

STEROWNIK MASZYNY CNC (via usb)

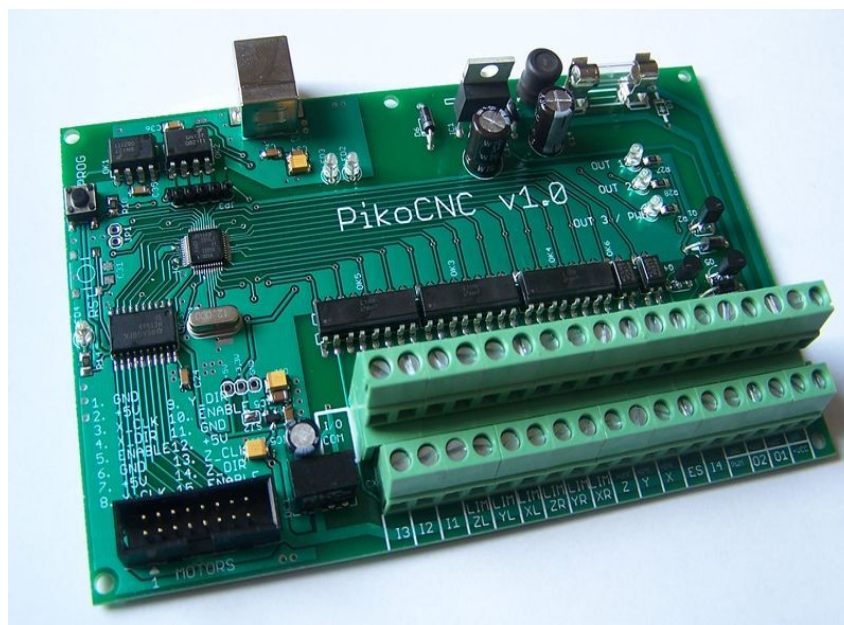
PikoCNC v1.01

instrukcja obsługi ver. 1.8 (18.11.2011)

copyright: Andrzej Woźniak 2011

P.P.H.U ELCOSIMO

wozniak_andrzej@wp.pl



LICENCJA

Prawa autorskie:

Właścicielem praw autorskich do programu PikoCNC oraz oprogramowania sterownika jest:

Andrzej Woźniak P.P.H.U ELCOSIMO Damasławek 2011

wozniak_andrzej@wp.pl

Licencja:

Autor udziela Użytkownikowi tej wersji programu PikoCNC bezpłatnej licencji na czas nieokreślony - zarówno do użytku komercyjnego jak i domowego.

Rozpowszechnianie:

Program w niezmienionej formie może być rozpowszechniany bezpłatnie bez wiedzy autora.

Ograniczenia:

Zabroniona jest modyfikacja, dekompilacja i deasemblacja każdej części programu, włączając w to zmiany lub usuwanie zasobów. Zabronione jest jakiekolwiek wykorzystywanie programu z innym sprzętem sterującym niż autora programu.

Program PikoCNC jest rozpowszechniany „jak jest”. Nie udziela się na niego żadnej gwarancji. Autor nie ponosi żadnej odpowiedzialności gwarancyjnej lub odszkodowawczej za skutki używania jego oprogramowania. Każdy używa go na własne ryzyko. Autor nie ponosi odpowiedzialności za utratę danych, szkody, utratę zysków lub jakąkolwiek inną stratę wynikłą z używania tego programu zgodnie lub niezgodnie z jego przeznaczeniem.

Jeśli Użytkownik nie zgadza się na przestrzeganie warunków niniejszej umowy licencyjnej, nie ma on prawa do używania programu PikoCNC i powinien go niezwłocznie usunąć ze swego komputera.

Podstawowe parametry

- liczba obsługiwanych osi: 3
- maksymalna częstotliwość sygnału STEP: 120kHz
- minimalna szerokość impulsów STEP: 3us
- bufor FIFO kontrolera: 160 wektorów
- zdolności przesyłowe PC->kontroler: 1000 wektorów/sek.
- pełna optoizolacja sterownika od strony USB
- liczba obsługiwanych wyjść: 3 (OUT1, OUT2, PWM)
- liczba obsługiwanych wejść 14 (wszystkie wejścia optoizolowane)
 1. I1-I4 (Start, Stop, Pauza, Reset)
 2. ES (E-Stop)
 3. 6 krańcówek limit po dwie na oś
 4. 3 krańcówki bazowania
- obsługa sygnału ENABLE
- 32-bitowe liczniki położenia osi (czyli obszar pracy praktycznie nieograniczony)
- sprzętowe rozpędzanie i hamowanie silników
- stała prędkość na łukach
- Obsługiwane formaty pliku: G-kod, HPGL, DXF, ESSI, BMP, BMP wykuwanie
- Skanowanie (za pomocą dalmierza laserowego)
- Obsługiwane G-kody: G0 G1 G2 G3 G4 G20 G21 G28 G30 G98 G80 G81
- Wymagany system operacyjny: WindowsXP, Vista, Windows7
- Pozostałe cechy programu:
 1. Możliwość rozpoczęcia obróbki w dowolnym miejscu programu
 2. Wbudowany prosty w obsłudze PikoCam.
 3. Możliwość dokonywania przekształceń programu typu: skalowanie, obroty, przesuwanie. (przesuwanie także za pomocą myszki)
 4. Możliwość rejestracji na dysku ścieżki narzędzia
 5. Bardzo mała obciążalność systemu.

Instalacja

Sam program nie wymaga żadnej instalacji folder z programem należy wyjąć z archiwum w docelowe miejsce na dysku. Jeżeli po podłączeniu sterownika system zapyta o sterowniki należy wybrać opcję ze wskazaniem lokalizacji a następnie wskazać na folder „drivers”.

W razie potrzeby, aktualnych wersji sterowników można szukać na stronie producenta: <http://www.ftdichip.com/Drivers/VCP.htm>

Aktualizacja FirmWare

Do aktualizacji wewnętrznego oprogramowania sterownika służy program „FWupdate”. Aby wprowadzić sterownik w tryb aktualizacji należy przy wyłączonym zasilaniu sterownika przytrzymać przycisk „prog” na jego płycie a następnie (cały czas go trzymając) załączyć zasilanie sterownika. Wejście w tryb sygnalizuje dioda stanu (mruga 2x wolniej niż normalnie). Następnie uruchomić program „FWupdate”, wybrać odpowiedni plik firmware (pliki z rozszerzeniem „.wsd”) i nacisnąć programuj.

Podłączenie

Wyprowadzenia złącze „Motors” (IDC16)

1. GND
2. +5V
3. X_CLK
4. X_DIR
5. ENABLE
6. GND
7. +5V
8. Y_CLK
9. Y_DIR
10. ENABLE
11. GND
12. +5V
13. Z_CLK
14. Z_DIR
15. ENABLE
16. NC

Na Złączu sygnały rozmieszczono tak aby można było łatwo rozdzielić taśmę na trzy części (po 5 linii) i w prosty sposób połączyć z poszczególnymi sterownikami osi. Dla sygnału ENABLE stanem aktywnym jest stan wysoki (H). **Do wyjść „+5V” absolutnie nie należy podłączać jakichkolwiek zewnętrznych źródeł napięcia.**

Listwa zaciskowa

COM	COM	COM	COM	COM	COM	COM	COM	COM	COM	COM	COM	COM	COM	GND	VCC	VCC	GND
IN 3	IN 2	IN 1	LIM ZL	LIM YL	LIM XL	LIM ZR	LIM YR	LIM XR	HOME Z	HOME Y	HOME X	ES	IN 4	PWM	O 2	O 1	VCC

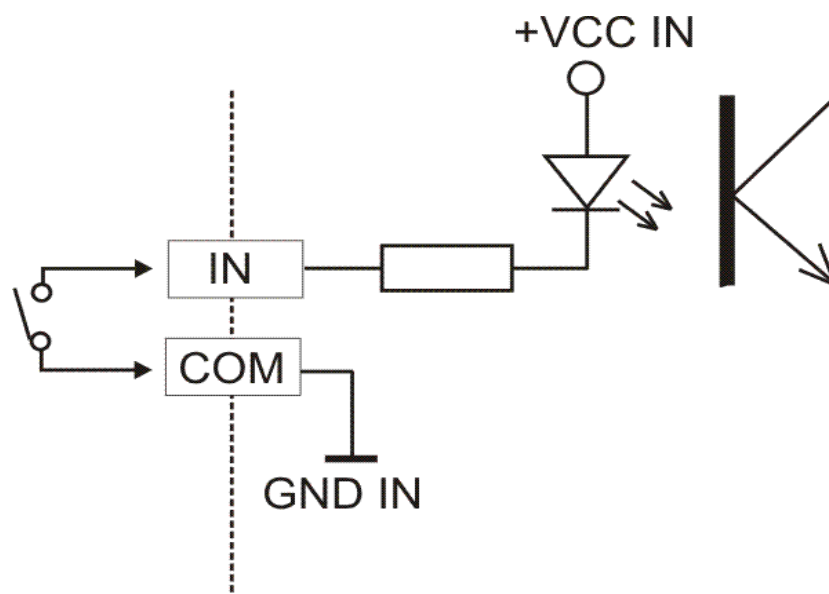
Nazwa	Funkcja	Typ wejścia
I3	Wejście przycisku „PAUZA”. Przyciśnięty razem z przyciskiem „RESET” inicjuje cykl jazdy referencyjnej.	NO
I2	Wejście przycisku „STOP”. Przyciśnięty razem z przyciskiem „RESET” inicjuje cykl pomiaru wysokości materiału.	NO
I1	Wejście przycisku „START”	NO
LIM ZL	Wejście krańcówki dolnej osi Z	NC
LIM YL	Wejście krańcówki lewej osi Y	NC
LIM XL	Wejście krańcówki lewej osi X	NC
LIM ZR	Wejście krańcówki górnej osi Z	NC
LIM YR	Wejście krańcówki prawej osi Y	NC
LIM XR	Wejście krańcówki prawej osi X	NC
HOME Z	Wejście krańcówki home osi Z	NO
HOME Y	Wejście krańcówki home osi Y	NO
HOME X	Wejście krańcówki home osi X	NO
ES	Wejście przycisku E-Stop	NO
I4	Wejście przycisku „RESET”	NO
PWM	Wyjście PWM	
O2	Wyjście sterowania przekaźnika 2	
O1	Wyjście sterowania przekaźnika 1	
VCC / GND	Wejścia zasilania sterownika (8-24V DC)	
COM	Wspólne dla wszystkich wejść	

Napięcie zasilające układ może być w przedziale 8-24V jednak należy pamiętać, że przekaźniki sterujące muszą być na to samo napięcie co zasilanie układu! Tak więc jeżeli zasilamy układ napięciem 12V to musimy zastosować przekaźniki z cewką na 12V. Napięcie zasilające nie musi być stabilizowane zatem wystarczy mostek graetza oraz koniecznie kondensator elektrolityczny np. 2200uF.

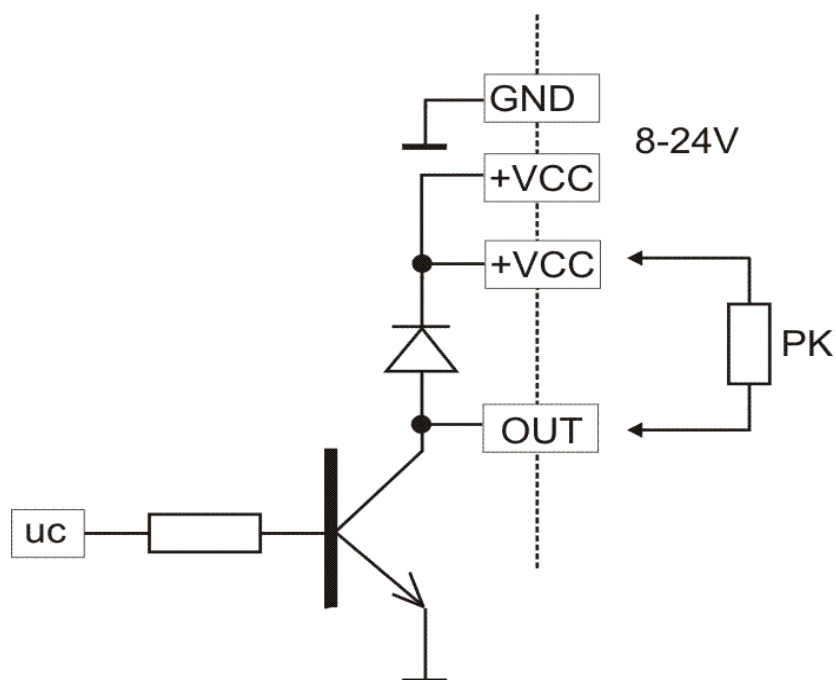
Należy też zwrócić uwagę że wejścia krańcówek LIMIT w odróżnieniu od pozostałych są NC (normalnie zamknięte) !.

Sygnały na listwie zostały tak wyprowadzone, że dla każdego dolnego gniazda parą jest

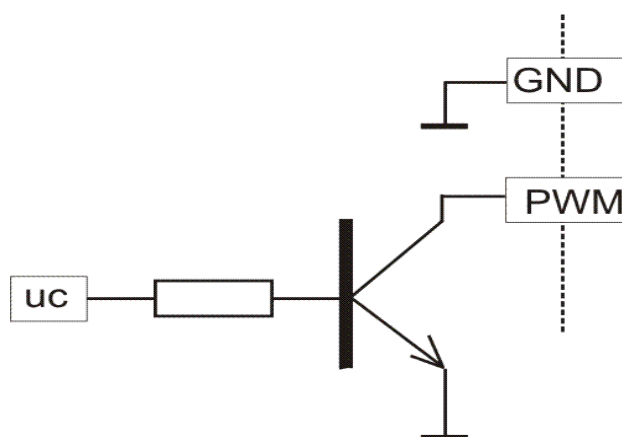
górne gniazdo (przesunięte trochę w lewo). Prawe skrajne gniazda (dolne VCC, i górne GND) to wejścia zasilania układu. Sterownik posiada zabezpieczenie na wypadek podłączenia odwrotnej polaryzacji.



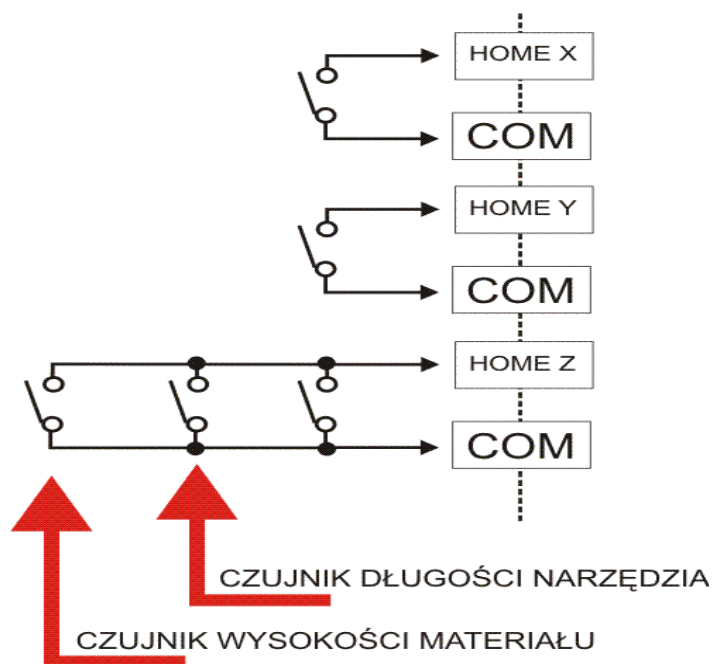
Schemat wewnętrzny toru wejściowego wszystkich wejść.



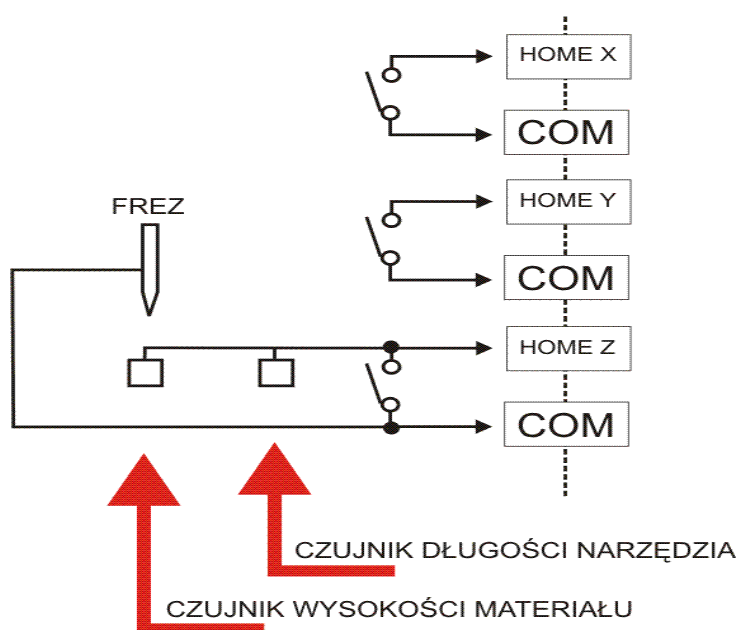
Schemat wewnętrzny wyjść O1 O2 sterownika oraz podłączenie przekaźnika.



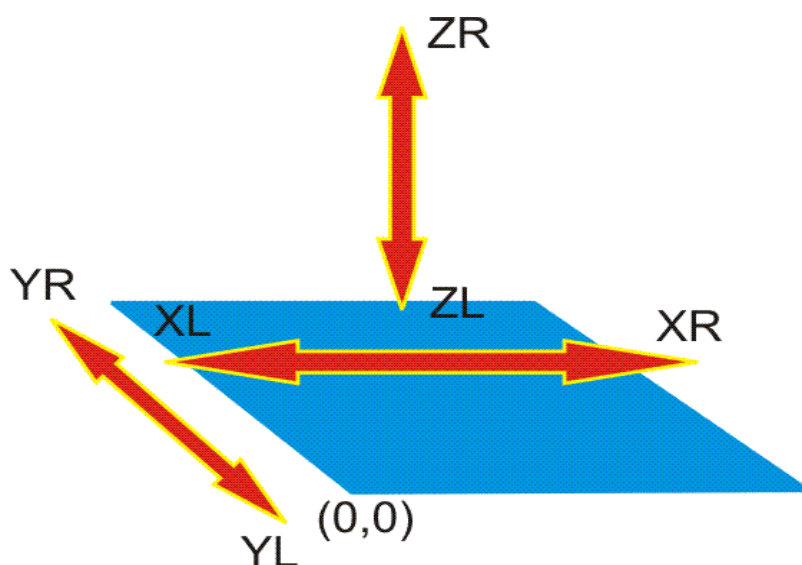
Schemat wewnętrzny wyjścia PWM.



Podłączenie krańcówek bazowania oraz czujników – wariant z czujnikiem długości narzędzia i wysokości materiału jako krańcówki.



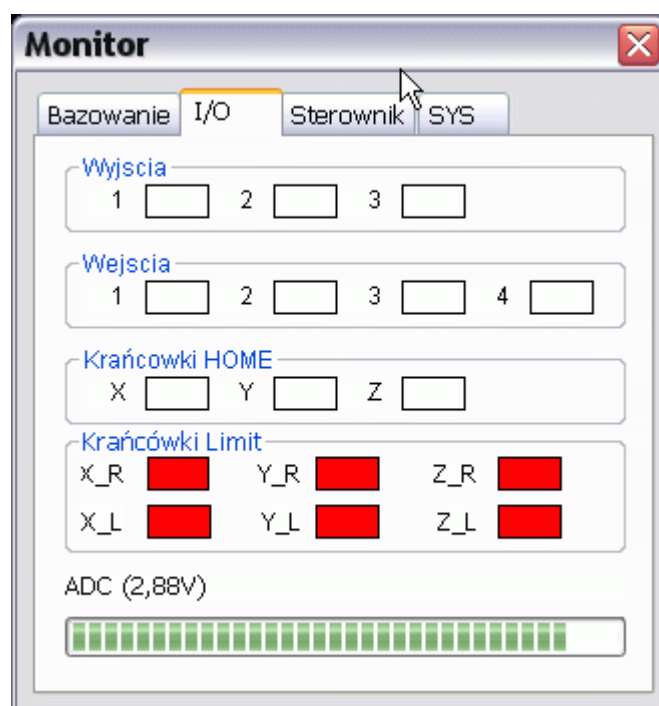
Podłączenie krańcówek bazowania oraz czujników – wariant z czujnikiem długości narzędzia i wysokości materiału jako płytki laminatu.



Rozmieszczenie krańcówek LIMIT na maszynie.

Krańcówki LIMIT muszą być tak zainstalowane aby po najechaniu na nie było jeszcze trochę miejsca (kilka mm) na ewentualne wyhamowanie silnika. Oczywiście krańcówki HOME muszą być przed krańcówkami LIMIT tzn. najpierw maszyna musi najechać na krańcówkę HOME.

Prawidłowość wszystkich sygnałów wejściowych można skontrolować w oknie „Monitor I/O”. Zaświecenie na czerwono oznacza, że styk na wejściu jest zamknięty.



Podłączenie od strony komputera.

Ze względu na bezpieczeństwo transmisji PC<->sterownik przy podłączaniu do komputera należy pamiętać o następujących kwestiach:

- Kabel USB musi być bardzo dobrej jakości (podwójny ekran, koralik ferrytowy)
- W komputerze do którego się podłączamy należy wybierać gniazda USB zainstalowane bezpośrednio na płycie głównej komputera!

Konfiguracja programu

Zakładka: „Maszyna 1/2”

Ustawienia

Maszyna 1/2 Maszyna 2/2 HOME Komunikacja Widok 1/2 Widok 2/2 G Kod

Oś X Oś Y Oś Z

Parametry osi X

Liczba kroków na obrót 400

Skok śruby 1,50

☒ Odwróć kierunek

Prędkości (mm/min)

Bezpieczna 350

MAX 900

Wymiary obszaru roboczego (mm)

Szerokość (X) 349

Długość (Y) 416

Dynamika jazdy

Wsp. przysp/ham. (procent / 1 ms) 4,00

Wsp. agresji na łukach 0,80

☒ "Ciche" dojazdy

OK

Grupa: Parametry osi X (Y,Z)

Liczba kroków na obrót: podajemy liczbę impulsów jaka przypada na pełny obrót śruby. Liczbę impulsów dla typowego silnika krokowego obliczamy według wzoru: $200 \cdot \text{podział kroku ustawiony w sterowniku silnika}$.

Skok śruby: skok gwintu śruby (w mm) – może być ułamkowy.

Odwróć kierunek: Jeżeli silnik nie kręci się w zamierzonym kierunku tutaj można to zmienić.

Nie używaj osi Z: Ta opcja związana jest tylko z osią Z. Należy ją zaznaczyć jeżeli używamy maszyny która nie korzysta z osi Z (laser, plazma etc.). Jej zaznaczenie ma następujące konsekwencje:

1. Wszystkie polecenia (G-kody) dla osi Z są ignorowane

2. Nie jest wykonywana jazda referencyjna do osi Z oraz pomiar długości narzędzia czy materiału
3. Ścieżka narzędzia jest rejestrowana gdy OUT1-ON (normalnie gdy $Z < 0$)

Bezpieczna: Jest to prędkość od której zaczyna się rampa przyspieszania i kończy rampa hamowania silnika.

Max: Jest to maksymalna prędkość do jakiej będą rozpędzane silniki jest to zarazem prędkość dojazdowa osi (G0).

Grupa: *Wymiary obszaru roboczego*

Szerokość (x), Długość (y): wpisujemy wymiary obszaru roboczego naszej maszyny (w mm)

Grupa: *Dynamika jazdy*

Zakładka ta określa parametry pracy silników umożliwiające obróbkę materiału powyżej prędkości bezpiecznej.

Wsp przyspieszenia/hamowania: Czas w jakim silnik przyspieszy (oraz wyhamuje) od prędkości bezpiecznej do maksymalnej (w procentach / 1ms).

Współczynnik agresji na łukach: Decyduje jak maszyna będzie pokonywała wszelkie zakręty (czy wyhamuje do prędkości bezpiecznej czy nie). Współczynnik ten może przyjmować wartości z zakresu 0.1 do 1. Im wyższy tym większa „agresja” w pokonywaniu łuków. Należy go dobrać eksperymentalnie do danej maszyny.

„Ciche” dojazdy: Przejazdy G0 będą wykonywane bez interpolacji liniowej.

Zakładka: „Maszyna 2/2”

Grupa: **Bazowanie**

Krańcówki zainstalowane: Informuje program, że funkcja jazdy referencyjnej ma być dostępna.

X,Y,Z: Wartości położenia osi jakie zostaną przypisane poszczególnym osiom po zakończonym bazowaniu.

Prędkość najazdu: Jest to prędkość najazdu na krańcówki HOME może to być prędkość maksymalna, gdyż właściwy pomiar dokonywany jest podczas zjazdu osi z krańcówki.

Prędkość zjazdu: Prędkość z jaką osie będą zjeżdżać z krańcówek. Musi to być wartość „rozsądna” aby pomiar był dokładny

Kierunek X,Y,Z: Informuje program w którym kierunku osie mają szukać krańcówek. Jeżeli ptaszek jest zaznaczony oś przesuwa się w kierunku rosnącym, jeżeli nie – malejącym.

Oś Z zawsze powinna być zaznaczona.

Grupa: Pomiar długości narzędzia

Czujnik zainstalowany: Informuje program, że funkcja pomiaru długości ma być dostępna.

Wysokość czujnika: Wysokość naszego czujnika (mm).

Prędkość najazdu: Prędkość najazdu na czujnik.

Grupa: Pomiar wysokości materiału

Czujnik zainstalowany: Informuje program, że funkcja pomiaru wysokości ma być dostępna.

Wysokość czujnika: Wysokość naszego czujnika (mm). Prędkość najazdu na czujnik jest taka sama jak przy pomiarze długości narzędzia.

Grupa: PWM

Wartość „S” dla 100%: Określa dla jakiej wartości parametru „S” wypełnienie przebiegu PWM będzie 100%

F nośna: Częstotliwość nośna przebiegu.

Odwróć fazę: Odwrócenie fazy przebiegu PWM.

Grupa: Wrzeciono

Czas rozpędzania (s): Czas na jaki zostanie wstrzymany program po załączeniu wrzeciona – dając tym samym czas na jego rozpędzenie.

Czas zatrzymania (s): j.w. W odniesieniu do zatrzymania wrzeciona.

Zakładka: „HOME”

Ustawienia

Maszyna 1/2 Maszyna 2/2 **HOME** Komunikacja Widok

Pozycja "HOME" (G28)

X: 186,59 Y: 405,00 Z: 56,56

Ustaw bieżącą pozycję

Pozycja "HOME" (G30)

X: 20,22 Y: 405,00 Z: 65,70

Ustaw bieżącą pozycję

OK

Grupa: Pozycja HOME 1 (G28)

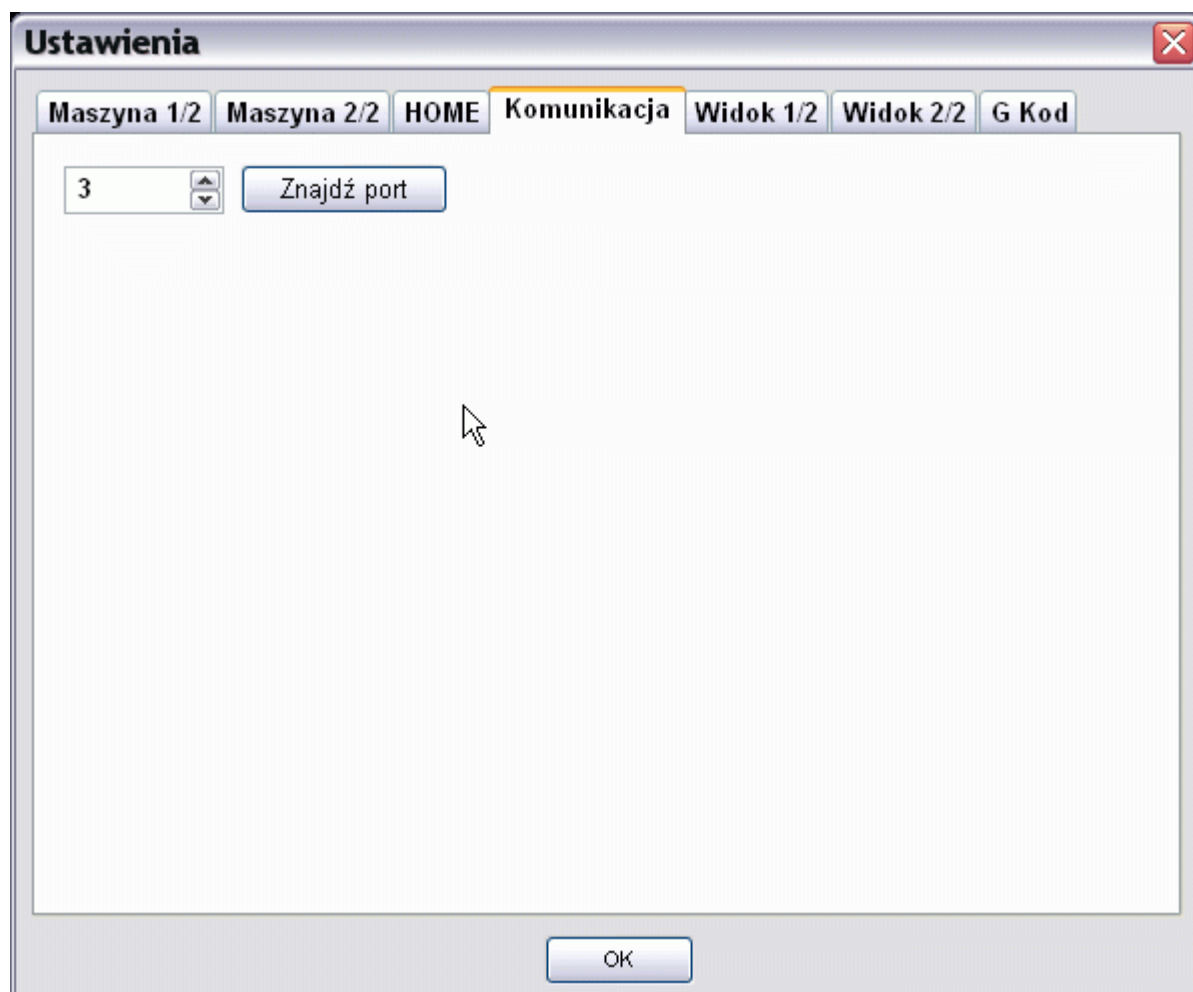
X,Y,Z: Wartości pozycji dla poszczególnych osi. **UWAGA!** Są to współrzędne maszynowe!

Ustaw bieżącą pozycję: Nadaje X,Y,Z aktualne położenie osi.

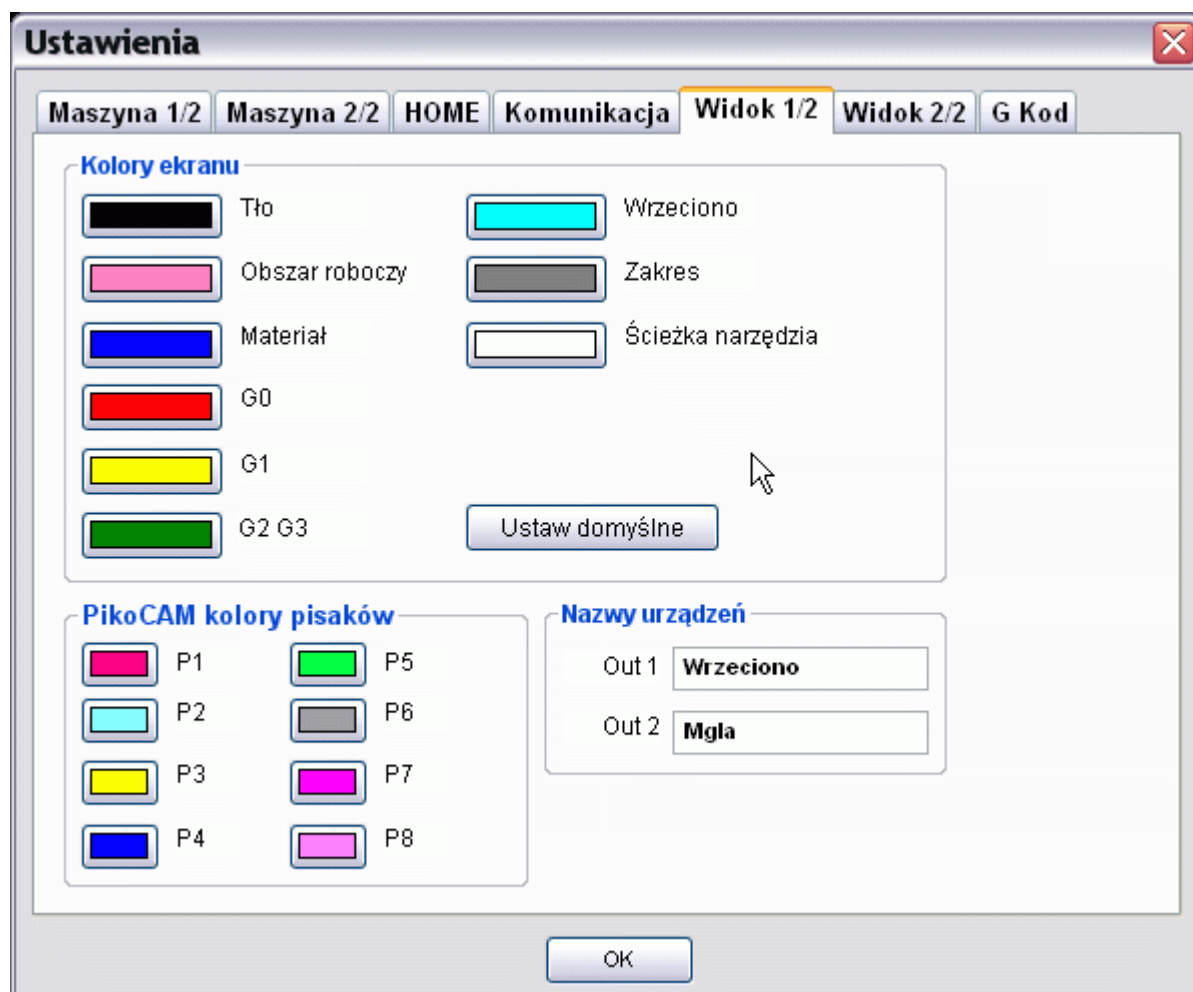
Grupa: Pozycja HOME 2 (G30)

X,Y,Z: Wartości pozycji dla poszczególnych osi. **UWAGA!** Są to współrzędne maszynowe!

Ustaw bieżącą pozycję: Nadaje X,Y,Z aktualne położenie osi.

Zakładka „Komunikacja”

Po zainstalowaniu sterowników i podłączeniu sterownika PikoCNC do USB należy wejść w tą zakładkę i kliknąć na „Znajdź port”. Jeżeli nie pojawi się żaden komunikat z błędem, to znaczy, że z komunikacją nie powinno być problemów.

Zakładka: „Widok”**Grupa: Kolory ekranu**

Umożliwia ustawienie kolorów dla poszczególnych elementów ekranu.

Grupa: PikoCAM kolory pisaków

Umożliwia ustawienie domyślnych kolorów dla PikoCAM-a.

Grupa: Nazwy urządzeń

Umożliwia nadanie nazwy poszczególnym urządzeniom.

Zakładka „G kod”

Numeracja linii, Skok numeracji: decydują czy linie G-kodu będą numerowane i z jakim skokiem.

Grupa: **Precyzja**

DOKŁADNOŚĆ G1: występujące w programie wektory krótsze niż wartość tutaj wpisana są łączone.

DOKŁADNOŚĆ G2 G3: łuki oparte o te komendy będą budowane z odcinka, którego długość tutaj wpisujemy. Z parametrem tym nie można przesadzać (nie może być zbyt mały) więcej informacji w rozdziale „**Panel czasu i postępu / Bufor**”.

Grupa: **Otwieranie**

Centruj automatycznie pod frezem: Wczytane programy ustawiane są tak aby ich środek znajdował się „pod frezem”.

Grupa: *Zmiana narzędzia (M6)*

Decydujemy jaka ma być reakcja programu na kod M6. Jeżeli wybierzemy „Ręczna zmiana” wtedy maszyna wykonuje następujący cykl:

- uniesienie osi „Z” na wysokość zdefiniowaną dla HOME1
- przejazd do punktu „XY” HOME1 (pozycja HOME1 staje się punktem wymiany narzędzia)
- wyświetlenie komunikatu o konieczności wymiany narzędzia i oczekiwanie na potwierdzenie.
- Kiedy potwierdzenie nastąpi wykonywany jest kompletny cykl bazowania wraz z pomiarem długości narzędzia.
- Po zakończeniu pomiaru automatyczny powrót do wykonywania dalszej części programu.

Uwagi: W oknie oczekiwania na potwierdzenie zmiany wyświetlana jest linia G-kodu w której znajduje się komenda M6 dlatego dobrym zwyczajem jest w komentarzu napisać o jaki frez chodzi np:

N1000 M6 (frez 4mm

Grupa: *Zakończenie programu*

Decyduje o sekwencji po zakończeniu programu. Jest ona wykonywana także gdy wykonywany jest tylko fragment programu. „Safe Z” to wysokość bezpieczna ustawiona w polu Materiał/Parametry dojazdowe/Wysokość bezpieczna.

Pierwsze uruchomienie

Po podłączeniu sterownika, zainstalowaniu sterowników, oraz wstępnej konfiguracji programu można pokusić się na pierwsze uruchomienie programu. Aby nawiązać połączenie ze sterownikiem należy kliknąć „połącz” w „Program/START”

UWAGA !!! Program musi być uruchamiany z konta administratora – inaczej komunikacja z kontrolerem nie będzie działać !!!

Dioda stanu na sterowniku powinna przestać mrugać aktywny robi się przycisk „RESET” - kliknięcie na niego powoduje załączenie sygnału ENABLE na sterowniki i umożliwienie pracy z maszyną (nawet jeżeli tego sygnału nie używamy w maszynie). Teraz wykonując ruchy z klawiatury poszczególnymi osiami można dopracować dynamiczne parametry pracy maszyny.

W oknie monitora można sprawdzić prawidłowość sygnałów wejściowych z krańcówek, przycisków. Należy sprawdzić też prawidłowość załączania urządzeń wyjściowych.

Pierwsze próby należy wykonywać bez założonego narzędzia a nawet wrzeciona, z ręką na E_Stop-ie !

Ekran główny

Grupa: **Położenie**

Położenie (mm)

X -25,352 ZERO

Y 36,200 ZERO

Z 44,704 ZERO

☐ Maszyna USTAW PM

X,Y,Z: Aktualne położenie osi x,y,z (zależnie od kontekstu).

Zero: Zerowanie poszczególnych osi (zależnie od kontekstu).

Ustaw: Otwiera okienko gdzie poszczególnym osiom można ręcznie nadać dowolne wartości.

Ustaw XYZ (material)

X 49,021

Y 71,483

Z -0,802

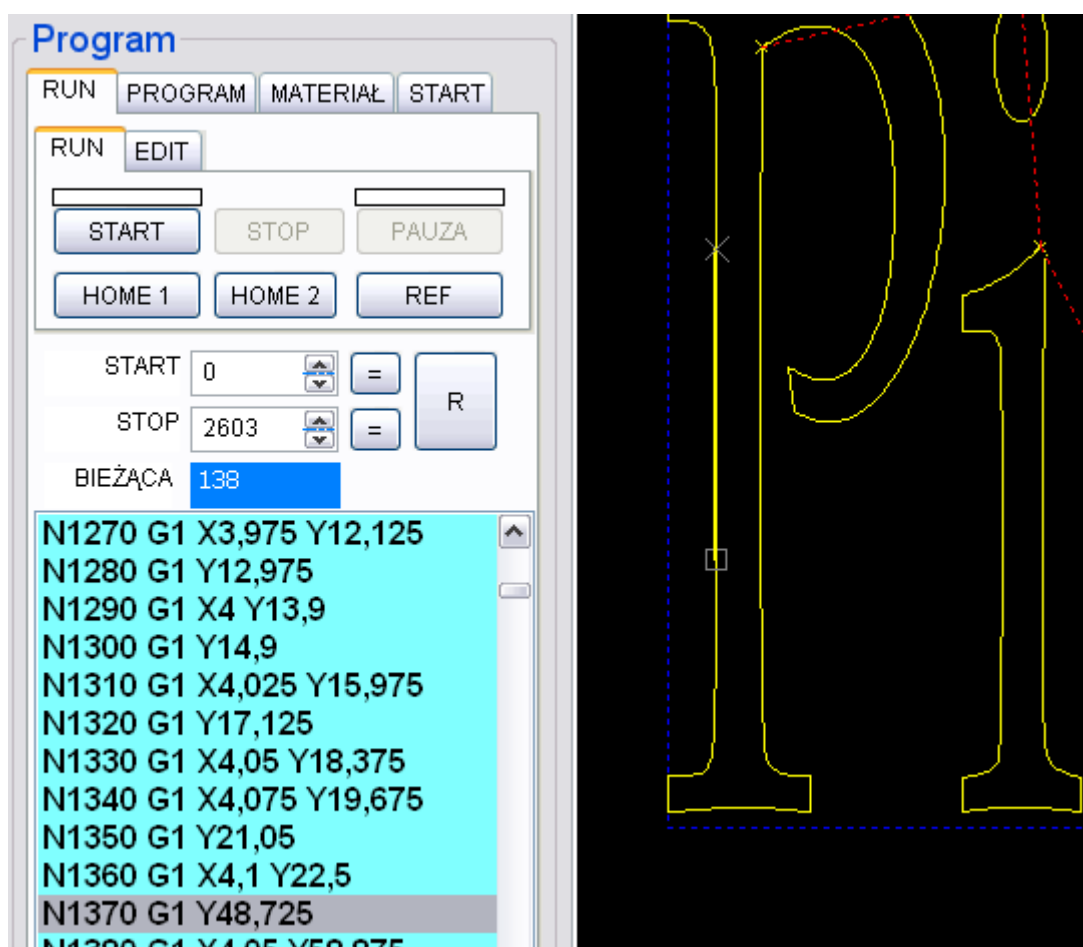
OK Anuluj

PM: inicjuje cykl pomiaru wysokości materiału. (oś Z zjeżdża w dół z prędkością ustawioną w [Pomiar długości narzędzia/Prędkość najazdu](#). Po dojechaniu do czujnika od aktualnej pozycji osi Z odejmowana jest jego grubość i punkt ten ustalany jest jako Z=0 materiału. Po pomiarze oś Z podnosi się 10mm nad czujnik.)

Maszyna: Normalnie wyświetlane są współrzędne materiałowe. Załączenie tego przycisku przełącza na wyświetlanie współrzędnych maszynowych. Wszystkie operacje (zerowanie, ustawianie wartości dotyczą teraz współrzędnych maszynowych. W tym trybie 'miga' kontrolka z lewej strony przycisku oraz zmienia się kolorystyka wyświetlacza.

Grupa: *Program*

Zakładka: „*RUN/RUN*”



START: Uruchomienie wczytanego programu. Program zawsze wykonywany jest od linii określonej w okienku „START PROGRAMU” a kończy się na linii określonej w „KONIEC PROGRAMU”. Cykl wykonywania programu zawsze zaczyna się od uniesienia Z na wysokość bezpieczną a następnie przejazdu na pozycję XY obróbki.

STOP: Zatrzymanie programu.

PAUZA: Inicjuje cykl pauzy. Oś unoszona jest na wysokość bezpieczną a następnie wszystkie urządzenia wyjściowe są wyłączane. W czasie stanu „Pauza” można używać posuwu ręcznego, można dokonać jazdy referencyjnej, przejechać do HOME. Powtórne naciśnięcie pauzy inicjuje cykl powrotu do obróbki czyli: oś Z na wysokość bezpieczną, następnie dojazd do XY początku wektora, którego wykonywanie przerwano, załączenie urządzeń które pracowały, zjazd do pozycji Z obróbki i kontynuacja programu.

HOME1: Inicjuje cykl przejazdu do pozycji HOME1. Najpierw oś Z, następnie XY. Ten sam efekt można uzyskać używając w programie kodu G28

HOME2: Inicjuje cykl przejazdu do pozycji HOME2. Najpierw oś Z, następnie XY. Ten sam efekt można uzyskać używając w programie kodu G30

REF: Inicjuje cykl jazdy referencyjnej czyli bazowania maszyny. Cykl przebiega w następujący sposób:

1. oś Z do górnej HOME_Z
2. kiedy Z osiągnie cel - osie XY do krańcówek HOME_X HOME_Y (jednocześnie)
3. kiedy XY osiągną cel – osie XYZ zjeżdżają z krańcówek (jednocześnie)
4. Nadanie wartości osiom (z ustawień).
5. Zjazd po 10mm wszystkich osi (jednocześnie) – chodzi o to aby na bezpieczną odległość oddalić się od dźwigni krańcówek.
6. Oś Z w dół do czujnika długości narzędzia. (o ile jest dostępny).
7. Po osiągnięciu czujnika oś Z (współrzędne maszynowe) przyjmuje wartość wysokości czujnika.
8. Oś Z w górę 10mm – oddalenie się od czujnika.

START PROGRAMU: Linia od której wykonywany będzie program. **UWAGA!** Jest to numer linii tekstowej na liście poniżej a nie według numeracji linii G-kodu. Na ekranie obszary wyłączane z obróbki zaznaczane są (domyślnie) kolorem szarym.

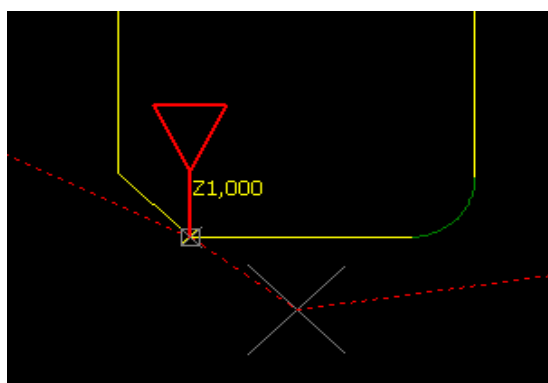
KONIEC PROGRAMU: j.w. Tylko koniec programu

BIEŻĄCA LINIA: Bieżąca linia wskazywana na liście.

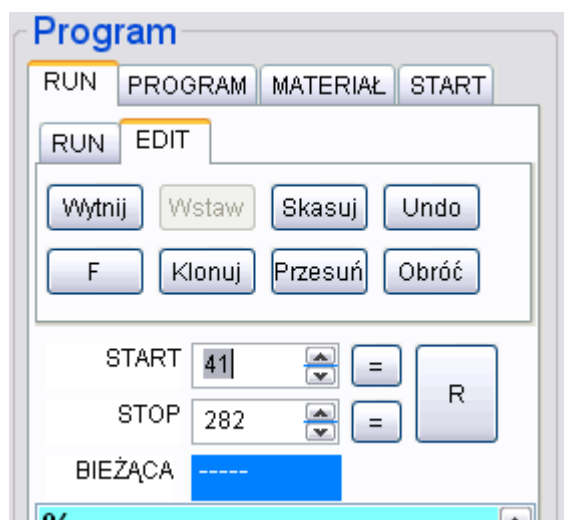
„=”: nadaje wartość wskazywaną przez bieżącą linie na liście.

„R”: Resetuje pola początku i końca tak aby wskazywały cały program.

„LISTA PROGRAMU”: Podświetlenie linii na liście powoduje podświetlenie wektora, któremu ta linia odpowiada. Początek wektora oznaczony jest kwadracikiem, koniec krzyżykiem. W przypadku gdy linia dotyczy zmiany pozycji w osi „Z” wyświetlana jest strzałka, której kierunek wskazuje kierunek ruchu, oraz informacja o docelowej pozycji.



Zakładka RUN/EDIT



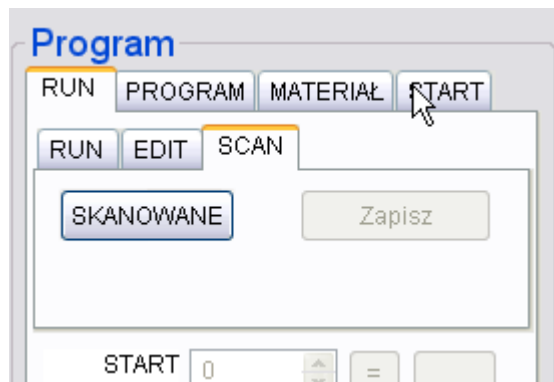
Zakładka ta służy do wykonywania prostych operacji na wczytanym G-kodzie.

Wszystkie przyciski (oprócz F) są normalnie wyłączone a aktywują się wtedy gdy użytkownik wybierze zakres programu, który chce przetworzyć. Zakres musi spełniać pewne kryteria początek jak i koniec musi wskazywać na G0 XY czyli przejazd w płaszczyźnie XY. Zakres można ręcznie wpisać w pola „START” „STOP” lub znacznie wygodniej trzymając przyciśnięty klawisz **CTRL** kliknąć na wektory końca i początku na ekranie (w dowolnej kolejności). Wybrany zakres zostanie na ekranie podświetlony.

Działanie większości funkcji nie wymaga chyba komentarza.

Funkcja „F” pozwala zmienić prędkość wykonywania danego fragmentu programu lub jeśli nic nie zaznaczymy to całego programu.

Wszystkie powyższe funkcje działają także jeżeli aktualny G-kod generowany jest przez PikoCam-a jednak należy pamiętać, że ponowne otwarcie PikoCama spowoduje utratę wprowadzonych zmian o czym przypomni stosowny komunikat.

Zakładka RUN/SCAN

Umożliwia rozpoczęcie procesu skanowania. Szczegóły w rozdziale „Skanowanie”.

Uwaga! Przycisk robi się aktywny dopiero gdy maszyna jest w stanie „RUN”.

Panel czasu i postępu



Czas: Czas wykonywania się obróbki (wliczając w to ewentualne „pauzy”)

Postęp: Stopień zaawansowania pracy.

Bufor: Stopień zapelnienia wewnętrznego bufora sterownika. Należy się tutaj trochę wyjaśnienia. Poziom zapelnienia bufora w czasie obróbki nie powinien spadać nigdy do zera gdyż spowalnia to maszynę i powoduje, że nie pracuje płynnie. Zbyt szybkie schodzenie bufora może mieć tylko jedną przyczynę – program generuje zbyt krótkie wektory przy zbyt dużej prędkości obróbki. Aby temu zaradzić należy zwiększyć wartości w polach „Dokładność G1 G2 G3” zakładki „PROGRAM”. Należy mieć na uwadze, że program może wysłać do sterownika maksymalnie 1000 wektorów na sekundę. Tak więc jeżeli np. Założymy że używamy wyłącznie wektorów o długości 0,01mm zatem maksymalna prędkość maszyny (ograniczona komunikacją) będzie wynosić $1000 \cdot 0,01 \cdot 60 = 600 \text{ mm/min}$ a już np. dla wektorów o długości 0,05 będzie to 3000mm/min

Zakładka: „PROGRAM”

OTWÓRZ: Otwieranie programów typu G-kod ,HPGL lub DXF zależnie od tego jak ustawimy filtr w oknie wyboru pliku.

AKTUALIZUJ: Powtórnie ładuje aktualnie otwarty plik (np. Po jego edycji)

ZAPISZ CNC: Umożliwia zapisanie aktualnego program jak G-kodu. Przycisk robi się aktywny jeżeli nawiążemy choć raz komunikację ze sterownikiem.

EDYTUJ: Otwiera aktualnie otwarty program w systemowym notatniku.

Ustaw pod frezem: Przesuwa program tak aby jego środek znajdował się dokładnie w aktualnym położeniu narzędzia.

DOMYŚLNE: Wszystkie manipulacje na programie są resetowane.

PikoCAM: W przypadku wybrania pliku HPGL lub DXF otwiera okno, w którym można ustalić szczegółowe parametry obróbki. (patrz dział PikoCAM). W przypadku otwarcia pliku BMP otwiera okno konwersji plików BMP.

Zakładka: „**MATERIAŁ**”

Rozmiar X,Y,Z: Wpisujemy tutaj wymiary naszego materiału. Warto to zrobić gdyż ma to szereg konsekwencji:

1. widzimy orientację materiału względem stołu oraz programu względem materiału.
2. Ścieżka narzędzia rejestrowana jest tylko w obrębie materiału.
3. Program w osi Z nie zejdzie niżej niż wpisana grubość materiału.

Grupa: **Parametry dojazdowe START/PAUZA**

Wysokość bezpieczna: Wysokość używana w czasie cyklu rozpoczęcia programu, pauzy.

Wysokość dojazdowa: j.w.

Prędkość schodzenia: Prędkość schodzenia osi Z poniżej wysokości dojazdowej.

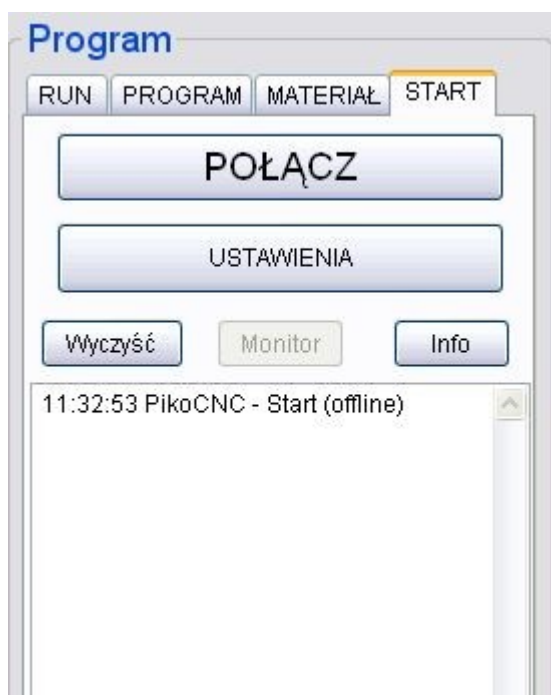
Grupa: *Rejestrowanie ścieżki narzędzia*

OTWÓRZ: Otwiera istniejący już materiał z zarejestrowaną ścieżką, bądź tworzony jest nowy jeżeli wpiszemy nazwę pliku, którego nie ma na liście. Jeżeli otworzymy rejestrację pliku nie można już ścieżki wyczyścić ręcznie do póki rejestracji nie zamkniemy.

ZAKOŃCZ: kończy rejestrację ścieżki.

Grupa: *Kontrola urządzeń*

Można tutaj ustalić w jaki sposób sterowane są urządzenia wyjściowe.

Zakładka: „START”

POŁĄCZ/ROZŁĄCZ : Rozpoczęcie/zakończenie komunikacji ze sterownikiem. Każdą sesję pracy z maszyną musimy rozpocząć/zakończyć od kliknięcia na „połącz/rozłącz”.

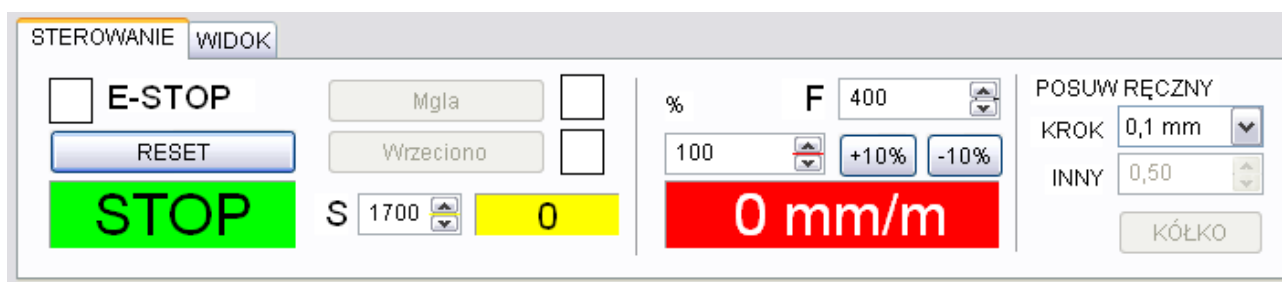
USTAWIENIA: Otwiera okno konfiguracji sterownika. Przycisk jest aktywny tylko wtedy gdy sterownik jest w trybie OFFLINE.

Wyczyść: czyszczenie zawartości okna LOG.

Monitor: otwiera okno monitora.

Info: O programie, autorze.

Grupa: Sterowanie



RESET: Aby rozpocząć pracę maszyny (lub wznowić po wystąpieniu E-STOP) musimy nacisnąć ten przycisk. (działa identycznie jak w Mach-u)

E-STOP: Kontrolka wystąpienia stanu E-STOP. Sygnał ten może wystąpić w trzech sytuacjach:

- naciśniemy przycisk E_STOP
- w czasie wykonywania programu maszyna najedzie na którąś z krańcówek
- wykonując posuw ręczny o zadany krok (z klawiatury lub kółkiem) maszyna najedzie na którąś z krańcówek – nie dotyczy to posuwu ciągłego (z przyciśniętym klawiszem shift lub Ctrl - wtedy krańcówki uniemożliwiają jedynie wykonanie ruchu w zabronionym kierunku)

Mgła/Wrzeciono: umożliwia ręczną kontrolę nad urządzeniami – o ile zezwoliliśmy na to. (patrz Grupa: [Kontrola urządzeń](#)).

S: Umożliwia ręczną kontrolę nad parametrem „S” czyli w wypadku naszego sterownika – kontrolą wypełnienia przebiegu na wyjściu PWM.

F: Ustalenie domyślnej prędkości obróbki. Wartość ta jest używana jeżeli nie jest ustalona w programie.

%: Ustalamy tutaj procentowe przyspieszenie/spowolnienie prędkości obróbki. Jeżeli zmian prędkości dokonujemy w czasie biegu programu, należy mieć świadomość, że zmiany prędkości będą widoczne z pewnym opóźnieniem. (a to ze względu na bufor sterownika)

Posuw ręczny: skok dla „ręcznego” poruszania osiami. (patrz rozdział „klawiatura”)

KÓŁKO: otwierane jest specjalne okno, które umożliwia ręczne poruszanie osiami maszyny za pomocą myszki (kółka). Lewym i prawym klawiszem myszki wybieramy oś którą chcemy poruszać a kręcąc kółkiem myszki góra-dół przesuwamy oś. Skok jest taki jak ustalony w okienku „Krok”. **Kursor myszy musi być cały czas w obrębie tego okna!**



Grupa: Widok
Zakres

MATERIAŁ/PRACA/STÓŁ/WSZYSTKO: ustalmy jaki widok chcemy mieć na ekranie. Tryb „**WSZYSTKO**” przydatny jest wtedy, gdy współrzędne naszego programu wybiegają poza obszar stołu.

Perspektywa

Wybór perspektywy.

Elementy

G0: decyduje czy przejazdy G0 mają być widoczne.

Ścieżka: decyduje czy ścieżka narzędzia ma być widoczna.

iZ: decyduje czy zmiany w osi Z mają być widoczne.

Materiał: decyduje czy obrys materiału ma być widoczny.

Stół: decyduje czy obrys obszaru roboczego ma być widoczny.

Narzędzie: decyduje czy kursor narzędzia będzie widoczny.

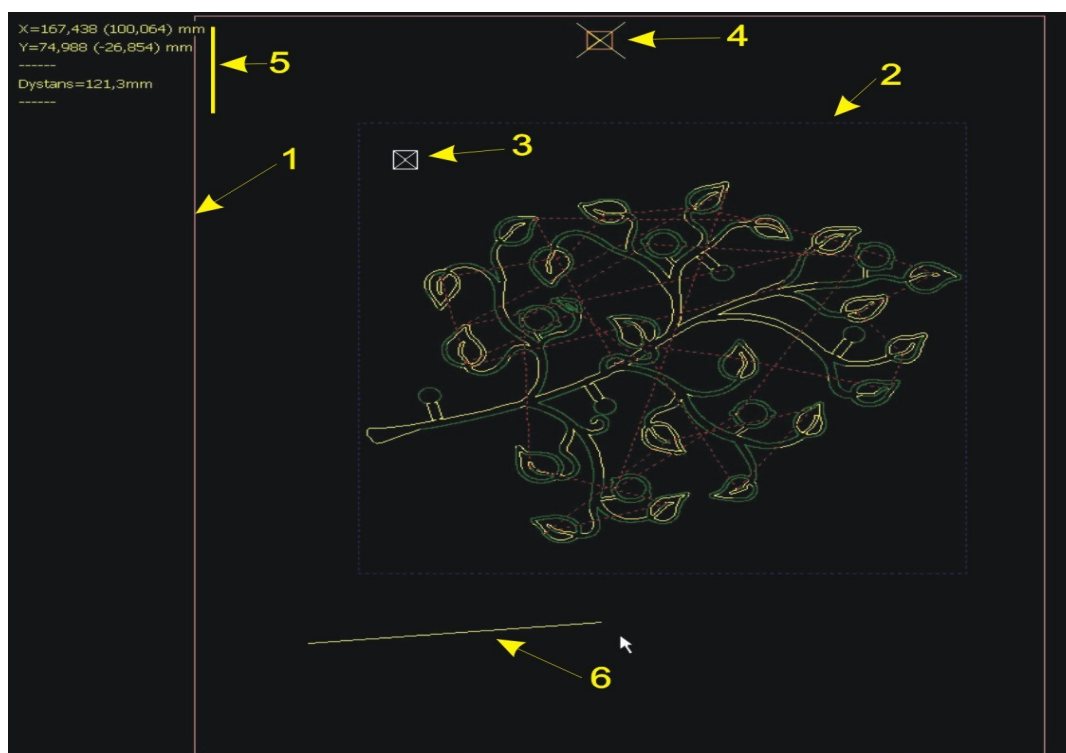
(F): włącza podgląd parametryzowany w funkcji F (prędkości obróbki). Wektory na ekranie rysowane są kolorem zależnym od prędkości (F) jego wykonywania. Dynamikę zmian koloru można ustawić w Ustawienia / Widok 2/2.

Ścieżka narzędzia

Czyść automatycznie: Czyści ścieżkę narzędzia przy każdym rozpoczęciu obróbki.

Wyczyść: Czyści ścieżkę narzędzia. W przypadku gdy jesteśmy w trybie rejestracji ścieżki - ręczne kasowanie ścieżki jest niemożliwe.

Ekran wykresu



1. Granice obszaru roboczego maszyny.

2. Granice materiału.

3. Pozycja HOME1.

4. Pozycja HOME2 (z zaznaczoną pozycją wrzeciona)

5. Informacje:

linia 1: współrzędne X położenia myszki (maszynowe i materiałowe).

linia 2: współrzędne Y położenia myszki (maszynowe i materiałowe).

linia 3: treść aktualnie wykonywanej (lub wskazanej myszką) linii programu. Ponieważ niekiedy cała linia programu nie mieści się na liście tutaj można ją zobaczyć w całości.

linia 4: „Miarka”. Służy do pomiaru odległości na stole. Naciskamy lewy klawisz myszy i przesuwamy wskaźnik myszy w punkt docelowy. Miarka pokazuje odległość od punktu początkowego do docelowego. Miarka służy też do wygodnego przesuwania programu względem materiału. Należy postępować jak wyżej tylko najpierw trzeba przycisnąć klawisz Shift.

linia 5: Aktualna wartość „Z”. Wskazując na liście jakąś linię programu niekiedy trudno się zorientować jakiej pozycji „Z” dotyczy. Tutaj można to zobaczyć.

6. Zaznaczona „miarka”

Interpretacja G-Kodu

Znak „%”

Każdy program (komendy dla maszyny) powinien rozpoczynać się od znaku „%”. Wszystkie linie, które występują w programie przed tym znakiem nie są interpretowane i są traktowane jako komentarz !!. Tak więc jeśli po wczytaniu jakiegoś programu na ekranie nic się nie pojawi - to znak, że brak „%” i trzeba go dopisać.

Np:

```

”
Krótki program          // linia bez znaczenia !
Frez 2mm                 // linia bez znaczenia !
G0 X30 Y30               // linia bez znaczenia !
%                          // zaczyna się program !!
G0 X10 Y10
G0 Z-1
G1 X20 Y20
”

```

Komentarze

Znaki występujące w linii po „ („ lub „ ’ ” (apostrof) lub „ ; ” są traktowane jako komentarz.

Np:

```

( Frez 2mm      - komentarz
' Frez 2mm      - komentarz
; Frez 2mm      - komentarz
G1 X20.6 Y36    (koniec

```

Zapis G-kodu.

Nie ma znaczenia wielkość liter, spacje, kropka czy przecinek, czy też sposób zapisu numeru G-kodu. Wszystkie linie poniżej są sobie równoważne:

```

G1 X45.5 Y30.2
G01 x45,5 Y 30,2
g1x45.5y30.2

```

Numeracja linii.

Numeracja jest całkowicie ignorowana przez program i jest obojętne czy jest czy jej nie ma. Program jest wykonywany według kolejności występowania komend w programie.

Obsługiwane polecenia

G0 (X,Y,Z) – Dojazdy. Ruch wykonywany zawsze z prędkością maksymalną.

G1 (X,Y,Z) – Ruch roboczy po prostej. Wykonywany z prędkością ustawioną parametrem F

G2 (X,Y,I,J,R) – Ruch roboczy po łuku. Kreśli łuk (zgodnie z kierunkiem zegara) od aktualnej pozycji do X,Y o środka o współrzędnych X+I,Y+J lub o promieniu R.

G3 (X,Y,I,J,R) – Ruch roboczy po łuku. Kreśli łuk (przeciwnie do kierunku zegara) od aktualnej pozycji do X,Y o środka o współrzędnych X+I,Y+J lub o promieniu R.

G4 (P) Czasowe zatrzymanie programu. Czas podawany w sekundach. Np G4 P0.5 zatrzyma wykonywanie programu na 0.5 sek.

G20 Program w systemie calowym.

G21 Program w systemie metrycznym.

G28 Przejazd do pozycji HOME1 (najpierw oś Z potem XY)

G30 Przejazd do pozycji HOME2 (najpierw oś Z potem XY)

G98 Ustalenie R-planu czyli wysokości przejazdowej dla cyklu G80.

G81 Podstawowy cykl wiercenia.

G0 Z2.

G98 (ustalenie aktualnego Z jako R-planu)

G81 X40 Y100 Z-3.2 R1. F150

X50 Y100

X60 Y110

X70 Y120

W wyniku działania tego programu zostaną wywiercone cztery otwory o głębokości 3.2mm. W czasie przejazdów frez będzie się unosił 2mm nad materiałem. Frez w czasie wyjazdu z otworu do wysokości 1mm będzie poruszał się z prędkością roboczą (tutaj F150) - powyżej z dojazdową.

G80 Odwołanie cyklu wiercenia.

Obsługiwane M-kody

M0 – Program stop. Program zostanie zatrzymany w ten sposób jakby naciśnięto „Pauze”. Po ponownym naciśnięciu pauzy program jest wznowiany.

M3/M4 – Załączenia wyjścia nr 1.

M5 – Wyłączenie wyjścia nr 1.

M6 – Zmiana narzędzia.

M8 - Załączenia wyjścia nr 2.

M9 – Wyłączenie wyjścia nr 2.

M10 - Załączenia wyjścia nr 3.

M11 – Wyłączenie wyjścia nr 3.

M30 – Zakończenie wykonywania programu

Pozostałe parametry:

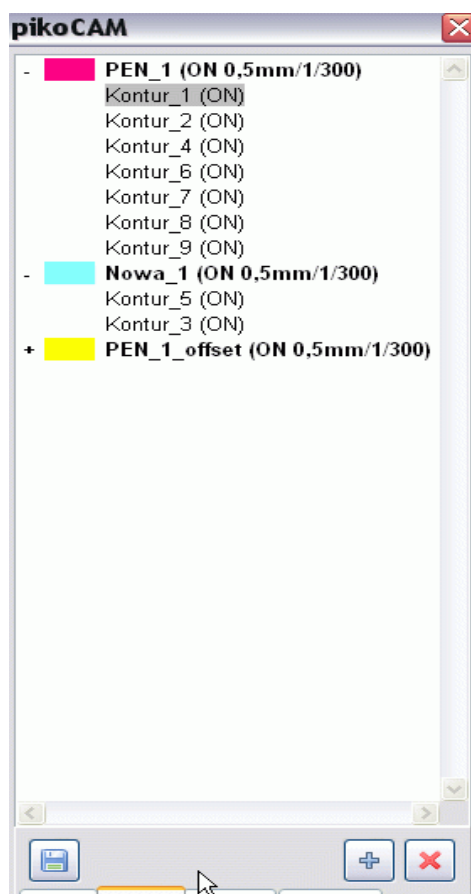
F - określa prędkość dla ruchu roboczego

S - określa wypełnienie przebiegu PWM

PikoCAM

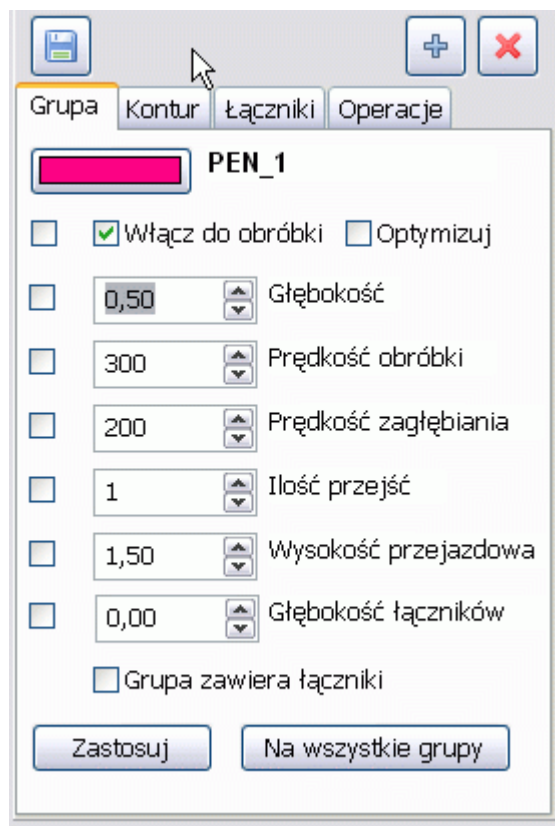
Po otwarciu pliku w formacie HPGL lub DXF w zakładce „Program” aktywny robi się przycisk PikoCAM, który otwiera okno gdzie można ustalić parametry obróbki.

Lista



Tutaj można wybrać grupę lub pojedynczy kontur, którego parametry obróbki chcemy zmienić. Elementy na liście można przesuwając metodą drag&drop zmieniając tym samym kolejność obróbki, a także usuwać (czerwony krzyżyk u dołu). Przycisk "+" służy do tworzenia nowych grup. Przycisk z ikoną dyskietki umożliwia zapisanie danych PikoCam-a na dysk.

Grupa



Umożliwia ustalenie parametrów obróbki dla wybranej grupy. Większość parametrów nie wymaga chyba komentarza.

Optymizuj: zmienia kolejność konturów w grupie tak aby przejazdy były jak najkrótsze.

Grupa zawiera łączniki: jeżeli grupa zawiera informację o łącznikach musimy tutaj zaznaczyć.

Głębokość łącników: jeżeli stosujemy w programie łączniki tutaj decydujemy jaką mają mieć głębokość dla wybranej grupy.

Aby zatwierdzić zmiany na końcu zawsze trzeba nacisnąć „Zastosuj”. Jeżeli chcemy aby te parametry dotyczyły wszystkich grup musimy nacisnąć „Na wszystkie grupy” - ale uwaga: na wszystkie grupy skopiowane zostaną tylko te parametry, które oznaczymy „ptaszkiem” po lewej stronie!

Kontur

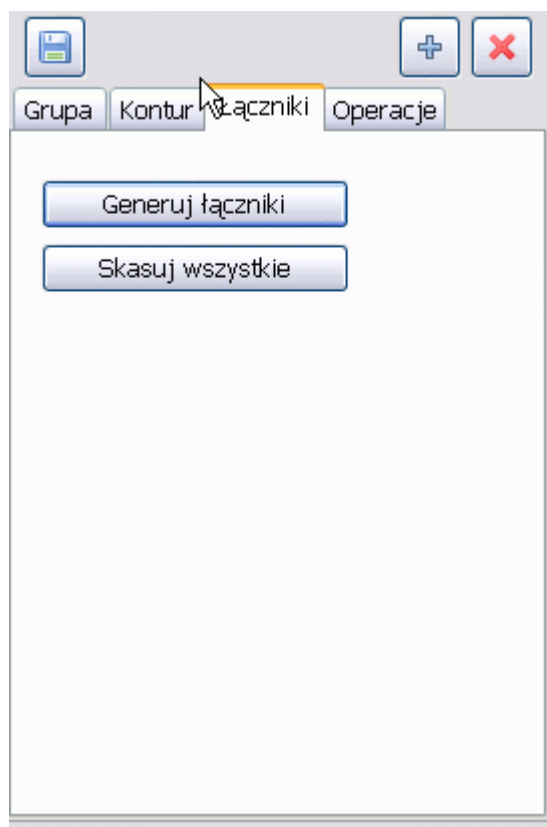
Umożliwia ustalenie parametrów obróbki dla wybranego konturu. Większość parametrów nie wymaga komentarza.

Zablokuj: jeżeli chcemy kontur zabezpieczyć przed globalnymi zmianami w zakładce „Grupa” to musimy tutaj zaznaczyć.

Offset/Wybieranie: Służy do generowania ścieżki offsetu i wybierania. Aby dało się to zrobić kontur musi być zamknięty oraz nie może zawierać wewnętrznych pętli (nie może przecinać się sam ze sobą). Dla konturów zewnętrznych możliwe jest tylko generowanie offsetu a dla wewnętrznych offsetu oraz wybierania. Przycisk „KONTUR ...” pokazuje z jakim konturem mamy aktualnie do czynienia. Jeżeli chcemy to zmienić musimy na niego kliknąć. Wybieranie na razie nie obsługuje wysp.

Cała grupa: Offset/wybieranie zostanie wygenerowane dla całej grupy zgodnie z aktualnymi kierunkami poszczególnych konturów.

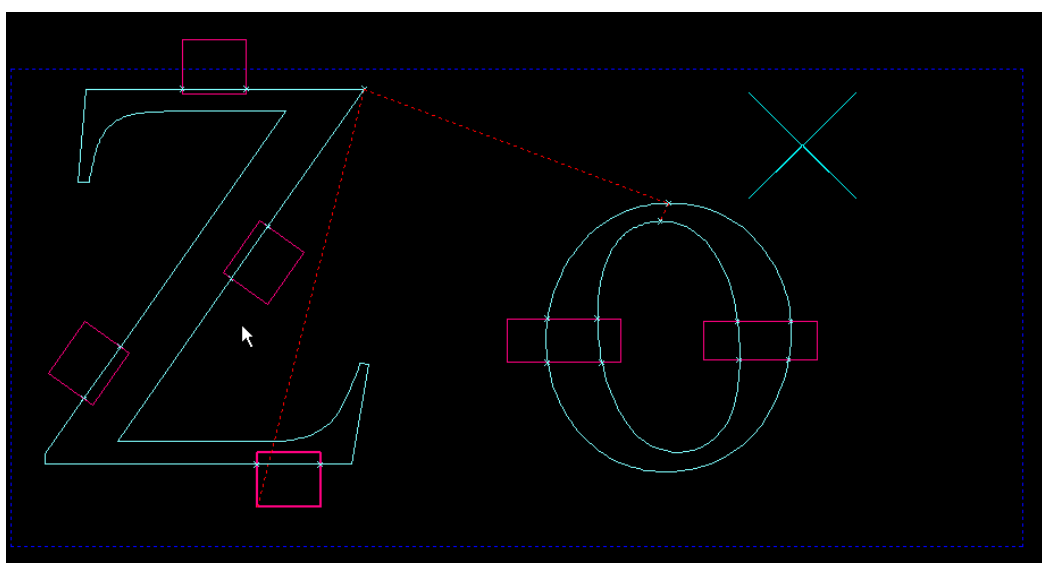
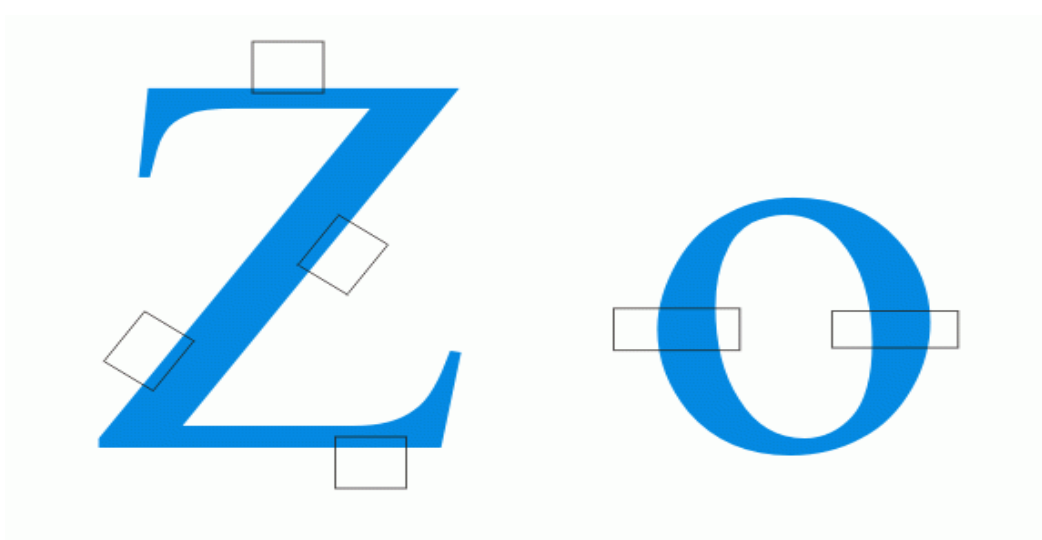
Aby zatwierdzić zmiany na końcu zawsze trzeba nacisnąć „Zastosuj”.

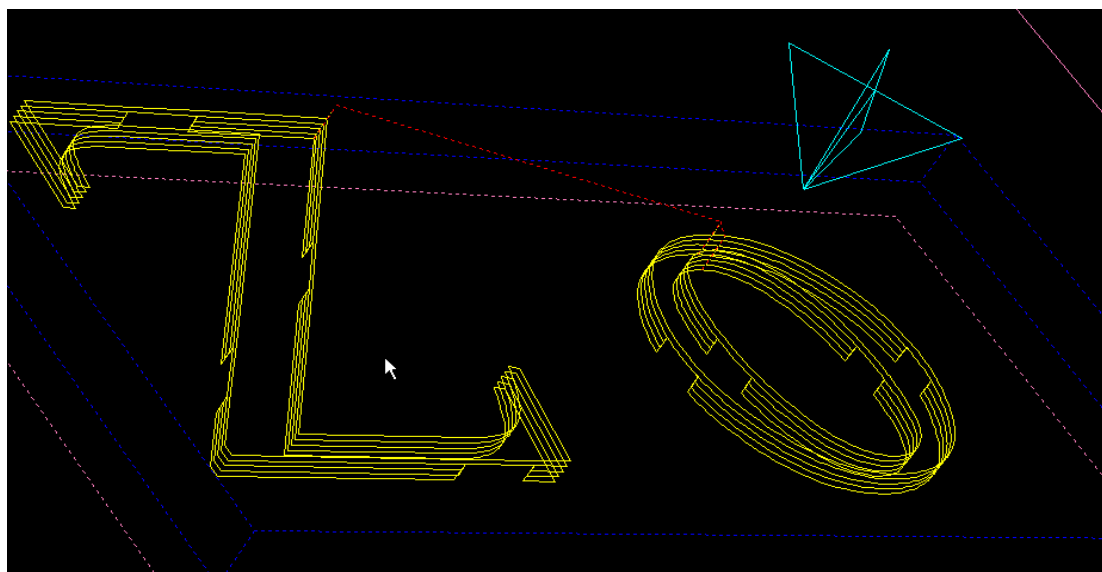
Łączniki

Jeżeli w którejś grupie zazaczyliśmy „zawiera łączniki” tutaj należy kliknąć na „Generuj łączniki” aby je stworzyć.

Tworzenie łączników

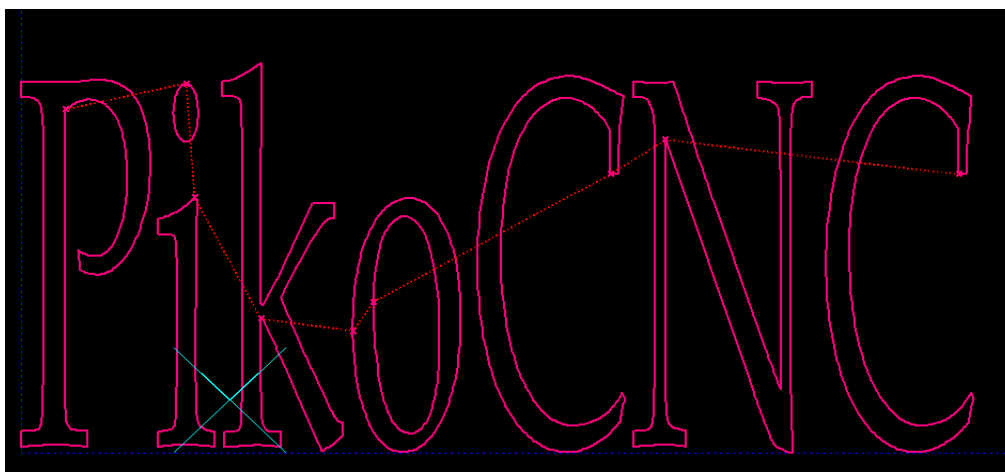
Program posiada możliwość tworzenia łączników – czyli fragmentów „niedociętych” aby wycinany element trzymał się nadal z materiałem. Jak robi się takie łączniki – po prostu w programie w którym tworzymy wzór do wycięcia rysujemy także łączniki rezerwując na ten cel jedno pióro (HPGL) lub warstwę (DXF). W pierwszym miejscu gdzie pióro łącznika przecina kontur zaczyna się łącznik - W następnym miejscu gdzie pióro łącznika przecina kontur kończy się łącznik – stąd prosty wniosek, że liczba przecięć musi być parzysta a jedna linia przecinająca wykorzystywana jest w jednym konturze tylko raz! Jest też warunek, że łącznik nie może przecinać łuków (G2 G3). Kiedy projekt już gotowy wczytujemy go do PikoCam-a. W grupie która zawiera łączniki zaznaczamy „**Grupa zawiera łączniki**” a w pozostałych grupach w polu „**Głębokość łączników**” wpisujemy docelową głębokość łączników. Na koniec w grupie „**Łączniki**” klikamy na „**Generuj łączniki**” - jeżeli nie zmieniamy położenia łączników wystarczy to zrobić tylko raz.



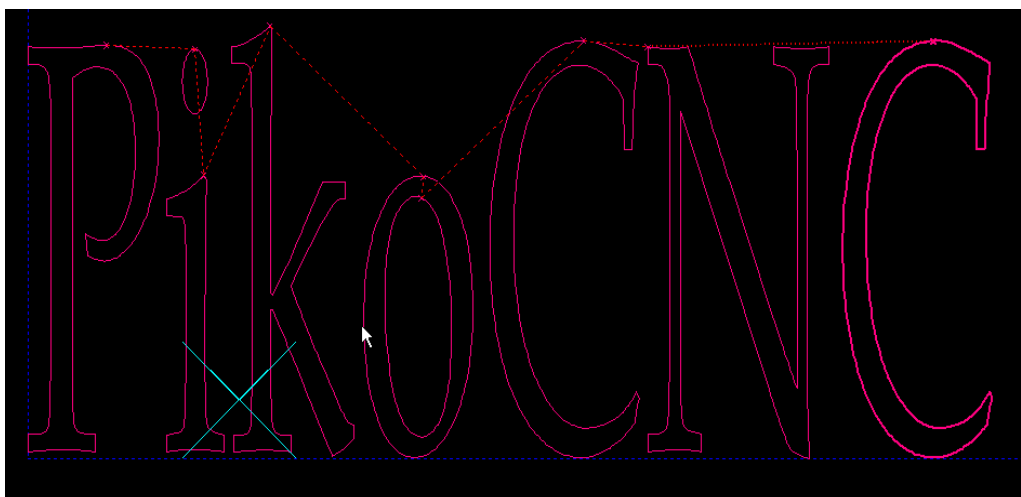


Zmiana punktu wejścia narzędzia

PikoCam umożliwia zmianę punktu wejścia narzędzia dla danego konturu. Aby to zrobić należy trzymając przyciśnięty klawisz **Alt** kliknąć na miejsce na konturze, które wybraliśmy na nowy punkt wejścia. Operacja ta działa oczywiście tylko dla konturów zamkniętych i należy ją wykonywać w widoku 2D (jak wszystkie operacje z udziałem myszki).



Przykład: oryginał.



I zmieniony...

Skrypty

Program umożliwia wykonywanie skryptów języka pascal. Opis samego języka wykracza daleko poza zakres tej instrukcji obsługi i szczegółów jego dotyczących należy szukać w sieci pod hasłami „pascal scripts” „pascal” „Delphi” „Lazarus”. Opisane będą tylko dodatkowe funkcje zaimplementowane dla potrzeb PikoCNC. Uruchamianie skryptów bez otwierania okna edytora można wykonać za pomocą klawisza F8.

Funkcje budowania linii G-kodu:

```

procedure SetX(v:Extended);
procedure SetY(v:Extended);
procedure SetZ(v:Extended);
procedure SetI(v:Extended);
procedure SetJ(v:Extended);
procedure SetR(v:Extended);
procedure SetF(v:Extended);
procedure SetS(v:Extended);
procedure SetP(v:Extended);
procedure SetM(v:byte);
procedure SetG(v:byte);
procedure Write();

```

Są to główne funkcje generowania pojedynczej linii G-kodu. Instrukcję „Write” wykonujemy na końcu aby zapisać linię np.

```

SetG(0); SetX(ODSTEP); SetY(ODSTEP); SetM(3); write;
SetZ(1); write;
SetG(1); SetZ(-1*DEPTH); SetF(Fz); write;

```

Należy też pamiętać, że wszystkie zmienne oprócz I,J,R są modalne - wynika z tego, że jeżeli w jakiejś linii ustawimy np. SetG(1) to G1 będzie obowiązywało do momentu aż funkcją SetG tego nie zmienimy! Dlatego powtarzanie w każdej linii np. SetG(1) nie ma sensu. Oczywiście dotyczy to też wszystkich pozostałych parametrów (oprócz I,J,R).

Funkcje ułatwiające prowadzenie ścieżki

```

procedure SetBulge(v:Extended);
procedure SetMove(len,alfa:extended);
procedure SetMoveR(len,alfa:extended);
procedure SetAngle(alfa:extended);
function GetX():Extended;
function GetY():Extended;
function GetZ():Extended;
function CirPointX(r,alfa:Extended):Extended;
function CirPointY(r,alfa:Extended):Extended;

```

Są to funkcje ułatwiające prowadzenie ścieżki.

procedure SetBulge(v:Extended);

Funkcja stosowana dla G2 G3. Tak dobiera parametr „R” aby łuk który powstanie miał wysokość „V”. Przed użyciem SetBulge musi być już ustawione G2 lub G3 oraz docelowy koniec łuku (x,y).

przykład:

```

SetG(3); SetX(ODSTEP);
SetBulge(LUK_WYS);
write;

```

procedure SetMove(len,alfa:extended);

Funkcja tak ustawia X,Y aby od bieżącego punktu powstała ścieżka o długości „len” i pochylona do osi X pod kątem „alfa”.

procedure SetMoveR(len,alfa:extended);

procedure SetAngle(alfa:extended);

Funkcja **SetMoveR** jest podobna do **SetMove** z tą różnicą, że kąt jest liczony względem poprzedniego wektora (ale wytyczonego tą funkcją!). Wartość początkową kąta ustala się funkcją **SetAngle**.

function GetX():Extended;

function GetY():Extended;

```
function GetZ():Extended;
```

Funkcje zwracają ostatnio zapisaną pozycję danej osi. (zapisaną poleceniem „write”)

```
function CirPointX(r,alfa:Extended):Extended;
```

```
function CirPointY(r,alfa:Extended):Extended;
```

Funkcje zwracają współrzędne X oraz Y punktu na okręgu o promieniu „r” i kącie położenia „alfa”;

Funkcje wprowadzania argumentów dla skryptu

```
procedure SetArg(num:cardinal; txt:string; v:variant);
```

```
function Arg(num:cardinal):variant;
```

```
procedure ArgEdit();
```

```
procedure SetArgPic(pname:string);
```

```
procedure ArgShow(num:cardinal);
```

```
procedure ArgNewPage(txt:string);
```

```
procedure ArgPageTitle(txt:string);
```

```
procedure ArgSeparator(txt:string);
```

```
procedure ArgShow(num:cardinal);
```

```
procedure ArgNewPage(txt:string);
```

```
procedure ArgPageTitle(txt:string);
```

```
procedure ArgSeparator(txt:string);
```

Są to funkcje, dzięki którym możemy stworzyć prosty interfejs do wprowadzania parametrów skryptu.

```
procedure SetArg(num:cardinal; txt:string; v:variant);
```

num – numer argumentu (maksymalnie może być 50);

txt – tekst objaśniający.

V – zmienna która jest wartością wyjściową do wpisywania jak również jej typ mówi jakiego rodzaju ma być okienko do wpisywania wartości np:

```
SetArg(0,'Test',120.0); // liczba zmiennoprzecinkowa
```



```

SetArg(0,'Test',120); // liczba całkowita
SetArg(0,'Test',TRUE); // liczba boolean – zostanie utworzony checkbox
SetArg(0,'*Test/raz/dwa/trzy',2); // liczba całkowita – zostanie utworzony combobox
                                z pozycjami raz dwa trzy z wybranym polem „trzy”
                                (Musi być gwiazdka na pierwszej pozycji ! )
SetArg(0,'Test','ala ma kota'); // string – zostanie utworzone pole edycji tekstu

```

Przykład:

```

if (RUN_CNT=0) then begin // wykonywane tylko przy pierwszym uruchomieniu
    SetArgPic('meblowy_info.jpg'); // obrazek objaśniający
    SetArg(0,'Szerokość frontu (mm)',120.0); // Arg. nr 0 liczba zmiennoprzecinkowa
    SetArg(1,'Wysokość frontu (mm)',60.0); // Arg. nr 1 liczba zmiennoprzecinkowa
    SetArg(2,'Wysokość łuku (mm)',20.0); // Arg. nr 2 liczba zmiennoprzecinkowa
    SetArg(3,'Odległość od krawędzi (mm)',10.0);
    SetArg(4,'*Górna linia/Łuk/Prosta',0); // Arg. nr 4 liczba całkowita - combobox
    SetArg(5,'Szybkość obróbki w osi XY (mm/min)',2000); // Arg. nr 5 liczba całkowita
    SetArg(6,'Zaokrąglone rogi',FALSE); // // Arg. nr 6 liczba typu Boolean
end;

```

ArgEdit; // otwiera okno edycji parametrów

// I odczyt wprowadzonych argumentów

szer:=Arg(0); wys:=Arg(1); luk_wys:=Arg(2); odstep:=Arg(3);

Fxy:=Arg(5); rogi:=Arg(6);

function Arg(num:cardinal):variant;

Umożliwia odczyt wprowadzonych argumentów (patrz przykład).

procedure ArgEdit();

Otwiera okno edycji.

procedure SetArgPic(pname:string);

Ustawia nazwę pliku obrazu, który służy jako ilustracja do wprowadzanych parametrów – można go jednak pominąć. Plik musi być w tym samym katalogu co skrypt i wystarczy podać samą nazwę.

```

procedure ArgShow(num:cardinal);
procedure ArgNewPage(txt:string);
procedure ArgPageTitle(txt:string);
procedure ArgSeparator(txt:string);

```

Powyższe procedury służą do organizowania sposobu wyświetlania parametrów.

Przykład:

```

ArgPageTitle('Wymiary frontu'); // nazwa pierwszej zakładki
ArgShow(0); // dodajemy pierwszy wyświetlany parametr
ArgShow(1); // drugi ....
ArgShow(2);
ArgSeparator(""); // dodanie separatora – tutaj pusta linia
ArgSeparator('Autor: Andrzej Woźniak Damasławek 2011');
ArgNewPage('Łuk 3'); // dodanie nowej zakładki o nazwie „Łuk 3”
ArgShow(10); // dodajemy parametry do nowej zakładki
ArgShow(3);
ArgShow(2);
ArgShow(1);
ArgNewPage('Łuk 2'); // i kolejna zakładka o nazwie „Łuk 2”
ArgShow(13); ArgShow(11); ArgShow(4); ArgShow(5); ArgShow(6);
ArgEdit; // i otwieramy okno

```

Jeżeli przed wywołaniem „*ArgEdit*” ani razu nie wywołamy żadnej z powyższych funkcji to wszystkie zdefiniowane parametry zostaną wyświetlone na jednej zakładce w kolejności nadanych przez funkcję „*SetArg*” numerów . Oczywiście najpierw należy parametry zdefiniować (*SetArg*) a dopiero później organizować porządek wyświetlania.

Funkcje matematyczne

```
function Tan(v:Extended):extended;
function ArcSin(v:Extended):extended;
function ArcCos(v:Extended):extended;
function ArcTan(v:Extended):extended;
```

Dodatkowo zaimplementowane funkcje matematyczne (funkcje sin() cos()) są zaimplementowane „w standardzie”).

Funkcje komunikacji z użytkownikiem

```
procedure MsgOk(txt:string);
function MsgYesNo(txt:string):boolean;
procedure Dbg(txt:string);
procedure Dbg_clr();
procedure Log(txt:string);
```

procedure MsgOk(txt:string);

komunikat z jednym przyciskiem „Ok”. np. MsgOk('ala ma kota');

function MsgYesNo(txt:string):boolean;

Komunikat z przyciskami „Yes” „No” zwraca TRUE jeśli naciśniemy „Yes”

procedure Dbg(txt:string);

Wypisuje tekst na konsoli pod oknem edytora.

procedure Dbg_clr();

Czyści zawartość konsoli pod oknem edytora.

procedure Log(txt:string);

Wypisuje tekst w oknie Log (zakładka „START” programu głównego)

Stałe

Program pewne stałe deklaruje automatycznie:

MAT_SIZE_X, MAT_SIZE_Y, MAT_SIZE_Z (rozmiary materiału) kopia wartości z pól „RozmiarXYZ” zakładki „Materiał”.

ACT_POS_X, ACT_POS_Y, ACT_POS_Z (aktualna pozycja osi - względem materiału)

ACT_POS_X_M, ACT_POS_Y_M, ACT_POS_Z_M (aktualna pozycja osi – współrzędne maszynowe)

ACT_F, ACT_S – są to kopie wartości wpisanych w okienka F i S programu głównego zakładka „Sterownie”

ACT_ZRAPID, ACT_F_Z – kopie wartości z pól „wysokość dojazdowa” i „prędkość schodzenia” z zakładki „Materiał” programu głównego.

TAB_SIZE_X, TAB_SIZE_Y -wymiary obszaru roboczego z ustawień.

RUN_CNT - licznik uruchomień skryptu zerowany po wczytaniu i wprowadzeniu zmian w skrypcie

MAKRA

Makra składniowo są podobne do zastosowanego w skryptach pascalem. Różnice są następujące:

- Wszystkie makra muszą być zapisane w katalogu „macros” programu.
- W jednej linii tylko jedna komenda!
- Blok z komendami musi być poprzedzony znakiem „%”
- Nie można wprost korzystać z typowych dla pascala zmiennych.
- Nie można deklarować dodatkowych funkcji czy procedur.
- Należy trzymać się poniższej konwencji:

```
//=====
// Przykład makra do pomiaru pozycji X materiału za pomocą sondy dotykowej
//=====
const           // deklaracja stałych (nie jest to konieczne)
F_NAJAZD=400;
F_ZJAZD=800;
%               // Znak procent oznacza rozpoczęcie programu
RefOFF(1,0,0,F_NAJAZD); // Najazd sondą na lewy bok materiału
SetXMat(PosX); // Ustalenie aktualnej pozycji jako X materiału
MoveF(PosX-10,PosY,PosZ,F_ZJAZD); // odjazd 10mm od materiału
//...koniec
```

Funkcje

```
function PosX():Extended;
```

```
function PosY():Extended;
```

```
function PosZ():Extended;
```

Funkcje zwracają aktualną pozycję maszyny. Są to zawsze współrzędne maszynowe.

*** MAKRA operują wyłącznie na współrzędnych maszynowych!**

```
procedure Move(x,y,z:Extended);
```

Przesuwa narzędzie do pozycji x,y,z. Jest odpowiednikiem G0.

```
procedure MoveF(x,y,z:extended;F:cardinal);
```

Tak samo jak funkcja Move() przesuwają narzędzie do pozycji x,y,z. Różnica jest taka, że

można podać prędkość (F) z jaką odbędzie się ruch.

procedure Wait();

Umożliwia oczekiwanie makra na jakieś zdarzenie np.

If (not Input(4)) then Wait;

linia ta zatrzymuje makro i czeka, aż użytkownik naciśnie przycisk 'RESET'.

procedure WaitMove();

Procedura wstrzymuje wykonywanie makra do momentu zakończenia ruchu maszyny. Niemal zawsze należy ją stosować po komendach Move, MoveF.

procedure Delay(time:word);

Procedura zatrzymuje makro na podaną ilość 1/100 sek.

procedure RefON(x,y,z:integer; f:cardinal);

procedure RefOFF(x,y,z:integer; f:cardinal);

Jazda określonych osi w poszukiwaniu krańcówek HOME. „RefON” przemieszcza oś tak długo aż nie zostanie zamknięta krańcówka tej osi. Natomiast „RefOFF” przemieszcza oś tak długo aż nie zostanie otwarta krańcówka tej osi. Parametry x,y,z określają które osie będą poszukiwały krańcówki, oraz to w którą stronę będzie się to działo. Parametr „F” określa prędkość jazdy.

np. RefON(0,0,-1,800); Oś „Z” jedzie w dół z prędkością 800mm/min do momentu załączenia krańcówki HOME_Z.

procedure SetPos(x,y,z:extended);

Ustawia aktualną pozycję maszyny

procedure SetOut(num:cardinal; st:boolean);

Ustawia stan wyjść np. SetOut(1,TRUE); załączy wyjście nr1

procedure EndMacro();

Umożliwia awaryjne zakończenie wykonywanego makra.

procedure SetVar(num:cardinal; v:variant);

function GetVar(num:cardinal):variant;

Obie funkcje umożliwiają pracę ze zmiennymi. Jest to jedyny sposób w makrach.

Parametr num określa numer zmiennej. Maksymalnie może wynosić 20.

przykład:

```
SetVar(0,100); //Nadanie zmiennej nr 0 wartości 100
```

```
SetVar(0,(GetVar(0)/2)+1);
```

```
Move(GetVar(0),0,0); // Przejazd maszyny do pozycji x=51 y=0 z=0
```

```
function Input(num:word):boolean;
```

```
function Output(num:word):boolean;
```

Funkcje zwracają stan wejść/wyjść o podanych numerach

```
function InputHomeX():boolean;
```

```
function InputHomeY():boolean;
```

```
function InputHomeZ():boolean;
```

Funkcje zwracają stan krańcówek HOME.

```
function GetXMat():extended;
```

```
function GetYMat():extended;
```

```
function GetZMat():extended;
```

Funkcje zwracają aktualne współrzędne materiału

```
procedure SetXMat(v:extended);
```

```
procedure SetYMat(v:extended);
```

```
procedure SetZMat(v:extended);
```

Funkcje ustawiają aktualne współrzędne materiału

```
function GetXSizeMat():extended;
```

```
function GetYSizeMat():extended;
```

```
function GetZSizeMat():extended;
```

Funkcje zwracają aktualny rozmiar materiału w poszczególnych osiach.

```
procedure SetXSizeMat(v:extended);
```

```
procedure SetYSizeMat(v:extended);
```

procedure SetZSizeMat(v:extended);

Procedury ustawiają aktualne położenie materiału w poszczególnych osiach.

function GetActTool():cardinal;

Funkcje zwraca numer aktualnego narzędzia

procedure SetActTool(num:cardinal);

Funkcja ustawia narzędzie o podanym numerze jako aktualne. Jeżeli jako numer narzędzia podamy zero oznacza to brak narzędzia w uchwycie a przede wszystkim wyzerowanie korekcji dla osi „Z”. Ważne jest to szczególnie w makrach wymiany narzędzia gdzie istotne jest aktualne położenie oprawki narzędzia a nie jego końca.

function GetReqTool():cardinal;

Funkcja zwraca numer narzędzia wymaganego przez program. (wykorzystywana w makrach automatycznej wymiany narzędzia)

procedure SetToolPos(posZ:extended);

Ustala pozycję „Z” końca aktualnego narzędzia. Jako „posZ” podajemy wysokość czujnika na którym mierzymy narzędzie.

function Base_Z():extended;

Zwraca pozycję krańcówki BASE_Z pomniejszoną o aktualną korekcję osi „Z”.

Zdefiniowane stałe:

G28X,G28Y,G28Z - pozycja x,y,z dla HOME1 (G28) (z ustawień).

G30X,G30Y,G30Z - pozycja x,y,z dla HOME2 (G30) (z ustawień).

BASE_X,BASE_Y - pozycja krańcówek HOME (z ustawień).

SMT_HEIGHT – wysokość czujnika długości narzędzia (z ustawień).

SMM_HEIGHT - wysokość czujnika wysokości materiału (z ustawień).

Makra wymiany narzędzia M6

Automatyczna wymiana narzędzia składa się z sekwencyjnego wywołania następujących makr (w tej kolejności) M6_beg, M6_put, M6_get, M6_mess, M6_end.

Załączenie automatycznej wymiany narzędzia dokonuje się w ustawieniach zakładka „G kod / Zmiana narzędzia (M6)”. Należy tam zaznaczyć „Makro M6”.

M6_beg

Wywoływane jeżeli narzędzie wymagane jest różne od aktualnego lub gdy narzędzie aktualne nie jest zmierzone. Zadaniem tej części jest głównie wyłączyć wrzeciono lub też inne czynności przygotowawcze, które muszą być wykonane zawsze.

M6_put

Wywoływane jeżeli narzędzie wymagane jest różne od aktualnego. Ale nie wywoływane jeżeli aktualne narzędzie jest „0” czyli brak narzędzia w uchwycie. Zadaniem tego makra jest odłożyć na miejsce narzędzie będące w uchwycie (opróżnić uchwyt). Jedną z pierwszych czynności w tym makro powinno być odwołanie korekcji w osi „Z” przez wywołanie **SetActTool(0);**

M6_get

Wywoływane jeżeli narzędzie wymagane jest różne od aktualnego. Zadaniem tego makra jest umieścić narzędzie w uchwycie. Makro powinno kończyć się komendą: **SetActTool(GetReqTool)** czyli ustawieniem pobranego narzędzia jako aktualnego.

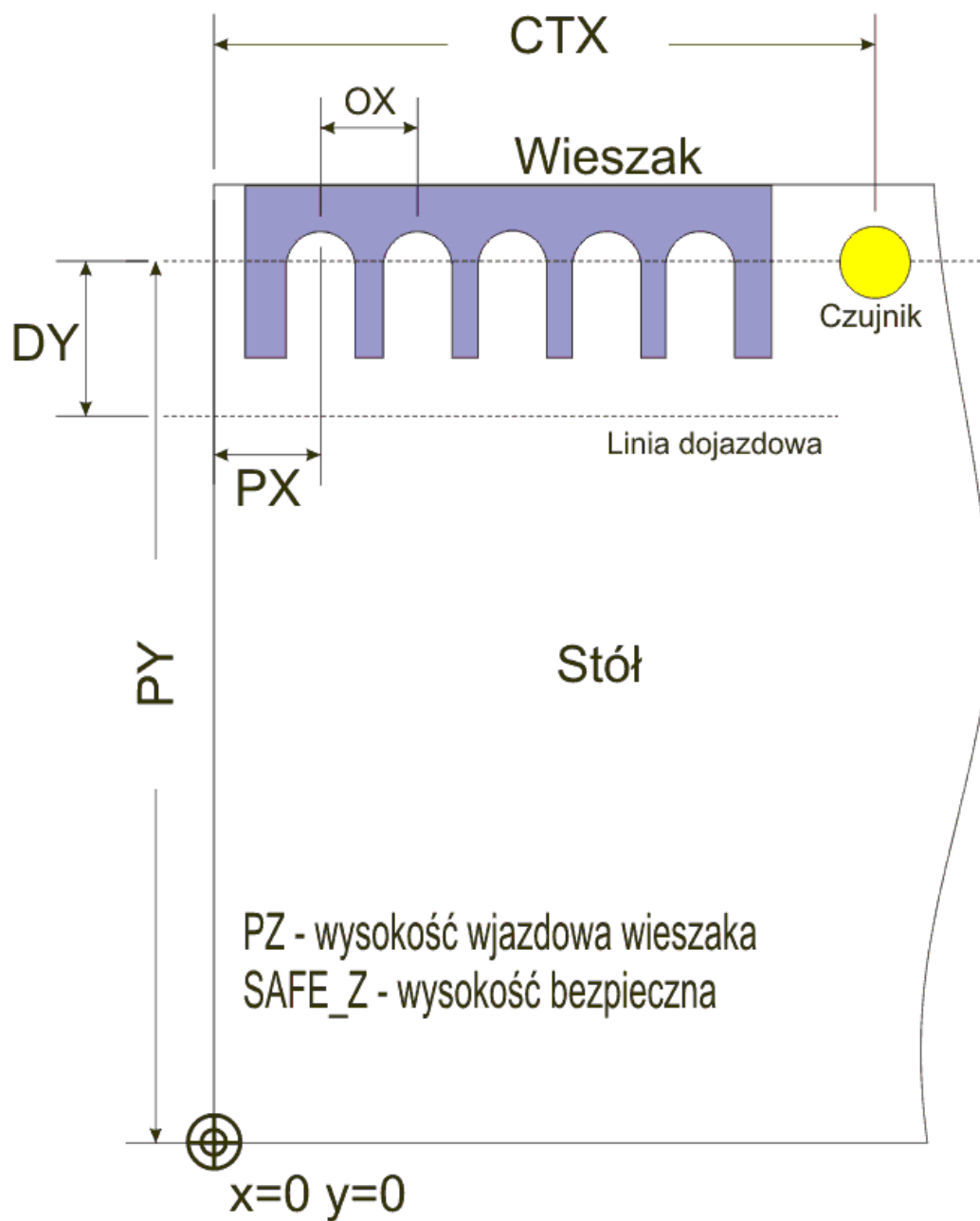
M6_mess

Wywoływane jeżeli będące aktualnie w uchwycie narzędzie nie jest zmierzone. Makro powinno zawierać funkcję **SetToolPos(posZ:extended)** czyli ustalenia pozycji końca freza. O tym czy narzędzie jest zmierzone czy nie decydujemy w oknie „Magazyn narzędzi” zaznaczając TAK lub NIE w kolumnie „Zmierzony”.

M6_end

Sekwencja końcowa wywoływana zawsze jeżeli wywołane było M6_beg. Powinna zawierać czynności końcowe jak np. załączenie wrzeciona.

Przykład: mamy wieszak na oprawki narzędzi zamocowany u góry stołu. Obok czujnik do pomiaru długości narzędzia. Wyjście Out2 służy do sterowania zamkiem uchwytu.



=====

Plik: M6_param

// plik ten służy wyłącznie do deklaracji wszystkich stałych. Można oczywiście ten
// fragment umieścić w każdym pliku osobno ale tak jest o wiele prościej wprowadzać
// ewentualne poprawki.

```
const          // deklaracja stałych
F_NAJAZD=400;
F_ZJAZD=800;
PX=10;
PY=200;
PZ=50;
DY=30;
OX=40;
SAFE_Z=90;
CTX=150;
CT_HEIGHT=29; //wysokość czujnika długości narzędzia
ZW=40;
// koniec M6_param
```

=====

plik: M6_beg

```
{%I macros/M6_param.txt} // wczytanie pliku z deklaracjami stałych
%
Move(PosX,PosY,SAFE_Z); // Uniesienie „Z” na wysokość bezpieczną
WaitMove;
SetOut(1,FALSE); //wyłączenie wrzeciona
// koniec M6_beg
```

=====

plik: M6_put

```
{%I macros/M6_param.txt} // wczytanie pliku z deklaracjami stałych
%
Move(PosX,PosY,SAFE_Z); // Uniesienie „Z” na wysokość bezpieczną
WaitMove;
Move(PX+((GetActTool-1)*OX),PY-DY,PosZ); // dojechanie do linii dojazdowej
WaitMove;
SetActTool(0); // odwołanie korekcji „Z” -- BARDZO WAŻNE !!!
Move(PosX,PosY,PZ); // Uchwyt na pozycję wjazdową „Z”
WaitMove;
Move(PosX,PosY+DY,PosZ); // wjechanie w uchwyt
WaitMove;
SetOut(2,TRUE); // zwolnienie zamka
Delay(20);
Move(PosX,PosY,SAFE_Z); // wyjazd w górę
SetOut(2,FALSE); // załączenie zamka
// koniec M6_put
```

=====

plik: M6_get

```
{ $I macros/M6_param.txt } // wczytanie pliku z deklaracjami stałych
%
Move(PosX,PosY,SAFE_Z);
WaitMove;
Move(PX+((GetReqTool-1)*OX),PY,PosZ); // dojazd nad odpowiednie narzędzie
WaitMove;
SetOut(2,TRUE); // zwolnienie zamka
Move(PosX,PosY,PZ); // „Z” na pozycję wyjazdową (najazd na narzędzie)
WaitMove;
SetOut(2,FALSE); // załączenie zamka
Delay(20);
Move(PosX,PosY-DY,PosZ); // Wyjazd z uchwytu
WaitMove;
SetActTool(GetReqTool); // Ustalenie narzędzia jako aktualnego
Move(PosX,PosY,SAFE_Z); // Uniesienie na wysokość bezpieczną
// koniec M6_get
```

=====

plik: M6_mess

```
{%I macros/M6_param.txt} // wczytanie pliku z deklaracjami stałych
%
Move(PosX,PosY,SAFE_Z); // uniesienie "Z" na wysokość bezpieczną
WaitMove;
Move(CTX,PY,SAFE_Z); // dojazd na pozycję XY czujnika
WaitMove;
RefOn(0,0,-1,F_NAJAZD); // zjazd do czujnika
SetToolPos(CT_HEIGHT); // ustalamy aktualną pozycję "Z"
Move(PosX,PosY,SAFE_Z); // uniesienie "Z" na wysokość bezpieczną
// koniec M6_mess
```

=====

plik: M6_end

```
{%I macros/M6_param.txt}
%
Move(PosX,PosY,SAFE_Z); // Uniesienie „Z” na wysokość bezpieczną
WaitMove;
SetOut(1,TRUE); //włączenie wrzeciona
```

Makra M3/M5

Program umożliwia zdefiniowanie makr dla komend M3 oraz M5, czyli dla załączenia i wyłączenia narzędzia. Opcję włączamy w ustawieniach G kod/"M3/M5 makro".

Przykład dla wypalarki plazmowej. Jako wejście potwierdzenia zajarzenia łuku wykorzystano wejście HOME_Z .

=====

plik: M3

%

SetOut(1,TRUE); // załączenie łuku

if (not InputHomeZ) then Wait; // czekamy na potwierdzenie zajarzenia łuku

Delay(100); // czekamy jeszcze np. sekundę na przebicie materiału

// koniec M3

=====

plik: M5

%

SetOut(1,FALSE); // wyłączenie łuku

// koniec M5

Skanowanie

Grupa: Wymiary obszaru (mm)

Szerokość: szerokość (wymiar w osi X) obszaru skanowanego. Jeżeli używamy osi X jako obrotowej pole zmienia nazwę na „Średnica” - i należy wpisać tutaj średnicę skanowanego obiektu.

Wysokość: wysokość (wymiar w osi Y) obszaru skanowanego. Jeżeli używamy osi Y jako obrotowej pole zmienia nazwę na „Średnica” - i należy wpisać tutaj średnicę skanowanego obiektu. Należy jeszcze dodać, że w czasie skanowania bez osi obrotowej głowica skanująca przemieszcza się na wysokości Z=0 natomiast w przypadku skanowania z osią obrotową na wysokości Z=średnica/2.

Oś obrotowa/Oś: wybór osi obrotowej.

Imp/360 stp.: Liczba impulsów przypadających na pełen obrót osi. Wszystkie pozostałe parametry osi (dynamika etc.) zostają takie jak normalnie przypisane do danej osi.

Grupa parametry

Rozdzielczość: Decyduje co ile mm będzie pobrana próbka. Wpisując tu jakąś wartość należy mieć na uwadze następujące kwestie:

- proces skanowania jest procesem bardzo czaso i pamięciożernym.
- Kontroler może w ciągu sekundy pobrać nie więcej niż 800 próbek.

Dlatego trzeba zachować rozsądne proporcje między wielkością obszaru skanowanego a rozdzielczością.

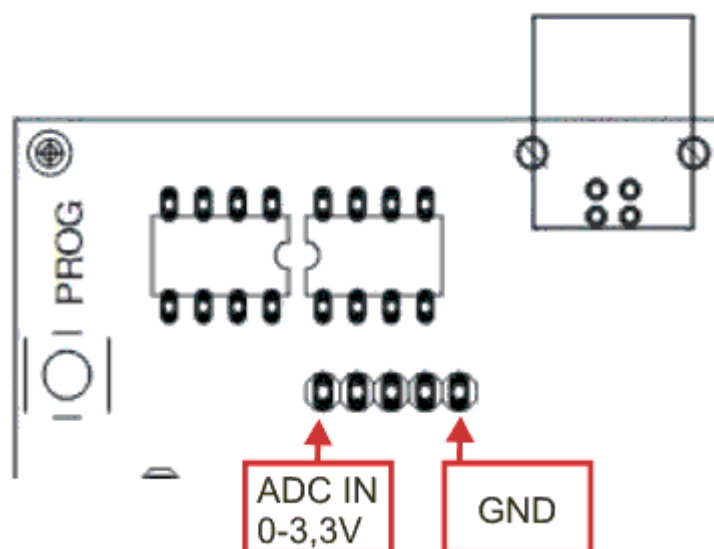
Szybkość: Tempo z jakim poruszać się będzie maszyna w czasie skanowania.

„Z dla Umax”: Głębokość „Z” dla maksymalnego napięcia na wejściu ADC.

Zapis

Po wykonaniu skanowania dane można zapisać. Po wyborze nazwy pliku program dane zapisuje od razu w dwóch formatach: bmp oraz „chmura punktów”. Tak więc jeżeli w wybieraczkę plików wybierzemy nazwę „figurka” zostaną utworzone pliki: figurka.bmp oraz figurka_cld.txt

Podłączenie dalmierza



Wejście układu pomiarowego znajduje się jak na rysunku. Na wyjściu dalmierza należy zastosować odpowiedni dzielnik napięcia wyjściowego tak aby:

napięcie wejściowe przetwornika ADC nigdy nie przekroczyło 3,3V - Nie zastosowanie się do tego grozi uszkodzeniem mikroprocesora !!!

Aktualny pomiar z przetwornika można zawsze zobaczyć w oknie monitora (zakładka I/O).

Klawiatura

klawisz	funkcja
S	(program) Start
E	(program) Stop
spacja	Pauza
Esc	E_stop
F1	Widok „material”
F2	Widok „praca”
F3	Widok „stół”
F4	Widok „wszystko”
F5	Przejazd do HOME 1
F6	Przejazd do HOME 2
shift+F5	Nadanie HOME 1 wartość XYZ aktualnej pozycji
shift+F6	Nadanie HOME 2 wartość XYZ aktualnej pozycji
F8	Uruchomienie skryptu jeżeli jest załadowany
F9	Zwiększenie prędkości obróbki o 10%
F10	Zmniejszenie prędkości obróbki o 10%
F11	Prędkość obróbki 100%
Strzałki kursora	Posuw ręczny XY o zadany krok
shift+Strzałki kursora	Posuw ręczny XY ciągły z prędkością dojazdową
ctrl+Strzałki kursora	Posuw ręczny XY ciągły z połową prędkości dojazdowej
PageUp (Down)	Posuw ręczny Z o zadany krok
shift+PageUp (Down)	Posuw ręczny Z ciągły z prędkością dojazdową
ctrl+PageUp (Down)	Posuw ręczny Z ciągły z połową prędkości dojazdowej
kółko	Zoom +/-
ctrl+Z	Przejazd z prędkością dojazdową do pozycji XY znacznika
ctrl+”<”	Przesunięcie znacznika do tyłu i dojazd do pozycji
ctrl+”>”	Przesunięcie znacznika do przodu i dojazd do pozycji

ctrl+X	Przejazd z prędkością dojazdową do 0,0 materiału
ctrl+C	Przejazd z prędkością dojazdową do pozycji XY środka materiału
ctrl+M	Przejazd z prędkością dojazdową do pozycji XY wskaźnika myszki (Działa tylko w widoku 2D i perspektywie Yx)
BACKSPACE	Czyszczenie ścieżki narzędzia