
		Instytut Informatyki Politechniki Śląskiej Zespół Mikroinformatyki i Teorii Automatów Cyfrowych			
Rok akademicki:	Rodzaj studiów*: SSI/NSI/NSM	Przedmiot (Języki Asemblerowe/SMiW):	Grupa	Sekcja	
2020/2021	SSI	SMiW	4	2	
Imię:	Oskar	Prowadzący: OA/JP/KT/GD/BSz/GB	JP		
Nazwisko:	Kuchejda				
<h2>Raport końcowy</h2>					
Temat projektu: <div style="text-align: center; margin-top: 100px;"> <h1>Tracker słoneczny poruszający panelem słonecznym w kierunku światła.</h1> </div>					
Data oddania: dd/mm/rrrr			09.02.2021		

Temat projektu, opis założeń, opis funkcji urządzenia.	3
Analiza zadania.	4
Specyfikacja wewnętrzna urządzenia.	6
3.1 Schemat blokowy i ideowy urządzenia.	6
3.2 Opis funkcji poszczególnych bloków układu oraz szczegółowy opis działania ważniejszych elementów układu.	7
3.3 Schemat montażowy obejmujący projekt płytki drukowanej, rozmieszczenie elementów na płytce oraz rozmieszczenie elementów urządzenia w obudowie.	9
3.4 Lista elementów.	12
3.5 Algorytm oprogramowania urządzenia.	13
3.6 Opis wszystkich ważniejszych zmiennych w programie.	15
3.7 Opis funkcji wszystkich procedur, występujących w programie.	15
3.8 Opis interakcji oprogramowania z układem elektrycznym.	17
3.9 Szczegółowy opis działania ważniejszych procedur.	19
Specyfikacja zewnętrzna urządzenia.	20
4.1 Opis funkcji elementów sterujących urządzeniem.	20
4.2 Opis funkcji elementów wykonawczych.	20
4.3 Opis reakcji oprogramowania na zdarzenia wewnętrzne.	20
4.4 Instrukcja obsługi urządzenia.	20
4.5 Schemat okablowania.	21
4.6 Opis montażu układu.	22
4.7 Opis sposobu programowania układu.	22
4.8 Opis sposobu uruchamiania oraz testowania układu.	22
4.9 Zdjęcia z gotowego układu.	23
Wnioski i uwagi z przebiegu pracy.	26
Wnioski końcowe.	26
Literatura.	26

1. Temat projektu, opis założeń, opis funkcji urządzenia.

Temat projektu:

Tracker słoneczny poruszający panelem słonecznym w kierunku światła.

Założenia projektu:

Urządzenie obraca panel słoneczny, w kierunku najintensywniejszego źródła światła. Panel słoneczny ładuje akumulator, z którego można ładować inne urządzenia. Wyświetlacz pokazuje aktualny poziom naładowania baterii oraz aktualne napięcie generowane przez panel słoneczny.

Opis funkcji urządzenia:

Panel słoneczny obracany jest pionowo oraz poziomo. Energia generowana przez panel jest magazynowana w akumulatorze, a stan naładowania oraz wydajność ogniwa jest prezentowany na wyświetlaczu. Stan baterii wyrażony jest w procentach, natomiast wydajność ogniwa w woltach oraz natężeniu. Pomiar prądów i napięć został zrealizowany przy użyciu specjalnych układów, opisanych w dalszej części raportu. Urządzenie umożliwia ładowanie telefonów oraz innych elektronicznych przedmiotów, dzięki wejściu USB.

2. Analiza zadania.

- **Silnik krokowy:**

Do obracania panelem wybrano silnik krokowy NEMA 17 - 17HS4401 zapewniający precyzję ruchów. Jako sterownik silnika wybrano A4988. Silnik obraca panelem w poziomie. Ogniwo zostało zamontowane na kolumnie wykonanej z aluminiowych mocowań. Wysokość kolumny wynosi około 30 cm. Łączna waga kolumny wraz z panelem wynosi około 2kg.

- **Serwo:**

Wybrano serwo AR-1201MG zapewniające 1.17Nm. Umieszczone na czubku kolumny serwo porusza panelem w pionie przy użyciu linki która jest zawijana bądź odwijana w zależności od podnoszenia lub obniżania ogniwa.

- **Panel słoneczny:**

Ogniwo monokrystaliczne o wymiarach 190mm x 430mm, mocy 10W, wadze 1.2kg i maksymalnym napięciu 20v, ładuje akumulator prądem o maksymalnym natężeniu 0.5A.

- **Wyświetlacz:**

LCD 2x16 znaków, wyświetlający użytkownikowi aktualne napięcie i natężenie generowany przez panel oraz stopień naładowania baterii w procentach.

- **Przetwornice:**

Zasilanie urządzeń 5v zostało zrealizowane przy użyciu przetwornicy XY-3606. Zastosowana przetwornica LM317 12v zostało użyta do zasilania silnika krokowego.

- **Rezonator kwarcowy:**

HC49 o częstotliwości 16 Mhz, wybrany w celu dokładnego odmierzenia czasu, między pomiarami.

- **Tranzystory:**

Tranzystor IRLZ44N służący do wyłączania zasilania, w przypadku kiedy akumulator jest naładowany w 100% oraz tranzystor A2222 otwierający, bądź zamykający kanał tranzystora mosfet.

- **Akumulator:**

W celu poprawnego działania urządzenia, wybrano akumulator AGM VRLA od firmy GreenCell o napięciu 12V i pojemności 10Ah.

- **Miernik prądu i napięcia:**

Do wykonywania pomiarów wykorzystano dwa układy o nazwie INA219, które zapewniły bardzo dobrą dokładność pomiarów. W celu komunikacji dwóch urządzeń z mikrokontrolerem użyto multipleksera TCA9548A.

- **Rezystory, kondensatory, listwy zaciskowe, goldpiny oraz dioda:**

Rezystory w celu ograniczenia prądów w układzie, oraz kondensatory do filtrowania napięcia, oraz dioda zabezpieczająca ogniwo przed prądem z akumulatora. Za pomocą listw zaciskowych możliwe było doprowadzenie prądów z akumulatora lub ogniwa. Piny umożliwiły komunikację z silnikiem, serwem i programatorem.

- **Mikrokontroler:**

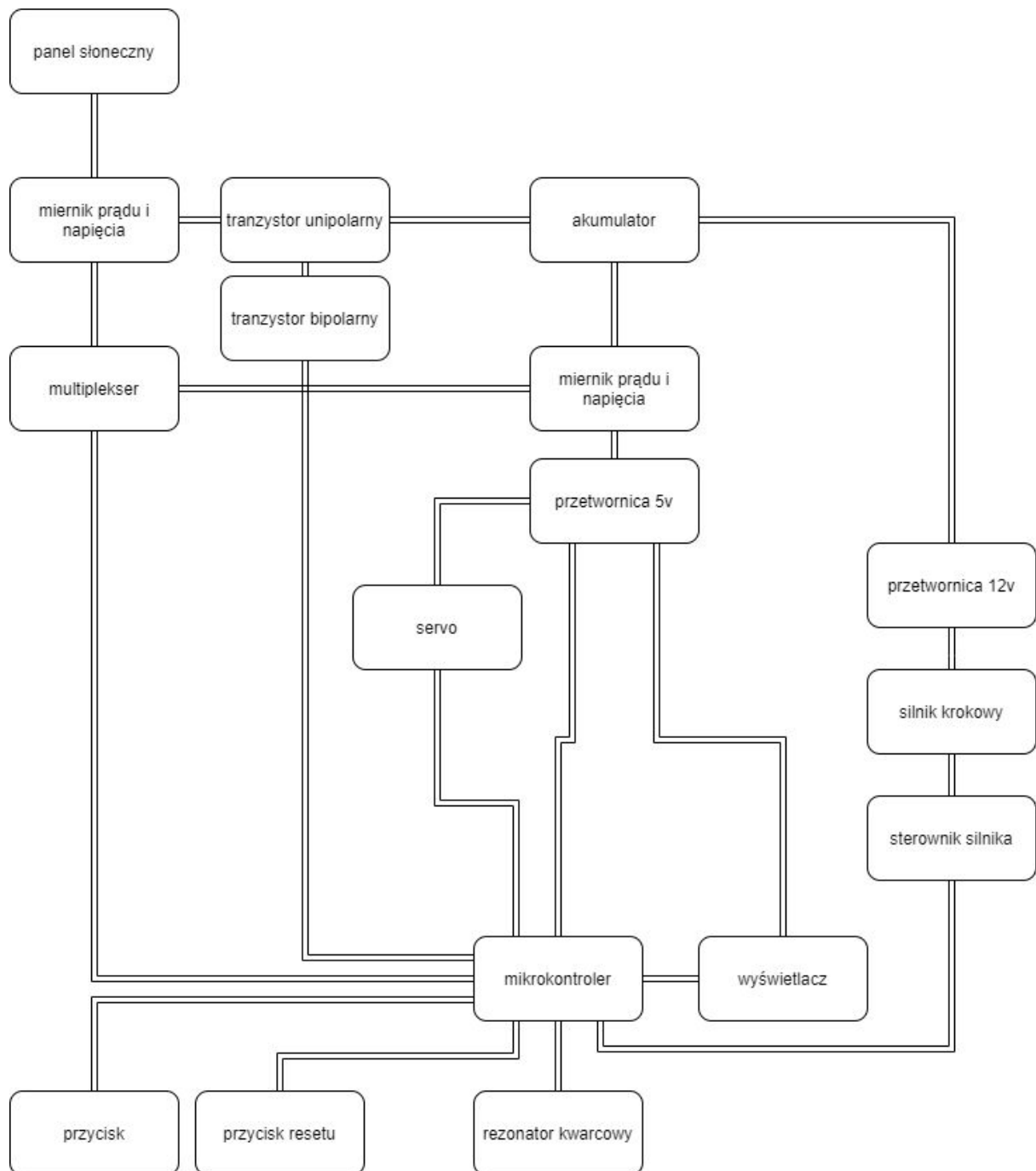
Na początku szukano mikrokontrolera posiadającego 2 kanały ADC, jednak wraz z ewolucją projektu zadanie należące do tych pinów przejęły dwa niezależne układy INA219, które były w stanie dokładniej zmierzyć napięcia i prądy w układzie. Mikrokontroler musiał obsługiwać wiele urządzeń, więc szukano takiego którego liczba pinów programowalnych była większa lub równa 15.

Przeglądano mikrokontrolery takie jak PIC16LF873A-I/SP, PIC18F2455-I/SP oraz PIC16F877A-I/P jednak zostały one odrzucone z powodu zbyt małej ilości pamięci Flash (poniżej 32 KB). Efektem szukania odpowiedniego mikrokontrolera było znalezienie trzech konkurencyjnych rozwiązań PIC16F18857-E/SP, STM8S105K6B6 oraz ATMEGA328P. Spośród wymienionych modeli, najwięcej pamięci flash posiada PIC16F18857-E/SP, 56 KB. Pozostałe modele mają po 32 KB. Wszystkie trzy mikrokontrolery mają odpowiednią ilość pinów programowalnych, jak i liczbę kanałów ADC, jednak z powodu problemów z dostępnością PIC16F18857-E/SP oraz STM8S105K6B6, został wybrany mikrokontroler ATMEGA328P, który spełnia warunki projektu.

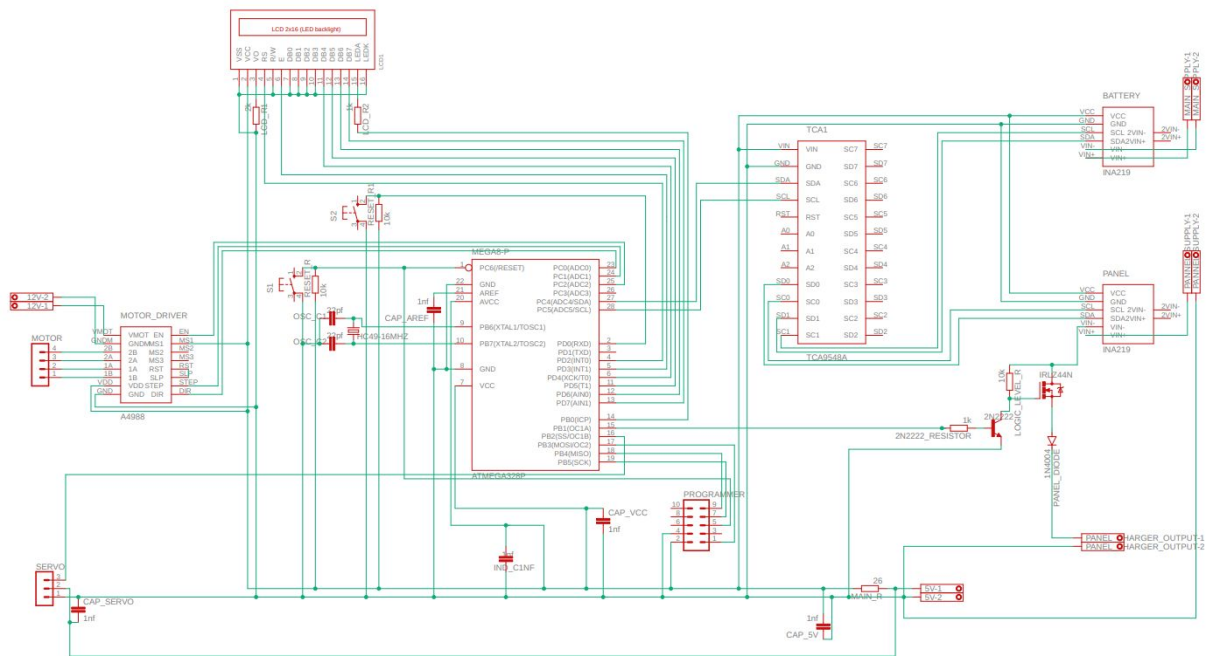
3. Specyfikacja wewnętrzna urządzenia.

3.1 Schemat blokowy i ideowy urządzenia.

Schemat blokowy:

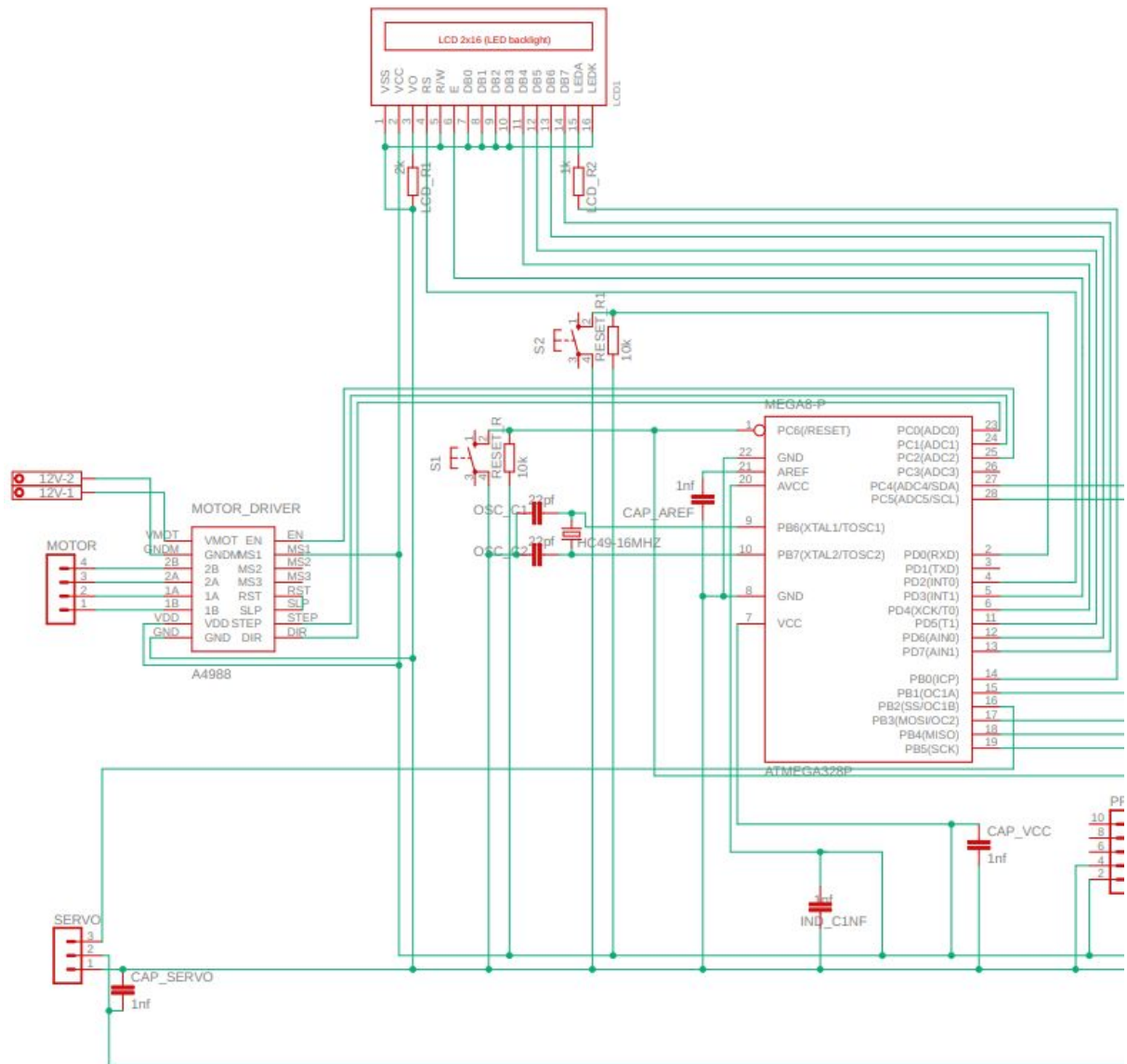


Schemat ideowy:

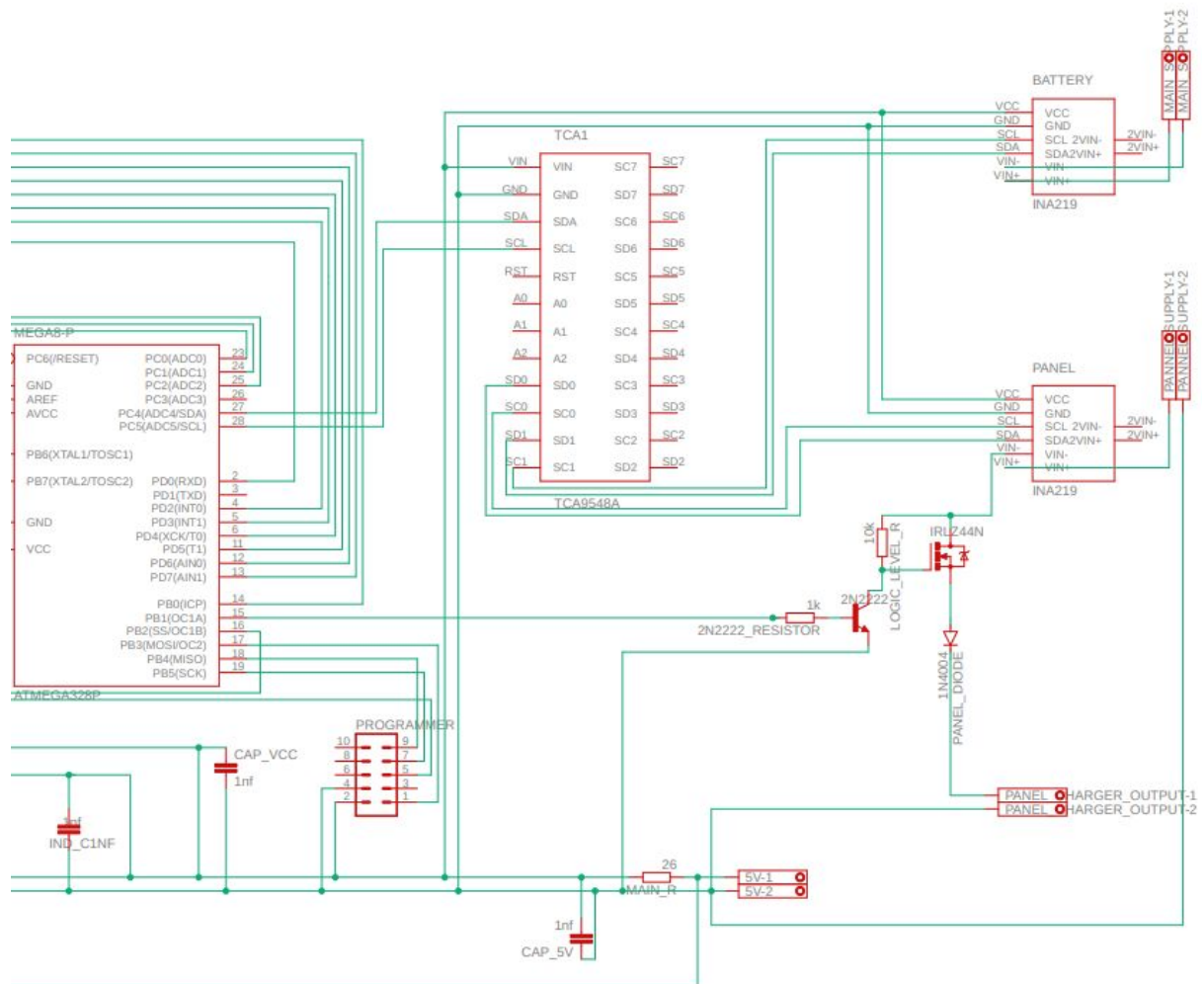


Poniżej zaprezentowano ten sam schemat, tylko przybliżony.

Lewa strona, podłączenie silnika krokowego, serwomechanizmu, wyświetlacza oraz przycisków.



Prawa strona, podłączenie multipleksera, modułów pomiarowych i przełącznika od ładowania akumulatora.



3.2 Opis funkcji poszczególnych bloków układu oraz szczegółowy opis działania ważniejszych elementów układu.

panel słoneczny - generuje energię, która zostaje użyta do ładowania akumulatora.

miernik prądu i napięcia - wykonuje pomiary prądów i napięć w układzie, dzięki czemu mikrokontroler może wywoływać potrzebne funkcję w celu poprawnej pracy urządzenia.

multiplekser - umożliwia przełączanie się pomiędzy dwoma miernikami prądu i napięcia w celu monitorowania parametrów akumulatora lub określenia napięcia i prądu generowanego przez panel słoneczny.

tranzystor unipolarny - umożliwia zamykanie bądź otwieranie obwodu zależnie od stanu naładowania akumulatora.

tranzystor bipolarny - umożliwia kontrolowanie tranzystora unipolarnego przy użyciu mikrokontrolera.

akumulator - zasila układ, oraz gromadzi energię wygenerowaną przez ogniwo.

przetwornica 5v - dostarcza prąd o odpowiednim napięciu dla mikrokontrolera, wyświetlacza, serwa oraz sterownika silnika krokowego.

przetwornica 12v - dostarcza prąd o odpowiednim napięciu dla silnika krokowego.

serwomechanizm - umożliwia obracanie panela słonecznego w górę lub w dół.

silnik krokowy - umożliwia obracania panela słonecznego w lewo lub w prawo.

sterownik silnika - umożliwia komunikację mikrokontrolera z silnikiem, zapewnia prawidłową pracę silnika krokowego.

wyświetlacz - prezentuje wyniki pomiarów użytkownikowi.

rezonator kwarcowy - jest zegarem dla mikrokontrolera.

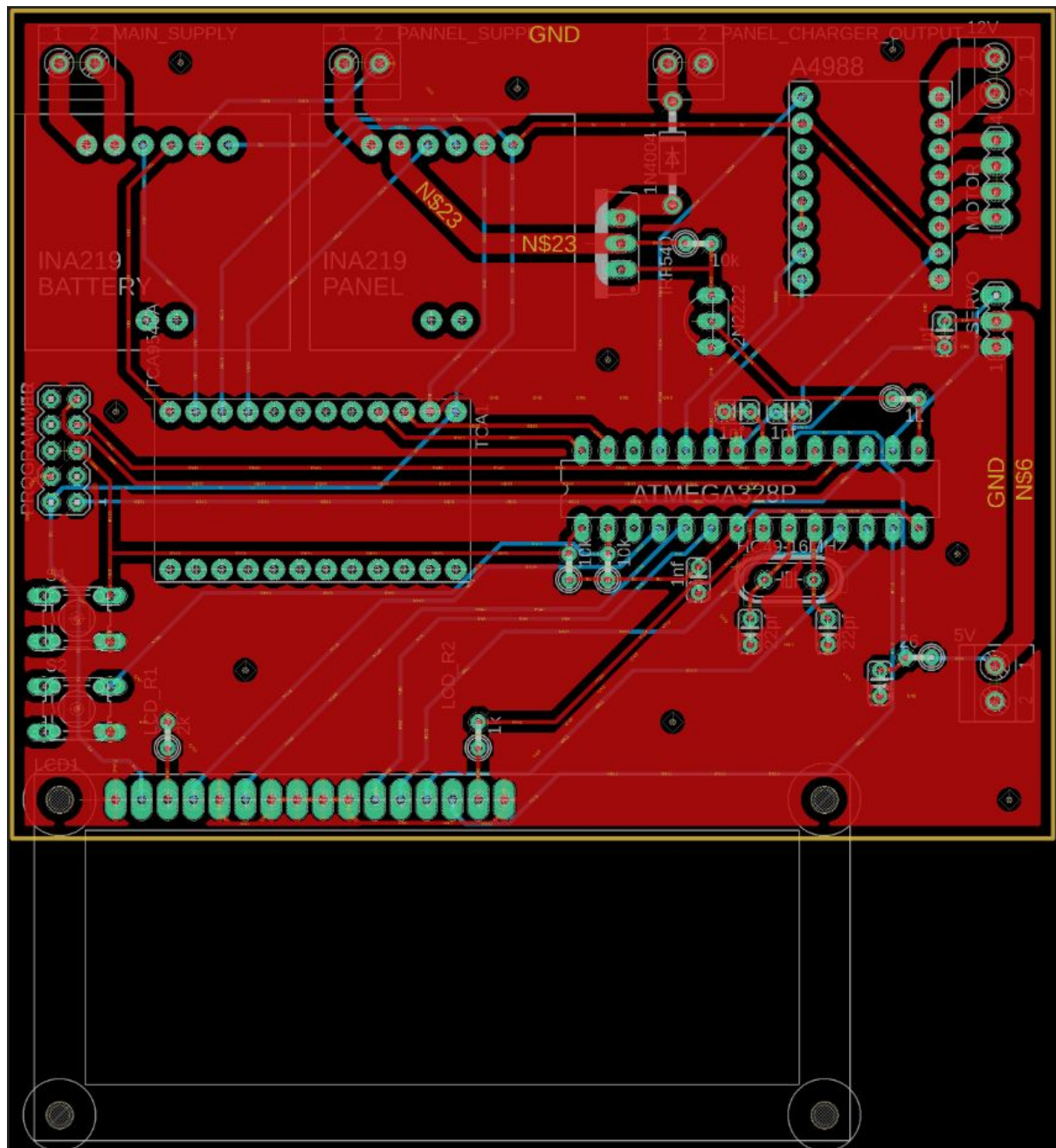
przycisk - umożliwia obsługę urządzenia, po jego naciśnięciu użytkownikowi prezentowane są informacje widoczne na wyświetlaczu.

przycisk resetu - umożliwia resetowanie urządzenia.

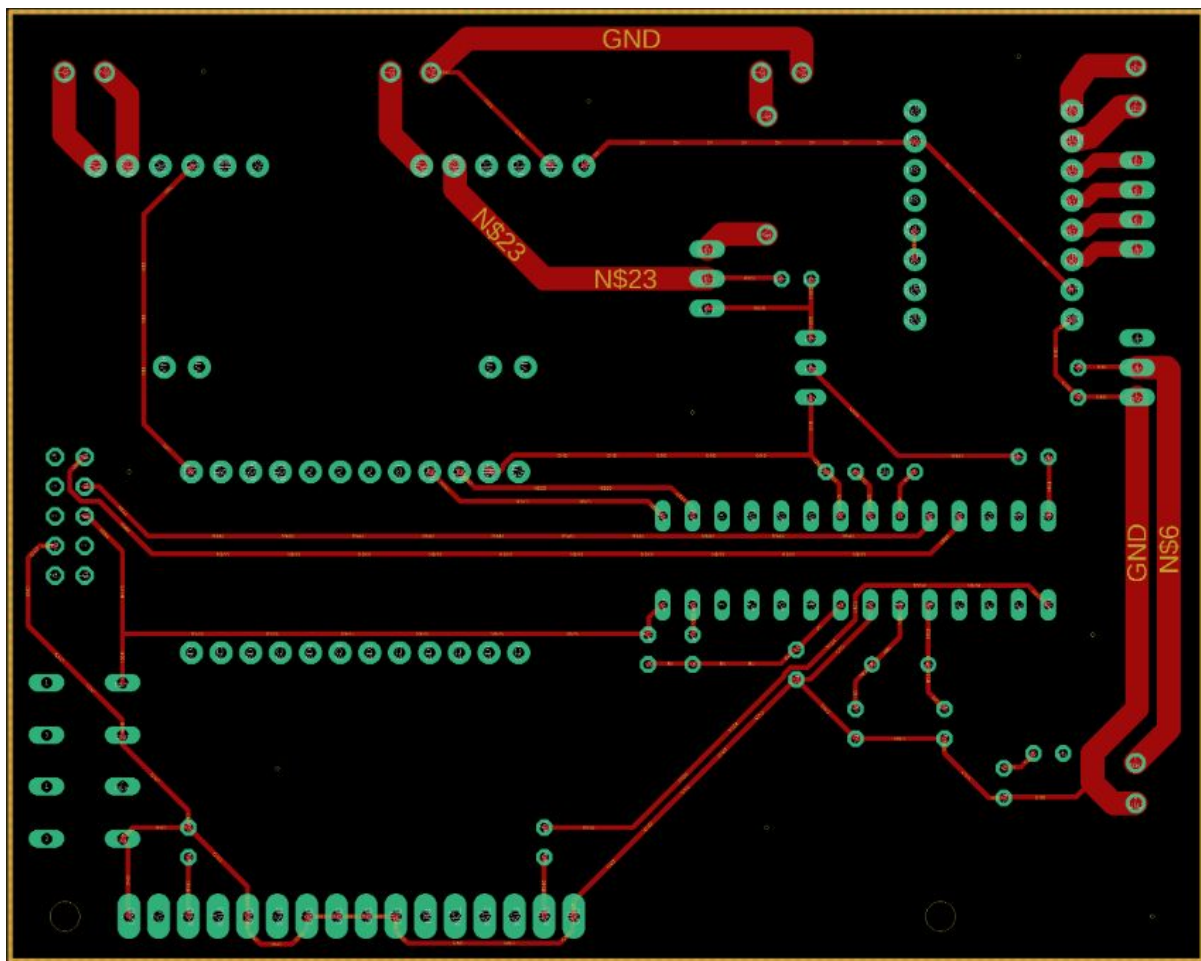
mikrokontroler - umożliwia sterowanie elementami układu, steruje silnikiem oraz serwomechanizmem.

3.3 Schemat montażowy obejmujący projekt płytki drukowanej, rozmieszczenie elementów na płycie oraz rozmieszczenie elementów urządzenia w obudowie.

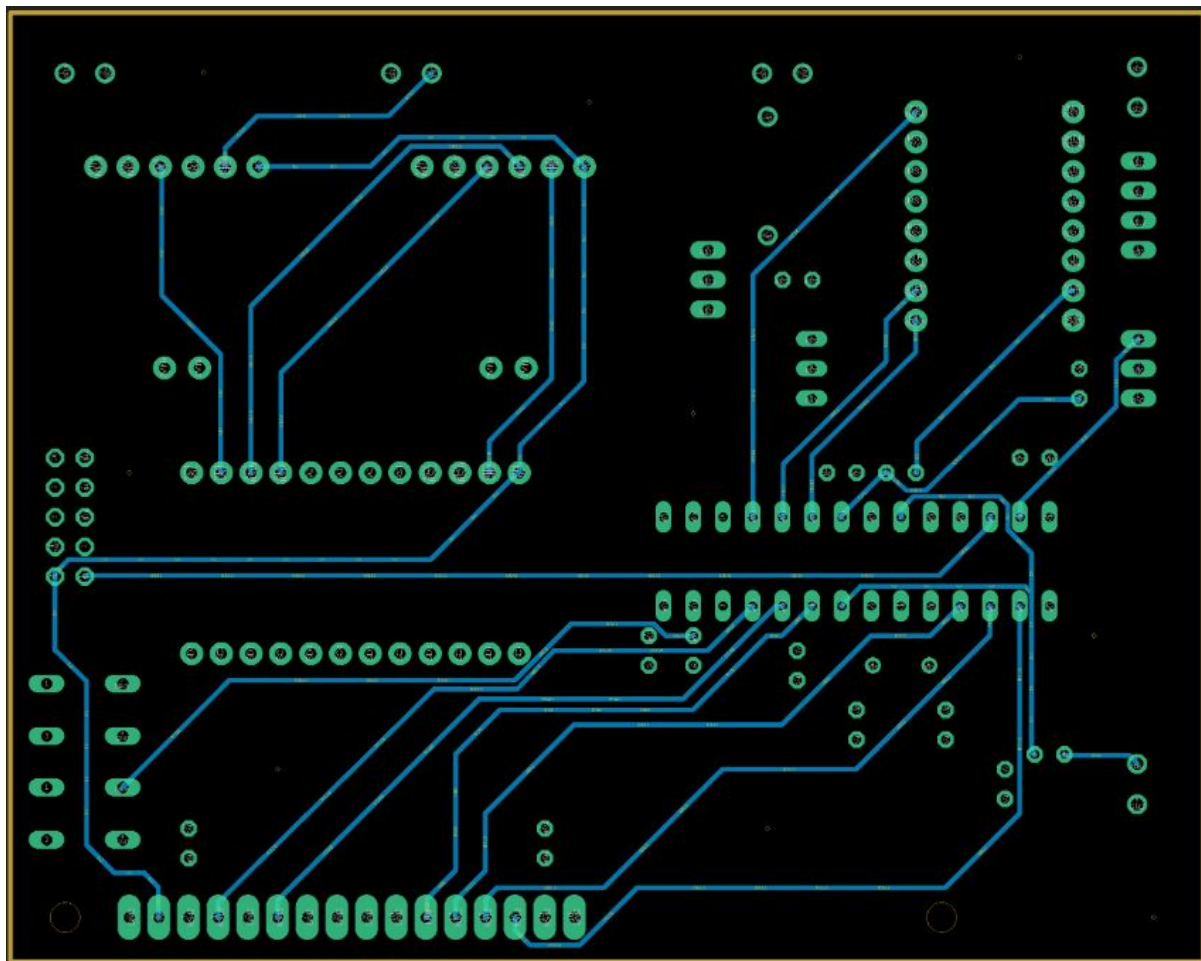
Schemat płytki pcb, widok ogólny:



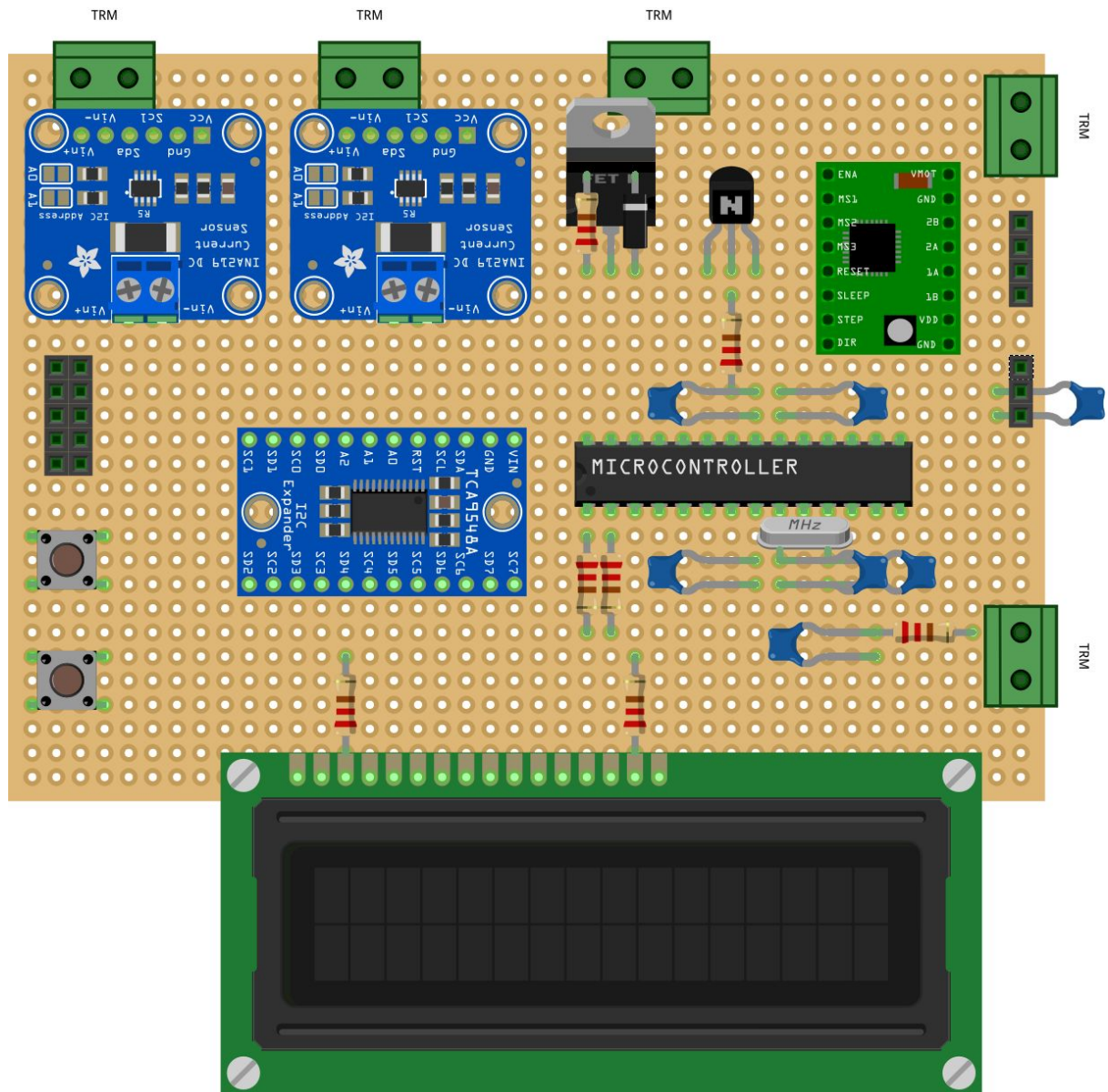
Schemat płytki pcb, góra:



Schemat płytki pcb, dół:



Rozmieszczenie elementów na płytce uniwersalnej:



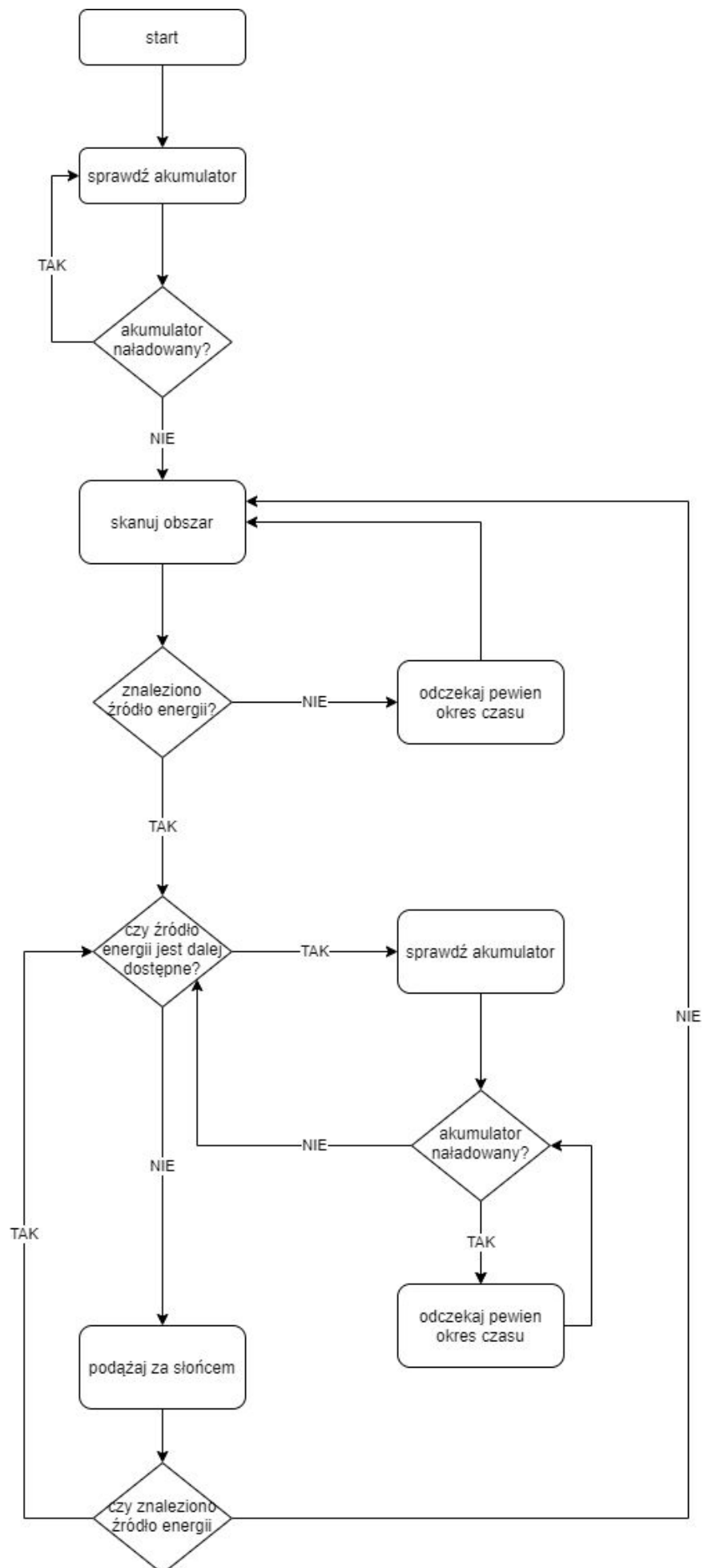
fritzing

3.4 Lista elementów.

1. Mikrokontroler Atmega328p,
2. Rezonator kwarcowy HC49 16MHz,
3. 2x przyciski,
4. 2x miernik prądu/napięcia - INA219,
5. Tranzystor unipolarny IRLZ44N,
6. Tranzystor bipolarny A2222,
7. Sterownik silnika krokowego - A4988,
8. Wyświetlacz - 2x16 znaków,
9. Multiplekser - TCA9548A,
10. 5x złącze ARK raster 5,0mm 2 pin,
11. Akumulator GreenCell 12v 10Ah,
12. Serwo AR-1201MG,
13. Silnik krokowy NEMA 17 - 17HS4401,
14. Panel słoneczny 10W firmy MAXX,
15. Rezystory 0.25W o wartościach: 26, 3x10k, 2x1k, 2k,
16. Kondensatory o wartościach: 2x22pF, 5x1nF,
17. Wtyki goldpin 2,54mm: 2x 5pin, 3pin, 4pin,
18. Dioda Schottky 1N5819 1A / 40V,
19. Podstawka pod układ scalony 7.62 mm 26 pin,
20. Przewód kynar 0.24mm,
21. Płytki uniwersalne wiercone dwustronne 8x12cm,
22. Radiator do chłodzenia sterownika silnika 9x9mm,
23. Przetwornica 5v XY-3606,
24. Przetwornica 12v LM317.

3.5 Algorytm oprogramowania urządzenia.

Poniżej pokazano algorytm urządzenia na schemacie blokowym:



1. Start algorytmu, zostają zainicjowane wszystkie wejścia a także wszystkie elementy, takie jak wyświetlacz, silnik.
2. Następuje sprawdzenie, czy akumulator jest naładowany, jeżeli tak to program czeka, aż napięcie akumulatora spadnie i przechodzi do kolejnego punktu.
3. Następuje pełny skan obszaru, w tym momencie silnik obraca panel a program mierzy moc generowaną i wynik zapisuje do tablicy. Z tablicy zostaje wybrane miejsce gdzie wartość była największa, a panel kieruje się w tę stronę. Serwomechanizm porusza panelem w pionie szukając najwydajniejszej pozycji. Jeżeli miejsce, gdzie wygenerowana moc była największa okazuje się za słabe, następuje oczekiwanie pewnego okresu czasu, po którym skan odbywa się ponownie. Jeżeli moc jest wystarczająca, algorytm przechodzi do następnego punktu.
4. W kolejnym kroku, gdy źródło światła zostało znalezione, algorytm cyklicznie sprawdza czy generowana moc na panelu nie spada poniżej deklarowanego progu, oraz sprawdza czy akumulator został naładowany. Jeżeli akumulator został naładowany podobnie jak poprzednio algorytm wyłącza ładowanie i czeka na rozładowanie.
5. Gdy moc generowana przez panel spada poniżej deklarowanego poziomu następuje wywołanie procedury podążania za słońcem, w tym celu panel obraca się w lewo i w prawo, a następnie porównuje zmierzone wartości i wybiera ten kierunek, gdzie wartość generowanej mocy była maksymalna. Następuje rekurencyjne podążanie w wybranym kierunku, do momentu kiedy moc generowana zaczyna spadać.
6. Po skończonej procedurze podążania za słońcem sprawdzane jest czy wartość generowanej mocy osiągnęła minimalny próg, jaki jest potrzebny by uznać źródło światła za znalezione. Jeżeli źródło światła zostało znalezione, algorytm wraca do punktu 4, jeżeli nie to do punktu 3.

3.6 Opis wszystkich ważniejszych zmiennych w programie.

LiquidCrystal lcd(LCD_RS, LCD_E, D4, D5, D6, D7); // Obiekt klasy odpowiedzialny za operacje na wyświetlaczu.

Adafruit_INA219 ina219; // Obiekt klasy odpowiedzialny za sterowanie miernikiem prądu/napięcia.

int servoState = 0; // Przechowuje aktualną pozycję serwa, maksymalnie 7.

int motorState = 0; // Przechowuje aktualną pozycję silnika krokowego, maksymalnie 25.

bool succedScan = false; // Zmienna określająca czy ostatni skan obszaru okazał się sukcesem.

float maxBatteryVoltage = 13.8; // Maksymalne napięcie jakie może osiągnąć akumulator.

float minBatteryVoltage = 12.3; // Minimalne napięcie jakie może mieć akumulator.

int waitingTime = 20; // Czas bezczynności jakie urządzenie musi odczekać po nieudanym skanie otoczenia.

int blinkingDelay = 250; // Czas po jakim wyświetlacz zapala się lub gaśnie gdy urządzenie przekazuje użytkownikowi informacje.

int motorSpeed = 13; // Czas mówiący ile milisekund musi upłynąć by sygnał STEP w sterowniku silnika zmienił się ze stanu wysokiego na niski lub odwrotnie.

int minimalPower = 200; // Minimalna moc jaką panel musi uzyskać aby skan został uznany jako sukces.

int measureTime = 100; // Odstęp w milisekundach między seryjnymi pomiarami prądu.

Servo servo; // Obiekt klasy odpowiedzialny za sterowanie serwomechanizmem.

3.7 Opis funkcji wszystkich procedur, występujących w programie.

Procedury odpowiedzialne za obsługę wyświetlacza:

- **void initLCD()** - Inicjalizuje wyświetlacz.
- **void turnBackgroundLight(bool turnOn)** - Zależnie od argumentu włącza lub wyłącza podświetlenie.
- **void writeOnLCD(String arg1, String arg2)** - Wyświetla informacje znajdujące się w arg1 i arg2 na wyświetlaczu.
- **void alert(int blinkNumber)** - Włącza i wyłącza oświetlenie kilkakrotnie, zależnie od argumentu blinkNumber.

Procedury odpowiedzialne za obsługę miernika prądu/napięcia INA219 jak i również multipleksera TCA9548A.

- **float roundNumber(float number)** - Zaokrągla liczbę znajdującą się w argumencie number do dwóch miejsc po przecinku, i zwraca ją.
- **void initINA219s()** - Inicjalizuje oba mierniki INA219.
- **void TCA9548A(uint8_t bus)** - Umożliwia wybranie odpowiedniego miernika zależnie od potrzeb. Dla wartości 0 argumentu bus zostanie wybrany miernik odpowiedzialny za prąd i napięcie akumulatora a dla 1 prąd i napięcie ogniwa słonecznego.
- **void initINA219()** - Inicjalizuje miernik INA219, jeżeli wystąpi błąd, to program zatrzyma się w tej funkcji i zgłosi błąd użytkownikowi.
- **float measureCurrent()** - Zwraca bieżące natężenie prądu przepływającego przez miernik.
- **float measureLoadVoltage()** - Zwraca całkowite napięcie akumulatora lub ogniwa słonecznego, zależnie od wcześniej wybranego miernika.
- **float measurePower()** - Zwraca średnią arytmetyczną z kilku pomiarów mocy.
- **measureAll()** - Funkcja mierząca wszystkie potrzebne wartości, które użytkownik odczytuje w momencie naciśnięcia przycisku.

Procedury odpowiedzialne za przełącznik doprowadzający prąd z panela słonecznego do akumulatora, zbudowany z dwóch tranzystorów.

- **void initChargerSwitch()** - Inicjalizuje pin odpowiedzialny za załączanie lub odłączanie obwodu. Wyłącza domyślnie ładowanie akumulatora.
- **void chargingON()** - Zamyka obwód ładujący akumulator.
- **void chargingOFF()** - Otwiera obwód ładujący akumulator.

Procedury odpowiedzialne za obsługę przycisku.

- **void initButton()** - Inicjalizuje odpowiedni pin, który będzie odczytywał zachowanie przycisku.
- **bool buttonPressed()** - Sprawdza czy przycisk został naciśnięty i zwraca odpowiednią wartość. Jeżeli wartość wynosi true, funkcja uruchamia procedurę **alert()** potwierdzającą wciśnięcie przycisku.

Procedura odpowiedzialna za obsługę serwomechanizmu.

- **void setServoState(int state)** - Ustawia serwo relatywnie do stanu podanego w argumencie funkcji.

Procedury odpowiedzialne za obsługę silnika krokowego, przy użyciu sterownika silnika.

- **void initMotor()** - Inicjalizuje piny podpięte do sterownika. Domyślnie wyłącza motor w celu ochrony przed przegrzaniem sterownika i oszczędzania energii.
- **void enableMotor()** - Włącza możliwość sterowania silnikiem.
- **void disableMotor()** - Wyłącza możliwość sterowania silnikiem.
- **void moveRight(int stepsCount)** - Porusza silnikiem w prawo o ilość kroków podanych w argumencie funkcji.
- **void moveLeft(int stepsCount)** - Porusza silnikiem w lewo o ilość kroków podanych w argumencie funkcji.
- **void setMotorState(int state)** - Ustawia odpowiedni stan motoru.

Procedury główne, tworzące algorytm pokazany na schemacie blokowym, które korzystają z funkcji opisanych powyżej.

- **void trackVertical()** - Ustawia panel przy użyciu serwomechanizmu na pozycję w której ilość produkowanej energii jest najwyższa.
- **void trackSunRight(float lastPower)** - Funkcja wywoływana rekurencyjnie do momentu znalezienia pozycji w której produkowana energia jest najwyższa. Obraca panelem w prawo.
- **void trackSunLeft(float lastPower)** - Funkcja wywoływana rekurencyjnie do momentu znalezienia pozycji w której produkowana energia jest najwyższa. Obraca panelem w lewo.
- **bool findSun()** - Funkcja wywoływana na początku działania urządzenia, jak i w momencie w którym nie powiodło się śledzenie słońca. Zwraca true jeżeli źródło światła zostało znalezione.
- **void trackSun()** - Po udanym wcześniejszym znalezieniu słońca, gdy ilość generowanej energii spada, metoda *trackSun* rozpoczyna śledzenie słońca.
- **void checkBattery()** - Sprawdza czy akumulator został naładowany, jeżeli tak to wyłącza ładowanie i czeka na rozładowanie akumulatora.
- **void controlCharging()** - Główna funkcja mierząca stan naładowania akumulatora. Wykonuje skanowanie obszaru lub śledzenie znalezionej już wcześniej słońca. Usypia również urządzenie jeżeli znalezienie słońca jest niemożliwe.
- **void showData()** - Prezentuje określone dane użytkownikowi.

3.8 Opis interakcji oprogramowania z układem elektrycznym.

Obsługa wyświetlacza:

Komunikacja z wyświetlaczem została zrealizowana dzięki poniższym pinom oraz przy użyciu biblioteki *LiquidCrystal*.

```
#define LCD_RS - 2
#define LCD_E - 3
#define D4 - 4
#define D5 - 5
#define D6 - 6
#define D7 - 7
```

Włączanie lub wyłączanie podświetlenia wyświetlacza zostało zrealizowane na pinie programowalnym numer 8. Podczas wyświetlania tekstu, na wyjście pinu LCD_A zostaje wprowadzony stan wysoki, który powoduje zaświecenie się podświetlenia wyświetlacza.

```
#define LCD_A - 8
```

Obsługa miernika prądu/napięcia INA219:

Komunikacja z modułami została rozwiązana przy użyciu biblioteki *Adafruit_INA219*, na pinach programowalnych o numerach 18(SDA) i 19(SCL). Moduły zostały podłączone do multiplexera TCA9548A, a ten do wyżej wymienionych pinów. Biblioteka **Wire** umożliwiła przełączanie się między modułami.

Obsługa przełącznika zbudowanego na tranzystorach:

W programie zdefiniowano poniższy pin:

```
#define MOSFET_GATE 9
```

Został on połączony z bazą tranzystora A2222, którego kolektor jest połączony z bramką tranzystora IRLZ44N. Po podaniu stanu wysokiego obwód zostaje zwarty, na bramce pojawia się stan wysoki który **rozłącza** obwód ładujący akumulator.

Obsługa przycisku:

Zdefiniowano następujący pin programowalny o numerze 0:

```
#define BUTTON_PIN 0
```

Po naciśnięciu przycisku na pinie BUTTON_PIN pojawia się stan niski, który jest odczytywany jako naciśnięcie przycisku. W takiej sytuacji funkcja odpowiedzialna za przycisk zwraca wartość true.

Obsługa serwomechanizmu:

Do kontroli serva skorzystano z biblioteki `Servo`. Komunikacja odbywa się przez niżej opisany pin:

```
#define SERVO_PIN 10
```

Po przekazaniu odpowiedniego argumentu do funkcji `writeMicroseconds()` z biblioteki `Servo`, serwomechanizm obraca się w lewo lub w prawo.

Obsługa silnika krokowego:

Do kontroli silnika, w programie zdefiniowano następujące piny:

```
#define DIR 16  
#define STEP 15  
#define EN 14
```

Analogicznie opisane są wyprowadzenia na sterowniku silnika. Pin `DIR` umożliwia zmianę kierunku silnika. Podanie wysokiego stanu na pin `STEP`, a następnie niskiego spowoduje ruch silnika w określonym kierunku o jeden krok. Pin `EN` umożliwia wyłączanie bądź włączanie sterownika co pomaga chronić sterownik przed przegraniem oraz oszczędza energię.

3.9 Szczegółowy opis działania ważniejszych procedur.

W programie można wyróżnić trzy najważniejsze procedury. Są to kolejno:

- **controlCharging()**,
- **findSun()**,
- **trackSun()**.

controlCharging()

Jest główną procedurą wykonującą się w pętli. Zawiera w sobie procedury umożliwiające skan obszaru, śledzenie źródła światła. Kontroluje stan naładowania akumulatora. Jeżeli akumulator zostanie naładowany to ładowanie zostaje zatrzymane. Procedura umożliwia również kontakt użytkownika z urządzeniem. Obsługuje przycisk po naciśnięciu którego na wyświetlaczu prezentowane są określone dane. Obsługuje również sytuację w której urządzenie nie może znaleźć źródła światła.

findSun()

Funkcja ta jest procedurą umożliwiającą znalezienie najwydajniejszego źródła światła, poprzez obsługę silnika oraz serwomechanizmu. Jest jedną z głównych funkcji rozpoczynających algorytm zaprezentowany na schemacie blokowym. Na początku

następuje ustawienie serwomechanizmu na pozycję 1, która daje największą szansę na wykrycie słońca. W kolejnym etapie następuje skan całego obszaru, a na końcu wybierany jest ten stan, w którym generowana energia była największa. Po ustawieniu się na pożądanym stanie zostaje uruchomiona funkcja szukająca najlepszego ustawienia ogniwa w pozycji pionowej. Jeżeli wszystko przebiegło pomyślnie, to zostaje zwrócona wartość true, która funkcja controlCharging() może użyć w swoim celu. W przypadku zwrócenia wartości false następuje odczekanie pewnego okresu czasu, po którym operacja jest ponawiana.

trackSun()

Procedura jest wywoływana w momencie którym funkcja *findSun()* zwróciła wartość true oraz minimalna moc generowana przez panel słoneczny przestała być osiągana. Procedura obraca panel w lewo i w prawo, a następnie dokonuje sprawdzenia w którym punkcie wartość ta była największa. W następnej kolejności, rekurencyjnie, w kierunku gdzie wykryto największą wartość panel jest obracany. Jeżeli mimo wszystko nie udało się znaleźć minimalnej wartości, panel ustawia się na pozycję 0 i rozpoczyna metodę findSun().

4. Specyfikacja zewnętrzna urządzenia.

4.1 Opis funkcji elementów sterujących urządzeniem.

Elementem sterującym urządzeniem jest przycisk, który należy wcisnąć po kalibracji urządzenia. Gdy urządzenie zakończy skan użytkownik może wcisnąć przycisk ponownie, w celu uzyskania informacji dotyczących stopnia naładowania akumulatora, lub aktualnej wydajności ogniwa.

4.2 Opis funkcji elementów wykonawczych.

- serwomechanizm - zapewnia możliwość ruchu ogniwa w pionie,
- silnik krokowy - umożliwia obracanie panelem w lewo lub w prawo,
- tranzystor MOSFET - umożliwia otwieranie bądź zamykanie obwodu ładującego akumulator,
- wyświetlacz LCD - prezentuje informacje użytkownikowi oraz pokazuje aktualny proces skanowania obszaru.

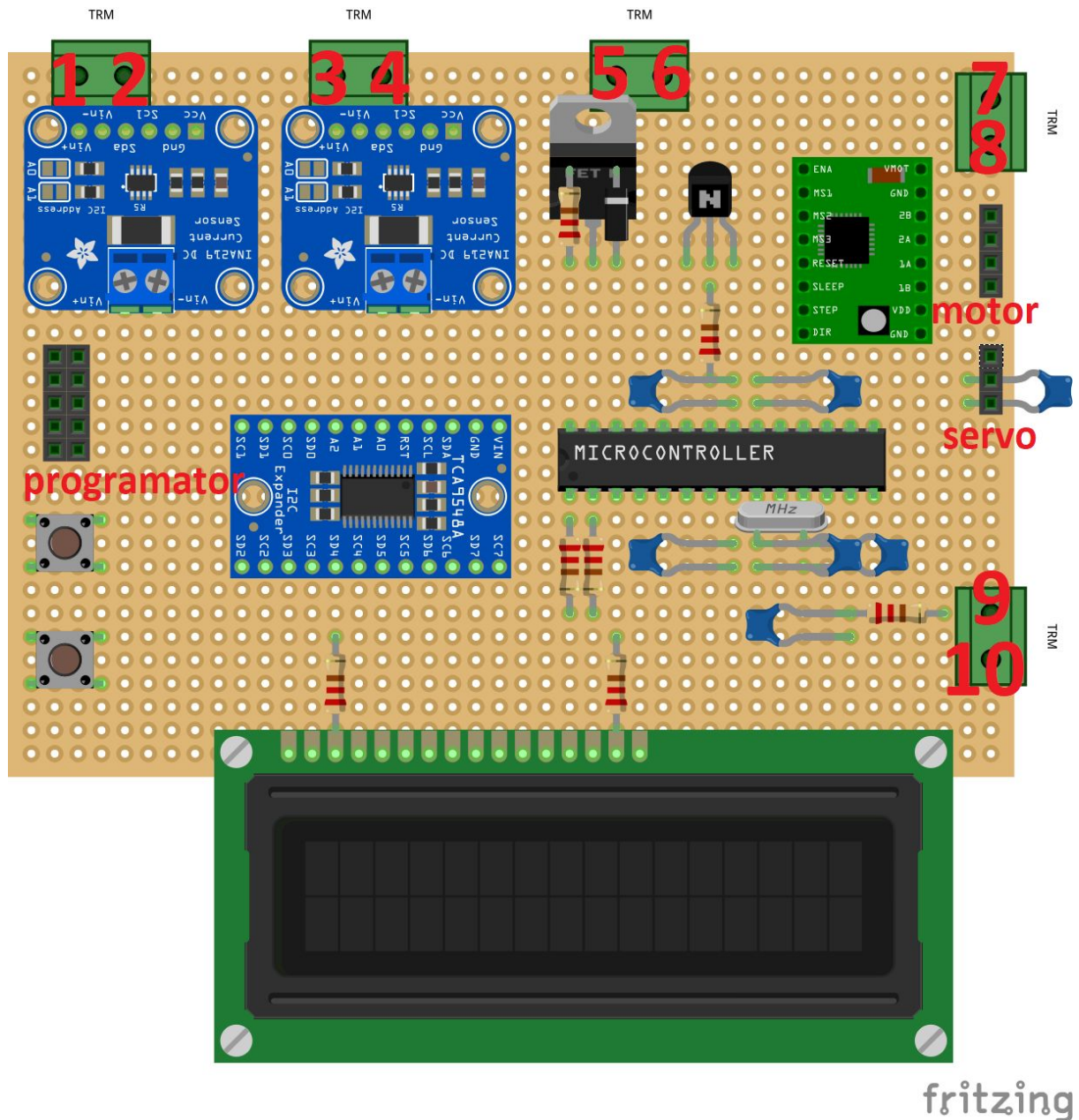
4.3 Opis reakcji oprogramowania na zdarzenia wewnętrzne.

Na początku, po naciśnięciu przycisku rozpocznie się algorytm urządzenia polegający na znalezieniu i w dalszej kolejności śledzeniu najintensywniejszego źródła światła. Oprogramowanie będzie reagowało na zmianę intensywności światła, poprzez manipulowanie silnikiem krokowym oraz serwomechanizmem w taki sposób, aby energia generowana przez panel była możliwie jak największa. Oprogramowanie reaguje na nagłe wyłączenie światła, poprzez uruchomienie metody ponownego skanowania a jeżeli i to nie przyniesie skutku, przejdzie w tryb oczekiwania i po pewnym czasie spróbuje ponownie. W celu ochrony przed przeładowaniem akumulatora oprogramowanie wyłączy ładowanie i gdy napięcie spadnie do bezpiecznej wartości ponownie uruchomi proces ładowania.

4.4 Instrukcja obsługi urządzenia.

- Ustawić urządzenie w dobrze nasłonecznionym miejscu.
- Zasilić urządzenie poprzez przymocowanie kabli do akumulatora, niebieski kabel do minusa natomiast czerwony do plusa.
- Po uruchomieniu urządzenia należy ustawić panel w pozycji 0 dla serwomechanizmu i silnika krokowego. Pozycja zero dla serwomechanizmu to ustawienie ogniwa w pozycji pionowej, natomiast pozycja zero dla serwa jest zaznaczona dwoma czarnymi punktami, które muszą znaleźć się możliwie jak najbliżej siebie.
- Nacisnąć przycisk i puścić go w momencie kiedy ekran zacznie migać.

4.5 Schemat okablowania.



Na schemacie pokazano w którym miejscu należy podpiąć programator, silnik krokowy oraz serwomechanizm. Do wejść oznaczonych numerami należy podpiąć kolejno:

1. napięcie z akumulatora,
2. napięcie wejściowe dla przetwornicy 12v i 5v,
3. plus z panela słonecznego,
4. masa z panela słonecznego,
5. napięcie z akumulatora,
6. masa z akumulatora,
7. napięcie z przetwornicy 12v,
8. masa z przetwornicy 12v,

9. napięcie z przetwornicy 5v,
10. masa z przetwornicy 5v.

4.6 Opis montażu układu.

Przed przystąpieniem do lutowania został wykonany schemat rozmieszczenia elementów na płytce uniwersalnej. W kolejnym kroku wszystkie elementy zostały wlutowane w określonym miejscu. Jako pierwsze z kabli zostały lutowane kable grube służące do zasilania silnika jak i ładowania akumulatora. W kolejnym kroku przystąpiono do lutowania kabli sygnałowych.

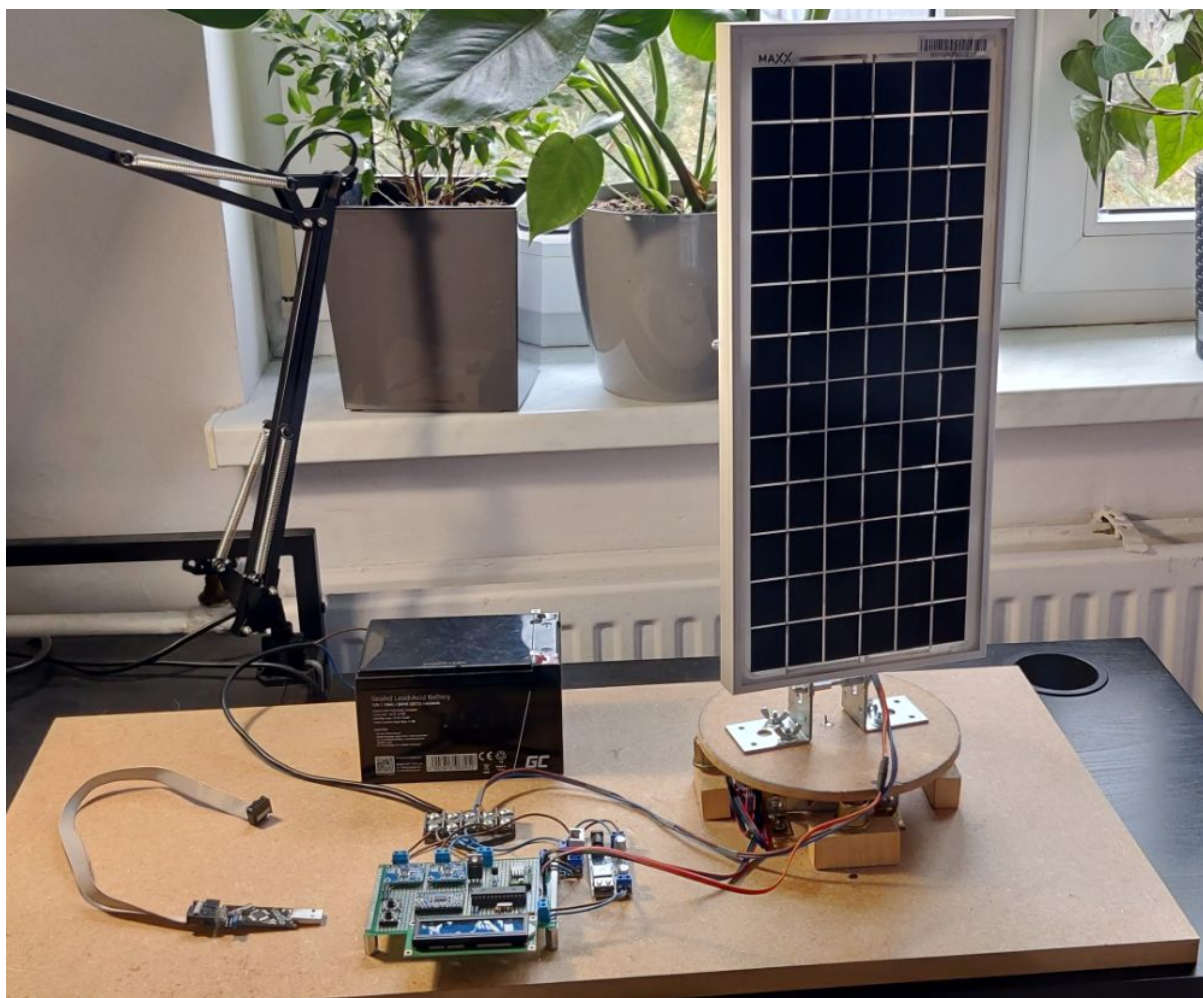
4.7 Opis sposobu programowanie układu.

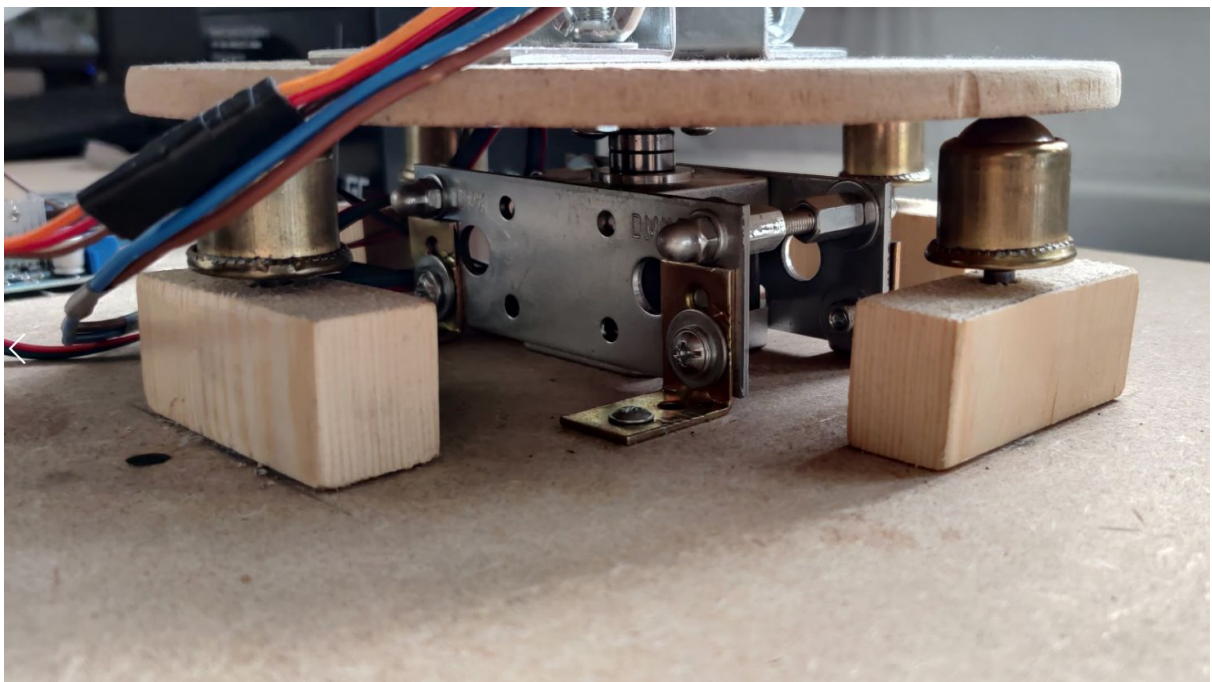
Układ posiada wejście na taśmę IDC, która jest zgodna z programatorami USBasp ISP. Programowanie odbywa się poprzez podłączenie wyżej wymienionego programatora w języku C++.

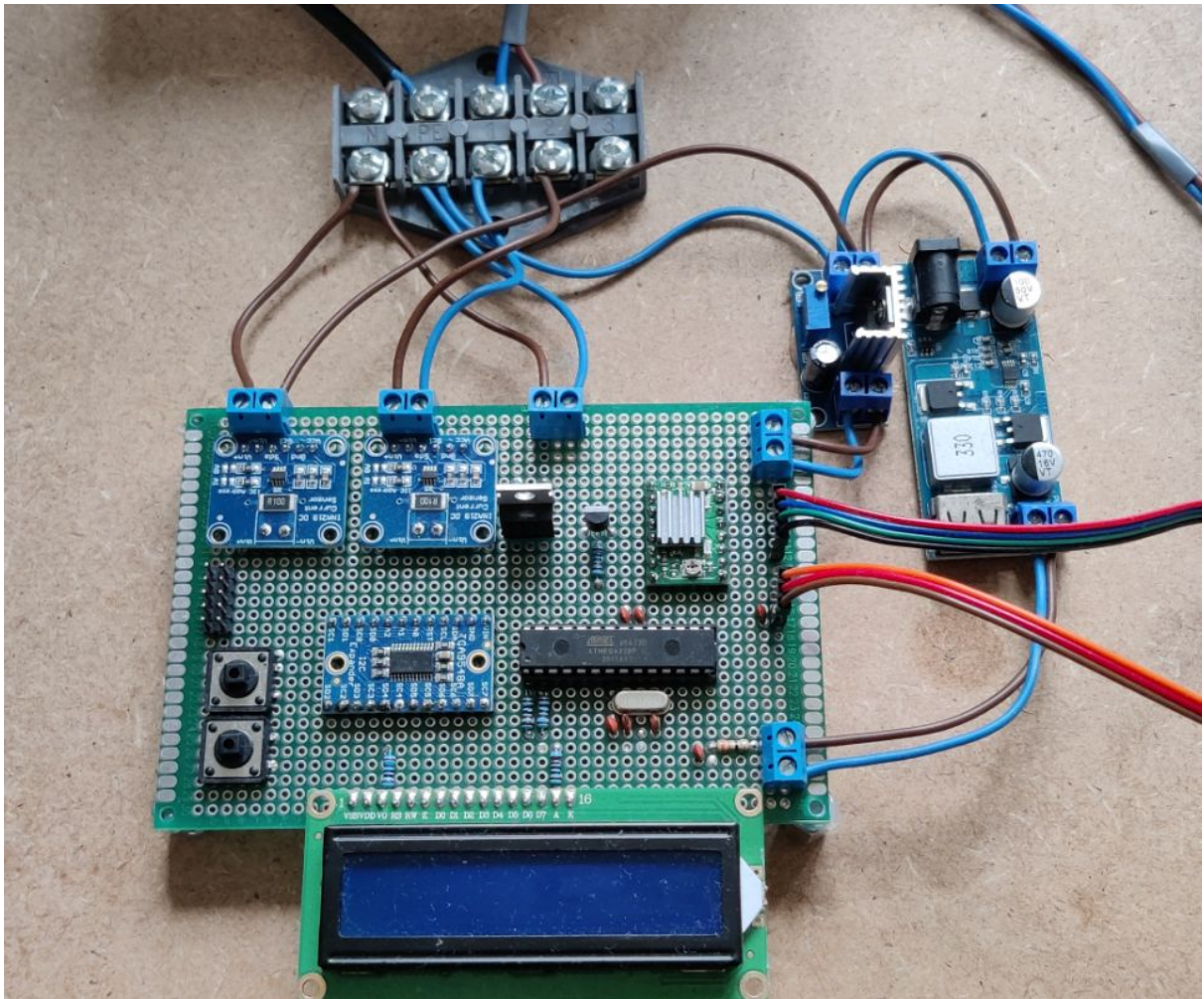
4.8 Opis sposobu uruchamiania oraz testowania układu.

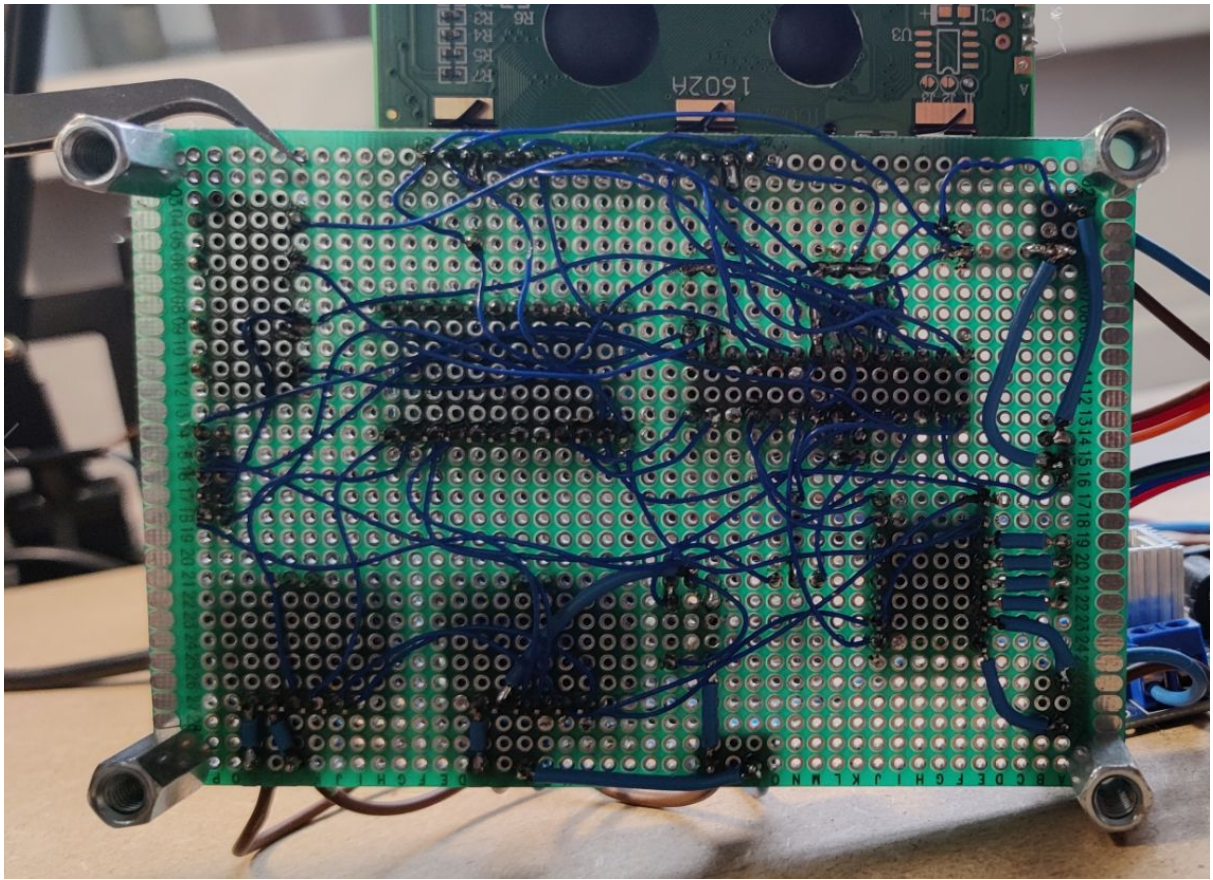
Przed przystąpieniem do tworzenia układu na płytce uniwersalnej, wszystkie elementy zostały połączone na płytce prototypowej. Do sprawdzenia działania zostały napisane proste funkcje, które uruchamiały każdy z podzespołów. Poprawność mierzenia modułów mierzących została sprawdzona za pomocą multimetru.

4.9 Zdjęcia z gotowego układu.









5. Wnioski i uwagi z przebiegu pracy.

Podczas pracy wiele rzeczy, które z pozoru wydawały się łatwe, wcale takie nie były. Początkowo projekt nie zakładał używania modułów pomiarowych, bo napięcie miało być mierzone przy pomocy atmegi, jednak końcowo postanowiłem skorzystać z modułów o nazwie INA219, które okazały się bardzo pomocne i sprawiły, że wszelkie operacje, takie jak szukanie światła czy pomiar napięcia na akumulatorze okazały się o wiele dokładniejsze. Kolejną przysłowiową ścianą na jaką się natknąłem, było stworzenie przełącznika przy pomocy tranzystora, sterowanego za pomocą mikrokontrolera. Początkowe testy na płytce prototypowej okazały się porażką. Po paru dniach, i przerobieniu kilkunastu schematów znalezionych na internecie znalazłem idealne rozwiązanie dla swojego projektu. Lutowanie całego układu na początku okazało się bardzo prostą i przyjemną rzeczą, jednak po krótkim użytkowaniu gotowego urządzenia doszło do zwarcia, które kosztowało mnie czas i pieniądze, gdyż wszystkie części musiałem zamówić ponownie. Okazało się, że złącza których użyłem w projekcie do montowania kabli miały za mały rozstaw, przez co odstęp między lutowanymi kablami okazał się zbyt mały. Nauczony na wcześniejszej płytce zastosowałem większe złącza, oraz zamontowałem dodatkowe nóżki, na których oparłem cały układ. Po skończonej płytce przyszedł czas na budowę całej platformy, na której obracał się panel. Początkowo całość miała być zbudowana na materiale typu plexi, jednak z powodu lepszej dostępności aluminiowych części postanowiłem skorzystać z tego typu materiału. Największym problemem okazało się przekazanie napędu z silnika na platformę. Po kilku dniach szukania pomysłu i części nałożyłem aluminiowy kołnierz pochodzący ze starego kołowrotka, który przewierciłem w 4 miejscach i przykręciłem do drewnianej platformy. Przez brak stabilności całej platformy postanowiłem dołożyć metalowe nóżki od mebli, posiadające na czubku kulki, które zmniejszyły opór i usztywniły całą konstrukcję.

6. Wnioski końcowe.

Projekt trackera słonecznego okazał się projektem bardzo wymagającym, jednocześnie dostarczył mi dużo satysfakcji, kiedy po skończonych pracach mogłem zobaczyć jak wszystkie elementy, które złożyłem ze sobą współpracują. Dzięki niemu nauczyłem się w jaki sposób zaprojektować układ od podstaw, jak i poszerzyłem swoją wiedzę dotyczącą fotowoltaiki. Projekt nauczył mnie w jaki sposób można programować mikrokontroler a także na jakie elementy zwrócić uwagę, aby projektowany układ był zabezpieczony przed ewentualnym zwarcie lub innym uszkodzeniem. Szukanie elementów potrzebnych do budowy trackera słonecznego umożliwiło poznanie mi wiele innych różnych elementów, które na pewno przydadzą mi się w innych projektach.

7. Literatura.

Podłączanie silnika:

https://www.youtube.com/watch?v=5M1KMmBmQ2g&feature=emb_title&ab_channel=Botland

Strona, która zainspirowała mnie do stworzenie przełącznika przy pomocy tranzystora:

<https://www.bristolwatch.com/solar2.htm>

Strony, dzięki którym zrozumiałem w jaki sposób akumulator jest ładowany przez panel słoneczny i jakie towarzyszą temu zjawiska:

<https://www.quora.com/Why-do-solar-panels-generate-a-high-voltage-but-a-low-current?share=1>

<https://electronics.stackexchange.com/questions/268025/solar-panel-output-voltage-drops>