

Projektowanie Elektronicznych Układów Sterowania

Układ sterowania automatycznym barmanem

Mateusz Twardowski 244722

Cel projektu

Głównym celem projektu było rozwinięcie kompetencji inżynierskich w dziedzinie projektowania elektronicznych układów sterowania. Główne dziedziny rozwoju to projektowanie rzeczywistych układów hydraulicznych, projektowanie i lutowanie układów elektronicznych oraz ich programowanie.

W ramach zajęć został zaproponowany system automatycznego barmana, którego zadaniem byłoby mieszanie dwóch płynów ze sobą w zadanych proporcjach. Ważnym założeniem była pełna automatyzacja tego procesu, tak aby koktajl powstawał praktycznie bez ingerencji użytkownika. W ramach zajęć należało stworzyć prototyp takiego układu posiadający dwa obiegi na różne rodzaje cieczy.

Założenia projektowe

W celu ułatwienia pracy podzielono całość na trzy grupy projektowe. Pierwsza z nich dotyczyła doboru elementów wykonawczych (sensorów oraz aktuatorów), które będą odpowiedzialne za poszczególne ciecze. Druga część związana była z zaprojektowaniem oraz zlutowaniem układu do sterowania elementami wybranymi w części pierwszej. Ostatnim zadaniem było zaprogramowanie logiki do sterowania całym układem. Całość została zrealizowana na mikrokomputerze RaspberryPi 4B.

1. Dobór elementów wykonawczych

Zdecydowano się na następujący podział elementów wykonawczych – pompa odpowiada za pompowanie napojów bezalkoholowych, z kolei układ zawór – przepływomierz odpowiadają za napoje alkoholowe oraz syropy.

Podział ten wynika z różnic w parametrach fizycznych – napoje bezalkoholowe mają podobną gęstość, więc można założyć, że przy liniowej charakterystyce pompy nie będzie potrzebny przepływomierz w celu oceny ilości nalanej cieczy. Ponadto zwykle używane są większe ilości tych napojów w stosunku do alkoholu lub syropu, przez co błąd względny jest mniejszy.

Zawór sam w sobie nie był wystarczający, ze względu na małą dostępność elektrozaworów w rozsądnych cenach. W związku z tym został dobrany tańszy zawór, który gorzej spełnia założenia projektowe, natomiast przepływomierz dzięki odpowiedniemu przetworzeniu sygnałów jest w stanie dokładnie odmierzyć czas otwarcia zaworu. Ponadto zawór jest elementem bardzo wolnym (czas pełnego otwarcia to 5 sekund), w związku z czym potrzebne jest dokładniejsze sterowanie.



Rysunek 1 - zdjęcie zaworu U.S. Solid USS- SV00011 z zamontowanymi reduktorami hydraulicznymi do średnicy rurki 4 mm

Na podstawie powyższych założeń dobrano następujące elementy.

Zawór jest zaworem kulowym z siłownikiem sterowanym napięciem 24V DC, marki US Solid, model USS- SV00011.



Rysunek 2 - zdjęcie przepływomierza YF-S402

Przepływomierz został dobrany tak, aby ilość jego maksymalny przepływ na minutę był jak najmniejszy, tj. 6 L/min. Celem było uzyskanie jak największej dokładności, poprzez zwiększenie czasu przepływu cieczy. Zaworem wykorzystanym w projekcie jest zawór „Water Flow Sensor YF-S402”.

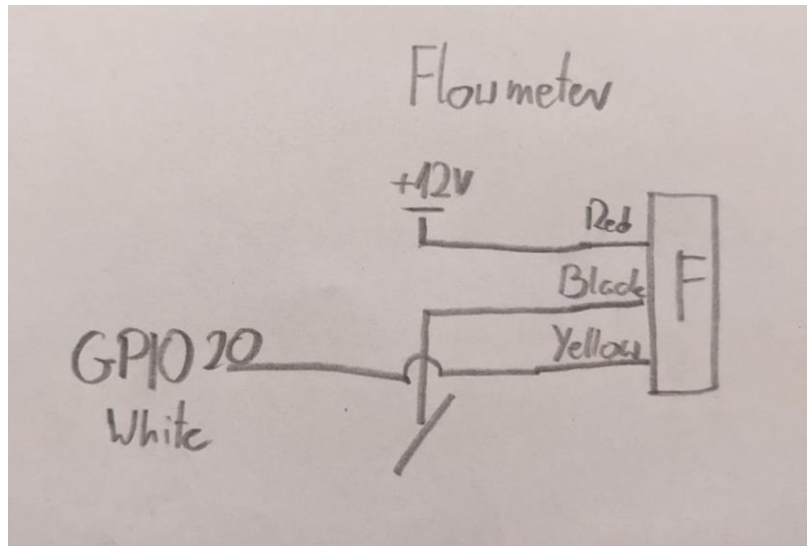


Rysunek 3 - zdjęcie pompy MOT-07206

Dobór pompy odbywał się w podobny sposób jak przepływomierza. Próbowano dobrać pompę o najmniejszym przepływie, tak aby czas nalewania był dłuższy, więc i większa dokładność pomiaru małych objętości. Niestety dostępność elementów o tak małym przepływie jest mała, a więc zdecydowano się na kompromis między ceną, a przepływem i ostatecznie maksymalna wartość to 110 L/h, czyli 1.83 L/min.

2. Projektowanie elektronicznego układu sterowania

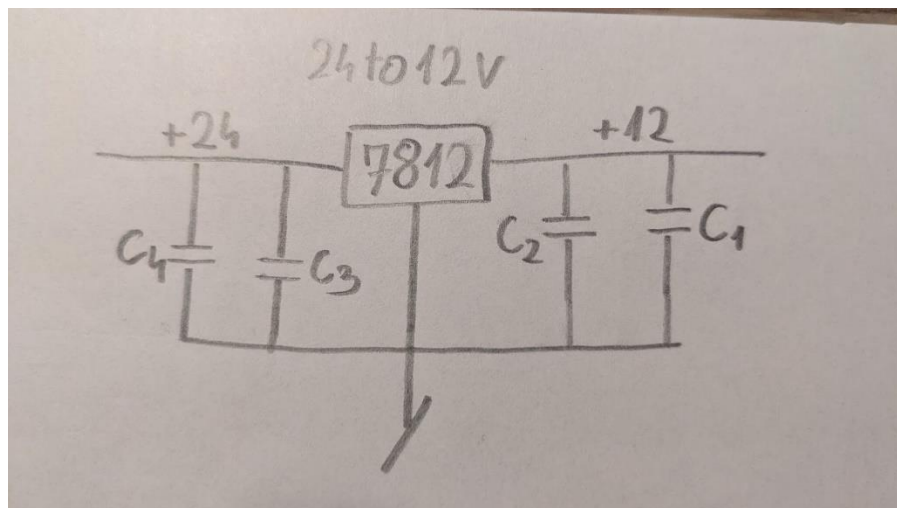
Zadaniem elektronicznego układu było przełożenie sygnałów cyfrowych z RaspberryPi na sygnały napięciowe sterujące dobranymi komponentami. O ile w przypadku przepływomierza nie było to skomplikowane, gdyż potrzebne sygnały to zasilanie +12 V DC, GND oraz sygnał wyjściowy do mikrokomputera (rysunek 4), to dla pozostałych dwóch komponentów niezbędny był dość rozbudowany układ sterowania.



Rysunek 4 - schemat podłączenia przepływomierza

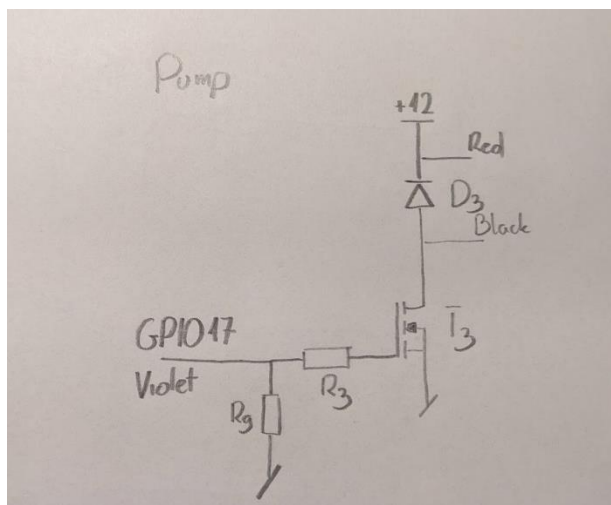
Najwyższym potrzebnym napięciem było + 24 V DC, w związku z tym dobrano zasilacz na to napięcie. Dobranym zasilaczem jest WX-DC2412. Bez niego niemożliwe byłoby sterowanie elementami wykonawczymi, gdyż RapsberyPi nie ma dostępnych napięć wyższych niż 5V.

Z kolei do sterowania pompą i zasilenia przepływomierza niezbędne było obniżenie tego napięcia do 12 V, w związku z czym skorzystano z układu 7812, w konfiguracji przedstawionej na rysunku 5.



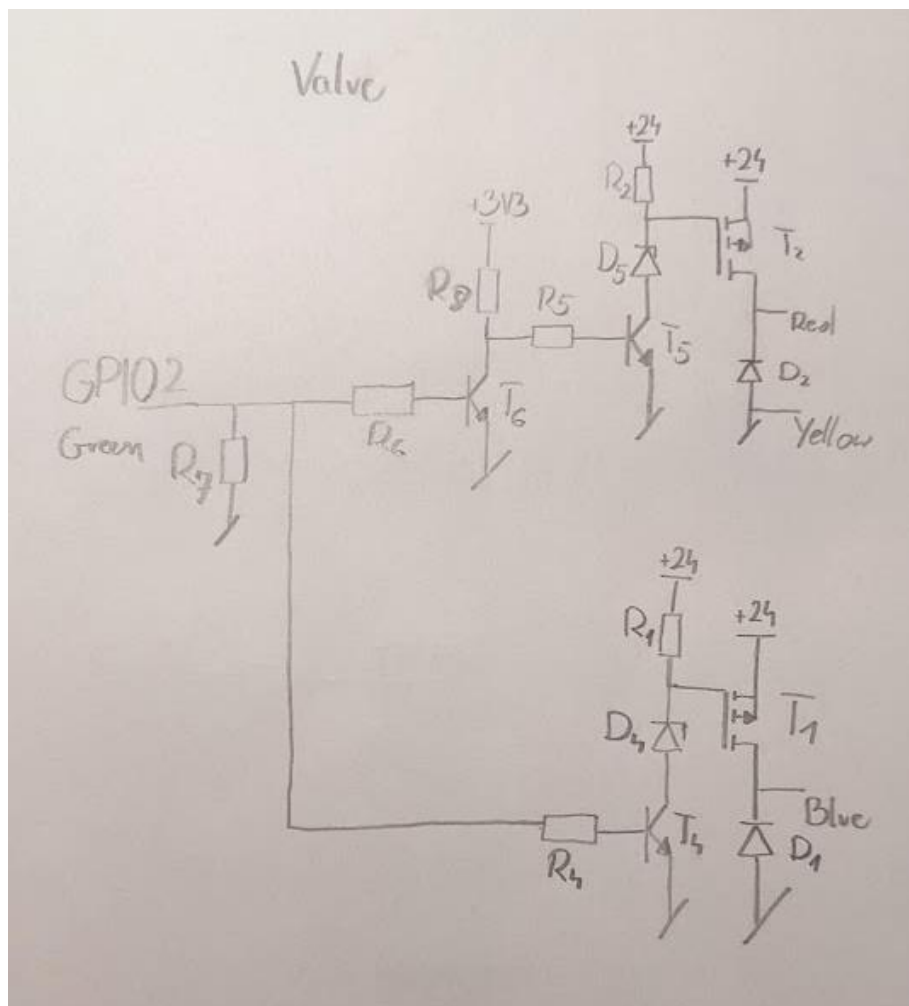
Rysunek 5 - schemat układu obniżającego napięcie 24VDC do 12 VDC

Następnie, korzystając z dostępnych sygnałów zaprojektowano układy do sterowania pompą (rys. 6) i zaworem (rys. 7).



Rysunek 6 - schemat elektryczny układu do sterowania pompą

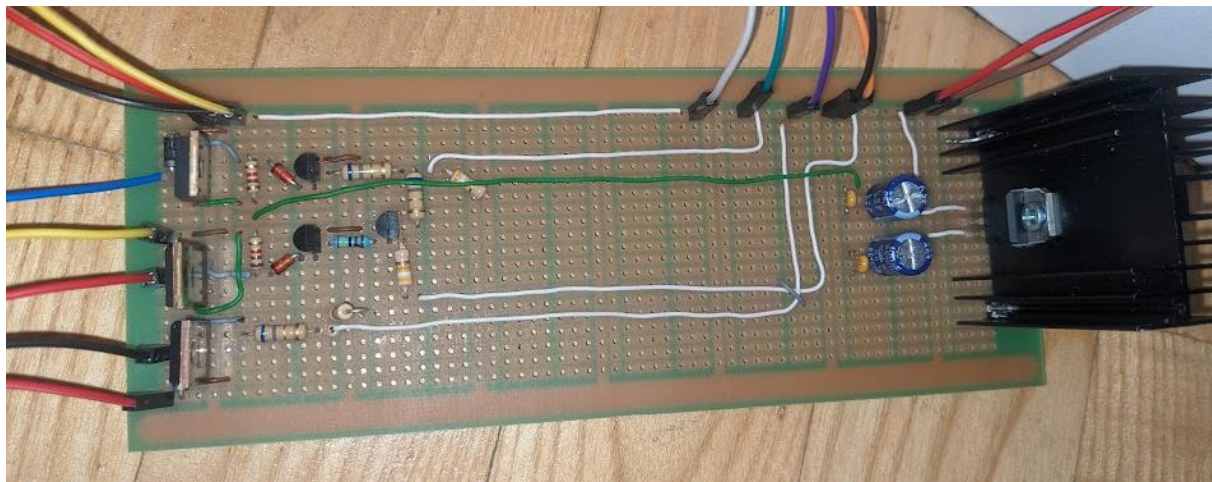
Do sterowania pompą zastosowano układ łącznika tranzystorowego.



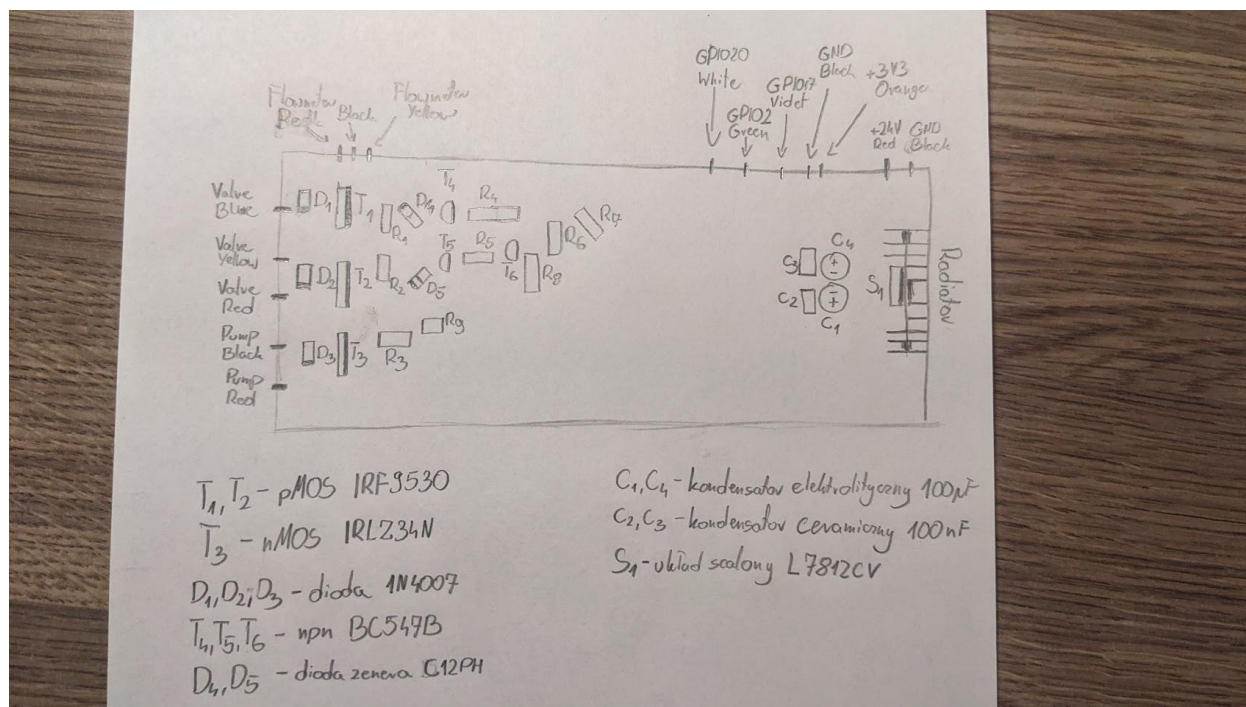
Rysunek 7 - schemat elektryczny układu do sterowania zaworem

Z kolei zawór posiada dwa przewody sygnałowe, czerwony i niebieski, z których jeden jest odpowiedzialny za zamykanie zaworu, a drugi za jego otwieranie. Przewód żółty jest na stałe podłączony do masy. Zadaniem tego układu sterowania jest przełączanie przewodu, do którego podłączone jest napięcie 24V. Układ górny i dolny są prawie identyczne, różnicą jest układ negacji tranzystorowej (T6, R8, R6). Tranzystory T1, T2 pełnią funkcję klucza tranzystorowego, z kolei T4 i T5 odpowiadają za wzmacnienie sygnału napięciowego z mikrokomputera. Są one niezbędne, gdyż napięcie na wyjściu GPIO w stanie wysokim jest równe 3.3V, co jest niewystarczające do wystawienia kluczy tranzystorowych.

Układ został stworzony na płytce uniwersalnej (rys. 8), a jego rzut z góry przedstawiony jest na rysunku 9.

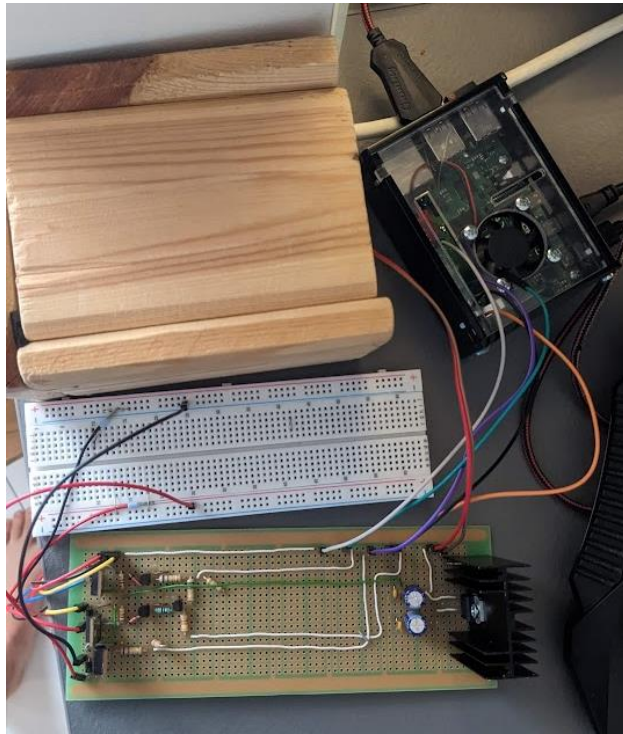


Rysunek 8 - zdjęcie ukończonego układu



Rysunek 9 - rzut z góry kompletnego układu sterowania wraz z opisem zastosowanych elementów

Następnie układ ten został podłączony do RaspberryPi oraz do podzespołów, którymi steruje.



Rysunek 10 - podłączenie układu do RaspberryPI



Rysunek 11 - prototyp automatycznego barmana

W pełni połączony układ (Rys. 12) był przygotowany do programowania.



Rysunek 12 - zdjęcie stanowiska testowego

3. Programowanie

Ze względu na znajomość języka Python i chęć jego doskonalenia, został on wybrany do napisania oprogramowania urządzenia. Cała dokumentacja oraz pliki programu dostępne są na GitHubie (<https://github.com/twardmat/brian/tree/PEUS>), w branchu PEUS.

Struktura programu zakłada, że każdy rzeczywisty element będzie osobną klasą, w ramach której będzie realizowane sterowanie. Umożliwia to skalowanie projektu w przyszłości, poprzez tworzenie nowych obiektów odpowiednich klas. Zostały stworzone trzy główne klasy, umieszczone w osobnych plikach – Valve() (zawór), Flowmeter() (przepływomierz) oraz Pump() (pompa).

Najprostszą jest klasa pompa – posiada ona zaledwie dwie metody dzięki którym zrealizowana jest cała funkcjonalność. Nad tą klasą została zbudowana inna – PumpService(), której głównym zadaniem jest stworzenie obiektu pompy, nowego wątku, na którym będzie realizowany jej program oraz uniemożliwienie ponownego uruchomienia pompy w momencie, gdy jeszcze działa. Posiada ona jedną metodę, która jest wywoływana z zewnątrz.

Kolejną klasą jest klasa Valve(), której zastosowanie jest bliźniacze do klasy Pump(). Czas otwierania zaworu jest podzielony na 2, aby cały proces otwierania i zamykania trwał tyle, ile zostało podane do funkcji w wyższych warstwach aplikacji.

Dla tej klasy nie ma napisanego serwisu, który zarządzałby wątkami oraz pilnował niewywoływania otwierania dopóki zawór się nie skończy, więc całość jest zrealizowana w jednej z metod klasy. Architektura całego układu przepływomierz – zawór zakłada, że przepływomierz jest jednostką nadrzędną, która kontroluje kiedy zawór ma być otwierany, a kiedy zamykany, a zatem do metody publicznej podawane są wartości logiczne (1 – otwieranie, 0 – zamykanie). Dodatkowo zawór został zaprogramowany tak, aby zamknął się automatycznie po 10 sekundach, nie zakłada się nalewania cieczy, których nalewanie zajęłoby dłużej, a jest to skuteczne zabezpieczenie przed przelaniem.

Najbardziej rozbudowanym elementem oprogramowania jest klasa Flowmeter(). Jest tak ze względu na sposób w jaki przetwarza ona sygnał przychodzący z czujnika. Na wyjściu przepływomierza pojawia się sygnał prostokątny (PWM o współczynniku wypełnienia 0.5) o zmiennej częstotliwości. Częstotliwość jest sygnałem, który zawiera informacje o prędkości przepływu zgodnie ze wzorem z dokumentacji $f = 73 \cdot Q$, gdzie f – częstotliwość, Q – aktualny przepływ w L/min. Po przeliczeniu tego na objętość oraz kalibracji układu uzyskano wzór rekurencyjny na całkowitą objętość cieczy, która przepłynęła
$$V_i = V_{i-1} + \frac{60}{90} \cdot T.$$

Do obliczania okresu T używa się częstotliwości f , która jest odczytywana poprzez wykrywanie zbocza sygnału. Sygnał jest filtrowany filtrem medianowym, wszystkie dane zapisywane są w 10-elementowym buforze kołowym.

Metoda publiczna po wywołaniu inicjalizuje nowy pomiar, a następnie otwiera zawór dopóki nie zostanie nalana połowa objętości zadanej. Po tym zdarzeniu zamyka zawór.

Wywołanie metod nalewających zarówno z zaworu jak i pompy odbywa się w funkcji main, którego wywołanie przyrządza koktajl.

Ponadto jako rozwinięcie projektu został stworzony web serwer operujący na framework'u Flask. W osobnym pliku serwer jest uruchamiany i możliwe jest wywołanie metody nalewającej napój przy użyciu pompy pod adresem IP RaspberryPi. Serwer uruchamiany jest z konsoli w języku Python.

Podsumowanie

Projekt został w pełni zrealizowany – założeniem było dobranie elementów wykonawczych i czujnika, zaprojektowanie i stworzenie układu sterowania, oraz zaprogramowanie mikrokomputera, tak aby dało się przygotować koktajl. Ponadto został napisany web serwer do sterowania zdalnego dla pompy. Barman jest w pełni funkcjonalny – podczas prezentacji nalał zadaną ilość cieczy do pojemnika i wykonał to zadanie dokładnie, bez względu na to czy wywoływany był z poziomu funkcji main, czy z innego komputera poprzez serwer.

Realizacja nie obyła się bez problemów na każdym z etapów – dobór tanich komponentów do takiego zadania był wyzwaniem, gdyż większość elementów dostępnych na rynku jest do zastosowań przemysłowych, gdzie jest albo duże ciśnienie, albo duże objętości cieczy. Z tego powodu zawór jest sterowany innym napięciem niż pompa, co jest znaczącym utrudnieniem.

W trakcie projektowania układu sterowania pompa się spaliła lub przyszła wadliwa, w związku z tym należało dodać dodatkowy układ obniżający napięcie dla nowej pompy (pierwotna miała być sterowana napięciem 5V, przepływomierz zgodnie z dokumentacją także mógłby być zasilony takim napięciem). Ponadto zawór okazał się sterowany poprzez przełączanie wysokiego napięcia między przewodami, a nie przełączanie masy, co początkowo nie zostało uwzględnione. W związku z tym konieczne było przeprojektowanie układu tak, aby sterowane były tranzystory pMOS zamiast nMOS.

Lutowanie też wymagało dwóch prototypów, gdyż pierwszy został zlutowany niepoprawnie. Dopiero drugi układ został poprawiony zgodnie z zasadami dobrego lutowania i nie występowały tzw. zimne luty.

Od strony programowej problemów było najmniej – program ewoluował od bardzo prostego skryptu sterującego stanem logicznym pinów do programu obiektowego rozbitego na pliki i obsługującego m. in. Framework Flask.

Postęp umiejętności uważam za znaczący od początku zajęć, projekt rozwinął mnie na wielu płaszczyznach i zdecydowanie pomógł ukształtować sposób myślenia o całym urządzeniu jako jednym, spójnym tworze, który musi być zgodny ze sobą na wszystkich poziomach.

Dalszy rozwój

Automatyczny barman będzie dalej rozwijany, natomiast prawdopodobnie ulegnie znaczącej zmianie pod wieloma względami. Po pierwsze elementy fizyczne zostaną zamówione z zagranicznych platform, co pozwoli obniżyć koszty przy jednoczesnej poprawie ich wspólnego działania. Ponadto zostanie rozbudowany, prawdopodobnie pojawią się trzy kolejne zawory i dodatkowa pompa. Jednakże zanim zostanie to zrobione zostanie przebadany pod kątem efektywności, gdyż zastosowane rozwiązanie nie musi być optymalne finansowo, ergonomicznie ani programowo. Jednakże warto zaznaczyć, że na ten moment rozwiązanie jest skalowalne – dorobienie kolejnych elementów nie wymaga przeprojektowywania całości.

W zależności od wybranych elementów układ elektroniczny może zostać zaprojektowany na płycie PCB oraz wydrukowany. Oszczędziłoby to dużo miejsca i byłoby rozwiązaniem docelowym.

Z kolei programowo jest dużo usprawnień do wprowadzania. Po pierwsze być może da się uprościć strukturę klasy Valve(). Po drugie, zostanie napisany serwis do obsługi układu przepływomierz – zawór, taki sam jak w przypadku pompy i zostanie udostępniony do serwera, tak aby można było go obsługiwać z sieci. Ponadto zostanie stworzona aplikacja mobilna do sterowania zdalnego maszyną.

Rozważana jest także zmiana języka programowania na C# w celu łatwiejszego łączenia aplikacji backendowej z bazą danych.

Ostatnią rzeczą, niemniej równie ważną jest budowa prawdziwej obudowy dla barmana – obecny prototyp pozostawia wiele do życzenia w kwestiach ergonomiczności i estetyki.

Podsumowując, pozostało wiele pracy do stworzenia rozwiązania, które mogłoby być wykorzystywane w życiu codziennym lub restauracjach, jednak duży krok został zrobiony, a pierwszy prototyp uważam za udany. Jak każdy większy projekt, niezbędne jest kilka iteracji, aby dojrzał on do właściwego kształtu.