ = A-

1.) A small local-area computer network (e.g. a network of two or three computers sharing files, printers, etc.—no Internet connection) could be made using USB connections or using Ethernet connections between the computers. Describe some of the challenges and advantages that would be encountered when using a USB connection for networking as opposed to using an Ethernet connection. How are these challenges rooted in the technological differences of the two connection types? (10 points)

10

For the sake of comparison, I will be referring to USB 3.0 for speed and what-not.

1) Ethernet
   a. Advantages
      i. No one computer can hog the serial bus all of the time.
      ii. Ethernet is a widely adopted protocol between devices while USB 3.0 is up and coming.
      iii. Can be converted to wireless signals like Wi-Fi very easily.
      iv. Only one style of plug (male or female).
      v. If internet is required in the future, it can easily attach to LAN and WAN sides.
   b. Disadvantages
      i. Slower speeds than USB 3.0
      ii. Only transfers data, no power This has changed. "PoE" is available.
      iii. Usually only one port per device and a hub will be required (extra hardware = more $$$)
      iv. Network will only be as fast as its slowest device
2) USB 3.0
   a. Advantages
      i. More power can be transferred along with data.
      ii. Quicker file transfer with different protocols to transfer them.
      iii. Usually more than one port available per device.
   b. Disadvantages
      i. Distance is limited.
      ii. Data can be bottlenecked as only one device can talk at a time with little limit to length.
      iii. Many types of cable connecters (micro, mini, type A, type B; all in male or female versions).
      iv. Bandwidth is distributed among all devices

The advantages and disadvantages stems from their protocols and purposes. Ethernet was designed before internet for networking. USB was designed to standardize drivers and interfacing for different devices. While both can be used, an ethernet network would work better and more reliably.

2.) A thirty-foot long asynchronous serial computer interface uses RS-232-style signaling but at "TTL" logic levels meaning 0 volts for logic-1 and idle and mark or 5 volts for logic-0 or space, nominal levels. The interface is in practically continuous use. (There are no significant periods of idle time.) About half of the time the transmission line is sending logic-1 and the other half of the time it is sending logic-0. A low-impedance driver is connected to a 50 Ω transmission line and a 50 Ω matched load is used to prevent reflections. It works fine but battery life is poor due to all the current flowing in the load. Suggest a change or changes to the source and/or load impedances that will improve battery life. (10 points)

8

Battery life depends on the amount of power. The amount of power delivered depends on the resistance of the circuit. Because we want to extend battery life, power needs to be reduced, meaning current needs to be reduced. Ultimately, to get a longer battery life, resistance needs to increase since current and resistance are inversely proportionate.

For the cabling to work correctly and to minimize reflections noticeable to the load, the transmission line impedance must match the load resistance. Therefore, we must increase the impedance of the transmission line and the load, such that they match. To do this, a different RS-232 cable will be required, and a resistor must be added in series with the load to match this new impedance.

While this is just one solution, other solutions may involve changing the source impedance or using a different style of communication (USB or Ethernet). While these are other changes, the one listed earlier will require the least amount of change to the system

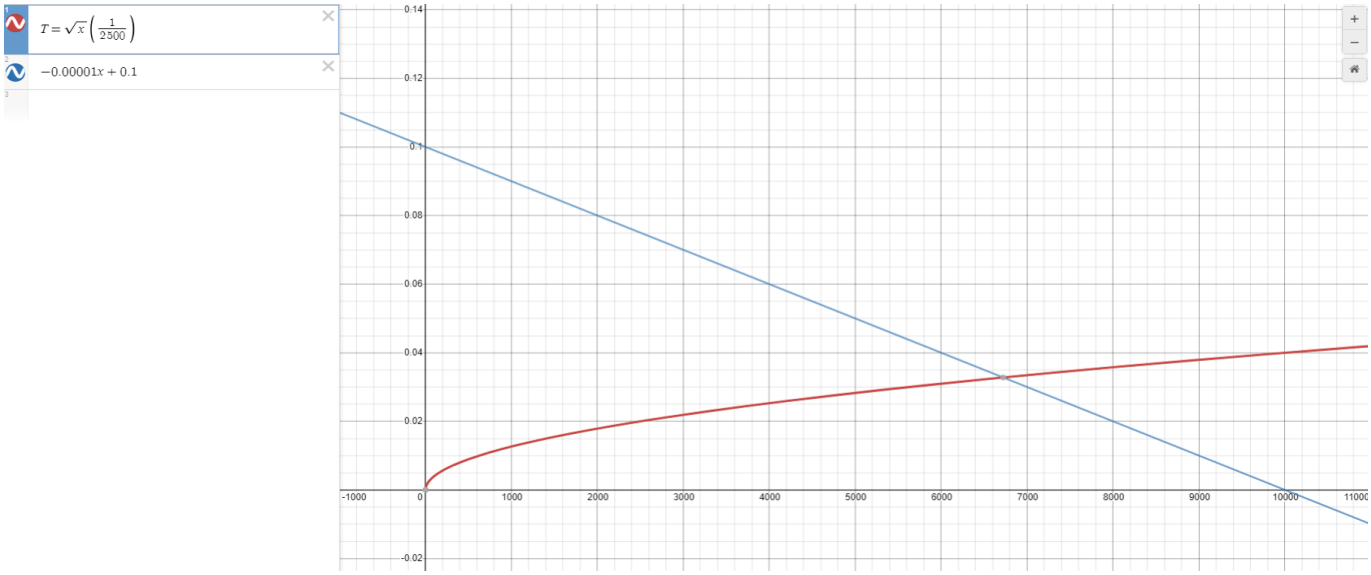
Changing the transmission line is usually expensive. (It may be already installed.) In any case, impedances above a few hundred ohms are very expensive, even impractical. The best solutions involve allowing reflections at the load but absorbing them well at the source.

3.) A DC motor with a permanent-magnet field, when connected to a 12 V DC source has a no-load speed of 10 000 RPM and a stall (locked rotor) torque (a.k.a. locked rotor torque) of 0.1 Nm. Assume the motor has an ideal linear torque vs. speed characteristic. This motor is connected to a low-impedance 12 V source on the electrical side and to a mechanical load that requires a driving torque that is related to the speed such that

$$T = \sqrt{S}\left(\frac{1}{2500}\,\text{Nm}/\text{RPM}^{0.5}\right)$$

Where T = required driving torque (in Nm)
S = load speed (in RPM)

Find the speed at which the motor will operate. (10 points)



Using the above graph, the X-axis is the speed in RPM while the Y-axis is the torque in Nm. The blue line represents the linear relationship of the motor. It has a no-load speed of 10000RPM and a locked torque of 0.1Nm. The red line shows the Torque to Speed relationship of the load.

To find the operating point, we need to find the intersection.

$$0.1 - 0.00001x = \sqrt{x} * \left(\frac{1}{2500}\right)$$

$$(0.1 - 0.00001x)^2 = \left(0.0004\sqrt{x}\right)^2$$

$$0.01 - 0.000002x + 0.0000000001x^2 = 0.00000016x$$

$$0.01 - 0.000002x - 0.00000016x + 0.0000000001x^2 = 0$$

$$0.01 - 0.00000216x + 0.0000000001x^2 = 0$$

$$roots = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-(-0.00000216) \pm \sqrt{(-0.00000216)^2 - 4(0.0000000001)(0.01)}}{2(0.0000000001)}$$

$$roots = 14879.2RPM\,, \qquad 6720.78RPM$$

Since our motor only gets up to 10000RPM, we can toss the 14879.2RPM speed and know that our motor is operating at 6720.78RPM

4.) Describe in words or pseudo-code how to use an interrupt driven tic-clock system with a microcontroller to generate a pulse of exactly 10ms each time a certain subroutine is called. Assume the tic-clock interrupt has a frequency of 10 kHz. (10 points)

$$f = 10kHz$$

$$T = \frac{1}{f} = \frac{1}{10000Hz} = 0.1ms$$

$$\#ofpulses = \frac{10ms}{T} = \frac{10ms}{0.1ms} = 100\ pulses$$

To get a pulse of 10ms, we need to count the interrupt 100 times. On the 100th time, switch the pulse from LOW to HIGH or vice versa. See below for code

```
int led_pin = 13;
int inter_pin = 2;
int counter = 0;
volatile int pulse10ms = LOW;

void mains_isr() {
   counter++;
   if(counter == 100) {
      counter = 0;
      if(pulse10ms == LOW) {
            pulse10ms = HIGH;
      else{
            pulse10ms = LOW;
      }
   digitalWrite(led_pin, pulse10ms);
}

void setup() {
   pinMode(led_pin, OUTPUT);
   digitalWrite(inter_pin, LOW);
   attachInterrupt(digitalPinToInterrupt(inter_pin), mains_isr, RISING);
}

void loop() {
   //do nothing
}
```

9

This would generate a 50 Hz square wave (100 ms high, 100 ms low.) It would not generate a 100 ms long pulse at the command of a subroutine call. A task scheduler is the best way to do what was requested.

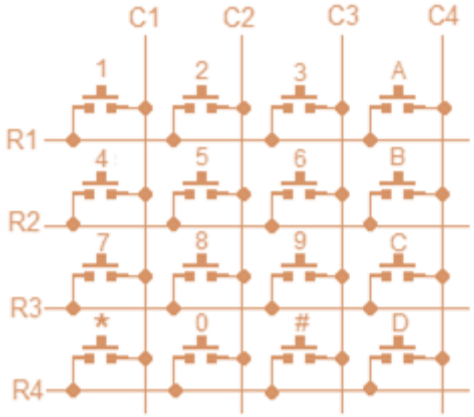
5.) Consider creating a keyboard-operated music synthesizer by connecting a keyboard to an Arduino microcontroller. Let the synthesizer have 24 keys (two octaves). Each key will have just one SPST switch. The keyboard has a 48-conductor ribbon cable connected to it, two conductors dedicated to each switch. Could this keyboard be connected directly to an Arduino microcontroller board using no additional parts such as diodes, gates, resistors, or transistors? The Arduino board has 13 digital I/O pins available. If so, describe how and describe any limitations of the connection. If not, give a rationale for your conclusion. (10 points)

Yes, this keyboard can be connected to an Arduino. The keyboard requires ten connections to the Arduino Uno I/O ports. You will need to connect the switches into a switch matrix (see picture below for an example of a 16-button matrix). The array for our keyboard will have 4 rows and 6 columns. The columns of the keyboard should be configured as inputs with pull-up resistors enabled. The program then will perform a matrix scan, that is, scan the six inputs to show if a button is pressed on the active row.

To explain further, imagine a button is pressed. Starting the scan at the first column, the only way it will return a pressed button (a LOW signal), is if the row was active. If it was low, the program knows which row was active and which column received the signal, so that button must have been pressed. To prevent debouncing, the program must delay for a few milliseconds (think 15-20ms), then re-read the same input again. If it is still LOW, the button is still pressed. Once it returns to HIGH, then the scan can continue. This prevents erroneous processing by only allowing one button to be pressed at a time.

Now imagine this time, no button is pressed. The program will scan through the first column and detect a HIGH signal. This means no button is pressed, so it will begin a scan of the second column. This will continue to repeat until a LOW signal is detected. If no LOW signal is detected by the time is finishes its scan of the last column, the process begins again at the first column. Whenever a LOW signal is detected, the program will return the correct keyboard note, go through the debounce and key-release process as described above, and continue scanning.

8



The problem of simultaneous key presses should be addressed.

6.) What is the difference between vectored interrupts and polled interrupts. (10 points)

Vectored interrupts have a specific location address within the memory which will handle the interrupt service routine capable of dealing with interrupts. This is like a doorbell on the front door. Once the ring goes off, the head of the household knows in his memory to go check the door and take care of the problem.

Polled interrupts are used when the device does not have this capability and dedicated memory space. In polled interrupts, the software must scan or poll each possible interrupt for service requests. This polling must happen regularly as to not miss out on a request. This is like checking the front door of your house to make sure there is no one there.

Regardless of the interrupt type used, both require quick response to ensure no requests are lost or drowned out.

7

With polled interrupts a device may request service but the CPU is not told which device requested service. The CPU only polls in response to a request. It does not poll to detect a request, and ISR handles that.

7.) A microcontroller has three sources of interrupts. Any of these can be eliminated by adding additional hardware. To keep costs down it is desirable to have the microcontroller handle as many of the interrupts as possible. Can the microprocessor handle all the interrupts below? If not, suggest which one (or ones) should be eliminated (requiring additional hardware). Try to eliminate as few interrupts as possible. Justify your answer. (15 points)

Listed from highest priority (on top) to lowest (on bottom)

Source 1) The interrupt service routine (ISR) takes 50 μs to run. Minimum time between interrupts is 100 μs. Prompt response to the interrupt request is not needed as long as the interrupt is serviced before it is requested again.

Source 2) ISR takes 10 μs. Minimum time between interrupts is 500 μs but the maximum allowable latency (time from request to end of execution of ISR) is 100 μs.

Source 3) ISR takes 5 μs. Minimum time between interrupts is 70 μs. Prompt response is not needed as long as the interrupt is serviced before it is requested again.

The longest critical region in the main software is 8 μs and the interrupts are disabled during each ISR. The longest instruction takes 1 μs.

(Please write your answer on the next page.)

(This page is provided for the answer to the problem stated on the previous page.)

Summary of the question: Will interrupts work? Eliminate as few as possible if any need to be eliminated. Justify your answer.

Source 1) ISR takes 50 µs to run, 100 µs between interrupts.
Source 2) ISR takes 10 µs to run, 500 µs between interrupts, latency < 100 µs req'd.
Source 3) ISR takes 5 µs to run, 70 µs between interrupts
Critical region is 8 µs, interrupts disabled during ISRs, longest instruction is 1 µs.

Time between same interrupt	Time to run interrupt	Max time delay from other interrupts
$T_{P1} = 100\mu s$	$T_1 = 50\mu s$	$T_{1t} = 10\mu s$
$T_{P2} = 500\mu s$	$T_2 = 10\mu s$	$T_{2t} = 8\mu s$
$T_{P3} = 70\mu s$	$T_3 = 5\mu s$	$T_{3t} = 8\mu s$

a. $T_{1t} + N(1,1)T_1 < T_{P1} \rightarrow$
$10\mu s + (1)50\mu s < 100 \rightarrow 60\mu s < 100$
OKAY!

b. $T_{2t} + N(2,1)T_1 + N(2,2)T_2 < T_{P2} \rightarrow$
$8\mu s + \left(\frac{500\mu s - 10\mu s}{100\mu s}\right)50\mu s + (1)10\mu s < 500\mu s \rightarrow 263\mu s < 500\mu s$
OKAY, but unfortunately it has a latency of $263\mu s$ which is more than the allowable $100\mu s$ which is NOT OKAY. But the calculation is not testing 100 us latency.

c. $T_{3t} + N(3,1)T_1 + N(3,2)T_2 + N(3,3)T_3 < T_{P3} \rightarrow$
$8\mu s + \left(\frac{70\mu s - 5\mu s}{100\mu s}\right)50\mu s + \left(\frac{70\mu s - 5\mu s}{500\mu s}\right)10\mu s + (1)5\mu s < 70\mu s \rightarrow 46.8\mu s < 70\mu s$
50 + stuff = 46.8?
Bad math here.
OKAY

To make these interrupts allowable, there are three solutions.
1) Remove the $100\mu s$ max latency on interrupt 2.
2) Reduce the time to run T_1 to $\frac{100\mu s - 18\mu s}{\left(\frac{500\mu s - 10\mu s}{100\mu s}\right)} = 16\mu s$, or increase T_{P1} to $\frac{500\mu s - 10\mu s}{\left(\frac{100\mu s - 18\mu s}{50\mu s}\right)} = 299\mu s$ or some combination of the two. If this cannot be done,
3) Interrupt 2 must be eliminated.

13

Correct technique.
A number of errors in execution.

8.) Design a rotational position sensor for sensing ***absolute*** position with a resolution of 11.25 degrees (1/32 of a revolution). The rotational position sensor needs to only keep track of one revolution. If the shaft of the sensor should roll past the home position, the position code can roll over too. Each time the rotational position of a disk changes by 1/32 of a revolution, an interrupt should be produced. You may assume that the interrupt will be serviced before the disk moves enough to generate another interrupt request. You may decide for yourself how much logic to do in hardware and how much to do in the ISR. Your design should be relatively easy to manufacture. (15 points)

a.) Show the design of the disk and number and location of the sensor(s).

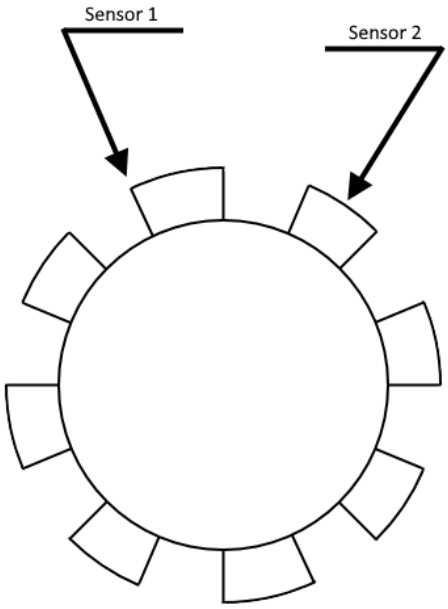
b.) Show a flow chart (or equivalent pseudo-code) to describe the processing that needs to be done in the ISR to keep track of ***absolute*** position.

a. Using a quadrature encoder design, I will need two sensors. To get the resolution, I need a certain number of teeth. Using simple math:

$Resolution = \frac{360deg}{4*n}$ where $n = number\ of\ teeth$

To get $11.25deg$ of resolution, $n = \frac{360deg}{4*resolution} = \frac{360deg}{4*11.25\text{deg}} = 8\ teeth$

$Teeth\ Width = \frac{360deg}{2*n} = \frac{360deg}{16} = 22.5deg$ wide with $22.5deg$ between teeth. See diagram below for placement of the sensors on the 8 teeth.



4

The location of Sensor 1 is used as the defined read line. Sensor 2 must be located $45deg + 22.5deg$ away from Sensor 1. Set up each sensor so that it will generate an interrupt in the microcontroller on each rising and falling edge. When a rising or falling edge occurs on sensor 1, sensor 2 will be in the middle of a tooth. Or gap Similarly, when a rising or falling edge occurs on sensor 2, sensor 1 will be in the middle of a tooth or gap. This way, when the encoder spins, an interrupt will occur on either sensor for each 11.25 degrees of rotation.

This is a good description of a quadrature position sensor, but that is a type of relative positions encoder.

b. The ISR for each sensor must sample each sensor before the wheel can turn half of a tooth (11.25 degrees). Suppose a sensor reads a HIGH signal while over a tooth.

   a. Sensor 1 ISR:

      i. Sensor 1 = Sensor 2 if the wheel is rotating counterclockwise. To keep track of position, a counter can be used. In this case, increment the counter

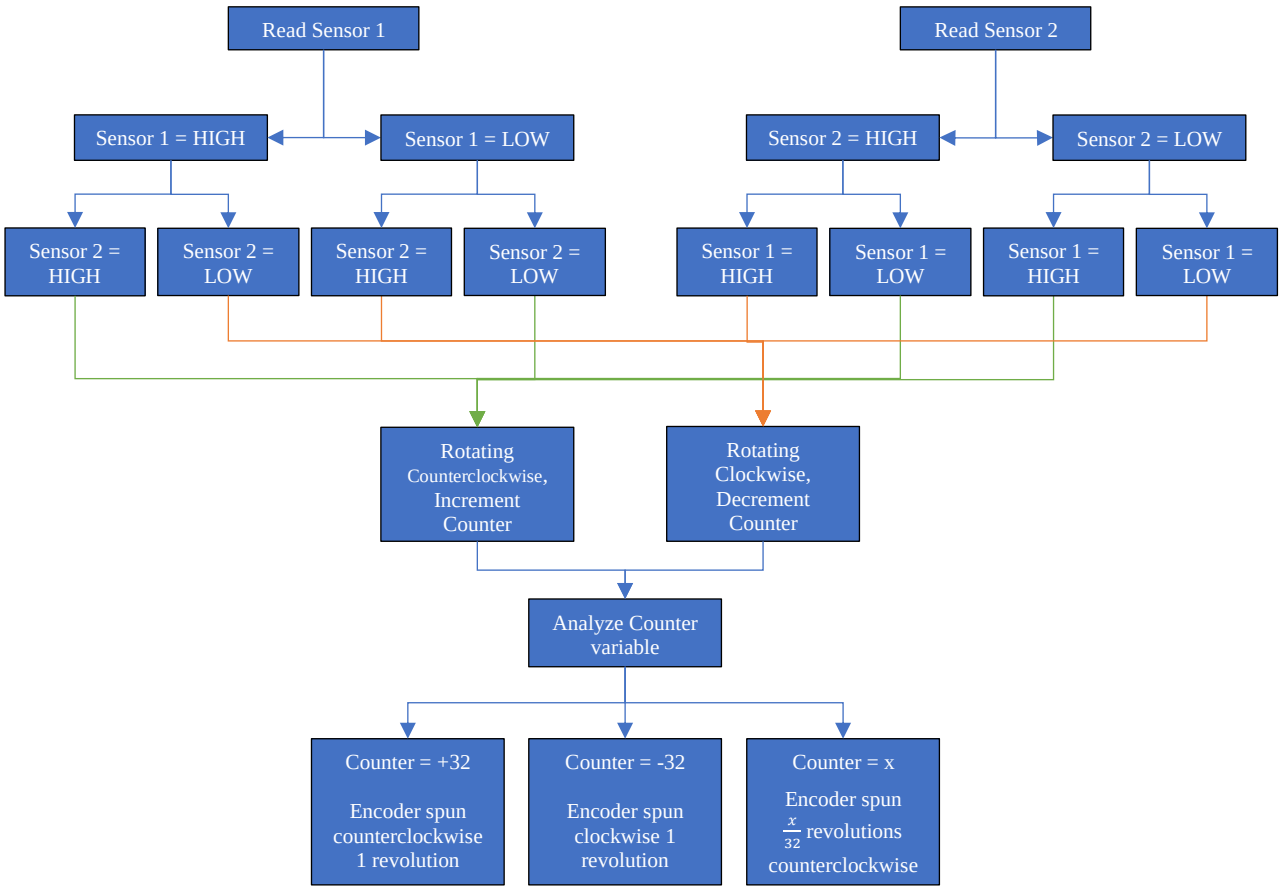
      ii. If Sensor 1 != Sensor 2, the wheel is rotating clockwise. In this case, decrement the counter

   b. Sensor 2 ISR:

      i. Sensor 2 = Sensor 1 if the wheel is rotating clockwise. In this case, decrement the counter

      ii. If Sensor 2 != Sensor 1, the wheel is rotating counterclockwise. And finally, for this case increment the counter

   c. To know how many revolutions the wheel is turning, we can look at the counter. A full revolution requires $\frac{360deg}{revolution} * \frac{1count}{11.25deg} = 32 \frac{counts}{revolution}$



9. The performance of a car's shock absorber needs to be monitored to assist with the design and testing of the shock absorber. The fastest rate at which the shock absorber might effectively move to absorb the energy of bumps is 100 Hz. (100 bumps per second for example.)

   The transducer used is a linear analog position sensor. It produces a voltage proportional to the length between the two anchor points of the shock absorber. The maximum voltage is +0.100 V at full extension of the shock absorber and the minimum is –0.100 V at full compression. The transducer has a frequency response up to 10000 Hz. Thus, it is not only is sensitive to the motion of the shock absorber, it is also sensitive to some of the sound of the wheel rolling over the road and conducted into the shock-absorber.

   A commercially available analog-to-digital converter is in use. The sample rate of the converter is 1000 Hz. (One sample is taken every millisecond). This analog-to-digital converter has an input range of +10 V to –10 V and provides a 10-bit output in a two's complement format such that +10 V produces the code $01\ 1111\ 1111_{TC}$ which corresponds to $+511_{10}$ and –10 V produces the code $10\ 0000\ 0001_{TC}$ which corresponds to -511_{TC}. This converter has no anti-aliasing filter. The user is expected to add their own anti-aliasing filter.

   The user figured that the 10000 Hz frequency response of the transducer and the 1000 Hz sample rate are 100 and 10 times (respectively) more than needed since the user is only interested in motions at 100 Hz or less. Thus, these were perceived as very good specifications. (The user may be naive in this determination.) The $\pm$10 V input range is also perceived as 100 times greater than needed so that should also be very good. (The user may be naive in this determination.) The user is getting poor results from this system.

   For an anti-aliasing filter the user has resorted to averaging 8 sequential samples (using digital addition of the sampled data, then dividing by 8) in order to get better results. This has reduced their effective sample rate to 1000/8 = 125 Hz. (Every eight digital samples get averaged to produce just one average sample for the interval.) However, the samples still are not accurate.

   Assume that averaging digital samples must be abandoned as it is an incorrect method of addressing the problems here. Assume analog filters and amplifiers are available to any specifications you might like. For a filter, specify if it should pass low or high frequencies and what the cutoff frequency should be. For an amplifier, specify the gain. How can you use filters and amplifiers to improve the results? Explain your choices. (10 points)

Awkward word

First, the transducer is only outputting $\pm 0.100\ V$ while the analog-to-digital converter has an input of $\pm 10\ V$. This means that they are only using a small portion of the converter's capacity. Therefore, an amplifier should be used between the transducer and the converter with a gain of 100. This will boost the transducer signal from $\pm 0.100\ V$ to $\pm 10V$, thus using the full input range and increasing resolution.

The second problem relates to the frequency inputs. The transducer outputs at 10000Hz, the converter outputs at 1000Hz, and the desired rate is 100Hz. The mismatch here results in excess noise delivering bad results. Since the filters available are only for analog signals, it must be placed between the transducer and the converter, and not between the converter and microcontroller (since the converter brings it to a digital signal). By utilizing band-pass frequency filters, the noise can be reduced. The filter must have a low-pass frequency filter to block out frequencies higher than 100Hz. This will bring out the highest resolution to the desired microcontroller.

10

Name: Charley

Dordt University Engineering Department

EGR 304, Microprocessor Interfacing — Final Exam, May 4, 2017. 77/100 = A-
Open book, open notes, take-home.
An Internet-connected computer is required. A calculator is also required.

1.) A small local-area computer network (e.g. a network of two or three computers sharing files, printers, etc.—no Internet connection) could be made using USB connections or using Ethernet connections between the computers. Describe some of the challenges and advantages that would be encountered when using a USB connection for networking as opposed to using an Ethernet connection. How are these challenges rooted in the technological differences of the two connection types? (10 points)

9

USB: universal serial bus
Data rate; Power is higher
Full duplex

Pros:
intended Good power + information eg. to connect devices to one CPU printers

length limited to 5 meters

Cons:
slower up to 480 Mbps
this will greatly inhibit a network that covers only one floor of a building

Ethernet
length longer, full duplex
Intended for networking protocol
power is lower

better for multiple CPU's up to 4 Gbps
more information inclined
still has low power

You should get specific here: peer-to-peer vs. host-client

2.) A thirty-foot long asynchronous serial computer interface uses RS-232-style signaling but at "TTL" logic levels meaning 0 volts for logic-1 and idle and mark or 5 volts for logic-0 or space, nominal levels. The interface is in practically continuous use. (There are no significant periods of idle time.) About half of the time the transmission line is sending logic-1 and the other half of the time it is sending logic-0. A low-impedance driver is connected to a 50 Ω transmission line and a 50 Ω matched load is used to prevent reflections. It works fine but battery life is poor due to all the current flowing in the load. Suggest a change or changes to the source and/or load impedances that will improve battery life. (10 points)

8

Change the source impedence to match the load impedence. In other words, add a 50Ω resistor in serie immediately after the low impedence driver. Because the impedence of the circuit will be doubled the current drain will be cut in half. Since current is related to power this will greatly improve battery life.

- Perhaps add a capacitor outside of the source to eliminate DC which will drain battery life

Capacitor in series between source and transmission line

An important further improvement is possible. Raise R_L as much as possible toward an open circuit. Reflections will now be absorbed at the source.

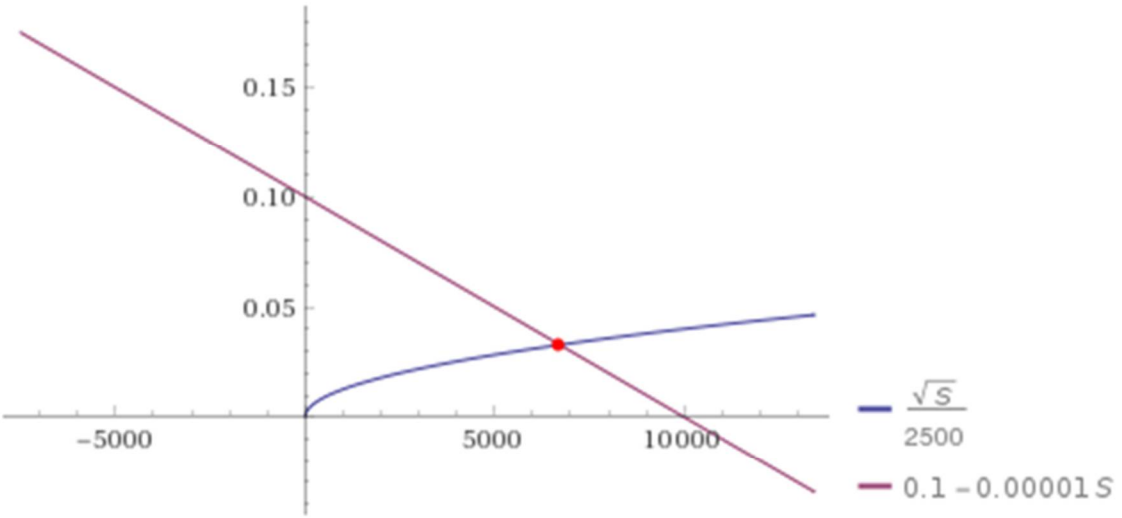
Input:

$$\sqrt{S} \times \frac{1}{2500} = 0.1 - 0.00001\,S$$

Result:

$$\frac{\sqrt{S}}{2500} = 0.1 - 0.00001\,S$$

Plot:



Alternate form assuming S is real:

$$S + 40.\sqrt{S} = 10\,000.$$

Alternate form:

$$\frac{\sqrt{S}}{2500} = -0.00001\,(S - 10\,000)$$

Alternate form assuming S is positive:

$$\sqrt{S} = 81.9804$$

For problem 3

3.) A DC motor with a permanent-magnet field, when connected to a 12 V DC source has a no-load speed of 10 000 RPM and a stall (locked rotor) torque (a.k.a. locked rotor torque) of 0.1 Nm. Assume the motor has an ideal linear torque vs. speed characteristic. This motor is connected to a low-impedance 12 V source on the electrical side and to a mechanical load that requires a driving torque that is related to the speed such that

$$T = \sqrt{S}\left(\frac{1}{2500}\ \text{Nm/RPM}^{0.5}\right)$$

Where T = required driving torque (in Nm)
S = load speed (in RPM)

10

Find the speed at which the motor will operate. (10 points)

graph this

1) graph ideal torque-speed curve it is linear and is represented by

see where it intersects with ideal T vs Speed (will be a line)

$T = 0.1 - 0.00001S$

2) Graph given equation $T = \sqrt{S}\left(\frac{1}{2500}\right)$

3) See where they intersect and that is where operation will occur

$\sqrt{S}\left(\frac{1}{2500}\right) = 0.1 - 0.00001S$

$\left(\frac{1}{2500}\right)\sqrt{S} + (0.00001)S = 0.1$

Plugged into wolfram Alpha

$\sqrt{S} = 81.98$

$S = 6720.79$ RPM

4.) Describe in words or pseudo-code how to use an interrupt driven tic-clock system with a microcontroller to generate a pulse of exactly 10 ms each time a certain subroutine is called. Assume the tic-clock interrupt has a frequency of 10 kHz. (10 points)

on the arduino

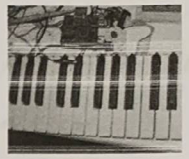
I could decide which pin I want the 10ms pulse on then set it to logic Low or zero, unless the certain sub routine is called the start a block of code that drives the pin HIGH. Then idles for 10 ms, then drives the pin Low again X

1

You must use a task scheduler to take advantage of a tic clock.

5.) Consider creating a keyboard-operated music synthesizer by connecting a keyboard to an Arduino microcontroller. Let the synthesizer have 24 keys (two octaves). Each key will have just one SPST switch. The keyboard has a 48-conductor ribbon cable connected to it, two conductors dedicated to each switch. Could this keyboard be connected directly to an Arduino microcontroller board using no additional parts such as diodes, gates, resistors, or transistors? The Arduino board has 13 digital I/O pins available. If so, describe how and describe any limitations of the connection. If not, give a rationale for your conclusion. (10 points)

A hint of a multiplexing strategy is here, but it is disorganized, and limitations are not addressed.



Multiplexing

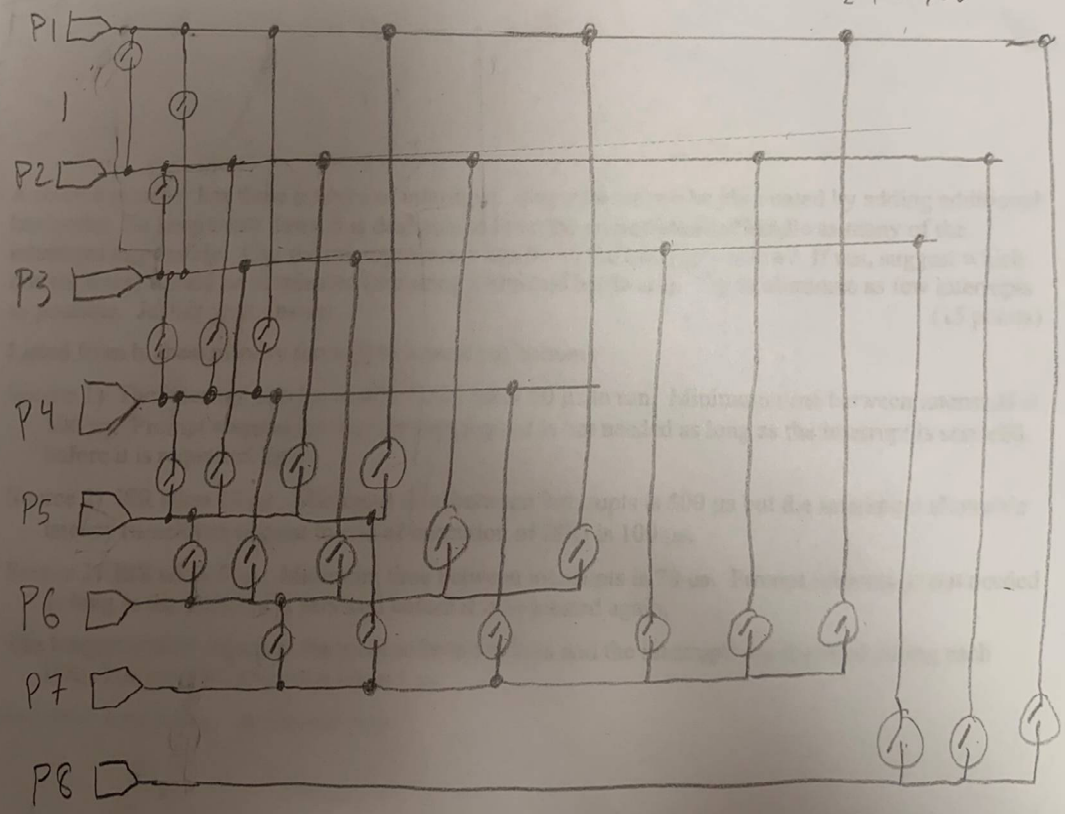
Yes via Charlieplexing

Limitations

would be better if we could include capacitors for the de bouncing of each key

I can do 21 keys with 7 pins therefore with 8 pins I can connect to all 24 keys.

5



6.) What is the difference between vectored interrupts and polled interrupts. (10 points)

Polled interrupt systems have a bit longer of a process as they have to check every potential interrupt source. X

7

A vectored interrup is a processing technique in which an interrupting device directs the processor to the interrupt service routine. ✓

With polled interrupts a device may request service, but the CPU is not told which device requested service. The CPU only polls in response to a request. It does not poll to detect a request, and ISR handles that.

7.) A microcontroller has three sources of interrupts. Any of these can be eliminated by adding additional hardware. To keep costs down it is desirable to have the microcontroller handle as many of the interrupts as possible. Can the microprocessor handle all the interrupts below? If not, suggest which one (or ones) should be eliminated (requiring additional hardware). Try to eliminate as few interrupts as possible. Justify your answer. (15 points)

Listed from highest priority (on top) to lowest (on bottom)

Source 1) The interrupt service routine (ISR) takes 50 μs to run. Minimum time between interrupts is

(This page is provided for the answer to the problem stated on the previous page.)

Summary of the question: Will interrupts work? Eliminate as few as possible if any need to be eliminated. Justify your answer.

Source 1) ISR takes 50 µs to run, 100 µs between interrupts.
Source 2) ISR takes 10 µs to run, 500 µs between interrupts, latency < 100 µs req'd.
Source 3) ISR takes 5 µs to run, 70 µs between interrupts
Critical region is 8 µs, interrupts disabled during ISRs, longest instruction is 1 µs.

12

$\frac{50}{100} + \frac{10}{500} + \frac{5}{70} = \frac{207}{350} < 1$ ✓

$T_1 = 50$ $T_{P1} = 100$
$T_2 = 10$ $T_{P2} = 500$ X
$T_3 = 5$ $T_{P3} = 70$

$T_{1+} = 10$
$T_{2+} = 5$ X
$T_{3+} = 1$ X

$N(2,1) = \left\lceil \frac{T_{P2} - T_2}{T_{P1}} \right\rceil$

$= \frac{500-10}{100} = \frac{490}{100}$

$= 4.9 = 5$ X

for #1 $T_{1+} + N(1,1)\,T_1 < T_{P1} \Rightarrow 10 + 50 < 100$

$60 < 100$ ✓ ok

for #2 $T_{2+} + N(2,1)\,T_1 + N(2,2)\,T_2 < T_{P2}$

$5 + 45\,(50) + 1\,(10) < 500$

$265 < 500$ ✓ ok

This all contributes to latency which needs to be under 265 + 100 µs

$N(3,1) = \left\lceil \frac{T_{P3} - T_3}{T_{P1}} \right\rceil$

$= \left\lceil \frac{70-5}{100} \right\rceil = 1$

for #3 $T_{3+} + N(3,1)\,T_1 + N(3,2)\,T_2 + N(3,3)\,T_3 < T_{P3}$

$N(3,2) = \left\lceil \frac{T_{P3} - T_3}{T_{P2}} \right\rceil$

$1 + (1)\,50 + 1\,(10) + 1\,(5) < 70$

$= \left\lceil \frac{65}{500} \right\rceil = 1$

$66 < 70$ ✓ ok

I think this needs to be 8µs to account for critical regions

For reliable operation eliminate interrupt 2 and it will operate because all that will be left is

$1 + 50 + 5 < 70$

$55 < 70$ ✓ ok

8.) Design a rotational position sensor for sensing ***absolute*** position with a resolution of 11.25 degrees (1/32 of a revolution). The rotational position sensor needs to only keep track of one revolution. If the shaft of the sensor should roll past the home position, the position code can roll over too. Each time the rotational position of a disk changes by 1/32 of a revolution, an interrupt should be produced. You may assume that the interrupt will be serviced before the disk moves enough to generate another interrupt request. You may decide for yourself how much logic to do in hardware and how much to do in the ISR. Your design should be relatively easy to manufacture. (15 points)

15

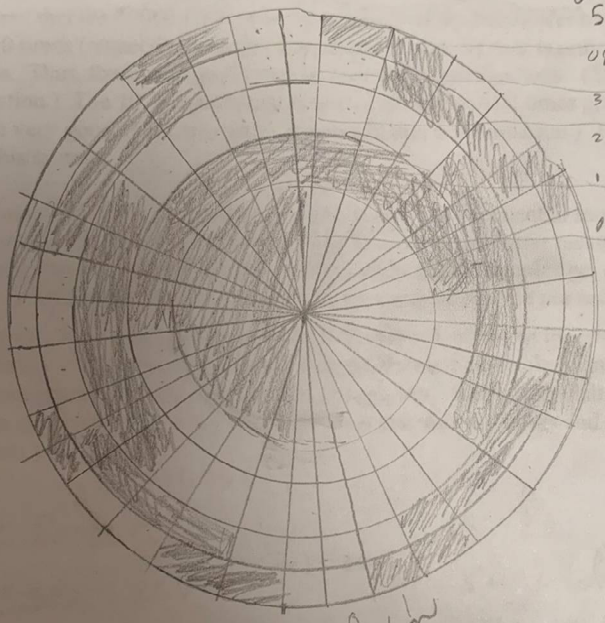
a.) Show the design of the disk and number and location of the sensor(s).

b.) Show a flow chart (or equivalent pseudo-code) to describe the processing that needs to be done in the ISR to keep track of ***absolute*** position.

a)

32 unique positions
5 "tracks"
5 bits allows for position numbered 0-31

GRAY CODE



least significant bit

on other side 5 lightsensors

4 5 photo transistors
3 pick send pulse
2 whenever
1 LED shines
0 through slots of each "track"

most sig bit

Rotary Encoder

on one side

5 LEDs one for each "track of slots"

b)

Interrupt service routine needs to calculate signal from each photo transistor as a 1 for its bit

eg. when sensing this position photo transistor number 4 will sense light from its LED

The ISR will assign a "1" to the least significant bit and know that that position: 00001 has been reached

9. The performance of a car's shock absorber needs to be monitored to assist with the design and testing of the shock absorber. The fastest rate at which the shock absorber might effectively move to absorb the energy of bumps is 100 Hz. (100 bumps per second for example.)

The transducer used is a linear analog position sensor. It produces a voltage proportional to the length between the two anchor points of the shock absorber. The maximum voltage is +0.100 V at full extension of the shock absorber and the minimum is –0.100 V at full compression. The transducer has a frequency response up to 10000 Hz. Thus, it is not only is sensitive to the motion of the shock absorber, it is also sensitive to some of the sound of the wheel rolling over the road and conducted into the shock-absorber.

A commercially available analog-to-digital converter is in use. The sample rate of the converter is 1000 Hz. (One sample is taken every millisecond). This analog-to-digital converter has an input range of +10 V to –10 V and provides a 10-bit output in a two's complement format such that +10 V produces the code 01 1111 1111_{TC} which corresponds to $+511_{10}$ and –10 V produces the code 10 0000 0001_{TC} which corresponds to -511_{TC}. This converter has no anti-aliasing filter. The user is expected to add their own anti-aliasing filter. 10

The user figured that the 10000 Hz frequency response of the transducer and the 1000 Hz sample rate ADC are 100 and 10 times (respectively) more than needed since the user is only interested in motions at 100 Hz or less. Thus, these were perceived as very good specifications. (The user may be naive in this determination.) The 10 V input range is also perceived as 100 times greater than needed so that should also be very good. (The user may be naive in this determination.) The user is getting poor results from this system.

Prob For an anti-aliasing filter the user has resorted to averaging 8 sequential samples (using digital addition of the sampled data, then dividing by 8) in order to get better results. This has reduced their effective sample rate to 1000/8 = 125 Hz. (Every eight digital samples get averaged to produce just one average sample for the interval.) However, the samples still are not accurate.

Assume that averaging digital samples must be abandoned as it is an incorrect method of addressing the problems here. Assume analog filters and amplifiers are available to any specifications you might like. For a filter, specify if it should pass low or high frequencies and what the cutoff frequency should be. For an amplifier, specify the gain. How can you use filters and amplifiers to improve the results? Explain your choices. (10 points)

10

0.1 – (–0.1) = 0.2

Transducer Range 0.2V

ADC Range 20V 10 – (–10) = 20

So an amplifier should be placed after the transducer with a gain of 100 to match voltages and utilizing the entire range of the ADC

A Low pass filter should be placed before the ADC at the Nyquist frequency which is ½ sample rate of the ADC which is $\frac{1000}{2} = 500$ Hz

Low Pass w/ cutoff 500 Hz

Name: Ryan Zevenbergen

Dordt University Engineering Department

EGR 304, Microprocessor Interfacing — Final Exam, May 4, 2017.
Open book, open notes, take-home.
An Internet-connected computer is required. A calculator is also required.

97/100 = A

1.) A small local-area computer network (e.g. a network of two or three computers sharing files, printers, etc.—no Internet connection) could be made using USB connections or using Ethernet connections between the computers. Describe some of the challenges and advantages that would be encountered when using a USB connection for networking as opposed to using an Ethernet connection. How are these challenges rooted in the technological differences of the two connection types? (10 points)

Where both of these connections are able to supply power, that feature will be disregarded in this comparison due to the likelihood of it being insufficient for the devices listed on the network.

Beginning with Ethernet: Ethernet provides many advantages for network connection. It uses the client/server architecture and can traverse greater distances that USB can. Currently, Ethernet can also achieve greater speeds than USB, but with USB 4.0 coming, the gap is closing. Because of the range and speed advantages, Ethernet is capable of setting up broader networks. In addition, its architecture allows every device to be simultaneously a client and a server, adding the advantage of various pathways and potentially faster transmission. This comes with the danger of introducing loops, however.

Next to USB: USB uses the host/hub architecture and can only traverse distances no greater than 15 ft. by the standard. One significant advantage, however, is the automatic loop detection. USB is also a more common connection type with devices that come with specific drivers, so it can support almost any kind of device. The shortcomings lie in its range and topography. Having only a 15 ft. range, a longer distance network would require one or multiple signal amplifiers. This increases the overall cost. In addition, the host/hub topography may lead to issues. Every device will have a host to reply to, so if multiple devices are hooked up to one host, the USB port could become overloaded. I was actually in this situation. At my internship, I had a laptop connected to a dock via USB. Into this dock, there were two monitors, a speaker/microphone headset, a high speed mouse, a second keyboard, and the phone line (so it could connect to the headset). This was too much data for a single USB port to handle, and I had to find other means to connect all of the devices.

10

The differences in these two comes in their original purposes. Ethernet was specifically designed for the physical and datalink layers of TCP/IP and thus, lends itself well to network connections specifically. USB, on the other hand, was designed for local device connection, such as mice, keyboards, etc. Thus, it was not necessarily originally intended for network usage.

2.) A thirty-foot long asynchronous serial computer interface uses RS-232-style signaling but at "TTL" logic levels meaning 0 volts for logic-1 and idle and mark or 5 volts for logic-0 or space, nominal levels. The interface is in practically continuous use. (There are no significant periods of idle time.) About half of the time the transmission line is sending logic-1 and the other half of the time it is sending logic-0. A low-impedance driver is connected to a 50 Ω transmission line and a 50 Ω matched load is used to prevent reflections. It works fine but battery life is poor due to all the current flowing in the load. Suggest a change or changes to the source and/or load impedances that will improve battery life. (10 points)

$$\Gamma_L = \frac{R_L - Z_c}{R_L + Z_c} = \frac{50 - 50}{50 + 50} = \frac{0}{100} = 0 \rightarrow No\ reflection$$

8

$$\Gamma_S = \frac{R_S - Z_c}{R_S + Z_c} = \frac{R_S - (50 + 50)}{R_S + (50 + 50)} = \frac{R_S - 100}{R_S + 100} \rightarrow R_S = 100\ \Omega\ for\ no\ reflections$$

By increasing the source resistance, the current of the system is decreased via Ohm's law. This will decrease the power necessary to drive the system. In addition, by choosing the Thevenin equivalent of the transmission and load impedances, no reflections will arise. Thus, the source impedance should be increased to 100 ohms.

Your ideas are sort of correct. The conclusion of a 100 ohm source is wrong.

3.)A DC motor with a permanent-magnet field, when connected to a 12 V DC source has a no-load speed of 10 000 RPM and a stall (locked rotor) torque (a.k.a. locked rotor torque) of 0.1 Nm. Assume the motor has an ideal linear torque vs. speed characteristic. This motor is connected to a low-impedance 12 V source on the electrical side and to a mechanical load that requires a driving torque that is related to the speed such that

$$T = \sqrt{S}\left(\frac{1}{2500}\,\text{Nm}/\text{RPM}^{0.5}\right)$$

Where T = required driving torque (in Nm)
S = load speed (in RPM)

Find the speed at which the motor will operate. (10 points)

$$T = 0.1\ Nm - \left[\frac{0.1\ Nm}{10000\ RPM}\right]S$$

8

$$\sqrt{S}\left(\frac{1\ Nm}{2500\ RPM^{0.5}}\right) = 0.1\ Nm - \left[\frac{0.1\ Nm}{10000\ RPM}\right]S$$

$$\sqrt{S}\left(\frac{1}{2500}\right) + \left(\frac{0.1}{10000}\right)S = 0.1$$

x

Missed the middle term.
(a + b)^2 is not equal to a^2 +b ^2

$$\left(\frac{0.1}{10000}\right)^2 S^2 + \left(\frac{1}{2500}\right)^2 S - 0.1^2 = 0$$

$$S = 9231.95\ RPM\ or - 10831.95\ RPM$$

Since a negative speed does not make sense in this case:

$S = 9231.95\ RPM$ = 6721 RPM

x

4.) Describe in words or pseudo-code how to use an interrupt driven tic-clock system with a microcontroller to generate a pulse of exactly 10 ms each time a certain subroutine is called. Assume the tic-clock interrupt has a frequency of 10 kHz. (10 points)

To obtain a pulse of exactly 10 ms generated, the clock would need to cycle 100 times, as the clock takes 100 µs to complete one cycle. The interrupt must be set up on pin 1 or 2 if using an Arduino Uno. Three variables are needed: one for the output pin, one for the interrupt, and a volatile variable for the change. A counter should be set up in the ISR to trigger on the rising edge of the tic-clock. Initially, both the volatile and output pins should be low. Once the counter reaches 100, the volatile value should change polarity, and the counter should be reset. Then the output should be set to whatever the volatile variable changed to, be that low or high. In this way, for every 100 pulses of the clock, the output will change, resulting in a pulse of exactly 10 ms.

9

This would generate a 50 Hz square wave (100 ms high, 100 ms low.) It would not generate a 100 ms long pulse at the command of a subroutine call. A task scheduler is the best way to do what was requested.

5.) Consider creating a keyboard-operated music synthesizer by connecting a keyboard to an Arduino microcontroller. Let the synthesizer have 24 keys (two octaves). Each key will have just one SPST switch. The keyboard has a 48-conductor ribbon cable connected to it, two conductors dedicated to each switch. Could this keyboard be connected directly to an Arduino microcontroller board using no additional parts such as diodes, gates, resistors, or transistors? The Arduino board has 13 digital I/O pins available. If so, describe how and describe any limitations of the connection. If not, give a rationale for your conclusion. (10 points)

This would not be feasible in the case of a music synthesizer. The only way to connect a 24-key device to a 13-port system would be to multiplex it. But in the case of a music synthesizer, often more than one key is pressed at a time to form a chord. This would cause a short on the multiplexing scheme and cause damage to the system (and its devices). If prevention methods were in place to make sure the user never pressed more than one key down at a time, then it could function, but this would be a severe limitation in its performance.

7

With matrix scanning and open-collector drivers (Arduino has these) for either the rows or the columns of the matrix any two keys simultaneously pressed can be easily and properly scanned. Sophisticated software can extend this performance a bit. Say a chord has 4 notes. Most likely there will be 1 ms or so between key presses, even if the person playing tries to press all four simultaneously. With rapid and continuous scanning the sequence of key presses can be estimated from the on-going scanning of the matrix. Some combinations are still ambiguous. They can be handled by ignoring the new key-press (no sound if necessary) or by choosing the most harmonious possible option. It might make a mistake, but it might also sound OK. This can be suitable for a toy instrument.

6.) What is the difference between vectored interrupts and polled interrupts. (10 points)

In a vectored interrupt, the device requests service from the CPU. The CPU then suspends its operations and find a location in memory (or vector) that tells it how to deal with the device's interrupt. This is then processed and performed, allowing the CPU to return to its previous state and complete its tasks.

Polled "interrupts" are actually generated by the CPU. Usually used in programs or architecture that do not have interrupts, the CPU "polls" each device to see if it needs service. If it does, the CPU accomplishes the task and goes back to either polling the rest of the devices or completing its original task. This one is much more computationally expensive, as the CPU needs to do both tasks of checking whether devices need service and completing the service necessary.

With polled interrupts a device may request service but the CPU is not told which device requested service. The CPU only polls in response to a request. It does not poll to detect a request, and ISR handles that.

7.) A microcontroller has three sources of interrupts. Any of these can be eliminated by adding additional hardware. To keep costs down it is desirable to have the microcontroller handle as many of the interrupts as possible. Can the microprocessor handle all the interrupts below? If not, suggest which one (or ones) should be eliminated (requiring additional hardware). Try to eliminate as few interrupts as possible. Justify your answer. (15 points)

Listed from highest priority (on top) to lowest (on bottom)

Source 1) The interrupt service routine (ISR) takes 50 μs to run. Minimum time between interrupts is 100 μs. Prompt response to the interrupt request is not needed as long as the interrupt is serviced before it is requested again.

Source 2) ISR takes 10 μs. Minimum time between interrupts is 500 μs but the maximum allowable latency (time from request to end of execution of ISR) is 100 μs.

Source 3) ISR takes 5 μs. Minimum time between interrupts is 70 μs. Prompt response is not needed as long as the interrupt is serviced before it is requested again.

The longest critical region in the main software is 8 μs and the interrupts are disabled during each ISR. The longest instruction takes 1 μs.

(Please write your answer on the next page.)

(This page is provided for the answer to the problem stated on the previous page.)

Summary of the question: Will interrupts work? Eliminate as few as possible if any need to be eliminated. Justify your answer.

Source 1) ISR takes 50 µs to run, 100 µs between interrupts.
Source 2) ISR takes 10 µs to run, 500 µs between interrupts, latency < 100 µs req'd.
Source 3) ISR takes 5 µs to run, 70 µs between interrupts
Critical region is 8 µs, interrupts disabled during ISRs, longest instruction is 1 µs.

$$T_1 = 50\ \mu s \qquad T_2 = 10\ \mu s \qquad T_3 = 5\ \mu s$$
$$T_{P_1} = 100\ \mu s \qquad T_{P_2} = 500\ \mu s \qquad T_{P_3} = 70\ \mu s$$
$$T_{1+} = 10\ \mu s \qquad T_{2+} = 50\ \mu s \qquad T_{3+} = 50\ \mu s$$

$$\frac{T_1}{T_{P_1}} + \frac{T_2}{T_{P_2}} + \frac{T_3}{T_{P_3}} = \frac{50}{100} + \frac{10}{500} + \frac{5}{70} = 0.591 < 1 \rightarrow OK$$

$$N(1,1) = 1 \qquad N(2,1) = 5 \qquad N(3,1) = 1$$
$$N(2,2) = 1 \qquad N(3,2) = 1$$
$$N(3,3) = 1$$

15

$$T_{1+} + (1)T_1 < T_{P1} \quad 10 + 50 = 60\ \mu s < 100\ \mu s$$
$$T_{2+} + (5)T_1 + (1)T_2 < T_{P2} \quad 50 + (5)50 + 10 = 310\ \mu s < 500\ \mu s$$
$$T_{3+} + (1)T_1 + (1)T_2 + (1)T_3 < T_{P3} \quad 50 + 50 + 10 + 5 = 115\ \mu s > 70\ \mu s \rightarrow BAD$$

Thus, Source 3 should be put into hardware. This will have added benefits because it means that Source 2 will never go more than 100 µs before being serviced, as even if it has to wait for Source 1 and the critical region, this will only take a total of 68 µs.

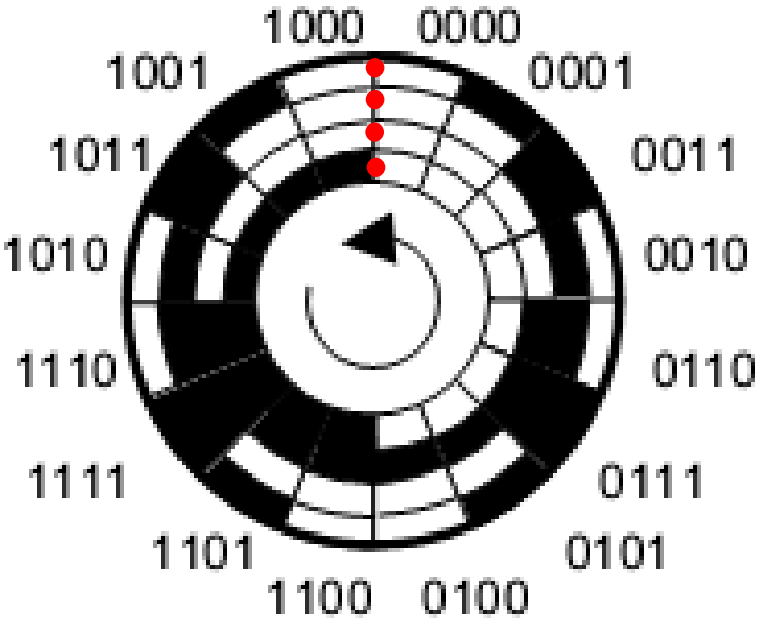
A better solution:
Observe that ISR 1 has margin, thus does not need priority. Demote it to lowest priority and everything will work.

Overall technique is correct.

8.) Design a rotational position sensor for sensing ***absolute*** position with a resolution of 11.25 degrees (1/32 of a revolution). The rotational position sensor needs to only keep track of one revolution. If the shaft of the sensor should roll past the home position, the position code can roll over too. Each time the rotational position of a disk changes by 1/32 of a revolution, an interrupt should be produced. You may assume that the interrupt will be serviced before the disk moves enough to generate another interrupt request. You may decide for yourself how much logic to do in hardware and how much to do in the ISR. Your design should be relatively easy to manufacture. (15 points)

a.) Show the design of the disk and number and location of the sensor(s).

b.) Show a flow chart (or equivalent pseudo-code) to describe the processing that needs to be done in the ISR to keep track of ***absolute*** position.



To achieve 11.25 degrees of resolution, 5 bits are needed. ***Note: the image to the left only has 4 bits. I could not find a suitable 5-bit counterpart, so just recognize that there is supposed to be one more layer and one more sensor.*** I would use a rotational sensor with gray code. Each of the five bits would be associated with a sensor, and each sensor would read a section of the rotator shown to the left. White indicates logic-0 and black indicates logic-1. Most of the processing would be done in hardware, as the sensors would send their corresponding logic value and the microcontroller would only need to interpret the values to find the position via a lookup table. The dots represent where the sensors will be located.

Image found at: https://scienceprog.com/using-gray-code-for-rotary-encoders/

15

9. The performance of a car's shock absorber needs to be monitored to assist with the design and testing of the shock absorber. The fastest rate at which the shock absorber might effectively move to absorb the energy of bumps is 100 Hz. (100 bumps per second for example.)

   The transducer used is a linear analog position sensor. It produces a voltage proportional to the length between the two anchor points of the shock absorber. The maximum voltage is +0.100 V at full extension of the shock absorber and the minimum is –0.100 V at full compression. The transducer has a frequency response up to 10000 Hz. Thus, it is not only is sensitive to the motion of the shock absorber, it is also sensitive to some of the sound of the wheel rolling over the road and conducted into the shock-absorber.

   A commercially available analog-to-digital converter is in use. The sample rate of the converter is 1000 Hz. (One sample is taken every millisecond). This analog-to-digital converter has an input range of +10 V to –10 V and provides a 10-bit output in a two's complement format such that +10 V produces the code $01\ 1111\ 1111_{TC}$ which corresponds to $+511_{10}$ and –10 V produces the code $10\ 0000\ 0001_{TC}$ which corresponds to -511_{TC}. This converter has no anti-aliasing filter. The user is expected to add their own anti-aliasing filter.

   The user figured that the 10000 Hz frequency response of the transducer and the 1000 Hz sample rate are 100 and 10 times (respectively) more than needed since the user is only interested in motions at 100 Hz or less. Thus, these were perceived as very good specifications. (The user may be naive in this determination.) The $\pm$10 V input range is also perceived as 100 times greater than needed so that should also be very good. (The user may be naive in this determination.) The user is getting poor results from this system.

   For an anti-aliasing filter the user has resorted to averaging 8 sequential samples (using digital addition of the sampled data, then dividing by 8) in order to get better results. This has reduced their effective sample rate to 1000/8 = 125 Hz. (Every eight digital samples get averaged to produce just one average sample for the interval.) However, the samples still are not accurate.

   Assume that averaging digital samples must be abandoned as it is an incorrect method of addressing the problems here. Assume analog filters and amplifiers are available to any specifications you might like. For a filter, specify if it should pass low or high frequencies and what the cutoff frequency should be. For an amplifier, specify the gain. How can you use filters and amplifiers to improve the results? Explain your choices. (10 points)

Obviously the noise from the road is effecting measurements. Amplifying the noise in addition to the signal and averaging the result is causing poor performance. The user must find a low pass filter, since the only thing they are interested in is the shock absorber. The low pass filter should allow 100 Hz to pass through, or perhaps 110 Hz depending on how steep the cutoff is. This will allow the 100 Hz signals to pass through to the sensor, while blocking any noise of higher frequency from the road. Afterwards, an amplifier with a gain of 7-10 can be applied to the signal. This will place it in a significant range for the sensor, allowing for accurate readings. Additionally, a Schmidt trigger could be used for additional accuracy, preventing false changes in the signal.

8

Too low. A gain of 100 is needed.

The proposed changes would help, but these are not based on sound theory and so optimal results have still not been found.