UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

MÁSTER UNIVERSITARIO EN INGENIERÍA BIOMÉDICA
TRABAJO FIN DE MÁSTER

DESIGN AND CONSTRUCTION OF AN INITIAL PROTOTYPE OF A NORMOTHERMIC PERFUSION SYSTEM FOR THE PRESERVATION AND POSSIBLE REHABILITATION OF A KIDNEY

INÉS FERNÁNDEZ ABALDE

2022

Universidad Politécnica de Madrid

Escuela Técnica Superior de
Ingenieros de Telecomunicación

MÁSTER UNIVERSITARIO EN INGENIERÍA BIOMÉDICA

TRABAJO FIN DE MÁSTER

Design and construction of an initial prototype of a normothermic perfusion system for the preservation and possible rehabilitation of a kidney

Author:
Inés Fernández Abalde

Mentors:
Victoria Gómez Dos Santos

Álvaro Gutiérrez Martín

June 2022

Resumen

El trasplante renal es un procedimiento quirúrgico en el que se coloca un riñón sano, procedente de un donante vivo o fallecido, en un paciente con insuficiencia o fallo renal. Según datos de la Organización Nacional de Trasplantes, España tiene una tasa de 40,2 donantes por millón de población (p.m.p.), la más alta del mundo. En 2021 se realizaron 2.950 trasplantes renales, un 9,2% más que el año anterior. De ellos, 2.627 fueron trasplantes de donantes fallecidos y 323 de donantes vivos.[1]

Con el paso de los años, la tasa de donantes jóvenes y sin patologías previas está disminuyendo, debido a que el número de donantes que fallecen en accidentes de tráfico es cada vez menor, siendo en la actualidad sólo el 4,7%. La principal causa de muerte de los donantes son las lesiones cerebrovasculares. En cuanto a la edad de los donantes, más de la mitad (53,7%) tienen más de 60 años, el 27,5% más de 70 y el 3,8% más de 80. Esto hace que los riñones trasplantados estén más deteriorados o hayan sufrido patologías previas. [1]

Actualmente, los riñones para trasplante se conservan en un sistema de perfusión hipotérmica. Esta técnica mantiene el órgano recién extraído en una solución hipotérmica durante el menor tiempo posible antes del trasplante. Los órganos se lavan y perfunden con una solución de conservación para conseguir una temperatura baja homogénea. Esta solución intenta ralentizar y detener todos los procesos de degradación celular y la actividad enzimática. Las líneas modernas de investigación han demostrado que el tratamiento previo del órgano con sangre y en condiciones normales de temperatura podría mejorar aún más la función post-trasplante, preservando tanto la estructura como la funcionalidad de sus células para superar las limitaciones actuales del trasplante de órganos.

Teniendo en cuenta la necesidad de mejorar la preservación de los injertos renales y el nicho comercial existente para este tipo de dispositivos surge este Trabajo Fin de Máster. Su objetivo principal es diseñar y construir un primer prototipo de un sistema de perfusión normotérmica para la preservación y posible rehabilitación de un riñón. Para lograr este objetivo, se diseña un sistema de circulación cerrada compuesto por una bomba centrífuga, un sensor de flujo y un sensor de temperatura. El sistema monitoriza los datos de perfusión y, mediante el software de control integrado, se autorregula modificando la velocidad de bombeo de la bomba centrífuga. Este sistema actúa cuando alguno de sus parámetros se sale del rango correcto, pero también puede ser controlado por un operador a través de la interfaz de usuario implementada.

Este dispositivo es de fácil uso, asequible y con plena integración del control informático. Permite la adaptación autónoma del sistema a las necesidades y características del órgano, aliviando la carga de trabajo del operador del dispositivo. Además, proporciona una plataforma amigable, capaz de trabajar por sí sola durante largos periodos de tiempo sin necesidad de supervisión humana.

Palabras clave

Perfusión, máquina de perfusión normotérmica, transplante renal, dispositivo autorregulado, software de control.

Summary

Renal transplantation is a surgical procedure in which a healthy kidney, from a living or deceased donor, is transplanted into a patient with renal insufficiency or failure. According to data from the National Transplant Organisation, Spain has a rate of 40.2 donors per million population (p.m.p.), the highest rate in the world. In 2021, 2,950 kidney transplants were performed, 9.2% more than the previous year. Out of these, 2,627 were transplants from deceased donors and 323 from living donors. [1]

Over the years, the rate of young donors and donors without previous pathologies is decreasing, due to the fact that the number of donors who die in traffic accidents is declining, currently it is only 4.7%.The main cause of death of donors is cerebrovascular injury. Regarding the age of donors, more than half (53.7%) are over 60 years old, 27.5% are over 70 and 3.8% are over 80. This causes transplanted kidneys to be more deteriorated or to have suffered from previous pathologies. [1]

Currently, kidneys for transplantation are preserved in a hypothermic perfusion system. This technique keeps the freshly extracted organ in a hypothermic solution for as short as possible before transplantation. The organs are washed and perfused with a preservation solution to achieve a homogenous low temperature. This solution attempts to slow down and stop all cell degradation processes and enzymatic activity. Modern lines of research have shown that pre-treatment of the organ with blood and under normal temperature conditions could further improve post-transplantation function, preserving both the structure and functionality of its cells to overcome current limitations on organ transplantation.

This Master Thesis intends to fulfill the need to improve the preservation of renal grafts and the existing commercial niche for this type of device. Its main objective is to design and build an initial prototype of a normothermic perfusion system for the preservation and possible rehabilitation of a kidney. To achieve this goal, a closed circulation system is designed consisting of a centrifugal pump, a flow sensor and a temperature sensor. The system monitors perfusion data and, by means of the integrated control software, self-regulates by modifying the pumping speed of the centrifugal pump. This system acts when any of its parameters goes out of the correct range but can also be controlled by an operator through the implemented user interface.

This device is easy to use, affordable and with full integration of computer control. It allows autonomous adaptation of the system to the needs and characteristics of the

organ, easing the workload of the device operator. It also provides a user-friendly platform capable of working on its own for long periods of time without the need for human supervision.

Keywords

Perfusion, normothermic perfusion machine, renal transplantation, autoregulated device, control software.

Agradecimientos

Me gustaría agradecer en primer lugar a los tutores de este Trabajo de Fin de Máster, Victoria Gómez Dos Santos y Álvaro Gutiérrez Martín. A Victoria, por haberme dado la oportunidad de realizar este proyecto tan interesante y a Álvaro por prestarme ayuda siempre que lo he necesitado.

Gracias a mi familia por el esfuerzo que han hecho para darme la posibilidad de cursar este Máster.

Y por último, gracias a mis amigos, en especial a Ventura y Juan por haberme apoyado y animado tanto este curso.

Contents

List of Figures

List of Tables

1 Introduction and objectives

This Master Thesis has been carried out in collaboration with the Hospital Universitario Ramón y Cajal, specifically with the Urology Department. The work is framed within the project *Modelo Traslacional de Perfusión Normotérmica Ex vivo para la evaluación de la Viabilidad y la Rehabilitación de Injertos Renales Pretrasplante* with funding from the Fundación Mutua Madrileña (FMM) 2020 grant for research projects in health, specifically in the area of transplantation.

The Fundación Mutua Madrileña, in its 17th call for research grants, has earmarked 2.3 million euros to support medical research in Spain. This project belongs to the category *Vías de mejora en la donación de órganos para trasplante*, led by Dr. Victoria Gómez, Head of Section in the Urology Service of the Hospital Universitario Ramón y Cajal and member of the Group *Investigación quirúrgica en urología y trasplante renal* included in Area 3 of the Instituto Ramón y Cajal de Investigación Sanitario (IRYCIS), whose head is Dr. Francisco Javier Burgos, also Head of the Urology Service and member of the project. The project has a duration of three years and has a clear multidisciplinary character, international projection and is supported by the Organización Nacional de Trasplantes (ONT). [2]

1.1 Motivation

Kidney transplantation is a surgical procedure that involves placing a healthy kidney from a living or deceased donor into a person whose kidneys no longer function properly, i.e. who has kidney failure.

In the second half of the 20th century, organ transplantation has been a revolution in the world of medicine, becoming a daily activity in hospitals. The Organización Nacional de Trasplantes (ONT) recorded a total of 2,950 renal transplants in Spain in 2021, as it is shown in Figure 1[1]. This has been possible thanks to advances in surgical techniques, the development of new immunosuppressive drugs and the improvement of organ preservation techniques. These techniques have had a decisive impact on organ viability and transplant success, increasing graft survival.

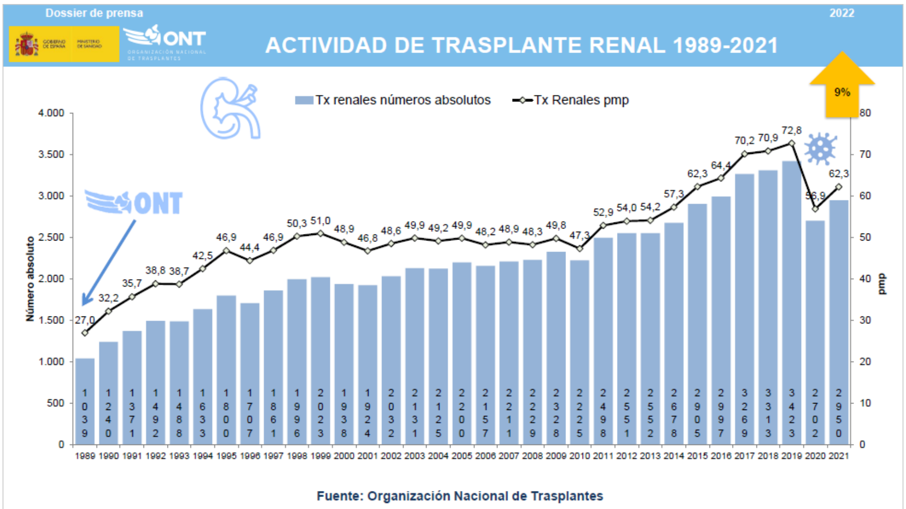


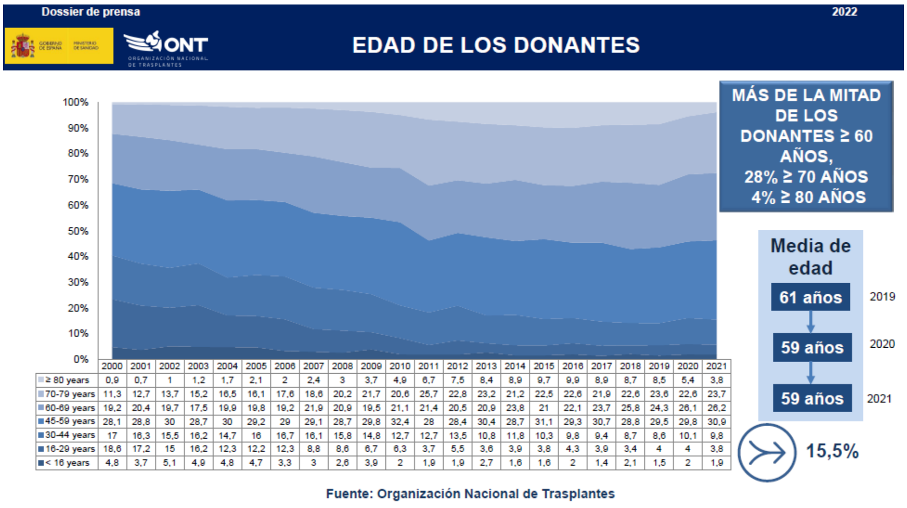
Figure 1: Growth in the number of kidney transplants [1]

An ongoing concern is the demand for transplantation, which far exceeds the number of available donor organs. One of the reasons for the shortage of organs available for transplantation is due to the large percentage of discarded grafts. The vast majority of which are not considered optimal for use and may cause complications or inferior survival. Although the donation rate achieved in Spain, of 40.2 per million population (p.m.p.), far exceeds that reported by the rest of the countries in the world in 2020 (Figure 2), the number of young healthy donors who died in traffic accidents remains very low, at only 4.7%. The main cause of death of donors is stroke. In terms of donor age, more than half (53.7%) are over 60, the 27.5% are over 70 and the 3.8% are over 80. The advanced age of donors leads to a higher number of discarded grafts, which further aggravates the problem of the lack of available donors. (Figure 3)[1].

Research teams around the world are working on new approaches to overcome the problem of high numbers of graft discards, maintaining organ viability during preservation. The conventional technique keeps the freshly harvested organ in a hypothermic solution for as short a time as possible before transplantation. The organs are washed and perfused with a preservation solution to reach a homogenous low temperature as soon as possible, the temperature of the applied solution attempts to slow down and stop all cell degradation processes by decreasing enzymatic activity.

The conventional preservation process is based on the idea that the organ can live longer under the previously explained conditions. But, nowadays, lines of research have

Figure 2: Organ donation rates in Spain compared to other countries [1]



	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
≥ 80 years	0,9	0,7	1	1,2	1,7	2,1	2	2,4	3	3,7	4,9	6,7	7,5	8,4	8,9	9,7	9,9	8,9	8,7	8,5	5,4	3,8
70-79 years	11,3	12,7	13,7	15,2	16,5	16,1	17,6	18,6	20,2	21,7	20,6	25,7	22,8	23,2	21,2	22,5	22,6	21,9	22,6	23,6	22,6	23,7
60-69 years	19,2	20,4	19,7	17,5	19,9	19,8	19,2	21,9	20,9	19,5	21,1	21,4	20,5	20,9	23,8	21	22,1	23,7	25,8	24,3	26,1	26,2
45-59 years	28,1	28,8	30	28,7	30	29,2	29	29,1	28,7	29,8	32,4	28	28,4	30,4	28,7	31,1	29,3	30,7	28,8	29,5	29,8	30,9
30-44 years	17	16,3	15,5	16,2	14,7	16	16,7	16,1	15,8	14,8	12,7	12,7	13,5	10,8	11,8	10,3	9,8	9,4	8,7	8,6	10,1	9,8
16-29 years	18,6	17,2	15	16,2	12,3	12,2	12,3	8,8	8,6	6,7	6,3	3,7	5,5	3,6	3,9	3,8	4,3	3,9	3,4	4	4	3,8
< 16 years	4,8	3,7	5,1	4,9	4,8	4,7	3,3	3	2,6	3,9	2	1,9	1,9	2,7	1,6	1,6	2	1,4	2,1	1,5	2	1,9

Figure 3: Age of organ donors [1]

shown that pre-treatment of the organ with blood and under normal temperature conditions could further improve post-transplantation function. Current organ preservation principles and techniques are reviewed below.

1.2 Preservation methods

Maintaining organ viability from harvest to transplantation is a crucial factor for proper function and graft survival. Organ damage occurs mainly as a result of ischaemia-reperfusion injury. To minimise this damage, different organ preservation techniques are used to optimise organ function once perfusion is restored. Organ damage during transplantation occurs in two phases:

1. Warm ischaemia phase: time elapsed from the moment the circulation of the donated organ is interrupted until it is perfused with the hypothermic preservation solution.

2. Cold ischaemia phase: the period of time between the organ being preserved in a hypothermic state and its transplantation into the receptor.

Ischaemia is the cellular stress caused by a decrease in blood flow, the consequent decrease in oxygen supply (hypoxia) and the elimination of products of tissue metabolism. Severe ischaemia can lead to anoxia, which means that the tissues in that region will not have the necessary energy to survive and tissue cell death (necrosis) occurs. [3]

A large part of the injury to transplanted organs does not occur during ischaemia, but during reperfusion. Reperfusion injury is defined as the adverse effect of restoring circulation and allowing blood and nutrients to reach previously ischaemic cells. Restoration of blood flow results in the removal of accumulated toxic metabolites. Although reperfusion is necessary for organ recovery after ischaemic injury, the release of these toxic metabolites into the systemic circulation can have metabolic consequences, triggering inflammatory processes that directly stimulate the receiver's immune system, contributing significantly to the development of acute graft rejection. [4]

The foundation of organ preservation is based on the suppression of metabolism and catabolic enzymes by hypothermia at 4ºC. Hypothermia has been shown to slow enzyme activity thereby reducing the probability of ischaemic injury. The most significant adverse effect of hypothermia is the increase in cellular edema, which results in excessive accumulation of fluid in the interstitial space caused by an imbalance between the forces that regulate the movement of fluid from one compartment to another. [5]

Organs exposed to normothermic ischaemia (over 37°C) remain viable for relatively short periods, usually less than an hour. However, by cooling them to 2 to 4°C, preservation time can be extended for a period of 12 to 13 hours [6]. This period can be significantly extended if an appropriate preservation solution is used. There are numerous preservation solutions but their goals are the same: to prevent cellular edema, delay cell destruction and maximise organ function once perfusion is restored.

The most commonly used hypothermic preservation techniques today are cold preservation and preservation in a hypothermic perfusion machine. These techniques will be explained below, as well as the normothermic preservation technique used in this project.

- **Cold preservation**: the most common and least expensive method of preservation. The organ is perfused with a cold preservation solution immediately after harvesting. It is stored at 4°C in this solution until implantation. It has a number of advantages, such as its almost universal availability and its easy transport. However, it is questionable whether this method is able to prevent deterioration of the quality of organs in the expanded criteria donor group. [7]

- **Preservation in a hypothermic perfusion machine**: after the initial wash in the operating room, the organ is introduced into a device that maintains a controlled continuous or pulsed flow of cold preservation solution (from 0 to 4ºC). This flow allows the circulatory stream to be cleared of microthrombus and facilitates the elimination of metabolic end products. Its advantages are a lower incidence of delay in the initial function of the graft, the possibility of assessing its viability in real time and the possibility of providing metabolic (oxygen and substrates) or pharmacological support during perfusion.[8]

- **Normothermic preservation**: the aim of normothermic perfusion is to recreate the normal physiological environment of the kidney outside the body to avoid hypoxia and ischaemia. Nowadays, research lines have shown that pre-treatment of the organ with blood and under normal temperature conditions could further improve post-transplantation function [9]. This system preserves temperature and oxygen supply to assess the viability of high-risk grafts and their physiological recovery prior to transplantation. This new preservation system validates non-invasive, sensitive and reliable biomarkers that would allow determination of graft viability, selection of the appropriate receptor and prediction of post-transplant outcome. [9]

1.3 Normothermic perfusion devices

This Master Thesis covers an area with a recent history of research so there is not a large supply of commercially available systems. It is an area in continuous expansion with a high demand for new alternatives and continuous progress.

Currently, there are several hypothermic kidney perfusion machines, such as: Life-Port Kidney Transporter from the manufacturer Organ Recovery System [10], RM3 Kidney Preservation System from the manufacturer Waters Medical Systems [11] or the Airdrive HMP device from the manufacturer Portable Organ Perfusion [12]. In contrast, there is only one commercial device available for ex vivo renal perfusion in normothermia, the XVIVO Kidney Assist machine.

The XVIVO Kidney Assist [13] is a dedicated device for ex vivo, oxygenated, pressure-controlled perfusion of donor kidneys. An integrated heater/cooler provides perfusion ranging from hypothermic to normothermic temperatures, from 12 to 37°C. The components that make up the system and their functionalities are shown in Figure 4.

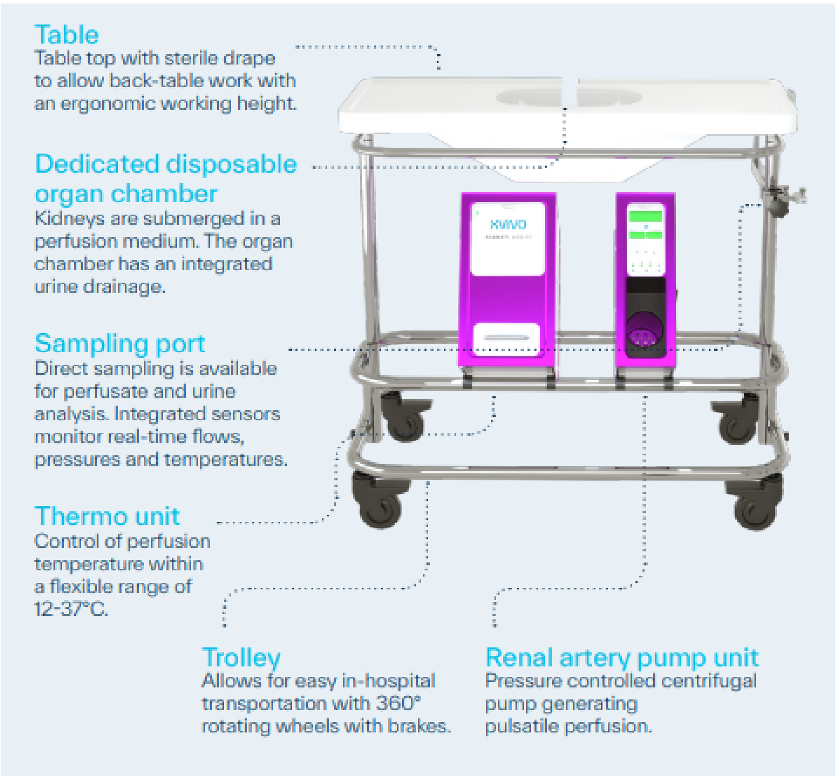


Figure 4: Kidney Assist Device with its components detailed [13]

As the only ex vivo normothermic renal perfusion device available on the market, its demand, and therefore its cost, is very high. This high cost makes its use and implementation in most hospitals unfeasible.

1.4 Objectives

Taking into account all the information gathered in the previous sections, the aim of this Master Thesis is to design and develop a first functional prototype of a normothermic perfusion system for an isolated kidney. The goal is to reduce the cost of such a device while preserving functionality. Subsequently, the device will be validated together with the head of the Urology Department of the Hospital Universitario Ramón y Cajal, Dr. Victoria Gómez.

To achieve this goal, the closed circulation system will consist of a centrifugal pump, a flow meter and a temperature sensor. The device sends the data collected by the sensors and receives commands from a control software designed in Python that communicates with the hardware devices through a microcontroller. Through this communication, the system monitors the temperature and flow rate of the perfused blood continuously, this data can be visualised in real time in the software developed for the control and visualisation. By means of this control software, the system will self-regulate the speed of the engine that pumps the blood, adapting the blood flow to the range of values decreed as the correct one. In addition, the operator of the device will be able to manually modify the speed of the motor as well as stop or restart the graphical display of the data from the sensors. In this way, the performance of the system can be objectively quantified and evaluated.

After defining the ultimate objective of the project, the intermediate objectives to be achieved for its correct implementation are specified below:

1. State of the art and similar solutions available on the market.

2. Design of the closed circulation circuit.

3. Study and selection of the necessary components for the hardware design.

4. Assembly of the device hardware and calibration of the sensors.

5. Programming of the microcontroller.
6. Data acquisition from the flow meter and the temperature sensor.
7. Adjustment of the centrifugal pump speeds.
8. Development of the control software and the graphical user interface.
9. Establishment communication between the control software and the microcontroller.
10. Testing and adjustment of the system.
11. Technical and clinical validation.

1.5 Document structure

After defining the objectives of this Master Thesis in the previous section (section 1.4), the structure of this document is set out below, highlighting the most important aspects of each section. The document consists of the following five chapters and three annexes:

- **Chapter 1**: this first chapter details the motivation and clinical context justifying the need for a normothermic perfusion machine for ex-vivo kidney preservation. In addition, a review of the art is made and the objectives of the project are presented.
- **Chapter 2**: this chapter is divided into two large blocks, the hardware and the software of the system. Each section details the processes and methodologies carried out to achieve the final objectives of the project.
- **Chapter 3**: after explaining the activities carried out, the results obtained are interpreted in this section.
- **Chapter 4**: the results obtained are discussed and a final conclusion is drawn. This chapter also details possible future lines for the improvement of the system.
- **Chapter 5**: this chapter gathers all the bibliography used in the document, these references are in IEEE format.
- **Appendix A**: in this annex the different impacts that the project may have on the social, economic, ethical and environmental fields are studied and analysed.

- **Appendix B**: details the financial budget necessary for the development of the project, which is funded by the Instituto Ramón y Cajal de Investigación Sanitario (IRYCIS)[14]
- **Appendix C**: this annex details the features of the hardware components used in the device.

2 Development

The closed-loop perfusion system comprises a whole set of devices that work together for organ homeostasis. A stable and relatively constant internal environment is maintained based on the baseline parameters established in the project *Modelo Traslacional de Perfusión Normotérmica Ex vivo para la evaluación de la Viabilidad y la Rehabilitación de Injertos Renales Pretrasplante*, this information has been provided by the project leader Dr. Victoria Gomez. In the Table 1 it can be seen the ranges of values in which the perfusion device will work.

Parameter	**Value**
Perfusion volume	500 ml
Flow	0,5 litres/min
Temperature	37ºC
Arterial cannula	3 mm
Venous cannula	5 mm

Table 1: Values for the parameters of the normothermic perfusion device

The work will be divided into a hardware and a software part as depicted in Figure 5. In the following sections, each of these parts will be split up for easier understanding.

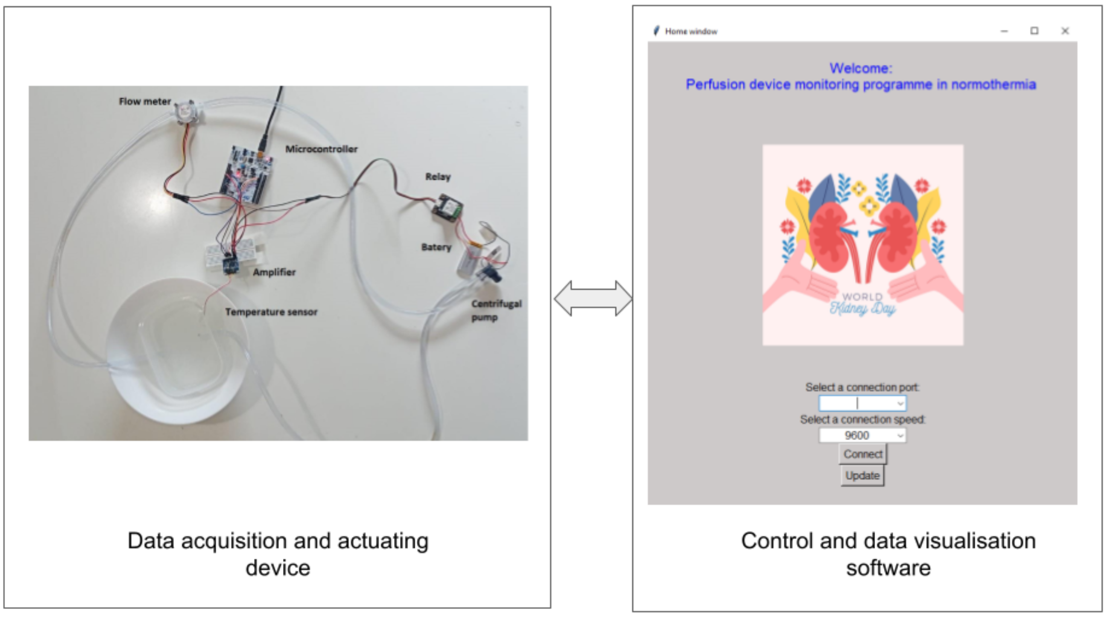


Figure 5: Diagram of the normothermic perfusion system

2.1 Hardware of the system

Figure 6 shows the interconnection of the hardware components and how they are assembled to make up the system. It mimics the blood circulation of a real body with the addition of those devices that continuously record the current state of the system and those that act on it. The parts that make up the system's hardware will be explained in depth in the Annex. C:

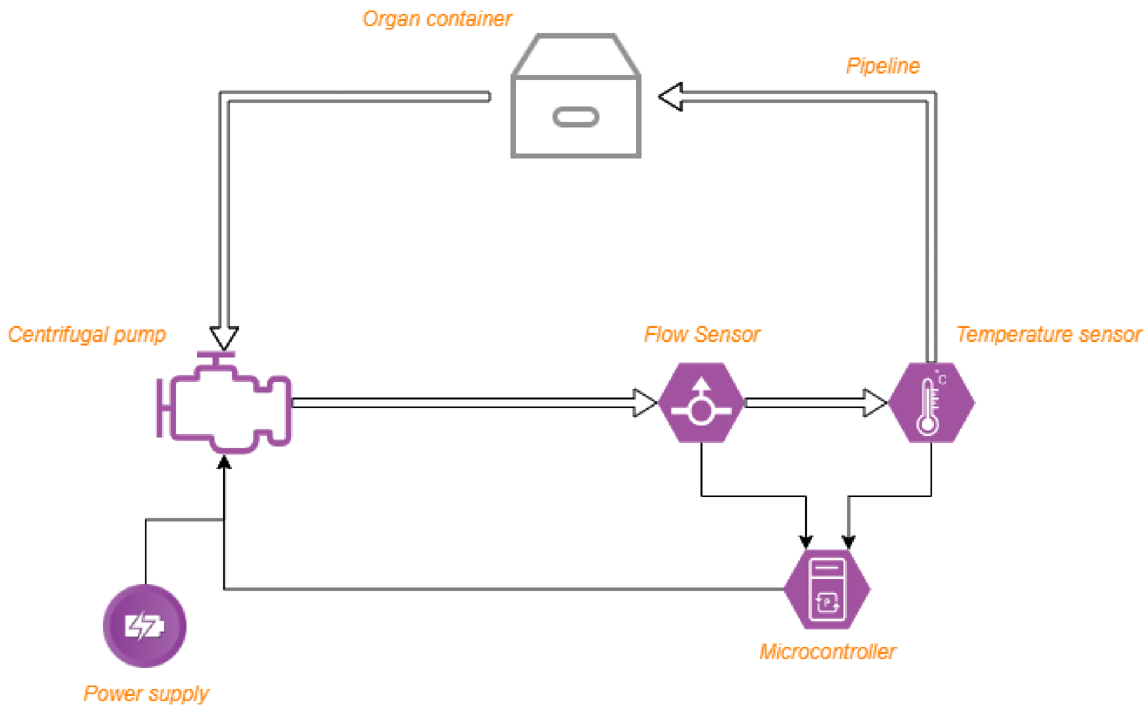


Figure 6: Schematic diagram of the normothermic perfusion device

Development board

The NUCLEO-F411RE development board is an STM32 Nucleo board from STMicroelectronics with STM32F411RET6 microcontroller that allows the user to develop their own prototypes for embedded applications [15], Annex C. This choice has been based on its high performance and power efficiency, as it contains 32-bit RISC processors with ARM Cortex-M architecture. In addition to its computational capability, its small size and cost make it an ideal choice for this project.

ARM is a 32-bit and 64-bit RISC (Reduced Instruction Set Computer) architecture found in electronics devices such as microprocessors that are smaller in size, low in power consumption and very low in cost. The RISC-based design allows ARM processors to require a small number of transistors, making the RISC architecture ideal for low-power applications.

This board interacts with each of the sensors (flow and temperature) and actuators (centrifugal pump), as well as carrying out the data transfer between these devices and the computer. The code developed to carry out all these actions is explained in detail in Section 2.2.1.

Flow sensor

The flow data is collected using the Hall effect measuring system G1&8 Water Flow Sensor from the manufacturer Seeed Studio [16], Annex C. A small and compact flow sensor was chosen, with a diameter to fit the size of the venous and arterial cannulas. In addition, this flowmeter has the ability to measure the flow corresponding to the perfusion system indicated in the Table 1, its flow measurement range is from 0.3 to 6L/min.

The sensor operates according to the Hall effect principle and consists of a plastic valve body, a rotor and a Hall effect sensor. The rotor axis is connected to the Hall-effect sensor, when the liquid flows through the rotor it rotates at a speed that varies according to the flow rate. The magnetic Hall-effect sensor emits an electrical pulse with each revolution, by counting the Hall-effect output pulses the flow rate of the liquid can be calculated, knowing that each pulse is approximately 2.25 millilitres [16]. This is an arrangement of a current flow coil and a magnet connected to the rotor axis, whereby a voltage/pulse is induced as this rotor spins, see Figure 7.

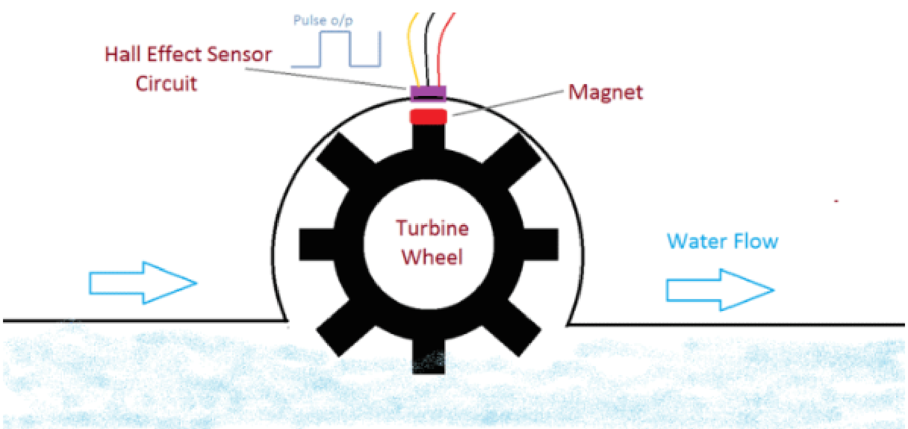


Figure 7: Hall effect schematic [17]

Temperature sensor

To acquire blood temperature data, the PT1000 thermoresistance detector (RTD) NB-PTCO-168 ([18], Annex C) belonging to the PTF family, from TE Connectivity, is placed at the entrance of the organ container . The PTF sensor family combines a group of resistance temperature detectors (RTD) that use a platinum resistor in thin-film technology as a sensing element. It consists of a platinum film structured on a ceramic substrate, passivated by a glass sheath. The basic principle of an RTD is that its wire sensor, made of a metal with a known electrical resistance, changes its resistance value as the temperature rises or falls. The resistance is a platinum strip with a resistance of 1000 ohms at 0°C, hence the name PT1000.

Platinum RTDs are the most common type of RTD as it has excellent corrosion resistance, excellent long-term stability and measures a wide temperature range (-30º to +300ºC). Due to its small size and low mass it has a low time constant, making it a suitable solution for fast (0.35 seconds response time in water) and accurate (±0.15°C) feedback control systems [18]. These values are in accordance with the parameters specified in the Table 1.

Acquisition card

To obtain the values from the PT1000 temperature sensor, an amplifier that is designed to read low resistance levels must be used. In addition, the resistance of the connection cables must be adjusted and compensated for. For this purpose, the Adafruit RTD sensor amplifier is used with the MAX31865 sensor card[19], Annex C.

The Adafruit RTD sensor amplifier integrates SPI (Serial Peripheral Interface) bus communication, this standard allows duplex communication between the microcontroller and the temperature data acquisition board. It includes a clock line, incoming data, outgoing data and a chip select pin, which connects or disconnects the operation of the device with the STM32 NUCLEO-F411RE board. To connect the RTD sensor to the amplifier we use the amplifier terminal blocks. The RTD sensor has two connection wires but the amplifier has one terminal block with four contacts, so in order to connect the sensor to the amplifier, the two terminal holes on the right side are jumpered together and the same for the left side, as shown in Figure 8.

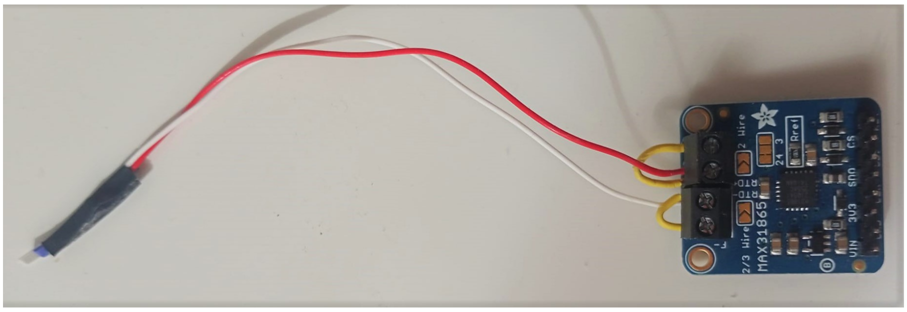


Figure 8: Short-circuit to connect the temperature sensor to the data acquisition card

Centrifugal pump

To avoid haemolysis (rupture of red blood cells) and to imitate the pulsating flow of the heart, the RS Pro centrifugal water pump [20] is used. This pump will be activated and regulated taking into account the values obtained by the flow meter, the microcontroller ensures that the pumping speed maintains the flow rate at the correct value specified in the Table 1.

Its high quality aluminium construction makes it highly resistant to corrosion. In addition, it has a very small size and the diameter of the connection to the pipe is 3.2 mm, which is perfectly adapted to the size of the arterial and venous cannulae. The maximum flow rate it can provide is 650 ml/min, so it can provide the flow rate established as correct in the Table 1.

External power supply

The centrifugal pump in the previous section must be supplied with voltage via an external power supply. The RS Pro motor operates on a supply voltage of 3 to 4V, so the NIMO Electronic BAT 528 rechargeable lithium battery is used [21]. It is a small and inexpensive battery that provides a voltage of 3.7V with a current of 500mAh, so it can be integrated into the system without taking up a lot of space or increasing its weight considerably.

Relay

The relay is an electromagnetic device that functions as a switch controlled by an electrical circuit in which, by means of a coil and an electromagnet, it opens or closes the passage of electric current. It can also function as an electrical amplifier as it has the ability to control an output circuit with higher power than the input circuit. The great advantage of electromagnetic relays is the complete electrical separation between the actuating current, which flows through the coil of the electromagnet, and the circuits controlled by the contacts. This makes it possible to handle high voltages/potential differences or high power with small control voltages. They also offer the possibility of controlling a device remotely using small control signals in a safe and cost-effective way.

To protect and control the system's centrifugal pump, the Gravity Digital Relay Module from the manufacturer DFRobot [22] is integrated. It is specially designed for low-voltage devices and can handle up to 10A of current. The relay module has an integrated LED indicator that shows its status. It has as input pins ground (-), power (+) and control signal (*D*); as output pins we only use the normally open (*NO*) pin and the common (*COM*) port. These pins can be seen in Figure 9.

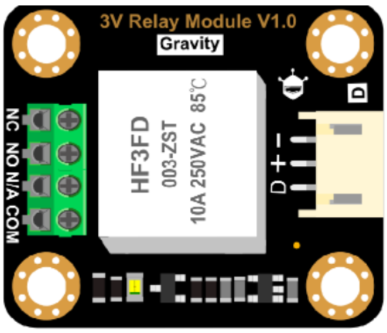


Figure 9: Relay module pins [22]

Pipelines and container

High quality flexible silicone pipes are used, as they are highly resistant to wear and abrasion and are non-toxic. They are 3 metres long and have an internal diameter of 3.2 mm, so they adapt perfectly to the diameter of the inlet and outlet of the peristaltic pump and therefore, to the diameter of the cannulas, as can be seen in the Table 1. In addition, a perforated plastic container with appropriate holes is used to provide the circulation of the fluid in which the organ is immersed.

Wires and conector

Multi-coloured cables with 10 cm long male-to-male, female-to-female or male-to-female connectors are used. The different connector types make them ideal for use with solderless BreadBoards or for connecting to square post headers or 0.1" pitch sockets on the development board or the different hardware devices. They use tin-plated beryllium copper in black plastic housings. [23]

A high quality USB A to mini USB B male cable is required to connect the microcontroller NUCLEO-F411RE to the control computer. The cable manufactured by RS Pro [24] has been chosen as it conforms to industry standards and has a length of 3m.

2.2 Software of the system

The system software has been developed in two different platforms:

- **STM32CubeIDE**: is ST's free, cross-platform, integrated development environment to work with STM32 microcontrollers [25]. The most important features of this tool are described below:

  – STM32CubeIDE integrates STM32CubeMX, a graphical tool that allows a very simple configuration of STM32 microcontrollers and microprocessors through a step-by-step process that allows a more efficient workflow. The first step is to select an STMicrolectronics STM32 microcontroller. The second step consists of configuring the pins, clock tree, peripherals (such as GPIO or USART) and middleware stacks (such as USB or TCP/IP). Finally, the user triggers the generation of initialisation C-code that matches the selected configuration options. At any time during development, the user can go back to the initialisation and configuration of the peripherals or middleware and regenerate the initialisation code without impacting the user's code.

  – It is fully compatible with Eclipse, being able to use familiar tools reduces the learning curve.

- It includes build and stack analysers that provide the user with useful information about project status and memory requirements. It also incorporates standard and advanced debugging features including views of CPU core registers, memories and peripheral registers, as well as live variable observation, the Serial Wire Viewer interface or the fault analyser.

- **Python**: this language has been chosen for programming the control software and the graphical interface for the following reasons:
  - It is developed under an open source licence, so it can be used and distributed freely, even for commercial use. The Python licence is administered by the Python Software Foundation. [26]
  - It is a general-purpose language, designed to create all kinds of programs.
  - It is cross-platform, originally developed for Unix, but any system is compatible with the language as long as there is an interpreter programmed for it.
  - It is an object-oriented language, offering a simple way to create programs with reusable components. It has many functions for the treatment of strings, numbers, files, etc. As well as libraries that can be imported for the treatment of specific topics such as window programming.
  - It is a high-level interpreted programming language whose philosophy emphasises the readability of its code.
  - The fact that Python is free and open source contributes to create a strong community. It is one of the most popular programming languages, so there is a large amount of information available.

(a) STM32CubeIDE [25] (b) Python [26] (c) Visual Studio Code [27]

Figure 10: Tools used for system software development

2.2.1 STM32CubeIDE

The programming of the NUCLEO-F411RE [15] development board has been carried out using ST's integrated development environment, STM32CubeIDE [25], using the C language. Below, each of the configurations made on the board for the correct operation of the system will be explained in more detail.

All connections between the electronic components that make up the hardware part of the system can be seen in Figure 11.

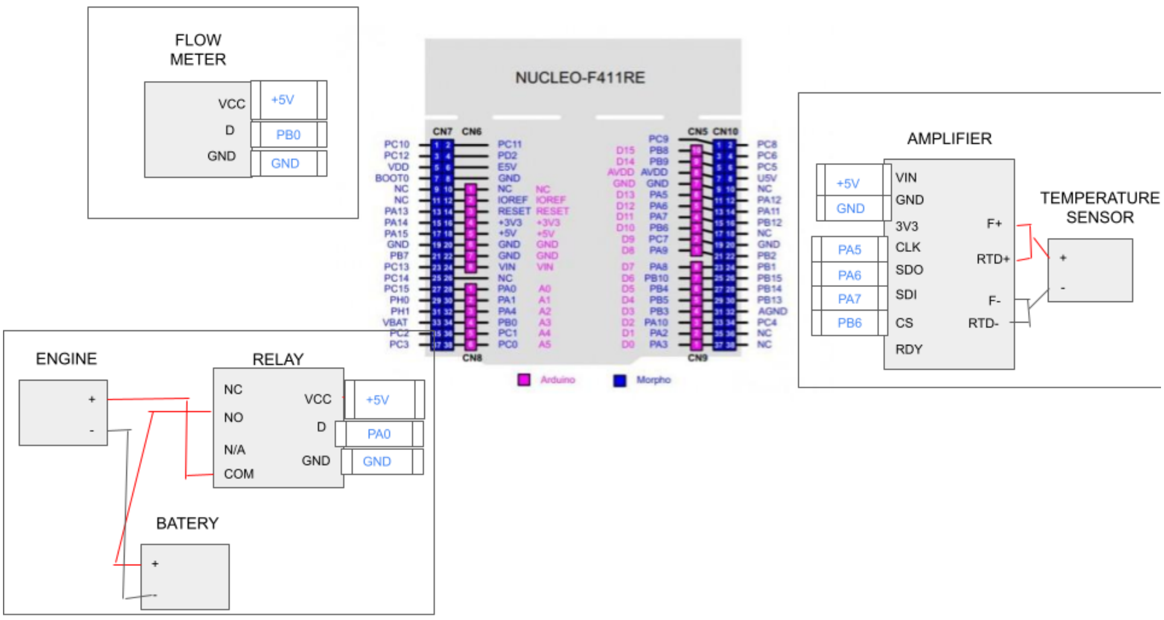


Figure 11: System hardware electrical connections

The board has been programmed using the STM32 Hardware Abstraction Layer (HAL). This abstraction layer contains drivers for all available peripherals with which pins, peripherals, interrupts and nearly all microcontroller operation can be configured. It has high-level programming, greatly simplifying code portability between STM32 subfamilies, hides the complexity of the electronics and speeds up programming times.

Each STM32 microcontroller has a variable number of general programmable I/O, the configuration of the microcontroller's peripherals is done through the GPIO (General Purpose Input Output) registers. GPIO is a series of registers associated with the general purpose inputs and outputs of the board. As seen above, the HAL is designed in a way that abstracts from peripheral-specific memory, providing a general and easier way to configure it, using the HAL_GPIO module.

Figure 12 shows the configuration of all the pins of the NUCLEO-F411RE microcontroller. Those in green are activated, those in yellow with the warning signal can be activated and partially configured, with some limited functionality. In the case of those in red, it is not possible to activate them, since there is a conflict with another already activated pin that cannot be resolved. It is possible to see which function each pin can acquire by simply clicking on the pin in question.

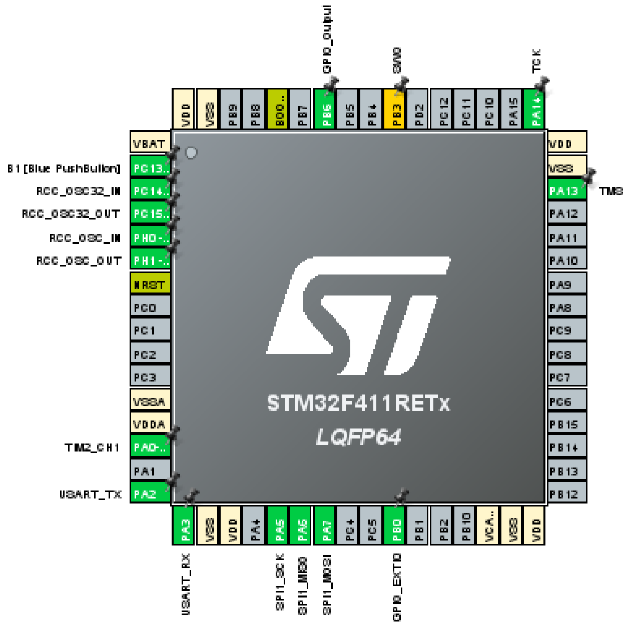


Figure 12: Microcontroller pin configuration

Serial communication

For communication between the microcontroller and the computer control software, the USART module contained in the microcontroller has been used. USART (Universal Synchronous/Asynchronous Receiver Transmitter) is a protocol used in dual communications, i.e. it has the capacity to receive and transmit simultaneously. Data is transmitted serially, only one frame is transferred over the channel at a time. The management of serial communication has many benefits, among which stand out, the control of systems through the computer, performing complex calculations, visualising and graphing data, the simplicity for its implementation and low cost.

A frame is the unit for sending data, each frame has the following elements, which are shown in Figure 13:

- Stop bit always high.
- Data bits (5, 6, 7, 8 or 9 depending on configuration).
- A parity bit: parity codes are used to detect and, in some cases, correct errors in transmission. This parity bit is determined so that the total number of bits 1 to be transmitted is either even (even parity code) or odd (odd parity code). The parity bit shall be a 0 if the total number of ones to be transmitted is even (bit 1 for an odd number of ones). The parity bit shall be a 0 if the total number of ones is odd (bit 1 for an even number of ones).
- One or two stop bits high.

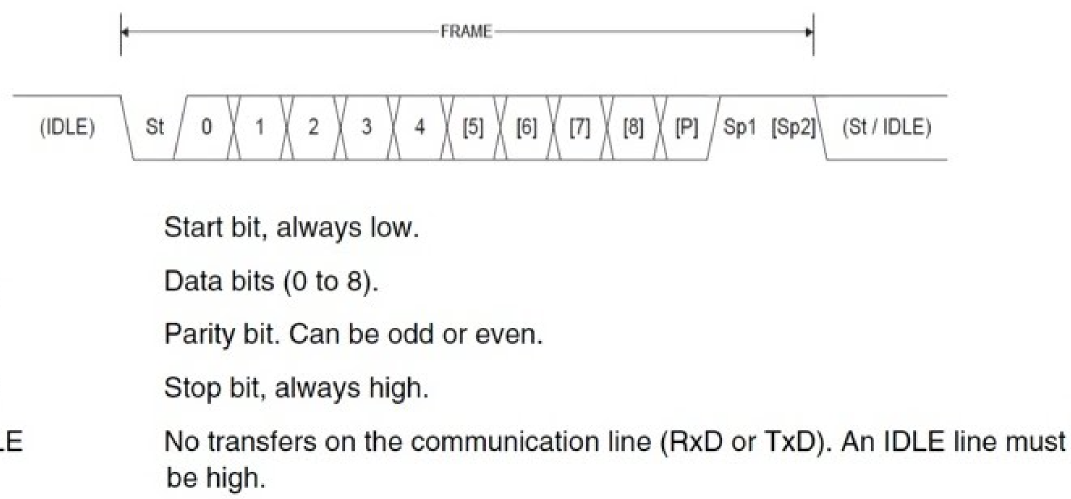


Figure 13: USART frame [28]

The board is connected to the computer by means of a high quality USB A to mini USB B male cable [24]. For the configuration of the USART module in the microcontroller, the USART2 module has been used. This module is connected internally to the board by default and therefore, the same COM port (serial connection I/O interface) that the board generates in the computer when connected to it can be used. The USART2 module was configured by following the steps below:

1. Two pins are set, one for Transmit Data Out (TX) and one for Receive Data In (RX), as shown in Figure 12. These will be pins *PA2* and *PA3* respectively.

2. An asynchronous connection mode is selected, so the assignment of the *CK* pin is not necessary.

3. We disable the Hardware Flow Control (RS232), so we do not need the *CTS* and *RTS* pins either.

4. The global NVIC (Nested Vectored Interrupt Control) interrupt is enabled. It is used to perform the sending and receiving of data via a priority 5 interrupt, so that the flow of code does not block each time one of these actions is performed.

5. Configured as a Full Duplex Operation (Independent Serial Receive and Transmit Registers).

6. A single stop bit.

7. No paired bit.

8. To establish an asynchronous communication between two agents it is necessary to agree on the byte transmission rate (baud), the transmission speed in transmitter and receiver must be the same. A transmission and reception rate of 9600 baud (number of bits per second) is established.

9. The data frame length is set to 8 bits of information.

10. The USART receiver implements different oversampling techniques for data recovery, discriminating between valid incoming data and noise. Oversampling by 16 is selected to increase the receiver's tolerance to clock drift. The receiver samples a period of a bit 16 times. To understand whether the bit is 1 or 0, the receiver engine takes 16 samples.

Concurrent programming

The normothermic perfusion device needs to perform multiple tasks at the same time, such as capturing measurements from the different sensors and executing commands on the actuator. To manage multiple jobs running simultaneously, a real-time operating system (RTO) is used. The operating system of choice is FreeRTOS [29], a free and open source real-time operating system that runs on many popular microcontrollers, including the STM32. In 2017, Amazon took control of the FreeRTOS project and now offers regular maintenance and support.

STM32CubeIDE does not make calls to FreeRTOS directly. As explained in Section 2.1, the microcontroller used is based on the 32-bit RISC Arm Cortex core. ARM has created the CMSIS-RTOS library, which allows calls to be made to an underlying RTOS, in our case FreeRTOS. This improves the portability of the code between various ARM processors. FreeRTOS is our "RTOS kernel", as shown in Figure 14 below:

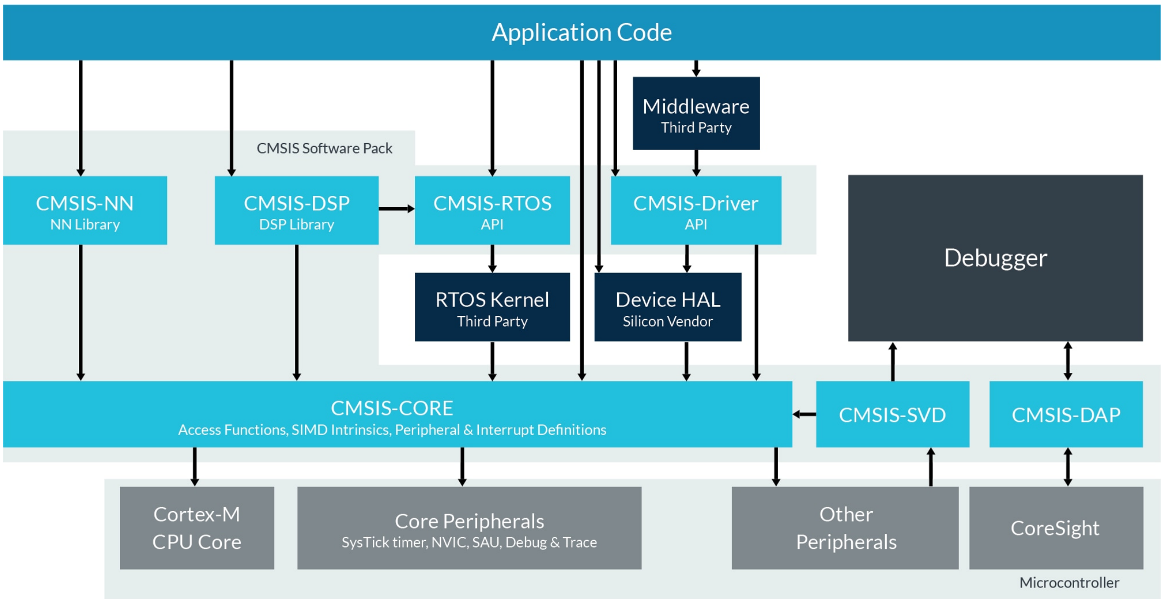


Figure 14: Diagram of the ARM CMSIS libraries [30]

The use of FreeRTOS allows task switching, scheduling of tasks, assigning priorities to each job, etc. Each of the tasks that are performed simultaneously are implemented as concurrent threads by following these steps:

1. From the category *Middleware* FREERTOS is selected. The interface mode CMSIS_V2 is chosen.

2. SysTick is a timer that is reserved for operating system purposes. As explained above, the Hardware Abstraction Layer (HAL) is used, SysTick is used by default by this layer with a very high priority. However, FreeRTOS needs SysTick for its scheduler and requires SysTick to have a much lower priority. To solve this, another

timer must be assigned as a time base source for HAL. Basic Timer 1 has been selected as the time base.

3. Three threads are created with normal priority, dynamic allocation and a stack size of 128 words. One thread is responsible for reading and sending data from the flow sensor, another for reading and sending data from the temperature sensor and finally, a thread that receives the commands to drive the centrifugal pump.

Flowmeter

As explained in Section 2.1 the sensor works on the Hall effect principle, where the magnetic Hall effect sensor emits an electrical pulse with each revolution. To count the Hall effect output pulses, the *PB0* PIN on the board is configured as an input PIN with an external *EXTI0* interrupt, as can be seen in Figure 11. EXTIs are interrupts that are executed when the state of a microcontroller pin is altered. As the GPIOs have 16 pins (0 to 15), there are external interrupts from *EXTI0* to *EXTI15*, only interrupts 0 to 4 have independent interrupt vectors.

The configuration of this external interrupt is done by the NVIC (Nested Vectored Interrupt Control) which is the module that handles the interrupts in a Cortex-M. The pin will trigger an interrupt when it detects a rising edge, i.e. when it goes from a low state to a high state due to the presence of an electrical pulse generated by the Hall effect sensor of the flow meter. The interruption produced is handled in the interruption file *stm32f4xx_it.c*, where the pulses produced will be counted in a global variable. Each time the interruption is triggered in the PIN, one unit will be added to the global counter.

The device operates with a clock frequency of 84 MHz (Figure 15). The clock signals of the processor core, peripherals and system buses can be configured and displayed under the *Clock Configuration* tab. Timer 3 is configured with the internal clock source, a prescalar of 840 and a counter period of 10000 are set. This results in a counter at a frequency of 10 Hz. This counter calculates the flow rate of the device every 100ms, taking into account the global variable *pulse* which contains the number of pulses emitted by the Hall sensor. The following formula is used to calculate the litres per minute that are flowing through the device

$$flow = \frac{pulse \cdot 2.25 \cdot 60}{1000} \tag{1}$$

The number of pulses is multiplied by 2.25 as each pulse is known to correspond to approximately 2.25 millilitres. This measurement is then rescaled to litres per minute. The decimal value obtained is converted into a string and sent every second via USART communication to the control computer.

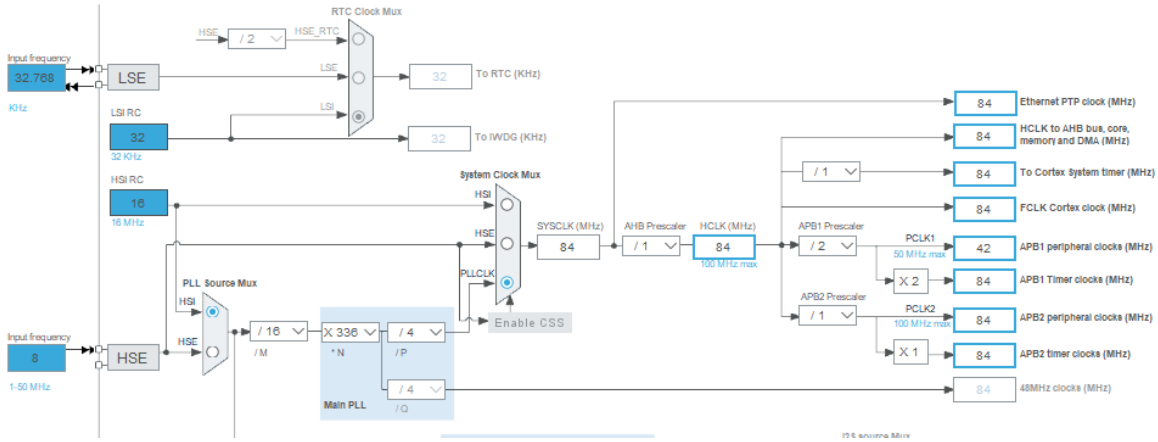


Figure 15: Configuration of the system clocks

The flowchart of the flowmeter function can be seen below, Figure 16.

Temperature sensor

To communicate between the microcontroller and the Adafruit MAX31865 RTD amplifier [19] , the SPI (Serial Peripheral Interface) bus is used. This is a communications standard for controlling digital electronic devices that accept a clock-regulated serial bit stream (synchronous communication). Its communication speed is faster than the USART standard used in the rest of the device and it is also full-duplex. This means that it can transmit and receive at the same time as it has separate receive and send data lines.

The SPI bus has four possible modes of operation, these modes refer to how data is sampled with clock pulses. Polarity (CPOL) and transmission phase (CPHA) are taken into account:

- With rising edge (CPOL = 0, clock line is low) without delay (CPHA = 0, bits are sampled on the rising edge of the clock).

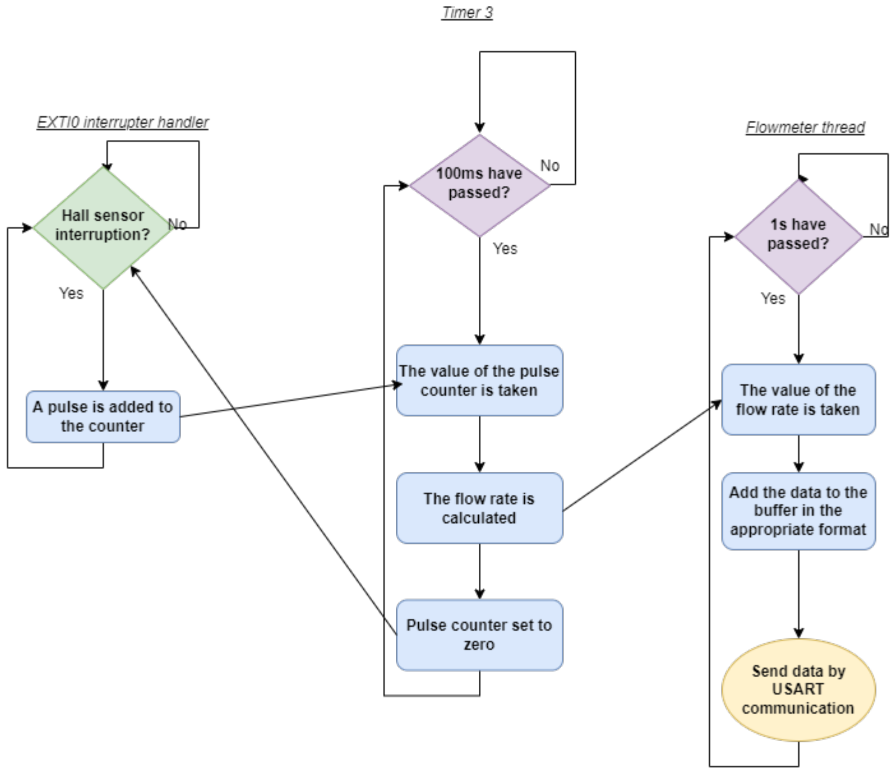


Figure 16: Flowmeter function flowchart

- With rising edge (CPOL = 0, clock line is low) with delay (CPHA = 1, bits are sampled on the falling edge of the clock).
- With falling edge (CPOL = 1, clock line is high) without delay (CPHA = 0, bits are sampled on the rising edge of the clock).
- With falling edge (CPOL = 1, clock line is high) with delay (CPHA = 1, bits are sampled on the falling edge of the clock).

As can be seen in Figure 11, the clock line is connected to PIN *PA5*, PIN *PA7* corresponds to the MOSI (Master Output Slave Input) output data line, the MISO (Master Input Slave Output) input data is connected to PIN *PA6* and the chip select pin is configured as the GPIO output PIN *PB6*. This last pin, which connects or disconnects the Slave to the Master, is idle at high level, i.e. the signal is lowered when talking to the device is required.

In *Connectivity* SPI1 is selected and Full-Duplex Master mode is set. Also, Hardware NSSP Signal mode is disabled, as it is not desired that the slave trigger pulse is handled by a hardware peripheral. The data size is set to 8 bits, where the most significant bit is

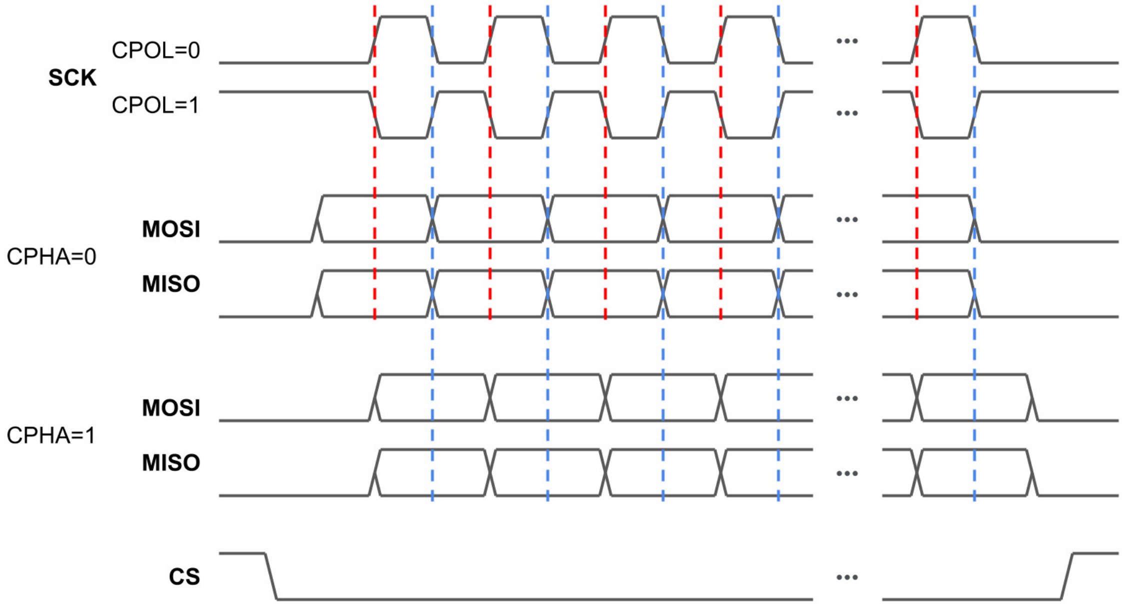


Figure 17: SPI operating modes [31]

the first one (MSB). A Prescaler value of 128 is selected, as it is desired that the Baud Rate stays below 2 Mbps. The device uses a low clock line (CPOL = 0) and samples bits on both the rising and falling edges of the clock (CPHA 2 edges).

To start reading the sensor data the MAX31865 STM32 HAL library is acquired from the GitHub user Nima Askari [32]. This library is based on the library provided by Adafruit for the acquisition board, Adafruit's MAX31865 library. The files are downloaded and implemented in the STM32CubeIDE project, managing all permissions and respecting the hierarchies and structures of the code.

After including the headers of these files in the code, the reference values for the temperature sensor [18], nominal resistance and reference resistance of the RTD are specified in the configuration file *Max31865Conf.h*.

- *rtd_nominal*: is the resistance value in ohms of the RTD at a nominal value (typically 0 degrees Celsius). This value is 1000 Ohms for a PT1000 sensor.

- *ref_resistor*: value of the reference resistor in Ohms. The default value is 4300 Ohms for a PT1000 breakout.

The average slope between 0°C and +100°C is called alpha (α). This value depends on the impurities and concentrations in the platinum of the RTD sensor. The resistance versus temperature curve is reasonably linear but has a certain curvature, as can be seen in Figure 18 for a PT100. This linear relationship is described with the Callendar-Van Dusen equation [19]::

$$R(t) = R(0) \cdot [1 + a \cdot T + b \cdot T^2 + c \cdot (T - 100C) \cdot T^3] \tag{2}$$

Where the European standard IEC 60751 (DIN EN 60751) specifies an alpha value of $\alpha = 0.00385055$ and the following Callendar-Van Dusen coefficient values:

$$a = 3.90830 \cdot 10^{-3} \tag{3}$$

$$b = -5.77500 \cdot 10^{-7} \tag{4}$$

$$c = -4.18301 \cdot 10^{-12} \tag{5}$$

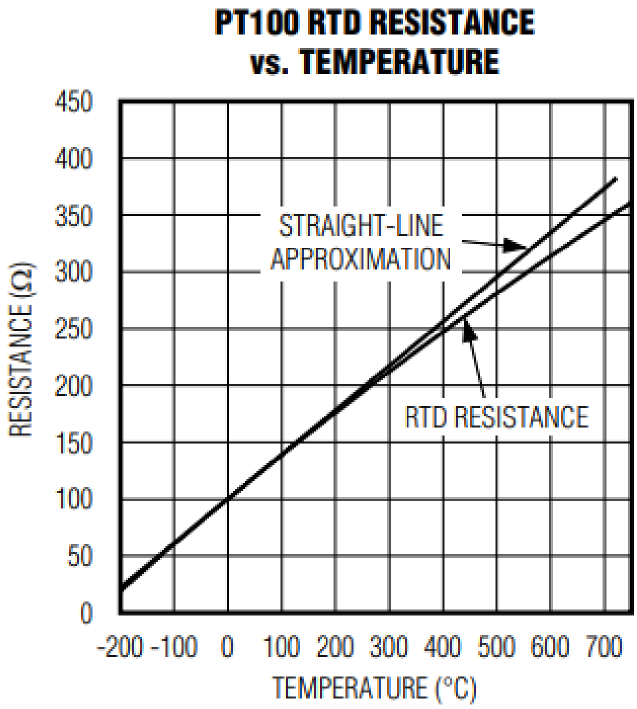


Figure 18: Resistance vs. temperature curve [19]

Firstly, the object for SPI bus communication between the amplifier and the RTD sensor is initialised. Indicating the pin of the chip select (*PB6*), the number of wires needed for the connection to the RTD sensor (2) and the filtering frequency in Hz (50Hz).

Once this communication object has been created, the *Max31865_readTempC* function of the new implemented library is called, which will return a temperature value that will be filtered by the *Max31865_Filter* function, also belonging to the same library. The temperature in degrees Celsius is obtained by applying the formula and the values seen in the Equation 2. The final decimal result of the temperature is converted to a string and sent by interruptions, every second, through the USART communication standard to the control computer. The flowchartfor temperature data acquisition can be seen below, Figure 19.

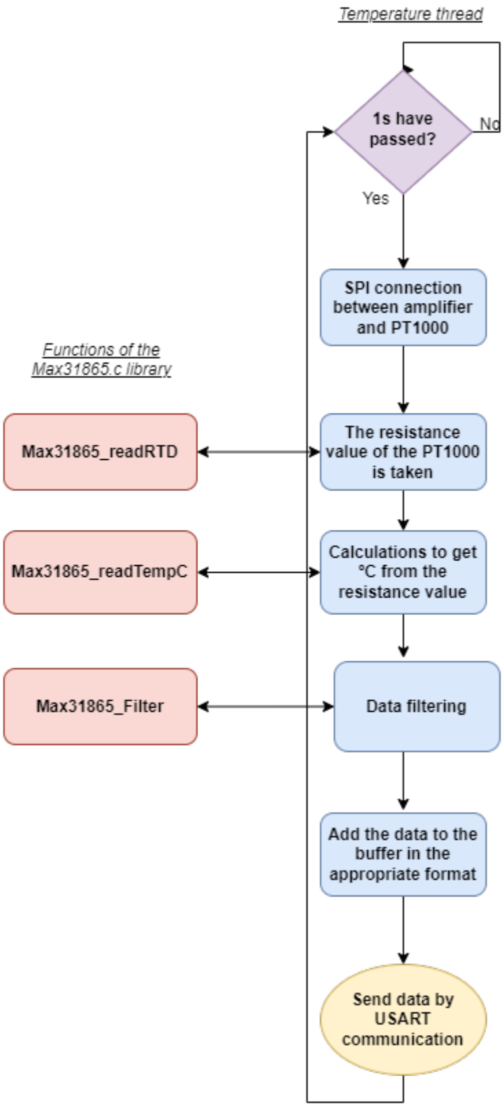


Figure 19: Temperature data acquisition flowchart

Centrifugal pump operation

The thread that manages the speeds of the centrifugal pump receives data from the control software via the USART communication. The receive buffer has a size of 3 bits, i.e. the maximum value it can receive is 999.

Among the most common methods for regulating the speed of a DC motor, the most widespread due to its reliability and simplicity is Pulse Width Modulation (PWM). This technique consists of working with a square digital signal, in which the duty cycle can be varied without varying the frequency. The duty cycle describes the amount of time the signal is in a logic high state, as a percentage of the total time it takes to complete a full cycle, as shown in Figure 20. The frequency determines how fast a cycle is completed, and therefore how fast it switches between the logic high and low states. As a formula, a duty cycle is expressed as:

$$D = \frac{Ton}{Period} \cdot 100\% \tag{6}$$

where D is the duty cycle and TON is the time that the signal is active. The pulse width is the duration of the TON, given the period. [33]

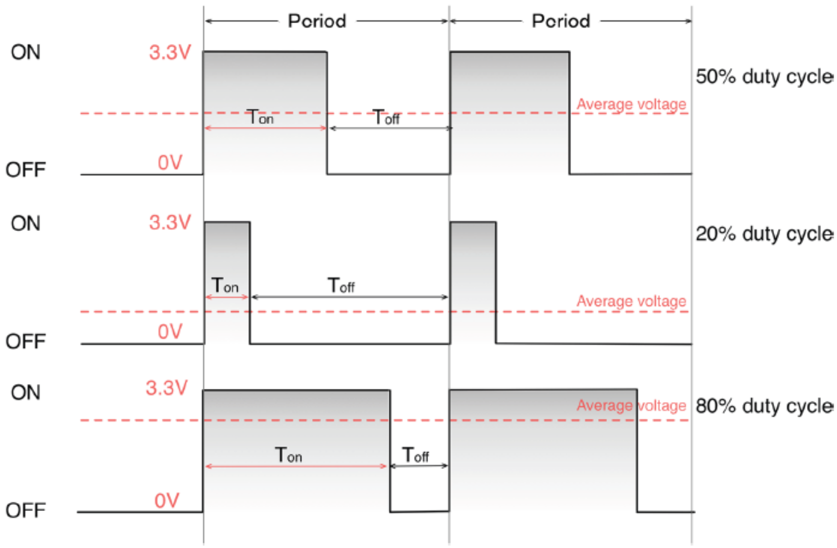


Figure 20: Duty cycle of 50%, 20% and 80%. [33]

To adapt the motor speed to the value received from the control software, the output voltage is regulated by means of pulse width modulation (PWM). To do this, the Timer

2 is used by configuring its channel 1 as a PWM generator. This configuration of the Timer 2 is assigned to the *PA0-WKUP* PIN on the board. The device operates with a clock frequency of 84 MHz (Figure 15), the timer uses the device's internal clock as a clock source. A Prescaler of 4204 and a counter period of 999 are set to lower the motor frequency to 20Hz, according to the following formula:[33]

$$frecuency\_timer2 = \frac{(\frac{f}{prescaler+1})}{period+1} = \frac{(\frac{84MHz}{4204+1})}{999+1} = 20Hz \quad (7)$$

The choice of the period, which determines the frequency of the output signal together with the internal clock of the timer, is adjusted to the operating frequencies of the motor. The frequency is lowered sufficiently so that the rotational speeds of the motor are perceptible, the maximum rotational frequency of the motor is 20 Hz.

The duty cycle goes from 0 (since we set the parameter *Pulse* to 0) to the value of the timer's *Period* field, in this case 999, the maximum value that the input buffer can receive. The longer the period, the wider the adjustment range, i.e. the finer the output voltage can be adjusted. There are two PWM modes available: PWM Mode 1 and 2. PWM Mode 1 is chosen because it is an upcounting mode, i.e. the channel is active as long as the value of the *Period* is less than the value of the *Pulse*, otherwise it is inactive. In downcounting, the channel is inactive as long as the period is greater than the pulse, otherwise it is active.

When an interruption occurs due to the arrival of data from the control software, this received data is added to the buffer, transformed into an integer value and set as the new value of *Pulse*. In this way, the duty cycle is varied and, consequently, the output voltage value that will reach the motor, affecting its speed.

This modified output voltage value is transmitted to the motor after passing through the relay. The relay functions as a switch, protecting and controlling the centrifugal pump. The control signal (D) of the relay, which opens or closes the connection to the motor, is connected to the *PA0-WKUP* PIN of the microcontroller where the previously explained pulse generator was configured, these connections can be seen in Figure 11. The flowchart of the centrifugal pump operation can be seen below, Figure 21.

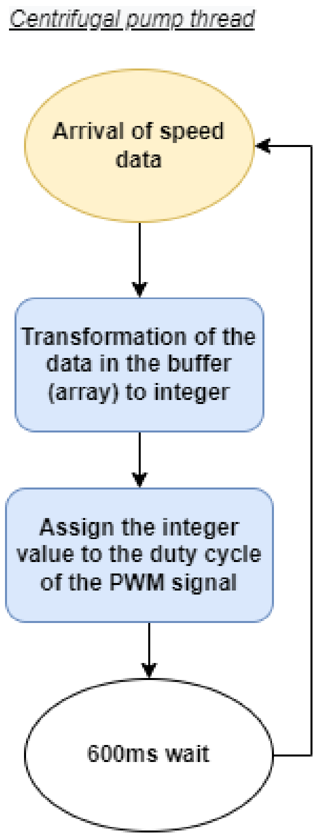


Figure 21: Centrifugal pump operation flowchart

2.2.2 Python

The control code and user interface of the system has been developed using the Python language [26] in the Visual Studio Code environment [27]. This is a source code editor developed by Microsoft for Windows, Linux, macOS and Web. It is free and open source and includes support for debugging, syntax highlighting and intelligent code completion.

The code programmed in Python is broken into two large blocks to make it easier to understand, the graphical user interface part and the control code.

User interface

The user interface is the way in which the user can communicate with the device. The objective is to allow the user to operate and control the system in a friendly and efficient way. It includes elements such as menus, windows, graphic content and, in general, all those channels through which communication between the human being and the device is allowed. The following libraries are used to program this user interface:

- Tkinter: is the most widely used package for creating graphical interfaces in Python. It is an object-oriented layer based on Tcl (a simple and versatile open-source programming language) and Tk (the standard graphical user interface tool for Tcl). This library is used to create the windows and widgets that, in a hierarchical way, will compose the interface.

- Python Imaging Library (PIL): allows image editing directly from Python. Adds support for opening, manipulating and saving different image file formats, such as GIF, JPEG, JPG and PNG.

- Matplotlib: is a cross-platform data visualisation and plotting library for Python and its numeric extension NumPy. It can make use of matplotlib APIs (Application Programming Interfaces) to embed static, animated or interactive graphics in GUI (graphical user interface) applications.

In the first place, a start-up window is displayed with a welcome message, a cover image, two drop-downs and two buttons. The first drop-down contains the available ports detected by the program, which can be updated by pressing the *Update* button, the other drop-down contains different serial communication speeds, by default, the system is programmed at 9600 baud. The user has to select the input port where the device is connected and press the button *Connect* (see Figure 22). Then, this window disappears and another window is displayed with two different tabs, one for data display and one for control (see Figure 24).

In the visualisation tab, Figure 23, two graphs are shown at the top of the window, two buttons in the centre and a frame at the bottom. Both graphs paint the real-time values acquired by the flow sensor (left graph) and the temperature sensor (right graph). The values start to be displayed when the user presses the *Start* button and stop painting if the *Stop* button is pressed. The X axis of both graphs represent the time, all the values collected in 100 seconds can be painted at the same time. On the Y axes we can see that the flow rate values are shown in a range from 0 to 3 LPM and the temperature values in a range from 25 to 35ºC. These ranges have been determined taking into account the values established as correct in the Table 1. To make it easier to read the data in real time, a frame has been added at the bottom of the window to show the current data

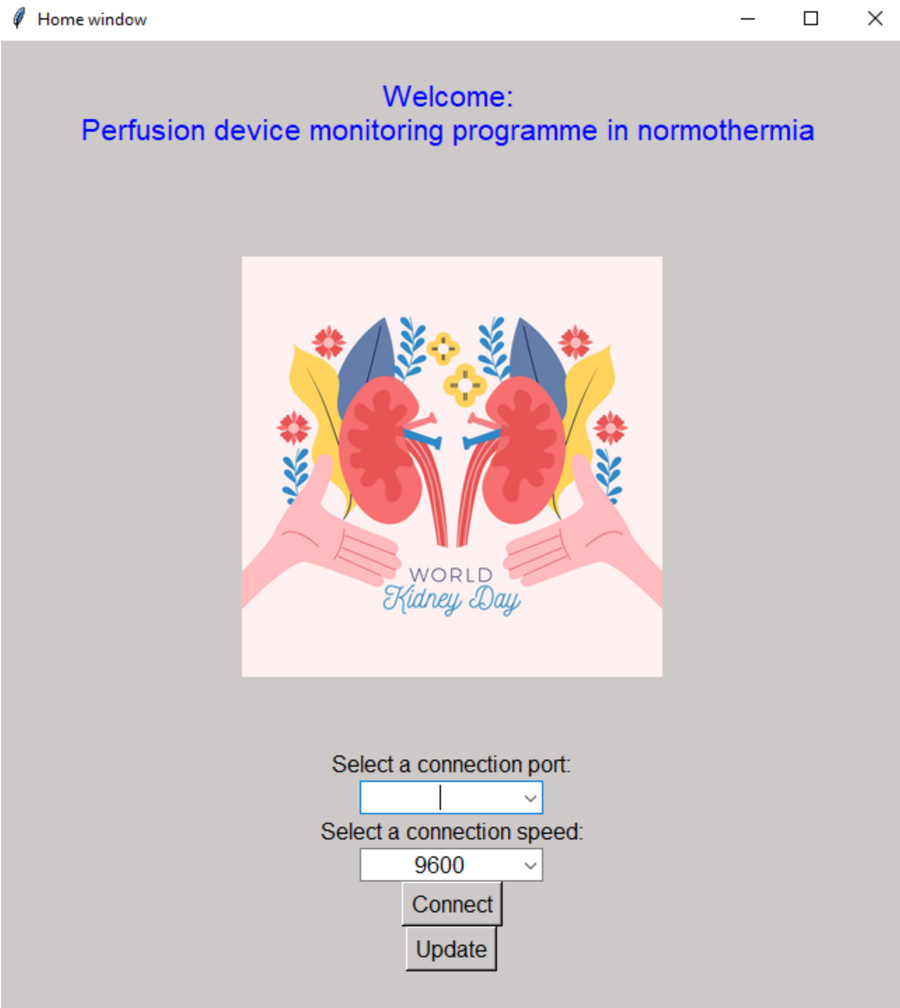


Figure 22: Programme start window

currently being plotted in the upper graphs.

The control tab, Figure 24, shows a text box at the top of the window, a slider on the left and a switch on the right. The text box informs the user that the centrifugal pump is self-regulated by the system taking into account the real-time values measured by the flow meter. Although the system is self-regulated, the operator is given the option to switch the motor on or off whenever it is necessary and even regulate the speed of the motor by means of the slider. The status of the motor (on or off) will be indicated by highlighting the text in green or grey and changing the state of the switch, as can be seen in Figure 25. By means of the slider control the user will be able to regulate the speed of the pump, this will be at maximum functioning when the value sent is 999 and it will be turned off if a value of 0 is sent using the button *Send*.

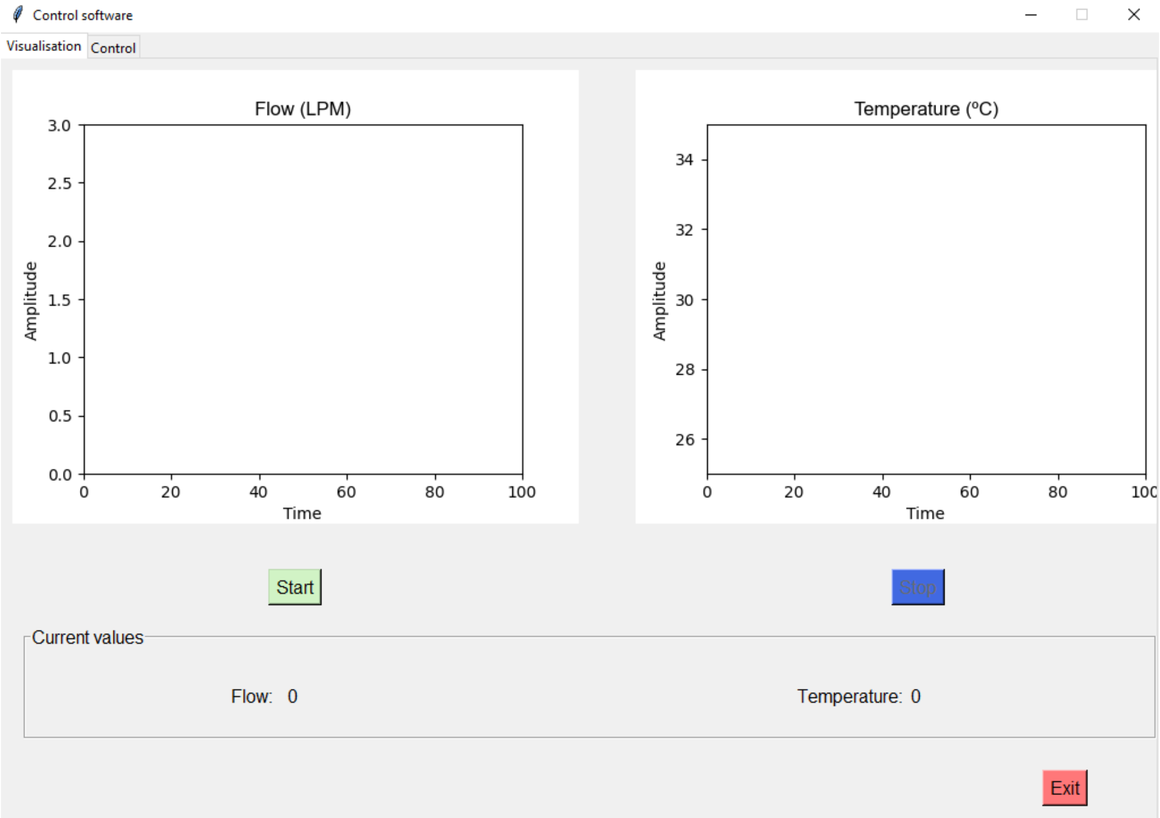


Figure 23: Data display window

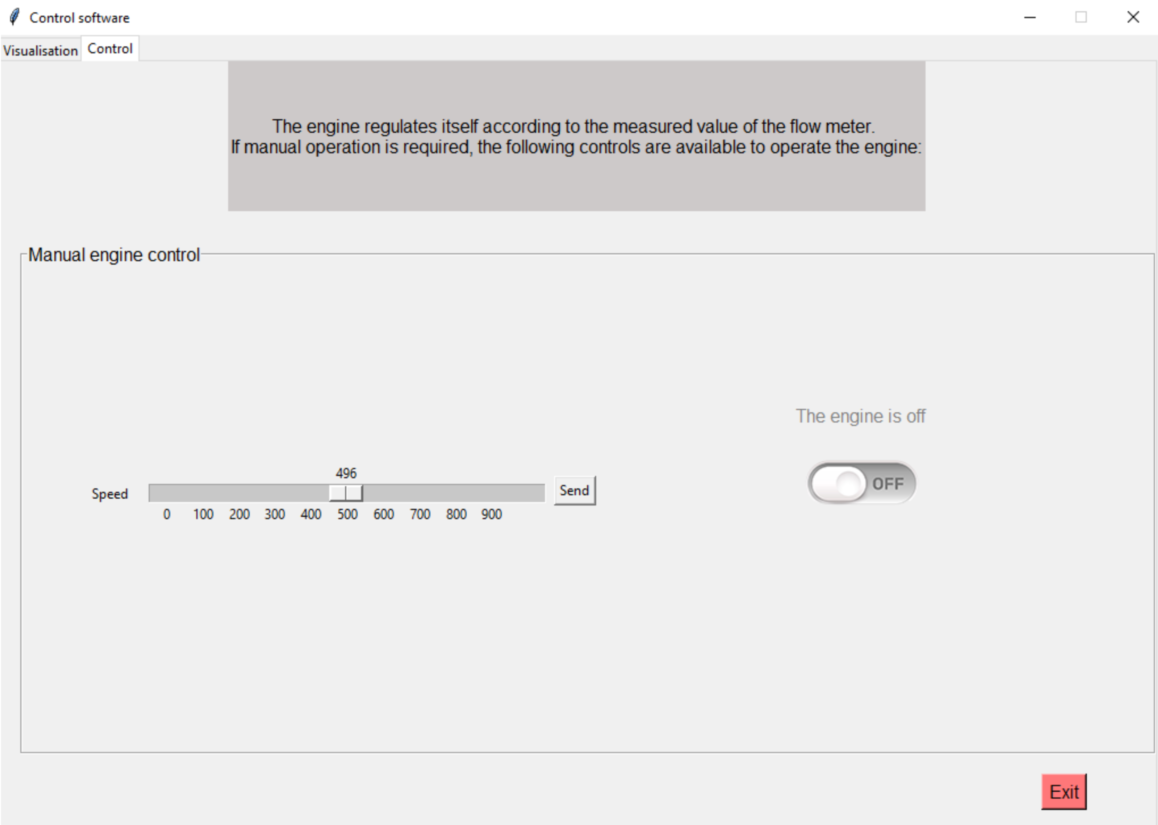
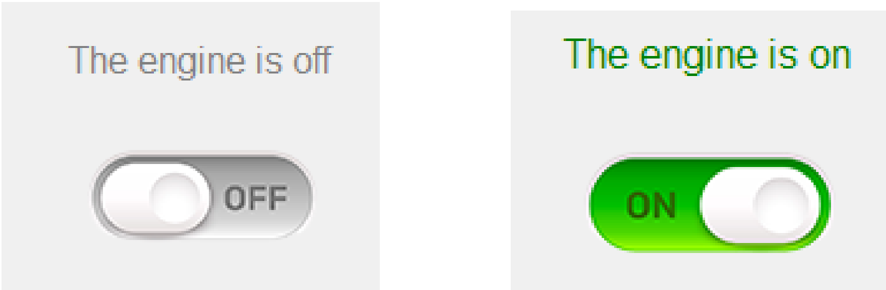


Figure 24: Centrifugal pump control window



(a) Off (b) On

Figure 25: Switch states

Both the visualisation and control tabs have an exit button in the bottom right corner. When the user presses this button or closes the window all processes will stop and the programme will close in an organised sequence.

Control code

The system control code was developed using the Python programming language and the libraries specified below. The aim is to communicate, receive and send data in real time between the hardware and the graphical user interface.

- Threading: using this module, threaded programming is possible, i.e. it allows a program to execute multiple operations simultaneously in the same process space. Each execution flow that originates during processing is called a thread or subprocess. The use of threads is very useful as it allows to check and control the operation of the different sensors and actuators of the system simultaneously.

- Serial: this module encapsulates the access to the serial port and configures the serial object necessary to carry out the serial communication with the microcontroller (both for reading and writing).

- Collections: this library helps to complete and manipulate data structures efficiently.

- NumPy: this library specialises in numerical calculation and data analysis, especially for large volumes of data. It incorporates a class of objects called arrays, that allows to represent collections of data of the same type in several dimensions. It also incorporates very efficient functions for their manipulation. Array processing is much faster (up to 50 times faster) than Python lists, making it ideal for processing large vectors.

When the system starts running, the available ports will be searched using the Serial library and they will be shown in the drop-down list of the start-up window. When the user selects a communication port and a communication speed rate, he/she will press the button *Connect*, previously explained in the above section, and the serial object will be created. This serial object will be used to receive and send data to the microcontroller, the configuration parameters of this serial object are the port and the transmission speed selected by the user in the start-up window.

Once the connection has been made, the thread is created and initialised. As long as this thread is open, the sensor data sent by the microcontroller will be constantly read. This data reading will only be interrupted when the operator or the programme itself sends a data through the serial port to control the centrifugal pump. The use of threads is very useful as it allows to check and control the operation of the different sensors and actuators of the system simultaneously. Threads running in a process share the same data space as the main thread and can therefore access the same information or communicate with each other more easily than if they were in separate processes. Running a multi-threaded process typically requires less memory resources than running the equivalent in separate processes and simplifies the design of applications that need to execute multiple operations concurrently.

The data from the temperature and flow sensors being received by the thread are automatically stored in their respective containers. The Deque container provided by the Collections library is used, this is an optimal version of the list normally used in Python for inserting and deleting elements. When the user presses the start data visualisation button the data contained in the temperature and flow rate containers will be painted in their respective graphs, and can be paused or resumed at any time. The value provided by the flowmeter will be checked continuously and, if it is outside the correct range (Table 1), a command will be sent via the serial port to the engine to provide more or less flow. This algorithm will run continuously until the user closes the window or presses the *Exit* button, killing the process thread. The flowchart of this control code can be seen below, Figure 26.

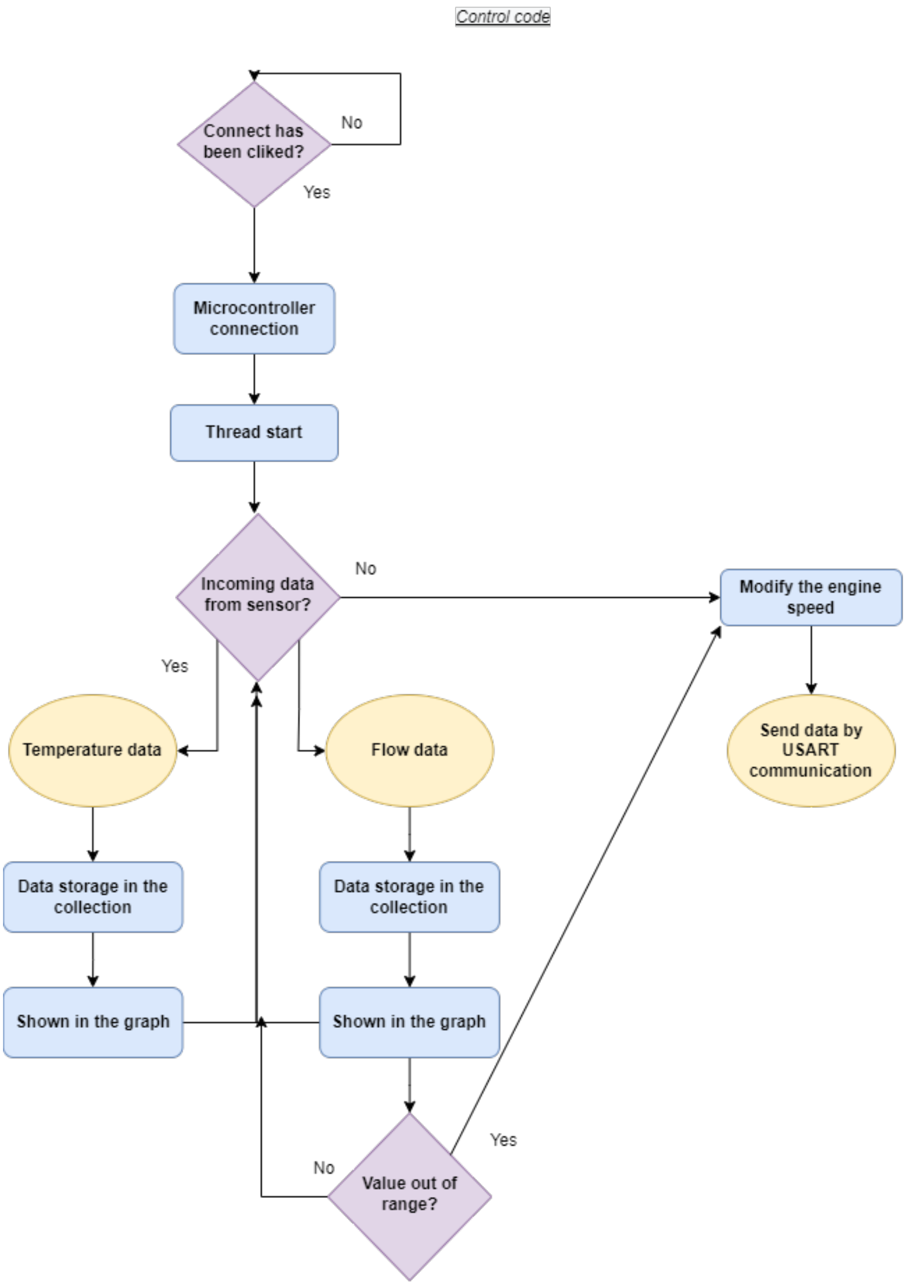


Figure 26: Control code flowchart

3 Results

The flow and temperature monitoring and control system has been tested using water. Water plays the role of perfusion fluid in the system, as there is no access to real pig blood. There is also no access to a porcine kidney, so the pipe endings are not connected to any cannulae of the organ. The water flows freely through the vessel that acts as a container for the organ, being driven by the rotation of the centrifugal pump without exerting any resistance related to the renal artery or vein. The aim of the experiments is to verify that the system is autonomously self-regulating, as well as responding appropriately to the commands given by the operator and measuring and displaying the parameters of interest. The operating system is shown in Figure 27, along with the identification of each of its individual parts.

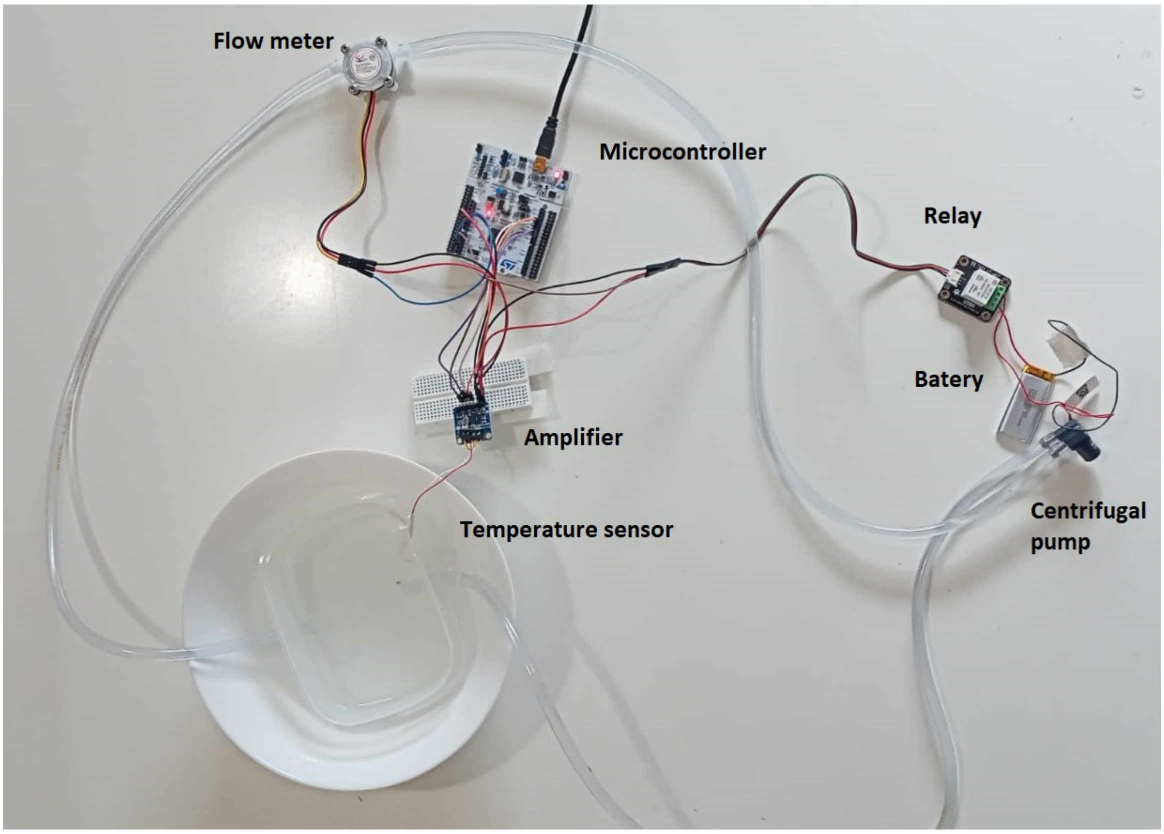


Figure 27: Circuit for testing

The following figures show the state of the water perfusion at three different times. In Figure 28 the experiment has been performed with water heated to a temperature of 27.5ºC. As can be seen in the *Visualisation* tab, the curves in the graphs are those corresponding to the flow rate and temperature collected every second in the system.

This was the first test carried out, so the values of the Y axis range in both graphs are not correctly delimited. This error was solved in later experiments by selecting a range from 0 to 3 LPM in the flowmeter graph and from 25 to 35ºC in the temperature graph.

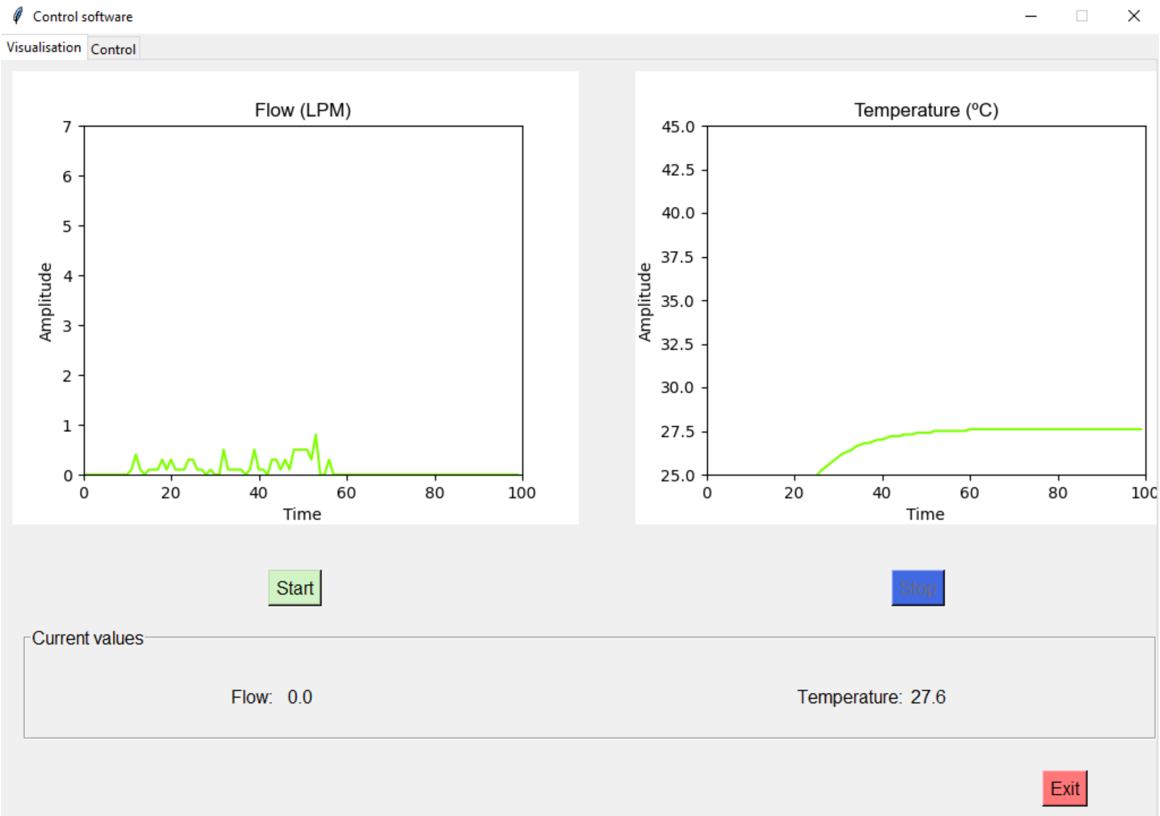


Figure 28: First perfusion system test

As it can be seen in the graph on the right (graph showing the measured temperature values), the system takes a few seconds to start displaying the data. The temperature sensor also takes a few seconds to reach the plateau and shows little or no oscillation in the temperature values read. In other words, there are almost no disturbances during this first part where stabilisation of the experiment takes place.

The graph on the left shows the values of the flow meter read every second, this sensor also takes a few seconds to adapt to the sampling of values. When the control software detects that the centrifugal pump is providing an insufficient flow rate for organ perfusion, it orders to increase the pumping rate, thus producing a rise in the graph. On the other hand, when very high values are detected that could damage the graft, the control software reduces the motor speed, thus decreasing the values on the graph.

These oscillations can be seen in the graph in the form of irregular peaks.

While testing the system, one difficulty has become apparent, which is the capability of the computer processor running the programme to perform all the necessary tasks in real time. In other words, the device executes several tasks concurrently, reading values from several devices, making verifications on this data, carrying out commands, creating data constructions and plotting this data. All these real-time tasks slow down the display of both graphs considerably, as the computer's processor does not have enough capacity to handle all tasks instantaneously. Therefore, in the following tests it has been decided to keep a constant value in one of the graphs while the other graph shows the data in real time. All other processes are running normally, performing the necessary controls and commands, the operator can also handle the engine in the normal way.

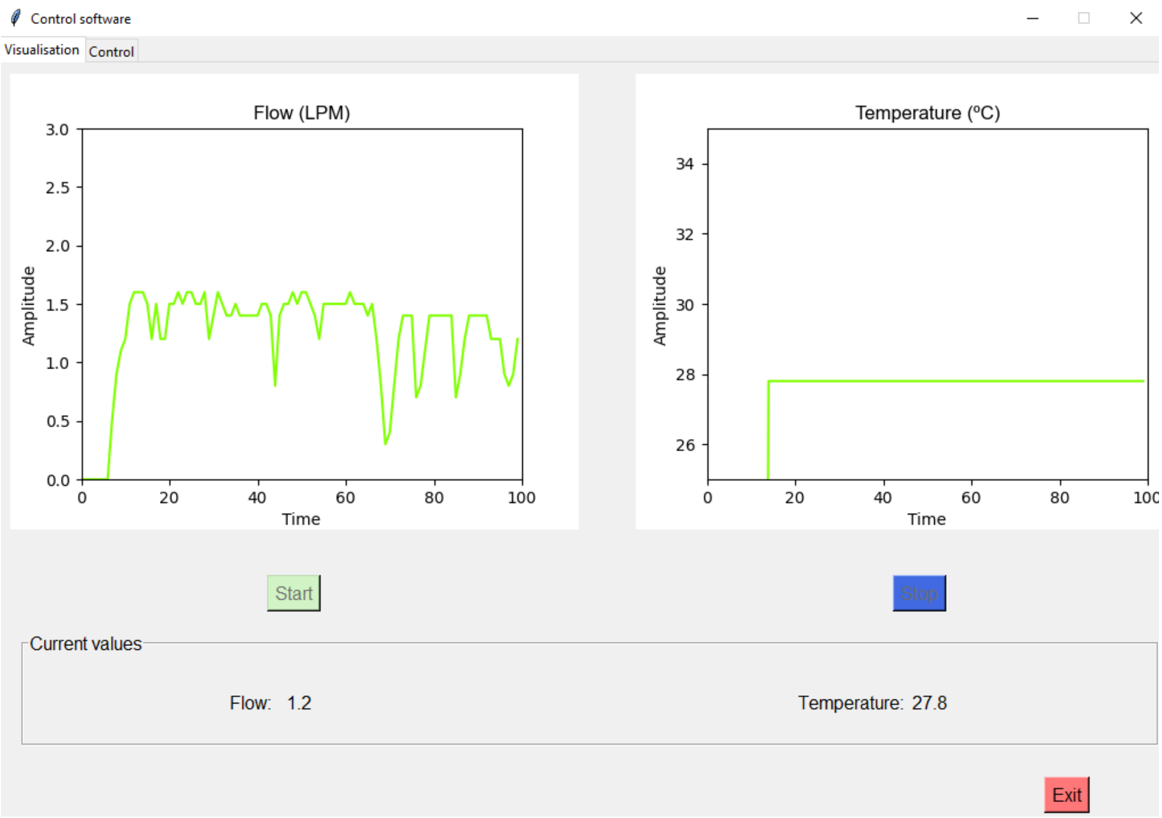


Figure 29: Plotting flow meter data

In Figure 29 it can be seen that the temperature value in the graph on the right is plotted as constant, this does not mean that the read values are also constant. The changing values read in real time from the temperature sensor are still displayed at the bottom, in the box on the tab, but they are not painted on the graph in order to lighten the load on the computer's processor. The graph on the left shows the real time values of the flow meter, the characteristic peaks of the self-regulating flow of the system are detected as well as abrupt variations caused by the manual modification of the engine speeds by the operator in the *Control* tab, shown in Figure 24.

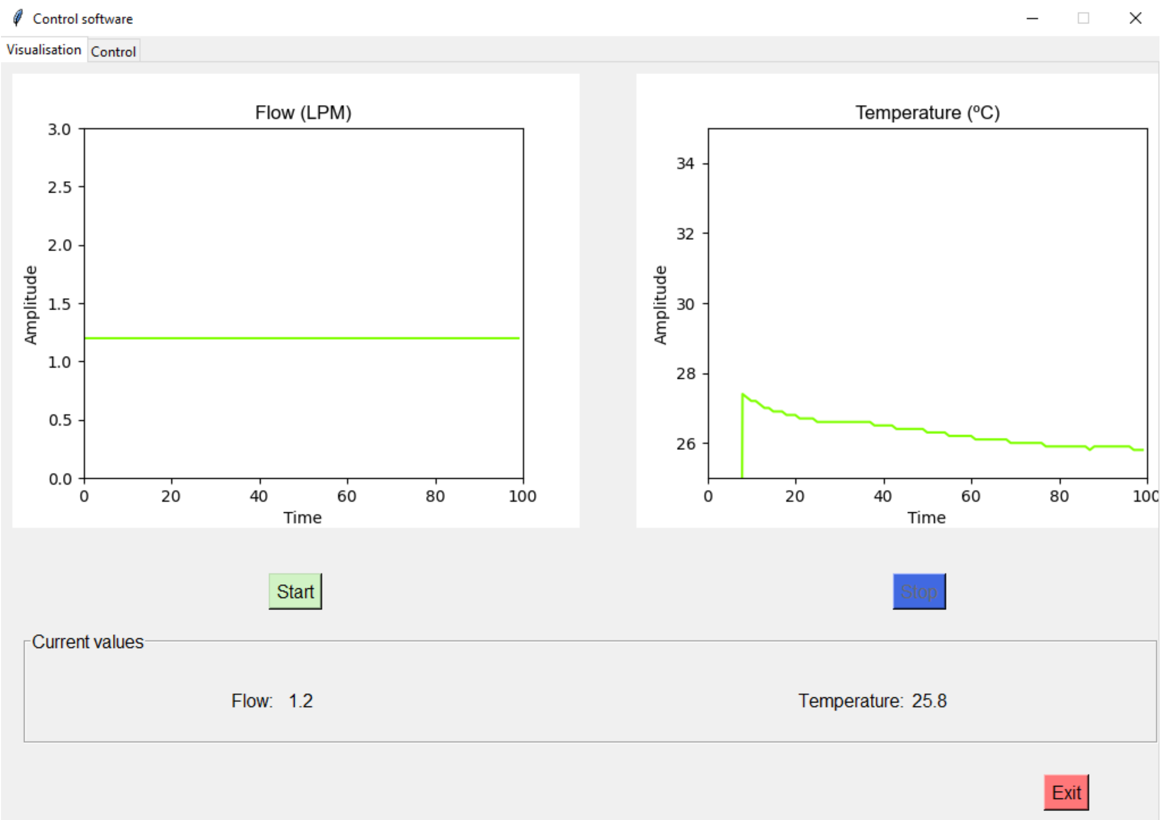


Figure 30: Plotting temperature sensor data

Figure 30 shows a constant flow rate value on the graph but the system is still correctly regulated by the changing values measured by the flow meter every second from the mircrocontroller. The temperature graph starts to take values after a few seconds at 27.7ºC, as this is the temperature at which the water was introduced into the system. After some time in circulation, this water starts to cool down, so the temperature sensor value gradually drops until it stabilises at around 25.8ºC.

This decision to display only the real-time values of one of the graphs is a temporary measure to improve the test results on the current computer. If the program was run on a computer with more processing power, this measure would not be necessary and the system would work optimally.

Since the data from the flowmeter was not displayed in real time on the graph, this data have been collected and saved in a text file to check the correct functioning of the device. This file was then plotted using the programming and numerical calculation tool, MATLAB [34]. The file data has been dumped into a graph to check that the data collected by the flowmeter reflected the behaviour of the system. Its changes in engine speed and compliance with the orders sent by the operator are reflected. The graph obtained with the flow rate data stored in a text file is as follows (Figure 31):

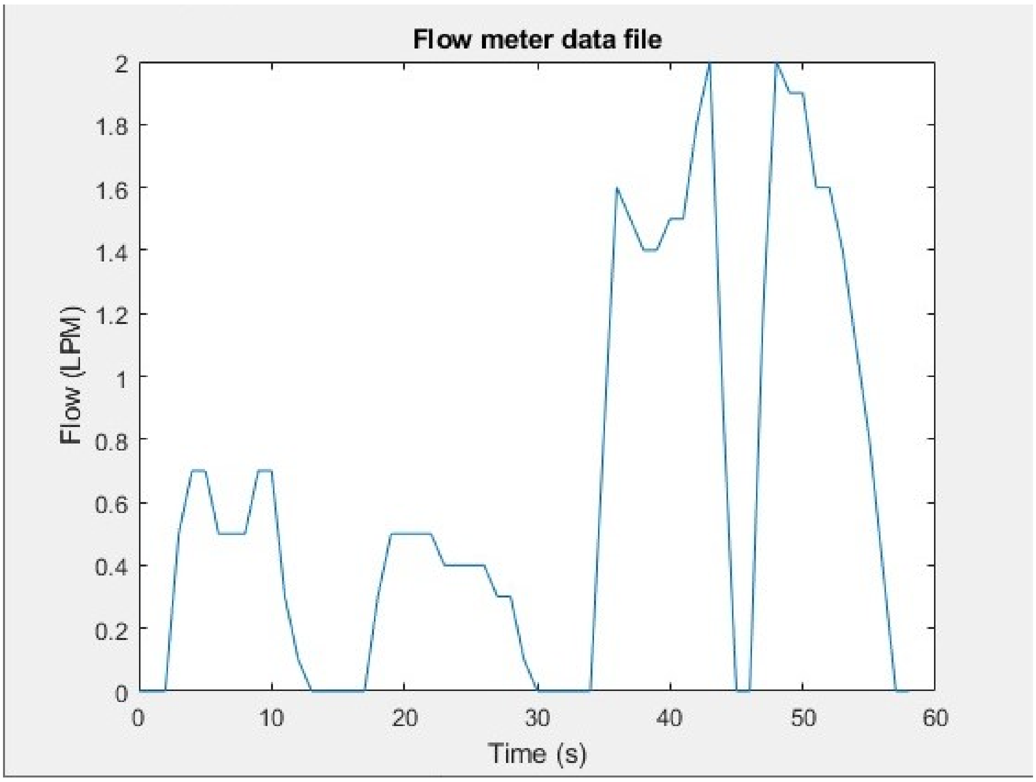


Figure 31: MATLAB plot of flowmeter data stored in text file

The tests carried out check the correct functioning and programming of the software implemented in the system. The configuration code for the NUCLEO-F411RE microcontroller as well as the control software and the graphical user interface can be accessed via the following GitHub link: GitHub Code

4 Conclusions and future lines

4.1 Conclusions

In Section 1.4 the final and intermediate objectives of this Master Thesis have been defined. It can be concluded that the final aim of the project, which is *The design and development of an initial prototype of a normothermic perfusion system for the preservation and possible rehabilitation of a kidney*, has been successfully achieved. In order to complete this final goal, all the intermediate objectives also specified in this section had to be fulfilled one by one.

One of the most important motivations for undertaking this project was the lack of affordable options for perfusing a kidney in normothermia and the need to pursue this line of research further. As we saw in Section 1.3, there is only one commercial option for perfusing an ex-vivo kidney in normothermia, the XVIVO Kidney Assist [13]. Therefore, the lack of competition and the high technology used makes this product unaffordable for a large majority of hospitals.

The aim of this project was to develop a circuit that is easy to use, affordable and with full integration of computer control. This allows autonomous adaptation of the system to the needs and characteristics of the organ. The quality of the graft to be transplanted differs greatly from one kidney to another, as explained in Section 1, so the preservation conditions vary depending on its characteristics. The software control implemented in the perfusion machine is a control system capable of maintaining the isolated kidney in a constant and stable state. It adapts autonomously to the characteristics of the organ, as well as to the changes that occur during perfusion.

The prototype allows constant measurement of the parameters. The use of sensors such as flowmeters and thermometer allows instantaneous values to be recorded without the need for manual calculations from time to time. In addition, the flow control allows autonomous adjustment that improves the manual adjustment, easing the workload of the device operator. It also provides a user-friendly platform capable of working on its own for long periods of time without the need for human supervision.

4.2 Future lines

In line with the objectives achieved, some of the possible proposals for improving the system are listed below:

- Increase the number of sensors and actuators to improve the preservation of the graft as well as widen the range of detection parameters and have more capacity to act on the organ. Some of the components that could be implemented are the following:
  - Pressure sensor: to ensure that the pressure of the perfusion fluid circulating in the organ is adequate to avoid tissue damage.
  - PH sensor: allows the alkalinity or acidity of a solution to be determined.
  - Peristaltic pump: to regulate chemical concentrations and amounts of perfusion fluid administered.
  - Heater/cooler module: to heat or cool the blood circulating in the system taking into account the values acquired by the temperature sensor.
  - Blood oxygenator: to remove carbon dioxide from the blood and to circulate oxygen-rich blood.
- Implementation of biomarkers that, together with the software, evaluate the viability of the organ. Once the perfusion has been performed and the parameters have been recorded, the system could autonomously evaluate these values and indicate whether or not the graft is valid for transplantation and what its possible response would be.
- Minimise the noise of the electric motor due to relay switching. This is caused by the PWM type control signal that regulates the output voltage.
- Design of an enclosure that supports the hardware components of the system and is easily portable.
- Optimisation of processes so as not to use so many of the computer's processor resources.
- Testing with real grafts and blood to get more accurate data on system response.

5 Bibliography

References

[1] Ministerio de sanidad. La organización nacional de trasplantes presenta su balance de actividad en 2021. pages 1–42, 2021. URL www.mscbs.es.

[2] La fundación mutua madrileña financia un proyecto al grupo del irycis "investigación quirúrgica en urología y trasplante renal", que consolida su línea de investigación en preservación renal en el trasplante | noticias | comunicación | instituto ramón y cajal de investigación sanitaria | instituto ramón y cajal de investigación sanitaria. URL https://www.irycis.org/es/comunicacion/noticias/98/.

[3] J. H. Southard and F. O. Belzer. Organ preservation. *Annual review of medicine*, 46:235–247, 1995. ISSN 0066-4219. doi: 10.1146/ANNUREV.MED.46.1.235. URL https://pubmed.ncbi.nlm.nih.gov/7598460/.

[4] Isamu Koyama, Gregory B. Bulkley, G. Melville Williams, and Michael J. Im. The role of oxygen free radicals in mediating the reperfusion injury of cold-preserved ischemic kidneys. *Transplantation*, 40:590–595, 1985. ISSN 0041-1337. doi: 10.1097/00007890-198512000-00003. URL https://pubmed.ncbi.nlm.nih.gov/3907028/.

[5] Pierre Alain Clavien, P. Robert C. Harvey, and Steven M. Strasberg. Preservation and reperfusion injuries in liver allografts. an overview and synthesis of current studies. *Transplantation*, 53:957–978, 1992. ISSN 0041-1337. doi: 10.1097/00007890-199205000-00001. URL https://pubmed.ncbi.nlm.nih.gov/1585489/.

[6] Isamu Koyama, Gregory B. Bulkley, G. Melville Williams, and Michael J. Im. The role of oxygen free radicals in mediating the reperfusion injury of cold-preserved ischemic kidneys. *Transplantation*, 40:590–595, 1985. ISSN 0041-1337. doi: 10.1097/00007890-198512000-00003. URL https://pubmed.ncbi.nlm.nih.gov/3907028/.

[7] medicina intensiva, 2008. URL www.elsevier.es/medicinaintensiva.

[8] Shawn D. St. Peter, Charles J. Imber, and Peter J. Friend. Liver and kidney preservation by perfusion. *Lancet (London, England)*, 359:604–613, 2 2002. ISSN 0140-6736. doi: 10.1016/S0140-6736(02)07749-8. URL https://pubmed.ncbi.nlm.nih.gov/11867131/.

[9] Introducción J Briceño, M A Gómez-Bravo, Perfusión Normotérmica, C Fondevila, A J Hessheimer, J C García-Valdecasas, Donación En, Asistolia R Ciria, D Garrote, J Briceño, K Muffak Granero, S Rufián, J A Ferrón, P López-Cillero, I Bilbao, M Abradelo, J L Lázaro, R Sanabria, J Bueno, E Moreno, R Charco, and C Jiménez. Medicina clinica split ex situ e in situ: técnica y resultados, 2012. URL www.elsevier.es/medicinaclinica.

[10] Lifeport kidney transporter | organ recovery systems. URL https://www.organ-recovery.com/lifeport-kidney-transporter/.

[11] Rm3 kidney perfusion system – waters medical systems. URL https://wtrs.com/rm3-kidney-perfusion-system/.

[12] Mediphenix – donor organ transport box – indes. URL https://indes.eu/en/portfolio/donor-organ-transport-box/.

[13] Xvivo - extending horizons in organ transplantation. URL https://www.organ-assist.nl/.

[14] Instituto ramón y cajal de investigación sanitaria. URL https://www.irycis.org/es/.

[15] Nucleo-f411re - stm32 nucleo-64 development board with stm32f411re mcu, supports arduino and st morpho connectivity - stmicroelectronics. URL https://www.st.com/en/evaluation-tools/nucleo-f411re.html.

[16] G1/8 inch water flow sensor - seeed wiki. URL https://wiki.seeedstudio.com/G1-8_Water_Flow_Sensor/.

[17] Arduino water flow sensor to measure flow rate volume. URL https://how2electronics.com/arduino-water-flow-sensor-measure-flow-rate-volume/.

[18] Pt temperature sensor – ptf family. 2015.

[19] General description benefits and features max31865. URL www.maximintegrated.com.

[20] Bomba de agua rs pro components. URL https://es.rs-online.com/web/p/bombas-de-agua/0480122?cm_mmc=ES-PLA-DS3A-_-google-_-CSS_ES_ES_Catchall_SSC-_-Ad+group-_-480122&matchtype=&pla-293946777986&gclid=CjwKCAjwloCSBhAeEiwA3hVo_eQpQWvFSFHh7V0aWzrKgsWsLFor5LfRO3G15Yggv3MhN3wFMiLaZBoCkxsQAvD_BwE&gclsrc=aw.ds.

[21] Nimo bateria recargable bat 528 3.7v 500 mah litio, polimero. URL https://laloelectronica.com/baterias-y-pilas/24387-nimo-bateria-recargable-bat-528-37v-500-mah-litio-polimero.html.

[22] Gravity__digital_relay_module__arduino__raspberry_pi_compatible__sku__dfr0473-dfrobot. URL https://wiki.dfrobot.com/Gravity__Digital_Relay_Module__Arduino_&_Raspberry_Pi_Compatible__SKU__DFR0473.

[23] Zipable multi-color jumper wires. 2015. URL www.BusBoard.com.

[24] Cables usb cable usb a a usb b macho mini usb b rs pro, 3m. URL https://uk.rs-.

[25] Stm32cubeide - integrated development environment for stm32 - stmicroelectronics. URL https://www.st.com/en/development-tools/stm32cubeide.html.

[26] Python software foundation | python software foundation. URL https://www.python.org/psf/.

[27] Visual studio code - code editing. redefined. URL https://code.visualstudio.com/.

[28] Usart básico - coffeebrain-wiki. URL http://www.coffeebrain.org/wiki/index.php?title=USART_B%C3%A1sico.

[29] Freertos - market leading rtos (real time operating system) for embedded systems with internet of things extensions. URL https://www.freertos.org/.

[30] :cmsis - modm barebone embedded library. URL https://modm.io/reference/module/modm-cmsis/.

[31] Getting started with stm32 - how to use spi. URL https://www.digikey.com/en/maker/projects/getting-started-with-stm32-how-to-use-spi/09eab3dfe74c4d0391aaaa99b0a8ee17.

[32] Github - nimaltd/max31865: Max31865 library for stm32 hal. URL https://github.com/nimaltd/max31865.

[33] Carmine Noviello. Mastering stm32 a step-by-step guide to the most complete arm cortex-m platform, using a free and powerful development environment based on eclipse and gcc, 2015. URL http://leanpub.com/mastering-stm32.

[34] Mathworks - creadores de matlab y simulink - matlab y simulink - matlab simulink. URL https://es.mathworks.com/?s_tid=gn_logo.

[35] The 17 goals | sustainable development. URL https://sdgs.un.org/es/goals.

A Ethical, economic, social and environmental aspects

A.1 Introduction

This annex analyses the social, economic, environmental and ethical impacts of this Master Thesis. It is a multidisciplinary work, in which tools derived from Biomedical Engineering are applied to a clinical need. The project is framed in the health sector together with the technological innovation sector, since the main focus is based on improving the preservation of renal grafts for transplants. In other words, this project has a direct impact on improving the quality of life and health of patients by making use of low-cost electronic components available in the current market.

The work is carried out in collaboration with the Urology Services of the Hospital Universitario Ramón y Cajal and the Robotics and Control System Group of the Escuela Técnica Superior de Ingenieros de Telecomunicación of the Universidad Politécnica de Madrid. The need to be covered by this project is the creation of a first low-cost prototype that allows the self-regulated perfusion in normothermia of an isolated ex-vivo kidney. Currently, only one device with these characteristics is commercialised (the XVIVO Kidney Assist [13]) although more and more lines of research are pointing to the normothermic perfusion method as the most suitable possible method for the preservation and rehabilitation of renal grafts. The aim of this work is to develop a device that allows perfusion data to be obtained and quantified in real time and that self-regulates when any of these parameters falls outside the range determined to be correct. The ultimate goal of this work is to improve the preservation of renal grafts by means of a low-cost device that is affordable for a large number of hospitals.

A.2 Description of relevant impacts related to the project

Having explained the development of the normothermic perfusion prototype, it is necessary to analyse the social, economic, ethical and environmental impact of the project. An in-depth analysis of the impacts of the project will be made in the following section. This section will only briefly describe the most relevant impacts identified in the following section.

- **Ethical impact**: patient safety is enhanced as the normothermic perfusion system allows professionals to make more informed decisions about what actions should be taken. This is possible thanks to the visualisation of the acquired perfusion data more accurately. By having access to real-time information, clinical errors made in the treatment of the organ will be reduced.
- **Social impact**: perfusion in normothermia aims to improve and restore the organ's

functionality. If the organ function improves, fewer grafts will be discarded, so the waiting list of patients will decrease and donated organs will have a better response and a longer lifespan.

- **Economic impact**: this device has a lower price than the main competitor ([13]). The low cost of the device makes it accessible to a larger number of hospitals. The greater number of devices available means that the number of patients on the waiting list is reduced. This is because a greater number of perfusion machines would result in a lower percentage of grafts being discarded, reducing the waiting list of patients.
- **Environmental impact**: reducing the time that clinical treatments are given to patients on waiting lists would also mean a reduction in the use of environmentally damaging materials or disposable products.[35]

A.3 Detailed analysis of some of the main impacts

The physician's code of ethics dictates that physicians aim to preserve the safety, integrity and proper care of patients. The normothermic perfusion system allows professionals to make more informed decisions about what actions should be taken, by being able to visualise the acquired perfusion data more accurately. Thanks to having access to real-time information, clinical errors made in organ treatment will be reduced. Therefore, patient safety will prevail above all else, which has an impact on the ethical aspect of the project.

As seen in Section 1, the number of kidney transplants is growing significantly every year, resulting in longer and longer waiting lists of patients in need of a transplant. This problem has worsened in recent years as the grafts for transplantation belong to increasingly older people, suffering from some kind of pathology. This results in higher numbers of grafts discards. By applying normothermic perfusion, the aim is to improve and recover the functionality of the organ. If the function of the organ improves, fewer grafts will be discarded, so the waiting list of patients will decrease and the donated organs will have a better response and a longer life expectancy. This would have a major impact on society.

This Master Thesis seeks to create a first low-cost prototype for normothermic perfusion of ex-vivo kidneys. The lack of commercial options currently available makes the acquisition of such a perfusion device extremely expensive. By lowering the cost of the device would make it possible for more hospitals to have access to this product. A greater number of perfusion machines would result in a lower percentage of grafts being discarded, reducing the waiting list of patients. These patients who are on the waiting list for a kidney are undergoing some kind of treatment, such as dialysis. These treatments reduce the patient's quality of life and have a large economic impact. If the waiting time is shorter, the duration of the treatments patients undergo will also be shorter, thus reducing hospital costs.

Related to the reduction of treatments for patients on the waiting list, an environmental impact of the project can be observed. Although it is not a direct environmental impact, it can be deduced that a reduction of clinical treatments to patients would also mean a reduction of materials or disposable products that cause damage to the environment.

A.4 Conclusions

The project generates a great positive social and economic impact, as well as a minor impact on ethical and environmental aspects. The development of this Master Thesis shows the future potential of normothermic perfusion of pre-transplanted organs. Although there is still a long way to go, we can see encouraging results in the search for techniques that improve the preservation and functionality of transplanted organs.

The work of this Master Thesis covers mainly two goals of the 2030 Agenda for Sustainable Development. This 2030 Agenda for Sustainable Development provides a shared plan for peace and prosperity for people and planet, now and in the future, for all UN Member States since 2015. Its core are the 17 Sustainable Development Goals (SDGs), which constitute an urgent call for action by all countries in a global partnership [35]. The third and tenth goals are covered. Goal 3 aims to ensure healthy lifestyles and promote well-being for everyone at all ages.t The project aims to increase the quality of life for people with a transplanted kidney by improving the function and lifespan of the graft, using the nomothermic preservation method. The tenth objective is to reduce inequalities within and between countries. This Master Thesis aims to develop a low-cost tool that can be affordable for all hospital centres, including those with lower purchasing power.

B Financial budget

This Master Thesis has been funded by the Instituto Ramón y Cajal de Investigación Sanitario (IRYCIS)[14]. The economic budget of the project has been calculated taking into account the costs related to the material resources used, mentioned in Section 2.1.

	Cost (€)
Microcontroller	12.30
Flow meter	8.65
Temperature sensor	3.86
Acquisition card	13.60
Centrifugal pump	49.44
Batery	6.53
Relay	7.52
Pipelines and container	16.25
Wires	4.43
Conector	3.5
TOTAL	**126.08**

Table 2: Material expenses

This total value of electronic products is subject to 21% VAT (Value Added Tax). This is the default rate applied to all products and services in Spain. The total budget adding VAT is 152.56€, as shown in Table :

	Cost (€)
Material expenses	126.08
VAT (21%)	26.48
TOTAL	**152.56**

Table 3: Total cost including VAT

C Hardware components

This Annex explains in detail the features of each component of the system mentioned in Section 2.1.

Development board

The STM32 NUCLEO-F411RE development board is characterised by the following features: [15]

- 32-bit ARM Cortex-M4 microcontroller Ref: STM32F411RET6
- Frequency 100 MHz max CPU
- VDD VD from 1.7 V to 3.6 V
- 512 KB Flash
- 128 KB SRAM
- GPIO (50) with 12-bit external interrupt capability
- ADC with 16 channels
- RTC
- Timers (8)
- I2C (3)
- USART (3)
- SPI (5)
- USB OTG Full Speed
- SDIO
- 32.768 kHz crystal oscillator
- ARDUINO Uno V3 expansion connector and ST morpho extension pin headers for full access to all STM32 I/Os
- Flexible power options: ST-LINK USB VBUS or external power supplies
- On-board ST-LINK debugger/programmer with USB re-routing capability: mass storage, virtual COM port and debug port

- Support for a wide range of integrated development environments (IDEs), including IAR Embedded Workbench, MDK-ARM and STM32CubeIDE
- External SMPS to generate Vcore logic power supply
- 24 MHz or 48 MHz HSE
- Dedicated connector for external SMPS experimentation
- Micro-B or Mini-B USB connector for the ST-LINK
- MIPI debug connector

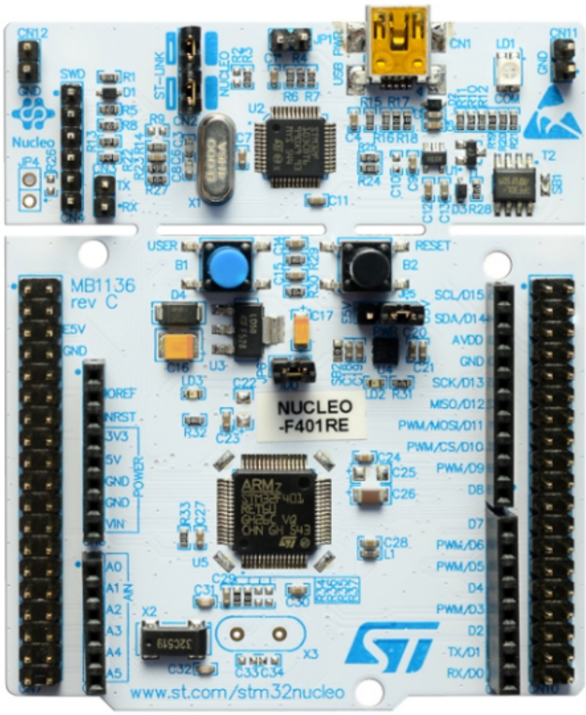


Figure 32: STM32 NUCLEO-F411RE [15]

Flow sensor

The features of the G1&8" Water Flow Sensor are as follows: [16]

- Working Voltage DC 5V to 24V
- Max. Working Current 15mA (DC 5V)
- Flow Rate Range 0.3 $\sim$ 6L/min
- Load Capacity $\leq$ 10mA(DC 5V)
- Operating Temperature $\leq$80℃

- Liquid Temperature $\leq$120℃
- Operating Humidity 35% $\sim$ 90%RH
- Water Pressure $\leq$0.8MPa
- Storage Temperature -25$\sim$ +80℃
- Storage Humidity 25%$\sim$95%RH
- Weight G.W 30g

Figure 33: G1&8" Water Flow Sensor [16]

Temperature sensor

The features of the thermoresistance are as follows: [18]

- RTD Element Type: Platinum Thin Film Temperature Element
- Lead Wire Style: Ni/Au
- Element Package: Ceramic
- Connector Type: Open Ends
- Tolerance Class: Class A / F0.15
- Wire Count: 2
- T_ref for Resistance(°C): 0
- Resistance (at T_ref)(Ω): 1000 (0 °C)
- Accuracy (at T_ref)(°C): ± 0.15

- TCR at (T1 and T2)(ppm/°C): 3850
- T1 and T2 for TCR(°C): 0 and +100
- Ambient Temperature Range: -30 ∼ 300°C[-22 ∼ 572°F]
- Body dimensions: 2x5mm

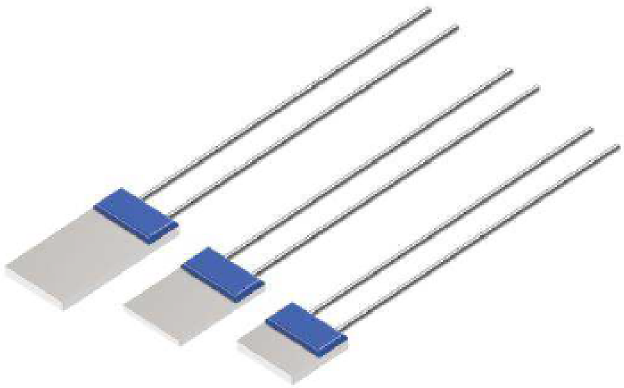


Figure 34: Thermoresistance [18]

Acquisition card

The features of the Adafruit RTD Sensor Amplifier are as follows: [19]

- Handles platinum RTDs from 100Ω to 1kΩ (at 0°C)(PT100 to PT1000)
- Compatible with 2, 3 and 4-wire sensor connections
- SPI compatible interface
- 20 pin TQFN and SSOP packages
- 15 bit ADC resolution; nominal temperature
- 0.03125NC resolution
- 21 ms conversion time (max.)
- ±45V input protection
- Fault detection (RTD element open, RTD short-circuit with voltage out of range, or short circuit through RTD element)
- Unit weight: 2,600 g

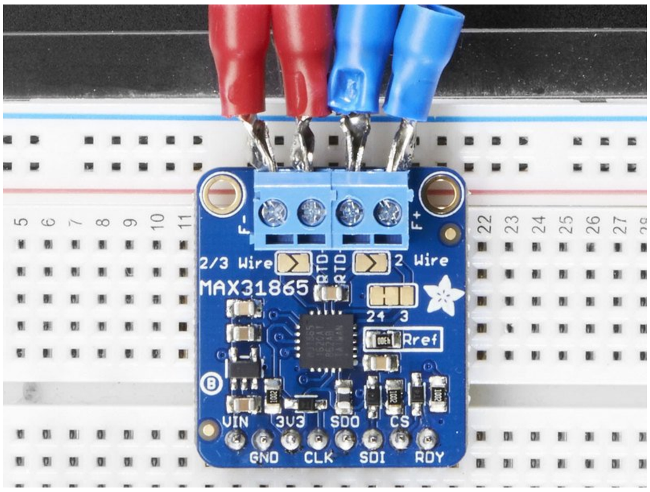


Figure 35: Adafruit RTD Sensor Amplifier MAX31865 [19]

Centrifugal pump

The features of the RS Pro centrifugal pump are as follows: [20]

- Body material: aluminium
- Pump type: centrifugal
- Supply voltage: 3 $\sim$ 4 V
- Input power: 0.84 $\sim$ 1.6W
- Maximum flow rate: 650ml/min
- Maximum operating pressure: 327 mbar
- Operating temperature range: 20 °C $\sim$ +100 °C
- IP Protection Rating IP64
- Inlet and outlet connections: 3.2mm
- Maximum head: 1.9m
- Small size: 16x16x29mm
- Weight: 11-13g

Figure 36: RS Pro centrifugal pump [20]

Relay

The features of the Gravity Digital Relay Module are as follows: [22]

- Operating voltage: 2.8V $\sim$ 5.5V
- Input signal: High level ($\geq$2.8V); Low level ($\leq$0.5V)
- Max. switching current: 10A
- Max. switching voltage: 35VAC; 30VDC
- Rating load: 10A@35VAC; 10A@28VDC
- Max. switching power: 300W
- Operate time: $\leq$10ms
- Release time: $\leq$5ms
- Dimension: 38x33x20 mm /1.5x1.3x0.81 inches
- Weight: 20 g
- Mechanical life: 1 million times
- Electrical life: 50 thousand times

Figure 37: Gravity Digital Relay Module [20]