

1. Podłączenie Arduino

- W pdfi-e schemat-podlacznie przedstawione jest jak **połączone są elementy**, które PINy na arduino itd. Dodatkowo rozpiska z pinami znajduje się w pliku **config.h** w folderze arduino. Wydaje mi się, że jak pakowałem Panu do pudełka ten prototyp to źle podpiąłem endstopy, więc trzeba sprawdzić wszystkie PINy.
- Na pewno trzeba będzie **zasilac serwo z oddzielnego stabilizatora 5V**, bo zakłóca pracę sterowników silnika jeśli jest podłączony do tego samego zasilania z nimi, także podłączenie serwa do zasilania Arduino też się nie sprawdzi.
- W tym moim prototypie, który Panu przyniosłem Arduino było **zasilane z portu USB** komputera (kabel nie służył do przesyłania Gcode, bo ten plik trzeba najpierw nagrać na kartę microSD na komputerze, a następnie włożyć w czytniki kart podpięty pod arduino), więc jeśli chcemy zasilać bez PC, potrzebny będzie **2-gi zasilacz albo połączenie zasilania 12V silników z zasilaniem Arduino**. Najpierw podłączamy Arduino i następnie silniki, a gdy odłączamy to najpierw silniki, a później Arduino.
- Klasycznie przed podpięciem nowego silnika trzeba ustawić odpowiednie napięcie/nateżenie prądu na sterowniku silnika, ale to chyba raczej Pan wie.
(<https://howtomechatronics.com/tutorials/arduino/how-to-control-stepper-motor-with-a4988-driver-and-arduino/>)

2. Program Arduino

- Do działania mogą być **niezbędne dodatkowe biblioteki**, nie pamiętam już, które trzeba było ściągnąć, a które są oryginalnie w oprogramowaniu, ale potrzeba:

SoftwareServo: <http://www.arduino.cc/playground/ComponentLib/Servo>

TimerOne: <http://www.arduino.cc/playground/Code/Timer1>

Keypad.h <https://playground.arduino.cc/Code/Keypad>

LiquidCrystal.h <https://playground.arduino.cc/Main/LiquidCrystal>

SD.h <https://www.arduino.cc/en/Reference/SD>

SPI.h <https://www.arduino.cc/en/reference/SPI>

- Program powstał na podstawie oprogramowanie do SphereBota:
<http://pleasantsoftware.com/developer/3d/spherebot/>
<https://github.com/zaggo/SphereBot>
- Plik config.h zawiera ustawienia, wartości domyślne i podłączenia PINów wszystkich elementów układu
- Plik StepperModel zawiera funkcje dotyczące silników
- Plik wycinarka_plazmowa zawiera główny kod programu, aby uruchomić i wgrać program na arduino, **trzeba zmienić nazwę folderu z arduino na wycinarka_plazmowa**.
- **Aby program dobrze działał trzeba ustawić odpowiednią średnicę dla silnika X (wzdłużnego), (XAXIS_DEFAULT_DIAMETER), którą tak trzeba dobrać, by jeden obrót silnika był równoznaczny z wartością $\pi \cdot \text{domyślnaŚrednica}$ i będzie to wynikać z przełożeniu ruchu obrotowego silnika X, na ruch obrotowy nakrętki na śrubie**

trapezowej, a więc ruch posuwisty całego układu wzdłuż osi X. Wartość tą można także zmienić za pomocą komendy Gcode *M400S...* Jeśli będą jeszcze jakieś przełożenia to można je uwzględnić jako gearRatio.

- W przypadku osi Y wartość średnicy do obliczeń posuwu I innych parametrów jest równoznaczna średnicy ciętej rury I jest zmieniana za pomocą komendy *M401S...* uwzględnionej w makrze do generacji Gcode.
- **Należy ustawić przełożenie mechaniczne silników GEAR_RATIO w pliku config.h, ja zostawiłem wartość 1**
- Ustawić mikrokroki zmieniając wartości VMS... w config.h, ustawić microstepping itd., więc opisu można znaleźć w pliku config.h, ale w miarę dokładnie opisałem wszystkie pliki.

• Dostępne komendy GCODE

G0X...Y...	Szybki ruch na współrzędne X i Y
G1X...Y...F...	Ruch roboczy na współrzędne X I Y, F- szybkość posuwu w mm/min (UWAGA NIE testowałem zmian posuwu, tylko komendy G1X..Y..)
G2X...Y...I...J...	Ruch po okręgu zgodnie z wskazówkami zegara, do punktu X,Y o środka w I,J (UWAGA, komenda dodana przeze mnie, więc też wymaga testów)
G3X...Y...I...J...	Ruch po okręgu przeciwnie do wskazówek zegara, do punktu X,Y o środka w I,J
G20	Oś Y wyrażona w mm, by odpowiednio odczytywać Gcode wyrażony w mm
G21	Oś Y wyrażona w radianach, by odpowiednio odczytywać Gcode wyrażony w radianach
G22	Oś Y wyrażona w stopniach , by odpowiednio odczytywać Gcode wyrażony w stopniach
G4P...	Przerwa ... milisekund
G90	Pozycjonowanie absolutne
G91	Pozycjonowanie przyrostowe
G92	AutoHoming - zerowanie silników
M18	Wyłączenie silników
M300S...	Ustawienie serwa od 0-180
M400S...	Zmiana domyślnej średnicy osi X (średnica wynika z konstrukcji przełożenia ruchu obrotowego na postępowy)
M401S...	Zmiana domyślnej średnicy osi Y (średnica==średnica ciętej rury)

- Odnośnie trybu wprowadzania z klawiatury do załączyłem w folderze literatura opis działania modułu oraz artykuł, na którym się wzorowałem wyprowadzając zależności

matematyczne na rozwinięcie krzywej na cylindrze.

3. Makra do FreeCada

- Instrukcja jak wgrać makra do FreeCada znajduje się w folderze FreeCadMakro
- By edytować I lepiej widzieć jak wygląda kod można zmienić rozszerzenie z *.FCMacro na *.py I otworzyć kod w programie ze wsparciem składni Pythona
- Okienka do wprowadzania danych we FreeCadzie zrobiono z użyciem PySide
<https://www.freecadweb.org/wiki/PySide>
[https://www.freecadweb.org/wiki/PySide Beginner Examples](https://www.freecadweb.org/wiki/PySide_Beginner_Examples)
[https://www.freecadweb.org/wiki/PySide Medium Examples](https://www.freecadweb.org/wiki/PySide_Medium_Examples)
[https://www.freecadweb.org/wiki/PySide Advanced Examples](https://www.freecadweb.org/wiki/PySide_Advanced_Examples)
- Podstawowy tutorial odnośnie skryptowania we FreeCadzie w celu tworzenia I manipulacji geometrią:
[https://www.freecadweb.org/wiki/FreeCAD Scripting Basics](https://www.freecadweb.org/wiki/FreeCAD_Scripting_Basics)
[https://www.freecadweb.org/wiki/Topological data scripting](https://www.freecadweb.org/wiki/Topological_data_scripting)
[https://www.freecadweb.org/wiki/Scripting examples](https://www.freecadweb.org/wiki/Scripting_examples)
- Przed uruchomieniem makra do generacji Gcode trzeba mieć zaznaczone krzywe, z których będziemy generować Gcode
- Jeśli będą problemy z zapisem to na sztywno trzeba będzie ustawić ścieżkę, w której będzie zapisywany nasz Gcode (jest to zmienna path, kilka linijek od końca skryptu go generacji Gcode).
- Trzeba też będzie wpisać kąt serwa dla włączonego I wyłączzonego palnika (servoON, servoOFF)
- Po generacji Gcode kopiujemy nasz plik z rozszerzeniem txt oraz tworzymy drugi plik z Gcode, tylko bez rozszerzenia txt I oba pliki kopiujemy na kartę microSD I wkładamy do naszego czytnika microSD w Arduino.

4. Ogólnie musi Pan się zagłębić trochę w kod I od arduino I od makra, ale w miarę dużo dodałem komentarzy do kodu.