Challenge 2: Global Health Care

EEE-CS Specialist Team Final Report

Authors: Miruna Cristiana Moraru, Tharmetharan Balendran (Liaison Engineer), An Vo, Hugues Codron, Zhang Weixi, Jiaho Xu, Longxi Yin, Minjae Kang, Roxana Tuglea, Zhou Andrea

1 Introduction

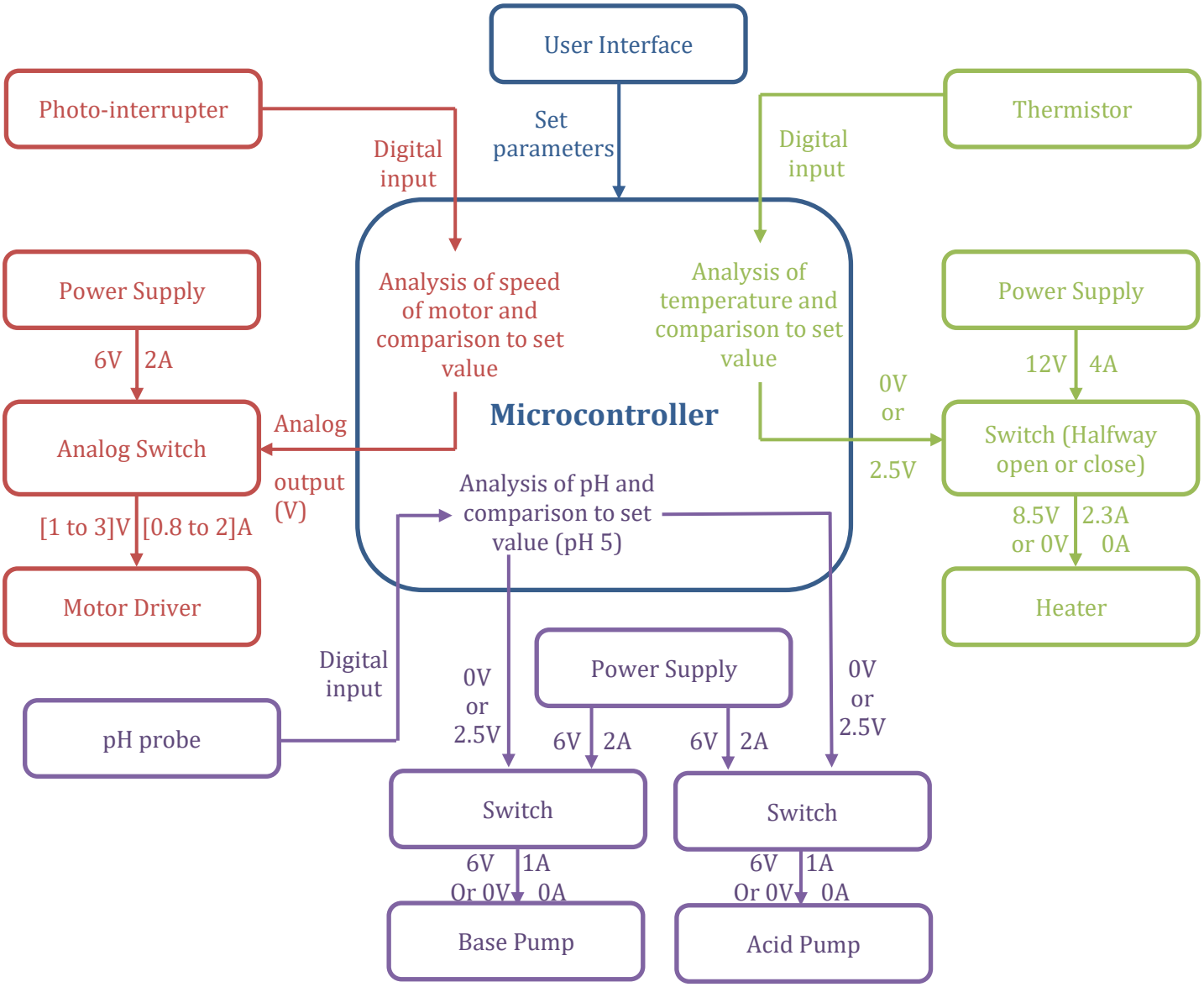
Tuberculosis has almost disappeared in developed regions, such as Northern American and European countries. However, it is still devastating in poorer regions in the world… One of the major goals, today, in terms of Healthcare, thus is to propose a solution to expand vaccines in these regions, where diseases are still one of the most common causes of death. This Challenge aimed at combining the work of different specialist engineers to provide a viable way to produce and expand a vaccine against tuberculosis in Uganda. The global project was split into 3 main aspects:

- The power provision for the plant
- The reactor and instrumentation design for the bioreactor
- The bioreactor control system

As Computer Scientists and Electrical and Electronic Engineers, we were in charge of the control system. Indeed, the potential user of the bioreactor needs to be able to control a couple of parameters concerning its content. Therefore, our team managed to enable the users to set the pH, the temperature and the stirring they wish in the bioreactor. The user enters the values in the interface, and the bioreactor maintains the content regarding the specifications.

After having put a liquid (within 10 and 20°C) in the bioreactor, our system is able to keep it at a set temperature between 25-35°C with a precision of ±0.5°C. The stirring speed can be set at any value between 500 and 1500 RPM, and the pH of the liquid can be maintained at 5.

To do so, our system follows this functional block diagram:



Our different probes enable the microcontroller to constantly collect data concerning the state of the content of the bioreactor, and to adapt the input to the other devices to fit the commands of the user. This design shall thus insure the set parameters inside the bioreactor. The control of these parameters will then enable the bioreactor to properly grow the organism that contains the antigen of tuberculosis. It is essential that the parameters are reliable, in order to make sure not to damage, in any way, the antigen.

However, this project doesn't only contain technical aspects. Indeed, the implantation of a vaccine plant cannot succeed if the population do not consent to it. A global promotion of the vaccine and the plant thus seems essential. Nevertheless, it may not be as straight forward as it could be expected. Indeed, the people of Uganda is probably not familiar with the idea of a vaccine, and may, in first place, be suspicious toward this new chemistry they don't understand.

To this extend, it would be interesting to highlight the fact that the plant would create a lot of jobs for the local population. They would of course be different from farming activities, but could still interest the population. They would be at first building jobs, for the construction of the plant, and then jobs concerning the production of the vaccine.

The power generated by the plant could also be an argument: the excess of electricity produced can be reversed to the population. Eventually, after a good promotion, some Ugandans may even try the vaccine, and realise it highly decreases the probability to be infected by Tuberculosis, what would encourage the rest of the population to take it.

All in all, the whole project is far from a simple scientific challenge, however, in this report, we will focus on the technical aspect of the project, concerning the bioreactor. We will thus present each subsystem (pH, heating, stirring) individually, before the overall integrated system.

2 Subsystem Descriptions

2.1 pH Probe Control Subsystem

The pH subsystem's purpose is to maintain the pH of the cultivating medium at an optimum level of pH 5 for maximum growth. This is achieved by continually measuring values of the current pH and injecting base and acid solution into the medium in order to keep the pH at the specified level. The brief requires the subsystem to be able to measure accurately the pH between 3 and 8; a requirement that has been sufficiently reached in our subsystem. Two main circuits make up the pH subsystem: one contains the pH probe (PH-100ATC, see Appendix 1.1) which inputs pH readings to the microcontroller, and the other contains two peristaltic pumps that will pump acid and base into the medium following the decisions of the microcontroller.

2.1.1 PH Probe

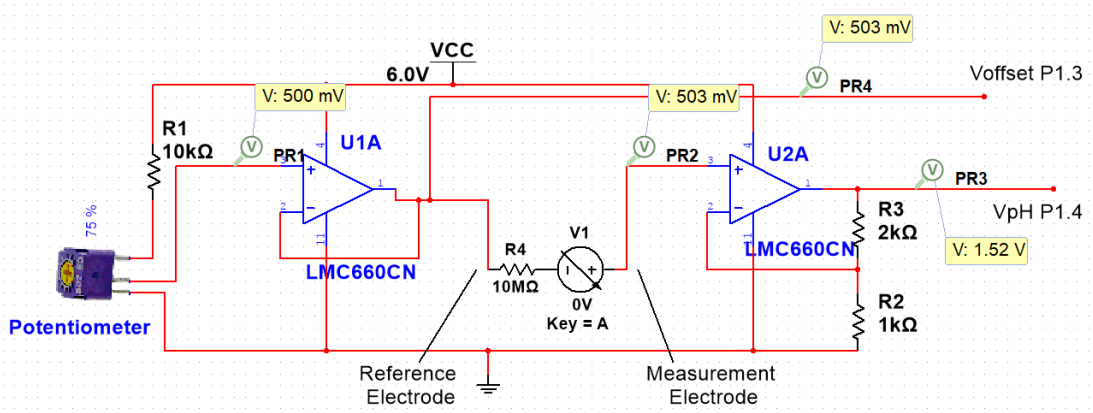


Figure 1: Circuit Schematic for pH probe circuit. See Appendix 1.2 for details

The pH probe measures the potential difference between the reference and measurement electrodes and the calculations carried out by the microcontroller determine the current pH of the solution. The pH reading is also dependent on temperature, so an extra input from the heating subsystem is required, see Appendix 1.3. However, the electrodes are able to produce bipolar potential differences, giving negative voltages when placed in a base solution. The range of these voltages needs to be shifted into a solely positive spectrum (unipolar) for the microcontroller to be able to read suitable values. This is achieved through the potential divider circuit on the left-hand side of the above schematic. The potential at the reference electrode is carefully adjusted to 500mV from a 6V supply, see Appendix 1.4, by setting the resistance of the potentiometer to a certain resistance, see Appendix 1.5. This offset voltage is inputted into the microcontroller, see Appendix 1.6. A buffer operational amplifier is also used between the potentiometer and the electrode, see Appendix 1.7.

The circuit connecting the measurement probe to the microcontroller contains an operational amplifier in a non-inverting amplifier configuration with a gain of three. When the peristaltic pumps and the stirring subsystem use the same 6V power supply to operate, the supply is unable to provide a constant 6V, causing the potential at the reference electrode to vary, and resulting in fluctuations, in the measurement potential.

See Appendix 1.8 for programming solution. The amplifier reduces these errors by using as much of the voltage reading range available from the microcontroller. See Appendix 1.9.

Testing the pH probe in a known value of pH 5.6, we found that the probe gave readings that fluctuated slightly above and below a pH 5.9, see Appendix 1.10 for results.

2.1.2 Peristaltic Pumps

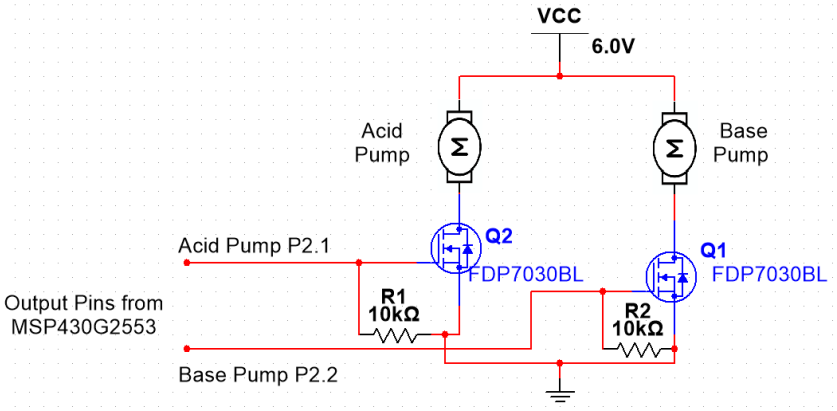


Figure 2: *Circuit Schematic for pH probe circuit*

The second part of the pH control subsystem consists of two 6V, 1A peristaltic pumps, connected to two MOSFET transistors that are used to switch the pumps on so that either alkaline or acid solution is injected to maintain medium at optimum pH. The gate of each transistor is connected to the appropriate output pin of the microcontroller. When one of the pumps needs to be operated, the microcontroller sends a voltage to the gate of the corresponding MOSFET, allowing current to flow through the drain and source, resulting in the pump turning on and pumping the desired solution. See Appendix 1.11 for pump calibration.

The two resistors connecting the gate with the drain are used to prevent the pumps running for longer than intended. When a voltage is applied to the gate, there is a build of charge between the gate and drain causing a capacitance between the terminals. When removing the voltage, there is still a build of charge that still allows current to flow through to the pump. The solution is the resistor, which quickly discharges this build up, turning off the electronic switch.

The pumps were tested by putting the probe in pH 5.9 and pH 4.7 solutions and our system was able to pump the correct solution after the pH value was measured, see Appendix 1.12 for results.

2.1.3 pH Control Subsystem Results/Validation

The following tests of the pH probe subsystem and the pump subsystem, in Appendix 1.8 and Appendix 1.10, shows that they can both work as separate units. When combined together, the operation of the entire pH control system was successful and met the specifications set by the brief. The control system continuously measures the pH to a suitable degree of accuracy and will inject the corresponding acid or base solution to main pH at a value of 5. Full test of integrated subsystem can be found in Appendix 1.13.

The pH control system can be improved by utilising the potentiometer built into the potential divider circuit. With our current system, we are able to measure across pH 0 to pH 14 at 100°C.

Our brief only requires us to measure between pH 3 and pH 8 between 25°C and 35°C, meaning in the future we could calibrate the probe so that our required range is mapped across the maximum range of voltages which the microcontroller can read, 0V to 3V. This can be done by using a substance of known pH 3, adjusting the offset voltage to 0 and measuring the resistance used. Repeat this for a known pH 8 solution, mapping this to 3V and then find a compromise for the resistance so that the largest possible range can be used. This will improve the accuracy of the pH readings since the microcontroller can utilise more of its measuring capabilities.

2.2 Temperature Control Subsystem

The bioreactor will be used to grow the vaccine using biological agents. If the temperature is too low, the rate at which the vaccine is produced will be too slow. On the other hand, if the temperature is too high the biological agents will be destroyed and the reaction will cease to happen. Therefore, the temperature must be kept constant at an optimum value.

The value for the optimum temperature was set to be in the range 25°C - 35°C. The chosen temperature should be controlled within a precision of ± 0.5°C. This means that if the chosen temperature was 30°C, the actual temperature of the bioreactor shouldn't drop below 29.5°C or rise above 30.5°C.

To achieve this temperature control system, a thermistor was used in conjunction with a heating element. The temperature of the medium can be determined using the thermistor. This data can be fed into a processor which will then decide if the temperature need to be adjusted and will turn the heater on or off accordingly. In our project, we used the MSP 430 Launchpad with the DILMSP430G2553 chip as our controller.

2.2.1 Thermistor

A thermistor is an electrical component that has a resistance which is dependent on temperature. There are two types of thermistors Negative Temperature Coefficient(NTC) thermistors and Positive Temperature Coefficient(PTC) thermistors. They differ in the way their resistances change with temperature. NTC thermistors have a resistance that decreases with increasing temperature whereas PTC thermistors have a resistance that increases with increasing temperature. The thermistor used in the bioreactor system is a 10kΩ NTC thermistor (ND06P00103K). Using data obtained from the datasheet for this thermistor, we were able to formulate an equation to relate resistance and temperature. Further details on how we obtained the equation from the data is given in appendix 2.1.

The temperature sensing system must be able to at least measure temperatures in the range of 25°C - 35°C as this is the desired optimum temperature of the bioreactor. It is also preferable that the system can sense temperatures outside this range to inform the user and other subsystems of the condition. As we are required to maintain the temperature within a margin of ± 0.5°C, the temperature readings should be accurate to at least ±0.1°C. This would ensure that determining if the temperature is within the range is relatively easy.

The circuit used to get readings of the temperature is as follows:

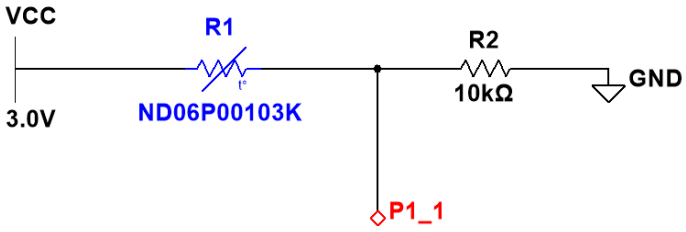


Figure 3: Circuit Diagram for the Temperature Sensing System

This is a simple potential divider circuit with the 10kΩ thermistor in series with a 10kΩ fixed resistor. As the temperature changes the resistance of the thermistor will change. If the thermistor was on its own the Potential difference across it would not change regardless of its resistance. Therefore, a resistor in series with the thermistor causes the resistance ratios to change as the temperature changes. The output to the MSP 430 is taken across R2, the fixed 10kΩ resistor. The value for the fixed resistor was chosen to be 10kΩ after having tried a few other resistances and realising that a resistance of 10kΩ provides reasonable variation in the desired temperature range (10°C - 40°C). This increases the accuracy of the temperature readings.

The analogue measurement of the potential difference across the fixed resistor is converted to a digital value (ADC value). Using this ADC value and the earlier derived Steinhart–Hart equations, it is possible to derive a function that will calculate the temperature from the ADC values. For details on the derivation of this equation see appendix 2.2.

Once we had completed building the circuit and writing the code for the temperature sensing part, we ensured that the readings were calibrated using a digital thermometer by

comparing the readings of the thermometer with our obtained readings. We were able to accurately measure the temperature of the solution between 15°C and 35°C.

2.2.2 Heater

The heating element will be used to increase the temperature of the medium when it is too cold. Electric current passing through the heating element is resisted by the conducting material. The energy that goes into overcoming this resistance in transferred in the form of heat. The heating element that we were provided with was a 3Ω, 30W heating element. Using this information, we can calculate the rated voltage of the heater:

$$P = \frac{V^2}{R}$$

$$V = \sqrt{PR}$$

$$V = \sqrt{(30)(3)} = \sqrt{90} = 9.49\,V\ (3s.f.)$$

The heater takes some time to heat up once switched on and cool off once switched off. This will need to be taken into consideration when integrating the heater with the thermistor and temperature code. The heater will be used to keep the bioreactor within the desired temperature range. It is assumed that the ambient temperature is lower than the desired temperature of the bioreactor. Therefore, the temperature control system will not need a cooling mechanism.

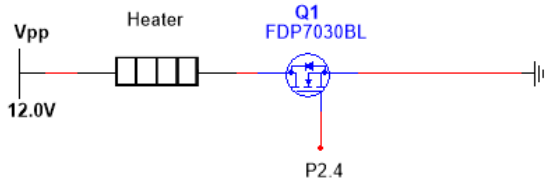


Figure 4: Circuit Diagram for the Heating System

Figure 4 shows the circuit diagram for heating system of the bioreactor. This circuit consists of the 12V power supply, a MOSFET and the heating element. Running the heater at 12V would probably result in the heating element burning out. To overcome this, we exploited the nature of the MOSFET (See Appendix 2.3 for further details). Depending on whether the input to the gate of the MOSFET is high or low, it will allow current to pass through the drain and source.

2.2.3 Heating Control Subsystem Results/Validation

Once both the temperature sensing system and the heating system were functioning properly on their own we went on to integrate them together. This involved programming the MSP 430 to switch off the heater when the temperature rises above a certain point and switch it back on when it gets too low. As the heating element has a bit of delay on it after being switched on or off, it is important to switch off the heater before it hits the desired temperature. This allows the heater to cool off and not overshoot the desired temperature. We decided that 1°C below the desired temperature, the heater should be switched off. This allowed enough time for the heater to cool off. See Appendix 2.4 for the evolution of temperature once the system is switched on.

2.3 Stirring Control Subsystem = Motor Speed Sensor + Motor Driver

The aim of this subsystem is to maintain the Revolution per minute (RPM) of the motor stirring the liquid between 500 and 1500 RPM. This is accomplished by continually measuring the values of the RPM using light photo-interrupter. The brief requires the subsystem to be able to measure and adjust accurately the RPM between 500 and 1500 RPM and this has been successfully applied. The main circuit that compose the stirring subsystem includes an input from the user (RPM value) to the microcontroller and an output to the power supply which controls the amount of the power supplied to the motor so that the RPM can be adjusted.

2.3.1 Motor Speed Sensor

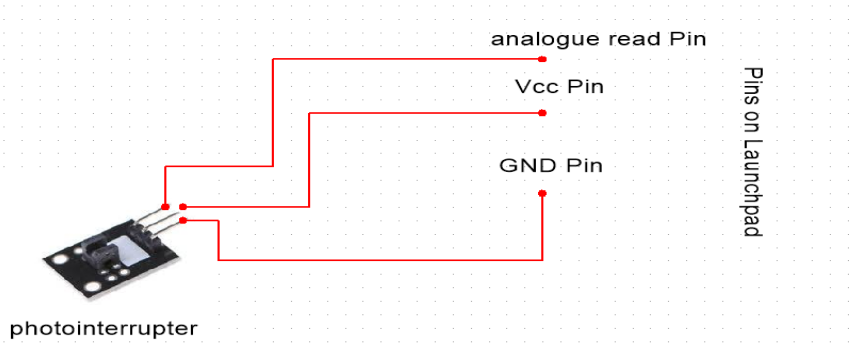


Figure 5: Circuit schematic for motor sensor

The speed sensor measures the speed of the motor in terms of RPM. It is composed of a light photo-interrupter and launchpad. The sensor could recognize the change of light intensity(LI) of the infra-red it receives; once it is connected to the launchpad, there would be some readings on the serial monitor. If LI is big, then the digital readings would be big, and vice versa. A code, applying this property of the photo-interrupter, is designed, and uploaded to record the speed of the motor, see Appendix 3.1. It is required that we should build a subsystem that could monitor the speed of the motor.

As is shown in the circuit above, the middle pin of the photo-interrupter is connected to the Vcc pin on the launchpad so that the launchpad could supply power to the sensor. The other two pins are connected to the analogue read pin and the GND pin respectively so that the whole circuit is connected in series. Once the code written to monitor the speed of the motor is uploaded to the launchpad, there will be readings on the serial monitor.

In the final test of the subsystem, the speed of the motor could be accessed from the serial monitor and the readings has almost the same value as the reading from the speedometer provided by the lab. For example, if the desired speed is set to be 500 RPM, the readings on the serial monitor will be all around 490-510 RPM. Therefore, the design meets the requirement.

2.3.2 Motor Driver

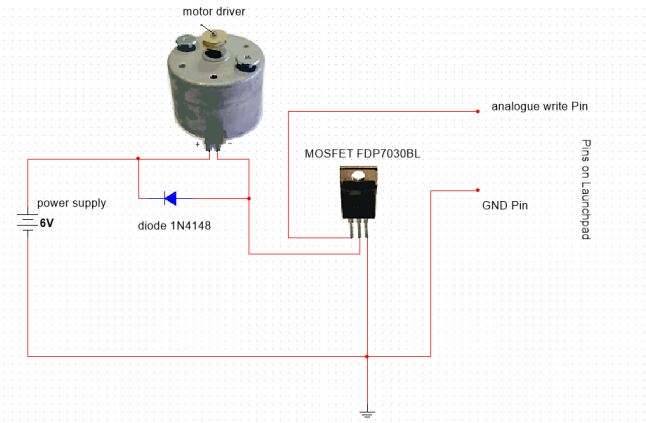


Figure 6: Circuit schematic for motor driver

The speed driver is the part spinning. It is composed of a motor, a diode, an external 6V power supply, a MOSFET switch and the launchpad. The motor could only work under 3V or lower voltages; hence, a MOSFET switch used to control the voltage across it is added to the circuit. Because the motor would generate a backward EMF, which might damage the switch, a diode is connected in parallel to prevent the potential damage. Besides, the switch also acts as an important role in adjusting the speed of the motor; it is connected to one end of the motor and an analogue write pin on the launchpad so that the launchpad could control the speed. It is required that the subsystem should be able to control the speed of the motor at any random value within 500-1500 RPM with an error range of ±20RPM.

As is shown in the circuit above, the switch, power supply and launchpad are connected to the common ground, so that the MOSFET could act as a switch. Drain pin of the MOSFET is connected to the motor and gate pin is connected to the analogue write pin on the launchpad; once the code written to control the speed of motor is uploaded to the launchpad, see Appendix 3.1, and the external power supply is turned on, the motor will run at the speed previously set in the code. However, as is shown in the appendix 3.3, there are some experimental errors.

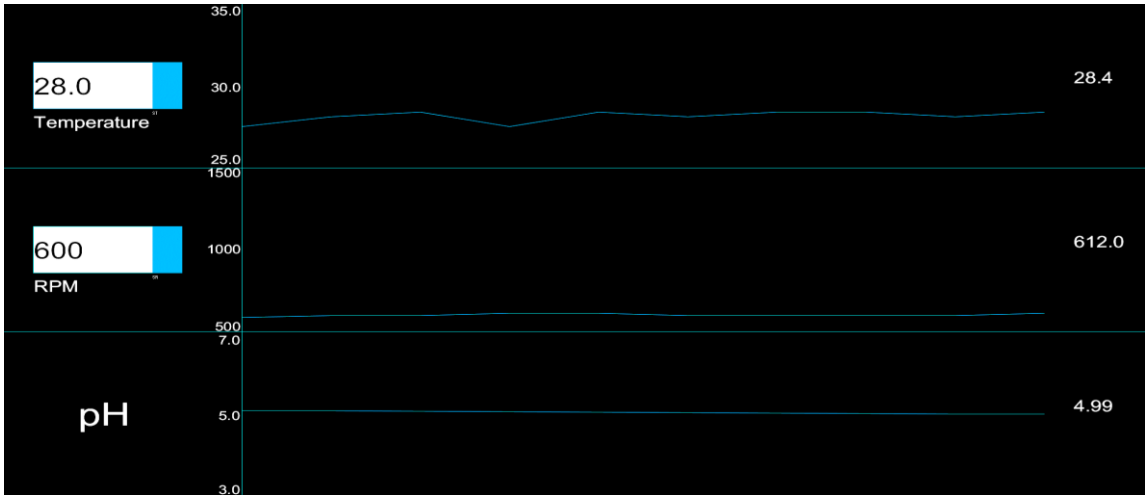
In the final test, the speed is set to 600 RPM, and the speed we got on the serial monitor was 585-615 RPM, which was perfect, and the reading on the speedometer was 597 RPM. However, after we adjusted the speed to 1500 RPM, the readings were 1425-1785 RPM. Therefore, our design meets the requirements if the speed is slow.

2.3.3 Stirring Control Subsystem Results/Validation

After confirming both the motor RPM sensing system and the motor power supplying system were working properly, we integrated the systems together. When a RPM value is given as an input via the user-interface, we programmed the system to check the RPM of the motor continuously and compare it with the input RPM value. If the motor RPM is lower than the input, power supplied to the motor increases and if it is greater, then the power supplied decreases. This process is repeated when the motor is running in order to maintain the RPM of motor. Due to this reason, the RPM value has some fluctuations. However, we decided not to fix the value to the input RPM value because the viscosity of the liquid could change as it is stirred. As for the uncertainty, the greater the input value, the greater range of uncertainty the motor RPM value has. See Appendix 3.2 for a graphical analysis of the motor RPM and its further explanations.

To further improve the subsystem, a voltage regulator is needed so that the readings could be more precise; a more sensible light photo-interrupter is also required to give more accurate analogue read.

3 Overall System Integration and Summary



To summarize the work presented before, our team succeeded in building a functional bioreactor used to grow the vaccine for Tuberculosis. In order for the growth of the vaccine to benefit from optimal conditions, we managed to build a control system that allows the user to control the following parameters: temperature, pH and stirring of the motors. To optimize our design process, we chose to build and test each subsystem separately and then added them together to create one complete system.

The test results of the final prototype were extremely favourable, showing that our control system performs just as expected. Our cultivating medium initially had a pH of 5.4 and temperature 26.9°C (known values). The control subsystem measured a pH of 5(inaccuracy of 0.4 pH) and a temperature of 27.3°C (inaccuracy of 0.4°C). In order to bring the temperature up to a value of 28.5°C the heating element was turned on by the microcontroller. By continually measuring the data of revolutions per minute using a light photo-interrupter the system managed to keep the motor stirring at around a value of 600 RPM. Therefore, we can say that our control system is able to both measure and control the temperature and pH of the solution as well as setting the stirring speed of the motor to a desired range of values.

In conclusion, even though the test results are extremely encouraging, we know that large 'non-ideal' bioreactors will be significantly different than the small laboratory-scale ones. Our concern is that in the bioreactors running under realistic conditions there will be a greater occurrence of noise, which was not previously considered in our pilot. Therefore, in the future, our team acknowledges adding a low pass filter in order to reduce the noise as much as possible. We are confident that by doing so our prototype will get closer to the idea of a non-ideal one utilized in large-scale viral vaccine manufacturing. With Africa having access to only 0.1% of the world's vaccine production, locally produced Tuberculosis vaccines in a large-scale bioreactor could significantly increase that percentage, assuring that the population's health needs are met.

4 Appendices

4.1.1 Appendix 1.1 PH-100ATC Probe

This probe consists of two electrodes. The measurement electrode is placed into an unknown solution while the reference electrode is placed into a standard solution of known pH. There will be a different potential at each probe since they are in separate solutions so a potential difference can be calculated and used to work out the real pH value. The data sheet states that the pH in the standard solution is pH 7.1, however it would have been ideal if we measured the true pH of this solution since it can change over time. If, the pH is different, all our pH values would contain a systematic error, shifting them from the correct value. Unfortunately, the probe needs to be completely dry with no solution (including the buffer solution) on the measurement electrode. Only then using an actual pH meter can we work out the real pH of the standard solution.

4.1.2 Appendix 1.2 Circuit Design/Modelling

The circuit diagram contains all the key elements of the pH probe circuit, including the offset voltage potential divider circuit and the pH measuring circuit. In order to model and test our circuit design, a very high resistance resistor and variable power supply is placed between the two electrodes. This is essentially what the pH probe is made up of and allows us to choose a suitable offset as well as the gain for amplification. The pH probe can produce a maximum range of -500mV to 500mV which we can simulate by varying the measurement voltage output.

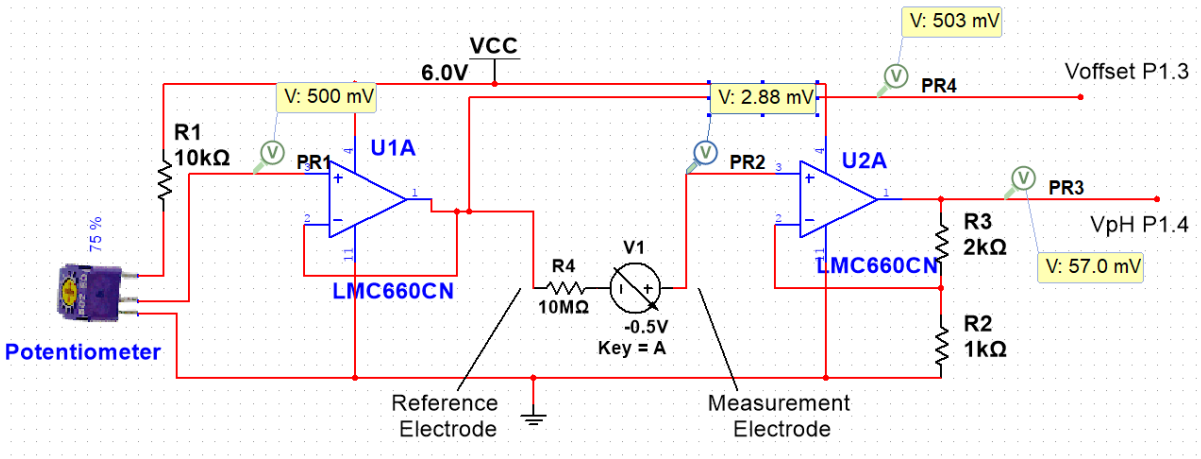


Figure 7: Circuit with Measurement voltage = -500mV

Since we know that the microcontroller can only measure positive voltages we must choose a suitable resistance ratio for the potential divider so that the offset voltage is greater than the minimum voltage of the pH probe. Setting the measurement probe to -500mV as shown in figure 7, allows us to see that the designed circuit would work because at the output pin (P1.4) which goes to the microcontroller has a positive voltage so can be read by the ADC.

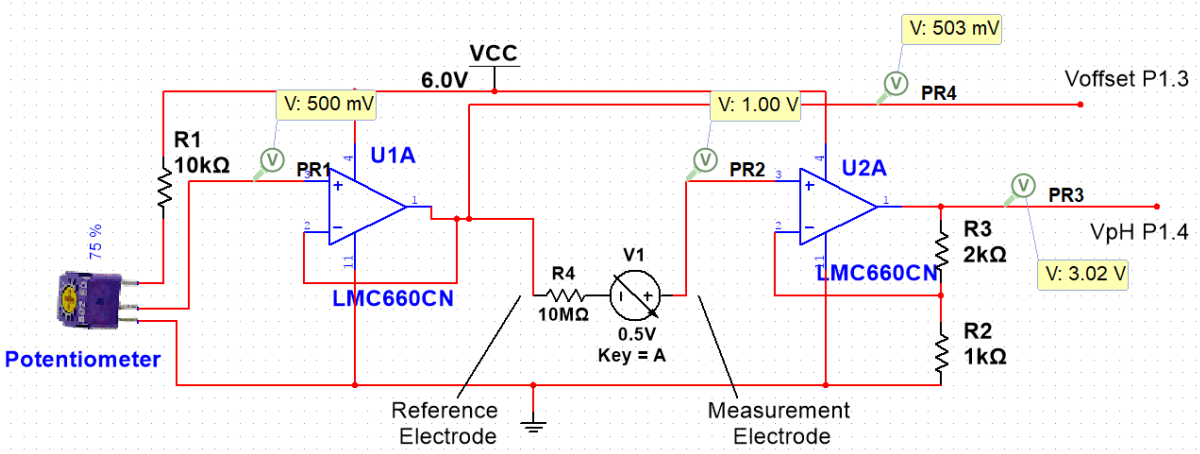


Figure 8: circuit with Measurement voltage = 500mV

We can also test what happens at the maximum measurement electrode voltage (500mV) and choose the appropriate gain for the non-inverting amplifier. From the circuit modelling we were able to conclude that a gain of three is sufficient as it utilises all the microcontroller's input voltage range, since it can only read a maximum of 3V.

4.1.3 Appendix 1.3 Calculation for pH reading

The formula used to calculate the pH is as follows:

$$pH(X) = pH(S) + \frac{(ES - EX)F}{RT\ln(10)}$$

Where

- pH(X) = pH of unknown solution
- pH(S) = pH of standard solution = 7.1
- ES = Electrical potential at reference probe
- EX = Electrical potential at pH measuring probe
- Faraday constant, F = 9.6485309 x 10^4 C mol^{-1}
- Universal gas constant, R = 8.314510 J K^{-1} mol^{-1}
- T = Temperature (Kelvin)

```
// Calculating the pH for this pair of sensor&offset (might be outlier)
double pH_raw = 7.1 + ((Voffset - VpH) * (9.6485309e4)) / (temperature * 8.314510 * log(10));
```

Code used to calculate the pH from the two potential readings at the electrodes. Temperature is taken from the reading that the temperature subsystem calculated.

4.1.4 Appendix 1.4 Reason for 500mV offset voltage

If the probe was placed in a solution of pH 14 at 100°C, then the potential difference across the two probes will be – 518mV which the microcontroller couldn't measure since it can only measure between 0V and 3V. Our idea was to shift the entire spectrum by adding an offset voltage at the reference electrode. The voltage produced by the measuring electrode rides on top of the offset voltage, meaning at the above conditions the voltage at the measuring electrode will be in the positive region (close to 0V) that would be readable by the microcontroller.

4.1.5 Appendix 1.5 Use of potentiometer

Initially, the circuit had a fixed potential divider, consisting of resistors with 11kΩ and a 1kΩ. This should have theoretically supplied an offset voltage of 500mV, however due to the tolerances in the resistors, the actual offset was only 430mV. By replacing the potential divider with a potentiometer, we expected to get a better accuracy of the offset. When calibrating it, we managed to get the offset to the exact value of 500mV. The 10kΩ resistor in series allows the potentiometer to vary the offset voltage between 0V and 3V.

$$\textbf{Maximum Potentiometer Voltage} = \textbf{Supply Voltage} * \frac{R_{series}}{R_{series}+R_{maxPot}} = 6 * \frac{10k}{20k} = 3\ \textbf{V}$$

By adjusting the resistance of the potentiometer and using a digital multi-meter, we could achieve an offset voltage at 500mV. By increasing the offset to its maximum voltage, we believed that the effect of fluctuations would be reduced, but increasing the offset to by 70mV made little impact on our readings. However, we kept the potentiometer for a potential future application.

4.1.6 Appendix 1.6 Reason for offset voltage input

To combat fluctuation errors, we tried adding the offset voltage as an extra input into the microcontroller, instead of setting a constant offset voltage. The fluctuation occurs due to the unstable voltage source so any change in the offset voltage will also lead to an equal change in the measurement voltage. Subtracting these two should eliminate the change caused by the supply, however this was not the case and fluctuations in pH readings still occurred. It was concluded that the voltage readings themselves were over a very small scale (0V to 1V), meaning any small change would cause significant change to the final pH reading. This was addressed in a future iteration where a non-inverting amplifier was utilised.

4.1.7 Appendix 1.7 Use of Buffer Operational Amplifier

The buffer operational amplifier is essential for the voltage offset circuit as it maintains the voltage level of the reference probe at the same level as the output of the potential divider. The key property of an op-amp is that it has incredibly high input resistance, meaning it draws very little current from the power source. The resistance of the pH probe will vary depending on the temperature and the pH of the solution, so if the buffer amplifier was not used, the load resistance (resistance of the entire probe circuit connected to the supply) is comparable to the resistance of the potential divider. This would cause the circuit to draw a significant current away from the potential divider, making the resistance ratio to fail and the desired supply voltage (500mV) to be altered.

By including the op-amp, with its extremely large input resistance, it will draw a tiny current (pA range), which would have very little effect compared to the current through the potential divider (mA range). In effect, it increases the load resistance to a much greater value than the resistance of the potential divider. This allows the potential divider circuit to operate as intended, supplying 500mV to the op-amp which due to the unity gain configuration will also supply the reference probe with 500mV. This means that whatever the changes in the resistance of the pH probe, the potential at the reference probe will always be at 500mV.

4.1.8 Appendix 1.8 Code Error Adjustments

Instead of changing the physical circuit, we decided to make changes within the program in order to reduce fluctuations in pH reading. The idea was to compare the current pH with the previous pH reading. The difference between the two values was calculated and a fraction of this change

was added to the previous reading. In effect, this reduces the impact of the change in the pH readings.

We tested different values for the fraction, starting off with 0.001. This certainly stabilised the pH readings, however it was too slow in changing the pH when the probe was put into a different solution. Fractional value of 0.01 still caused too much variety in readings, but finally a good compromise between stability and adjustment to new solution was found at 0.005.

The final code is as followed:

```
// Weighting the pH value
double get_pH_weighted(double pH_raw)
{
  // Reducing the effect of pH_raw on pH_weighted
  pH_weighted = pH_weighted + (pH_raw - pH_weighted)*0.005;
  return pH_weighted;
}
```

4.1.9 Appendix 1.9 Non-Inverting Amplifier

By amplifying the voltage at the measurement electrode, the changes in the readings due to the fluctuating voltage supply are reduced significantly. Another advantage of the amplifier, is that it allows more of the microcontroller's input reading range to be used. Without the amplifier, the maximum range of voltages is from 0V (500mV - 518mV, pH 14 base at 100°C) to 1V (500mV + 518mV, pH 0 acid at 100°C), however with the amplifier, the range increases from 0V to 3V, utilising the entire readable voltage range. The gain was set to three so that the microcontroller with a voltage reading range of 0V to 3V can still detect the potential difference of a pH 0 acid at 100°C.

4.1.10 Appendix 1.10 pH Probe Testing

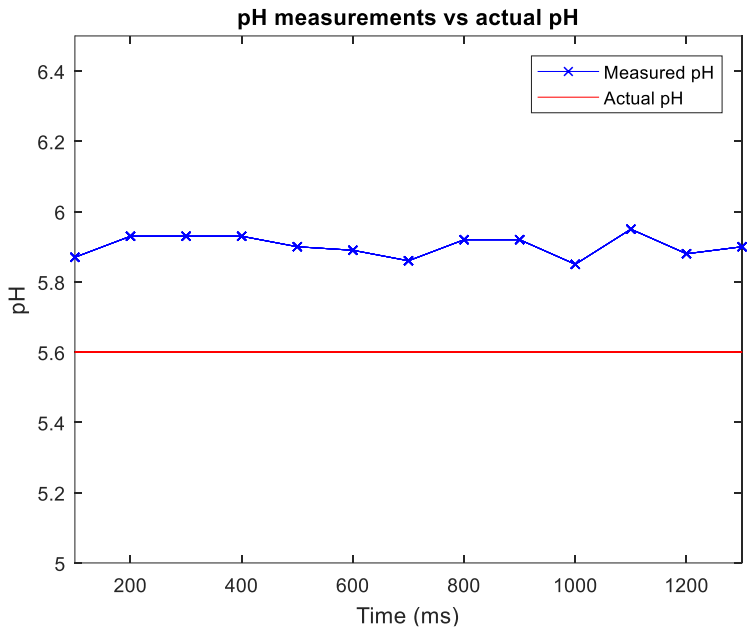


Figure 9: Graph of actual pH and measured pH

Testing the probe in a solution of pH 5.6 gave us the following results. It clearly shows that the implementations discussed before were able to reduce the fluctuations down significantly since our measured values lie in between a very small range (+/- 0.1 pH) around pH 5.9. This offset of pH 0.3 is mainly due systematic error in the probe itself and is perhaps due to the pH of the standard solution within the probe changing (not pH 7.1 as specified), causing a constant shift for all pH readings. This can be solved by adding a voltage calibration into the program to account for this error.

4.1.11 Appendix 1.11 Calibration of Pumps

The pumps have a maximum voltage of 6V, so is not sensible to power them at 6V. Instead we decided to use post width modulation in order to provide a smaller voltage despite using the same 6V supply. This can be achieved by using the analogue write function within the program for the microcontroller. The microcontroller essentially, switches the gate of the MOSFET on and off at a rate defined by the program, meaning the pump is not continuously supplied with 6V. Because the switching is at very high speeds, the voltage over the pump is at a constant value, but lower than 6V since the pump is off at some points in time.

This was a matter of trial and error in finding the required rate for the pumps to operate minimally. The end result was that the acid pump can run at 0.5 x 6 = 3V and the base pump can run at 0.8 x 6 = 4.8V. The difference is due to the thickness of the tubes in the pumps, since the base pump has thicker tubes so there is more friction, requiring more energy to move liquid through the tubes.

However, a final adjustment was done to improve the reliability of the system which, was to power the pumps at maximum voltage for a very short amount of time (10ms). Initially, when the pump is stationary, it requires more energy to start rotating. We were able to run the pumps so that they inject a few (3-4) drops of base/acid per cycle of the program.

The final code is as followed:

```
if (pH_weighted > pH_TARGET + pH_RANGE)
{
  analogWrite(BASE_PUMP_PORT, 0);
  if (IsPumpOn == false)
  {
    digitalWrite(ACID_PUMP_PORT, HIGH);
    delay(10);
    analogWrite(ACID_PUMP_PORT, 0.5 * 255);
    IsPumpOn = true;
  }
  else
  {
    analogWrite(ACID_PUMP_PORT, 0);
    IsPumpOn = false;
  }
}
else if (pH_weighted < pH_TARGET - pH_RANGE)
{
  analogWrite(ACID_PUMP_PORT, 0);
  if (IsPumpOn == false)
  {
    digitalWrite(BASE_PUMP_PORT, HIGH);
    delay(10);
    analogWrite(BASE_PUMP_PORT, 0.8 * 255);
    IsPumpOn = true;
  }
  else
  {
    analogWrite(BASE_PUMP_PORT, 0);
    IsPumpOn = false;
  }
}
else
{
  analogWrite(ACID_PUMP_PORT, 0);
  analogWrite(BASE_PUMP_PORT, 0);
  IsPumpOn = false;
}
```

4.1.12 Appendix 1.12 Pump Operation Testing

Test no 1
4.68
A drop of base solution
4.75
A drop of base solution
4.79
A drop of base solution
4.84

Table 1: Pump test for acidic solution

Test no 2
5.90
A drop of acid
5.82
A drop of acid
5.74
A drop of acid
5.68

Table 2: Pump test for alkaline solution

When the pH probe was placed in an acidic solution, the microcontroller would apply a voltage to the gate of the MOSFET connected to the base pump, allowing current to pass through and turns on the pump. We calibrated it so that only a few drops of the solution would be injected so that the pH value doesn't change dramatically. From table 1, we can see that after a few cycles the measured pH increases towards the desired pH 5.

Similarly, when the probe was placed in an alkaline solution, a voltage is applied to the gate of the MOSFET connected to the acid pump, turning on the pump and injecting a couple drops of acid into the solution. This clearly shows that the pumps worked as expected.

4.1.13 Appendix 1.13 Integration of pH Subsystem Testing

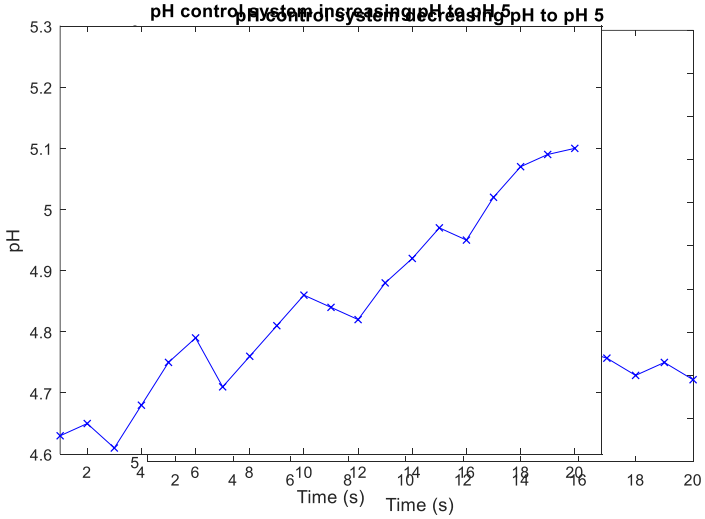


Figure 11: Graph showing control system pumping acid solution to decrease pH of unknown solution to pH 5

Figure 10 clearly shows that the control system can increase the pH of the solution to the desired pH (5) for optimum vaccine growth. The solution initially starts at around pH 4.64 but the control system turns on the base pump, causing the pH to increase and begins to stabilise at around pH 5. The fluctuations caused during the increase is mainly due to the pumps running. The pumps and the potential divider circuit both use the same 6V power supply. When the pumps draw current from the supply, the supply is no longer able to provide a constant 6V. This would affect the offset voltage and hence affect the pH reading. This can also be seen in figure 11 where there is some variation in pH as it decreases to pH 5.

A possible solution to this is to use a separate power supply for each element of the circuit, however this would add extra complexity as well as cost in installing a new power supply. The optimum solution is to implement a voltage rectifying circuit to the reference probe, using a Zener diode. The Zener diode will supply a constant voltage to the reference probe because even small

changes in current through the diode will cause little change in potential drop, which is significant in a potential divider circuit.

4.2.1 Appendix 2.1 Modelling resistance against temperature of thermistor

Temperature (°C)	Resistance (Ω)	Temperature (°C)	Resistance (Ω)
-55	1214000	50	3379
-50	833500	55	2769
-45	579200	60	2281
-40	407200	65	1890
-35	289500	70	1573
-30	208000	75	1316
-25	151000	80	1106
-20	110700	85	933.7
-15	81970	90	791.8
-10	61230	95	674.3
-5	46150	100	576.6
0	35080	105	495
5	26880	110	426.6
10	20760	115	369.1
15	16160	120	320.4
20	12670	125	279.1
25	10000	130	243.9
30	7949	135	213.9
35	6359	140	188.1
40	5120	145	166
45	4148	150	146.9

Table 3: Resistive values of the thermistor for different temperatures taken from the datasheet for the ND06P00103K.[1]

[1] NTC Disc Thermistors' Datasheet(AVX) - http://datasheets.avx.com/NTC_DiscThermistors.pdf

There are multiple approaches of modelling the temperature-resistance relationship of thermistors. The method that we used is called the Steinhart–Hart equations. This equation is in the following form:

$$\frac{1}{T} = A + B\ln(R) + C(\ln(R))^3$$

Where T is the temperature in Kelvin and R is the Resistance at that temperature. The remaining variables A, B and C are constants unique to the thermistor. To calculate these constants, we used the data from the table above. As there are three constants to be determined we picked 3 data points and formed 3 equations to solve simultaneously:

$$(0,35080)\ ,(20,12670)\ ,(50,3379)$$

$$\frac{1}{0+273.15} = A + B\ln(35080) + C(\ln(35080))^3$$
$$\frac{1}{20+273.15} = A + B\ln(12670) + C(\ln(12670))^3$$
$$\frac{1}{30+273.15} = A + B\ln(3379) + C(\ln(3379))^3$$

Solving these 3 equations is most easily accomplished by matrix multiplication:

$$\begin{pmatrix} 1 & \ln(35080) & (\ln(35080))^3 \\ 1 & \ln(12670) & (\ln(12670))^3 \\ 1 & \ln(3379) & (\ln(3379))^3 \end{pmatrix}\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} \frac{1}{0+273.15} \\ \frac{1}{20+273.15} \\ \frac{1}{30+273.15} \end{pmatrix}$$

$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} 1 & \ln(35080) & (\ln(35080))^3 \\ 1 & \ln(12670) & (\ln(12670))^3 \\ 1 & \ln(3379) & (\ln(3379))^3 \end{pmatrix}^{-1}\begin{pmatrix} \frac{1}{0+273.15} \\ \frac{1}{20+273.15} \\ \frac{1}{30+273.15} \end{pmatrix}$$

Using MATLAB to calculate the multiplication we get:

$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} 1.26368563274486 \times 10^{-3} \\ 2.19645005235890 \times 10^{-4} \\ 8.60542729501152 \times 10^{-8} \end{pmatrix} \approx \begin{pmatrix} 1.2637 \times 10^{-3} \\ 2.1965 \times 10^{-4} \\ 8.6054 \times 10^{-8} \end{pmatrix} [5s.f.]$$

Thereby we deduce that the equation linking temperature and resistance for the ND06P00103K is:

$$\frac{1}{T} = (1.2637 \times 10^{-3}) + (2.1965 \times 10^{-4})\ln(R) + (8.6054 \times 10^{-8})(\ln(R))^3$$

4.2.2 Appendix 2.2 Converting ADC values to temperature readings

The measured output is an analogue value. The microcontroller converts this analogue value to a digital value. This is known as an ADC conversion. During this process, the range of potentials 0 – 3V is mapped across digital values 0-1023. The output to the MSP 430 is the potential difference across the fixed resistor which can be calculated as follows:

$$V_o = V_{cc}\left(\frac{10k}{R_T + 10k}\right)$$

The MSP 430 maps the potential values from 0 - V_{cc} over the digital values 0 – 1023. Using this fact, we can deduce the following:

$$A_{DC} = \frac{V_R}{V_{CC}}(1023)$$

Eliminating V_R:

$$A_{DC} = \left(\frac{10k}{R_T + 10k}\right)(1023)$$

Rearranging for R_T:

$$R_T = \frac{A_{DC}}{10230000} - 100000$$

Previously derived Steinhart–Hart equations:

$$\frac{1}{T} = (1.2637 \times 10^{-3}) + (2.1965 \times 10^{-4})\ln(R) + (8.6054 \times 10^{-8})(\ln(R))^3$$

Substituting in R_T for R:

$$\frac{1}{T} = (1.2637 \times 10^{-3}) + (2.1965 \times 10^{-4})\ln(R_T) + (8.6054 \times 10^{-8})(\ln(R_T))^3$$

$$where \quad R_T = \frac{A_{DC}}{10230000} - 100000$$

Final Equation:

$$T = \frac{1}{= (1.2637 \times 10^{-3}) + (2.1965 \times 10^{-4})\ln(R_T) + (8.6054 \times 10^{-8})(\ln(R_T))^3}$$

$$where \quad R_T = \frac{A_{DC}}{10230000} - 100000$$

4.2.3 Appendix 2.3 Powering the heating element

The external power supply available to us had voltages of 12V and 6V. It was decided that we would use the 12V supply and limit the current to the heating element. Initially we were going to use the FDP7030BL MOSFET as a variable voltage supply by supplying the correct voltage to the gate. However, the MSP430 is not able to output analogue voltages. Following this we decided to run the heater at 12V but limit the current flowing through it by using Pulse Width Modulation. By changing the duty cycle of the modulation, we can control the amount of current that flows through the heater. By doing so we could decide on a value for the duty cycle and thereby safely power the heater:

Current that would be drawn is run using 12V supply:

$$I = \frac{V}{R} = \frac{12}{3} = 4\ A$$

Current for which the heating element is rated at:

$$P = I^2R$$

$$I = \sqrt{\frac{P}{R}} = \sqrt{\frac{30}{30}} = \sqrt{10} = 3.16\ A\ (3s.f.)$$

Calculating duty cycle:

$$\% = \frac{3.16}{4} \times 100 = 79\%$$

Calculating the corresponding parameter for the analogWrite function:

$$p = \frac{79}{100} \times 255 = 201.45$$

Therefore, to run the heater at its optimum, a parameter value of 201 should be used. Having done further tests, we found it hard to predict how quickly the heater cools down when it operates at this rate. This made it difficult to state when the heater should be switched off. Hence, we decided to go with a lower value for the parameter to allows for a gentler heating and cooling which is more predictable.

4.2.4 Appendix 2.4 How temperature changes in the bioreactor

The graph below shows how the temperature of the bioreactor changed from initial start-up at around 19°C:

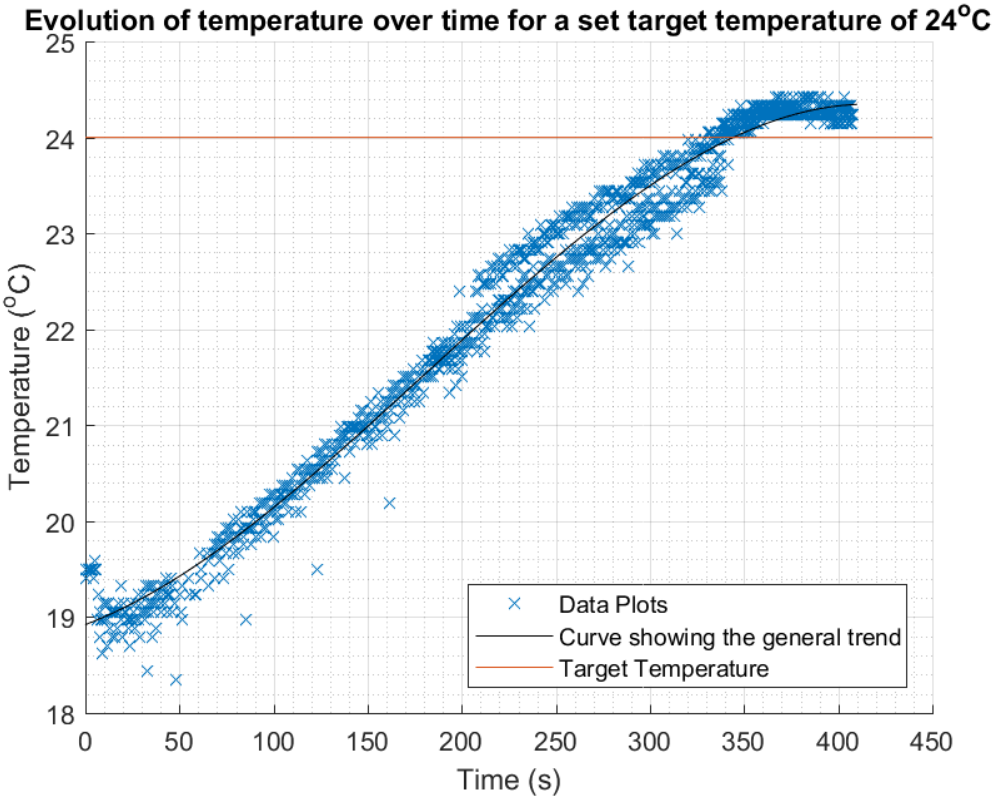


Figure 12: Graph showing how the temperature of the bioreactor changes once started up at a relatively low temperature.

The delay in switching of the heater is clearly portrayed by the graph above. There is no significant change in temperature in the first 20 seconds despite the temperature being below the target temperature. As discussed previously, this is because the heater takes some time to warm up and start heating the surrounding water. After 20 seconds, the temperature slowly increases to a maximum rate of increase at which the temperature continues to rise until a temperature of 23.5°C is reached. At 23°C, the heater is switched off and the heating element will start to cool down causing the rate of increase of the temperature to fall until eventually it settles at a temperature. This is represented by the decreasing gradient.

4.3.1 Appendix 3.1 Code for measuring RPM and controlling speed

```
void updateRpm ()
{
    period = 2 * (millis() - prevmillis);
    prevmillis = millis();
    rpm = 60000 / period;
}

void setup()
{
  Serial.begin(9600);
  pinMode(sensorIn, INPUT_PULLUP);
  pinMode(motorOut, OUTPUT);

  attachInterrupt(sensorIn, updateRpm, RISING);
  interrupts();
}
```

This is the code to measure the RPM

```
// Adjusting the speed of motor
void adjustRpm()
{
  Serial.println(rpm);
  if (desiredRpm > 1500 || desiredRpm < 500)
  {
    analogWrite(motorOut, 0);
    return;
  }

  if (rpm > (desiredRpm+10) && power > 0)
    power--;
  if (rpm < (desiredRpm-10) && power < 255)
    power++;
  analogWrite(motorOut, power/2);
}
```

This is the code used to control the motor speed.

Since we used the 6V power supply while the maximum voltage across the motor should be 3V, we set the final output power to be 'power/2' so that the motor won't break down.

4.3.2 Appendix 3.2 Final results of stirring

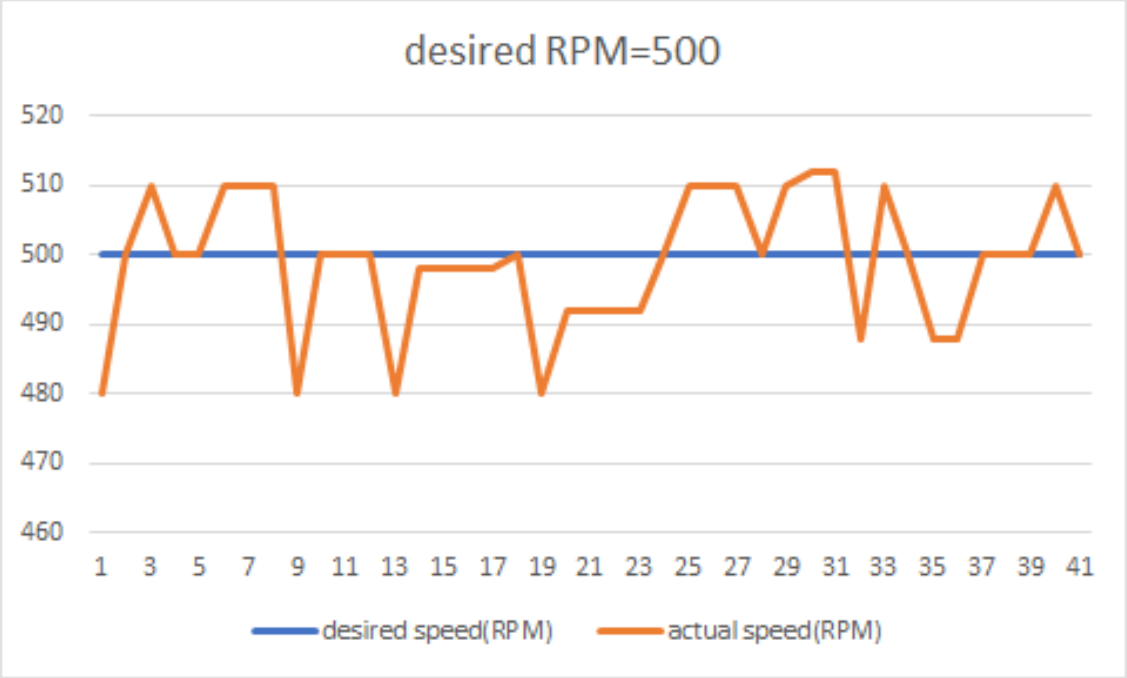


Figure 13: Graph showing how the RPM of the motor changes when a relatively low RPM is given as an input

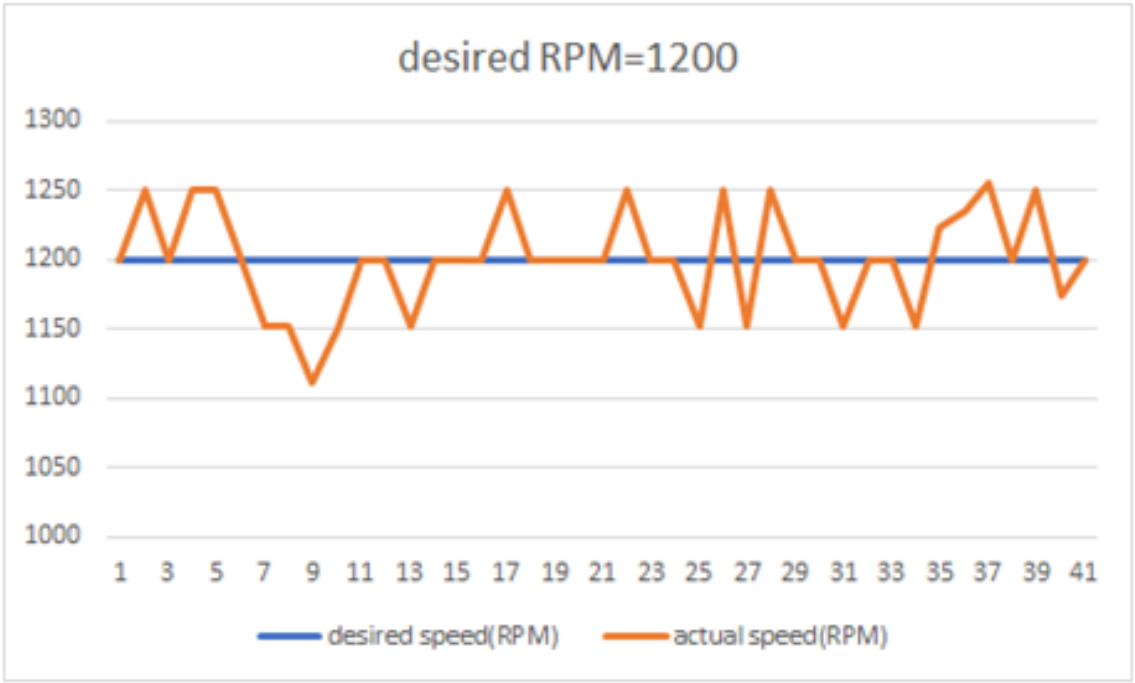


Figure 14: Graph showing how the RPM of the motor changes when a relatively high RPM is given as an input

Figure 13 shows that when a small number is given as an input, the value of the motor RPM is relatively accurate with a small range of uncertainty. The graph shows that the uncertainty is +/- 20 RPM. On the other hand, figure 14 shows the range of uncertainty increases to +/-100 RPM when a greater input value is given. This is due to the existence of a percentage uncertainty.

4.3.3 Appendix 3.3 Results and error estimation

The results fit the requirements perfectly when the motor spins at low speed ranging from 500-1200 RPM; however, the results don't fit the requirements when the motor spins at low speed. For example, 500 RPM in the code is set as the desired RPM, then the motor would run with speed being around 490-510 RPM which would be shown on the serial monitor. This would run perfectly with the error range being ±20 at speed slower than 1100 RPM. However, because of the inaccuracy of the equipment, if the speed is set faster than 1200RPM, the error range would be around ±50-±200, increasing with speed. This could not be modified by changing the circuit or code.