



Instrukcja

Panel operatorski HMI35 z wyświetlaczem TFT 3,5"

wersja dla kompilatora 6.xx

OPIS PANELU

Panel operatorski HMI35 jest swobodnie programowalnym panelem HMI z 3,5" ekranem dotykowym. Rozdzielczość ekranu wynosi 480x320 pikseli. Panel umożliwia komunikację z maksymalnie ośmioma urządzeniami typu SLAVE. Komunikacja może odbywać się poprzez magistralę szeregową RS485 z wykorzystaniem protokołu Modbus RTU lub poprzez sieć Wi-Fi (LAN) z wykorzystaniem protokołu Modbus TCP.

WEJŚCIA i WYJŚCIA

Widok złącz panelu



Oznaczenia	Opis
Uz	Napięcie zasilania 1224VDC:
	+ - biegun dodatni (+) napięcia zasilania,
	 biegun ujemny (masa) napięcia zasilania.
RS485	Port RS485.
SW	Przełącznik funkcji panelu. Opis:
	SW1 – nieobsługiwany,
	SW2=ON – wejście do trybu programowania,
	SW3=ON – wejście do trybu konfiguracji modułu Wi-Fi,
	SW4=ON – wejście do trybu kalibracji ekranu dotykowego.
	W trybie normalnej pracy wszystkie przełączniki powinny być ustawione na OFF.
Prg	Przycisk trybu aktualizacji firmware.
USB A	Złącze USB A przeznaczone do podłączenia pendrive z oprogramowaniem
	użytkownika.
USB B	Złącze mikro USB B przeznaczone do aktualizacji firmware.

DIODY

Panel posiada w prawym górnym rogu trzy diody.

Oznaczenia	Opis
PWR	Świeci na zielono, jeśli panel ma podłączone zasilanie.
STATUS	Miga na zielono, jeśli nie ma żadnej awarii. Wszystkie rejestry statusowe zadeklarowane w projekcje mają wartość równą "0". Miga na czerwono, jeśli wykryto awarię lub brak komunikacji z zadeklarowanym sterownikiem. Awaria oznacza, że przynajmniej jeden rejestr statusowy zadeklarowany w projekcie ma wartość różną od "0". Rejestry statusowe deklaruje się w plikach predefiniowanych o nazwach ster0.txt ster7.txt w parametrze ADR STATUS.
LAN	Świeci na zielono, gdy aktywne jest połączenie przez Wi-Fi (niezależnie od trybu pracy wbudowanego modułu).

KALIBRACJA EKRANU DOTYKOWEGO

W celu skalibrowania ekranu dotykowego panelu trzeba wyłączyć jego zasilanie, przełączyć SW4 na ON i włączyć zasilanie. Następnie naciskać na punkty pojawiające się na ekranie wyświetlacza.



Po poprawnym przejściu procedury kalibracji wyświetli się komunikat informacyjny: "Kalibracja ekranu zakończona powodzeniem". Następnie trzeba wyłączyć zasilanie,

przełączyć SW4 w pozycję OFF i włączyć ponownie zasilanie.

PROGRAMOWANIE

Programowanie panelu odbywa się poprzez proste skrypty zapisywane w plikach tekstowych. Do napisania skryptów wystarczy standardowy notatnik Windows. Do kompilacji służy dedykowany, dostarczany bezpłatnie, kompilator "Kompilator5". Wgranie gotowego programu do panelu HMI35 ogranicza się do zapisania skompilowanego wynikowego pliku na pendrive i uruchomieniu procedury programowania na panelu. Kompilator i przykładowy projekt można ściągnąć ze strony <u>https://frisko.com.pl</u> z działu Do Pobrania.

Projekt składa się plików tekstowych (z rozszerzeniem .txt), które mogą być napisane za pomocą zwykłego notatnika lub innego programu tekstowego. Umożliwia to stworzenie programu na panel HMI35 przez zwykłego użytkownika po przeczytaniu tej instrukcji. Nie jest wymagany żaden specjalistyczny program. Po napisaniu wszystkich potrzebnych plików projektowych trzeba uruchomić kompilator, który zweryfikuje i połączy wszystkie pliki, a następnie utworzy plik wynikowy **hmi35.bin**. Gotowy plik z projektem należy skopiować na pendrive - nośnik pamięci z USB typu A.

W celu zaprogramowania panelu trzeba wyłączyć jego zasilanie, przełączyć SW2 na ON

i włożyć pendrive do gniazda USB. Po włączeniu zasilania postępować zgodnie z instrukcjami wyświetlanymi na ekranie.

Jeśli na nośniku pamięci wykryty zostanie plik z programem o nazwie: hmi35.bin to wyświetli się następujący ekran:



Jeśli chcemy zaprogramować nasz panel HMI35 nowym programem to wystarczy dotknąć ekran. Jeśli nie chcemy nowego programu trzeba wyłączyć zasilanie, wyjąć nośnik

pamięci i przełączyć SW2 w pozycję OFF

Podczas wgrywania programu następuje weryfikacja pamięci i w przypadku wystąpienia błędów wyświetlony zostanie odpowiedni komunikaty, a nowy program nie będzie zaakceptowany.

Jeśli program zostanie poprawnie wgrany to wyświetli się następujący komunikat:



Po wgraniu należy wyłączyć zasilanie, wyciągnąć nośnik pamięci i przełączyć SW2 na OFF a następnie włączyć zasilanie.

Jeśli wyświetli się następujący komunikat:



oznacza to, że system w panelu i kompilator użyty do stworzenia programu mają różne wersje.

KONFIGURACJA MODUŁU WI-FI

Konfiguracja wewnętrznego modułu Wi-Fi w panelu HMI35 odbywa się poprzez sieć Wi-Fi z poziomu przeglądarki WWW. Najlepiej do tego celu użyć smartfona lub tabletu. <u>W ce</u>lu

włączenia trybu konfiguracji trzeba wyłączyć zasilanie, przełączyć SW3 na ON włączyć zasilanie. Dioda 'LAN' zacznie migać na zielono a na ekranie powinien się wyświetlić następujący ekran informacyjny:



Teraz panel HMI35 pracuje jako access point oraz serwer WWW. Musisz połączyć się z nim poprzez Wi-Fi. Włącz w swoim urządzeniu (smartfonie, tablecie) obsługę sieci Wi-Fi. Znajdź sieć o nazwie "HMI35" i połącz się z nią.

	🐝 👷 🖘 📶 84% 🛢 08:49	
< Wi-Fi	Wi-Fi Direct	
WŁĄCZONY		
BIEŻĄCA SIEĆ		
HMI35	oże być niedostępny	
DOSTĘPNE SIECI		

Hasło dostępu to "12345678".

HMI	35	
Hasło 1234	5 45678	
	Wyświetl hasło	
	Automatyczne ponov łączenie	wne
\cap	Dokaż oncie zaawan	eowana
	ANULUJ	POŁACZ

Następnie uruchom przeglądarkę, w polu adresu wpisz "192.168.4.1" i naciśnij klawisz Enter (lub OK).



Na ekranie smartfona (tabletu) zostanie wyświetlona strona umożliwiająca konfigurację czujnika.

	Voi)	,ı∣ 84% í	08:51
⚠ ▲ 192.168.4.1	+	1	:
HMI35			
_Nazwa WiFi			
TEST_WIFI			
Hasło WiFi			
1234567890			
-DHCP			
Tak 🗸			
Adres IP			
192.168.1.151			
-Maska			
255.255.255.0			
Brama			
192.168.1.1			
I			

Zapisz

Znaczenie poszczególnych parametrów przedstawia poniższa tabela.

Parametr	Opis
Nazwa WiFi	Nazwa sieci Wi-Fi (ssid) do której ma być podłączony panel HMI35.
Hasło WiFi	Hasło dostępu do sieci Wi-Fi do której ma być podłączony panel HMI35.
DHCP	Obsługa funkcji DHCP. Wybór opcji "Tak" oznacza, że adres IP panelu HMI35 zostanie przydzielony automatycznie poprzez usługę DHCP. Wybór opcji "Nie" oznacza wyłączenie funkcji DHCP. Adres IP czujnika, maskę i bramę należy skonfigurować ręcznie.
Adres IP	Adres IP czujnika, gdy DHCP=Nie.
Maska	Maska sieci, gdy DHCP=Nie.
Brama	Brama sieci, gdy DHCP=Nie.

Po poprawnym wypełnieniu listy parametrów naciśnij klawisz "Zapisz" znajdujący się na dole strony WWW. Po tym w przeglądarce zostanie wyświetlony poniższy komunikat.

▲ 192.168.4.1/inde: + ②

HMI35

Zmiana parametrów...

Proszę ustawić SW w trybie pracy i zrestartować panel

Następnie trzeba wyłączyć zasilanie, przełączyć SW3 w pozycję OFF i włączyć ponownie zasilanie.

(and W przypadku nieprawidłowego ustawienia parametrów nie zostaną one zapisane, a po naciśnieciu klawisza "Zapisz" zostanie ponownie wyświetlona strona z parametrami.

KOMUNIKACJA ZE STEROWNIKAMI

Panel umożliwia komunikację maksymalnie z ośmioma sterownikami typu SLAVE. Komunikacja ze sterownikami może odbywać się:

- poprzez magistralę szeregową RS485 z wykorzystaniem protokołu Modbus RTU,
- poprzez sieć Wi-Fi (LAN) z wykorzystaniem protokołu Modbus TCP.

Wyboru trybu komunikacji z danym sterownikiem dokonuje się na etapie tworzenia deklaracji tego sterownika (pliku **ster_.txt**) wpisując odpowiednią wartość parametru "MAG". Prędkość oraz format danych magistrali RS485 jest wspólny dla wszystkich sterowników z niej korzystających i definiowany jest w pliku **projekt.txt**.

Przykładowy schemat podłączenia przedstawia poniższy rysunek.



W przypadku wykrycia braku komunikacji z danym sterownikiem SLAVE zostanie wyświetlony poniższy komunikat o błędzie:



JAK ZACZĄĆ PROJEKT

Najprostszy projekt posiada jeden ekran i komunikuje się z jednym sterownikiem. Każdy projekt musi posiadać pliki **projekt.txt**, **ster0.txt** oraz **ekran0.txt**. Wszystkie pliki projektu muszą być w jednym katalogu, w którym znajduje się również kompilator. Pliki projektu tworzymy i edytujemy za pomocą dowolnego edytora tekstu np. dobrze znanego z Windows Notatnika. Schemat blokowy przykładowego projektu, znajduje się poniżej.



PROJEKT

Projekt jest zapisany w pliku projekt.txt. Każdy parametr projektu musi być zapisany w osobnej linii pliku.

Wszystkie wymagane parametry projektu przedstawia poniższa tabela.

ID= Numer ID projektu. NAZWA= Nazwa projektu. Maksymalna długość nazwy to 16 znaków. Nazw projektu jest wyświetlana na ekranie startowym. WERSJA= Wersja projektu: np. data. Maksymalna długość wersji to 16 znaków Wersja projektu jest wyświetlana na ekranie startowym. PASS- Hasło do projektu. Wszystkie objekty, które maja możliwość odycji mog
NAZWA= Nazwa projektu. Maksymalna długość nazwy to 16 znaków. Nazw wersja projektu jest wyświetlana na ekranie startowym. WERSJA= Wersja projektu: np. data. Maksymalna długość wersji to 16 znaków Wersja projektu jest wyświetlana na ekranie startowym. PASS- Hasło do projektu. Wszystkie objekty, które maja możliwość odycji mog
projektu jest wyświetlana na ekranie startowym. WERSJA= Wersja projektu: np. data. Maksymalna długość wersji to 16 znaków Wersja projektu jest wyświetlana na ekranie startowym. PASS- Hasło do projektu. Wszystkie objekty, które maja możliwość odycji mog
WERSJA= Wersja projektu: np. data. Maksymalna długość wersji to 16 znakóv Wersja projektu jest wyświetlana na ekranie startowym. PASS-
Wersja projektu jest wyświetlana na ekranie startowym. PASS- Hasło do projektu. Wszystkie objekty, które maja możliwość odycji mog
PASS- Hasło do projektu. Wszystkie objekty, które maja możliwość odycji mog
T noo- T nasio do projektu. Wszystkie obiekty, które mają możniwoso edycji mog
być chronione hasłem. Podane tutaj hasło jest stałe nie podlegając
edycji. Jeśli podamy PASS=0 to cały projekt nie będzie korzystał
funkcji hasła i wszystkie obiekty będą normalnie dostępne do edyc
przez wszystkich użytkowników.
KOLOR_TLO= Kolor tła w formacie RGB 24b (24 bitowym). Liczby muszą by
rozdzielone przecinkiem.
KOLOR_SYS= Kolor systemowy w formacie RGB 24b. W tym kolorze będ
generowane komunikaty systemowe, klawiatura i listy wyboru.
Liczba zdefiniowanych ekranow w ramach projektu. Maksymalna liczb
ekranow, jaką mozna zdefiniować to 32.
ekranů tyte ekranů tyt
Pierwszy ekran projektu ma numer zero i jest opisany w plik
ekran0.txt. Drugi ekran opisany jest w pliku ekran1.txt itd. Liczb
plików opisujących ekrany musi się zgadzać z liczba przypisana d
parametru EKRANY .
Przykład:
DIa EKRANY=1 tworzymy tylko plik ekran0.txt.
Dla EKRANY=3 tworzymy pliki ekran0.txt, ekran1.txt, ekran2.txt.
Ekran startowy (z nazwa projektu oraz logo EBISKO) nie jest wliczan
do liczby ekranów.
STER= Liczba sterowników, z którymi ma się komunikować panel w ramac
projektu. Maksymalna liczba sterowników to 8. Sterowniki muszą by
opisane w plikach o predefiniowanych nazwach ster0.txt ster7.txt.
Pierwszy sterownik projektu ma numer zero i jest opisany w plik
ster0.txt. Drugi sterownik opisany jest w pliku ster1.txt itd. Liczb
plików opisujących sterowniki musi się zgadzać z liczbą przypisaną d
parametru SIEK.
Na FIZyMau. Dla STER-1 tworzymy tylko plik ster0 tyt
Dia STER=3 tworzymy tjiko piki ster0.txt. ster1.txt. ster2.txt.
COM DATA= Konfiguracja portu COM (magistrali szeregowej RS485) - format danycł
Opcje:
 8n1 - 8 bitów danych, brak kontroli parzystości, 1 bit stopu,
8e1 - 8 bitów danych, kontrola parzystości parzystej EVEN, 1 bit stopu
801 - 8 bitów danych, kontrola parzystości nieparzystej ODD, 1 b
stopu.
COM_V= Kontiguracja portu COM (magistrali szeregowej RS485) - prędkoś
transmisji. Opcje:
■ 1200 - 1200 bps, ■ 2400 - 2400 bps
- 2400 - 2400 ups,
= 9600 - 9600 bps,
■ 19200 - 19200 bps.

Pokazany poniżej plik **projekt.txt** opisuje projekt o nazwie "MyProjekt1" (NAZWA=MyProjekt1) i wersji "20.10.2023" (WERSJA=20.10.2023). Dostęp do parametrów nie będzie chroniony hasłem (PASS=0). Projekt zawiera 4 ekrany (EKRANY=4). Ekrany musza być opisane w plikach ekran0.txt, ekran1.txt, ekran2.txt oraz ekran3.txt. Kolorem tła ekranów będzie kolor biały (KOLOR_TLO=255, 255, 255). Kolorem podstawowym, systemowym będzie kolor zielony (KOLOR_SYS=0, 255, 0). Projekt będzie obsługiwał dane z jednego sterownika (STER=1). Właściwości sterownika muszą być opisane w pliku ster0.txt. Parametry portu RS485 (magistrali com): prędkość 9600bps (COM_V=9600), format znaku 8N1 (COM_DATA=8N1).

Plik projekt.txt

ID=2 NAZWA=MyProjekt1 WERSJA=20.10.2023 PASS=0 KOLOR_TLO=255,255,255 KOLOR_SYS=0,250,0 EKRANY=4 STER=1 COM_DATA=8n1 COM_V=9600

STEROWNIKI

Liczbę sterowników, z którymi ma się komunikować panel w ramach projektu określa parametr **STER w** pliku "projekt.txt". Maksymalna liczba sterowników to 8.

Sterowniki muszą być opisane w plikach o predefiniowanych nazwach **ster0.txt...ster7.txt**. Pierwszy sterownik projektu ma numer zero i jest opisany w pliku **ster0.txt**. Drugi sterownik opisany jest w pliku **ster1.txt** itd. Liczba plików opisujących sterowniki musi się zgadzać z liczbą przypisaną do parametru **STER**.

Każdy parametr sterownika musi być zapisany w osobnej linii pliku. Wszystkie wymagane parametry sterownika przedstawia poniższa tabela.

Parametr	Opis
SL_ADR=	Adres SLAVE sterownika, z którym komunikuje się panel.
SL_DFORMAT=	Format zapisu liczb 16 bitowych (rejestrów) w sterowniku. Opcje:
	zm - format: znak-moduł. Format używany między innymi w
	sterownikach firmy FRISKO serii: MR65, MR208, MR210, RX910,
	RX911, S20.
	u2 - format: U2 uzupełnienia do dwóch. Format używany w większości
	sterowników dostępnych na rynku, między innymi w sterownikach firmy
	FRISKO serii: ATTO i S1000.
SL_DADR=	Adres pierwszego 16 bitowego rejestru (holding register) dostępnego do
	doczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus
01 00175	
SL_DSIZE=	Liczba dostępnych rejestrow w sterowniku. Panel HMI umozliwia
	odczyt/zapis rejestrow 16 bitowych o adresach od SL_DADR do
	SL_DADR+SL_DSIZE-1. Adrea pierwazai flagi (apil) depterpai de depzytu/zapieu z
SL_FADE	wykorzystaniem protokołu Modbus BTU lub Modbus TCP
SI ESIZE-	Liczba dostepnych flag. Panel HMI umożliwia odczyt/zanis flag. o
	adresach od SL FADR do SL FADR+SL FSIZE-1.
ADR STATUS=	Adres rejestru statusowego w sterowniku. Panel HMI wykorzystuje rejestr
	statusowy do sterowania diodą "STATUS". W przypadku, gdy odczytane
	wartości rejestrów statusowych ze wszystkich zadeklarowanych
	sterowników są równe 0 dioda miga kolorem zielonym. Odczytanie z
	rejestru statusowego dowolnego sterownika wartości różnej od 0
	powoduje mruganie diody kolorem czerwonym.
MAG=	Typ magistrali wykorzystywanej do komunikacji ze sterownikiem SLAVE.
	• com - komunikacja poprzez RS485 (port COM), z wykorzystaniem
	Modbus RTU, Jap komunikacia poprzez sięć LAN (połaszonia HML z siesia
	realizowane przez Wi-Fi) z wykorzystaniem protokołu Modbus TCP.
TOUT=	Czas timeout odpowiedzi (maksymalny czas na odpowiedź sterownika)
	wyrażony w sekundach. Zalecana nastawa TOUT dla MAG=com to
	5001000, dla MAG=lan 20005000.
IP=	Adres IP sterownika. Parametr wymagany, gdy ustawiono MAG=lan.

Pokazany poniżej plik **ster0.txt** opisuje sterownik o adresie SLAVE równym 1 (SL_ADR=1). Sterownik posiada rejestry 16 bitowe dostępne w przestrzeni adresowej 4000...4999 (SL_DADR=4000, SL_DSIZE=1000) oraz

flagi dostępne w przestrzeni adresowej 7000...7999 (SL_FADR=7000, SL_FSIZE=1000). Rejestry zapisywane są w formacie znak-moduł (SL_DFORMAT=zm). Adres rejestru statusowego wynosi 4011 (ADR_STATUS=4011). Komunikacja ze sterownikiem będzie realizowana poprzez magistralę RS485 z wykorzystaniem protokołu Modbus RTU (MAG=com). Timeout odpowiedzi ustawiony na 1 sekundę.

Plik ster0.txt

SL_ADR=1 SL_DFORMAT=zm SL_DADR=4000 SL_DSIZE=1000 SL_FADR=7000 SL_FSIZE=1000 ADR_STATUS=4011 MAG=com TOUT=1000

EKRANY

Maksymalna liczba ekranów, jaką można zdefiniować to 32.

Ekrany muszą być opisane w plikach o predefiniowanych nazwach ekran0.txt ... ekran31.txt.

Pierwszy ekran projektu ma numer zero i jest opisany w pliku **ekran0.txt**. Drugi ekran opisany jest w pliku **ekran1.txt** itd. Liczba plików opisujących ekrany musi się zgadzać z liczbą przypisaną do parametru **EKRANY w** pliku "projekt.txt".

Plik **ekran0.txt** zawiera wszystkie obiekty umieszczone na pierwszym ekranie.

Każdy opis obiektu zaczyna się znacznikiem otwierającym, czyli nagłówkiem z nazwą w ostrych nawiasach: <nazwa> i kończy znacznikiem zamykającym: </>. Przykład:

Przykład pliku ekran0.txt z jednym elementem – nagłówkiem.

```
<nazwa>
TYP=NAGLOWEK
TXT=Menu
KOLOR=25,25,112
L_ICON=0
P_ICON=-1
</>
```

Maksymalna liczba obiektów, jaką można zdefiniować na jednym ekranie to 48.

Dla przejrzystości projektu możemy stosować puste linie pomiędzy kolejnymi dodawanymi obiektami.

Przykład pliku ekran0.txt

<nazwa> TYP=NAGLOWEK TXT=Menu KOLOR=25,25,112 L_ICON=0 P ICON=-1 </> <klaw1> TYP=KLAWISZ_EKRAN NAZWA=Demo1 X=240 Y=55 W=300 H=50 ROWNAJ=centralnie KOLOR=25,25,112 KOLOR_ON=192,192,192 EKRAN=2 STER=0 </> <klaw2> TYP=KLAWISZ_EKRAN NAZWA=Demo2 X=240 Y=105 W=300 H=50 ROWNAJ=centralnie KOLOR=25,25,112 KOLOR_ON=192,192,192 EKRAN=3 STER=0 </> <klaw3> TYP=KLAWISZ_EKRAN NAZWA=Demo3 X=240 Y=155 W=300 H=50 ROWNAJ=centralnie KOLOR=25,25,112 KOLOR_ON=192,192,192 EKRAN=4 STER=0 </>

Komentarze

W plikach opisujących ekran możliwe jest szybkie wyłączenie z kompilacji danej linii poprzez użycie symbolu "//". Jest to znana metoda komentarzowania z języka C.

LISTA REJESTRÓW

W celu wygodniejszego pisania programów na panel HMI35 można wcześniej przygotować sobie listę rejestrów i flag ze zdefiniowanymi nazwami tych rejestrów i flag. Taka mapa pamięci zapisana w pliku rejestry.txt wygląda następująco:

Przykład pliku rejestry.txt

MINUTY;4001 GODZINY;4002 DZIEN_TYG;4006 AW_TAZ;7187 AIN_TZEW;4094 OUT_NG;4211 STAT_PRACY;4112 NAST_PRACA_CTR;4086

Po zdefiniowaniu takiej listy możemy w projekcie stosować w parametrach ADR lub ENABLE zamiast liczby 4002 oznaczającej rejestr, nazwę GODZINY, lub zamiast wartości 4211 nazwę OUT_NG.

Przykład: ADR=GODZINY

OBIEKTY

Obiekty umieszczone na ekranach mogą być różnego typu i maja oczywiście różne parametry.

Typy obiektów, jakie mogą wystąpić w projekcje HMI:

- NAGLOWEK nagłówek ekranu,
- VAR16 wyświetlacz liczby 16-bitowej,
- TXT wyświetlacz tekstu,
- AKCJA_EKRAN akcja zmiany ekranu,
- AKCJA_NC_EKRAN akcja zmiany ekranu bez odświeżania,
- AKCJA_DEC akcja dekrementacji,
- AKCJA_INC akcja inkrementacji,
- AKCJA_SET akcja ustawienia zmiennej,
- AKCJA_FLAG_SET akcja ustawienia flagi,
- KLAWISZ_EKRAN przycisk zmiany ekranu,
- KLAWISZ_NC_EKRAN przycisk zmiany ekranu bez odświeżania,
- KLAWISZ_DEC przycisk dekrementacji,
- KLAWISZ_INC przycisk inkrementacji,
- KLAWISZ_SET przycisk ustawienia zmiennej,
- KLAWISZ_FLAG_SET przycisk ustawienia flagi,
- BMP obrazek typu RGB 24-bitowy
- IKONA obrazek typu mono,
- ONOFF_IKONA migająca ikona,
- LISTA_TXT lista tekstów,
- LISTA_ALERT lista alarmów,
- LISTA_BMP lista obrazków typu RGB 24-bitowy,
- DYNLISTA_BMP dynamiczna lista obrazków typu RGB 24-bitowy,
- LISTA_IKON lista obrazków typu mono,
- PASS obiekt umożliwiający wpisanie hasła,
- PROGRAM interpretacja graficzna programu tygodniowego
- RAMKA prostokątna ramka,
- BOX_ALERT obiekt wyświetlający na całym ekranie alarmy,
- STATUS_COM status komunikacji ze sterownikiem.

Poniżej zostały szczegółowo opisane wszystkie możliwe typy obiektów wraz ze wszystkimi ich parametrami.

Obiekt typu NAGLOWEK – nagłówek ekranu

Obiekt NAGLOWEK zawiera nazwę danego ekranu i umożliwia wyświetlenie dwóch systemowych klawiszy w lewym i prawym górnym rogu. Klawisze te umożliwiają przejście do innych ekranów.

zmienna	opis
TYP=	Typ danego obiektu.
TXT=	Nazwa danego ekranu. Maksymalna długość nazwy to 16 znaków. HMI nie obsługuje polskich znaków.
KOLOR=	Kolor całego nagłówka (nazwy i przycisków) w formacie RGB 24b (24-bitowym). Liczby muszą być rozdzielone przecinkiem.
L_ICON=	Lewy klawisz. Wartość tego parametru określa numer ekranu, do którego ma skoczyć program. Przypisanie wartości -1 powoduje, że dany klawisz nie będzie wyświetlany.
P_ICON=	Prawy klawisz. Opis analogiczny jak powyżej.

Przykład

Pokazany poniżej obiekt <nazwa> opisuje nagłówek ekranu (TYP=NAGLOWEK) o nazwie "Ekran startowy" (TXT=Ekran startowy). Nagłówek ten posiada jeden klawisz po prawej stronie po dotknięciu, którego zostaje wyświetlony ekran o numerze 1 (P_ICON=1). Cały nagłówek ma kolor niebieski (KOLOR=0, 0, 255).

<nazwa> TYP=NAGLOWEK TXT=Ekran startowy KOLOR=0,0,255 L_ICON=-1 P_ICON=1 </>

Obiekt typu VAR16 – wyświetlacz liczby 16-bitowej

Obiekt VAR16 umożliwia wyświetlenie lub edycję parametrów liczbowych 16-bitowych, czyli rejestrów w sterowniku.

zmienna	opis					
TYP=	Typ danego obiektu.					
NAZWA=	Nazwa obiektu, będzie wyświetlana tylko podczas edycji tego parametru.					
	Maksymalna długość	nazwy to 16:	znaków.			
X=	Współrzędna x leweg	o górnego ro	ku wyświetla	nego obiektu. 0 o	oznacza,	
	że obiekt będzie narys	sowany maks	symalnie przy	lewej krawędzi e	kranu.	
Y=	Współrzędna y leweg	o górnego ro	ku wyświetla	nego obiektu. 0 (oznacza,	
	że obiekt będzie narysowany maksymalnie przy górnej krawędzi ekranu.					
ROWNAJ=	Parametr określa, od której strony ma być narysowany obiekt. Opcje:					
	• do lewej - obiekt będzie narysowany od lewej strony czyli					
	współrzędne x i y oznaczają lewy górny róg obiektu.					
	• centralnie - obieł	kt będzie na	rysowany tal	k, że współrzędr	nexiy	
	oznaczają środek gór	nej krawędzi	obiektu,			
	• do prawej – obi	ekt bedzie	narysowany	od prawej stro	ny czyli	
	współrzędne x i y ozn	aczaja prawy	górny róg ob	piektu.	, ,	
MIN=	Minimalna wartość licz	zby, jaka moz	żna wprowadz	zić.		
MAX=	Maksymalna wartość	liczby, jaka m	nożna wprowa	adzić.		
FORMAT=	Format wyświetlania I	iczby. Opcie:				
-	fx1 – X – iedna cvfr	a.				
	• fx2 – XX – dwie cyf	iry,				
	• fx9 – XXXXXXXXX	– dziewięć c	yfr,			
	fx02 – 0X – dwie cy	/fry z zerem v	viodącym,			
	 fx03 – 00X – trzy cyfry z zerami wiodącymi, 					
	 fx09 – 00000000X – dziewięć cyfr z zerami wiodacymi, 					
	fx1.1 – X X – jedna cvfra i jedna cvfra no przecinku					
	 fx2.1 – XX.X – dwie cyfry i jedna cyfra po przecinku, 					
				, ,		
	- i.e. $ -$					
	■ fx02.1 - 0X.X -	dwie cyfry z	zerem wio	dącym i jedna	cyfra po	
	przecinku,					
	• fx03.1 - 00X.X - trzy cyfry z zerami wiodącymi i jedna cyfra po					
	przecinku,					
	• IXU9.I – UUUUUUUUX.X – dziewięc cytr z zerami wiodącymi i jedna					
FONT-	Wielkość czcionki. Or	ncia: 12 16 2	0 24 36 45			
		Joje. 12, 10, 2	.0, 24, 30, 43			
	FONT	wysokość	szerokość	szerokość małych		
	12	12	8	4		
	16	16	11	5		
	20	20	13	6		
	24	24	15	7		
	36	36	24	10		
	45	45	32	14		
	Powyższa tabola moż	a hvé przyda	tna joéli obo	amy na ekranie d	okładnia	
	rozmieścić kilka tak	stów lub ww	świetlaczy n	umerycznych o	różnych	
	rozmiarach ozoionok i różnoj długości znaków. Mozyctkie litery i oviny					
	maja taka sama	szerokoś		atkiem kilku	znaków	
	interpunkcvinvch::	; .itp.				

KOLOR=	Kolor wyświetlanej liczby w formacie RGB 24b (24-bitowym). Wagi					
	poszczególnych kolorów muszą być rozdzielone przecinkiem.					
AKCJA=	Rodzaj wyświetlacza. Opcje:					
	 edit – liczba edytowalna, 					
	brak – tylko wyświetlanie.					
STER=	Numer sterownika, którego dotyczy dana zmienna VAR16.					
	W pliku o podanym numerze zapisane są wszystkie parametry transmisji					
	w tym adres Modbus sterownika.					
	Przykład:					
	STER=U oznacza, ze zmienna dotyczy sterownika opisanego w pliku steru.txt.					
ADR=	Adres 16-bitowego rejestru (noiding register) dostępnego do					
	TCD. Adree musi być z zekreeu zedeklerowenego w pliku sterV tyt adzie					
	Y oznacza numor storownika (SL DADR do SL DADR, SL DSIZE 1) W					
	tvm parametrze jako adres rejestru zamiast warteści liczbowej można					
	nodać nazwe rejestru. W tym celu należy najpierw przygotować plik					
	(rejestry tyt) z taka lista nazw rejestrów					
ENABLE=	Warunek wyświetlania danego obiektu Jeśli chcemy żeby dany obiekt					
	był wyświetlany zawsze to możemy całkowicje pominać linijke z tym					
	warunkiem ENABLE.					
	Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś					
	warunku to podajemy go w tej linijce po słowie ENABLE.					
	Przykłady:					
	ENABLE=4165=1 – obiekt będzie wyświetlany, jeśli w					
	rejestrze 4165 jest wartość równa 1.					
	ENABLE=4165>5 – ODIEKT DĘdZIE WySWIETIANY, JESII W					
	ENABLE=4165<3 - obiekt bedzie wyświetlany, jeśli w					
	rejestrze 4165 jest wartość mniejsza niż 3.					
	ENABLE=4165!0 – obiekt będzie wyświetlany, jeśli w					
	rejestrze 4165 jest wartość różna od 0.					
	rejestrze 4165 jest wartość różna od 0. W warunkach równy '=' lub różny '!', zamiast rejestru może być użyta					
	rejestrze 4165 jest wartość różna od 0. W warunkach równy '=' lub różny '!', zamiast rejestru może być użyta flaga. W tym parametrze jako adres rejestru (flagi) zamiast wartości					
	rejestrze 4165 jest wartość różna od 0. W warunkach równy '=' lub różny '!', zamiast rejestru może być użyta flaga. W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy					

Pokazany poniżej obiekt <godziny> opisuje wyświetlacz 16-bitowego rejestru (TYP=VAR16) o nazwie "Godziny" (NAZWA=Godziny). Obiekt będzie wyświetlany na współrzędnych X=5; Y=60 (X=5, Y=60) podajemy współrzędne lewego górnego rogu obiektu (ROWNAJ=do lewej). Minimalna (MIN=0) i maksymalna (MAX=23) wartość, jaką możemy wprowadzić podczas edycji tego obiektu. Wyświetlana liczba będzie miała dwie cyfry w tym zero wiodące (FORMAT=fx02) i wielkość czcionki (FONT=24) w kolorze czerwonym (KOLOR=255,0,0). Obiekt ten będzie edytowalny (AKCJA=edit) i będzie można zmienić wartość w zadeklarowanym w pliku ster0.txt sterowniku (STER=0) w rejestrze o adresie (ADR=4002). Dodatkowo ten obiekt będzie wyświetlany jeśli w rejestrze o adresie 4165 będzie wartość 1 (ENABLE=4165=1).

<godziny> TYP=VAR16 NAZWA=Godziny X=5 Y=60 ROWNAJ=do lewej MIN=0 MAX=23 FORMAT=fx02 FONT=24 KOLOR=255,0,0 AKCJA=edit STER=0 ADR=4002 ENABLE=4165=1 </>

Obiekt typu TXT – wyświetlacz tekstu

zmienna	Opis						
TYP=	Typ danego obiektu.						
TXT=	Tekst do wyświetlenia. Maksymalna długość to 32 znaki. HMI nie						
	obsługuj	e polskich z	naków. Aby	wyświetlić z	nak stopni Cel	sjusza'°'	
	trzeba uz	trzeba użyć znaku kratki (hash) '#'.					
	Przykład:						
	Jeśli w te	j linijce wpiszei	my TXT=#C to	na ekranie wy	∕świetli się: ℃.		
X=	Współrze	ędna x poz	ycji wyświe	tlanego obie	ektu. W zależ	ności od	
	parameti	ru ROWNAJ	może to być	lewa lub pra	wa krawędź obi	ektu albo	
	środek.						
Y=	Współrze	ędna y pozy	cji wyświetla	inego obiekti	u. O oznacza, z	że obiekt	
	będzie n	arysowany m	aksymalnie p	orzy górnej kra	awędzi ekranu.		
ROWNAJ=	Paramet	r określa, od l	której strony	ma być narys	owany obiekt. O	pcje:	
	■ do le	wej – obie	ekt będzie	narysowany	od lewej stro	ony czyli	
	współrzę	dne x i y ozna	aczają lewy g	jórny róg obie	ektu,		
	 centra 	Inie – obiek	kt będzie na	rysowany tal	<, że współrzęc	lnexiy	
	oznaczaj	a środek góri	nej krawędzi	obiektu,			
	l∎ do pi	r awej – obi	ekt będzie	narysowany	od prawej str	ony czyli	
	współrzę	dne x i y ozna	aczają prawy	gorny róg ob	nektu.		
FONT=	Wielkość	c czcionki, Op	cje: 12, 16, 2	20, 24, 36.			
		FONT	www.ackaćć	a – a valva ć ć	szerokość małych	7	
		FONT	wysokosc	SZEROKOSC	elementów (.,:;)		
		12	12	8	4	_	
		16	16	11	5	_	
		20	20	13	6	-	
		24	24	15	10	-	
		30	30	24	10		
	Powyższ	Powyższa tabela może być przydatna, jeśli chcemy na ekranie dokładnie					
	rozmieśc	cić kilka teks	stów lub wy	świetlaczy n	umerycznych o	różnych	
	rozmiara	ch czcionek	i różnej dług	gości znaków	. Wszystkie lite	ry i cyfry	
	mają	taka samą	szerokoś	ć z wyja	ątkiem kilku	znaków	
	interpunk	kcyjnych:.,:	;.itp.		-		
KOLOR=	Kolor wy	/świetlanego	tekstu w for	rmacie RGB	24b (24-bitowy	m). Wagi	
	poszczeg	gólnych kolor	ów muszą by	ć rozdzielone	przecinkiem.		
STER=	Numer s	terownika, któ	órego dotyczy	/ dany obiekt.			
	W pliku (o podanym ni	umerze zapis	sane są wszy	stkie parametry	transmisji	
	w tym ac	Ires Modbus	sterownika.				
	Numer s	terownika jes	st potrzebny	tylko, jeśli ko	rzystamy z waru	inkowego	
	wyświetla	ania podaneg	o w następne	ej linijce – EN	ABLE.		
	Przykład:	oznacza ża ak	ialit datuatur	tarawaika ania	anaga w pliku ata	·0 +v+	
	Warupok	vyvéwiotlani	neki udiyezy s	ierownika opis vioktu lośli o	anego w pliku slei boomy żoby da	<u>u.ixi.</u> ny obiokt	
	był wyś	viotlany zawa	a uaneyo oo	my całkowie	incenty zeby ud ie nominać linii	ko z tvm	
	warunkie	$m = N \Delta R I =$		any carowic	ie pominąc imj	κę z tym	
	lośli cho	emv abv dai	ny objekt był	wyświatlany	w zależności od	Liakionoś	
	Jesh chicenny, aby dany objekt byr wyswieliany w zalezności od jaklegos						
		,	go w toj innjo				
	Przykła ENADIE-	dy: 1165-1	objokt	hadria www	áuiotlanu -		
	reiestr	70 4165 too	t wartość	równa 1	swielially,	ESTT M	
	ENABLE=	4165>5 -	obiekt I	bedzie wv	świetlanv, T	eśli w	
	rejestr	ze 4165 jes	st wartość	większa niż	5.		
	ENABLE=	4165<3 -	obiekt .	będzie wy	świetlany, j	eśli w	
	rejestr	ze 4165 jes	st wartość .	mniejsza ni	ż 3.		

Obiekt typu TXT wyświetla zwykły tekst w dowolnym miejscu na ekranie.

ENABLE=4165!0	-	obiekt	będzie	wyświetlany,	jeśli	W
rejestrze 4165	jes	t wartość	różna o	d 0.		
W warunkach róv	/ny '	=' lub róż	ny'!', zam	iast rejestru moż	e być uż	yta
flaga. W tym pai	ame	etrze jako	adres reje	estru (flagi) zami	ast warto	ści
liczbowej można	poc	lać nazwę	rejestru	(flagi). W tym	celu nale	эżу
najpierw przygotow	vać	plik (rejesti	ry.txt) z tak	a lista nazw rejes	trów.	-

Pokazany poniżej obiekt <tekst1> opisuje wyświetlacz tekstu (TYP=TXT) z tekstem "°C" (TXT=#C). Obiekt będzie wyświetlany na współrzędnych X=240; Y=200, podajemy współrzędne lewego górnego rogu obiektu (ROWNAJ=do lewej). Wyświetlany tekst będzie miał wielkość czcionki 36 (FONT=36) w kolorze niebieskim (KOLOR=0,0,255). Tekst ten będzie wyświetlany pod warunkiem, że w zadeklarowanym w pliku sterowniku (STER=0) w rejestrze o adresie 4165 będzie wartość równa 1 (ENABLE=4165=1).

<tekst1></tekst1>		
TYP=TXT		
TXT=#C		
X=240		
Y=200		
ROWNAJ=do lewej		
FONT=36		
KOLOR=0,0,255		
STER=0		
ENABLE=4165=1		

Obiekt typu AKCJA_EKRAN i AKCJA_NC_EKRAN – akcja zmiany ekranu

Obiekt AKCJA_EKRAN jest niewidocznym na ekranie prostokątem, który umożliwia przejście do innego ekranu.

zmienna	opis
TYP=	Typ danego obiektu.
	 AKCJA EKRAN – przy zmianie ekranu nastąpi wyczyszczenie całego
	ekranu na biało.
	 AKCJA NC EKRAN – przy zmianie ekranu nie bedzie czyszczony
	ekran na biało.
X=	Współrzędna x pozycji obiektu. W zależności od parametru ROWNAJ
	może to być lewa lub prawa krawędź obiektu albo środek.
Y=	Współrzędna y pozycji obiektu. 0 oznacza, że obiekt będzie umieszczony
	maksymalnie przy górnej krawędzi ekranu.
W=	Szerokość obiektu.
H=	Wysokość obiektu.
ROWNAJ=	Parametr określa, od której strony ma być interpretowana pozycja
	objektu Opcie:
	• do lewei – obiekt bedzie umieszczony od lewei strony czyli
	współrzedne x i v oznaczaja lewy górny róg objektu
	 centralnie – objekt bedzie umjeszczony tak że współrzedne x i y
	oznaczaja środek górnej krawedzi objektu
	• do prawej – objekt bedzie umieszczony od prawej strony czyli
	współrzedne x i v oznaczaja prawy górny róg objektu
FKRAN-	Parametr określa numer ekranu, do którego ma skoczyć dany objekt po
	iego dotknieciu
STER-	Numer sterownika, którego dotyczy dany objekt
STERE	W pliku o podanym numerze zapisane sa wszystkie parametry transmisij
	w tym adres Modbus sterownika
	Numer sterownika jest potrzebny tylko, jeśli korzystamy z warunkowego
	wyświetlania podanego w nastepnej linijce – ENABLE
	Przykład
	STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.
ENABLE=	Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt
	był wyświetlany zawsze to możemy całkowicie pominać linijke z tym
	warunkiem ENABLE.
	Jeśli chcemy, aby dany objekt był wyświetlany w zależności od jakiegoś
	warunku to podajemy go w tej linijce po słowie ENABLE.
	Przykłady:
	ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość równa 1.
	ENABLE=4165>5 – obiekt będzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość większa niż 5.
	ENABLE=4165<3 - obiekt będzie wyswietlany, jesli w
	FNABLF=416510 – objekt bedzie wyświetlany jeśli w
	rejestrze 4165 jest wartość różna od 0.
	W warunkach równy '=' lub różny 'l' zamiast rejestru może być użyta
	flaga W tym parametrze jako adres rejestru (flagi) zamiast wartości
	liczbowej można podać nazwe rejestru (fladi). W tym celu należy
	naipierw przygotować plik (rejestry txt) z taka lista nazw rejestrów

Pokazany poniżej obiekt <akcjal> opisuje niewidzialny przycisk, który umożliwia przejście do innego ekranu (TYP=AKCJA_EKRAN) o numerze trzecim (EKRAN=3). Obiekt ten jest nie widocznym prostokątem o wymiarach 100 na 50 pikseli (W=100, H=50) umieszczonym centralnie (ROWNAJ=centralnie) na współrzędnych x:50 i y:80 (X=50, Y=80). Dodatkowo obiekt będzie aktywny pod warunkiem, że w zadeklarowanym w pliku sterowniku (STER=0) w rejestrze o adresie 4165 będzie wartość równa 1 (ENABLE=4165=1).

<akcjal> TYP=AKCJA_EKRAN X=50 Y=80 W=100 H=50 ROWNAJ=centralnie EKRAN=3 STER=0 ENABLE=4165=1 </>

Obiekt typu AKCJA_DEC – akcja dekrementacji

Obiekt AKCJA_DEC umożliwia odjęcie 1 od wartości wybranego rejestru. Obiekt jest niewidocznym na ekranie prostokątem.

zmienna	opis
TYP=	Typ danego obiektu.
X=	Współrzędna x pozycji obiektu. W zależności od parametru ROWNAJ może to być lewa lub prawa krawedź obiektu albo środek
Y-	Współrzedna v pozycji objektu 0 oznacza że objekt bedzie
	umieszczony maksymalnie przy górnej krawędzi ekranu.
W=	Szerokość obiektu.
H=	Wysokość obiektu.
ROWNAJ=	Parametr określa, od której strony ma być interpretowana pozycja
	obiektu. Opcje:
	• do lewej – obiekt będzie umieszczony od lewej strony czyli
	współrzędne x i y oznaczają lewy górny róg obiektu,
	• centralnie – obiekt będzie umieszczony tak, że współrzędne x i y
	oznaczają środek górnej krawędzi obiektu,
	• do prawej – obiekt będzie umieszczony od prawej strony czyli
	współrzędne x i y oznaczają prawy górny róg obiektu.
MIN=	Minimalna wartość, do której można obniżyć dany rejestr za pomocą tej
	akcji dekrementacji.
STER=	Numer sterownika, którego dotyczy dany obiekt.
	w pliku o podanym numerze zapisane są wszystkie parametry transmisji
	W Tym adres Modbus sterownika.
	STER–0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0 tyt
ADR=	Adres 16-bitowego rejestru (holding register) dostephego do
	odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus
	TCP. Adres musi bvć z zakresu zadeklarowanego w pliku sterX.txt gdzie
	X oznacza numer sterownika (SL DADR do SL DADR+SL DSIZE-1).
	W tym parametrze jako adres rejestru zamiast wartości liczbowej można
	podać nazwę rejestru. W tym celu należy najpierw przygotować plik
	(rejestry.txt) z taką listą nazw rejestrów.
ENABLE=	Warunek aktywności danego obiektu. Jeśli chcemy żeby dany obiekt był
	aktywny zawsze to możemy całkowicie pominąć linijkę z tym warunkiem
	ENABLE.
	Jeśli chcemy, aby dany obiekt był aktywny w zależności od jakiegoś
	warunku to podajemy go w tej linijce po słowie ENABLE.
	PIZYKIADY: FNABLE=4165=1 – objekt bedzie aktywny jeśli w rejestrze
	4165 jest wartość równa 1.
	ENABLE=4165>5 - obiekt będzie aktywny, jeśli w rejestrze
	4165 jest wartość większa niż 5.
	ENABLE=4165<3 - obiekt będzie aktywny, jeśli w rejestrze
	4165 jest wartosc mniejsza niz 3. ENNRIE-416510 – obiekt bodzie aktuuru, jeśli w rejestrze
	4165 jest wartość różna od 0
	W warunkach równy '=' lub różny 'l' zamiast rejestru może być użyta
	flaga. W tym parametrze jako adres rejestru (flagi) zamiast wartości
	liczbowej można podać nazwe rejestru (flagi). W tvm celu należy
	naipierw przygotować plik (rejestry.txt) z taka lista nazw rejestrów.

Pokazany poniżej obiekt <akcja2> opisuje niewidzialny przycisk, który umożliwia dekrementacje rejestru (TYP=AKCJA_DEC) czyli odjęcie 1 od wartości w rejestrze 4103 (ADR=4103) w sterowniku opisanym w piku ster0.txt (STER=0). Minimalna wartość, do której możemy dekrementować to 0 (MIN=0). Obiekt ten jest nie widocznym prostokątem o wymiarach 100 na 50 pikseli (W=100, H=50) umieszczonym na współrzędnych x:50 i y:200 (X=50, Y=200) gdzie te współrzędne określają lewy górny róg prostokąta (ROWNAJ=do lewej).

<akcja2> TYP=AKCJA_DEC X=50 Y=200 W=100 H=50 ROWNAJ=do lewej MIN=0 STER=0 ADR=4103 </>>

Obiekt typu AKCJA_INC – akcja inkrementacji

Obiekt AKCJA_INC umożliwia dodanie 1 do wartości wybranego rejestru. Obiekt jest niewidocznym na ekranie prostokątem.

opis
Typ danego obiektu.
Współrzędna x pozycji obiektu. W zależności od parametru ROWNAJ
może to być lewa lub prawa krawędź obiektu albo środek.
Współrzędna y pozycji obiektu. 0 oznacza, że obiekt będzie
umieszczony maksymalnie przy górnej krawędzi ekranu.
Szerokość obiektu.
Wysokość obiektu.
Parametr określa, od której strony ma być interpretowana pozycja
obiektu. Opcje:
 do lewej – obiekt będzie umieszczony od lewej strony czyli
współrzędne x i y oznaczają lewy górny róg obiektu,
 centralnie – obiekt będzie umieszczony tak, że współrzędne x i y
oznaczają środek górnej krawędzi obiektu,
do prawej – obiekt będzie umieszczony od prawej strony czyli
współrzędne x i y oznaczają prawy górny róg obiektu.
Maksymalna wartość, do której można zwiększyć dany rejestr za
pomocą tej akcji inkrementacji.
Numer sterownika, którego dotyczy dany obiekt.
W pliku o podanym numerze zapisane są wszystkie parametry transmisji
w tym adres Modbus sterownika.
Przykład:
STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.
Adres 16-bitowego rejestru (holding register) dostępnego do
odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus
ICP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie
X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1).
W tym parametrze jako adres rejestru zamiast wartości liczbowej można
podac nazwę rejestru. W tym celu nalezy najpierw przygotowac plik
(rejestry.txt) z taką listą nazw rejestrow.
Warunek aktywności danego obiektu. Jesli chcemy żeby dany obiekt był
aktywny zawsze to możemy całkowicie pominąc linijkę z tym warunkiem
ENABLE. Jośli obcomy, oby dony, objekt był oktywny w zalożności, od jekiegoś
Jesii chcenny, aby dany obiekt był aktywny w zależności od jakiegos
Brauk lodu.
FIZYKIAUY: ENABLE=4165=1 – objekt bedzie aktywny, jeśli w rejestrze
ENABLE=4165=1 – obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość równa 1.
ENABLE=4165=1 – obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 – obiekt będzie aktywny, jeśli w rejestrze
ENABLE=4165=1 – obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 – obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość większa niż 5.
ENABLE=4165=1 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 - obiekt będzie aktywny, jeśli w rejestrze
ENABLE=4165=1 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość mniejsza niż 3.
ENABLE=4165=1 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 - obiekt będzie aktywny, jeśli w rejestrze
<pre>F12ykiady: ENABLE=4165=1 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość różna od 0. W warunkoch równy '' ub różny '' zamiest rejestrze meże być wiete</pre>
<pre>F12ykiady: ENABLE=4165=1 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość różna od 0. W warunkach równy '=' lub różny '!', zamiast rejestru może być użyta flaga W tym parametrze jeke adree rejestru (flagi) zemiast wartości</pre>
<pre>F12ykiady: ENABLE=4165=1 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 - obiekt będzie aktywny, jeśli w rejestrze 4165 jest wartość różna od 0. W warunkach równy '=' lub różny '!', zamiast rejestru może być użyta flaga. W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwe rejestru (flagi) W tym colu paloży.</pre>

Pokazany poniżej obiekt <akcja3> opisuje niewidzialny przycisk, który umożliwia inkrementacje rejestru (TYP=AKCJA_INC) czyli dodanie 1 do wartości w rejestrze 4103 (ADR=4103) w sterowniku opisanym w piku ster0.txt (STER=0). Maksymalna wartość, do której możemy inkrementować do 15 (MAX=15). Obiekt ten jest nie widocznym prostokątem o wymiarach 100 na 100 pikseli (W=100, H=100) umieszczonym na współrzędnych x:300 i y:200 (x=300, Y=200) gdzie te współrzędne określają prawy górny róg prostokąta (ROWNAJ=do prawej).

<akcja3> TYP=AKCJA_INC X=300 Y=200 W=100 H=100 ROWNAJ=do prawej MAX=15 STER=0 ADR=4103 </>>

Obiekt typu AKCJA_SET – akcja ustawienia zmiennej

Obiekt AKCJA_SET umożliwia ustawienie wartości wybranego rejestru. Obiekt jest niewidocznym na ekranie prostokątem.

zmienna	opis
TYP=	Typ danego obiektu.
X=	Współrzędna x pozycji obiektu. W zależności od parametru ROWNAJ
	może to być lewa lub prawa krawędź obiektu albo środek.
Y=	Współrzędna y pozycji obiektu. 0 oznacza, że obiekt będzie
	umieszczony maksymalnie przy górnej krawędzi ekranu.
W=	Szerokość obiektu.
H=	Wysokość obiektu.
ROWNAJ=	Parametr określa, od której strony ma być interpretowana pozycja
	obiektu. Opcje:
	 do lewej – obiekt będzie umieszczony od lewej strony czyli
	współrzędne x i y oznaczają lewy górny róg obiektu,
	• centralnie – obiekt będzie umieszczony tak, że współrzędne x i y
	oznaczają środek górnej krawędzi obiektu,
	 do prawej – obiekt będzie umieszczony od prawej strony czyli
	współrzędne x i y oznaczają prawy górny róg obiektu.
SET=	Wartość, która będzie wpisana do danego rejestru po dotknięciu tego
	obiektu.
STER=	Numer sterownika, którego dotyczy dany obiekt.
	W pliku o podanym numerze zapisane są wszystkie parametry transmisji
	w tym adres Modbus sterownika.
	Przykład:
455	STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.
ADR=	Adres 16-bitowego rejestru (noiding register) dostępnego do zapisu z
	wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi
	byc z zakresu zadekiarowanego w pliku sterX.txt gozle X oznacza numer
	sterownika (SL_DADR do SL_DADR+SL_DSIZE-I). W tym parametrze
	jako adres rejestru zamiast waności liczbowej można podać nazwę
	liete pozw rejectrów
	Marunak aktuwnaćaj danaga abjaktu Jaćli abaamu żabu danu abjakt był
	aktywny zawsza to możomy osłkowicjo pominać linijka z tym warunkiem
	ENARI E
	leśli chcemy, aby dany objekt był aktywny w zależności od jakiegoś
	warunku to podajemy do w tej linijce po słowie ENABLE
	Przykłady:
	ENABLE=4165=1 - obiekt będzie aktywny, jeśli w rejestrze
	4165 jest wartość równa 1.
	ENABLE=4165>5 - obiekt będzie aktywny, jeśli w rejestrze
	4165 jest wartość większa niż 5.
	ENABLE=4165<3 - obiekt będzie aktywny, jesli w rejestrze
	4165 jest wallosc mniejsza niz 5. ENARLE=416510 – objekt bedzie aktywny jeśli w rejestrze
	4165 jest wartość różna od 0.
	W warunkach równy '=' lub różny 'l' zamiast rejestru może być użyta
	flaga. W tym parametrze jako adres rejestru (flagi) zamiast wartości
	liczbowej można podać nazwe reiestru (fladi). W tvm celu należy
	nainierw przygotować plik (rejestry tyt) z taka lista nazw rejestrów

Pokazany poniżej obiekt <akcja4> opisuje niewidzialny przycisk, który umożliwia wpisanie (TYP=AKCJA_SET) wartości 8 (SET=8) do rejestru o adresie (ADR=4103) w sterowniku opisanym w piku ster0.txt (STER=0). Obiekt ten jest nie widocznym prostokątem o wymiarach 100 na 50 pikseli (W=100, H=50) umieszczonym na współrzędnych x:175 i y:250 (X=175, Y=250) gdzie te współrzędne określają lewy górny róg prostokąta (ROWNAJ=do lewej).

<akcja4> TYP=AKCJA_SET X=175 Y=250 W=100 H=50 ROWNAJ=do lewej SET=8 STER=0 ADR=4002 </>

Obiekt typu AKCJA_FLAG_SET – akcja ustawienia flagi

Obiekt AKCJA_FLAG_SET umożliwia ustawienie stanu wybranej flagi. Obiekt jest niewidocznym na ekranie prostokątem.

zmienna	opis
TYP=	Typ danego obiektu.
X=	Współrzędna x pozycji obiektu. W zależności od parametru ROWNAJ
	może to być lewa lub prawa krawędź obiektu albo środek.
Y=	Współrzędna y pozycji obiektu. 0 oznacza, że obiekt będzie
	umieszczony maksymalnie przy górnej krawędzi ekranu.
W=	Szerokość obiektu.
H=	Wysokość obiektu.
ROWNAJ=	Parametr określa, od której strony ma być interpretowana pozycja
	obiektu. Opcie:
	 do lewej – obiekt bedzie umieszczony od lewej strony czyli
	współrzedne x i v oznaczaja lewy górny róg obiektu.
	• centralnie – obiekt będzie umieszczony tak, że współrzędne x i y
	oznaczaja środek górnej krawedzi objektu.
	• do prawej – obiekt będzie umieszczony od prawej strony czyli
	współrzędne x i y oznaczają prawy górny róg obiektu.
SET=	Wartość '0' lub '1', która będzie wpisana do danej flagi po dotknięciu
	tego obiektu.
STER=	Numer sterownika, którego dotyczy dany obiekt.
	W pliku o podanym numerze zapisane są wszystkie parametry transmisji
	w tym adres Modbus sterownika.
	Przykład:
	STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.
ADR=	Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu
	Modbus RTU lub Modbus TCP.
	Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X
	oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W
	tym parametrze jako adres flagi zamiast wartości liczbowej można podać
	nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z
	taką listą nazw rejestrów.
ENABLE=	Warunek aktywności danego obiektu. Jeśli chcemy żeby dany obiekt był
	aktywny zawsze to możemy całkowicie pominąć linijkę z tym warunkiem
	ENABLE.
	Jeśli chcemy, aby dany obiekt był aktywny w zależności od jakiegoś
	warunku to podajemy go w tej linijce po słowie ENABLE.
	Przykłady:
	ENABLE=4165=1 - Oblekt będzie aktywny, jesii w rejestrze
	ENABLE=4165>5 – objekt bedzie aktywny, jeśli w rejestrze
	4165 jest wartość wieksza niż 5.
	ENABLE=4165<3 - obiekt będzie aktywny, jeśli w rejestrze
	4165 jest wartość mniejsza niż 3.
	ENABLE=4165!0 - obiekt będzie aktywny, jeśli w rejestrze
	4165 jest wartość różna od 0.
	W warunkach równy '=' lub różny '!', zamiast rejestru może być użyta
	flaga. W tym parametrze jako adres rejestru (flagi) zamiast wartości
	Liczbowej można podać nazwe rejestru (flagi) W tym celu należy

Pokazany poniżej obiekt <akcja5> opisuje niewidzialny przycisk, który umożliwia ustawienie flagi (TYP=AKCJA_FLAG_SET). Do flagi o adresie (ADR=7083) w sterowniku opisanym w piku ster0.txt (STER=0) zostanie wpisana wartość 1 (SET=1). Obiekt ten jest nie widocznym prostokątem o wymiarach 100 na 50 pikseli (W=100, H=50) umieszczonym na współrzędnych x:300 i y:80 (x=300, Y=80) gdzie te współrzędne określają lewy górny róg prostokąta (ROWNAJ=do lewej).

<akcja5> TYP=AKCJA_FLAG_SET X=300 Y=80 W=100 H=50 ROWNAJ=do lewej SET=1 STER=0 ADR=7083 </>>

Obiekt typu KLAWISZ_EKRAN i KLAWISZ_NC_EKRAN – przycisk zmiany ekranu

Obiekt KLAWISZ_EKRAN jest przyciskiem, który umożliwia przejście do innego ekranu.

zmionna	Onic
	Tura danaga ahiaktu
ITP=	Typ danego obiektu.
	Typ danego obiektu.
	• KLAWISZ_EKRAN – przy zmianie ekranu nastąpi wyczyszczenie
	całego ekranu na blało,
	KLAWISZ_NC_EKRAN – przy zmianie ekranu nie będzie czyszczony
	ekran na biało.
NAZWA=	Wyświetlana nazwa dla danego przycisku. Maksymalna długość nazwy
	to 16 znaków. HMI nie obsługuje polskich znaków.
X=	Współrzędna x pozycji wyświetlanego obiektu. W zależności od
	parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo
	środek.
Y=	Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt
	będzie narysowany maksymalnie przy górnej krawędzi ekranu.
W=	Szerokość obiektu.
H=	Wysokość obiektu.
ROWNAJ=	Parametr określa, od której strony ma być narysowany objekt. Opcie:
	• do lewei – objekt bedzie narysowany od lewei strony czyli
	współrzedne x i v oznaczaja lewy górny róg objektu
	 centralnie – objekt bedzie narvsowany tak że współrzedne x i y
	oznaczaja środek górnej krawedzi objektu
	• do prawei – obiekt bedzie parvsowany od prawei strony czyli
	współrzedne x i v oznaczaja prawy górny róg objektu
	Kolor przycisku w formacje BGB 24b (24-bitowym) Wagi
	poszczególnych kolorów musza być rozdzielone przecinkiem
	Kolor detknietogo przycieku w formacja PCP 24b (24 bitowym). Wagi
KOLON_ON=	Rolof dolkniętego przycisku w formacie RGB 240 (24-bilowym). Wagi
EKDAN	poszczegolitych kolorow muszą byc rozuzielone przecinkiem.
	iogo dotknicoju
OTED	Jego doknieciu.
SIER=	Numer sterownika, ktorego dotyczy dany obiekt.
	w piku o podanym numerze zapisane są wszystkie parametry transmisji
	w tym adres moduls sterownika.
	Numer sterownika jest potrzebny tyjko, jesi korzystamy z warunkowego
	wyswietiania podanego w następnej linijce – ENABLE.
	Pizykłau. STEP-0 oznacza, żo obiekt dotyczy storownika opisanogo w pliku stor0 tyt
ENABLE-	Warunek wyświatania dapogo obiektu lośli choomy żoby dapy obiekt
	był wyświetlany zawsze to możemy całkowicje pominać linijke z tym
	warunkiem ENARI E
	leśli chcemy, aby dany objekt był wyświetlany w zależności od jakiegoś
	warunku to podajomy go w toj lipijco po słowio ENABLE
	Przykłady:
	ENABLE=4165=1 – obiekt bedzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość równa 1.
	ENABLE=4165>5 - obiekt będzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość większa niż 5.
	ENABLE=4165<3 - obiekt będzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość mniejsza niż 3.
	ENABLE=4165!0 – obiekt będzie wyświetlany, jeśli w
	rejestrze 4165 jest wartosc rozna od 0.
	vv warunkach rowny = lub rozny ¹ , zamiast rejestru moze byc uzyta
	liaga. w tym parametrze jako adres rejestru (flagi) zamiast wartości
	liczbowej mozna podać nazwę rejestru (tlagi). W tym celu należy
	najpierw przygotowac plik (rejestry.txt) z taka lista nazw rejestrów.

Pokazany poniżej obiekt <klaw1> opisuje przycisk (klawisz), który umożliwia przejście do innego ekranu (TYP=KLAWISZ_EKRAN) o numerze drugim (EKRAN=2). Obiekt ten jest widocznym prostokątem z zaokrąglonymi rogami i napisem "Ekran2" (NAZWA=Ekran2) w kolorze zielonym (KOLOR=0,255,0) i wymiarach 100 na 50 pikseli (W=100, H=50) umieszczonym na współrzędnych x:50 i y:80 (X=50, Y=80) gdzie są to współrzędne lewego górnego rogu (ROWNAJ=do lewej). Po wciśnięci przycisku zmieni on kolor na ciemno szary (KOLOR_ON=25,25,25). Dodatkowo obiekt będzie aktywny pod warunkiem, że w zadeklarowanym w pliku sterowniku (STER=0) w rejestrze o adresie 4165 będzie wartość 1 (ENABLE=4165=1).

<klaw1>
TYP=KLAWISZ_EKRAN
NAZWA=Ekran2
X=50
Y=80
W=100
H=50
ROWNAJ=do lewej
KOLOR=0,255,0
KOLOR_ON=25,25,25
EKRAN=2
STER=0
ENABLE=4165=1
</></>

Obiekt typu KLAWISZ_DEC – przycisk dekrementacji

Obiekt KLAWISZ_DEC jest przyciskiem, który odejmuje 1 od zadeklarowanej zmiennej.

zmienna	Opis
TYP=	Typ danego obiektu.
NAZWA=	Wyświetlana nazwa dla danego przycisku. Maksymalna długość nazwy
	to 16 znaków. HMI nie obsługuje polskich znaków.
X=	Współrzędna x pozycji wyświetlanego obiektu. W zależności od
	parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo
	środek.
Y=	Współrzedna v pozycji wyświetlanego obiektu. 0 oznacza, że obiekt
	będzie narysowany maksymalnie przy górnej krawędzi ekranu.
W=	Szerokość obiektu.
H=	Wysokość obiektu.
ROWNAJ=	Parametr określa, od której strony ma być narysowany objekt. Opcie:
	• do lewei – obiekt bedzie narvsowany od lewei strony czyli
	współrzedne x i v oznaczaja lewy górny róg objektu.
	• centralnie – objekt bedzie narvsowany tak, że współrzedne x i v
	oznaczaja środek górnej krawedzi objektu
	• do prawei – objekt bedzie narvsowany od prawej strony czyli
	współrzedne x i v oznaczaja prawy górny róg objektu.
KOLOR=	Kolor przycisku w formacie RGB 24b (24-bitowym). Wagi
	poszczególnych kolorów musza być rozdzielone przecinkiem.
KOLOB ON=	Kolor dotknietego przycisku w formacie BGB 24b (24-bitowym) Wagi
	poszczególnych kolorów musza być rozdzielone przecinkiem
MIN=	Minimalna wartość, do której można obniżyć dany rejestr za pomoca tej
	akcii dekrementacii
STER=	Numer sterownika, którego dotyczy dany objekt
01211-	W pliku o podanym numerze zapisane sa wszystkie parametry transmisii
	w tym adres Modbus sterownika <i>Przykład</i>
	STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.
ADR=	Adres 16-bitowego rejestru (holding register) dostepnego do
	odczytu/zapisu z wykorzystaniem protokołu Modbus BTU lub Modbus
	TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie
	X oznacza numer sterownika (SL DADR do SL DADR+SL DSIZE-1).
	W tym parametrze jako adres rejestru zamiast wartości liczbowej można
	podać nazwę rejestru. W tym celu należy najpierw przygotować plik
	(rejestry.txt) z taką listą nazw rejestrów.
ENABLE=	Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt
	był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym
	warunkiem ENABLE.
	Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś
	warunku to podajemy go w tej linijce po słowie ENABLE.
	Przykłady:
	ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość równa 1.
	ENABLE=4165>5 - ODIEKL DĘdzie wyswieliany, jesii w
	ENABLE=4165<3 - objekt bedzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość mniejsza niż 3.
	ENABLE=4165!0 – obiekt będzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość różna od 0.
	W warunkach równy '=' lub różny '!', zamiast rejestru może być użyta
	flaga. W tym parametrze jako adres rejestru (flagi) zamiast wartości
	liczbowej można podać nazwę rejestru (flagi). W tym celu należy
	najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.

Pokazany poniżej obiekt <klaw2> opisuje przycisk (klawisz), który umożliwia dekrementacje rejestru (TYP=KLAW_DEC), czyli odjęcie 1 od wartości w rejestrze 4103 (ADR=4103) w sterowniku opisanym w piku ster0.txt (STER=0). Minimalna wartość, do której możemy dekrementować to 0 (MIN=0). Obiekt ten jest widocznym prostokątem z zaokrąglonymi rogami i napisem "Dec" (NAZWA=Dec) w kolorze niebieskim (KOLOR=0,0,255) i wymiarach 100 na 100 pikseli (W=100, H=100) umieszczonym na współrzędnych x:50 i y:200 (X=50, Y=200) gdzie są to współrzędne lewego górnego rogu (ROWNAJ=do lewej). Po wciśnięci przycisku zmieni on kolor na ciemno szary (KOLOR_ON=25,25,25).

<klaw2>
TYP=KLAWISZ_DEC
NAZWA=Dec
X=50
Y=200
W=100
H=100
ROWNAJ=do lewej
KOLOR=0,0,255
KOLOR_ON=25,25,25
MIN=0
STER=0
ADR=4103
</>></>>

Obiekt typu KLAWISZ_INC – przycisk inkrementacji

Obiekt KLAWISZ_INC jest przyciskiem, który dodaje 1 do zadeklarowanej zmiennej.

zmienna	Opis
TYP=	Typ danego objektu.
NAZWA=	Wyświetlana nazwa dla danego przycisku. Maksymalna długość nazwy
	to 16 znaków. HMI nie obsługuje polskich znaków.
X=	Współrzedna x pozycji wyświetlanego obiektu. W zależności od
	parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo
	środek.
Y=	Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt
	będzie narysowany maksymalnie przy górnej krawędzi ekranu.
W=	Szerokość obiektu.
H=	Wysokość obiektu.
ROWNAJ=	Parametr określa, od której strony ma być narysowany obiekt. Opcje:
	• do lewej – obiekt bedzie narysowany od lewej strony czyli
	współrzędne x i y oznaczają lewy górny róg obiektu,
	• centralnie – obiekt będzie narysowany tak, że współrzędne x i y
	oznaczają środek górnej krawędzi obiektu,
	• do prawej – obiekt będzie narysowany od prawej strony czyli
	współrzędne x i y oznaczają prawy górny róg obiektu.
KOLOR=	Kolor przycisku w formacie RGB 24b (24-bitowym). Wagi
	poszczególnych kolorów muszą być rozdzielone przecinkiem.
KOLOR_ON=	Kolor dotkniętego przycisku w formacie RGB 24b (24-bitowym). Wagi
	poszczególnych kolorów muszą być rozdzielone przecinkiem.
MAX=	Maksymalna wartość, do której można zwiększyć dany rejestr za pomocą
	tej akcji inkrementacji.
STER=	Numer sterownika, którego dotyczy dany obiekt.
	W pliku o podanym numerze zapisane są wszystkie parametry transmisji
	w tym adres Modbus sterownika. Przykład:
	STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.
ADR=	Adres 16-bitowego rejestru (holding register) dostępnego do
	odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus
	I CP. Adres musi byc z zakresu zadeklarowanego w pilku sterx.txt gdzle
	X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W
	tym parametrze jako adres rejestru zamast wartosci liczbowej mozna
	(rejestru tyt) z taka lista pazw rejestrów
	Marunak wyówiatlania dapaga obiektu lośli obcomy żoby dapy obiekt
	był wyświetlany zawsze te możemy całkowicje pominać linijke z tym
	warunkiem ENARI E
	leśli chcemy, aby dany objekt był wyświetlany w zależności od jakiegoś
	warunku to podajemy go w tej linijce po słowie ENABLE
	Przykłady:
	ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość równa 1.
	ENABLE=4165>5 – obiekt będzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość większa niż 5.
	ENADLE=4103<3 - ODIEKT DĘdZIE WYSWIETIANY, JESII W rejestrze 4165 jest wartość mniejsza niż 3
	ENABLE=4165!0 – obiekt bedzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość różna od 0.
	W warunkach równy '=' lub różny '!', zamiast rejestru może być użyta
	flaga. W tym parametrze jako adres rejestru (flagi) zamiast wartości
	liczbowej można podać nazwę rejestru (flagi). W tym celu należy
	najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.

Pokazany poniżej obiekt <klaw3> opisuje przycisk (klawisz), który umożliwia inkrementacje rejestru (TYP=KLAW_INC), czyli dodanie 1 do wartości w rejestrze 4103 (ADR=4103) w sterowniku opisanym w piku ster0.txt (STER=0). Maksymalna wartość, do której możemy inkrementować to 15 (MAX=15). Obiekt ten jest widocznym prostokątem z zaokrąglonymi rogami i napisem "Inc" (NAZWA=Inc) w kolorze czerwonym (KOLOR=255,0,0) i wymiarach 100 na 100 pikseli (W=100, H=100) umieszczonym na współrzędnych x:300 i y:200 (X=300, Y=200) gdzie są to współrzędne lewego górnego rogu (ROWNAJ=do lewej). Po wciśnięci przycisku zmieni on kolor na ciemno szary (KOLOR_ON=25,25,25).

<klaw3>
TYP=KLAWISZ_INC
NAZWA=Inc
X=300
Y=200
W=100
H=100
ROWNAJ=do lewej
KOLOR=250,25,25
KOLOR_ON=25,25,25
MAX=15
STER=0
ADR=4103
</>></>>

Obiekt typu KLAWISZ_SET – przycisk ustawienia zmiennej

Obiekt KLAWISZ_SET jest przyciskiem, który ustawia wartość zadeklarowanej zmiennej.

zmienna	opis
TYP=	Typ danego obiektu.
NAZWA=	Wyświetlana nazwa dla danego przycisku. Maksymalna długość nazwy to 16 znaków. HMI nie obsługuje polskich znaków.
X=	Współrzędna x pozycji wyświetlanego obiektu. W zależności od parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo środek.
Y=	Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt będzie narysowany maksymalnie przy górnej krawędzi ekranu.
W=	Szerokość obiektu.
H=	Wysokość obiektu.
ROWNAJ=	 Parametr określa, od której strony ma być narysowany obiekt. Opcje: do lewej – obiekt będzie narysowany od lewej strony czyli współrzędne x i y oznaczają lewy górny róg obiektu, centralnie – obiekt będzie narysowany tak, że współrzędne x i y oznaczają środek górnej krawędzi obiektu, do prawej – obiekt będzie narysowany od prawej strony czyli współrzędne x i y oznaczają prawy górny róg obiektu.
KOLOR=	Kolor przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem.
KOLOR_ON=	Kolor dotkniętego przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem.
SET=	Wartość, która będzie wpisana do danego rejestru po dotknięciu tego przycisku.
STER=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i>
ADR=	Adres 16-bitowego rejestru (holding register) dostępnego do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.
ENABLE=	Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE.
	Przykłady: ENABLE=4165=1 – obiekt będzie wyświetlany, jeśli w
	EJESTIZE 4165 JEST WAILOSC LOWNA I. ENABLE=4165>5 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5.
	ENABLE=4165<3 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3.
	 ENABLE=4165!0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość różna od 0. W warunkach równy '=' lub różny '!', zamiast rejestru może być użyta flaga. W tym parametrze iako adres rejestru (flagi) zamiast wartości
	liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.

Pokazany poniżej obiekt <klaw4> opisuje przycisk (klawisz), który umożliwia wpisanie (TYP=KLAWISZ_SET) wartości 8 (SET=8) do rejestru o adresie (ADR=4103) w sterowniku opisanym w piku ster0.txt (STER=0). Obiekt ten jest widocznym prostokątem z zaokrąglonymi rogami i napisem "Set:8" (NAZWA=Set:8) w kolorze zielonym (KOLOR=0,255,0) i wymiarach 100 na 50 pikseli (W=100, H=50) umieszczonym na współrzędnych x:175 i y:250 (X=175, Y=250) gdzie te współrzędne określają lewy górny róg prostokąta (ROWNAJ=do lewej). Po wciśnięci przycisku zmieni on kolor na ciemno szary (KOLOR_ON=25,25,25).

<klaw4>
TYP=KLAWISZ_SET
NAZWA=Set:8
X=175
Y=250
W=100
H=50
ROWNAJ=do lewej
KOLOR=0,255,0
KOLOR_ON=25,25,25
SET=8
STER=0
ADR=4103
</>></>>

Obiekt typu KLAWISZ_FLAG_SET – przycisk ustawienia flagi

Obiekt KLAWISZ_FLAG_SET jest przyciskiem, który ustawia wartość zadeklarowanej flagi.

TYP= Typ danego obiektu. NAZWA= Wyświetlana nazwa dla danego przycisku. Maksymalna długość nazwy to 16 znaków. HMI nie obsługuje polskich znaków. X= Współrzędna x pozycji wyświetlanego obiektu. W zależności od parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo środek. Y= Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt będzie narysowany maksymalnie przy górnej krawędzi ekranu. W= Szerokość obiektu. H= Wysokość obiektu. ROWNAJ= Parametr określa, od której strony ma być narysowany obiekt. Opcje: • do lewej – obiekt będzie narysowany od lewej strony czyli współrzędne x i y oznaczają lewy górny róg obiektu. • centralnie – obiekt będzie narysowany od prawej strony czyli współrzędne x j oznaczają prawy górny róg obiektu. KOLOR= Kolor przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. SET= Stan, który będzie wpisany do danej flagi po dotknięciu tego przycisku. Możliwe stany 0 albo 1. STER= Numer sterownika, (SL_FADR do SL_FADR do SL_FALR dzie. Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadekłarowanego w pliku ster0.bt. ADR= Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres mu rejestrów. ENABLE	zmienna	opis
NAZWA= Wyświetlana nazwa dla danego przycisku. Maksymalna długość nazwy to 16 znaków. HMI nie obsługuje polskich znaków. X= Współrzędna x pozycji wyświetlanego obiektu. W zależności od parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo środek. Y= Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt będzie narysowany maksymalnie przy górnej krawędzi ekranu. W= Szerokość obiektu. H= Wysokość obiektu. ROWNAJ= Parametr okresla, od której strony ma być narysowany obiekt. Opcje: • do lewej – obiekt będzie narysowany od lewej strony czyli współrzędne x i y oznaczają środek górnej krawędzi obiektu, • centralnie – obiekt będzie narysowany d prawej strony czyli współrzędne x i y oznaczają prawg górny róg obiektu. KOLOR= Kolor przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. SET= Stan, który będzie wpisany do danej flagi po dotknięciu tego przycisku. Możliwe stany 0 albo 1. STER= Numer sterownika, którego dotyczy sterownika opisanego w pliku ster0.txt. Możliwe stany 0 albo 1. STER= Numer sterownika (SL_FADR do zapisau z wykorzystaniem protokolu Modbus RTU lub Modbus TCP. Adres flagi (coli) dostępnej do zapisu z wykorzystaniem protokolu Modbus RTU lub Modbus TCP. Adres flagi (coli) dostępnej do zapisu z wykorzystaniem protokolu Modbus RTU lub Modbus TCP. Adres flagi zamiast wartości liczbowej polizkt z dzie X oznacza rejest wzy dako dores flagi zamiast	TYP=	Typ danego obiektu.
X= Współrzędna x pozycji wyświetlanego obiektu. W zależności od parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo środek. Y= Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt będzie narysowany maksymalnie przy górnej krawędzi ekranu. W= Szerokość obiektu. H= Wysokość obiektu. ROWNAJ= Parametr określa, od której strony ma być narysowany obiekt. Opcje: • do lewej – obiekt będzie narysowany od lewej strony czyli współrzędne x i y oznaczają prawg dorny róg obiektu. • centralnie – obiekt będzie narysowany at k, że współrzędne x i y oznaczają prawg dorny róg obiektu. Kolor przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. Ster= Stan, którego dotyczy dany obiekt. W pilku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykła: STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pilku ster0.txt.</i> Adres musi być z zakresu zadeklarowanego w pilku ster0.txt. Adres musi być z zakresu zadeklarowanego w pilku ster0.txt. Adres musi być z zakresu zadeklarowanego w pilku ster0.txt. zaką listą nazw rejestrów. ENABLE= Warunek wyświetlanią canego obiektu będzie pomiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować pilk (rejestry.txt) z taką listą nazw rejestrów. ENABLE= Warunek wyświetlany zawsze to możeny całkowicie pomiaąć linjikę z tym warunku to podajem y dw tej linjice po słowie ENABLE. <i>Przykłady:</i> ENABLE=165/55 – obiekt będzie wyświetlany, jeśli w rejestr	NAZWA=	Wyświetlana nazwa dla danego przycisku. Maksymalna długość nazwy to 16 znaków. HMI nie obsługuje polskich znaków.
Y= Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt będzie narysowany maksymalnie przy górnej krawędzi ekranu. W= Szerokość obiektu. ROWNAJ= Parametr określa, od której strony ma być narysowany obiekt. Opcje: • do lewej – obiekt będzie narysowany od lewej strony czyli współrzędne x i y oznaczają lewy górny róg obiektu, • centralnie – obiekt będzie narysowany tak, że współrzędne x i y oznaczają środek górnej krawędzi obiektu, • do prawej – obiekt będzie narysowany od prawej strony czyli współrzędne x i y oznaczają prawy górny róg obiektu. KOLOR= Kolor przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. SET= Stan, który będzie wpisany do danej flagi po dotknięciu tego przycisku. Możliwe stany 0 albo 1. STER= Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> STER=0 czacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt ADR= Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika. (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast watości liczbowej można podać nazwę flagi. W tym celu należy najbierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. ENABLE= Warunek wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany, ześli w rejestrze 4165 jest wartość równa 1. ENABLE=416551 – obiekt będzie wyświetlany, jeśli w rej	X=	Współrzędna x pozycji wyświetlanego obiektu. W zależności od parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo środek.
W= Szerokość obiektu. H= Wysokość obiektu. ROWNAJ= Parametr określa, od której strony ma być narysowany obiekt. Opcje: 	Y=	Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt będzie narysowany maksymalnie przy górnej krawędzi ekranu.
H= Wysokość obiektu. ROWNAJ= Parametr określa, od której strony ma być narysowany obiekt. Opcje: • do lewej – obiekt będzie narysowany od lewej strony czyli współrzędne x i y oznaczają lewy górny róg obiektu, • centralnie – obiekt będzie narysowany tak, że współrzędne x i y oznaczają środek górnej krawędzi obiektu, • do prawej – obiekt będzie narysowany od prawej strony czyli współrzędne x i y oznaczają prawy górny róg obiektu. KOLOR= Kolor przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. KOLOR_ON= Kolor dotkniętego przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. SET= Stan, który będzie wpisany do danej flagi po dotknięciu tego przycisku. Możliwe stany 0 albo 1. STER= Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład: STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> ADR= Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. ENABLE= Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunku to podajemy gow tej linijce po słowie ENABLE. <i>Przykłady:</i> ENABL	W=	Szerokość obiektu.
ROWNAJ= Parametr określa, od której strony ma być narysowany obiekt. Opcje: • do lewej obiekt będzie narysowany od lewej strony czyli współrzędne x i y oznaczają środek górnej krawędzi obiektu, • centralnie obiekt będzie narysowany tak, że współrzędne x i y oznaczają środek górnej krawędzi obiektu, • do prawej obiekt będzie narysowany tak, że współrzędne x i y oznaczają środek górnej krawędzi obiektu, • kolor otkniętego przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. KOLOR_ON= Kolor dotkniętego przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. SET= Stan, który będzie wpisany do danej flagi po dotknięciu tego przycisku. Możliwe stany 0 albo 1. STER= Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> STER= Numer sterownika, którego dotyczy sterownika opisanego w pliku ster0.txt. ADR= Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku ster2.txt gdzie X oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. ENABLE= Warunek wyświetlani danego obiektt. był wyświetlany, jeśli w rejestrze 4165	H=	Wysokość obiektu.
 do lewej – obiekt będzie narysowany od lewej strony czyli współrzędne x i y oznaczają iewy górny róg obiektu, centralnie – obiekt będzie narysowany tak, że współrzędne x i y oznaczają środek górnej krawędzi obiektu, do prawej – obiekt będzie narysowany od prawej strony czyli współrzędne x i y oznaczają frawy górny róg obiektu. Kolcor przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. KOLOR= Kolor dotkniętego przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. Stan, który będzie wpisany do danej flagi po dotknięciu tego przycisku. Możliwe stany 0 albo 1. STER= Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład: STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt</i>. Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokolu Modbus STCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. ENABLE= Warunek wyświetlania danego obiektu Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pomiąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1.5. ENABLE=4165/1 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165/1 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165/1 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsz	ROWNAJ=	Parametr określa, od której strony ma być narysowany obiekt. Opcje:
 Centralme – objekt będzie narysowany od prawej strony czyli vspółrzędne x i y oznaczają środk sórnej krawędzi obiektu. do prawej – obiekt będzie narysowany od prawej strony czyli współrzędne x i y oznaczają prawy górny róg obiektu. KOLOR= Kolor przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. KOLOR_ON= Kolor dotkniętego przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. SET= Stan, który będzie wpisany do danej flagi po dotknięciu tego przycisku. Możliwe stany 0 albo 1. STER= Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> STEF-0 <i>oznacza, ze obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku ster2.txt gdzie X oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. ENABLE= Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=4165-1 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165-5 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość wieksza niż 5. ENABLE=4165-10 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość wieksza niż		 do lewej – obiekt będzie narysowany od lewej strony czyli współrzędne x i y oznaczają lewy górny róg obiektu, centralnie – obiekt będzie narysowany tak że współrzędne x i y
Image: State in the intervent of the inte		- Centralme – obleki będzie narysowany tak, że wspoliżędne x r y
StorFinally of a prawej strony of prawej strony of the prawej strony of the prawej strony of the prawej strony of the presentation of the pre		do prawej – objekt bedzie parvsowany od prawej strony czyli
KOLOR= Kolor przycisku w formacie HGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. KOLOR_ON= Kolor dotkniętego przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. SET= Stan, który będzie wpisany do danej flagi po dotknięciu tego przycisku. Możliwe stany 0 albo 1. STER= Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. Przykład: STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt. ADR= Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. ENABLE= Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunku to podajemy go w tej linijce po słowie ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165 ENABLE=4165 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3.		współrzędne x i y oznaczają prawy górny róg obiektu.
KOLOR_ON= Kolor dotkniętego przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem. SET= Stan, który będzie wpisany do danej flagi po dotknięciu tego przycisku. Możliwe stany 0 albo 1. STER= Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład: STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> ADR= Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. ENABLE= Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. <i>Przykłady:</i> <i>ENABLE=4165-5 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1.</i> <i>ENABLE=4165-5 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość minejsza niż 3.</i> <i>ENABLE=4165:0 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość minejsza niż 3.</i> <i>ENABLE=4165:0 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość minejsza niż 3.</i> <i>ENABLE=4165:0 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość minejsza niż 3.</i> <i>ENABLE=4165:0 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość minejsza niż 3.</i>	KOLOR=	Kolor przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem.
SET= Stan, który będzie wpisany do danej flagi po dotknięciu tego przycisku. Możliwe stany 0 albo 1. STER= Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. Przykład: STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt. ADR= Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. ENABLE= Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=41655 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=41655 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=41655 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość wieksza niż 5. ENABLE=41655 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość wieksza niż 3. ENABLE=416510 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość różna od 0.	KOLOR_ON=	Kolor dotkniętego przycisku w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem.
Możliwe stany 0 albo 1.STER=Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.ADR=Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.ENABLE=Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. <i>Przykłady:</i> ENABLE=4165=1 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165<3 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165	SET=	Stan, który będzie wpisany do danej flagi po dotknięciu tego przycisku.
STER=Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. Przykład: STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.ADR=Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.ENABLE=Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=4165>5 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165 S - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość różna od 0.		Możliwe stany 0 albo 1.
W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. Przykład: STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.ADR=Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.ENABLE=Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunku to podajemy go w tej linijce po słowie ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=416555 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 5. ENABLE=416553 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=416550 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=416510 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=416510 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=41655 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=416510 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=416510 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=416510 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3.	STER=	Numer sterownika, którego dotyczy dany obiekt.
w tym adres Modbus sterownika. Przykład: STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.ADR=Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.ENABLE=Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=416555 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165ENABLE=4165/0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość minejsza niż 3. ENABLE=4165/0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość minejsza niż 3. ENABLE=4165/0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość minejsza niż 3. ENABLE=4165/0 - obiekt będzie wyświetlany, jeśli w		W pliku o podanym numerze zapisane są wszystkie parametry transmisji
STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.ADR=Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.ENABLE=Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165ENABLE=4165- obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość minejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość minejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość minejsza niż 3.		w tym adres Modbus sterownika. Przykład:
ADR=Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.ENABLE=Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165ENABLE=4165- obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość minejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość minejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość minejsza niż 3.		STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.
Modbus RTU lub Modbus TCP.Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.ENABLE=Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165 S - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3.	ADR=	Adres flagi (coil) dostępnej do zapisu z wykorzystaniem protokołu
Adres musi być z zakresu zadeklarowanego w piku sterx.kt gdzie x oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.ENABLE=Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 5. ENABLE=4165 S - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3.		Modbus RTU lub Modbus TCP. Adres, reusi, buć, z zakrosu, zadaklarowana w pliku, starX tut odzie. X
Oznacza numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres flagi zamiast wartości liczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.ENABLE=Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt 		Adres musi byc z zakresu zadekiarowanego w pliku sterX.txt gozie X
ENABLE=Variative jako adres hagi zamiast wartoschiczbowej można podać nazwę flagi. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.ENABLE=Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165ENABLE=4165obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3.		02/1ac2a numer sterownika (SL_FADR do SL_FADR+SL_FSIZE-1). W
ENABLE=Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165 S - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3. ENABLE=4165:0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość miejsza niż 3.		tym parametrze jako aures nagi zamiast wartości liczbowej można podac pazwe flagi. W tym colu paloży pajniorw przygotować plik (rojestry tyt) z
ENABLE=Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś 		taka lista nazw rejestrów
był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=4165=1 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3.	ENABLE=	Warunek wyświetlania danego obiektu Jeśli chcemy żeby dany obiekt
<pre>warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3.</pre>		był wyświetlany zawsze to możemy całkowicie pominać linijke z tym
Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=4165=1 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość różna od 0.		warunkiem ENABLE.
<pre>warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość różna od 0.</pre>		Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś
Przykłady: ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość różna od 0.		warunku to podajemy go w tej linijce po słowie ENABLE.
ENABLE=4165=1 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość różna od 0.		Przykłady:
ENABLE=4165 jest wartość równa 1. ENABLE=4165>5 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość różna od 0.		ENABLE=4165=1 – obiekt będzie wyświetlany, jeśli w
rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość różna od 0.		rejestrze 4165 jest wartosc rowna 1. ENABLE=4165>5 – objekt bedzie wyświetlany jeśli w
ENABLE=4165<3 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość różna od 0.		rejestrze 4165 jest wartość wieksza niż 5.
rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165!0 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość różna od 0.		ENABLE=4165<3 – obiekt będzie wyświetlany, jeśli w
ENABLE=4165!0 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość różna od 0.		rejestrze 4165 jest wartość mniejsza niż 3.
rejestrze 4165 jest Wartosć rozna od U.		ENABLE=4165!0 - obiekt będzie wyświetlany, jeśli w
W worupkach równy (-) lub różny (!) zomiast rojasty, moża być użuta		rejestrze 4165 jest wartosc rozna od U. W warupkach równy '-' lub różny 'l' zamiast rojastru maża być użyta
flaga W tym parametrze jako adres rojestru (flagi) zamiast worteści		flaga W tym parametrze jako adres rejestru (flagi) zamiast wartości
liczhowej możną podąć nazwe rejestru (fladi) W tym celu należy		liczbowej można podać nazwe rejestru (flagi). W tym celu należy
najpierw przvootować plik (rejestry.txt) z taka lista nazw rejestrów.		najpierw przygotować plik (rejestry.txt) z taka lista nazw rejestrów.

Pokazany poniżej obiekt <klaw5> opisuje przycisk (klawisz), który umożliwia ustawienie flagi (TYP=KLAWISZ_FLAG_SET). Do flagi o adresie (ADR=7083) w sterowniku opisanym w piku ster0.txt (STER=0) zostanie wpisana wartość 1 (SET=1). Obiekt ten jest widocznym prostokątem zaokrąglonymi rogami i napisem "Serwis" (NAZWA=Serwis) w kolorze zielonym (KOLOR=0,255,0) i wymiarach 100 na 50 pikseli (W=100, H=50) umieszczonym na współrzędnych x:300 i y:80 (X=300, Y=80) gdzie te współrzędne określają lewy górny róg prostokąta (ROWNAJ=do lewej). Po wciśnięci przycisku zmieni on kolor na ciemno szary (KOLOR_ON=25,25,25).

<klaw5>
TYP=KLAWISZ_FLAG_SET
NAZWA=Serwis
X=300
Y=80
W=100
H=50
ROWNAJ=do lewej
KOLOR=0,255,0
KOLOR_ON=25,25,25
SET=1
STER=0
ADR=7083
</>></>

Obiekt typu BMP – obrazek typu RGB 24-bitowy

Obiekt typu **BMP** umożliwia wyświetlenie obrazka na podstawie pliku graficznego 24-bitowego o rozszerzeniu .bmp.

zmienna	opis
TYP=	Typ danego obiektu.
X=	Współrzędna x pozycji wyświetlanego obiektu. W zależności od
	parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo
	środek.
Y=	Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt
	będzie narysowany maksymalnie przy górnej krawędzi ekranu.
W=	Szerokość obiektu. Trzeba podać rzeczywisty rozmiar obrazka. Jeśli
	obrazek ma mieć inne wymiary trzeba go przyciąć w programie
	graficznym i podać tutaj nowe wymiary. Kompilator nie przycina
	oryginalnych rozmiarów obraza.
H=	Wysokość obiektu. Trzeba podać rzeczywisty rozmiar obrazka. Jeśli
	obrazek ma mieć inne wymiary trzeba go przyciąć w programie
	graficznym i podać tutaj nowe wymiary. Kompilator nie przycina
	oryginalnych rozmiarów obraza.
ROWNAJ=	Parametr określa, od której strony ma być narysowany obiekt. Opcje:
	 do lewej – obiekt będzie narysowany od lewej strony czyli
	współrzędne x i y oznaczają lewy górny róg obiektu,
	 centralnie – obiekt będzie narysowany tak, że współrzędne x i y
	oznaczają środek górnej krawędzi obiektu,
	 do prawej – obiekt będzie narysowany od prawej strony czyli
	współrzędne x i y oznaczają prawy górny róg obiektu.
BMP=	Nazwa pliku graficznego 24-bitowego z rozszerzeniem .bmp. Plik ten
	musi być umieszczony w tym samym katalogu co cały projekt.
STER=	Numer sterownika, którego dotyczy dany obiekt.
	W pliku o podanym numerze zapisane są wszystkie parametry transmisji
	w tym adres Modbus sterownika.
	Numer sterownika jest potrzebny tylko, jeśli korzystamy z warunkowego
	wyświetlania podanego w następnej linijce – ENABLE.
	Przykład: STER 0 oznacza, ża obielu dotyczy starownika opiecnogo w pliku star0 tyt
	STER=0 oznacza, ze obiekt dolyczy sterownika opisanego w pliku stero.txi.
ENADLE=	waluliek wyświetlania udliego obiektu. Jesli cicelny zeby udliy obiekt
	warunkiom ENARI E
	leśli chcemy, aby dany objekt był wyświetlany w zależności od jakiegoś
	warunku to podajemy do w tej linijce po słowie ENARI E
	Przykłady:
	ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość równa 1.
	ENABLE=4165>5 – obiekt będzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość większa niż 5.
	ENABLE=4165<3 - Obiekt będzie wyswietlany, jesli w
	ENABLE=4165/0 – objekt bedzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość różna od 0.
	W warunkach równy '=' lub różny '!', zamiast reiestru może bvć użvta
	flaga. W tym parametrze jako adres rejestru (flagi) zamiast wartości
	liczbowej można podać nazwę rejestru (flagi). W tym celu należy
	najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.

Pokazany poniżej obiekt <obraz1> opisuje wyświetlany obrazek (TYP=BMP) na podstawie 24-bitowego pliku graficznego o nazwie "grafika24.bmp" (BMP=grafika24) o wymiarach 50 na 50 pikseli (W=50, H=50). Będzie on wyświetlany na współrzędnych x:80 i y:80 (X=80, Y=80), podane współrzędne dotyczą środka górnej krawędzi (ROWNAJ=centralnie). Obrazek ten będzie wyświetlany pod warunkiem, że w zadeklarowanym w pliku sterowniku (STER=0) w rejestrze o adresie 4103 będzie wartość większa od 5 (ENABLE=4103>5).

<obraz1>
TYP=BMP
X=80
Y=80
W=50
H=50
ROWNAJ=centralnie
BMP=grafika24
STER=0
ENABLE=4103>5
</>></>>

Obiekt typu IKONA – obrazek typu mono

Obiekt typu **IKONA** jest obiektem mono czyli jednokolorowym. Umożliwia wyświetlenie obrazka na podstawie pliku graficznego 24-bitowego o rozszerzeniu .bmp. Każdy bit w kolorze innym niż biały traktowany jest jako "1" i rysowany jest w kolorze zdefiniowanym w obiekcie. Plik kolorowy w czasie kompilacji przekonwertowany będzie do postaci mono, czyli jednokolorowej ikony.

zmienna	
TYP=	Typ danego obiektu.
X=	Współrzędna x pozycji wyświetlanego obiektu. W zależności od parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo środek.
Y=	Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt będzie narysowany maksymalnie przy górnej krawędzi ekranu.
W=	Szerokość obiektu. Trzeba podać rzeczywisty rozmiar obrazka. Jeśli obrazek ma mieć inne wymiary trzeba go przyciąć w programie graficznym i podać tutaj nowe wymiary. Kompilator nie przycina oryginalnych rozmiarów obraza.
H=	Wysokość obiektu. Trzeba podać rzeczywisty rozmiar obrazka. Jeśli obrazek ma mieć inne wymiary trzeba go przyciąć w programie graficznym i podać tutaj nowe wymiary. Kompilator nie przycina oryginalnych rozmiarów obraza.
ROWNAJ=	 Parametr określa, od której strony ma być narysowany obiekt. Opcje: do lewej – obiekt będzie narysowany od lewej strony czyli współrzędne x i y oznaczają lewy górny róg obiektu, centralnie – obiekt będzie narysowany tak, że współrzędne x i y oznaczają środek górnej krawędzi obiektu, do prawej – obiekt będzie narysowany od prawej strony czyli współrzedne x i y oznaczają prawy górny róg obiektu.
KOLOR=	Kolor wyświetlanego obrazka (ikony) w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem.
IKONA=	Nazwa pliku graficznego 24-bitowego z rozszerzeniem .bmp, który będzie przekonwertowany do postaci mono czyli jednokolorowej ikony.
STER=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. Numer sterownika jest potrzebny tylko, jeśli korzystamy z warunkowego wyświetlania podanego w następnej linijce – ENABLE. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i>
ENABLE=	 Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. Przykłady: ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165<3 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość mniejsza niż 3. ENABLE=4165 ENABLE=4165 jest wartość różna od 0. W warunkach równy '=' lub różny '!', zamiast rejestru może być użyta flaga. W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy

Pokazany poniżej obiekt <**ikona1>** opisuje wyświetlany monochromatyczny obrazek (TYP=IKONA) na podstawie pliku "icon_dom.bmp" (IKONA=icon_dom) o wymiarach 70 na 32 pikseli (W=70, H=32). Będzie on wyświetlany na współrzędnych x:10 i y:200 (x=10, Y=200), podane współrzędne dotyczą lewego górnego narożnika (ROWNAJ=do lewej). Ikona będzie wyświetlana w niebieskim kolorze (KOLOR=0,0,255). Ikona ta będzie wyświetlana pod warunkiem, że w zadeklarowanym w pliku sterowniku (STER=0) w rejestrze o adresie 4103 będzie wartość mniejsza od 4 (ENABLE=4103<4).

<ikonal> TYP=IKONA X=10 Y=200 W=70 H=32 ROWNAJ=do lewej KOLOR=255,0,0 IKONA=icon_dom STER=0 ENABLE=4103<4 </>>

Obiekt typu LISTA_TXT – lista tekstów

Obiekt typu LISTA_TXT umożliwia wyświetlenie różnych tekstów w zależności od wartości danego rejestru lub flagi. Np. dzień tygodnia.

zmienna	opis					
TYP=	Typ dan	ego obiektu.				
NAZWA=	Nazwa obiektu, będzie wyświetlana tylko podczas edycji tego parametru.					
	Maksymalna długość nazwy to 16 znaków.					
X=	Współrzędna x pozycji wyświetlanego obiektu. W zależności od					
	parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo					
	środek.					
Y=	Współrz	ędna y pozy	cji wyświetla	anego obiekti	u. 0 oznacza, że obiekt	
	będzie n	arysowany m	naksymalnie p	orzy górnej kr	awedzi ekranu.	
ROWNAJ=	Paramet	r określa, od	której strony	ma być narys	sowany obiekt. Opcje:	
	■ do le	ewej – obie	ekt będzie	narysowany	od lewej strony czyli	
	współrzę	edne x i y ozn	aczają lewy g	górny róg obie	ektu,	
	 centra 	alnie – obieł	kt będzie na	rysowany tal	k, że współrzędne x i y	
	oznacza	ją środek gór	nej krawędzi	obiektu,		
	■ do p	r awej – obi	ekt będzie	narysowany	od prawej strony czyli	
	współrzę	edne x i y ozn	aczają prawy	v górny róg ob	piektu.	
FONT=	Wielkoś	ć czcionki, Op	ocje: 12, 16, 2	20, 24, 36.		
		-	-	1		
		FONT	wysokość	szerokość	szerokosc małych elementów (::)	
		12	12	8	4	
		16	16	11	5	
		20	20	13	6	
		24	24	15	7	
		36	36	24	10	
	Davini			the infliction		
	POWyZSZ	a tabela moz	e byc przyda	atna, jesii chco	emy na ekranie dokładnie	
	rozmieso	CIC KIIKA LEK	SLOW IUD Wy i rá≐naci dluv	swieliaczy n	umerycznych o roznych	
	roziniara	ICH CZCIONEK		JOSCI ZNAKOW	7. WSZYSIKIE IIIERY I CYIFY	
	intorpup	lana Sallia	į szerokos	C Z Wyja		
	Kolor w	véwietlenei	,.iip. liety w form	ania PCR (24b (24 bitowwm) Waai	
KULUN=		gólnych kolor		iacle nGD i	240 (24-bilowyiii). Wayi	
	Podzaj li	icty Opoio:	ow muszą by			
ANCJA=	nouzaj listy. Opuje. • adit — lista odvtowalna, pojowia sio okionko uvihoru, mekovmelna,					
	- euri – iista euytowalna, pojawia się okienko wydoru, maksymalna –					
	uugusu listy euytowalnej to 7 elementow (tekstow po 19 znakow),					
	zmiana r	nsta cykik na koleiny tek	et z liety ma	ksymalna dłu	anéć listy cyklicznej to 15	
	element	ńw (tekstów r	o 19 znaków	n)		
	■ brak -	– lista tvlko d	lo wyświetlan), jia maksyma	Ina długość tej listy to 15	
	elementów (tekstów no 19 znaków)					
LISTA=	Nazwa pliku z lista tekstów do wyświetlania. Wartości i teksty im					
	nrzynisane zawarte sa w dodatkowym pliku					
LISTA EDIT=	Nazwa	pliku z lista	tekstów, któ	fre sie poiav	via podczas edvcii tego	
	paramet	ru czvli tvlka	o dla list z i	ustawionym r	parametrem AKC ₁ /A=edit	
	Wartośc	i i tekstv im	przypisane	sa w dodat	kowym pliku. Parametry	
	LISTA i	LISTA EDIT	umożliwiaia	przygotowar	nie dwóch list. np. iednei	
	wvrówna	anei do lewei	stronv. a dr	uaiei wvrówn	anei do prawei. Możliwe	
	jest tez	zastosowan	ie w iednei	liście skród	convch tekstów. Poniżej	
	przvkład	v takich plikó	W.		, ,	
STER=	Numer s	terownika. kt	órego dotvcz	v danv obiekt		
	W pliku	o podanvm n	umerze zapis	sane sa wszv	stkie parametrv transmisii	
	w tym ac	dres Modbus	sterownika. F	Przykład:		
	STER=0	oznacza, że ol	biekt dotyczy s	sterownika opis	anego w pliku ster0.txt.	

ADR=	Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.
ENABLE=	VVarunek wyswietlania danego obiektu. Jesli chcemy žeby dany obiekt
	był wyświetlany zawsze to możemy całkowicje pominać linijke z tym
	warunkiem ENABLE
	Valunkiem LINADEL.
	Jesii chcerny, aby dany obiekt by wyswietiany w zależności od jakiegos
	warunku to podajemy go w tej linijce po słowie ENABLE.
	Przykłady:
	ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość równa 1.
	ENABLE=4165>5 – obiekt będzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość większa niż 5.
	ENABLE=4165<3 - obiekt bedzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość mniejsza niż 3.
	ENABLE=4165!0 – obiekt bedzie wyświetlany, jeśli w
	rejestrze 4165 jest wartość różna od 0.
	W warunkach równy '=' lub różny 'l' zamiast rejestru może być użyta
	flaga W tym parametrze jako adres rejestru (flagi) zemiest werteści
	liaga. w tym parametrze jako aures rejestru (nagi) zamiast wartosci
	liczbowej można podać nazwę rejestru (flagi). W tym celu nalezy
	najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.

Pokazany poniżej obiekt <dzien> opisuje edytowalną (AKCJA=edit) listę dni tygodnia w postaci tekstowej (TYP=LISTA_TXT). Obiekt będzie wyświetlany na współrzędnych x:475 i y:60 (X=475, Y=60), podajemy współrzędne prawego górnego rogu obiektu (ROWNAJ=do prawej). Wyświetlany tekst będzie miał wielkość czcionki 36 (FONT=36) w kolorze niebieskim (KOLOR=0, 0, 255). Teksty będą wyświetlane w zależności od wartości rejestru 4006 (ADR=4006) w sterowniku zadeklarowanym w pliku ster0.txt (STER=0).

Lista wyświetlanych tekstów i wartości do nich przypisanych będzie zadeklarowana w pliku "I_dni_skrot.txt" (LISTA=1_dni_skrot), poniżej przykład takiego pliku. Podczas edycji tego parametru będzie wyświetlone dodatkowe okienko wyskakujące zawierające lista tekstów zadeklarowanych w pliku "I_dni_lewa.txt" (LISTA=1_dni_lewa), poniżej przykład takiego pliku.

```
<dzien>

TYP=LISTA_TXT

NAZWA=Dzien tygodnia

X=475

Y=60

ROWNAJ=do prawej

FONT=36

KOLOR=0,0,255

AKCJA=edit

LISTA=1_dni_skrot

LISTA_EDIT=1_dni_lewa

STER=0

ADR=4006

</>>
```

W poniższym pliku w każdej linijce mamy dwa parametry oddzielone średnikiem. Pierwszy to wartość rejestru, dla którego ma być wyświetlony dany tekst, drugi to tekst o maksymalnej długości 19 znaków. Możemy zadeklarować do 15 elementów.

Plik I_dni_skrot.txt

0;Po			
1;Wt			
2;Sr			
3;Cz			
4;Pi			
5;So			
6 ; Ni			

W poniższym pliku w każdej linijce mamy dwa parametry oddzielone średnikiem. Pierwszy to wartość rejestru, drugi to tekst o maksymalnej długości 19 znaków. Trzeba pamiętać, aby teksty miały tą samą długość, w razie potrzeby trzeba uzupełnić brakujące litery "SPACJAMI".

Ten plik opisuje listę tekstów w okienku wyskakującym podczas edycji, możemy w nim zadeklarować maksymalnie do 7 elementów.

Plik I_dni_lewa.txt

0;Poniedzialek 1;Wtorek 2;Sroda 3;Czwartek 4;Piatek 5;Sobota 6;Niedziela

Obiekt typu LISTA_ALERT – lista alarmów

Obiekt typu LISTA_ALERT umożliwia wyświetlenie naprzemiennie wszystkich aktualnie występujących awarii.

zmienna	opis								
TYP=	Typ danego obiektu.								
X=	Współrz	Współrzędna x pozycji wyświetlanego obiektu. W zależności od							
	paramet	parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo							
	środek.	środek.							
Y=	Współrz	Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt							
	będzie n	arysowany m	iaksymalnie p	orzy górnej kr	awędzi ekranu.				
ROWNAJ=	Paramet	r określa, od	której strony	ma być narys	owany obiekt. Op	ocje:			
	■ do le	ewej – obie	ekt będzie	narysowany	od lewej stro	ny czyli			
	współrzę	edne x i y ozn	aczają lewy ę	górny róg obie	ektu,				
	 centra 	alnie – obieł	kt będzie na	rysowany tal	<, że współrzędr	ıехіу			
	oznacza	ją środek gór	nej krawędzi	obiektu,					
	■ do p	r awej – obi	ekt będzie	narysowany	od prawej stro	ny czyli			
	współrzę	dne x i y ozn	aczają prawy	v górny róg ob	oiektu.				
FONT=	Wielkość	ć czcionki, Op	ocje: 12, 16, 2	20, 24, 36.					
		FONT			szerokość małych				
		FONT	wysokość	szerokość	elementów (. , : ;)				
		12	12	8	4				
		16	16	11	5				
		20	20	13	6				
		24	24	15	/				
		36 36 24 10							
	Powyższa tabela może być przydatna, jeśli chcemy na ekranie dokładnie								
	rozmieścić kilka tekstów lub wyświetlaczy numerycznych o różnych								
	rozmiarach czcionek i różnej długości znaków. Wszystkie litery i cyfry								
	maja taka sama szerokość z wyjatkiem kilku znaków								
	interpun	kcyjnych:.,:	;.itp.		·				
KOLOR=	Kolor wyświetlanej listy z alarmami w formacie RGB 24b (24-bitowym).								
	Wagi po	szczególnych	kolorów mus	szą być rozdz	ielone przecinkie	m.			
BLINK=	Paramet	r określa czy	dany obiekt l	oędzie migał.					
	■ 0 - ob	 0 – obiekt będzie wyświetlany ciągle, 							
	1 – ob	iekt będzie m	nigał z często	tliwością 1s.					
LISTA=	Nazwa p	oliku z listą a	larmów do w	vyświetlania.	W tym dodatkow	ym pliku			
	zapisane	e są następuj	ące informac	cje: numer ste	erownika, numer	rejestru,			
	wartość	tego rejestru	i treść komu	unikatu. Tej s	amej listy można	i użyć w			
	obiekcie	BOX ALERI	Г.			-			

Pokazanv poniżej obiekt <alert> opisuje listę wvświetlanvch awarii (TYP=LISTA_ALERT). Obiekt będzie wyświetlany na współrzędnych x:240 i y:280 Y=280), podajemy współrzędne środka górnej krawędzi obiektu (X=240,(ROWNAJ=centralnie). Wyświetlany tekst będzie miał wielkość czcionki 24 (FONT=24) w kolorze czerwonym (KOLOR=250,0,0) i będzie migać (BLINK=1). Lista wyświetlanych alarmów będzie zadeklarowana w pliku "I alert.txt" (LISTA=1 alert), poniżej przykład takiego pliku.

```
<alert>
TYP=LISTA_ALERT
X=240
Y=280
ROWNAJ=centralnie
FONT=24
KOLOR=250,0,0
BLINK=1
LISTA=1_alert
</>
```

W poniższym pliku w każdej linijce mamy cztery parametry oddzielone średnikiem:

- numer sterownika, (np. 0 oznacza, że zmienna dotyczy sterownika opisanego w pliku ster0.txt),

- numer rejestru,

- wartość tego rejestru, dla którego występuje dana awaria,

- wyświetlany tekst dla danej awarii,

Możemy zadeklarować do 64 elementów (różnych awarii) o maksymalnej długości tekstu 32 znaki.

W poniższym przykładzie z listą alarmów widzimy, że jeśli w sterowniku o numerze 0 opisanym w pliku ster0.txt w rejestrze 4101 będzie wartość 1 to zastanie wyświetlony komunikat "Awaria czujnika Tkan!".

Plik I_alert.txt

```
0;4101;1;Awaria czujnika Tkan!
0;4102;1;Awaria czujnika Tzew!
0;4111;1;Brudny filtr czerpni!
0;4112;1;Brudny filtr nawiewu!
1;4201;1;Awaria czujnika Twel!
1;4202;1;Awaria czujnika Twe2!
```

W liście alrtów jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.

Obiekt typu LISTA_BMP – lista obrazków typu RGB 24-bitowy

Obiekt typu LISTA_BMP umożliwia wyświetlenie różnych obrazków w zależności od wartości danego rejestru lub flagi. Obrazki te muszą być 24-bitowymi bitmapami o rozszerzeniu .bmp.

Zimerina	opis
TYP=	Typ danego obiektu.
X=	Współrzędna x pozycji wyświetlanego obiektu. W zależności od parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo środek
V_	Wepółrzedna v pozycji wyświotlapogo obioktu. O oznacza, żo obiokt
1=	bedzie narysowany maksymalnie przy górnej krawedzi ekranu
W–	Szerokość objektu
H_	Wysokość obiektu
	Parametr ekrośla, od któroj stropy ma być parysowany objekt. Opcje:
NOWNAJ-	 do lewej – obiekt będzie narysowany od lewej strony czyli współrzodno x i v oznaczaja lowy górny róg obiektu
	spolizeune x i y oznaczają lewy gorny rog obiektu,
	- centralme – obleki będzie narysowany tak, że wspoliżędne x i y
	de prawoj objekt bodzio parveowany od prawoj strony ozvij
	• do prawej – obiekt będzie narysowany od prawej strony czyn
	Peremetr ekreśle ezy depy ebiekt bedzie migeł
	• 0 objekt bodzie wyćwietleny ciegle
	 U – objekt będzie wyswietiany ciągie, 1 – objekt będzie migał z częstotliwościa 1s
	Rodzaj listy Opcje:
	riouzaj listy. Opoje. ■ cadit – lista cyklicznie przewijana, po dotkniecju objektu nastepuje.
	zmiana na nastenna w kolejności grafike z listy
	Iniana na następną w kolejności grankę z listy, ■ hrak – lista tylko do wyświetlania
LISTA-	Nazwa nliku z lista wartości i przypisanymi do nich nlikami graficznymi
	Pliki te musza być 24-bitowymi bitmanami o rozszerzeniu, bmn. Poniżej
	nrzykład takiego pliku
STER-	Numer sterownika, którego dotyczy dany obiekt
OTEN=	Willie Sterownika, Rolego dotyczy dany obiekt.
	I W DIIKU O DODADVM DUMERZE ZADISADE SA WSZVSTKIE DARAMETRY TRADSMISII I
	W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. Przykład:
	W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.
ADR=	W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego
ADR=	W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus
ADR=	W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie
ADR=	W pliku o podaným numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1)
ADR=	W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze
ADR=	W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać
ADR=	W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik
ADR=	W pliku o podaným numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.
ADR=	W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt
ADR= ENABLE=	W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym
ADR= ENABLE=	W pliku o podaným numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE.
ADR= ENABLE=	W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś
ADR= ENABLE=	W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE.
ADR= ENABLE=	 W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. <i>Przykłady:</i>
ADR= ENABLE=	 W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. <i>Przykłady:</i> <i>ENABLE=4165=1 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1</i>
ADR= ENABLE=	 W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. <i>Przykłady:</i> <i>ENABLE=4165=1 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1.</i> <i>ENABLE=4165>5 – obiekt będzie wyświetlany, jeśli w</i>
ADR= ENABLE=	 W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. <i>Przykłady:</i> <i>ENABLE=4165=1 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1.</i> <i>ENABLE=4165>5 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1.</i>
ADR= ENABLE=	 W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. <i>Przykłady:</i> <i>ENABLE=4165=1 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1.</i> <i>ENABLE=4165>5 – obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5.</i> <i>ENABLE=4165</i> <i>ENABLE=4165</i> <i>obiekt będzie wyświetlany, jeśli w</i>
ADR= ENABLE=	 W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. <i>Przykłady:</i> ENABLE=4165=1 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość równa 1. ENABLE=4165>5 - obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5. ENABLE=4165 <i>ENABLE=4165</i> <i>obiekt będzie wyświetlany, jeśli w rejestrze 4165 jest wartość większa niż 5.</i>
ADR= ENABLE=	W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostępnego do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1) lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE. <i>Przykłady:</i> <i>ENABLE=4165=1 – obiekt będzie wyświetlany, jeśli w</i> rejestrze 4165 jest wartość równa 1. <i>ENABLE=4165>5 – obiekt będzie wyświetlany, jeśli w</i> rejestrze 4165 jest wartość większa niż 5. <i>ENABLE=4165<3 – obiekt będzie wyświetlany, jeśli w</i> rejestrze 4165 jest wartość miejsza niż 3. <i>ENABLE=4165/0 – obiekt będzie wyświetlany, jeśli w</i> rejestrze 4165 jest wartość miejsza niż 3. <i>ENABLE=4165/0 – obiekt będzie wyświetlany, jeśli w</i> rejestrze 4165 jest wartość miejsza niż 3. <i>ENABLE=4165/0 – obiekt będzie wyświetlany, jeśli w</i> rejestrze 4165 jest wartość miejsza niż 3. <i>ENABLE=4165/0 – obiekt będzie wyświetlany, jeśli w</i> rejestrze 4165 jest wartość miejsza niż 3. <i>ENABLE=4165/0 – obiekt będzie wyświetlany, jeśli w</i>

W warunkach równy '=' lub różny '!', zamiast rejestru może być użyta
flaga. W tym parametrze jako adres rejestru (flagi) zamiast wartości
liczbowej można podać nazwę rejestru (flagi). W tym celu należy
najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.

Pokazany poniżej obiekt <bmp_dni> opisuje cyklicznie edytowalną (AKCJA=cedit) listę dni tygodnia w postaci graficznej czyli obrazków z 24-bitową grafiką (TYP=LISTA_BMP) o wymiarach 135 na 90 pikseli (W=135, H=90). Obiekt będzie wyświetlany na współrzędnych x:5 i y:230 (X=5, Y=230), podajemy współrzędne lewego górnego rogu obiektu (ROWNAJ=do lewej).

Grafiki będą wyświetlane w zależności od wartości rejestru 4006 (ADR=4006) w sterowniku zadeklarowanym w pliku ster0.txt (STER=0) i nie będą migać (BLINK=0).

Lista wyświetlanych grafik i wartości do nich przypisanych będzie zadeklarowana w pliku "l_dni_bmp.txt" (LISTA=l_dni_bmp), poniżej przykład takiego pliku.

<bmp_dni> TYP=LISTA_BMP X=5 Y=230 W=135 H=90 ROWNAJ=do lewej BLINK=0 AKCJA=cedit LISTA=1_dni_bmp STER=0 ADR=4006 </>>

W poniższym pliku w każdej linijce mamy dwa parametry oddzielone średnikiem. Pierwszy to wartość rejestru, dla którego ma być wyświetlona dana grafika, drugi to nazwa pliku graficznego 24-bitowego o rozszerzeniu .bmp.

Plik Idni_bmp.txt

0 ; dt_pn	
1;dt_wt	
2;dt_sr	
3;dt_cz	
4;dt_pt	
5 ; dt_so	
6;dt_ni	

Obiekt typu DYNLISTA_BMP – dynamiczna lista obrazków typu RGB 24-bitowy

Obiekt typu DYNLISTA_BMP umożliwia wyświetlenie różnych obrazków w zależności od zadeklarowanej listy z możliwymi zdarzeniami. W danym momencie wyświetlany będzie jeden obrazek o najwyższym priorytecie. Priorytet ustalany jest w zależności od kolejności wpisanych zdarzeń. Najwyższy priorytet ma zdarzenie na samej górze listy. Lista ta jest zapisana w osobnym pliku.

zmienna	opis					
TYP=	Typ danego obiektu.					
X=	Współrzędna x pozycji wyświetlanego obiektu. W zależności od					
	parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo					
	środek.					
Y=	Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt					
	będzie narysowany maksymalnie przy górnej krawędzi ekranu.					
W=	Szerokość obiektu.					
H=	Wysokość obiektu.					
ROWNAJ=	Parametr określa, od której strony ma być narysowany obiekt. Opcje:					
	• do lewej – obiekt będzie narysowany od lewej strony czyli					
	współrzędne x i y oznaczają lewy górny róg obiektu,					
	• centralnie – obiekt będzie narysowany tak, że współrzędne x i y					
	oznaczają środek górnej krawędzi obiektu,					
	• do prawej - obiekt będzie narysowany od prawej strony czyli					
	współrzędne x i y oznaczają prawy górny róg obiektu.					
LISTA=	Nazwa pliku z listą możliwych zdarzeń i przypisanymi do nich plikami					
	graficznymi. Pliki te muszą być 24-bitowymi bitmapami o					
	rozszerzeniu .bmp. Poniżej przykład takiego pliku.					
STER=	Numer sterownika, którego dotyczy dany obiekt.					
	W pliku o podanym numerze zapisane są wszystkie parametry transmisji					
	w tym adres Modbus sterownika. Przykład:					
	STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.					
ENABLE=	Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt					
	był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym					
	warunkiem ENABLE.					
	Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś					
	warunku to podajemy go w tej linijce po słowie ENABLE.					
	Przykłady:					
	ENABLE=4165=1 – obiekt będzie wyświetlany, jeśli w					
	rejestrze 4165 jest wartosc rowna 1.					
	ENABLE-4105/5 – Oblekt Dędzie wyswietlany, jesii w					
	ENABLE=4165<3 – objekt bedzie wyświetlany, jeśli w					
	rejestrze 4165 jest wartość mniejsza niż 3.					
	ENABLE=4165!0 – obiekt będzie wyświetlany, jeśli w					
	rejestrze 4165 jest wartość różna od 0.					
	W warunkach równy '=' lub różny '!', zamiast rejestru może być użyta					
	flaga. W tym parametrze jako adres rejestru (flagi) zamiast wartości					
	liczbowej można podać nazwę rejestru (flagi). W tym celu należy					
	najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.					

Pokazany poniżej obiekt <dynlist> opisuje dynamiczną listę bitmap (TYP=LISTA_BMP) o wymiarach 36 na 96 pikseli (W=36, H=96). Obiekt będzie wyświetlany na współrzędnych x:240 i y:100 (X=240, Y=100), podajemy współrzędne środka górnej krawędzi obiektu (ROWNAJ=centralnie). Grafiki będą wyświetlane w zależności od zadeklarowanej listy zapisanej w pliku "l_dynlista.txt" (LISTA=l_dynlista). Obrazek ten będzie wyświetlany pod warunkiem, że w zadeklarowanym w pliku sterowniku "ster0.txt" (STER=0) w rejestrze o adresie 4103 będzie wartość większa od 0 (ENABLE=4103>0).

```
<dynlist>
TYP=DYNLISTA_BMP
X=240
Y=100
W=36
H=96
ROWNAJ=centralnie
LISTA=l_dynlista
STER=0
ENABLE=4103>0
</>>
```

W poniższym pliku "I_dynlista.txt", w każdej linijce mamy cztery parametry oddzielone średnikiem i są to:

- numer sterownika,
- numer rejestru lub flagi i warunek logiczny, dla którego wyświetlana będzie dana grafika, Możliwe logiczne warunki są takie same jak dla parametru ENABLE:
 - równe 7187=1,
 - różne 7188!1,
 - większe 4212>2,
 - mniejsze 4212<1,
- informacje o miganiu danej grafiki, jeśli spełniony jest warunek zdarzenia,

- nazwa pliku graficznego 24-bitowego o rozszerzeniu .bmp. który ma być wyświetlony jeśli spełniony jest warunek zdarzenia.

Plik I_dynlista.txt

0;7187=1;1;ng_err 0;7188!1;0;ng_war 0;4212>2;0;ng1 0;4212<1;0;ng0

W danym momencie będzie wyświetlany tylko jeden obrazek o najwyższym priorytecie, czyli w kolejności od góry do dołu. Najpierw sprawdzany jest pierwszy warunek: czy flaga 7187 jest równa 1 (7187=1). Jeśli tak to będzie migał obrazek zapisany w pliku "ng_err.bmp". Jeśli nie to sprawdzany jest drugi warunek czy flaga 7188 jest rózna od 1 (7188!1). Jeśli tak to będzie wyświetlany obrazek zapisany w pliku "ng_war.bmp". Jeśli nie to sprawdzany jest kolejny warunek czy rejestr 4212 jest większy niż 2 (4212>2). Jeśli tak to będzie wyświetlany obrazek zapisany w pliku "ng1.bmp". Jeśli nie to sprawdzany jest kolejny warunek czy rejestr 1 (4212<1). Jeśli nie to sprawdzany w pliku "ng0.bmp". Jeśli nie to kolejnych warunków do sprawdzania z listy w tym przykładzie już nie ma i nie będzie wyświetlana żadna grafika.

Obiekt typu LISTA_IKON – lista obrazków typu mono

Obiekt typu LISTA_IKON umożliwia wyświetlenie różnych obrazków w zależności od wartości danego rejestru lub flagi. Obrazki te muszą być 24-bitowymi bitmapami o rozszerzeniu .bmp, które podczas kompilacji będą przekonwertowane do obrazków mono czyli jednokolorowych. Każdy bit w kolorze innym niż biały traktowany jest jako "1" i rysowany jest w kolorze zdefiniowanym w obiekcie. Dla każdej wartości rejestru możemy zdefiniować inny kolor wyświetlania ikony.

zmienna	opis
TYP=	Typ danego obiektu.
X=	Współrzędna x pozycji wyświetlanego obiektu. W zależności od
	parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo
	środek.
Y=	Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt
	będzie narysowany maksymalnie przy górnej krawędzi ekranu.
W=	Szerokość obiektu.
H=	Wysokość obiektu.
BLINK=	Parametr określa czy dany obiekt będzie migał.
	 0 – obiekt będzie wyświetlany ciągle,
	 1 – obiekt będzie migał z częstotliwością 1s.
ROWNAJ=	Parametr określa, od której strony ma być narysowany obiekt. Opcje:
	 do lewej – obiekt będzie narysowany od lewej strony czyli
	współrzędne x i y oznaczają lewy górny róg obiektu,
	• centralnie – obiekt będzie narysowany tak, że współrzędne x i y
	oznaczają środek górnej krawędzi obiektu,
	 do prawej – obiekt będzie narysowany od prawej strony czyli
	współrzędne x i y oznaczają prawy górny róg obiektu.
AKCJA=	Rodzaj listy. Opcje:
	• cedit – lista cyklicznie przewijana, po dotknięciu obiektu następuje
	zmiana na następną w kolejności grafikę z listy,
	brak – lista tylko do wyswietlania.
LISTA=	Nazwa pliku z listą wartości, kolorow i przypisanymi do nich plikami
	graficznymi. Pliki te muszą byc 24-bitowymi bitmapami o
OTED	rozszerzeniu .bmp. Ponizej przykład takiego pliku.
SIER=	Numer sterownika, ktorego dotyczy dany obiekt.
	w pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym odrog Modbug storownika
	Przykład:
	STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0 txt
ADR=	Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostepnego
	do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus
	TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie
	X oznacza numer sterownika (SL DADR do SL DADR+SL DSIZE-1)
	lub z zakresu (SL_FADR do SL_FADR+SL_FSIZE-1). W tym parametrze
	jako adres rejestru (flagi) zamiast wartości liczbowej można podać
	nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik
	(rejestry.txt) z taką listą nazw rejestrów.
ENABLE=	Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt
	był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym
	warunkiem ENABLE.
	Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś
	warunku to podajemy go w tej linijce po słowie ENABLE.
	FIZYKIAUY: FNARLF=4165=1 – objekt bedzie wyświetleny jośli w
	rejestrze 4165 jest wartość równa 1.
	ENABLE=4165>5 - obiekt będzie wyświetlany, ieśli w
	rejestrze 4165 jest wartość wieksza niż 5

ENABLE=4165<3	– obiekt	będzie	wyświetlany,	jeśli	W
rejestrze 4165	jest warto	ść mniejsz	a niż 3.		
ENABLE=4165!0	– obiekt	będzie	wyświetlany,	jeśli	W
rejestrze 4165	jest warto	ść różna c	d 0.		
W warunkach ró	wny '=' lub ro	óżny '!', zam	niast rejestru moz	że być uż	yta
flaga. W tym pa	rametrze jak	o adres rej	estru (flagi) zam	iast warto	JŚCI
liczbowej można	podać naz	wę rejestru	(flagi). W tym	celu nale	eży
najpierw przygoto	wać plik (reje	stry.txt) z tal	ka lista nazw reje	strów.	•

Pokazany poniżej obiekt <ikonal> opisuje wyświetlaną listę ikon (TYP=LISTA_IKON), które mają sygnalizować różne stany awaryjne w sterowniku opisanym w pliku (STER=0) i nie będą migać (BLINK=0). Adres rejestru z informacją o awariach to 4011 (ADR=4011). Obiekt będzie wyświetlał ikony zadeklarowane w pliku "l_ierror.txt" (LISTA=1_ierror) bez możliwości edycji (AKCJA=brak). Ikony będą miały rozmiar 180 na 120 pikseli (W=180, H=120), które będą wyświetlane na współrzędnych x:100 i y:200 (X=100, Y=200), podane współrzędne dotyczą lewego górnego narożnika (ROWNAJ=do lewej).

<ikonal> TYP=LISTA_IKON X=100 Y=200 W=180 H=120 ROWNAJ=do lewej AKCJA=brak LISTA=l_ierror STER=0 ADR=4011 </>>

Pokazany poniżej plik **I_ierror.txt** opisuje dokładnie listę ikon wyświetlaną w powyższym obiekcie. W każdej linijce są trzy parametry oddzielone średnikiem:

- wartość rejestru, dla której dana ikona jest wyświetlana (2),
- kolor tej ikony, dla każdej wartości może być inny kolor (255, 0, 0),
- nazwa pliku z daną ikoną (icon_t_error).

Kolor wyświetlanej ikony jest zapisany w formacie RGB 24b (24-bitowym). Liczby muszą być rozdzielone przecinkiem (np. dla koloru czerwonego to będzie następujący zapis 255,0,0). Każdy plik ikony musi być plikiem graficznym 24-bitowym o rozszerzeniu .bmp, który w czasie kompilacji przekonwertowany będzie do postaci mono czyli jednokolorowej ikony. Każdy bit w kolorze innym niż biały traktowany jest jako "1" i rysowany jest w kolorze zdefiniowanym osobno dla każdej ikony.

Plik I_ierror.txt

```
2;255,0,0;icon_t_error
3;255,0,0;icon_t_error
4;255,0,0;icon_t_error
5;255,0,0;icon_t_error
9;255,0,0;icon_t_error
203;255,165,0;icon_t_max
209;255,165,0;icon_t_max
```

Pliki "icon_t_error.bmp" i "icon_t_max.bmp" są plikami graficznymi zapisanymi w formacie 24-bitowej bitmapy.

Obiekt typu ONOFF_IKONA – migająca ikona

Obiekt typu ONOFF_IKONA jest specyficznym obiektem. Posiada dwa stany: wyłączenia i załączania. W stanie załączenia obiekt cyklicznie zmienia swój kolor, dzięki czemu można zrobić efekt migania. Stan załączenia można uwarunkować przez dodatkowy rejestr. Plik graficzny musi być 24-bitowa bitmapą o rozszerzeniu .bmp, która podczas kompilacji będzie przekonwertowana do obrazka mono czyli jednokolorowej ikony.

zmienna	opis			
TYP=	Typ danego obiektu.			
X=	Współrzędna x pozycji wyświetlanego obiektu. W zależności od			
	parametru ROWNAJ moze to byc lewa lub prawa krawędz obiektu albo			
	srodek.			
Y=	Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt			
	Deuzie narysowany maksymainie przy gornej krawędzi ekranu.			
W=	Szerokość obiektu.			
H=	Wysokość obiektu.			
ROWNAJ=	Parametr określa, od której strony ma być narysowany obiekt. Opcje:			
	• do lewej – obiekt będzie narysowany od lewej strony czyli			
	wspołrzędne x i y oznaczają lewy górny róg obiektu,			
	• centralnie – objekt będzie narysowany tak, że współrzędne x i y			
	oznaczają srodek gornej krawędzi obiektu,			
	• do prawej – obiekt będzie narysowany od prawej strony czyli			
	wspołrzędne x i y oznaczają prawy gorny rog obiektu.			
KOLOR_OFF=	Kolor Ikony w stanie wyłączenia w formacie RGB 24b			
	(24-bitowym). Wagi poszczegolnych kolorow muszą być rozdzielone			
	przecifikiem. Diarwany (nacrotkowy) kolor ikony w stania zalaszania w formacia BCB.			
KOLOR_ONIE	Pierwszy (początkowy) kolor ikony w stanie załączenia w formacie RGB			
	24b (24-bilowyiii). Wagi poszczegolitych kolorow muszą być rozdzielone przecinkiem Jkopa w stanie załaczenia bodzie owklicznie zmieniać kolor			
	przecifikiem. Ikona w stanie załączenia będzie cyklicznie zmieniać kolor pomiodzy KOLOB, ONI a KOLOB, ON2			
	Drugi (końcowy) kolor ikony w stanie załaczonia w formacja PGB 24h			
KOLON_ONZ=	(24-bitowy) Wagi poszczogólnych kolorów mucza być rozdzielene			
	(24-bitowym). Wagi poszczegomych kolorów muszą być rozuzielone			
	Nazwa pliku graficznego 24-bitowego z rozszerzeniem hmp. który			
	bedzie przekonwertowany do postaci mono czyli jednokolorowej ikony			
ON=	Warunek załaczenia Jeśli wartość odczytana z rejestru			
•	zadeklarowanego w parametrze ADR jest równa temu parametrowi to			
	ikona jest w stanie załączenia.			
	Przykłady:			
	Jeśli ON=0 i ADR=7093 to			
	R7093=0 to ikona załączona,			
	R7093≠1 to ikona wyłączona.			
	Jesli ON=100 i ADK=4094 to			
	R4094=100 to ikona wybozona			
STER_	Numor storownika, którogo dotyczy dany obiekt			
SILN=	W nliku o podanym numerze zanisane sa wszystkie narametry transmisii			
	w tym adres Modbus sterownika Przykład:			
	STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.			
ADR=	Adres 16-bitowego rejestru (holding register) lub flagi (coil) dostepnego			
	do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus			
	TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie			
	X oznacza numer sterownika (SL DADR do SL DADR+SL DSIZE-1)			
	lub z zakresu (SL_FADR do SL FADR+SL FSIZE-1). W tym			
	parametrze jako adres rejestru (flagi) zamiast wartości liczbowej można			
	podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować			
	plik (rejestry.txt) z taką listą nazw rejestrów.			

IF_ADR=	Dodatkowy warunek załączenia. Jeśli wartość odczytana z rejestru lub flagi IF_ADR jest większa od 0 to stan ikony zależy od rejestru (lub flagi) określonego w parametrze ADR. Jeśli wartość jest równa 0 to stan ikony jest wyłączony. Jeśli IF_ADR= -1 to warunek jest nieaktywny. <i>Przykład:</i> <i>Jeśli ON=0, ADR=4094 i IF_ADR=4095 to</i> <i>R4094=0 i R4095>0 to ikona jest załączona,</i> <i>R4094=1 i R4095>0 to ikona wyłączona,</i> <i>R4094=0 i R4095=0 to ikona wyłączona.</i>			
ENABLE=	Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt			
	był wyświetlany zawsze to możemy całkowicje pominać lipijke z tym			
	by wyswieliany zawsze to mozemy carowicie pominąc imijkę z tym			
	warunkiem ENABLE.			
	Jeśli chcemy, aby dany objekt był wyświetlany w zależności od jakiegoś			
	warunku to podajemy do w tej linijce po słowie ENABLE			
	warunku to podajemy go w tej imijce po słowie ENABLE.			
	Przykłady:			
	ENABLE=4165=1 – obiekt będzie wyświetlany, jeśli w			
	rejestrze 4165 jest wartość równa 1.			
	ENABLE=4165>5 – obiekt bedzie wyświetlany, jeśli w			
	rejestrze 4165 jest wartość wieksza niż 5.			
	ENABLE=4165<3 – objekt bedzie wyświetlany, jeśli w			
	rejestrze 4165 jest wartość mniejsza niż 3			
	ENADIE = 416E10 objekt bodzie wardzietler			
	ENABLE=4105:0 - ODIEKU DĘCZIE WYSWIELIANY, JESII W			
	rejestrze 4165 jest wartosc różna od U.			
	W warunkach równy '=' lub różny '!', zamiast rejestru może być użyta			
	flaga. W tym parametrze jako adres rejestru (flagi) zamiast wartości			
	liczbowej można podać nazwe rejestru (flagi). W tym celu należy			
	nozoowoj mozna podao nazwę rojostru (nagr). W tym obu nalezy			
	najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrow.			

Pokazany poniżej obiekt <ikona_cwu> opisuje wyświetlaną dwustanową ikonę (TYP=ONOFF_IKONA). Ikona pobrana z pliku graficznego 24-bitowego o nazwie "icon_cwu.bmp" (IKONA=icon_cwu) zmienia swój stan w zależności od rejestru 4074 (ADR=4074) w sterowniku opisanym w pliku "ster.txt" (STER=0). W stanie wyłączenia ma kolor (KOLOR_OFF=25,25,112), natomiast w stanie załączenia, jeśli rejestr równa się 100 (ON=100), płynnie zmienia swój kolor pomiędzy dwoma zdefiniowanymi kolorami (KOLOR_ON1=32,128,32, KOLOR_ON2=0,255,0). Ikona w rozmiarze 36 na 30 pikseli (W=36, H=30), będzie wyświetlana na współrzędnych x:250 i y:200 (X=250, Y=200), podane współrzędne dotyczą lewego górnego narożnika (ROWNAJ=do lewej). Ikona ta będzie wyświetlana zawsze, nie posiada dodatkowego warunku wyświetlania (IF_ADR=-1).

<ikona cwu> TYP=ONOFF_IKONA X=250 Y=200 W=36 H=30 ROWNAJ=do lewej KOLOR_OFF=25,25,112 KOLOR_ON1=32,128,32 KOLOR_ON2=0,255,0 IKONA=icon_cwu ON=100 STER=0 ADR=4074 IF ADR=-1 </>

Obiekt typu PASS – hasło do trybu edycji

Obiekt typu PASS umożliwia przejście do trybu edycji wszystkich parametrów. Obiekt ten to obrazek 24-bitowy o rozszerzeniu .bmp, który po dotknięciu otwiera okienko z klawiaturą umożliwiającą wpisanie hasła. Po poprawnym wpisaniu hasła zadeklarowanego w pliku projekt.txt w linijce PASS panel HMI przechodzi w tryb edycji co umożliwia zmianę wszystkich parametrów, które maja zadeklarowaną taką możliwość w swoich parametrach (AKCAJ=edit lub AKCJA=cedit). Dotyczy to następujących obiektów:

- VAR16 wyświetlacz liczby 16-bitowej,
- AKCJA_DEC akcja dekrementacji,
- AKCJA_INC akcja inkrementacji,
- AKCJA_SET akcja ustawienia zmiennej,
- AKCJA_FLAG_SET akcja ustawienia flagi,
- KLAWISZ_DEC przycisk dekrementacji,
- KLAWISZ_INC przycisk inkrementacji,
- KLAWISZ_SET przycisk ustawienia zmiennej,
- KLAWISZ_FLAG_SET przycisk ustawienia flagi,
- LISTA_TXT lista tekstów,
- LISTA_BMP lista obrazków typu RGB 24-bitowy,
- LISTA_IKON lista obrazków typu mono.

Do czasu podania poprawnego hasła żaden obiekt nie podlega edycji i jest możliwe tylko ich przeglądanie i przechodzenie pomiędzy wszystkimi ekranami.

Funkcja zabezpieczenia hasłem działa tylko wtedy, gdy podamy hasło różne od zera w pliku projekt.txt w linijce PASS.

zmienna	opis
TYP=	Typ danego obiektu.
X=	Współrzędna x pozycji wyświetlanego obiektu. W zależności od parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo środek.
Y=	Współrzędna y lewego górnego roku wyświetlanego obiektu. 0 oznacza, że obiekt będzie narysowany maksymalnie przy górnej krawędzi ekranu.
W=	Szerokość obiektu.
H=	Wysokość obiektu.
ROWNAJ=	 Parametr określa, od której strony ma być narysowany obiekt (nie dotyczy to napisu określonego w parametrze NAZWA). Opcje: do lewej – obiekt będzie narysowany od lewej strony czyli współrzędne x i y oznaczają lewy górny róg obiektu, centralnie – obiekt będzie narysowany tak, że współrzędne x i y oznaczają środek górnej krawędzi obiektu, do prawej – obiekt będzie narysowany od prawej strony czyli współrzędne x i y oznaczają prawy górny róg obiektu.
BMP_ON=	Nazwa pliku graficznego 24-bitowego z rozszerzeniem .bmp. Obrazek ten jest wyświetlany w trybie edycji, czyli po poprawnym podaniu hasła. Plik ten musi być umieszczony w tym samym katalogu co cały projekt. Może to być np. obrazek z otwartą kłódka.
BMP_OFF=	Nazwa pliku graficznego 24-bitowego z rozszerzeniem .bmp. Obrazek ten jest wyświetlany w trybie blokady edycji, czyli bez poprawnie wpisanego hasła. Plik ten musi być umieszczony w tym samym katalogu co cały projekt. Może to być np. obrazek z zamkniętą kłódką.

Pokazany poniżej obiekt <pass> umożliwia wejście do trybu edycj. Obrazek o wymiarach 64 na 64 pikseli (W=64, H=564) będzie wyświetlany na współrzędnych x:O i y:97 (X=0, Y=97), podane współrzędne dotyczą lewego górnego narożnika (ROWNAJ=do lewej). Grafika wyświetlana w trybie edycji jest w pliku icon_pass_on.bmp (BMP_ON=icon_pass_on) natomiast grafika wyświetlana w trybie blokady edycji jest w pliku icon_pass_off.bmp (BMP_ON=icon_pass_off).

<pass> TYP=PASS X=0 Y=97 W=64 H=64 ROWNAJ=do lewej BMP_ON=icon_pass_on BMP_OFF=icon_pass_off </>

Obiekt typu PROGRAM – interpretacja graficzna programu dobowego.

Obiekt typu PROGRAM jest specyficznym obiektem, który umożliwia graficzne przedstawienie dwóch przedziałów czasowych programu dobowego w formie wykresu słupkowego. Jest to obiekt współpracującym z programami w sterownikach Frisko.

zmienna	opis			
TYP=	Typ danego obiektu.			
X=	Współrzędna x pozycji wyświetlanego obiektu. W zależności od			
	parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo			
	środek.			
Y=	Współrzędna y pozycji wyświetlanego obiektu. 0 oznacza, że obiekt			
	będzie narysowany maksymalnie przy górnej krawędzi ekranu.			
ROWNAJ=	Parametr określa, od której strony ma być narysowany obiekt. Opcje:			
	• do lewej – obiekt będzie narysowany od lewej strony czyli			
	współrzędne x i y oznaczają lewy górny róg obiektu,			
	• centralnie – obiekt będzie narysowany tak, że współrzędne x i y			
	oznaczają środek górnej krawędzi obiektu,			
	• do prawej – obiekt będzie narysowany od prawej strony czyli			
	współrzędne x i y oznaczają prawy górny róg obiektu.			
KOLOR_HI=	Kolor, który reprezentuje aktywny program dla dwóch zadeklarowanych			
	odcinków czasu:			
	od G11:M11 do G12:M12,			
	od G21:M21 do G22:M22.			
	Kolor zapisany jest w formacie RGB 24b (24-bitowym). Wagi			
	poszczegolnych kolorów muszą być rozdzielone przecinkiem.			
KULUH_LU=	Kolor, ktory reprezentuje nieaktywny program, czyli pokazuje odcinki ktory zadalelarowanymi. Kalar			
	poza uwoma zauekiarowanymi pomzej przeuziałami czasowymi. Kolor			
	kolorów mucza być rozdziolono przeciekiem			
	Kolor ozoionki w tym obiokojo w formacjo PCP 24b (24 bitowym). Wagi			
KOLON_IXIE	noszczogólnych kolorów musza być rozdziolono przecinkiem			
	poszczegolitych kolorów maszą być rozdzielone przecinkiem.			
STER_	Numer sterownika, którego dotyczy dany objekt			
STER=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane sa wszystkie parametry transmisij			
STER=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i>			
STER=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i>			
STER=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godzina poczatku			
STER= ADR_G11=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z			
STER= ADR_G11=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi			
STER= ADR_G11=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer			
STER= ADR_G11=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze			
STER= ADR_G11=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę			
STER= ADR_G11=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką			
STER= ADR_G11=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.			
STER= ADR_G11= ADR_M11=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Adres 16-bitowego rejestru (holding register)			
STER= ADR_G11= ADR_M11=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Adres 16-bitowego rejestru (holding register) z minutami początku pierwszego przedziału czasowego.			
STER= ADR_G11= ADR_M11= ADR_G12=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Adres 16-bitowego rejestru (holding register) z minutami początku pierwszego przedziału czasowego.			
STER= ADR_G11= ADR_M11= ADR_G12=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Adres 16-bitowego rejestru (holding register) z minutami początku pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną końca pierwszego przedziału czasowego			
STER= ADR_G11= ADR_M11= ADR_G12= ADR_M12=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Adres 16-bitowego rejestru (holding register) z minutami początku pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną końca pierwszego przedziału czasowego Adres 16-bitowego rejestru (holding register)			
STER= ADR_G11= ADR_M11= ADR_G12= ADR_M12=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Adres 16-bitowego rejestru (holding register) z minutami początku pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną końca pierwszego przedziału czasowego.			
STER= ADR_G11= ADR_M11= ADR_G12= ADR_M12= ADR_G21=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Adres 16-bitowego rejestru (holding register) z minutami początku pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z minutami końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z minutami końca pierwszego przedziału czasowego.			
STER= ADR_G11= ADR_M11= ADR_G12= ADR_M12= ADR_G21=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Adres 16-bitowego rejestru (holding register) z minutami początku pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z minutami końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z minutami końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z minutami końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z minutami końca pierwszego przedziału czasowego.			
STER= ADR_G11= ADR_M11= ADR_G12= ADR_M12= ADR_G21= ADR_M21=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Adres 16-bitowego rejestru (holding register) z minutami początku pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z minutami końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną początku drugiego przedziału. Adres 16-bitowego rejestru (holding register) z godziną początku drugiego przedziału. Adres 16-bitowego rejestru (holding register)			
STER= ADR_G11= ADR_M11= ADR_G12= ADR_M12= ADR_G21= ADR_M21= ADR_022	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Adres 16-bitowego rejestru (holding register) z minutami początku pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z minutami końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z minutami końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z minutami końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną początku drugiego przedziału. Adres 16-bitowego rejestru (holding register) z godziną początku drugiego przedziału.			
STER= ADR_G11= ADR_M11= ADR_G12= ADR_G12= ADR_G21= ADR_G21= ADR_G22=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Adres 16-bitowego rejestru (holding register) z minutami początku pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z minutami końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną początku drugiego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną początku drugiego przedziału. Adres 16-bitowego rejestru (holding register)			
STER= ADR_G11= ADR_M11= ADR_G12= ADR_G12= ADR_G21= ADR_G21= ADR_G22= ADR_G22= ADR_G22=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Adres 16-bitowego rejestru (holding register) z minutami początku pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną początku drugiego przedziału. Adres 16-bitowego rejestru (holding register) z godziną początku drugiego przedziału. Adres 16-bitowego rejestru (holding register) z godziną początku drugiego przedziału. Adres 16-bitowego rejestru (holding register) z godziną końca dugiego przedziału. Adres 16-bitowego rejestru (holding register) z godziną końca dugiego przedziału czasowego.			
STER= ADR_G11= ADR_M11= ADR_G12= ADR_G12= ADR_M12= ADR_G21= ADR_G21= ADR_G22= ADR_G22= ADR_M22=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.</i> Adres 16-bitowego rejestru (holding register) z godziną początku pierwszego przedziału czasowego. Rejestr dostępny do odczytu/zapisu z wykorzystaniem protokołu Modbus RTU lub Modbus TCP. Adres musi być z zakresu zadeklarowanego w pliku sterX.txt gdzie X oznacza numer sterownika (SL_DADR do SL_DADR+SL_DSIZE-1). W tym parametrze jako adres rejestru zamiast wartości liczbowej można podać nazwę rejestru. W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów. Adres 16-bitowego rejestru (holding register) z minutami początku pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną końca pierwszego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną początku drugiego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną początku drugiego przedziału. Adres 16-bitowego rejestru (holding register) z godziną początku drugiego przedziału. Adres 16-bitowego rejestru (holding register) z godziną początku drugiego przedziału. Adres 16-bitowego rejestru (holding register) z godziną końca dirugiego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z minutami początku drugiego przedziału czasowego. Adres 16-bitowego rejestru (holding register) z godziną końca drugiego przedziału czasowego. Adres 16-bitowego rejestru (holding register)			

Pokazany poniżej obiekt <programbar> opisuje wyświetlany program tygodniowy (TYP=PROGRAM) w postaci graficznej. Jest to program tygodniowy zwykle używany w sterownikach firmy Frisko. Obiekt będzie wyświetlany na współrzędnych x:240 i y:50 (X=240,Y=50), podane współrzędne dotycza środka górnej krawędzi (ROWNAJ=centralnie). Okresy zawierające się w zadeklarowanym przedziale czasowym będą wyświetlane w kolorze czerwonym (KOLOR_HI=255,0,0), natomiast okresy z poza zadeklarowanego przedziału w kolorze niebieskim (KOLOR_LO=0, 0, 205). Wszystkie teksty w tym obiekcie będą w kolorze szarym (KOLOR TXT=105,105,105). Wszystkie dane potrzebne do wyświetlenia tej graficznej reprezentacji programu są odczytywane ze sterownika zdefiniowanego w pliku ster0.txt (STER=0).

<programbar> TYP=PROGRAM X=240 Y=50 ROWNAJ=centralnie KOLOR HI=255,0,0 KOLOR_LO=0,0,205 KOLOR_TXT=105,105,105 STER=0 ADR G11=4181 ADR_M11=4185 ADR G12=4182 ADR_M12=4186 ADR_G21=4183 ADR_M21=4187 ADR_G22=4184 ADR M22=4188 </>

Obiekt typu RAMKA – kolorowy prostokąt

Obiekt typu RAMKA jest obiektem, który umożliwia narysowanie kolorowego prostokąta z zaokrąglonymi rogami z napisem. Tekst wyświetlany jest na górnej krawędzi ramki i zaczyna się zawsze z lewej strony. Ten obiekt często stosuję się z obiektami typu VAR16.

zmienna	opis					
TYP=	Typ danego obiektu.					
NAZWA=	Wyświetlana nazwa na górnej krawędzi ramki. Maksymalna długość nazwy to 16 znaków. Należy zadbać, aby napis nie był zbyt długi i nie wykraczał poza szerokość ramki. Napis zawsze zaczyna się od lewej strony.					
X=	Współrzędna x pozycji wyświetlanego obiektu. W zależności od parametru ROWNAJ może to być lewa lub prawa krawędź obiektu albo środek.					
Y=	Współrzędna y lewego górnego roku wyświetlanego obiektu. 0 oznacza, że obiekt będzie narysowany maksymalnie przy górnej krawędzi ekranu.					
W=	Szeroko	ść obiektu.				
H=	Wysokos	ść obiektu.				
ROWNAJ=	 Parametr określa, od której strony ma być narysowany obiekt (nie dotyczy to napisu określonego w parametrze NAZWA). Opcje: do lewej – obiekt będzie narysowany od lewej strony czyli współrzędne x i y oznaczają lewy górny róg obiektu, centralnie – obiekt będzie narysowany tak, że współrzędne x i y oznaczają środek górnej krawędzi obiektu, do prawej – obiekt będzie narysowany od prawej strony czyli współrzędne x i y oznaczają prawy górny róg obiektu. 					
FONT=	Wielkość	czcionki, Op	ocje: 12, 16, 2	20, 24, 36.		
		FONT	wysokość	szerokość	szerokość małych elementów (. , : ;) 4	
		16	16	11	5	
		20	20	13	6	
		24	24	15	7	
		36	36	24	10	
	Powyższa tabela może być przydatna, jeśli chcemy na ekranie dokładnie rozmieścić kilka tekstów lub wyświetlaczy numerycznych o różnych rozmiarach czcionek i różnej długości znaków. Wszystkie litery i cyfry mają taka samą szerokość z wyjątkiem kilku znaków interpunkcyjnych: .,:;.itp. Zbyt duża czcionka może spowodować że napis wyjdzie w prawo poza zarys ramki.					
KOLOR=	Kolor ramki w formacie RGB 24b (24-bitowym). Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem.					
STER=	Numer sterownika, którego dotyczy dany obiekt. W pliku o podanym numerze zapisane są wszystkie parametry transmisji w tym adres Modbus sterownika. Numer sterownika jest potrzebny tylko, jeśli korzystamy z warunkowego wyświetlania podanego w następnej linijce – ENABLE. <i>Przykład:</i> <i>STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0 txt</i>					
ENABLE=	Warunek wyświetlania danego obiektu. Jeśli chcemy żeby dany obiekt był wyświetlany zawsze to możemy całkowicie pominąć linijkę z tym warunkiem ENABLE. Jeśli chcemy, aby dany obiekt był wyświetlany w zależności od jakiegoś warunku to podajemy go w tej linijce po słowie ENABLE.					

Przykłady:					
ENABLE=4165=1	- obiekt	będzie	wyświetlany,	jeśli	W
rejestrze 4165	jest wartość	równa 1.			
ENABLE=4165>5	– obiekt	będzie	wyświetlany,	jeśli	W
rejestrze 4165	jest wartość	większa	niż 5.		
ENABLE=4165<3	- obiekt	będzie	wyświetlany,	jeśli	W
rejestrze 4165	jest wartość	mniejsza	a niż 3.		
ENABLE=4165!0	- obiekt	będzie	wyświetlany,	jeśli	W
rejestrze 4165	jest wartość	różna od	d 0.		
W warunkach róv	vnv '=' lub różr	ıv '!'. zam	iast reiestru moż	że być uż	vta
flaga W tym na	rametrze jako	adres reie	stru (flagi) zami	ast warto	ści
liozbowoj możno	nodoć nozwo	rojostru	(flogi) M/ tym		3
	pouac nazwę		(ilagi). W tylii		≠∠y
najpierw przygoto	wac plik (rejestr	y.txt) z tak	ą listą nazw rejes	strow.	

Pokazany poniżej obiekt <ramka1> opisuje prostokątną ramkę (TYP=RAMKA) z napisem "Tzadana" (NAZWA=Tzadana). Ramka o wymiarach 80 na 50 pikseli (W=80, H=50) będzie wyświetlana na współrzędnych x:200 i y:100 (X=200, Y=100), podane współrzędne dotyczą lewego górnego narożnika (ROWNAJ=do lewej).

Ramka i napis będzie w kolorze czerwonym (KOLOR=255,0,0), a czcionka będzie w rozmiarze 20 (FONT=20).

<ramka1> TYP=RAMKA NAZWA=Tzadana X=200 Y=100 W=80 H=50 ROWNAJ=do lewej FONT=20 KOLOR=255,0,0 STER=0 </> Obiekt typu BOX_ALERT – obiekt wyświetlający na całym ekranie alarmy

Obiekt typu BOX_ALERT wykorzystuje cały ekran i umożliwia wyświetlenie w wierszach do 8 aktualnie występujących awarii. Wielkość czcionki jest stała i wynosi 24.

zmienna	opis
TYP=	Typ danego obiektu.
KOLOR=	Kolor wyświetlanej listy z alarmami w formacie RGB 24b (24-bitowym).
	Wagi poszczególnych kolorów muszą być rozdzielone przecinkiem.
LISTA=	Nazwa pliku z listą alarmów do wyświetlania. W tym dodatkowym pliku
	zapisane są następujące informacje: numer sterownika, numer rejestru,
	wartość tego rejestru i treść komunikatu. Tej samej listy można użyć w
	obiekcie LISTA_ALERT.

Przykład

Pokazany poniżej obiekt <alert> opisuje listę wyświetlanych awarii (TYP=BOX_ALERT). Wyświetlane teksty będą w wierszach na całym ekranie w kolorze czerwonym (KOLOR=250,0,0). Lista wyświetlanych alarmów będzie zadeklarowana w pliku "I_alert.txt" (LISTA=1_alert), poniżej przykład takiego pliku.

```
<boxalert>
TYP=BOX_ALERT
KOLOR=250,0,0
LISTA=l_alert
</>
```

W poniższym pliku w każdej linijce mamy cztery parametry oddzielone średnikiem:

- numer sterownika, (np. 0 oznacza, że zmienna dotyczy sterownika opisanego w pliku ster0.txt),

- numer rejestru,

- wartość tego rejestru, dla którego występuje dana awaria,

- wyświetlany tekst dla danej awarii,

Możemy zadeklarować do 64 elementów (różnych awarii) o maksymalnej długości tekstu 32 znaki.

W poniższym przykładzie z listą alarmów widzimy, że jeśli w sterowniku o numerze 0 opisanym w pliku ster0.txt w rejestrze 4101 będzie wartość 1 to zastanie wyświetlony komunikat "Awaria czujnika Tkan!".

```
Plik I_alert.txt
```

```
0;4101;1;Awaria czujnika Tkan!
0;4102;1;Awaria czujnika Tzew!
0;4111;1;Brudny filtr czerpni!
0;4112;1;Brudny filtr nawiewu!
1;4201;1;Awaria czujnika Twel!
1;4202;1;Awaria czujnika Twe2!
```

W liście alertów jako adres rejestru (flagi) zamiast wartości liczbowej można podać nazwę rejestru (flagi). W tym celu należy najpierw przygotować plik (rejestry.txt) z taką listą nazw rejestrów.

Obiekt typu STATUS_COM – status komunikacji ze sterownikiem

Obiekt typu STATUS_COM wyświetla symbol statusu komunikacji ze sterownikiem.

zmienna	opis
TYP=	Typ danego obiektu.
X=	Współrzędna x lewego górnego roku wyświetlanego obiektu. 0 oznacza,
	że obiekt będzie narysowany maksymalnie przy lewej krawędzi ekranu.
Y=	Współrzędna y lewego górnego roku wyświetlanego obiektu. 0 oznacza,
	że obiekt będzie narysowany maksymalnie przy górnej krawędzi ekranu.
STER=	Parametr określa, którego zadeklarowanego sterownika dotyczy status
	komunikacji.
	Przykład:
	STER=0 oznacza, że obiekt dotyczy sterownika opisanego w pliku ster0.txt.

Przykład

Pokazany poniżej obiekt <com> opisuje status komunikacji (TYP=STATUS_COM) ze sterownikiem opisanym w pliku ster0 (STER=0).

<com> TYP=STATUS_COM X=5 Y=300 STER=0 </>

Specyfikacja techniczna

Wyświetlacz	Тур	3,5" TFT
-	Aktywny obszar	73,44 x 48,96 mm (SxW)
	Rozdzielczość	480 x 320 pikseli
	Rozmiar pixela	0,153 x 0,153 mm
	Liczba kolorów	65K RGB 16bit
	Jasność (cd/m2)	300
	Rekomendowany kierunek patrzenia	12 godzina
	Kąty widzenia	50°, 60°, 60°, 60°
	Typ podświetlania	LED
	Żywotność podświetlanie	>30000 godzin
Matryca dotykowa	Τνρ	Bezystancvina
	Precyzia	(X)+2% $(Y)+2%$ (danei długości
	1 10092ju	boku obszaru aktywnego)
	Draaaaar	
Jeonosika centraina	Procesor	96MHz
	Pamieć	1.4MB RAM
	L L	2MB Flash (system)
		8MB ExFlash (program i dane)
	RTC	Wbudowany
Porty	LISB Host	LISB 2.0 typ A (programowanie z
1 only		wykorzystaniem Pendrive)
	USB Klient	USB 2.0 typ micro-B (złacze
		systemowe)
	СОМ	RS485 2W
		obsługa Modbus RTU MASTER,
		prędkości transmisji:1200, 2400,
		4800, 9600, 19200, 38400 bps,
		format: 8-NONE-1, 8-EVEN-1,
		8-ODD-1.
	Ehternet	Wi-Fi (moduł ESP8266)
		obsługa Modbus TCP MASTER
Zasilanie	Napięcie zasilania	1224VDC
	Pobór prądu (tryb aktywny)	55mA@12VDC30mA@24VDC
	Pobór prądu (tryb uśpienia)	16mA@12VDC10mA@24VDC
Obudowa		Tablicowa, ABS
obudona	Wymiary	119 x 84 x 33 5 mm (SxWxG)
	Wymiary otworu montażowego	109 x 74 mm (SxW)
	Masa	165 g
Środowieka	- Staniań achrany	IP65 (front popolici) IP20 (roo-to
SIUUUWISKU		obudowy)
	Temperatura przechowywania	±5°C ±40°C
	Temperatura przechowywania	±5°C ±40°C
	Wilgotność względna	10% 90% (bez kondensacii)
1		