POLITECHNIKA ŚLĄSKA

WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI

Projekt z Systemów Mikroprocesorowych

Inteligentny sterownik bramy

Autorzy:	Jakub Kaniowski, Szymon Ciemała
Rok / Semestr:	2020 / 2021 Semestr V
Grupa / Sekcja:	TI-4 Sekcja: 5
Kierujący pracą:	Dr inż. Piotr Krauze

Gliwice, październik - grudzień 2020

Spis treści

Spis rysunków

Spis tabel

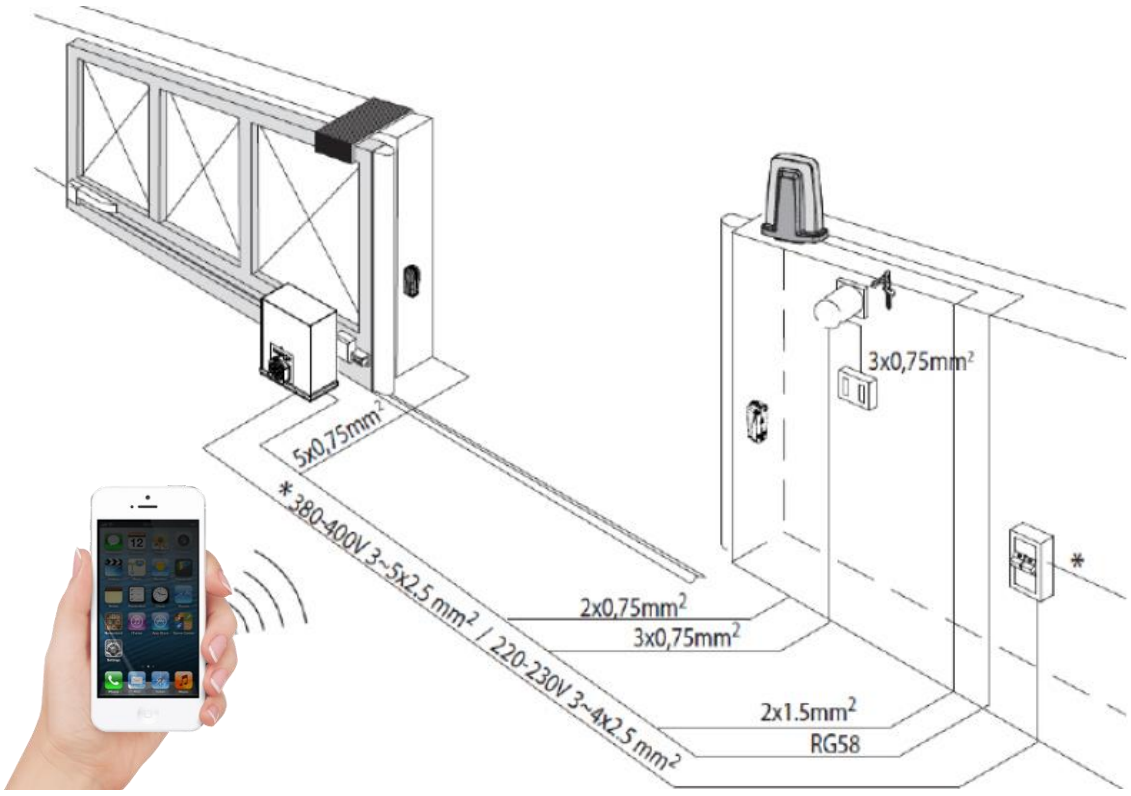
1. Wstęp

Inteligentny sterownik bramy to system mikroprocesorowy umożliwiający sterowanie i symulowanie rzeczywistego automatu bramowego. Przewagą tego rozwiązania, od konwencjonalnych układów sterowniczych jest łączność z siecią domową, dzięki czemu możemy sterować bramą np. za pomocą telefonu, komputera, czy za pomocą asystenta głosowego Google, Siri.

W zaprezentowanym projekcie, zostanie zasymulowany automat bramy przesuwnej. Będzie on zawierał wszystkie niezbędne czujniki i urządzenia wykonawcze, aby zbliżyć prezentowany układ mikroprocesorowy do układów rzeczywistych, stosowanych do użytku domowego.

W późniejszym czasie po ukończeniu projektu zaliczeniowego, część systemu mikroprocesorowego zostanie zaimplementowana do rzeczywistych sterowników bramy wjazdowej. Zostanie wykorzystana funkcja zdalnego otwierania bramy. Zaletą powyższego rozwiązania jest zwiększenie niezawodności dotyczącej przesyłania polecenia „otwarcia" bramy. Obecnie wykorzystywane urządzenia radiowe (pilot i odbiornik) w popularnych urządzeniach sterowniczych nie mają dużego zasięgu i często mają również problemy z transmisją przy złych warunkach atmosferycznych.

Prezentowany system mikroprocesorowy zwiększy również bezpieczeństwo użytkowników. Będzie można bezdotykowo (głosowo), i znacznie wcześniej przygotować bramę wjazdową do otwarcia, co w przypadku dużego natężenia ruchu drogowego wyeliminuje niebezpieczeństwo związane z zatrzymaniem pojazdu, i późniejszym włączeniem się do ruchu w celu wykonania odpowiedniego manewru, lub skorzystania z telefonu podczas jazdy.



Rysunek 1 Wizualizacja projektu [9]

1.1 Cel i zakres projektu

Niniejszy projekt, będzie obejmował w swoim zakresie dwa aspekty – aspekt sprzętowy oraz programowy.

W przypadku aspektu sprzętowego, głównym celem jest wykonanie fizycznego układu kontrolera, w oparciu o własny laminat PCB. Istotnym faktem będzie dobór odpowiednich elementów elektronicznych – w tym mikroprocesora oraz czujników. Z uwagi na projekt wykonywany w skali, znacznie mniejsze znaczenie ma kwestia urządzeń wykonawczych – np. silnik/siłownik bramy.

W przypadku aspektu programowego, wysoki priorytet jest ustanowiony na odpowiednie zaprogramowanie systemu mikroprocesorowego, w taki sposób aby spełniał wszystkie założenia projektu i był przy tym odporny na błędy i zakłócenia z zewnątrz. Niższym priorytetem zostanie obarczona kwestia komunikacji ze smartfonem, z uwagi na (najprawdopodobniej) urządzenie pośredniczące w tym procesie (serwer).

Zakres projektu będzie obejmował wykonanie odpowiedniej dokumentacji technicznej, rozwiązań programowych oraz wykonania fizycznego układu na dedykowanym laminacie PCB.
Dodatkowo, jednym z ważniejszych punktów jest budowa taniego urządzenia umożliwiającego sterowanie rzeczywistymi układami automatyki domowej, bądź rozszerzanie ich obecnej funkcjonalności. Ponadto, równie istotnym czynnikiem, będzie skonstruowanie układu, który poradzi sobie w sytuacjach wyjątkowych, i uchroni pojazd przed zniszczeniem, lub użytkownika przed utratą zdrowia. Całość obsługi urządzenia ma być podwyższeniem komfortu życia – interakcja z użytkownikiem ma być prosta i intuicyjna, a działanie urządzenia ma być niezawodne.

2. Harmonogram

2.1 Harmonogram zatwierdzony

- Zakup potrzebnych części i materiałów do budowy projektu
- Montaż części na płytce prototypowej
- Napisanie obsługi multipleksera
- Napisanie podstawowego kodu programu otwieranie bramy, odczyt czujników
- Projekt i wykonanie dedykowanej płytki PCB dla układu mikroprocesorowego.
- Projekt i wykonanie (wydrukowanie) części mechanizmu automatu bramy.
- Napisanie obsługi sieci LAN
- Napisanie obsługi komend głosowych
- Testowanie układu

2.2 Harmonogram wykonany

- Opracowanie założeń (19.10 - 26.10)
- Sporządzenie ogólnego projektu (26.10 - 02.11)
- Zapoznanie się z elementami elektronicznymi, czytnik RFID, multiplekser itd. Napisanie programowania obsługującego dany element (02.11- 09.11)
- Wykonanie i testy płytki (09.11 – 16.11)
- Napisanie końcowego oprogramowania wraz z implementacją komunikacji głosowej (16.11 – 23.11)

Wszystkie pozycje wyszczególnione w harmonogramie zatwierdzonym zostały skumulowane do działań zawartych w harmonogramie wykonanym. Gdyby podsumować, co zajęło najwięcej czasu, to zdecydowanie najdłuższym czasem realizacji charakteryzował się punkt dotyczący zapoznania się z elementami elektronicznymi. Należało przetestować każdy z elementów, i zweryfikować jego zasadność w projekcie. Najtrudniejszym fragmentem projektu, był bez wątpienia podpunkt dotyczący wykonania i testów płytki. To on zużył najwięcej zasobów zarówno psychicznych jak i fizycznych. Najkrótszym z kolei do wykonania, był podpunkt dotyczący opracowania założeń. Punkt ten zrealizowano poprzez krótką konferencję online i wymianę poglądów, pomysłów i sugestii związanych z urządzeniem. Podział pracy nastąpił dopiero po realizacji podpunktu dot. sporządzenia ogólnego projektu. Do tego etapu, projekt był wykonywany wspólnie. Realizacja tego projektu została rozdzielona po równo. Dotyczyło to zarówno testowania poszczególnych elementów, jak i części związanej z oprogramowaniem. W tym przypadku podział nastąpił względem właściwego oprogramowania (tzn. procesu otwarcia bramy) oraz całej komunikacji sieciowej wraz z asystentami głosowymi. Wyjątkiem w podziale ról stanowiło wykonanie laminatu PCB. Tym zajęła się jedna osoba, wykonując dwa laminaty (zarówno dla siebie, jak i współautora projektu).

3. Kosztorys

Koszty wykonania inteligentnego sterownika bramowego prezentują się następująco:

Tabela 1 Kosztorys projektu

Element dostarczany	Opis	Cena
ESP8266 (nodeMCU v3)	Mikrokontroler sterujący całym urządzeniem	8.98 PLN
Ekspander wyjść/wejść	Multiplekser CD4067BE wykorzystany do zwiększenia ilości wejść d/a	3.20 PLN
Dioda i odbiornik IR	Stosowana jako symulacja fotokomórek automatu bramowego.	2.45 PLN
Moduł RFID	Bezpieczna alternatywa otwierania bramy	7.90 PLN
Stabilizator LM7805	Stosowany do zasilenia układu mikroprocesorowego	0.50 PLN
Mostek H L293D	Stosowany jako sterownik makiety silnika bramy.	2.19 PLN
Przekaźnik 5V	Stosowany w celu obsługi rzeczywistego automatu bramowego	2.90 PLN
Wyłączniki krańcowe	Stosowane do określenia pozycji makiety bramy	3.48 PLN
Laminat PCB	Wykorzystany do wytworzenia płytki PCB	2.40 PLN
Inne elementy	Podstawowe elementy elektroniczne (kondensatory, rezystory, dip switch, złącza ARK)	2.50 PLN
Suma		**37.00 PLN**

Jak można zauważyć, koszt projektu nie jest wysoki, co docelowo może stanowić dobry argument w przypadku np. masowego wdrożenia. Oprócz oryginalności pomysłu i nowemu podejściu systemów automatyki bramowej, Inteligentny sterownik bramy jest urządzeniem nisko kosztowym, i uniwersalnym (dodatkowa możliwość pracy jako moduł rozszerzający).

Konkurencyjne rozwiązania należy liczyć w setkach złotych, co stawia potencjał tego systemu mikroprocesorowego na bardzo wysokim poziomie. Przykłady zostały omówione podczas prezentacji systemu mikroprocesorowego.

4. Urządzenie wraz z aplikacją

Projekt ma na celu stworzeniu inteligentnego sterownika bramy na miarę XXI wieku. Obsługa tego sterownika w dużej mierze bazuje na asystentach głosowych oraz module RFID.
Zastosowanie kontrolera RFID, pozwoli kontrolować dostęp określonych użytkowników. W przypadku zastosowań dla firm, można ustanowić odpowiedni dostęp dla posiadaczy kart, bądź breloków. Można założyć że będą to głównie osoby pieszo docierające do np. pracy.
Dla kierowców, chcących dostać się na posesję swojego domostwa bądź firmy, przeznaczony jest asystent głosowy, który nie rozprasza kierowcy podczas jazdy, bądź wykonywania manewru.

Warto również dodać, że sterownik jest w stanie analizować aktualne położenie bramy oraz przeciążenie napędu. Ponadto, jest odpowiedzialny za sterowanie napędem bramy, obsługę czujników i połączenia sieciowego, które umożliwi odbiór danych poleceń głosowych.

W tym rozdziale omówiono kwestię analizy problemu i jego kompleksowego rozwiązania i wykonania technicznego. Poniżej znajdują się główne założenia projektowe.

4.1 Założenia projektowe

4.1.1 Zasilanie

System mikroprocesorowy, będzie zawierał własny układ zasilania:

- Układ zasilania zostanie oparty na stabilizatorze. Do układu będzie doprowadzone napięcie stałe rzędu ok. 12V lub 24V.
- System mikroprocesorowy będzie posiadał zabezpieczenie chroniące przed uszkodzeniem, w postaci bezpiecznika.

4.1.2 Czujniki

System mikroprocesorowy zostanie wyposażony w czujniki:

- Skrajnego położenia bramy (otwarta/zamknięta)
- Prądu dostarczanego do napędu bramy
- Przerwania wiązki podczerwonej – umieszczone na słupkach ogrodzenia

Ponadto układ powinien zawierać elementy umożliwiające interakcję użytkownika z systemem, takie jak:

- Główny wyłącznik zasilania
- Możliwość sterowania manualnego (przycisk mechaniczny oraz kontroler RFID)
- Przełącznik konfiguracyjny określający prędkość otwierania bramy

4.1.3 Elementy wykonawcze

System mikroprocesorowy, powinien zawierać:

- Sterownik napędu wraz ze stosowną regulacją parametrów
- Moduł komunikacyjny WIFI (zintegrowane z mikroprocesorem)

4.1.4 Oprogramowanie

Oprogramowanie powinno wspierać:

- Obsługę WIFI, asystentów głosowych
- Obsługę wyjątków w postaci zacięcia bramy, nieznanego położenia.
- Obsługę wszystkich urządzeń składających się na system mikroprocesorowy
- Raportowanie o błędach.
- Możliwość korzystania z dwóch trybów: sterownika, modułu rozszerzającego

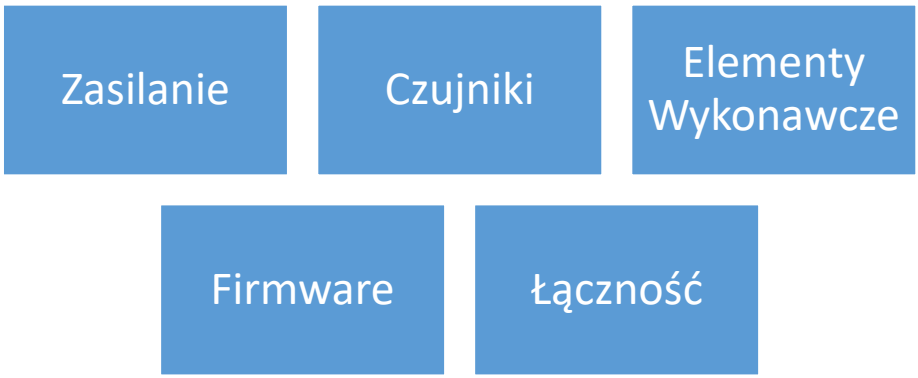
4.1.5 Łączność

Mikrokontroler powinien umożliwić sterowanie poprzez sieć lokalną, bądź zdalne usługi zarządzania urządzeniami Internetu Rzeczy (dalej zwane: „IoT").

- Moduł WIFI zaimplementowany bezpośrednio w mikrokontroler.
- Łączność bezprzewodowa, możliwość podłączenia do sieci LAN.
- Komunikacja z Blynk.

4.2 Określenie problemu

Istotnym założeniem tego projektu jest łączność sieciowa, i możliwość zdalnego zarządzania procesem otwierania bramy. Ponadto, aby system mógł spełniać swoje założenia powinien zostać wyposażony w czujniki, elementy wykonawcze i odpowiednie oprogramowanie. Oprócz wspomnianych elementów, istotnym będzie również układ zasilania – system mikroprocesorowy wraz z innymi elementami musi otrzymać odpowiedni zasilacz. Poniżej (rys. 2) przedstawiono pięć bloków, z którego będzie się składał Inteligentny Sterownik Bramy.



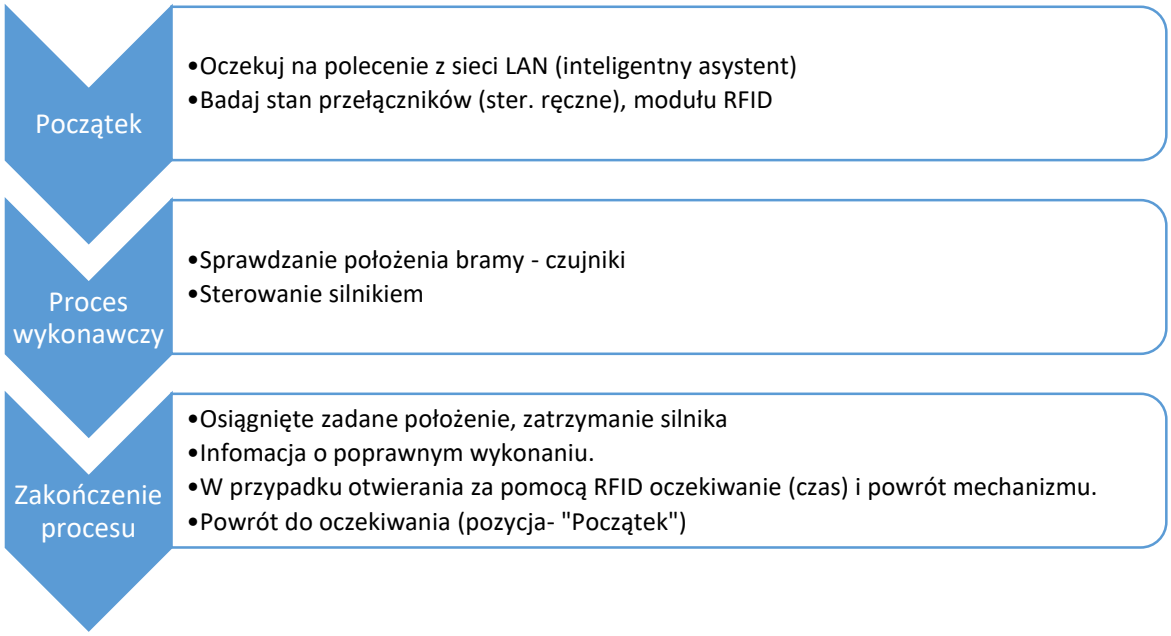
Rysunek 2 Blok systemu dla inteligentnego sterownika bramowego

Na podstawie obserwacji rzeczywistych układów automatyki bramowej, postanowiono wykonać krótki diagram przedstawiający sekwencje zdarzeń składające się na cały proces otwarcia lub zamknięcia bramy (rys. 3). Takie podejście znacznie usprawniło pracę nad oprogramowaniem oraz potwierdziło zasadność poszczególnych elementów w projekcie.

Czynność podzielono na 3 etapy. Etap początkowy w odróżnieniu od tradycyjnych urządzeń posiada alternatywne źródło, z którego będzie pochodzić informacja o rozpoczęciu działania automatu. (np. poprzez sterowanie serwisem web, lub inteligentnym asystentem głosowym). W konwencjonalnych produktach powinna znaleźć się tutaj transmisja radiowa (pilot). Etap wykonawczy przedstawia klasyczny przebieg czynności jakie są wykonywane podczas otwierania/zamykania się bramy. Mikroprocesor ma za zadanie przetwarzać informacje z wielu czujników w odpowiednio krótkim czasie, zbliżonym do przetwarzania w czasie rzeczywistym. W przypadku wykrycia przeszkód, lub pełnego przemieszczenia się bramy na jej skrajne punkty, powinna zostać podjęta decyzja o zatrzymaniu napędu.

Etap trzeci, jest odpowiedzialny za zakończenie trwającego procesu. Należy wysłać informację o poprawności wykonania (np. na serwer), bądź w przypadku użycia kart lub breloków RFID odczekać ustaloną ilość czasu, i powrócić do jednego ze skrajnych położeń bramy. Po zakończonym procesie, układ mikroprocesorowy, powinien powrócić w tryb czuwania - nasłuchiwania komunikatów z lokalnej

sieci, lub badania statusu czujników sterowania manualnego, czy też wspomnianego wcześniej modułu RFID.

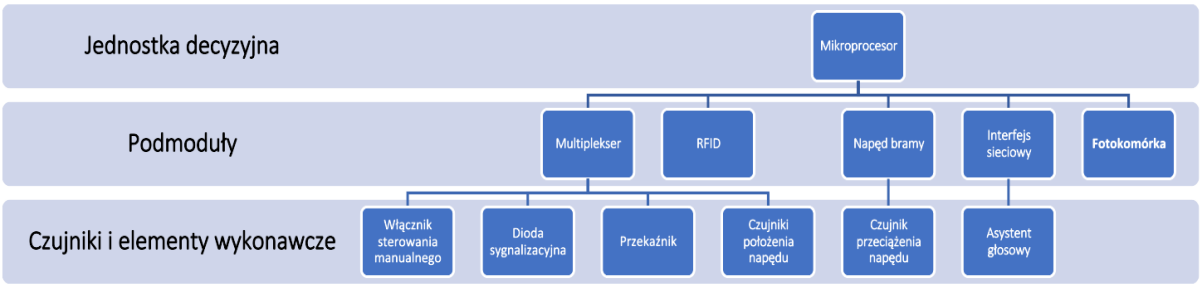


Rysunek 3 Lista czynności - otwarcie/zamknięcie bramy wjazdowej

4.3 Analiza rozwiązań, projekt urządzenia

Dokonując wyboru części składowych i rozwiązań dla systemu mikroprocesorowego, kierowano się wymaganiami postawionymi w punkcie 4.1 Założenia projektowe. Poniżej (rys.4) zaprezentowano diagram, ukazujący wypracowaną strukturę systemu mikroprocesorowego:

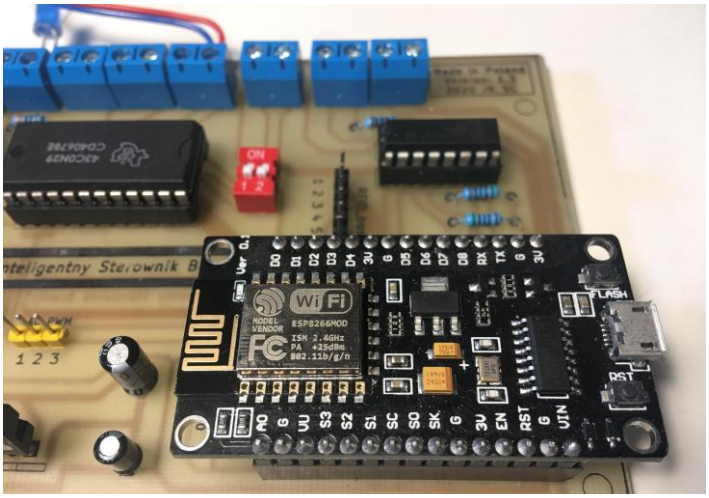
Powyższy diagram ma strukturę hierarchiczną. Oznacza to, że najistotniejszym elementem



Rysunek 4 Diagram strukturalny systemu mikroprocesorowego

sterownika jest mikroprocesor. Niżej w hierarchii są podmoduły rozszerzające możliwości samego mikrokontrolera i umożliwiające jakąkolwiek pracę układu. Najniżej w hierarchii znajdują się czujniki i elementy wykonawcze, co wcale nie umniejsza w „ważności" tych elementów. Ten diagram po prostu klasyfikuje stopień skomplikowania/pracy jaką te elementy muszą wykonać. Dla przykładu: procesor wykonuje wiele operacji na sekundę, gdy np. przekaźnik może być tylko zwarty, bądź rozwarty. W dalszej części podrozdziału omówiona zostanie szczegółowa analiza i implementacja rozwiązań wraz z fragmentami schematu elektrycznego. Pełen schemat elektryczny umieszczono w podrozdziale 4.4.3.

Wybrany mikroprocesor, to popularny układ firmy **Espressif ESP8266**, na platformie uruchomieniowej nodeMCU. Jest to urządzenie z wbudowaną kartą sieciową WIFI, co spełnia nasze główne wymaganie dot. dostępu do sieci LAN. Wspomniany układ, to 32-bitowy procesor, taktowany z częstotliwością **80MHz**, posiadający **64kB RAM** oraz **4MB pamięci flash**, co w zupełności wystarcza do skonstruowania tego urządzenia. Oprócz ogromnej zalety jakim jest **wbudowany moduł WIFI**, platforma nodeMCU pozwala na stosowanie napięcia zasilania nawet do +10V DC. W tym systemie mikroprocesorowym zasilanie tego mikrokontrolera będzie wynosiło +5V DC, należy jednak pamiętać, iż cała logika będzie odbywała się na napięciu +3.3V. Dodatkowo zaimplementowano wbudowany programator, który pozwala na zaprogramowanie układu po podpięciu do portu USB w komputerze.



Rysunek 5 Mikrokontroler ESP8266 (nodeMCU v.1.0)

Tabela 2 Dane techniczne mikrokontrolera [1]

Specyfikacja nodeMCU (v.1.0)	
Procesor	Espressif ESP8266 32-bit 80MHz
Pamięć RAM	64kB (SRAM)
Pamięć Flash	4MB
Programator	CH340G (USB – Serial)
Napięcie operacyjne	3.3V
Napięcie zasilania	4.5V – 10V (DC)
Wyprowadzenia	11 cyfrowych (We/Wy), 1 ADC (max. 3.3V!)
Interfejsy/Magistrale	UART, I^2C, SPI, WIFI (802.11 b/g/n)
Zakres temperatur	-40°C → 120°C

Zastosowany mikroprocesor jest wydajnym urządzeniem, jednak sam mikrokontroler nie będzie w stanie spełnić postawionych wymagań. W celu rozszerzenia jego funkcjonalności zastosowano kilka podmodułów takich jak multiplekser, moduł RFID.

Po analizie wszystkich elementów, które powinny zostać podłączone do układu, zdecydowano się na multiplekser, który rozszerzy skromne i niewystarczające 11 wejść/wyjść cyfrowych.

W tym celu zaimplementowano układ CMOS CD4067BE, który posiada 16 wejść/wyjść rozszerzających. Dokładna realizacja zostanie omówiona (jako część sprzętowa) w punkcie 4.3.2 Multiplekser oraz 4.4.4 Właściwe oprogramowanie (jako część programowa). Układ ten skonstruowano w oparciu o obudowę DIP-24. W obecnych czasach taki multiplekser można znaleźć głównie w formie SMD, jednak z uwagi na dużą trudność w wytworzeniu odpowiedniej płytki PCB w standardzie SMD, i dużej ilości tych układów w domowym zapleczu - zdecydowano się na implementacje tego właśnie multipleksera. Co istotne, multiplekser operuje w logice 3.3V, a więc zgodnej z zastosowanym mikrokontrolerem. Na rysunku 6 znajduje się krótka nota katalogowa multipleksera, zawierająca napięcie zasilania

Characteristic	Min.	Max.	Units
Supply-Voltage Range (T_A=Full Package-Temp. Range)	3	18	V
Multiplexer Switch Input Current Capability	–	25	mA
Output Load Resistance	100	–	Ω

Rysunek 6 Notka katalogowa multipleksera CD4067BE. [6]

Silnik elektryczny napędu bramy jest sterowany za pomocą sygnału PWM. W tym procesie pośredniczy mostek H (układ scalony L293d). Mostek H posiada wydajność prądową do 1.2A, co w zupełności wystarcza w przypadku zastosowania 5V silnika DC pochodzącego z napędu DVD.

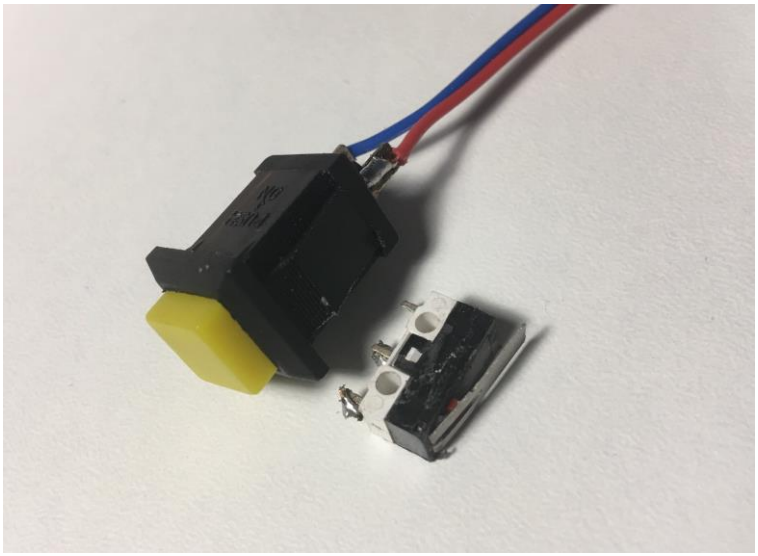
W układzie występuje również czujnik przerwania wiązki podczerwieni. W tym projekcie zrealizowano to za pomocą diody nadawczej IR oraz odbiornika IR VS1838B. Elementy te są zwrócone do siebie, w jednej osi. W momencie przerwania wiązki, napęd powinien się zatrzymać. Podczas procesu otwierania/zamykania bramy, wysyłana jest wiązka podczerwona z nadajnika poczerwieni (zwykłej diody IR). Skierowana na odbiornik wiązka jest interpretowana jako stan domyślny. Przerwanie wiązki, czyli stan niski powoduje przerwanie procesu. Dane techniczne tego elementu przedstawiono w tabeli 3:

Tabela 3 Dane techniczne odbiornika IR [2]

Specyfikacja VS1838B	
Napięcie zasilania	2.7V – 5.5V (DC)
Maks. zasięg / kąt	23 metry / 35°
Napięcie operacyjne	0.4 V – niski stan, 5V – stan wysoki.
Częstotliwość pracy	38kHz

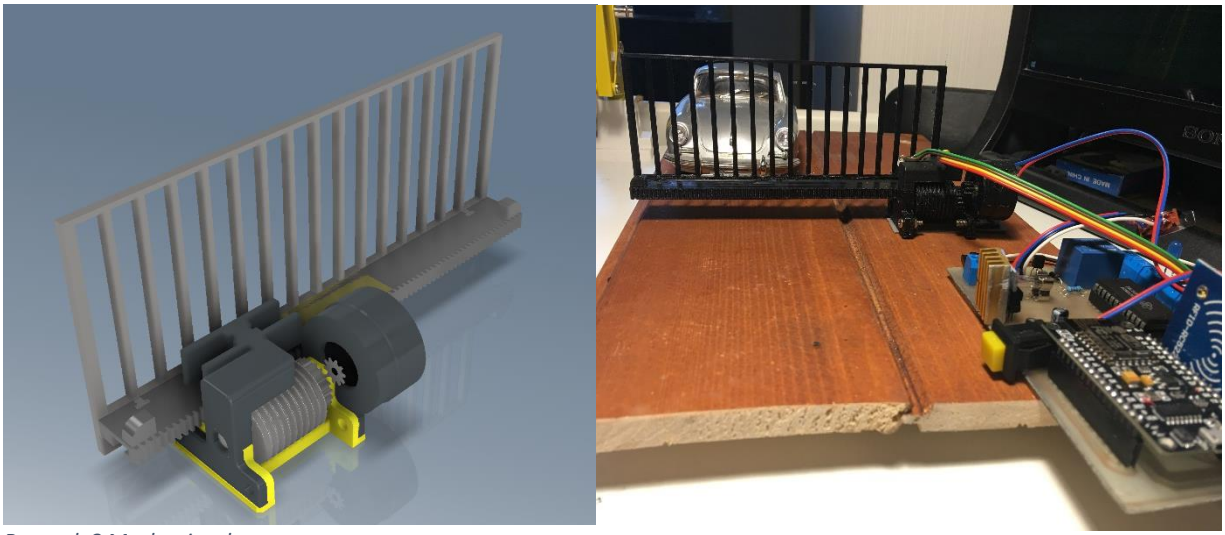
Czujnik przeciążenia napędu bazuje na dzielniku napięcia, który jest podłączony do wyjścia analogowego mikrokontrolera.

Czujniki położenia napędu, to dwa zwykłe wyłączniki krańcowe. Są jednak w wersji mini, gdyż ich rozmiar to zaledwie 12x5,8x6,6mm. Napięcie maksymalne takiego wyłącznika to 125V, a maksymalny prąd to 1A. Tym samym, nie ma żadnych przeciwskazań do stosowania tych elementów w projekcie. Czujniki te, są aktywowane fizycznymi „zgrubieniami" mechanizmu bramy (elementy wydrukowane na drukarce 3D). Mikrokontroler odczytuje odpowiednią pozycje bramy, która jest przypisana do danego wyłącznika (czyli otwarte bądź zamknięte). Na rys. 7 przedstawiono (po lewej) przycisk sterowania ręcznego oraz wyłącznik krańcowy (po prawej).



Rysunek 7 Wyłączniki mechaniczne

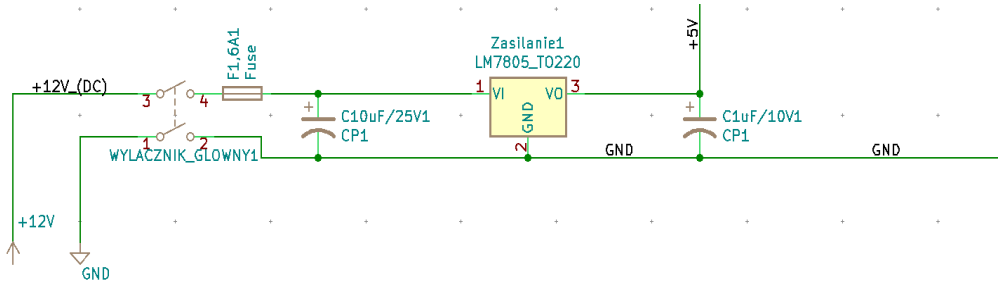
Aby cały system mikroprocesorowy można było zwizualizować, i przetestować jego działanie na obiekcie fizycznym, postanowiono wykonać mechanizm bramy. Posłużono się do tego programem Autodesk Inventor. Po sporządzeniu projektu, wydrukowano go na drukarce 3D. Efekty pracy można zobaczyć na rysunku 8.



Rysunek 8 Mechanizm bramy

4.3.1 Układ zasilania

Układ zasilania został oparty na stabilizatorze **LM7805**, który dostarcza stałe napięcie 5V, i maksymalny prąd 1,5A. Jest to bardzo popularny model stabilizatora, stąd jego niska cena oraz prostota zastosowania. Głównym powodem jego wykorzystania w tym projekcie, nie jest cena, a szeroka tolerancja napięć wejściowych układu: zakres od ~7V – 35V (DC). Dodatkowo w celu eliminacji zakłóceń zastosowano dwa kondensatory filtrujące. Główne zabezpieczenie stanowi bezpiecznik rurkowy 1,6A, który chroni cały układ przed uszkodzeniem.



Rysunek 9 Fragment schematu - układ zasilania

Sterownik bramy można wykorzystać na dwa sposoby – jako niezależne urządzenie sterujące – tutaj sugerowany dedykowany zasilacz, bądź jako moduł rozszerzeń do rzeczywistych automatów bramowych – ta druga możliwość wymaga dołączenia układu do istniejącego już źródła zasilania. Stąd tak pożądany szeroki zakres napięć wejściowych. Większość układów automatyki bramowej pracuje na zasilaniu +12V DC, lub +24V DC. Należy tylko pamiętać, iż **w przypadku większych napięć (więcej niż +9V), wymagane jest zastosowanie radiatora**, który odprowadzi ciepło ze stabilizatora, i uchroni go przed przegrzaniem. Warto w tym miejscu dodać, że układ ten posiada wbudowane zabezpieczenie przed przegrzaniem, a obudowa TO220 umożliwia łatwy montaż radiatora. Parametry techniczne tego stabilizatora zostały zaprezentowane w poniższej nocie katalogowej (rys. 10)

Table 10. Electrical characteristics of L7805C

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
V_O	Output voltage	T_J = 25°C	4.8	5	5.2	V
V_O	Output voltage	I_O = 5 mA to 1 A, V_I = 7 to 18 V	4.75	5	5.25	V
V_O	Output voltage	I_O = 1 A, V_I = 18 to 20V, T_J = 25°C	4.75	5	5.25	V
ΔV_O(1)	Line regulation	V_I = 7 to 25 V, T_J = 25°C		3	100	mV
		V_I = 8 to 12 V, T_J = 25°C		1	50	
ΔV_O(1)	Load regulation	I_O = 5 mA to 1.5 A, T_J = 25°C			100	mV
		I_O = 250 to 750 mA, T_J = 25°C			50	
I_d	Quiescent current	T_J = 25°C			8	mA
ΔI_d	Quiescent current change	I_O = 5 mA to 1 A			0.5	mA
		V_I = 7 to 23 V			0.8	
$\Delta V_O/\Delta T$	Output voltage drift	I_O = 5 mA		-1.1		mV/°C
eN	Output noise voltage	B = 10 Hz to 100 kHz, T_J = 25°C		40		µV/V_O
SVR	Supply voltage rejection	V_I = 8 to 18 V, f = 120 Hz	62			dB
V_d	Dropout voltage	I_O = 1 A, T_J = 25°C		2		V
R_O	Output resistance	f = 1 kHz		17		mΩ
I_{sc}	Short circuit current	V_I = 35 V, T_J = 25°C		0.75		A
I_{scp}	Short circuit peak current	T_J = 25°C		2.2		A

1. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty cycle is used.

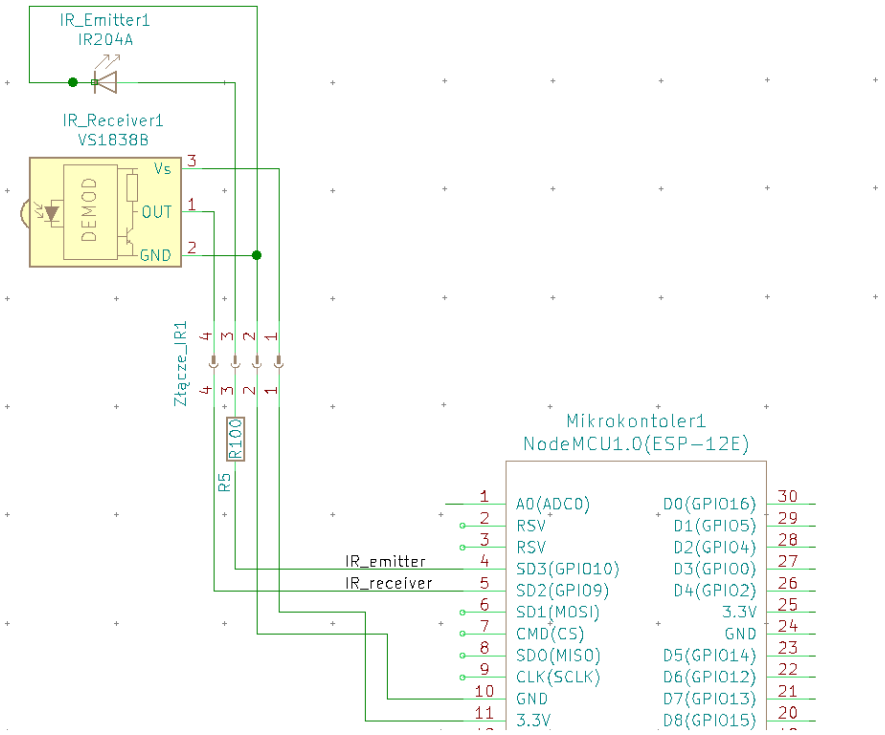
Rysunek 10 Nota katalogowa stabilizatora LM7805. [3]

4.3.2 Układy czujników i przełączników

W tym miejscu zostanie omówione zastosowanie czujnika podczerwieni oraz czujników i przełączników mechanicznych. Należy wspomnieć, że czujnik podczerwieni (nadajnik i odbiornik) zostały podłączone do bezpośrednich wyprowadzeń mikrokontrolera (GPIO9, 10). Wszelkie przełączniki mechaniczne zostały podłączone do multipleksera celem oszczędności wyjść cyfrowych. Ponadto dostęp do takich czujników może być realizowany właśnie w taki sposób, z uwagi na idealną równowagę pomiędzy elementami, które potrzebują stałego i bezpośredniego dostępu do mikrokontrolera. W tym przypadku czujnik/przełącznik mechaniczny, który technicznie rzecz biorąc jest zwykłym wyłącznikiem (krańcowym), nie przetwarza danych i nie jest układem cyfrowym. Odczyt poszczególnych stanów (wysoki/niski) odbywa się w narzuconej przez oprogramowanie kolejności.

Czujnik podczerwieni

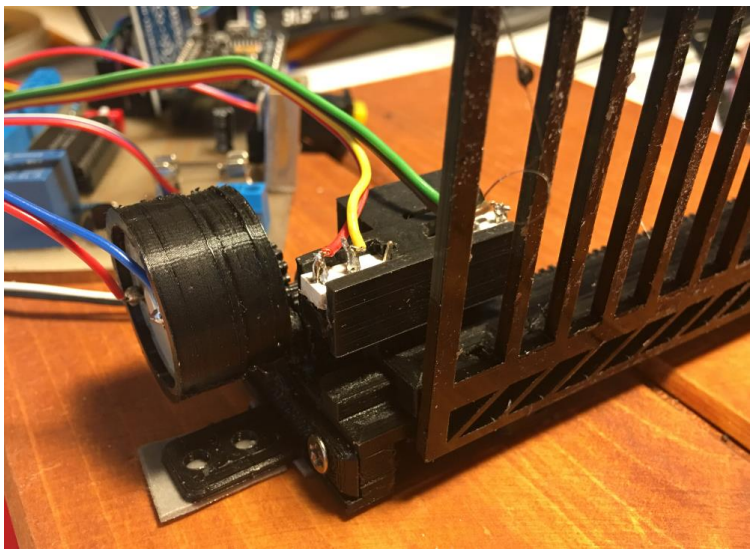
W tym wypadku zasymulowano taki czujnik układem diody nadawczej IR oraz odbiornika podczerwieni VS1838B. Odbiornik zwraca napięcie na wyjściu od 0.4V do 5V. W momencie rejestracji zmiany natężenia światła podczerwonego, napięcie na wyjściu czujnika gwałtownie spada. W ten sposób otrzymywany jest stan niski, który jest wykrywany przez mikrokontroler. Na podstawie tego sygnału podejmowana jest decyzja o zatrzymaniu napędu celem uniknięcia kolizji. Rozwiązanie w postaci schematu elektrycznego przedstawiono na rys. 11. Rezystor 100 Ohm zabezpiecza diodę nadawczą oraz definiuje długość emitowanej fali podczerwonej (tutaj 940nm).



Rysunek 11 Fragment schematu - układ czujnika podczerwieni

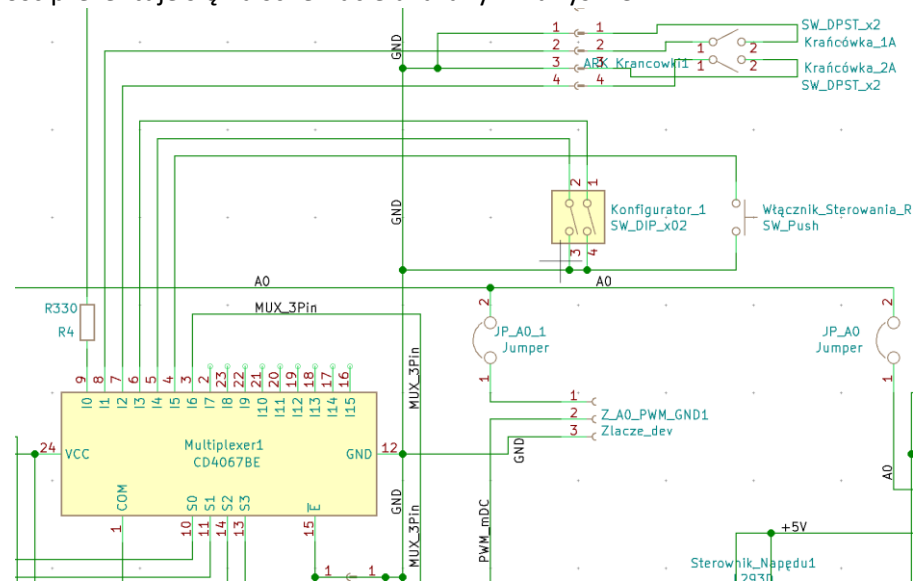
Czujniki mechaniczne

System mikroprocesorowy, jest wyposażony w czujniki mechaniczne, służące do określania pozycji skrzydła bramy. Składa się on z dwóch wyłączników krańcowych, które mają odpowiednie znaczenie w układzie - jeden wyłącznik odpowiada za pozycję „otwartą”, drugi wyłącznik za „zamkniętą”. Całość jest podłączona do multipleksera, z którego wysterowany jest dany kanał odpowiadający podłączonemu wyłącznikowi.



Rysunek 12 Czujniki położenia napędu

Na rysunku 12 zaprezentowano zamontowane w napędzie wyłączniki krańcowe. Wyłącznik po lewej stronie odpowiada za pozycję „zamkniętą”, drugi (trochę mniej widoczny) odpowiada za pozycje otwartą. Całość prezentuje się na schemacie ukazanym na rys. 13:

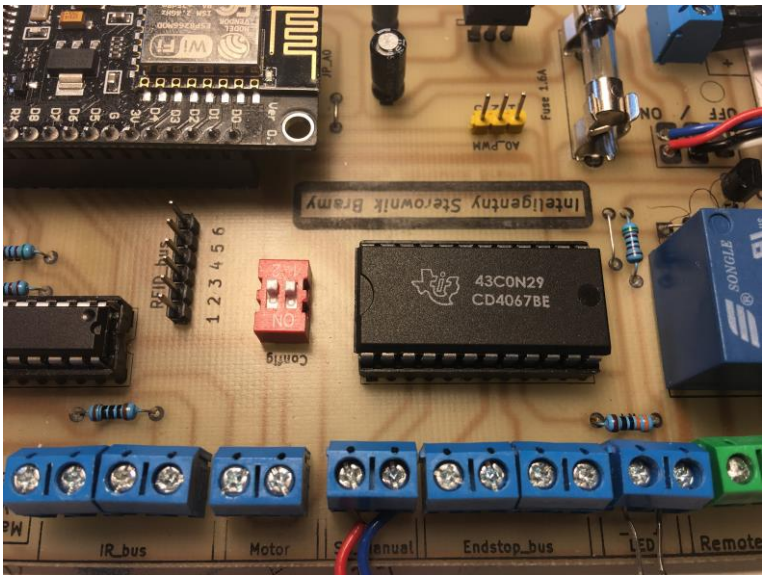


Rysunek 13 Fragment schematu – czujniki mechaniczne

Jak można zauważyć, oba wyłączniki krańcowe są podłączone do wyjść multipleksera. Ponieważ w tym miejscu prezentowane są czujniki mechaniczne, warto zwrócić uwagę na przycisk sterowania manualnego (oznaczony na rys. 13, jako „Wyłącznik_Sterowania_R[..]”) oraz dodatkowy przełącznik konfiguracyjny (widoczny na rys.13 pod nazwą „Konfigurator_1” i na rys.14, jako czerwony element na płytce PCB.)

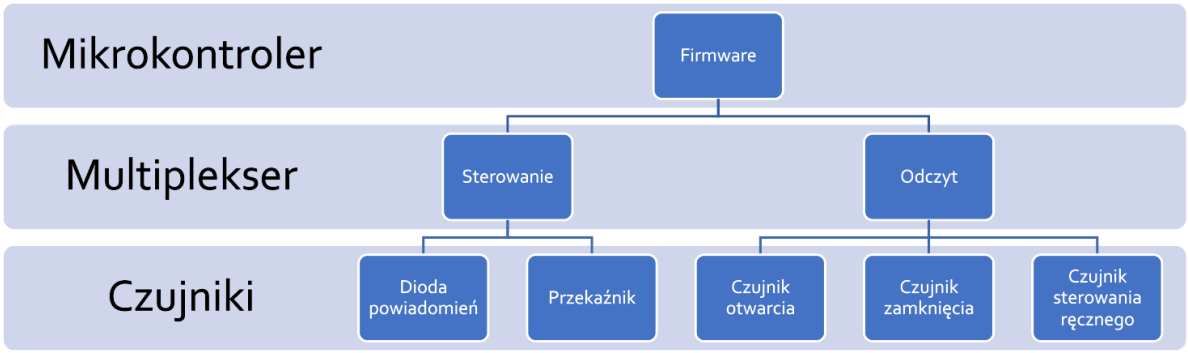
Multiplekser

W systemie mikroprocesorowym zastosowano multiplekser CD4067BE. Jak wspomniano powyżej, liczba wyjść/wejść w mikrokontrolerze była niewystarczająca. Postanowiono więc rozszerzyć standardowe wyjścia za pomocą 16-bitowego multipleksera.



Rysunek 14 Multiplekser CD4067BE - Inteligentny Sterownik Bramy

Warto poruszyć ten temat właśnie w tym podrozdziale, z uwagi na elementy podłączone do tego układu scalonego. Większość stanowią właśnie przełączniki, w tym czujniki położenia napędu, i włącznik sterowania ręcznego. Poniżej diagram wyjaśniający połączenia:



Rysunek 15 Diagram przedstawiający działanie i funkcjonalność multipleksera

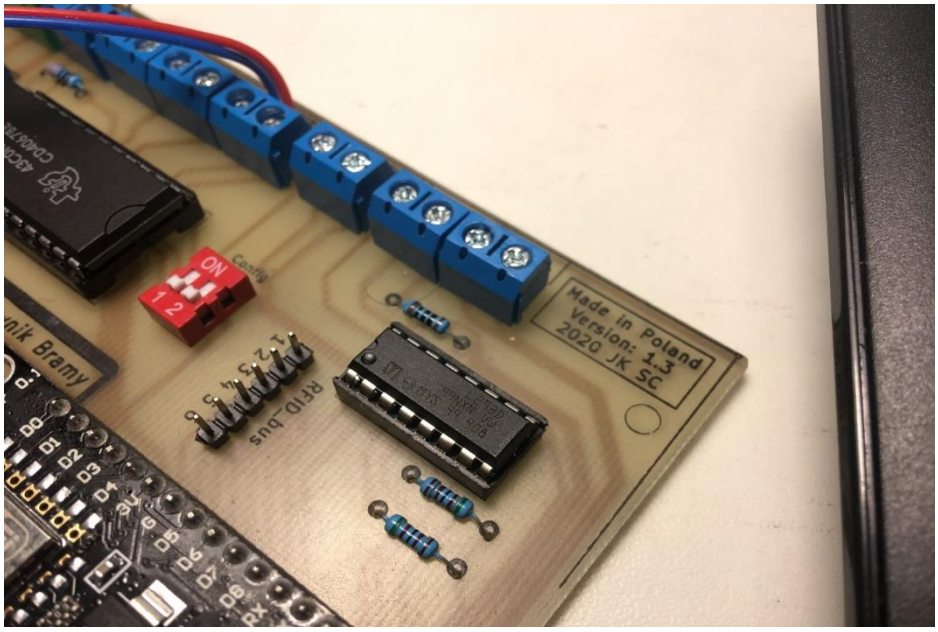
Nad poprawnym dostępem do urządzeń czuwa firmware mikrokontrolera, a sam multiplekser jest elementem wykonującym polecenia w dwóch kierunkach – można zarówno sterować elementami (takimi jak dioda powiadomień, przekaźnik), jak również odczytywać stany poszczególnych czujników, które zostały opisane wcześniej. Potencjalną wadą takiego rozwiązania jest brak dostępności do wszystkich elementów w jednym czasie. Działanie multipleksera polega na przekierowaniu głównego wyjścia/wejścia, na rozszerzane wyprowadzenia, w zależności od bitów jakie podamy na wejścia konfiguracyjne multipleksera. Nie możliwy jest zatem dostęp do każdego wyjścia jednocześnie. Nie jest to jednak uciążliwe, z uwagi na dużą częstotliwość wykonywania odczytu, i szybki czas przełączania multipleksera na poszczególne wejścia układu. Umożliwia to niemalże bezzwłoczną reakcję urządzenia. Dodatkowo, w niemalże tym samym czasie jest możliwy odczyt wszystkich czujników mechanicznych przy jednoczesnym cyklicznym załączaniu diody powiadomień (lampki kontrolnej).

Przekaźnik, który jest widoczny na diagramie (rys. 15) jest zaimplementowany w układzie z uwagi na podwójną funkcjonalność systemu mikroprocesorowego - w tym przypadku - umożliwia korzystanie jako moduł rozszerzający dla rzeczywistych układów sterowniczych automatyki bramowej.

4.3.3 Elementy wykonawcze

Napęd bramy

Sterowanie napędu bramy powierzono **układowi scalonemu L293d**. Układ ten posiada dwa mostki H (w tym projekcie wykorzystywany jest tylko jeden), które są sterowane za pomocą mikroprocesora ESP8266.



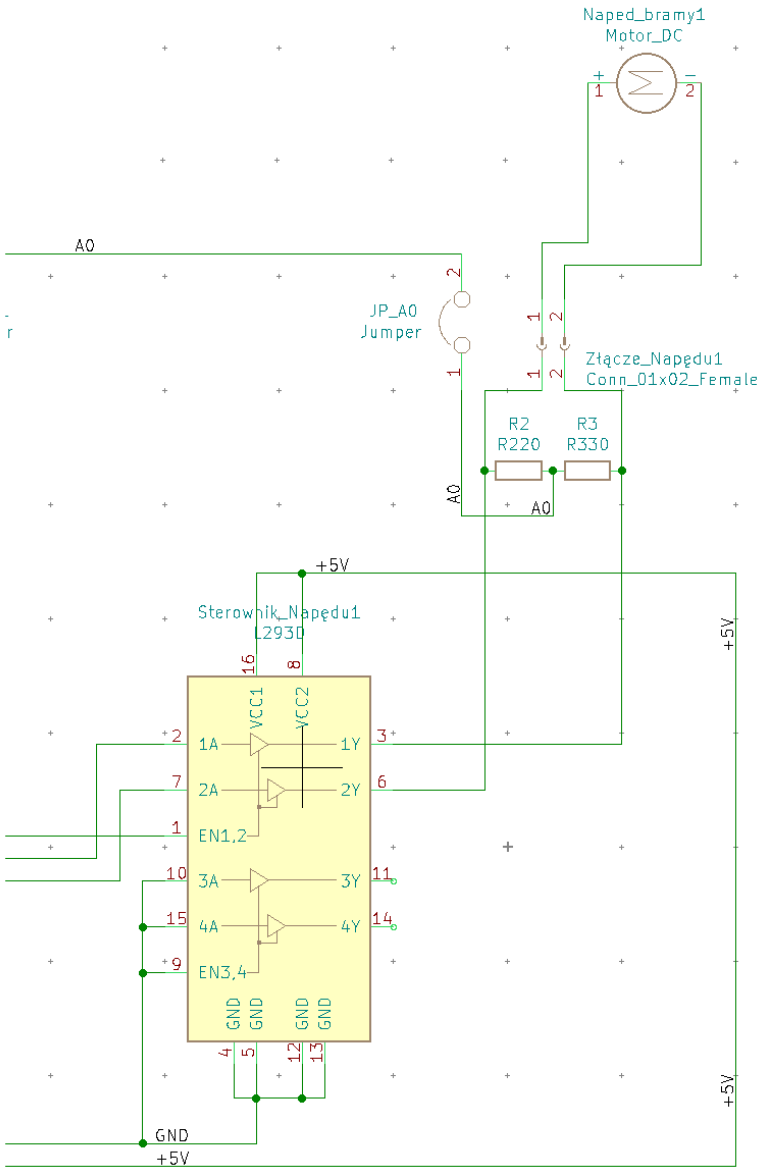
Rysunek 16 Sterownik silnika -układ L293d widok na płytce PCB

Sterowanie odbywa się poprzez sygnał PWM wysterowany z mikroprocesora (połączenie D8 (GPIO15) –EN1,2 – widoczne na pełnym schemacie rys. 22). Wyprowadzenia oznaczone na schemacie (rys. 17) jako 1A, 2A służą do sterowania kierunkiem obrotów silnika, lub np. hamowania przeciwprądem. Wszystko to realizowane jest na podstawie prostej logiki ukazanej w tabeli 4:

Tabela 4 Sposoby sterowania układem L293d

Wyprowadzenie 1A	Wyprowadzenie 2A	Wykonywana czynność
Stan **WYSOKI**	Stan **NISKI**	Kierunek obrotów: **LEWO**
Stan **NISKI**	Stan **WYSOKI**	Kierunek obrotów: **PRAWO**
Stan **NISKI**	Stan **NISKI**	Normalne wyhamowanie
Stan **WYSOKI**	Stan **WYSOKI**	Hamowanie przeciwprądem

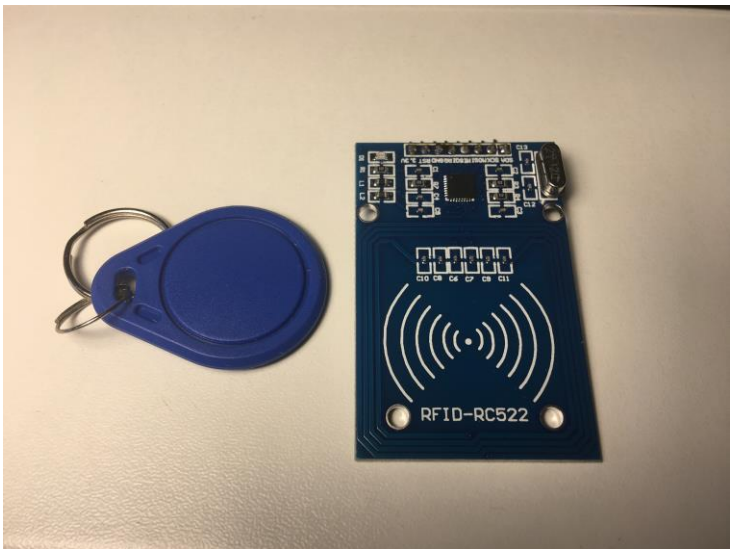
Wyjście układu, oznaczone jako wyprowadzenie 3, 6 prowadzi bezpośrednio do silnika DC. Silnik jaki zastosowano, pochodzi z napędu DVD. Jego napięcie pracy wynosi 5V. Takim napięciem jest również zasilany układ L293d [4]. Jego wydajność prądowa wynosi 0.6A na kanał, jednak prąd maksymalny to 1.2A. Podczas normalnego cyklu otwarcia/zamknięcia bramy, pobór prądu oscyluje w granicach 0,1-0,3A. Aby uniknąć zbyt długiego przeciążenia mostka H, zastosowano prosty dzielnik napięcia, który jest podłączony do jedynego wyprowadzenia analogowego (oznaczona na schemacie linia „A0") w mikroprocesorze ESP. Na jego podstawie badany jest stan wyjścia (czyli de facto prądu silnika). W momencie kiedy silnik natrafi na zbyt duże obciążenie, mikrokontroler zauważa to zjawisko i przerywa procedurę otwarcia/zamknięcia bramy. Dwa rezystory o rezystancji kolejno: 220 Ohmów i 330 Ohmów oprócz pełnienia funkcji zwykłego dzielnika napięcia, zapewniają bezpieczną pracę przetwornika analogowo cyfrowego zastosowanego w układzie ESP8266, bowiem jego maksymalne napięcie wejściowe wynosi 3.3V. Na rysunku 17 przedstawiono omówione rozwiązanie na schemacie elektrycznym (fragment).



Rysunek 17 Fragment schematu –układ sterownika silnika

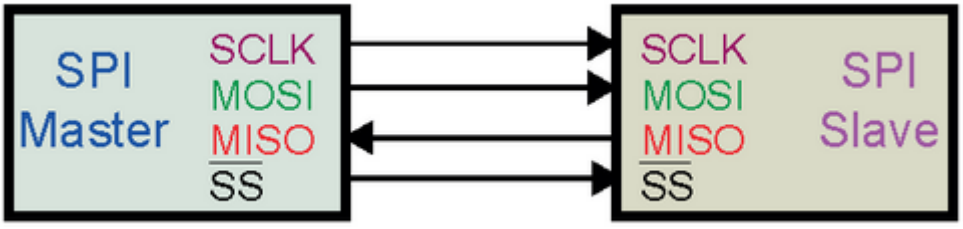
Moduł RFID

Moduł rejestruje unikalne identyfikatory karty lub breloków, wykorzystując do tego fale radiowe. Jest to popularny kontroler RFID-RC522 (ukazany na rys. 18), korzystający z magistrali SPI (*Serial Peripheral Interface*). W projekcie, moduł jest wykorzystany jako alternatywna metoda obsługi bramy. W momencie przyłożenia karty do modułu, brama otwiera się w trybie “dla przechodnia”, co oznacza automatyczne zamknięcie po jej pełnym otworzeniu się (z uwzględnieniem pewnej zwłoki czasowej).



Rysunek 18 Zastosowany kontroler RFID

Magistrala SPI pozwala na przesyłanie danych między dowolnymi urządzeniami obsługującymi tą magistralę. Budowa takiego połączenia polega na określeniu jednego tzw. *mastera* –tzn. urządzenia nadrzędnego oraz praktycznie dowolnej ilości *slave’ów* (tj. urządzeń podrzędnych) taki układ nazywa się *układem łańcuchowym*. W przypadku projektu, układ składa się jedynie z jednego master’a i jednego slave’a. Określenie roli urządzenia odbywa się za pomocą wyprowadzenia SS. Przesyłanie danych wymaga synchronizacji zegarów urządzeń, co można osiągnąć łącząc zegary szeregowe poprzez wyprowadzenie SCLK. Komunikacja odbywa się przy pomocy wyprowadzeń MOSI (Master Out Slave In) - przepływ danych z master’a do slave’s oraz MISO (Master In Slave Out) - przepływ danych ze slave’a do master’a. Opisywana sytuacja jest przedstawiona na rys. 19.



Rysunek 19 Schemat magistrali SPI [8]

4.4 Wykonanie fizyczne urządzenia

4.4.1 Wykorzystane oprogramowanie

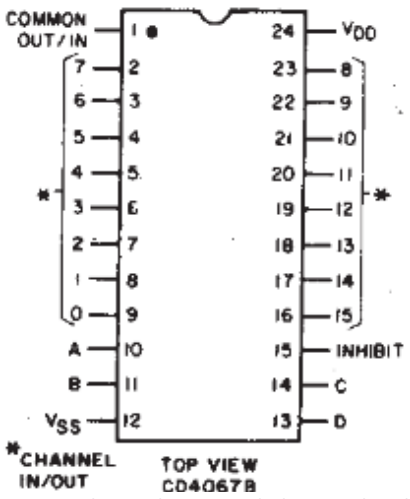
System mikroprocesorowy został zmontowany na własnoręcznie przygotowanej, dedykowanej płytce PCB. Do projektowania schematu i trasowania ścieżek zastosowano darmowy program komputerowy KiCAD. Narzędzie to umożliwiło bardzo przyjazne zaprojektowanie schematu ideowego urządzenia, co pozwoliło na względnie szybkie przekonwertowanie schematu ideowego w schemat płytki drukowanej.

W celu zaprogramowania systemu mikroprocesorowego, zastosowano również darmowe narzędzie, pod nazwą Visual Studio Code. Program ten cechuje się prostotą i szybkością działania. Dużą zaletą jest również fakt, iż ma on ogrom dodatków, które umożliwiają programowanie mikrokontrolerów poprzez środowisko Arduino IDE. Visual Studio Code wspiera również kontrolę nad wersją oprogramowania taką jak Git. W celu efektywnej pracy, stworzono odpowiednie repozytorium dla całego projektu. Dzięki temu jest rejestrowana każda zmiana kodu, bądź innych plików. W przypadku pomyłki, bądź poważnych problemów z nowa wersją oprogramowania, z pomocą repozytorium Git, istnieje możliwość szybkiego przywrócenia poprzednich wersji. Do przygotowania dokumentacji w postaci raportu oraz prezentacji interaktywnej urządzenia wykorzystano pakiet Microsoft Office.

4.4.2 Prototypowanie, prototypy modułów, płytka PCB

Pierwszym etapem wykonania urządzenia, było zapoznanie się z notami katalogowymi podzespołów, które miały być wykorzystane w projekcie. Ten etap został zaprezentowany w podrozdziale 4.3 Analiza rozwiązań, projekt urządzenia. Następnie na podstawie wybranych elementów elektronicznych, wykonano schemat elektryczny układu. Równolegle stworzono i testowano oprogramowanie do obsługi poszczególnych „modułów” z osobna. W dużej mierze odbywało się to na płytce stykowej.

Pierwszym elementem był multiplekser CD4067BE. Na płytce prototypowej został umieszczony wspomniany element. Po dogłębnej analizie okazało się, że nie wymaga on implementacji wszystkich szesnastu wejść/wyjść. Wykonano pewne ograniczenie do 8 wyjść multipleksera.



CD4067 TRUTH TABLE

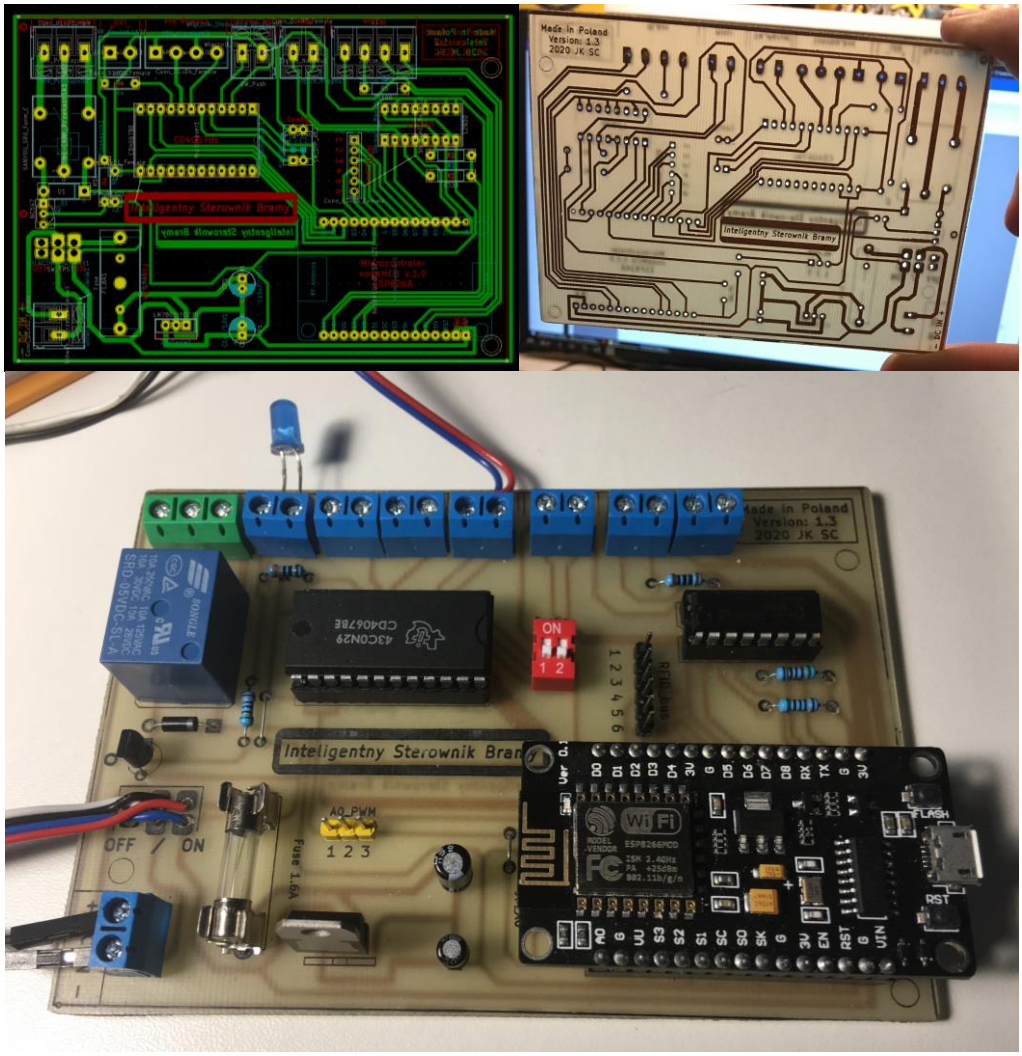
A	B	C	D	Inh	Selected Channel
X	X	X	X	1	None
0	0	0	0	0	0
1	0	0	0	0	1
0	1	0	0	0	2
1	1	0	0	0	3
0	0	1	0	0	4
1	0	1	0	0	5
0	1	1	0	0	6
1	1	1	0	0	7
0	0	0	1	0	8
1	0	0	1	0	9
0	1	0	1	0	10
1	1	0	1	0	11
0	0	1	1	0	12
1	0	1	1	0	13
0	1	1	1	0	14
1	1	1	1	0	15

Rysunek 20 Schemat i tabela prawd multipleksera CD4067BE [6]

Wyprowadzenia sterujące tego multipleksera (oznaczone jako ABCD na rys. 20), zostały ograniczone do trzech, co sprawiło, że multiplekser stał się multiplekserem 8-bitowym, a nie 16-bitowym. Oszczędziło to jedno wyjście cyfrowe mikrokontrolera, które można było wykorzystać do

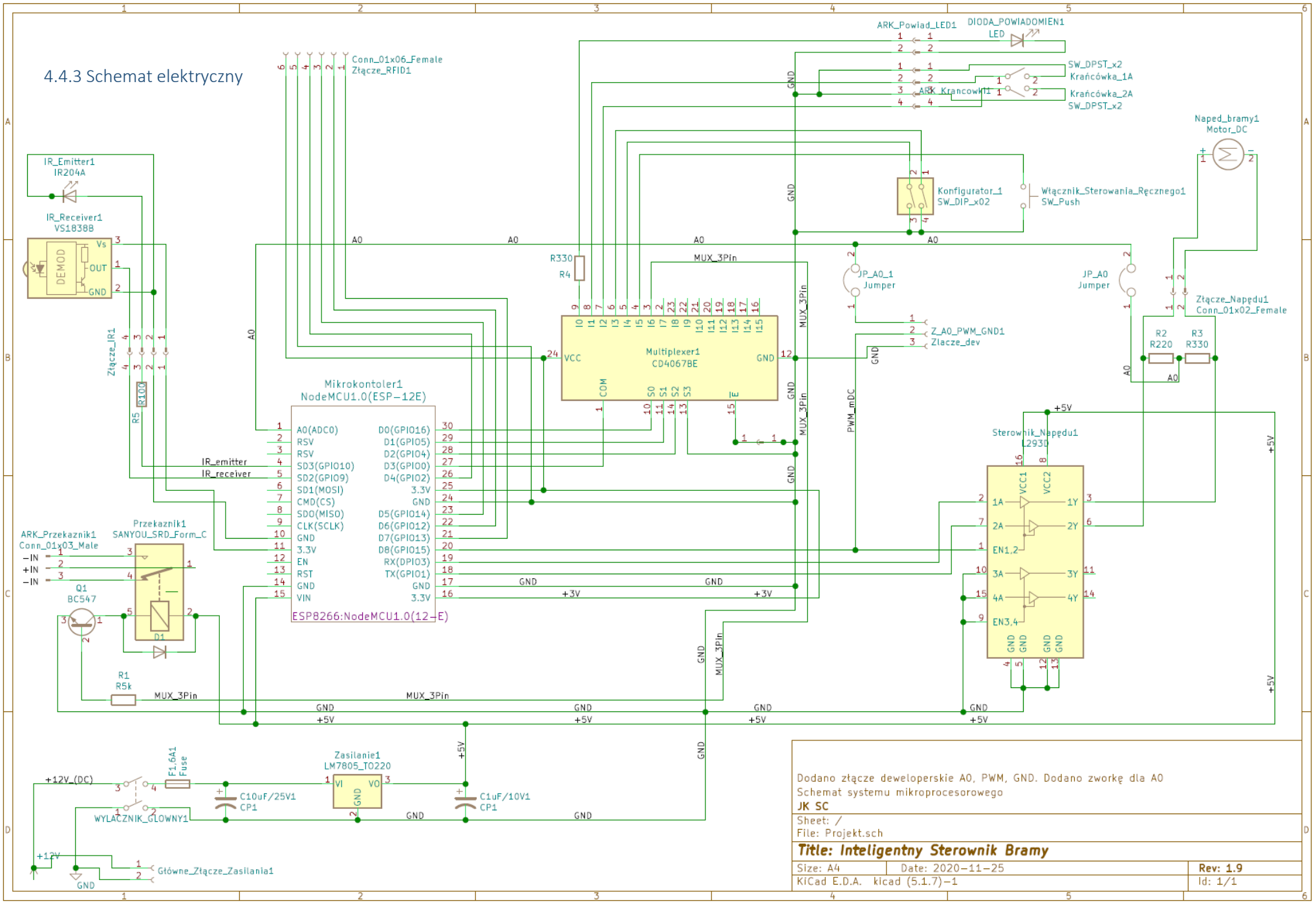
sterowania innymi elementami podwykonawczymi. Efekt ten uzyskano zwierając jedno z wyprowadzeń sterujących do masy urządzenia. Dzięki temu otrzymano „zero” logiczne. W kwestii oprogramowania sterującego multiplekserem, bardzo pomocna okazała się tabela prawd przedstawiona na rysunku 20. Tabela ta, ukazuje w jaki sposób należy wysterować wyprowadzenia sterujące multiplekserem (oznaczone jako ABCD).

Następnie, każdy z zaplanowanych elementów został przygotowany na płytce prototypowej. Po sprawdzeniu poprawności działania, wykonano pełen schemat elektryczny, znajdujący się w podrozdziale 4.4.3 Schemat elektryczny. Tak przygotowany schemat, został zaimportowany do podprogramu KiCAD - PCB Layout Editor, za jego pomocą powstał nadruk do metody termo transferowej. Zwieńczeniem etapu projektowania było wytrawienie gotowego laminatu PCB i montaż gotowego urządzenia. Efekty pracy widoczne są na rys. 21.



Rysunek 21 Projekt i wykonanie płytki PCB

4.4.3 Schemat elektryczny



Rysunek 22 Pełny schemat elektryczny systemu mikroprocesorowego

4.4.4 Właściwe oprogramowanie

W pierwszej fazie projektu oprogramowanie było pisane modułowo, czyli każdy element został przetestowany pod kątem końcowego projektu, osobnym programem przeznaczonym jedynie dla danego czujnika lub elementu wykonawczego. Każdy z modułu był pisany pod kątem jego roli w pełnej wersji programu.

W kolejnej fazie pisania oprogramowania, każdy z modułów służył jako wzór do oprogramowania końcowego. Taki sposób pisania programu pozwolił na uniknięcie problemów wynikających z implementacji rozbudowanego oprogramowania.

Na rysunkach 23 i 24, przedstawiono najważniejsze fragmenty kodu – prototypy funkcji i dwie główne struktury programu:

```
struct motor {        // Short structure for DC Motor
  int ctrlPin[3];     //  Array of control Pins (ENABLE1, INPUT1, INPUT2)
  int value;          //  PWM value
  int direction;      //  Direction for run DC motor
  int actualStatus;   //  Actual status of DC Motor Working (value)/Stopped(0).
  int position;       // Gate position - closed(0), opened(1)
};

struct multiplexer {
  int ABCD[3];       //  Table for pinout multiplexer (eg. 16, 5, 4)
  int read_pin;      //  Pin read for multiplexer (IN)
  int write_pin;     //  Pin write for multiplexer (OUT)
  int channel;       //  Current channel. In declaration - should be NULL
  double ret_val;    //  Received value from readPin() - should be NULL
  double send_val;   //  Send value to multiplexer writePin() - should be NULL
};
```

Rysunek 23 Struktury opisujące napęd bramy oraz multiplekser

```
/*
 * Function Prototypes
*/
  void    setMotor(struct motor &);
  void    goMotor(int,  struct motor &);
  void    stopMotor(struct motor &);
  void    showPosition(struct motor &);
  void    setMultiplexer(struct multiplexer &);
  void    showMultiplexer(struct multiplexer &);
  int     setChannel(int, int *);
  double  readPin(int, struct multiplexer & , bool );
  double  writePin(int, int,  struct multiplexer & , bool );
  int checkEndStop();
  int checkConfig();
  int checkIR(int, int);
  String detect_rfid_user();
```

Rysunek 24 Główne prototypy funkcji w programie

Struktura `motor` jest odpowiedzialna za przechowywanie informacji dot. sterowania silnikiem. Zawiera informacje na temat:

- Wyprowadzeń mikrokontrolera sterujących silnikiem `int ctrlPin[3];`
- Prędkości zadanej silnika `int value;`
- Kierunku obrotów silnika `int direction;` (zdefiniowanej jako LEFT(0)/RIGHT(1))
- Aktualnego stanu silnika `int actualStatus;` (czy silnik zatrzymany, czy pracujący)
- Pozycji skrzydła bramy – otwarta/zamknięta `int position;`

Struktura `multiplexer` definiuje rzeczywisty multiplekser CD4067BE. Zawiera informacje dotyczące:

- wyprowadzeń mikrokontolera sterujących multiplekserem `int ABCD[3];`
- wejścia mikrokontrolera `int read_pin;`
- wyjscia mikrokontrolera `int write_pin;`
- aktualnie wybranego kanału multipleksera `int channel;`
- ostatniej wartości zwróconej przy odczycie wyjścia `double ret_val;`
- ostatniej wartości wysłanej do wyjścia multipleksera `double send_val;`

Obie struktury posiadają charakter globalny. Funkcje wymagające tych struktur, odwołują się do nich, na zasadzie referencji. Sprawia to, że dane są przechowywane w jednym miejscu, i są dostępne dla innych fragmentów kodu. Funkcje kolejno: `setMotor()`, `goMotor()`, `stopMotor()` - umożliwiają inicjalizacje danego napędu; ruch danym napędem z zadaną prędkością (kierunek, bądź prędkość są przekazywane automatycznie); zatrzymanie danego napędu.

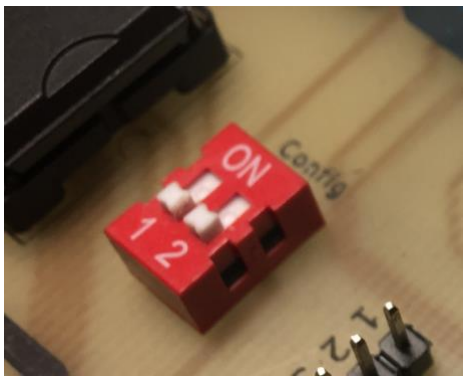
Funkcje `showMultiplexer()`, `showPosition()` umożliwiają odczytanie struktur za pomocą monitora portu szeregowego, obsługiwanego przez Arduino IDE (VS Code – również). Funkcje te służyły głównie do wyeliminowania błędów w oprogramowaniu.

Funkcja `setMultiplexer()` służy do inicjalizacji multipleksera przy rozruchu systemu. Jako że oprogramowanie starano się pisać z naciskiem na uniwersalność i możliwość przyszłego wykorzystania w innych projektach, umożliwiono definiowanie wielu „silników” oraz multiplekserów. W tym projekcie korzystamy z jednego sterownika silnika i z jednego multipleksera.

Funkcje `readPin()`, `writePin()` służą do odczytu i zapisu danych poprzez multiplekser. Działają w bardzo zbliżony sposób, co podstawowe funkcje z Arduino IDE z tą różnicą, iż tutaj oprócz wskazania danego numeru wyjścia, wskazujemy na multiplekser (z którego chcemy skorzystać) i wartość jaką chcemy wysłać oraz tryb odczytu/zapisu – tzn. odczyt/zapis cyfrowy lub analogowy.

Funkcja `checkEndStop()` odpowiedzialna jest za sprawdzanie stanu wyjść czujników mechanicznych – dwóch wyłączników krańcowych (położenie bramy) oraz jednego przycisku do sterowania manualnego. Funkcja ta zwraca odpowiednie wartości całkowite, w przypadku uaktywnienia któregoś z podłączonych wyłączników mechanicznych.

Funkcja `checkConfig()` wywoływana jest przy uruchamianiu urządzenia. Sprawdza ona konfigurację przełącznika DIP-Switch, który jest widoczny na rys. 25. Na jej podstawie zapisywana jest informacja o zadanej prędkości silnika napędu bramy.



Rysunek 25 Przełącznik konfiguracyjny

Funkcja `checkIR()` sprawdza stan czujników przerwania wiązki podczerwonej. W przypadku wystąpienia przeszkody (przerwania wiązki), zwraca wartość 1. W przeciwnym wypadku zwraca 0.

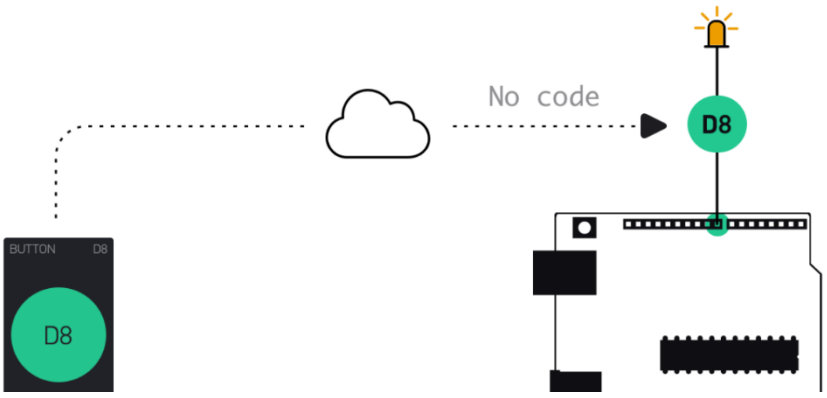
Funkcja `detect_rfid_user()` zwraca kody kart i breloków RFID, które zostały odczytane przez moduł. Na tej podstawie w dalszym etapie, jest wydawana decyzja dot. otwarcia bramy, bądź (w przyszłości) zgłoszenia incydentu do serwera.

4.4.5 Łączność

Łączność z mikrokontrolerem odbywa się z wykorzystaniem wbudowanych bibliotek producenta ESP8266 (dot. połączenia z siecią), technologii Blynk oraz IFTTT. NodeMcu korzystając z odpowiednich bibliotek łączy się z europejskim serwerem firmy Blynk, który przekazuje informacje od użytkownika, takie jak stan danego wyprowadzenia lub wartość liczbową.

Blynk

Blynk to platforma za pomocą której, można połączyć dowolny mikrokontroler dzięki udostępnionym bibliotekom, serwerom oraz aplikacji mobilnej. Aplikacja mobilna generuje klucz, czyli tak zwany *„token”*, za pomocą którego można zidentyfikować dane urządzenie IoT. Wymiana danych odbywa się za pomocą wywołania URL wysłanego na adres IP odpowiedniego serwera, podając odpowiedni klucz i dane opisane zgodnie z założeniami producenta.



Rysunek 26 Przedstawiona na schemacie idea działania platformy Blynk [7]

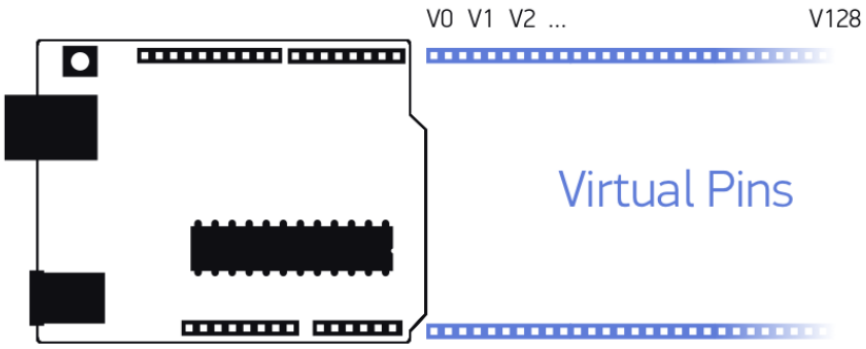
W projekcie przepływ danych odbywa się za pomocą tzw. wyprowadzeń wirtualnych. W tej metodzie, wyprowadzenia (*ang. pins*) to są tak naprawdę kanały komunikacji między serwerem, a mikrokontrolerem. Każdy kanał może przyjmować wartości zakresie od 0 do 255. W przypadku projektu sterownika bramy stosowany zakres stanów to po prostu 0 oraz 1. Stan niski oznacza polecenie zamykające bramę, natomiast stan wysoki polecenie otwierające bramę.

Inteligentni asystenci

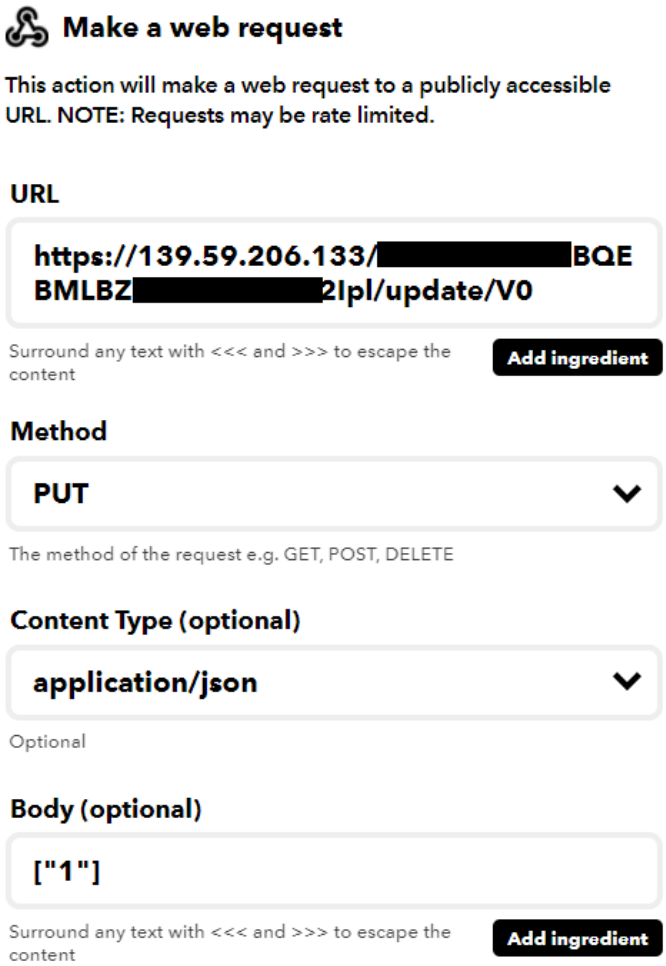
Głównym założeniem projektu jest możliwość stosowania inteligentnych asystentów głosowych, w tym przypadku Asystent Google'a oraz Siri firmy Apple. W przypadku Siri, implementacja jest prostsza, ponieważ aplikacja sama w sobie posiada możliwość wywoływania poleceń URL w momencie wypowiedzenia odpowiedniej komendy głosowej. Niestety w przypadku asystenta Google'a jego wykorzystanie wymaga skorzystania z aplikacji firm trzecich.

IFTTT

Usługa IFTTT (z ang. If This Then That) pozwala na automatyzację procesów i działania aplikacji oraz usług internetowych, takich jak Gmail, Facebook oraz wielu innych. W przypadku projektu usługa pozwala na połączenie asystenta Google'a razem z platformą Blynk. W momencie aktywacji asystenta za pomocą odpowiedniej komendy opisanej w aplecie IFTTT zostaje wysłane polecenie URL, o którym mowa była powyżej.



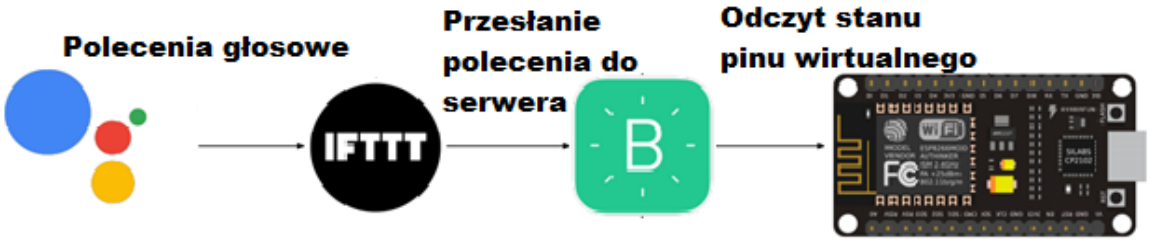
Rysunek 27 Przedstawiona na schemacie idea Wirtualnych wyprowadzeń [7]



Rysunek 28 Przykładowy aplet w usłudze IFTTT [5]

Przepływ danych

Na rys. 29 przedstawiono w jaki sposób przebiega proces przesłania informacji od Asystenta głosowego do mikrokontrolera. Tak jak to już było wcześniej napisane, w przypadku Siri, nie istnieje potrzeba wykorzystania IFTTT.



Rysunek 29 Przepływ danych z wykorzystaniem IFTTT

4.4.6 Testowanie i implementacja poprawek.

Dedykowana płytka PCB

Testowanie rozpoczęło się od sprawdzenia ciągłości połączeń na płytce drukowanej. Okazało się, że trzy pierwsze modele płytki miały w sobie wady fizyczne. Dopiero wersja czwarta była spełniła wszystkie założone wymagania tj. Ciągłość wszystkich połączeń, i brak zwarć w miejscach zagęszczonych. Poprawki polegały na dobraniu odpowiedniej temperatury oraz czasu ogrzewania płytki, w kluczowym procesie przenoszenia nadruku z papieru kredowego na laminat.

Czujnik światła podczerwonego

Problemem związanym z implementacją czujnika światła podczerwonego było dokładne zbadanie jego działania. Po pierwsze, trudno określić czy dioda IR działa poprawnie, ponieważ nie emituje ona żadnego światła widzialnego. Według noty katalogowej przy napięciu zasilania 3V najlepszym rezystorem ograniczającym prąd jest rezystor o oporze 100 Ω, dzięki któremu diody emituje światło o długości fali 940 nm. Zakres czułości czujniki znajduje się w przedziale od 850nm do 1050nm, czyli częstotliwość światła emitera jest identyfikowalna.

Czujnik na wyjściu zwraca napięcie od 0 do napięcia zasilania. W momencie przerwania wiązki światła podczerwonego napięcie gwałtownie spada i może zostać zarejestrowane przez mikrokontroler.

Testy polegały na podłączeniu wyjścia czujnika do wejścia analogowego mikrokontrolera i korzystając z portu szeregowego, monitorowaniu jak zmienia się jego napięcie w zależności od tego co dzieje się z wiązką światła emitera.

Moduł RFID

Moduł RFID posiada osiem połączeń: 2 piny zasilające, 4 piny magistrali SPI, RST i IRQ. Z tego powodu powstało parę problemów. Po pierwsze magistrala SPI wymaga połączenia konkretnych wyprowadzeń mikrokontrolera, których nie można zastąpić, po drugie w przypadku zajęcia ośmiu wyprowadzeń na NodeMcu drastycznie obniża się możliwość podpięcia innych czujników. Z tego względu została podjęta decyzja o pozbyciu się części funkcjonalności modułu RFID , mowa o RST i IRQ, które odpowiadają za przerwania i restart modułu. Dzięki takiemu zabiegowi możliwe było wykorzystanie dwóch niewykorzystanych wejść do innych celów.

Oprogramowanie

Oprogramowanie podczas realizacji projektu było testowane niemalże bez przerwy – na bieżąco. Wynikało to z modularnej struktury kodu. Wpierw pisząc np. oprogramowanie dla multipleksera został on przetestowany czy kieruje wyjścia/wejścia na odpowiednio zadane wartości w funkcji `setChannel()` – rys. 24. Następnie, sprawdzono poprawne odczytywanie wyjść, i wysyłanie danych na wejścia, sprawdzono również jak zareaguje firmware w przypadku błędnie zdefiniowanego kanału multipleksera. Po wyeliminowaniu problemów, dany „fragment” oprogramowania był gotowy, i oczekiwał na końcową – wspólną implementacje.

Takie podejście uchroniło autorów projektu, od dużej ilości błędów, która mogła się nagromadzić podczas pisania całości w jednym momencie. Na tej samej zasadzie odbywało się pisanie implementacji dla kontrolera RFID, modułu sieciowego (łącznie z Blynk), sterowania napędem, itd.

4.5 Problemy w trakcie tworzenia sprzętu i aplikacji

Podczas realizacji projektu, nie wszystko odbyło się bez problemów. Poniżej przedstawiono największe problemy z jakimi należało się zmierzyć:

a) Oprogramowanie obsługujące multiplekser zostało napisane w dwóch wersjach. Pierwsza wersja zakładała użycie tablicy konfiguracyjnej, opisującej wejścia sterujące multipleksera w sposób binarny (tzn. każde wyjście posiadało swój kod binarny np. wyjście 4 = 0100). Takie podejście do problemu nie było rozwiązaniem optymalnym ze względu na korzystanie z dużych tablic. W kolejnej wersji programu podejście zostało zmienione, w programie zostały wykorzystane odpowiednie przekształcenia liczb dziesiętnych na liczby binarne opisujące wyprowadzenia obsługujące multiplekser. Takie rozwiązanie okazało się dużo krótsze i czytelniejsze.

b) Serwer Blynk przesyła informacje do mikrokontrolera dopiero w momencie wysłania żądania poprzez protokół HTTPS. W przypadku asystenta głosowego Siri, który posiada wbudowaną możliwość wywołania odpowiedniego adresu w momencie otrzymania komendy głosowej adaptacja takiego zadania nie jest problemem. Niestety w przypadku asystenta Google'a okazało się, że asystent potrzebował dodatkowej aplikacji IFTTT, której działanie zostało opisane wcześniej.

c) Wykorzystanie diody IR wymagało ograniczenia prądu rezystorem o odpowiedniej rezystancji. Ze względu na to, że światło emitowane przez diodę nie jest widoczne, określenie poprawności działania nie było możliwe i należało polegać na dokumentacji diody. Z tego powodu podczas testów niektóre egzemplarze uległy zniszczeniu.

d) Proces tworzenia płytki z wykorzystaniem laminatu PCB metodą termo transferu wymagał wielu prób i dobrania odpowiedniej temperatury oraz czasu grzania. Pierwsze próby zostały zakończone niepowodzeniem, ścieżki nie posiadały ciągłości. Dopiero po opracowaniu odpowiedniego procesu tworzenie płytki zakończyło się powodzeniem.

5. Podsumowanie

Projekt końcowy spełnił pierwotnie wyznaczone założenia i cele. Udało się uzyskać funkcjonalność pozwalającą na prostą obsługę głosową. Rozwiązanie udowadnia możliwość implementacji intuicyjnej obsługi sterowników urządzeń codziennego użytku, w tym wypadku sterownika bramowego. Korzystanie z takiego urządzenia znacznie wpływa na polepszenie komfortu życia i bezpieczeństwa użytkowników.

Wprowadzenie takich funkcji w wypadku pojedynczego egzemplarza sterownika nie wymaga dodatkowych kosztów wynikających z użycia narzędzi typu Blynk czy IFTTT. Natomiast, podłączenie większej ilości urządzeń wymaga wykupienia odpowiedniej licencji. W takim wypadku warto w przyszłości stworzyć podobną platformę obsługującą urządzenia z możliwością podłączenia do sieci.

W projekcie komercyjnym mostek L293d sterujący silnikiem za pomocą sygnału PWM, powinien zostać zastąpiony sterownikiem o wyższej mocy dobranej pod silniki standardowo wykorzystywane w systemach automatyki bramowej. Dzięki temu sterownik może zostać zastosowany praktycznie do większości tego typu systemów.

Oprogramowanie sterownika oraz jego budowa pozwala zastosować go do innych celów niż obsługa bramy wjazdowej. Urządzenie może zostać wykorzystane, na przykład do obsługi automatyki bramy garażowej. Pomimo innej konstrukcji, funkcje spełniane przez czujniki pozostają takie same.

Warto również wspomnieć o ważnym aspekcie dydaktycznym tego projektu. Wiedza i doświadczenie zdobyte podczas budowy, programowania urządzenia, rozwiązywania problemów, jak i również opracowywania odpowiedniej dokumentacji technicznej zdecydowanie przełożyły się na świadomość autorów projektu dot. konstruowania od podstaw systemów mikroprocesorowych. Jest to bezcenna, wszechstronna wiedza, mająca wpływ na dalszy rozwój osobisty autorów projektu, i postrzegania przez nich wielu rozwiązań technicznych w otaczającej rzeczywistości. Praktyka jest najlepszym sposobem pozyskiwania wiedzy, gdyż charakteryzuje się trwałością i pełnym zrozumieniem danego zagadnienia, w odróżnieniu do stricte teoretycznego podejścia, które pełni równie ważną rolę, jednak nie zapewnia fizycznego i naocznego kontaktu z rzeczywistymi obiektami.

Bibliografia

[1] Make-It.ca, „Dane techniczne mikrokontrolera NodeMcu,” 2015. [Online]. Available: https://www.make-it.ca/nodemcu-arduino/nodemcu-details-specifications/. [Data uzyskania dostępu: 13 Grudzień 2020].

[2] MakerFabs, „Dokumentacja techniczna czujnika VS1838B,” 2020. [Online]. Available: https://www.makerfabs.com/infrared-receiver-vs1838b.html. [Data uzyskania dostępu: 18 Grudzień 2020].

[3] STMicroelectronics, „Dokumentacja techniczna stabilzatora LM7805,” 2014. [Online]. Available: https://botland.com.pl/index.php?controller=attachment&id_attachment=973. [Data uzyskania dostępu: 12 Grudzień 2020].

[4] Texas Instruments, „Dokumentacja techniczna układu L293d (mostek H),” 2016. [Online]. Available: https://www.ti.com/lit/ds/symlink/l293.pdf. [Data uzyskania dostępu: 12 Grudzień 2020].

[5] IFTT.com, „Opis technologii IFTT. Użycie,” 2020. [Online]. Available: https://ifttt.com/home. [Data uzyskania dostępu: 18 Grudzień 2020].

[6] Texas Instruments, „CD4067BE, Dokumentacja techniczna multipleksera,” 2003. [Online]. Available: https://pdf1.alldatasheet.com/datasheet-pdf/view/157680/TI/CD4067BE.html. [Data uzyskania dostępu: 17 Grudzień 2020].

[7] BlynkDocs, „Dokumentacja oprogramowania Blynk,” 2020. [Online]. Available: https://docs.blynk.cc/#blynk-main-operations-virtual-pins. [Data uzyskania dostępu: 18 Grudzień 2020].

[8] Serwis elektroniki, „Jak działa magistrala SPI do komunikacji z Arduino?,” 2018. [Online]. Available: https://www.serwis-elektroniki.com.pl/jak-dziala-magistrala-spi-serial-peripheral-interface-do-komunikacji-z-arduino/. [Data uzyskania dostępu: 18 Grudzień 2020].

[9] Olbud, „Wizualizacja, schemat bramy przesuwnej,” 2019. [Online]. Available: https://olbud.com.pl/img/cms/automaty%20BFT/schemat-automat-do-bramy-przemyslowej.jpg. [Data uzyskania dostępu: 11 Grudzień 2020].