

Cięcie CNC gorącym drutem Jedicut-Alden SuperLuks V3b



AERODEN Jedicut-Alden SL

Ta nowa wersja pozostaje praktycznie identyczna z działaniem poprzedniej wersji. Z drugiej strony nowy szkic jest inny i znacznie wydajniejszy w przetwarzaniu poleceń wydawanych przez Jedicut. W wersji [2.4.1 Jedicut](#) operacja może zostać zablokowana na niektórych komputerach ze starym szkicem. Nowy szkic nosi nazwę "[LMFAO_V4_7_0](#)". Dwóch modelarzy (Vincent i Ollivier) dokonało przeglądu części komunikacyjnej USB i dekodowania poleceń wydanych przez Jedicut bez korzystania z bibliotek Arduino, co oszczędza czas programu.

Łącząc ten szkic z moim dla całej automatyki i opcji, otrzymujemy bardzo potężny nowy szkic. Sterowanie ogrzewaniem jest możliwe w trybie ręcznym i PC.

Dla tych, którzy już przeszli na Super Luks, pozostaje tylko przestanie i skonfigurowanie szkicu.

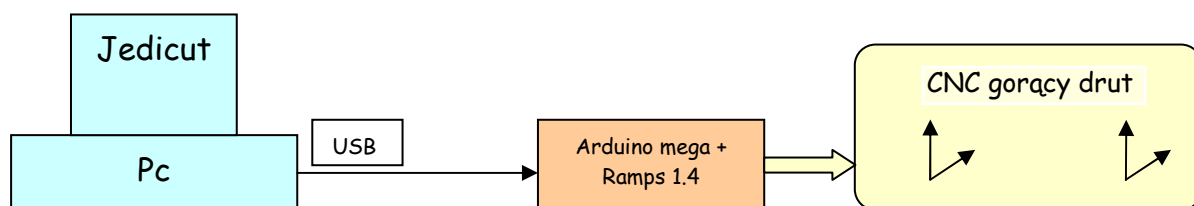
Ten dokument z pewnością nie jest idealny, starałem się udostępnić go wszystkim zainteresowanym, aby mogli bez problemu zrealizować swoją maszynę. Bardziej doświadczeni wykonują próby, ale i także zwracają mi uwagę na możliwe błędy. Kompletny plik [jedicut-Alden_SuperLuxe3.zip](#) zawiera całą niezbędną dokumentację do realizacji. Przed rozpoczęciem należy przeczytać cały podręcznik. Znajomość Arduino, karty Ramps 1.4 i środowiska Arduino IDE nie są szczegółowo opisane w tej instrukcji. [Dlatego przeczytaj "Arduino_Ramps 1.4_IDE_3.pdf"](#)

Niniejsza instrukcja umożliwia bezpośrednie wykonanie elektroniki Jedicut Alden Super Luks lub modyfikację Jedicut Alden Luks, aby uzyskać możliwości Super Luks. Zasadnicza różnica polega na zarządzaniu wyłącznikami krańcowymi (możliwe zastosowanie różnych rodzajów wyłączników krańcowych, aby powrócić do pozycji zerowej i aby skorzystać z wstępnego pozycjonowania początku cięcia.

Ograniczenie: Uwaga ten interfejs działa poprawnie pod warunkiem, że maszyna z gorącym drutem CNC jest wyposażona w 4 osie o identycznej charakterystyce (silnik krokowy ze śrubami / nakrętkami lub silnik krokowy z kołem zębatym i pasem) nie można mieszać charakterystyk.

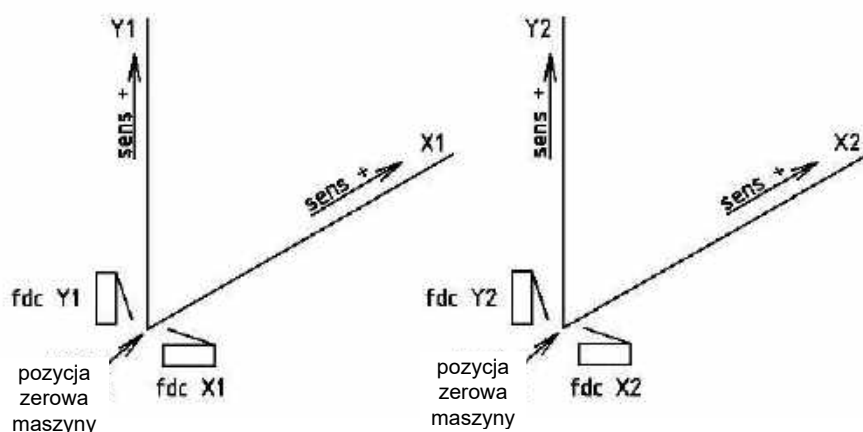
Życzę Ci dobrej zabawy.

Przypomnienie:

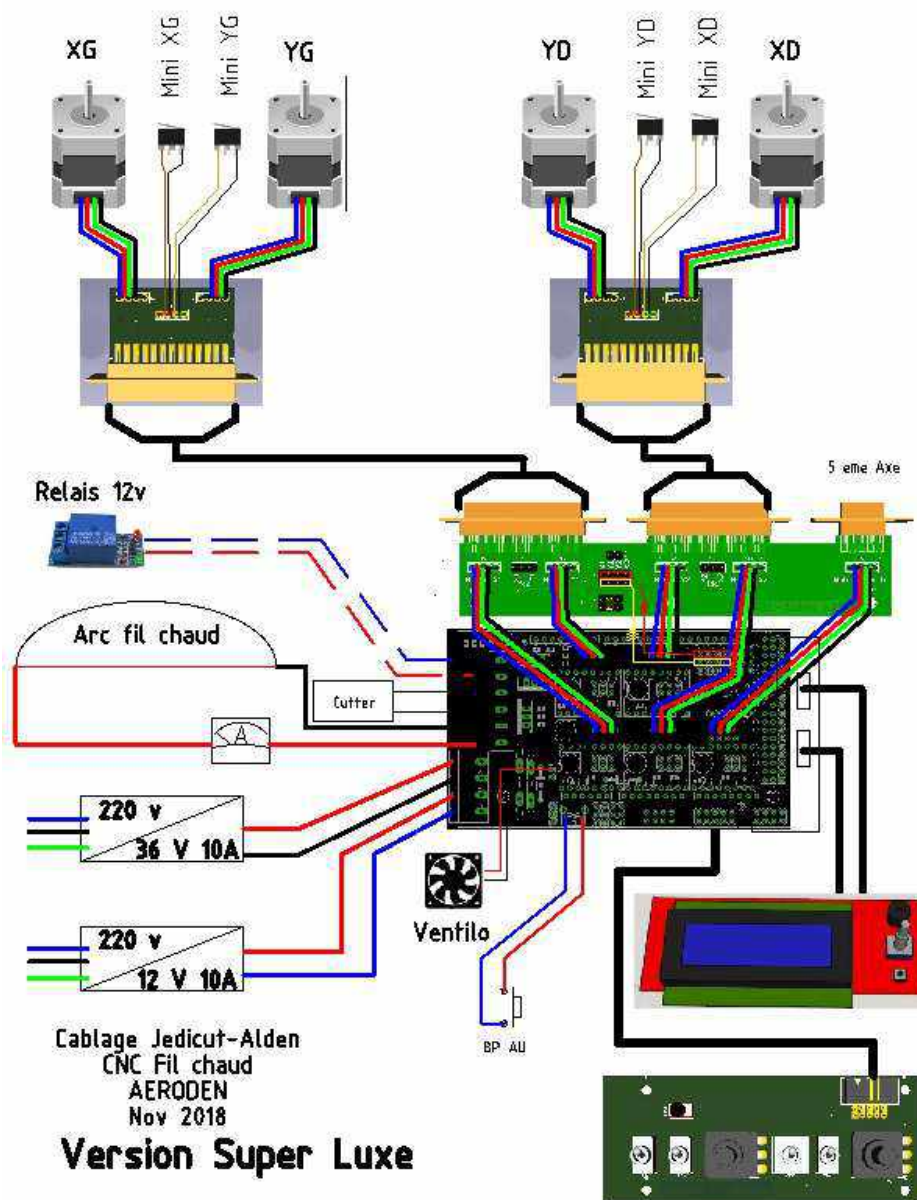


Powszechne korzystanie z drukarek 3D wprowadziło na rynek niedrogą i o wysokiej wydajności elektronikę do sterowania silnikami krokowymi. To jest opcja, którą proponuję

Ustawienie wózków X1-Y1 i X2-Y2. Kiedy stoisz przed maszyną, wózki X1 i Y1 są po lewej stronie.



Okablowanie wersji SuperLuks:



Wersja SuperLuks: zasilacz 12V, zasilanie 36V dla drutuu łuku, amperomierz, Arduino Mega, Ramps1.4, sterowanie PWM, wentylator sterownika, wyświetlacz LCD 4 linie po 20 znaków (**wyświetlacz graficzny nie jest zgodny**). Na ekranie LCD wyświetlana jest zadana wartość grzania drutu, zadana wartość prędkości, wartości konfiguracyjne, ręczne sterowanie ogrzewaniem drutu i noża, przesyłanie poleceń z komputera, wyłączanie silników krokowych, wyświetlanie wartości ręcznych elementów sterujących. Zarządzanie wyłącznikami krańcowymi (**fdc**), które akceptuje wszystkie typy urządzeń **fdc**. Umożliwia to ustawienie bazowania i pozycjonowanie wstępne. Został dodany ekran pomocy dotyczący wprowadzenia obsługi **fdc**.

Sterowanie PWM grzania drutu.

Sterowanie przełącznikiem odcięcia grzałki, przełącznik ten służy do sterowania odcinaniem grzania drutu, i jest używany przez tych, którzy mają ustawienie dla zewnętrznego grzania, lub tych, którzy nie ufają sterowaniu PWM i umieszczają styk przełącznika szeregowo z łukiem.

Podczas inicjalizacji:

- daje możliwość sprawdzenia statusu **fdc** (wyłączniki krańcowe).
- wyświetla parametry konfiguracji maszyny.

Obowiązkowe jest bazowanie osi po każdym włączeniu zasilania i za każdym razem, gdy silniki są wyłączone przez przełącznik STOP.

Wybór wstępnego pozycjonowania po bazowaniu w ekranie konfiguracji.

Wybór pozostawienia włączonych silników (bardzo przydatne w przypadku maszyn pasowych).

Podczas cięcia wyświetla prędkość cięcia i moc grzania drutu w %.

Podczas cięcia grzanie można regulować potencjometrem.

Zatrzymanie awaryjne maszyny odbywa się poprzez manipulację jednym z 2 przełączników (TRYB, SILNIK WŁ/WYŁ lub zadziałanie **fdc**. (Cięcie nie może zostać ponownie uruchomione).

Umożliwione jest ręczne podgrzewanie drutu do czyszczenia lub opuszczenia bloku po awarii. Ręczne grzanie drutu jest dozwolone podczas cięcia, co jest bardzo interesujące przy wyszukiwaniu parametrów materiałowych.

Z wyłączonym trybem PC, pozwala na sterowanie grzaniem dodatkowej przecinarki elektrycznej, (na przykład do cięcia wnek na serwo).

Wybór ręcznego sterowania grzaniem drutu przez potencjometr lub enkoder obrotowy. Sugeruję wybór potencjometru, gdyż może się zdarzyć, że następna wersja skorzysta z enkodera obrotowego w celu uzyskania dostępu do menu i zmian parametrów.

Wybór sterowania grzaniem przez obliczenie prędkości drutu na trajektorii. Nie używaj tej opcji, gdy dynamiczne grzanie Jedicut będzie działał na USB.

Wyłącznik alarmu alarmu dźwiękowego.

Wyłączniki krańcowe są niezbędne dla wersji Super Luks.

Bez indywidualnych wyłączników krańcowych jest to wersja luksusowa bez opcji bazowania i wstępnego pozycjonowania.

Ta wersja używa tej samej obudowy, plany w plikach **.dxf** są częścią tego folderu.

Płytki drukowane są takie same jak dla wersji Luks, dostępne są pod następującym adresem:

"alain@aeroden.fr", kompletne plany są dostępne do pobrania na stronie "<https://www.jedicut.com/>" oraz pod adresem "<https://alainfelixdenis.wordpress.com/>".

W wersji Super Luks wymagana jest płytka drukowana panelu sterowania ręcznego.

Specyfikacja :

Wersja	Super Luks	Luks	Prosta z LCD
Zasilacz 12V 10A	x	x	x
Zasilacz 36V 10A	x	x	x
Arduino + ramps + sterownik	x	x	x
Arduino + ramps + ster. + wyświetlacz + kable	x	x	x
Elementy sterowania ręcznego	x	x	
Test kroków	x	x	x
Amperomierz	x	x	x
mechaniczny fdc	x	x	
mechaniczny fdc na płytce	x		
Wybór języka	x	x	x
Odwroćcie obrotów silników	x	x	x
Zatrzymanie przez fdc	x	x	
Odwroćcie odblokowania sterowników	x	x	x
Wybór podtrzymania silników	x	x	x
Odwroćcie działania fdc	x		
optyczny fdc	x		
optyczny fdc na płytce	x		
Pozycjonowanie	x		
Pozycjonowanie wstępne	x		
Pełne sterowanie grzaniem	x	x	
Szkic Arduino	LMFAO_V4_5_0	LMFAO_V4_5_0	LMFAO_V1_4_0

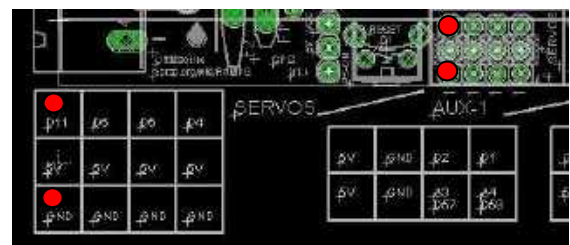
12.04.2019 - zmieniono szkice na
07.11.2019 - nowy szkic

LMFAO_V4_6_0
LMFAO_V4_7_0

LMFAO_V4_6_0
LMFAO_V4_7_0

LMFAO_V1_5_0
brak

Wyciągnąłem na kołki złącza "SERVOS" w RAMPS 1.4 (D11) kopię impulsów krokowych X1, pozwala to na kontrolę za pomocą oscyloskopu. Punkty testowe D11 i GND są reprezentowane przez czerwone kropki.

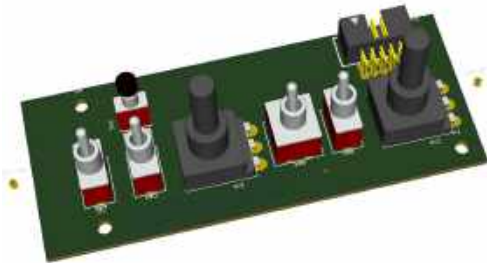


Jeśli budujesz CNC z napędem pasowym, wybierz sterowniki DRV8825 zamiast A4988, ponieważ masz większą możliwość dopasowania mikrokroków. (Do 1/32 kroków)

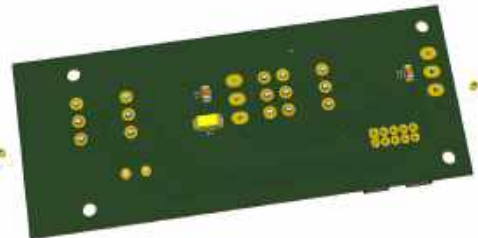
Dla obliczania ustawień zgodnie z mechaniką, sporządzono pomoc, która znajduje się w podfolderze "Calcul des transmissions", pliku do pobrania "jedicut-Alden_SuperLuxe3.zip"
Lub jako osobny plik "CNC_Calcul_chaine_cinématique.zip" do pobrania tutaj:

https://drive.google.com/open?id=1qNQppsRI-tL_dDOI4VHaxyTbLJSHEqS

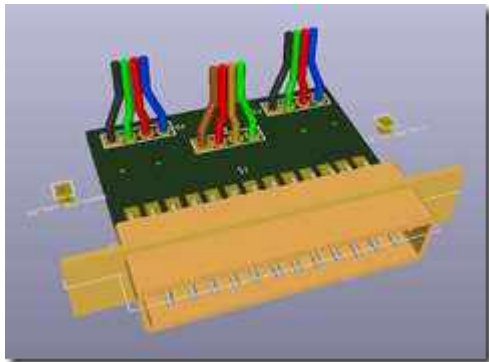
Oto widok opracowanych obwodów drukowanych:



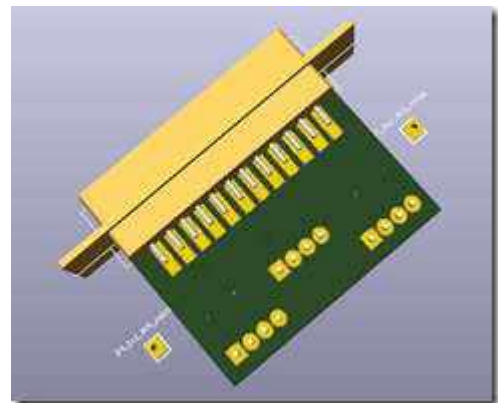
Płyta sterowania ręcznego z płaskim złączem kablowym do ramps 1.4



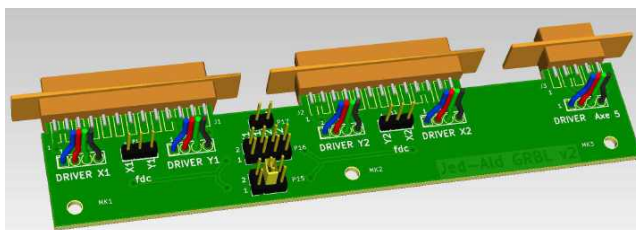
Płyta sterowania ręcznego po stronie miedzi



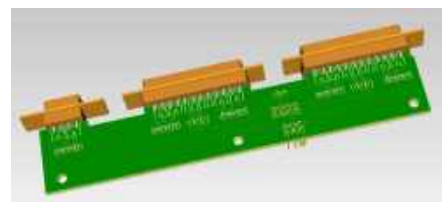
Złącze DB25 dla wózków X1-Y1 lub X2-Y2 i wyłączników krańcowych.



Dla silników piny są podwójne dla każdego przewodu. Zajmuje to 2 płytki z DB25.



Dwa złącza DB25 do podłączenia silników i krańcówek X1-Y1 i X2-Y2. oraz DB9 dla możliwej piątej osi.



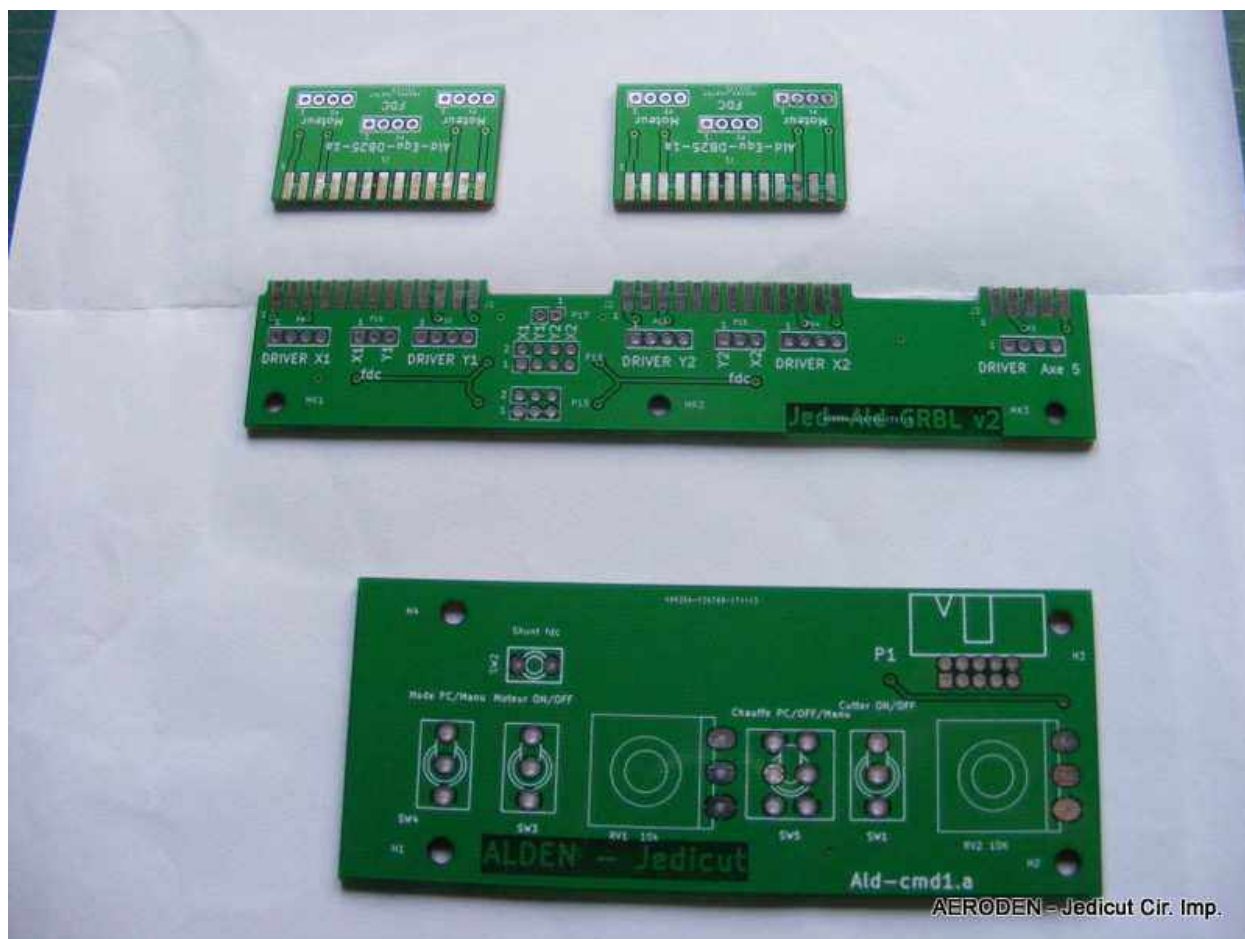
Używane Oprogramowanie
KiCad v 4.0.1

Korzystam z oprogramowania [KiCad](https://www.kicad.org/) od wielu lat, nowa wersja oferuje wyjątkowe możliwości routingu, przyjemnie jest robić skomplikowane płytki. Zanim opanowałem nową wersję, miałem trochę kłopotów, ważne jest, aby ćwiczyć, ale wynik jest zadowalający.

Generowanie plików gerber dla płytki dwustronnej, plik wiercenia, plik konturu jest realizowany przez [KiCad](#). Pliki te są bezpośrednio czytane przez oprogramowanie [CamBam](#). Generuję 4 pliki kodu do mojej frezarki CNC.

W folderze "[Jed-alden-electronique](#)" i podfolderze "[Schema](#)" znajdziesz wszystkie pliki w formacie *.pdf i *.jpg, dla tych, którzy nie znają oprogramowania Kicad. Pliki te zawierają widok ścieżek i wiercenia i wymiary płytek

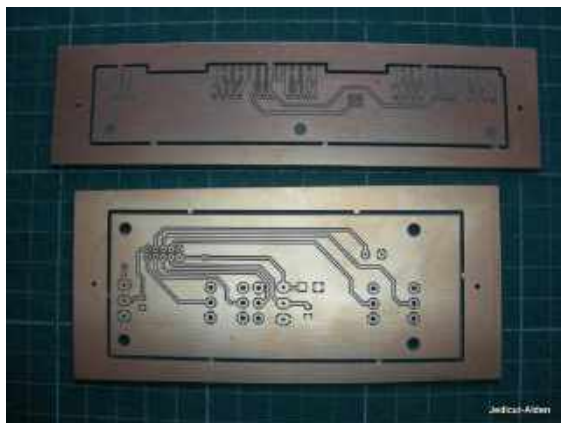
KiCad generuje niezbędne pliki do zlecenia produkcji PCB, próbowałem na zestawie 10 kart, tydzień terminów, metalizowane obwody otworów z sitodrukiem w bardzo przystępnej cenie dla producentów modeli, więcej niż chemikalia. Tak więc udostępniłem te obwody producentom gorącego drutu CNC. Płytki są dostępne pisząc do mnie pod adresem "alain@aeroden.fr"



Realizacja obwodów elektronicznych:

Realizuję swoje płytki grawerując CNC z płyt epoksydowych o grubości 15/10 mm.

Z KICAD w CamBam dla Gcode, a następnie na frezarce CNC z 0,2 mm frezem 30 °, 25000 rpm, 800 mm/min, 1 przejście 0,1 mm. Kontury frez diamentowy 1 mm w 2 przejściach.



Płytki wychodzące z frezarki CNC



Płytki są odtłuszczone acetonem i cynowane na zimno.



Płytką sterowania ręcznego.



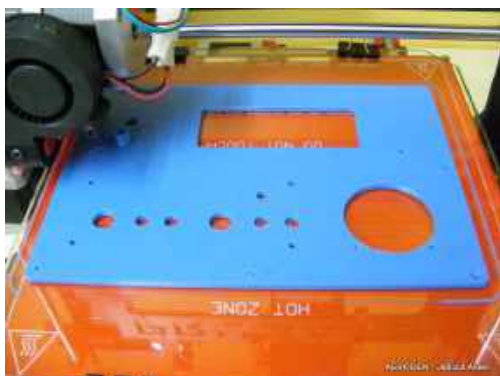
Spód płytki sterowania ręcznego



Przygotowanie taśm kabli i gniazd do połączenia z Ramps 1.4



Do DB25 przewody są przylutowane



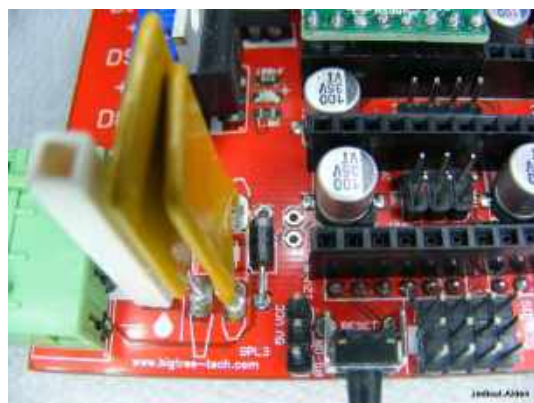
Drukowana płyta czołowa o grubości 2,6 mm. .



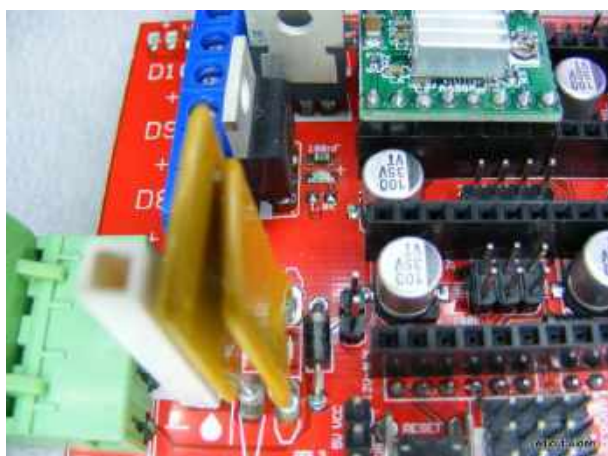
Płyta czołowa, amperomierz wskazuje natężenie prądu przepływającego przez gorący drut.



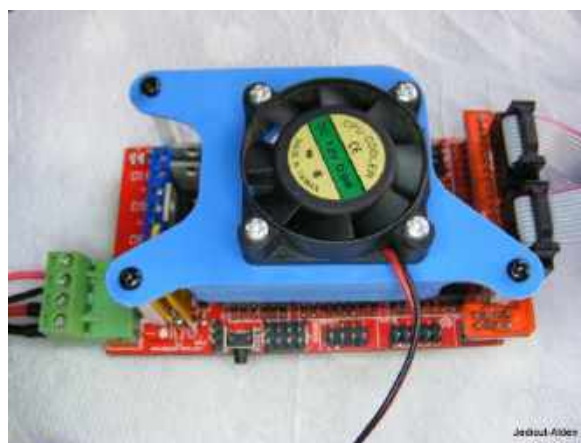
Tył płyty czołowej.



Na mojej karcie Ramps brakuje 2 szpilek dla wentylatora sterowników, widzimy 2 otwory obok diody.

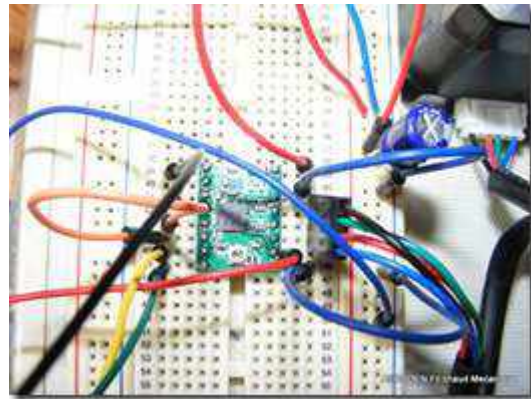
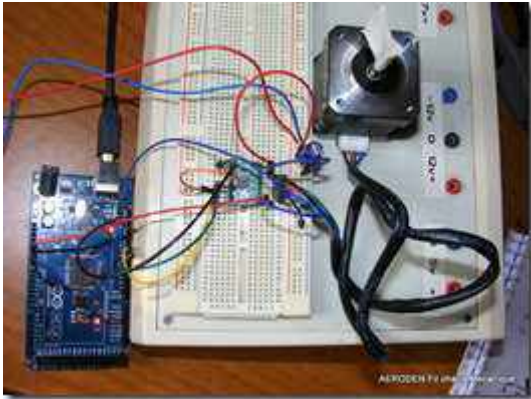


Włutowanie 2 szpilek i wentylator będzie zasilany z 12 V.



Prezentacja wspornika wentylatora. Jest wydrukowany w 3D, możesz to zrobić na gorąco z plastiku.

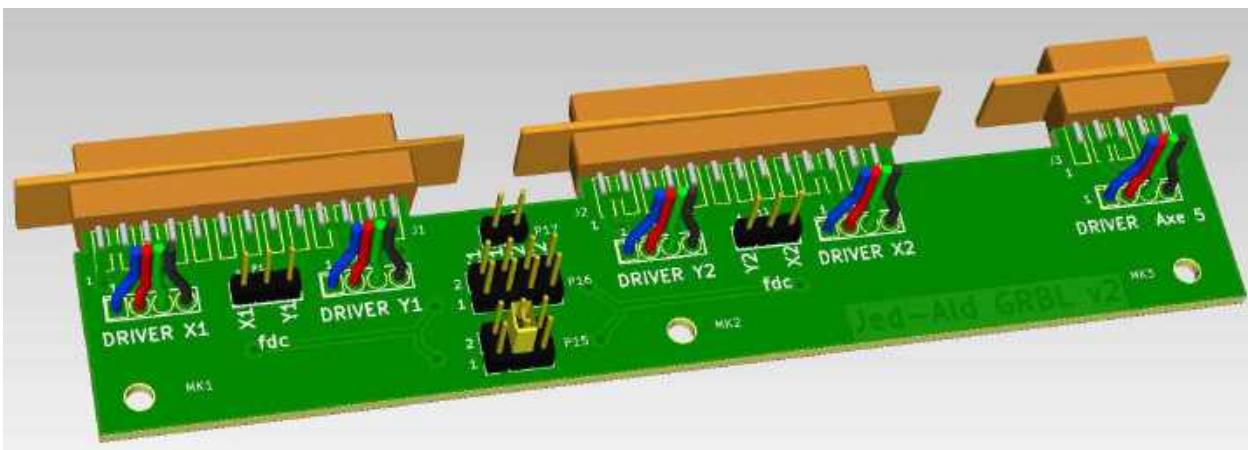
Testowanie płytek i regulacja sterownika silnika krokowego:



[Zobacz folder ARD_RAMP_IDE](#)

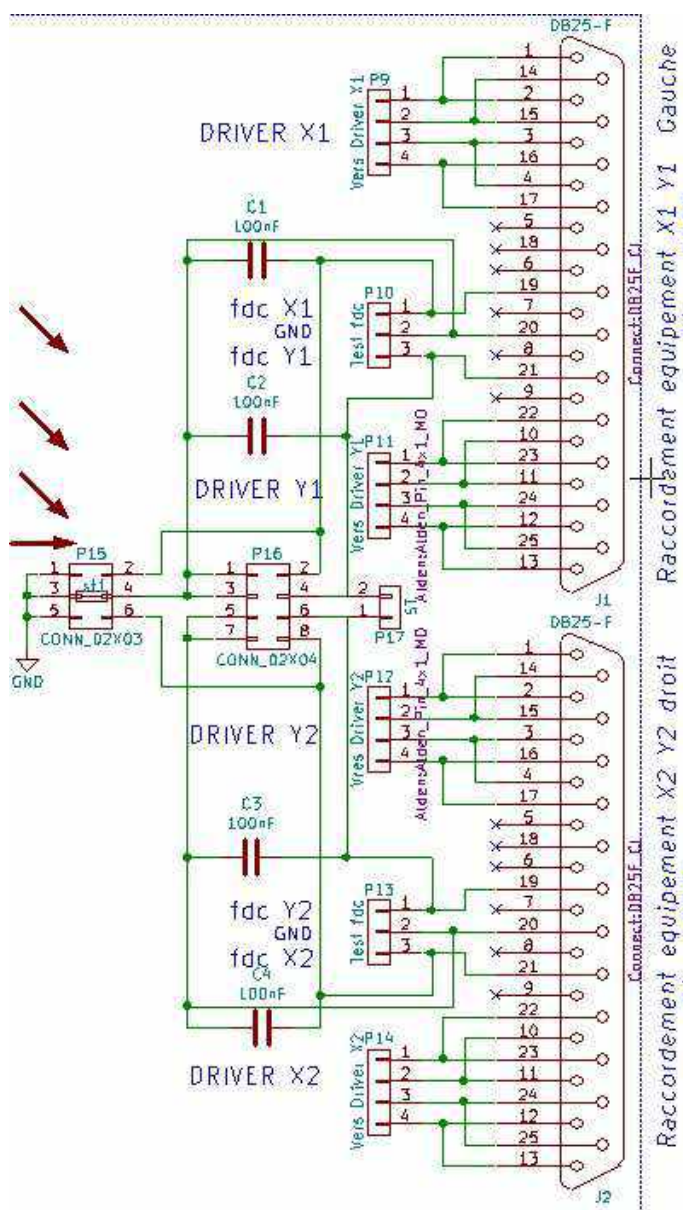
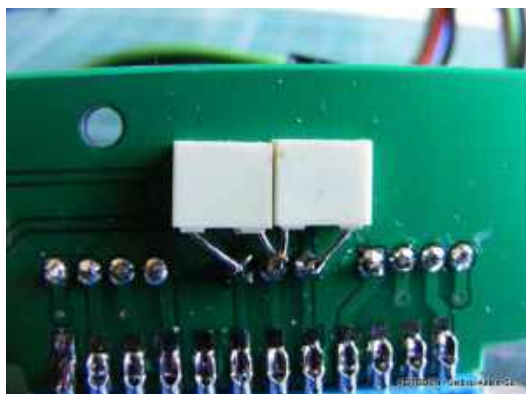
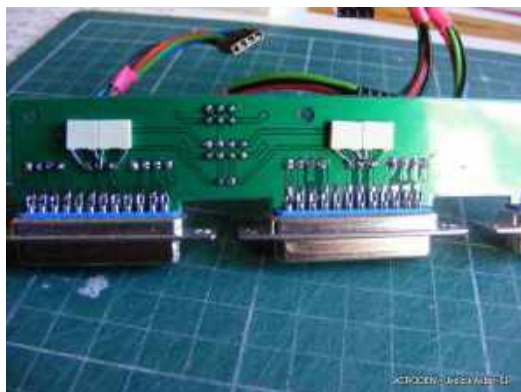
Modyfikacja płytki wyjściowej DB25 dla silników:

Po opracowaniu Jedicut-GCode i Grbl zaprojektowałem płytę DB 25 kompatybilną z Jedicut-Aldenem i Grblem. Są tylko szpilki do modyfikacji. Dzięki temu użytkownik Jedicut-Alden może migrować swoją elektronikę bez zmiany okablowania wyjściowego na Grbl.



W przypadku Jedicut-Alden Super Luxe, należy umieścić zworę na P15, jak na powyższym obrazku. Aby uniknąć jakiegokolwiek problemu z przypadkowymi zakłóceniami, (przewody wyłączników krańcowych znajdują się obok przewodów silników krokowych), zalecam dodanie kondensatorów 100nF do pinów **f1c** po stronie lutowania - w sposób pokazany niżej.

Ta karta V2 jest wspólna dla GRBL i Jedicut-Alden, kiedy ją projektowałem, przewidywałem, że pewnego dnia będzie konieczność przejścia w Jedicut-Alden na indywidualne **fdc** zamiast szeregowych, dlatego jest większa liczba łączników, które pozwalają zworkami wybierać różne połączenia samego mechanicznego **fdc**. Dla GRBL kondensatory odsprężające dla każdego **fdc** znajdują się na pośredniej karcie, w przypadku Jedicut-Alden SuperLuks konieczne jest dodanie ich w następujący sposób:



Te 4 kondensatory to 100nF 63V raster 5.08. Zastosowanie kondensatorów o rastrze 2,54 byłoby łatwiejsze. Ze względu na cynowane płytki, lutowania są bardzo proste do wykonania. Dla tych, którzy chcą użyć mechanicznego **fdc** na płytkach z sygnalizacją lub czujnikami optycznymi, konieczne jest posiadanie napięcia +5V, którego nie brałem pod uwagę przy tworzeniu karty. Jest to możliwe do wykonania i wyjaśnione w podręczniku "[Jed_FDC2.pdf](#)" oznaczonym jako **AD / CNC / JED / FDC**. Ta instrukcja jest częścią tego folderu.

Podłączanie wyłączników krańcowych na Ramps 1.4.

To połączenie fdc może być używane tylko z wersją Super Luks.

Podłączanie wyłączników krańcowych
na płycie Ramps 1.4

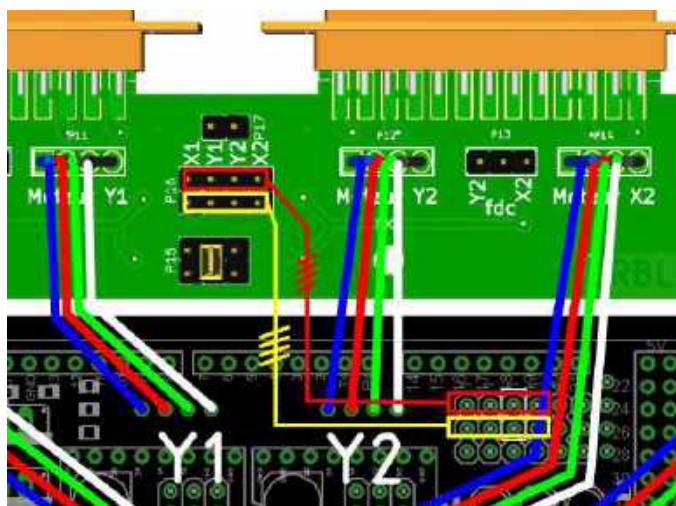


Złącze krańcówek
na ramps 1.4

Połączenie przewodowe
między złączem ramps a
złączem P16 na płycie DB25



Zworka
na złączu P15-3-4



Pomiędzy złączem P16 a złączami krańcówek karty Ramps potrzeba 8 przewodów w tym 4 dla GND.
4 żółte dla GND i 4 przewody czerwone dla fdc jak na powyższym obrazku.

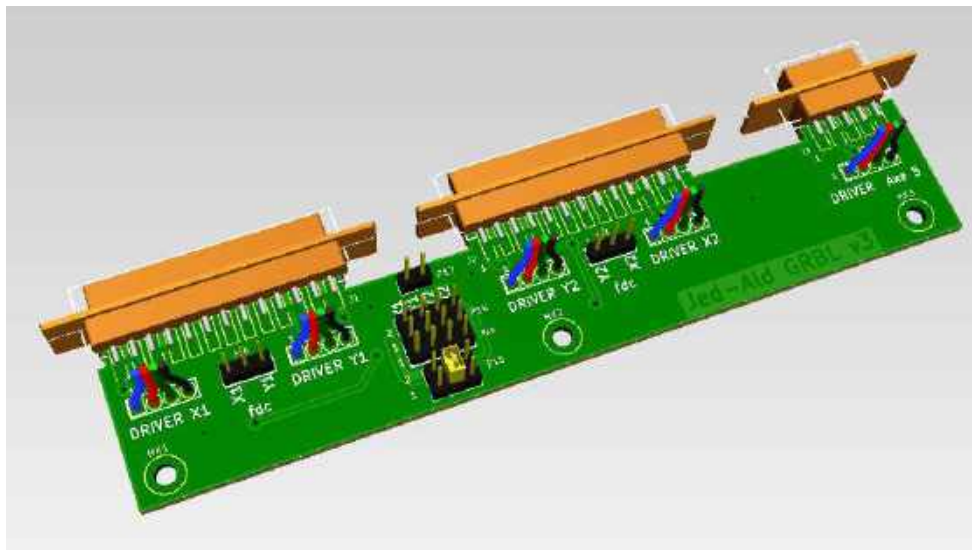
Fdc X1 połączenie od P16-1 do P8-1 na Ramps.

Fdc Y1 połączenie od P16-3 do P9-1 na Ramps.

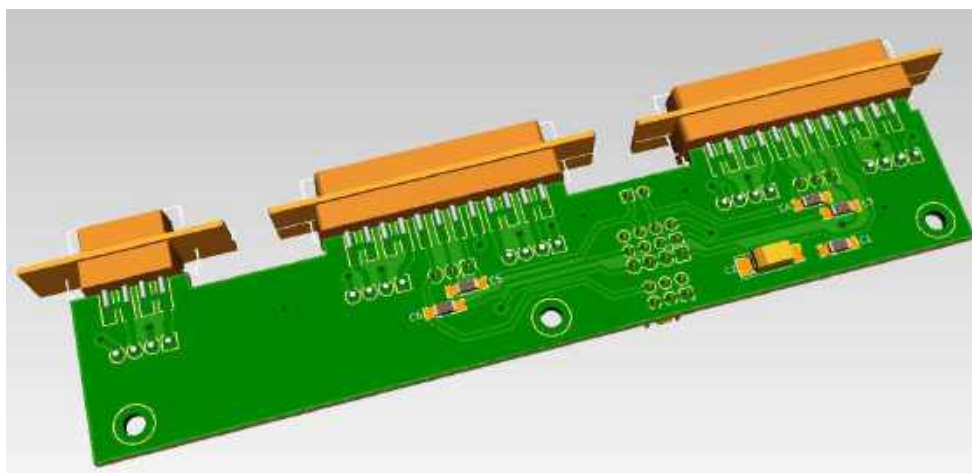
Fdc Y2 połączenie od P16-5 do P10-1 na Ramps.

Fdc X2 połączenie od P16-7 do P11-1 na Ramps.

Ponieważ konieczne jest dodanie kondensatorów dla wyłączników krańcowych, wersja płytki 2 została zmieniona na wersję 3. Dodano na druku pady dla kondensatorów smd i piny zasilania 5 wolt, dla optycznych wyłączników krańcowych lub 3 przewodowych z sygnalizacją. Oto nowa płytka:



Widok płytki z dołu

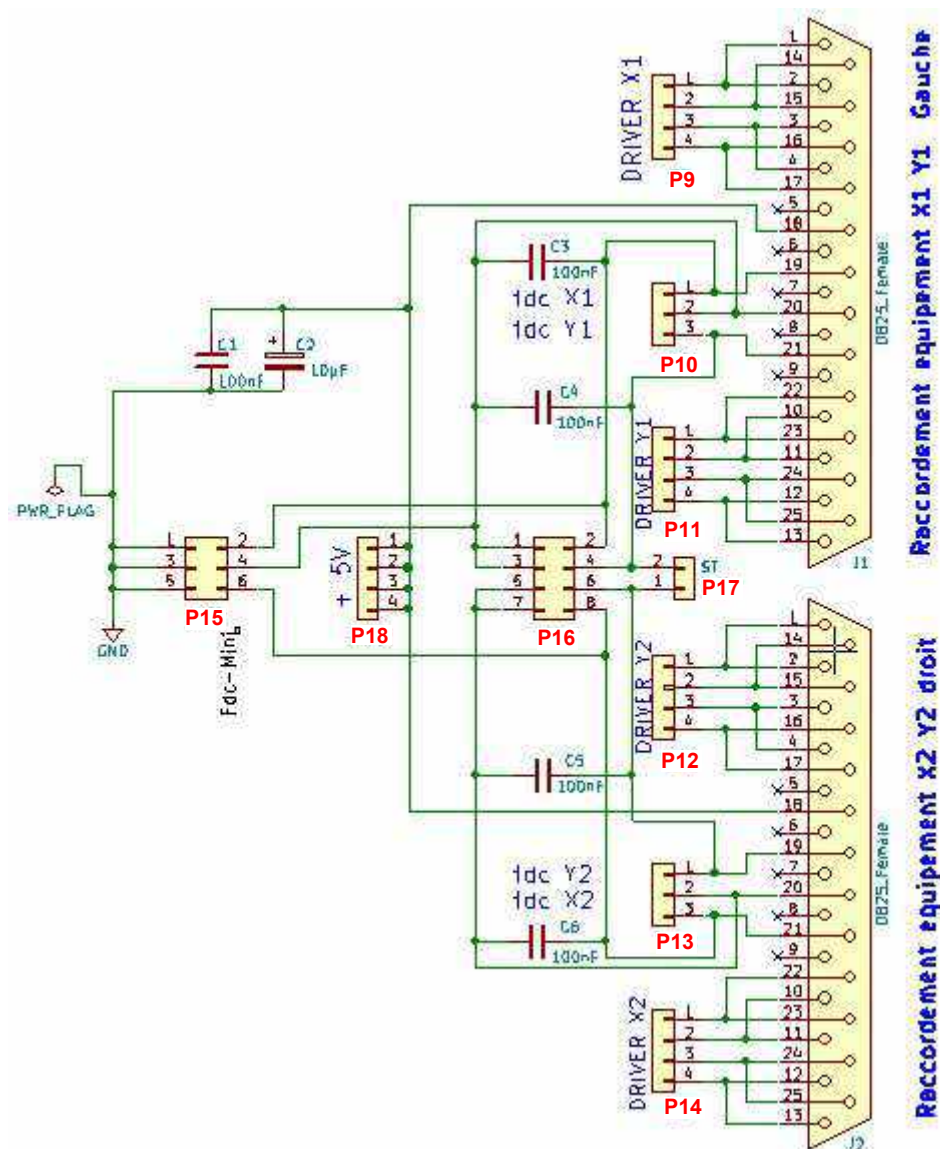


Jeśli nie używasz optycznych lub 3-przewodowych wyłączników krańcowych z sygnalizacją, kondensatory C1 i C2 nie są potrzebne.

Kondensatory 100nF to CMS 1208

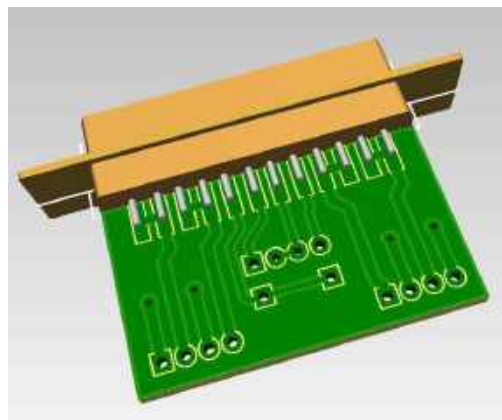
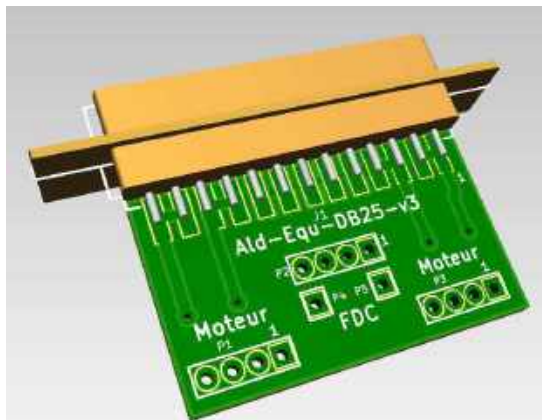
Kondensator 10 μ F to Tantalum CMS 6032

Schemat płytki v3:



Widzimy nowe złącze **P18**, które doprowadza +5v na DB 25 pin 18, aby je dostarczyć dla optycznych lub 3 przewodowych z sygnalizacją. Gdy do krańcówek używane jest łącze 5V, co najmniej jeden przewód musi być umieszczony między pinem złącza **P18** a płytą ramps. Kondensatory **C2** o wartości 10µF i **C1** o wartości 100nF służą do odsprzęgania, aby uniknąć zakłóceń pasożytniczych w kierunku Ramps 1.4. Nie zapomnij umieścić zworki na **P15** na 3 i 4. Pozwala to na podłączenie GND płyty DB25 do GND płytki Ramps przez **P16** piny 1,3,5,7 ..

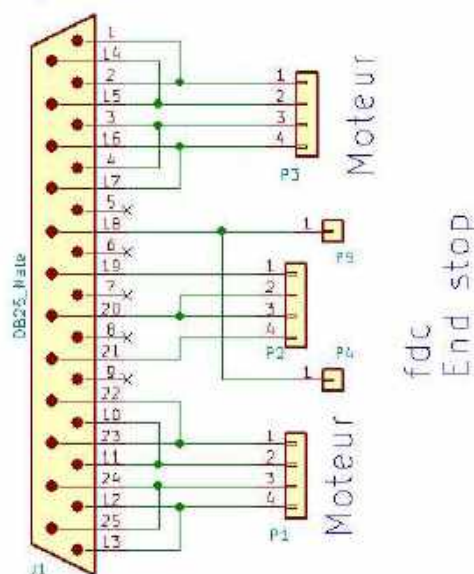
Płytki DB25 wózków zostały również zmienione w celu podłączenia optycznego lub 3-przewodowego **fdc**.



Schemat:

Znajdujemy więc nowe połączenia **P4 i P5** dla optycznego **fdc** lub 3-przewodowego dla + 5V pochodzącego ze szpilki **18** wtyku **DB25**.

Les bornes P4 et P5 reçoivent du +5V lorsque l'on utilise des fins de course optiques ou avec signalisation.



Teraz, gdy twoja nowa maszyna jest skończona, lub wersja Luks zmodyfikowana na SuperLuks, przejdziesz do testów; aby wszystko przebiegło zgodnie z planem, bez ryzyka zablokowania

maszyny. Polecam Ci działanie krok po kroku. Zalecam przeczytanie każdej fazy testu przed rozpoczęciem manipulacji.

Plan testów:

- Odczyt parametrów konfiguracyjnych szkicu arduino.
- Instalacja i konfiguracja Jedicut
- Konfiguracja szkicu.
- Przesyłanie szkicu
- Weryfikacja funkcji i kierunku wyłączników krańcowych.
- Weryfikacja kierunku ruchu 4 osi.
- Rekonfiguracja szkicu do odwrócenia fdc i możliwych osi.
- Nowe testy fdc i kierunku ruchu.
- Uruchomienie testu "bazowania".
- Uruchomienie wstępnego pozycjonowania.
- Zatrzymanie kontroli za pomocą fdc i zresetowanie.
- Uruchomienie grzania PWM.

Objaśnienie parametrów konfiguracyjnych:

Proszę przeczytać wszystkie objaśnienia parametrów przed modyfikacją. Następnie zobaczysz procedurę uruchomienia.

Po załadowaniu pliku do Arduino IDE masz 2 zakładki, w tym "conf.h", to właśnie nas interesuje.

```
// Konfiguracja maszyny
#ifndef _CONF_H
#define _CONF_H

//=====
// Definicja nazwy maszyny :

#define MACHINE_NAME " CNC FIL AERODEN"

//-----

//=====
// Definicja twojej wersji

#define VERSION "v4.5.0"

//-----

//=====
// Wybór języka: wybrany język nie może być odkomentowany (dwa // na początku linii)

#define LANG_FRENCH
// #define LANG_ENGLISH

//-----

//=====
// milimetr na krok
// Przykład 1: Ustawienie sterownika krokowego: pełny krok, silnik krokowy: 400 kroków na obrót, M6 Śruba prowadząca: 1 mm gwint
=> (mm na obrót * ustawienie sterownika) / krok na obrót = (1 * 1) / 400 = 0,0025
```


Objaśnienie:

Parametry w tej instrukcji definiowane są w następujący sposób, dla lepszego zrozumienia:

```
#define LIM_P_CUTTER 50
```

Na niebiesko jest instrukcja definicji

W kolorze czarnym i duże litery to parametr

Podświetlona na żółto jest zazwyczaj wartość liczbową, natomiast tekst jest ciągiem znaków w cudzysłowie „abc” to można zmieniać.

// 2 następujące po sobie ukośniki wskazują, że następujący tekst jest komentarzem aż do końca linii. Program nie bierze tego pod uwagę, to tylko pomoc.

Uważaj, aby nie wstawić "o" zamiast zera.

Analiza pliku konfiguracji szkicu: conf.h

Definicja nazwy maszyny

```
// Definicja nazwy maszyny
```

```
#define MACHINE_NAME " CNC FIL AERODEN"
```

```
#define MACHINE_NAME " CNC FIL AERODEN"
```

Możesz nadać nazwę swojej maszynie, masz prawo do 20 znaków i spacji między cudzysłowami.

Zmiana nazwy swojej wersji:

```
// Definicja twojej wersji
```

```
#define VERSION "v4.5.0"
```

```
#define VERSION "v4.7.0"
```

Aby zidentyfikować konfigurację, możesz zastąpić ostatnie "0" pierwszą literą swojego imienia. (Masz tylko jeden znak, w przeciwnym razie wartość na wyświetlaczu LCD będzie zła).

Wybór języka:

```
// Wybór języka: wybrany język nie może być komentowany
```

```
#define LANG_FRENCH
```

```
// #define LANG_ENGLISH
```

```
#define LANG_FRENCH
```

```
// #define LANG_ENGLISH
```

W tym przypadku jest wybrany jest język francuski, aby wybrać język angielski, należy usunąć 2 ukośniki na początku linii angielskiej i umieścić 2 ukośniki na początku linii francuskiej. Umieszczając 2 ukośniki na początku linii, zmieniasz ją w komentarz. Jeśli w pliku "lang.h" będą tłumaczenia również innych języków, to ich wybór w "conf.h" jest identyczny. Pamiętać należy że, wybrać można tylko jeden pozostałe muszą być komentarzami (dwa ukośniki na początku linii)

```
//=====
// milimetr na krok
// Przykład 1: Ustawienie sterownika krokowego: pełny krok, silnik krokowy: 400 kroków na obrót, M6 skok gwintu: 1 mm na obrót =>
// (mm na obrót * ustawienie sterownika) / krok na obrót = (1 * 1) / 400 = 0,0025
// Przykład 2: Ustawienie sterownika krokowego: 1/8 kroku, silnik krokowy: 200 kroków na obrót, pasek: 2 mm między zębami, koło
// pasowe: 20 zębów => (zęby koła pasowego * przestrzeń między zębami paska * ustawienie sterownika) / krok na obrót = (20 * 2 * 1/8)
// / 200 = 0,025
#define MM_PER_STEP 0.0025
```

#define MM_PER_STEP 0.0025

Jest to wartość w mm przesunięcia wózka dla jednego kroku silnika.

```
//=====
// Limit mocy noża: nóż 6V zasilany 12V PWM przy 75%
#define MAX_PERCENTAGE_CUTTER 75

// Ograniczenie mocy grzania drutu w %
#define MAX_PERCENTAGE_WIRE 80
```

#define MAX_PERCENTAGE_CUTTER 75

Limit mocy noża: nóż 6V zasilany 12V PWM przy 75%

#define MAX_PERCENTAGE_WIRE 80 Ograniczenie mocy do drutu do 80% przez Jedicut, potencjometr lub enkoder.

```
// Wybór między potencjometrem i enkoderem do ręcznego ogrzewania drutu
// wybrana opcja nie powinna być komentowana

// #define HEAT_CONSIGN_ROTARY_ENCODER
#define HEAT_CONSIGN_POTENTIOMETER
```

```
// #define HEAT_CONSIGN_ROTARY_ENCODER
#define HEAT_CONSIGN_POTENTIOMETER
```

W tym przypadku jest to potencjometr, który jest wybrany, aby wybrać enkoder, usunąć 2 ukośniki na początku linii enkodera i umieścić 2 ukośniki na początku linii potencjometru. Umieść dwa ukośne znaki na początku linii, a linia zmieni się w komentarz

```
// Brzęczyk włączony lub wyłączony dla alarmu dźwiękowego wyłącznika krańcowego,
// aby uniknąć alarmu dźwiękowego, wstaw "/" na początku linii
#define BUZZER_ON
```

#define BUZZER_ON

Włączanie i wyłączanie brzęczyka dla alarmu dźwiękowego, aby uniknąć alarmu dźwiękowego, należy umieścić "/" na początku linii.

Wybór, podtrzymania zasilania silników:

Parametr używany do określenia, czy silniki pozostają pod napięciem po zatrzymaniu, w przypadku urządzeń mechanicznych ze śrubami nie jest konieczne, z drugiej strony dla urządzeń pasowych lepiej jest pozostawić je pod napięciem. Silniki można wyłączyć za pomocą przełącznika „ON / OFF”.

"MOTEUR_ON_ASSERVI" ustawiony na „1” określa, że zasilanie obsługiwane jest rozkazami szkicu Arduino i Jedicut.
Dla „0” silniki są zawsze pod napięciem.

```
#define MOTEUR_ON_ASSERVI 0 // Przy "0" silniki są zawsze pod napięciem.
```

```
#define MOTEUR_ON_ASSERVI 0
```

Odwroćcie obrotów silników.

UWAGA: Na karcie "Konfiguracja maszyny" w Jedicut: wszystkie pola "Odwroćcie" są odznaczone. Odwrócenie kierunku, jeśli to konieczne, musi być wykonane przez maskę poniżej, aby uzyskać poprawne działanie sekwencji bazowania.

Odwroćcie kierunku ruchu w masce przyjmuje wartość „1” (dla najbardziej znaczących bitów, Y2,Y1,X2,X1)

Ob,Y2, Y1, X2, X1, 0,0,0,0; („Ob” oznacza że jest to wartość binarna)

*/

```
#define INV_DIR_MASK 0b00000000 // "1"odwrócenie 4 osi => 0b11110000
```

```
#define INV_DIR_MASK 0b00000000
```

Odwroćcie działania wyłączników krańcowych.

W zależności od rodzaju użytych krańcówek czasami trzeba je odwrócić.

Normalne: krańcówka nie jest naciśnięta, daje "0" na wejście ramps

Odwroćcie: krańcówka, nie jest naciśnięta, daje "1" na wejście ramps

*/

```
#define INV_FDC_X1 0 // "0" Normalny, "1" Odwrócony
```

```
#define INV_FDC_Y1 0 // "0" Normalny, "1" Odwrócony
```

```
#define INV_FDC_X2 0 // "0" Normalny, "1" Odwrócony
```

```
#define INV_FDC_Y2 0 // "0" Normalny, "1" Odwrócony
```

```
#define INV_FDC_X1 0 // "0" Normalny, "1" Odwrócony
```

```
#define INV_FDC_Y1 0 // "0" Normalny, "1" Odwrócony
```

```
#define INV_FDC_X2 0 // "0" Normalny, "1" Odwrócony
```

```
#define INV_FDC_Y2 0 // "0" Normalny, "1" Odwrócony
```

Parametry operacyjne wymagane do naprowadzania i wstępnego pozycjonowania:

Parametry umożliwiające obliczenie danych niezbędnych do zerowania maszyny

*/

```
#define VIT_RECH_FDC 4.50 // Szybkość wyszukiwania fdc w mm/s Format XX.XX
```

```
#define VIT_AJUST_FDC 1.00 // Prędkość regulacji fdc w mm/s Format XX.XX
```

#define VIT_RECH_FDC 4.50 : Ustawiasz prędkość, z jaką przesuwają się osie podczas wyszukiwania wyłączników krańcowych, jest to również prędkość wstępnego pozycjonowania. Domyślnie umieszczam 4.5mm / s, możesz umieścić wartość, która Ci odpowiada.

#define VIT_AJUST_FDC 1.00 : ustawiasz prędkość, z jaką osie dojadą do FDC. Niska prędkość zwiększa dokładność. Domyślnie umieszczam 1mm / s.

Parametry określające różne typy sekwencji bazowania

```
#define SEQ_HOMING 0 // "1" sekwencja bazowaniaa aktywna, "0" sekwencja nieaktywna
#define POS_SECU_Y 1 // "1" wykonuje ruch Y przed bazowaniem
#define MM_POS_SECU_Y 2 // podnosi Y o x mm przed naprowadzeniem
#define PREPOS 1 // "1" Wstępne pozycjonowanie po zezwoleniu na naprowadzanie, "0" bez wstępnego pozycjonowania
#define MM_PREPOS_Y 1 // Wartość ustawienia wstępnego Y w mm
#define MM_PREPOS_X 1 // Wartość ustawienia wstępnego X w mm
```

#define SEQ_HOMING 0 // "1" sekwencja bazowaniaa aktywna, "0" sekwencja nieaktywna
Domyślnie nie ma sekwencji bazowania, najpierw sprawdź FDC i kierunek osi

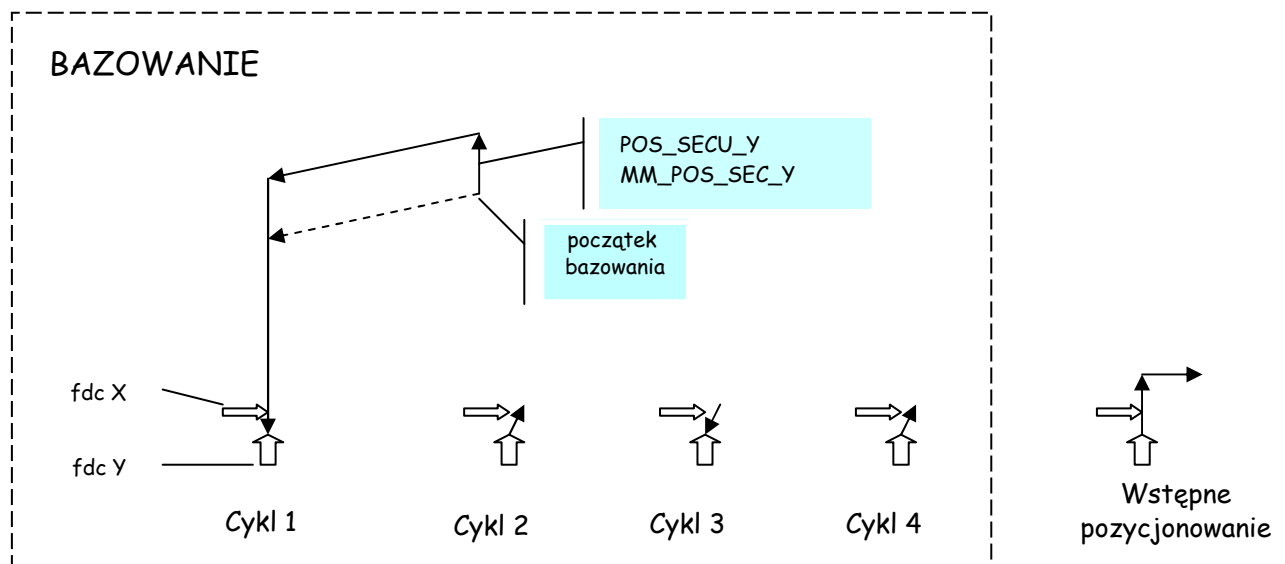
#define POS_SECU_Y 1 // "1" wykonuje ruch Y przed bazowaniem, "0" nie wykonuje przesunięcia w Y.
Domyślnie "1", jeśli maszyna zatrzymała się na fdc Y, lepiej jest podnieść drut, aby nie dotknął stołu.

#define MM_POS_SECU_Y 2 // wartość ruchu Y o 2mm przed domyślnym bazowaniem.

#define PREPOS 0 // "1" Wstępne pozycjonowanie po zezwoleniu na naprowadzanie, "0" bez wstępnego pozycjonowania, domyślnie "0" pozwala to ustawić stół na zero.

#define MM_PREPOS_Y 1 // Wartość ustawienia wstępnego Y w mm, domyślnie 1mm ale możesz wstawić wartość 10 200mm.

#define MM_PREPOS_X 1 // Wartość ustawienia wstępnego X w mm, domyślnie 1mm ale możesz wstawić wartość 10 200mm.



Oto schemat bazowania i wstępnego pozycjonowania. (Reprezentowane są tylko 2 osie)

Poniższy opis dotyczy krańcówek mechanicznych, których kontakt jest otwarty, gdy wyłącznik nie jest aktywny

Cykl bazowania 1: odbywa się to z prędkością VIT_RECH_FDC.

Wybierz bazowanie, aby przesunąć oś Y do góry o "x" mm. Planowałem ten ruch w przypadku, gdy osiągnąłeś fdc Y i drut dotyka stołu, robiąc ruch w górę o kilka mm, drut nie dotyka stołu i możemy wykonać bazowanie X, a następnie bazowanie Y do fdc. (Styki fdc są otwarte).

Cykl bazowania 2: odbywa się to z prędkością VIT_AJUST_FDC

Wszystkie 4 osie przesuwają się w kierunku dodatnim, aż zamkną się styki fdc.

Cykl bazowania 3: odbywa się to z prędkością VIT_AJUST_FDC

Wszystkie 4 osie przesuwają się w kierunku ujemnym do momentu otwarcia styków fdc.

Cykl bazowania 4: odbywa się to z prędkością VIT_AJUST_FDC

Wszystkie 4 osie przesuwają się w kierunku dodatnim, aż zamkną się styki fdc.

Koniec bazowania:

Jeśli zdecydujesz się na wstępne pozycjonowanie, koniec bazowania przechodzi przez wstępne pozycjonowanie, ty decydujesz odległości RUCHU X i Y, aby osiągnąć pożądaną pozycję, prędkość ruchu jest prędkością skonfigurowaną "VIT_RECH_FDC". Obie osie Y przesuwają się najpierw, a następnie osie X. Jeśli twoja pozycja "0" fdc jest niższa od twojego stołu, przesunięcie osi Y zapobiegnie zawieszaniu się drutu, gdy osie X się poruszają.

Szybkość komunikacji USB:

```
// Szybkość komunikacji szeregowej USB, wybierz jedną z następujących wartości:  
// 9600, 14400, 38400, 57600, 115200, 250000  
// W pliku comport.ini JediCut musi być dokładnie taka sama wartość  
#define BAUDRATE 115200
```

```
#define BAUDRATE 115200
```

Odwrócenie aktywacji sterownika:

```
// Jeśli sterownik Stepper Driver ma odwrócony poziom pinu aktywacji, odkomentuj to,  
// jeśli sterownik jest odblokowany przez "1" zamiast "0" usuń // z  
// następnej linii
```

```
//#define INVERT_STEPPER_DRIVER_ENABLE_PIN
```

```
#ifdef INVERT_STEPPER_DRIVER_ENABLE_PIN
```

```
    #define STEPPER_DRIVER_ENABLE_HIGH_LEVEL 1
```

```
    #define STEPPER_DRIVER_ENABLE_LOW_LEVEL 0
```

```
#else
```

```
    #define STEPPER_DRIVER_ENABLE_HIGH_LEVEL 0
```

```
    #define STEPPER_DRIVER_ENABLE_LOW_LEVEL 1
```

```
#endif
```

```
//#define INVERT_STEPPER_DRIVER_ENABLE_PIN
```

Dla sterowników na Ramps 1.4 pozostaw 2 ukośniki na początku linii.

Korekta grzania dla cięcia skośnego i po łuku

Ostrzeżenie: jeśli używasz tej opcji, nie używaj grzania dynamicznego w Jedicut. To się powieli.

Poprzez dopasowanie kroków możliwe jest określenie prędkości na trajektorii i sterowanie ogrzewaniem poprzez wprowadzenie korekty ogrzewania. Prędkość trajektorii jest największa, gdy drut przechodzi przez segment 45°. Jeśli prędkość X i Y wynosi 2 mm / s, prędkość drutu wynosi $2 \times 1,414 = 2,828$ mm/s. Z wykresu twoich materiałów dla danego łuku, zanotujesz wartość w % dla 2 mm/s -> 30%, a następnie wartość dla prędkości 2,828 -> 39,5%, teraz obliczasz, $39,5 / 30 = 1,3166$, współczynnik korygujący wyniesie zatem 1,32. Szkic sprawdza się dobrze w obliczeniach prędkości pośredniej drutu.

*/

define CHAUFFE_ASSERV 1 // „0” brak kontroli grzania, „1” z kontrolą

define CORRECT_CHAUFFE 1.32 // współczynnik grzania w funkcji prędkości

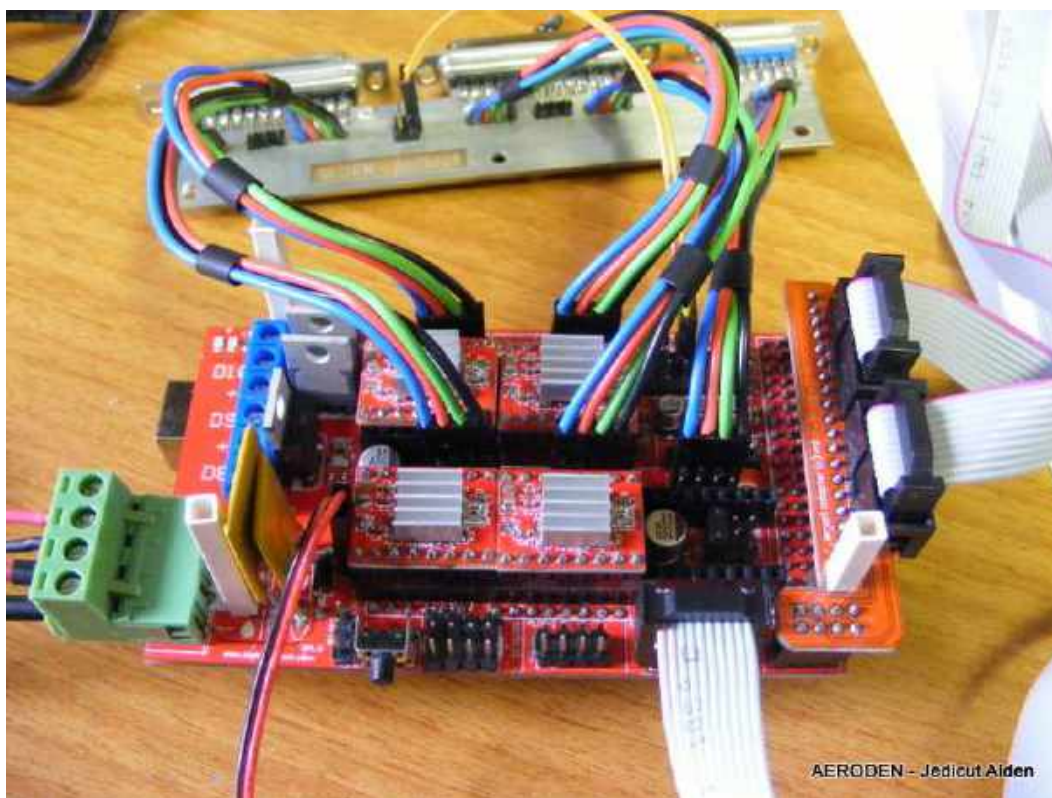
Przejrzeliśmy opcje konfiguracji, teraz dostosujemy to do twojej maszyny i twoich potrzeb.

Test karty Ramps 1.4

Zobacz folder [ARD_RAMPS_IDE](#)

Po przetestowaniu płyty Ramps 1.4 można przykleić radiatory do sterowników. Uważaj, aby nie dotykały potencjometrów. Ustawieś REF na prąd znamionowy twoich silników. Uruchamiasz wszystko od wentylatora.

Podłączasz zespół, aby przeprowadzić testy z twoim urządzeniem (**mechanicznie musi on być kompletnie zakończony, fdc działa**), ustawienie w skrzynce zostanie wykonane po testach.





Przygotowanie Jedicut:

Zainstaluj Jedicut

Obecna wersja Jedicut to **2.4.2**

Sprawdź, czy:

Plik "**USBSerial_2.dll**" znajduje się w podfolderze "**DLL**" katalogu "**Jedicut**".

Sprawdź także, czy plik "**comport.ini**" znajduje się w katalogu "**Jedicut**".

Jeśli ich nie ma, wykonaj poniższą procedurę.

(Nowe pliki „**DLL**” i „**comport.ini**” są częścią pobranego pliku Jedicut, lub pobranych instrukcji [folder "**Plugging**"]).

Ostrzeżenie: W Windows 10 nie można zmieniać plików w folderze „**Program Files (x86)**”. Aby uzyskać dostęp do pliku w folderze „**Program Files (x86)**”, kliknij prawym przyciskiem myszy folder "**Jedicut**, **Właściwości, Security**", daj sobie **wszystkie uprawnienia**. Będziesz mógł wykonywać następujące operacje. Oto bardzo ważne katalogi i podkatalogi.


Dodaj w folderze "**Jedicut**" w "**Program Files (x86)**" następujące pobrane pliki:

Plik dll: "**USBSerial_2.dll**" w katalogu "**DLL**" "**Jedicut**".

Plik "**comport.ini**" w katalogu głównym "**Jedicut**".

Konfiguracja Jedicut:

Otwórz program Jedicut a następnie

Otwórz: "Narzędzia", "Opcje" lub kliknij  i otwórz kartę "Komunikacja".

Wypełnij tak:

Aktywny stół CNC: **twój**.

Tryb komunikacji: **USBSerial_2.dll**

Konfiguracja sygnałów: wartości są narzucone przez program Arduino.

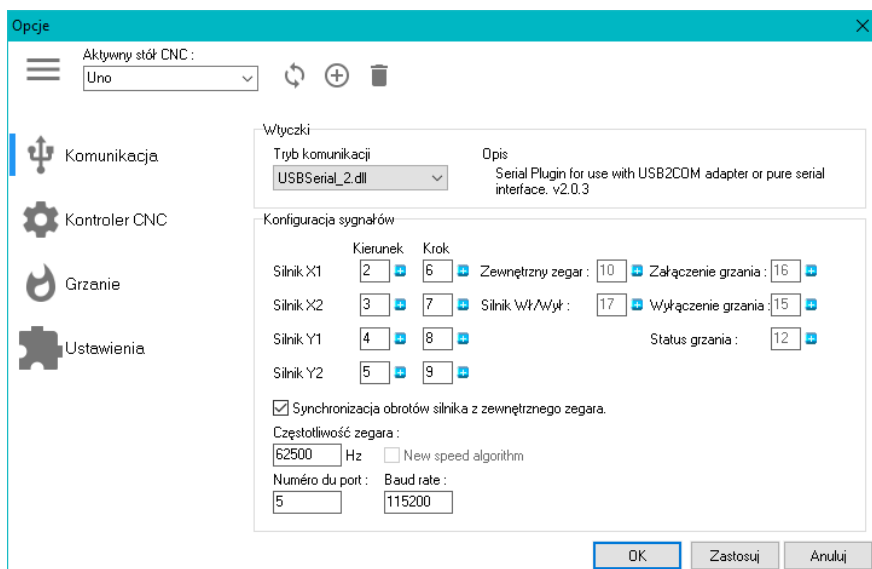
Pola wyboru:


"Synchronizacja z": **zaznaczone**
(niezaznaczone, grzejnik jest zablokowany)

"Nowy algorytm prędkości": **nie powinien być zaznaczony** dla maszyn połączonych przez USB

Częstotliwość zegara: **62500 Hz**.

Wybierz "Zastosuj" i "OK"

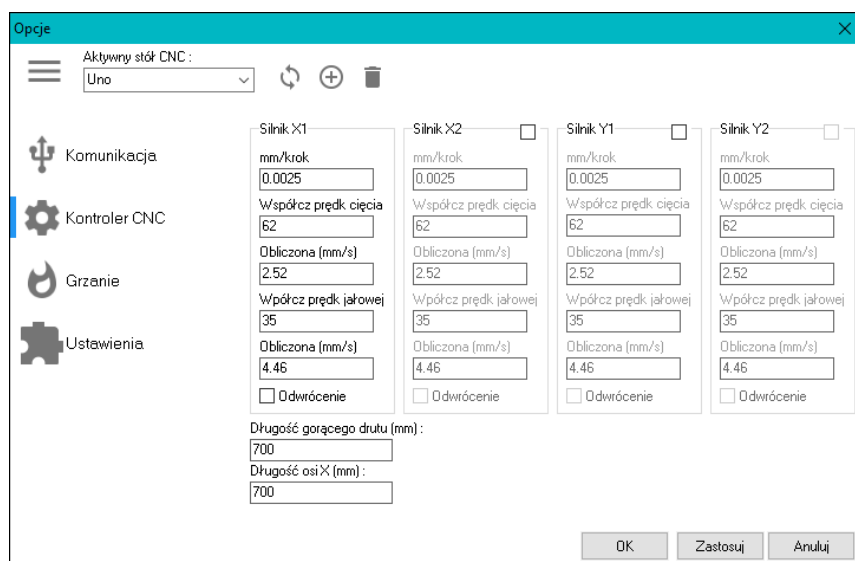


Otwórz: "Narzędzia", "Opcje" lub kliknij  i otwórz kartę "Kontroler CNC"

Charakterystyka mechaniczna Twojej maszyny. Pręt gwintowany 6 mm x 1 mm; silnik 200 kroków na obrót, zestaw elektroniczny na $\frac{1}{2}$ kroku; musimy zatem zapewnić 400 kroków na obrót, co daje $1 \text{ mm}/400 = 0,0025 \text{ mm/krok}$.

Wstaw współczynniki prędkości cięcia i ruchu jałowego:

W poprzednim widoku ustawiamy częstotliwość taktowania 62500 Hz, z wartością 62, która zostanie przesłana do Arduino (podzielimy częstotliwość $62500/62 = 1008 \text{ Hz}$), które wygeneruje kroki z tą częstotliwością. Mamy $0,0025 \text{ mm} \times 1008 = 2,52 \text{ mm/s}$. Obliczenie Jedicut jest dobre. Dla dużej prędkości współczynnik 35 daje 4,46 mm/s




W przypadku Jedicut-Alden Super Luxe wszystkie pola "Odwróć" powinny być **odznaczane**. Odwrócenie kierunków obrotów musi być wykonane bezwzględnie w "conf.h" szkicu Arduino, inaczej bazowanie nie zadziała.

Uwaga: te wartości współczynnika prędkości są ograniczone w programie Arduino do 32766 dla maks. Operacja jest gwarantowana do 63 dla mini. Jeśli chcesz obniżyć, jest to możliwe, ale musisz mieć Hub USB między komputerem a Arduino, Hub poprawia prędkość transmisji (nie znalazłem dlaczego). Konieczne jest jednak ograniczenie do niskiej wartości 30. (Hub NGS 4 portowy bez zasilania).

Nie używaj pełnego kroku dla sterowników, operacja nie jest płynna.



Wybierz "Zastosuj" i "OK"

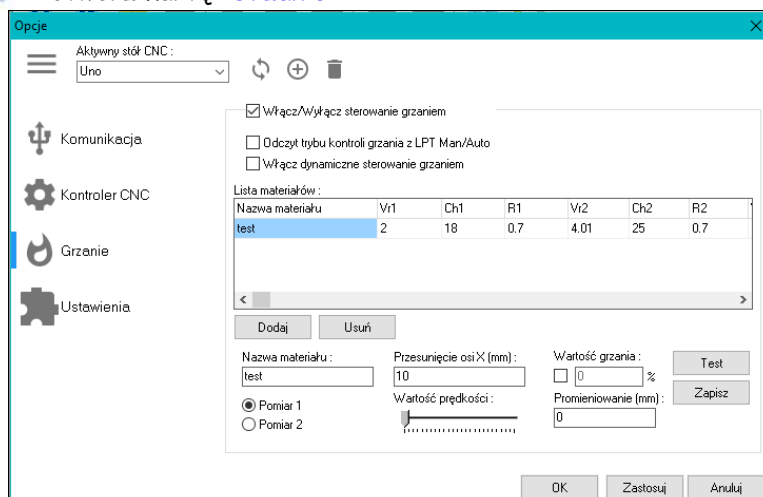
Otwórz: "Narzędzia", "Opcje" lub kliknij  i otwórz kartę "Grzanie"

Zaznacz pole wyboru **Włącz/wyłącz sterowanie grzaniem**.

Pozostałe pola nie muszą być zaznaczone przy połączeniu USB.

Aby zapoznać się z materiałami, zobacz instrukcję obsługi Jedicut, oraz plikami podkatalogu "Chauffe", który jest częścią pliku do pobrania "jedicut_Alden_SuperLuxe3.zip".

Jednak materiały te nie są przydatne w opisanych tutaj testach.



Zamknij Jedicut.

Programowanie wersji SuperLuks

Adaptacja programu "LMFAO_V4_7_0" do twojej maszyny.

Najpierw skopiuj folder szkicu "LMFAO_V4_7_0" do podkatalogu "Applications" w katalogu "Arduino".

Uruchom "Arduino IDE" Będziesz miał domyślny program, który się otworzy lub program, który był wcześniej używany.

Kliknij "Plik, Preferencje, Sprawdź wiersz" Pokaż numery linii ", jeśli nie zostało to wcześniej zrobione. Kliknij "Otwórz plik", wyszukaj folder "Applications" twojego programu, otwórz folder i kliknij dwa razy na "LMFAO_V4_7_0.ino" Otworzy się nowe okno i twój program będzie dostępny. Masz 3 zakładki, są one częścią twojego szkicu, są to pliki, umożliwiają tworzenie krótszych plików, łatwiejszych do edycji, często są to również pliki konfiguracyjne.

W naszym przypadku mamy plik "conf.h", plik "lang.h", plik "LMFAO_V4_7_0.ino" Kliknij na zakładkę "conf.h", gdzie należy skonfigurować parametry specyficzne dla twojej maszyny. Powracasz do karty "LMFAO_V4_7_0"

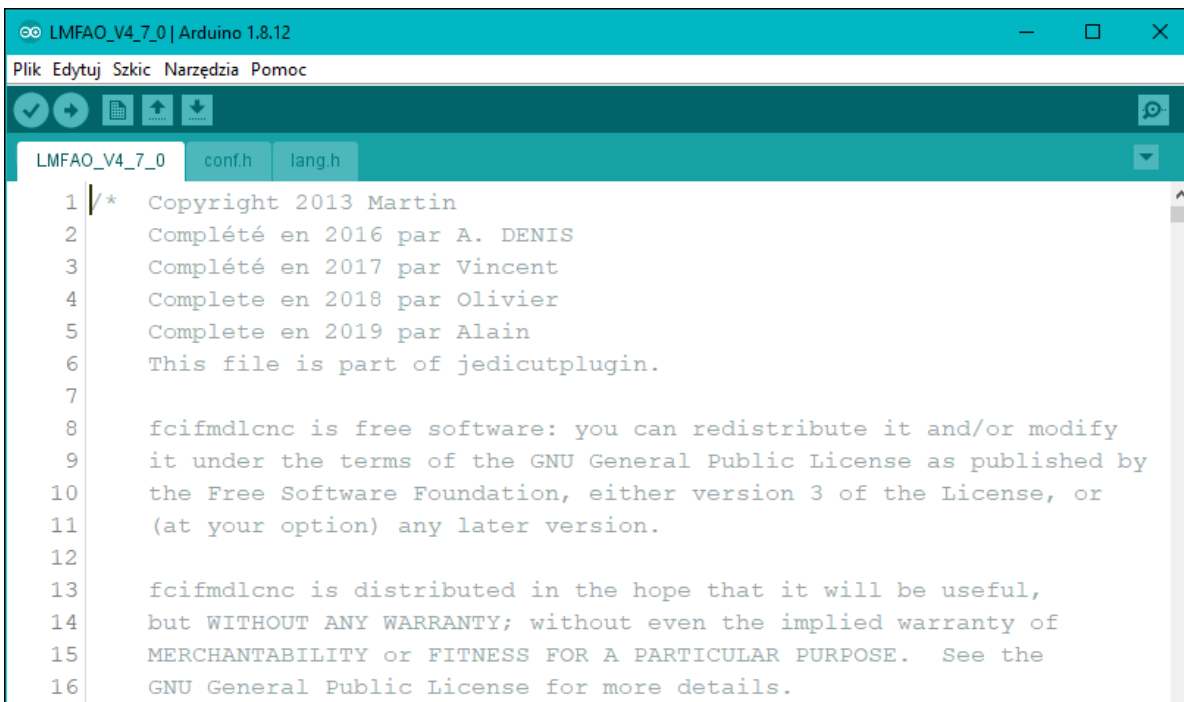
Radzę przed jakąkolwiek modyfikacją, zapisać szkic pod inną nazwą, aby zachować oryginał jak opisano w instrukcji Ard_Ramps_IDE_3.pdf.

Otwórz "Plik => Zapisz jako" wybierz katalog "Aplikacja" w "Arduino", a następnie zmień nazwę, w "LMFAO_V4_7_0", zastąp ostatnie "0" pierwszą literą swojego imienia np. "LMFAO_V4_7_A", a następnie kliknij "Zapisz".

Twój szkic ma nową nazwę, i jest widoczny w zakładce.

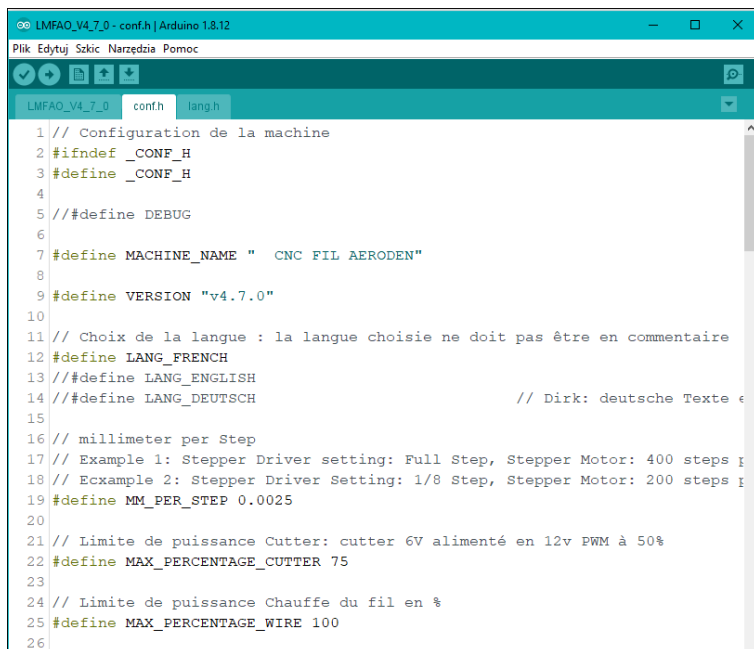
Podłącz kabel USB

Sprawdź w IDE wybór płytki Arduino i numer portu Comx.



```
1 /* Copyright 2013 Martin
2   Compl  t   en 2016 par A. DENIS
3   Compl  t   en 2017 par Vincent
4   Complete en 2018 par Olivier
5   Complete en 2019 par Alain
6   This file is part of jedicutplugin.
7
8   fcifmdlnc is free software: you can redistribute it and/or modify
9   it under the terms of the GNU General Public License as published by
10  the Free Software Foundation, either version 3 of the License, or
11  (at your option) any later version.
12
13  fcifmdlnc is distributed in the hope that it will be useful,
14  but WITHOUT ANY WARRANTY; without even the implied warranty of
15  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
16  GNU General Public License for more details.
```

Kliknij kartę "Conf.h"



```
1 // Configuration de la machine
2 #ifndef _CONF_H
3 #define _CONF_H
4
5 // #define DEBUG
6
7 #define MACHINE_NAME " CNC FIL AERODEN"
8
9 #define VERSION "v4.7.0"
10
11 // Choix de la langue : la langue choisie ne doit pas être en commentaire
12 #define LANG_FRENCH
13 // #define LANG_ENGLISH
14 // #define LANG_DEUTSCH // Dirk: deutsche Texte
15
16 // millimeter per Step
17 // Example 1: Stepper Driver setting: Full Step, Stepper Motor: 400 steps
18 // Example 2: Stepper Driver Setting: 1/8 Step, Stepper Motor: 200 steps
19 #define MM_PER_STEP 0.0025
20
21 // Limite de puissance Cutter: cutter 6V alimenté en 12v PWM à 50%
22 #define MAX_PERCENTAGE_CUTTER 75
23
24 // Limite de puissance Chauffage du fil en %
25 #define MAX_PERCENTAGE_WIRE 100
26
```

Adaptacja konfiguracji dla twojej maszyny: otwórz "conf.h"

W linii **9** : możesz zmienić nazwę swojej maszyny. Pamiętaj że masz tylko 20 znaków między cudzysłowami, (w tym spacje i inne).

`#define MACHINE_NAME " CNC FIL AERODEN"`

W linii **16**: możesz zmienić wersję, zastępujesz ostatnie "0" w numerze wersji pierwszą literą swojego imienia.

`#define VERSION "v4.7.0"` na `#define VERSION "v4.7.A"` - (mam na imię Alain)

W liniach **23 i 24**, Wybór języka wyświetlacza LCD

`#define LANG_FRENCH`

`// #define LANG_ENGLISH`

Wybrałem francuski, (odkomentowany brak//)

W linii **33**: określasz wartość kroku maszyny w mm

`#define MM_PER_STEP 0.0025`

Tutaj 0,0025 => pręt gwintowany 1mm, silnik 200 kroków, sterownik w $\frac{1}{2}$ kroku.

To ta sama wartość, którą umieściłeś w konfiguracji Jedicut

W linii **39**: ograniczamy moc noża elektrycznego do 75%.

`#define MAX_PERCENTAGE_CUTTER 75`

W linii **42**: ograniczamy moc gorącego drutu do np.80 (80% mój drut zaczyna się rumienić).

`#define MAX_PERCENTAGE_WIRE 80`

W linii **50**: Wybrałem potencjometr do ręcznego ustawiania nagrzewnicy.

`// #define HEAT_CONSIGN_ROTARY_ENCODER`

`#define HEAT_CONSIGN_POTENTIOMETER`

W linii **57**: ustawiamy alarm zadziałania wyłączników krańcowych.

```
#define BUZZER_ON
```

W linii **70**: opcja ciągłego utrzymywania silników pod napięciem.

```
#define MOTEUR_ON_ASSERVI 1 // "1" dla maszyn śrubowych, "0" silniki są zawsze zasilane, jest to przydatne dla urządzeń pasowych, aby uniknąć przesunięć w czasie postoju.
```

W linii **81**: zostaw to w ten sposób.

```
#define INV_DIR_MASK 0b00000000 // "1" odwrócenie 4 osi -> 0b11110000
```

W linii **26**: ustawiamy prędkość transmisji danych 115200.

```
#define BAUDRATE 115200
```

W liniach **90, 91, 92, 93**: dla używających **fdc** na płytce lub optycznych, modyfikacja wstawiając "1" uzyskujemy odwrócenie kierunku działania krańcówek, jak podano w instrukcji "[Jed_FDC2.pdf](#)".

```
#define INV_FDC_X1 0 // "0" Normalne, "1" Odwrócone,
```

```
#define INV_FDC_Y1 0 // "0" Normalne, "1" Odwrócone,
```

```
#define INV_FDC_X2 0 // "0" Normalne, "1" Odwrócone,
```

```
#define INV_FDC_Y2 0 // "0" Normalne, "1" Odwrócone,
```

Linie **107 i 108**: pozostaw na razie na "0".

```
#define SEQ_HOMING 0 // "1" Sekwencja bazowania jest aktywna, "0" Brak sekwencji bazowania
```

```
#define POS_SECU_Y 0 // "1" wykonuje wycofanie Y przed bazowaniem
```

Linia **119**: Szybkość transmisji USB: 115200

```
#define BAUDRATE 115200
```

Linie **128-136**: Jeśli masz sterowniki Ramps, nic nie dotykasz.

```
// jeśli sterownik jest odblokowany przez "1" zamiast "0", usuń "/" z następnej linii.
```

```
//#define INVERT_STEPPER_DRIVER_ENABLE_PIN
```

```
#ifndef INVERT_STEPPER_DRIVER_ENABLE_PIN
```

```
    #define STEPPER_DRIVER_ENABLE_HIGH_LEVEL 1
```

```
    #define STEPPER_DRIVER_ENABLE_LOW_LEVEL 0
```

```
#else
```

```
    #define STEPPER_DRIVER_ENABLE_HIGH_LEVEL 0
```

```
    #define STEPPER_DRIVER_ENABLE_LOW_LEVEL 1
```

```
#endif
```

Linia **154**: pozostaw na razie "0":

```
#define CHAUFFE_ASSERV 0 // "0" bez sterowania grzaniem, "1" sterowanie grzaniem
```

Wszystko w "[config.h](#)" jest ustawione do pierwszych testów.

Wróć do karty "[LMFAO_V4_7_0](#)".

Teraz będziemy mogli przesłać szkic do Arduino Mega 2560.

Zachowaj ostrożność, korzystając z nowego szkicu, zachęcam do wykonania kolejnych czynności związanych z oddaniem do eksploatacji, może być konieczne przeczytanie tego kroku przed rozpoczęciem pracy.

Poprzez ręczną manipulację **przesuń 4 wózki maszyny około 60 mm od wyłączników krańcowych**. Ma to na celu sprawdzenie kierunku przesuwu.

Włącz zasilanie swojej elektroniki.

Podłącz kabel USB (jeśli podłączysz kabel USB przed włączeniem zasilania elektroniki, port USB przejmie to zasilanie, a nie zawsze jest wystarczająco wydajny dla prądu wyświetlacza LCD).

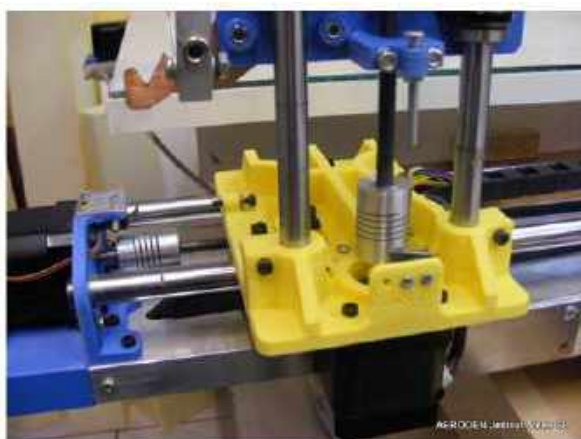
Podczas wyłączenia urządzenia, należy zachować zwyczaj odłączania portu USB przed wyłączeniem zasilania elektroniki.

Sprawdź w IDE wybór płytki Arduino i nr. portu comx

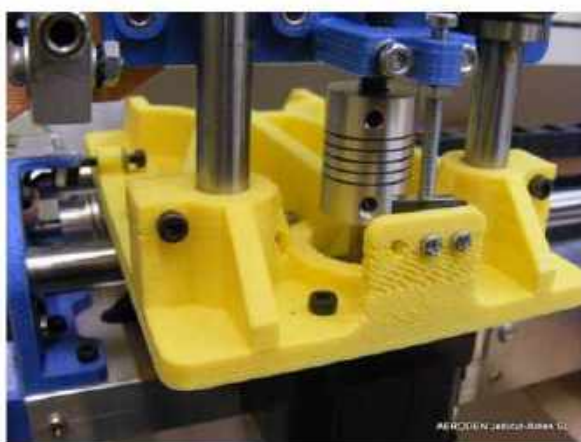
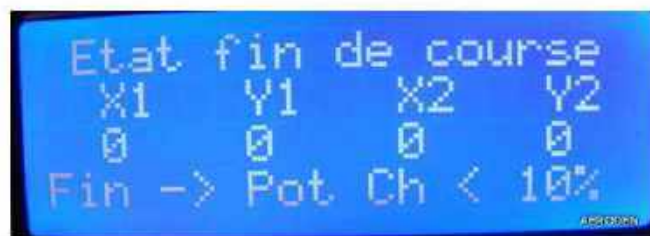
Weryfikacja krańcówek

Na pulpicie przekręcasz **potencjometr grzania całkowicie w kierunku zgodnym z ruchem wskazówek zegara na maks**, co pozwoli Ci przejść przez sekwencję wizualizacji 4 wyłączników krańcowych podczas inicjalizacji szkicu.

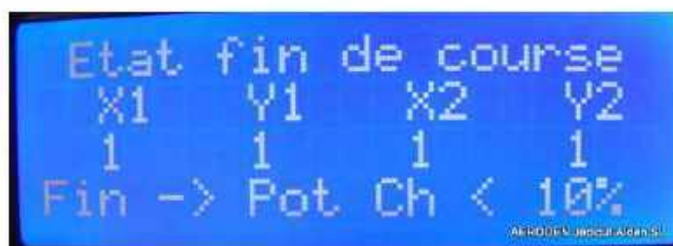
Przypomnienie: Poprzednio przesunąłeś wózki o 60 mm od **fdc**. Do testu musisz przesunąć wózki bliżej **fdc**.



krańcówka wolna



krańcówka wciśnięta



Załaduj szkic.

Podczas inicjalizacji wyświetlacz pokazuje:

Nazwę ustawień konfiguracji.

Wersja szkicu.

Prędkość transmisji.

Brzęczyk jest włączony



Następnie wyświetli się następujący ekran:

Jeśli masz "1" zamiast "0", będziesz musiał odwrócić krańcówki w szkicu "conf.h".

Manipulując **fdc** ręcznie naciskając lub przecinając wiązkę teksturą dla optyki, sprawdzasz, czy stan **FDC** osi zmienia się z 0 na 1 lub 1 na 0 jeśli są odwrócone.



Odnótowujesz **fdc**, które powinny być odwrócone. Jeśli nadal nie masz poprawnej odpowiedzi dla każdego z **fdc**, znajdź przyczynę, sprawdź okablowanie.

Powracasz do pliku "conf.h" i odwracasz zapisany FDC

```
#define INV_FDC_X1 0 // "0" Normalne      "1" Odwrócone
#define INV_FDC_Y1 0 // "0" Normalne      "1" Odwrócone
#define INV_FDC_X2 0 // "0" Normalne      "1" Odwrócone
#define INV_FDC_Y2 0 // "0" Normalne      "1" Odwrócone
```

Nie dotykając swojej elektroniki, ponownie wgrywasz zmodyfikowany szkic.

Teraz masz "0" dla wszystkich fdc i "1", gdy ręcznie je aktywujesz. Jest to niezbędne dla prawidłowego funkcjonowania tego, co nastąpi później. Szukaj przyczyn, jeśli nie masz właściwego wyniku..

Jeśli jest dobrze, przekręcasz potencjometr ogrzewania na zero. Inicjalizacja kończy się.

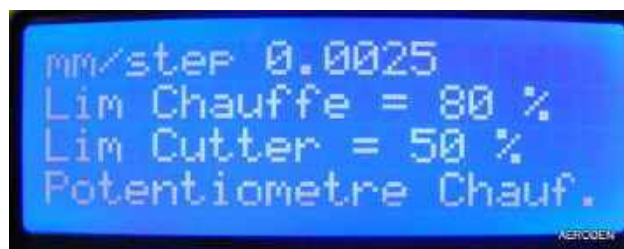
Następny ekran pokazuje:

Przesunięcie osi na każdy krok w mm/step

Limit ogrzewania drutu przy 80%

Granica ogrzewania noża przy 50%

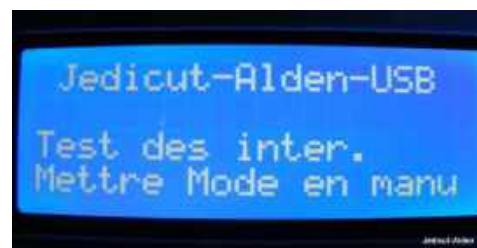
Zmiana ręcznego grzania odbywa się za pomocą potencjometru.



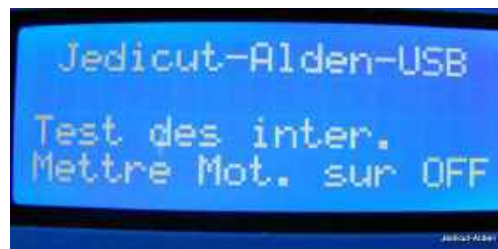
Podczas inicjalizacji sprawdzana jest pozycja przełączników na płycie czołowej sterowania ręcznego.

W tym przypadku przełącznik trybu PC / MANU pozostał na PC.

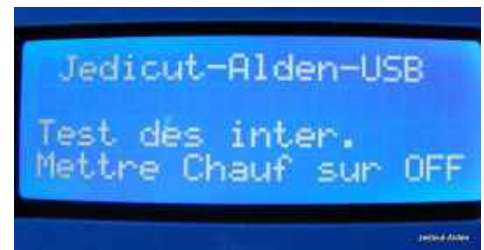
Ustaw przełącznik na MANU, test będzie kontynuowany.



Jeśli przełącznik silników **MOT** był już **WYŁ(OFF)**, ten ekran nie pojawi się.



Jeśli przełącznik grzania **Chauf** był już **WYŁ (OFF)**, ten ekran też nie pojawi się.



Rozruch jest zakończony, linie 3 i 4 są podzielone na 4 kolumny



Tryb	E dla Krańcówek	Silniki	Grzanie drutu	Grzanie noża
Ręczn./PC	I lub K	Wył/Wł/PC	PC/Wył/ręczn	Wył/Wł
	I = krańc. wolna		Ręczne ustawienie grzania	Ustawienie grzania noża

Po lewej stronie wskaźnika **WYŁ (OFF)** silnika znajduje się **I**. Jest to wskaźnik działania krańcówek, jeśli **I** zmieni się na **K** to znaczy że jedna z krańcówek zadziałała. Jeśli włączyłeś brzęczyk w konfiguracji, usłyszysz dźwięk ostrzegawczy. Będzie to opisane nieco bardziej szczegółowo.

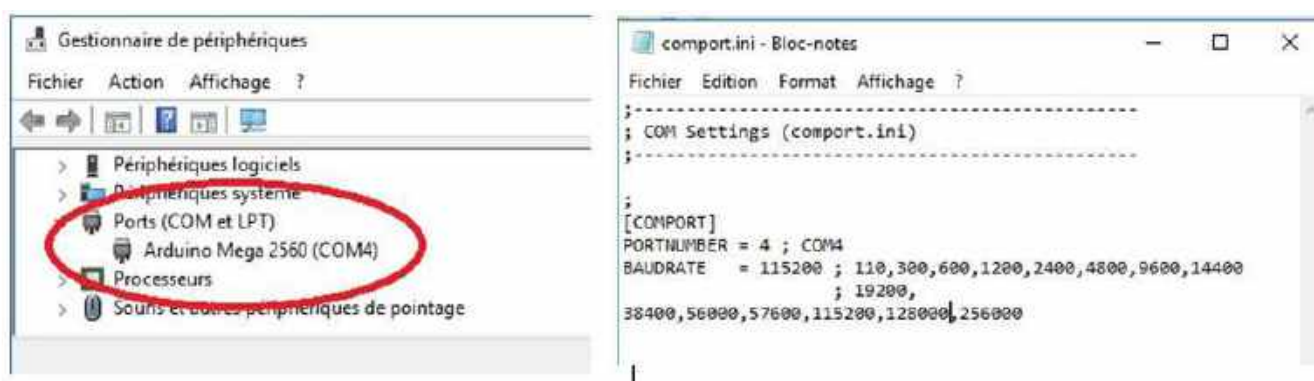
[Zamknij Arduino IDE](#)

[Otwórz Jedicut](#)

Weryfikacja kierunku ruchów:

Przypomnienie: skonfigurowałeś Jedicut na stronie **18**. Musisz jednak sprawdzić i skonfigurować plik **Comport.ini**, który znajduje się w głównym katalogu plików programu Jedicut.

Przejdź do [menedżera urządzeń](#) i znajdź port używany przez Arduino, otwórz plik **comport.ini** w notatniku i wpisz numer portu **COM** zgodnie z numerem z [menedżera urządzeń](#), jak pokazano poniżej.



Zapisz plik "**comport.ini**" w tym samym miejscu. Zamknij Jedicut, otwórz ponownie, aby Jedicut ponownie odczytał zmiany w pliku "**comport.ini**".

Weryfikacja kierunków ruchu wózków.

Ustaw przełączniki Trybu na PC, Silniki Wł, Grzanie Wył, Nóż Wył.

Jeśli twoje urządzenie jest całkowicie gotowe możesz teraz wypróbować grzanie drutu, ustawisz przełącznik "grzanie na Wł", zobaczysz prąd na amperomierzu.

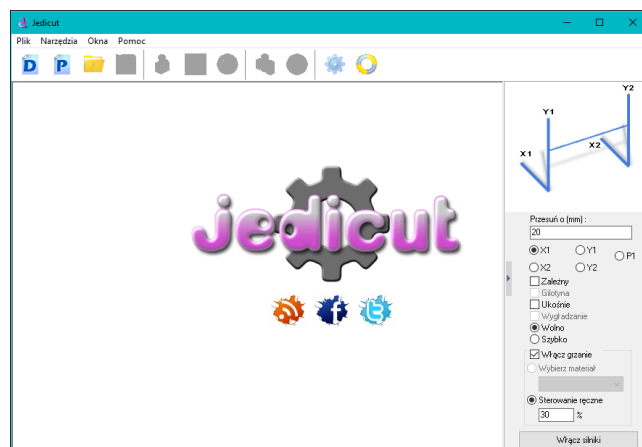
Ekran zmienia się po przejściu do trybu PC



Będziesz używał ręcznego sterowania z panelu Jedicut.

W oknie "Przesuń o" wpisz 10 mm, jest to ruch dodatni, trzeba odejść od FDC. Odnazcz "Zależny", zaznacz "Wolno", zaznacz "Włącz grzanie", zaznacz "Sterowanie ręczne" grzaniem i wprowadź 30%

Zaznacz oś X1, sprawdź, czy wózek odsuwa się od FDC, jeśli wózek powraca w kierunku FDC, to w tabeli poniżej w polu "Odwr." umieść "X" pod "DX1". Robisz to samo dla X2, Y1, Y2, (zaznaczasz pola "Dobry" lub "Odwr." pod DX2,DY1,DY2).



W tabeli po prawej masz przykład:

X1 => Dobry, w masce bitu 4 zaznaczamy "0", ponieważ po lewej "Dobry" jest "0"

X2 => Odwrotny, w masce bitu 5 umieszczamy "1", ponieważ po lewej "Odwr." jest "1"

Y1 => Odwrotny, w masce bitu 6 umieszczamy "1" ponieważ po lewej "Odwr." jest "1"

Y2 => Dobry, w masce bitu 7 umieszczamy "0", ponieważ po lewej "Dobry" jest "0"

W "conf.h" szkicu ustawimy dla #define INV_DIR_MASK 0b01100000

Twoja maszyna

8-bitowe słowo "Silnik-Kierunek"

		Kierunek							
		7	6	5	4	3	2	1	0
Znaczy		DY2	DY1	DX2	DX1	0	0	0	0
0	Dobry								
1	Odwr.								

Konfiguracja maski

		7	6	5	4	3	2	1	0
0	b					0	0	0	0

Przykład

8-bitowe słowo "Silnik-Kierunek"

		Kierunek							
		7	6	5	4	3	2	1	0
Znaczy		DY2	DY1	DX2	DX1	0	0	0	0
0	Dobry			x				x	
1	Odwr.				x	x			

Konfiguracja maski

		7	6	5	4	3	2	1	0
0	b	0	1	1	0	0	0	0	0

Dla twojej maszyny otrzymasz: #define INV_DIR_MASK 0b__ __ 0000

miejsca __ __ __ wypełniasz zgodnie wartościami ustawionymi w tabeli "Konfiguracja maski"

Podczas ruchu sprawdź czy drut się nagrzewa, patrząc na amperomierz. Jeśli się nie nagrzewa, sprawdź okablowanie.

Zamknij Jedicut,

Otwórz Arduino IDE.

W pliku "conf.h", w linii **44** zmodyfikuj maskę INV_DIR_MASK z wartościami, które zdefiniowałeś:
`#define INV_DIR_MASK 0b00000000` następnie ponownie wgraj szkic.

Zamknij Arduino IDE,

Uruchom Jedicut i sprawdź przemieszczenie każdej osi w kierunku dodatnim i ujemnym. Jeśli wszystko jest dobrze, możesz przejść do uruchomienia sekwencji "bazowania".

Zamknij Jedicut,

Otwórz Arduino IDE

W pliku "conf.h" poszukaj linii: **101 i 102**

`#define VIT_RECH_FDC 4.50` // Szybkość wyszukiwania fdc w mm/s Format XX.XX

`#define VIT_AJUST_FDC 1.00` // Szybkość regulacji fdc w mm/s Format XX.XX

Do testowania fdc, polecam zachowanie powyższych wartości prędkości wyszukiwania i regulacji, kiedy wszystko będzie działać poprawnie, możesz zwiększyć prędkość wyszukiwania.

W linii **107** zezwalasz na sekwencję "bazowania".

`#define SEQ_HOMING 0` // "1" Bazowanie aktywne, "0" Brak bazowania

Umieść "1" zamiast "0".

`#define SEQ_HOMING 1`

Linie **108 i 109** pozostaw tak do testów.

`#define POS_SECU_Y 1` // "1" wykonuje ruch osi Y przed bazowaniem

`#define MM_POS_SECU_Y 2` // wartość ruchu osi Y o x mm przed bazowaniem

Jeśli nie chcesz podnosić osi Y przed bazowaniem, ustawisz "0" na linii 108.

Możesz zmieniać wartość przesunięcia.

Nie dotykaj innych wartości, wstępne pozycjonowanie nie jest aktywne.

Prześlij szkic

Po zakończeniu inicjalizacji pojawia się następujący ekran:

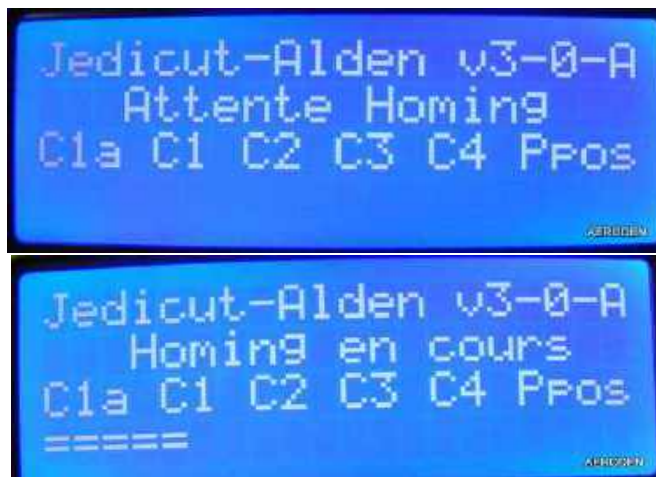
Dla wykonania sekwencji "bazowania" w celu ustawienia osi maszyny na zero.

Ustaw przełącznik "Silniki WŁ".

Naciśnij przycisk BP aktywujący bazowanie.

To automatyczna sekwencja. W przypadku pierwszego użycia należy zachować ostrożność, można zatrzymać sekwencję, ustawiając silniki w pozycji "Wył." lub przyciskiem awaryjnym BP, jeśli go okablowano.

Podczas ruchu masz pasek, który się przesuwają,



Następnie na ekranie pojawi się komunikat "Koniec bazowania", widzimy że nie ma wstępnego pozycjonowania, pasek zatrzymał się na **C4** i po 3 sekundach powraca do normalnego ekranu. Widzimy że wstępne pozycjonowanie nie jest aktywne, dzięki czemu można skorygować mechanicznie położenie **fdc**. Spójrz, jak drut jest ustawiony w stosunku do stołu, (**X** ustawia drut prostopadłe do stołu, a **Y** ustawia drut równoległe do twojego stołu)



Kiedy nieznacznie zmienisz ustawienie zadziałania **fdc** na osi, następuje powrót do bazowania. Silniki zostają wyłączone, co skutkuje przerwaniem bazowania. Włączasz z powrotem silniki i naciskasz **BP** aktywując bazowanie.

Możesz powtórzyć tę akcję tyle razy, ile potrzeba, aby uzyskać idealne zero.

Dla tych, którzy mają napęd pasowy i zdecydowali się utrzymać silniki po napięciu, należy sprawdzić, czy nie można ich obracać, gdy przełącznik silników jest włączony.

Przypomnienie: Po wyłączeniu silników następuje **utrata bazowania**. Dlaczego tak jest? Ponieważ w maszynach z napędem pasowym, po wyłączeniu napięcia silników, osie (szczególnie pionowe Y) mogą ulec przesunięciu. Nie różnicowałem szkicu odrębnie dla maszyny śrubowych, ponieważ ponowne bazowanie jest tak szybkie, że nie stanowi problemu.

Teraz zamierzamy uruchomić pozycjonowanie wstępne, to jest pozycja robocza, którą określasz zgodnie ze swoimi nawykami. Dla mojej maszyny umieściłem **1mm** od **fdc** na osiach **X** i **1mm** od **fdc** na osiach **Y**. Tak jest ustawione domyślnie w pliku "conf" (możesz to dowolnie zmienić).

Zauważyłeś, że nie otworzyliśmy **Jedicuta**, Arduino IDE jest nadal otwarte, i teraz w "conf.h" zmodyfikuj linię **110**

```
#define PREPOS 0 // "1" Wstępne pozycjonowanie po zezwoleniu na naprowadzanie  
po #define PREPOS 1
```

Dla linii **112** i **113** można umieścić **5 mm** i **10mm** będzie bardziej widoczny niż **1 mm**
domyślne ustawienie:

```
#define MM_PREPOS_Y 1 // Wartość w mm odległości pozycjonowania wstępnego Y  
#define MM_PREPOS_X 1 // Wartość w mm odległości pozycjonowania wstępnego X
```

po zmianie:

```
#define MM_PREPOS_Y 5 // Wartość w mm odległości pozycjonowania wstępnego Y  
#define MM_PREPOS_X 10 // Wartość w mm odległości pozycjonowania wstępnego X
```

Prześlij szkic

Poczekaj na uruchomienie bazowania, ustaw przełącznik silnika w pozycję ON "Bazowanie", zobaczysz, że pozycjonowanie wstępne jest realizowane. Twój drut jest **5 mm** na **Y** i **10 mm** na **X** od pozycji zerowej. Zawsze ustawiaj najpierw osie **Y**, aby zapobiec opadnięciu drutu na stół.

Zamknij Arduino IDE i otwórz Jedicut:

Sprawdźmy zatrzymanie na **fdc**.

Przez panel sterowania ręcznego Jedicut:

W elektronice ustawiamy, **Tryb PC**, **Silniki Wył** (już wykonałeś bazowanie),.

Oś po osi, wykonujesz ujemne przesunięcie **-10 mm**, jeśli masz pozycjonowanie wstępne ustawione na **5 mm**, natomiast jeśli masz pozycjonowanie ustawione **25 mm**, wykonujesz przesunięcie ujemne **-35 mm**.

Gdy wózek załączy **FDC**:

Rozlegnie się sygnał dźwiękowy

Silniki się zatrzymują.

Pojawia się "ekran wyłączników krańcowych", informujący, która **fdc** spowodowała zatrzymanie.

Naciskasz "BP" (uruchamiający bazowanie):

Sygnal dźwiękowy gaśnie,

Jesteś poproszony o manipulowanie przełącznikami.

Naprowadzanie odbywa się przez włączenie silników, wciśnięcie BP naprowadzania.

Przełączysz się na inną oś, powinno być to samo.

Tutaj są symulacje, w przypadku gdy jeden wózek atakuje FDC podczas cięcia, zatrzymuje wszystko, ale gorący drut jest uwięziony w bloku.

Po potwierdzeniu sygnału dźwiękowego, użytkownik zostanie poproszony o przestawienie przełączników w położenie, a następnie ręcznie włączy grzejnik, który pozwala za pomocą potencjometru lub enkodera sterować ogrzewaniem drutu i uwolnić drut z bloku.

Następnie możesz wykonać naprowadzanie, a twoja maszyna jest gotowa.

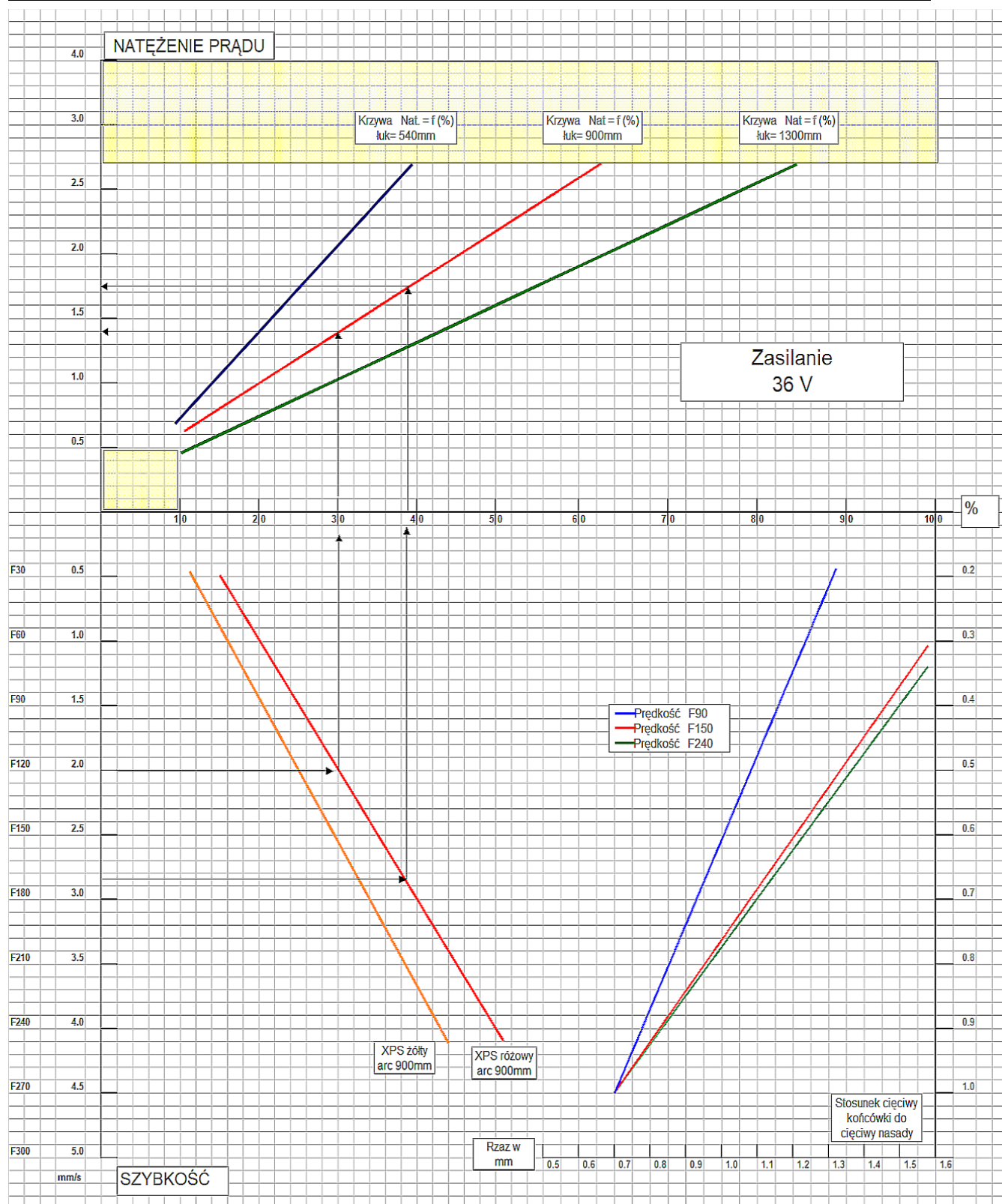
Kontrolowanie grzania:

Przeprojektowanie programu Jedicut spowodowało problem z połączeniem USB dla dynamicznego sterowania grzaniem. Jedicut musi pozostać kompatybilny ze starym sprzętem i musi współpracować z kartami USB. Wraz z Jerome'em trochę posunęliśmy się naprzód w tej kwestii, ale nie rozwiązaliśmy jeszcze wszystkiego. Kiedy opracowałem bazowanie wersji Super Luks, zastanowiłem się nad alternatywą z kolejnością kroków, traktując matematycznie kroki, dają one trygonometryczny obraz trajektorii i prędkości na trajektorii, te obliczenia pozwolą mi zmieniać ogrzewanie. Wada: kiedy chciałem zmieniać PWM grzejnika, przyjmowały wartość 100%. Po kilku godzinach badań znalazłem problem. Moje obliczenia są wykonywane z częstotliwością kroków, dla mojego komputera jest to 2000 Hz; domyślna częstotliwość PWM używanego wyjścia Arduino wynosi 976 Hz generowane przez timer 4, więc aktualizuję rejestr PWM przed wygenerowaniem sygnału PWM, co powoduje 100% wzrost temperatury. Rozwiązaniem było uzyskanie wyższej częstotliwości PWM. Częstotliwość timera 4 wpływa w naszej aplikacji tylko na PWM, a timer 0 generuje opóźnienia, tak było od początku szkiców dla Jedicut-Alden. Teraz dla generowania przerw wybrałem timer 5, który nie jest używany przez biblioteki Arduino. Mogę więc zmienić częstotliwość PWM, działając na rejestr w momencie konfiguracji. Zmieniłem częstotliwość na 7800 Hz i teraz nie ma problemu, zmiana ogrzewania jest taka sama jak moje obliczenia.

Prędkość trajektorii jest największa, gdy drut przechodzi przez odcinek pod kątem 45°.

Dla prędkości 2mm/s w osi X i Y prędkość drutu wynosi $2 \times 1,414 = 2,828$ mm/s.

Z wykresu materiału dla danego łuku, zanotuj wartość w % dla 2 mm/s --> 30% następnie wartość dla prędkości 2,828 --> 39,5% teraz dzielisz $39,5 / 30 = 1,3166$, wynik jest współczynnikiem korekcji który wynosi 1,32. Szkic dobrze sobie radzi w obliczeniach prędkości pośredniej drutu. Dotyczy to danego łuku. Linie materiałów są równoległe do siebie. Dla swoich obliczeń, powinieneś przyjąć normalną prędkość cięcia pomnożoną przez 1,414.



W folderze znajduje się plik "[Intensite_pourcentage_chauffe.pdf](#)". Są to 2 puste arkusze do prześledzenia twoich ustawień. Dla tych, którzy znają arkusz kalkulacyjny Excel, masz plik "[Intensite_pourc.xls](#)", wystarczy wprowadzić swoje wartości zamiast moich.

Jeśli chcesz włączyć sterowanie grzaniem, otwórz "conf.h" i w liniach 154 i 155. Wpisujesz 1 dla CHAUFFE_ASSERV oraz wpisujesz współczynnik korekcji (1.32) dla CORRECT_CHAUFFE.

#define CHAUFFE_ASSERV 0 // "0" bez sterowania grzaniem, "1" ze sterowaniem.

#define CORRECT_CHAUFFE 1.32 // "1.32" to współczynnik korekcji grzania zależny od prędkości.

Ponownie przesyłasz nowy szkic.

[Zamknij Arduino IDE](#)

[Otwórz Jedicut](#)

Uruchamiasz pionową gilotynę z ręcznymi parametrami grzania, i odnotowujesz wskazania amperomierza dla swojej prędkości cięcia.

Następnie uruchamiasz skośną gilotynę pod kątem 45°, umieszczając jednakowe wartości w osiach X oraz Y. Notujesz wskazania amperomierza. Widzisz, że jest to bardziej istotne.

Kiedy wycinasz kształty za pomocą krzywych, zobaczysz bardzo małe zmiany, ale gdy wycinasz krawędź natarcia w jednym przejściu, zobaczysz znaczące różnice na amperomierzu. Rzeczywiście, korekty nie są widoczne na wyświetlaczu LCD, zmiany przy 2000 Hz dawałyby nieczytelny obraz wartości grzania.

Korzystanie z poleceń:

Przełączniki poleceń **Tryb/MODE** i **Siln/MOT** muszą być odwrócone w górę, aby uwzględnić polecenia pochodzące z **Jedicut**. Jeśli przełączysz jeden z tych 2 przełączników podczas cięcia, nastąpi wyłączenie silnika, zatrzymanie grzania, **Jedicut** zakończy swoją sekwencję, a Arduino opróżni swój bufor. Masz możliwość kontrolowania ogrzewania drutu przez komputer (**Jedicut**) lub w trybie ręcznym nawet podczas cięcia. Ustawienie ogrzewania odbywa się za pomocą potencjometru lub enkodera obrotowego, zależnie od wyboru.

Polecam potencjometr jest znacznie szybszy.
Więc możesz użyć tych 2 przełączników **Tryb**, **Siln**.
jako zatrzymanie awaryjne.

Jeśli chcesz użyć ustawienia wartości zadanej ogrzewania przez enkoder:

Musisz wybrać enkoder obrotowy w konfiguracji szkicu Arduino.

Korzystanie z enkodera

Naciśnij przycisk enkodera, zobaczysz znak ">" przed nastawą ogrzewania, obrócisz enkoder do wybranej wartości, tutaj **26%**, ponownie wciskamy przycisk enkodera, znak ">" znika, wartość jest brana pod uwagę przy ogrzewaniu.

Jeśli nagrzewnica jest ograniczona do **80%** w konfiguracji szkicu Arduino, ustawienie ręczne będzie ograniczone do **80%**, jak w przypadku sygnału pochodzącego z **Jedicut**. Zaleta enkodera obrotowego, ustawienie zostaje zachowane, gdy poruszyś przyciskiem, jednak jest wolniejszy niż potencjometr. Enkoder jest z wyświetlaczem, zrobiłem mały program dla mojej wiedzy o enkoderze, jednak wolę potencjometr.



Sterowanie ogrzewaniem noża elektrycznego jest dostępne, gdy **Tryb** znajduje się w poz. **Ręczn**.

Ekran podczas cięcia z wolną prędkością 2,52 mm/s i 28% grzaniem.

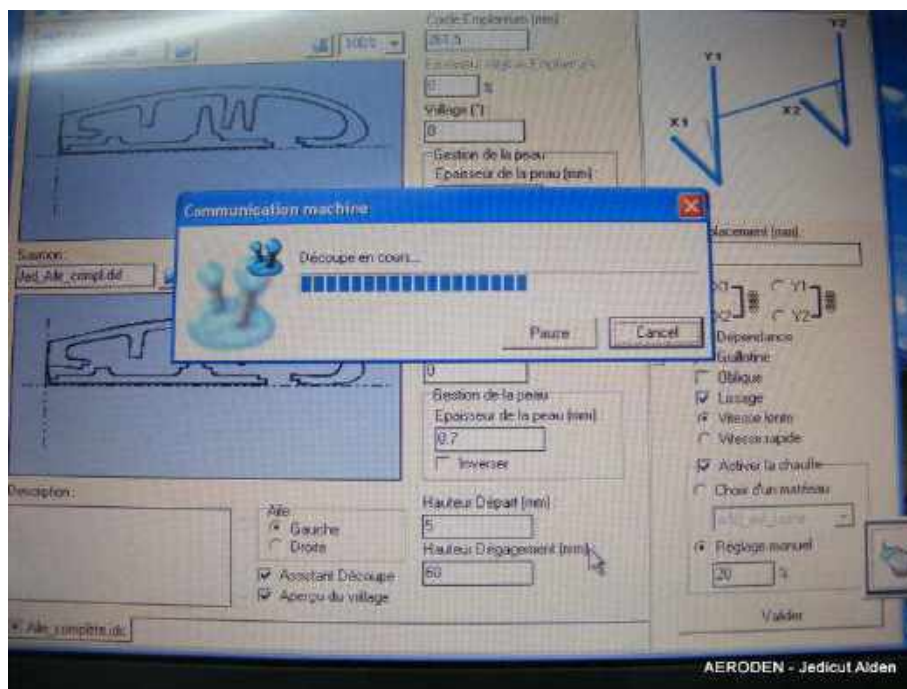


Ekran podczas cięcia z wysoką prędkością 4,46 mm/s i 30% grzaniem.

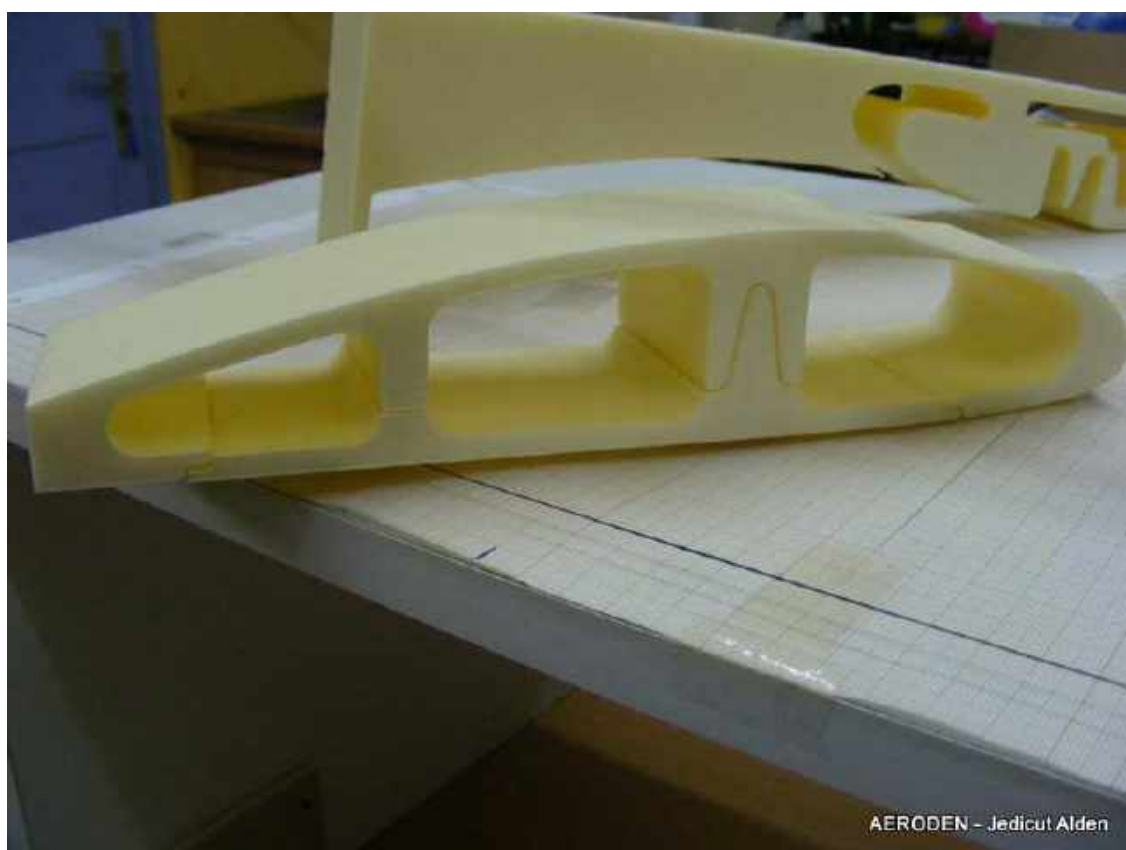


Na zakończenie twoich testów polecam wypróbować dołączony kompletny plik cięcia:

[Aile_complete.jdc](#)



Ponieważ są to próby, brakuje krótkiego fragmentu.



Wykonanie obudowy

Plany mechaniczne obudowy są umieszczone w folderze [Jedicut_Alden3D](#) i są częścią pliku



Obudowa z dużą ilością otworów wentylacyjnych.



Zasilacz gorącego drutu po prawej stronie 36V 10A i zasilacz elektroniki 12V 10A po lewej stronie, bez obudowy w celu lepszego chłodzenia.



Ustawienie płytek i złącza DB25



Widok złącza sieciowego wyposażonego w bezpiecznik $5 \times 2 \text{ 8 A}$;
Idealem byłby bezpiecznik opóźniony o wartości 5 A wytrzymujący pobór prądu rozruchowego z 2 zasilaczy impulsowych.



Tylna ścianka



Panel sterowania ręcznego

Gotowa obudowa:

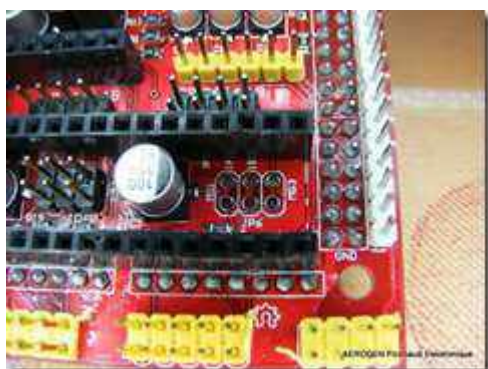


Opiszę tutaj awarię, z którą miałem do czynienia w innej instalacji.

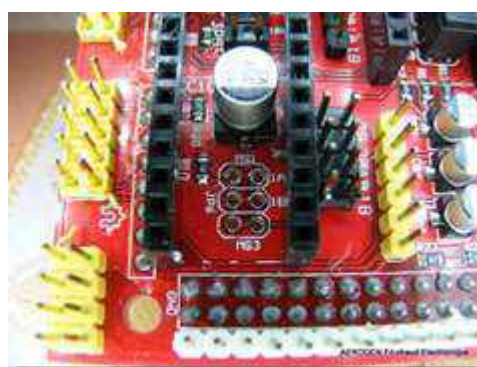
W testach silnik krokowy porusza się niewiele w jednym kierunku i powraca do pozycji wyjściowej.

Wnioskuje, że jest problem z wyborem mikro-kroków.

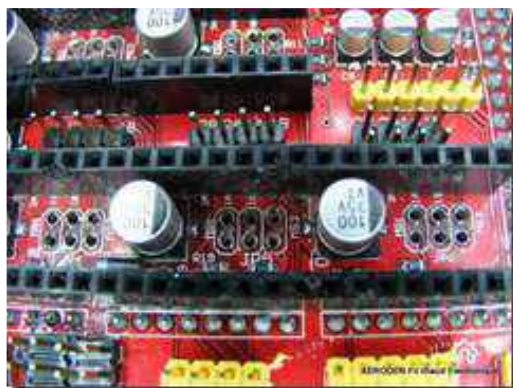
Nie myliłem się w kwestii położenia zworek, **dla całych kroków nie ma zworek**. Testuję omomierzem i widzę, że piny są połączone. Odrywam podstawki pinów i tam odkrywam, że są połączone miedzianą ścieżką. Wybór był zatem na stałe 1/16 kroku, co nie tłumaczy, dlaczego silnik obrócił się bardzo niewiele. To było coś podejrzanego.



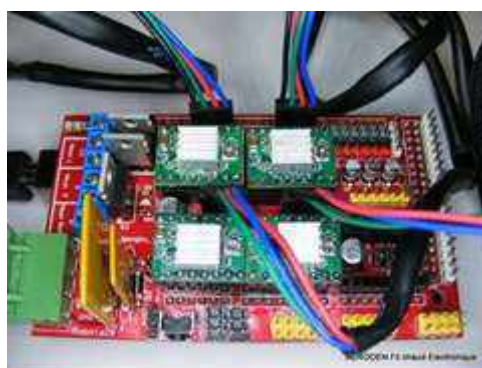
Nieprzylutowane kołki pokazują ścieżki łączące pady.



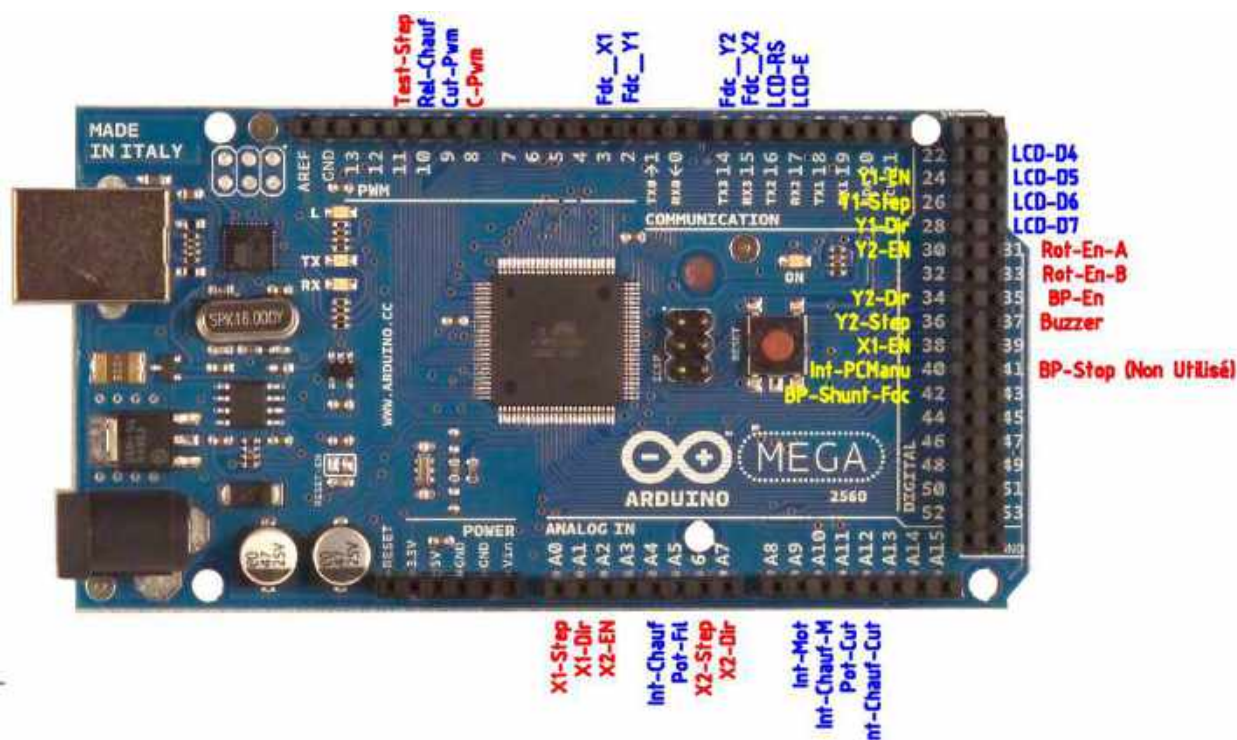
Inny widok



Cięcie ścieżek za pomocą frezu i mini-wiertarki. Identyczne traktowanie wszystkich 5 osi płytki Ramps 1.4.



Czyszczenie, i ponowny montaż zworek ustawionych na 1/2 kroku. W moim programie testów umieściłem 400 impulsów zamiast 200 tak aby silniki wykonały 1 obrót. Test jest rozstrzygający, działa 5 osi.

Identyfikacja wejść i wyjść na płytce Arduino Mega 2560:

Rozwiązywanie problemów:

Jeśli nie masz komunikacji między Jedicutem i elektroniką (kiedy uruchamiasz ruch w Jedicutem, pasek postępu jest bardzo szybki, trudno go zobaczyć) musisz sprawdzić plik "[comport.ini](#)" w katalogu głównym Jedicut z edytorem tekstu. Patrz strona 24.

Dokumentacja jaką przekazuję do dyspozycji w celu dokończenia zrozumienia całości:

- [Ard_Ramps_IDE_3.pdf](#) - autor Alain
- [Jed_Fdc2.pdf](#) - autor Alain
- Folder [Calcul de transmission CNC](#) - (pliki *.pdf, *.xls, *.ods) - autor Alain
- Folder [Chauffe](#) - (pliki *.pdf, *.xls, *.ods) - autor Alain

Historia zmian

Wersja: **0**

Pierwsze wydanie

Wersja: **1**

Nowy szkic części komunikacyjnej między komputerem a elektroniką (nie używający bibliotek Arduino, jako zbyt wolnych do przetwarzania). Ten szkic jest kompilacją szkicu Oliviera i Vincenta (część komunikacji USB i poleceń dekodowania, z motywem Alaina (automatyzm, opcja, korekcja ciepła, opcja językowa)

Dodano wybór języka dla wyświetlacza LCD. - strona **16**.

Dodano obrazy dla wyłącznika krańcowego. - strona **31**.

Dodano konfigurację "comport.ini". - strona **34**.

Wersja: **2** z dn. 01.03.2019

Nowy szkic [LMFAO_V4_5_0](#) części komunikacyjnej między komputerem a elektroniką (nie używający bibliotek Arduino, zbyt wolny do przetwarzania). Ten szkic kompilacją szkicu Oliviera i Vincenta (część komunikacji USB i poleceń dekodowania, z motywem Alaina (automatyzm, opcja, korekcja ciepła, opcja językowa)

Wersja: **3** z dn. 17.04.2019

Dodano na stronie **2 ograniczenie** dotyczące identyczności wszystkich osi

Wersja: **3a** z dn. 11.04.2019

Nowa wersja szkicu: [LMFAO_V4_6_0](#) naprawa pauzowania

Wersja: **3b** z dn 07.11.2019

Nowa wersja szkicu: [LMFAO_V4_7_0](#) możliwość pracy z enkoderem, zatrzymanie ręcznego grzania przez inter, regulacja grzania podczas cięcia,