ENGS101P: Team 4 Report

Stirring

Goal

The objective of the stirring subgroup was to produce a microprocessor controlled circuit that alters the frequency of a motor depending on the required rpm and the actual rpm of the motor.

Circuit Diagram

How it works.

The circuit consists of a transistor and a diode. The source of the transistor is connected to GND, the drain is connected to the supply and the gate is connected to the MSP430. The purpose of the transistor is to alter the voltage of the circuit as required to alter the frequency of the motor in the stirring equipment. As the voltage is increased, the frequency of the motor increases. The diode is to ensure that when the circuit is turned on, creating a power surge, the rest of the circuit is safe from over-heating. The phototransistor on the motor returns 1’s or a 0’s to the MSP430 which are used to calculate the Revolutions per Minute (RPM).

As for the code, to measure the RPM of the motor, the code creates a time variable which starts measuring time once the state of the phototransistor (1 or 0) changes and then uses a while loop which waits for the state to change again. After the second change in state, the code creates another time variable which measures the time again and then calculates the difference between the 2 time variables. From this, the Time period of the motor can be calculated and therefore the RPM can be calculated. Once the RPM is measured, the code compares this value to the value input into the User Interface. If the measured value is too low, the user interface sends information to the MSP430 which increases the voltage in order to speed up the motor. If the measured value is within a suitable range of the wanted value, the speed of the motor remains constant.

Ways to Improve

The system worked very well with the user interface and interacted correctly with user input. A way to improve the system would be to show the constant change in the rpm using a graph to monitor what the system is doing and to ensure that it is consistent with the input throughout the entire use. Also, the code could be improved by setting a limit to how high or low the RPM can be set to. This is because having no limits on input could be dangerous when implementing the system with a bioreactor.

Temperature

Goal

The objective of the temperature subgroup was to produce a system which monitors and controls the temperature of the environment to be able to provide suitable temperatures required to produce the vaccine in a bioreactor.

How it works

The system has 2 states, the Waiting State and the Operating State. By default, the system will be in the Waiting State, the serial monitor will keep printing “Waiting” to show the system is working and waiting to be initiated. Waiting State is also the only state that allows changing pre-set optimum temperature by user input.

Pressing the button 1.3 will initiate the system and it will go into Operating State after a delay of 1 second. It will then read the value of the pin connected to the thermistor and translate it into temperature (Kelvin or Celsius) using the Steinhart-Hart equation. The equation has several variables that could be modified to be compatible with different types of thermistors and can be calibrated by changing one value. The heating system could only be turned ON or OFF by sending a HIGH or LOW digitalWrite to the pin that connects to the heater.

If the temperature is much lower than pre-set temperature, the heating element is set to increase the temperature (ON for 4s and OFF for 1s during which it reads the new temperature). When the temperature reaches 2 degrees Celsius below the optimum temperature, the heater is powered less frequently (4s ON and 3s OFF).

Once it reaches 0.5 degrees Celsius, the heater remains OFF as it will still be hot and it takes time to cool down. Reading will be taken every 2 secs the heater turns ON again if the temperature falls 0.5 degrees below the optimum temperature.

For a small amount of water, it is likely that the temperature after the first break of heating will be 1-2 degrees higher than the optimum temperature if it is heated from a low temperature. If the pre-set optimum temperature needs to be changed, or, for some reason, the temperature seems to be too high, simply press and hold button 1.3 until next reading, the system will return to Waiting State and “Terminated” will be displayed on the monitor.

Ways to Improve

The code worked well with the UI and responded to user input correctly. The amount of water we added was too small, and after the heating was turned off, the temperature finally reached 34°C (close to max temperature 35°C) degrees and then eventually fell to optimum temperature. This could be improved by setting optimum temperature a little bit lower than 30 degrees during the first heating process and changed to normal afterwards.

pH – (STILL IN PROGRESS)

Goal

The objective of the pH subgroup was to provide a system that monitors and controls the pH of the environment. Some chemical reactions require certain pH levels to produce the best yield and so it is crucial that the pH of the system is kept constant.

How it works

Ways to Improve

User Interface – (STILL IN PROGRESS)

Goal

The objective of the user interface subgroup was to create a program that takes a required value for each subgroup using a user input, send data to the microcontrollers, and then receiving data which would then be displayed on the monitor.

How it works

The UI uses processing to produce a program that takes user input, communicates with the MSP430 and then

The user interface can only run one with one sub-system at a time so in order to display all three sub-systems, three monitors are required.

Ways to Improve

The user interface worked really well with all the sub-systems and produced correct outputs. One way to improve it would be to combine all the UI’s into one so that all the subgroups could be monitored on one screen. This would enhance the system as it would become easier to control and track each of the sub-systems.