

NOTA APLIKACYJNA

| | |
|----------------------------|-----------------------------------|
| TYTUŁ: | Sekwencyjne sterowanie zaworami |
| URZĄDZENIA: | NeoGSM-IP-64 /-PS |
| WERSJA: | 1.1JH |
| DATA PIERWSZEJ PUBLIKACJI: | 17-08-2020 |

Prezentowana nota opisuje sposób konfiguracji centrali NeoGSM-IP-64 dla sekwencyjnego sterowania wyjściami, rozwiązanie może być użyte w przypadkach kiedy niezbędne jest uruchamianie poszczególnych urządzeń jedno po drugim a ich działanie w jednym czasie jest niepożądane, czyli na przykład w celu utrzymania odpowiedniego ciśnienia wody przy podlewaniu upraw w przypadku gdy istnieje instalacja do tego przeznaczona wyposażona w wiele zaworów, w takim przypadku otwarcie kilku zaworów na raz może spowodować nadmierny spadek ciśnienia wody w instalacji a co za tym idzie spadek wydajności lub nawet niepodlanie części areału do którego nie sięga wtedy zbyt słaby strumień wody. Przedstawione rozwiązanie kontroluje ilość i kolejność załączonych wyjść sterujących nie pozwalając na pracę więcej niż jednego na raz oraz pozwala na zautomatyzowanie całego procesu dzięki możliwości podłączenia odpowiednich czujników których występowanie może spowodować rozpoczęcie sekwencji podlewania bez potrzeby ingerencji użytkownika. W dalszej części opis jest ukierunkowany właśnie na takie rozwiązanie, należy jednak pamiętać, że tak przygotowana centrala może w analogiczny sposób sterować dowolnymi urządzeniami podłączonymi do jej wyjść.

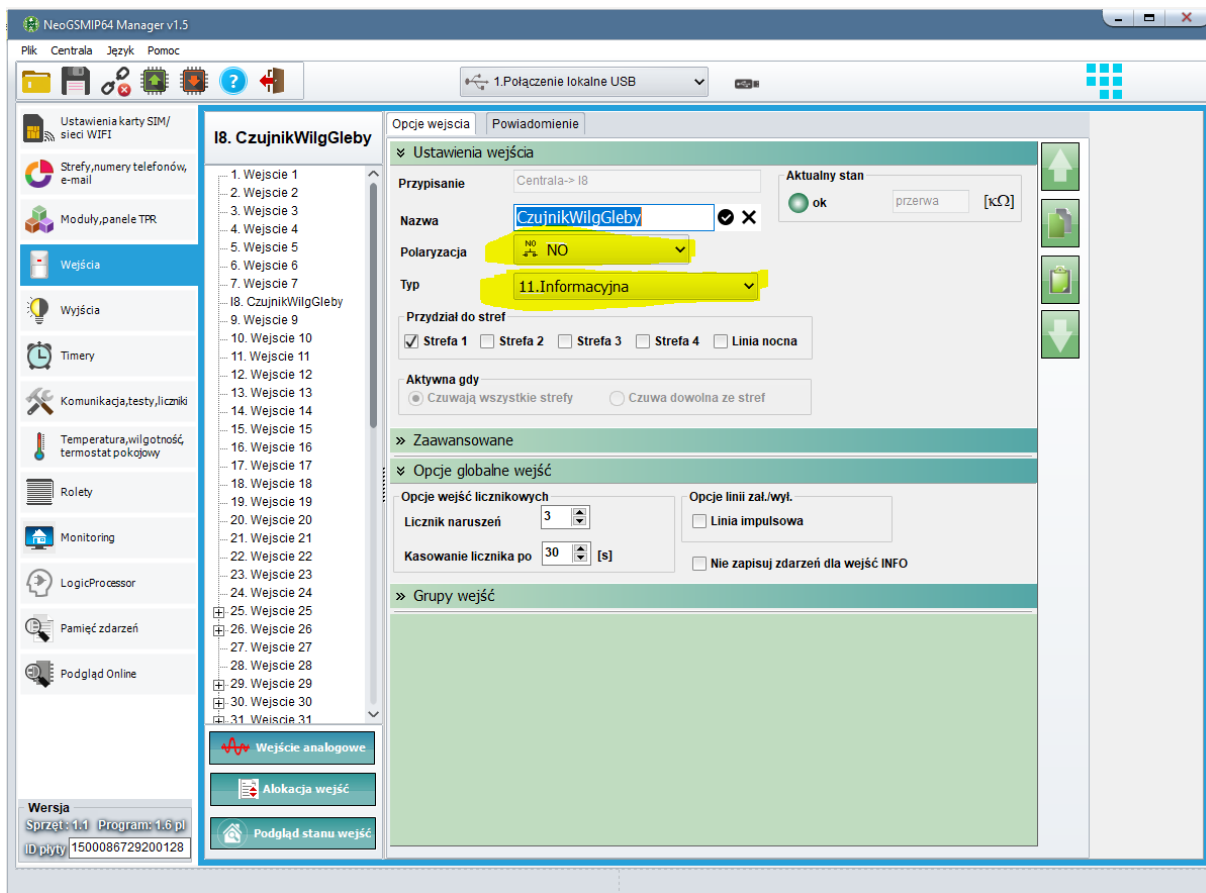
Ponieważ nota z zasady jest przeznaczona do sterowania urządzeniami zewnętrznymi dlatego zaleca się wykorzystanie jako wyjść ekspanderów przekaźnikowych [EXP-O8R-RN-D9M](#) z naszej oferty.

Założenia:

- Możliwość sekwencyjnego sterowania kilkoma/kilkunastoma wyjściami.
- Na raz może być załączone tylko jedno wyjście.
- Każde wyjście ma indywidualnie ustawiony czas pracy.
- Praca w trybach: automatycznym (w pełni automatyczne wyzwolenie sekwencji), półautomatycznym (sekwencja jest wyzwalana przez użytkownika), ręcznym (każdy zawór jest sterowany indywidualnie przez użytkownika).
- Priorytet sekwencji automatycznej, nad ręczną i sterowaniem indywidualnym.
- Możliwość zatrzymania sekwencji w dowolnym momencie.
- Kontrola i sterowanie zarówno lokalne (np. panel TPR), jak i zdalne (aplikacja mobilna RopamNeo).

Konfiguracja wejścia:

W przykładzie do wejścia podłączony jest czujnik wilgotności gleby (można użyć dowolnego pod warunkiem, że na wyjściu generuje sygnał binarny). W takim przypadku wejście należy skonfigurować jako typu informacyjnego z polaryzacją NC lub NO w zależności od typu podłączonego czujnika. Dane wejście jest jednym ze źródeł sygnału dla automatycznego wyzwolenia sekwencji.

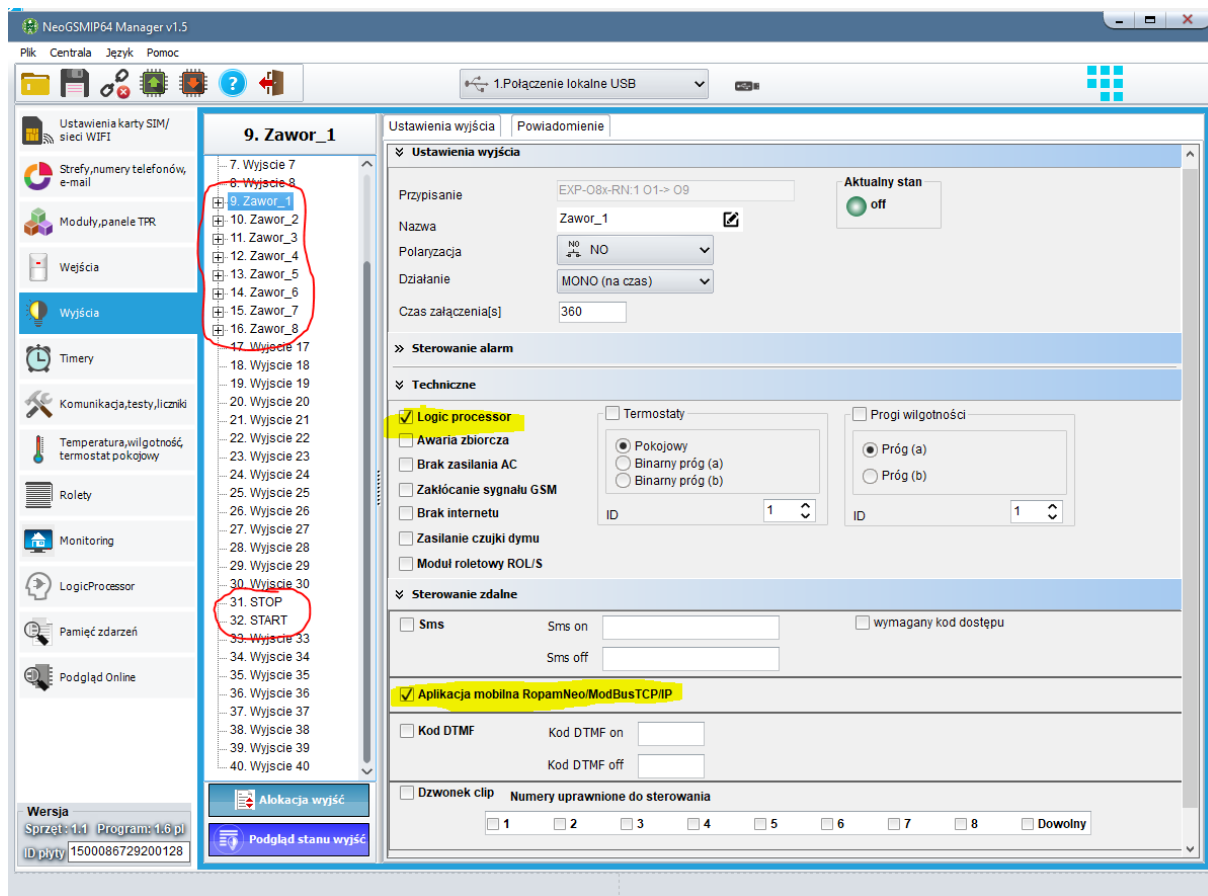


Konfiguracja wyjść:

Nota opisuje konfigurację ośmiu wyjść, przy czym w danej aplikacji wyjść może zarówno mniej jak i więcej, maksymalna ilość dostępny do sterowania wyjść wynosi około 18 co jest związane z maksymalną wielkością skryptu Logic Procesora (patrz opis skryptu), jak wyżej wspomniano ponieważ zaleca się użycie przekaźnikowego ekspandera wyjść dlatego w rozwiązaniu został on zastosowany (jakkolwiek konfiguracja wyjść nie przekaźnikowych wygląda identycznie). Oprócz ośmiu wyjść

wykonawczych konieczne jest użycie dwóch wyjść sterujących, jedno do wyłączenia działającej sekwencji i drugie do ręcznego wyzwolenia sekwencji, przy czym oba wyjścia są wirtualne tzn. nie mają fizycznego odpowiednika w instalacji i służą tylko doysterowania innych wyjść lub funkcji.

Na poniższym screenie widać konfigurację jednego z wyjść, pozostałe powinny być ustawione w identyczny sposób, czyli zaznaczone sterowanie przez Logic Procesor oraz jeżeli wyjście ma być dostępne również przez aplikację zaznaczony dostęp przez aplikację mobilną (można również zaznaczyć sterownie SMS lub kodem DTMF). Pierwsze osiem wyjść to wyjścia wykonawcze na ekspanderze przekaźnikowym, ostatnie dwa do wirtualne wyjścia sterujące.



Konfiguracja timera

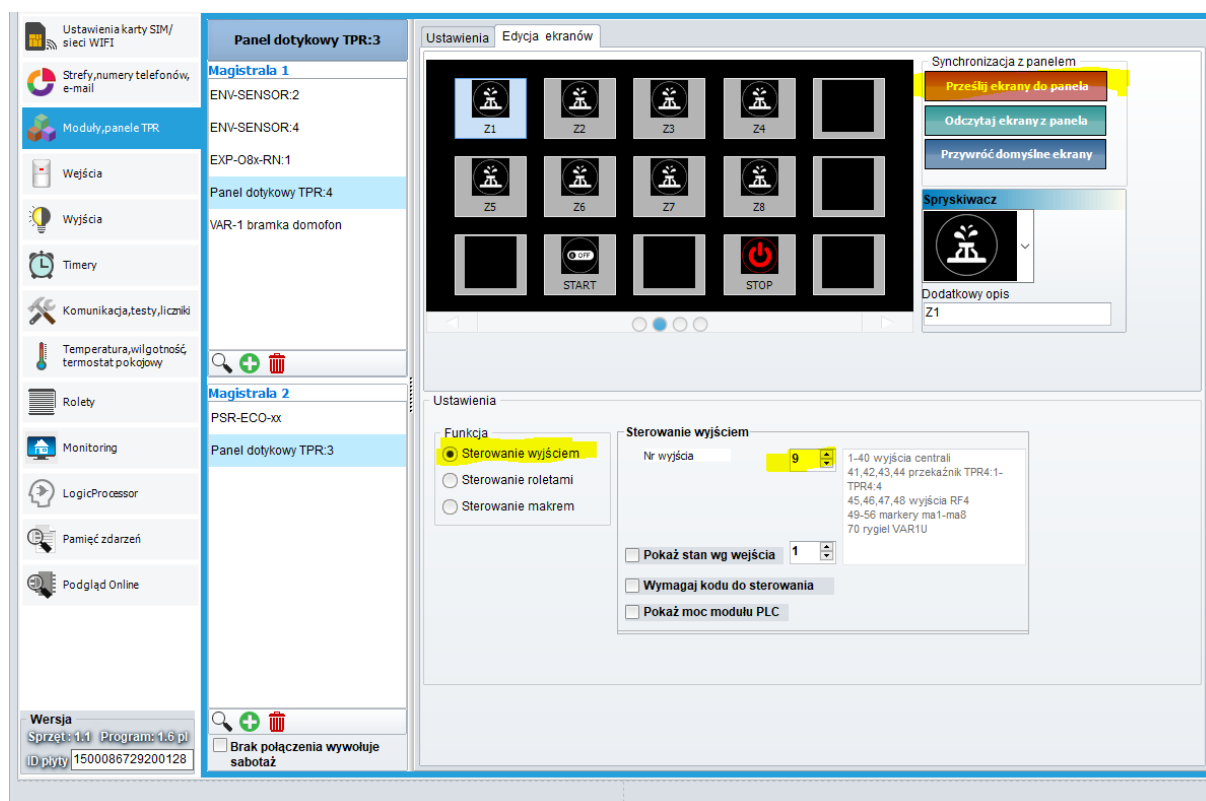
W prezentowanym rozwiązaniu automatyczna sekwencja może uruchomić się tylko w określonych godzinach (tutaj dokładnie pomiędzy 21:00 a 06:00), w tym celu należy odpowiednio skonfigurować jeden z timerów kalendarzowych centrali (rys. poniżej)

| Lp | Stan | Rok | Miesiąc | Dzień | Czas | Dzień tyg. | | |
|----|-------|-----|---------|-------|----------|------------|---|---|
| 1 | 1 ON | | | | 21:00:00 | | + | - |
| 2 | 0 OFF | | | | 06:00:00 | | + | - |

Tryb timera
 roczny dzienny
 miesięczny stały
 tygodniowy wyłączony

Konfiguracja sterowania z TPR-a:

Istnieje możliwość sterowania z poziomu panela TPR w tym celu należy skonfigurować jeden z ekranów tak aby udostępniał sterowanie każdym wyjściem z osobna oraz umożliwiał ręczne wystartowanie sekwencji, jej zatrzymanie i wyłączenie aktywnych wyjść, można to zrobić tak jak na rysunku poniżej.



Ikony od Z1 do Z9 są skonfigurowane do sterowania konkretnym wyjściem (czyli Z1 steruje wyjściem nr 9 Z2 wyjściem nr 10 itd.), START uruchamia sekwencję (jest do niego przypisane wyjście nr 32), zaś stop powoduje wyłączenie wszystkich wyjść sterujących i zatrzymanie sekwencji (wyjście nr 31).

W analogiczny sposób można skonfigurować panel sterowniczy w aplikacji mobilnej.

Logic procesor i diagram editor

Prezentowany skrypt zawiera w sobie kompletny układ sterowania wyjściami, gdzie wyjście może byćysterowane indywidualnie lub automatycznie jako część sekwencji (czyli kolejnego załączania wyjść na określony czas), sterowanie sekwencyjne posiada możliwość wyzwolenia ręcznego (sterowanie półautomatyczne) lub automatycznego po spełnieniu określonych warunków. Układ posiada zabezpieczenia przed włączeniem kilku wyjść na raz zarówno w przypadku sterowania sekwencyjnego jak i indywidualnego, posiada również priorytet sterowania automatycznego, czyli w przypadku kiedy zostanie uruchomiona sekwencja automatyczna najpierw wszystkie wyjścia są wyłączane (co powoduje również przerwanie sekwencji jeżeli akurat trwała) a dopiero potem następuje start nowej sekwencji. Ponadto istnieje również możliwość ręcznego zatrzymania trwającej sekwencji, jak i wyłączenia wyjściaysterowanego indywidualnie.

Całość jest stworzona głównie z myślą o sterowaniu zaworami służącymi do podlewania upraw, gdzie przy większej liczbie nie można otworzyć wszystkich naraz albowiem mogłoby to spowodować nadmierny spadek ciśnienia wody co powodowałoby nierównomierne nawadnianie (im dalej od ujęcia wody tym mniejszy przepływ). Dlatego skrypt uruchamia zawory po kolei i uniemożliwia włączenie na raz więcej niż jednego (testy praktyczne wykazały, że przy próbie włączenia więcej niż jednego wyjścia te dodatkowe mogą się załączyć na krótki okres rzędu ~0,5 sekundy.)

Ponieważ rozwiązanie, w szczególności, obejmujące więcej niż 2 – 4 wyjścia zawiera dużo elementów na schemacie, co zmniejsza jego czytelność, dlatego całość została podzielona na mniejsze układy które spełniają określone funkcje (np.: sterują wyjściem lub generują sygnał zatrzymania sekwencji), każdy taki układ posiada linie wejściowe na które są przekazywane sygnały z innych i na ich podstawie albo zmienia swój stan, albo generuje inny sygnał który poprzez linie wyjściowe służy do sterowania całości. Taka modułowa konstrukcja umożliwia również łatwą rozbudowę o kolejne wyjścia, ponieważ w takim wypadku ogranicza się to do dodania elementów stanowiących układ sterowania wyjściem i połączeniu go z liniami I/O reszty układu.

Sygnały generowane przez układ są zerojedynkowe czyli linia sygnałowa może posiadać dwa stany wysoki lub niski (1 lub 0) dlatego w opisie dana nazwa linii odnosi się również do sygnału, czyli np.:

wysterowanie linii SP znaczy to samo co podanie sygnału SP w obu przypadkach chodzi o podanie na linię SP stanu wysokiego.

Opis linii sygnałowych:

SP – Sygnał trwającej sekwencji podlewania, stan wysoki („1”) oznacza uruchomioną sekwencję

SS – Sygnał inicjalizujący sekwencję

SOx2 – Sygnał włączenia wyjścia, jeżeli wyjście jest włączone na linii panuje stan wysoki, linię SOx2 łączy się z linią SOx1 następnego wyjścia, oraz blokiem sumy logicznej w generatorze sygnałów SOx i ISOx (każde wyjście generuje własny SOx2).

SOx1 – sygnał załączenia poprzedniego wyjścia, informuje o stanie w jakim jest wyjście poprzedzające w sekwencji dane wyjście (w wyjściu które jest pierwsze w sekwencji linia jest nie podłączona), stan wysoki oznacza włączone wyjście, linia jest podłączana do linii SOx2 poprzedniego wyjścia.

WY – sygnał wyłączenia, podanie na linię WY stanu wysokiego spowoduje wyłączenie wszystkich aktywnych wyjść i przerwanie sekwencji.

SOx – Sygnał załączonego dowolnego wyjścia, stan wysoki („1”) oznacza, że jedno z wyjść jest aktywne.

ISOx – Zanegowany sygnał załączonego dowolnego wyjścia, stan wysoki („1”) oznacza, że żadne z wyjść nie jest aktywne.

RS – Sygnał automatycznego zatrzymania, czyli wyzwalacz sygnału WY.

Układ sterowania wyjściem:

Sterowanie wyjściem może odbywać się w sposób automatyczny lub półautomatyczny w sekwencji, pierwsze wyjście jest wysterowane przez zadany Trigger (w trybie półautomatycznym ręcznie), a następne są wysterowane poprzez wyłączenie poprzedniego. Sterowanie może odbywać się również poprzez indywidualne załączenie wybranego wyjścia w takim wypadku wyłączenie wyjścia nie spowoduje załączenia następnego. Warunkiem koniecznym wysterowania wyjścia manualnie lub półautomatycznie jest wyłączenie wszystkich innych wyjść, czyli nie może trwać sekwencja ani żadne

wyjścia na czas określony w bloku R1 (tutaj jest to 15 sekund). W momencie wyłączenia wyjścia na linii SOx2 jest generowane zbocze opadające które może być użyte do wysterowania kolejnego wyjścia w sekwencji. Warto zwrócić uwagę na to, że w pierwsze wyjście jest uruchamiane nie przez sygnały SOx2 i SP a przez osobny sygnał inicjalizujący sekwencję czyli SS, a więc w takim wypadku brak jest bloków M6 i E2 ze schematu powyżej a sygnał SS jest podawany bezpośrednio na wejście S bloku R1.

Uruchomienie ręczne:

Jeżeli wyjście zostało wysterowane ręcznie na wyjściu z O10 pojawi się sygnał „1” jest on podawany na jedno z wejść M7 (AND), na drugie wejście jest podawany sygnał będący zanegowana sumą logiczną wszystkich pozostałych wyjść (ma wartość „1” gdy wszystkie wyjścia są nieaktywne i „0” gdy którekolwiek jest aktywne) jeżeli wszystkie wyjścia są wyłączone nastąpi wysterowanie wyjścia z M7 a przez to wysterowanie wejścia „O” w bloku R1, co spowoduje załączenie wyjścia na ustalony czas. Jednocześnie ponieważ linia SP nie jest wysterowana wyłączenie wyjścia i spowodowane tym pojawienia się na linii SOx2 zbocza opadającego nie spowoduje załączenia kolejnego.

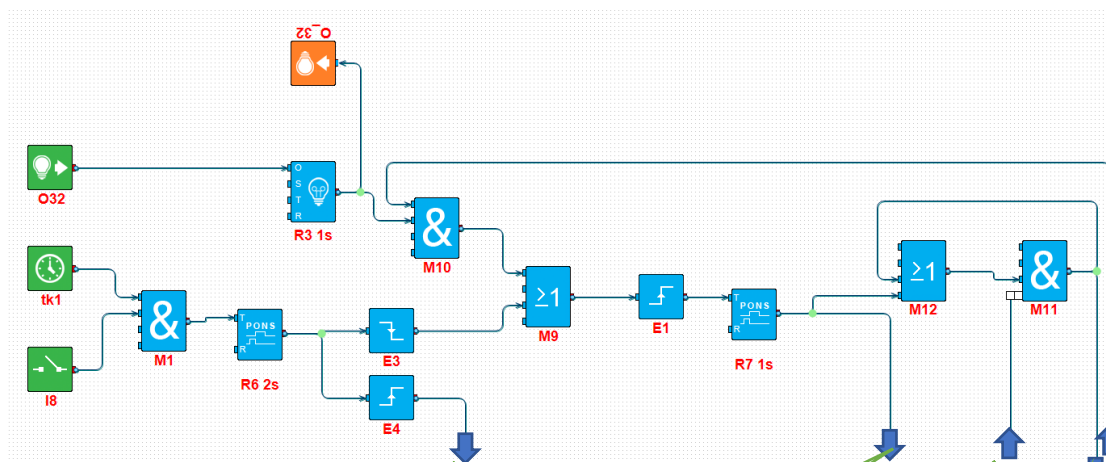
Wyłączenie wyjść:

Podanie na linię RW sygnału „1” powoduje wysterowanie wejścia R w bloku R1 i tym samym wyłączenie wyjścia, jeżeli RW jest podpięta do każdego wyjścia spowoduje to wyłączenie wszystkich wyjść na raz i tym samym przerwanie sekwencji.

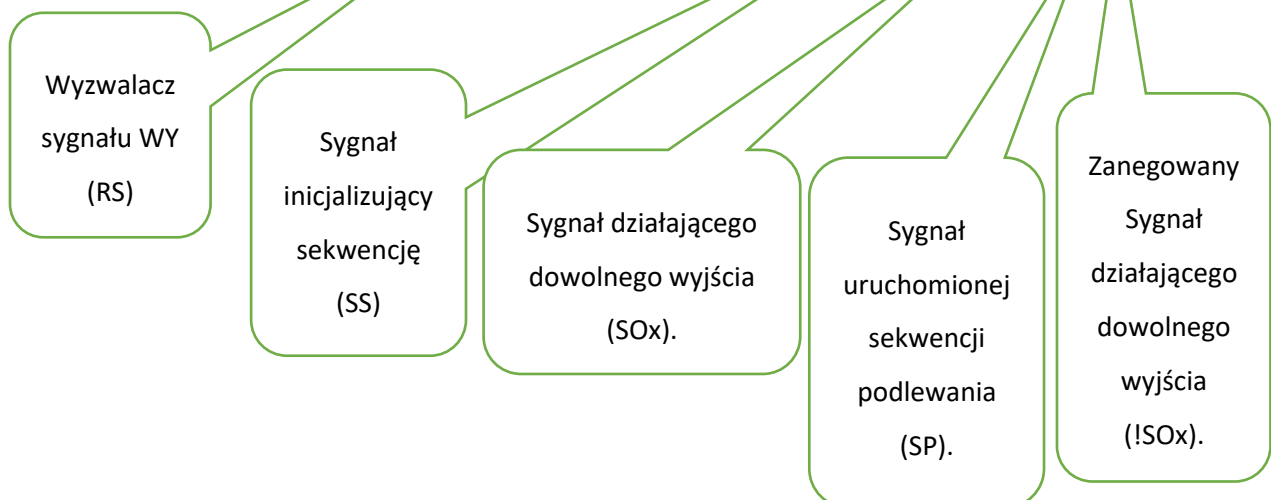
Układ wyzwalający sekwencję (trigger):

Sekwencyjne sterowanie wyjść można uruchomić automatycznie (w prezentowanym rozwiązaniu do automatycznego wyzwolenia sekwencji służy timer kalendarzowy tk1 i wejście I8 do którego jest podpięty czujnik wilgotności gleby) lub na żądanie w dowolnym momencie w tym celu należy przesterować wyjście O32.

Poniżej widać screen z fragmentem diagramu przedstawiającym kompletny układ wyzwalacza sekwencji podlewania.



Schemat 2



Ręczne wyzwolenie sekwencji (sekwencja półautomatyczna):

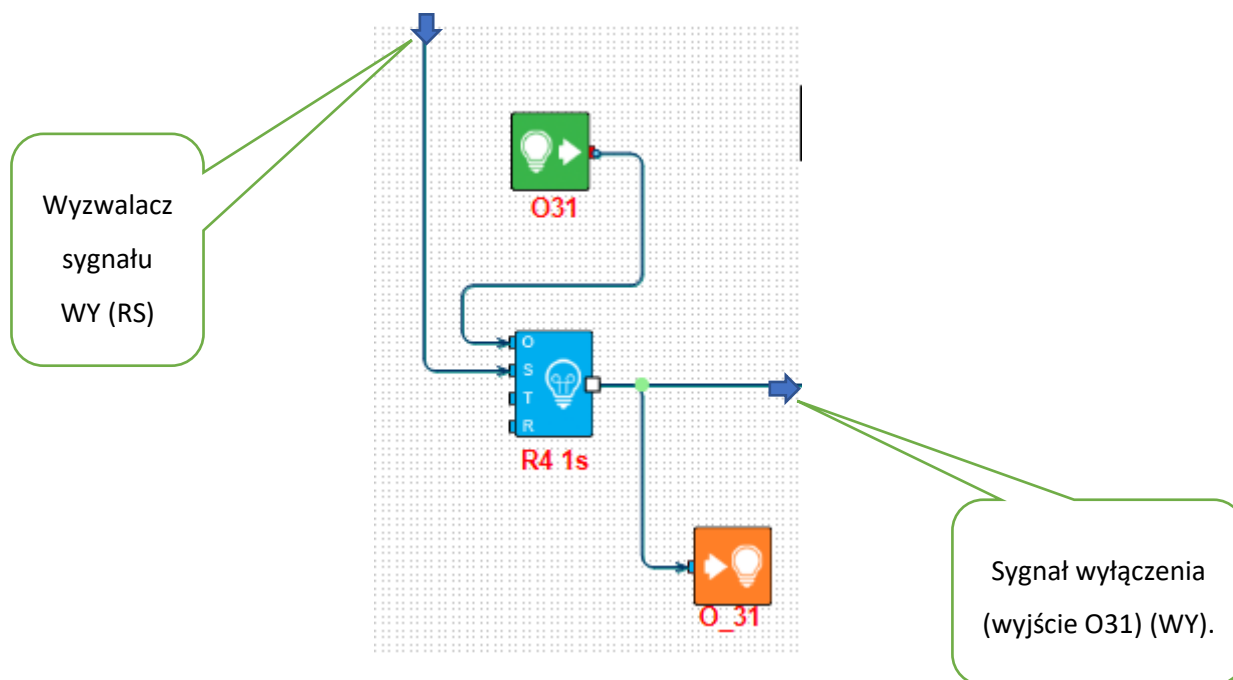
Po wystereowaniu wyjścia O32 na wyjściu bloku R3 pojawi się sygnał „1” jest on podawany wspólnie z sygnałem !SOx na wejście bloku iloczynu logicznego M10, jeżeli jest aktywne którekolwiek wyjście (czyli !SOx będzie w podawać „0”) M10 nadal będzie podawał „0” i sekwencja nie zostanie wyzwolona (w takim wypadku należy albo poczekać do wyłączenia wyjść, albo uczynić to ręcznie przy pomocy wyjścia O31), w przypadku gdy żadne z wyjść nie jest aktywne (czyli !SOx =1) M10 poda na blok sumy logicznej M9 sygnał „1”, na wyjściu M9 powstanie zbocze narastające wykryte przez E1 co poprzez przełącznik czasowy R7 wygeneruje jednosekundowy sygnał startowy sekwencji (SS), SS jest analogiczny do sygnału SOx1 różnica jest tylko w źródle, SOx1 jest generowany przez załączenie poszczególnych wyjść, zaś SS jest generowany przez Trigger i jest sygnałem startowym dla pierwszego wyjścia. SS jest podawany również na wejście bloku sumy logicznej M12, z M12 sygnał jest podawany na blok iloczynu logicznego M11, na drugie jego wejście jest podawany sygnał załączenia wyjść SOx, a zatem na wyjściu M11 (czyli na linii SP) pojawi się „1” tylko wtedy gdy wyjście zostało uruchomione przez sekwencję, SP jest podawane następnie z powrotem na M12, co tworzy samopodtrzymujący się układ który jest aktywny tak długo jak długo są załączone wyjścia. W skrócie, do wystereowania sygnału SP jest niezbędne uruchomienie sekwencji a podtrzymanie jest realizowane przez działające wyjścia.

Automatyczne uruchamianie sekwencji:

Sekwencja zostanie uruchomiona automatycznie jeżeli są spełnione warunki tk1 i I8 w takim wypadku zostanie wystereowane wyjście z bloku M1 co z kolei spowoduje uruchomienie przełącznika czasowego R6 na czas dwóch sekund. Należy zaznaczyć, że cały układ tk1, I8 i M1 może być dowolny tj. mogą pojawić się tutaj różne warunki do spełnienia, sama sekwencja uruchamia się w momencie podania na wejście R6 sygnału „1”. Uruchomienie R6 jest wykrywane przez blok E4 i generuje sygnał RS który poprzez układ wyłączający generuje sygnał WY który wyłącza wszystkie włączone wyjścia i tym samym umożliwia sterowanie. Po dwóch sekundach następuje wyłączenie R6 co jest wykrywane przez blok E3 który generuje sygnał podawany na blok sumy logicznej M9, dalsze działanie jest analogiczne do działania sekwencji uruchamianej ręcznie. Warto zwrócić uwagę na to, że w przypadku automatycznego uruchomienia sekwencji wykona się ona zawsze ponieważ bezpośrednio przed startem (czyli podaniem sygnału na M9) wyłączane są wszystkie ewentualnie włączone wyjścia, a zatem w momencie startu sekwencji na pewno żadne wyjście nie jest aktywne.

Układ wyłączający:

W razie potrzeby jest możliwe ręczne wyłączenie trwającej sekwencji, lub wyłączenie ręcznie włączonego wyjścia jest to zrealizowane na prostym układzie widocznym poniżej.

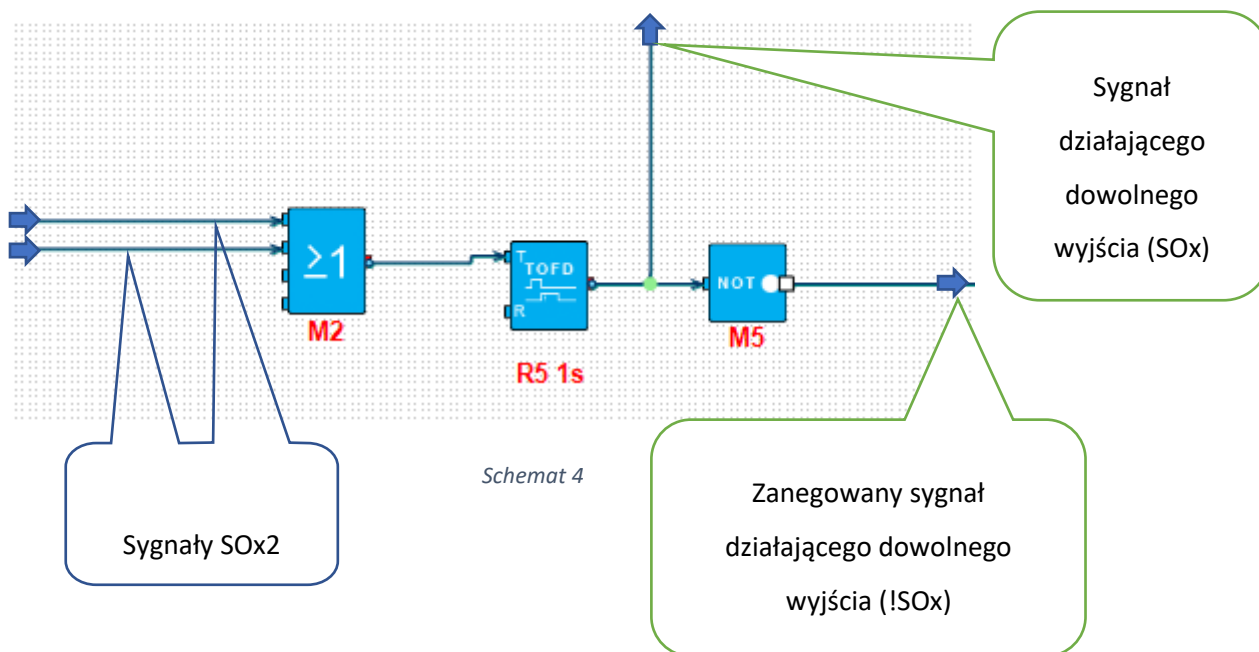


Schemat 3

Przesterowanie wyjścia O31 spowoduje wystawienie „1” na linii WY, WY jest podłączone do wejścia R w blokach sterujących wyjściami sygnał „1” podawany na to wejścia powoduje wyłączenie wyjścia (porównaj z opisem działania poszczególnych wyjść), sygnał może być też wyzwolony przez podanie „1” na linii RS dzieje się tak w momencie startu automatycznej sekwencji.

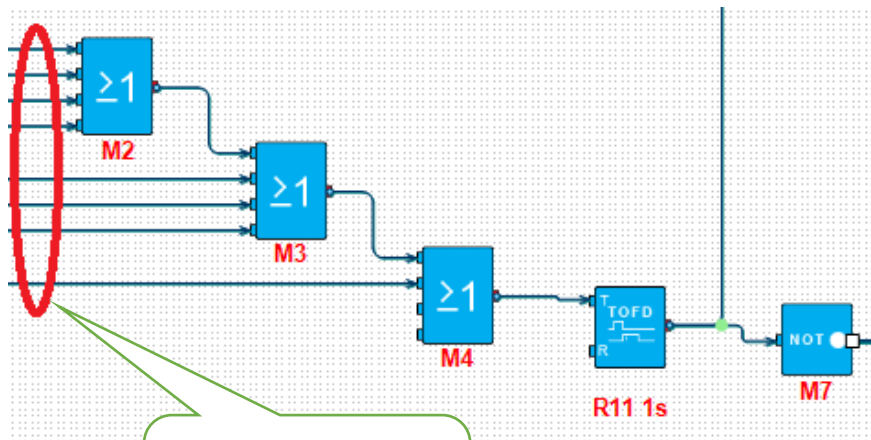
Układ generatora sygnałów SOx i !SOx:

Poniżej przedstawiony jest układ generujący sygnały informacyjne SOx i !SOx używane do sterowania.



Na blok sumy logicznej M2 są podawane sygnały załączenia poszczególnych wyjść (SOx2), a zatem na wyjściu z M2 pojawi się stan wysoki kiedy którekolwiek z wyjść jest włączone, sygnał jest kierowany na wejście T Przeźniacza czasowego R5 typu Time off Delay, powoduje to wystawienia wyjścia z R5, powrót w stan niski następuje dopiero sekundę po wyłączeniu wszystkich wyjść, zabezpiecza to przed zmianami sygnału w trakcie trwania sekwencji kiedy istnieją krótkie przerwy pomiędzy włączeniem poszczególnych wyjść. Dalej sygnał jest kierowany na bramkę NOT M5 która na wyjściu podaje odwrotność sygnału wejściowego (czyli jeżeli na linii SOx jest „1” na linii !SOx będzie „0” i na odwrót).

W przypadku używania większej ilości wyjść sterujących (a więc będących źródłem sygnału SOx2) należy bloki sum logicznych łączyć kaskady jak na schemacie poniżej, gdzie jest zaprezentowany układ generujący sygnały SOx i !SOx zbudowany dla ośmiu sygnałów SOx2 (a więc ośmiu wyjść sterujących).

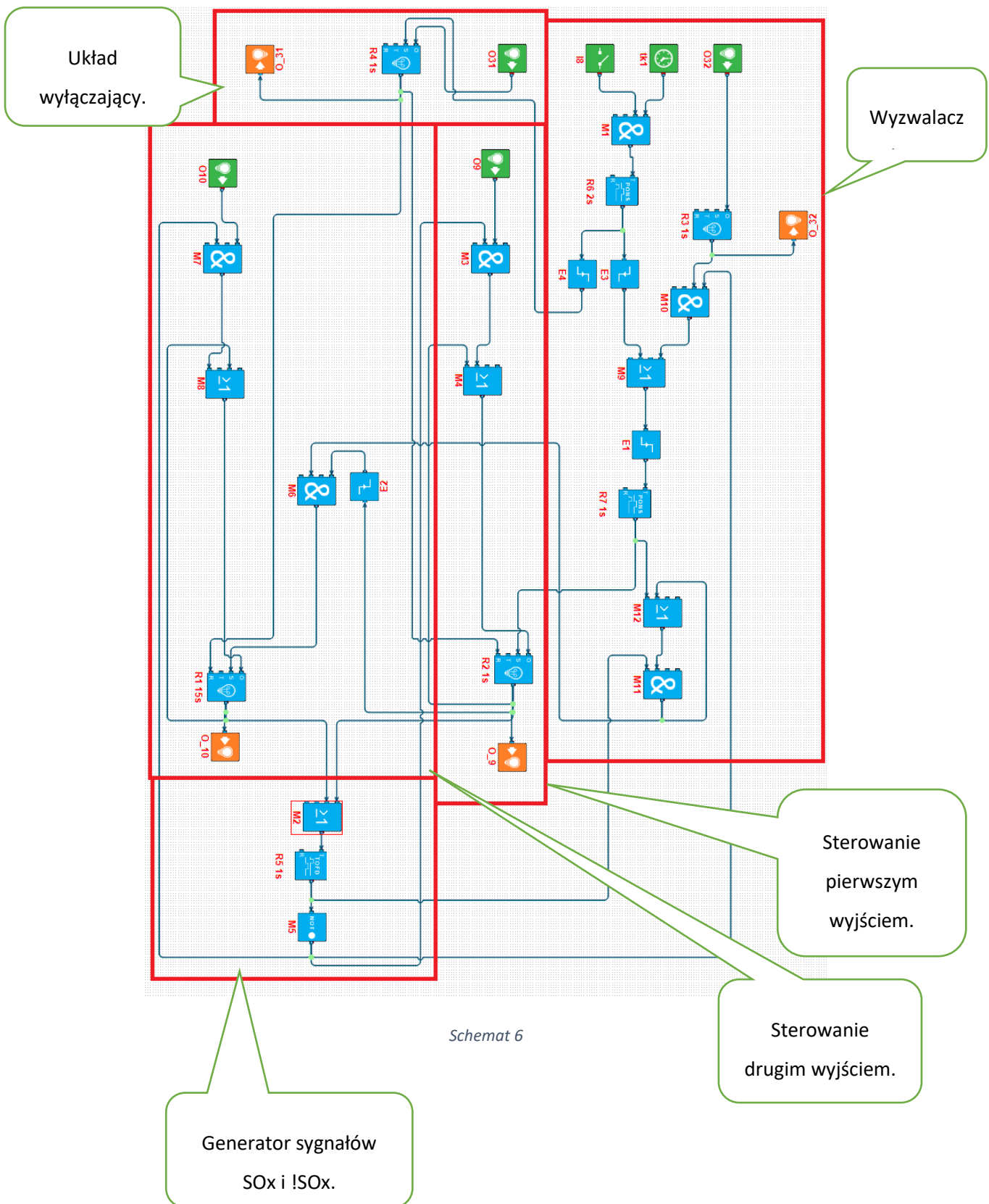


Schemat 5

Sygnaly SO_x2 z
poszczególnych wyjść

Połączenie poszczególnych sekcji w jeden układ:

Na sąsiedniej stronie widoczny jest przykład połączenia opisanych sekcji w jeden w układ sterowania, w celu zwiększenia czytelności schematu jest on zrealizowany tylko dla dwóch wyjść, rozbudowa o kolejne odbywa się przez dokładanie kolejnych sekcji sterujących wyjściami (patrz schemat 1) i ewentualną rozbudowę generatora sygnałów informacyjnych SOx i !SOx (schematy 4 i 5).



Kod wygenerowany dla powyższego schematu:

int E1;

int E2;

int E3;

int E4;

int I8;

int M1 ;

int M10 ;

int M11 ;

int M12;

int M2;

int M3 ;

int M4;

int M5;

int M6 ;

int M7 ;

int M8;

int M9;

int O10;

int O31;

int O32;

int O9;

int R1;

int R2;

int R3;

```
int R4;
int R5;
int R6;
int R7;
int tk1;
main()
{
while(1)
    {
        O9=geto(9);
        I8=geti(8);
        O10=geto(10);
        O32=geto(32);
        O31=geto(31);
        tk1=getenv("tk1");
        M1=I8&tk1;
        M3=O9&M5;
        M4=M3|R2;
        M2=R2|R1;
        R3=lsw(2,O32,0,0,0,1,1);
        R5=tofd(4,M2,0,1,1);
        M11=M12&R5;
        M5=!R5;
        M10=R3&M5;
        M7=O10&M5;
```

```
M8=R1|M7;

R6=pons(5,M1,0,2,1);

R2=lsw(1,M4,R7,0,R4,1,1);

M12=M11|R7;

R7=pons(6,E1,0,1,1);

E2=edge(1,0,R2);

M6=E2&M11;

R1=lsw(0,M8,M6,0,R4,15,1);

E3=edge(2,0,R6);

M9=M10|E3;

E1=edge(0,1,M9);

E4=edge(3,1,R6);

R4=lsw(3,O31,E4,0,0,1,1);

seto(9,R2);

seto(10,R1);

seto(32,R3);

seto(31,R4);

};

};
```

Warto zwrócić uwagę na to, że cały kod wykonywany jest w obrębie jednej pętli co powoduje dużą oszczędność pamięci. Prezentowany przykład ma objętość zaledwie 767 bajtów.

Na następnych stronach jest pokazany schemat i kod układu rozbudowanego do ośmiu wyjść sterujących, takie rozwiązanie zawiera 1815 bajtów, wynika z tego, że dodanie pojedynczego wyjścia zajmuje niecałe 200 bajtów pamięci, a więc w NeoGSM-IP-64 można w taką sekwencję połączyć ok. 18 – 19 wyjść z których każde posiada sterowanie zarówno indywidualne jak i sekwencyjne i ma osobno ustawiony czas pracy. Przy założeniu, że jedno wyjście może obsługiwać po kilka zaworów jest to więc rozwiązanie nadające się do obsługi instalacji składającej się z kilkudziesięciu zakończeń.

Kod dla ośmiu wyjść:

int E1;

int E10;

int E2;

int E3;

int E4;

int E5;

int E6;

int E7;

int E8;

int E9;

int I6;

int I8;

int M1 ;

int M10;

int M11 ;

int M12 ;

int M13 ;

int M14;

int M15 ;

int M16 ;

int M17;

int M18 ;

int M19 ;

int M2;

int M20;

int M21 ;

int M22 ;

int M23;

int M24 ;

int M25 ;

int M26;

int M27 ;

int M28;
int M29;
int M3;
int M30 ;
int M31 ;
int M32;
int M4;
int M5 ;
int M6;
int M7;
int M8 ;
int M9 ;
int O10;
int O11;
int O12;
int O13;
int O14;
int O15;
int O16;
int O31;
int O32;
int O9;
int R1;
int R10;
int R11;
int R12;
int R13;
int R2;
int R3;
int R4;
int R5;
int R6;
int R7;
int R8;

```
int R9;
main()
{
    while(1)
    {
        O9=geto(9);
        I6=geti(6);
        O10=geto(10);
        O11=geto(11);
        O12=geto(12);
        O13=geto(13);
        O14=geto(14);
        O15=geto(15);
        O16=geto(16);
        O32=geto(32);
        O31=geto(31);
        I8=geti(8);
        M1=I6&I8;
        M5=O9&M7;
        M6=M5|R8;
        M2=R8|R7|R6|R5;
        M3=M2|R4|R3|R2;
        M4=M3|R1;
        R9=Isw(8,O32,0,0,0,1,1);
        R11=tofd(10,M4,0,1,1);
        M31=M32&R11;
        M7=!R11;
        M30=R9&M7;
        M27=O16&M7;
        M28=M27|R1;
        M24=O15&M7;
        M26=M24|R2;
        M22=O14&M7;
        M23=M22|R3;
```


M18=O13&M7;
M20=R4|M18;
M15=O12&M7;
M17=M15|R5;
M13=O11&M7;
M14=M13|R6;
M9=O10&M7;
M10=R7|M9;
R12=pons(11,M1,0,2,1);
R8=lsw(7,M6,R13,0,R10,1,1);
M32=M31|R13;
R13=pons(12,E1,0,1,1);
M25=E2&M31;
R1=lsw(0,M28,M25,0,R10,12,1);
M21=E3&M31;
R2=lsw(1,M26,M21,0,R10,3,1);
E2=edge(1,0,R2);
M19=E4&M31;
R3=lsw(2,M23,M19,0,R10,10,1);
E3=edge(2,0,R3);
M16=E5&M31;
R4=lsw(3,M20,M16,0,R10,4,1);
E4=edge(3,0,R4);
M12=E6&M31;
R5=lsw(4,M17,M12,0,R10,7,1);
E5=edge(4,0,R5);
M11=E7&M31;
R6=lsw(5,M14,M11,0,R10,2,1);
E6=edge(5,0,R6);
E8=edge(7,0,R8);
M8=E8&M31;
R7=lsw(6,M10,M8,0,R10,15,1);
E7=edge(6,0,R7);
E9=edge(8,0,R12);

```
M29=M30|E9;  
E1=edge(0,1,M29);  
E10=edge(9,1,R12);  
R10=lsw(9,O31,E10,0,0,1,1);  
seto(9,R8);  
seto(10,R7);  
seto(11,R6);  
seto(12,R5);  
seto(13,R4);  
seto(14,R3);  
seto(15,R2);  
seto(16,R1);  
seto(32,R9);  
seto(31,R10);  
};  
};
```

Historia wersji dokumentu:

| Nr wersji: | Data publikacji: | Zmiany: |
|------------|------------------|---|
| 1.0 | 17-08-2020 | Pierwsza wersja do publikacji |
| 1.1 | 19-08-2020 | Poprawa widoczności opisów screenów. Drobne zmiany merytoryczne w opisie konfiguracji wejścia. Drobne poprawki edycyjne. |

Uwagi.

Firma Ropam Elektronik z zasady ukierunkowana jest na innowacyjność i rozwój swoich rozwiązań dlatego zastrzega sobie prawo wprowadzania bez uprzedzenia zmian parametrów technicznych, wyposażenia i specyfikacji oferowanych towarów.

Ropam Elektronik nie ponosi odpowiedzialności za działanie aplikacji lub produktu w określonym wdrożeniu u Klienta. Integrator lub projektant jest odpowiedzialny za swoje produkty i aplikacje wykorzystujące elementy Ropam Elektronik. Informacje, dokumenty czy oprogramowanie, które można przeglądać lub pobrać z serwisu Ropam Elektronik są "tak jak są" („as is”) bez jakiegokolwiek gwarancji bezpośredniej lub domyślnej przydatności do wykorzystania, wdrożenia, zastosowania.

Wszystkie użyte nazwy, znaki towarowe i handlowe są własnością stosownych podmiotów i zostały użyte wyłącznie w celach informacyjnych oraz identyfikacyjnych.