

Petr Kocián

Master's thesis

Computer Science and Engineering
Computer Engineering and Networks
2023/2024

Thesis supervisor:

doc. Ing. Tomáš Koutný, Ph.D.

Real-Time Concept for SmartCGMS

Abstract

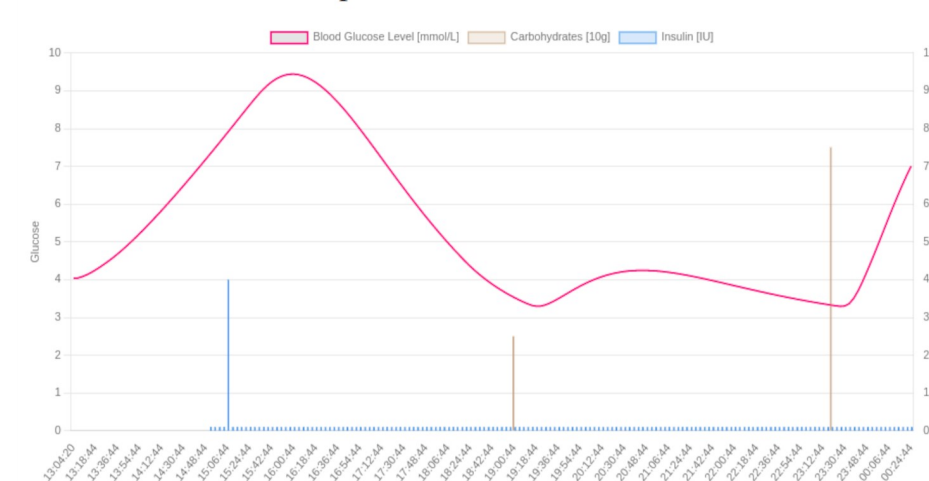
SmartCGMS is a framework for continuous glucose monitoring (CGM) systems. It has been used extensively for testing and simulations, but it needs to improve its support for execution on low-power devices. This paper proposes and implements a real-time SmartCGMS concept based on FreeRTOS. It has been deployed as a native application on an ESP32 and a Raspberry Pi Zero W. For experimentation with loadable modules, the concept was compiled to WebAssembly (WASM), which can be executed on ESP32, Raspberry Pi, x86-64/AMD64, and on the web.

In addition to targeting ESP32 and Raspberry Pi Zero W running FreeRTOS, an environment for WASM runtimes was prepared. This allowed us to execute the SmartCGMS concept on the web.

This project has expanded the range of devices supported by SmartCGMS to include the following platforms:

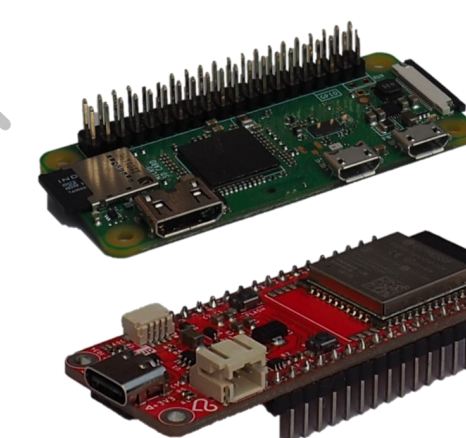
- Raspberry Pi Zero W (ARMv6 ISA)
- ESP32 (Xtensa ISA)
- WASM (portable format)

Blood Glucose Level Graph



Send commands
to an insulin pump

Visualize data



Raspberry Pi Zero W / ESP32
running SmartCGMS framework
to process data from CGM sensor

Transfer estimated
blood glucose level

Introduction

The purpose of bringing SmartCGMS to low-power devices is to create a closed-loop CGM system. A closed-loop CGM system is a set of devices that can autonomously control diabetic patients' blood glucose levels based on measurements from a sensor by injecting insulin using an insulin pump. The SmartCGMS framework is capable of implementing a closed-loop system with SmartCGMS components.

Therefore, the goal of this thesis was to propose, implement, and evaluate the SmartCGMS framework for low-power devices (ARMv6 and Xtensa). The concept must maintain compatibility with desktop devices (x86-64/AMD64), where computationally-intensive pre-clinical simulations take place.

The SmartCGMS Concept

Only necessary parts of SmartCGMS were modified to maintain maximum compatibility across high performance and low-power platforms.

In addition to modifying the SmartCGMS code-base, it was necessary to provide a way how to compile SmartCGMS components for low-power devices. A preprocessor tool that modifies source code files of SmartCGMS components was implemented. It allows the compilation of SmartCGMS components across all supported platforms.

To be able to execute the SmartCGMS concept on the low-power devices, environments with real-time operating system, drivers, and other software components had to be prepared.

Evaluation and Results

The first part of the evaluation focused on verifying the correct functionality of the SmartCGMS concept on the supported platforms. The results confirmed that the SmartCGMS concept works as expected on all supported platforms.

The second part of the evaluation focused on comparing the SmartCGMS concept performance when compiled as a native application versus a WASM module. The evaluation was conducted on GNU/Linux using WASM micro runtime (WAMR). The results were in line with findings published from other, independent benchmarks. The execution time has prolonged, but it still meets practical deadlines.

WAMR is also available on ESP32, but the SmartCGMS WASM module was too large for its limited DRAM. This issue could be addressed either by a SW upgrade (e.g., using a different WASM interpreter) or by a HW upgrade.

Conclusion

The proposed concept was successfully implemented and tested. It offers a way to execute SmartCGMS on low-power devices. Future work should focus on practical experimentation with low-power devices and making any empirically justified modifications.



**FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI**

