Hi this is the schedule I have planned for us to finish

I need to type out and work on your code urgently reason being the deadline is coming up and I need to test your code to see if itworks with the rest of the project

I will be in the lab from 8 am - nz time to about 10 pm everyday

Update me on the code as you go along , I will read through your code and ask to implement certain funtionalities to the code

step by step what needs to be done

1)Today help me write finish the UART receiver side code so that I can transmit code from the pc putty terminal to the linear microcontroller. What I want you to type into putty is in the client document

In this format “ “

2) from this I want you to extract the mfc value above the mfc value is 123 this is the max flow rate im not sure whether we need to convert the 123 value back to binary value and I want you to take 123/255(max flow value) and change the pwm signal period on and off look at PWM code provided so that the effect of the pwm on the motor will cause the motor to slow down.

3) I want this to trigger a message transmitted from the LCC to the UART stating the voltage value, current value, back emf value, resonant frequency I want the format to be transmitted in the form of stated in the client document” “ where we will store the values into the specific spaces provided

4) I might need help checking whether the values measured are correct voltage sensing, current sensing

5) I also need help setting up the short circuit protection measuring for large current through the coil trigger the turning off of our PWM by closing our mosfets by going active high

Preferably able to do 2-3 items a day don't worry I will stay up with you all day trying to understand the code

If you can help me finish everything by Friday if possible

Hi how is the UART receiver coming along could you help me implement this feature once you are done with the UART receiver I want to be able to transmit the following message below to the putty terminal the putty terminal only reads ASCII VALUE so you will have to store the below function in a string first after which you have to convert each character I at a time toASCII to be transmitted across to be displayed on putty terminal the putty terminal is the pc hyperterminal

An example of a fully populated ASCII JSON structure from the LCC:

```
{
   "3":
   {
         "mfc":
         {
               "req": "120",
               "cur": "123"
         },
         "ver": "1.2.3",
         "param":
         {
               "pwr":  "1.2W",
               "freq": "100Hz",
               "curr": "97mA",
               "volt": "12.32V"
         },
         "clr": "ew",
         "ew": ["cmprStalled", "pistonCollision"],
         "user":
         {
               "coolFeature": "on",
         }
   }
}
```

I want to be able to transmit this today to the putty terminal

Could help me store the value into a string I want to then convert each of this characters to ASCII value using the ASCII convert function in the UART.C file

What I mean is

1) Store the string above and convert it into and array or pointer

2) Send each character in the string to the ASCII convert function in the UART.c

3) Send this ASCII converted string to be stored in the UDR , the UDR will immediately help transmit the ASCII values to the putty terminal

4) I also need to be able to acess and store power(pwr) , frequency(fre),current value(curr),voltage value into the string to be transmitted could you help me send the voltage value in the ADC.C file and store it in the "12.32v" position to be sent to the PC (putty terminal) I want to check whether it can be read

5) Could you help me store all the values for the different variables inside as requested

6) Note could you write a function such that this message above is transmitted only if the following message is sent to the microcontroller

Example request of mass flow control value 130:

```
{"3": {"mfc": {"req": "130"}}}
```

Trigger the above function by sending and identifying the max flow rate

Hi how is the string to ASCII conversion coming along could you send me what you have done so far , also I need to implement this feature next so I need to send from the PC which is the receiver the following instructions

Example request of mass flow control value 130:

```
{"3": {"mfc": {"req": "130"}}}
```

This is one of the main steps I need to get done

I need to type this into the putty (PC terminal) and then I will need to transmit the value

130 from from the PC and store it in the UDR to be transmitted back to the microcontroller this value is then converted from string take this value 123 and divide it by 255 and multiply by PWM the pulse width of the PWM the PWM will affect the flow rate of the linear motor

(Please read the PWM code to understand it)

If you are able to do this then I should be able to test it in the lab to reduce the speed of the linear motor

Next step Could you help me write a short circuit protection code

Meaning we need to measure current sensing whether the value is greater then 300A short circuit value if that is the case then I want to stop and cut the PWM power completely

Use and interrupt to check if current is greater than a certain value

If it is greater then top the PWM output temporarily

Use a timer to monitor the current value every 1 microsecond , if current does reduce then enure that is the case 1 min later then restart the motor or the coil

Could you send error detection message from linear microcontroller back to the PC stating the short circuit in this format should be displayed store under ew: shortcircuit and voltage should read 0v

An example of a fully populated ASCII JSON structure from the LCC:

```
{
   "3":
   {
         "mfc":
         {
               "req": "120",
               "cur": "123"
         },
         "ver": "1.2.3",
         "param":
         {
               "pwr":  "1.2W",
               "freq": "100Hz",
               "curr": "97mA",
               "volt": "12.32V"
         },
         "clr": "ew",
         "ew": ["cmprStalled", "pistonCollision"],
         "user":
         {
               "coolFeature": "on",
         }
   }
}
```

Next step is to help me check whether back emf is working and to test for resonant frequency