Grit Rhinow*, Carsten Tautorat*, Swen Grossmann, Niels Grabow, Stefan Siewert, Klaus-Peter Schmitz and Wolfram Schmidt

Universal single-board computer based control unit for biomedical test benches

Abstract: Continuous adaptation of international standards for medical devices requires recurrent modification of test benches. A universal control unit using an open software environment is presented to simplify the maintenance of various test benches. Our developed control unit is equipped with a Raspberry Pi 4, with standard communication interfaces and application-specific electronic assemblies. The software is based on Node-RED, a browser-based editor.

A measuring setup was adapted for a flow perfusion system. Our control unit simplifies the handling of the flow perfusion system by controlling a hydraulic pump and all required valves. A software programmed PID algorithm adapts the speed of the pump to adjust the pressure automatically. Actuators like proportional pinch valves are handled to control volumetric flow and pressure within the circulation. Consequently, the user directly observes changes inside the system. The measured data are stored and are available for documentation.

Keywords: Biomedical, testing, embedded system, automation, Raspberry Pi, Node-RED

<https://doi.org/10.1515/cdbme-2021-2160>

1 Introduction

Proper testing of medical devices and its improvement represents an essential part of biomedical developments. Following continuous adaption of international standards for medical devices, test benches need further modifications as

well. Many test benches consist of various hardware interfaces and software versions and are controlled with outdated personal computers. The measurement software is written in different programming languages (e.g. LabView, Matlab) or consists of commercial software written for single system components. As a result, special programming skills are required to make fast adaptations. Especially outdated software impedes modifications in a proper amount of time.

An existing self-developed flow perfusion system [1, 2] for example used LabView to measure the flow and pressure. Although it is also possible to read the actual flow rate from the converter via Modbus RTU, the system used the analogue output for data logging in LabView. The initial hydraulic pump was operated manually, which makes automation difficult. Consequently, it was replaced by a hydraulic pump that sets the pump speed via Modbus interface. Subsequently the user controls the speed in the pump software and observes the actual measurement data parallel in LabView. Two manual valves were included to control the flow and pressure.

Nevertheless, the overall handling of the system was inconvenient. As a result, we developed a measuring setup that uses standard communication interfaces for the peripheral devices and can be handled in just one user interface. The presented system can be easily adapted to multiple requirements because only standard interfaces are used for hardware as well as software connections.

2 Methods

To build a flexible system for many test situations, the developed control unit is based on a Raspberry Pi 4 Model B. The Raspberry Pi is a single board computer and an inexpensive alternative to conventional controllers for many laboratory applications. In addition to connecting the Raspberry Pi to various sensors and actuators, the Raspberry Pi also offers a possibility to develop built-in graphical user interfaces (GUI) with commonly used programming languages like Python, HTML/CSS, QT or Node-RED.

*Grit Rhinow, Carsten Tautorat: Institute for Biomedical Engineering, Rostock University Medical Center, Friedrich-Barnewitz-Straße 4, Rostock-Warnemünde, Germany, e-mail: grit.rhinow@uni-rostock.de

Niels Grabow, Wolfram Schmidt, Klaus-Peter Schmitz: Institute for Biomedical Engineering, Rostock University Medical Center, Rostock-Warnemünde, Germany

Swen Grossmann, Stefan Siewert, Klaus-Peter Schmitz: Institute for ImplantTechnology and Biomaterials e.V., Rostock-Warnemünde, Germany

2.1 Hardware design

The control unit is built in a modular way and consists of a Raspberry Pi 4 Modell B, a SD card with 32 GB memory, a voltage regulator, a real time clock and a Wi-Fi antenna. **Figure 1** illustrates the principle block diagram of the developed device.

Figure 1: Principle block diagram of the control unit.

In the first experimental setup, a CPU temperature of 80 °C was observable. With a temperature greater than 80 °C the Raspberry Pi throttles the CPU back to prevent permanent overheating of the CPU [3]. Therefore the device is equipped with an independent fan-speed controller that reduces the temperature to about 55 °C. For each measurement, the system needs reliable timestamps. Since the Raspberry Pi has no real-time clock (RTC) onboard, the control unit is also equipped with an RTC module. Using the Raspberry Pi without a monitor and keyboard requires a network connection and enabled SSH and VNC [4] in the Raspberry Pi's configuration file for remote access.

Figure 2: Control unit for biomedical test benches including Raspberry Pi 4. The application specific assemblies are connected via Modbus, I²C or the USB ports of the Raspberry Pi.

With the help of a Wi-Fi antenna and the wireless router software RaspAP [5] users are able to connect a laptop or a

handheld device to the control unit. The control unit works as a Wi-Fi access point and for identification purpose each test bench needs its unique Wi-Fi SSID. For data communication to all peripheral devices like sensors and actuators, the control unit contains two industrial standard communication interfaces like Modbus RTU and I²C. **Figure 2** shows the front and back panel of the control unit. All the application-specific assemblies like analogue-digital converter (ADC), digital-analogue converter (DAC), level shifter and relay driver can be placed on a second printed circuit board (PCB) nearby the test bench and are connected to the control unit via I²C or Modbus RTU.

2.2 Software design

In addition to the hardware design, a new software concept was realised. A simple and modern way of creating user interfaces with networking capabilities is Node-RED [6].

Node-RED is a browser-based editor and is running on the Raspberry Pi. Programs can be edited within a browser without installing special programming environments. All the functions are described in so-called nodes and wired together in flows. Node-RED stores incoming data in JavaScript objects called messages and passes them through the nodes, the basic building blocks (see **Figure 3**).

Figure 3: Simplified block diagram of a Node-RED structure, consisting of three nodes. The input node generates a message containing the most useful data. A message is processed inside the function nodes and output nodes deliver messages to peripheral devices or the GUI

There are three different types of nodes. Input nodes create messages for the flow. Predefined function nodes (e.g. delayed message) forward the message to the output nodes. There is also a “function” node for user-specific functionality. Within this node it is possible to write own JavaScript functions. Node-RED also offers a simple GUI called dashboard. Dashboard widgets containing gauges, buttons, text inputs and other GUI elements can be created and designed within the Node-RED editor.

In our control unit, Node-RED starts automatically after powering the system. The LEDs on the front panel switch from “STANDBY” to “ACTIVE” and indicate that the program is running. The system administrator is able to connect to the Node-RED editor via a web browser. For security reasons, this editor can be password protected.

For most common applications, standard flows were generated that can be adapted for different test benches. These flows are saved as subflows and contain reading analogue inputs, a PID controller node, logging measurement values into a databank for visualisation or a shutdown button. The subflows serve as a workaround for new test bench scenarios in order to reduce developing time.

3 Results

The first prototype was built for an existing flow perfusion system. All data measurement and system handling is implemented in Node-RED using the standard subflows. **Figure 4** shows the block diagram of the modified flow perfusion system.

Figure 4: Block diagram of the modified flow perfusion. In this example the pressure controlled pump conveys water from a water tank through a proportional valve to the measurement path for measuring low flow rates. The water passes through the opened pinch valve and the flow sensor. Then the water passes the sample chamber and the pressure is measured.

To simplify the overall handling of the system as well as future modifications and maintenance, the system is equipped with the control unit and extended with an application-specific PCB. According to the application, the PCB contains eight analogue inputs and two analogue outputs as well as additional general digital IOs. At the moment, two pressure sensors with different measurement ranges (series 86A 5 psi and 5 bar, Measurement Specialties Inc., USA) are connected to the PCB and can be selected for the measurement in the control unit. In this way, the user applies the determined pressure sensor to the sample chamber with a luer lock connector and selects the sensor for the speed control in the GUI. Additional sensors can easily be added to the system

due to its modular software design and a user-friendly scaling inside the GUI. This opens the system to varying test situations.

The PCB contains analogue outputs for connecting with analogue command signals to the control unit. In this setup two stepper motor pinch valves were added for a manual fine adjustment of the volumetric flow and pressure inside the water circulation. One valve is used for adjusting the flow and pressure during the measurements and a second valve implements a bypass which enables an optimisation of the pressure and flow range.

To simplify the test bench handling, two flow sensors with different measurement ranges (Leviflow LFS-008 and Leviflow LFS-08, Levitronix, Switzerland) operate in parallel in the system. The matching flow converters (Leviflow LFC-1C, Levitronix, Switzerland) are located in a separate case. On the back panel this case offers sockets for the analogue output signals and the Modbus RTU connection to the control unit whereas the fluidic connections for the flow sensors are located on the front panel.

Eight relay driver and two opto-couplers offer the possibility to connect actuators like valves and other inductive loads. By adding two solenoid pinch valves to the system, the user is able to select the flow sensor for the measurement. Inside the GUI, the measurement path is selected and one solenoid pinch valve hydraulically shuts off the idle path from the water circuit. Considering overheating of the pinch valves as a systematic measurement uncertainty, special relay driver for inductive loads (iC-GE, iC-Haus, Germany) reduce the power consumption and thus the overheating of the pinch valves [7].

A further improvement is the adjustment of the software driven pump (BPS-i30, Levitronix, Switzerland) to the control unit via Modbus RTU. Thus the user is able to control the pump within the GUI (see **Figure 5**). The automated speed control for the hydraulic pump is a major improvement of the handling. In automatic mode, the user determines a pressure setpoint. To achieve the desired setpoint, a software programmed PID controller adjusts the pump automatically to control the pressure. It is also possible to increase the pressure in 20 mmHg steps until the controller recognises a pressure drop of 50 %. In case of a pressure drop, the PID controller stops automatically the pump to prevent damage to the system or further damage of the samples investigated within the test bench. The PID controller is adjustable for different control scenarios like controlling temperature. Any changes of the settings are directly observable inside the GUI and in automatic mode. All measured data are stored automatically in an ASCII file, CSV or TXT format, for

instance. The date and time ordered ASCII files can be downloaded for documentation and further analysis.

Figure 5: Screenshot of user control inside GUI. GUI is accessible via web browser on different computing devices

4 Conclusion

A universal control unit using an open software environment was developed and implemented into an existing flow perfusion test bench for extending the system functionalities. Major improvements are an overall simplified handling of the test bench. Further sensors can be easily added to the application specific assemblies. Standard pluggable connections also minimise errors due to unintentional wrong connections. Only one GUI performs the data measurement and settings. By connecting to the control unit via Wi-Fi, the program can be modified with Node-RED in a browser. No special development environment is needed for programming. The added PID controller adjusts the pressure automatically by controlling the hydrodynamic pump. Due to the intrinsic modular design, the control unit can be used for further laboratory applications. In general, only the Node-RED program needs adaptation to achieve the system requirements of future test benches. However, when using

special sensors (e.g. sensors with different communication interfaces), the PCB with the application-specific assemblies needs further adjustment.

Author Statement

Research funding: This work was partially funded by the Federal Ministry of Education and Research (BMBF) within the project RESPONSE “Partnership for Innovation in Implant Technology”. **Conflict of interest:** Authors state no conflict of interest. **Informed consent** is not applicable. **Ethical approval:** Not applicable.

Grit Rhinow and Carsten Tautorat equally contributed to this paper.

References

- [1] Ott R, Wüstenhagen C, Martin H, Stiehm M, Schmidt W, Einolf N, Frerich B, Grabow N, Schmitz KP and Siewert S: Experimental and numerical investigations of fluid flow for optimized in vitro stem cell loading in xenograft. *Current Directions in Biomedical Engineering* 2017; 3(2): 799–802
- [2] Grossmann S, Siewert S, Kohse S, Ott R, Schmidt W, Grabow N and Schmitz KP: Standardized technique of water permeability measurement for biomedical applications. *Current Directions in Biomedical Engineering* 2018; 4(1): 633–636
- [3] Raspberry Pi Foundation: Over temperature warning (80–85°C) <https://www.raspberrypi.org/documentation/computers/configuration.html#over-temperature-warning-80-85c>, September 2021
- [4] Raspberry Pi Foundation: Remote Access, <https://www.raspberrypi.org/documentation/remote-access/>, June 2021
- [5] RaspAp team: Documentation, <https://docs.raspap.com/>, June 2021
- [6] The OpenJS Foundation: Node-RED Official Documentation, <https://nodered.org/>, June 2021
- [7] Malzahn U: Magnetspulen in Ventilen, Relais und Aktoren intelligent ansteuern. *ElektronikPraxis* 2012; 21:60-61