1 Modbus TCP Master

1.1 `modbus-tcp-master`

Program `modbus-tcp-master` implementuje funkcjonalność Modbus TCP Master. Służy do połączenia się z jednym serwerem Modbus TCP Slave i okresowej wymiany skonfigurowanych bloków danych.

Program `modbus-tcp-master` obsługuje następujące polecenia protokołu Modbus:

Read Holding Registers (0x03) — odczyt rejestrów,

Read Input Registers (0x04) — odczyt wejść,

Write Multiple Registers (0x10) — zapis rejestrów.

Wywołanie:

```
$ modbus-tcp-master [-p port] -c config host
```

gdzie *port* to numer portu TCP (domyślnie 502), *config* jest plikiem konfiguracyjnym o składni opisanej w sekcji 1.2, a *host* nazwa hosta na którym działa serwer Modbus TCP Slave.

Przykład:

```
$ modbus-tcp-master -p 2000 192.168.0.3
```

1.2 Plik konfiguracyjny

Plik konfiguracyjny składa się z linii o następującej strukturze:

op file offset slave address length period

Dodatkowo każda linia zaczynająca się od znaku '#' traktowana jest jako komentarz.

op — Operacja, która ma być wykonana na danym bloku danych:

`read` — odczyt rejestrów,

`read-input` — odczyt wejść,

`write` — zapis rejestrów,

file — Plik służący do dostępu do danych, które mają być odczytane lub zapisane do zdalnego urządzenia Modbus Slave. Np. `/dev/mtd/data`.

offset — Offset w pliku *file*.

slave — Numer urządzania Slave, 1–247.

address — Adres początkowy rejestrów Modbus, 0–65535[1].

[1] są to adresy w Modbus PDU, w modelu danych Modbus bloki danych numerowane są od 1 do n, tzn. adres 0 to block danych 1

length — Ilość rejestrów do zapisu/odczytu.

period — okres aktualizacji w milisekundach.

Przykład:

```
# operation   file            offset  slave  address  length  period
read          /dev/mtd/data   0       1      0        1       1000
read          /dev/mtd/data   0       1      0        8       1000
read-input    /dev/mtd/data   0       1      8        1       1000
write         /etc/passwd     0       2      0        8       2000
```

Dane są przekazywane bezpośrednio w postaci jaka używana jest w protokole Modbus — słowa 16-bitowe w kolejności bajtów big-endian.

1.3 Działanie programu Modbus TCP Master

Głównym modułem programu implementującego funkcjonalność Modbus TCP Master jest plik `tcp-master.c` w pakiecie `modbus-utils`.

Program `modbus-tcp-master` po starcie przetwarza listę argumentów, a następnie czyta plik konfiguracyjny. Po inicjalizacji łączy się z wybranym serwerem Modbus TCP Slave, a następnie przechodzi do okresowego wysyłania poleceń do Slave zgodnie z plikiem konfiguracyjnym.

Wysyłanie poleceń dla każdego wpisu w pliku konfiguracyjnym odbywa się całkowicie niezależnie. Polecenie wysyłanie jest od razu jak upłynął skonfigurowany okres, ale tylko wtedy, gdy poprzednie polecenie dotyczące danego bloku danych się zakończyło.

1.4 `modbus-tcp-masterd`

Program `modbus-tcp-master` kończy pracę w przypadku błędu. By zapewnić ponowne uruchomienie używany jest skrypt `modbus-tcp-masterd`:

```
#!/bin/sh

while :; do
        modbus-tcp-master "$@"
        sleep 1
done < /dev/null > /dev/null 2>&1 &
```

2 Modbus TCP Slave

2.1 `modbus-tcp`

Program `modbus-tcp` implementuje funkcjonalność Modbus TCP Slave. Implementacja znajduje się w pliku `tcp-slave.c` w pakiecie `modbus-utils`.

`modbus-tcp` udostępnia zawartość wybranego pliku jako zestawu rejestrów Modbus, które klient Modbus TCP Master może dowolnie odczytywać i modyfikować.

Program `modbus-tcp` obsługuje następujące polecenia protokołu Modbus:

Read Holding Registers (0x03) — odczyt rejestrów,

Read Input Registers (0x04) — odczyt wejść,

Write Single Register (0x06) — zapis pojedynczego rejestru.

Write Multiple Registers (0x10) — zapis rejestrów.

Wywołanie:

```
$ modbus-tcp --emu [-l port] [-d device]
```

gdzie *port* to numer portu TCP pod którym dostępny jest Modbus TCP Slave (domyślnie 502), a *device* to plik który udostępniany jest jako dane urządzenia.

Przykład:

```
$ modbus-tcp --emu -l 3000 -d /dev/mtd/data
```

Za pomocą protokołu Modbus możliwy jest dostęp do odczytu i zapisu pierwszych 65536 16-bitowych słów wybranego pliku (128 KiB).