

Opis płytki Arduino Mega 2560	Strona 2
Opis płytki Ramps 1.4	Strona 8
Opis sterownika A4988	Strona 10
Budowa sterownika Pololu A4899.	Strona 12
Wyjaśnienie obliczeń i regulacji prądu silnika	Strona 13
Opis wyświetlacza LCD	Strona 15
Instalowanie Arduino IDE i Pierwsze kroki.	Strona 19
Test płyty Mega 2560	Strona 21
Test sterownika Pololu	Strona 24
Sprawdzanie i testowanie płyty Ramps 1.5	Strona 25
Opis i ustawienia sterownika DRV8825	Strona 28

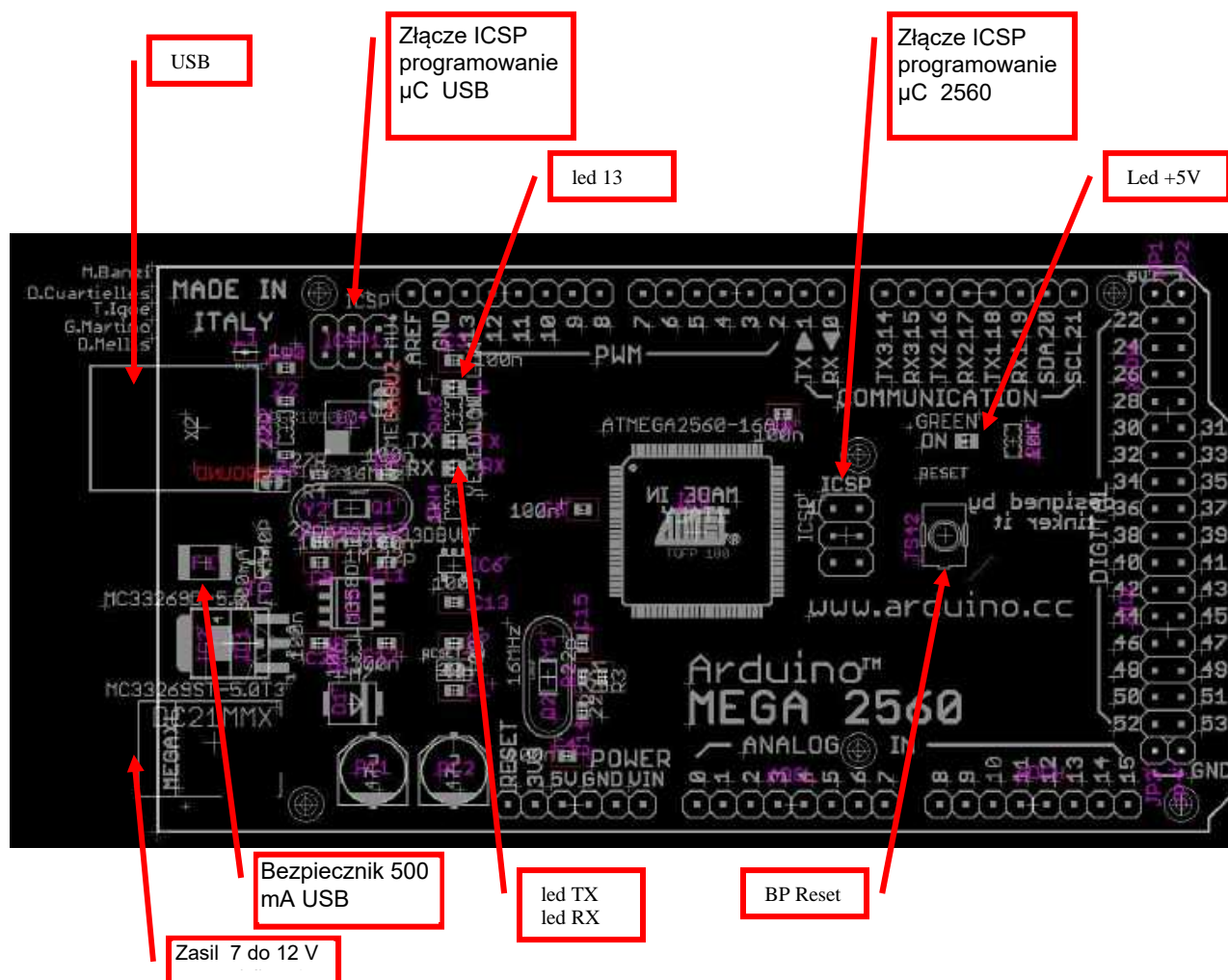
Ten dokument jest syntezą moich badań, moich notatek i doświadczeń związanych z używaniem Arduino mega 2560 i Ramps 1.4. Chciałbym podziękować wszystkim tym, którzy napisali artykuły i szkice, które pomogły mi poczynić postępy.

Ta synteza z pewnością pozwoli tym, którzy zaczynają szybko się rozwijać, unikając spędzania czasu na żmudnym poszukiwaniu informacji rozproszonych po internecie.

Zalecam przeczytanie tego dokumentu w całości, zanim przystąpisz do pracy, uratuje cię to od awarii. Poświęć czas i środki ostrożności, aby uniknąć zwarcia na 5V, regulatory płyty mega 2560 nie są chronione.

Życzę udanego opanowania tych płyt i sukcesu w twoich projektach.

## Płyta Arduino Mega 2560



Mikrokontroler Atmega 2560; Częstotliwość 16 MHz

Napięcie pracy 5 V,

54 wyjścia cyfrowe, w tym 15 wyjść zgodnych z PWM

16 wejść analogowych (mogą być używane również jako wejścia cyfrowe)

Maksymalny prąd na pin 40 mA.

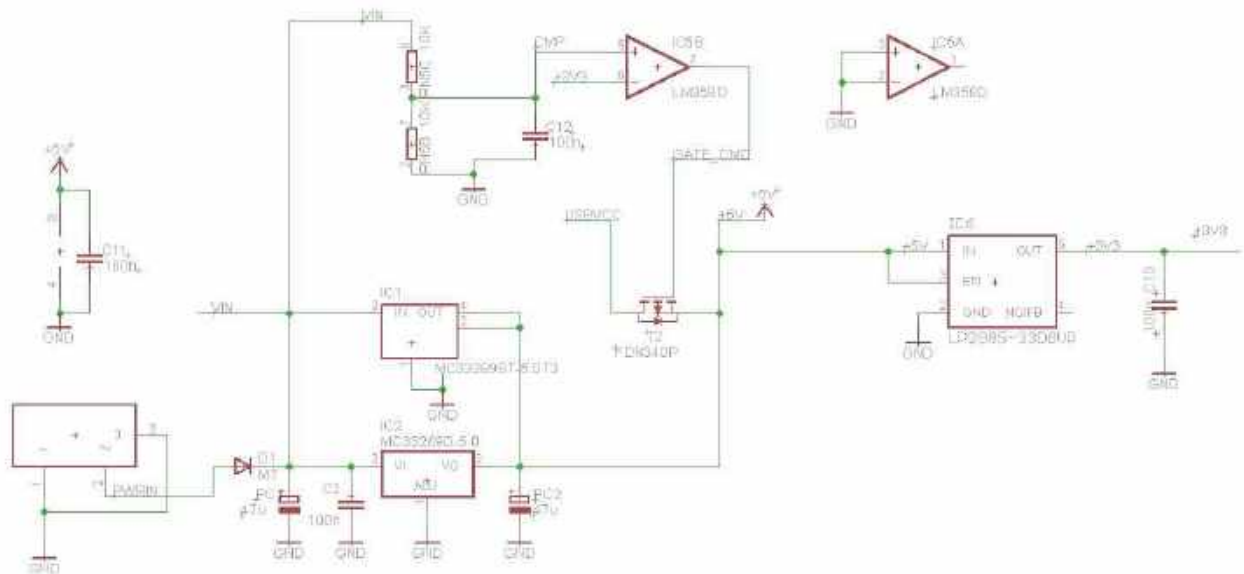
Pamięć: 256 KB Flash, 8 KB SRAM, 4 KB EEPROM

8 KB boot loader w pamięci Flash

2 złącza ICSP, są to złącza do bezpośredniego programowania mikrokontrolerów (bootloader) głównego i USB. Przy zakupie karty, bootloader 2560 jest już zaprogramowany. Mikrokontroler łączy USB jest również zaprogramowany, więc nie będziemy używać tych złączy podczas normalnej pracy.

LED 13 jest podłączony bezpośrednio do wyjścia cyfrowego 13, LED + 5 V to obecność 5V.

Diody TX i RX pokazują ruch USB na wyjściu mikrokontrolera USB

**Specyfika zasilaczy:**

Istnieją 3 sposoby zasilania Arduino:

Zasilacz USB VCC eliminuje zasilanie podczas programowania, jeśli sterowniki mocy nie są podłączone do karty.

Na płytce Arduino znajduje się złącze zasilania 2,1 mm (7 do 12V =), wyjście złącza przechodzi do regulatorów 5 V przez diodę.

Trochę wyżej niż dioda D1. Jest obwód o nazwie VIN, znajduje się on na złączu zasilaczy, jest on do połączenie z inną kartą, w szczególności Ramps 1.4.

Za regulatorami 5 V, istnieje regulator 3.3 V, który umożliwia podłączenie urządzeń pomocniczych wymagających zasilania z 3.3 V jak karta SD.

Do czego służą inne komponenty?

Istnieje wzmacniacz operacyjny LM358 zastosowany jako komparator pomiędzy napięciem 3.3v a mostkiem dzielącym  $\frac{1}{2}$  napięcia VIN, wyjście komparatora steruje tranzystorem MOS na 5 V linii USB.

Jest to automatyczne przełączanie między napięciem USB a zasilaniem z VIN.

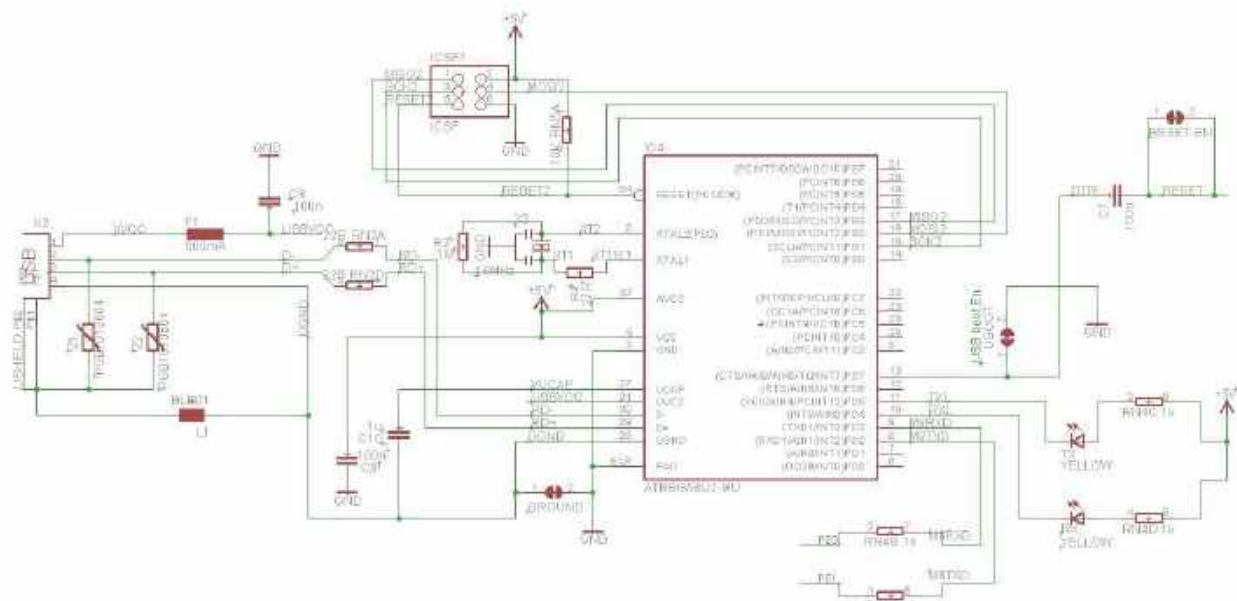
Działanie:

Bez napięcia VIN bramka tranzystora Mos jest na 0 V, ponieważ 3,3 V znajduje się na wejściu odwracającym wzmacniacza. Tranzystor przewodzi i zasila elektronikę. Kiedy jeden z zasilaczy VIN jest połączony przez mostek dzielący, napięcie o wartości 6 V jest podawane na nieodwracające wejście wzmacniacza, wyjście następnie przechodzi do +5 V i blokuje tranzystor MOS. , zasilacz USB VCC zostaje odłączony.

Podsumowując, płytka może być zasilana przez USB, gniazdo zasilania płytki Mega lub zasilana z płytki Ramps, gdy ta jest używana.

Blok pinów zasilania: Pin VIN jest pinem, który akceptuje zasilanie od 7 do 12 V pochodzące z innej karty jak np. Ramps 1.4, 2-piny GND tworzą połączenie masy, pin 5 V wychodzi z regulatorów 5V płytki Mega i pozwala zasilić inną płytkę. Pin 3.3 V wychodzi z regulatora 3v3 płytki Mega i pozwala zasilić inną płytkę napięciem 3v3. Pin Reset pozwala na innej płytce, która jest podłączona, mieć dodatkowy przycisk Reset, ponieważ Reset płytki Mega staje się niedostępny.



**Połączenie USB:**

Po lewej stronie znajduje się gniazdo USB z bezpiecznikiem 500 mA F1, który zabezpiecza przed przeciążeniem. Ten bezpiecznik jest komponentem SMD. Komponenty Z1, Z2, L1 tworzą filtr odcinający dla tłumienia zakłóceń zewnętrznych przez kabel. Rezystory RN2A i RN2B o wartości 22 omów ograniczają wpływy między USB  $\mu C$  8U2 płytki Mega a wyjściem USB komputera PC lub koncentratora USB. W niektórych zestawach, które można znaleźć w internecie, rezystory te są często pomijane, co często prowadzi do losowego działania portu USB, stąd potrzeba posiadania krótkich kabli USB. Zalecam umieszczenie tych 22 lub 27 omowych rezystorów w tym obwodzie, jeśli pracujesz za pomocą USB.

Sygnał USB wchodzi do  $\mu C$  8U2 przez D- i D+, jest przetwarzany i pojawia się na portach PD2, PD3 i jest kierowany do portów PE0 i PE1 z  $\mu C$  2650 zwanych RX0 i TX0 przez rezystory 1K RN4A i RN4B (dla uniknięcia wpływów między 2  $\mu C$  z jednej strony a wejściami cyfrowymi 0 i 1  $\mu C$  2560 z drugiej).

Diody TX i RX wskazują ruch w linii USB

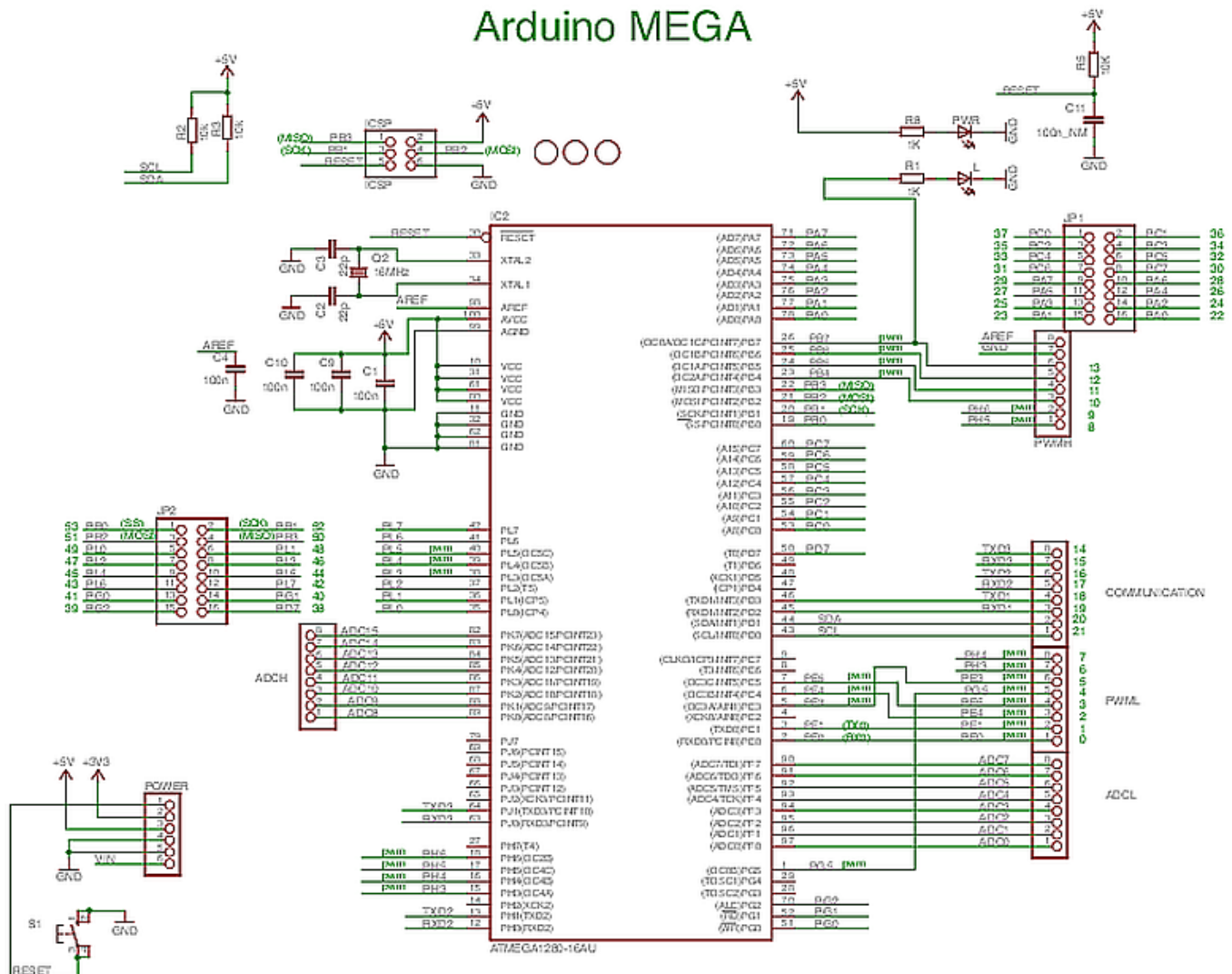
Dioda TX wskazuje, że mega-karta wysyła informacje do komputera.

Dioda RX wskazuje, że mega karta odbiera informacje z komputera.

Złącze ISP1 umożliwia programowanie  $\mu C$  8U2 w czasie produkcji karty.



## Arduino MEGA



Powyżej, po lewej stronie, złącze ICSP do wgrywania Bootloadera w procesie produkcji płytki.

Liczby na krawędzi prostokąta ograniczającego  $\mu C$  są numerami pinów obudowy.

Na przykład: w prawym górnym rogu szczegóły dotyczące  $\mu C$  poniżej

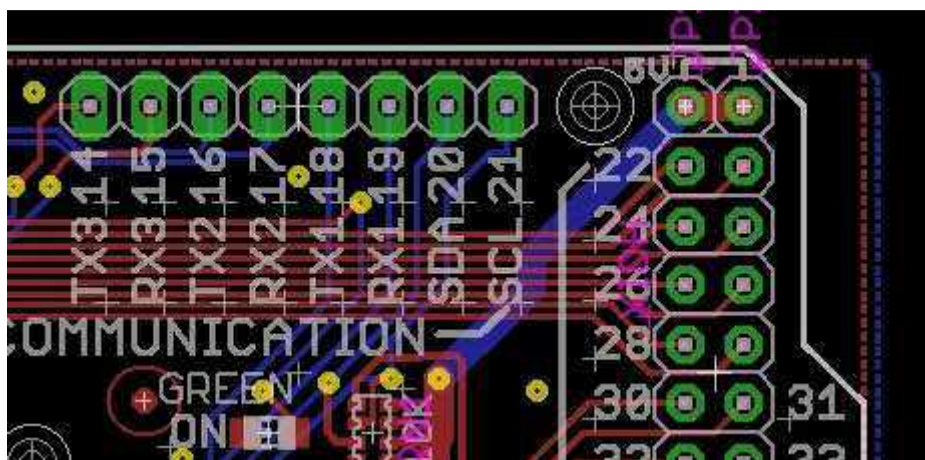


Jeśli rozpoczniemy od portu A mamy 8 bitów od PA0 do PA7, port ten jest wyspecjalizowany do operacji AD (AD0-AD7) port ten może być używany dla adresów danych do zapisu danych w zewnętrznej pamięci. Nie należy mylić z ADC, który jest funkcją Analog Digital Converter.

Na prawo od PA 0 znajduje się numer 78, który jest numerem pinu obudowy.

Następnie znajdujemy PA0, to jest nazwa połączenia (tutaj nadano mu nazwę portu, ale mógł to być na przykład "toto"). Czy linia jest podłączona? Nie, aby schemat był czytelny. Ciągłość przewodów nie jest zachowywana podczas rysowania schematu, trzeba szukać innego przewodu w powietrzu, która ma tę samą nazwę (jest to nazwa "net") w obwodach drukowanych. Ten przewód znajduje się na prawym złączu XIOH na pinie 16. Jest to odpowiednik połączenia znajdującego się pomiędzy pinem 78  $\mu C$  a pinem 16 złącza XIOH. Tę nazwę można oczywiście znaleźć w innych częściach schematu. Na prawo od linii PA0 na złączu XIOH jest numer 22, który jest numerem wejścia/wyjścia płytki Mega.

Opisy na płycie pozwalają na łatwą identyfikację pinów  $\mu C$  np. PA0 odpowiada pinowi 22 na podwójnym złączu po prawej na zdjęciu. Numery pinów 23, 25, 27, 29 nie są opisane, brak miejsca na płycie



Piny "wejść cyfrowych" w programach są również numerowane od 0 do 53 na płycie.

Piny analogowe na płycie mają numery od 0 do 15 i są oznaczone jako "ANALOG IN". W programach są one nazywane od A0 do A15, nawet jeśli są używane jako wejścia cyfrowe.

Wszystko to jest dość skomplikowane, ale istnieje wyjaśnienie:

Najczęściej używanym językiem programowania  $\mu C$  jest C/C++, narzędziem programistycznym dla  $\mu C$  Atmel jest "Atmel Studio", jeśli konieczne jest zdefiniowanie na przykład danych wyjściowych z rejestrami używającymi PA0 ..... jest to bardzo żmudne i trzeba użyć arkusza danych  $\mu C$ .

Arduino urodził się w Anglii, aby uczyć programowania nastolatków. Następnie inżynierowie ułatwili zadanie, rozwijając biblioteki, aby użytkownik Arduino nie musiał znać wewnętrznych rejestrów  $\mu C$ . Użytkownik Arduino użyje pinów karty, a nie portów  $\mu C$ . Z drugiej strony językiem jest "C", i w rezultacie rozwój programu i programowanie mikroprocesora  $\mu C$  odbywa się za pomocą narzędzia "IDE Arduino",



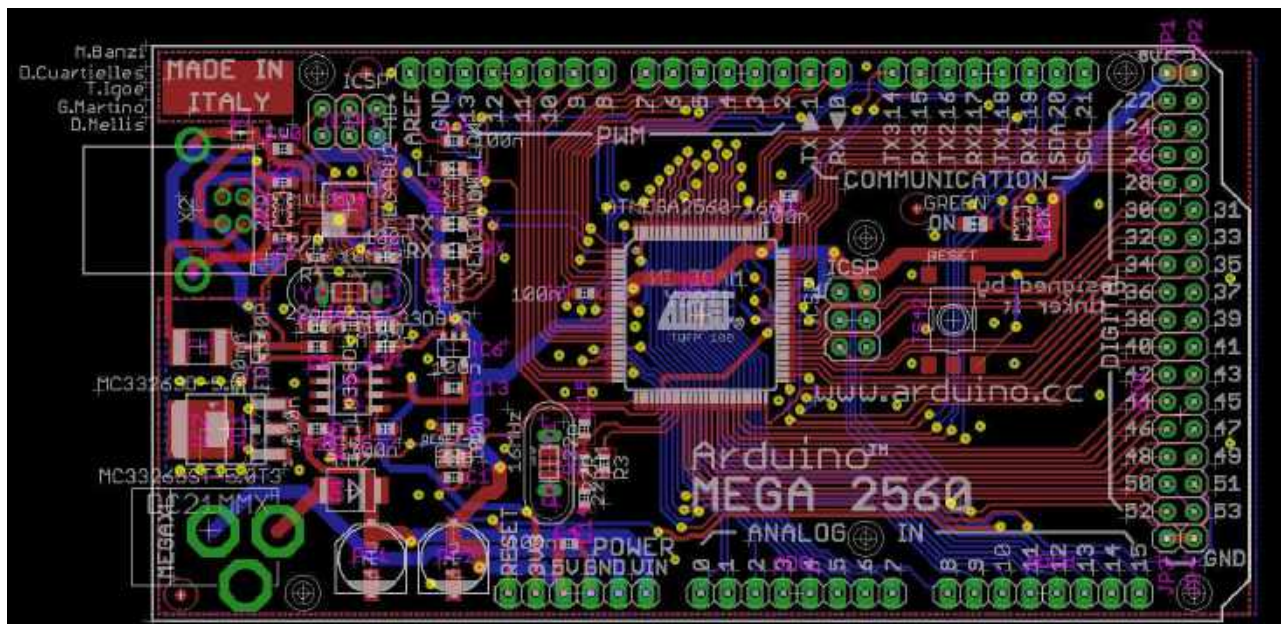
Narzędzie to jest bezpłatne, zawiera wszystkie biblioteki, również możemy dodawać biblioteki; kompilator, który przekształca język C w plik .hex dla transferu do Arduino. Obsługuje również błędy składniowe.

Oto przykład: Miganie diodą Led 13 na płytce Mega co 2 sekundy.

<pre>int ledPin = 13 // On declare la Pin 13 pour la Led  void setup() {   // initialise en sortie.   pinMode(ledPin, OUTPUT); }  // boucle infinie void loop() {   digitalWrite(ledPin, HIGH); // allume la led   delay(1000);               // attend une seconde   digitalWrite(ledPin, LOW);  // éteint la led   delay(1000);               // attend une seconde }</pre>	<pre>Int ledPin = 13 // Deklarujemy Pin 13 dla diody.  void setup() {   //Uruchamia wyjście.   pinMode(ledPin, OUTPUT); }  //Nieskończona pętla void loop() {   digitalWrite(ledPin, HIGH); // włączanie diody LED   delay(1000);               // poczekaj sekundę   digitalWrite(ledPin, LOW); // wyłącza leda   delay(1000);               // poczekaj sekundę }</pre>
---	---

Wszystkie programy Arduino mają funkcję „setup” i „loop”. W tym przykładzie widzimy, że nie korzystamy z rejestrów.

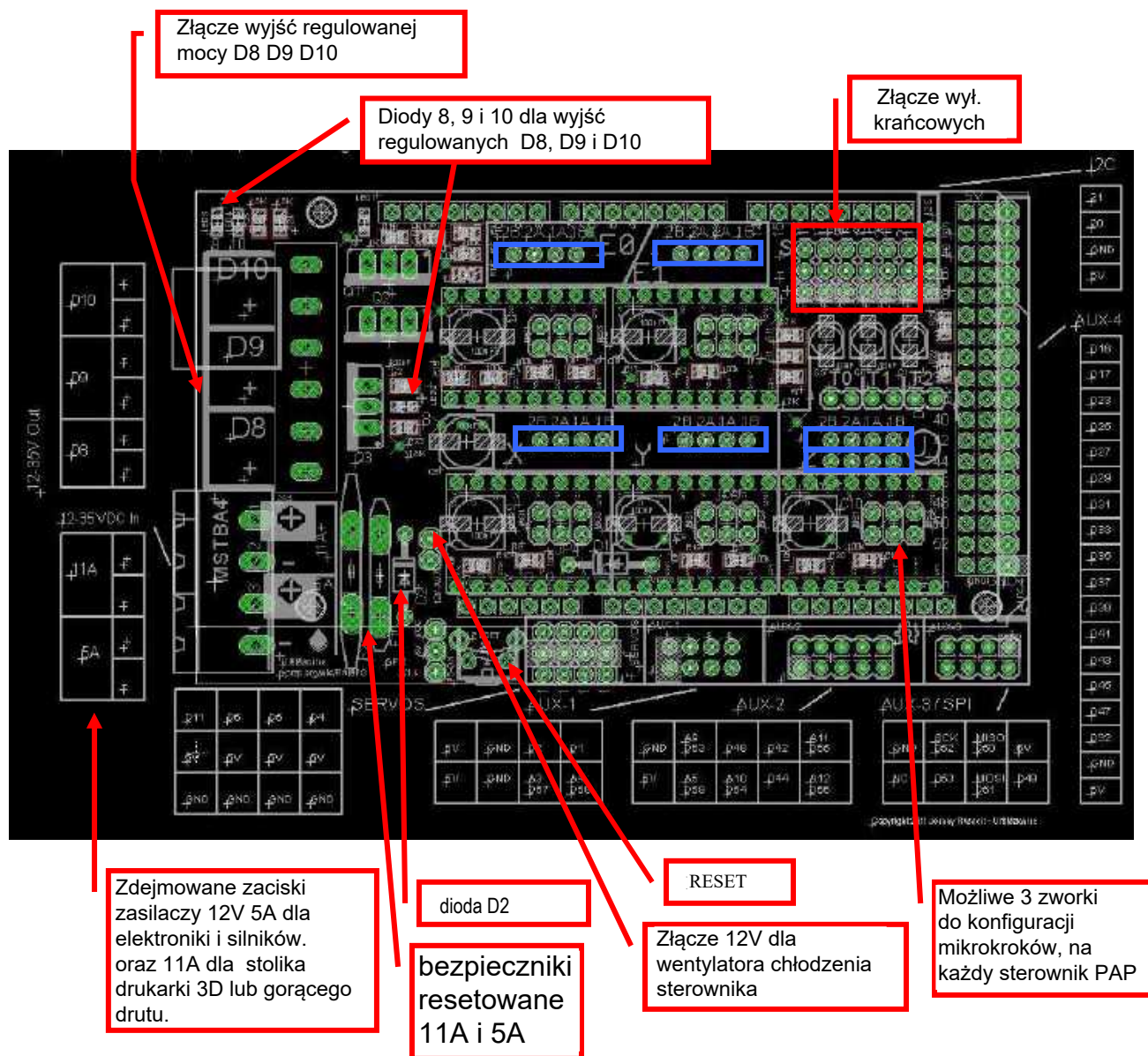
Jest to naprawdę bardzo praktyczne, z drugiej strony działa wolniej, ponieważ korzysta z bibliotek. W przypadku niektórych skomplikowanych aplikacji należy zachować ostrożność, ponieważ niektóre biblioteki nie są ze sobą kompatybilne..



Podane przeze mnie informacje pozwolą ci korzystać z płytek Arduino Mega, ładować istniejący program i analizować anomalie operacyjne.

To streszczenie jest kopią moich notatek, od kiedy po raz pierwszy zacząłem używać płytek Arduino. W internecie jest wiele rzeczy, ale nie znalazłem dokumentu podsumowującego, więc to napisałem. Są bardzo dobre książki, które pozwalają uczyć się krok po kroku, ale są drogie. Na przykład książka ARDUINO (37 EUR). Dla entuzjastów CNC nie sądzę, aby inwestowanie w książki było konieczne, aby stworzyć ich maszynę..

## Karta Ramps 1.4



Płytkę Ramps 1.4 jest nakładką na płytkę Arduino Mega 2560 przeznaczoną do sterowania drukarką 3D, pozwala kontrolować 6 bipolarnych silników PAP (krokowych).

**W przypadku drukarek 3D:** 1 silnik X, 1 silnik Y, 2 silniki równolegle dla Z, Ekstruder 1 na E0, Ekstruder 2 na E1.

Wyjście mocy regulowane: D8 ogrzewanie stolika, D9 ogrzewanie ekstrudera 1, D10 ogrzewanie ekstrudera 2 lub zasilanie wentylatora chłodzącego wylot dyszy wytłaczarki.

Złącze wyłączników krańcowych.

Wyświetlacz LCD 4 x 20, obrotowy enkoder do nawigacji w menu, czytnik kart SD do korzystania z programów do drukowania bez komputera.

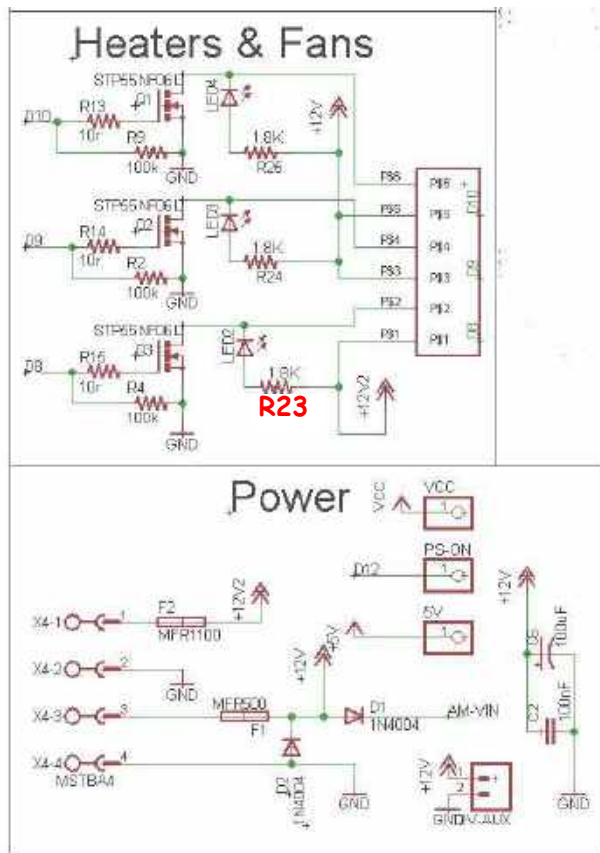
**Do cięcia gorącym drutem:** 4 silniki krokowe na E0, E1, X, Y. Wyjścia mocy: D8 drut łuku tnącego, D9 przecinarka elektryczna, D10 przełącznik do nagrzewnicy łukowej.

Korzystanie z wyłączników krańcowych. Wykorzystanie wyświetlacza do informacji o mocy grzania drutu i prędkości cięcia.

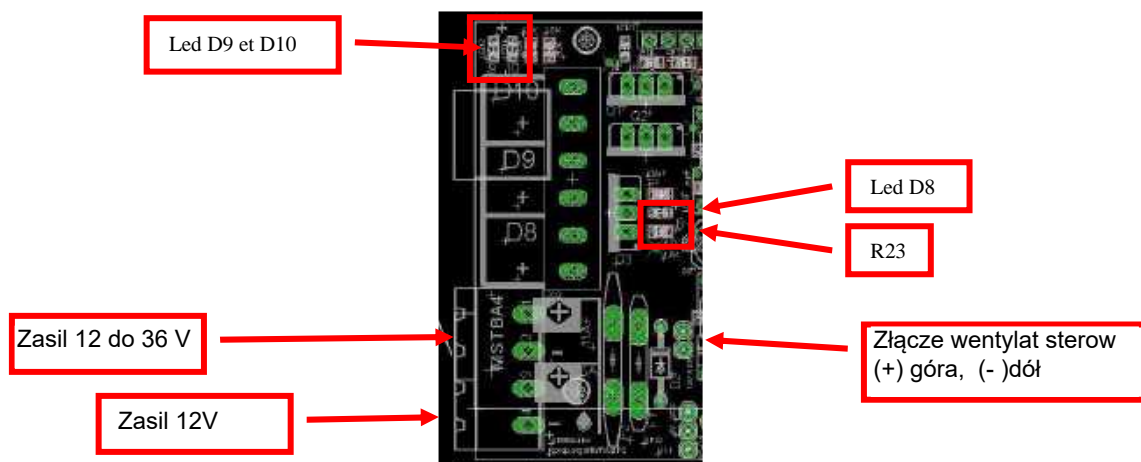


## Zasilanie i wyjścia zasilania:

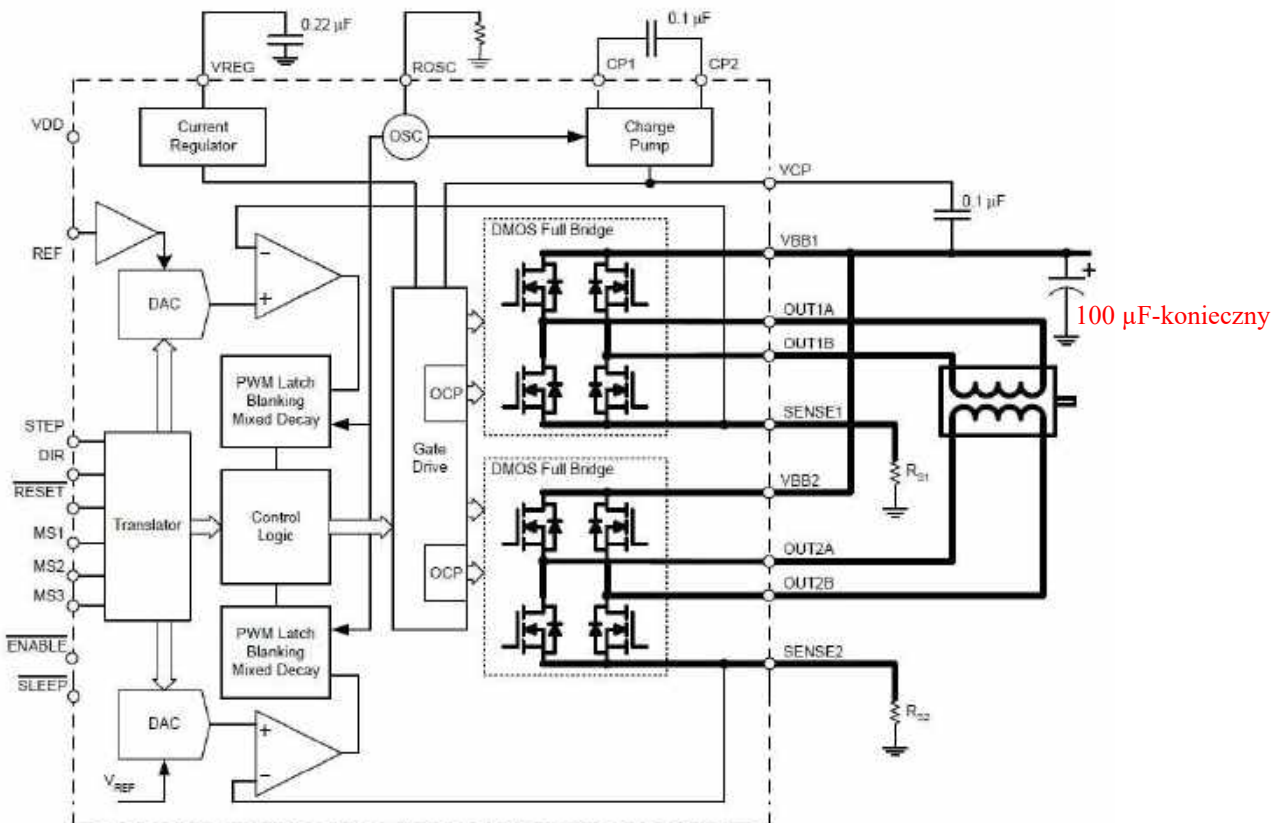
Przy normalnym użytkowaniu zasilacz 12V (7 do 16 V) jest podłączony do X4-3, X4-4 przechodzi przez resetowalny bezpiecznik F1-5 A (są to półprzewodniki, których rezystancja wzrasta, gdy przepływający przez nie prąd przekracza swoją wartość nominalną), gdy prąd wraca do normy, zasilanie jest przywracane. Po tym bezpieczniku 12V jest dystrybuowane do sterowników mocy silnika PAP i złącza V-Aux dla wentylatora chłodzącego sterowników, to 12 V można znaleźć na zaciskach zasilania regulowanego D9 i D10. Następnie przez diodę D1 przechodzi do pinu VIN, który zasilą płytke Arduino Mega. Spójrzmy na drugi zasilacz w X4-1 X4-2, ten zasilacz przyjmuje napięcie od 12 do 36V które przechodzi przez resetowalny bezpiecznik F2 o wartości 11 A, oznaczone 12V2 i przechodzi do bloku zacisków zasilania regulowanego D8. Napięcia regulowane są modulowane PWM przez Tranzystory MOS Q1, Q2, Q3, sygnały MLI lub PWM pochodzą z cyfrowych pinów 8, 9, 10 płytki Mega, a na płytce Ramps są nazywane D8, D9, D10. Diody LED wskazują modulację mocy.



**Uwaga:** jeśli używasz napięcia zasilania większego niż 12 V dla D8, konieczne jest usunięcie rezystora **R23**.



## Sterowniki A4988:



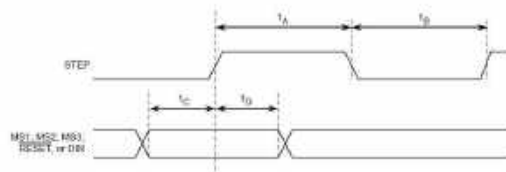
A4988 jest kompletnym sterownikiem silnika krokowego z wbudowanym regulatorem prądu. Przeznaczony jest do współpracy z dwubiegunowymi silnikami krokowymi w trybach pełnego, pół-, ćwierć-, jedna ósma i jedna szesnasta kroku, o napięciu wyjściowym 35 V i prądzie maks.  $\pm 2$  A.

Część pogrubiona przedstawia zasilanie dla obu uzwojeń silnika krokowego. Każde uzwojenie jest zasilane przez tranzystorowy (DMOS) mostek H. Silniki są zasilane przez VBB1 i VBB2 odsprężone przez kondensator **100 µF, kondensator ten jest bardzo ważny**. Powrót do GND jest podawany przez rezystory do pomiaru prądu. Odczyt z pomiaru prądu jest wartością SENSE1 dla 1A i 1B oraz SENSE2 dla 2A i 2B, wartość ta jest przesyłana do wzmacniacza operacyjnego i porównywana z Vref. To trochę bardziej skomplikowane, bo trzeba... zsynchronizować to wszystko z krokami; taka jest rola translatora i logiki kontroli. To jest po prostu konieczne do zrozumienia zasady, ufa się producentowi. Szczegółowo opiszę to dzięki czemu ustawienie prądu dostosowane do silników jest łatwe do przeprowadzenia. Wszystko, co musisz zrobić, to zmierzyć napięcie REF multimetrem. Zacisk REF jest podawany do wzmacniacza i staje się VREF, a następnie spotyka się na obu przetwornikach DAC.

Terminale VREG, ROS, CP1, CP2, VCP wymagają komponentów do podstawowej operacji. Użyjemy 3 terminali "step", "dir" i "enable barre" (odblokowanie sterownika odbywa się przez "0"), dla rozkazów kroku i kierunku konieczne jest respektowanie czasów przejścia pomiędzy impulsami,

Poniżej jest diagram.

W naszym przypadku nie będziemy brać pod uwagę wyboru  $\mu$  kroków i resetowania, ponieważ są to operacje, które są wykonywane, gdy maszyna nie pracuje..



Time Duration	Symbol	Typ.	Unit
STEP minimum, HIGH pulse width	$t_A$	1	$\mu s$
STEP minimum, LOW pulse width	$t_B$	1	$\mu s$
Setup time, input change to STEP	$t_C$	200	ns
Hold time, input change to STEP	$t_D$	200	ns

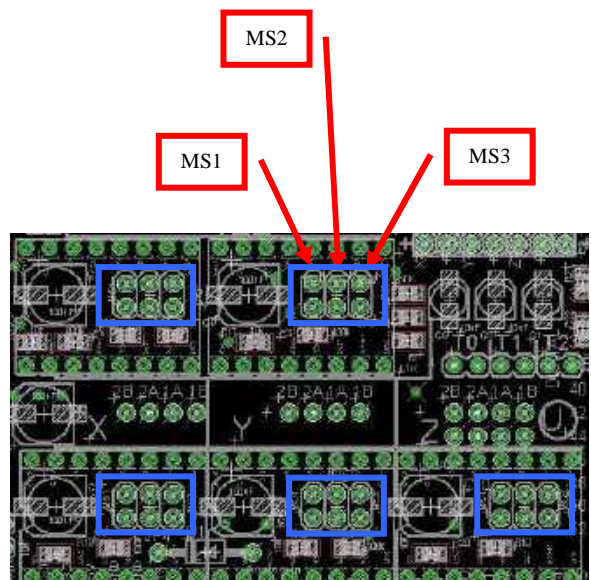
Widzimy, że impuls "kroku" musi nadejść po zmianie rozkazu "DIR" na " $t_C$ " > 200 ns. Czas impulsu krokowego " $t_A$ " musi być większy niż 1  $\mu s$ ; a czas " $t_B$ " pomiędzy 2 impulsami krokowymi musi być większy niż 1  $\mu s$ . Zmiana kierunku nie może nastąpić przed czasem " $t_D$ " > 200ns. W większości przypadków znajdujemy się w tych zakresach, ale trzeba być czujnym, jeśli zauważy się buczenie silników lub utratę kroków.

Teraz przyjrzyjmy się wyborowi mikrokroków ( $\mu$ stepów), mamy do dyspozycji 3 wejścia MS1. MS2 MS3

Table 1. Microstepping Resolution Truth Table

MS1	MS2	MS3	Microstep Resolution	Excitation Mode
L	L	L	Full Step	2 Phase
H	L	L	Half Step	1-2 Phase
L	H	L	Quarter Step	W1-2 Phase
H	H	L	Eighth Step	2W1-2 Phase
H	H	H	Sixteenth Step	4W1-2 Phase

Pięć sterowników jest konfigurowalnych w  $\mu$ krokach, na zdjęciu znajdują się złącza pozwalające na ustawienie zworek zgodnie z wybranym  $\mu$ krokiem. Są pod sterownikami "Pololu". Bez zworki dostaniemy L,L,L to pełny krok, przez umieszczenie zworki na MS1 wybieramy  $\frac{1}{2}$  kroku, i tak dalej w/g tablicy.

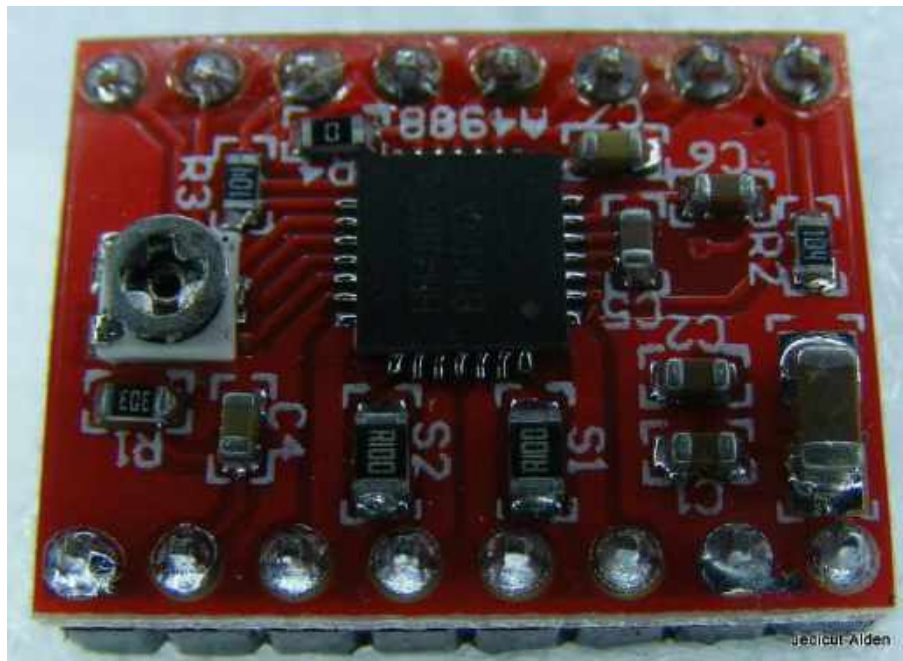


Polecenia RESET i SLEEP nie są używane, zostają ustawione na "1"

Sterowniki te są montowane z kilkoma niezbędnymi elementami na małej płytce (Pololu), którą podłącza się do płyty Ramps 1.4 lub innej. Obwód A4899 musi być zwieńczony **radiatorem**, dla prądów  $I < 1A$ . Zaś dla prądów,  $I > 1A$  musi być **wymuszone chłodzenie wszystkich radiatorów**.



## Opis "POLOLU" A4899:



W zależności od producenta płytek wartości rezystorów i oznaczeń są różne. Jednak schemat pozwala je odnaleźć.

Ważne jest zlokalizowanie rezystorów pomiarowych. To całkiem proste, to są te największe, na zdjęciu i oznaczone jako S1 i S2. Teraz musisz znać ich wartość, jeśli nie możesz odczytać wartości, możliwe jest zrobienie zdjęć makro pod różnymi kątami oświetlenia. Następnie odczytanie obrazów na komputerze ułatwia to odczytanie znaków. Tutaj rezystory S1 i S2 oznaczone są jako R100.

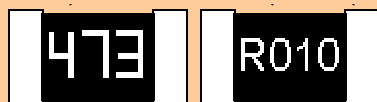
Przypomnienie o oznaczeniu oporności na SMD:

pierwsza i druga to znaczące liczby,

Trzeci to liczba zer.

Litera "R" zastępuje kropkę dziesiętną.

Przykład: 473 to 4, a następnie 7, a następnie 3 zera, czyli 47000 omów lub 47 kiloomów (47 K $\Omega$ ).



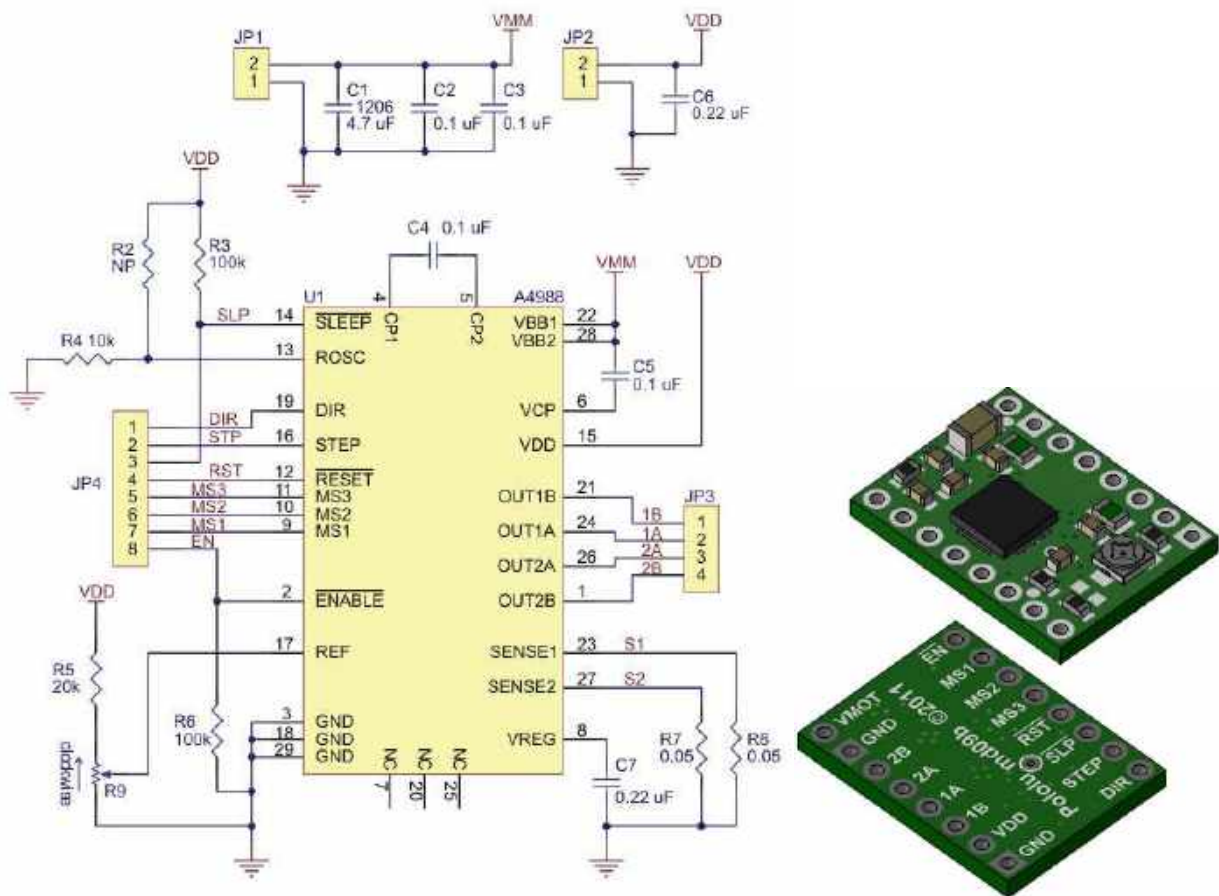
R = 47 K ohms R = 0.010 ohm

W naszym przypadku R100 = 0,1  $\Omega$

Ustawienie prądu REF jest dokonywane za pomocą potencjometru umieszczonego na płycie.

**Uwaga:** do ustawiania użyj izolowanego śrubokręta.

Teraz spójrzmy na schemat:



Patrząc na układ od dołu widać dwa rzędy pinów 8x1, a na schemacie jest tylko jeden 8x1[JP4], lewe złącze jest podzielone na 3 złącza, jedno 1x4[JP3] i dwa 1x2[JP1 i JP2].

W widoku z góry zauważysz, że pin na dole po lewej jest kwadratowy, to jest GND i to na tym pinie umieszysz końcówkę(-) multimetru do pomiaru REF końcówka(+) multimetru będzie w kontakcie z suwakiem potencjometru.

Na tym schemacie wartości rezystorów pomiaru prądu R8 i R7 wynoszą 0,05  $\Omega$

Wartość potencjometru to 10 k $\Omega$ , a rezystora szeregowego R5=20 k $\Omega$  Możliwe ustawienie REF to:  
 $5\text{ V} \times 10 / (10 + 20) = 1,66\text{ V}$ .

Używając wzoru z arkusza danych  $I_{Trip\text{ Max}} = REF / (8 \times R_s)$

Otrzymujemy:  $REF = I_{Trip\text{ Max}} \times 8 \times R_s$ .

**Przykład:** silnik obsługuje 1,5 A > ustawienie REF będzie wynosić  $1,5 \times 8 \times 0,05 = 0,6\text{ V}$

Na zdjęciu moich sterowników, rezystory S1 i S2 są oznaczone jako R100 co oznacza 0.1  $\Omega$ ,

Tras zastępujemy  $R_s$  przez 0.1 i otrzymujemy :  $V_{REF} = I_{Trip\text{ Max}} \times 8 \times 0.1$

Mój silnik pozwala na maks. prąd=1,65 A, więc ustawię dla bezpieczeństwa  $V_{REF} = 1,65 \times 0,8 = 1,32\text{ V}$ .

Podczas regulacji nie mogę uzyskać 1,32 V, ale 1,25 V, na zdjęciu Pololu widać, że rezystancja obok potencjometru posiada oznaczenie 303 to jest 30 k $\Omega$ , więc możliwe ustawienie to :

$5\text{ V} \times 10 / (10 + 30) = 1,25\text{ V}$ . To wyjaśnia wszystko.

**Uwaga: w zależności od źródła, rezystancja z potencjometru 10 kΩ może mieć różną wartość;**

W moim przypadku, maksymalny prąd będzie wynosił  $1,25 / (8 \times 0,1) = 1,56 \text{ A}$ . Jeśli nie jest to zadowalające, należy wymienić rezystor połączony szeregowo z potencjometrem.

**Zwracam uwagę, że wydajność i działanie silników krokowych PAP jest bezproblemowe, jeśli przyjrzyj się temu dokładnie. Dzisiaj jesteśmy zalewani tanim sprzętem, tym lepiej dla naszego hobby, ale musimy opanować różnice w produkcji.**

Nieco później zobaczymy, jak ustawić REF.

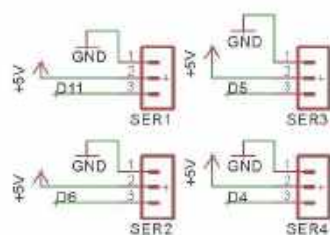
## Główne połączenia płytki Ramps 1.4:

Tutaj znajdujemy połączenie płytki Ramps 1,4 z płytą Mega 2560.

Liczby wewnątrz prostokąta oznaczają piny mega 2560. I tak liczby wejść cyfrowych są poprzedzone "D". Wskazania na kreskach odpowiadają połączeniom elementów wewnętrznych płyty Ramps 1.4.

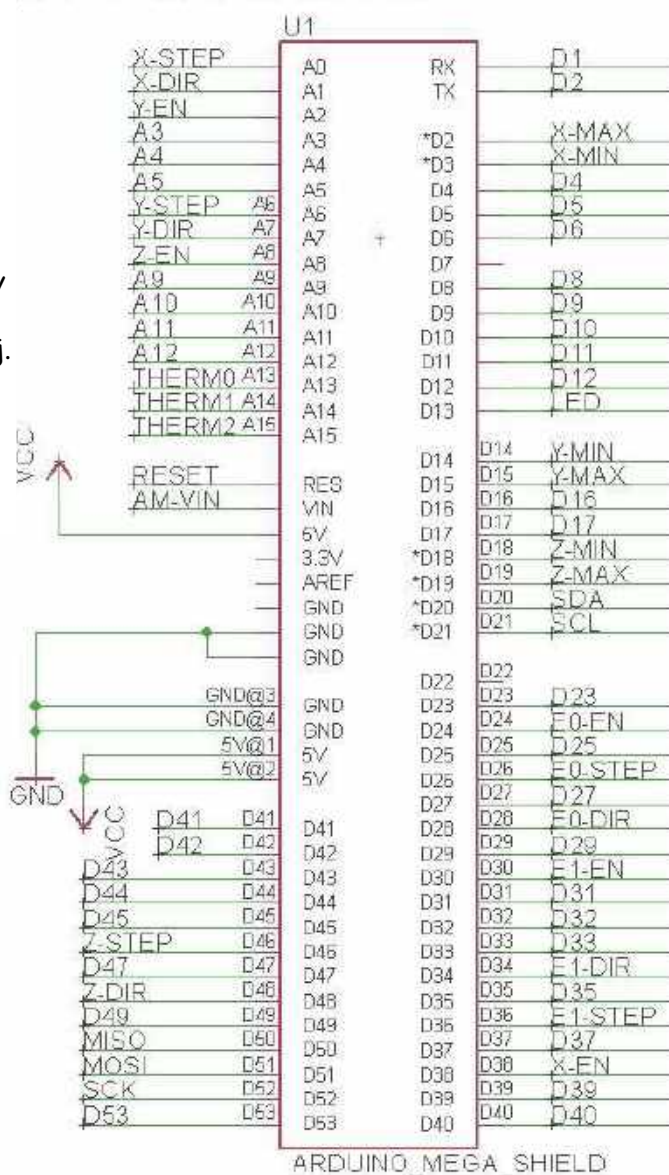
Przykład: jeśli weźmiemy A0(Mega), znajdziemy opis "X-STEP"(Ramps), więc ta linia jest połączona z A4988 opis "Step(krok)" i tak dalej. Kiedy widzimy D5 (Mega) w kierunku D5 (Ramps), odnajdujemy D5 na złączu serwa SER3

### Servos



Płytkę Ramps 1.4 może być użyta do sterowania robotem, można użyć 4 serwonapędów, istnieją uniwersalne gniazda, ale trzeba napisać program, który dostarcza sygnały do serw. Rozumie się, że można zaprogramować D5 do czegokolwiek innego, korzystając z tego konektora.

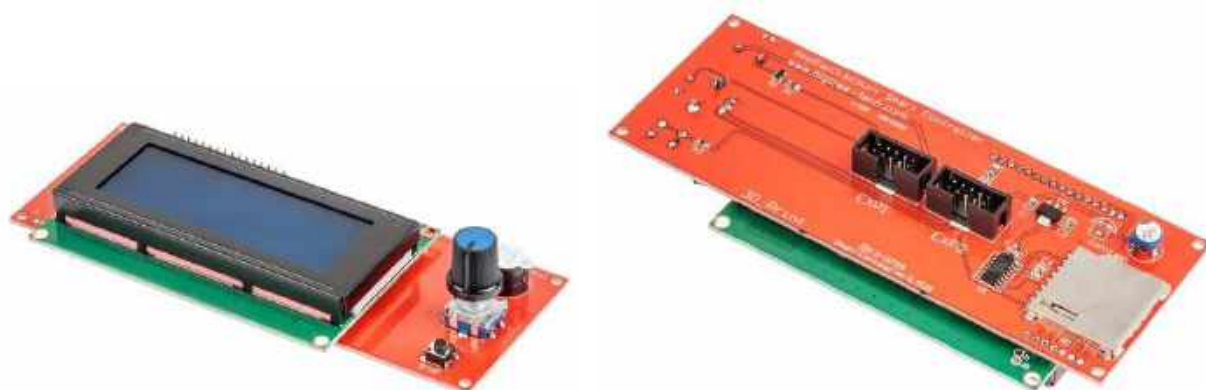
## MEGA Conn.



Informacje na tej stronie przedstawiają użycie drukarki 3D.

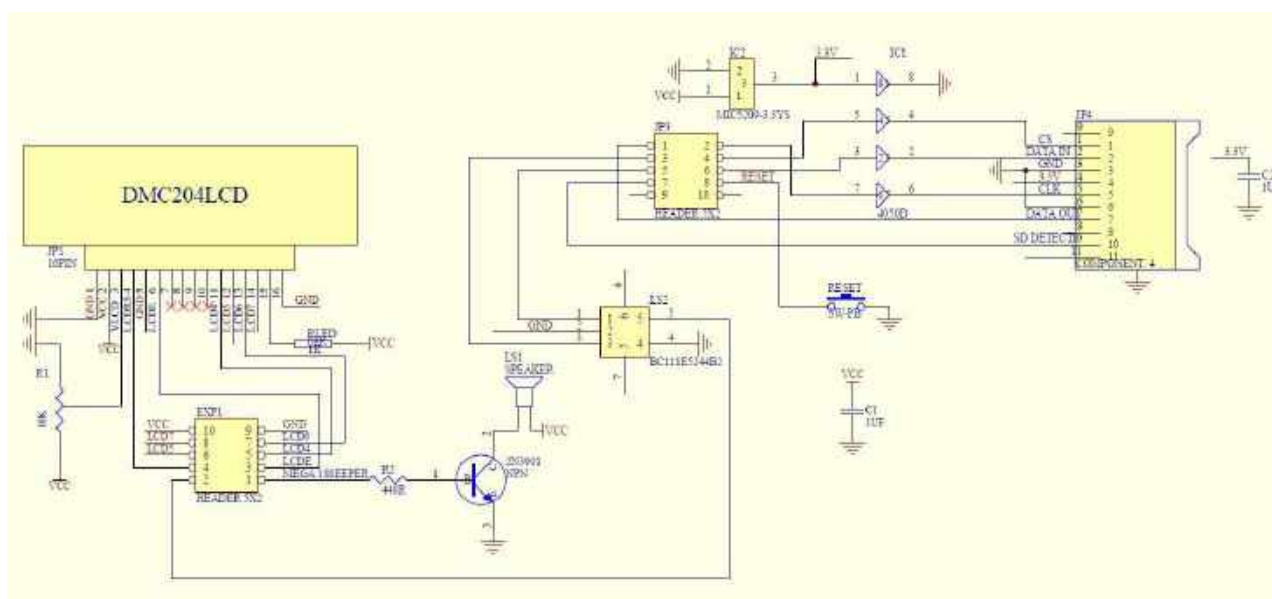
Z tą płytą opracowałem drut elektryczny CNC, zachowałem piny Pololu, piny do wyświetlania, do reszty użyłem połączeń pomiędzy Mega 2560 i Ramps 1.4 - najbardziej praktycznych.



*Płytki wyświetlacza LCD:*

Wyświetlacz LCD składa się z:

- Ekranu na którym są 4 linie z 20 znakami podświetlone na niebiesko.
- Przycisku reset, którego można użyć do zatrzymania maszyny.
- Dwudrożnego enkodera obrotowego z przyciskiem.
- Brzęczyk.
- Z tyłu czytnik kart SD, kontroler 3,3 V dla karty SD i potencjometr regulacji kontrastu wyświetlacza.

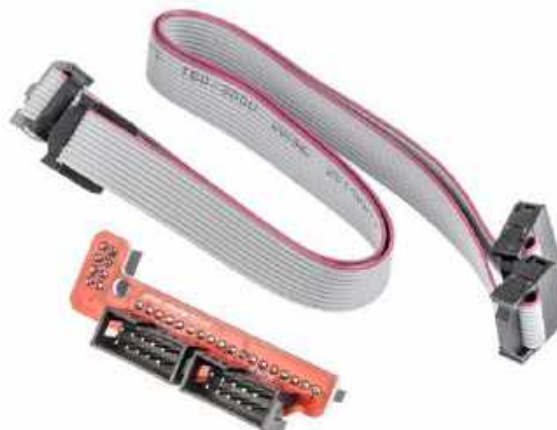


Na tym schemacie złącze oznaczone JP3 to na płytce EXP2, i razem z EXP1 są to 2 złącza 10 żyłowych kabli taśmowych.

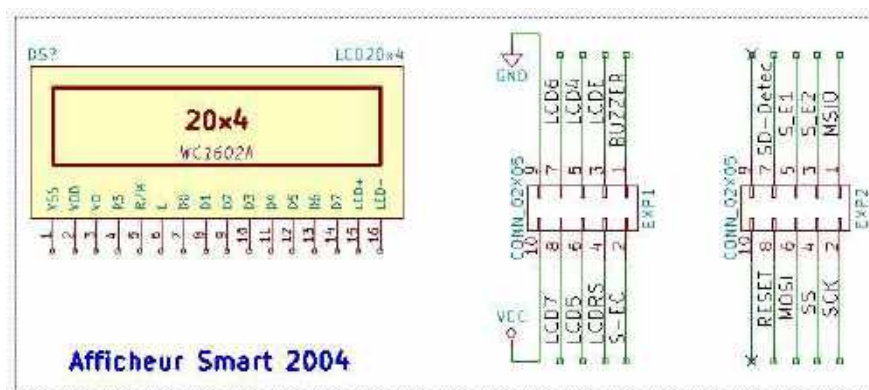
LS2 jest dwukierunkowym enkoderem obrotowym z przyciskiem BP "RESET".

BP "RESET" to przycisk, który nie jest podłączony do resetu  $\mu C$ , jest natomiast podłączony do zacisku wejścia cyfrowego D41. płytki mega 2560 ten zaś odpowiada portowi PGO. Jeśli chcesz go używać, musisz go obsłużyć w programie.

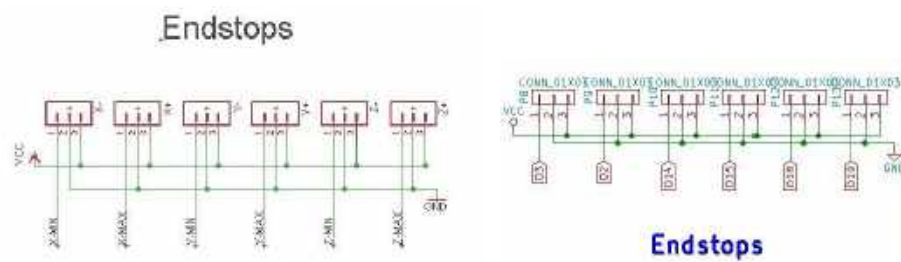
IC 4050 służy do adaptacji napięcia ( $5\text{ V} > 3,3\text{ V}$ ) dla sygnałów przesyłanych do czytnika kart SD.



Poniżej również uproszczony schemat wyświetlacza.

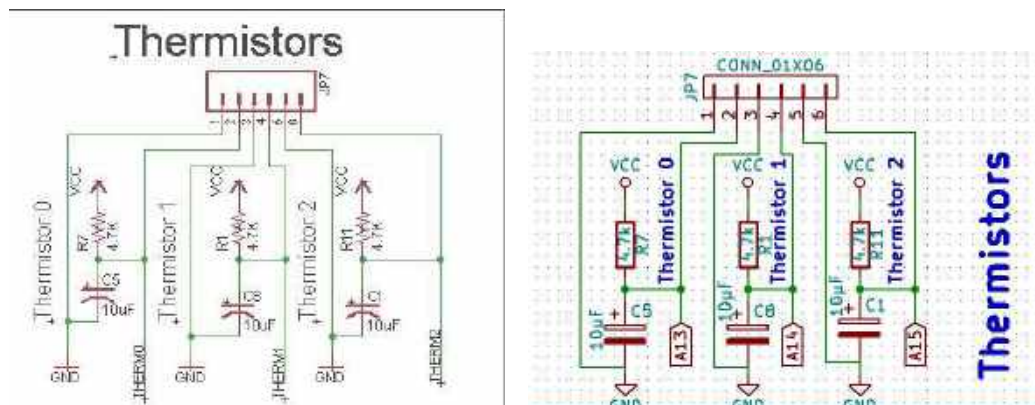


Spójrzmy na inne złącza:  
End Stop



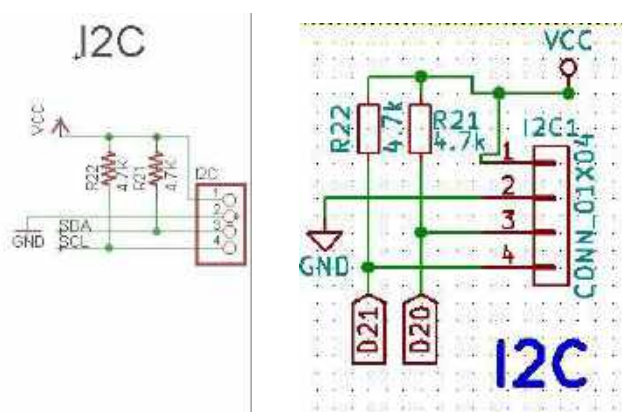
Jeśli nie używasz wyłącznika krańcowego, wyjścia cyfrowe można wykorzystać do czegoś innego.

Złącza termistorów:



Widzimy, że te wejścia do pomiaru temperatury przez termistory mają wbudowaną rezystancję, aby realizować mostek dzielący i kondensator filtrujący. Bez modyfikacji trudno jest używać tych wejść. Dlatego są dedykowane tylko dla termistorów.

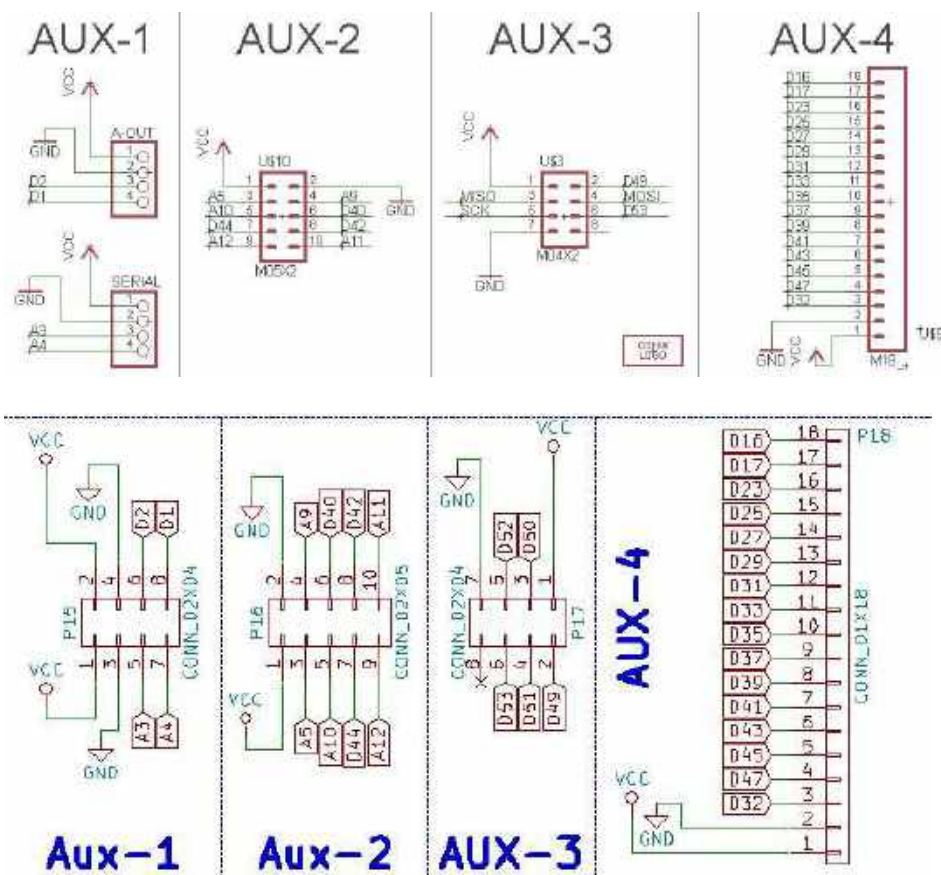
Złącze I2C:



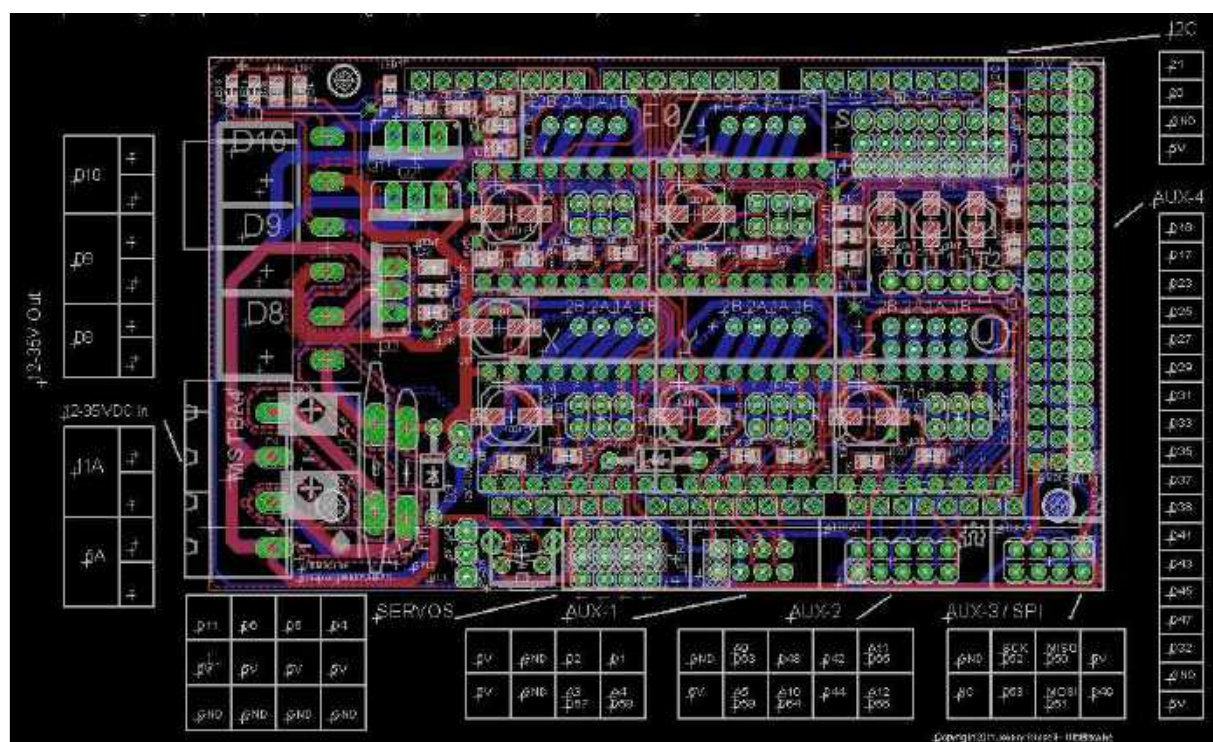
Połączenie to jest dedykowane do komunikacji I2C dla określonych obwodów, przy ograniczeniu możliwości wykorzystania ich jako wejść, są tam umieszczone rezystory podciągające (Pullup).



Złącza pomocnicze:



Złącza AUX 3 i AUX4 są przeznaczone dla wyświetlacza LCD i karty SD, jednak jeśli ich nie używasz, mogą być używane inaczej, użycie AUX1 i AUX2 jest całkowicie dowolne.



## Instalowanie IDE ARDUINO:

[Pobierz najnowszą wersję tutaj:](#)

### Download the Arduino Software



W systemie Windows wybierz Instalator Windows

### Support the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more on how your contribution will be used.](#)



Klikasz: "JUST DOWNLOAD"

Rozpocznie się pobieranie: ( 89.3 Mb)

Zainstaluj oprogramowanie, klikając bezpośrednio na [pobranym pliku](#).

Odpowiadasz jak zwykle na konwencjonalne pytania.

Pojawi się kolejne okno:



Klikasz: **"Installer"**.

Pojawi się kolejne okno tego samego rodzaju: kliknij również **"Installer"**.

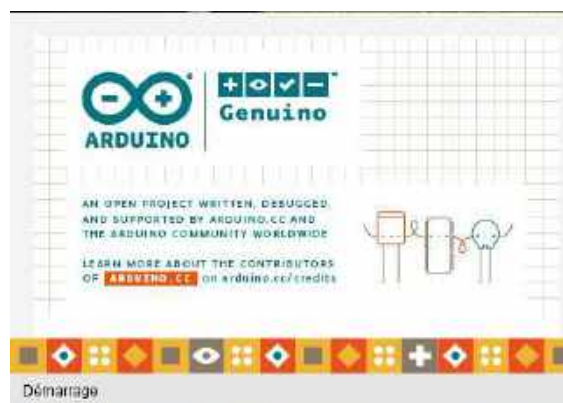
Na koniec kliknij: **"Close"** I ikona Arduino znajduje się na pulpicie.

Na koniec instalacji, spójrz do eksploratora plików, jeśli masz w **"Dokumentach"** katalog **"Arduino"** i 2 podkatalogi **"Aplikacje"** i **"Biblioteki"** (nie pamiętam czy są one tworzone automatycznie, jeśli nie, to będziesz musiał je utworzyć).

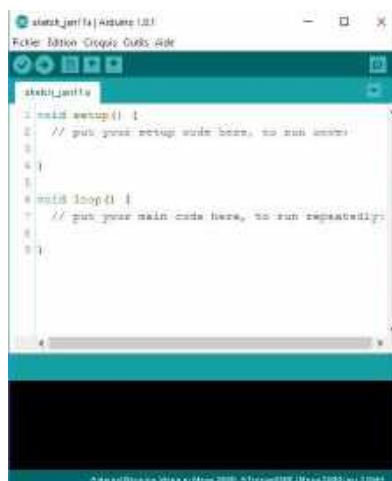
Teraz możesz przetestować Arduino IDE i swoją płytkę UNO lub Mega2560, aby sprawdzić działanie.

Podłącz kabel USB do karty Mega lub UNO.

Uruchom **"Arduino IDE"** z ikony na pulpicie.



Będziesz miał domyślny program, który się otworzy lub program, który był wcześniej używany.



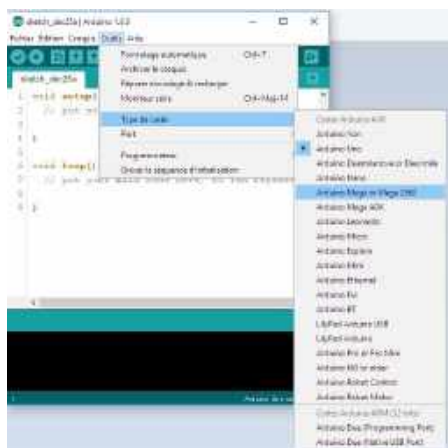


Wyszukaj wykryty "com port": przejdź do menadżera urządzeń Windows, rozwiń "com Ports", znajdź comx Arduino Mega, zanotuj numer portu.

W IDE:

W zakładce "Narzędzia, typ karty", kliknij Arduino Mega;

W zakładce "Narzędzia, Port", kliknij Com x Arduino Mega.

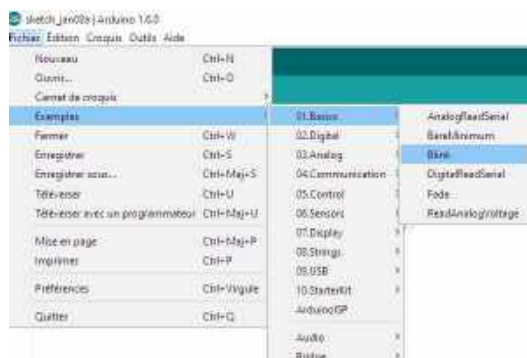


Wybierz port, który odpowiada temu, który zanotowałeś.

Przetestuj swoje karty UNO lub Mega za pomocą prostego programu do błyskania diodą LED na wyjściu 13.

Ładujemy program przykładowy "Blink" w Arduino IDE.

W zakładce "Pliki, Przykłady, 01Basics" i kliknij "Blink".



Otworzy się nowe okno z programem.

Kliknij na ikonę poziomej strzałki, program będzie się kompilował i jest ładowany do Arduino. Jeśli nie masz błędu, dioda LED 13 powinna błyskać. Twoje IDE jest dobrze zainstalowane i twoja płyta Arduino działa poprawnie.

**Zachowaj ostrożność, jeśli chcesz wypróbować inną płytę Arduino, po podłączeniu nowej karty musisz znaleźć numer portu com(x)**

**(UWAGA! na 4 płytach Arduino, które ja posiadam każda ma inny port comx).**



**Ikony poleceń ogólnych:**

- 1 od lewej: Zapisz i skompiluj do weryfikacji.
- 2 Zapisywanie, kompilowanie i wgrywanie na płytę Arduino.
- 3 Nowy plik
- 4 Otwórz plik
- 5 Zapisz plik.
- 6 po prawej Otwarcie "monitora szeregowego".

**Korzystanie z pobranych programów:**

Programy są umieszczone zawsze w folderze, plik lub pliki są źródłami programu. Wśród nich jest plik z rozszerzeniem ".ino". To jest program, który zostanie załadowany.  
Uwaga, folder zawsze ma nazwę pliku ".ino".

Przykład:

Documents

Arduino

Applications

Toto

Toto.ino

Lili

Lili.ino

Ardu

Ardu.ino, conf.h, conf.ccp

Katalog pobranego programu zostanie umieszczony w folderze "Arduino, aplikacje"

W języku Arduino unikamy mówienia o programach, wolimy używać słowa "szkic", jest to łatwiejsze, ponieważ oznacza to, co jest otoczone na czerwono.

**Zalecenia:**

Po pobraniu szkicu, aby zachować go w stanie oryginalnym, lepiej jest zmienić jego nazwę. Aby to zrobić, należy zmienić nazwę folderu i pliku ".ino" (**muszą mieć tę samą nazwę**). Najprostszym sposobem jest załadowanie programu do IDE i natychmiastowe zapisanie go pod nową nazwą "Plik, Zapisz jako". Zaletą tego sposobu jest stworzenie nowego szkicu, gotowego do użycia, i pozostawienie starego w wersji oryginalnej.

W przypadku nowej nazwy zalecam zachowanie oryginalnej nazwy i dodanie 2 liter (drugiego imienia) oraz numeru wersji.

Przykład: szkic **Ardu**, załadowałem **Ardu.ino**, razem z nim załadowały się pliki **conf.h** i **conf.ccp**, w IDE znajdziemy karty **Ardu.ino**, **conf.h** i **conf.ccp**.

Chcę go zapisać pod inną nazwą: Wybieram "Plik, i Zapisz jako" Podaję nową nazwę "**Ardu\_al1**" i klikam "Zapisz", w IDE znajdziemy karty **Ardu\_al1.ino**, **conf.h**, i **conf.ccp**.

AERODEN	Opis Arduino + Ramps Programowanie Arduino	AD/ARD/PRO/01
---------	---	---------------

I patrząc na pliki mam to:

Documents

Arduino

Applications

Toto

Toto.ino

Lili

Lili.ino

Ardu

Ardu.ino, conf.h, conf.ccp

Ardu\_al1

Ardu\_al1.ino, conf.h, conf.ccp

Ten sposób jest znacznie prostszy. Nie zapomnij skomentować zmian w nagłówkach plików, po 3 lub 4 dniach nie będziesz wiedział, co zrobiłeś.

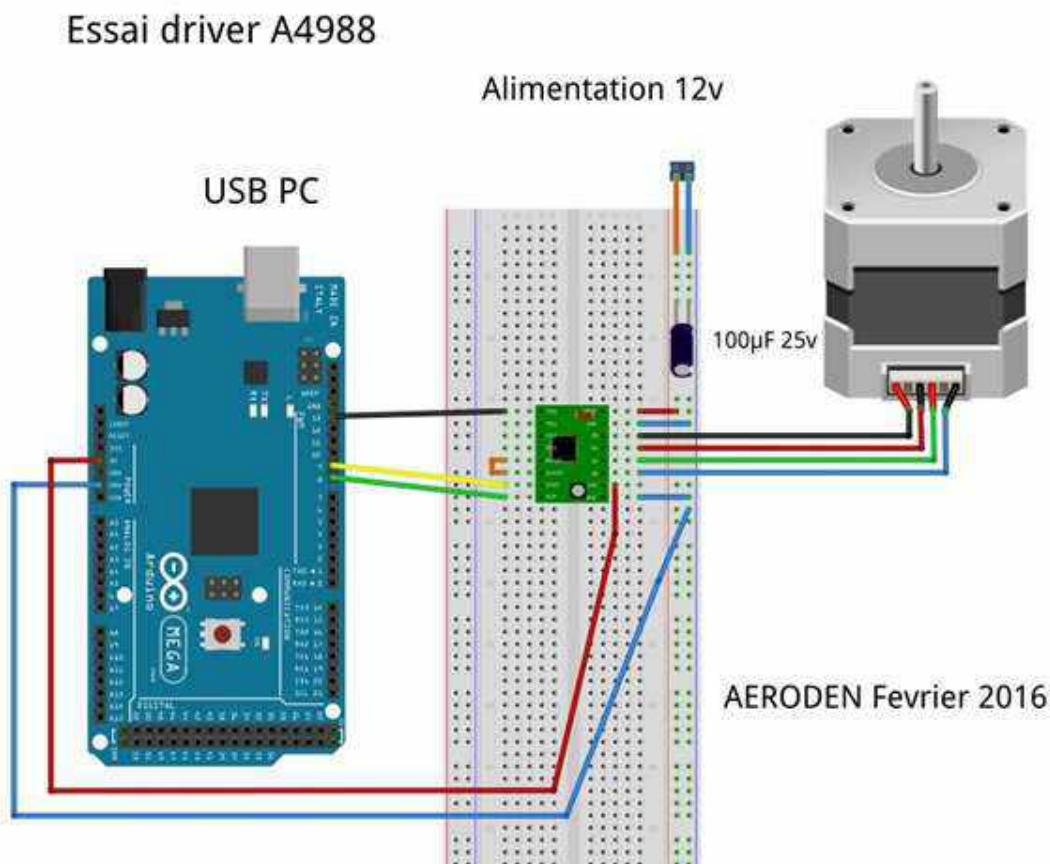
Bravo! Zdobyłeś już niezbędną wiedzę  
z wykorzystaniem tego materiału.

Teraz przejdźmy do testowania i ustawienia sterowników  
POLOLU

**Środki ostrożności:** Ponieważ jest to tani materiał i ma różne pochodzenie, poświęcam czas na wypróbowanie sterowników przed umieszczeniem ich na płytce Ramps.

Jeśli nie masz płytki testowej, możesz przejść bezpośrednio do testów płyty Ramps 1.4



**Testy i regulacja sterowników POLOLU:**

Wykonałem montaż jak wyżej.

```
#define pinEnable 13 // Activation du driver/pilote  
#define pinStep 9 // Signal de PAS (avancement)  
#define pinDir 8 // Direction
```

Przed wszystkim nie zapomnij o spolaryzowanym kondensatorze 100µF.

Uwaga: lepiej najpierw podłączyć gniazdo USB, a następnie zasilanie 12 V (moc silników). Aby wyłączyć, odłącz 12 V, a następnie wtyczkę USB. (To dla ochrony sterowników).

Podłącz gniazdo USB.

Do testów, silnik bez obciążenia, ustawiam REF na 0,5 V, co odpowiada  $0,5 / (8 \times 0,1) = 0,625$  A. Dla tych ustawień radiatory nie są konieczne.

Jeśli rezystory pomiaru prądu Twojej płytki są inne niż tutaj opisane, przelicz ponownie ustawienie REF zgodnie z opisem poniżej.

Pomiar tego REF odbywa się za pomocą multimetru na zakresie 2V, COM podłączony do GND i + na suwaku potencjometru. REF dla A4988 jest zawsze obecny, gdy tylko pojawi się 5 V.

Szkic testu nie jest mój, pobrałem go w Internecie i nazywa się A4988\_Test. Jest dostarczany z tym dokumentem, umieszczasz go w "[Arduino, Applications](#)", uruchamiasz IDE, i wybierasz: "[Plik, Otwórz, Dokumenty, Arduino, Aplikacje, A4988\\_Test, A4988\\_Test.ino, Otwórz](#)"

Twój plik jest otwarty; ;

Test wykonuje powolny obrót 200 kroków w jednym kierunku, a obrót 10 razy szybciej w drugim kierunku, przy końcu silnik pozostaje pod napięciem przez 5 sekund. Aby ponownie uruchomić program, naciśnij Reset na płycie Mega i nowy cykl rozpoczyna się ponownie.

Sprawdź wybór płyty Arduino i portu COM. (USB jest już podłączone)

Podłącz silnik, a następnie zasilanie silnika 12V.

Kliknij na ikonę poziomej strzałki, szkic zostanie skompilowany i przestany do Arduino.

Silnik się uruchamia.

Możesz otworzyć monitor szeregowy klikając na prawą ikonę (szkło powiększające), wybierz w prawym dolnym rogu okna 9600. Zobaczysz kolejność czynności.

Być może masz pomruki silnika z dużą prędkością, to nie ma znaczenia, brakuje trochę mocy.

Jeśli twój sterownik działa, wyłącz zasilanie 12V potem USB, zmień sterownik i ponownie podłącz USB, ustaw właściwy REF, podłącz silnik, potem zasilacz 12V i wciśnij Reset na płytce Mega, jeśli silnik działa poprawnie wypróbuj tak samo pozostałe sterowniki Pololu.

Sprawdziłem 5 sterowników A4988, które działają poprawnie.

Teraz możesz przykleić radiatory do obwodów A4988, **zachowaj ostrożność, i umieść je tak, by nie dotykały żadnego elementu.**

## Test płytki Ramps 1.4:

Przygotowanie płytki Ramps 1.4: Zaleca się sprawdzenie zwarć na zworkach wyboru mikrokroków. Weź swój multimetr na zakresie 200  $\Omega$  i sprawdź, czy nie ma zwarcie między pinami, na które zakładamy zworki, miałem płytkę gdzie było zwarcie między pinami, więc bez zworki, sterownik Pololu pracował z 1/16 kroku. Płytki nie są drogie ...! Jeśli tak jest w twoim przypadku, to nie masz szczęścia, należy wylutować piny i przeciąć zwarcia frezem..

Teraz możesz wstawić płytkę Ramps do płyty Arduino Mega. Przygotuj zasilanie 12 V, aby podłączyć je do listwy zaciskowej płyty Ramps 1.4.

Wybierz mikrokroki i umieść zworki zgodnie z wcześniejszym opisem.

Wstaw sterownik Pololu.

Podłącz silniki.

Szkic do testowania płyty Ramps 1.4 nazywa się Ramps14\_Test.

Szkic opisany jest w nagłówku. Zaprojektowałem ten program, czerpiąc inspirację z tego, którego używałeś do testu sterowników. Zwykle gniazda sterowników to E0, E1 X, Y, Z. W moim szkicu, ([nie jest to drukarka 3D](#)), sterowniki oznaczone są E0, E1, E2, E3, E4.

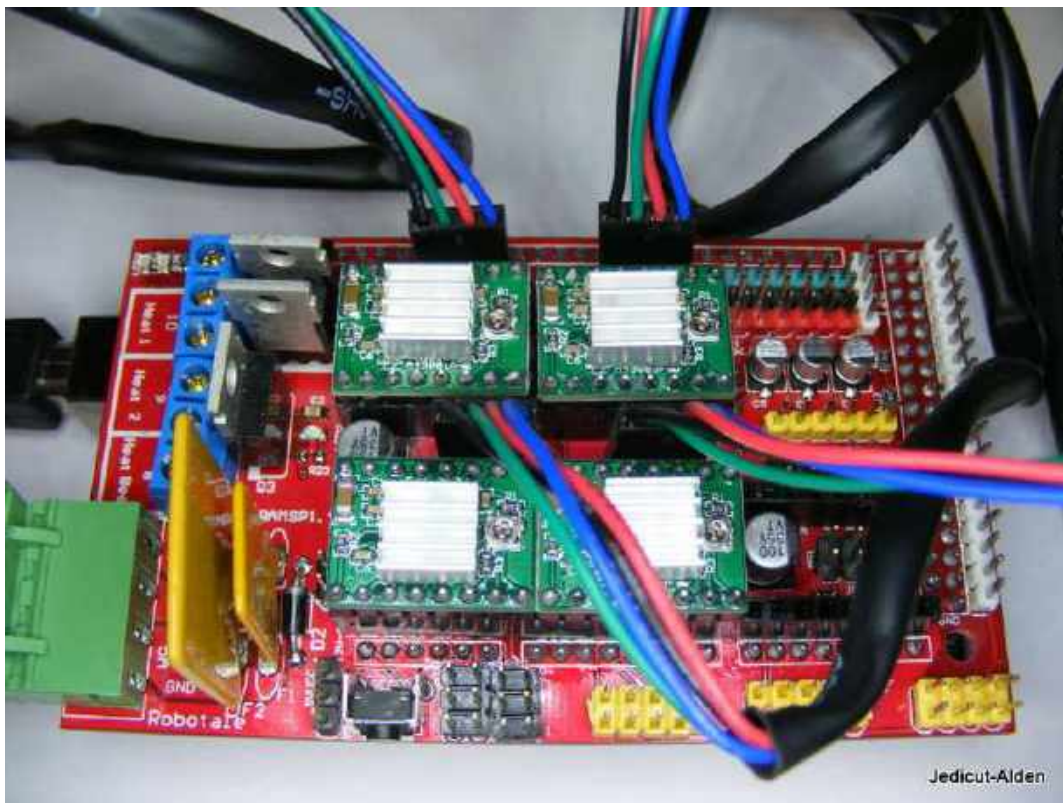
Nie użyłem 5 sterowników, ale 4, pozwoliłem ci wcielić w życie to, co opisałem, by zrobić kolejny szkic dla 5 sterowników.

Teraz jesteś gotowy na test płyty Ramps1.4 i 4 sterowników Pololu.

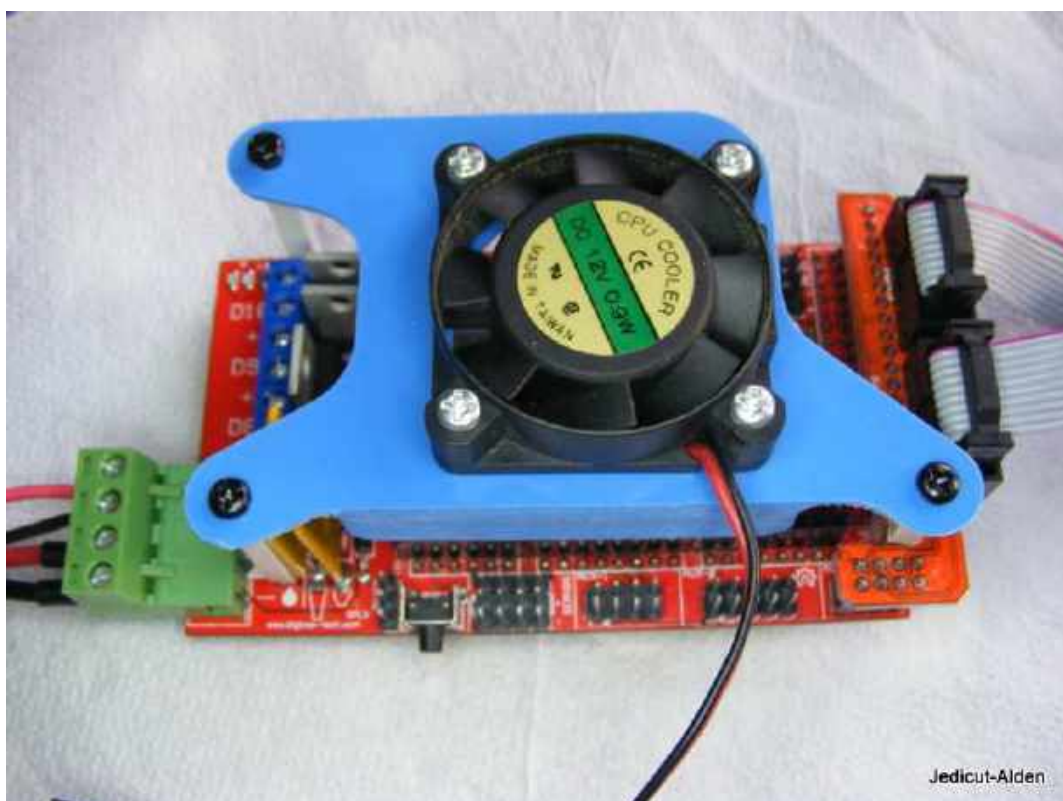
Zrób te same manipulacje, co w procesie testowania sterowników.

Ustaw prąd znamionowy silników, Zamontuj radiatory i wentylator. .

Oto połączenia do testowania płyty Ramps 1.4.

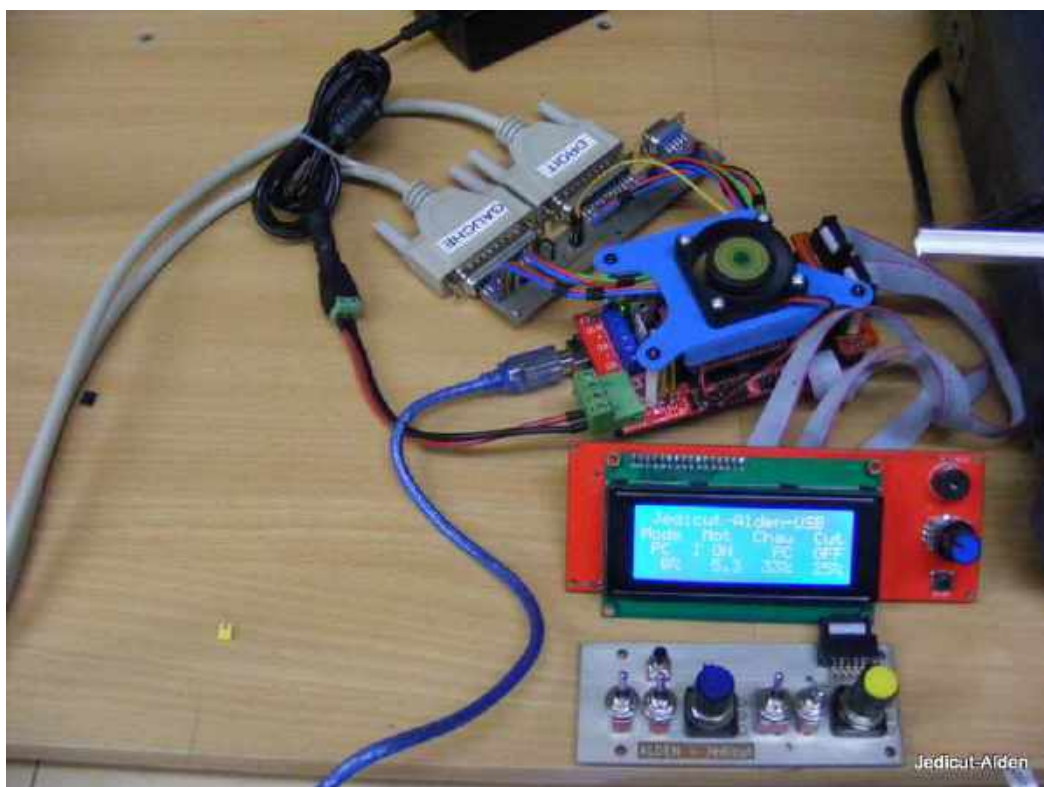


Płytkę wyposażoną w wentylator:





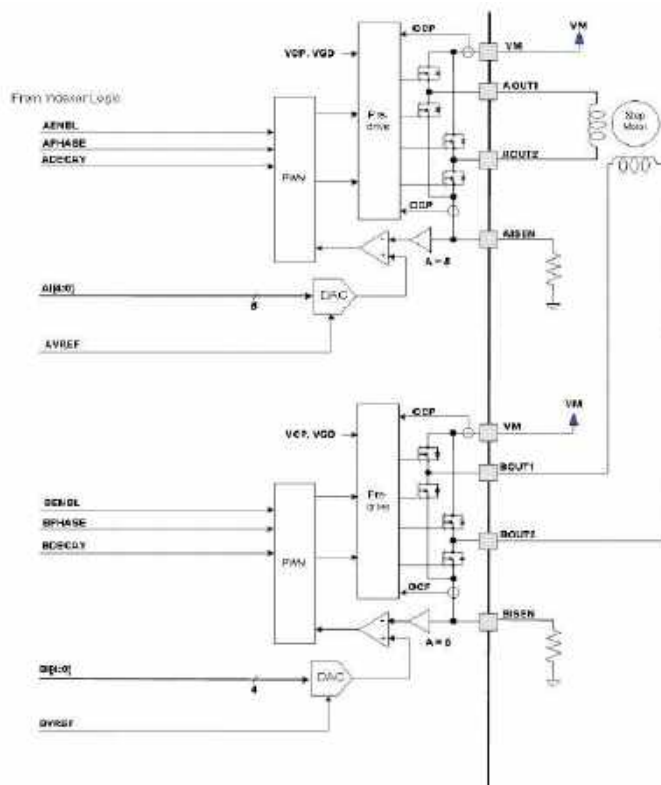
Oto realizacja sterowania i oprogramowania do cięcia gorącym drutem Jédicut zastępuje doskonałą kartę MM2001



To właśnie otrzymujemy z tego materiału.



## Sterowniki DRV 8825:



DRV8825 to kompletny sterownik silnika krokowego ze zintegrowanym regulatorem prądu.

Przeznaczony jest do współpracy z krokowymi silnikami dwubiegunowymi w pełnych, 1/2, 1/4, 1/8, 1/16 i 1/32 krokach, z napięciem wyjściowym 45V i prądem maks.  $\pm 2,2$  A.

Część po prawej stronie przedstawia obwód mocy i 2 uzwojenia silnika krokowego. Każde uzwojenie jest zasilane przez tranzystorowy przetwornik H DMOS. Zasilanie silników jest realizowane przez VM, musi być odsprężone przez kondensator 100  $\mu$ F, **kondensator ten jest bardzo ważny**. Powrót do GND odbywa się poprzez rezystory do pomiaru prądu. Wynikiem pomiaru prądu jest wartość AISEN dla AOUT1 i AOUT2 oraz BISEN dla BOUT1 i BOUT2, wartość ta jest przesyłana do wzmacniacza operacyjnego w celu porównania z AVref i BVref. Doprecyzowałem to tak, aby łatwo było uzyskać aktualne ustawienie dostosowane do silników. Wystarczy zmierzyć napięcie REF za pomocą multimetru. Potencjometr regulacji VREF pololu jest połączony z zaciskami AVref i BVref, które znajdują się na 2 przetwornikach DAC. Będziemy używać 3 terminali "step", "dir" i "enable barre" (sterownik jest odblokowany przez "0"), dla rozkazów "step" i "dir", musisz przestrzegać czasów przejścia między rozkazami w szkicu Arduino.

Oto schemat.

W naszym przypadku nie będziemy brać pod uwagę wyboru mikrokroków i resetowania, ponieważ są to operacje, które są wykonywane poza pracą.

## TIMING REQUIREMENTS

			MIN	MAX	UNIT
1	$f_{STEP}$	Step frequency		250	kHz
2	$t_{WH(STEP)}$	Pulse duration, STEP high	1.9		$\mu s$
3	$t_{WL(STEP)}$	Pulse duration, STEP low	1.9		$\mu s$
4	$t_{SU(STEP)}$	Setup time, command to STEP rising	650		ns
5	$t_{H(STEP)}$	Hold time, command to STEP rising	650		ns
6	$t_{ENBL}$	Enable time, nENBL active to STEP	650		ns
7	$t_{WAKE}$	Wakeup time, nSLEEP inactive to STEP	1.7		ms

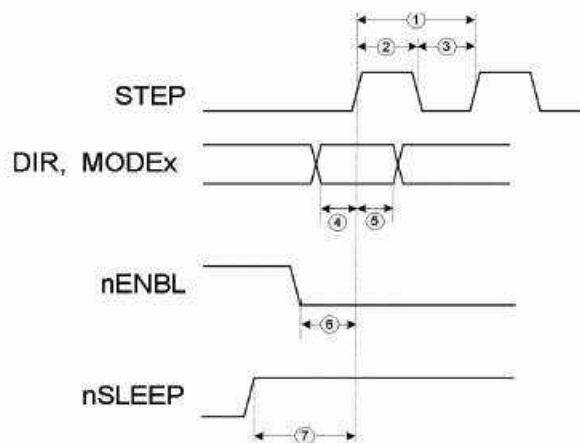


Figure 1. Timing Diagram

Widzimy, że impuls "STEP(krok)" musi nadejść po zmianie rozkazu "DIR(kierunek)" na "4" > 650 ns. Że czas impulsu krokowego "2" musi być większy niż 1,9  $\mu s$ ; Że czas "3" pomiędzy dwoma impulsami krokowymi musi być większy niż 1,9  $\mu s$ .

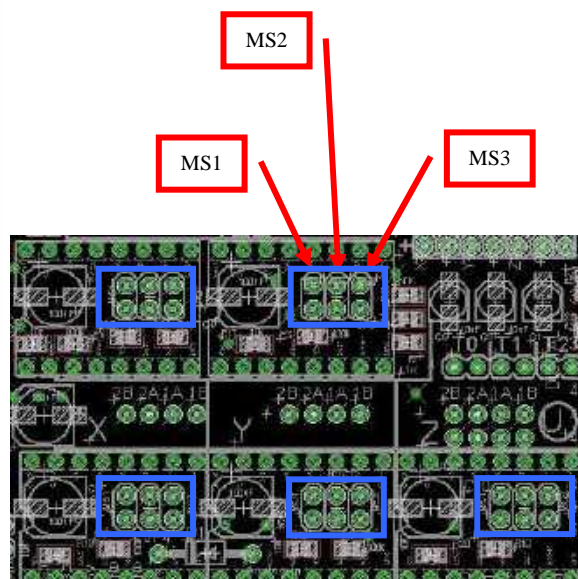
Zmiana kierunku nie może nastąpić przed czasem "6" > 650ns.

W większości przypadków jesteśmy w tych zakresach, ale trzeba być ostrożnym, jeśli zauważy się pomruki silników lub utratę kroku.

Zobaczmy wybór mikrokroków ( $\mu pas$ ) mamy do dyspozycji 3 wejścia MODE0 => MS1: MODE1 => MS2: MODE2 => MS3.

Table 2. Stepping Format

MODE2	MODE1	MODE0	STEP MODE
0	0	0	Full step (2-phase excitation) with 71% current
0	0	1	1/2 step (1-2 phase excitation)
0	1	0	1/4 step (W1-2 phase excitation)
0	1	1	8 microsteps / step
1	0	0	16 microsteps / step
1	0	1	32 microsteps / step
1	1	0	32 microsteps / step
1	1	1	32 microsteps / step



Wszystkie pięć sterowników posiada konfigurowalne mkrokroki, na zdjęciu znajdują się złącza pozwalające na ustawienie zworek zgodnie z wybranym mikrokrokiem. Są pod sterownikami "Pololu".

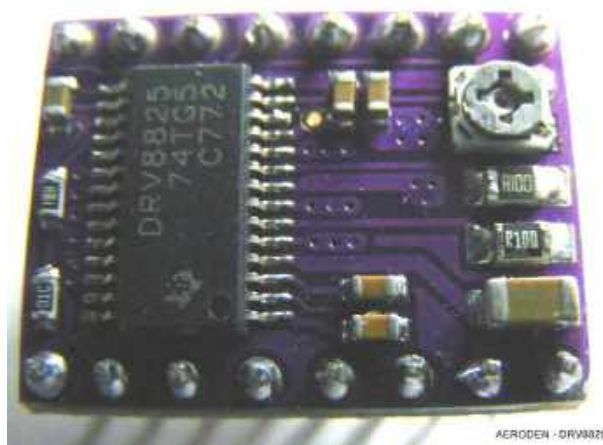


Bez zworki dostajemy "000" to pełny krok, wstawiając zworkę na MS1 otrzymujemy  $\frac{1}{2}$  kroku. Istnieją 3 możliwości ustawienia zworek dla kroku 1/32, najprostsza "111" 3 zworki założone.

Polecenia RESET i SLEEP nie są używane, zostaną ustawione na "1"

Sterowniki te są montowane z innymi komponentami na małej karcie (Pololu), która jest podłączona do Ramps 1.4 lub innej karty. Obwód DRV8825 musi być zaopatrzony w radiator. Dla prądów mniejszych niż 1,5 A to wystarczy, ale dla  $I > 1,5$  A trzeba wymusić chłodzenie wszystkich radiatorów.

### **Badanie "POLOLU" DRV 8825:**



W zależności od producenta obwodów wartości rezystorów i oznaczeń mogą być różne. Jednak schemat pozwala je odnaleźć.

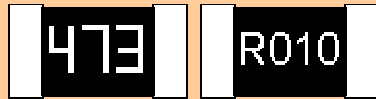
Ważne jest zlokalizowanie obecnych rezystorów pomiarowych. To całkiem proste, to te tuż pod potencjometrem. Teraz musisz znać ich wartość, jeśli nie możesz odczytać wartości, możliwe jest zrobienie zdjęć makro z różnymi kątami oświetlenia. Następnie otwarcie obrazów na komputerze co ułatwia odczytanie napisów. Tutaj rezystory oznaczone są symbolem R100.

Oznakowanie rezystorów SMD:

pierwsza i druga to znaczące liczby,  
Trzeci to liczba zer.

Litera "R" zastępuje kropkę dziesiętną.

Przykład: 473 to 4, a następnie 7, a następnie 3 zera, czyli 47000 omów lub 47 kiloomów (47 KΩ).

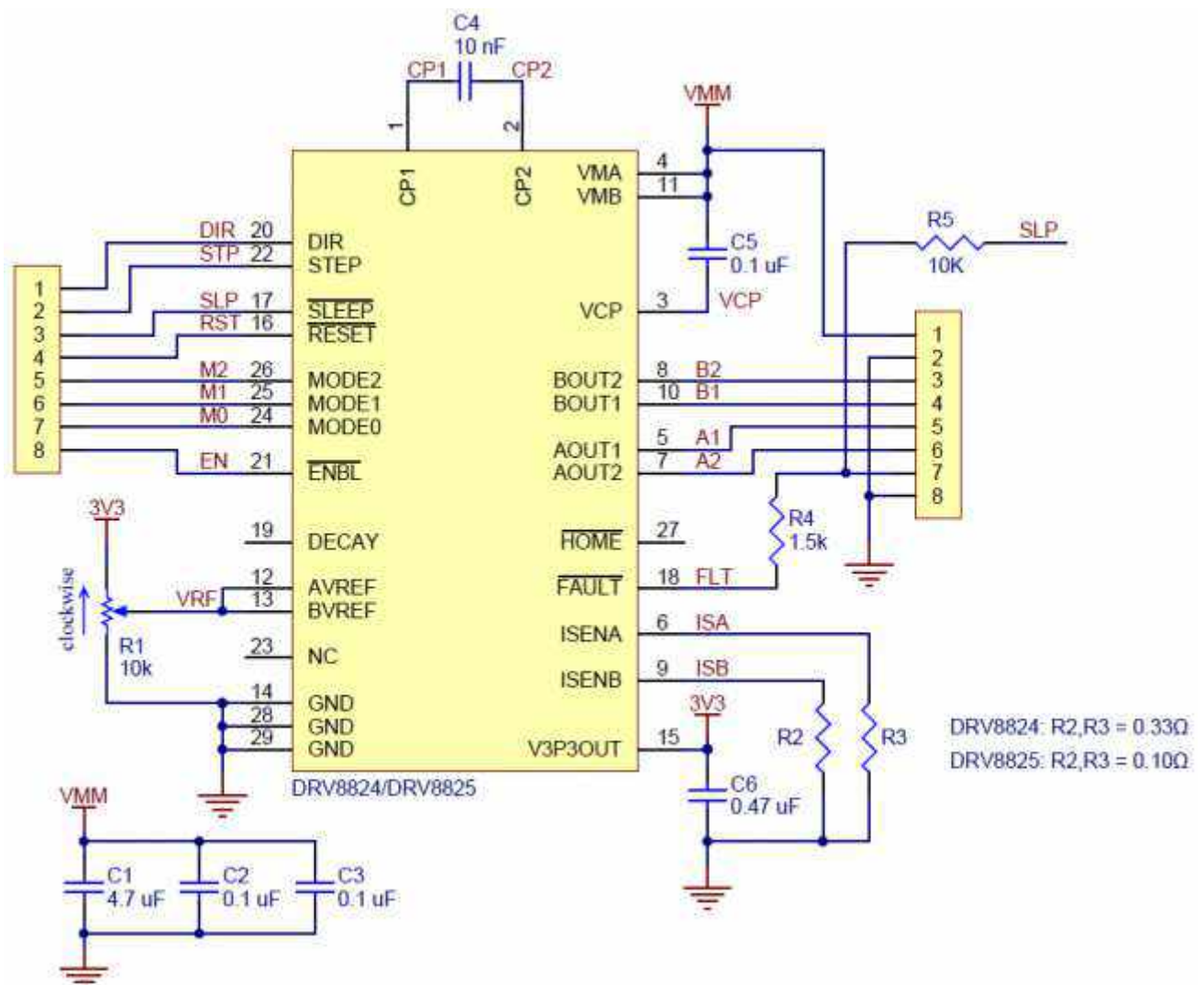


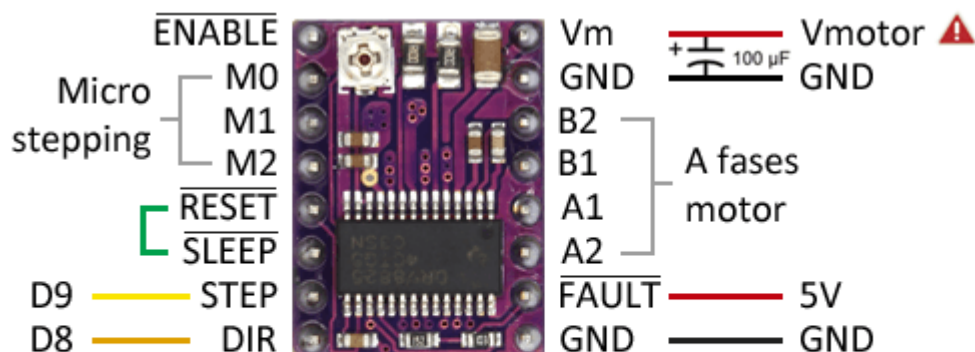
$R = 47 \text{ K ohms}$   $R = 0.010 \text{ ohm}$

W naszym przypadku  $R_{100} = 0,1 \Omega$

Ustawienie zadawania prądu REF będzie dokonywane za pomocą potencjometru. **Uwaga do ustawienia użyj izolowanego śrubokręta.**

Teraz spójrzmy na schemat:





Vmotor entre 8 y 35V. Al usar alimentación externa SIEMPRE poner con GND común.

W widoku z góry zauważysz pin w prawym dolnym rogu, to jest GND i na tym pinie umieścisz (-) miernika do pomiaru VREF, a (+) multimetru będzie w kontakcie z suwakiem potencjometru.

Wygodna jest sonda plusowa miernika w kształcie śrubokręta

Na tym schemacie wartości rezystorów pomiarowych prądu R2 i R3 wynoszą 0,1 Ω jak na sfotografowanym modelu.

Potencjometr wynosi 10 kΩ. Możliwe ustawienie REF wynosi od 0 do 3,3V

za pomocą wzoru arkusza danych:  $I_{Trip\ Max} = VREF / (5 \times R_{Isense})$

Co daje:  $VREF = I_{TripMax} \times 5 \times R_{Isense}$

Przykład: silnik obsługuje 1,5 A > ustawienie VREF będzie  $1,5 \times 5 \times 0,1 = 0,75\ V$

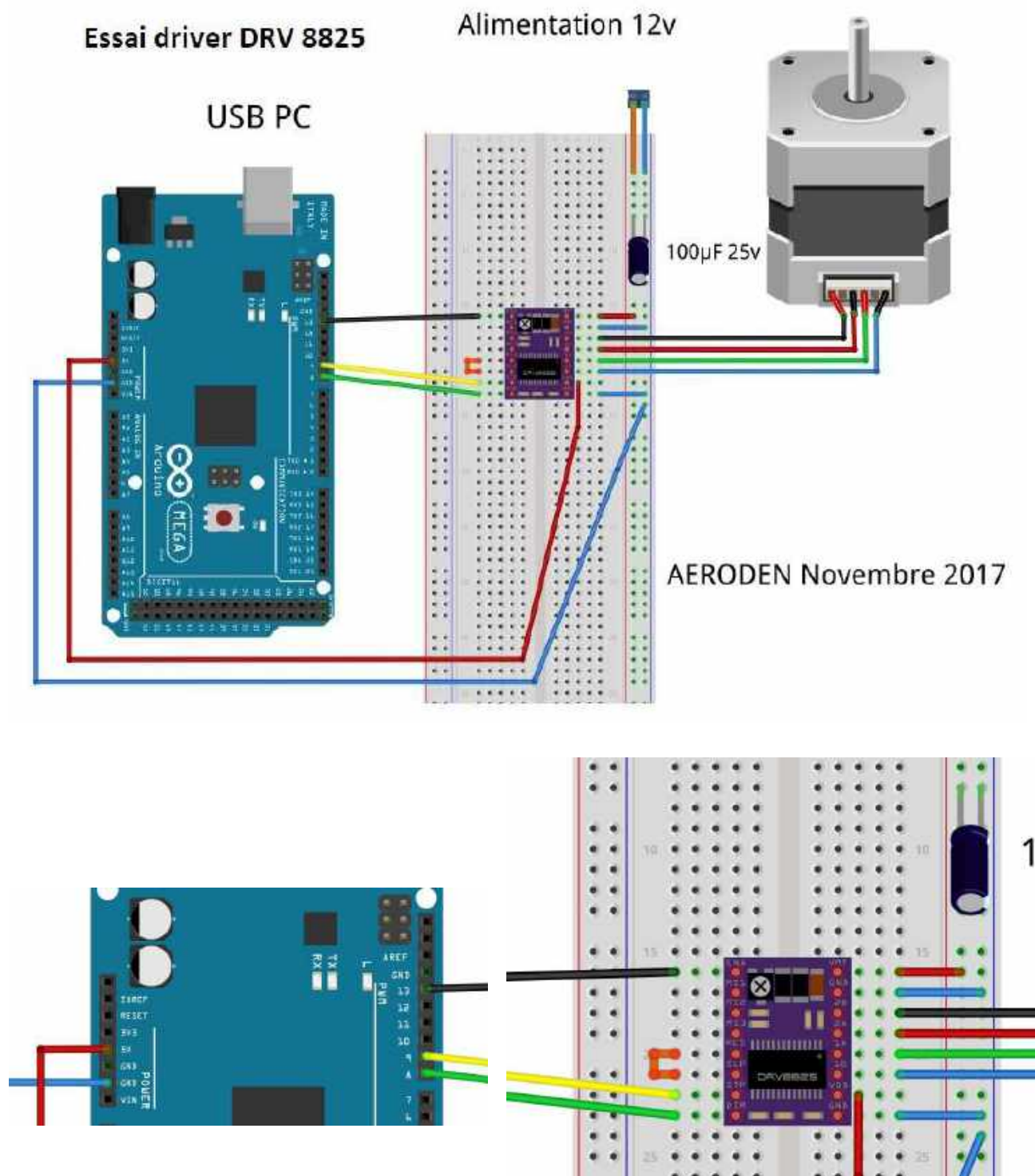
Dla 1,65 A > ustawienie VREF wyniesie  $1,65 \times 5 \times 0,1 = 0,825\ V$

Uwaga: w zależności od źródła, rezystancje  $R_{Isense}$  mogą mieć różną wartość; w takim przypadku dostosuj formułę

Zwracam uwagę, że wydajność i działanie twoich silników PAP jest bezproblemowe, jeśli przyjrzyj się temu dokładnie. Dzisiaj jesteśmy zalewani tanim sprzętem, tym lepiej dla naszego hobby, ale musimy opanować różnice w produkcji.

Zobaczmy dalej, jak dostosować VREF.



**Testowanie i regulacja POLOLU DRV8825:**

Wykonałem montaż jak wyżej.

```
#define pinEnable 13 // Activation du driver/pilote
#define pinStep 9 // Signal de PAS (avancement)
#define pinDir 8 // Direction
```

**Przed wszystkim nie zapomnij o spolaryzowanym kondensatorze 100µF.**

AERODEN	Opis Arduino + Ramps Programowanie Arduino	AD/ARD/PRO/01
---------	---	---------------

Uwaga: lepiej jest najpierw podłączyć wtyczkę USB, a następnie zasilacz 12 V (zasilanie silników). Aby wyłączyć zasilanie, należy odłączyć wtyczkę 12V, a następnie USB. (To jest dla ochrony sterowników).

Podłącz gniazdo USB.

W przypadku badania silnika bez obciążenia ustawiamy VREF na 0,3 V, co odpowiada  $0,3/(5 \times 0,1) = 0,6A$ . Dla tego prądu i bez obciążenia, radiatory nie są konieczne.

Jeśli nie posiadasz tych samych oporników pomiarowych prądu, oblicz ponownie ustawienie zgodnie z opisem w dalszej części.

Pomiar napięcia VREF odbywa się za pomocą multimetru na zakresie 2V, minus podłączony do GND a plus do sywaka trimera. **W przeciwieństwie do A4988 VREF jest obecny tylko wtedy, gdy obecny jest silnik i jego zasilanie 12 V.** Napięcie 3,3 v jest dostarczane przez układ scalony DRV8825.

Szkic do testu nie jest mój, pobrałem go w Internecie i nazywa się "A4988\_Test", jest kompatybilny z DRV8825. Jest dostarczany z tym dokumentem, umieszczasz go w "[Arduino, Applications](#)", uruchamiasz IDE, wykonaj:

["Plik, Otwórz, Dokumenty, Arduino, Aplikacje, A4988\\_Test, A4988\\_Test.ino, Otwórz"](#)

Twój plik jest otwarty:

Test wykonuje powolny obrót 200 kroków w jednym kierunku, a obrót 10 razy szybciej w drugim kierunku, przy końcu silnik pozostaje pod napięciem przez 5 sekund. Aby ponownie uruchomić program, naciśnij Reset na płytce Arduino Mega i nowy cykl rozpoczyna się ponownie.

Sprawdź wybór płyty Arduino i portu COM.

Kliknij na ikonę poziomej strzałki, szkic zostanie skompilowany i przestany do Arduino.

Podłącz 12 V, ustaw trymer na około 0,3 V, naciśnij Reset na Arduino Mega, Silnik się uruchomi.

Możesz otworzyć monitor szeregowy klikając na prawą ikonę (szkło powiększające), wybierz w prawym dolnym rogu okna 9600. Zobaczysz kolejność czynności.

Być może masz pomruki silnika z dużą prędkością, to nie ma znaczenia, brakuje odrobiny mocy; zwiększ VREF.

Jeśli Twój sterownik działa, ustaw trymer na wartość prądu znamionowego Twoich silników obliczoną powyżej. **Nie uruchamiaj teraz silnika, radiator nie jest na miejscu**, wyłącz zasilanie, zmień sterownik i podłącz ponownie USB, podłącz 12 V, ustaw prąd VREF i naciśnij reset karty mega, silnik obraca się OK Wypróbuj tak samo pozostałe sterowniki Pololu.

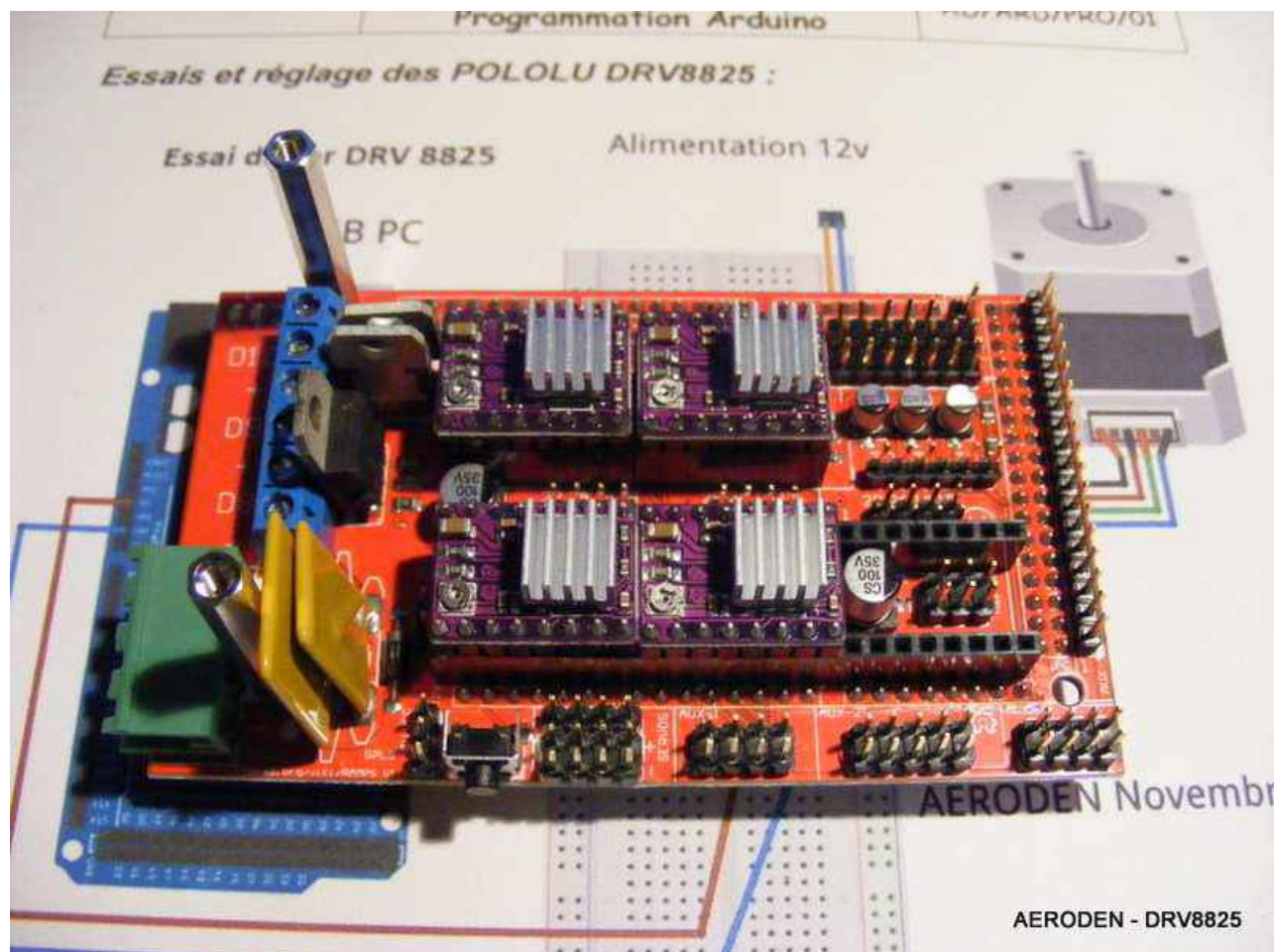
Testuj 5 sterowników DRV8825, które powinny działać poprawnie.

Teraz twój sterownik jest ustawiony, wkładasz go w płytkę Ramps - kierunek jak na zdjęciu niżej.

Teraz możesz przykleić radiatory do obwodów DRV8825, **zachowaj ostrożność, umieść je tak, aby nie dotykały żadnego elementu.**

Łatwiej jest wyjąć sterownik bez radiatorów.

Kierunek ustawiania DRV 8825 na płycie Ramps 1.4.

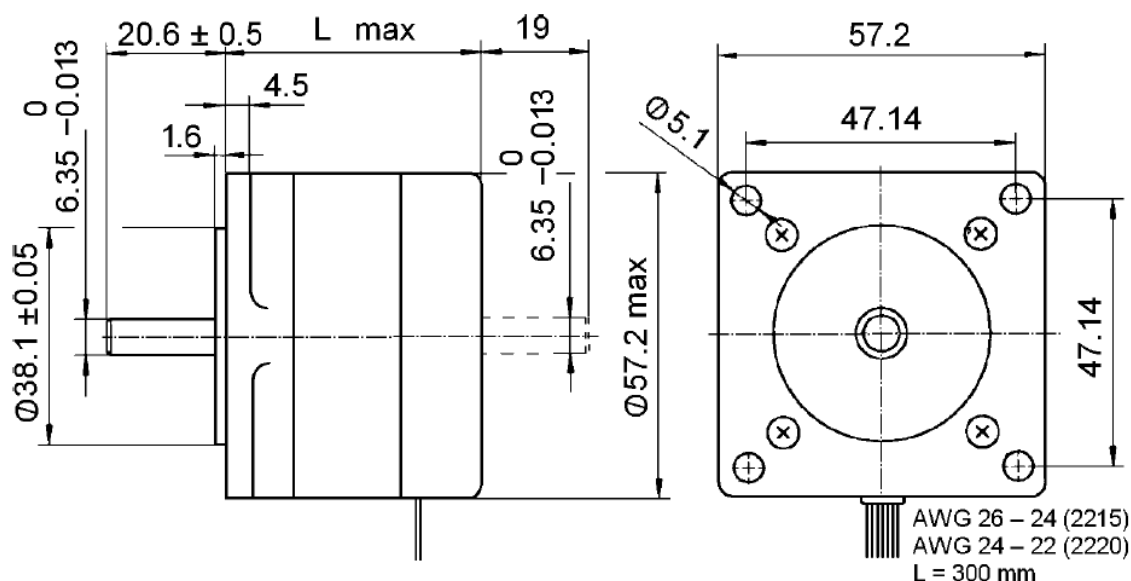




Unipolarny lub bipolarny silnik krokowy

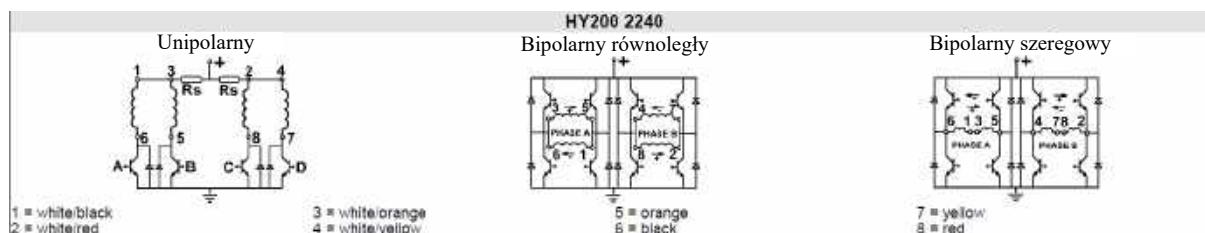
1 = biało-czarny

2 = biało-czerwony

**KLUCZ OPISU SILNIKÓW KROKOWYCH**

HY200	34	24	0085	A	*	X	*	*	08
kroki/obr średn. silnika 3.4 "	dłg. silnika 2.4 "	prąd na fazę 0085=0.85A	nic = bez enkodera wersja endera	X = brochure Y = temporary others = special	nic = standardowe wyjście przewodów U = wyprowadza wyjście z PG C = złącze na płycie	nic = wałek fi 9,525 płaski L = wałek fi 9,525 okrągły R = wałek fi 9,525 z gniazdem klucza S = wałek fi 12 z gniazdem klucza T = wałek fi 12,7 z gniazdem klucza			
typ silnika HY = hybryda HS/HN = hybryda wysoka wydajność			A=1 wał B=2 wały E=enkoder 3kan.200ppr. bez LD L=enkoder 3kan.200ppr. z LD R=z przekładnią S=bryzgoszczelny inne litery=specjalny						

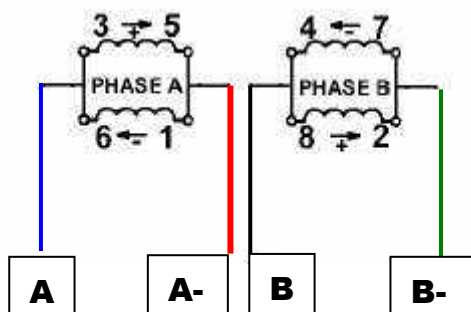
Specifications	HY200 2215					HY200 2220		
	0150 AX08 0150 BX08	0100 AX08 0100 BX08	0033 AX04 0033 BX04	0220 AX04 0220 BX04		0044 AX08 0044 BX08	0100 AX08 0100 BX08	0210 AX08 0210 BX08
step angle	1,8°	1,8°	1,8°	1,8°		1,8°	1,8°	1,8°
step angle accuracy	%	5	5	5		5	5	5
rated phase current	A	1,5	1,0	0,33	2,2	0,44	1,0	2,1
phase resistance	Ω	1,5	3,4	33,8	0,7	23	5	1,1
phase inductance	mH	1,5	3,8	54,6	1,2	39,2	8	1,7
holding torque unipolar *	Ncm	25	27	-	-	52	53	52
holding torque bipolar *	Ncm	33	34	32	31	65	69	64
detent torque	Ncm	3,4	3,4	3,4	3,4	5,3	5,3	5,3
rotor inertia	g cm <sup>2</sup>	56	56	56	56	124	124	124
mass	Kg	0,34	0,34	0,34	0,34	0,5	0,5	0,5
max. length	mm	40	40	40	40	52	52	52
max. applicable voltage	V	75	75	75	75	75	75	75
insulation class	B	B	B	B	B	B	B	B

**HY200 2240**

Dwubiegunowy równoległy  
Bipolarny równoległy

1 = biało/czarny      3 = biało/pomarańczowy  
2 = biało/czerwony      4 = biało/żółty

5 = pomarańczowy      7 = żółty  
6 = czarny      8 = czerwony



Do pracy ze sterownikami A4988 lub DRV 8825 konieczne jest wykonanie połączenia w układzie równoległym bipolarnym.. Aby to zrobić:

Połącz przewody 3, 6, i przylutuj niebieski przewód.

Połącz przewody 5, 1, i przylutuj czerwony przewód

Połącz przewody 8, 4, i przylutuj czarny przewód.

Połącz przewody 2, 7, i przylutuj zielony przewód.

Natężenie prądu dla tych silników wynosi 0,7A na cewkę. Jako że uzwojenia połączone są równolegle, bieżące ustawienie będzie 1,4A na sterowniku.

Przetestuj przy pomocy szkicu dołączonego do pliku Arduino + Ramps 1.4.

<b>AERODEN</b>	<b>Opis Arduino + Ramps Programowanie Arduino</b>	<b>AD/ARD/PRO/01</b>
----------------	---	----------------------

**Historia zmian w dokumencie:**

Wersja 1.0 Z dnia 03/11/2017 Dodano opis i dostosowanie sterowników DRV 8825> strona 28-34

Wersja 2.0 Z dnia 22/11/2018 Uzupełnienie o uzwojenia silnika krokowego.

Wersja 3.0 Z dnia 13/11/2019 Poprawiono błąd pisarski w aktualnej formule na stronie 13