

ESE Health Programming Finale Challenge 2024-2025

v1.1 Nov 22, 2024

Case: Advanced Life Support Patient Monitor Simulation together with UtrechtUMC Wilhelmina Children's Hospital

- [Click here to enroll](#)

0.1 Partners

The partners in the Finale are:



HEALTH CONCEPT LAB

The logo features the word "Capgemini" in a blue cursive font, followed by a blue teardrop-shaped graphic.

The logo consists of the letters "G", "E", and "T" in a bold black font. The "E" is composed of three horizontal bars of increasing length from top to bottom, colored dark blue, medium blue, and light blue respectively.

GEMINI EMBEDDED TECHNOLOGY

Introduction

Wilhelmina Children's Hospital (WKZ) in Utrecht (part of UtrechtUMC) is always looking for new ways to improve skills of the nurses and physicians.

Together with the HAN Health Concept Lab they are developing tools for life support training including a manikin to practice high quality CardioPulmonary Resuscitation (CPR) .

Case

Wilhelmina Children's Hospital wants to improve their training by adding vital signs monitoring. In the past we already developed the box to trick real patient monitors using Arduino that gives an idea of the end goal of the challenge (see Fig. 1).



Figure 1: The end goal: a simulated vital signs monitor for training advanced life support.

Figure 2 shows a simulated vital signs monitor. The monitor shown is already available for life support training. WKZ wants to be able to use such a monitor with a manikin to increase realism. This means real vital signs sensors will need to be connected. When a particular sensor (e.g. Electrocardiogram (ECG)) is connected, the simulated vital signs should show on the monitor (in case of the ECG , heart rythm, heart beat, i.e. the green signal in from Fig. 2).

Requirements

Based on the same principle WKZ wants to:

- Only show particular vital signs when the appropriate (simulated) vital sign leads are attached.
- Run the simulator on a Raspberry Pi, also when there is no external network connection is available (hotspot mode).



Figure 2: Example of a vital signs monitor. We see the version of RadboudUMC which is available to use as reference during the challenge.

The Challenge

0.1.1 High Level vs Low Level

Your assignment is to program (parts) of this simulator. You will need to combine both high and in low-level C/C++ code. At the Raspberry Pi you will need to use C++ and the Object Oriented Paradigm (optimized for code quality) possibly together lower level code (optimized for speed to generate the vital signs). On the simulation modules (see Fig. 3) you might need to use low-level C-code as well.

The following vital signs

- ECG : difficult (Must)
- Blood oxygenation (SpO2): difficult (Should)
- Non-invasive blood-pressure (NIBP) (sensor + actuators): moderate (Should)
- Temperature (cavity and surface/skin): easy (Should)
- Breathing (via the other signals): difficult (Could)

Note: the ECG is mandatory and therefor has MoSCoW priority Must, the other items are Should. Real patient monitors determine breathing via the other signals. Breathing patterns are a Could. Code quality and vital signs quantity all add up to the final result.

0.1.2 High Level

The patient monitor needs to look and feel like a real patient monitor for the nurses.

Explain Monitor Login with: prog6 and WeLikeAChallenge2025!

This is a great app but:

- It lacks performance (limited hardware);
- It cannot run together with hardware (limited simulation options).

First find out what frameworks you can best use on your Raspberry Pi to make such a Graphical User Interface (GUI) that does have great performance.

0.1.3 Low Level

Then we need to be able to interface between the Raspberry Pi and Arduino-like boards. That also helps to separate concerns.

Hardware and Software

In order to be able to support this project also after the challenge, we decided to prescribe the hardware and parts of the software. The Advanced Life Support simulation modules use SAMD21 microcontrollers. The modules are in essence all i2c peripherals. The next section lists the contents of the simulation kit.

The SAMD21 modules can be programmed using Arduino+FreeRTOS or other tools. If you want to use Arduino you can use them as Adafruit Feather M0 Express boards.

Simulation Kit content

There are modules available for all the vital signs described: they each have their characteristics and (estimated) difficulty level.

ALS Simulation Kit content

Fig. 3 shows the simulation kit with the modules listed above.



Figure 3: The simulation modules as interface to connect the vital sign sensors to the simulated monitor.

Fig. 4 shows the SensorHub Module.

The entire kit contents is listed below:

- 2x SensorHub Allows to connect multiple sensors and uses a SAMD21

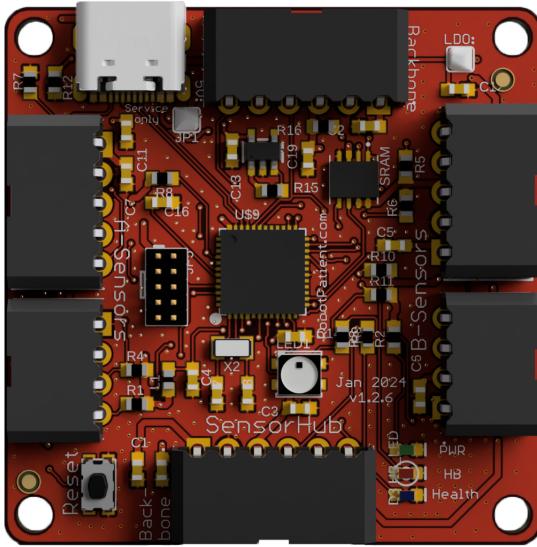


Figure 4: The SensorHub module acts as a communication hub between the sensors and the simulated monitor.

microcontroller: has a USB-C interface to program.

- 1x SimModule ECG + 3 ECG leads and Lead detector board.
- 1x SimModule SpO₂ + blood oxygen saturation sensor.
- 2x SimModule Temp + 1x surface + 1x cavity probe.
- 1x SimModule NIBP + neonatal cuff.
- 1x SimModule Single Row adapter (i2c multiplexer)
- 1x SimModule Adapter (i2c multiplexer)
- 1x SimModule PiInterface with SAMD51 (communication with the Raspberry Pi)
- 1x Raspberry Pi 3/4 + Power Supply + SD-card
- 1x HDMI-screen

Practical Information

Schedule

Date: Wednesday Jan, 29 2025 - 8.30 - 17.00 R29/H2.12-H2.14 (still subject to change)

8.30	Arrival of the engineers. Let's fill our cup with Coffee/Tea
8.45	Meeting the pro's
9.00 - 12.30	Programming
12.30 - 13.15	Break / poster sessions H1xx
13.15 - 15.30	More programming
15.30 - 16.00	Present your work
16.15	Awards
16.30	Drinks

Table 1: (Preliminary) schedule.

Note: Please bring your own lunch and cups. Coffee/tea/hot water will be provided.

Awards

Apart from the grading there will be rewards: (To be determined).

And of course you get credits for Prog 6 (if you pass, see Rubrics).

Please use the Open-Closed principle for this! - The Raspberry Pi software can be higher level C++, preferably object oriented. You should generate vital signs (e.g. an electrocardiogram) to be displayed on a screen via one of the HDMI-ports. The characteristics should be configurable. For the communication between possible control devices and the Pi, MQTT is preferable.

Reference materials

Note: more reference material will be added before the hackaton.

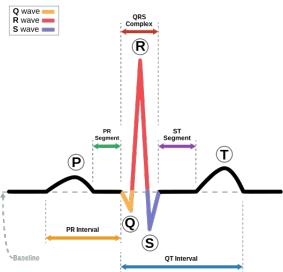


Figure 5: Electrocardiogram.

Fig. 5 shows the basic morphology of an ECG : reference.

The sinus rythm is the standard rythm without anomalies. There are also many anomalies (arrythmia). These could be simulated as well. You can see this in the simulated cloud-based monitor Explain Monitor.