**Elektronika i oprogramowanie urządzenia**

**Wstęp:**

Głównym zadaniem elektroniki i zaimplementowanego programu jest sterowanie silnikami krokowymi w taki sposób, by teleskop ustawił się na konkretną gwiazdę, a następnie ją śledził w miarę jej ruchu po nocnym niebie. W tym rozdziale będą omówione podstawy teoretyczne, które posłużą do napisania algorytmów sterowania silnikami, zaprojektowany układ elektroniczny, schemat blokowy sterowania silnikami oraz zademonstrowane zostanie działanie programu.

**Podstawy teoretyczne**

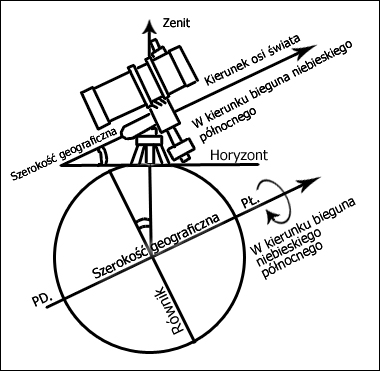
Uwagi do poniższego punktu: opis skupia się na prostym wytłumaczeniu zjawisk zachodzących podczas ruchu naszej planety bez zagłębiania się w szczegółowe zagadnienia i problemy jak różnica pomiędzy dobą słoneczną, a dobą gwiezdną albo nieliniowość czasu gwiezdnego. Takie podejście wynika z tego, że byłby potrzebny bardzo obszerny opis tych zjawisk, a i tak algorytm programu nie będzie ich brał pod uwagę ze względu na bardzo trudną implementację – przykładowo: doba gwiezdna trwa 23h i 56 minut albo wpływ położenia naszej planety w układzie słonecznym. Uznano, że błędy są pomijalnie małe oraz głównym celem jest obserwacja odległych gwiazd, a nie np. Marsa.

Na poniższym rysunku 1 przedstawiono oznaczenie poszczególnych osi w teleskopie o montażu paralaktycznym.



Rysunek 1 Osie w teleskopie o montażu paralaktycznym. Latitude - szerokość, Polar Axis - oś biegunowa; Declination Axis - oś deklinacji

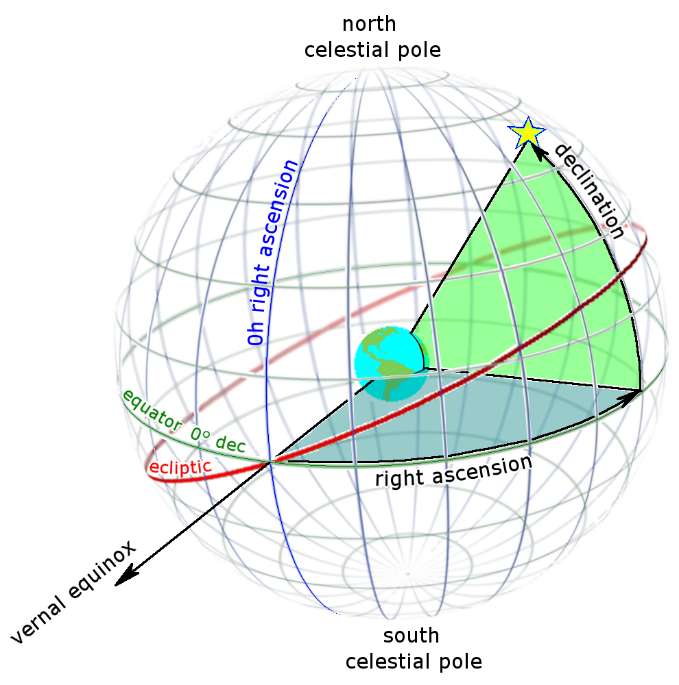
Kąt szerokości ustawiamy na zgodny z naszą szerokością geograficzną oraz kątem pochylenia teleskopu. Na rysunku 2 przedstawiono jakie ma znaczenie prawidłowe ustawienie tego kąta.



Rysunek 2 Zobrazowanie znaczenia ustawienie kąta szerokości

Oś deklinacji ustawiamy na wartość równą pozycji na nieboskłonie naszej obserwowanej gwiazdy. Na rysunku 3 przedstawiono zobrazowanie znaczenia ustawienia kąta deklinacji. Dzięki ustawieniu tych dwóch omówionych parametrów nasz teleskop jest nakierowany na wybrany przez nas punkt – obiekt astronomiczny. Niemniej ze względu na ciągły obrót kuli ziemskiej wokół własnej osi obserwujemy pozorny ruch gwiazd na niebie. Należy stosować w takim wypadku nadążano korekcję, by móc zastosować m.in. długą ekspozycję do zdjęć.

Do w pełni prawidłowego ustawienia teleskopu trzeba ustawić oś biegunową.



Rysunek 3 Zobrazowanie znaczenia deklinacji i rektascensji

Oś biegunową teleskopu powinniśmy ustawić zgodnie z wynikiem sumy:

LHAobject – lokalny kąt godzinny

GST – Czas gwiazdowy Greenwhich

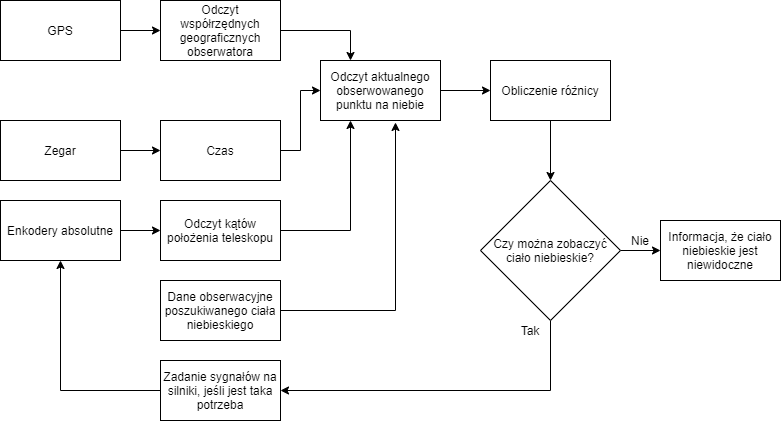
λ – długość geograficzna obserwatora

α – Rektascensja poszukiwanego obiektu na nieboskłonie

W praktyce błędne ustawienie osi rektascensji teleskopu wpływa jedynie na to, że obraz będzie obrócony o pewien kąt. Niemniej kompensując obrót ziemi względem własnej osi obraz w obiektywie będzie nieruchomy.

**Schemat blokowy algorytmu**

Uwaga, teleskop musi być ustawiony w osi Północ-Południe. Wynika to z tego, że użytkownik będzie wstanie znacznie precyzyjnej ustawić w tej osi niż program oparty o odczyt z enkoderów. Wynika to m.in. z błędów od deklinacji magnetycznych oraz że nie każdy teleskop ma ruchomą 4 oś.



Rysunek Schemat blokowy algorytmu

**Zaprojektowany układ elektronicznych**

Realizowany układ oprócz wykonywanego algorytmu musi spełniać również szereg innych wymogów wynikających z miejsca zastosowania:

- źródło zasilania pozwalające na uzyskanie odpowiednich prądów i napięcia

- napięcie bezpieczne

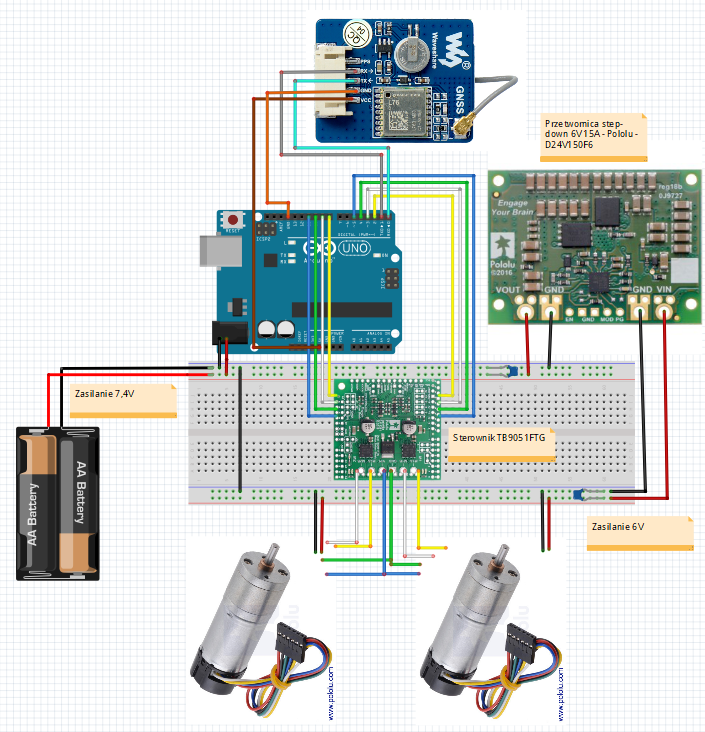
- temperatura pracy: od -10 st. C do 30 st. C

- łączna waga do 0.5 kg

- możliwość uzyskania prądów rozładowywania do 8A

- małe wymiary (łączna powierzchnia mniejsza niż 200 cm2)

Wyliczenia sprawdzające spełnienie tych kryteriów będą umieszczone już po opisie zastosowanych części.



Rysunek Schemat elektryczny podłączeń poszczególnych elementów elektronicznych

Użyte komponenty:

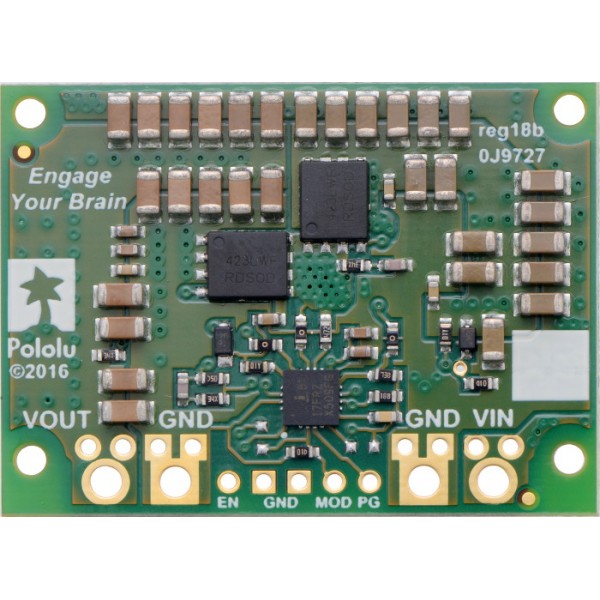
Arduino Uno Rev3

Został użyty ze względu na odpowiednią moc obliczeniową, wystarczającą ilość wyprowadzeń pinów cyfrowych, posiadanie interfejsów komunikacyjnych UART i USB. Może być zasilane prądem o napięciu 7-12V. Również ważne jest to, że moduł jest bardzo łatwo programować, oraz jest w posiadaniu członków zespołu.

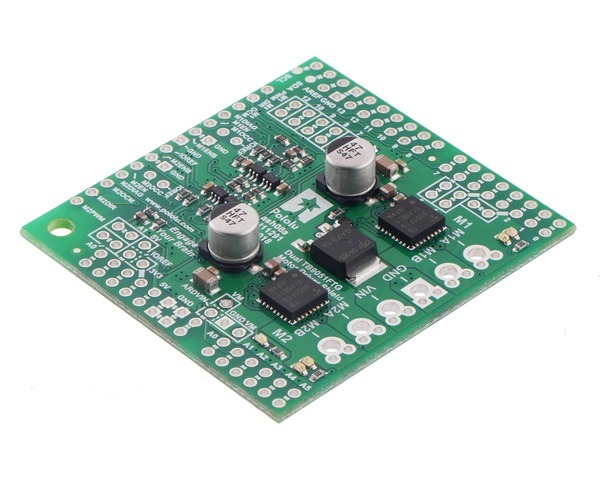
Zasilanie – akumulator Li-Ion LG INR18650 MG1 3,7V 2900mAh 10A

Ze względu na to, że teleskop będzie działał bez możliwości pobierania prądu z sieci, postanowiono użyć zasilania akumulatorowego. Parametrami branymi do wyboru zasilania całej elektroniki i mikrosilników były napięcie znamionowe, pojemność oraz prąd rozładowywania. Do prawidłowego funkcjonowania całej elektroniki napięcie musi być 6V dla silników oraz 7V-12V dla modułu Arduino. Akumulatory są podłączone szeregowo w celu zapewnienia docelowego napięcia 7,4V. Oprócz tego, prądy rozładowania zgodnie z obliczeniami będą wynosiły do 5,8A. Akumulatory to gwarantują z dużą nawiązką, co jest szczególnie ważne w momencie, gdy cały system jest uruchamiany i przez krótki czas są duże prądy rozładowywania. Również ważnym czynnikiem jest pojemność, która powinna być wystarczająca do zasilania całego ponad godzinę Innym ważną przyczyną wyboru tych ogniw był rozmiar AA, co ułatwia dobór ładowarki przez użytkownika. Pewnym mankament jest to, że ten typ akumulatorów ma zmniejszoną pojemność w temperaturach poniżej 0 stopni Celsjusza.

Przetwornica step down – D24V150F6 6V 15A – Pololu 2882

Wybór tej przetwornicy wynika z posiadania odpowiednego napięcia wejściowego 7,2V do 40V zasilanego ogniwami, napięcia wyjściowego 6V umożliwiające uzyskanie odpowiednich napięć na silnikach, oraz wysoki prąd wyjściowy 15A, co bez problemowo zasili zastosowane silniki i pozwala na ewentualne zastosowanie bardziej prądożernych silników w przyszłych modyfikacjach.

Sterownik silnika krokowego – Dual TB9051FTG Motor Driver Shield

Silniki krokowe wymagają sterowników, by móc łatwiej nimi sterować za pomocą mikrokontrolera. Stosuje się zwykle dedykowany układ elektroniczny. Zastosowanie powyższego układu wynika z tego, że jest dedykowany do zastosowanych silników. Oprócz tego producent zapewnia odpowiednie wsparcie w postaci bibliotek na Arduino.

Moduł GPS – L76X Multi GNSS

 Powyższy moduł lokalizacyjny zastosowano ze względu na szybkie i dokładne pozycjonowanie (do 30s), małe zużycie energii, napięcie zasilania umożliwiające zasilanie bezpośrednio z pinu Arduino, posiadanie już gotowych bibliotek na Arduino oraz małe rozmiary (32x25 mm).

1. Wymagania prądowe

Maksymalne prądy wymagane przez silniki: 2,4A x 2 = 4,8 A

Maksymalne prądy wymagane przez moduł Arduino Uno: 1A

Łączne: 5,8 A

Użyte ogniwa mają dopuszczalne prądy rozładowania do 10 A. A więc uda się zasilić całe urządzenie oraz pozostaje dalej możliwość rozwoju urządzenia dodając kolejne elementy jak np. nowy silnik albo ekran użytkownika.

1. Czas działania.

Przy podłączeniu szeregowo dwóch ogniwach ich pojemność łączna wyniesie 5,8Ah. Przy maksymalnym zużyciu prądów całe urządzenie będzie działało co najmniej 1h. Należy pamiętać, że w praktyce tylko jeden silnik będzie cały czas pracował kompensując ruch obrotowy ziemi oraz moduł Arduino w tej aplikacji realnie będzie wymagał dużo mniejszych prądów niż 1A. Realny czas działania należy szacować na 1,5-2h. Nie mniej dodając kolejną parę ogniw, w sposób równoległy, czas ten można zwiększyć do 3-4h

1. Masa

Silniki: 2x108g

Arduino UNO: 25 g

Ogniwa: 2x31g

Przetwornica step down: 19g

Sterownik silnika krokowego: 11g

Moduł GPS: 9g

Osprzęt dodatkowy (np. kable, kondensatory, płytka stykowa - szacunkowo): do 100g

Łącznię: do 423g

Łączna masa jest wystarczająca, by nie obciążać zbytnio statywu.

Arduino UNO: 70x55

Ogniwa: 2x 51x15

Przetwornica step down: 43x32

Sterownik silnika krokowego: 50x52

Moduł GPS: 35x26

Łączna powierzchnia: 102,66 cm2

Przy takiej powierzchni powinno się zastosować pudełko o wymiarach np. 11x11 cm oraz wysokości do 5 cm. Nie mniej należy pamiętać, że elementy nie muszą być umieszczone na jednej płytce, ale już w trójwymiarowym pudełku, co prawdopodobnie zmniejszy jego wymiary.

**Implementacja programu w wersji uproszczonej**

Poniżej znajduje się kod gotowy do wgrania na Arduino, który z wykorzystaniem wszystkich elementów teleskopu nakieruje soczewkę na gwiazdę północną (Polaris), nadążnie ją śledzi oraz odczytuje lokalizacje obserwatora i dokładny czas. Głównym celem algorytmu jest sprawdzenie poprawności działania całego programu.

#include "DualTB9051FTGMotorShield.h"

DualTB9051FTGMotorShield md;

// Definicja PINÓW do silników jest nie potrzebna. trzeba połączyć tak jak na schemacie od producenta.

#include <SoftwareSerial.h>

#include "DEV\_Config.h"

#include "L76X.h"

GNRMC GPS1;

//Tutaj też nie trzeba definiować dodatkowych pinów . Wykorzystuje piny RX i TX.

int ImpulsyNaObrot = 48\*499\*4; // ilość impulsów na jeden obrót całej osi urządzenia

int PolarisDeklinacja = 89.26411; //w stopniach

int PrawdziweZeroEnkoderaM1 = 0; //do określenia prawdziwej pozycji

//Zalozenie - silnik jeden jest do osi deklinacji, a silnik 2 do osi biegunowej

// Stosujemy enkodery inkrementalny

void setup() {

md.init(); //inicjalizacja sterownika do silników

Serial.begin(115200); //do wysylania wiadomosci do konsoli - w celach testowych

}

void loop() {

md.enableDrivers();

delay(1); // wait for drivers to be enabled so fault pins are no longer low

zerowanie\_pozycji() ; //zerowanie teleskopu i enkoderow do określonej pozycji, w celu odnalezienia swojej pozycji

while ( ImpulsyNaObrot\*PolarisDeklinacja/360 >= readM1Encoder() - PrawdziweZeroEnkoderaM1 ){ // gdy dojedzie to stanie na gwiezdzie polarnej

void setM1Speed(400) ;

}

void setM1Speed(0) ; //ustwiona os deklinacji

while(1){

while ( ImpulsyNaObrot\*PolarisDeklinacja/360 >= readM1Encoder() - PrawdziweZeroEnkoderaM1 ){ // gdy dojedzie to stanie na gwiezdzie polarnej

void setM1Speed(400) ;

}

void setM1Speed(0) ; //ustwiona os deklinacji

// nadążne ustawienia osi biegunowej

void setM2Speed(50.4);

//Odczyt czasu i położenia z GPSu

GPS1 = L76X\_Gat\_GNRMC();

Serial.print("\r\n");

Serial.print("Time:");

Serial.print(GPS1.Time\_H);

Serial.print(":");

Serial.print(GPS1.Time\_M);

Serial.print(":");

Serial.print(GPS1.Time\_S);

Serial.print("\r\n");

Serial.print(GPS1.Lon); //Długosć geograficzna

Serial.print("\r\n");

Serial.print(GPS1.Lan); //Szerokość geograficzna

Serial.print("\r\n");

}

}

void zerowanie\_pozycji()

{

while ( 0.1 >= getM1CurrentMilliamps()){ // gdy dojedzie do maksimu to stanie

void setM1Speed(-300)

}

void setM1Speed(0)

PrawdziweZeroEnkoderaM1 = readM1Encoder(); //Zresestowanie enkodera

//Silnik 2 nie wymaga kalibracji

}

Jak widać wersja uproszczona programu nie wykorzystuje odczytu czasu do nadążnego śledzenia obserwowanego obiektu na niebie. Wynika to z tego, że prościej jest lecz mniej dokładniej jest nastawić stały obrót silnika niż wykonywać obrót o 1 impuls co określony czas. Wartość 31,5 została obliczona w następujący sposób.

Wartość, dla których silnik ma maksymalne obroty: 400

Maksymalne obroty silnika: 11 obr/min

Przy przełożeniach 4:1 oraz 499:1 obrót teleskopu trwa: 0,008818 obr/min = 0,5291 obr/h = 12,6974 obr/dzień.

Poszukiwana wartość wyniesie: 400\*1[obr/dzień] /12,6974[obr/dzień] = 31,5

**Kosztorys**

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa | Sztuk | Cena za 1 szt. | Koszt |
| Arduino UNO | 1 | 92,00 zł | 92,00 zł |
| 499:1 Metal Gearmotor 25Dx73L mm LP 6V with 48 CPR Encodera firmy Pololu | 2 | 130,00 zł | 260,00 zł |
| Akmulator LG INR18650 MG1 | 2 | 7,00 zł | 14,00 zł |
| L76X GPS Module | 1 | 73,00 zł | 73,00 zł |
| Przetwornica step-down D24V150F6 | 1 | 170,00 zł | 170,00 zł |
| Sterownik TB9051FTG | 1 | 75,00 zł | 75,00 zł |
| Drobny osprzęt: kondensatory 220nF/kable itd. | - | 100,00 zł | 100,00 zł |
|  |  | Całość: | 784,00 zł |

**Porównanie z komercyjnymi rozwiązaniami**

Cena jest dosyć wysoka, ale warto porównać z komercyjnymi rozwiązaniami dostępnymi na rynku. Dla podsumowania: nasz uchwyt ma napędzane dwie osie przez silniki krokowe, kosztuje do 1000 złotych, waga wynosi 2 kg. Z funkcji dodatkowych posiada GPS, oraz komunikację bluetooth przez telefon.



 Uchwyt do teleskopu posiada obie osie sterowane silnikami krokowymi, bazę danych ponad 40 000 obiektów na niebie. Jego masa wynosi 5,4 kg. Nie posiada GPS. Cena wynosi 899$.

Rysunek Teleskop Celestron Advanced VX GoTo

Rysunek Skywatcher HEQ5

Rysunek Celestron Omni CG-4

<https://www.keuwl.com/apps/bluetoothelectronics/userguide/remote_code.html>

Uchwyty są poruszane manualnie. Jego waga wynosi 4,5 kg. Nie posiada GPS. Cena wynosi 289$.

Porównać z 3 innymi rozwiązaniami

Waga, cena, gwarancja, do czego są uchwyty,

Co zrobić:

Zdjęcia większe. Kod z krańcówkami. Poprawić schemat elektroniczny. Zamienić na Leonardo. Łączyć się przez apkę w telefonie. (Zrzut z pseudo ekranu z klawiszami co o czego) n Oraz sterowanie z panel. Bluetooth electronics.