



Połączenie USB Jedicut dla MM2001 lub innych sterowników

Sterowanie grzaniem drutu
zależne od prędkości i trajektorii

Aeropassion AERODEN	Realizacja Jedicut USB MM2001 z Arduino UNO	AD/ELE/ARD/02
------------------------	--	---------------

Kontekst:

Ewolucja sprzętu komputerowego powoduje, że porty równoległe znikają. Jednym z rozwiązań jest użycie portu USB. Aby zachować stare karty sterujące sprzętem CNC, konieczne jest utworzenie interfejsu USB / DB25.

Ograniczenie: Uwaga: ten interfejs działa poprawnie, pod warunkiem że mechanika przecinarki CNC jest wyposażona w 4 osie o identycznych charakterystykach (silnik krokowy ze śrubami / nakrętkami, silnik krokowy z zębniakiem i pasem), których nie możemy zmieniać.

Zasada:

Jedicut wysyła informacje bardzo szybko, są one kodowane na dwóch bajtach, jeden bajt oznacza polecenie, a drugi bajt reprezentuje wartość tego polecenia. Interfejs USB/DB25 wyposażony w Arduino UNO i jego program musi zebrać wszystkie te informacje, zdekodować je, wygenerować polecenia, zsynchronizować je, aby uzyskać odpowiednie prędkości i skierować je na odpowiednie wyjścia DB25; i to bez utraty jakichkolwiek informacji. Ponieważ ważna jest ilość przetwarzanych informacji, Arduino nie może odbierać wszystkich informacji jednocześnie. Program Arduino używa bufora 900 bajtów, aby to zrobić. Gdy bufor jest prawie pełny, Arduino wysyła do Jedicut rozkaz „S” zatrzymaj transmisję, gdy przestrzeń zostanie zwolniona w buforze, Arduino wyśle Jedicutowi rozkaz „C” Kontynuuj transmisję. To jedyne 2 rozkazy, które wracają do Jedicut.

Podsumowanie:

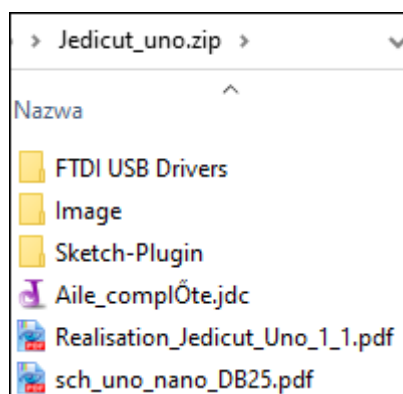
Jedicut oblicza parametry cięcia, trajektorii, grzania, prędkości i dostarcza to wszystko do biblioteki DLL. Biblioteka DLL wysyła dane do USB i odczytuje informacje pochodzące z Arduino. W ten sposób reguluje przepływ, dzięki czemu można sprawdzić, czy istnieje komunikacja między Jedicutem a Arduino. Bez komunikacji, pasek przewijania Jedicut jest bardzo szybki, z poprawną komunikacją pasek przewijania zatrzymuje co jakiś czas, gdy bufor Arduino jest prawie pełny.

Arduino ma zegar, który pracuje z częstotliwością **62500 Hz**. Jest to timer, który pozwala na zachowanie prędkości cięcia i prędkości jałowej (poza materiałem). Ten zegar generuje przerwę za każdym razem, gdy musi zostać wysłany krok do silników.

Wyjaśnię trochę bardziej szczegółowo. Szerokość impulsu krokowego dla płyty sterującej można regulować. Dla MM2001 potrzeba 20 µs; zaś 2µs wystarczą dla karty Ramps. Na kartach Arduino, sygnał USB nie wchodzi bezpośrednio na mikrokontroler, jest przekształcany przez dedykowany obwód FTDI, µC Atmeg, lub CH340. Gdy karta arduino jest wyposażona w obwód CH340, nie jest zadeklarowana jako zgodna. Jednak działa tak samo.

Nowość tego szkicu polega na wykorzystaniu możliwości obliczeniowych Arduino w celu skorygowania ogrzewania w zależności od prędkości na trajektorii.

Zebrałem w folderze do pobrania [Jedicut_Uno.zip](#), wszystkie pliki niezbędne do realizacji.



Folder z plikami sterowników dla portu COM: [FTDI USB Drivers](#)

Folder obrazów: [Image](#)

Folder z program Arduino i bibliotekami DLL: [Sketch-Plugin](#)

Plik cięcia testowego: [Complete_Aile.jdc](#)

Plik instrukcji do realizacji: [Realization_Jedicut_Uno.pdf](#)

Pliki schematu interfejsu: [sch_uno_nano_DB25.pdf](#)

Realizacja:

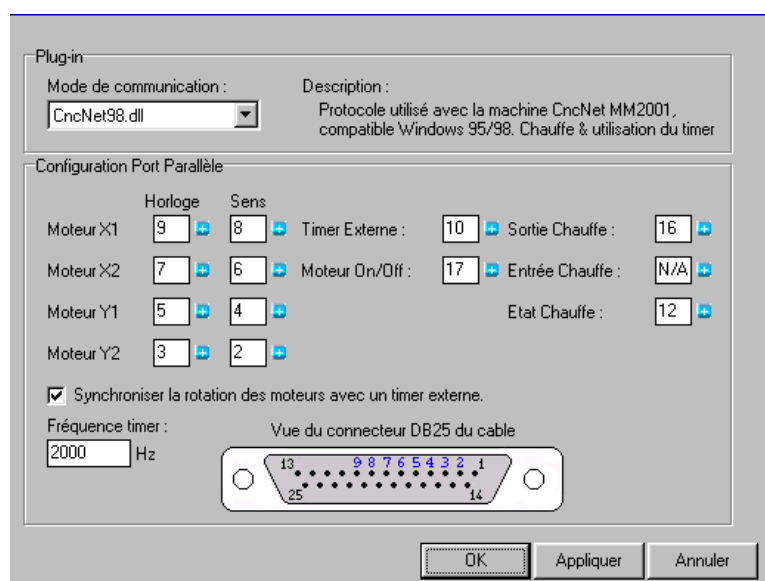
Krok 1:

Sposób okablowania płytki interfejsu, pozwala pobrać różne sygnały z arduino i podłączyć je do terminala DB25 urządzenia. Załączony schemat jest okablowaniem mojego MM2001 odpowiadający mojej maszynie. Okablowanie musi być zgodne ze starą konfiguracją Jedicut dla portu równoległego dla złącza DB25 LPT.

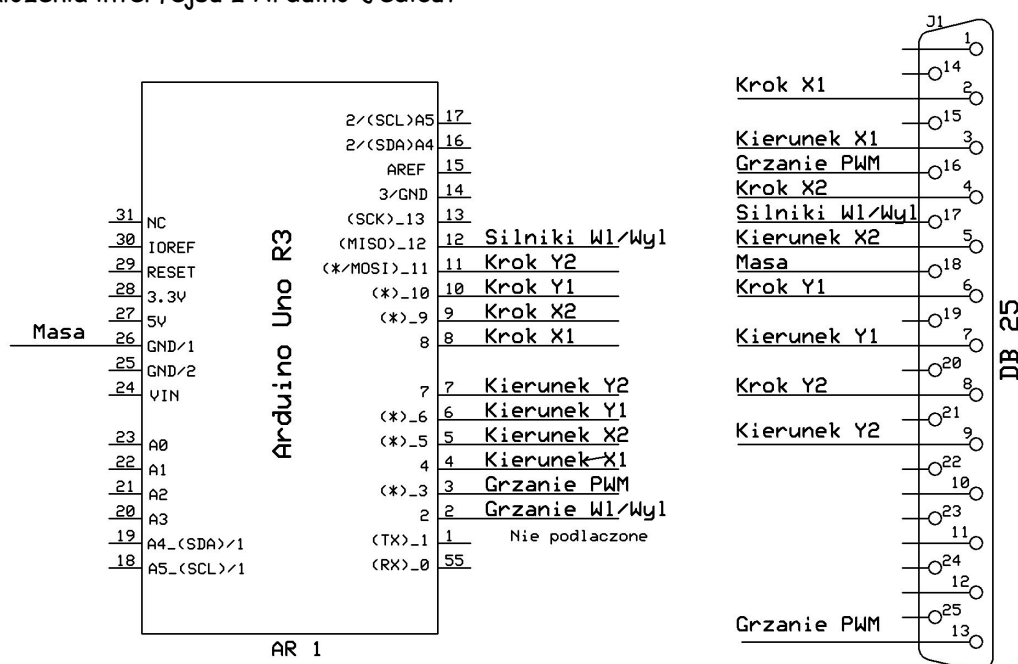
PONIŻEJ znajdziesz plany realizacji na obwodach drukowanych.

https://drive.google.com/open?id=12vuPN4txzyhFu8e1aj8RURFNJ_IXvVtq

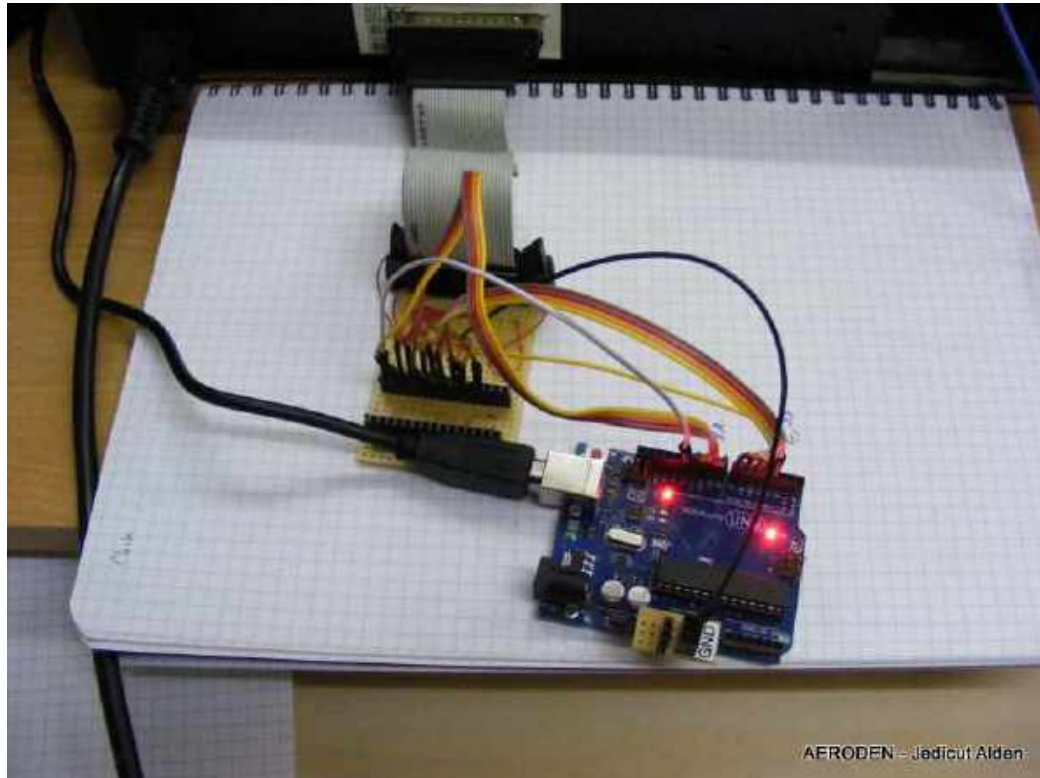
Oto, jak wyglądała stara konfiguracja Jedicut:



A tu założenia interfejsu z Arduino-Jedicut



Okablowanie jest zatem zgodne z kartą sterowania silnika.



Krok 2:

Przesyłanie programu do płyty Arduino.

Nie opisuję szczegółowo instalacji Arduino IDE, zobacz początek pliku do pobrania.

[Plik do pobrania PONIŻEJ.](#)

https://drive.google.com/open?id=1p8_Ft5xPQFgAiQq7KEPnrMGNCgcVxUW5

Po zainstalowaniu **Arduino IDE** na PC, w „twoich dokumentach” musisz znaleźć katalog „**Arduino**” i podkatalog „**Applications**”. Kopiujesz folder „**ad_fc_uno-4**”, który pobrałeś do tego podkatalogu. Otwierasz **Arduino IDE**, otwierasz szkic i w **Arduino IDE**, widzisz 2 zakładki: „**ad_fc_uno-4**” i „**conf.h**”, klikasz na zakładkę „**conf.h**”



Aeropassion AERODEN	Realizacja Jedicut USB MM2001 z Arduino UNO	AD/ELE/ARD/02
------------------------	--	---------------

Dostosowanie konfiguracji do parametrów twojej maszyny w pliku: „conf.h”

// Ograniczenie mocy grzania drutu

define LIM_P_FIL 100

W tym przypadku moc grzewcza nie jest ograniczona, jest równa 100%

Jeśli twój drut zaczerwieni się przed osiągnięciem mocy 100%, możesz zmniejszyć ustawienie limitu np. na 80%. Wpisujesz 80 zamiast 100.

Limit ten jest aktywny tylko wtedy, gdy falownik w MM2001 jest w położeniu na PC.

// Prędkość USB

define BAUDRATE 115200 // 110,300,600,1200,2400,4800,9600,14400

//19200, 38400,56000,57600,115200,128000,256000

Jest to prędkość transmisji USB, nie zmieniasz tej wartości pozostawiając 115200.

Jest to wielka innowacja tej wersji, grzanie drutu może być kontrolowane przez prędkość na trajektorii.

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

define CORRECT_CHAUFFE 1.32 // współczynnik grzania w funkcji prędkości

Na razie nie zmieniasz niczego w tych dwóch parametrach, wróćmy do tego, gdy przeprowadzisz pierwsze testy działania.

Przesyłasz program [ad_fc_uno-4.ino](#) do arduino. Zauważ, że używany jest port COM.


Krok 3:

Na komputerze sprawdź folder Jedicut w „Program Files” czy w katalogu dll widnieją następujące dwa pliki; „USBSerial.dll” i „USBSerial_2_P.dll” .

Otwórz plik [comport.ini](#) w katalogu głównym Jedicut. Zmodyfikuj plik [comport.ini](#), aby nadać mu port komunikacyjny, który zanotowałeś podczas przesyłania.

Konfiguracja Jedicut w wersji 2.4.1:

Otwórz Jedicut

Wybierz: „Narzędzia”, „Opcje” lub kliknij ikonę  otwórz kartę „Komunikacja”

Wypełnij w ten sposób:

Aktywny stół CNC: Nazwa twojego stołu.

Tryb komunikacji:

wybierz USBSerial_2_P.dll

Konfiguracja portu równoległego:

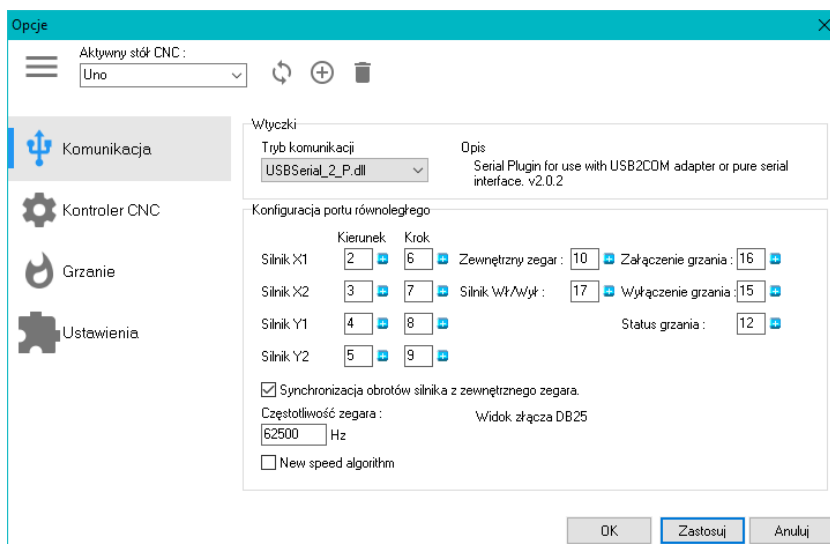
wartości są narzucone przez program Arduino.

Zaznacz pole „Synchronizacja obrotów silnika ”

Częstotliwość zegara: wpisz 62500 Hz.

Przy połączeniu USB ważne jest, aby **nie zaznaczać** „New speed algorithm”

Kliknij „Zastosuj” i „OK”



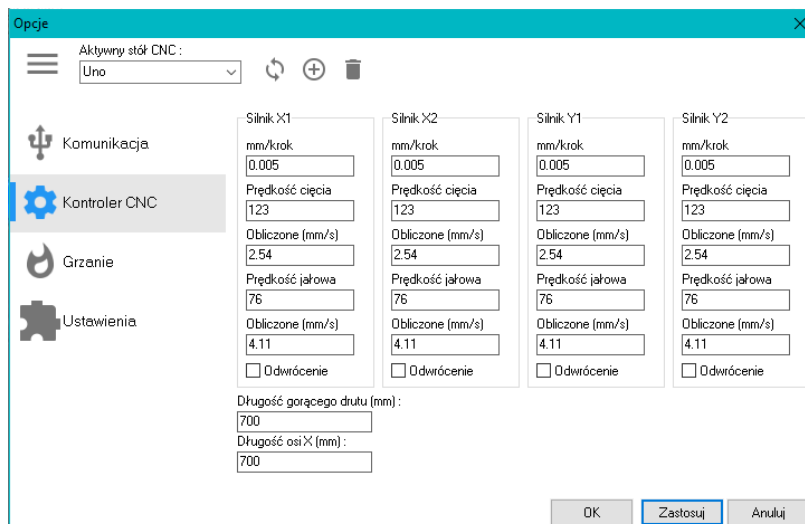
Otwórz: „Narzędzia”, „Opcje” lub ikonę  otwórz kartę „Kontroler CNC”

Charakterystyka mechaniczna

twojej maszyny to: Pręt gwintowany 6 mm x 1 mm, silnik 100 kroków na obrót, elektronika MM2001 ustawiona na $\frac{1}{2}$ kroku, musimy zatem przyjąć 200 kroków na obrót, co daje $1 \text{ mm} / 200 = 0,005 \text{ mm} / \text{krok}$.

Dobieramy współczynniki prędkości cięcia i prędkości jałowej a prędkości w mm/s zostaną obliczone przez Jedicut:

W poprzednim widoku ustawiliśmy częstotliwość taktowania 62500 Hz, podzielimy tę częstotliwość przez wartość współczynnika 123, i otrzymamy częstotliwość = 508 Hz, która zostanie przesłana do Arduino, i wygeneruje kroki z tą częstotliwością.



Teraz obliczymy prędkość posuwu $0,005 \text{ mm} \times 508 = 2,54 \text{ mm/s}$ Obliczenia Jedicut są dobre.

Dla prędkości jałowej (poza materiałem) wstawiamy 76 co daje nam 4,11 mm/s.

Ostrzeżenie: te wartości współczynników prędkości są ograniczone w programie Arduino od 32766 dla max. do 63 dla min. Jeśli musisz zejść niżej, jest to możliwe, ale konieczne jest posiadanie koncentratora USB między komputerem a Arduino. Hub poprawia prędkość transmisji (nie znalazłem powodu dlaczego). Współczynnik nie powinien być jednak niższy niż 30.

(Hub NGS 4 porty bez zasilania).

Jeśli chcesz szybszych prędkości, wybierz pełny krok w twojej elektronice MM2001 i osiągniesz 0,01 mm / kroki.



Podczas prób możesz powrócić do tego widoku, aby odwrócić ruchy w razie potrzeby zaznaczając okienko "Odwrócenie" dla wybranej osi.

„Zastosuj” i „OK”

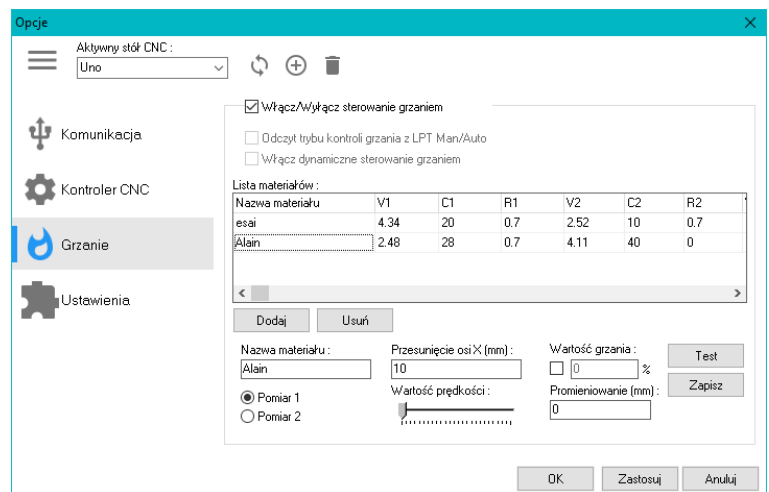
Kliknij: „Narzędzia”, a następnie „Opcje” i „Grzanie”

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

Pozostałe pola nie mogą być zaznaczone w połączeniu USB.

Aby wprowadzić materiały, zobacz instrukcję obsługi Jedicut.

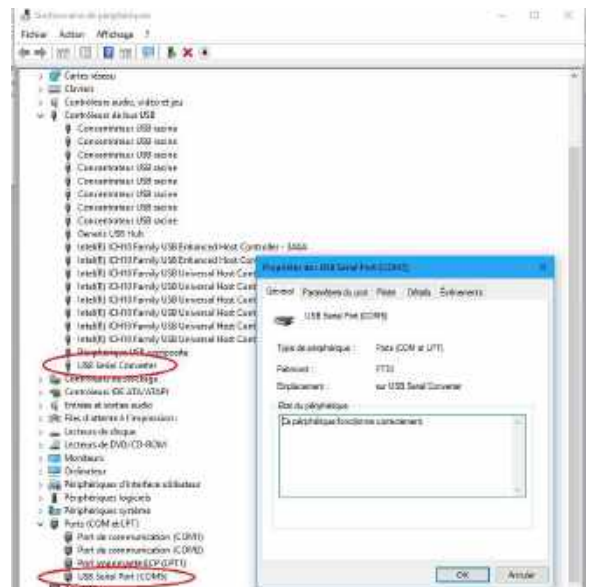
Materiały nie są przydatne do naszych testów do czasu kalibracji grzania



Zamknij Jedicut.

Podłącz kartę Uno do portu USB, ale nie podłączaj jeszcze DB-25 do MM2001.

Sprawdź, czy interfejs został wykryty w widoku menedżera urządzeń.

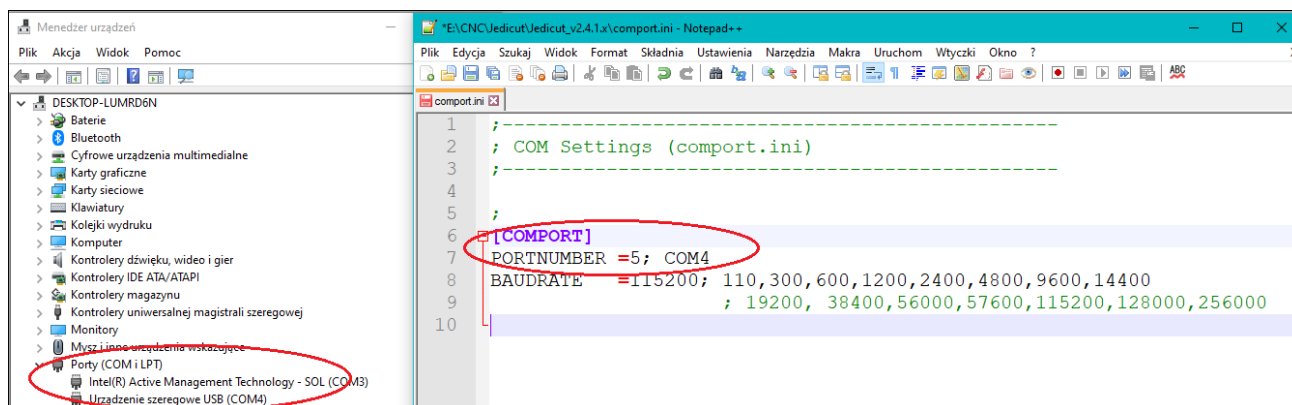


Jeśli masz znaki zapytania na sprzęcie, spróbuj zaktualizować sterownik, dołączyłem dla ciebie sterowniki w pobranych folderach.

Wydaje się, że USB Serial Converter instaluje się tylko ze sterownikiem FTDI.

Jedicut jest skonfigurowany, ale musisz sprawdzić i skonfigurować [Comport.ini](#), który znajduje się w głównym katalogu plików programu Jedicut.

Przejdź do menedżera urządzeń i poszukaj numeru portu [com](#) używanego przez Arduino, otwórz plik [comport.ini](#) za pomocą notatnika, i zmień jeśli trzeba numer portu [com](#) na ten z menedżera urządzeń, jak poniżej. [Zapisz plik \(z tą samą nazwą\)](#).



Nie podłączaj jeszcze karty interfejsu do elektroniki MM2001 za pomocą DB25.

Otwórz Jedicut i włącz ręczne sterowanie gilotyną 10 mm; musisz zobaczyć na płytce UNO migające zielone diody LED transmisji, dioda L13 może zgasnąć, zaświeci się ponownie po zakończeniu transmisji. Podczas tego polecenia Jedicut generuje pasek przewijania zadania. Jeśli pasek przewijania zostanie zamknięty natychmiast, oznacza to brak komunikacji. Szukasz przyczyny.

Jeśli wszystko w porządku, zamknij Jedicut, podłącz kartę Uno do elektroniki sterującej MM2001 przez DB25.

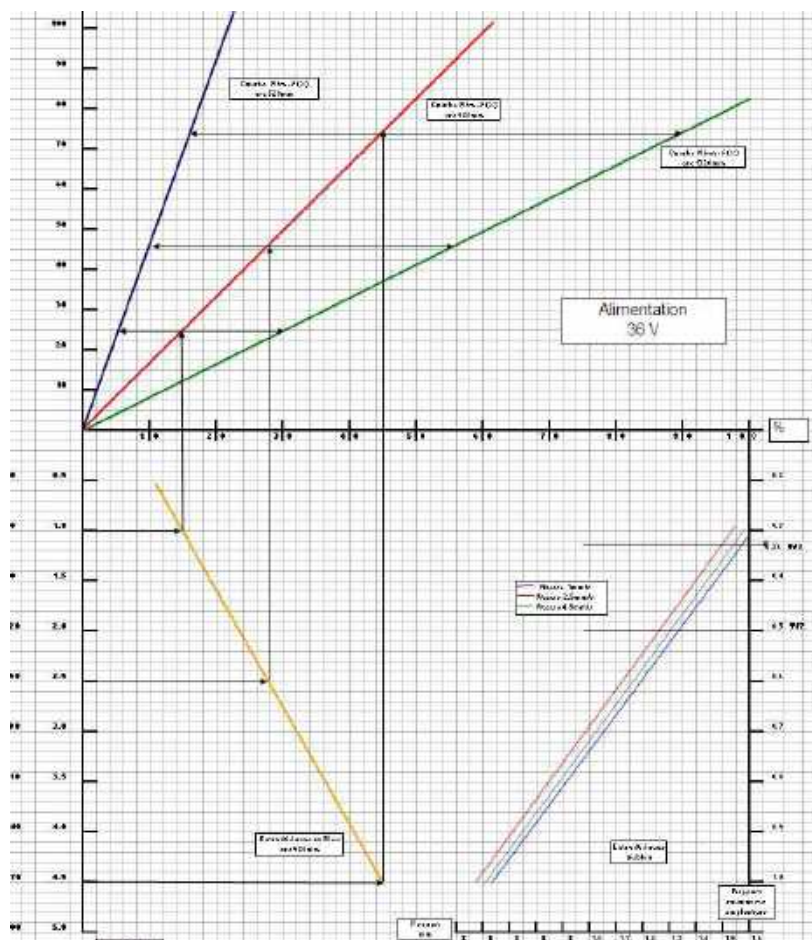
Uruchom Jedicut i sterowanie ręczne osi, aby sprawdzić kierunek obrotów. Jeśli konieczne jest odwrócenie kierunku, przejdź do „[Narzędzia](#)”, następnie „[Opcje](#)” i zakładki „[Kontroler CNC](#)”, zaznacz pole „[Odwrócenie](#)” dla wymaganej osi, a następnie „[Zastosuj](#)” i „[OK](#)” teraz sprawdź działanie.

Zmierz również długość ruchu osi, sprawdzając czy faktyczna długość przesunięcia osi odpowiada zadanej w programie Jedicut.

Gdy wszystko działa prawidłowo, można skalibrować grzanie drutu. Zapoznaj się z pełnym dokumentem. „JA_Chauffe_Paramétrage.pdf”. [Pobierz go z linku poniżej.](#)

https://drive.google.com/open?id=1LLHFi_ekmk7PBEPfjuj80FvRpWPHuRe6

Teraz masz wszystkie dane dla swojej maszyny,



Będziesz mógł przeprowadzić konfigurację kontroli grzania w powiązaniu z prędkością.

Zamykasz Jedicut, otwierasz Arduino IDE. **Nigdy nie powinno się otwierać Jedicut i Arduino IDE w tym samym czasie, jest konflikt, ponieważ używają tego samego portu komunikacyjnego.**

Kliknij kartę „Conf.h”.

kontrolowanie grzania drutu

Przeprojektowanie programu Jedicut spowodowało problem z połączeniem USB do dynamicznego ogrzewania. 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:** jednak 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;

Aeropassion AERODEN	Realizacja Jedicut USB MM2001 z Arduino UNO	AD/ELE/ARD/02
------------------------	--	---------------

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. Opcja ta ma zastosowanie do prostej wersji wyświetlacza, dzięki czemu można z niej korzystać.

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

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

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.

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

```
# define CORRECT_CHAUFFE 1.32 // współczynnik grzania w funkcji prędkości.
```

Aby aktywować kontrolę grzania, wstawiasz „1” zamiast zera.

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

Dla wartości współczynnika korekcji grzania, wstawiasz obliczoną wartość.

Ponownie przesyłasz zmieniony program [ad_fc_uno-4.ino](#) do arduino

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 prądu, ale gdy wycinasz krawędź natarcia w jednym przejściu, zobaczysz znaczące zmiany na amperomierzu.

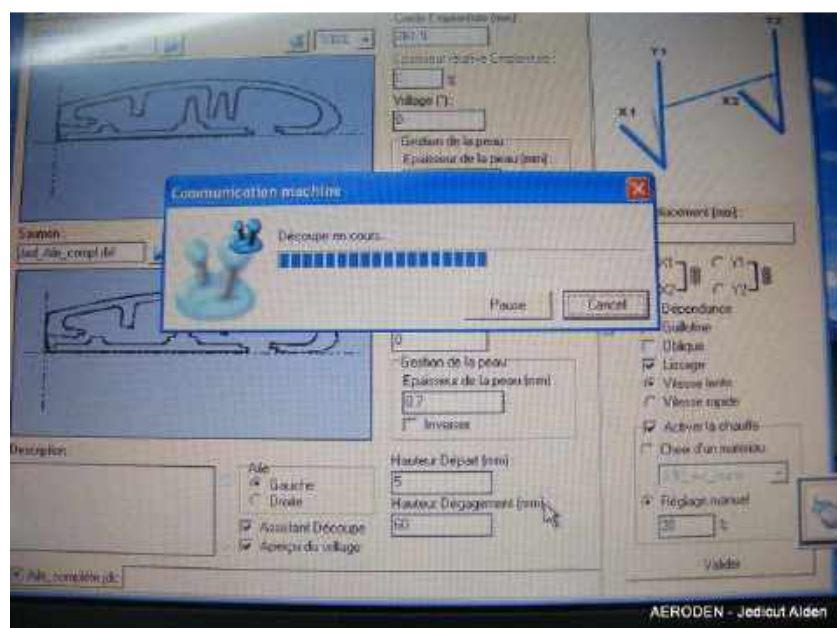
Aby przetestować w pełni prawidłowość ustawień, wykonaj cięcie za pomocą pliku "[Profil test vitesse chauffe.jdc](#)" z folderu "[JA Paramétrage chauffe](#)"

Oto co się dzieje:



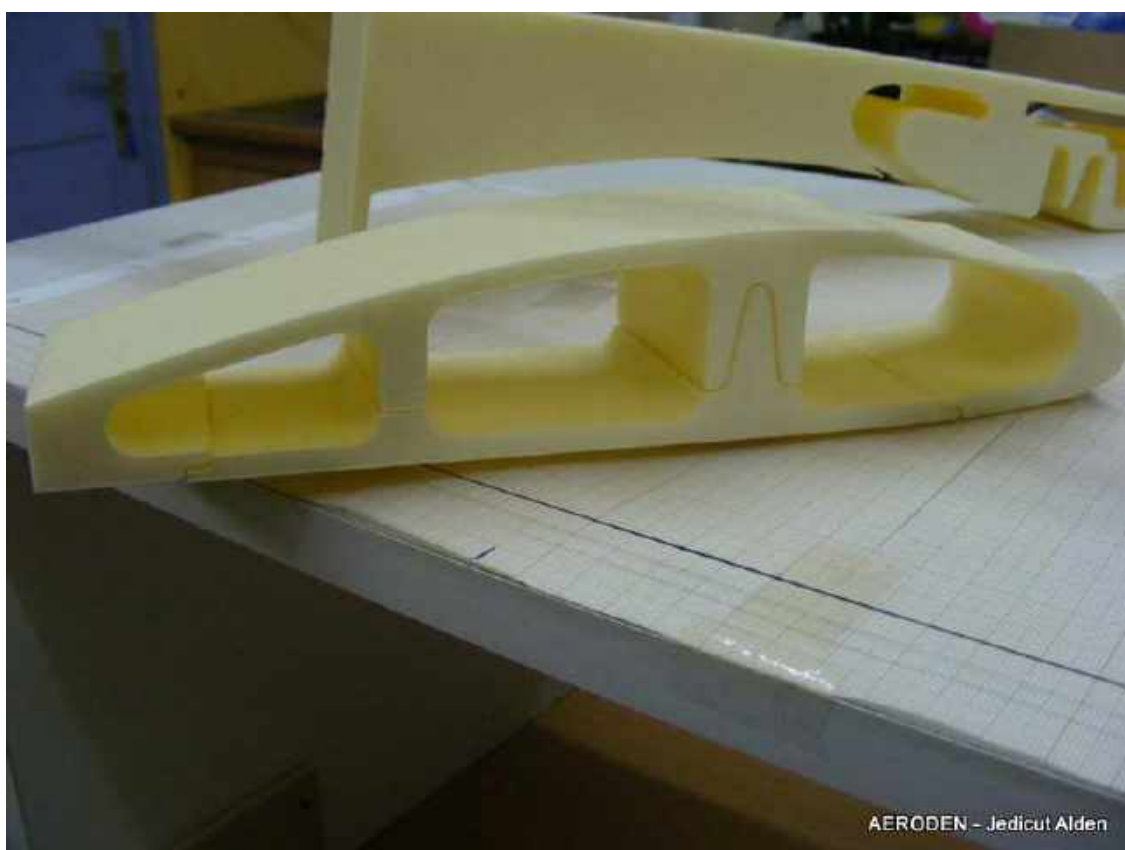
Z moją starą maszyną i elektroniką MM2001 Wynik jest bardzo dobry.

Podsumowując testy tutaj, znajduje się plik cięcia: "[Aile_complète.jdc](#)". Uruchom go, i wykonaj cięcie.





Wygląda to tak, a ponieważ to tylko testy, brakuje krótkiego fragmentu.



Z moją starą maszyną i elektroniką MM2001 Wynik jest bardzo dobry. Jeśli Twój wynik jest podobny, to ustawienia Twojej maszyny są poprawne .

Wersja 1.1 z 17.04.2019

Dodano ostrzeżenie na stronie 2 dot. identyczności wszystkich osi