TECHBASE
AUTOMATYKA ELEKTRONIKA INFORMATYKA PRZEMYSŁOWA

Seria urządzeń
iMOD

Instrukcja iMod + PLC

© TechBase S.A. - kontakt: info@a2s.pl | www.a2s.pl

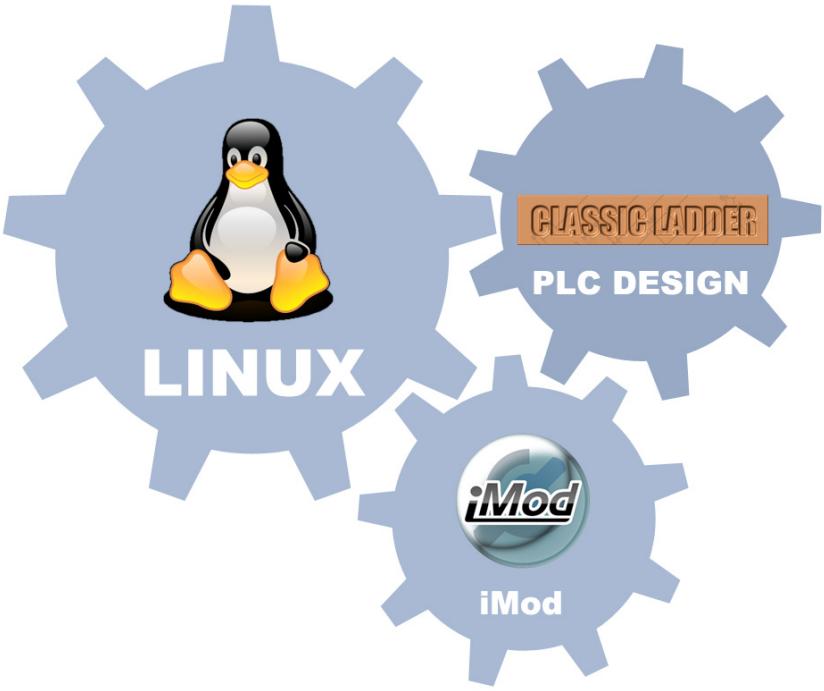
Spis treści:

- Rozdział 1. Wstęp
- Rozdział 2. Przygotowanie środowiska pracy
  - System Linux na dowolnym systemie (virtual-box + image)
  - Aplikacja Ladder Design
    - Konfiguracja programu
  - iMod
- Rozdział 3. Przykładowa konfiguracja
  - 1. Zliczanie zmian statusu diody USER_LED
    - Budowa i konfiguracja LadderClassic
      - Opis programu
      - Konfiguracja
    - Budowa i konfiguracja iMod
      - Definicja kanałów
      - Definicja parametrów
    - Weryfikacja przykładu
      - Ustanowienie połączenia
      - Weryfikacja przykładu
  - 2. Powiadomienie e-mail przy 10 krotnej zmianie statusu diody USER_LED
    - Budowa i konfiguracja Classic Ladder
      - Opis programu
      - Konfiguracja
    - Budowa konfiguracji iMod
      - Definicja kanałów
      - Definicja parametrów
    - Weryfikacja przykładu
      - Ustanowienie połączenia
      - Weryfikacja przykładu
    - Dodatek A: Opis bloków funkcyjnych programu Classic Ladder

Rozdział 1. Wstęp

iMod działający na platformie NPE ma możliwość pracy jako sterownik PLC. W tym celu poza wewnętrznym plikiem konfiguracyjny, należy wykorzystać aplikację ClassicLadder służącą do przekonwertowania języku drabinkowego LD na język C.

Rozdział 2. Przygotowanie środowiska pracy



Do uzyskania funkcjonalności PLC na iModzie potrzebny jest:

1. System Linux lub program do obsługi wirtualnych maszyn np. Oracle VirtualBox wraz z obrazem Linux'a
2. Oprogramowanie ClassicLadderDesign
3. Najnowsza wersja oprogramowania iMod Engine na platformie NPE (więcej na ten temat w rozdziale poświęconym przygotowaniu aplikacji iMod)

Pliki projektów użytych w niniejszej instrukcji znajdują się na serwerze ftp:
adres: ftp.a2s.pl
login: npe_imod@ftp.a2s.pl
hasło: npe_1m0d

System Linux na dowolnym systemie (virtual-box + image)

Aby zbudować aplikację PLC potrzebny jest system Linux wspierający środowisko graficzne GTK+.

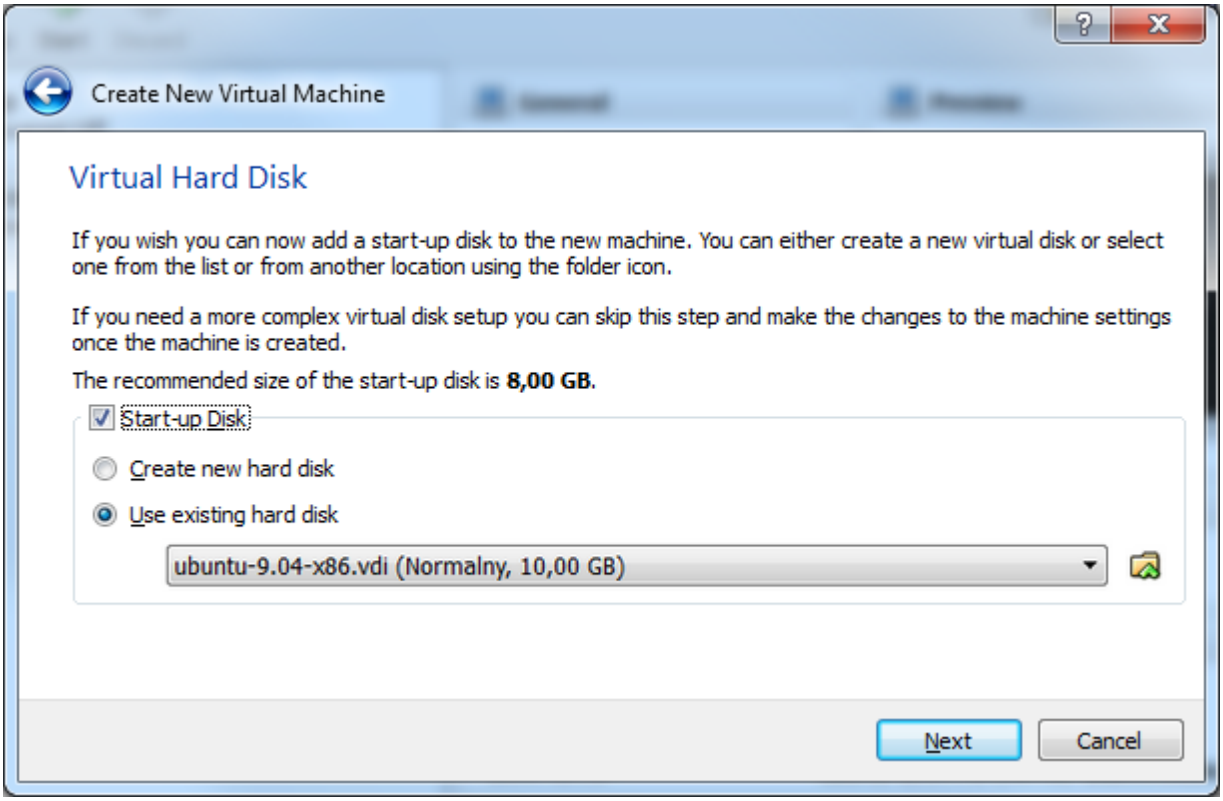
System Linux można uruchomić bezpośrednio z systemu Windows za pomocą wirtualnej maszyny.

Oprogramowanie VirtualBox można pobrać ze strony:
https://www.virtualbox.org/

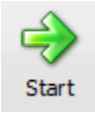
Obraz Linuxa można pobrać ze strony:
http://virtualboxes.org/images/ubuntu/

1. Uruchom program VirtualBox

2. Stwórz nową wirtualną maszynę poprzez kliknięcie przycisku Nowa
3. Nadaj nazwę wirtualnej maszynie, system operacyjny wybierz system Linux oraz odpowiednią wersję systemu
4. Przydziel min. ilość pamięci RAM dla maszyny wirtualnej (256 MB)
5. Wybierz opcję 'istniejącego dysku twardego' i wskaż plik z obrazem systemu Ubuntu

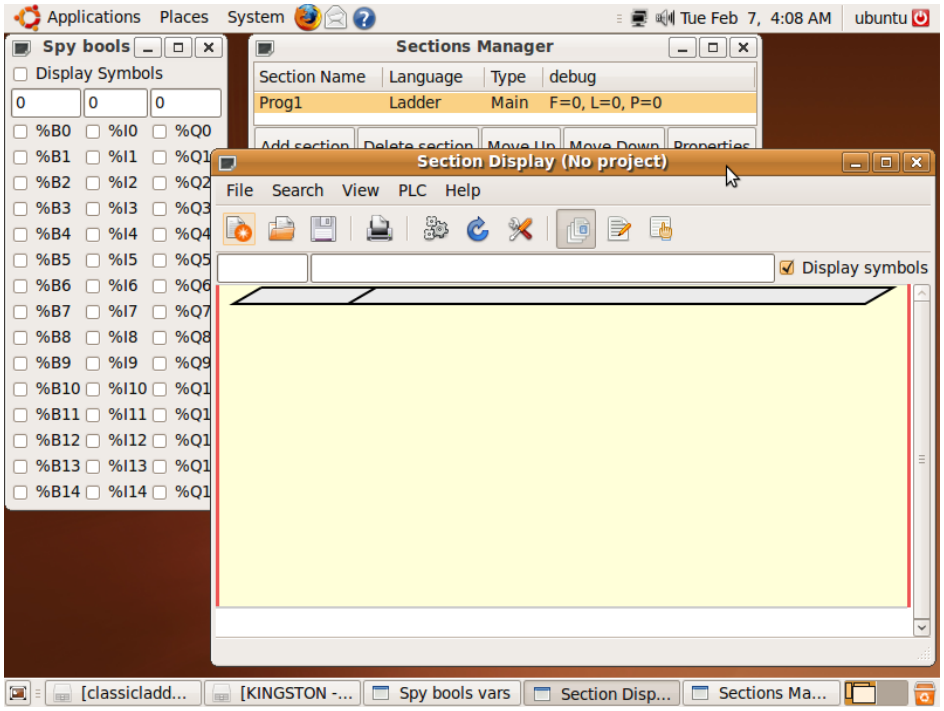


6. Po prawidłowym załadowaniu obrazu, pojawi się okno z parametrami wirtualnej maszyny. Kliknij prawym przyciskiem na dodaną wirtualną maszynę oraz wybierz opcję „Start”.



Hasło oraz login do Linuxa powinien być podawany w miejscu downloadu.
Dla wersji Ubuntu Linux 9.04:
login: ubuntu
password: reverse

Aplikacja Ladder Design



Na potrzeby tej instrukcji został wykorzystany program ClassicLadder.

Oprogramowanie Classic Ladder można pobrać ze strony:

http://sourceforge.net/projects/classicladder/

bądź z ftp.a2s.pl

W celu dodania nowych elementów w języku drabinkowym w aplikacji, uruchom okno edytora:

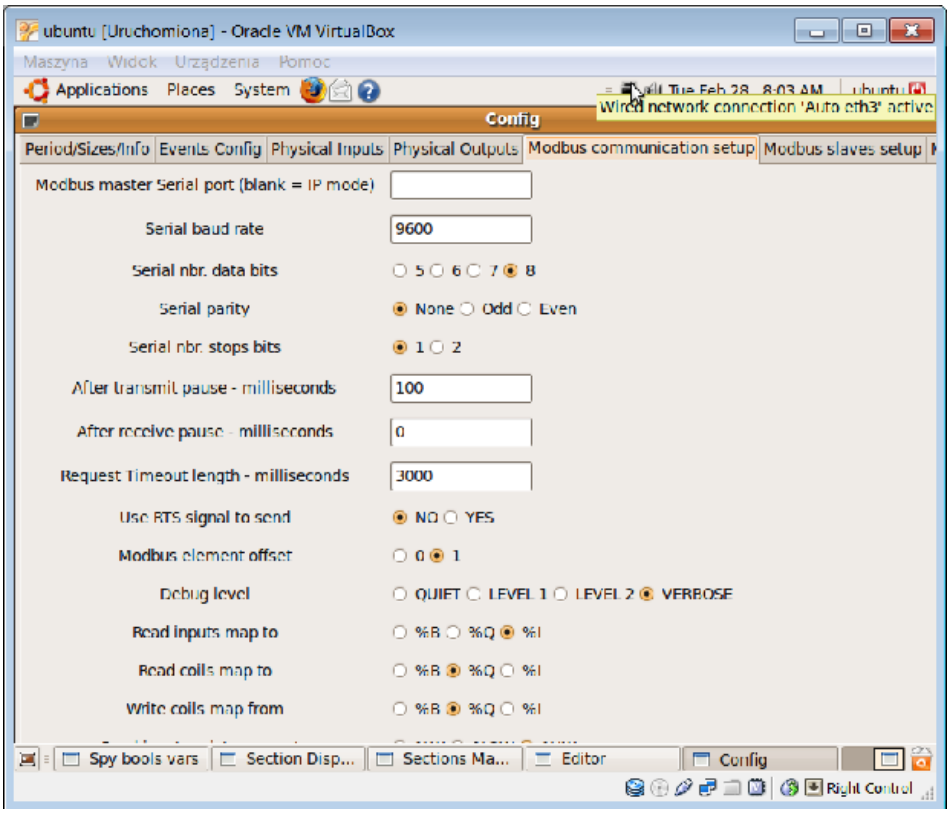
View-> Editor Window -> „Modify”

W oknie *Properties* możesz nadać adres zmiennej oraz ewentualnie pozostałe parametry bloku funkcyjnego.

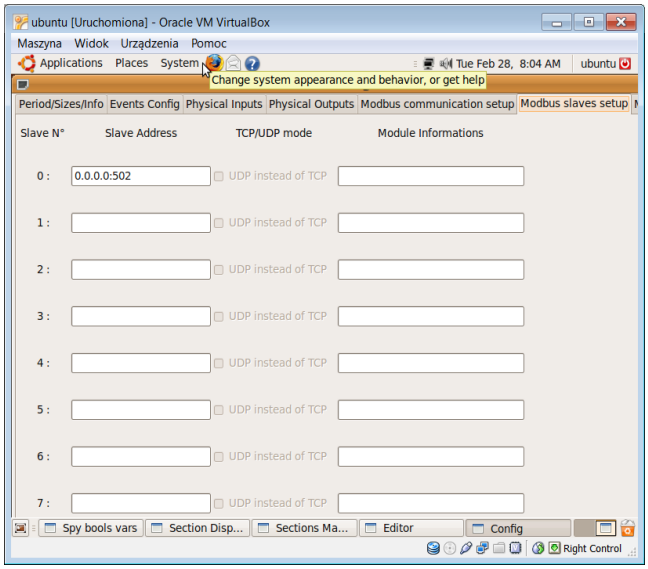
Zwróć uwagę że przed zapisem projektu, należy w oknie *Editor* zamknąć tryb edycji poprzez kliknięcie przycisku **OK**.

Konfiguracja programu

Aby wejść w ustawienia konfiguracji musisz przejść do PLC Configuration

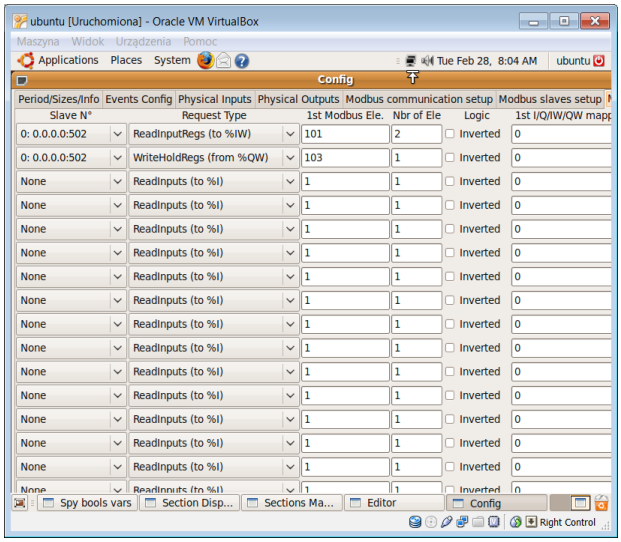


W zakładce *Modbus communication setup* ustaw parametry komunikacyjne z urządzeniem. W momencie zostawienia pustego pola *Modbus master Serial port*, komunikacja odbywać się będzie poprzez interfejs Ethernet.



W zakładce Modbus slaves setup podaj adres IP oraz port komunikacyjny dla odpowiednich urządzeń typu slave modbus. Zwróć uwagę na numer id urządzenia slave.

Adres IP 0.0.0.0 oznacza lokalny host



W zakładce Modbus I/O Setup podaj wszystkie parametry, wraz z adresami modbusowymi oraz typem żądania. Każdy z parametrów należy przypisać

odpowiedniemu urządzeniu slave.

Aby dokonać zapisu zmian parametrów konfiguracji, wystarczy zamknąć okno *Config*. Zmiany zostaną zapisane automatycznie.

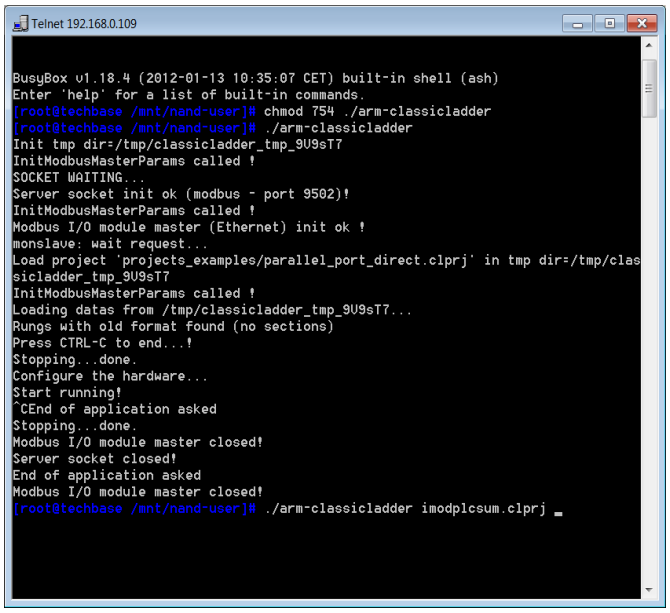
iMod

Przed wykonaniem kolejnych kroków upewnij się za pomocą aplikacji *softmgr* posiadasz najnowsze oprogramowanie.

Bezpłatne wsparcie techniczne dotyczy zawsze najnowszych stabilnych pakietów. Możesz dokonać aktualizacji iMod engine za pomocą polecenia:

softmgr update imod

W celu przejścia w tryb PLC, uruchom na iModzie projekt ClassicLadder poprzez cross skompilowaną aplikację arm-classicladder dostępną na serwerze ftp.a2s.pl (login i hasło na początku rozdziału 2.)



1. Skopiuj w trybie binarnym pliki *arm-classicladder* oraz *projekt.clprj* (projekt

wcześniej stworzonej aplikacji PLC) do katalogu:
/mnt/nand-user/

2. Nadaj odpowiednie uprawnienia dla pliku arm-classicladder poprzez komendę
chmod 754 /mnt/nand-user/arm-classicladder
3. Uruchom aplikację *arm-classicladder*
/mnt/nand-user/arm-classicladder
4. Uruchom projekt poprzez aplikację *arm-classicladder*
/mnt/nand-user/arm-classicladder projekt.clprj

Zakończenie działania aplikacji PLC odbywa się poprzez przerwanie programu, kombinacją klawiszy CTRL+C, bądź przy restarcie iModa.

Przy restarcie iModa aplikacja PLC zostanie przerwana, dlatego wskazane jest dodanie skryptu startowego do pliku /mnt/mtd/rcs co spowoduje auto uruchomienie skryptu po restarcie urządzenia.

Aby uruchomić lokalnie aplikację, należy przejść do menu *PLC -> RUN*.
Przy pomocy okna *Spy bools var* (*View -> Bools var window*) oraz okna *Spy free vars* (*View -> Free var window*) można sterować stanami oraz wartościami parametrów.

Rozdział 3. Przykładowa konfiguracja

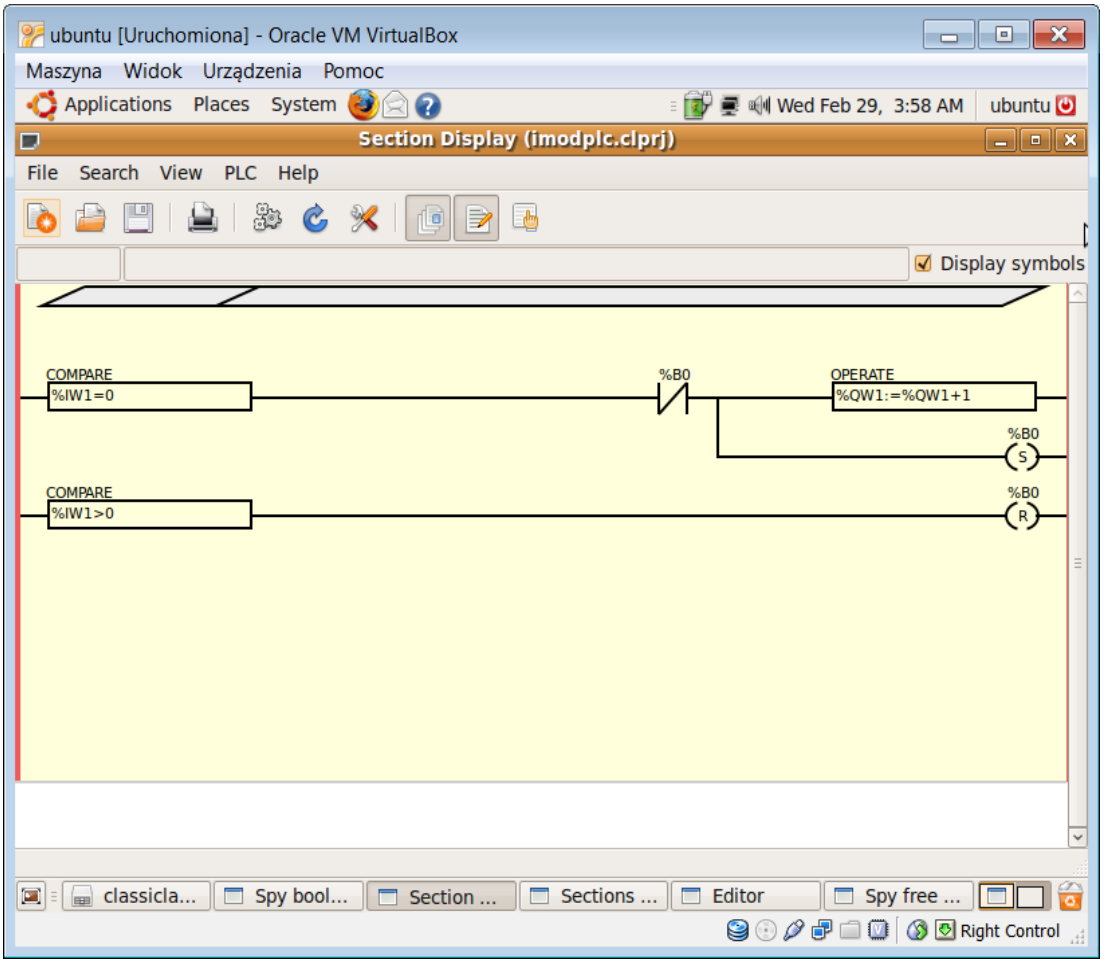
W przykładowej konfiguracji modułu telemetrycznego iMod w trybie pracy jako sterownik PLC zostaną przedstawione dwa przykłady:

1. zliczenia zmian statusu diody USER_LED na stan wysoki
2. dla liczby 10 zmian statusu diosy USER_LED zostanie wysłany e-mail.

1. Zliczanie zmian statusu diody USER_LED

W tym przykładzie, aplikacja będzie zliczała ilość zmian stanów diody USER_LED i zapisywała tą ilość do rejestru w iModzie.

Budowa i konfiguracja LadderClassic



W tej aplikacji wykorzystywane są dwie zmienne główne:

- IW1 – sczytywanie stanu diody USER_LED
- QW1 – zapisywanie ilości zmian

oraz jedna zmienna pomocnicza:

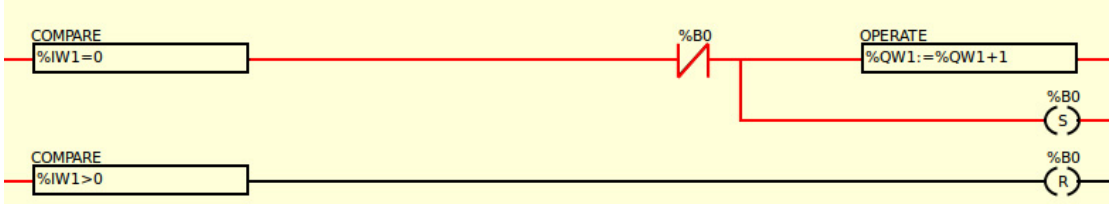
- B0

Wykorzystywane są również bloki funkcyjne:

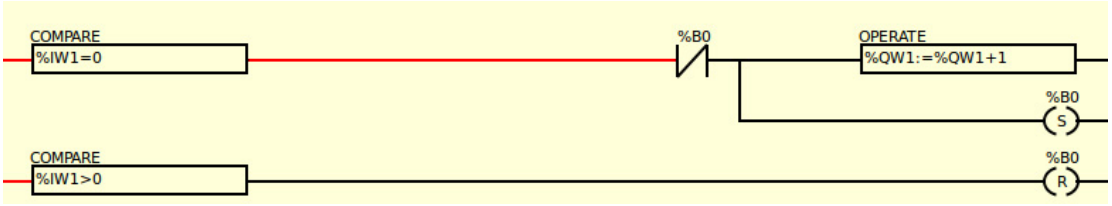
- Variable Comparison
- Variable Assignment
- Set Output
- Reset Output
- N.C. Input

Opis programu

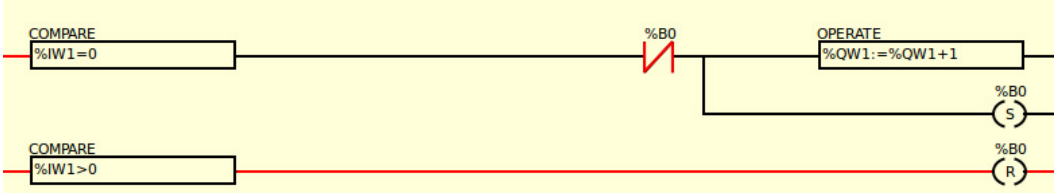
iMod domyślnie ustawione ma zapamiętywanie stanów I/O dlatego aplikacja została ustawiona na zliczanie zmian stanu diody USER_LED w momencie, kiedy dioda przyjmować będzie stan 0 (dioda wyłączona).



- W momencie zmiany stanu diody USER_LED na 0, sygnał zostanie puszczony przy pomocy bloku funkcyjnego COMPARE %IW1=0 poprzez styk normalnie zamknięty ze zmienną dodatkową B0 i dotrze w pierwszej kolejności do bloku funkcyjnego OPERATE %QW1:=%QW1+1 a zaraz potem do cewki SET ze zmienną B0.



- Dzięki bloku funkcyjnemu OPERATE, zmienna QW1 wzrośnie o jeden. Cewka SET powoduje zmianę stanu zmiennej B0, co spowoduje rozłączenie cewki normalnie zamkniętej z tą zmienną, dzięki czemu sygnał nie będzie docierał do bloku funkcyjnego OPERATE. Jest to zabezpieczenie, aby parametr QW1 został zwiększony wyłącznie o jeden punkt.



- W momencie zmiany stanu diody USER_LED na 1, zmienna B0 zostanie zresetowana przy pomocy bloku funkcyjnego %IW1>0, dzięki czemu cewka normalnie zamknięta zostanie ponownie zamknięta i przy kolejnej zmianie

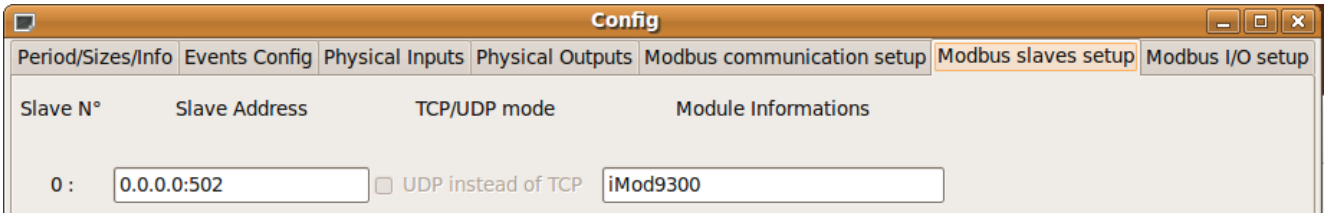
stanu diody USER_LED na 0 będzie przepuszczać sygnał.

Konfiguracja

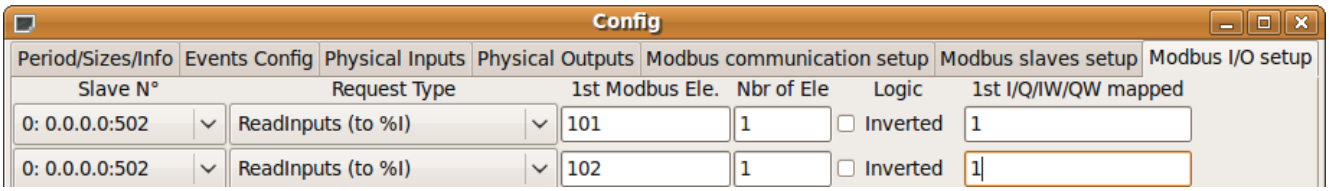
Konfiguracja komunikacji modbusowej
(*PLC Configuration Modbus communication setup*):

Dla tego przykładu będziemy się łączyć z urządzeniem poprzez port Ethernet, dlatego Modbus Master Serial Port zostawiamy pusty, resztę ustawień pozostawiamy domyślnych.

Konfiguracja adresacji modbusowej
(*PLC Configuration Modbus slaves setup*):



Konfiguracja parametrów modbusowych
(PLC Configuration Modbus I/O setup):



Należy zwrócić uwagę na adresację modbusową. Często zdarza się tak że adres ulega przesunięciu o 1 w górę lub w dół. W tym przypadku, adresacja parametrów w ClassicLadder powinna być przesunięta o jeden w górę względem iModa.

Budowa i konfiguracja iMod

Konfiguracja iModa odbywa się wyłącznie poprzez plik MainConfig.xml. W pierwszej kolejności musisz odpowiednio skonfigurować kanały dostępu oraz adresację parametrów.

Definicja kanałów

Zgodnie z konfiguracją komunikacji modbusowej w programie ClassicLadder, kanał dostępu trzeba stworzyć pod adresem modbusowym urządzenia 0, poprzez Ethernet na porcie 502. Źródłem parametrów jest nasze urządzenie iMod.

```
<?xml version="1.0" encoding="UTF-8"?>

<imod version="1.1.0">

<group name="Definicje kanalow">

        <access-channel name="Modbus_S1">
                <protocol name="MODBUS" />
                <port>"ET-502-TCP"</port>
                <property name="device-id" value="0"/>
        </access-channel>

        <source-channel name="NPE_io">
                <protocol name="HARDWARE"/>
                <gap>0</gap>
                <cycle>2</cycle>
        </source-channel>

</group>

</imod>
```

Definicja parametrów

Tworzymy dwa parametry, jeden odpowiedzialny za sczytywanie wartości stanu diody USER_LED (parametr id 100), drugi w którym będzie zapisywana ilość zmian stanów (parametr id 101).

```
<imod version="1.1.0">
 <group name="Definicje kanalow">

        <parameter>
                <id>"100"</id>
                <description>"USER_LED"</description>
                <source-channel channel-name="NPE_io" parameter-id="USER_LED"/>
                <access-channel channel-name="Modbus_S1" parameter-id="100">
                        <property name="varspace" value="DISCRETE"/>
                </access-channel>
        </parameter>

        <parameter>
                <id>"counter"</id>
                <access-channel channel-name="Modbus_S1" parameter-id="101"/>
        </parameter>

</group>

</imod>
```

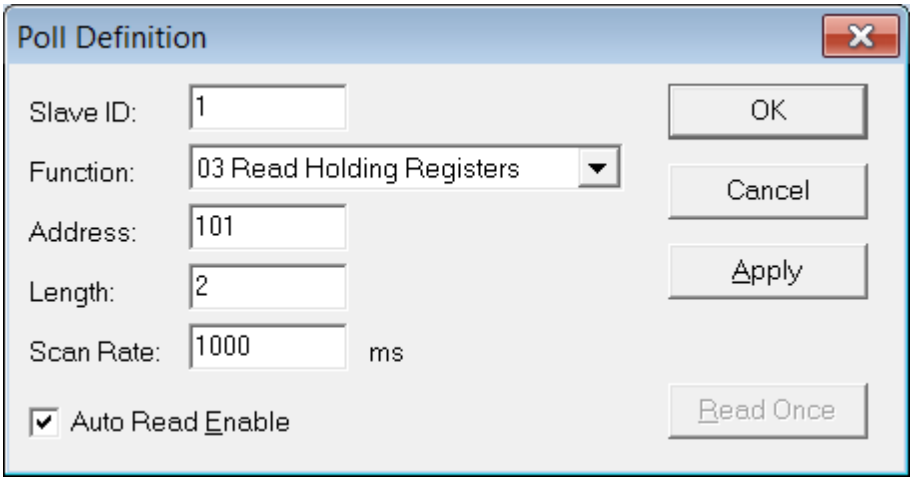
Weryfikacja przykładu

Do poprawnej weryfikacji przykładu zostanie użyta aplikacja typu modbus master - Modbus Poll

Aplikacja Modbus Poll jest dostępna w internecie pod adresem:
http://www.modbustools.com/modbus_poll.asp

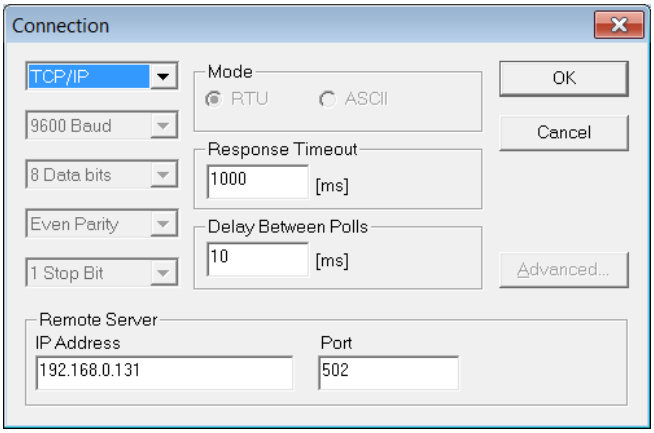
Dodatkowo należy wgrać plik konfiguracyjny omawiany wcześniej oraz uruchomić aplikację *arm-classicladder* wraz z projektem wygenerowanym przez ClassicLadder.

Ustanowienie połączenia



W pierwszej kolejności należy ustawić parametry które będziemy odbywać za pomocą aplikacji Modbus Poll. W tym celu przechodzimy do okna Poll Definition (*Setup -> Poll Definition*).

W przykładzie tym odczytujemy urządzenie Slave jakim jest iMod. Jego adres modbusowy to 1, wykorzystujemy kod funkcji modbusowej 03, sczytujemy rejestry o adresie 100, 101 oraz 102 dlatego zaczynamy sczytywać od 101 i 3 adresy (występuje tutaj przesunięcie modbusowe).

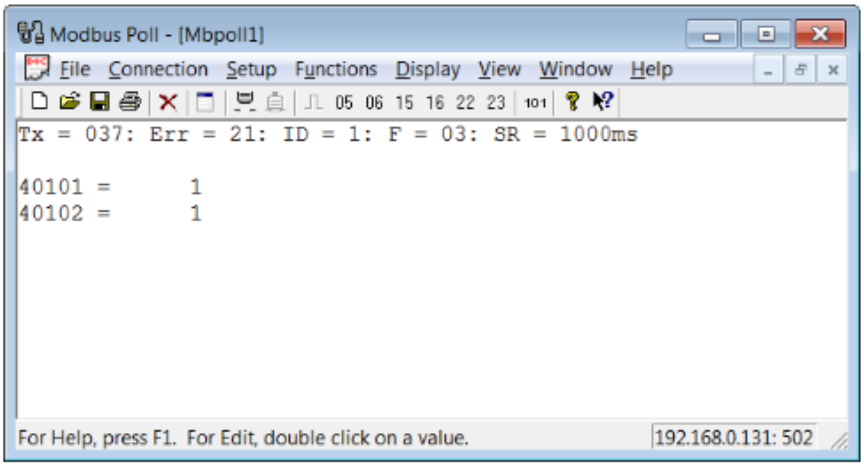


Kolejnym krokiem jest połączenie się z iMod'em (Connection -> Connect).

Zgodnie z plikiem konfiguracyjnym, łączymy się poprzez TCP/IP i port 502.

Domyślne IP modułu telemetrycznego iMod to:
192.168.0.101

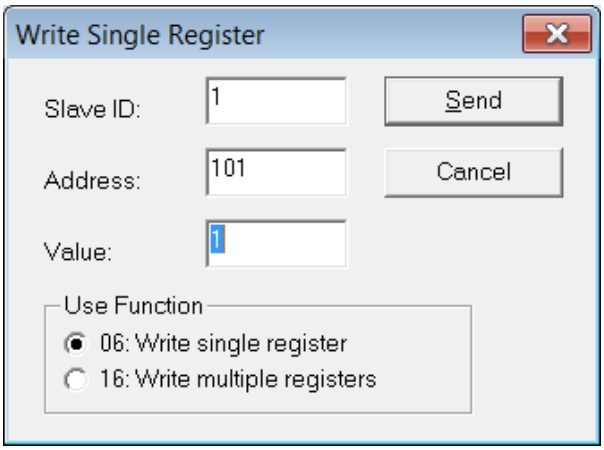
Weryfikacja przykładu



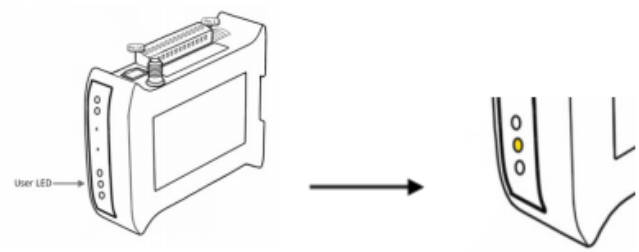
Po poprawnym połączeniu powinno wyświetlić się okno jak poniżej (w zależności od wersji Modbus Polla okno może się różnić).

Opis parametrów w oknie Modbus Poll:

Adres	Opis
101	Dioda USER_LED
102	Licznik - podaje ilość zmian statusu diody

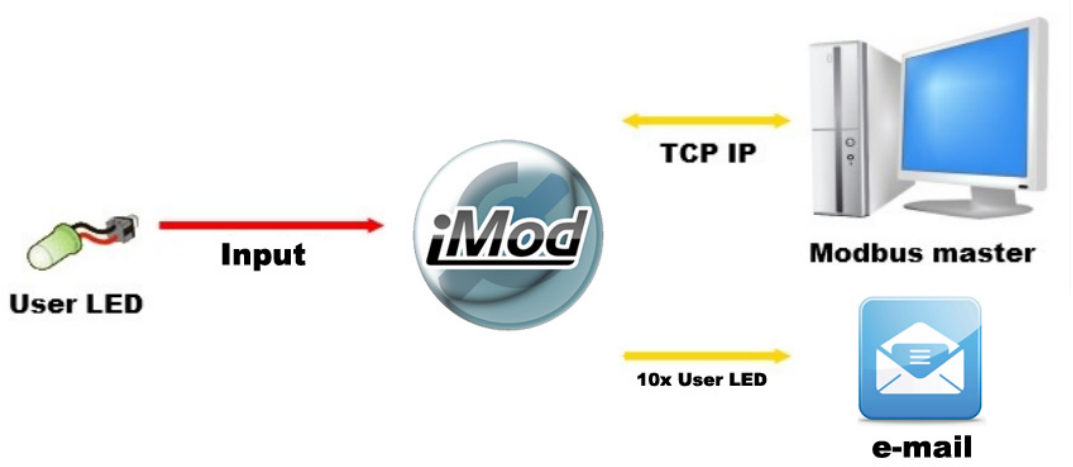


Aby zmienić wartość wybranego parametru należy dwukrotnie kliknąć w jego adres lub wartość wyświetlaną.



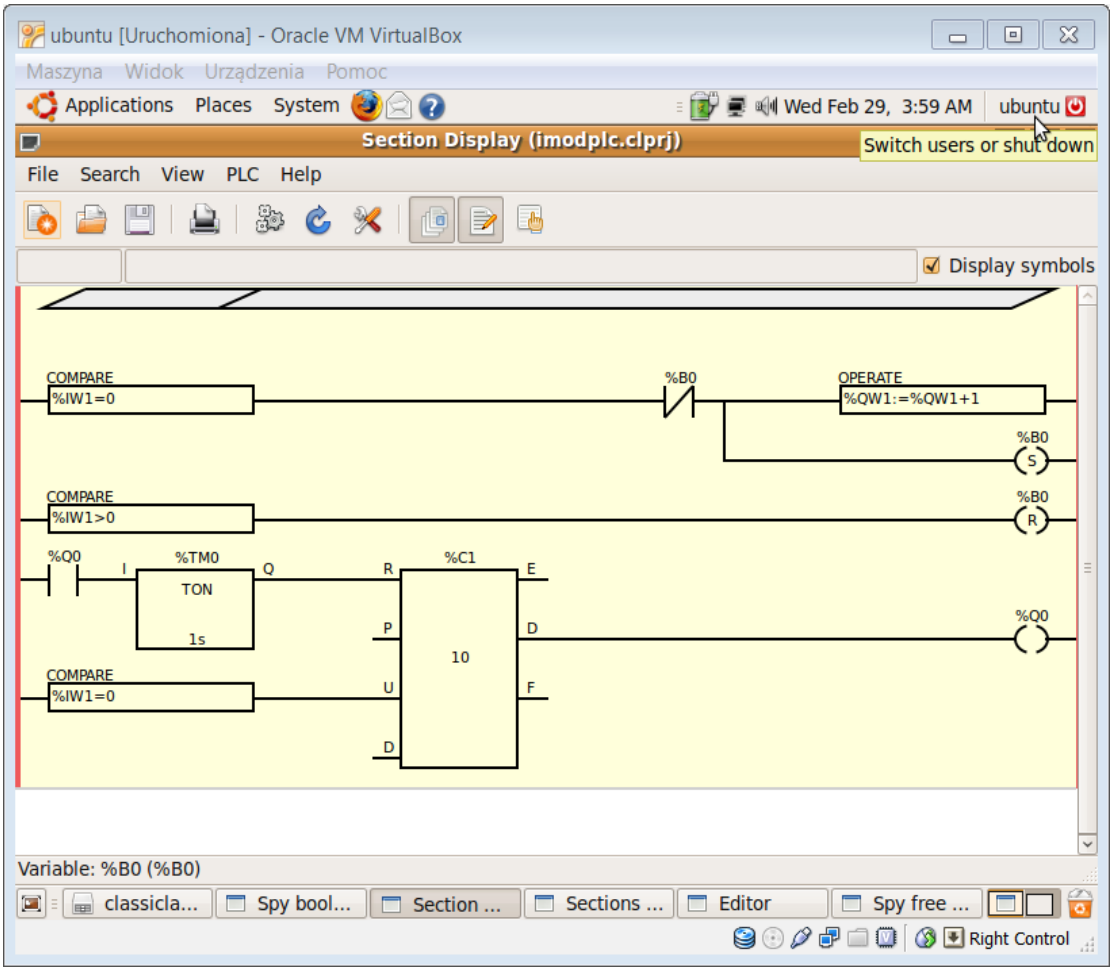
Aby zmienić wartość parametru, należy wpisać ją w okno Value następnie wysłać do urządzenia za pomocą przycisku Send. Stan diody USER_LED na urządzeniu powinien się zmienić, a w wartość parametru 102 powinna wzrosnąć o jeden punkt.

2. Powiadomienie e-mail przy 10 krotnej zmianie statusu diody USER_LED



Aplikacja będzie wysłać wiadomość e-mail na wskazany adres w pliku konfiguracyjnym w momencie kiedy wartość zmian stanu diody USER_LED przekroczy 10. W momencie wysłania e-mail, ilość zmian stanu diody zostaje wyzerowany i naliczanie odbywa się od początku.

Budowa i konfiguracja Classic Ladder

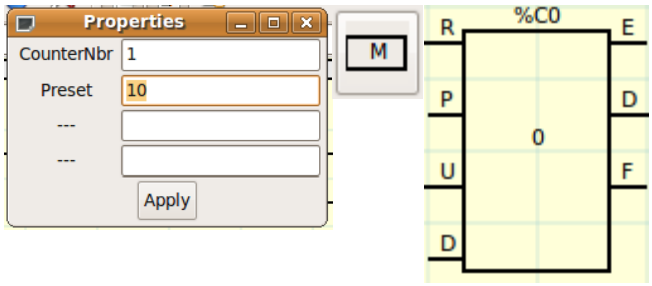


Aplikacja ta jest rozszerzoną wersją aplikacji z poprzedniego przykładu dlatego wszystkie parametry są takie same. Dodatkowo poza zmiennymi które zostały przedstawione w poprzednim przykładzie, występuje dodatkowa zmienna:

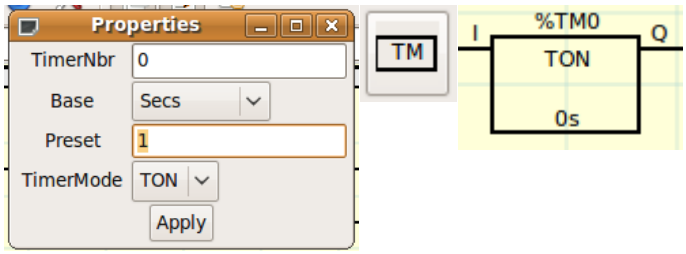
- Q0 – zmienna odpowiadająca za wysłanie e-maila

Opis programu

Program ten zlicza ilość zmian statusu diody USER_LED, co 10 zmian stanów diody na stan , następuje powiadomienie e-mailem wcześniej zdefiniowanego odbiorcy.



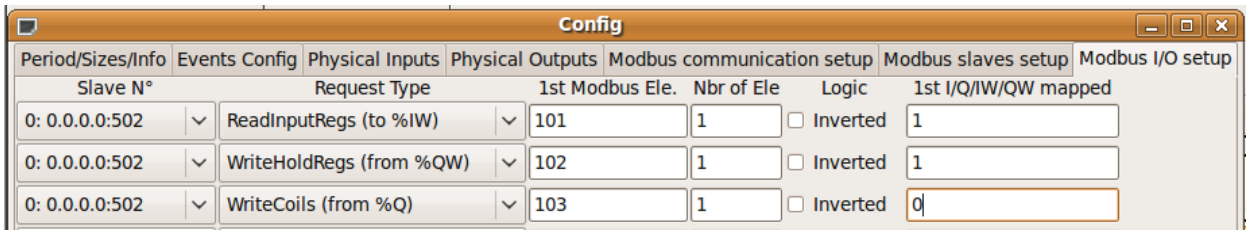
Wykorzystywany jest tutaj blok funkcyjny licznika (Counter Block) oraz timera (Timer IEC Block). Blok licznika został ustawiony na uruchomienie po i przesłanie sygnału do cewki %Q0 w momencie gdy liczba zliczeń będzie większa od 10.



1. W momencie gdy status USER_LED jest równy zero, sygnał dochodzi do licznika do złącza U, co powoduje wzrost liczby o 1 (należy zwrócić uwagę że jest to jednorazowa zmiana i nie trzeba stosować zabezpieczeń tak jak poprzednio).
2. W momencie gdy liczba zmian będzie wynosić 10, na wyjście D zostanie podany sygnał co uruchomi cewkę %Q0. Parametr Q0 odpowiada za wysłanie maila.
3. Uruchomienie stanu wysokiego resetuje również licznik, aby mógł zacząć zliczanie od początku. Jednak zanim licznik zostanie zresetowany, stan wysoki przechodzi do timera, który jest ustawiony na jedną sekundę.
4. Dopiero po upływie jednej sekundy licznik zostanie zresetowany. Należy go użyć gdyż zbyt szybki reset licznika, może spowodować iż iMod nie zdąży odczytać wartości parametru Q0 a co za tym idzie wysłać maila.

Konfiguracja

Konfiguracja komunikacji modbusowej oraz konfiguracja adresacji modbusowej pozostaje bez zmian.
Dla konfiguracji parametrów modbusowych należy dodać nową zmienną:
(*PLC Configuration Modbus I/O setup*)



Należy zwrócić uwagę na kolumnę 1st mapped – w tym przypadku ustawione jest zero dlatego że zmienna Q0 zaczyna się od liczby 0 a nie jak poprzednio zmienne zaczynały się od 1

Budowa konfiguracji iMod

Plik konfiguracyjny iMod jest rozszerzeniem pliku z poprzedniego przykładu. Do kanałów oraz parametrów należy dodać odpowiedni linie kodu.

Definicja kanałów

Do definicja kanałów należy dodać kanał powiadomienia, który będzie odpowiedzialny konfigurację maila oraz kanał treści wiadomości mailowej.

```
<message-channel name="Email_sender">
        <protocol name="EMAIL">
                <property name="user" value="testnpe"/>
                <property name="password" value="123npe"/>
        </protocol>
        <port>"poczta.o2.pl"</port>
        <recipient>"info@a2s.pl"</recipient>
    </message-channel>

 <message id="Mess_1">
        <![CDATA[
```

```
                    "USER_LED change state 10 times"
                    ]]>
        </message>
```

Definicja parametrów

Aby uzyskać funkcjonalność wysyłania wiadomości e-mail po 10 krotnym wysłaniu wiadomości e-mail, należy dodać parametr odpowiedzialny za wysłanie e-maila po zmianie stanu.

```
<parameter>
        <id>"10"</id>
        <access-channel channel-name="Modbus_S1" parameter-id="102">
                <property name="varspace" value="COIL"/>
        </access-channel>
        <event type="OnChange">
                <message-channel channel-name="Email_sender"/>
                <message-id>"Mess_1"</message-id>
        </event>
</parameter>
```

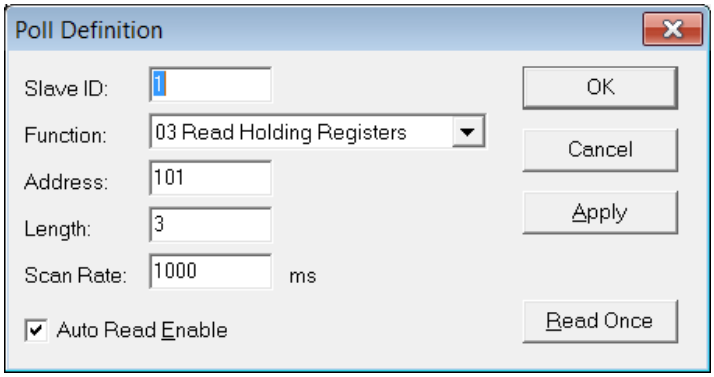
Weryfikacja przykładu

Do poprawnej weryfikacji przykładu zostanie użyta aplikacja typu modbus master - Modbus Poll

Aplikacja Modbus Poll jest dostępna w internecie pod adresem:
http://www.modbustools.com/modbus_poll.asp

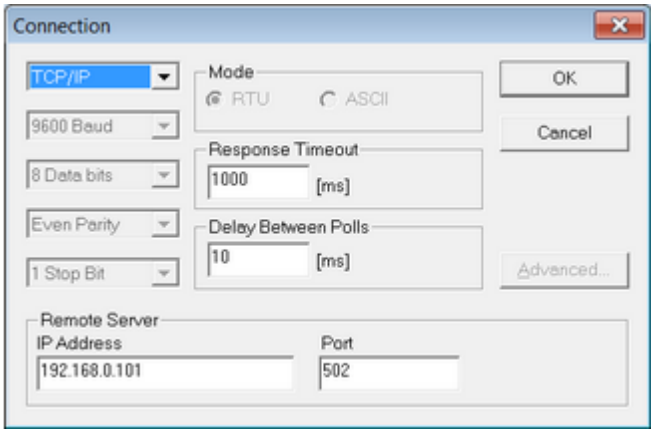
Dodatkowo należy wgrać plik konfiguracyjny omawiany wcześniej oraz uruchomić aplikację arm-classicladder wraz z projektem wygenerowanym przez ClassicLadder.

Ustanowienie połączenia



W pierwszej kolejności należy ustawić parametry które będziemy odbywać za pomocą aplikacji Modbus Poll. W tym celu przechodzimy do okna Poll Definition (*Setup -> Poll Definition*).

W przykładzie tym odczytujemy urządzenie Slave jakim jest iMod. Jego adres modbusowy to 0. Wykorzystujemy kod funkcji modbusowej 03. Adresy do sczytywania to rejestry o adresie 100, 101 oraz 102 dlatego zaczynij sczytywać od 101 i 3 adresy (występuje tutaj przesunięcie modbusowe).

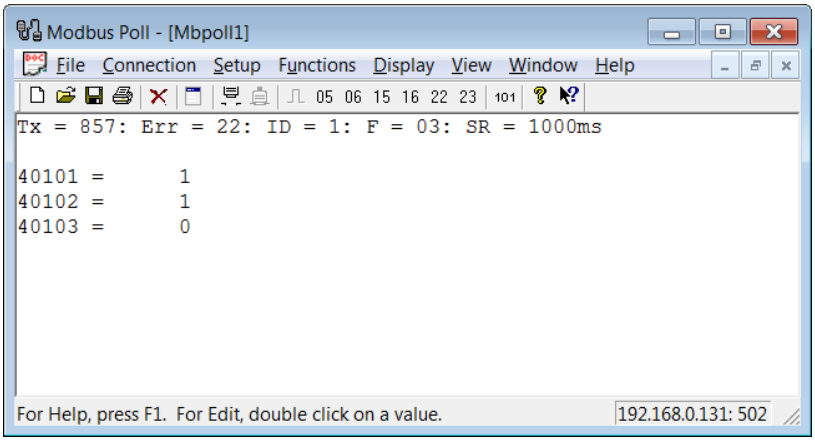


Kolejnym krokiem jest połączenie się z iMod'em (Connection -> Connect).

Zgodnie z plikiem konfiguracyjnym, łączymy się poprzez TCP/IP i port 502.

Domyślne IP modułu telemetrycznego iMod to:
192.168.0.101

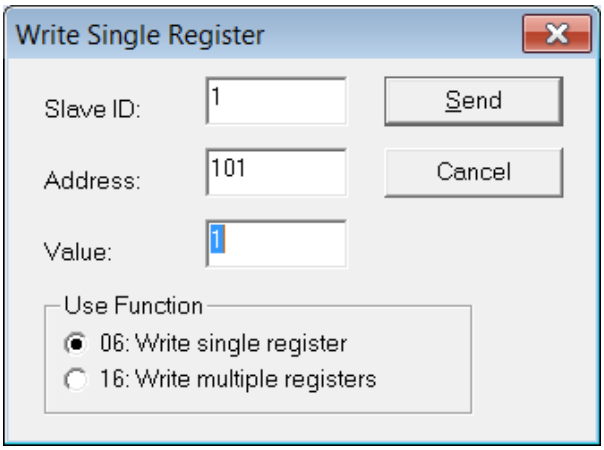
Weryfikacja przykładu



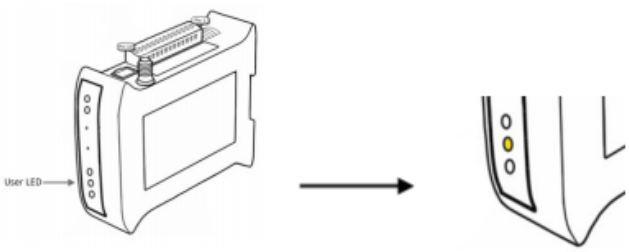
Po poprawnym połączeniu powinno wyświetlić się okno jak poniżej (w zależności od wersji Modbus Polla okno może się różnić).

Opis parametrów w oknie Modbus Poll:

Adres	Opis
101	Dioda USER_LED
102	Licznik - podaje ilość zmian statusu diody
103	Licznik - w momencie gdy parametr ten osiągnie wartość 10 zostanie wysłane powiadomienie mailowe oraz wartość zostaje zresetowana



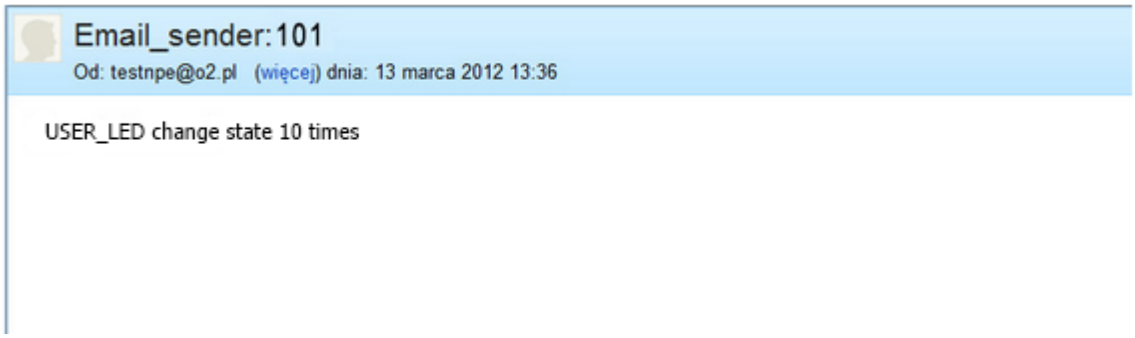
Aby zmienić wartość wybranego parametru należy dwukrotnie kliknąć w jego adres lub wartość wyświetlaną.



Aby zmienić wartość parametru, należy wpisać ją w okno Value następnie wysłać do urządzenia za pomocą przycisku Send. Stan diody USER_LED na urządzeniu powinien się zmienić, a w wartość parametrów 102 oraz 103 powinna wzrosnąć o jeden punkt.

Przy 10 krotnej zmianie stanu diody USER_LED aplikacja iMod wyśle wiadomość

mailową na zdefiniowany adres.



Dodatek A: Opis bloków funkcyjnych programu Classic Ladder

Symbol	Opis	Funkcja
	Styk normalnie otwarty	Przewodzi sygnał (zwiera styki), gdy wartość logiczna przypisanej zmiennej wynosi "1"
	Styk normalnie zamknięty	Styk normalnie zamknięty. Przewodzi sygnał (zwiera styki), gdy wartość logiczna przypisanej zmiennej wynosi "0"
	Styk z rosnącym zboczem	Styk ten wysyła impuls w momencie zmiany stanu z otwartego na zamknięty
	Styk z opadającym zboczem	Styk ten wysyła impuls w momencie zmiany stanu z zamkniętego na otwarty
	Linia łącząca pozioma	Linia pozioma łącząca elementy
	Linia łącząca pionowa	Linia pionowa łącząca elementy
	Linia łącząca pozioma do końca	Tworzy linie poziomą do końca linii
TM	Blok funkcyjny timera	Timer, do wyboru nazwa timera, wartość, stała, oraz tryb: • ON • OFF
C	Blok funkcyjny licznika	Licznik R- reset licznika P- ustawia na wartość zadaną aby na wyjściu było 1 U- zliczanie w górę D po lewej - zliczanie w dół D po prawej – wyjście, gdy wartość zliczonych równa się lub jest większa od wartości zadanej to stan 1
>	Blok funkcyjny porównania	Za pomocą tego bloku można użyć funkcji matematycznych porównania

T	Blok funkcyjny timera	Timer, zadany odlicza czas.
M	Blok monostabilny	Blok ten ustawia wartość wyjścia na „1” na określony czas, po czym wyjście uzyskuje stan „0”
	Cewka normalnie otwarta	Przekaźnik ustawia wartość przypisanej zmiennej na "1", gdy podany zostanie do niego sygnał. Jest to przekaźnik o stykach otwartych.
	Cewka normalnie zamknięta	Przekaźnik ustawia wartość przypisanej zmiennej na "0", gdy podany zostanie do niego sygnał. Jest to przekaźnik o stykach zwartych.
S	Cewka ustawiania	Przekaźnik ustawialny "SET". Przekaźnik ustawia na "1" zmienną.
R	Cewka resetu	Przekaźnik ustawialny "RESET" przystosowany do współpracy z przekaźnikiem "SET". Gdy do przekaźnika podany zostanie sygnał wartość przypisanej mu zmiennej ustawiona zostanie na "0".
J	Cewka przeskoku	Przeskok do wybranej linii
C	Cewka wezwania	Wezwanie wybranej linii
=	Działanie	Możliwość wykonywania działań matematycznych
	Wskaźnik	Wskaźnik