



**POLITECHNIKA
RZESZOWSKA**
im. IGNACEGO ŁUKASIEWICZA



**WYDZIAŁ
BUDOWY MASZYN
I LOTNICTWA**
POLITECHNIKI RZESZOWSKIEJ

PROJEKT INŻYNIERSKI

Autor:

Michał Danek

nr albumu:

163469

Temat projektu:

Wykorzystanie technologii RFID w automatycznym kategoryzowaniu przesyłki

Koordynator przedmiotu: dr inż. Paweł Penar

Forma i stopień studiów: stacjonarne I stopnia

Kierunek studiów: Mechatronika

Rok akademicki: 2022/23

Profil studiów: ogólnoakademicki

Zadania do wykonania:

1. Przegląd literatury w zakresie czytników RFID i ich zastosowaniu w logistyce
2. Przygotowanie algorytmu uwzględniającego obsługę czytnika RFID w kontekście kategoryzowania przesyłek
3. Budowa makiety symulującej przyjęte rozwiązanie dotyczące automatycznego kategoryzowania przesyłek.

Temat projektu
otrzymałem:

Zatwierdzam

Data i podpis Studenta

podpis Koordynatora

*podpis
Kierownika Katedry*

*Data, pieczęć i podpis
Dziekana*

Spis treści

1.WSTĘP.....	3
1.1 Wprowadzenie.....	3
1.2 Cel i zakres pracy	4
2. PRZEGLĄD ROZWIĄZAŃ	5
2.1 Wprowadzenie do technologii RFID.....	5
2.1.1 Częstotliwości i standardy	5
2.1.2 Tag RFID	7
2.1.3 Czytniki	8
2.1.4 Anteny	9
2.2 Przykłady zastosowanie systemów RFID	10
2.2.1 Selektywna bramka RFID.....	10
2.2.2 Inwentaryzacja i kontrola stanów i położenia palet. RFID w magazynie.....	11
2.2.3 System parkingowy dalekiego zasięgu	12
3.OPIS STEROWANIA ZA POMOCĄ ARDUINO/BLNYK	14
3.1 Opis aplikacji BLYNK.....	14
3.1.1 Server BLYNK	15
3.1.2 Aplikacja na Urządzenie mobilne.....	15
3.2 Specyfikacja Arduino Uno R3	17
3.2.1 Specyfikacja techniczna	18
3.2.2 Budowa Płytki	19
3.2.3 Pinout - opis wyprowadzeń	20
3.3 Specyfikacja Ethernet Shield	21
3.4 Specyfikacja serwomechanizmu SG90	22
3.5 Specyfikacja silnika krokowego 28BYJ-48	23
3.6 Specyfikacja czujnika RC522	25
4.REALIZACJA PROJEKTU	26
4.1 Zaprogramowanie Arduino	26
4.1.1 Czytnik RC522 (firmy INDUINO)	26
4.1.2 Komunikacja z aplikacją internetową BLYNK	32
4.1.3 Sterowanie serwomechanizmem SG90	33
4.1.4 Sterowanie silnikiem krokowy 28BYJ-48	35
4.1.5 Obsługa Ethernet Shielda.....	37
4.2 Utworzenie interfejsu w BLYNKU.....	38
4.2.1 Utworzenie pinów komunikacyjnych w zakładce „Datastreams”	38
4.2.2 Utworzenie eventów w zakładce „Events”	39
4.2.3 Utworzenie interfejsu w zakładce „Web Dashboard”	40
4.3 Zaprojektowanie modelu 3D w Programie Inventor Studio	42
4.4 Wydruk elementów na drukarce 3D.....	43
5.TESTOWANIE FUKCJONALNOŚCI OPRACOWANEGO ROZWIĄZANIA.....	45
6.PODSUMOWANIE	48
Bibliografia	49

1. WSTĘP

1.1. Wprowadzenie

W dzisiejszych czasach przy ciągłym zwiększaniu się procentu zakupów wykonywanych przez Internet przedsiębiorstwa zajmujące się magazynowaniem oraz dostarczaniem przesyłek opierały swoje systemy kategoryzowania przesyłek na kodach kreskowych naklejanych na pudełka. Jednak z czasem rozwiązanie to stało się niewystarczająco wydajne. Dlatego większe firmy zaczęły szukać wydajniejszych metod kategoryzowania przesyłek, które pozwoliłyby na przyjmowanie większej liczby przesyłek w takim samym czasie.

Rozwiązaniem, które spełniało te warunki i było dodatkowo, jest opłacalne finansowo, okazała się Technologia RFID (z ang. Radio Frequency Identification). Jest to system identyfikacji z wykorzystaniem fal radiowych, który zyskał szerokie zastosowanie w różnych branżach, w tym w logistyce i magazynowaniu. RFID pozwala na automatyczne i bezdotykowe odczytanie danych przy pomocy specjalnych czytników, co znacząco ułatwia i przyspiesza wiele procesów.

Jednym z zastosowań technologii RFID jest automatyczne kategoryzowanie przesyłek. Możliwe jest to dzięki specjalnym etykietom RFID, które są umieszczane na opakowaniach lub na samych przesyłkach. Etykiety te zawierają informacje dotyczące rodzaju przesyłki, jej wagi, rozmiarów czy też adresu docelowego. Dzięki temu, w momencie, gdy przesyłka trafia do magazynu lub na sortownię, czytnik RFID automatycznie odczytuje dane z etykiety i na tej podstawie dokonuje jej kategoryzacji.

Automatyczne kategoryzowanie przesyłek z wykorzystaniem technologii RFID pozwala na znaczne przyspieszenie i usprawnienie procesu sortowania i przeprowadzenie go z większą dokładnością. Ponadto, dzięki automatycznemu odczytywaniu danych, możliwe jest szybsze wyszukiwanie konkretnych przesyłek oraz łatwiejsze nadzorowanie i zarządzanie całym procesem logistycznym.

Technologia RFID jest coraz częściej wykorzystywana w logistyce i magazynowaniu, jednak nie brakuje również innych zastosowań tej technologii, takich jak np. zarządzanie flotą pojazdów, kontrola dostępu do budynków czy też identyfikacja zwierząt hodowlanych. Warto jednak pamiętać, że technologia RFID nie jest pozbawiona wad i ograniczeń, takich jak np. konieczność stosowania specjalnych etykiet lub brak możliwości odczytu bez specjalnych urządzeń.

1.2.Cele i zakres pracy

Niniejsza praca ma na celu opracowanie algorytmu uwzględniającego wykorzystanie technologii RFID w kontekście automatycznego kategoryzowanie przesyłek na podstawie mikrokontrolera Arduino Uno.

Do realizacji celu zostanie wykorzystany język C++, Ethernet shield, moduł RC522, serwomechanizm, silnik krokowy, aplikacja internetowa BLYNK. Do opracowanego rozwiązania zostanie zaprojektowana w programie makieta 3D, a następnie otrzymany model zostanie wydrukowana na drukarce 3D.

2. PRZEGLĄD ROZWIĄZAŃ

2.1. Wprowadzenie do technologii RFID

RFID (z ang. Radio Frequency Identification) jest technologią, która pozwala bezprzewodowo identyfikować prawie każdy obiekt za pomocą danych przesyłanych za pośrednictwem fal radiowych [7]. Technologia ta pozwala na odczyt i przesyłanie danych oraz w zależności od użytego znacznika możliwość odczytywania od jednego do kilkunastu kart RFID. Technologię RFID stosuje się w wielu dziedzinach życia codziennego między innymi w logistyce, procesach przemysłowych, produkcji, kontroli dostępu, czy również zapewnianiu bezpieczeństwa.

2.1.1. Częstotliwości i standardy

W technologii RFID wykorzystuje się fale radiowe o częstotliwości LF, HF, VHF, UHF, Microwave.

	Zakres częstotliwości				
	LF 125/134,2 kHz	HF 13,56 MHz	VHF 433 MHz	UHF 860-960 MHz	Microwave 2,45 GHz
Nazwa normy	ISO 11784 ISO 11785 ISO/IEC 18000-2 ISO/IEC 18047-2	ISO/IEC 14443 ISO/IEC 15693 ISO/IEC 18000-3 ISO/IEC 18047-3	ISO/IEC 18000-7 ISO/IEC 18047-7 ISO/IEC 24730-3	ISO/IEC 18000-6 ISO/IEC 18047-6	ISO/IEC 18000-4 ISO/IEC 18047-4 ISO/IEC 24730-2

Tab.1. Zestawienie wybranych standardów ISO dla technologii RFID [6]

Przykładowo standard ISO/IEC 7810:2003 definiuje kilka formatów kart dla technologii RFID:

- ID-1 (85,60 x 53,98 mm) – wykorzystywany w kartach płatniczych, prawach jazdy i dowodach osobistych wydawanych m.in. w Polsce,
- ID-2 (105 x 74 mm) – stosowany w niemieckich dowodach osobistych do listopada 2010 roku,
- ID-3 (125 x 88 mm) – ogólnowoświatowy format paszportów i wiz,
- ID-000 (25 x 15 mm) – dla kart SIM.

Wyróżniamy trzy główne standardy częstotliwości RFID: LF- (z ang. Low Frequency) Niskiej częstotliwości, HF- (z ang. High Frequency) Wysokiej częstotliwości oraz UHF- (z ang. Ultra-High Frequency) Ultra wysokiej częstotliwości [8].

RFID Niskiej częstotliwości (LF):

- Zakres częstotliwości: od 30 do 300 kHz, zwykle 125 kHz lub 134 kHz,
- Nazwa standardu: Unique,
- Zasięg odczytu: do 10 centymetrów,
- Zastosowanie: kontrola dostępu, śledzenie zwierząt, breloki samochodowe,
- Zalety: dobrze sprawdza w pobliżu cieczy,
- Wady: niewielki zasięg odczytu, ograniczona ilość pamięci, niska szybkość transmisji danych,
- Liczba równocześnie czytanych znaczników RFID: pojedynczo.

RFID Wysokiej częstotliwości (HF):

- Zakres częstotliwości: od 3 do 30 MHz, zwykle 13,56 MHz,
- Nazwa standardu: Mifare, NFC,
- Zasięg odczytu: do 50 cm,
- Zastosowania: systemy biletowe, książki w bibliotekach, identyfikatory, żetony do gier,
- Zalety: duża pojemność pamięci, globalne standardy bezpieczeństwa,
- Wady: niewielki zasięg odczytu, niska szybkość transmisji danych,
- Liczba równocześnie czytanych znaczników RFID: pojedynczo.

RFID Ultra wysokiej częstotliwości (UHF):

- Zakres częstotliwości: od 300 MHz do 3 GHz,
- Nazwa standardu: UHF Gen2,
- Podstawowe zakresy częstotliwości: 433 MHz, 860 - 960 MHz,
- Zasięg odczytu: 20 metrów,
- Zastosowania: systemy kontroli dostępu, namierzanie oraz identyfikacja towarów,
- Zalety: daleki zasięg odczytu, duża pojemność pamięci, globalne standardy bezpieczeństwa,
- Wady: wysoka częstotliwość pracy oraz stosunkowo duży zasięg mogą potencjalnie zakłócać inne transmisje radiowe,
- Liczba równocześnie czytanych znaczników RFID: wiele.

2.1.2. Tag RFID

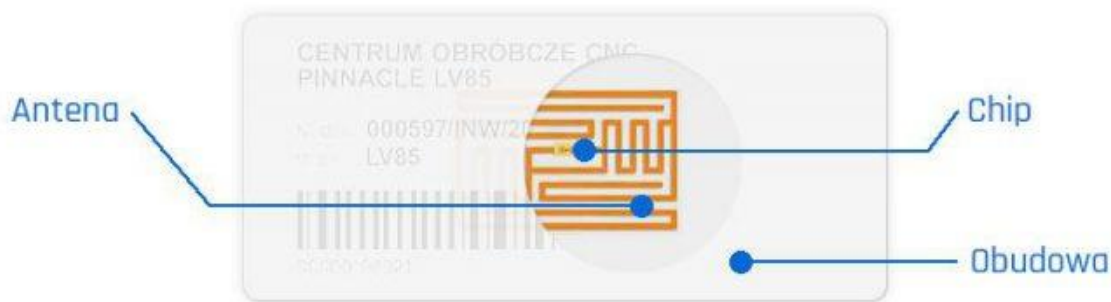
Tagi RFID zwane również transponderami to małe nośniki danych dołączone do przedmiotów za pomocą kleju lub innych substancji. Mogą one przybierać rozmaite formy np.: etykiet samoprzylepnych, małych pudełek, czy plastikowych krążków. Tagi w większości modeli nie wymagają zasilania, ponieważ są zasilane pasywnie poprzez antenę nadajnika. Istnieją także tagi aktywne wyposażone w niewielką baterię. Zasięg takiego transpondera może wahać się od 0.5 do 12 metrów. Umożliwiają zapisywanie, odczytywanie i przechowywanie danych.

Transpondery RFID możemy podzielić ze względu na.:

- Sposób zapisu danych,
 - RO (z ang. Read only) Tylko odczyt – jest to najbardziej podstawowy rodzaj tagu. Służy, jedynie do odczytu danych i znajduje swoje zastosowanie przy przekazywaniu informacji o obecności obiektu,
 - WORM (z ang. Write Once Read Many) Zapisz raz czytaj wiele razy – tagi z możliwości jednorazowego zapisu danych bez możliwości ich modyfikacji czy usunięcia,
 - RW (z ang. Rewritable) Możliwość wielokrotnego zapisu – tagi, które posiadają możliwość wielokrotnego zapisu i odczytu danych.
- Sposób zasilania,
 - Pasywne - Tagi bez własnego źródła zasilania, tagi emitują sygnał dopiero, wtedy kiedy zostaną zasilone z pola elektromagnetycznego emitowanego z anteny nadajnika. Są najtańsze tagi RFID, które umożliwiają użytkowanie przez bardzo długi czas,
 - Sami-aktywne – posiadają one własne źródło zasilania w postaci baterii, ale służące jedynie do zasilania chipa, tag sam nie emituje fal radiowych,
 - Aktywne – transpondery z własnym źródłem zasilania w postaci baterii która zasil chip jak i antenę tagu dzięki czemu emitują fale radiowe. Tagi te posiadają zwiększoną pamięć w stosunku do innych rodzajów, dzięki tym zaletom sprawdzają się w trudnych warunkach. Tagi te mają żywotność od 2 do 10 lat [1].

Tagi RFID zbudowane są z trzech głównych elementów przedstawionych na Rysunku 2.1 :

- Układu scalonego – mikroprocesor z pamięcią, które odpowiedzialne są za przechowywanie i przetwarzanie danych,
- Anteny – odpowiedzialna za wysyłanie i odbiór sygnału radiowego,
- Obudowy – Przykładowo.: plastikowa karta, etykieta papierowa, pojemnik, folia przylepna.



Rysunek 2.1 Budowa Tagu RFID [1]

Każdy tag RFID bez względu na swój rodzaj czy formę fizyczną posiada cztery rodzaje pamięci.:

- Pamięć EPC (EPC Memory) – bank pamięci przechowujący Electronic Product Code, czyli Elektroniczny Kod Produktu. Jest to pamięć edytowalna, w której możemy zapisać dowolny numer HEX o długości maksymalnej do równi pamięci Taga RFID (pamięci EPC – zwykle 96 bitów),
- Pamięć użytkownika (User Memory) – jest to dodatkowa pamięć, którą wykorzystuje się w sytuacji, gdy 96 bitów pamięci EPC okazuje się niewystarczające. Pamięć użytkownika to zwykle dodatkowe 512 bitów do zapisu danych w tagu,
- Pamięć TID (TID Memory) – TID jest pamięcią przechowującą unikalny numer danego transpondera, którego nie można zmienić. Numer TID jest nadawany przez producenta,
- Pamięć bezpieczeństwa – to bank pamięci, w którym znajdują się dwa 32-bitowe hasła: hasło dostępu do blokowania i udostępniania możliwości zapisu danych poufnych oraz hasło niszczące. To drugie hasło całkowicie niszczy tag i uniemożliwia jego działanie [1] .

2.1.3. Czytniki

Czytniki RFID są to urządzenia stacjonarne lub mobilne wyposażone w antenę nadawczo-odbiorczą potrafiące odczytać i zapisywać dane w znaczniku RFID. Czytniki RFID mogą przybierać wiele postaci, od biurowego czytnika krótkiego zasięgu do czytnika UHF dalekiego zasięgu (montowanego na masztach). Czytniki RFID mogą być łączone z antenami RFID w celu zwiększenia zasięgu zbierania informacji z tagów.

Rodzaj czytnika RFID	Zasięg	Typowe zastosowanie
Czytnik RFID biurkowy	1–5cm	Odczyt kart RFID LF (Unique 125kHz), HF (Mifare) i UHF, identyfikacja osób, autoryzacja dostępu do programu, ewidencja dokumentów, identyfikacja pacjentów, wydawanie posiłków, obsługa pralni, rejestracja czasu pracy, program do obsługi konferencji
Stacjonarne bramy RFID	1–15m	Odczyt tagów RFID, identyfikacja przedmiotów, magazyn, handel detaliczny, logistyka, produkcja, handel, obsługa pralni
Przemysłowe czytniki RFID	1–12m	Odczyt znaczników RFID na produktach, maszynach, identyfikacja urządzeń, pojazdów, maszyn, kontrola przepływu towarów, surowców i materiałów, monitoring produkcji, kontrola ruchu na zakładzie
Kolektory danych RFID (terminal RFID)	0,3–5m	Odczyt kart, tagów i breloczków RFID, identyfikacja osób, identyfikacja przedmiotów, obsługa magazynu, handel hurtowy i detaliczny, logistyka, produkcja, inwentaryzacja, ewidencja dokumentów, identyfikacja pacjentów, obsługa konferencji

Tab.2 Czytniki RFID USB - rodzaje i zastosowania [9].

2.1.4. Anteny

Są to urządzenia, które łączy się czytnikami RFID w celu wzmocnienia zasięgu odbierania sygnałów z tagów. Systemy RFID mogą być wyposażone w większość ilość anten, można posłużyć się stwierdzeniem, że czynniki RFID jest mózgiem systemu a anteny jego ramionami. Anteny RFID przeważnie nie różnią się wyglądem fizycznym, to co je od siebie odróżnia to specyfikacja techniczna.

Anteny RFID możemy podzielić w zależności od :

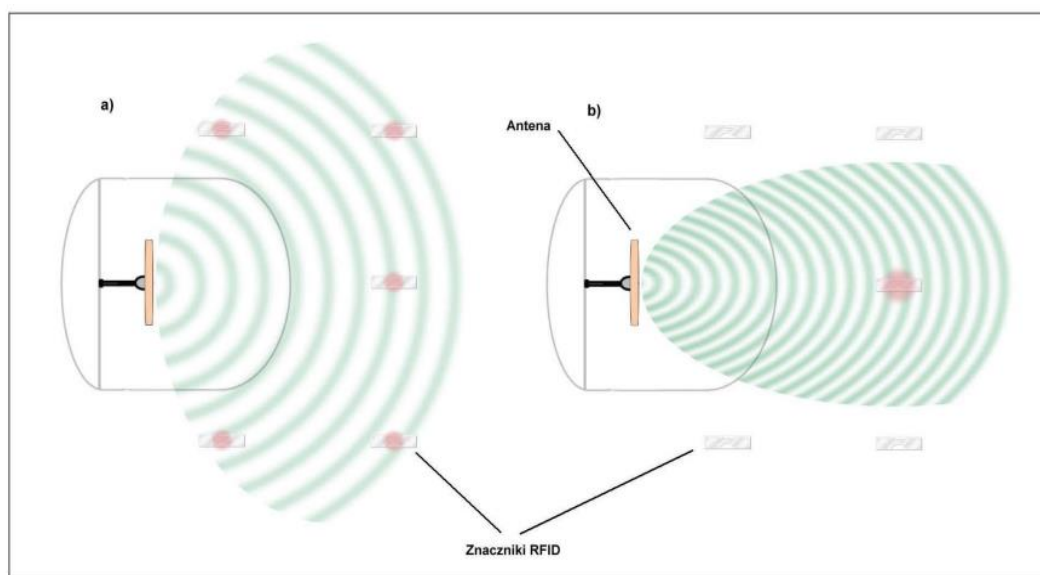
- Polaryzacji,
 - Anteny o polaryzacji liniowej – emitują fale elektromagnetyczne w jednej płaszczyźnie poziomo lub pionowo. Są wyposażone w dużą moc i daleki zasięg odczytu. Ważne jest jednak aby tagi, które mają być zidentyfikowane, zostały ułożone pod podobnym kątem. Duże znaczenie odgrywa zatem orientacja transpondera [1],
 - Anteny o polaryzacji kołowej – anteny o polaryzacji kołowej wysyłają sygnał jednocześnie w dwóch płaszczyznach. Fale rozchodzą się w sposób kołowy, dzięki czemu sygnał nie jest wrażliwy na położenie i orientację tagu. Anteny kołowe mają za to mniejszy zasięg oddziaływania [1],

- Zakres częstotliwości – zakresy są określone w zależności od przepisu obowiązującego w danym kraju/regionie, wyróżniamy trzy najbardziej rozpowszechnione [10]:
 - 865-868 MHz (EU / ETSI),
 - 902-928 MHz (US / FCC),
 - 860-960 MHz (globalny).
- Wzmocnienia i kąt otwarcia – wielkości te są ze sobą ściśle powiązane, ponieważ im większe wzmocnienie, tym węższa fala radiowa co skutkuje węższym obszarem pokrycia, ale i większą długością.

2.2. Przykłady zastosowania RFID

2.2.1. Selektywna bramka RFID

Jest to jedna z powszechniejszych form wykorzystania RFID w magazynach i nie tylko. Odczyt transpondera RFID następuje podczas przejścia przez tak zwaną bramkę RFID, bramka ta jest zestawem anten osadzonych w konstrukcji, która może być dostępna jako element gotowy lub stworzona pod określone rozwiązanie. Właściwości konstrukcyjne bramki mogą wpłynąć bezpośrednio na jej odczyt, a co za tym idzie na jej niezawodność całego procesu automatycznej identyfikacji [2].



Rysunek 2.2 Schemat poglądowy obszaru działania bramki RFID, a) standardowa, b) selektywna [2]

Na rysunku 2.2 pokazano zamierzony efekt działania selektywnej bramki RFID, cecha selektywności może być szczególnie pożądana w takich zastosowaniach jak:

- identyfikacja przemieszczającego się personelu (na przykład wyrobiska kopalniane, platformy wiertnicze),
- automatyzacja procesów odprawy środowiska transportowego (lotnisko),
- inwentaryzacja elementów wchodzących i wychodzących z obiektu (magazyny).



Rysunek 2.3 Selektywna bramka RFID [2]

Na rysunku 2.3 pokazano widok bramki RFID w pełnej konfiguracji prototypowej 2×5 modułów + poprzeczka, światło bramki to 3×3 m. Zaletą bramki jest, modułowa budowa umożliwia dostosowanie jej konfiguracji do konkretnych potrzeb, w zależności od przewidywanego rodzaju i przepływu identyfikowanych obiektów.

2.2.2. Inwentaryzacja i kontrola stanów położenia palet. RFID w Magazynie

Proces inwentaryzacji zaprezentowany na rysunku 2.4, polega na identyfikacji składowanych palet oznaczonych etykietami RFID z pomocą ręcznego czytnika RFID UHF. Przechodząc między paletami, z uruchomionym odczytem skanujemy palety bez konieczności kontaktu wzrokowego z etykietą. Prędkość skanowania determinuje głównie prędkość przemieszczania się pracownika po magazynie. Dla przykładu, z jednego z rzeczywistych naszych wdrożeń, trzyosobowy zespół wyposażony w terminale RFID, inwentaryzuje w przeciągu mniej niż 1,5 godziny magazyn z 6000 palet [5].



Rysunek 2.4 Poglądowy sposób przedstawia inwentaryzacji z wykrzyczeniem RFID [5]

Innym sposobem wykorzystania systemu RFID przy inwentaryzacji i kontroli położenia jest system anten RFID rozmieszczonych na stałe w magazynie, np. nad regałami paletowymi, wzdłuż alejek. Pozwoli to na inwentaryzację w czasie rzeczywistym. W zależności od sposobu realizacji mogą być to systemy automatycznej identyfikacji. Dostarczające danych w przeciągu kilkunastu sekund bądź kilku minut. Cała operacja wykonywana jest bez udziału pracowników [5].

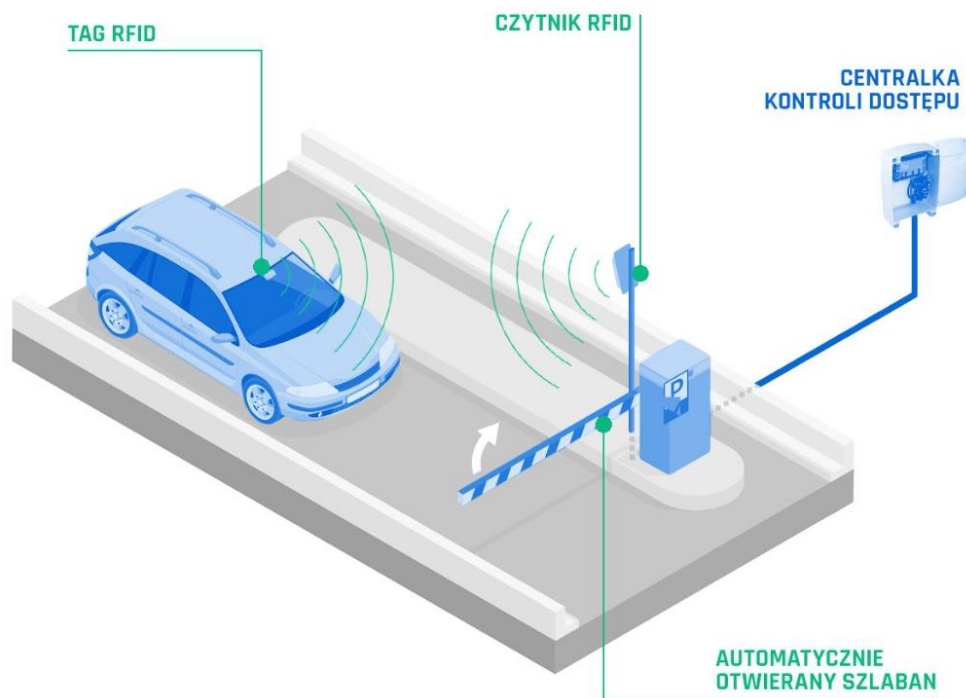
2.2.3. System parkingowy dalekiego zasięgu

Bezobsługowe systemy parkingowe pomagają zarządzać przestrzenią i dokładnie ewidencjonować wjazdy oraz wyjazdy pojazdów. Automatyczna kontrola dostępu ułatwia korzystanie z parkingu zarówno kierowcą, jak i administratorom obiektów. Technika zbliżeniowa RFID pozwala zaoszczędzić czas [11].

System ten działa na zasadzie abonamentu, gdzie kierowca wykupuje miejsce parkingowe na dłuższy okres czasowy. Rozwiązanie to doskonale znajduje zastosowania na parkingach przyzakładowych czy osiedlowych zamkniętych, nie jest jednak zalecane stosowanie tego rozwiązania dla parkingów obsługiwanych poprzez bileterki, takich jak parkingi sklepowe, lotniskowe czy dworcowe.

Budowę takowego systemu zaprezentowano na rysunku 2.5, składa się ona z kilku powiązanych ze sobą elementów, tworząc sprawnie działający monitoring ruchu na parkingu:

- czytnik RFID – zamontowany w pobliżu szlabanów (bram), które odczytują i identyfikują sygnał z chipów umieszczonych na/w zbliżającym się pojeździe,
- centrala kontroli dostępu – mająca za zadanie przechowanie w pamięci wszystkie kody pojazdów uprawnionych do wjazdu na teren parkingu oraz rejestruje wszystkie zdarzenia związane z kontrolą dostępu,
- oprogramowanie – umożliwia poprawne działanie całego systemu oraz analizę historii i danych zgromadzonych w centrali parkingowej,
- znaczniki RFID – umieszczonych najczęściej na szybach samochodów w postaci naklejek z chipem (można również wykorzystywać plastikowe karty RFID) [11].



Rysunek 2.5 Schemat budowy szlabanu wjazdowego na czytnik RFID [11]

3. OPIS STEROWANIA ZA POMOCĄ ARDUINO/BLYNK

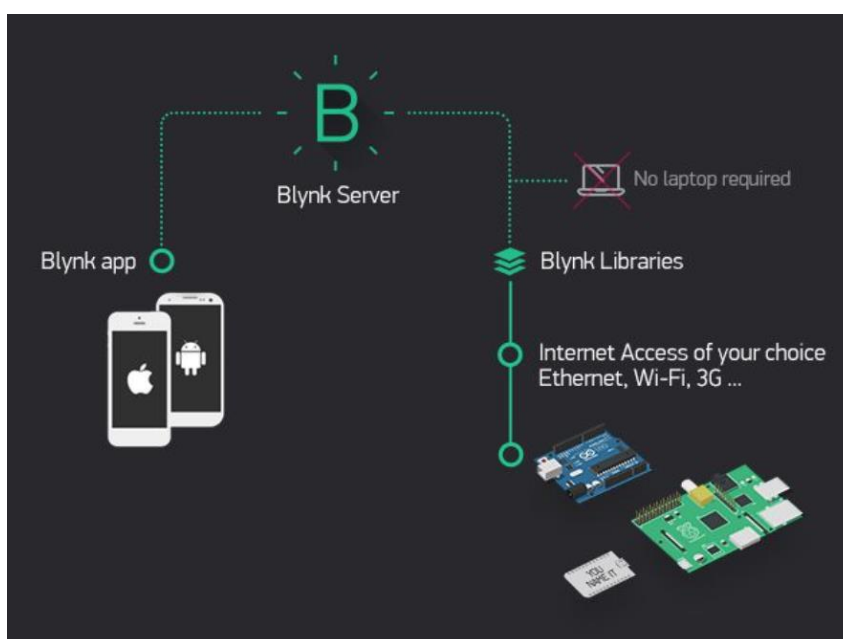
3.1.Opis aplikacji BLYNK

Blynk jest rozwiązaniem typu „multi-tenant”, oznacza to, że pojedynczy system jest wykorzystywany przez wielu użytkowników, których dane są rozdzielone na różnych partycjach aplikacji. Posiada konfiguracyjny dostęp do danych, który na podstawie wcześniej utworzonych ról jest przyznawany użytkownikom. Aplikacje wykonane przy użyciu tego rozwiązania są gotowe dla użytkowników końcowych. Producent Blynk na swojej stronie deklaruje, powszechność swojego rozwiązania, stwierdzając „Niezależnie od tego, czy jest to członek Twojej rodziny, pracownik, czy osoba, która zakupiła Twój produkt, będzie mogła pobrać aplikację, podłączyć urządzenie i rozpocząć korzystanie z niego”. Blynk oferuje również rozwiązanie „white-label” (część BiznesPlanu), co oznacza, że możesz dodać logo swojej firmy, ikonę aplikacji, wybrać motyw, kolory i opublikować aplikację w App Store i Google Play pod nazwą swojej firmy. [14].

Blynk jest to rozwiązanie systemowe IoT, składa się on z trzech elementów zaprezentowanych na rysunku 3.1.

W skład budowy wchodzi:

- aplikacja w telefonie/tablecie,
- serwer Blynk odpowiedzialny za zarządzanie systemem,
- aplikację zaimplementowaną w mikrokontrolerze/urządzeniu klienta.

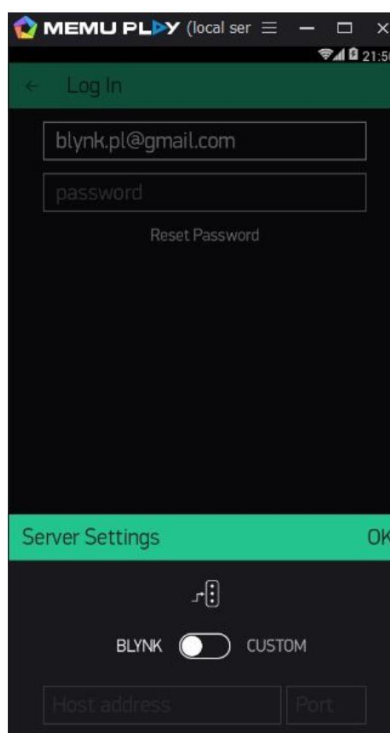


Rysunek 3.1 Schemat działania BLYNKa [12]

3.1.1. Server BLYNK

Blynk serwer w standardowej konfiguracji praktycznie nie jest on dostępny dla użytkownika, działa on bez konfiguracji i nie wymaga dodatkowych linijek kodu. Serwer działa w ten sposób, ponieważ nie wymaga od użytkownika żadnych parametrów do umieszczonego w chmurze serwera. Jego istnienie widać, dopiero gdy chcemy zmienić domyślne ustawienia połączenia pokazane na rysunku 3.2. Choć oczywiście istnieje pod adresem blynk-cloud.com. Warto zapamiętać, bo przyda nam się w przyszłości do kilku ciekawych eksperymentów [12].

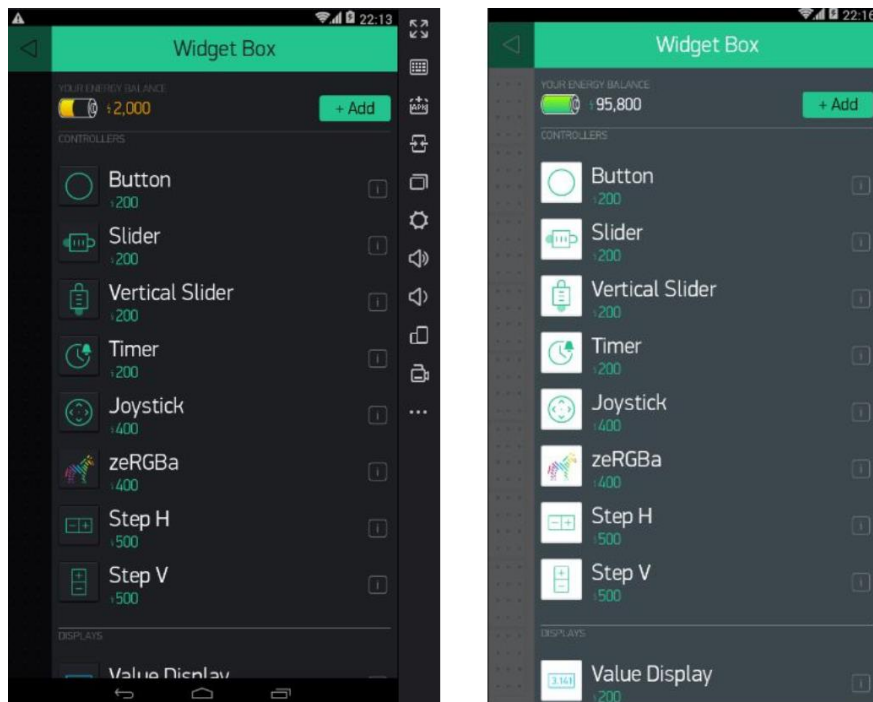
Serwer Blynk działa bez potrzeby konfiguracji, a także bez potrzeby dodawania dodatkowych linii kodu. Nie wymaga on od użytkownika podawania żadnych parametrów do serwera znajdującego się w chmurze. Instalacja serwera Blynk jest standardowym procesem, trwającym zazwyczaj pięć minut. Po tym czasie mamy dostęp do własnego, autonomicznego i darmowego systemu IoT. [12].



Rysunek 3.2 Zmiana ustawień serwera w Blynku [12].

3.1.2. Aplikacja na urządzenia mobilne

Aplikacje na urządzenia mobilne są dostępne na systemach operacyjnych Android oraz iOS. Aplikacja pozwala na dostęp do systemu z każdego miejsca na świecie, z zachowaniem najnowszej wersji oprogramowania. Tworzenie interfejsu w aplikacji mobilnej jest pokazane na rysunku 3.3, jednak ma ono swoje ograniczenia w darmowej wersji - w postaci danej liczby widżetów, którą można poszerzyć poprzez zakup dodatkowych pakietów ulepszeń.



Rysunek 3.3 Tworzenie interfejsu w aplikacji BLYNK [12]

Aplikacja mobilna pozwala na:

- Ustawić nazwę przycisku oraz jego kolor,
- Przypisać, czym dany Pin ma sterować, do wyboru mamy 3 rodzaje pinów, :
 - Digital – sterowanie wyprowadzeniami cyfrowymi,
 - Analog – osobna kategoria na wyprowadzenia,
 - Virtual – to zmienne, które możemy wykorzystywać w programie.
- Wybrać tryb przycisku – bistabilny/monostabilny,
- Ustawić etykiety, które będą się wyświetlały w zależności od stanu przycisku,
- Tworzenie Eventów w przypadku wystąpienia konkretnych zdarzeń, które będą wyświetlane w różnych formach.

3.2.Specyfikacja Arduino Uno R3

Arduino UNO R3 (ostatni człon oznacza trzecią rewizję, czyli wersję płytki) pokazane na rysunku 3.4 jest to jeden z najpopularniejszych płytek zawierających mikrokontroler ARmega328, na płytce umiejscowiono 14 uniwersalnych pinów (wejścia/wyjścia), z czego 6 można wykorzystać jako wyjścia PWM (np. do sterowania silnikami) oraz 6 wyjść analogowych. Płytką tą powstała jako pomoc dydaktyczna dla włoskich studentów, szybko stała się jednak międzynarodowym hitem, który używany jest obecnie przez miliony osób z całego świata [15].



Rysunek 3.4 Płytką Arduino UNO R3 [15]

Arduino wyposażone jest w trzy interfejsy komunikacyjne, a są to:

- UART - opiera się na szeregowym wysłaniu ciągu bitów, które następnie składane są w informację,
- I2C - To szeregową magistralę synchroniczną o prędkości zegara 100 lub 400 kHz. Jego wyprowadzenia znajdują się na lewym końcu gniazdek pinów cyfrowych. Są oznaczone jako "SCL" (sygnał zegara) i "SDA" (sygnał danych). Pozwala na podłączenie do tych samych przewodów do 127 urządzeń i niezależną komunikację z każdym [16],
- SPI- jest to szybka synchroniczna magistrala szeregową. W Arduino UNO jej zegar może pracować maksymalnie z prędkością 8 MHz. Wyprowadzenia magistrali mają oznaczenia "SCK" (sygnał zegara), "MOSI" (wyjście danych), "MISO" (wejście danych) i znajdują się w 6-pinowym wtyku ICSP z prawej strony płytki [16].

Arduino jest rozwiązaniem otwartym źródłowym, co oznacza, że każdy ma dostęp do warstwy sprzętowej oraz programistycznej. Celem tego rozwiązania było umożliwienie każdej osobie zbudowanie własnej wersji płytki lub rozwijanie kompatybilnych projektów we własnym zakresie. Jednak to rozwiązanie okazało się korzystne również dla wielu firm, które zaczęły masowo produkować kopie oryginalnych płytek, tworząc tzw. klony, które przypominały oryginalne Arduino pod względem wyglądu.

3.2.1. Specyfikacja techniczna

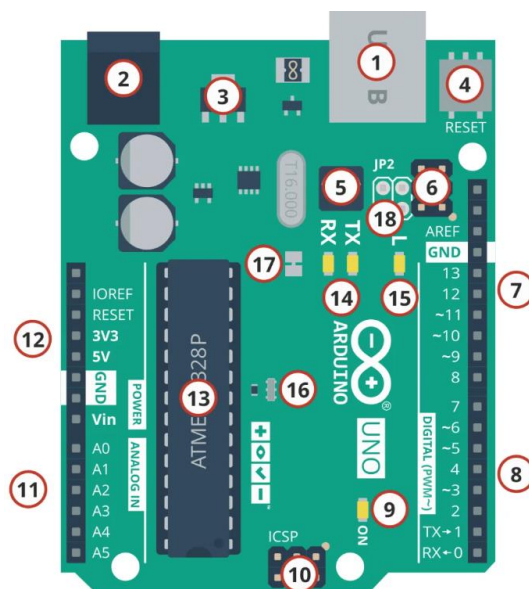
Specyfikacja Arduino UNO R3

- Zasilanie,:
 - złącze koncentryczne 5,5/2,1 mm – napięcie od 7 – 12 V,
 - złącze USB typu B – napięcie 5V.
- Mikrokontroler: ATmega328P w obudowie DIP-28,:
 - pamięć SRAM: 2 kB,
 - pamięć Flash: 32 kB (0,5 kB zarezerwowane dla bootloadera),
 - pamięć EEPROM: 1 kB,
 - częstotliwość zegara: 16 MHz.
- 20 uniwersalnych wyprowadzeń wejść/wyjść,:
 - 14 cyfrowych wejść/wyjść (w tym 6 z możliwością generowania 8-bitowego sygnału PWM),
 - 6 wejść analogowych o rozdzielczości 10 bitów (mogących pełnić funkcję cyfrowych wejść/wyjść).
- Interfejsy komunikacyjne: UART, I2C, SPI,
- Zewnętrzne przerwania: 2,
- Maksymalny prąd,:
 - dla wyjścia 5 V: 500 mA,
 - dla wyjścia 3,3 V: 50 mA,
 - dla poszczególnych GPIO: 20 mA.
- Rozmiary płytki: 68,6 × 53,4 mm.

3.2.2. Budowa Płytki

Skrótowy opis najważniejszych elementów, które znajdziemy na PCB zaprezentowanej na rysunki 3.5.

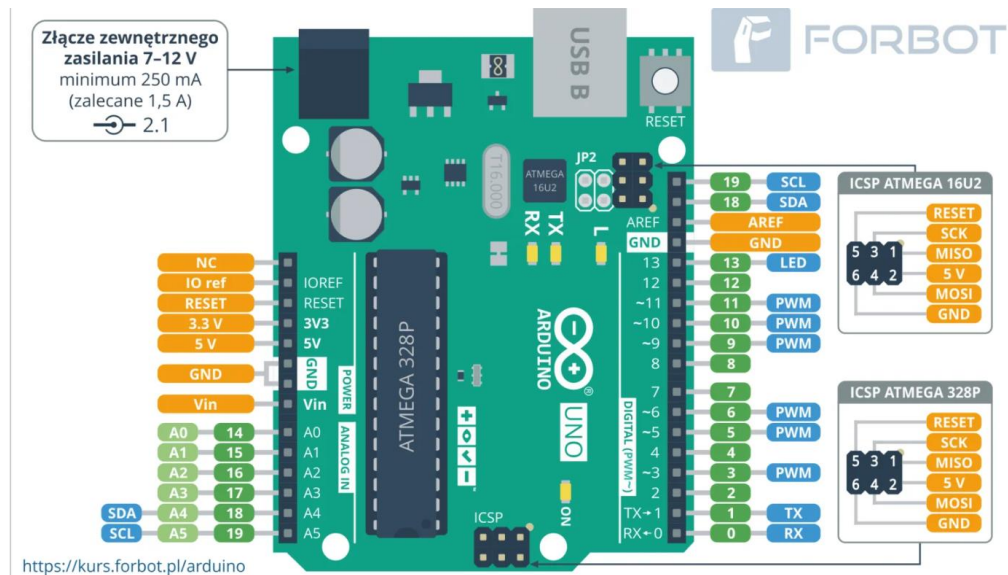
1. Złącze USB - wykorzystywane do zasilania, programowania oraz komunikacji z komputerem,
2. Koncentryczne złącze zasilania (7V - 12V),
3. Stabilizator napięcia - napięcie wejściowe ze złącza nr 2 obniżane jest do 5V dzięki temu układowi,
4. Przycisk resetu - resetuje płytkę Arduino,
5. Mikrokontroler odpowiedzialny za komunikację z komputerem przez USB,
6. Złącze programowania do mikrokontrolera z punktu 5,
7. Złącze sygnałowe,
8. Złącze sygnałowe,
9. Dioda LED sygnalizująca podłączenie zasilania do Arduino,
10. Wyjście programatora dla mikrokontrolera z punktu 13 (Port szeregowy),
11. Złącze sygnałowe,
12. Złącze zasilania,
13. Serce Arduino, główny mikrokontroler AVR ATmega328,
14. Diody LED sygnalizujące transmisję do/z komputera,
15. Dioda LED do dyspozycji użytkownika,
16. Rezonator ceramiczny taktujący mikrokontroler (punkt 13) z częstotliwością 16 MHz,
17. Zworka, której przecięcie wyłącza automatyczne resetowanie Arduino,
18. Pola lutownicze z wyprowadzonymi sygnałami mikrokontrolera z punktu 5, używane ekstremalnie rzadko w bardzo specyficznych i niestandardowych sytuacjach [15].



Rysunek 3.5 Płytko Arduino UNO R3 [15]

3.2.3. Pinout- opis wyprowadzeń

Opis wyprowadzeń znajdujących się na płytce Arduino zaprezentowany na rysunku 3.6 jest bardzo pomocny przy tworzeniu projektów z wykorzystaniem dodatkowych elementów jak np.: silniki krokowe. Z poniższego schematu można się dowiedzieć, jaka jest lokalizacja pinów (od 0 do 13 oraz od A0 do A5) oraz jakie dodatkowe funkcje mogą one pełnić [15].



Rysunek 3.6 Pinout Arduino Uno [15]

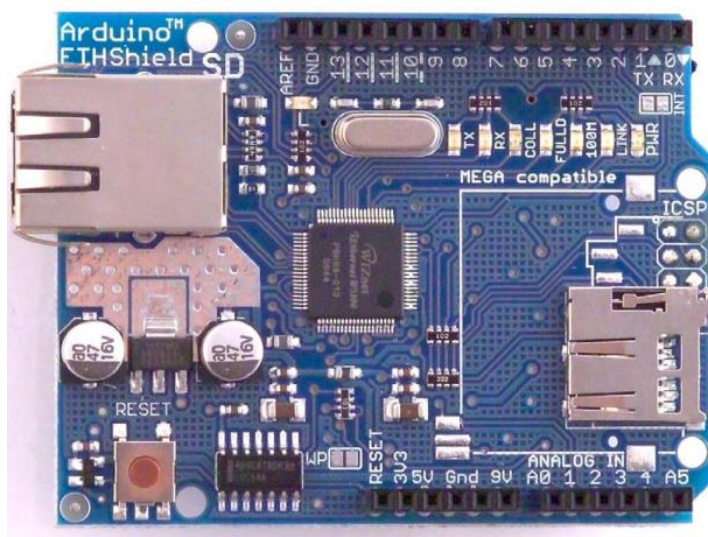
Piny znajdujące się na płytce Arduino możemy podzielić na:

- Piny zasilania - pośredniczą w przekazywaniu napięcia. Wyróżnia się cztery rodzaje pinów,:
 - 3V3 – o napięciu wyjściowym 3,3V,
 - 5V – o napięciu wyjściowym 5V,
 - "GND" (masa) to oznaczenie drugiego bieguna zasilania, popularnie zwanego "minusem".
 - "VIN" to napięcie z wtyczki zasilacza [16].
- Pin RESET – pełniący bliźniaczą funkcję co przycisk reset czyli ponowne uruchomienie mikrokontrolera,
- Pin IOREF - sygnalizuje modułom rozszerzającym Arduino jakim napięciem operują piny cyfrowe,
- Piny cyfrowe (Digital) - "Każdy z nich można zaprogramować tak, by pełnił rolę wejścia lub wyjścia. Charakteryzują się tym, że jako wyjścia mogą mieć napięcie albo 0V, albo 5V. Jako wejścia akceptują też dwa poziomy napięć: około 0V oraz napięcie między 2,5 a 5 V [16]. Z pinów cyfrowych możemy wyróżnić jeszcze:
 - Piny PWM – które pozwalają regulować moc dostarczaną do podpiętego urządzenia,
 - Piny INT – odpowiedzialne za przerwania. Same przerwania działają na zasadzie, że kontroler przerywa główny program i wykonuje specjalnie przygotowany kod programu.

- Piny analogowe (Analog) - Mogą mierzyć napięcie od 0 do 5 V. Wejścia analogowe mają rozdzielczość 10 bitów, co pozwala na rozpoznanie 1024 poziomów napięcia, co daje dokładność około 0.005 V. Dokładność można zwiększyć, zmniejszając programowo zakres napięć od 0 do 1.1 V. Do pinu "AREF" można podłączyć inne napięcie regulujące ten zakres, jednak nie może ono przekraczać 5 V [16].

3.3.Specyfikacja Ethernet Shield

Arduino Ethernet Shield, przedstawiony na rysunku 3.7, pozwala na połączenie Arduino z Internetem za pomocą kabla Ethernet. Shield oparty jest na układzie ethernetowym Wiznet W5100. Ten układ zapewnia stos sieciowy (IP) obsługujący zarówno TCP, jak i UDP, oraz obsługuje do czterech jednoczesnych połączeń gniazdowych [17]. Shield łączy się z płytką Arduino za pomocą pinów umiejscowionych na jego spodzie. Shield posiada także czytnik kart pamięci microSD oraz jest kompatybilny z Arduino Uno, Mega, Leonardo, oraz Nucleo.



Rysunek 3.7 Ethernet Shield W5100 [17]

Ethernet shield zawiera również szereg informacyjnych diód LED [17]:

- PWR – wskazuje, że płyta i shield są zasilane,
- LINK – wskazuje na obecność połączenia sieciowego i miga, gdy tarcza nadaje lub odbiera dane,
- FULLD – wskazuje, że połączenie sieciowe odbywa się w trybie full duplex,
- 100M – wskazuje na obecność połączenia sieciowego 100 Mb/s (w przeciwieństwie do 10 Mb/s),
- RX – miga, gdy shield odbiera dane,
- TX – miga, gdy shield wysyła dane,
- COLL – miga, gdy wykryte zostaną kolizje sieciowe.

Specyfikacja techniczna Etherent Shield:

- Zasilanie: 5V (pobierane z Arduino),
- Kontroler Ethernet: W5100 z wbudowanym buforem 16K i oscylatorem 25MHz,
- Prędkość transmisji: 10/100 Mb/s,
- Komunikacja z Arduino: SPI,,
- układ ACS712ELC-20A,
- interfejs SPI,
- Slot na kartę MicroSD – SPI,
- Wbudowany stabilizator AMS 3,3V,
- Średni pobór prądu: 10 mA,
- Wymiary: 74x54x29 mm,
- Długość pinów: 9,5mm,
- Wymiary znaku: 2,45 x 5,00 mm,
- Zakres temperatur pracy: od -20 do +70 °C.

3.4.Specyfikacja serwomechanizmu SG90

Serwomechanizm SG90 przedstawiony na rysunku 3.9 zbudowany jest z:

- szczotkowego silnika DC,
- przekładni redukcyjnej o wyjściowym momencie obrotowym na poziomie 1,8 kg*cm,
- analogowego kontrolera z potencjometrem odczytującym aktualną pozycję serwomechanizmu.

Rozmieszczanie tych elementów w obudowie pokazano na rysunku 3.8.

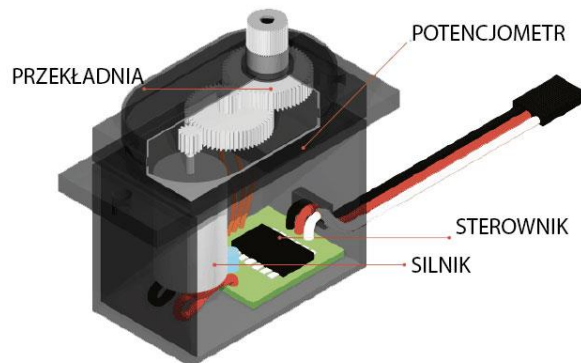
Sam serwomechanizm pracuje w pętli sprzężenia zwrotnego, czyli poprzez podawany sygnał PWM zostaje automatycznie ustalony pozycja wału wyjściowego, na bieżąco korygując wszystkie odchyłki. Zasilanie serwomechanizmu jest to napięcie stałe z przedziału 4,8V-6V, a zakres obrotu wynosi, 180 stopi.

Specyfikacja techniczna:

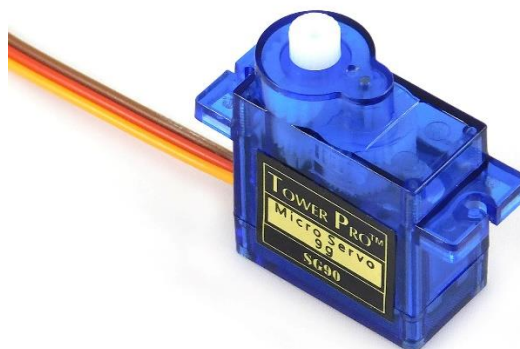
- Wymiary: 22 x 11.5 x 27 mm,
- Waga: 9 g,
- Zasilanie: od 3 do 6 V,
- Pobór prądu: 550 mA,
- Standard stanów logicznych: 3,3 V lub 5 V,
- Sterowanie: PWM,
- Zakres wypełnienia PWM: 500 - 2400 mikrosekund,
- Szybkość: obrót o 60 stopni w 0.12 s,
- Siła: 1,2 kg / cm ramienia przy zasilaniu 4.8V.

Wyprowadzenie:

- brązowy przewód: GND (masa zasilania),
- czerwony przewód: Zasilanie 5 V,
- pomarańczowy przewód: Sygnał sterujący PWM.



Rysunek 3.8 Budowa Serwomechanizmu [18]



Rysunek 3.9 Serwomechanizm modelarski TowerPro SG90 [19]

3.5. Specyfikacja silnika krokowego 28BYJ-48

Silnika krokowego 28BYJ-48 jest to element stosowany między innymi z mikrokontrolerami, najczęściej dobieranym do niego jest Arduino. Jest to element bardzo popularny w wielu dziedzinach przez wzgląd na swoją cenę, jak i niskie napięcie zasilania oraz niski poziom hałasu, jaki generuje. Silnik ten z wykorzystaniem dodatkowej biblioteki dostaje dostęp do szeregu nowych funkcji przykładowo:

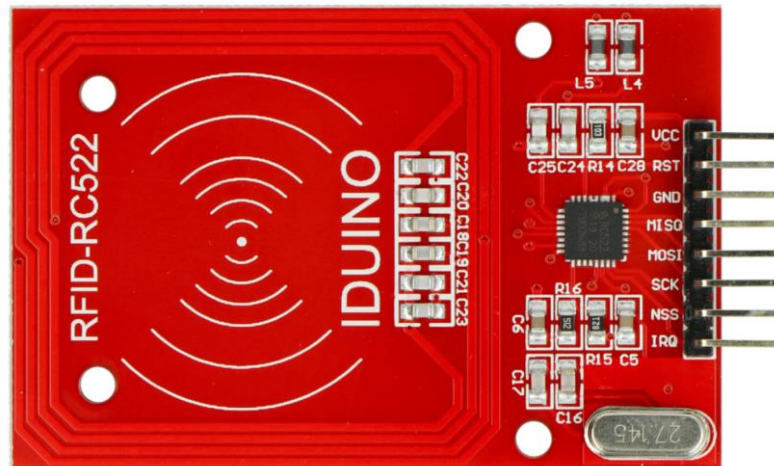
- Dokładna ilość kroków,
- Określona prędkość działania,
- Zadany kierunek obrotów.

Budowa wyprowadzeń silnika przedstawiono na rysunku 3.9, oraz jego wymiary na rysunku 3.10.

3.6.Specyfikacja czujnika RC522

Czytnik RC522 zaprezentowany na rynku 3.11, jest to płytka umożliwia odczyt i zapis danych z urządzeń RFID na częstotliwości 13,56 MHz. Moduł oparty jest na popularnym układzie RC522 posiada wbudowaną antenę, zasilany jest napięciem 3,3 V, komunikuje się poprzez SPI [21].

Moduł może odczytywać taki RFID działające na częstotliwości 13,56 MHz. Antena wbudowana w czytnik pozwala na odczyt maksymalnie na dystansie do 10 cm z uwzględnieniem niewielkich przeszkód.



Rysunek 3.11 Czytnik RC522 firmy IDUINO [21]

Specyfikacja techniczna:

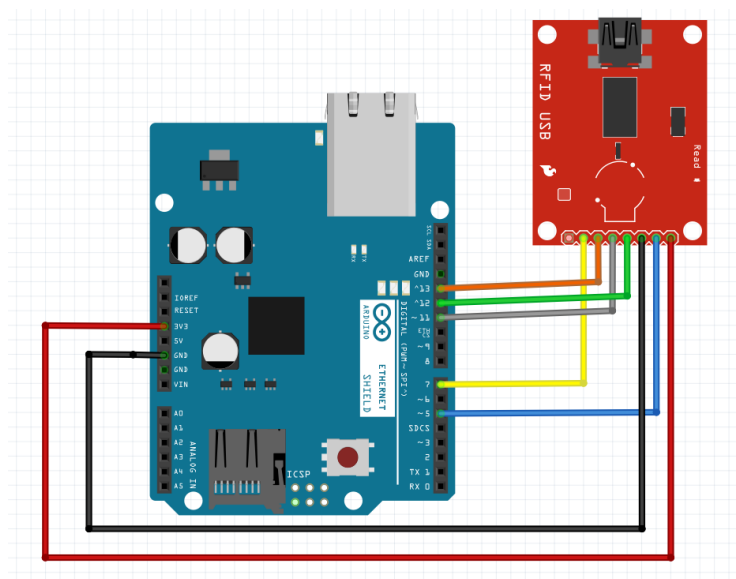
- Napięcie pracy: 3,3 V,
- Kontroler: RC522,
- Możliwość odczytu i zapisu danych,
- Zasięg: do 10 cm,
- Komunikacja: SPI (biblioteka dla Arduino),
- Wymiary płytki: 60 x 40 x 5 mm.

4. REALIZACJA PROJEKTU

4.1. Zaprogramowanie Arduino

4.1.1. Czytnik RC522 (firmy INDUINO)

Do rozpoczęcia pracy z czytnikiem RC522 należy odpowiednio podpiąć go do Ethernet Shielda, schemat połączeń przewodów możemy zobaczyć na rysunku 4.1.



Rysunek 4.1 Podłączanie czytnika RC522.

Następnym krokiem będzie napisanie kodu umożliwiającego odczyt, jak i zapis pamięci karty magnetycznej. Dlatego należy najpierw zapoznać się z budową pamięci znacznika, wykorzystana pamięć w użytych znacznikach to MIFARE Classic 1K i jest ona podzielona na 16 sektorów (od 0 do 15). Każdy sektor jest w następnej kolejności podzielony na 4 bloki (od 0 do 3). A dany blok może pomieścić 16 bajtów danych (od 0 do 15). Czyli podsumowując, powyższe informacje otrzymujemy:

16 sektorów x 4 bloki x 16 bajtów danych = 1024 bajty = 1k pamięci

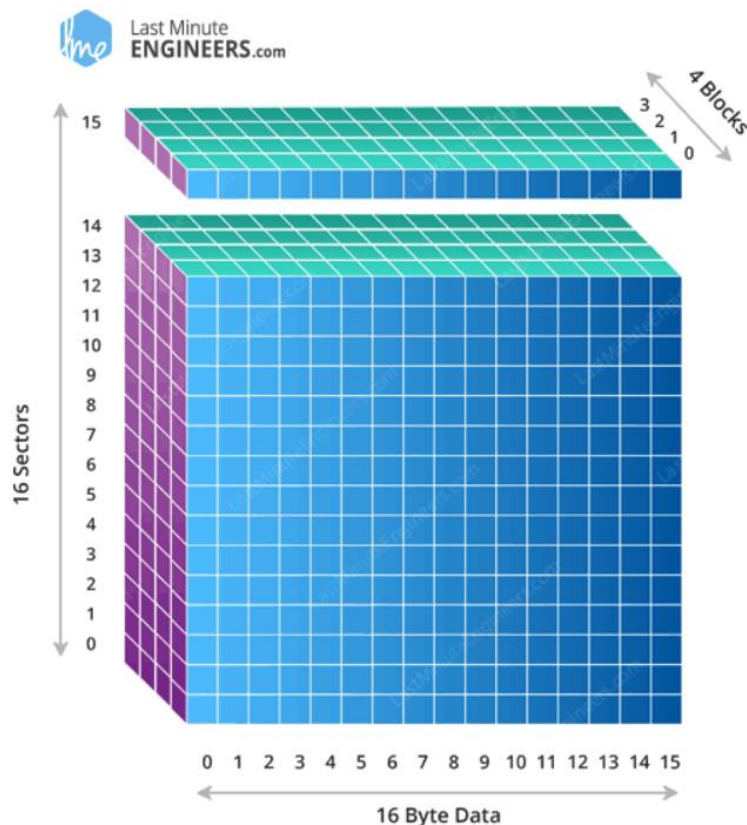
```

Firmware Version: 0x92 = v2.0
Scan PICC to see UID, SAK, type, and data blocks...
Card UID: 20 C3 93 5E
Card SAK: 08
PICC type: MIFARE 1KB
Sector Block 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 AccessBits
15 63 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
    62 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
    61 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
    60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
14 59 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
    58 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
    57 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
    56 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
13 55 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]
    54 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
    53 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
    52 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 [ 0 0 0 ]
12 51 00 00 00 00 00 00 FF 07 80 69 FF FF FF FF FF FF [ 0 0 1 ]

```

Rys 4.2 Podgląd serial monitora w programie Arduino IDE

Pamięć taką możemy przedstawić graficznie w następujący sposób:



Rys 4.3 Reprezentacja 3D układu mapy pamięci MIFARE Classic 1K [5]

Warto zwrócić uwagę jeszcze na ostatni bloki danego sektora, który nazywamy „Sector Trailer”, zawierają one informacje, które nazywamy „Access Bits”, które natomiast umożliwiają odczyt oraz zapis pozostałej pamięci sektora [5]. Czyli w każdym sektorze pamięci tylko trzy pierwsze bloki (od 0 do 2) są, faktycznie zapisywalne a oznacza to, że mamy do dyspozycji 48 bajtów na sektor.

Innym blokiem, na który warto zwrócić uwagę jest blok 0 sektora 0 jest to tzw. „Blok producenta”, który zawiera dane producenta układu scalonego oraz unikalny indykator (**UID**)

	16	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
3	15	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF	[0 0 1]
	14	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	13	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	12	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
2	11	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF	[0 0 1]
	10	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	9	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	8	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
1	7	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF	[0 0 1]
	6	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	5	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	4	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
0	3	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF	[0 0 1]
	2	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	1	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	0	20 C3 93 5E	2E 88 04 00	00 00 00 00	00 00 00 00	[0 0 0]

Rys 4.4 Podgląd serial monitora w programie Arduino IDE
z podkreślonym blokiem producenta.

Ważne jest, aby nie nadpisywać bloku producenta, ponieważ grozi to stałym zablokowaniem znacznika/karty.

Przechodząc to analizy kodu, w pierwszej kolejności należy zainicjować użycie biblioteki MFRC522.h, która będzie niezbędna do obsługi czujnika RFID.

```
#include <MFRC522.h>
```

W następnej kolejności należy skonfigurować piny „slave” oraz „reset”. Oraz zainicjować MFRC522 jak i utworzyć strukturę „MIFARE_Key” o nazwie „key”, która będzie przechowywała informacje o karcie oraz „StatusCode”, który będzie nas informował o statusie karty.

```
#define RST_PIN      5
#define SS_PIN       7
MFRC522 mfrc522(SS_PIN, RST_PIN);    // Tworzenie instancji MFRC522
MFRC522::MIFARE_Key key;
MFRC522::StatusCode status;
```

Plik konfiguracyjny powinien wyglądać w następujący sposób.

```
void setup() {
  Serial.begin(9600);           // Inicjalizacja komunikacji szeregowej
  // z komputerem PC
  SPI.begin();                 // Inicjalizacja magistrali SPI
  mfrc522.PCD_Init();          // Inicjowanie karty MFRC522
  Serial.println(F("RFID Sorting:")); // Pokazanie nazwy projektu informujące
  // o gotowości programu
```

W głównej pętli programowej, żeby uzyskać dostęp do karty, musimy zainicjować klucz bezpieczeństwa chipa „security key”. Bazowo klucz ten jest ustawiony na „0xFF”, w przypadku użycia karty ze zmienionym kluczem należy to uwzględnić w programie.

```
for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
```

W dalszej części kodu należy ustawić warunek sprawdzający, czy nie pojawiła się nowa karta do odczytu pod nazwą „PICC_IsNewCardPresent”. W przypadku wykrycia nowej karty program odczyta jej numer seryjny, a program przejdzie do dalszego odczytywania zawartości, jeżeli operacja będzie niemożliwa, zostanie wyświetlony komunikat o błędzie karty.

```
if (mfrc522.PICC_IsNewCardPresent()) {  
    if (mfrc522.PICC_ReadCardSerial()) {  
        RFID();  
    }  
    else  
    {  
        Serial.println("Bład karty");  
        Blynk.logEvent("bd_karty");  
    }  
}
```

W podprogramie o nazwie RFID będzie wykonywana operacja odczytywania wybranych danych z pamięci karty. Wybieramy sektor 0 blok 4. Pamiętaj, aby nigdy nie wybierać bloku nr 3 żadnego sektora. Pisanie w bloku „Sector Trailer” może sprawić, że blok nie będzie nadawał się do użytku.

```
byte block = 4;  
byte len = 18;
```

Następnie definiujemy tablicę 18 bajtów o nazwie „buffer”. Użyjemy go do odczytania treści pisanych. Dla wyjaśnienia 18 bajtów wynika z tego, że „MIFARE_Read” wymaga buforowania o długości co najmniej 18 bajtów, w których mieści się 16 bajtów bloku.

```
byte buffer1[18];
```

Następnie poprzez wcześniej utworzoną zmienną status należy sprawdzić, czy klucz bezpieczeństwa jest prawidłowy oraz możliwość odczytania wybranych danych z karty.

```

status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 4, &key,
&(mfrc522.uid));

if (status != MFRC522::STATUS_OK)
{
    Serial.print(F("Uwierzytelnianie nie powiodło się: "));
    Blynk.logEvent("bd_autoryzacji_rfid");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}
status = mfrc522.MIFARE_Read(block, buffer1, &len);
if (status != MFRC522::STATUS_OK)
{
    Serial.print(F("Nieudany Odczyt: "));
    Blynk.logEvent("nieudany_odczyt_");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
}

```

Przed procesem odczytu danych zalecane jest zapisanie w wybranym sektorze danych. Dane można zapisać za pomocą poniższego kodu.:

```

Serial.setTimeout(20000L);
byte block = 4;
byte buffr[] = {0x44,0x72,0x20,0x69,
                0x6E,0x7A,0x2E,0x20,
                0x50,0x61,0x77,0x65,
                0x6C,0x20,0x50,0x65,
                0x6E,0x61,0x72};
writeBytesToBlock(block, buffr);

```

Dane są zapisywane w formacie szesnastkowym, dlatego zaleca się korzystanie z konwertera [patrz rysunek 4.5]. W celu zapisania danych w kodzie szesnastkowym do zmiennej "buffr" należy skorzystać z polecenia "writeBytesToBlock". Pamiętaj, że każdy blok pamięci musi mieć długość 16 bajtów, dlatego dane zapisane muszą być odpowiedniej długości.

ASCII Text to Hex Code Converter

Enter ASCII/Unicode text string and press the *Convert* button:

The screenshot shows a web application interface for converting ASCII text to hexadecimal. At the top, there are two dropdown menus labeled 'From' and 'To'. 'From' is set to 'Text' and 'To' is set to 'Hexadecimal'. Below these are two buttons: 'Open File' with a folder icon and a search button with a magnifying glass icon. A text input area contains the string '12345678998765432112'. Below the input area is a 'Character encoding' dropdown set to 'ASCII'. Underneath is an 'Output delimiter string (optional)' section with a dropdown set to 'Space' and an empty text input field. At the bottom of the input section are three buttons: a green 'Convert' button with a circular arrow icon, a grey 'Reset' button with an 'X' icon, and a grey 'Swap' button with a double-headed arrow icon. The output area at the bottom shows the converted hexadecimal string: '31 32 33 34 35 36 37 38 39 39 38 37 36 35 34 33 32 31 31 32'.

Rys 4.5 Prezentacja zmiana na format szesnastkowy [22]

Kiedy cały proces weryfikacji przebiegł pomyślnie oraz uprzednio zostały zapisane dane w pamięci, wtedy w pętli `for` do zmiennej „value” zapisujemy zawartość pamięci.

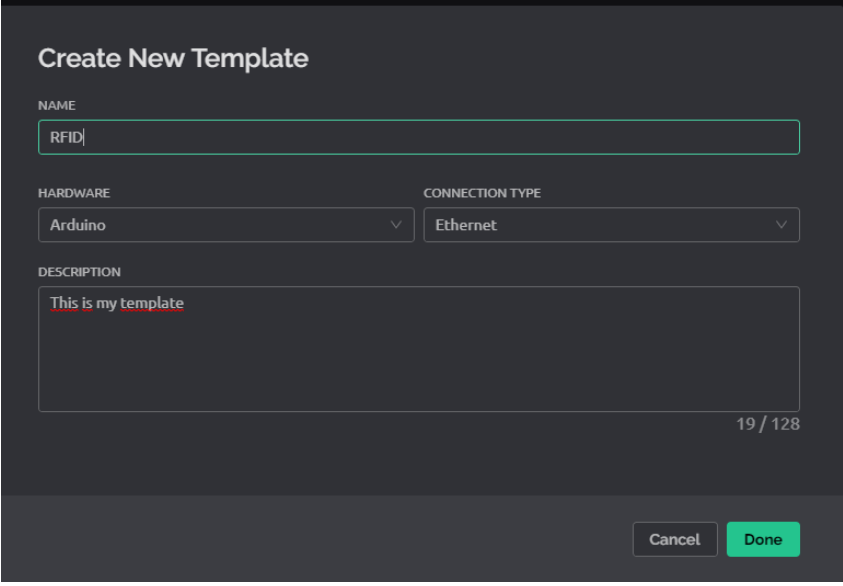
```
for (uint8_t i = 0; i < 16; i++)
{
    value += (char)buffer1[i];
}
```

Na koniec procesu wyświetlamy odczytane dane w „serial monitor” oraz komunikat o zakończeniu odczytu danych.

```
Serial.print(value);
Serial.println(F("\n**Zakończenie czytania danych**\n"));
```

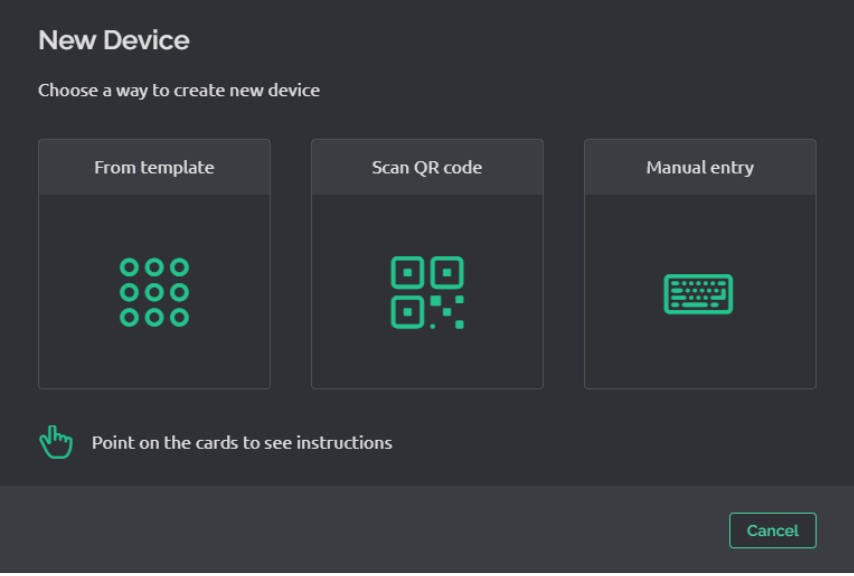
4.1.2. Komunikacja z aplikacją internetową BLYNK

Kolejnym krokiem jest stworzenie komunikacji między mikrokontrolerem a serwerem BLYNK. Aby to zrobić, najpierw należy otworzyć stronę Blynka w przeglądarce, a następnie przejść do zakładki "Templates". Tam, należy utworzyć nowy szablon. Tworzenie przedstawione jest na rysunku 4.5.

The screenshot shows the 'Create New Template' interface. At the top, the title 'Create New Template' is displayed. Below it, there are three main sections: 'NAME' with a text input field containing 'RFID'; 'HARDWARE' with a dropdown menu set to 'Arduino'; and 'CONNECTION TYPE' with a dropdown menu set to 'Ethernet'. Below these is a 'DESCRIPTION' section with a large text area containing the text 'This is my template'. A character count '19 / 128' is visible at the bottom right of the description area. At the bottom of the form, there are two buttons: 'Cancel' and 'Done'.

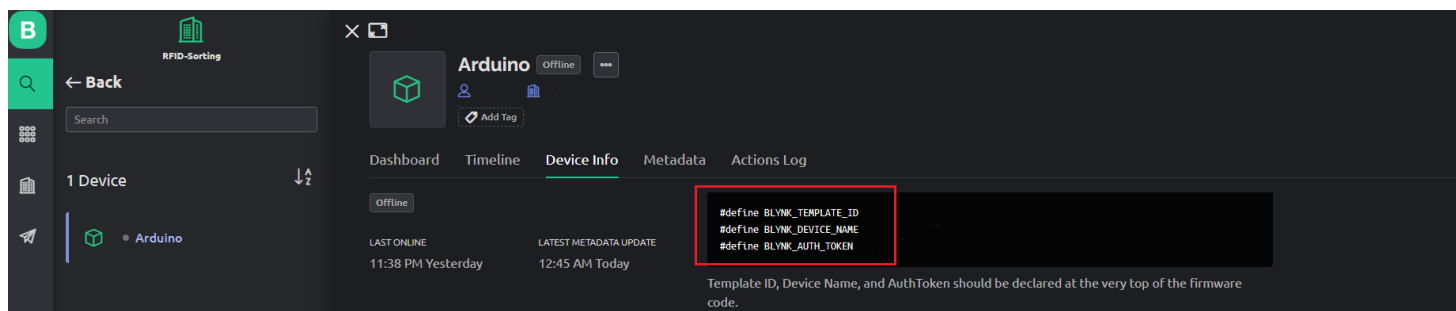
Rys 4.5 Tworzenie templatki w aplikacji BLYNK.

Po utworzeniu szablonu w zakładce „Search” należy dodać nowe urządzenie, pierwsze okienko tworzenie urządzenia można zobaczyć na rysunku 4.6, należy użyć opcji tworzenia urządzenia z poprzednio wykreowanego szablonu.

The screenshot shows the 'New Device' screen. At the top, the title 'New Device' is displayed. Below it, the instruction 'Choose a way to create new device' is shown. There are three main options, each in a card: 'From template' with a 3x3 grid of dots icon, 'Scan QR code' with a QR code icon, and 'Manual entry' with a keyboard icon. Below these cards, there is a hand icon pointing to the cards with the text 'Point on the cards to see instructions'. At the bottom right, there is a 'Cancel' button.

Rys 4.6 Tworzenie nowego urządzenie BLYNK.

Następnie, przechodzimy do nowo wstawionego urządzenia, a następnie do zakładki „Device Info” znajdziemy tam informacje o tym, co należy uwzględnić w kodzie Arduino, aby Blynk mógł połączyć się z mikrokontrolerem, informacje te wskazano w czerwonej ramce na rysunku 4.7,



Rys 4.7 Informacje konfiguracyjne BLYNK.

Po uwzględnieniu powyższych danych w kodzie, należy wprowadzić fragment odpowiedzialny za konfigurację. Należy dodać poniższą linijkę, która będzie odpowiadać za rozpoczęcie konfiguracji z Blynkiem.

```
Blynk.begin(auth, "blynk.cloud", 80);
```

Do dopełnienia konfiguracji pomiędzy Arduino a Blynikiem należy dodać jeszcze jedną linijkę kodu, na początku głównej pętli programowej. Funkcja ta ma na celu przetworzenie przychodzących poleceń i wykonania porządkowania połączenia Blynk.

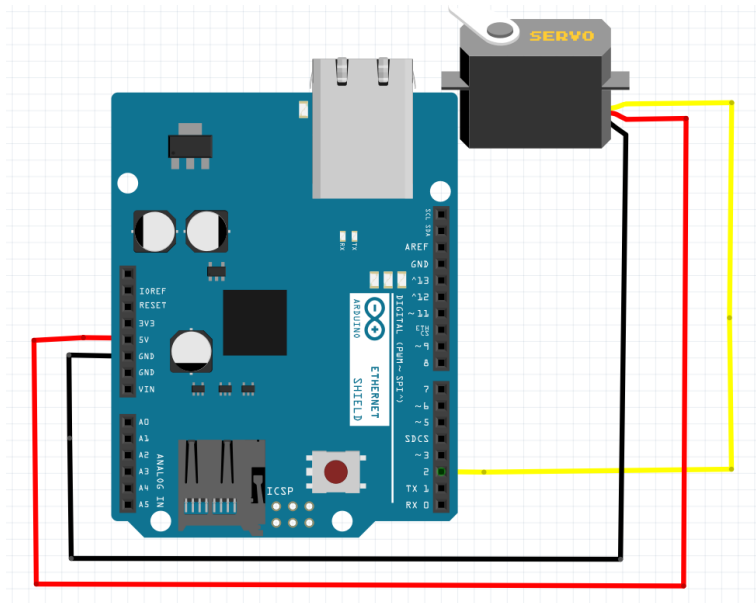
```
Blynk.run();
```

4.1.3. Sterowanie serwomechanizmem SG90

Aby zrealizować funkcjonalność serwomechanizmu, należy najpierw połączyć go zgodnie ze schematem przedstawionym na Rysunku 4.8. Następnie należy skorzystać z odpowiedniej biblioteki. Biblioteka ta pozwala na sterowanie 12 serwomechanizmami za pomocą standardowej płytki Arduino. Należy jednak pamiętać, że korzystanie z tej biblioteki uniemożliwia użycie funkcji "analogWrite" z pinami numer 9 i 10 na standardowej płytce Arduino.

W pierwszej kolejności została użyta funkcja „myservo.attach”, pozwala ona oznaczyć pin Arduino, do którego podłączony jest serwomechanizm. Natomiast funkcja „myservo.write” Określa kąt w stopniach, o jaki ma się obrócić serwomechanizm.

```
//Konfiguracja Serva
#include <Servo.h>
myservo.attach(2);
myservo.write(90);
```



Rysunek 4.8 Podłączanie serwomechanizmu SG90.

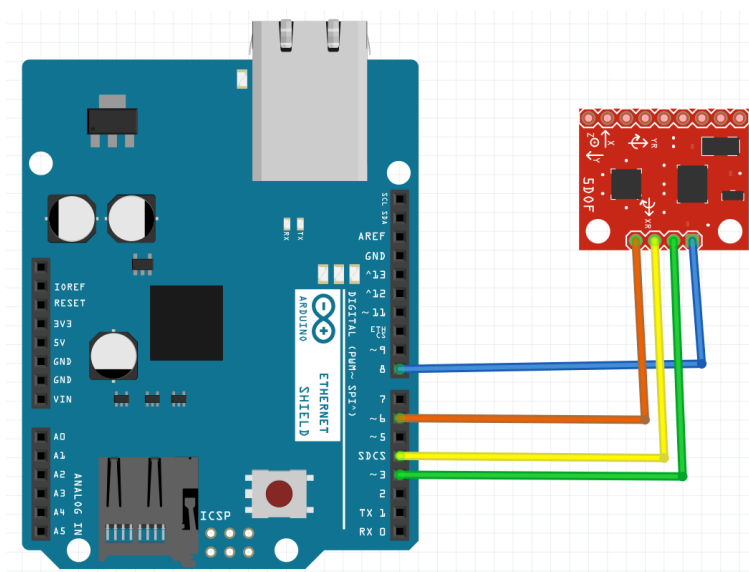
Funkcja realizująca sterowanie serwomechanizmem ma postać:

```
void Servoo () //Sterownie Serwem
{
int stopnie;
if (RFID_data=="1234567891234567")
{
stopnie=45;
myservo.write(stopnie);
Blynk.virtualWrite(V3, stopnie);
Odliczanie();
}
else if (RFID_data=="1234567891234568")
{
stopnie=135;
myservo.write(stopnie);
Blynk.virtualWrite(V3, stopnie);
Odliczanie();
}
else
{
stopnie=90;
myservo.write(stopnie);
}
}
```

Powyższa funkcja najpierw sprawdza dane odczytane przez RFID. Sprawdzania zachodzi, poprzez warunki „if” gdzie sprawdzany jest, czy odczytana wartość zgadza się z tą zapisaną w warunku, jeżeli ze znacznika RFID odczytano, dane odpowiadające któremuś z warunków następuje wystawienie serwomechanizmu do danego konta oraz wysłanie kąta wystawiania do Blynka. Jeżeli żaden warunek nie został spełniony, kąt serwomechanizmu zostaje ustawiony na 90 stopni.

4.1.4. Sterowanie silnikiem krokowym 28BYJ-48

Do realizacji funkcjonalności silnika krokowego użyte zostanie sterowanie 4 cewkami zaprezentowanymi na Rysunku 3.9. Pierwszą czynnością jest podłączenie sterownika ULN2003, który to jest sterownikiem silnika krokowego wedle schematu z Rysunku 4.9.



Rysunek 4.9 Podłączanie sterownika ULN2003.

Po podłączeniu sterownika w kodzie definiujemy piny do obsługi silnika krokowego, kod wykonujący tą czynność ma postać:

```
#define P1 6
#define P2 4
#define P3 3
#define P4 8
```

Aby, zrealizować funkcje obrotów silnika w prawo należy, w odpowiedniej kolejności załączać i wyłączać cewki w silniku. Sterowanie obrotami będzie realizowane za pomocą zmniejszania/zwiększania opóźnienia pomiędzy załączeniami. Kod realizujący powyższe założenie przyjmuje postać:

```

void motor_P(int dly)  // Podprogram do sterowania prawymi obrotami silnika
{
    digitalWrite (P1, HIGH);
    delay(dly);
    digitalWrite (P1, LOW);
    digitalWrite (P2, HIGH);
    delay(dly);
    digitalWrite (P2, LOW);
    digitalWrite (P3, HIGH);
    delay(dly);
    digitalWrite (P3, LOW);
    digitalWrite (P4, HIGH);
    delay(dly);
    digitalWrite (P4, LOW);
}

```

Analogicznie należy postąpić w przypadku lewych obrotów, jedyną różnicą będzie odwrotna kolejność załączania cewek.

Do załączenia, jak i sterowania silnikiem zostanie użyty Blynk, gdzie poprzez sprawdzanie wartości zmiennej otrzymywanej z Blynka będą sprawdzane warunki w pętli głównej czy silnik załączyć, a jeśli tak to, jakie ma obroty oraz jakie występuje opóźnienie w załączaniu.

Kod realizujący powyższe założenie przyjmuje postać:

```

//odczyt zadanego stanu silnika (czy jest włączony czy nie)
BLYNK_WRITE(V4)
{
    Stan_silnika = param.asInt();
}

// odczyt zadanych obrotów silnika (prawe czy lewe)
BLYNK_WRITE(V5)
{
    Obroty = param.asInt();
}

//odczyt zadanej prędkości silnika
BLYNK_WRITE(V6)
{
    P_silnika = param.asInt();
}

```

```

if(Stan_silnika==0)
{
    Stan_silnika=0;
}
else{
    if(Obroty==0)
    {
        motor_P(P_silnika);
    }
    else
    {
        motor_L(P_silnika);
    }
}
}

```

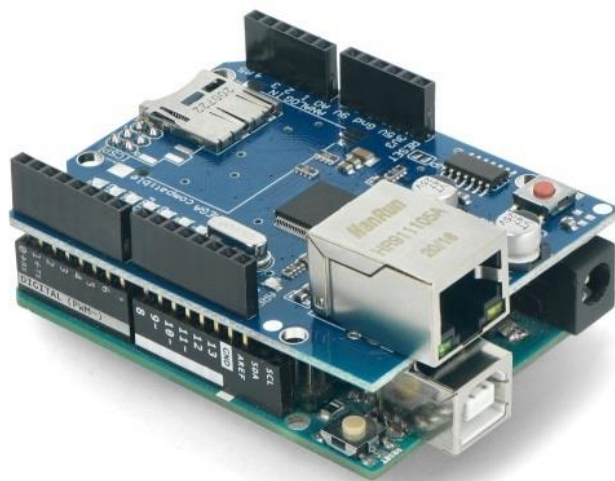
4.1.5. Obsługa Ethernet Shielda

Montaż Ethernet shielda na Arduino polega na, zamontowaniu go na Arduino, poprawnie zamontowany Shield pokazano na Rysunku 4.10. Do jego obsługi zostanie wykorzystana biblioteka „Ethernet.h”, której implementacja wygląda następująco:

```

#include <Ethernet.h>
#include <BlynkSimpleEthernet.h>

```



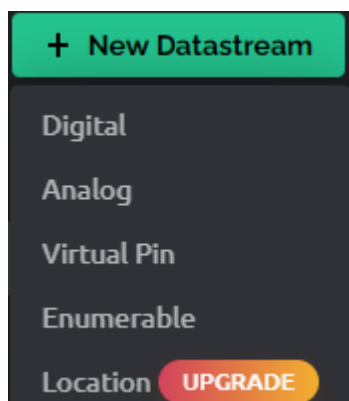
Rysunek 4.10 Zamontowanie Ethernet Shielda.

4.2. Utworzenie interfejsu w BLYNKU

4.2.1. Utworzenie pinów komunikacyjnych w zakładce „Datastream”

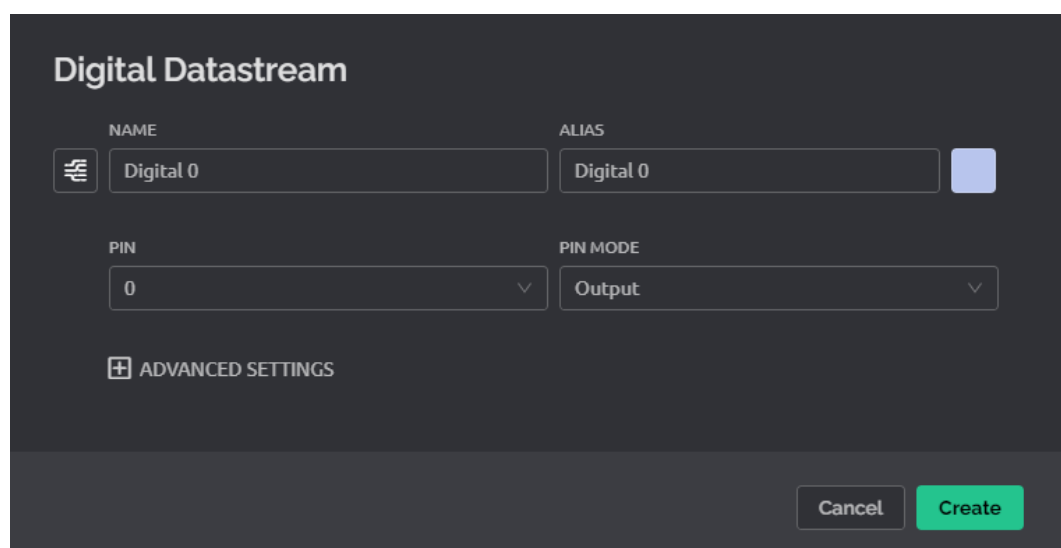
Do realizacji komunikacji pomiędzy Arduino a Blynkiem używane są wirtualne piny, które same w sobie nie wpływają na mikrokontroler, istnieje też wariant pinów komunikacyjnych, który steruje fizycznymi wyjściami płytki. Przechowywane są w nich informacje, które mogą zostać odczytane w kodzie programu oraz pokazane na „Dashbordzie” w Blynku.

W celu utworzenie pinów komunikacyjnych należy otworzyć Blynka w przeglądarce internetowej lub na urządzeniu mobilnym, następnie w zakładce „Templates” otwieramy wybrany szkic, po jego wybraniu w zakładce „Datastream” włączmy tryb edycji, a następnie klikamy w „+ New Datastream”, po tym zostanie wyświetlony wybór, jakiego rodzaju pin zostanie utworzony, lista pinów pokazano na Rysunki 4.11.



Rysunek 4.11 Możliwe do utworzenie piny komunikacyjne.

Z podanych możliwości zostanie wybrany pin wirtualny, następnie wybrana zostanie jego nazwa, jaki pin to jest oraz jakiego jest typu (wejści/wyjście). Po wpisaniu wszystkich informacji po wciśnięciu „Create” zostanie utworzony wirtualny pin komunikacyjny.



Rysunek 4.12 Tworzenie wirtualnego pina w aplikacji Blynk.

Do odczytania wartości pinów wirtualnych w programie służy funkcja „BLYNK.WRITE”, a jak przykładem jest:

```
BLYNK_WRITE (V3)
{
  int kat = param.asInt();
  myservo.write(kat);
}
```

Fragment ten odczytuje wartość trzeciego pinu wirtualnego, to zmiennej „kat”.

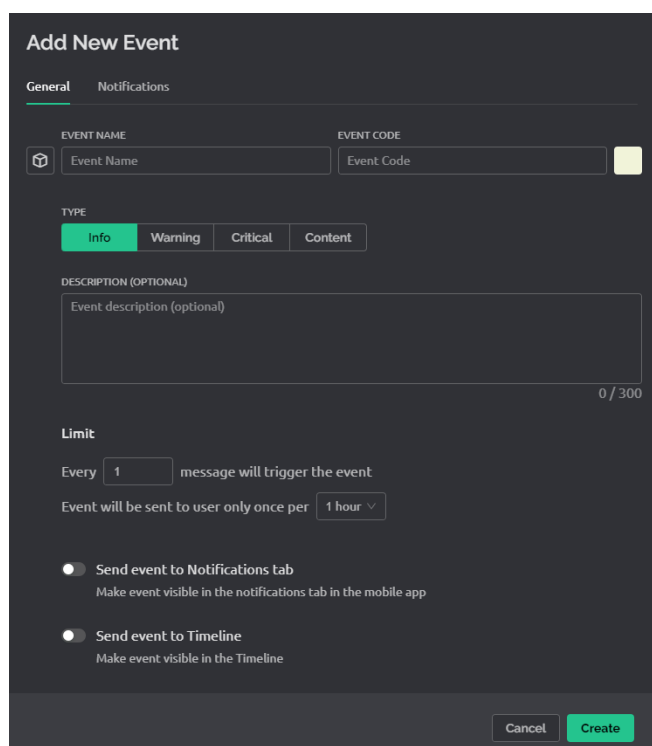
4.2.2. Utworzenie eventów w zakładce „Events”

Zdarzenia (z ang. Events), służą do śledzenia, rejestrowania i pracy z ważnymi zdarzeniami, które mają miejsce w urządzeniu. Zdarzenia są również wykorzystywane do powiadomień, które mogą być wysyłane przez e-mail, dostarczane jako powiadomienia push na smartfon użytkownika lub wysyłane jako SMS.

Zdarzenie możemy podzielić na 3 rodzaje:

- zdarzenia systemowe (z ang. System Events) – Domyślne zdarzenia platformy Blynk,
- zdarzenia niestandardowe (z ang. Custom Events) – Zdarzenia, które możesz stworzyć i skonfigurować do swoich potrzeb,
- zdarzenia informacyjne (z ang. Content Events): Wydarzenia informacyjne, które są pokazywane osobno w aplikacji,

Żeby utworzyć wydarzenie, należy przejść: Blynk strona główna -> Templates -> Utworzony szablon -> Events



Rysunek 4.12 Tworzenie eventu aplikacji Blynk.

Po wejście do zakładki z wydarzeniami, wchodząc to trybu edycji, wcisnąć należy „+ Add new Event”. Po tej operacji pokaże się okno tworzenie nowego zdarzenia pokazane na Rysunku 4.12, w oknie należy wybrać rodzaj eventu, czy ma wysyłać powiadomienie (i limit takowych powiadomień), po wykonaniu tych czynności, tworzymy nowe zdarzenie, klikając „Create”.

Obsługa zdarzeń w programie jest dość prosta, należy umieścić poniższą funkcję razem z wykryciem błędu lub innej czynności, która ma zostać zarejestrowana.

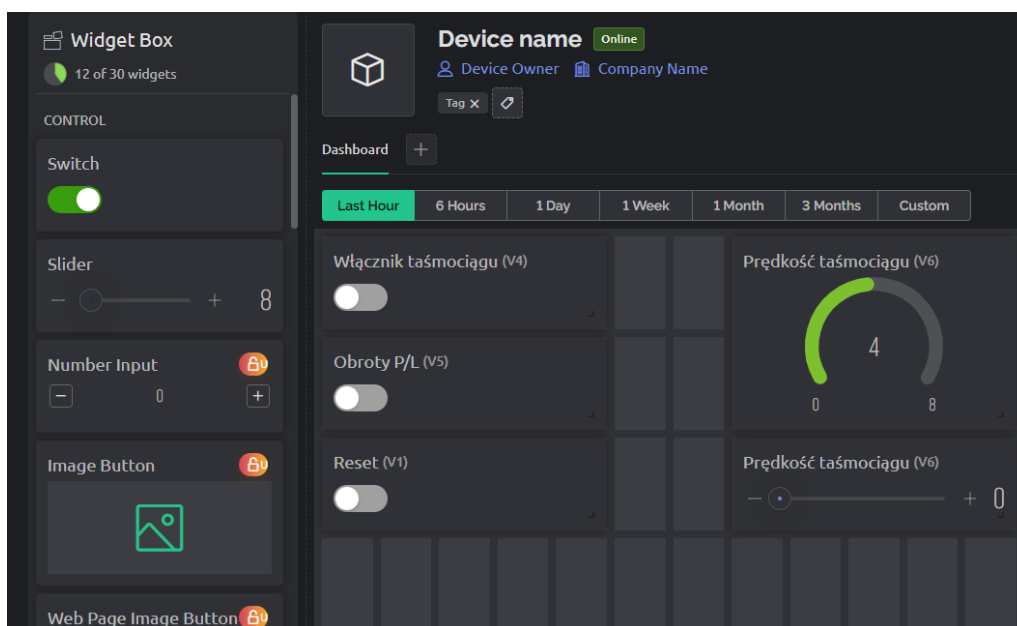
```
Blynk.logEvent("nieudany_odczyt_");
```

4.2.3. Utworzenie interfejsu w zakładce „Web Dashboard”

Tablica internetowa jest jedna z dwóch możliwości sterowania urządzeniem (druga to aplikacja mobilna).

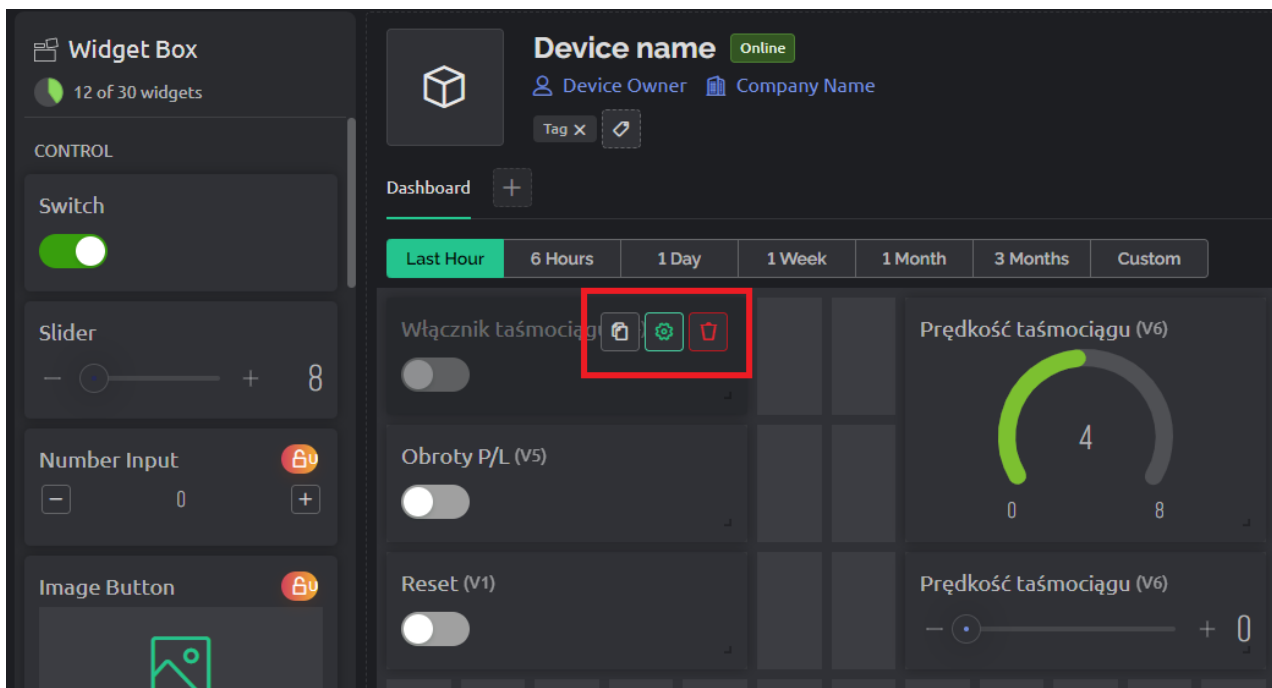
W celu utworzenia interfejsu należy przejść: Blynk strona główna -> Templates -> Utworzony szablon -> Web Dashboear,

Po wykonaniu tych czynności należy włączyć tryb edycji, następstwem włączenia tego trybu powinna być dostępna możliwość dodawanie widżetów, tak ja to zaprezentowano na Rysunku 4.13

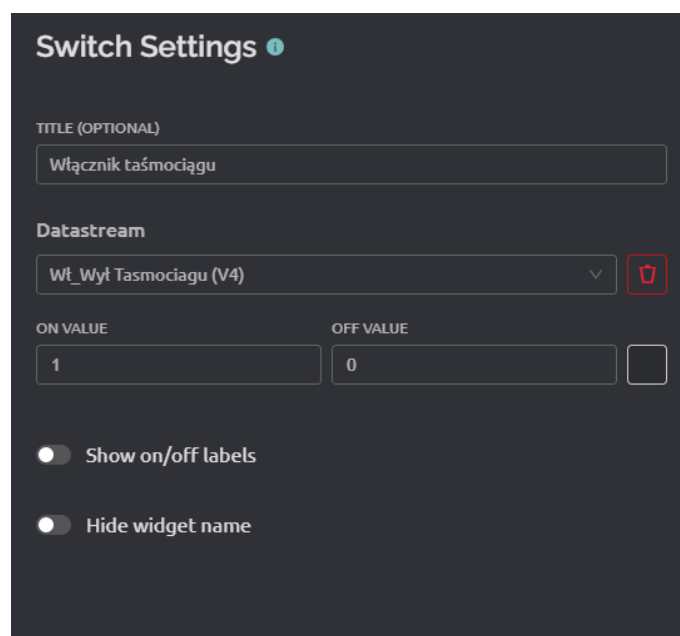


Rysunek 4.13 Tworzenie eventu aplikacji Blynk.

W celu konfiguracji widżetów należy najechać kursorem na dany widżet, po tej czynności powinny pojawić się trzy opcje takie jak przedstawione na Rysunku 4.14. Kolejną czynnością jest kliknięcie zębatki, po jej kliknięciu powinno pojawić się okno takie jak na Rysunku 4.15, w oknie tym mamy możliwość przypisania do widżeta pinu sygnałowego, zmiany nazwy czy jak w przypadku przełącznika jest stan aktywny i nieaktywny.



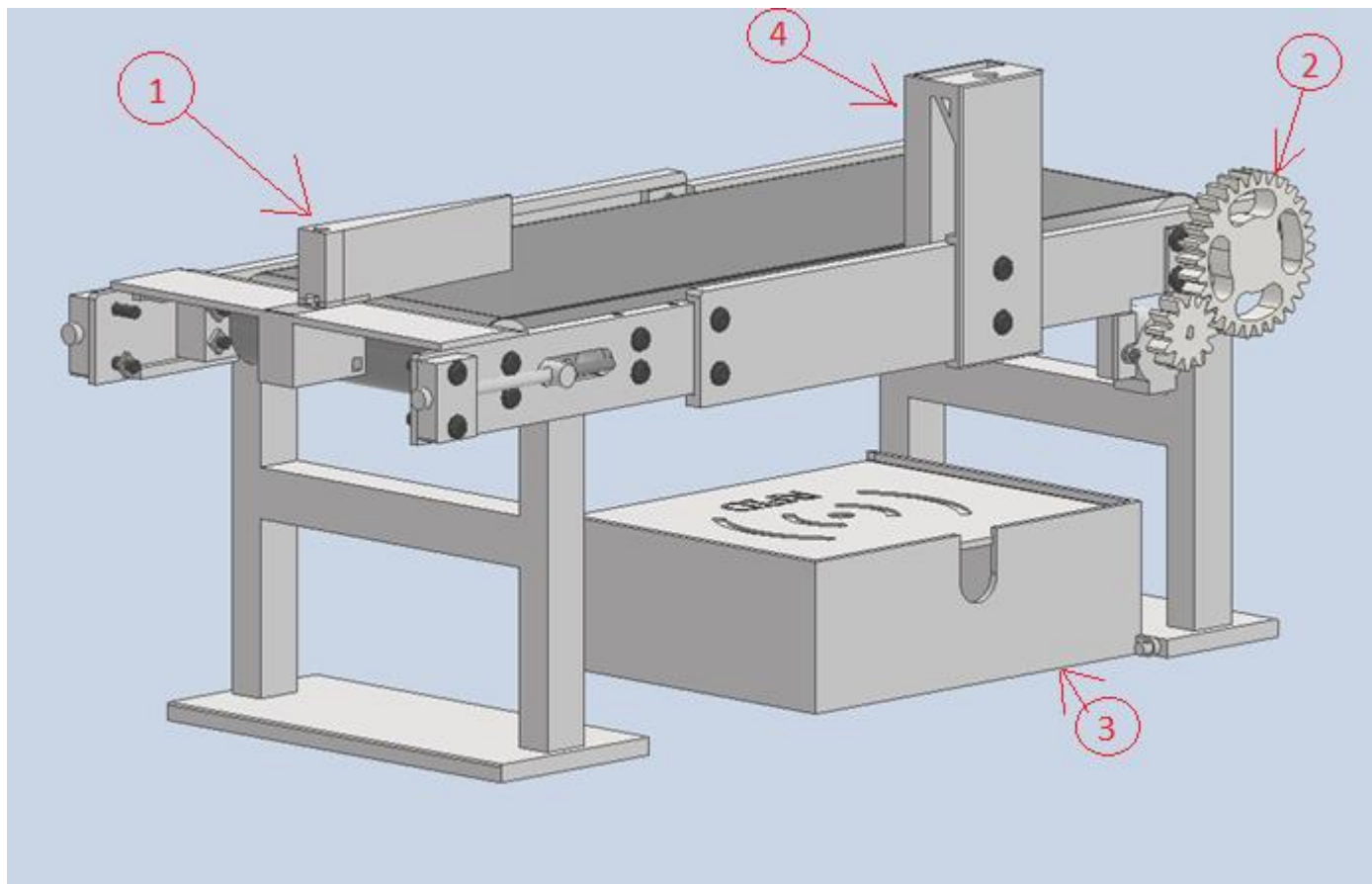
Rysunek 4.14 Wskazanie edycji widżetu.



Rysunek 4.15 Tworzenie eventu aplikacji Blynk.

4.3.Zaprojektowanie modelu 3D w programie Inventor Studio

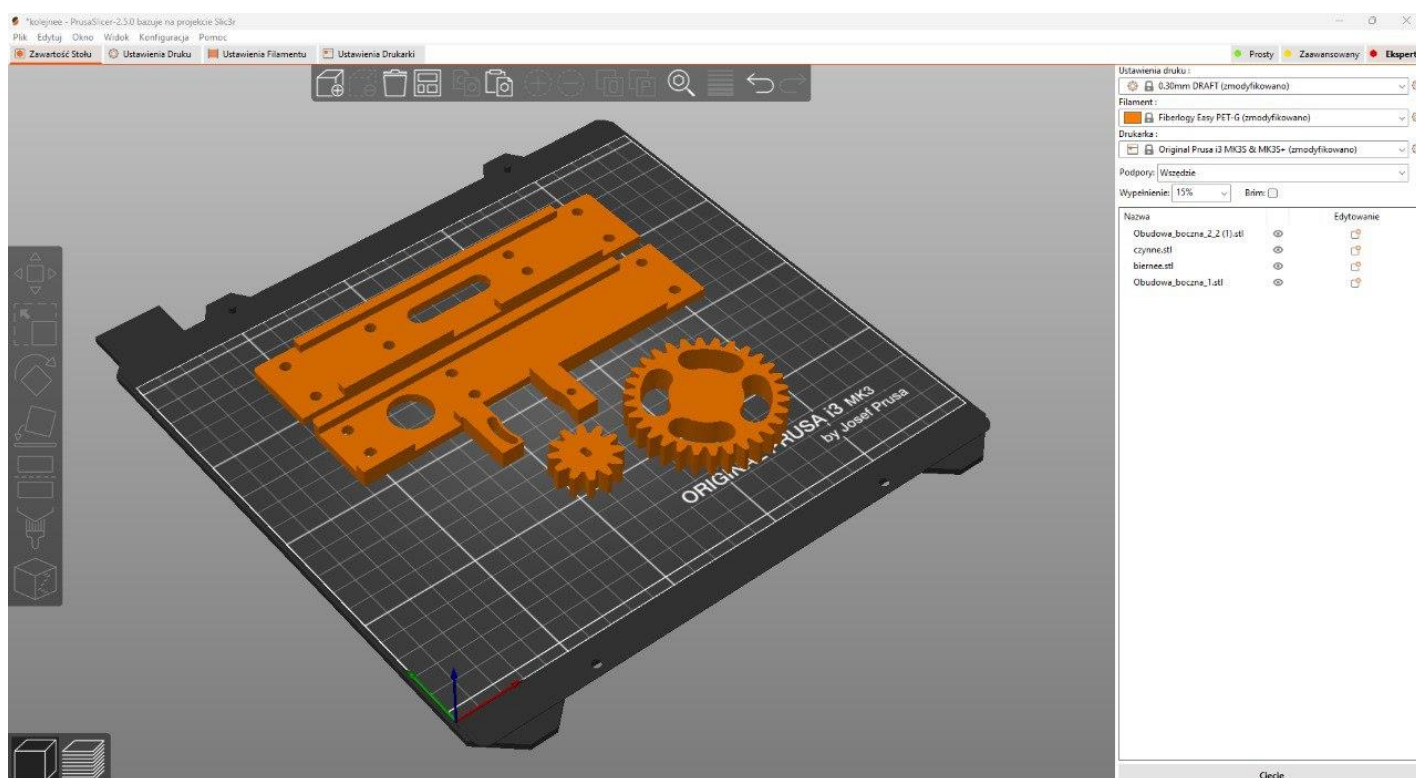
W celu symulacji przyjętego rozwiązania dotyczącego automatycznego kategoryzowania przesylek w programie Inventor Studio zaprojektowano model 3d taśmociągu z wykorzystanie elementów pokazanych na rysunku 4,16. Końcowe wymiary modelu 3d to 469mm x 189mm x 239mm. Model 3d składa się z trzydziestu jeden elementów, które zostały zapisane w formacie „ipt”, a następnie przekonwertowane do „stl” w celu ich późniejszego wydruku na drukarce 3d.



Rysunek 4.16 Makieta wykonana w programie Inventor, 1-serwomechanizm, 2- silnik krokowy, 3- Arduino + Ethernt Shield, 4 -czytnik RC522.

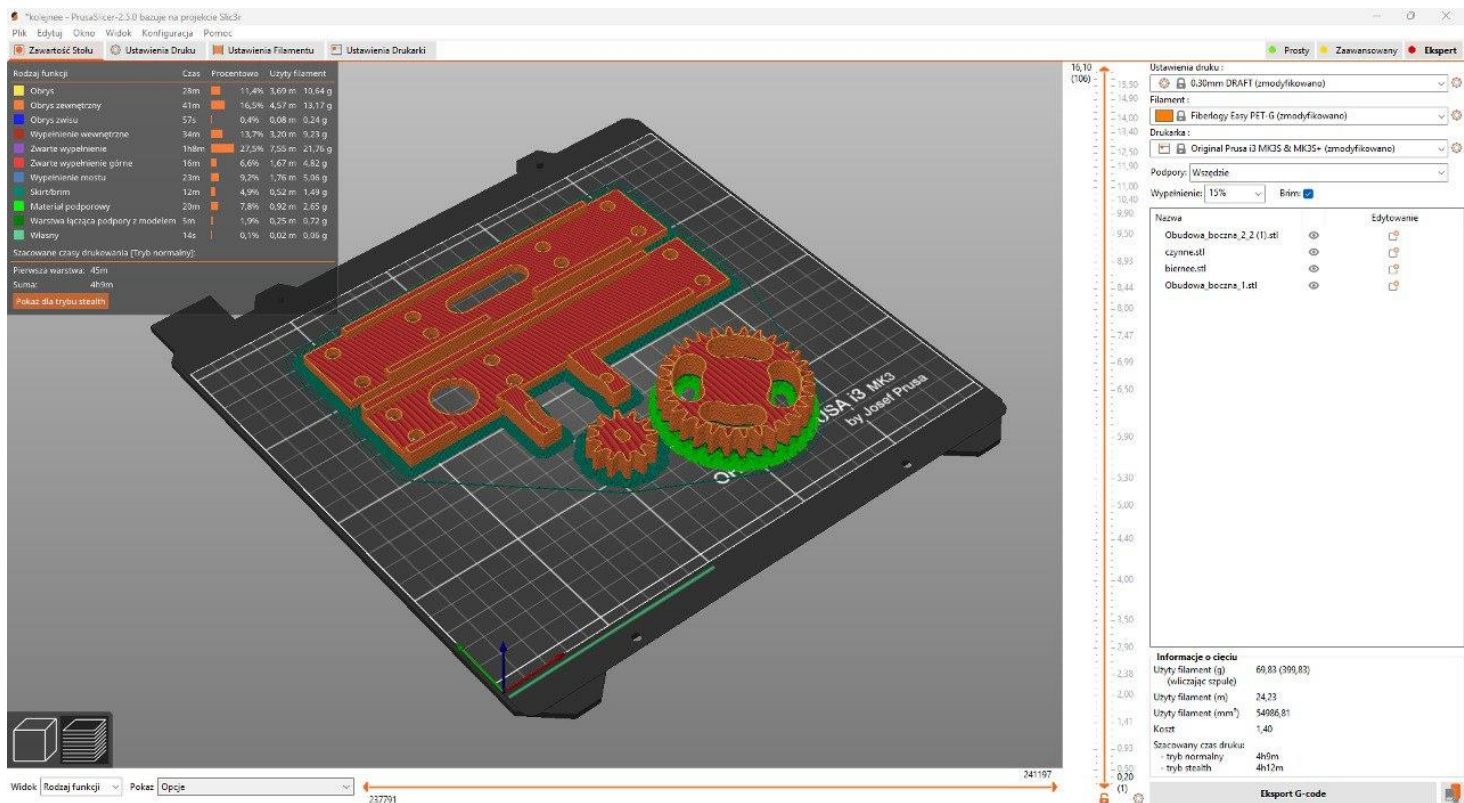
4.4. Wydruk elementów na drukarce 3D

Wydruk modelu 3d był wykonywany przy pomocy drukarki Prusa Mk3. Wydruk polegał na wgraniu do oprogramowania producenta „PrusaSlicer” wcześniej przygotowanych modeli 3d w programie Inventor. Widok niektórych z nich zaprezentowano na Rysunku 4.17.



Rysunek 4.17 Elementy modelu w programie PrusaSlicer.

Do wydruku elementów użyto filamentu rodzaju „PETG”. Program producenta przekonwertowuje otrzymane pliki w formacie „stl” na „G-code”. Na rysunku 4.18 możemy zauważyć modele z Rysunku 4.17, lecz po konwersji na „G-code” oraz z automatycznie wygenerowanymi podporami ułatwiającymi wydruk. Na wydruk wszystkich elementów wykorzystano 1.5 kg filamentu oraz przeznaczono na sam proces wydruku ok. stu godzin roboczych,

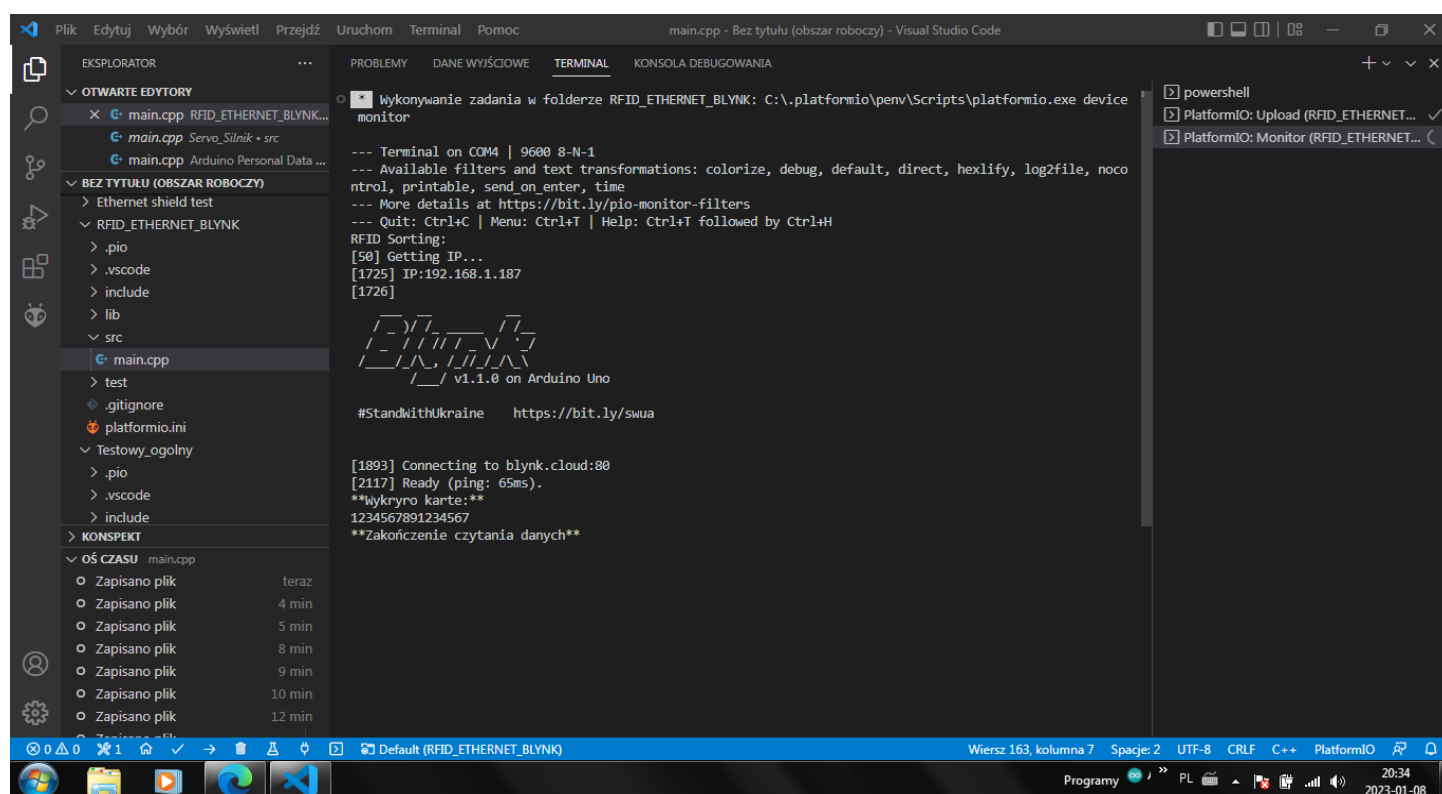


Rysunek 4.18 Elementy modelu przygotowane do wydruku w PrusaSlicer.

ROZWIĄZANIA

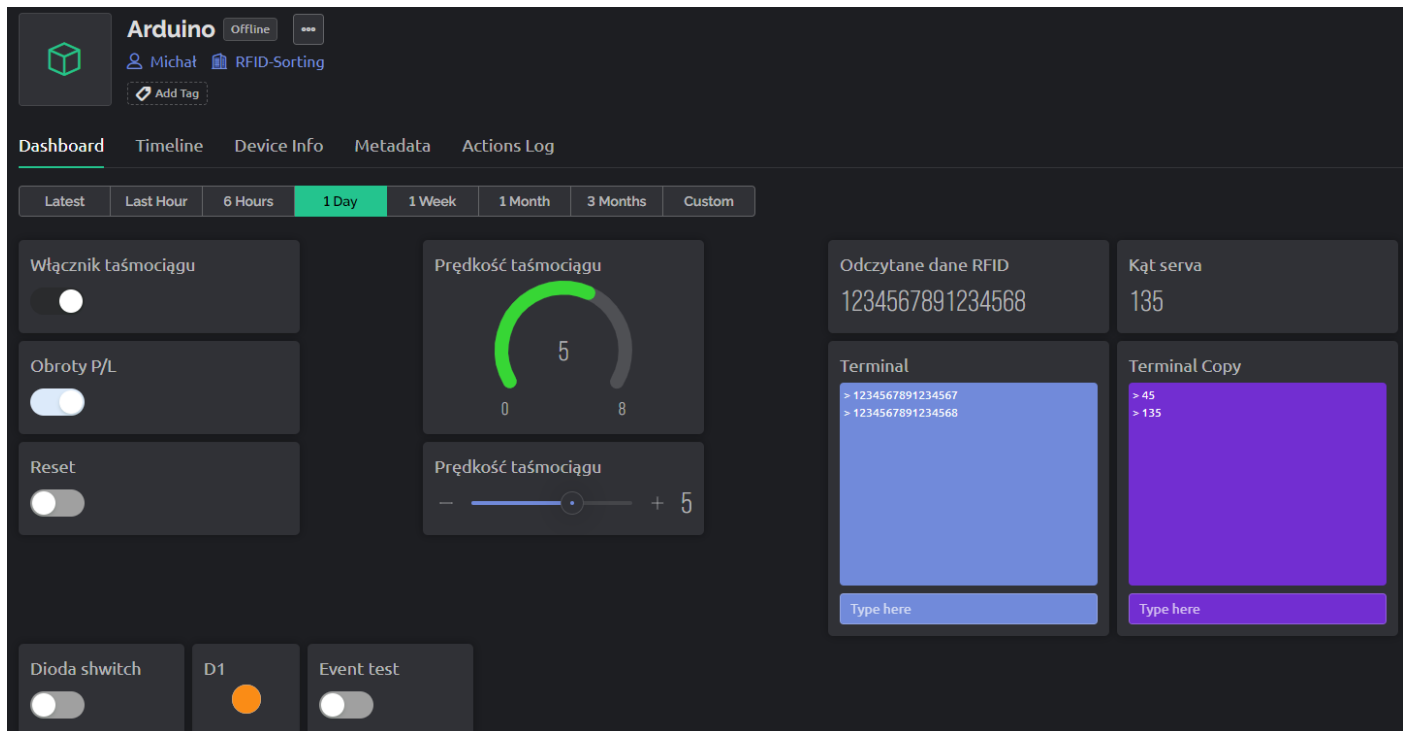
Testowanie funkcjonalności oprogramowania rozpoczęto od testu w programie Microsoft Visual Code, przy pomocy rozszerzenia programu występującego pod nazwą Platformio, uzyskamy prostą możliwość wgrania programu do mikrokontrolera, a następnie jego weryfikację.

Po wgraniu programu i otwarciu podglądu w „serial monitor” program działa zgodnie z założeniami, po zeskanowaniu karty RFID otrzymamy wynik z Rysunku 5.1. Program uruchamia się poprawnie, a następnie następuje procedura łączenia się z serwerem Blynk. Po której znajduje się komunikat o poprawnych wykryciu znacznika RFID oraz pozytywny komunikat informujący o poprawnym zakończeniu procesu czytania danych ze znacznika.



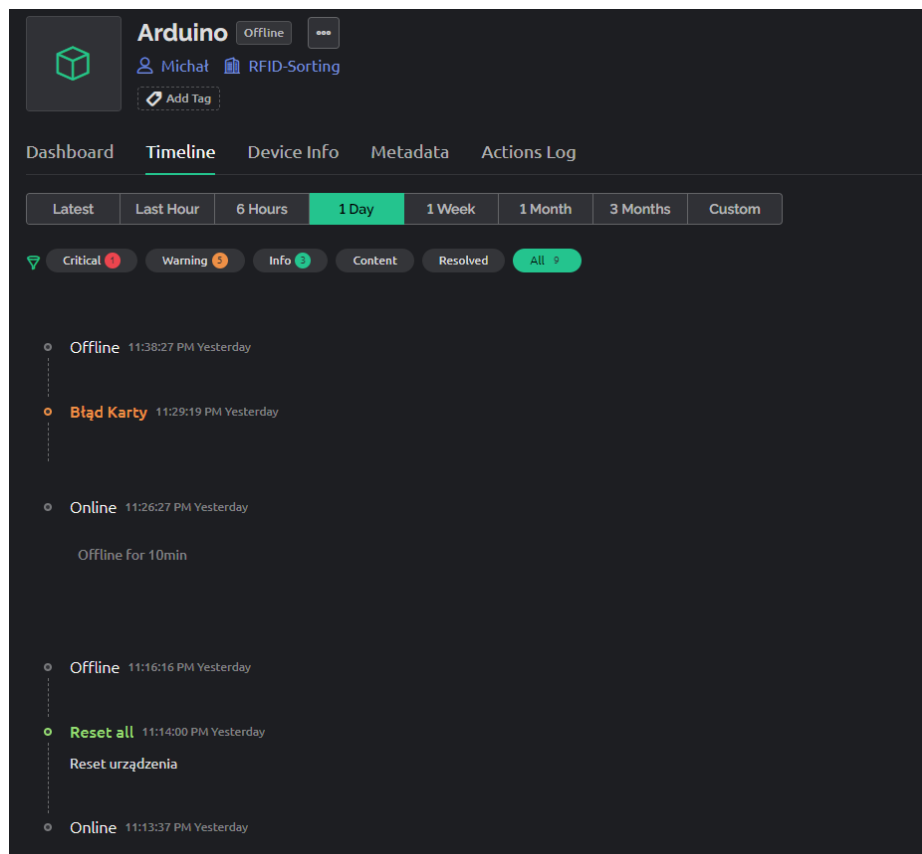
Rysunek 5.1 Program Microsoft Visual Code z działającym kodem rozwiązania.

Kolejnym etapem sprawdzanie poprawności działania programu jest przejście do panelu w aplikacji Blynk. W oknie aplikacji przedstawionym na Rysunku 5.2 można zauważyć, że program poprawnie skomunikował się poprzez wirtualne piny komunikacyjne, potwierdzeniem tego jest wyświetlane w terminalach danych, które pokrywają się z programem Code Studio.

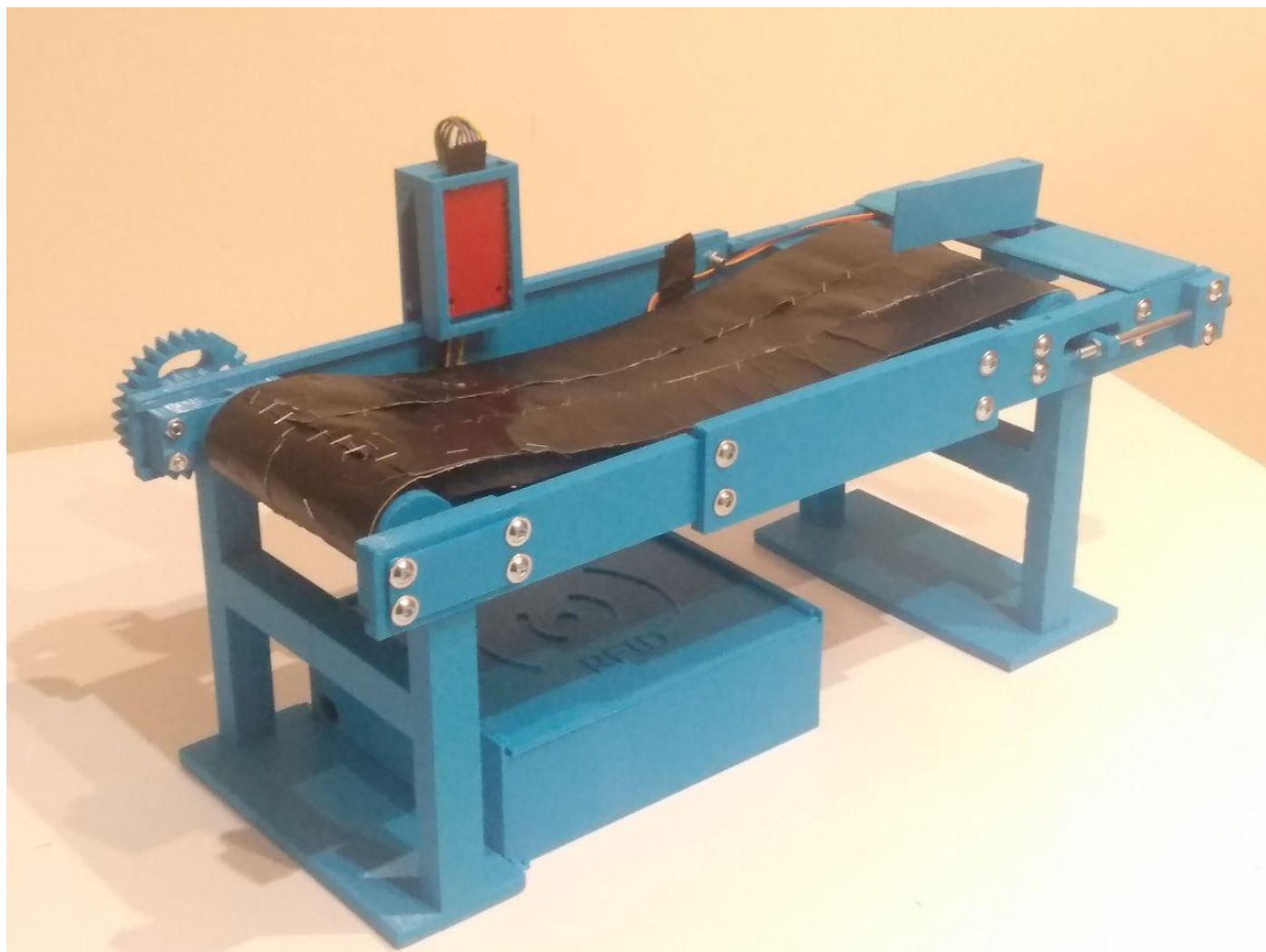


Rysunek 5.2 Dashboard Blynka z odczytami RFID oraz sterowaniem sortującym .

W czasie pracy programu wystąpił błąd „Błąd Karty”, zastał on wyświetlony na osi czasu zdarzeń, zaprezentowanej na Rysunku 5.3. Pojawienie się alertów informuje o poprawności kodu oraz synchronizacji z Blynkiem.



Rysunek 5.3 Logi zdarzeń z działania rozwiązania.



Rysunek 5.4 Gotowa makieta symulacyjna.

Podsumowując na podstawie powyżej przedstawionych danych możemy zakładać że zaprojektowane rozwiązanie działa poprawnie. Wszystkie testy zostały wykonane w połączeniu z makietą z rysunku 5.4, co świadczy o tym, że rozwiązanie jest również funkcjonalne w aspekcie fizycznym

6. PODSUMOWANIE

Celem wykonywanej pracy było zaprojektowanie oraz wykonanie programu dla płytki Arduino, jak i fizycznej makiety symulującej przyjęte rozwiązanie dotyczące automatycznego kategoryzowania przesylek. Oprogramowanie oraz makietę miały na celu zaprezentowanie poprawności rozwiązania automatycznego kategoryzowania przesylek. Oprogramowanie zostało napisane przy pomocy języka programowania „C++”, z wykorzystaniem bibliotek ułatwiających kontrolę nad podzespołami dobranymi do realizacji projektu. Do użytych bibliotek możemy zaliczyć: „MFRC522.h” (RFID), „Ethernet.h” (Ethernet Shield), „Servo.h” (serwomechanizm). W ramach części programowej wykonano również połączenie pomiędzy Arduino a aplikacją Blynk, połączenie to miało na celu umożliwienie zdalnego sterowania oraz odczytywania danych z czytnika RFID. Część modelowa odpowiadająca za fizyczną makietę została wykonana przy pomocy takich programów jak: „Autodesk Inventor Profesjonal” czy „PrusaSlicer”.

Opracowane rozwiązanie posiada dalsze możliwości rozwojowe – przykładowo poprzez dodanie dodatkowych czujników wykrywających kolizję przesylek, lub implementację dodatkowych linii, lub napędów. Przyspieszyłoby to proces kategoryzowania. Inną możliwością jest wymiana obecnie wykorzystywanych podzespołów na odpowiedniki o lepszych parametrach technicznych. Konstrukcja bazowa makiety nie wymaga dodatkowych ulepszeń, ponieważ spełnia ona swoją funkcję pokazową. W przypadku zastosowania rozwiązania przemysłowo wymagana będzie stworzenie nowej makiety pod konkretne potrzeby nabywcy rozwiązania.

Bibliografia

- [1] „Technologia RFID – co to jest?”, *RFID Polska*
<https://www.rfidpolska.pl/technologia-rfid-co-to-jest/> (dostęp 5 styczeń 2023)
- [2] A. Maćkowiak, K. Sieczkarek, P. Kaźmierczak, A. Wojciechowski „Selektywna bramka RFID”, *logistyka.net.pl*.
https://www.logistyka.net.pl/bank-wiedzy/item/download/79101_f60a75534220d9b7f41fd9a32c1b9f22
(dostęp 05 styczeń 2023)
- [3] „RFID w Magazynie”, *HAXON*
<https://haxon.com.pl/rfid-w-magazynie/> (dostęp 5 styczeń 2023)
- [4] „RFID w logistyce i w magazynie – Automatyczna identyfikacja w systemach logistycznych” *RFID Solutions*
<https://rfid.zone/rfid-w-logistyce/> (dostęp 5 styczeń 2023)
- [5] „What is RFID? How It Works? Interface RC522 RFID Module with Arduino”, *Last Minute Engineers*
<https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/> (dostęp 5 stycznia 2023)
- [6] Z. Mazur, H. Mazur „Automatyczna identyfikacja obiektów w technologii RFID”, *MEDIA Logistyka*.
https://www.logistyka.net.pl/bank-wiedzy/logistyka/item/download/77941_a29aba1053c67945c50bab95b8e431f8 (dostęp 5 stycznia 2023)
- [7] „RFID – co to jest?”, *IBCS Poland*.
<https://rfid.com.pl/technologia-rfid-co-to-jest/> (dostęp 5 stycznia 2023)
- [8] A. Bernasińska „Technologia RFID - co to jest, jak działa i gdzie znajduje zastosowanie?” *UNICARD*
<https://www.unicard.pl/news-technologia-rfid-co-to-jak-dziala> (dostęp 5 stycznia 2023)
- [9] „Czytniki RFID (3 rodzaje i najlepsze zastosowania)”, *PWSK*.
<https://www.pwsk.pl/czytnik-rfid/> (dostęp 5 stycznia 2023)
- [10] „Anteny RFID - jakie wybrać?”, *IBCS Poland*.
<https://rfid.com.pl/anteny-rfid-jaka-wybrac/> (dostęp 5 stycznia 2023)
- [11] „Systemy Parkingowy RFID – bezobsługowe systemy kontroli Wjazdu”, *RFID Polska*.
<https://www.rfidpolska.pl/bezobslugowy-system-parkingowy/> (dostęp 6 stycznia 2023)
- [12] Krzycho „BLYNK – opis ogólny systemu”, *Blynk.pl - fanklub*.
<https://blynk.pl/blynk-opis-ogolny-systemu/> (dostęp 6 stycznia 2023)
- [13] „Standardy RFID: LF (125kHz), HF (13,56MHz) i UHF (860-956 MHz)”, *RFID Polska*.
<https://www.rfidpolska.pl/standardy-rfid/> (dostęp 6 stycznia 2023)
- [14] „Introduction”, *docs.blynk.io* .

<https://docs.blynk.io/en/> (dostęp 6 stycznia 2023)

[15] „Arduino UNO - co warto wiedzieć?”, *FORBOT*.

<https://forbot.pl/blog/leksykon/arduino-uno> (dostęp 6 stycznia 2023)

[16] „Arduino UNO – idealna płytką kontrolera”, *Akademia Nattigo*.

http://akademia.nettigo.pl/arduino_uno/ (dostęp 6 stycznia 2023)

[17] „Arduino Ethernet Shield”, *Mouser*.

https://www.mouser.com/catalog/specsheets/a000056_datasheet.pdf (dostęp 6 stycznia 2023)

[18] „Serwo – obsługa serwomechanizmów i zasada działania”, *Elektroweb*.

<https://blog.elektroweb.pl/serwo-obsługa-serwomechanizmow-i-zasada-dzialania/> (dostęp 6 stycznia 2023)

[19] „Serwomechanizm modelarski TowerPro SG90”, *NETTIGO*.

<https://nettigo.pl/products/serwomechanizm-modelarski-towerpro-sg90> (dostęp 6 stycznia 2023)

[20] J. Tyburski „Tani, uniwersalny silnik krokowy 28BYJ-48 z wbudowaną przekładnią”, *LPORTAL*.

<https://elportal.pl/blog/silniki-i-serwo/817-tani-uniwersalny-silnik-krokowy-28byj-48-z-wbudowana-przekladnia> (dostęp 6 stycznia 2023)

[21] „Czytnik RFID RC522”, *Botland*.

<https://botland.com.pl/moduly-i-tag-i-rfid/10574-czytnik-rfid-rc522-1356mhz-spi-karta-i-brelok-czerwony-iduino-me138-5903351241168.html> (dostęp 6 stycznia 2023).

[22] „ASCII Text to Hex Code Converter”, *RapidTables*.

<https://www.rapidtables.com/convert/number/ascii-to-hex.html> (dostęp 7 stycznia 2023).

[23] „The Software Servo Library”. *arduino.cc*.

<https://playground.arduino.cc/> (dostęp 7 stycznia 2023).

[24] „Ethernet Shield W5100 dla Arduino z czytnikiem kart microSD”, *Botland*.

<https://botland.com.pl/arduino-shield-komunikacja/3779-ethernet-shield-w5100-dla-arduino-z-czytnikiem-kart-microsd-5904422359461.html>

STRESZCZENIE PROJEKTU INŻYNIERSKIEGO

Wykorzystanie technologii RFID w automatycznym kategoryzowaniu przesylek

Autor: **Michał Danek**

Promotor: **dr inż. Paweł Penar**

Słowa kluczowe: RFID, Arduino, Blynk, RC522, Serwomechanizm, Silnik krokowy, Makieta.

Streszczenie:

W pracy zaprezentowano proces programowania Arduino oraz projektowania makiety 3d do wybranego rozwiązania, którego celem jest automatyczne kategoryzowanie przesylek poprzez wykorzystanie technologii RFID oraz aplikacji Blynk. Praca składa się z części teoretycznej w której omówiono zawartości tegoż rozwiązania oraz praktycznej gdzie sprawdzane jest czy opracowania teoretycznie działa w praktyce. W części teoretycznej została ogólnie przybliżona technologia RFID oraz opis z specyfikacją poszczególnych komponentów wykorzystanych w projekcie. Część praktycznej głównie opierała się o opracowanie modelu 3D w programie a następnie ich wydruk na drukarce 3D po końcowy montaż. Testy działania rozwiązania wykonane na makiecie potwierdziły poprawność działania rozwiązania.

DIPLOMA THESIS ABSTRACT

Use of RFID technology in automatic shipment categorisation

Author:: **Michał Danek**

Promoter: **dr inż. Paweł Penar**

Key words: RFID, Arduino, Blynk, RC522, Servo, Stepper motor, Mock-up.

Abstract:

The paper presents the process of Arduino programming and designing a 3d mockup for a selected solution, which aims to automatically categorise shipments by using RFID technology and the Blynk application. The paper consists of a theoretical part, where the intricacies of the solution are discussed, and a practical part, where it is verified that the theoretical development works in practice. The theoretical part provided a general overview of RFID technology and a description with specifications of the various components used in the project. The practical part was mainly based on the development of a 3D model in the software, followed by its printing on a 3D printer and final assembly. Functional tests performed on a mock-up confirmed the correctness of the solution.