



**Politechnika Śląska**

**Wydział Automatyki, Elektroniki i Informatyki**

**Sterowniki dedykowane**

*Raport końcowy*

***Obiekt z regulatorem***

## Cel projektu

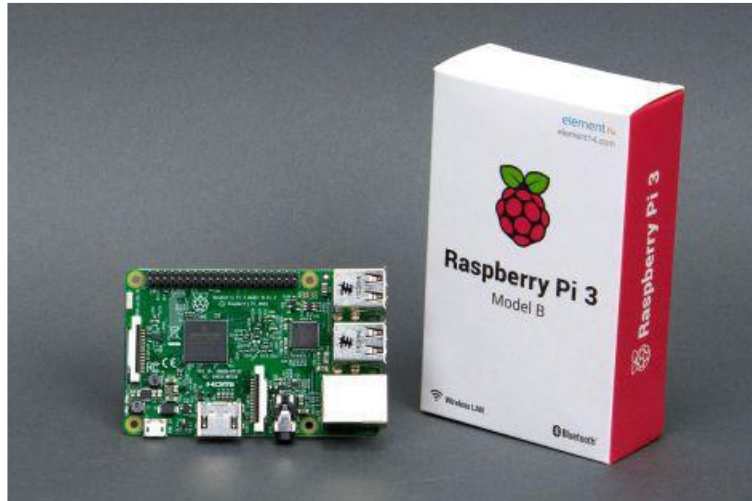
Celem projektu było zasymulowanie przykładowego układu sterowania oraz zaprojektowanie sterownika regulującego wybraną wielkość.

## Sterownik

W projekcie wykorzystano Raspbberry Pi. Jest to platforma komputerowa składająca się z pojedynczego obwodu drukowanego. Jej specyfikacja zależy od wersji. W projekcie użyte zostało Raspberry PI 3. Jest ono dość bogato wyposażone.

Posiada m. in.:

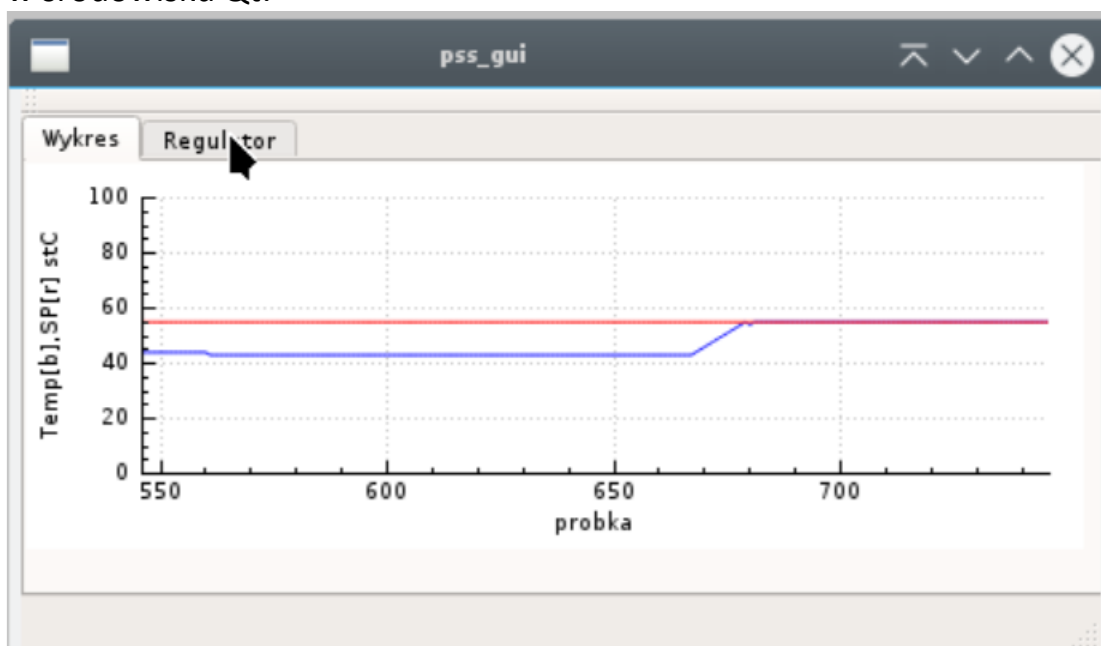
- procesor Broadcom BCM2837 quad-core 64-bitowy ARM-8 Cortex-A53 1,2 Ghz,
- 1 GB pamięci RAM,
- wbudowany moduł WiFi,
- wbudowany moduł Bluetooth 4.1,
- cztery gniazda USB,
- 40 GPIO,
- złącze na kartę microSD,
- port Ethernet,
- złącze HDMI.



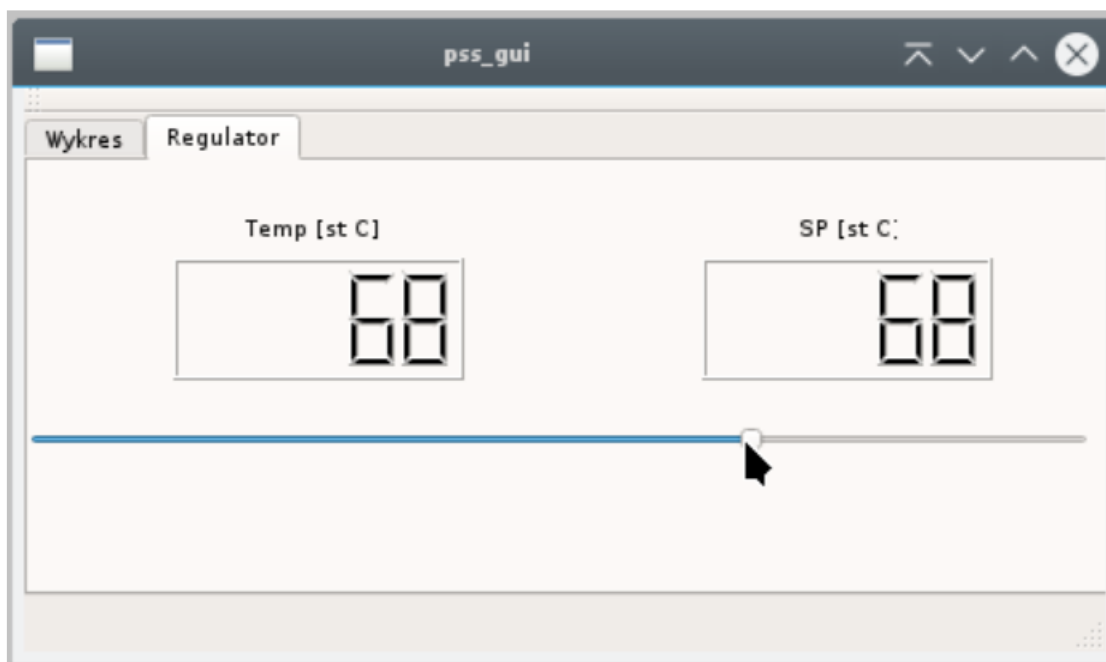
Rysunek 1 Raspberry Pi 3

## Opis działania sterownika

Na platformie Raspberry wgrany został system operacyjny Raspbian. Zaimplementowany został regulator PID w języku C++ sterujący obiektem, w sposób gwarantujący poprawne działanie i tak, by możliwa była zmiana nastaw. W sterowniku zaprogramowany został interfejs użytkownika w środowisku Qt:



Rysunek 2 Przykładowy przebieg wielkości regulowanej oraz wartości zadanej



Rysunek 3 Interaktywny zadajnik wartości zadanej

Użytkownik miał możliwość zadawania wartości zadanej, odczytywania aktualnego wyjścia z obiektu oraz obserwacji przebiegów sterowania. Do platformy mógł być podpięty ekran dotykowy, lub interfejs mógł być wyświetlany na osobnym komputerze. Sterownik łączył się z układem symulującym obiekt za pomocą UARTa, w taki sposób, że obiekt wysyłał bajt danych, który był odczytywany przez sterownik jako wyjście z obiektu, następnie wywoływany był algorytm regulacji. Sterowanie wysyłane było w podobny sposób, Raspberry wysyłało bajt danych, który traktowany był przez obiekt jako wypełnienie sygnału sterującego.

## Obiekt

Obiekt został zaimplementowany przy pomocy zestawu uruchomieniowego EVK1100 z 32 bitowym procesorem AVR oraz napisany w środowisku AtmelStudio w języku C.



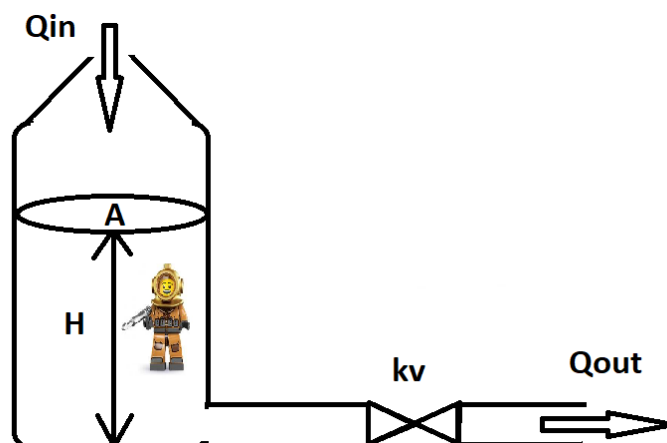
Rysunek 4 EVK110 oraz AtmelStudio wersji 7

Napisany program symuluje działanie zbiornika z wodą dla nurków.

Model matematyczny:

$$A \frac{dH}{dt} = Q_{in}^* - Q_{out}^* \quad (1)$$

$$\frac{dH}{dt} = \frac{1}{A} (Q_{in}^* - k_v H) \quad (2)$$



Rysunek 5 Schemat symulowanego układu

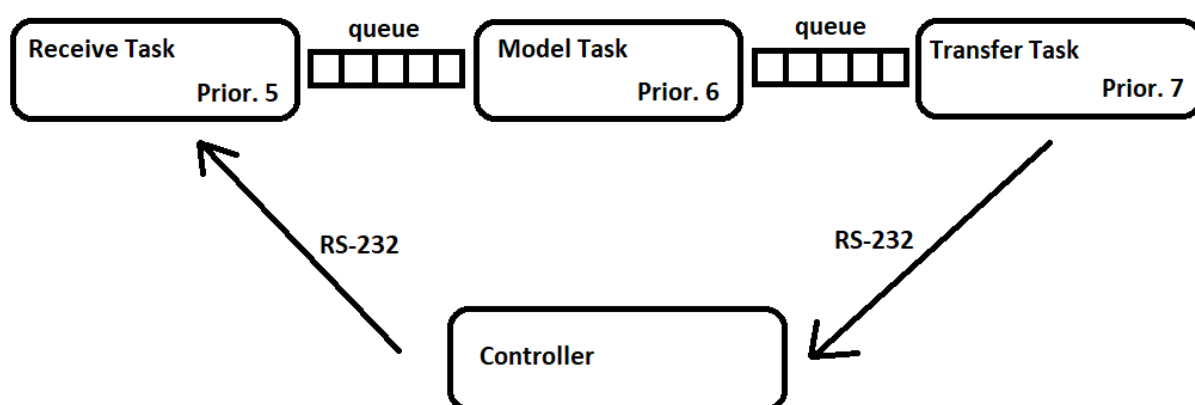
## System czasu rzeczywistego RTOS

Program został napisany zgodnie z zasadami działania systemu czasu rzeczywistego RTOS (freeRTOS). Działanie urządzenia zostało podzielone na 3 odrębne funkcjonalności, do których zostały przydzielone poszczególne wątki czyli taski. Komunikacja między taskami odbywa się za pomocą kolejek o odpowiedniej długości. System obsługuje dany wątek przez ściśle określony czas, a następnie przechodzi do kolejnego zgodnie z nadanym priorytetem.



Rysunek 6 Logo freeRTOS

Poniżej znajduje się schemat ideowy działania systemu:



Rysunek 7 Schemat ideowy działania systemu

## **Wnioski**

Udało się stworzyć sterownik obiektu. Układ działa prawidłowo i stabilnie oraz dąży do wartości zadanej (o ile jest to możliwe w obiekcie). Raspberry okazało się być dobrym i łatwym w użyciu komputerem. Szereg dostępnej dokumentacji i dość duża społeczność w internecie pozwala łatwo rozwiązać większość problemów. UART jako interfejs komunikacyjny sprawdza się znakomicie, jest prosty w użyciu i nie sprawia problemów. Qt jak narzędzie do programowania interfejsu użytkownika również się sprawdziło.

Implementacja obiektu na systemie czasu rzeczywistego zapewniła deterministyczność, poszczególne operacje zostały od siebie odseparowane w osobnych taskach, między którymi komunikacja następowała między kolejkami.

## **Literatura**

Do realizacji wykorzystano liczne materiały dostępne w internecie, między innymi:

1. <https://forbot.pl/blog/kurs-raspberry-pi-od-podstaw-wstep-spis-tresci-id23139>
2. <https://mlodytechnik.pl/eksperymenty-i-zadania-szkolne/kursy/28558-praktyczny-kursraspberrypi>
3. <https://www.raspberrypi.org/blog/tag/tutorials/>

Ponadto korzystano z oficjalnej dokumentacji sprzętu użytego do wykonania zadania.