

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: Automatyka i Robotyka (AIR)
SPECJALNOŚĆ: Robotyka (ARR)

**PRACA DYPLOMOWA
INŻYNIERSKA**

Projekt systemu sensorycznego bazującego na
protokole MQTT

Design a sensor system based on MQTT protocol

AUTOR:
Marcin Bober

PROWADZĄCY PRACĘ:
Dr inż., Mateusz Cholewiński, K29W04D02

Spis treści

1	Wprowadzenie	1
1.1	Założenia	1
1.2	Opis komponentów	1
1.2.1	Aplikacja dostępowa	1
1.2.2	Urządzenie wykonawcze	2
1.2.3	Serwer	3
2	Schemat działania projektu	4
3	Szczegółowy opis sterownika	5
3.1	System operacyjny	5
3.2	System operacyjny	5
3.3	Schemat funkcjonowania	5
3.4	Regulator PID	5
3.5	Pomiar napięcia	5
3.6	Łączenie z brokerem	5
4	Szczegółowy opis brokera MQTT	6
4.1	Sposób działania	6
4.2	Implementacja i konfiguracja	6
5	Szczegółowy opis aplikacji dostępowej	7
5.1	Wykorzystana technologia	7
5.2	Implementacja komunikacji	7
5.2.1	Odbiór danych	7
5.2.2	Wysyłanie danych	7
5.3	Wykresy	7
6	Instrukcja obsługi	8
7	Podsumowane	9
	Literatura	9

Rozdział 1

Wprowadzenie

Celem pracy jest stworzenie systemu sensorycznego potrafiącego zarządzać silnikiem prądu stałego.

1.1 Założenia

System składa się z urządzenia wykonawczego serwera oraz aplikacji dostępowej. Komunikacja pomiędzy elementami systemu następuje poprzez serwer z wykorzystaniem protokołu MQTT.

Urządzenie wykonawcze składa się z mikrokontrolera, mostka H oraz silnika prądu stałego wyposażonego w enkoder kwadraturowy. Oprócz tego posiada układ pomiaru napięcia, a także wszystkie elementy niezbędne do prawidłowej pracy procesora.

Aplikacja dostępowa służy do łączenia się z serwerem poprzez protokół MQTT. Ma za zadanie stanowić prosty i przejrzysty interfejs do sterowania urządzeniem. Za jej pomocą mamy możliwość:

- zadawania prędkości obrotowej silnika,
- odczytu aktualnej prędkości silnika,
- zmiany nastaw regulatora PID,
- odczytu napięcia zasilania urządzenia,
- odczytu wypełnienia sygnału PWM.

Ostatnim elementem systemu jest serwer hostujący broker komunikatów w protokole MQTT. Ma on za zadanie być pośrednikiem w komunikacji między aplikacją dostępową, a urządzeniem wykonawczym.

1.2 Opis komponentów

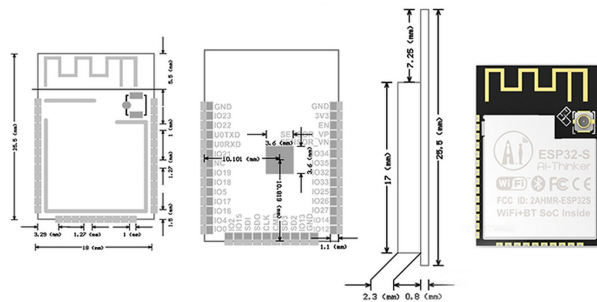
1.2.1 Aplikacja dostępowa

W celu komfortowej obsługi sterownika silnika, zdecydowałem się na stworzenie aplikacji dostępowej w języku C++, która będzie komunikowała się w protokole MQTT. W tym celu wyzystałem otwarto źródłową wersję narzędzia programistyczne jakim jest QT [1]. Moja decyzja była uwarunkowana moją szeroką znajomością tego narzędzia. Ponadto biblioteki QT posiadają dobrze zdefiniowane warstwy abstrakcji oraz bardzo wylewną dokumentację, co sprawia że tworzenie zaawansowanych, przenośnych aplikacji okienkowych jest wybitnie proste i przyjemne.

1.2.2 Urządzenie wykonawcze

Urządzenie końcowe zostało oparte na ogromnie popularnym SoC ESP32-S, chińskiej firmy Espressif Systems. Został on wybrany ze względu na wbudowany moduł WiFi oraz bardzo niską cenę wynoszącą poniżej 2\$ za sztukę. Ponadto oferuje on:

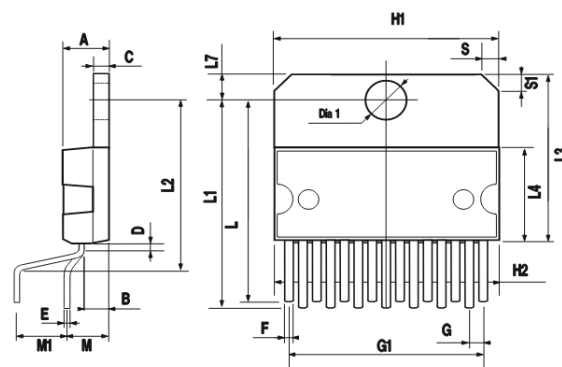
- dwurdzeniowy procesor o taktowaniu do 240MHz,
- 320 KiB RAM, 448 KiB ROM,
- Bluetooth w standardzie 4.2 oraz BLE,
- 34 wyprowadzenia GPIO.



Rysunek 1.1 Schemat wykorzystanej jednostki centralnej

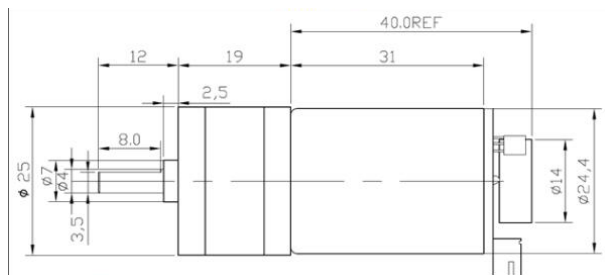
Można zauważyć że posiada on dość imponujące parametry jak na tak tani produkt. Pozwoli to na zadowalającą wydajność całego systemu bez obaw o niewystarczające zasoby lub kiepską optymalizację.

W roli mostka H zastosowałem bardzo popularny układ scalony L298N w obudowie Multi-watt15. Nie jest to najdoskonalszy wybór, ale zdecydowanie wystarczający dla tego zastosowania. Może się on poszczycić napięciem zasilania silników do 46V i prądem szczytowym na poziomie 2A. Ma on dwa kanały co sprawia że może jednocześnie sterować dwoma osobnymi silnikami, jednakże mój projekt zakłada wykorzystanie tylko jednego silnika, więc drugi kanał pozostanie nieaktywny. Układ tego typu znacząco redukuje poziom skomplikowania urządzenia ponieważ, aby go uruchomić wystarczą cztery diody i kondensator filtrujący napięcie. [5]



Rysunek 1.2 Schemat wykorzystanego mostka H

Silnik jest to bardzo konkurencyjnie wyceniany produkt firmy DFROBOT. Został on wybrany w głównej mierze ze względu na swoją niską cenę i parametry które w zupełności wystarczą do zrealizowania tego projektu. Jest on wyposażony w metalową przekładnię zapewniającą przełożenie napędu w proporcji 20:1. Ponadto, na końcu silnika jest fabrycznie zamontowany enkoder kwadraturowy, który produkuje 11 impulsów na każdy obrót wału głównego.



Rysunek 1.3 Schemat wykorzystanego silnika

1.2.3 Serwer

Ostatnim elementem projektu jest Broker MQTT. Jego zadaniem jest bycie łącznikiem pomiędzy urządzeniami. Pozwala on na łatwe subskrybowanie jak i publikowanie informacji. Ja zdecydowałem się na użycie w tej roli otwartoźródłowego brokera Mosquitto od fundacji Eclipse. Jest to jeden z popularniejszych programów tego typu, a zawdzięcza to małym wymaganiom sprzętowym, skalowalności oraz dostępności na wielu architekturach.

Rozdział 2

Schemat działania projektu

diagram wymiany danych, tematów mqtt

Rozdział 3

Szczegółowy opis sterownika

3.1 System operacyjny

jaki? po co?

3.2 System operacyjny

użyty framework

3.3 Schemat funkcjonowania

diagram wątków, kolejki

3.4 Regulator PID

implementacja, dobór nastaw

3.5 Pomiar napięcia

implementacja, regresja liniowa, błędy pomiarowe

3.6 Łączenie z brokerem

implementacja,

Rozdział 4

Szczegółowy opis brokera MQTT

4.1 Sposób działania

sposób działania,

4.2 Implementacja i konfiguracja

docker, konfiguracja

Rozdział 5

Szczegółowy opis aplikacji dostępowej

5.1 Wykorzystana technologia

Qt

5.2 Implementacja komunikacji

5.2.1 Odbiór danych

5.2.2 Wysyłanie danych

5.3 Wykresy

Rozdział 6

Instrukcja obsługi

Rozdział 7

Podsumowane

Literatura

- [1] Qt Development Frameworks: Dokumentacja biblioteki Qt
<https://doc.qt.io/qt-5/>
Dostęp 14.09.2021
- [2] STMicroelectronics: Dokumentacja L298H
https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf
Dostęp 14.09.2021
- [3] Espressif Systems: Dokumentacja ESP32 https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
Dostęp 14.09.2021
- [4] OASIS: Dokumentacja standardu MQTT 5.0
<https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf>
Dostęp 14.09.2021
- [5] V.P. Eloranta, J. Koskinen, M. Leppanen, V. Reijonen: Designing Distributed Control Systems: A Pattern Language Approach, Wiley, 2014.

Spis rysunków

1.1	Schemat wykorzystanej jednostki centralnej	2
1.2	Schemat wykorzystanego mostka H	2
1.3	Schemat wykorzystanego silnika	3