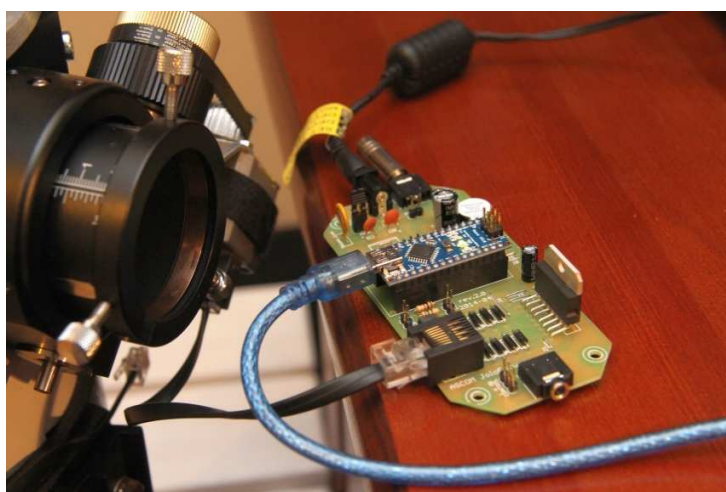




ASCOM Jolo Focuser wersja 2.1

Spis treści

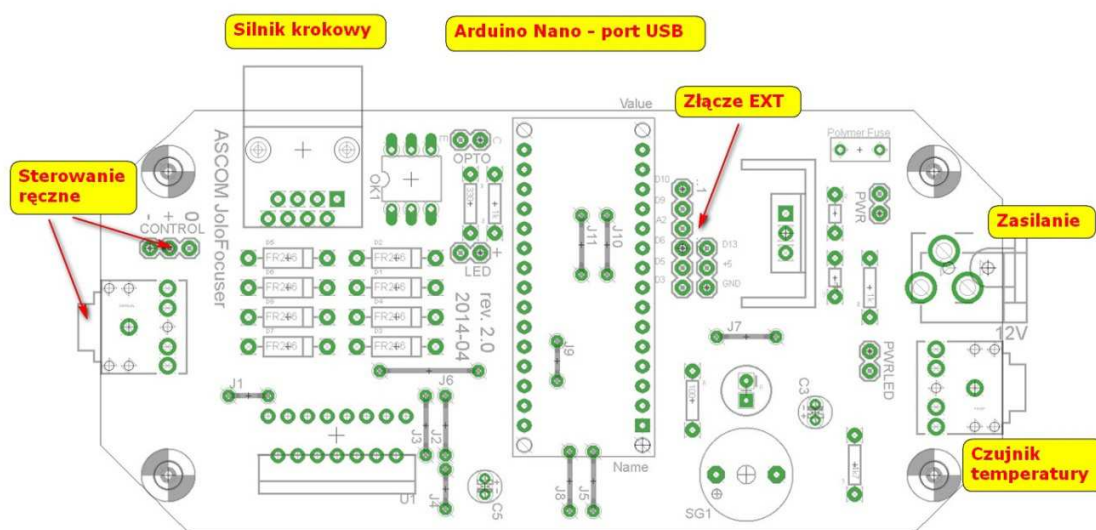
Opis układu.....	2
Podłączenie urządzenia.....	3
Czujnik temperatury	4
Brzęczyk oraz sygnalizacja świetlna	4
Sterowanie ręczne	5
Podłączenie wyświetlacza LCD	5
Silnik krokowy.....	6
Przełożenie silnika i krok focusera	7
Schemat ideowy focusera	7
Złącze dodatkowe EXT oraz OPTO	8
Aktualizowanie sterownika ASCOM.....	8
Opcje sterownika ASCOM	9
Okienko monitora	10
Aktualizowanie wsadu Arduino	11
Sterowanie mikrokrokowe	12





Opis układu

Układ oparty jest na platformie Arduino Nano 3.0 (<http://arduino.cc/en/Main/ArduinoBoardNano>) która odpowiada za sterowanie poszczególnymi komponentami urządzenia oraz za komunikację z komputerem przy użyciu wbudowanego interfejsu USB-RS232 FTDI. Po stronie komputera zainstalowany musi być sterownik ASCOM (do pobrania ze <https://github.com/sirJolo/ascom-jolo-focuser/tree/master/Downloads>) który pośredniczy w komunikacji pomiędzy układem focusera oraz aplikacjami sterującymi ustawianiem ostrości (*Maxim DL*, *Focus Max*, *APT* i inne umożliwiające sterowanie urządzeniami opartymi o interfejs ASCOM).



Rysunek przedstawia płytke urządzenia z zaznaczonymi gniazdami i złączami do wykorzystania przez użytkownika.

1. Gniazdo zasilania 5.5/2.1mm. Urządzenie może być zasilane napięciem z zakresu od 7 do 16V
2. Gniazdo jack 3.5mm stereo do podłączenia czujnika temperatury DS1820 lub czujnika temperatury i wilgotności DHT22/DHT11
3. Gniazdo USB mini (umieszczone na płytce Arduino Nano) zapewniające komunikację z komputerem
4. Gniazdo RJ45 posiadające 8 wyprowadzeń połączonych w cztery pary po dwa wyprowadzenia sterujące bipolarnym silnikiem krokowym
5. Gniazdo jack 3.5mm stereo do podłączenia dwóch przycisków do sterowania ręcznego.

Oprócz wyżej wymienionych gniazd na płytce znajdują się dodatkowo złącza:

- Złącze PWR do podłączenia dodatkowego wyłącznika zasilania
- Złącze PWRLED do podłączenia diody LED sygnalizującej zasilanie



<http://astrojolo.blogspot.com>

- Złącze EXT podłączone do wyprowadzeń modułu Arduino do wykorzystania przez użytkownika (domyślnie skonfigurowane do podłączenia typowego wyświetlacza LCD 2x16 znaków przez konwerter I²C)
- Złącze OPTO izolowane optycznie do sterowania dowolnym urządzeniem (na przykład migawką aparatu fotograficznego)
- Złącze do podłączenia diody LED sygnalizującej pracę urządzenia
- Złącze do podłączenia przycisków do sterowania ręcznego, które można umieścić np. wewnątrz obudowy urządzenia

Po podłączeniu urządzenia do portu USB komputera powinno ono zostać rozpoznane i po zainstalowaniu sterownika ASCOM będzie ono widoczne w aplikacjach przy wybieraniu dostępnych urządzeń do ustawiania ostrości jako *Jolo ASCOM focuser*. W tym momencie będzie już działała komunikacja z urządzeniem, które będzie zasilane z portu USB komputera. Po podpięciu napięcia zasilania do urządzenia oraz silnika otrzymamy w pełni działający układ. Wszystkie pozostałe komponenty są opcjonalne.

Podłączenie urządzenia

W sekcjach poniżej opisano szczegółowo sposób podłączenia poszczególnych elementów do urządzenia. Przy pierwszym uruchomieniu możemy skorzystać z poniższego scenariusza:

1. Podłączamy urządzenie kablem USB do komputera. Po chwili system wykryje urządzenie i zainstaluje odpowiedni sterownik interfejsu USB-RS232 FTDI. Jeżeli system nie zainstaluje sterowników automatycznie znajdziemy je na stronie FTDI albo w folderze *Program Files/Arduino/drivers* o ile mamy zainstalowane środowisko *Arduino IDE*. Po zainstalowaniu w naszym systemie pojawi się dodatkowy port szeregowy COM
2. W urządzeniu domyślnie zainstalowane jest oprogramowanie w wersji 2.1. Jeśli chcemy je zaktualizować albo zmienić niektóre parametry możemy to zrobić w tym momencie używając środowiska *Arduino IDE*
3. Następnie możemy zainstalować sterownik ASCOM. Po jego zainstalowaniu z dowolnego programu obsługującego focusery ASCOM (np. *Maxim DL* albo *APT*) wybieramy *Jolo ASCOM focuser* i po otwarciu okienka ustawień wprowadzamy odpowiednie wartości
 - a. Ustawiamy odpowiedni port COM
 - b. Ustawiamy obliczony przesuw wyciągu odpowiadający jednemu krokowi silnika (możemy to zrobić później)
 - c. Wprowadzamy maksymalną wartość kroku odpowiadającą całkowitemu wysunięciu wyciągu
4. Po wprowadzeniu tych wartości możemy już połączyć się z urządzeniem i sprawdzić reakcje na wprowadzane z oprogramowania polecenia
5. Następnie podłączamy do urządzenia zewnętrzne zasilanie i silnik. Po tej czynności polecenia wydawane z programu powinny skutkować odpowiednimi ruchami silnika krokowego

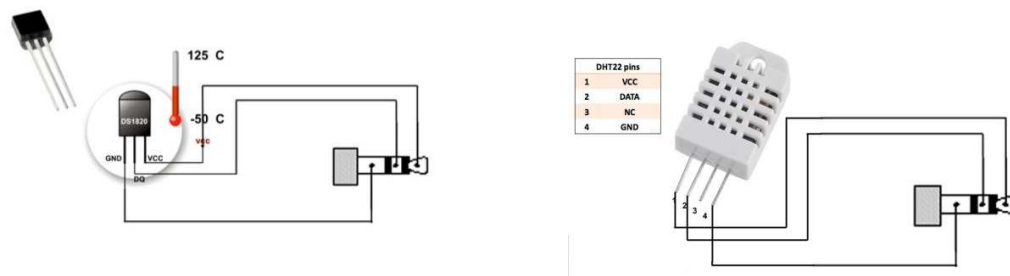


6. Kolejnym krokiem (opcjonalnym) będzie podłączenie czujnika temperatury. Po wykonaniu tej czynności program powinien wyświetlać aktualną temperaturę czujnika, który możemy umieścić w odpowiednim miejscu (na przykład przymocować do tubusu naszego teleskopu). Czujnik temperatury należy podłączać przy wyłączonym zasilaniu urządzenia, w przeciwnym razie nie zostanie on wykryty
7. Następnym opcjonalnym krokiem może być podłączenie przycisków do ręcznego sterowania wyciągu
8. Następny opcjonalny krok to podłączenie wyświetlacza LCD przy użyciu magistrali I²C i innych peryferii do złącza EXT (sterowania grzałkami albo wentylatorami przez wyjścia PWM, pomiar napięcia zasilania albo pobierania prądu, itp.)

Czujnik temperatury

Oparty jest o popularny układ DS18B20 i podłączany do gniazda jack 3.5mm na górze płytki obok gniazda zasilania. Aby układ wykrył czujnik musi on być podpięty przed włączeniem zasilania – podłączenie czujnika do włączonego urządzenia nie umożliwi nam pomiaru temperatury. Inną opcją jest zastosowanie czujnika temperatury i wilgotności DHT22. Przy prawidłowym podłączeniu czujnika urządzenie samo wykryje jego typ.

Schemat podłączenia czujników temperatury:



Brzęczyk oraz sygnalizacja świetlna

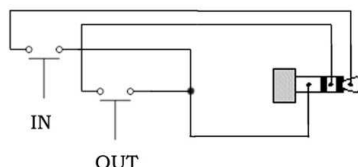
Na płytce umieszczono piezoelektryczny brzęczyk oraz złącze do podłączenia dodatkowej diody LED sygnalizującej działanie układu. Sygnalizowane są akcje takie jak: podłączenie urządzenia do komputera i nawiązanie komunikacji, zakończenie ruchu silnika, błędna komenda. W celu obniżenia głośności brzęczyka zamiast zworki R2 (obok brzęczyka) można zastosować rezystor o wartości 30-200Ω. Brzęczyk można wyłączyć odpowiednim ustawieniem w sterowniku ASCOM.



Sterowanie ręczne

Urządzenie umożliwia podpięcie dwóch przycisków do sterowania ręcznego. Naciśnięcie przycisku powoduje uruchomienie silnika krokowego z minimalną szybkością która rośnie aż do osiągnięcia maksymalnej zadanej w ustawieniach sterownika ASCOM wartości. Po zwolnieniu przycisku silnik zatrzymuje się. Krótkie naciśnięcia przycisku umożliwiają sterowanie z dokładnością do pojedynczych kroków. Przyciski można podpiąć do gniazda jack 3.5mm na dole płytki (przyciski zewnętrzne) oraz do złącza CONTROL na płytce położonego obok tego gniazda (np. dla umieszczenia przycisków wewnątrz obudowy). Można używać obu złączy jednocześnie – przyciski będą pracowały niezależnie. Bieżące położenie focusera jest aktualizowane w urządzeniu i wyświetlane w aplikacji sterującej (o ile jest podłączona). Po osiągnięciu pozycji 0 albo maksymalnej zadanej w konfiguracji sterownika ASCOM silnik zatrzymuje się.

Schemat podłączenia przycisków sterowania ręcznego:



Podłączenie wyświetlacza LCD

Do urządzenia można podpiąć typowy wyświetlacz alfanumeryczny LCD 2x16 znaków sterowany przez magistralę I²C. Niektóre dostępne wyświetlacze są sterowane bezpośrednio przez tę magistralę, jednak do praktycznie każdego można dokupić niedrogi konwerter I²C -> LCD. Do astrozastosowań polecałbym wyświetlacze z czerwonym podświetleniem zwykłe lub negatywowe, ewentualnie czerwone wyświetlacze OLED. Warto zakupić wyświetlacz z rozszerzonym zakresem pracy do -20°C. Na zdjęciu poniżej pokazano konwerter I²C zamontowany do wyświetlacza LCD:



Pin GND podłączamy do piny GND złącza EXT focusera. VCC podłączamy do +5V, SDA podłączamy do D3 a SCL do piny D5 złącza EXT. Widocznym na zdjęciu potencjometrem możemy ustawić kontrast wyświetlacza, natomiast aby obniżyć jasność podświetlenia w miejsce zworki po lewej stronie konwertera należy wlutować dodatkowy rezystor o wartości kilkunastu – kilkudziesięciu omów. Można również tutaj wpiąć tranzystor małej mocy (BC548 albo podobny) kluczowany przez jedno z wyjść PWM



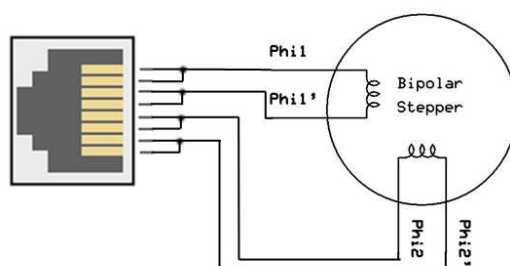
na złączu EXT i będziemy mogli sterować wtedy jasnością wyświetlania programowo. Wyświetlane na LCD informacje można konfigurować z poziomu ustawień sterownika ASCOM. W czasie przesuwania wyciągu wyświetlacz nie jest odświeżany i wyświetlany jest napis „FOCUSER MOVING”. W opcjach sterownika ASCOM można włączyć odświeżanie wyświetlacza w czasie pracy silnika krokowego, ale mogą wtedy wystąpić niewielkie zakłócenia w jego pracy (nie wpłyną one jednak na końcową pozycję wyciągu).

Silnik krokowy

Do podłączenia silnika zamontowane zostało gniazdo RJ45 z ośmioma stykami. Każde dwa sąsiadujące styki są połączone ze sobą tworząc cztery niezależne linie sterujące silnikiem bipolarnym. W urządzeniu może zostać użyty dowolny silnik krokowy bipolarny z czterema wyprowadzeniami. Silnik będzie zasilany napięciem podanym do zasilania urządzenia pomniejszonym o około 2V (taki jest spadek napięcia na układzie sterownika silnika L298). Jest możliwość użycia silnika na niższe napięcie, ale wymaga to zamontowania do płytki dodatkowego regulatora napięcia w miejscu zworki obok złącza EXT. Zasilając np. urządzenie napięciem 12V i podłączając silnik na napięcie 7V można zastosować regulator 7808 o napięciu wyjściowym 8V, które po spadku o 2V da nam 6V na silniku. Regulator należy wyposażyć w odpowiedni radiator w zależności od prądu pobieranego przez silnik. Zasilanie silnika niższym napięciem spowoduje pewien spadek momentu obrotowego oraz maksymalnej prędkości obrotowej, ale przy różnicy 20-30% nie będzie to znaczny spadek.

Układ bez modyfikacji umożliwia sterowanie silników z maksymalnym prądem 1A. Po wymianie bezpiecznika polimerowego (obok gniazda zasilania) na większy i dołożeniu niewielkiego radiatora na układ L298 można sterować silnikami z prądem do 2A. Większe obciążenie może spowodować nadmierne nagrzewanie się ścieżek na płytce i doprowadzić do jej uszkodzenia. Oczywiście zastosowany zasilacz musi zapewnić odpowiedni prąd. Zasilając układ napięciem 12V bez regulatora można stosować silniki np. 42BYGH118-01B, 39BYGH402B albo dowolny podobny bipolarny na napięcie 9-15V i prąd do 1A.

Schemat podłączenia silnika bipolarnego do urządzenia:



Jeśli po zmontowaniu układu wyciąg porusza się w odwrotnym kierunku (a więc przy sterowaniu w kierunku *In* albo przyciskiem ‘-’ porusza się na zewnątrz) należy zamienić końcówkami jedną z par

połączeń silnika (np. *Phi1* i *Phi1'*) albo w kodzie Arduino (plik *focuser.ino*) zmienić kolejność w deklaracji silnika z

```
AccelStepper stepper = AccelStepper(AccelStepper::HALF4WIRE, A5, A4, A3, A2);
```

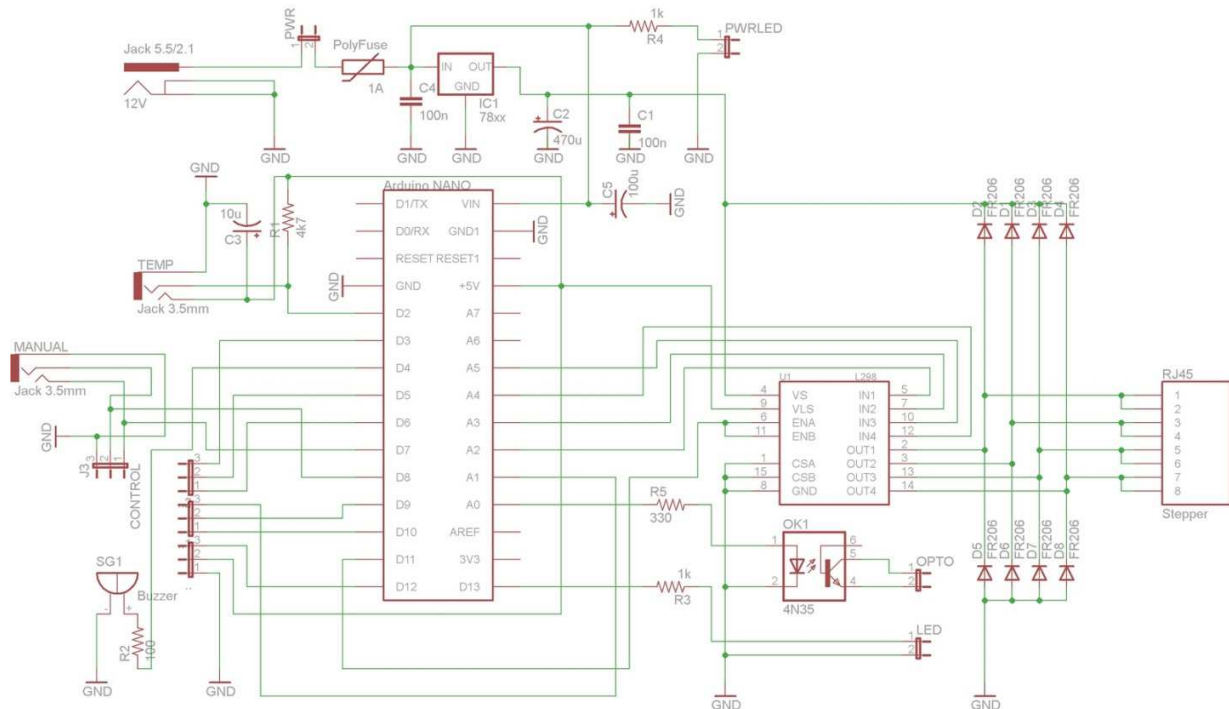
np. na:

```
AccelStepper stepper = AccelStepper(AccelStepper::HALF4WIRE, A4, A5, A3, A2);
```

Przełożenie silnika i krok focusera

Urządzenie steruje silnikiem w tzw. trybie półkrokowym. Czyli podłączając silnik o kroku równym 1.8 stopnia (200 kroków na obrót) jeden krok zostanie podzielony na dwa dodatkowe półkroki i wynikowo nasz silnik będzie wymagał 400 impulsów na pełen obrót. Dość istotnym parametrem dla automatyzacji ustawiania ostrości jest przełożenie kroku silnika na dystans pokonywany przez wyciąg. Jeśli mamy zmontowane urządzenie możemy go prosto zmierzyć suwmiarką w jednej pozycji, następnie wykonać przesunięcie na przykład o 1000 kroków i dokonać drugiego pomiaru a następnie podzielić otrzymane przesunięcie przez liczbę kroków. Możemy go również obliczyć znając całkowite przełożenie zastosowanej przekładni oraz średnicę osi przesuwającej tubus wyciągu. Zalecane jest aby rozmiar kroku był przynajmniej 3-4 razy mniejszy niż tak zwany CFZ (ang. *Critical Focus Zone*). CFZ możemy obliczyć przy użyciu prostego kalkulatora dostępnego w okienku konfiguracyjnym sterownika ASCOM focusera . Jeśli w naszym zestawie przykładowe CFZ wynosi 80µm to pojedynczy krok silnika nie powinien przesunąć wyciągu o więcej niż 25-30µm.

Schemat ideowy focusera





Złącze dodatkowe EXT oraz OPTO

Złącze EXT umieszczone jest obok płytki Arduino i zawiera 9 pinów. Umożliwiają one podpięcie do urządzenia dodatkowych urządzeń peryferyjnych takich jak np. przekaźniki włączające określone urządzenia, dodatkowe czujniki temperatury albo wilgotności, sterowanie opaskami grzewczymi na zestawie, oraz wiele innych.

W opisie złącza EXT na płytce wkradły się dwie pomyłki – zamiast pinu ~~D13~~ wyprowadzony jest pin D12, a zamiast A2 wyprowadzony jest A1.

Napięcie +5V oraz piny złącza EXT zasilane są bezpośrednio ze stabilizatora w module Arduino. Z tego powodu nie powinno się ich obciążać sumarycznie większym prądem niż 60mA, a pojedynczego pinu większym prądem niż 30mA. Dlatego sterowanie przekaźnikami czy opaskami grzewczymi musi być zrealizowane za pomocą dodatkowego klucza na tranzystorze (najlepiej polowym, np. IRLZ44) albo na przykład układzie ULN2003 (który zawiera 7 takich kluczy). Domyślnie piny złącza EXT są skonfigurowane jak poniżej:

- D3 jako SCL magistrali I²C
- D5 jako SDA magistrali I²C
- D6, D9, D10 jako sterowane z komputera wyjścia PWM (możliwość przełączenia w tryb automatyczny sterowany pomiarem wilgotności z czujnika DHT22)
- A1 jako wejście analogowe

Wyjściami PWM możemy sterować z komputera, a wartość napięcia na wejściu A1 możemy odczytywać na komputerze. Stan wszystkich tych pinów może być również pokazywany na wyświetlaczu LCD. Wejście analogowe A1 możemy np. użyć do pomiaru napięcia zasilającego nasz zestaw lub pobieranego przez zestaw prądu. W pierwszym przypadku musimy zastosować dzielnik napięcia tak, żeby napięcie na pinie A1 nie przekroczyło 5V. Do pomiaru prądu najwygodniej zastosować czujnik oparty np. na układzie ACS712.

Poniżej płytki Arduino znajduje się dwupinowe złącze oznaczone OPTO. Jest to dodatkowe złącze do sterowania na przykład migawką aparatu fotograficznego albo innego urządzenia które chcemy galwanicznie oddzielić od układu focusera. Wyjściem OPTO możemy sterować z komputera a jego stan pokazywany może być na wyświetlaczu LCD. Użytkownik może zastosować własne rozwiązania, ponieważ zarówno wsad Arduino jak i sterownik ASCOM to oprogramowanie otwarte i jest dostępne pod adresem <https://github.com/sirJolo/ascom-jolo-focuser>

Aktualizowanie sterownika ASCOM

Najnowsze i poprzednie wersje sterownika ASCOM do focusera można znaleźć na stronie https://github.com/sirJolo/ascom-jolo-focuser/tree/Production_21/Downloads. W celu instalacji albo aktualizacji drivera odpowiedni instalator należy pobrać oraz uruchomić na swoim komputerze. Po zainstalowaniu należy sprawdzić czy w okienku dialogowym sterownika pojawiła się nowa wersja (w lewym dolnym rogu okienka). Jeśli wciąż jest tam poprzednia, należy odinstalować sterownik z panelu sterowania i zainstalować nową wersję „na czysto”.



<http://astrojolo.blogspot.com>

Opcje sterownika ASCOM

Zakładka Basic

COM port – wybór portu szeregowego do komunikacji z urządzeniem

Test port – przycisk służący do sprawdzenia wybranego portu

Max focuser position – ilość kroków urządzenia odpowiadająca maksymalnemu wysunięciu wyciągu

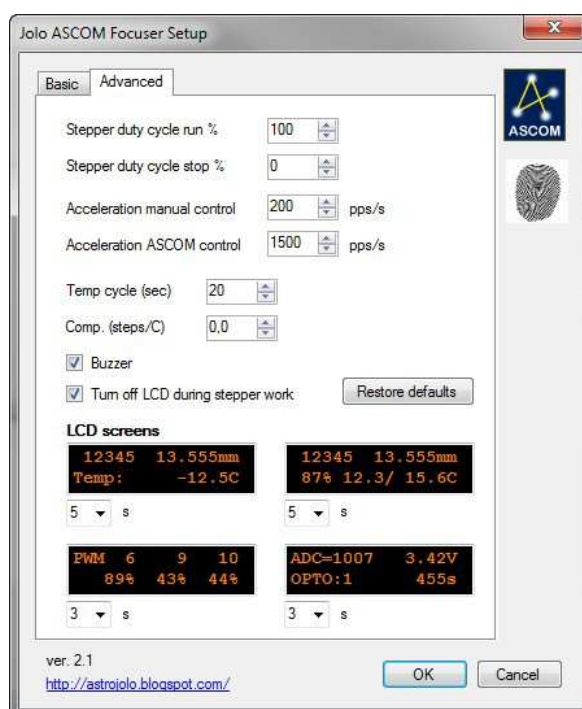
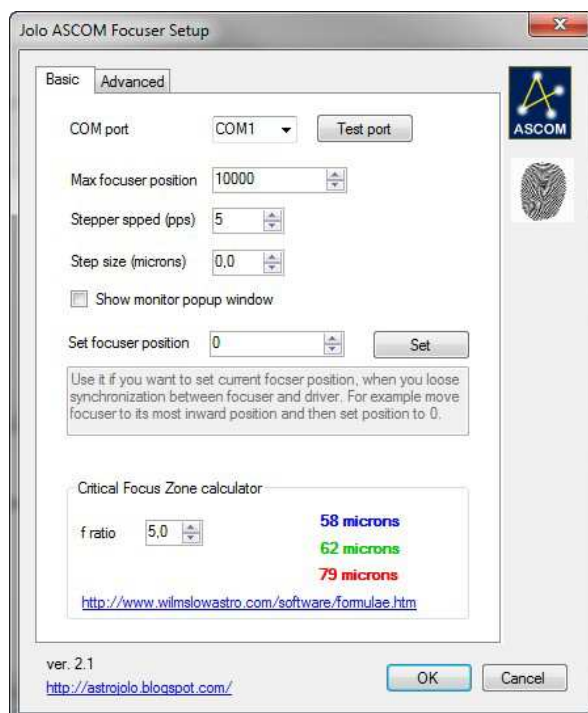
Stepper speed (pps) – prędkość obrotowa silnika krokowego w impulsach na sekundę. Przy wartości 400 i ilości kroków silnika 200 będzie on wykonywał jeden obrót na sekundę, ponieważ jest sterowany półkrokowo

Step size (microns) – przesuw wyciągu odpowiadający jednemu krokowi silnika.

Show monitor popup window – włączenie tego parametru będzie powodowało po podłączeniu urządzenia pojawienie się niewielkiego okienka do monitorowania stanu focusera w którym będziemy mieli dostęp do dodatkowych funkcji urządzenia takich jak stan wyjść PWM, OPTO oraz możliwość ich kontrolowania, a także możliwość odczytania wartości z wejścia A1 przetwornika ADC w kontrolerze

Set focuser position – możemy ręcznie ustawić w pamięci urządzenia nową wartość pozycji. Ma to głównie zastosowanie po resynchronizowaniu się rzeczywistej pozycji wyciąg z tą zapamiętaną w urządzeniu, albo przy pierwszym uruchomieniu. Po przesunięciu wyciągu do pozycji minimalnej możemy ustawić w tym polu wartość 0 i po naciśnięciu przycisku **Set** będzie ona zapamiętana w pamięci wewnętrznej kontrolera

Critical Focus Zone Calculator – to prosty kalkulator do wyznaczania rozmiaru strefy ostrości naszego instrumentu. Po wprowadzeniu światłosiły zostanie wyznaczona wartość CFZ dla trzech długości fali odpowiadających trzem podstawowym barwom





<http://astrojolo.blogspot.com>

Zakładka Advanced

Stepper duty cycle run – tutaj określimy z jaką mocą ma pracować silnik krokowy podczas przesuwania wyciągu

Stepper duty cycle stop – tutaj wprowadzamy wartość momentu hamującego silnika w czasie spoczynku. W większości przypadków możemy pozostawić wartość domyślną

Acceleration manual control – to przyspieszenie z jakim rośnie prędkość obrotowa silnika w czasie ręcznego sterowania urządzeniem

Acceleration ASCOM control – to przyspieszenie z jakim rośnie prędkość obrotowa silnika w czasie sterowania przez sterownik ASCOM

Temp cycle (sec) – czas pomiędzy kolejnymi odczytami temperatury z czujnika dla kompensacji

Comp. (steps/C) – wartość kompensacji temperaturowej urządzenia. Wprowadzając np. wartość 35 dla przy zmianie temperatury o 1 stopień wyciąg przesunie się o 35 kroków. Oczywiście wartości ujemne są również dopuszczalne. Ustawienie wartości na 0 spowoduje wyłączenie kompensacji

Buzzer – odznaczenie tego pola wyłączy nam sygnalizację akcji brzęczykiem

Turn off LCD during stepper work – wyłącza odświeżanie podłączonego wyświetlacza LCD podczas pracy silnika krokowego. Odświeżanie wyświetlacza LCD zabiera dość sporo czasu, co skutkuje nierównomierną pracą silnika krokowego

Restore defaults – powoduje przywrócenie wartości domyślnych na zakładce *Advanced* ustawień sterownika

LCD Screens – jeśli do urządzenia podpięty jest wyświetlacz LCD możemy tutaj wybrać które z predefiniowanych ekranów przez jaki czas mają być wyświetlane. Po wybraniu wartości 0 dany ekran nie będzie pokazywany w ogóle

Okienko monitora

Jeśli w ustawieniach sterownika zaznaczymy opcję *Show monitor popup window* wtedy po podłączeniu urządzenia pojawi się niewielkie okienko umożliwiające kontrolę focusera, a także dostęp do kilku dodatkowych funkcji urządzenia.

Zakładka Monitor

Position – wyświetla bieżącą pozycję focusera w krokach, oraz wysuw w milimetrach w oparciu o rozmiar kroku zapisany w ustawieniach sterownika

GO, STOP, -, + - umożliwiają podstawowe operacje focusera: przestawianie w zadaną pozycję, przesuwanie o zadaną ilość kroków oraz zatrzymanie silnika w trakcie działania

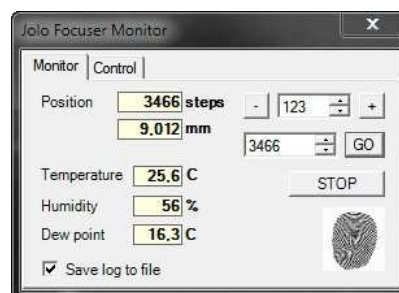


<http://astrojolo.blogspot.com>

Temperature, Humidity, Dewpoint – wyświetlają bieżącą temperaturę, wilgotność oraz obliczoną temperaturę punktu rosy. Dwie ostatnie wartości dostępne są tylko przy podpiętym czujniku DHT22

Save log to file – włącza logowanie danych do pliku. Co 10 sekund zapisywane są następujące wartości: pozycja wyciągu, temperatura, wilgotność, temperatura punktu rosy, wartości wyjść PWM, wartość odczytana z wejścia analogowego oraz stan wyjścia OPTO. Pliki logów zapisywane są w folderze *Moje dokumenty\JOLOFocuser*:

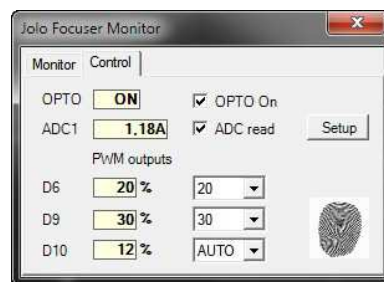
```
Time;position;temperature;humidity;dewpoint;pwm6;pwm9;pwm10;adc;opto
2014-07-09 20:49:13;3466; 25.7; 64; 18.4; 30; 80; 28;716;1
2014-07-09 20:49:23;3466; 25.6; 64; 18.3; 30; 80; 28;718;1
2014-07-09 20:49:43;3476; 25.6; 64; 18.3; 30; 80; 28;712;1
```



Zakładka Control

OPTO On – służy do włączania i wyłączenia wyjścia OPTO w urządzeniu. Pole *OPTO* wyświetla bieżący stan tego wyjścia

ADC Read – służy do włączenia pomiaru na wejściu analogowym A1 urządzenia. Pole *ADC1* wyświetla bieżący pomiar, a sposób i wyświetlane wartości możemy ustalić po kliknięciu na przycisk *Setup*. Przykładowo dla pomiaru napięcia przez dzielnik złożony z rezystorów 1 i 3kΩ należy w polu *A* wpisać wartość '0.0195', w polu *B* '0' a w polu *Suffix* 'V'



PWM outputs – wyświetla bieżące wartości wypełnienia sygnału PWM na wyjściach 6, 9 i 10. Rozwijane listy umożliwiają ustawienie wartości PWM w %, a po ustawieniu *AUTO* wartość PWM będzie obliczana na podstawie bieżącej wilgotności

Aktualizowanie wsadu Arduino

Aby zaktualizować firmware kontrolera Arduino będziemy potrzebowali:

1. Plik *Arduino sketch x.x.zip* w odpowiedniej wersji
2. Plik *Arduino libs x.x.zip* w odpowiedniej wersji
3. Arduino IDE

Dwa pierwsze pobierzemy ze strony <https://github.com/sirJolo/ascom-jolo-focuser/tree/master/Downloads> , natomiast IDE pobierzemy ze strony <http://arduino.cc/en/Main/Software> . Po zainstalowaniu IDE biblioteki (plik *Arduino libs*) należy rozpakować do folderu *Moje Dokumenty\Arduino\libraries* i następnie uruchamiamy IDE, otwieramy plik

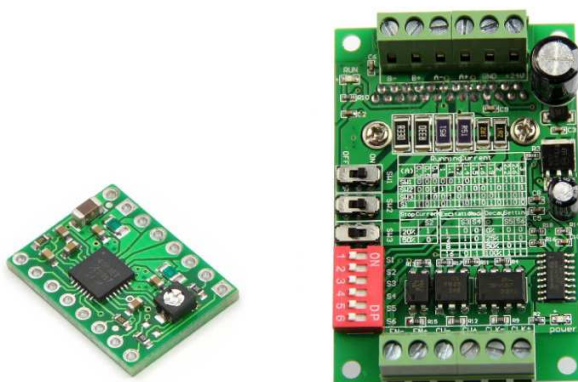


<http://astrojolo.blogspot.com>

focuser.ino z rozpakowanego archiwum *Arduino sketch*, kompilujemy i po podłączeniu urządzenia kablem USB do komputera ładujemy firmware do układu Arduino. Do tej operacji nie jest wymagane podłączanie zewnętrznego zasilania – moduł Arduino będzie zasilany z gniazda USB. Jeśli dla własnych potrzeb zmienialiśmy parametry konfiguracyjne wsadu Arduino, należy przed załadowaniem zmienić je również w nowej wersji (patrz sekcja poniżej).

Sterowanie mikrokrokowe

Urządzenie domyślnie pracuje w trybie półkrokowym. Jest możliwa modyfikacja urządzenia i zastosowanie dodatkowego kontrolera który umożliwi sterowanie mikrokrokowe jeśli potrzebna jest większa precyzja ustawiania ostrości. Modyfikacja polega na wylutowaniu z płytki sterownika L298, umieszczeniu w odpowiednich otworach pinów i podpięcie do nich dodatkowego kontrolera ze sterowaniem mikrokrokowym na przykład opartego na popularnym układzie A4988 albo TB6560:



Powyżej po lewej mikro moduł z układem A4988 umożliwiający sterowanie z dokładnością do 1/16 kroku z maksymalnym prądem 1A (do 2A z radiatorem). Po prawej moduł z układem TB6560 dający taką samą dokładność, ale więcej możliwości konfiguracji oraz większy prąd maksymalny (do 3A). Moduły takie można zakupić na allegro lub w sklepach elektronicznych w cenie około 30-50zł. Są również dostępne gotowe moduły sterujące z większą dokładnością (1/64 albo nawet 1/128 kroku) ale ich ceny są mało atrakcyjne.

W celu podłączenia modułu konieczne jest poprowadzenie 9 dodatkowych przewodów między układem focusera oraz dodatkowym kontrolerem:

1. 2 przewody doprowadzające zasilanie
2. Sygnał CLK taktujący sterownik (Arduino pin A4)
3. Sygnał DIR wskazujący kierunek obrotu (Arduino pin A3)
4. Sygnał ENABLE kluczący zasilanie silnika (Arduino pin D6)
5. 4 przewody zasilające silnik krokowy

Ostatnią czynnością będzie zmiana konfiguracji silnika krokowego w kodzie Arduino:

```
AccelStepper stepper = AccelStepper(AccelStepper::DRIVER, A4, A3)
```