

R4ATNR

Instrukcja dla programisty systemu R4ATNR.



z@tkom

Prosimy uważnie przeczytać instrukcję!

Wersja 11.1.6C

Autor MG, ZC

Spis treści

Wstęp:.....	3
Co możemy uzyskać pisząc własny program?.....	3
Co należy wiedzieć przed rozpoczęciem programowania:.....	3
Instalacja urządzenia i programu.	3
Jak to działa:.....	4
Obsługa zdarzeń:.....	4
Zasoby:.....	5
Rejestry i znaczniki procesora:.....	5
Zestaw instrukcji:.....	6
Zestawienie instrukcji:.....	37
Przykładowy program.....	43
1. Migamy lampką.....	43
2. Migamy lampkami naprzemiennie.....	44
3. Program generowany przez edytor graficzny.....	44
4. Sterbox – najpierw migamy lampką w Sterboxie.....	46
5. Odbieramy komendy ze Sterboxa.....	47
6. Odbierz ze Sterboxa komendę wybrania numeru i odtworzenia komunikatu po złószeniu się rozmówcy.....	48
7. Prosta DISA.....	50
8. Automatyczna sekretarka.....	54
9. Nagrywarka plus sekretarka.....	56

R4ATNR

Wstęp:

Instrukcja ma charakter poglądowy i nie wyczerpuje w pełni tematu. Jest przeznaczona dla osób które zamierzają tworzyć własne aplikacje w oparciu o system R4ATNR. Firma Zetkom nie zapewnia wsparcia technicznego w zakresie pisania własnych aplikacji w ramach standardowej obsługi klienta, warunkiem uzyskania pomocy technicznej jest udział w szkoleniu organizowanym przez firmę „Zetkom” i uzyskanie odpowiedniego certyfikatu wraz z nadaniem numerem klienta.

Co możemy uzyskać pisząc własny program?

Urządzenie dostarczamy z graficznym edytorem który buduje program „Nagrywanie rozmów”, „DISA” czy też „IVR”. Możemy to prześledzić ustawiając parametry dla np. Nagrywania rozmów kanału 1, na zakładce *Podstawowe*, następnie klikając na *Zaawansowane*. W zakładkach klikamy na *Kanał 1* w okienku *Program główny (N)* widzimy utworzony przez edytor program. Wyjście poza możliwości edytora graficznego jest możliwe tylko poprzez modyfikację automatycznie utworzonego programu.

Natomiast utworzenie całkowicie nowego programu umożliwia stworzenie nowego urządzenia. Można tworzyć urządzenia do automatycznego ostrzegania telefonicznego lub głosem (np. o podwyższonej temperaturze w serwerowni, alarmie – do wielu osób), systemy sterowania przy pomocy DTMF z wyjściami przekaźnikowymi, pocztę głosową wysyłającą majle o nagraniu, dzwoniące po otrzymaniu majla itd.

Co należy wiedzieć przed rozpoczęciem programowania:

Zadaniem programisty oprócz zaprogramowania określonego algorytmu działania systemu jest oprogramowanie reakcji na wszelkie możliwe anomalie działania zewnętrznych systemów z którymi urządzenie R4ATNR współpracuje. Często przyczyną problemów są czasy reakcji zewnętrznych urządzeń współpracujących. W związku z tym musimy pamiętać o pewnych zasadach np. po zajęciu linii telefonicznej centrala potrzebuje chwili czasu po której będzie gotowa do odbierania numeru (tak więc zajmujemy linię, odczekujemy chwilę i wybieramy numer), przy połączeniach przychodzących musimy pamiętać o tym że, centrale potrzebują chwili czasu na otwarcie kanałów rozmównych w przeciwną stronę (tak więc przy odtwarzaniu zapowiedzi po odebraniu połączenia odczekajmy chwilę i dopiero odtwarzajmy zapowiedź). Podobne zasady obowiązują po innych interakcjach np. flash-u, przy sprawdzaniu sygnału zajętości itp.

Z kolei przy współpracy ze Sterbox musimy urządzeniom zapewnić komunikację poprzez LAN lub połączenie RS485.

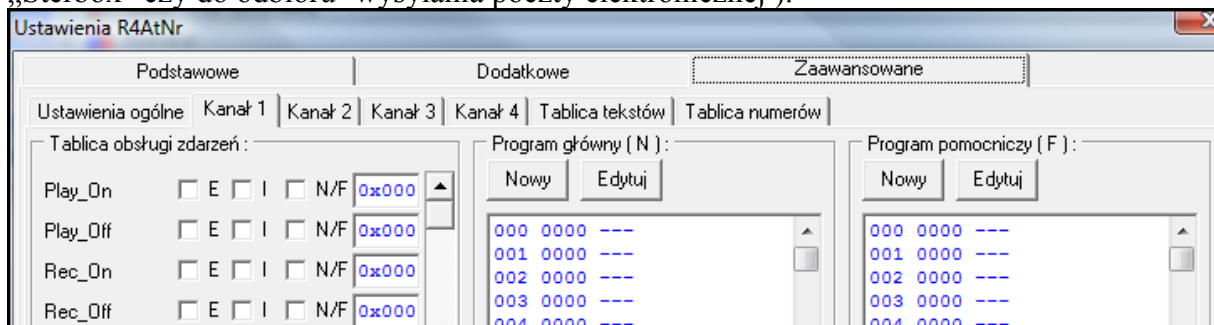
Instalacja urządzenia i programu.

Wszystkie ustawienia dostępne są w programie Tele32. Należy zainstalować go z pliku Setup.exe wybierając jako urządzenie obsługiwane nagrywarkę R4ATNR. Jako folder docelowy instalacji najlepiej wybrać jakiś katalog spoza Program files. Po zainstalowaniu programu podłączyć urządzenie poprzez USB. Sterowniki znajdują się w folderze do którego został zainstalowany program. Aby wejść w ustawienia należy zalogować się (fabrycznie użytkownik: instalator, hasło: 1234) i otworzyć *Konfiguracja, Ustawienia, R4ATNR*.

R4ATNR

Jak to działa:

Urządzenie R4ATNR jest wyposażone w cztery wirtualne procesory sterujące (dwa pierwsze są skojarzone z kanałami 1 i 2, procesory 3 i 4 w niektórych wykonaniach sprzętu są skojarzone z kanałami odpowiednio 3 i 4, w wersjach wyposażonych w dwa kanały procesory 3 i 4 można w ograniczonym zakresie również wykorzystywać np. do obsługi urządzeń „Sterbox” czy do odbioru- wysyłania poczty elektronicznej).

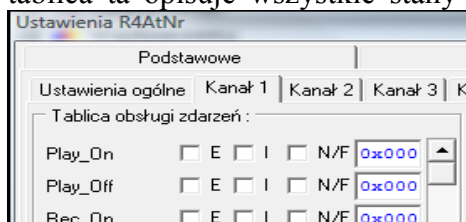


Rysunek 1. Otwarte okno z *Ustawieniami ATNR* na zakładce *Zaawansowane* , *Kanał 1*.

Procesory mogą działać w oparciu o ciągle wykonywanie programu sterującego, w oparciu o system zdarzeń lub w oparciu o obie te techniki jednocześnie. W panelu konfiguracji znajdują się dwa miejsca gdzie ustawiamy obsługę zdarzeń : pierwsze miejsce to zdarzenia kanałowe (na zakładkach kanałów-), a drugie miejsce to tablica tekstów. Każdy procesor posiada zestaw flag stanu oraz bank rejestrów uniwersalnych numerycznych jaki i tekstowych oraz zestaw rejestrów specjalnych (dla potrzeb generowania billingu i poczty elektronicznej). Dodatkowo każdy procesor ma dostęp do wspólnego banku tekstów (*Tablica tekstów*) i wartości numerycznych (*Tablica numerów*), do komunikacji między procesorowej mamy dostęp do zestawu markerów. Procesor może wykonywać program z dwóch przestrzeni adresowych głównej (*Program główny*) i dodatkowej (*Program pomocniczy*). Pomędzy przestrzeniami można się poruszać za pomocą skoków i podprogramów. Wykonywanie programu rozpoczyna się zawsze od programu głównego.

Obsługa zdarzeń:

Zdarzenia są generowane na podstawie tablicy zdarzeń przypisanej do każdego kanału, tablica ta opisuje wszystkie stany które można obsłużyć. Wpis do tablicy „E” określa czy



Rysunek 2. Tablica obsługi zdarzeń, fragment rysunku 1.

zdarzenie jest obsługiwane, czy nie. Wpis „I” określa czy obsługa zdarzenia może być przerwana przez obsługę innego zdarzenia. Wpis „N/F” określa czy zdarzenie jest obsługiwane przez program główny czy dodatkowy. Kolejny wpis określa adres obsługi zdarzenia w programie. Zdarzenia mogą być również generowane od systemu rozpoznawania tekstów (*Tablica tekstów*), dodatkowym parametrem w tablicy obsługi jest wtedy numer

R4ATNR

procesora do którego zostanie przekazane sterowanie. System rozpoznawania tekstów wyszukuje teksty w temacie maila przychodzącego lub w buforze odbiornika portu RS485. Obsługa zdarzenia powinna się zakończyć instrukcją **RET**, co spowoduje wznowienie aktualnego programu lub zatrzymanie procesora jeżeli program nie był uruchomiony.

Zasoby:

Ograniczenia. Program główny może mieć maksymalnie 1024 kroki, podobnie program dodatkowy. Mamy więc do dyspozycji 2048 kroki programu dla każdego procesora¹. Stos wywołań ma ograniczenie do 16 pozycji, po przekroczeniu pojemności stosu najstarszy zapis jest nadpisywany, kolejność wywołań stosu nie jest zaburzona. Kolejka wywołań zdarzeń ma głębokość 16 wywołań, przy przekroczeniu pojemności kolejki najstarszy wpis jest nadpisywany i nie zostanie wykonany, kolejność kolejki nie zostaje zaburzona.

Rejestry i znaczniki procesora:

Kursywą opisano indywidualne zasoby procesora.

Pogrubioną czcionką opisano zasoby wspólne dla procesorów.

- A) Znacznik prawda fałsz (F) - ustawiany za pomocą instrukcji TEST*
 - B) Znacznik pomocnicze (f0 do f7) - ustawiane za pomocą instrukcji TEST (większości), na znacznikach pomocniczych można wykonywać operacje logiczne które ustawiają stan flago głównej (F)*
 - C) Znacznik (N/F) - określa czy jest wykonywany program główny czy dodatkowy, znacznik nie jest dostępny bezpośrednio, jego zmiany są dokonywane za pomocą odpowiednich instrukcji skoków*
 - D) Znacznik (I) - określa czy aktualny program może być przerwany przez zdarzenie, jeżeli znacznik jest ustawiony program do czasu skasowania znacznika jest nieprzerywalny, a zdarzenia są dodawane do kolejki zdarzeń oczekujących i zostaną wykonane w kolejności zgłoszenia*
 - E) Rejestr numerów (num0 do num7) – uniwersalne rejestry numeryczne o rozmiarze 32 bitów ze znakiem*
 - F) Rejestry tekstów (txt0 do txt7) - uniwersalne rejestry tekstowe o pojemności max 64 znaków*
 - G) Timery (tim0 do tim7) - zestaw timerów do pomiarów czasów i do generowania opóźnień*
 - H) Rejestry specjalne generowania billingu (ustawiane specjalnymi rozkazami) :*
 - licznik dzwonków*
 - rejestr typu rozmowy*
 - rejestr numeru wybranego*
 - rejestr prezentacji*
 - rejestr nazwy numeru*
- I. Rejestry specjalne poczty elektronicznej (ustawiane specjalnymi rozkazami) :**

¹ Do wersji 11.4.1 włącznie, etykiety są osobne dla każdego z programów: głównego i pomocniczego. Poruszanie się pomiędzy nimi może następować jedynie na podstawie fizycznych adresów. Proponujemy ominięcie tego problemu poprzez zbudowanie na początku programu tablicy skoków.

R4ATNR

Działanie: Instrukcja powoduje przeskoczenie do określonej etykiety w programie, z programu głównego do dodatkowego lub odwrotnie

Znaczniki: Zmienia stan znacznika N/F na przeciwny

Przykład: **JMP_!** **procedura_zakonczenia**

Instrukcja: **JMP_T!** *etykieta_skoku*

Działanie: Instrukcja wykonuje przeskoczenie do określonej etykiety w programie gdy jest ustawiony znacznik **F** prawda, z programu głównego do dodatkowego lub odwrotnie

Znaczniki: Zmienia stan znacznika N/F na przeciwny

Przykład: **JMP_T!** **polecenie_systemu**

Instrukcja: **JMP_F!** *etykieta_skoku*

Działanie: Instrukcja wykonuje przeskoczenie do określonej etykiety w programie gdy jest skasowany znacznik **F** fałsz, z programu głównego do dodatkowego lub odwrotnie

Znaczniki: Zmienia stan znacznika N/F na przeciwny

Przykład: **JMP_F!** **analiza_numeru**

Instrukcja: **CALL** *etykieta_skoku*

Działanie: Instrukcja wykonuje skok do podprogramu

Znaczniki: Brak

Przykład: **CALL** **start**

Instrukcja: **CALL_T** *etykieta_skoku*

Działanie: Instrukcja wykonuje skok do podprogramu gdy jest ustawiony znacznik **F** prawda

Znaczniki: Brak

Przykład: **CALL_T** **koniec**

R4ATNR

Instrukcja: **CALL_F** *etykieta_skoku*

Działanie: Instrukcja wykonuje skok do podprogramu gdy jest skasowany znacznik **F** fałsz

Znaczniki: Brak

Przykład: **CALL_F** **koniec**

Instrukcja: **CALL_!** *etykieta_skoku*

Działanie: Instrukcja powoduje przeskok do podprogramu, z programu głównego do dodatkowego lub odwrotnie

Znaczniki: Zmienia stan znacznika N/F na przeciwny

Przykład: **CALL_!** **procedura_zakonczenia**

Instrukcja: **CALL_T!** *etykieta_skoku*

Działanie: Instrukcja wykonuje przeskok do podprogramu gdy jest ustawiony znacznik **F** prawda, z programu głównego do dodatkowego lub odwrotnie

Znaczniki: Zmienia stan znacznika N/F na przeciwny

Przykład: **CALL_T!** **polecenie_systemu**

Instrukcja: **CALL_F!** *etykieta_skoku*

Działanie: Instrukcja wykonuje przeskok do podprogramu gdy jest skasowany znacznik **F** fałsz,

Znaczniki: Zmienia stan znacznika N/F na przeciwny

Przykład: **CALL_F!** **analiza_numeru**

Instrukcja: **RET**

Działanie: Instrukcja powoduje powrót z obsługi podprogramu do miejsca wywołania lub kończy obsługę zdarzenia

Znaczniki: Zmienia stan znacznika N/F gdy powrót następuje pomiędzy programami głównym, a dodatkowym

R4ATNR

Przykład: **RET**

Instrukcja: **RET_T**

Działanie: Instrukcja powoduje powrót z obsługi podprogramu do miejsca wywołania lub kończy obsługę zdarzenia gdy ustawiony jest znacznik **F** prawda

Znaczniki: Zmienia stan znacznika **N/F** gdy powrót następuje pomiędzy programami głównym, a dodatkowym

Przykład: **RET_T**

Instrukcja: **RET_F**

Działanie: Instrukcja powoduje powrót z obsługi podprogramu do miejsca wywołania lub kończy obsługę zdarzenia gdy skasowany jest znacznik **F** fałsz

Znaczniki: Zmienia stan znacznika **N/F** gdy powrót następuje pomiędzy programami głównym, a dodatkowym

Przykład: **RET_F**

Instrukcja: **SKIP_T** [*ilość_instrukcji*]

Działanie: Instrukcja pomija podaną ilość instrukcji (max 16) występujących za tą instrukcją w sytuacji gdy jest ustawiony znacznik **F** prawda

Znaczniki: Brak

Przykład: **SKIP_T** [8]

Instrukcja: **SKIP_F** [*ilość_instrukcji*]

Działanie: Instrukcja pomija podaną ilość instrukcji (max 16) występujących za tą instrukcją w sytuacji gdy jest skasowany znacznik **F** fałsz

Znaczniki: Brak

Przykład: **SKIP_F** [6]

Instrukcja: **TEST_PLAY** <*flaga*>

R4ATNR

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy trwa odtwarzanie zapowiedzi czy nie

Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_PLAY** <**f0**>

Instrukcja: **TEST_REC** <*flaga*>

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy trwa nagrywanie rozmowy czy nie

Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_REC** <**f0**>

Instrukcja: **TEST_LINE_FREE** <*flaga*>

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy linia telefoniczna jest wolna czy nie (dotyczy wykrywania prądu w łączy)

Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_LINE_FREE** <**f0**>

Instrukcja: **TEST_LINE_POLA** <*flaga*>

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy linia telefoniczna ma umowną polaryzację A

Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_LINE_POLA** <**f0**>

Instrukcja: **TEST_LINE_POLB** <*flaga*>

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy linia telefoniczna ma umowną polaryzację B

Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_LINE_POLB** <**f0**>

R4ATNR

Instrukcja: **TEST_LINE_CURR** <flaga>
Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy linia telefoniczna ma umowną polaryzację A lub B
Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**
Przykład: **TEST_LINE_CURR** <**f0**>

Instrukcja: **TEST_LINE_CHANG** <flaga>
Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy na linii telefonicznej nastąpiła zmiana polaryzacji na przeciwną
Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**
Przykład: **TEST_LINE_CHANG** <**f0**>

Instrukcja: **TEST_LINE_RING** <flaga>
Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy linia telefoniczna jest w stanie dzwonienia, czy też nie
Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**
Przykład: **TEST_LINE_RING** <**f0**>

Instrukcja: **TEST_DTMF** <flaga> ['cyfra']
Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy odbiornik DTMF odebrał podaną cyfrę
Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**
Uwagi: Odbiornik DTMF musi być włączony odpowiednią instrukcją **SET_DSP_ON** [flagi sygnałów]
Przykład: **TEST_DTMF** <**f0**>['*']

Instrukcja: **TEST_SIGN** <flaga> [sygnał]
Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy zdekodowano sygnał synchroniczny np. zajętości
Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

R4ATNR

Uwagi: Odbiornik sygnałów musi być włączony instrukcją **SET_DSP_ON** [*flagi sygnałów*]

Przykład: **TEST_SIGN** <f0>[s0]

Instrukcja: **TEST_TEXT_FILE** <flaga>(rejestr tekstowy){tablica tekstów}

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy plik którego ścieżkę dostępu określa pozycja tablicy tekstów zawiera tekst zawarty w podanym rejestrze tekstowym **txt0** do **txt7** zajętości

Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_TEXT_FILE** <f0>(txt0){tab23}

Instrukcja: **TEST_NUM_ZERO** <flaga>(rejestr numeryczny)

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy wskazany rejestr numeryczny **num0** do **num7** jest równy zeru

Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_NUM_ZERO** <f0>(num0)

Instrukcja: **TEST_NUM_COMP** <flaga>(rejestr numeryczny1)(rejestr numeryczny2)

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy wskazane rejestry numeryczne **num0** do **num7** są równe

Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_NUM_COMP** <f0>(num0)(num1)

Instrukcja: **TEST_NUM_LESS** <flaga>(rejestr numeryczny1)(rejestr numeryczny2)

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy wskazany rejestr numeryczny 1 jest mniejszy od wskazanego rejestru numerycznego 2

Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_NUM_LESS** <f0>(num0)(num1)

R4ATNR

Instrukcja: **TEST_NUM_MORE** <flaga>(rejestr numeryczny1)(rejestr numeryczny2)

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy wskazany rejestr numeryczny 1 jest większy od wskazanego rejestru numerycznego 2

Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_NUM_MORE** <f0>(num0)(num1)

Instrukcja: **TEST_TIMER** <flaga>(rejestr timera)

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy został przekroczony czas podany rejestrze timera **tim0** do **tim7**, gdy czas nie został przekroczony zwracana jest prawda

Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_TIMER** <f0>(tim0)

Instrukcja: **TEST_TEXT_CHAR** (rejestr tekstowy)[rejestr numeryczny]['znak']

Działanie: Instrukcja ustawia flagę **F** w zależności od tego czy podany rejestr tekstowy **txt0** do **txt7** zawiera na pozycji określonej zmienną numeryczną **num0** do **num7** podany znak

Znaczniki: Zmienia stan flagi głównej **F**

Przykład: **TEST_TEXT_CHAR** (txt0)[num1]['A']

Instrukcja: **TEST_STER** <flaga>[numer wejścia sterbox-a]

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od stanu wejścia **p0** do **p7** sterboxa

Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_STER** <f0>[p0]

Instrukcja: **TEST_TEXT_COMP** <flaga>(rejestr tekstowy1)(rejestr tekstowy2)

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od tego czy wskazany rejestr tekstowy 1 zawiera taki sam tekst jak wskazany rejestr tekstowy 2

Znaczniki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

R4ATNR

Przykład: **TEST_TEXT_COMP** <f0>(txt0)(txt1)

Instrukcja: **TEST_IO_0** <flaga>

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od stanu portu IO 0.

Uwaga: Porty IO są dostępne w niektórych wykonaniach sprzętu.

Znaczniiki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_IO_0** <f0>

Instrukcja: **TEST_IO_1** <flaga>

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od stanu portu IO 1.

Uwaga: Porty IO są dostępne w specjalnych wykonaniach sprzętu.

Znaczniiki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_IO_1** <f0>

Instrukcja: **TEST_MARK** <marker>

Działanie: Instrukcja ustawia flagę **F** oraz jedną z flag dodatkowych **f0** do **f7** w zależności od stanu portu markera między procesorowego **m0** do **m7**.

Uwaga:

Znaczniiki: Zmienia stan flagi głównej **F** oraz jednej z dodatkowych **f0** do **f7**

Przykład: **TEST_MARK** <m0>

Instrukcja: **GET_RING** (*rejestr numeryczny*)

Działanie: Instrukcja pobiera do rejestru numerycznego zliczoną przez układ kontroli linii ilość dzwonków

Znaczniiki: Brak

Przykład: **GET_RING** (num0)

R4ATNR

Instrukcja: **GET_DTMF** (*rejestr tekstowy*)

Działanie: Instrukcja pobiera do rejestru tekstowego cyfry odebrane przez odbiornik DTMF

Uwaga: Odbiornik DTMF musi być włączony instrukcją **SET_DSP_ON** [*flagi sygnałów*]

Znaczniki: Brak

Przykład: **GET_DTMF (txt0)**

Instrukcja: **GET_CLIP** (*rejestr tekstowy*)

Działanie: Instrukcja pobiera do rejestru tekstowego numer odebrany przez odbiornik identyfikacji

Uwaga: Odbiornik clip-u musi być włączony instrukcją **SET_DSP_ON** [*flagi sygnałów*]

Znaczniki: Brak

Przykład: **GET_CLIP (txt0)**

Instrukcja: **GET_DATE** (*rejestr tekstowy*)

Działanie: Instrukcja pobiera do rejestru tekstowego aktualną datę w formacie tekstowym rrrr-mm-dd

Znaczniki: Brak

Przykład: **GET_DATE (txt0)**

Instrukcja: **GET_TIME** (*rejestr tekstowy*)

Działanie: Instrukcja pobiera do rejestru tekstowego aktualną godzinę w formacie tekstowym gg:mm:ss

Znaczniki: Brak

Przykład: **GET_TIME (txt0)**

Instrukcja: **GET_CHAN** (*rejestr tekstowy*)

Działanie: Instrukcja pobiera do rejestru tekstowego numer kanału w postaci tekstowej

Znaczniki: Brak

R4ATNR

Przykład: **GET_CHAN (txt0)**

Instrukcja: **GET_TIME_ZONE (rejestr numeryczny)**

Działanie: Instrukcja pobiera do rejestru numerycznego aktualną godzinę w postaci BCD (\$hhmmss). Wpis np. do *Tablicy numerów* danej do porównania powinien wyglądać następująco (przykład dla godziny 16:00): \$160000

Znaczniki: Brak

Przykład: **GET_TIME_ZONE (num0)**

Instrukcja: **GET_HOUR (rejestr numeryczny)**

Działanie: Instrukcja pobiera do rejestru numerycznego aktualną godzinę

Znaczniki: Brak

Przykład: **GET_HOUR (num0)**

Instrukcja: **GET_MINUTE (rejestr numeryczny)**

Działanie: Instrukcja pobiera do rejestru numerycznego aktualną minutę

Znaczniki: Brak

Przykład: **GET_MINUTE (num0)**

Instrukcja: **GET_YEAR (rejestr numeryczny)**

Działanie: Instrukcja pobiera do rejestru numerycznego aktualny rok

Znaczniki: Brak

Przykład: **GET_YEAR (num0)**

Instrukcja: **GET_MONTH (rejestr numeryczny)**

Działanie: Instrukcja pobiera do rejestru numerycznego aktualny miesiąc

Znaczniki: Brak

R4ATNR

Przykład: **GET_MONTH (num0)**

Instrukcja: **GET_DAY** (*rejestr numeryczny*)

Działanie: Instrukcja pobiera do rejestru numerycznego aktualny dzień

Znaczniki: Brak

Przykład: **GET_DAY (num0)**

Instrukcja: **GET_WEEK_DAY** (*rejestr numeryczny*)

Działanie: Instrukcja pobiera do rejestru numerycznego aktualny numer dnia tygodnia. Poniedziałek 0, wtorek 1,... niedziela 6.

Znaczniki: Brak

Przykład: **GET_WEEK_DAY (num0)**

Instrukcja: **GET_PLAY** (*rejestr numeryczny*)

Działanie: Instrukcja pobiera do rejestru numerycznego pozostałą do odtworzenia ilość sekund nagrania

Znaczniki: Brak

Przykład: **GET_PLAY (num0)**

Instrukcja: **GET_REC** (*rejestr numeryczny*)

Działanie: Instrukcja pobiera do rejestru numerycznego aktualną długość nagrania

Znaczniki: Brak

Przykład: **GET_REC (num0)**

Instrukcja: **GET_TIME_STAMP** (*rejestr numeryczny*)

Działanie: Instrukcja pobiera do rejestru numerycznego aktualny stempel czasowy liczony w sekundach od daty 2001-01-01 00:00:00

Znaczniki: Brak

Przykład: **GET_TIME_STAMP (num0)**

R4ATNR

Instrukcja: **GET_SUBJ_BUF** (*rejestr tekstowy*)

Działanie: Instrukcja pobiera do rejestru tekstowego temat ostatnio odebranego maila

Znaczni: Brak

Przykład: **GET_SUBJ_BUF (txt0)**

Instrukcja: **GET_SUB_TEXT** (*rejestr tekstowy*)(*rejestr numeryczny1*)(*rejestr numeryczny2*)

Działanie: Instrukcja pobiera do rejestru tekstowego tekst od znaku określonego rejestrem numerycznym 1 w ilości znaków określonych rejestrem numerycznym 2

Znaczni: Brak

Przykład: **GET_SUB_TEXT (txt0)(num0)(num1)**

Instrukcja: **GET_CNAM** (*rejestr tekstowy*)

Działanie: Instrukcja pobiera do rejestru tekstowego nazwę numer odebraną przez odbiornik identyfikacji

Uwaga: Odbiornik clip-u musi być włączony instrukcją **SET_DSP_ON** [*flagi sygnałów*]

Znaczni: Brak

Przykład: **GET_CNAM (txt0)**

Instrukcja: **GET_VU_LEV** (*rejestr numeryczny*)

Działanie: Instrukcja pobiera do rejestru numerycznego aktualny poziom sygnału w kanale (w zakresie od 0 do 7, 7 największy)

Uwaga: Odbiornik poziomu sygnału musi być włączony instrukcją **SET_DSP_ON** [*flagi sygnałów*]

Znaczni: Brak

Przykład: **GET_VU_LEV (num0)**

Instrukcja: **GET_SYS_LEV** (*rejestr numeryczny*)

R4ATNR

Działanie: Instrukcja pobiera do rejestru numerycznego aktualny poziom uruchomienia systemu

Znaczniiki: Brak

Przykład: **GET_SYS_LEV (num0)**

Instrukcja: **GET_DISC_CAL** (*rejestr numeryczny*)

Działanie: Instrukcja pobiera do rejestru numerycznego poziom zajętości karty pamięci w promilach

Uwaga: Instrukcja powoduje wyliczenie pojemności karty co może być czasochłonne, wskazane jest używanie instrukcji **GET_DISC_REG** która pobiera zajętość dysku z rejestru zajętości który jest automatycznie odświeżany w chwilach bezczynności

Znaczniiki: Brak

Przykład: **GET_DISC_CAL (num0)**

Instrukcja: **GET_DISC_REG** (*rejestr numeryczny*)

Działanie: Instrukcja pobiera do rejestru numerycznego poziom zajętości karty pamięci w promilach z rejestru zajętości odświeżanego okresowo w chwilach bezczynności

Znaczniiki: Brak

Przykład: **GET_DISC_REG (num0)**

Instrukcja: **SET_PLAY_ON** {*tablica tekstów*}

Działanie: Instrukcja uruchamia odtwarzanie zapowiedzi identyfikowanej w systemie plików poprzez wpis do tablicy tekstów

Znaczniiki: Brak

Przykład: **SET_PLAY_ON {tab2}**

Instrukcja: **SET_PLAY_OFF**

Działanie: Instrukcja zatrzymuje odtwarzanie zapowiedzi

Znaczniiki: Brak

R4ATNR

Przykład: **SET_PLAY_OFF**

Instrukcja: **SET_REC_ON**

Działanie: Instrukcja uruchamia nagrywanie w kanale

Znaczniki: Brak

Przykład: **SET_REC_ON**

Instrukcja: **SET_REC_OFF**

Działanie: Instrukcja zatrzymuje nagrywanie w kanale

Znaczniki: Brak

Przykład: **SET_REC_OFF**

Instrukcja: **SET_HOOK_ON**

Działanie: Instrukcja uruchamia obciążenie linii (podniesienie mikrotelefonu)

Znaczniki: Brak

Przykład: **SET_HOOK_ON**

Instrukcja: **SET_HOOK_OFF**

Działanie: Instrukcja dezaktywuje obciążenie linii (odłożenie mikrotelefonu)

Znaczniki: Brak

Przykład: **SET_HOOK_OFF**

Instrukcja: **SET_FLASH** (*rejestr numeryczny*)

Działanie: Instrukcja wywołuje flash-a (przerwa kalibrowana) na linii o czasie określonym przez rejestr numeryczny, czas należy podawać jako wielokrotność 20ms

Znaczniki: Brak

Przykład: **SET_FLASH (num0)**

R4ATNR

Instrukcja: **SET_DTMF** (*rejestr tekstowy*)

Działanie: Instrukcja powoduje wybranie w DTMF cyfr zawartych we wskazanym rejestrze tekstowym

Znaczniki: Brak

Przykład: **SET_DTMF (txt0)**

Instrukcja: **SET_DSP_ON** [*flagi sygnałów*]

Działanie: Instrukcja powoduje uruchomienie procesora rozpoznawania sygnałów
S – rozpoznawanie sygnałów synchronicznych (zajętości, ostrzegawczy itp.)
D – uruchamia odbiornik DTMF
C – uruchamia odbiornik clip-u (identyfikacji)
V – uruchamia rozpoznawanie poziomu sygnału

Uwaga: Procesora rozpoznawania sygnałów należy używać z umiarem ponieważ powoduje duże obciążenie procesora urządzenia R4ATNR co powoduje np. wolniejszy odczyt rejestracji z urządzenia

Znaczniki: Brak

Przykład: **SET_DSP_ON [S,D,C,V]**

Instrukcja: **SET_DSP_OFF**

Działanie: Instrukcja powoduje wyłączenie procesora rozpoznawania sygnałów

Znaczniki: Brak

Przykład: **SET_DSP_OFF**

Instrukcja: **SET_TIMER** (*rejestr timera*)(*rejestr numeryczny*)

Działanie: Instrukcja powoduje ustawienie wskazanego timera **tim0** do **tim7**, wartością numeryczną z rejestru numerycznego. Czas podajemy w milisekundach. Rozdzielczość timera wynosi 20ms

Znaczniki: Brak

Przykład: **SET_TIMER (tim0)(num0)**

Instrukcja: **SET_ROZM_NEW**

R4ATNR

Działanie: Instrukcja powoduje wyczyszczenie rejestrów generowania billingu i zachowuje godzinę rozpoczęcia rozmowy

Uwaga: Za pomocą specjalnych instrukcji należy uzupełnić wpisy billingu

Znaczniiki: Brak

Przykład: **SET_ROZM_NEW**

Instrukcja: **SET_ROZM_END**

Działanie: Instrukcja zamyka wpis billingu i zachowuje połączenie

Znaczniiki: Brak

Przykład: **SET_ROZM_END**

Instrukcja: **SET_LED** (*rejestr numeryczny*)

Działanie: Instrukcja ustawia tryb pracy diody led odpowiedniej dla kanału zgodnie z rejestrem numerycznym który określa numer wzorca (paternu)

Znaczniiki: Brak

Przykład: **SET_LED (num0)**

Uwaga: wzorzec „świecenia” LED jest w *Ustawieniach R4ATNR*, zakładka *Ustawienia ogólne, LED1..8 i LED9..16*. Liczby za *LED* wpisujemy do rejestru numerycznego aby otrzymać kolor i sposób migania.

Instrukcja: **SET_LCDA** (*rejestr tekstowy*)

Działanie: Instrukcja powoduje wyświetlenie na wyświetlaczu tekstowym rejestru tekstowego

Uwaga: Tylko specjalne wersje wykonania R4ATNR posiadają wyświetlacz LCD

Znaczniiki: Brak

Przykład: **SET_LCDA (txt0)**

Instrukcja: **SET_LCDH** (*rejestr tekstowy*)

Działanie: Instrukcja powoduje wyświetlenie na wyświetlaczu tekstowym rejestru tekstowego w górnej linii

R4ATNR

Uwaga: Tylko niektóre wersje wykonania R4ATNR posiadają wyświetlacz LCD

Znaczники: Brak

Przykład: **SET_LCDH (txt0)**

Instrukcja: **SET_LCDL** (*rejestr tekstowy*)

Działanie: Instrukcja powoduje wyświetlenie na wyświetlaczu tekstowym rejestru tekstowego w dolnej linii

Uwaga: Tylko niektóre wersje wykonania R4ATNR posiadają wyświetlacz LCD

Znaczniki: Brak

Przykład: **SET_LCDL (txt0)**

Instrukcja: **SET_FLAG** <*flaga*>

Działanie: Instrukcja ustawia jedną z flag dodatkowych **f0** do **f7**

Znaczniki: Zmienia stan jednej z flag dodatkowych **f0** do **f7**

Przykład: **SET_FLAG <f0>**

Instrukcja: **SET_MAIL_NEW**

Działanie: Instrukcja powoduje wyczyszczenie rejestrów generowania maila

Uwaga: Za pomocą specjalnych instrukcji należy uzupełnić wpisy maila

Znaczniki: Brak

Przykład: **SET_MAIL_NEW**

Instrukcja: **SET_MAIL_END**

Działanie: Instrukcja zamyka wpis maila i wysyła pocztę

Znaczniki: Brak

Przykład: **SET_MAIL_END**

R4ATNR

Instrukcja: **SET_FI**

Działanie: Instrukcja ustawia flagę **I** co powoduje, że program nie może zostać przerwany przez obsługę zdarzenia. Zdarzenia zostają dodane do kolejki obsługi zdarzeń

Znaczniki: Ustawia flagę **I**

Przykład: **SET_FI**

Instrukcja: **SET_IO_0_OUT**

Działanie: Instrukcja konfiguruje port IO 0 jako wyjście

Znaczniki: Brak

Uwaga: Porty IO są dostępne w niektórych wykonaniach sprzętu.

Przykład: **SET_IO_0_OUT**

Instrukcja: **SET_IO_1_OUT**

Działanie: Instrukcja konfiguruje port IO 1 jako wyjście

Znaczniki: Brak

Uwaga: Porty IO są dostępne w niektórych wykonaniach sprzętu.

Przykład: **SET_IO_1_OUT**

Instrukcja: **SET_IO_0_INP**

Działanie: Instrukcja konfiguruje port IO 0 jako wejście

Znaczniki: Brak

Uwaga: Porty IO są dostępne w niektórych wykonaniach sprzętu.

Przykład: **SET_IO_0_INP**

Instrukcja: **SET_IO_1_INP**

Działanie: Instrukcja konfiguruje port IO 1 jako wejście

Znaczniki: Brak

R4ATNR

Uwaga: Porty IO są dostępne w niektórych wykonaniach sprzętu.

Przykład: **SET_IO_1_INP**

Instrukcja: **SET_IO_0_H**

Działanie: Instrukcja ustawia na porcie IO 0 stan wysoki

Znaczniki: Brak

Uwaga: Porty IO są dostępne w niektórych wykonaniach sprzętu.

Przykład: **SET_IO_0_H**

Instrukcja: **SET_IO_0_L**

Działanie: Instrukcja ustawia na porcie IO 0 stan niski

Znaczniki: Brak

Uwaga: Porty IO są dostępne w niektórych wykonaniach sprzętu.

Przykład: **SET_IO_0_L**

Instrukcja: **SET_IO_0_TOG**

Działanie: Instrukcja ustawia na porcie IO 0 stan przeciwny do obecnego

Znaczniki: Brak

Uwaga: Porty IO są dostępne w niektórych wykonaniach sprzętu.

Przykład: **SET_IO_0_TOG**

Instrukcja: **SET_IO_1_H**

Działanie: Instrukcja ustawia na porcie IO 1 stan wysoki

Znaczniki: Brak

Uwaga: Porty IO są dostępne w niektórych wykonaniach sprzętu.

Przykład: **SET_IO_1_H**

R4ATNR

Instrukcja: **SET_IO_1_L**

Działanie: Instrukcja ustawia na porcie IO 1 stan niski

Znaczniki: Brak

Uwaga: Porty IO są dostępne w niektórych wykonaniach sprzętu.

Przykład: **SET_IO_1_L**

Instrukcja: **SET_IO_1_TOG**

Działanie: Instrukcja ustawia na porcie IO 1 stan przeciwny do obecnego

Znaczniki: Brak

Uwaga: Porty IO są dostępne w niektórych wykonaniach sprzętu.

Przykład: **SET_IO_1_TOG**

Instrukcja: **SET_CONN_ON**

Działanie: Instrukcja uruchamia akustyczny komutator kanału 1 z 2

Znaczniki: Brak

Uwaga: Komutator kanałów jest dostępny niektórych wykonaniach sprzętu.

Przykład: **SET_CONN_ON**

Instrukcja: **SET_CONN_OFF**

Działanie: Instrukcja wyłącza akustyczny komutator kanału 1 z 2

Znaczniki: Brak

Uwaga: Komutator kanałów jest dostępny niektórych wykonaniach sprzętu.

Przykład: **SET_CONN_OFF**

Instrukcja: **SET_MARK** <marker>

R4ATNR

Działanie: Instrukcja ustawia jeden z markerów **m0** do **m7**

Znaczники: Zmienia stan jednego z markerów **m0** do **m7**

Przykład: **SET_MARK <m0>**

Instrukcja: **PUT_RING** (*rejestr numeryczny*)

Działanie: Instrukcja ustawia licznik dzwonek w rejestrach generowania billingu

Znaczники: Brak

Przykład: **PUT_RING (num0)**

Instrukcja: **PUT_TYPE** (*rejestr numeryczny*)

Działanie: Instrukcja ustawia rejestr typu rozmowy w rejestrach generowania billingu

0 – wychodząca
1 – przychodząca
2 – niedoszła (nieudana)
3 - nieodebrana

Znaczники: Brak

Przykład: **PUT_TYPE (num0)**

Instrukcja: **PUT_DTMF** (*rejestr tekstowy*)

Działanie: Instrukcja ustawia rejestr numeru wybranego w rejestrach generowania billingu

Znaczники: Brak

Przykład: **PUT_DTMF (txt0)**

Instrukcja: **PUT_CLIP** (*rejestr tekstowy*)

Działanie: Instrukcja ustawia rejestr identyfikacji w rejestrach generowania billingu

Znaczники: Brak

Przykład: **PUT_CLIP (txt0)**

Instrukcja: **PUT_CNAM** (*rejestr tekstowy*)

R4ATNR

Działanie: Instrukcja ustawia rejestr nazwy numeru w rejestrach generowania billingu

Znaczniki: Brak

Przykład: **PUT_CNAM (txt0)**

Instrukcja: **PUT_SUBJ** (*rejestr tekstowy*)

Działanie: Instrukcja ustawia rejestr tematu w rejestrach generowania maila

Znaczniki: Brak

Przykład: **PUT_SUBJ (txt0)**

Instrukcja: **PUT_RECI** (*rejestr tekstowy*)

Działanie: Instrukcja ustawia rejestr adresu odbiorcy maila w rejestrach generowania maila

Znaczniki: Brak

Przykład: **PUT_RECI (txt0)**

Instrukcja: **CLR_DTMF**

Działanie: Instrukcja zeruje bufor odbiornika DTMF

Znaczniki: Brak

Przykład: **CLR_DTMF**

Instrukcja: **CLR_CLIP**

Działanie: Instrukcja zeruje bufory odbiornika identyfikacji

Znaczniki: Brak

Przykład: **CLR_CLIP**

Instrukcja: **CLR_SIGN**

Działanie: Instrukcja zeruje znaczniki rozpoznawania sygnałów

R4ATNR

Znaczniiki: Brak

Przykład: **CLR_SIGN**

Instrukcja: **CLR_FLAG** <*flaga*>

Działanie: Instrukcja kasuje jedną z flag dodatkowych **f0** do **f7**

Znaczniiki: Zmienia stan jednej z flag dodatkowych **f0** do **f7**

Przykład: **CLR_FLAG** <**f0**>

Instrukcja: **CLR_FI**

Działanie: Instrukcja kasuje flagę **I** co powoduje, że program może zostać przerwany przez obsługę zdarzenia

Znaczniiki: Kasuje flagę **I**

Przykład: **CLR_FI**

Instrukcja: **CLR_MARK** <*marker*>

Działanie: Instrukcja kasuje jeden z markerów **m0** do **m7**

Znaczniiki: Zmienia stan jednego z markerów **m0** do **m7**

Przykład: **CLR_MARK** <**m0**>

Instrukcja: **NUM_CLOAD** (*rejestr numeryczny*){*tablica numerów*}

Działanie: Instrukcja ładuje do rejestru numerycznego **num0** do **num7** wartość z tablicy numerów

Znaczniiki: Brak

Przykład: **NUM_CLOAD** (**num0**){**idx5**}

Instrukcja: **NUM_ILOAD** (*rejestr numeryczny*)[*stała numeryczna*]

Działanie: Instrukcja ładuje do rejestru numerycznego **num0** do **num7** stałą numeryczną podaną w rozkazie (od 0 do 127)

Znaczniiki: Brak

R4ATNR

Przykład: **NUM_ILOAD (num0)[100]**

Instrukcja: **NUM_IADD** (*rejestr numeryczny*)[*stała numeryczna*]

Działanie: Instrukcja dodaje do rejestru numerycznego **num0** do **num7** stałą numeryczną podaną w rozkazie (od 0 do 127)

Znaczni: Brak

Przykład: **NUM_IADD (num0)[100]**

Instrukcja: **NUM_ISUB** (*rejestr numeryczny*)[*stała numeryczna*]

Działanie: Instrukcja odejmuje od rejestru numerycznego **num0** do **num7** stałą numeryczną podaną w rozkazie (od 0 do 127)

Znaczni: Brak

Przykład: **NUM_ISUB (num0)[100]**

Instrukcja: **NUM_IMUL** (*rejestr numeryczny*)[*stała numeryczna*]

Działanie: Instrukcja mnoży rejestr numeryczny **num0** do **num7** przez stałą numeryczną podaną w rozkazie (od 0 do 127)

Znaczni: Brak

Przykład: **NUM_IMUL (num0)[100]**

Instrukcja: **NUM_IDIV** (*rejestr numeryczny*)[*stała numeryczna*]

Działanie: Instrukcja dzieli rejestr numeryczny **num0** do **num7** przez stałą numeryczną podaną w rozkazie (od 0 do 127)

Znaczni: Brak

Przykład: **NUM_IDIV (num0)[100]**

Instrukcja: **NUM_ADD** (*rejestr numeryczny 1*)(*rejestr numeryczny 2*)

Działanie: Instrukcja dodaje do rejestru numerycznego 1 zawartość rejestru numerycznego 2

Znaczni: Brak

R4ATNR

Przykład: **NUM_ADD (num0)(num1)**

Instrukcja: **NUM_SUB** (*rejestr numeryczny 1*)(*rejestr numeryczny 2*)

Działanie: Instrukcja odejmuje od rejestru numerycznego 1 zawartość rejestru numerycznego 2

Znaczniki: Brak

Przykład: **NUM_SUB (num0)(num1)**

Instrukcja: **NUM_MUL** (*rejestr numeryczny 1*)(*rejestr numeryczny 2*)

Działanie: Instrukcja mnoży rejestr numeryczny 1 przez zawartość rejestru numerycznego 2

Znaczniki: Brak

Przykład: **NUM_MUL (num0)(num1)**

Instrukcja: **NUM_DIV** (*rejestr numeryczny 1*)(*rejestr numeryczny 2*)

Działanie: Instrukcja dzieli rejestr numeryczny 1 przez zawartość rejestru numerycznego 2

Znaczniki: Brak

Przykład: **NUM_DIV (num0)(num1)**

Instrukcja: **NUM_CONV** (*rejestr numeryczny*)(*rejestr tekstowy*)

Działanie: Instrukcja dokonuje konwersji rejestru tekstowego na jego wartość numeryczną

Znaczniki: Brak

Przykład: **NUM_CONV (num0)(txt0)**

Instrukcja: **NUM_LTEXT** (*rejestr numeryczny*)(*rejestr tekstowy*)

Działanie: Instrukcja ładuje do rejestru numerycznego ilość znaków które zawiera tekst zawarty w rejestrze tekstowym

Znaczniki: brak

Przykład: **NUM_LTEXT (num0)(txt0)**

Uwaga: **od wersji ATNR 0.1 (program Tele32 v.11.1.6)**

R4ATNR

Instrukcja: **TEXT_CLOAD** (*rejestr tekstowy*){*tablica tekstów*}

Działanie: Instrukcja ładuje do rejestru tekstowego **txt0** do **txt7** wartość z tablicy tekstów

Znaczники: Brak

Przykład: **TEXT_CLOAD (txt0){tab5}**

Instrukcja: **TEXT_CLR** (*rejestr tekstowy*)

Działanie: Instrukcja zeruje rejestr tekstowy **txt0** do **txt7**

Znaczniki: Brak

Przykład: **TEXT_CLR (txt0)**

Instrukcja: **TEXT_RADD** (*rejestr tekstowy*)['znak']

Działanie: Instrukcja dodaje do rejestru tekstowego **txt0** do **txt7** podany znak z prawej strony

Znaczniki: Brak

Przykład: **TEXT_RADD (txt0) ['#']**

Instrukcja: **TEXT_LADD** (*rejestr tekstowy*)['znak']

Działanie: Instrukcja dodaje do rejestru tekstowego **txt0** do **txt7** podany znak z lewej strony

Znaczniki: Brak

Przykład: **TEXT_LADD (txt0) ['0']**

Instrukcja: **TEXT_LDEL** (*rejestr tekstowy*)[*ilość znaków*]

Działanie: Instrukcja kasuje w rejestrze tekstowym **txt0** do **txt7** podaną ilość znaków od lewej strony (od 1 do 16 znaków)

Znaczniki: Brak

Przykład: **TEXT_LDEL (txt0)[5]**

Instrukcja: **TEXT_RDEL** (*rejestr tekstowy*)[*ilość znaków*]

R4ATNR

Działanie: Instrukcja kasuje w rejestrze tekstowym **txt0** do **txt7** podaną ilość znaków od prawej strony (od 1 do 16 znaków)

Znaczniiki: Brak

Przykład: **TEXT_RDEL (txt0)[5]**

Instrukcja: **TEXT_CONV (rejestr tekstowy)(rejestr numeryczny)**

Działanie: Instrukcja dokonuje konwersji numeru na tekst

Znaczniiki: Brak

Przykład: **TEXT_CONV (txt0)(num1)**

Instrukcja: **TEXT_SUM (rejestr tekstowy 1)(rejestr tekstowy 2)**

Działanie: Instrukcja dokonuje do rejestru tekstowego 1 zawartość rejestru tekstowego 2

Znaczniiki: Brak

Przykład: **TEXT_SUM (txt0)(txt1)**

Instrukcja: **MASK_AND [numery flag]**

Działanie: Instrukcja wykonuje funkcję logiczną AND na flagach **f0** do **f7** o podanych numerach flag i w zależności od wyniku operacji ustawia flagę główną **F**. Numery flag w zakresie od 7 do 0

Znaczniiki: Ustawia znacznik **F** w zależności od wyniku operacji

Przykład: **MASK_AND [3,2,1,0]**

Instrukcja: **MASK_OR [numery flag]**

Działanie: Instrukcja wykonuje funkcję logiczną OR na flagach **f0** do **f7** o podanych numerach flag i w zależności od wyniku operacji ustawia flagę główną **F**. Numery flag w zakresie od 7 do 0

Znaczniiki: Ustawia znacznik **F** w zależności od wyniku operacji

Przykład: **MASK_OR [7,0]**

R4ATNR

Instrukcja: **MASK_DTMF_AND7** [*cyfry DTMF zobacz zakres*]

Działanie: Instrukcja wykonuje funkcję logiczną AND na podanych cyfrach kodu DTMF i w zależności od wyniku operacji ustawia flagę główną **F**. Zakres rozpoznawanych kodów DTMF 0,1,2,3,4,5,*,#

Znaczniki: Ustawia znacznik **F** w zależności od wyniku operacji

Przykład: **MASK_DTMF_AND7 [0123]**

Instrukcja: **MASK_DTMF_OR7** [*cyfry DTMF zobacz zakres*]

Działanie: Instrukcja wykonuje funkcję logiczną OR na podanych cyfrach kodu DTMF i w zależności od wyniku operacji ustawia flagę główną **F**. Zakres rozpoznawanych kodów DTMF 0,1,2,3,4,5,*,#

Znaczniki: Ustawia znacznik **F** w zależności od wyniku operacji

Przykład: **MASK_DTMF_OR7 [0123]**

Instrukcja: **MASK_DTMF_AND6** [*cyfry DTMF zobacz zakres*]

Działanie: Instrukcja wykonuje funkcję logiczną AND na podanych cyfrach kodu DTMF i w zależności od wyniku operacji ustawia flagę główną **F**. Zakres rozpoznawanych kodów DTMF 0,5,6,7,8,9,*,#

Znaczniki: Ustawia znacznik **F** w zależności od wyniku operacji

Przykład: **MASK_DTMF_AND6 [0123]**

Instrukcja: **MASK_DTMF_OR6** [*cyfry DTMF zobacz zakres*]

Działanie: Instrukcja wykonuje funkcję logiczną OR na podanych cyfrach kodu DTMF i w zależności od wyniku operacji ustawia flagę główną **F**. Zakres rozpoznawanych kodów DTMF 0,5,6,7,8,9,*,#

Znaczniki: Ustawia znacznik **F** w zależności od wyniku operacji

Przykład: **MASK_DTMF_OR6 [0123]**

Instrukcja: **MASK_LINE_AND5** [*stany linii zobacz zakres*]

Działanie: Instrukcja wykonuje funkcję logiczną AND na podanych stanach linii telefonicznej i w zależności od wyniku operacji ustawia flagę główną **F**.
Stany linii : a – polaryzacja A

R4ATNR

b – polaryzacja B
c – zmiana polaryzacji
r – dzwonenie
f – wykryty flash

Znaczniki: Ustawia znacznik **F** w zależności od wyniku operacji

Przykład: **MASK_LINE_AND5 [abc]**

Instrukcja: **MASK_LINE_OR5 [stany linii zobacz zakres]**

Działanie: Instrukcja wykonuje funkcję logiczną OR na podanych stanach linii telefonicznej i w zależności od wyniku operacji ustawia flagę główną **F**.

Stany linii : a – polaryzacja A
b – polaryzacja B
c – zmiana polaryzacji
r – dzwonenie
f – wykryty flash

Znaczniki: Ustawia znacznik **F** w zależności od wyniku operacji

Przykład: **MASK_LINE_OR5 [abc]**

Instrukcja: **MASK_SIGN_AND4 [sygnały]**

Działanie: Instrukcja wykonuje funkcję logiczną AND na podanych bitach wykrywania sygnałów i w zależności od wyniku operacji ustawia flagę główną **F**.
Sygnały: s0 do s3

Znaczniki: Ustawia znacznik **F** w zależności od wyniku operacji

Przykład: **MASK_SIGN_AND4 [s0,s1]**

Instrukcja: **MASK_SIGN_OR4 [sygnały]**

Działanie: Instrukcja wykonuje funkcję logiczną OR na podanych bitach wykrywania sygnałów i w zależności od wyniku operacji ustawia flagę główną **F**.
Sygnały: s0 do s3

Znaczniki: Ustawia znacznik **F** w zależności od wyniku operacji

Przykład: **MASK_SIGN_OR4 [s0,s1]**

Instrukcja: **WAIT_TIMER (rejestr timera)**

R4ATNR

Działanie: Instrukcja czeka na koniec czasu określony przez rejestr timera **tim7** do **tim0**

Znaczniki: Brak

Przykład: **WAIT_TIMER (tim0)**

Instrukcja: **WAIT_PLAY**

Działanie: Instrukcja czeka na koniec odtwarzania zapowiedzi

Znaczniki: Brak

Przykład: **WAIT_PLAY**

Instrukcja: **WAIT_FLASH**

Działanie: Instrukcja czeka na koniec generowania flash-a

Znaczniki: Brak

Przykład: **WAIT_FLASH**

Instrukcja: **WAIT_DTMF**

Działanie: Instrukcja czeka na koniec wybierania tonowego

Znaczniki: Brak

Przykład: **WAIT_DTMF**

Instrukcja: **SAVE_CREATE** (*rejestr tekstowy*)

Działanie: Instrukcja tworzy plik o ścieżce określonej przez rejestr tekstowy **txt0** do **txt7**

Znaczniki: Brak

Przykład: **SAVE_CREATE (txt0)**

Instrukcja: **SAVE_TEXT** (*rejestr tekstowy 1*)(*rejestr tekstowy 2*)

Działanie: Instrukcja zapisuje do pliku o ścieżce określonej przez rejestr tekstowy 1 rejestr tekstowy 2

Znaczniki: Brak

R4ATNR

Przykład: **SAVE_TEXT (txt0)(txt1)**

Instrukcja: **SAVE_NUM** (*rejestr tekstowy*)(*rejestr numeryczny*)

Działanie: Instrukcja zapisuje do pliku o ścieżce określonej przez rejestr tekstowy rejestr numeryczny w postaci tekstowej

Znaczники: Brak

Przykład: **SAVE_TEXT (txt0)(num1)**

Instrukcja: **SAVE_STER_ON** [*alias sterbox-a*](*rejestr tekstowy*)

Działanie: Instrukcja wysyła polecenie włączenia wejścia określonego przez **txt0** do **txt7** do sterbox-a określonego poprzez alias **p0** do **p7**

Znaczniki: Brak

Przykład: **SAVE_STER_ON [p0](txt0)**

Instrukcja: **SAVE_STER_OFF** [*alias sterbox-a*](*rejestr tekstowy*)

Działanie: Instrukcja wysyła polecenie wyłączenia wejścia określonego przez **txt0** do **txt7** do sterbox-a określonego poprzez alias **p0** do **p7**

Znaczniki: Brak

Przykład: **SAVE_STER_OFF [p0](txt0)**

Instrukcja: **END**

Działanie: Instrukcja kończy wykonywanie programu

Znaczniki: Brak

Przykład: **END**

Zestawienie instrukcji:

Parametry instrukcji:

[n16] - stała numeryczna 0 do 15
<fff> - flaga f0 do f7

R4ATNR

<mmm> - marker m0 do m7
 ['d'] - kod DTMF
 [sig] - flagi sygnałów
 (nnn) - rejestr numeryczny num0 do num7
 (ttt) - rejestr tekstowy txt0 do txt7
 {tab} - tablica tekstów tab0 do tab127
 {idx} - tablica numerów idx0 do idx127
 (tim) - rejestr timera tim0 do tim7
 ['c'] - znak
 [nn] - stała numeryczna od 0 do 127
 [st] - alias sterbox-a p0 do p7
 [dsp] - flagi dsp

Instrukcja	Parametry	Opis
JMP	etykieta	Skok bezwarunkowy
JMP_T	etykieta	Skok gdy F = prawda
JMP_F	etykieta	Skok gdy F = fałsz
JMP_!	etykieta	Przeskok do programu dodatkowego/głównego
JMP_T!	etykieta	Przeskok do programu dodatkowego/głównego gdy F = prawda
JMP_F!	etykieta	Przeskok do programu dodatkowego/głównego gdy F = fałsz
CALL	etykieta	Wywołanie podprogramu
CALL_T	etykieta	Wywołanie podprogramu gdy F = prawda
CALL_F	etykieta	Wywołanie podprogramu gdy F = fałsz
CALL_!	etykieta	Przeskok do podprogramu dodatkowego/głównego
CALL_T!	etykieta	Przeskok do podprogramu dodatkowego/głównego gdy F = prawda
CALL_F!	etykieta	Przeskok do podprogramu dodatkowego/głównego gdy F = fałsz
RET		Powrót z podprogramu lub zakończenie zdarzenia
RET_T		Powrót z podprogramu lub zakończenie

R4ATNR

		zdarzenia gdy F = prawda
RET_F		Powrót z podprogramu lub zakończenie zdarzenia gdy F = fałsz
SKIP_T	[n16]	Pomiń instrukcje gdy F = prawda
SKIP_F	[n16]	Pomiń instrukcje gdy F = fałsz
TEST_PLAY	<fff>	Czy odtwarzanie
TEST_REC	<fff>	Czy nagrywanie
TEST_LINE_FREE	<fff>	Czy linia wolna
TEST_LINE_POLA	<fff>	Czy polaryzacja A
TEST_LINE_POLB	<fff>	Czy polaryzacja B
TEST_LINE_CURR	<fff>	Czy lina aktywna
TEST_LINE_CHANG	<fff>	Czy zmiana polaryzacji
TEST_LINE_RING	<fff>	Czy dzwonenie
TEST_DTMF	<fff>['d']	Czy kod DTMF
TEST_SIGN	<fff>[sig]	Czy wykryty sygnał
TEST_TEXT_FILE	<fff>(ttt){tab}	Czy tekst zawarty w pliku
TEST_NUM_ZERO	<fff>(nnn)	Czy zero
TEST_NUM_COMP	<fff>(nnn)(nnn)	Czy numery równe
TEST_NUM_LESS	<fff>(nnn)(nnn)	Czy numer mniejszy
TEST_NUM_MORE	<fff>(nnn)(nnn)	Czy numer większy
TEST_TIMER	<fff>(tim)	Czy czas timera
TEST_TEXT_CHAR	(ttt)(nnn)['c']	Czy tekst zawiera znak
TEST_STER	<fff>[st]	Stan wejścia sterbox-a
TEST_TEXT_COMP	<fff>(ttt)(ttt)	Czy teksty równe
TEST_IO_0	<fff>	Jaki stan wejścia IO
TEST_IO_1	<fff>	Jaki stan wejścia IO
TEST_MARK	<mmm>	Czy marker
GET_RING	(nnn)	Pobierz licznik dzwonków
GET_DTMF	(ttt)	Pobierz DTMF
GET_CLIP	(ttt)	Pobierz identyfikację
GET_DATE	(ttt)	Pobierz datę

R4ATNR

GET_TIME	(ttt)	Pobierz czas
GET_CHAN	(ttt)	Pobierz numer kanału
GET_TIME_ZONE	(nnn)	Pobierz strefę czasową
GET_HOUR	(nnn)	Pobierz godzinę
GET_MINUTE	(nnn)	Pobierz minutę
GET_YEAR	(nnn)	Pobierz rok
GET_MONTH	(nnn)	Pobierz miesiąc
GET_DAY	(nnn)	Pobierz dzień
GET_WEEK_DAY	(nnn)	Pobierz dzień tygodnia
GET_PLAY	(nnn)	Pobierz pozostały czas odtwarzania
GET_REC	(nnn)	Pobierz czas nagrania
GET_TIME_STAMP	(nnn)	Pobierz stempel czasowy
GET_SUBJ_BUF	(ttt)	Pobierz bufor tematu maila
GET_SUB_TEXT	(ttt) (nnn) (nnn)	Wytnij tekst
GET_CNAM	(ttt)	Pobierz nazwę numeru
GET_VU_LEV	(nnn)	Pobierz poziom sygnału
GET_SYS_LEV	(nnn)	Pobierz poziom uruchomienia systemu
GET_DISC_CAL	(nnn)	Wylicz zajętość karty
GET_DISC_REG	(nnn)	Pobierz zajętość karty
SET_PLAY_ON	{tab}	Uruchom odtwarzanie
SET_PLAY_OFF		Zakończ odtwarzanie
SET_REC_ON		Uruchom nagrywanie
SET_REC_OFF		Zakończ nagrywanie
SET_HOOK_ON		Zajmij linię
SET_HOOK_OFF		Zwolnij linię
SET_FLASH	(nnn)	Wygeneruj flash-a na linii
SET_DTMF	(ttt)	Wybierz w DTMF
SET_DSP_ON	[dsp]	Włącz procesor dsp
SET_DSP_OFF		Wyłącz procesor dsp
SET_TIMER	(tim) (nnn)	Ustaw timer

R4ATNR

SET_ROZM_NEW		Inicjuj billing
SET_ROZM_END		Zamknij billing
SET_LED	(nnn)	Ustaw led-a
SET_LCDA	(ttt)	Tekst na wyświetlacz
SET_LCDH	(ttt)	Tekst na wyświetlacz górną linię
SET_LCDL	(ttt)	Tekst na wyświetlacz dolną linię
SET_FLAG	<fff>	Ustaw flagę
SET_MAIL_NEW		Inicjuj maila
SET_MAIL_END		Zamknij maila
SET_FI		Ustaw znacznik I
SET_IO_0_OUT		Konfiguruj port IO 0 jako wyjście
SET_IO_1_OUT		Konfiguruj port IO 1 jako wyjście
SET_IO_0_INP		Konfiguruj port IO 0 jako wejście
SET_IO_1_INP		Konfiguruj port IO 1 jako wejście
SET_IO_0_H		Ustaw na porcie IO 0 stan wysoki
SET_IO_0_L		Ustaw na porcie IO 0 stan niski
SET_IO_0_TOG		Ustaw na porcie IO 0 stan przeciwny do obecnego
SET_IO_1_H		Ustaw na porcie IO 1 stan wysoki
SET_IO_1_L		Ustaw na porcie IO 1 stan niski
SET_IO_1_TOG		Ustaw na porcie IO 1 stan przeciwny do obecnego
SET_CONN_ON		Włącz komutator kanałów
SET_CONN_OFF		Wyłącz komutator kanałów
SET_MARK	<mmm>	Ustaw marker
PUT_RING	(nnn)	Dzwonki
PUT_TYPE	(nnn)	Typ rozmowy
PUT_DTMF	(ttt)	Dtmf
PUT_CLIP	(ttt)	Identyfikacja

R4ATNR

PUT_CNAM	(ttt)	Nazwa
PUT_SUBJ	(ttt)	Temat
PUT_RECI	(ttt)	Odbiorca maila
CLR_DTMF		Wyzeruj rejestr DTMF
CLR_CLIP		Wyzeruj rejestry identyfikacji
CLR_SIGN		Wyzeruj flagi sygnałów
CLR_FLAG	<fff>	Skasuj flagę
CLR_FI		Skasuj znacznik I
CLR_MARK	<mmm>	Skasuj marker
NUM_CLOAD	(nnn){idx}	Załaduj numer z tablicy numerów
NUM_ILOAD	(nnn)[nn]	Załaduj stałą
NUM_IADD	(nnn)[nn]	Dodaj stałą
NUM_ISUB	(nnn)[nn]	Odejmij stałą
NUM_IMUL	(nnn)[nn]	Przemnóż przez stałą
NUM_IDIV	(nnn)[nn]	Podziel przez stałą
NUM_ADD	(nnn)(nnn)	Dodaj
NUM_SUB	(nnn)(nnn)	Odejmij
NUM_MUL	(nnn)(nnn)	Przemnóż
NUM_DIV	(nnn)(nnn)	Podziel
NUM_CONV	(nnn)(ttt)	Konwersja tekstu na numer
NUM_LTEKST	(nnn)(ttt)	Ilość znaków w tekście
TEXT_CLOAD	(ttt){tab}	Załaduj tekst z tablicy tekstów
TEXT_CLR	(ttt)	Wyzeruj tekst
TEXT_RADD	(ttt)['c']	Dodaj znak z prawej do tekstu
TEXT_LADD	(ttt)['c']	Dodaj znak z lewej do tekstu
TEXT_LDEL	(ttt)[n16]	Obetnij tekst od lewej
TEXT_RDEL	(ttt)[n16]	Obetnij tekst od prawej
TEXT_CONV	(ttt)(nnn)	Konwersja numeru na tekst
TEXT_SUM	(ttt)(ttt)	Dodaj teksty
MASK_AND	[7,6,5,4,3,2,1,0]	Maska and na flagach

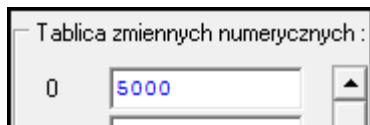
R4ATNR

MASK_OR	[7,6,5,4,3,2,1,0]	Maska or na flagach
MASK_DTMF_AND7	[012345*#]	Maska and na DTMF
MASK_DTMF_OR7	[012345*#]	Maska or na DTMF
MASK_DTMF_AND6	[056789*#]	Maska and na DTMF
MASK_DTMF_OR6	[056789*#]	Maska or na DTMF
MASK_LINE_AND5	[abcrf]	Maska and na stanie linii
MASK_LINE_OR5	[abcrf]	Maska or na stanie linii
MASK_SIGN_AND4	[sig]	Maska and na sygnałach
MASK_SIGN_OR4	[sig]	Maska or na sygnałach
WAIT_TIMER	(tim)	Czekaj do końca timera
WAIT_PLAY		Czekaj do końca odtwarzania
WAIT_FLASH		Czekaj do końca flash-a
WAIT_DTMF		Czekaj do końca wybierania DTMF
SAVE_CREATE	(ttt)	Utwórz plik
SAVE_TEXT	(ttt) (ttt)	Zapisz tekst do pliku
SAVE_NUM	(ttt) (nnn)	Zapisz numer do pliku
SAVE_STER_ON	[st] (ttt)	Polecenie włączenia w sterbox-ie
SAVE_STER_OFF	[st] (ttt)	Polecenie wyłączenia w sterbox-ie
END		Zakończenie programu

Przykładowy program.

1. Migamy lampką.

Miga lampka w kanale do którego wpisujemy skrypt. Czas mignięcia i czas przerwy jest taki sam i jest wpisany w:



```

NUM_ILOAD (num0) (idx0) // załaduj do rejestru 0 liczbę z tablicy z pozycji 0
NUM_ILOAD (num1) [0] //załaduj do rejestru 1 cyfrę 0
NUM_ILOAD (num2) [1]

```

R4ATNR

```
powt: SET_LED (num1)           //zgaś LED
      SET_TIMER (tim0) (num0)  //załaduj timer0 (5000 milisekund=5sekund)
      WAIT_TIMER (tim0)        //odlicz czas
      SET_LED (num2)           //zapal LED
      SET_TIMER (tim0) (num0)
      WAIT_TIMER (tim0)
      JMP powt
```

2. Migamy lampkami naprzemiennie.

W jednym kanale migniemy na zielono, w drugim na czerwono. Do komunikacji pomiędzy kanałami użyjemy markera <m0>. Czas jak poprzednio w *Tablicy numerycznej*. Najpierw skrypt wpisany do *Kanału 1*:

```
NUM_ILOAD (num0)[0]
      NUM_ILOAD (num1)[1]
      NUM_CLOAD (num2){idx0}
etyk: SET_LED (num0)
      CLR_MARK <m0>
      SET_TIMER (tim0)(num2)
      WAIT_TIMER (tim0)
      SET_LED (num1)
      SET_MARK <m0>
      SET_TIMER (tim0)(num2)
      WAIT_TIMER (tim0)
      JMP etyk
```

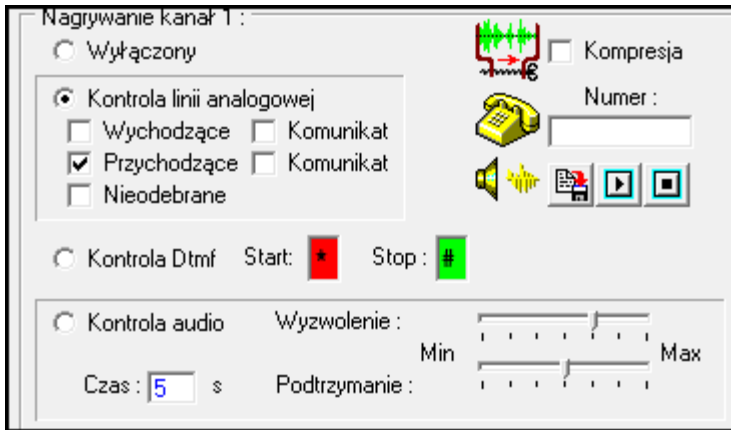
A teraz tekst skryptu w *Kanale 2*:

```
etyk: TEST_MARK <m0>
      JMP_F etyk1
      NUM_ILOAD (num0)[0]
      SET_LED (num0)
      JMP etyk
etyk1: NUM_ILOAD (num0)[2]
      SET_LED (num0)
      JMP etyk
```

3. Program generowany przez edytor graficzny.

Poniżej pokazano program wygenerowany przez edytor graficzny dla nagrywania rozmów.

R4ATNR



Rysunek 3. Ustawienia edytora graficznego.

Program jest umieszczony tylko w *Programie głównym*, nie zawiera żadnych wpisów w *tablicy obsługi zdarzeń*, *Tablicy tekstów* czy *Tablicy numerów* Tekst programu:

```
// Program sterujący nagrywaniem
// z linii analogowej

start: TEST_LINE_RING <f0>           // czy dzwonek
      CALL_T ring                     // tak
      TEST_LINE_CURR <f0>            // czy linia ?
      CALL_T line_o                  // tak
      JMP start                       // pętla

      // obsługa połączenia przychodzącego
ring:  NUM_ILOAD (num0)[10]          // } led
      SET_LED (num0)                 // }
      SET_DSP_ON [C]                 // włącz odbiornik clip-u
      SET_ROZM_NEW                   // nowa rozmowa
wring: TEST_LINE_CURR <f0>          // czy odebrane ?
      JMP_T line_i                   // tak
      TEST_LINE_RING <f0>           // czy dzwonięcie ?
      JMP_T wring                    // tak
      SET_DSP_OFF                    // wyłącz dsp
      NUM_ILOAD (num0)[0]           // } led
      SET_LED (num0)                 // }
      RET                            //
```

R4ATNR

```
    ;obsługa połączenia przychodzącego
line_i: NUM_ILOAD (num0)[1]      // } led
        SET_LED (num0)           // }
        SET_DSP_OFF              // wyłącz dsp
    ;inicjowanie rozmowy przychodzącej
        SET_ROZM_NEW             //
        GET_RING (num0)          // dzwonki
        PUT_RING (num0)          //
        NUM_ILOAD (num0)[1]      // przychodząca
        PUT_TYPE (num0)          //
        GET_CLIP (txt0)          // clip
        PUT_CLIP (txt0)          //
        SET_REC_ON                // no to nagrywamy
    ;rozmowa zainicjowana
wline_i:TEST_LINE_CURR <f0>     // czy prąd ?
        JMP_T wline_i            // jeszcze tak
    ;generowanie rozmowy przychodzącej
        SET_REC_OFF              // koniec nagrania
        SET_ROZM_END             // rozmowa !!!
    ;rozmowa wygenerowana
        NUM_ILOAD (num0)[0]      // } led
        SET_LED (num0)           // }
        RET                       //

    ;obsługa połączenia wychodzącego
line_o: NUM_ILOAD (num0)[4]      // } led
        SET_LED (num0)           // }
    ;nadzór połączenia
wline_t:TEST_LINE_CURR <f0>     // czy prąd ?
        JMP_T wline_t            // jeszcze tak
        NUM_ILOAD (num0)[0]      // } led
        SET_LED (num0)           // }
        RET                       //
```

4. Sterbox – najpierw migamy lampką w Sterboxie.

R4ATNR

Sieć	Sterbox	Pocztą	Ftp	Led 1..8	Led 9..16
Alias p0	Adres IP	Port	Port Sterbox		
	192.168. 0.177	7080	7080		
Alias p1	0. 0. 0. 0	RS485	Hasło Sterbox		
			1234		
Alias p2	0 0 0 0	RS485			

Ustawienia portów				
Nazwa	Stan aktywny	Stan po włącz.	Tylko wejście	Opis
ww0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ww0

Tablica tekstów i tablica skoków :	
0	ww0

W ATNR *Zaawansowane, Ustawienia ogólne, Sterbox*. Wpisany IP Sterboxa (Sterbox i ATNR musi być w tej samej podsieci). *Hasło Sterbox* z ustawień generalnych (w Sterboxie).

W Sterboxie migamy lampką związaną z wyjściem *ww0* (wartość umieszczona w kolumnie *Opis*).

W ATNR *Tablica tekstów* na pozycji 0 zawiera nazwę wyjścia ze Sterboxa

```

NUM_LOAD (num0){idx0} //załadowaj czas z tablicy numerycznej, pozycji 0
TEXT_LOAD (txt0){tab0} //załadowaj nazwę wyjścia Sterboxa z tablicy tekstów
etyk: SAVE_STER_ON [p0](txt0) //wyślij komendę do Sterboxa o IP=[p0] na wyjście=(txt0)
etyk1: SET_TIMER (tim0)(num0)
      WAIT_TIMER (tim0)
      SAVE_STER_OFF [p0](txt0)
      SET_TIMER (tim0)(num0)
      WAIT_TIMER (tim0)
      JMP etyk
      SET_TIMER (tim0)(num2)
      WAIT_TIMER (tim0)
      JMP etyk1

```

5. Odbieramy komendy ze Sterboxa.

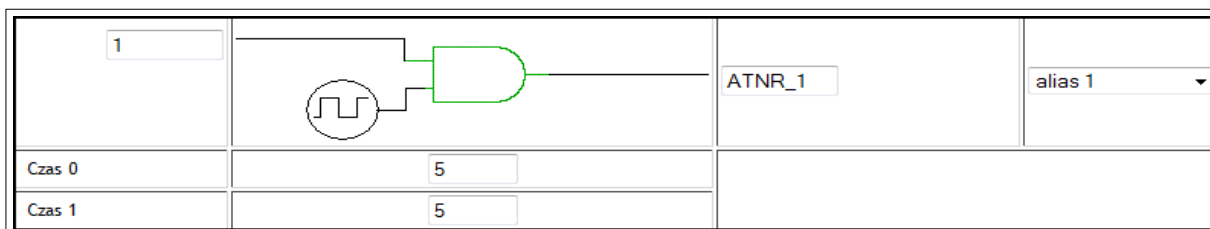
Proste ćwiczenie: generator Sterboxa zaświeca i gasi lampkę w R4ATNR.

Ustawienia aliasów	
Alias Sterbox 1	
192.168.0.197	7080

Ustawienia w Sterboxie: najpierw wpisujemy adres ATNR do tablicy *Alias*.

Następnie ustawiamy generator.

R4ATNR



W R4ATNR wpisujemy następujące ustawienia:

Port Sterbox 7080	Wejścia p0 ATNR_1	Zaawansowane, Ustawienia ogólne, Sterbox wpisujemy port dla sterowania ze Sterboxa oraz nazwę wejścia <p0>
----------------------	----------------------	--

Teść skryptu umieszczona np. w *Kanał 1*:

```

NUM_ILOAD (num0) [0] //wartość wyłączenia LED
NUM_ILOAD (num1) [1] //wartość włączenia LED na czerwono
ety1: TEST_STER <f0> [p0] //sprawdź co w Sterboxie
      JMP_T tak //jeśli w Sterboxie 1 to skocz
      SET_LED (num0) //w Sterboxie 0, zgaś LED
      JMP ety1 //sprawdź ponownie
tak: SET_LED (num1) //zapal LED na czerwono
      JMP ety1 //sprawdź ponownie
    
```

6. Odbierz ze Sterboxa komendę wybrania numeru i odtworzenia komunikatu po zgłoszeniu się rozmówcy.

Ustawienia Sterboxa:

Ustawienia klawiszy ekranowych				
Nazwa	Sposób działania	Opis	Powiązanie	Alias
kl0	Impuls	dzwoń	ATNR_1	alias 1

Ustawienia aliasów	
Alias Sterbox 1	
192.168.0.197	7080

Ustawienia: powyżej *klawiszy ekranowych* – klawisz w pierwszym wierszu będzie miał napis „dzwoń”, obok *aliasów* – adres IP ATNR

Widok wycinka ekranu użytkownika Sterboxa:

	Naciśnięcie klawisza na ekranie spowoduje zadzwonienie i odtworzenie zapowiedzi.
--	--

Ustawienia R4ATNR:

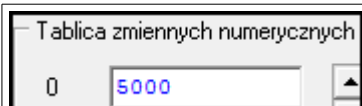
Port Sterbox 7080	Wejścia p0 ATNR_1	Zaawansowane, Ustawienia ogólne, Sterbox wpisujemy port dla sterowania ze Sterboxa oraz nazwę wejścia <p0>
----------------------	----------------------	--

R4ATNR



W pozycji {tab0} numer do wybrania, w pozycji {tab1} plik do odtworzenia. Uwaga: plik został załadowany przy pomocy edytora graficznego na zakładce *Podstawowe*. Jego parametry określone są w instrukcji od R4ATNR.

Uwaga: linia telefoniczna z centrali abonenckiej, numer podany w tablicy powoduje uruchomienie w telefonach cyfrowych funkcji rozgłaszania – uruchamia głośniki. Gdybyśmy chcieli wykorzystać normalną linię od operatora należałoby w najprostszym wypadku wydłużyć czas odliczany przez timer po wybraniu DTMF. Innym rozwiązaniem jest czekanie z odtworzeniem na zmianę polaryzacji żył rozmównych, oczywiście jeżeli taka usługę świadczy operator.



Wartość {idx0} używana do opóźnień.

```
ster:  TEST_STER <f0>[p0]           //testuj wejście ze Sterboxa
        JMP_F ster                   // na wejściu 0, sprawdzaj dalej
        NUM_ILOAD (num0)[2]         //wejście aktywne, załaduj świecenie lampki na zielono
        SET_LED (num0)              //włącz lampkę
        SET_HOOK_ON                 //zajmij linię
        NUM_CLOAD (num0){idx0}     //załaduj z tablicy
        SET_TIMER (tim0)(num0)     //ustaw timer
        WAIT_TIMER (tim0)           //odlicz czas
        TEXT_CLOAD (txt0){tab0}    //załaduj z tablicy
        NUM_ILOAD (num0)[1]
        SET_LED (num0)              //zapal lampkę na czerwono
        SET_DTMF (txt0)             //wybierz numer
        WAIT_DTMF                   //czekaj na wybranie numeru
        NUM_CLOAD (num0){idx0}
        SET_TIMER (tim0)(num0)
        WAIT_TIMER (tim0)           //odczekaj na zgłoszenie rozmówcy
        SET_PLAY_ON {tab1}         //odtwórz nagranie
        WAIT_PLAY                    //czekaj do końca odtwarzania
        NUM_CLOAD (num0){idx0}
        NUM_IMUL (num0)[2]
        SET_TIMER (tim0)(num0)
        WAIT_TIMER (tim0)           //odczekaj
        SET_HOOK_OFF                //zwolnij linię
        NUM_ILOAD (num0)[0]
        SET_LED (num0)              //zgaś lampkę
```

R4ATNR

JMP ster

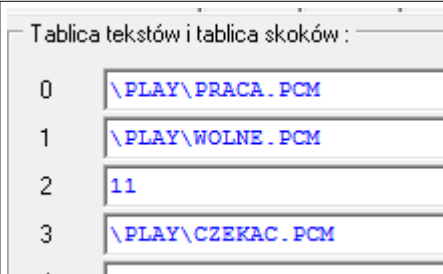
//czekaj na następane uaktywnienie ze Sterboxa

7. Prosta DISA.

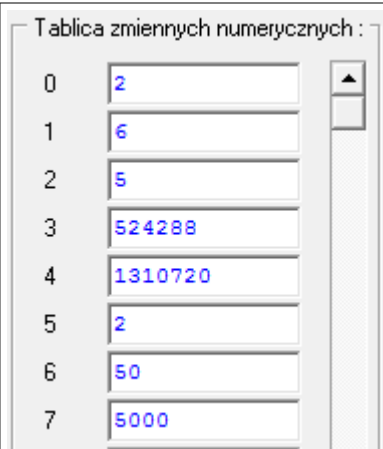
Odbiera dzwonięcie z linii, odtwarza zapowiedź powitalną zależnie od pory dnia i łączy na wybrany numer.

Co robi dodzwaniający się	Czas pracy	Czas wolny
Dodzwania się	Zapowiedź dla czasu pracy	Zapowiedź dla czasu wolnego
Wybiera numer wewnętrzny	Transfer do wybranego numeru	Transfer do wybranego numeru
Wybiera 0	Łączy do operatora	Łączy do operatora
Nic nie robi	Łączy do operatora	Rozłącza

Ustawienia Tabeli tekstów

	<p>{tab0} Zapowiedź w pracy, {tab1} zapowiedź po pracy.</p> <p>{tab2} Numer operatora.</p> <p>{tab3} Zapowiedź "łączy".</p>
--	---

Ustawienia Tablicy zmiennych numerycznych

	<p>//{idx1} Tablica numerów indeks 1 -> pierwszy dzień tygodnia wolny od pracy.</p> <p>//{idx2} Tablica numerów indeks 2 -> drugi dzień tygodnia wolny od pracy.</p> <p>//Godziny pracy, rozpoczęcia {idx3} , końca {idx4}. Uwaga czas podawać poprzedzony znakiem \$ np \$170000</p> <p>//Ile cyfr DTMF oczekujemy? {idx5}.</p> <p>//{idx6} czas trwania flasha, czas należy podawać jako wielokrotność 20ms.</p> <p>//{idx7} Jak długo oczekujemy na DTMF po zakończeniu zapowiedzi.</p>
---	--

Tekst programu:

```
start: TEST_LINE_RING <f0> // czy dzwonek
      JMP_T ring // tak
      JMP start // pętla
```

R4ATNR

```
ring:  NUM_ILOAD (num0)[10]           // } led
        SET_LED (num0)
        NUM_ILOAD (num0)[30]           //załaduj opóźnienie w 1/10sek
        NUM_IMUL (num0)[100]
        SET_TIMER (tim0)(num0)         //ustaw timer
        WAIT_TIMER (tim0)              //odlicz czas

//sprawdź wolne czy praca?
//Czy godziny pracy? Programowanie godzin, rozpoczęcia {idx3} , końca {idx4}
//dni tygodnia wolne od pracy: 1 dzień {idx1}, 2 dzień {idx2}
        CLR_MARK <m0>                  //wskaźnik po godzinach
pracy
        NUM_ILOAD (num7)[1]
        GET_WEEK_DAY (num0)            //aktualny dzień tygodnia
        NUM_CLOAD (num1){idx1}        //załaduj z tablicy tekstów
        NUM_CLOAD (num2){idx2}
        TEST_NUM_COMP <f0>(num0)(num1) //czy dzień tyg wolny od pracy?
        TEST_NUM_COMP <f1>(num0)(num2) //czy dzień tyg wolny od pracy?

//sprawdź godziny
        GET_TIME_ZONE(num0)            //aktualna godzina w formacie
$hhmmss
        NUM_CLOAD (num1){idx3}        //godzina rozpoczęcia pracy
        NUM_CLOAD (num2){idx4}        //godzina zakończenia pracy
        TEST_NUM_LESS <f2>(num0)(num1) //po rozpoczęciu pracy?
        TEST_NUM_MORE <f3>(num0)(num2) //po skończeniu pracy?
        MASK_OR [0,1,2,3]             //wolne?
        SKIP_T [2]                    //ustaw praca
        SET_MARK <m0>                 //ustaw wolne
        NUM_ILOAD (num7)[2]

//odbierz wywołanie
        SET_LED (num7)                //zapal LED ziel -praca, czer
-wolne
        SET_HOOK_ON                   //zajmij linię
        NUM_ILOAD (num6)[10]          //załaduj opóźnienie w 1/10sek
        NUM_IMUL (num6)[100]
        SET_TIMER (tim0)(num6)        //ustaw timer
        WAIT_TIMER (tim0)              //odlicz czas
        SET_DSP_ON [D]                //uruchomiony odb DTMF
        SKIP_T [2]
```

R4ATNR

```
                SET_PLAY_ON {tab0}                //odtwórz nagranie dla czasu pracy
                SKIP_F [1]
                SET_PLAY_ON {tab1}                //odtwórz nagranie dla czasu
wolnego
play1: GET_DTMF (txt0)                            //czy odebrana choćby 1 cyfra?
                NUM_LTEXT (num0)(txt0)
                TEST_NUM_ZERO <f0>(num0)
                JMP_F silence
                TEST_PLAY <f0>                    //zapowiedź zakończona?
                JMP_T play1                        //nie, graj dalej

//zapowiedź skonczonea
silence: SET_PLAY_OFF
                TEST_DTMF <f0> ['0']              //sprawdź czy wybrano 0
                JMP_T oper
                NUM_CLOAD (num6){idx7}            //czas oczekiwania na wybranie
DTMF
                NUM_CLOAD (num5){idx5}            //ile cyfr oczekujemy
                NUM_ILOAD (num1)[1]
                SET_TIMER (tim0)(num6)
getdtmf: GET_DTMF (txt0)                          //ile cyfr odebrane?
                NUM_LTEXT (num0)(txt0)
                TEST_NUM_COMP <f0>(num0)(num1)   //czy tylko 1 cyfra wybrana?
                SKIP_F [2]
                TEST_DTMF <f0> ['0']              //sprawdź czy wybrano 0
                JMP_T oper
                TEST_NUM_COMP <f0>(num0)(num5)   //oczekiwana ilość cyfr?
                JMP_T complet
                TEST_TIMER <f0>(tim0)            //czas minął?
                JMP_T getdtmf
oper:  TEST_MARK <m0>                              //sprawdź czy pracujemy
                SKIP_F [3]                        //wolne przeskocz 3 rozkazy
                TEXT_CLOAD (txt0){tab2}          //załaduj numer operatora
                CALL transfer                      //skocz do podprogramu łączenia
rozmowy
                JMP start                          //koniec
                CALL stop
                JMP start

//łączenie do numeru wybranego
```

R4ATNR

```
complet: CALL transfer
                JMP start                                //koniec

//łączenie
transfer: NUM_ILOAD (num7)[4]                          //załaduj miganie lampki
                SET_LED (num7)                          //włącz lampkę
                SET_PLAY_ON {tab3}                      //odtwórz zapowiedź "łącze".
                WAIT_PLAY                                //zaczekaj do końca zapowiedzi
                NUM_CLOAD (num0){idx6}                 //załaduj czas trwania sygnału flash
                SET_FLASH (num0)                       //wygeneruj przerwę na linię
                WAIT_FLASH
                NUM_ILOAD (num0)[15]                   //odczekaj po flashu w 1/100ms
                NUM_IMUL (num0)[100]                   //
                SET_TIMER (tim0)(num0)                 //
                WAIT_TIMER (tim0)
                SET_DTMF (txt0)                        //wybierz numer w DTMF
                WAIT_DTMF
                NUM_ILOAD (num0)[15]                   //odczekaj po DTMF w 1/100ms
                NUM_IMUL (num0)[100]                   //
                SET_TIMER (tim0)(num0)                 //
                WAIT_TIMER (tim0)
stop:           SET_HOOK_OFF                           //zwolnij linię- potłącz do AB B
                NUM_ILOAD (num0)[15]                   //odczekaj w 1/100ms
                NUM_IMUL (num0)[100]                   //
                SET_TIMER (tim0)(num0)                 //
                WAIT_TIMER (tim0)
                CLR_DTMF
                NUM_ILOAD (num7)[3]                   //załaduj zgaszenie lampki
                SET_LED (num7)                          //zgaś lampkę
                NUM_ILOAD (num6)[50]
                NUM_IMUL (num6)[100]
                SET_TIMER (tim0) (num6)
                WAIT_TIMER (tim0)
                NUM_ILOAD (num7)[0]
                SET_LED (num7)
                RET                                      //koniec podprogramu
```

R4ATNR

8. Automatyczna sekretarka.

Urządzenie odbiera wywołanie po ustalonej ilości dzwonek, zapisuje przez ustalony czas lub do odłożenia słuchawki przez rozmówcę. Odłożenie słuchawki wykrywane jest przez wykrycie sygnału zajętości.

Ilość dzwonek do odebrania	<i>Tablica numerów {idx0}</i> ← proszę nie wpisywać wartości 0
Maksymalny czas nagrania	<i>Tablica numerów {idx1}</i> w sekundach
Zapowiedź sekretarki	<i>Tablica tekstów {tab3}</i>

Tekst programu:

```
// Nagrywanie rozmów z linii + sekretarka
//{idx0} - ilość dzwonek do odebrania przez sekretarkę NIE WPISYWAĆ 0 !
//{idx1} - max czas nagrania w sekundach
//{tab3} - zapowiedź sekretarki

start: NUM_CLOAD (num5){idx0} //ilość dzwonek do odebrania jako sekretarka
    TEST_LINE_RING <f0>                // czy dzwonek
    CALL_T    ring                      // tak
    TEST_LINE_CURR <f0>                // czy linia ?
    CALL_T    line_o                    // tak
    JMP start                          // pętla

    // obsługa połączenia przychodzącego
ring:   NUM_ILOAD (num0){10}           // } led
        SET_LED (num0)                 // }
        SET_DSP_ON [C]                 // włącz odbiornik clip-u
        SET_ROZM_NEW                   // nowa rozmowa
wring:  TEST_LINE_CURR <f0>           // czy odebrane ?
        JMP_T line_i                   // tak
        GET_RING (num4)                //dzwonki
        TEST_NUM_COMP <f0>(num4)(num5) //czy odebrać jako sekretarka?
        JMP_T sekretarka

        TEST_LINE_RING <f0>           // czy dzwonenie ?
        JMP_T wring                    // tak
        SET_DSP_OFF                   // wyłącz dsp
sss:   NUM_ILOAD (num0){0}            // } led
        SET_LED (num0)                 // }
        RET                             //

        ;obsługa połączenia przychodzącego
line_i: NUM_ILOAD (num0){1}           // } led
        SET_LED (num0)                 // }
        SET_DSP_OFF                   // wyłącz dsp
wline_i:TEST_LINE_CURR <f0>           // czy prąd ?
        JMP_T wline_i                  // jeszcze tak
        NUM_ILOAD (num0){0}           // } led
```

R4ATNR

```
SET_LED (num0)                                // }
RET                                            //

;obsługa połączenia wychodzącego
line_o: NUM_ILOAD (num0)[1]                    // } led
        SET_LED (num0)                        // }
wline_o:TEST_LINE_CURR <f0>                    // czy prąd ?
        JMP_T wline_o                          // jeszcze tak
        NUM_ILOAD (num0)[0]                    // } led
        SET_LED (num0)                        // }
        RET                                    //

;obsługa sekretarki
sekretarka: SET_DSP_OFF                        // wyłącz dsp
        SET_HOOK_ON                            //zajmij linię
        NUM_ILOAD (num0)[2]                    // led zielony
        SET_LED (num0)                        // zapal
        //
        NUM_ILOAD (num6){10}                  //załaduj opóźnienie w 1/10sek
        NUM_IMUL (num6){100}                  //opóźnienie 1 sekunda
        SET_TIMER (tim0)(num6)                //ustaw timer
        WAIT_TIMER (tim0)                      //odlicz czas
        SET_PLAY_ON {tab3}                    //odtwórz nagranie
sekplay: TEST_PLAY <f0>                        // zapowiedz skończona?
        JMP_T sekplay                          //trwa czekaj
        NUM_ILOAD (num0)[13]                  // led zielony miga x2
        SET_LED (num0)                        // zapal
        //
        NUM_CLOAD (num6){idx1}                //załaduj maksymalny czas nagrania w sekundach
        NUM_IMUL (num6) [100]
        NUM_IMUL (num6) [10]                  //zamień na milisekundy
        SET_TIMER (tim0)(num6)                //ustaw timer
        SET_DSP_ON [S]                        //rozpoznaj sygnał zajętości
        SET_ROZM_NEW                           // inicjujemy zapis bilingu rozmowy
        GET_RING (num0)                        // dzwonki
        PUT_RING (num0)                        //
        NUM_ILOAD (num0)[1]                    // przychodząca
        PUT_TYPE (num0)                        //
        GET_CLIP (txt0)                        // clip
        TEXT_LADD (txt0)[S]                    //dodaj do CLIPU info Sekretarka
        PUT_CLIP (txt0)
        //
        SET_REC_ON                             //uruchom nagrywanie
sekrec: TEST_SIGN <f0>[s1]                    //czy wykryto sygnał zajętości?
        TEST_SIGN <f1>[s2]                    //jW o innym tempie
        TEST_SIGN <f3>[s3]
        MASK_OR[0,1,2]
        JMP_T sekrend                           //tak
        TEST_TIMER <f1>(tim0)                  //przekroczony maksymalny czas?
        JMP_F sekrend                           //tak
```

R4ATNR

```
JMP sekrec                                     // nagrywaj dalej
//
sekrend: SET_DSP_OFF
SET_HOOK_OFF                                 //zwolnij linię
SET_REC_OFF                                  // koniec nagrania
SET_ROZM_END                                 // rozmowa !!!
NUM_ILOAD (num6)[10]                         //załaduj opóźnienie w 1/10sek
NUM_IMUL (num6)[100]                         //opóźnienie 1 sekunda
SET_TIMER (tim0)(num6)                       //ustaw timer
WAIT_TIMER (tim0)                            //odlicz czas
JMP sss
```

9. Nagrywarka plus sekretarka.

Nagrywa rozmowy przychodzące i wychodzące, dodatkowo jest sekretarką.

Wpisy do tablic jak w poprzednim przykładzie.

Tekst programu:

```
// Nagrywanie rozmów z linii + sekretarka
//{{idx0} - ilość dzwonek do odebrania przez sekretarkę NIE WPISYWAĆ 0 !
//{{idx1} - max czas nagrania w sekundach
//{{tab3} - zapowiedź sekretarki
//Rozmowy odebrane przez sekretarkę mają w numerze CLIP jako pierwszy znak literę "S"
```

```
start: NUM_CLOAD (num5){idx0} //ilość dzwonek do odebrania jako sekretarka
```

```
TEST_LINE_RING <f0>           // czy dzwonek
CALL_T ring                   // tak
TEST_LINE_CURR <f0>           // czy linia ?
CALL_T line_o                 // tak
JMP start                     // pętla
```

```
// obsługa połączenia przychodzącego
```

```
ring: NUM_ILOAD (num0)[10]    // } led
SET_LED (num0)                // }
SET_DSP_ON [C]                // włącz odbiornik clip-u
SET_ROZM_NEW                  // nowa rozmowa
wring: TEST_LINE_CURR <f0>    // czy odebrane ?
JMP_T line_i                  // tak
GET_RING (num4)               //dzwonki
TEST_NUM_COMP <f0>(num4)(num5) //czy odebrać jako sekretarka?
JMP_T sekretarka
```

```
TEST_LINE_RING <f0>          // czy dzwonicie ?
```


R4ATNR

```
JMP_T wring                // tak
SET_DSP_OFF                // wyłącz dsp
;generowanie rozmowy nieodebranej
GET_RING (num0)           // dzwonki
PUT_RING (num0)           //
NUM_ILOAD (num0)[3]       // nieodebrana
PUT_TYPE (num0)           //
GET_CLIP (txt0)           // clip
PUT_CLIP (txt0)           //
sss: SET_ROZM_END          // rozmowa !!!
    ;rozmowa wygenerowana
    NUM_ILOAD (num0)[0]    // } led
    SET_LED (num0)        // }
    RET                    //

;obsługa połączenia przychodzącego
line_i: NUM_ILOAD (num0)[1] // } led
        SET_LED (num0)     // }
        SET_DSP_OFF       // wyłącz dsp
;inicjowanie rozmowy przychodzącej
        SET_ROZM_NEW      //
        GET_RING (num0)   // dzwonki
        PUT_RING (num0)   //
        NUM_ILOAD (num0)[1] // przychodząca
        PUT_TYPE (num0)   //
        GET_CLIP (txt0)   // clip
        PUT_CLIP (txt0)   //
        SET_REC_ON        // no to nagrywamy
;rozmowa zainicjowana
wline_i: TEST_LINE_CURR <f0> // czy prąd ?
        JMP_T wline_i     // jeszcze tak
;generowanie rozmowy przychodzącej
        SET_REC_OFF       // koniec nagrania
        SET_ROZM_END      // rozmowa !!!
;rozmowa wygenerowana
        NUM_ILOAD (num0)[0] // } led
        SET_LED (num0)    // }
        RET                //

;obsługa połączenia wychodzącego
line_o: NUM_ILOAD (num0)[1] // } led
        SET_LED (num0)    // }
;inicjowanie rozmowy wychodzącej
```

R4ATNR

```
SET_ROZM_NEW //
NUM_ILOAD (num0)[0] // wychodząca
PUT_TYPE (num0) //
SET_REC_ON // no to nagrywamy
SET_DSP_ON [D] // odbiornik dtmf-u
NUM_ILOAD (num0)[100] // 10 sekund
NUM_IMUL (num0)[100] //
SET_TIMER (tim0)(num0) // czas na wybranie
TEXT_CLR (txt1) // porównywanie dtmf-u
;rozmowa zainicjowana
wline_o:GET_DTMF (txt0) //
TEST_TEXT_COMP <f0>(txt0)(txt1) // czy nowa cyfra
JMP_T no_dial // nie
;dodanie czasu aktywności odbiornika dtmf-u
SET_TIMER (tim0)(num0) // czas na wybranie
GET_DTMF (txt1) // nowy numer do porównania
no_dial:TEST_TIMER <f0>(tim0) // czy czas
JMP_T tim_on // timer aktywny
SET_DSP_OFF // wyłącz dsp
JMP wline_p // nadzoruj połączenie lub wykryj zmianę polaryzacji
tim_on: TEST_LINE_CHANG <f0> // czy zmiana polaryzacji
JMP_T line_p // do zapowiedzi
TEST_LINE_CURR <f0> // czy prąd ?
JMP_F line_e // brak, rozłączenie
JMP wline_o // jeszcze tak
;nadzór połączenia i oczekiwanie zmianę biegunowości
wline_p:TEST_LINE_CHANG <f0> // czy zmiana polaryzacji
JMP_T line_p // do zapowiedzi
TEST_LINE_CURR <f0> // czy prąd
JMP_F line_e // brak, rozłączenie
JMP wline_p // jest prąd
;zmiana biegunowości
line_p: NUM_ILOAD (num0)[3] // } led
SET_LED (num0) // }
SET_DSP_OFF // wyłącz dsp
;nadzór połączenia
wline_t:TEST_LINE_CURR <f0> // czy prąd ?
JMP_T wline_t // jeszcze tak
line_e: SET_DSP_OFF // wyłącz dsp
;generowanie rozmowy wychodzącej
GET_DTMF (txt0) // dtmf
PUT_DTMF (txt0) //
SET_REC_OFF // koniec nagrania
```

R4ATNR

```
SET_ROZM_END                // rozmowa !!!
;rozmowa wygenerowana
NUM_ILOAD (num0)[0]         // } led
SET_LED (num0)              // }
RET                          //

;obsługa sekretarki
sekretarka: SET_DSP_OFF     // wyłącz dsp
SET_HOOK_ON //zajmij linię
NUM_ILOAD (num0)[2]         // led zielony
SET_LED (num0)              // zapal
//
NUM_ILOAD (num6)[10]        //załaduj opóźnienie w 1/10sek
NUM_IMUL (num6)[100]        //opóźnienie 1 sekunda
SET_TIMER (tim0)(num6)      //ustaw timer
WAIT_TIMER (tim0)           //odlicz czas
SET_PLAY_ON {tab3}          //odtwórz nagranie
sekplay: TEST_PLAY <f0>     // zapowiedź skończona?
JMP_T sekplay               //trwa czekaj
NUM_ILOAD (num0)[13]        // led zielony miga x2
SET_LED (num0)              // zapal
//
NUM_CLOAD (num6){idx1}     //załaduj maksymalny czas nagrania w sekundach
NUM_IMUL (num6) [100]
NUM_IMUL (num6) [10]        //zamień na milisekundy
SET_TIMER (tim0)(num6)      //ustaw timer
SET_DSP_ON [S]              //rozpoznaj sygnał zajętości
SET_ROZM_NEW                // inicjujemy zapis bilingu rozmowy
GET_RING (num0)             // dzwonki
PUT_RING (num0)             //
NUM_ILOAD (num0)[1]         // przychodząca
PUT_TYPE (num0)             //
GET_CLIP (txt0)             // clip
TEXT_LADD (txt0) ['S']      //dodaj do CLIPU info Sekretarka
PUT_CLIP (txt0)
//
SET_REC_ON                  //uruchom nagrywanie
sekrec: TEST_SIGN <f0>[s1]   //czy wykryto sygnał zajętości?
        TEST_SIGN <f1>[s2] //jW o innym tempie
        TEST_SIGN <f3>[s3]
        MASK_OR[0,1,2]
JMP_T sekrend               //tak
TEST_TIMER <f1>(tim0)       //przekroczony maksymalny czas?
```

R4ATNR

```
JMP_F sekrend //tak
JMP sekrec // nagrywaj dalej
//
sekrend: SET_DSP_OFF
SET_HOOK_OFF //zwolnij linie
SET_REC_OFF // koniec nagrania
NUM_ILOAD (num6)[10] //załaduj opoznienie w 1/10sek
NUM_IMUL (num6)[100] //opoznienie 1 sekunda
SET_TIMER (tim0)(num6) //ustaw timer
WAIT_TIMER (tim0) //odlicz czas
JMP sss
```