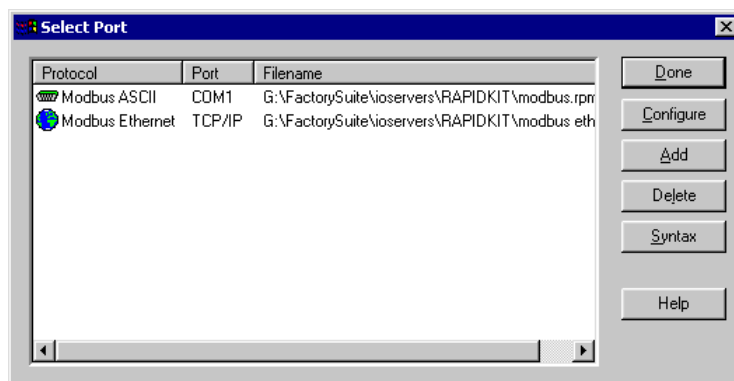


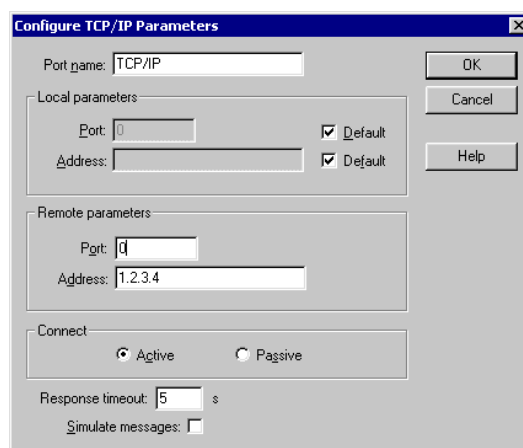
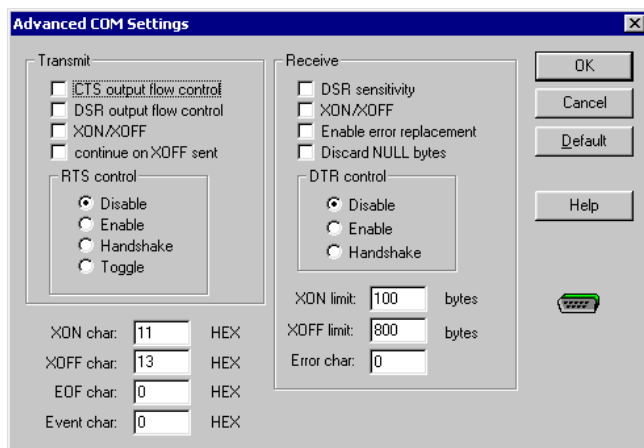
Rapid Protocol Modeler Kit

Na początku roku 2001 w ofercie produktów firmy Wonderware pojawił się produkt o nazwie Rapid Protocol Modeler Kit – w skrócie RapidKit. Jest to konfigurowalny program komunikacyjny, który umożliwia skonstruowanie własnego protokołu wymiany danych z urządzeniami komunikującymi się z komputerem PC za pomocą łącza RS232 lub Ethernet. W RapidKit można tworzyć i uruchamiać wiele protokołów jednocześnie co daje dosyć dużą elastyczność – można zastosować jeden program do wielu protokołów.



Proces tworzenia protokołu składa się z kilku części. Jedną z nich jest określenie czy komunikacja ma przebiegać po kablu szeregowym czy sieci Ethernet. Oprócz standardowej konfiguracji numeru portu, liczby bitów stopu, parzystości można także określić kilka bardziej zaawansowanych opcji.

Także konfiguracja łącza TCP/IP ma konfigurowalne parametry które umożliwiają podanie adresu i portu TCP/IP który ma zostać użyty do komunikacji.

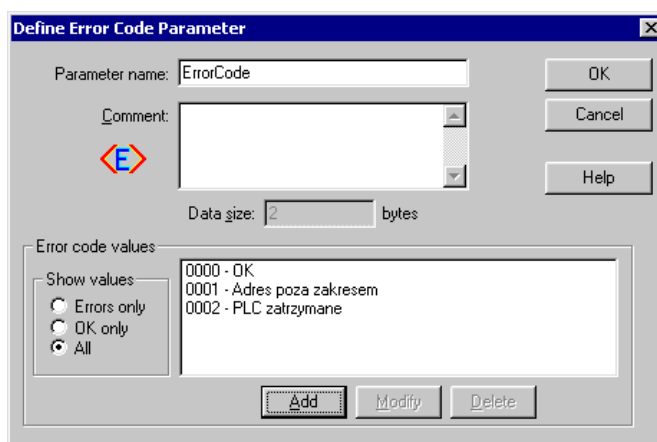


RapidKit ma także możliwość monitorowania danych przepływających przez port, podglądania aktywności poszczególnych topików, wartości zmiennych odczytywanych, statystyki uaktualnień i wiele innych. RapidKit DataMonitor umożliwia także zrzucanie do plików tekstowych i CSV danych które są w danej chwili aktywne.

Oprócz podziału komunikatów na binarne i ASCII w programie RapidKit można wyszczególnić typy takiej jak:

- Request – komunikaty wysyłane w celu pobrania wartości z urządzenia (zapytanie – odpowiedź).
- Poke – komunikaty wysyłane w celu zapisania wartości do urządzenia (zapisanie – potwierdzenie).
- Unsolicited – luźne komunikaty przychodzące z urządzenia.

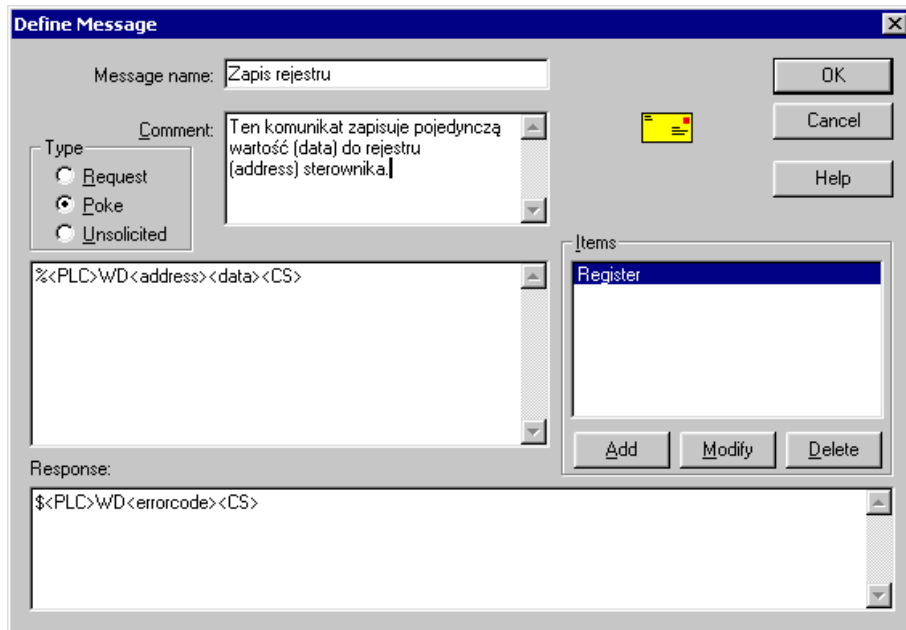
Komunikaty można parametryzować tak, aby zawierały np. adresy komórek pamięci do których chcemy mieć dostęp, adresy urządzeń do których się odwołujemy. Można także wstawiać pole sumy kontrolnej obliczanej zgodnie z kilkoma dostępnymi typami: XOR(LRC), CRC16, SUM(BCC) lub USER DLL czyli suma kontrolna użytkownika umieszczona w bibliotece DLL w funkcji o podanej w dokumentacji składni. W protokole można także zdefiniować kody błędów wraz z opisami co ułatwia diagnostykę problemów z komunikacją.



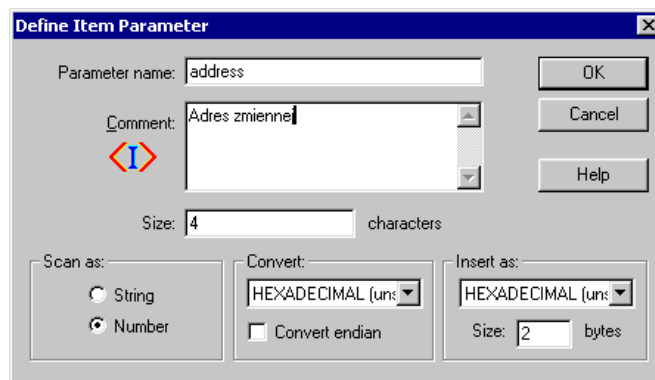
Jak w każdym programie komunikacyjnym także i w programie RapidKit można mieć dynamiczny dostęp do statystyk i pewnych parametrów komunikacyjnych driver'a. Można modyfikować takie parametry (*ang. ITEM*) jak:

- UPDATEINTERVAL - częstotliwość odpytywania danego tematu.
- MAXINTERVAL – najdłuższy czas odpytania jednego elementu (Item) w ramach danego tematu (Topic).
- STORESETTINGS – zapisanie zmodyfikowanego parametru UPDATEINTERVAL na stałe tak aby przy kolejnym uruchomieniu programu RapidKit został on wczytany automatycznie.
- ITEMCOUNT – liczba aktywnych zmiennych (Item) w ramach danego tematu.
- ERRORCOUNT – liczba aktywnych zmiennych z błędem.
- WRITECOMPLETE – stan procesu zapisu wartości do sterownika, które oczekują na wysłanie.
- READCOMPLETE – analogicznie do powyższego z tym, że dotyczy odczytów.
- POLLNOW – ustawienie na 1 wymusza odpytanie wszystkich zmiennych w ramach tematu.

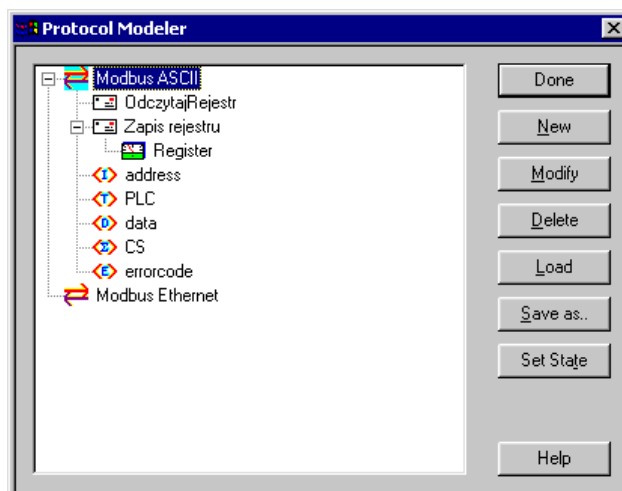
Oto przykład skonfigurowanego komunikatu zapisu wartości do sterownika.



Parametry komunikatu są oznaczone nawiasami <>. Po podwójnym kliknięciu na dany parametr np. <address> otwiera się okienko konfiguracyjne danego parametru. Można tutaj określić typ parametru, jego długość, typ danych itd.



A oto okno konfiguracyjne całego protokołu. Każdy protokół jest oddzielną gałęzią, w ramach protokołu konfigurujemy komunikaty, a w ramach komunikatów zmienne, które z ich pomocą odczytujemy. Oddzielnie konfigurujemy typy danych, które komunikaty mogą zawierać.



RapidKit ma opcję symulacji protokołu – aby zasymulować podłączone urządzenie należy do pliku simulate.txt wpisać odpowiednie ramki, które RapidKit ma traktować tak jakby przychodziły do niego od sterownika poprzez RS232 lub Ethernet. Jest to bardzo przydatna funkcjonalność ponieważ często faza projektowania protokołu odbywa się bez dostępu do urządzeń. Dzięki tej możliwości można przetestować protokół zanim przystąpimy do testów z urządzeniami. Rozdzielenie tych faz zmniejsza liczbę problemów przy pierwszych próbach uruchomienia protokołu z urządzeniem. Oto przykład pliku symulacyjnego Simulate.txt:

```
;data simulation
[ASCII test protocol]
data=BAFC

;poll response simulation
[COM1 ASCII test protocol]
readdata=\02S0155BAFCDD/03

;unsolicited message simulation
[COM1 ASCII test protocol]
msg1=\02S0155BAFCDD/03
receive=msg1
```

Pełny opis konfiguracji i zastosowania programu komunikacyjnego RapidKit można znaleźć w dołączonej do programu dokumentacji w pliku PDF. Wersję demo można otrzymać pobierając ją ze strony <http://www.wonderware.com> lub zwracając się do firmy Astor o przysłanie takiej wersji pocztą. Podsumowując ten krótki opis, produkt RapidKit można polecić wszystkim firmom które chciałyby szybko i bez zaangażowania programistów stworzyć dowolny protokół komunikacyjny z urządzeniami oparty na RS232 lub sieci Ethernet.

Porównując technologię tworzenia programu komunikacyjnego technikami programistycznymi z RapidKit'em można wymienić kilka jego zalet i wad.

Zalety to

- łatwość tworzenia protokołu,
- szybkość tworzenia protokołu,
- niski koszt (licencja Runtime zawarta w InTouch'u w wersji z IO),
- doświadczenie w programowaniu nie jest wymagane.

Do wad można zaliczyć:

- mała elastyczność – dotyczy głównie protokołów zawierających zależności czasowe między poszczególnymi komunikatami lub skomplikowane łańcuchy komunikatów,
- mniejsza wydajność.