Spis instrukcji CC4 z uwzględnieniem wersji programu i silnika :

```
//                                                  - komentarz tylko w linii

if (wyrażenie warunek wyrażenie) "polecenie"       - polecenie warunkowe

               warunki  =    - równe
                        >    - większe
                        <    - mniejsze
                        <>   - różne
                        >=   - większe lub równe
                        <=   - mniejsze lub równe
                        ~    - string zawiera znak(i) (dtmf~'123')
                        ^    - string odpowiada masce (lcd^'061*')
                        %    - string zawiera string  (dtmf%'123')

label=?                                            - etykieta skoku

goto=?                                             - rozkaz skoku do etykiety

goto=str1                                          - rozkaz skoku pod etykietę ze zmiennej

on_error                                           - do tego miejsca skok w przypadku błędu w
                                                     programie

engine=x.x                                         - wymagana wersja CC4 dla wykonania programu

hook_on                                            - dołącz się do linii

hook_off                                           - odłącz się od linii

record_on                                          - uruchom nagrywanie

record_off                                         - zakończ nagrywanie

play_on ('plik.wav')                               - uruchom odtwarzanie, plik do odtworzenia musi
                                                     znajdować się w katalogu wskazanym jako
                                                     katalog komunikatów

play_off                                           - zakończ odtwarzanie

play_str (zmienna_str,'selektor języka')           - uruchom odtwarzanie zawartości zmiennej
                                                     ( czytanie cyfr i liter ) wymaga plików o z
                                                     nagraniami znaków np. 1.wav, a.wav, dla selek-
                                                     tora języka 'pl' np. 1pl.wav, apl.wav.

dial_sys (klawisz byte)                            - wyslij polecenie do kanału

dial_dtmf (zmienna_str)                            - wybierz numer w dtmf-ie ( Uwaga ! Wymaga
                                                     plików 1dtmf.wav itd. w katalogu komunikatów )

clr_dtmf                                           - kasuje bufor odbioru dtmf-u

clr_stoper1                                        - zeruje stoper 1 odmierzający czas

clr_stoper2                                        - zeruje stoper 2 odmierzający czas

clr_stoper3                                        - zeruje stoper 3 odmierzający czas

clr_stoper4                                        - zeruje stoper 4 odmierzający czas

clr_stoper5                                        - zeruje stoper 5 odmierzający czas

wait_timer1                                        - czekaj do końca timera 1

wait_timer2                                        - czekaj do końca timera 2

wait_timer3                                        - czekaj do końca timera 3

wait_timer4                                        - czekaj do końca timera 4

wait_timer5                                        - czekaj do końca timera 5

wait_play                                          - czekaj do końca odtwarzania

restart                                            - uruchom program od początku

debugprint (zmienna)                               - wyświela wartość zmiennej w oknie debugowania
                                                     i wstrzymuje program ( wymaga włączenia prze-
                                                     łącznika DEBUG )
```

clr_semaphore01 .. clr_semaphore32 - wyzeruj semafor 1..32

init_semaphore - wyzeruj wszystkie semafory

get_semaphore01 .. get_semaphore32 - sprawdź i pobierz semafor
w instrukcji if (get_semaphore01) goto=semafor
pobrany

is_record - czy nagrywanie bool

is_play - czy odtwarzanie bool

is_hook - stan linii bool

is_find - wynik przeszukania bazy numerów bool

is_file {'filename'} - czy jest taki plik bool

dtmf - wybrane cyfry w dtmf-ie str

clip - numer clip str

to_dial - numer do wybrania z bazy str

dtmf_cnt - ilość wybranych cyfr w dtmf-ie byte

ring - stan dzwonienia byte

hour - aktualna godzina byte

minute - aktualna minuta byte

time - godzina w minutach word

year - aktualny rok word

month - aktualny miesiąc byte

day - aktualny dzień byte

day_of_week - dzień tygodnia byte
1 - niedziela

date - dzisiejsza data w formacie 'rrrrmmdd' str

level_rec - poziom sygnału byte

stoper1 - stoper podaje czas w sekundach od ostatniego zerowania

stoper2 - stoper podaje czas w sekundach od ostatniego zerowania

stoper3 - stoper podaje czas w sekundach od ostatniego zerowania

stoper4 - stoper podaje czas w sekundach od ostatniego zerowania

stoper5 - stoper podaje czas w sekundach od ostatniego zerowania

{ uwaga ! stopery można zmniejszać instrukcją dec (stoper1,milisekundy) }

timer1=? - timer sekundowy 1

timer2=? - timer sekundowy 2

timer3=? - timer sekundowy 3

timer4=? - timer sekundowy 4

timer5=? - timer sekundowy 5

is_bool1=? - przełącznik 1

is_bool2=? - przełącznik 2

```
  is_bool3=?                                    - przełącznik 3
  is_bool4=?                                    - przełącznik 4
  is_bool5=?                                    - przełącznik 5
  var1=?                                        - zmienna lokalna 1
  var2=?                                        - zmienna lokalna 2
  var3=?                                        - zmienna lokalna 3
  var4=?                                        - zmienna lokalna 4
  var5=?                                        - zmienna lokalna 5
  str1=?                                        - zmienna tekstowa lokalna 1
  str2=?                                        - zmienna tekstowa lokalna 2
  str3=?                                        - zmienna tekstowa lokalna 3
  str4=?                                        - zmienna tekstowa lokalna 4
  str5=?                                        - zmienna tekstowa lokalna 5
  gl_bool1=?                                    - przełącznik globalny 1
  gl_bool2=?                                    - przełącznik globalny 2
  gl_bool3=?                                    - przełącznik globalny 3
  gl_bool4=?                                    - przełącznik globalny 4
  gl_bool5=?                                    - przełącznik globalny 5
  gl_var1=?                                     - zmienna globalna 1
  gl_var2=?                                     - zmienna globalna 2
  gl_var3=?                                     - zmienna globalna 3
  gl_var4=?                                     - zmienna globalna 4
  gl_var5=?                                     - zmienna globalna 5
  inc (zmienna)                                 - zwiększ zmienną var o 1
  inc (zmienna,wartość)                         - zwiększ zmienną var o wartość
  dec (zmienna)                                 - zmniejsz zmienną var o 1
  dec (zmienna,wartość)                         - zmniejsz zmienną var o wartość
  { uwaga !  instrukcją dec (stoper1,5000) => można zmniejszyć wartość naliczoną przez stoper o 5
sekund }
  save_str ('filename',zmienna_str)             - zapisz zmienną tekstową do pliku
  load_str ('filename',zmienna_str)             - odczytaj zmienną tekstową z pliku

  false         = 0                             - stała oznaczająca fałsz
  true          = 1                             - stała oznaczająca prawdę
  b_flash100ms                                   - flash 100 ms
  b_flash200ms                                   - flash 200 ms
  b_flash300ms                                   - flash 300 ms
  b_flash400ms                                   - flash 400 ms
  b_flash500ms                                   - flash 500 ms
  b_flash600ms                                   - flash 600 ms
  b_flash700ms                                   - flash 700 ms
```

b_flash800ms	- flash 800 ms
b_flash900ms	- flash 900 ms
roz_out	- dodaj rozmowę wychodzącą
roz_in	- dodaj rozmowę przychodzącą
roz_outb	- dodaj rozmowę niedoszłą
roz_inb	- dodaj rozmowę nieodebraną
roz_kom (opis)	- dodaj komunikat
roz_nr=?	- numer wybrany
roz_ab=?	- numer abonenta
roz_ko=?	- kod konta
roz_tm=?	- czas trwania
roz_nt=?	- notatka
iocontrol1 ($hh)	- ustawia stan portu lpt1
file_delete ('filename')	- kasuje plik
email_r_mail=?	- adres odbiorcy poczty
email_r_subj=?	- temat poczty
email_r_body=?	- treść poczty
email_r_atta=?	- plik załącznika
email_s_mail=?	- nadawca poczty
email_s_name=?	- nadawca poczty
email_s_user=?	- konto nadawcy
email_s_pass=?	- hasło nadawcy
email_s_host=?	- serwer smtp
email_s_auth=?	- autoryzacja serwera
record_off+mail	- zakończ nagrywanie i wyślij mailem
roz_out+mail	- dodaj rozmowę wychodzącą i wyślij mailem
roz_in+mail	- dodaj rozmowę przychodzącą i wyślij mailem
roz_outb+mail	- dodaj rozmowę niedoszłą i wyślij mailem
roz_inb+mail	- dodaj rozmowę nieodebraną i wyślij mailem
roz_kom+mail (opis)	- dodaj komunikat i wyślij mailem
roz_out+mail+link	- dodaj rozmowę wychodzącą, połącz z nagraniem i wyślij mailem
roz_in+mail+link	- dodaj rozmowę przychodzącą, połącz z nagraniem i wyślij mailem
roz_outb+mail+link	- dodaj rozmowę niedoszłą, połącz z nagraniem i wyślij mailem
roz_inb+mail+link	- dodaj rozmowę nieodebraną, połącz z nagraniem i wyślij mailem
roz_kom+mail+link (opis)	- dodaj komunikat, połącz z nagraniem i wyślij mailem
roz_out+link	- dodaj rozmowę wychodzącą i połącz z nagraniem
roz_in+link	- dodaj rozmowę przychodzącą i połącz z nagraniem

roz_outb+link - dodaj rozmowę niedoszłą i połącz z nagraniem

roz_inb+link - dodaj rozmowę nieodebraną i połącz z nagraniem

roz_kom (opis)+link - dodaj komunikat i połącz z nagraniem

Polecenia od engine 1.4 ----------

// zmienne tekstowe mogą zawierać duże litery i spacje

play_bin (zmienna_bin,'selektor języka') - uruchom odtwarzanie zmiennej binarnej
wymaga plików z nagraniami tak jak play_str

is_bool6=? - przełącznik 6

is_bool7=? - przełącznik 7

is_bool8=? - przełącznik 8

is_bool9=? - przełącznik 9

is_bool0=? - przełącznik 0

var6=? - zmienna lokalna 6

var7=? - zmienna lokalna 7

var8=? - zmienna lokalna 8

var9=? - zmienna lokalna 9

var0=? - zmienna lokalna 0

str6=? - zmienna tekstowa lokalna 6

str7=? - zmienna tekstowa lokalna 7

str8=? - zmienna tekstowa lokalna 8

str9=? - zmienna tekstowa lokalna 9

str0=? - zmienna tekstowa lokalna 0

gl_bool6=? - przełącznik globalny 6

gl_bool7=? - przełącznik globalny 7

gl_bool8=? - przełącznik globalny 8

gl_bool9=? - przełącznik globalny 9

gl_bool0=? - przełącznik globalny 0

gl_var6=? - zmienna globalna 6

gl_var7=? - zmienna globalna 7

gl_var8=? - zmienna globalna 8

gl_var9=? - zmienna globalna 9

gl_var0=? - zmienna globalna 0

Polecenia od engine 1.5 ----------

level_max - poziom sygnału maksymalny byte (readonly)

clr_level_max - wyzeruj poziom sygnału

gl_str1=? - zmienna tekstowa globalna 1

gl_str2=? - zmienna tekstowa globalna 2

gl_str3=? - zmienna tekstowa globalna 3

gl_str4=? - zmienna tekstowa globalna 4

gl_str5=? - zmienna tekstowa globalna 5

gl_str6=? - zmienna tekstowa globalna 6

gl_str7=? - zmienna tekstowa globalna 7

gl_str8=? - zmienna tekstowa globalna 8

gl_str9=? - zmienna tekstowa globalna 9

gl_str0=? - zmienna tekstowa globalna 0

gl_str{0..99}=? - zmienna tekstowa globalna 0..99

gl_bool{0..99}=? - przełączniki globalne 0..99

gl_date0=? - data globalna 0

gl_date1=? - data globalna 1

gl_date2=? - data globalna 2

gl_date3=? - data globalna 3

gl_date4=? - data globalna 4

gl_date5=? - data globalna 5

gl_date6=? - data globalna 6

gl_date7=? - data globalna 7

gl_date8=? - data globalna 8

gl_date9=? - data globalna 9

gl_date{0..99}=? - daty globalne 0..99

rec_pause_on - wstrzymaj nagrywanie

rec_pause_off - uruchom nagrywanie

level_mid - poziom sygnału średni byte (readonly)

clr_level_mid - wyzeruj poziom średniego

gosub=? - rozkaz do podprogramu możliwe 16 poziomów podprogramów

return - powrót z podprogramu możliwe 16 poziomów podprogramów

Polecenia od engine 1.6 ----------

sms_out (port,'pin','numer','text') - wysłanie sms-a, wymaga biblioteki dll i zainstalowanego modemu GSM

Polecenia od engine 1.7 ----------

logprint (zmienna) - wysyła komunikat do logu

gl_var{0..99}=? - zmienna globalna 0..99

Wprowadzona możliwość adresowania zmiennej zmienną tak jak w przykładzie poniżej

gl_var{var0}=?

chanel - numer aktualnego programu

line_state{1..32} - stan linii z M3

port_state{1..32} - stan linii z R3

line_state - stan linii dla bieżącego portu

port_state - stan linii dla bieżącego portu

roz_nr{1..32}=? - numer wybrany

roz_ab{1..32}=? - numer abonenta

roz_ko{1..32}=? - kod konta

roz_tm{1..32}=? - czas trwania

gl_record{1..32} - czy nagrywanie bool

gl_play{1..32} - czy odtwarzanie bool

gl_hook{1..32} - stan linii bool

isdn_clip{1..32} - ostatni clip odczytany z linii isdn str

isdn_ddi_msn{1..32} - ostatni ddi lub msn odczytany z linii isdn str

Polecenia od engine 1.8 ----------

cti_ext_on (nr_wewnętrzny) - zajęcie wewnętrznego

cti_ext_off (nr_wewnętrzny) - zwolnienie wewnętrznego

cti_ext_dial (nr_wewnętrzny,cyfry) - wybranie cyfr

cti_ext_state {nr_wewnętrzny} - stan numeru wewnętrznego
stany : 0 - wolny
1 - zajęty
2 - połączony
3 - dzwoni
Uwaga ! w zmiennych będą dostępne dalsze informacje

cti_ext_numb - zmienna str numer wybrany lub clip

cti_ext_ddi - zmienna str numer ddi lub wzywany

cti_co_state {nr_portu} - stan portu miejskiego
stany : 0 - wolny
1 - zajęty
2 - połączony
3 - dzwoni
Uwaga ! w zmiennych będą dostepne dalsze informacje

cti_new_connect - odczytuje informacje o nowym połączeniu w centrali (cti)

stany : False - brak informacji
True - informacja w zmiennych

aktualizuje zmienne : cti_con_type;cti_con_ext1;cti_con_ext2;cti_con_co1;cti_con_co2;cti_con_time;cti_con_dir

cti_ext_connect {nr_wewnętrzny} - odczytuje informacje o połączeniu dla wewnętrznego

stany : False - brak informacji (połączenia)
True - informacja w zmiennych (jest połączenie)

cti_co_connect {nr_portu} - odczytuje informacje o połączeniu dla portu miejskiego

stany : False - brak informacji (połączenia)
True - informacja w zmiennych (jest połączenie)

cti_con_type - typ połączenia : 0 - ext<->ext
1 - ext<->co
2 - co<->co

cti_con_ext1 - numer dołączony 1

cti_con_ext2 - numer dołaczony 2

cti_con_co1 - port dołączony 1

cti_con_co2 - port dołaczony 2

cti_con_time - aktualny czas trwania połączenia

cti_con_dir - kierunek połączenia : 0 - wychodzące
1 - przychodzące

cc4_ext - zwraca numer wewnętrzny przypisany do kanału CC4

cc4_con - zwraca nazwę linii miejskiej przypisaną do kanału

Polecenia od engine 1.9 ----------

str_del (zmienna_str0-9,pozycja_znaku_do_usunięcia,ilość znaków) - dotyczy lokalnych zmiennych str0 .. str9

sql_connect - dołączenie do bazy danych

sql_create_db ('nazwa bazy') - utworzenie bazy danych

sql_select_db ('nazwa bazy') - wybranie pazy danych

sql_drop_db ('nazwa bazy') - usunięcie bazy danych

sql_close - odłączenie od bazy danych

sql_quwery: tutaj zapytanie/polecenie SQL - wykonuje polecenie/zapytanie sql
{var1} {str1} - tak wpisane zmienne zostaną zamienione na wartości zmiennych

sql_first - ustawia na pierwszy wiersz odpowiedzi

sql_next - następny wiersz odpowiedzi

sql_prior - poprzedni wiersz odpowiedzi

sql_last - ostatni wiersz odpowiedzi

is_sql - czy jest połączenie z bazą danych bool

sql_fieldscount - zwraca ilość pól odpowiedzi var

sql_rowscount - zwraca ilość wierszy odpowiedzi var

sql_field_var {numer pola} - zwraca pole jako zmienną var

sql_field_str {numer pola} - zwraca pole jako zmienną str

clr_event - skasuj zdarzenia

is_event_loop - wykryto prąd w łączu bool

is_event_noloop - wykryto brak prądu w łączu bool

is_event_busy - wykryto sygnał zajętości (tylko przy nagrywaniu)

is_event_alert - wykryto sygnał ostrzegawczy (tylko przy nagrywaniu)

Polecenia od engine 2.0 ----------

cti_ext_exec (nr_wewnętrzny,cyfry,akcja) - wybranie cyfr, transfer, hold itp.

play_autostop - zmienna bool sterująca stanem zatrzymywania odtwarzania od wykrycia dtmf-u

info (zmienna) - wyświela wartość zmiennej w oknie info (karta powiązania)

b_drop = 0 - rozłączenie po tapi

b_ansfer = 1 - odebranie po tapi

b_dial = 2 - wybranie numeru po tapi

b_hold = 3 - hold po tapi

b_unhold = 4 - unhold po tapi

b_park = 5 - park po tapi

b_unpark = 6 - unpark po tapi

b_trans_set = 7 - przygotowanie transferu po tapi

b_trans_comp = 8 - zakończenie transferu po tapi

float0=? - zmienna (zmiennoprzecinkowa)

```
float1=?                                          - zmienna (zmiennoprzecinkowa)
float2=?                                          - zmienna (zmiennoprzecinkowa)
float3=?                                          - zmienna (zmiennoprzecinkowa)
float4=?                                          - zmienna (zmiennoprzecinkowa)
float5=?                                          - zmienna (zmiennoprzecinkowa)
float6=?                                          - zmienna (zmiennoprzecinkowa)
float7=?                                          - zmienna (zmiennoprzecinkowa)
float8=?                                          - zmienna (zmiennoprzecinkowa)
float9=?                                          - zmienna (zmiennoprzecinkowa)
play_curr (zmienna_float)                         - odczytuje cenę, wymaga plików wav
sql_field_float {numer pola}                      - zwraca pole jako zmienną float
Polecenia od engine 2.1 ---------------------------------------------------
play_link ('plik1.wav','plik2.wav','plik3.wav'...   )    - łączy nagrania i odtwarza razem
on_usb_err                                         - do tego miejsca skok po błędzie usb
on_tapi_err                                        - do tego miejsca skok po błędzie tapi
on_sql_err                                         - do tego miejsca skok po błędzie sql
b_clear_station = 254                              - wyzerowanie numeru
b_clear_all     = 255                              - wyzerowanie całego tapi
restore                                            - powrót do programu po obsłudze błedu
break=                                             - wyskoczenie z podprogramu do etykiety
played                                             - ile czasu miało odtworzone nagranie
sign_rst                                           - wyślij reset do nagrywarki
peerip                                             - numer zdalnego ip programu CRM
peerport                                           - numer portu zdalnego programu CRM
peeropen                                           - numer portu w programie CRM do komunikacji
peerkey                                            - numer klucza zalogowanego użytkownika
```

Opis języka programowania CC4:

Założenia :

- każdy kanał może mieć swój program sterujący.
- niektóre zmienne mogą być dostępne dla wszystkich programów co umożliwia komunikację między programową lub sterowanie kilku programów jednym zdarzeniem.
- program jest wykonywany sekwencyjnie dlatego w jednej linii poza poleceniami warunkowymi nie można wpisywać więcej niż jednej instrukcji.
- zdarzenia sygnalizowane zmiennymi są całkowicie asynchroniczne względem programu dlatego zawartość zmiennej np. reprezentującej odebrane cyfry w dtmf-ie może zmienić się pomiędzy instrukcjami, należy mieć tego świadomość
- prędkość wykonywania programu wynosi 200 instrukcji/sekundę dla każdego kanału (czyli 25'600 instrukcji/sekundę) dla wszystkich kanałów.
- wykonanie niektórych instrukcji może trwać dość długo np. Zapytania do bazy danych, w takiej sytuacji wstrzymanie wykonania programu dotyczy tylko jednego kanału
- kontrola programu wykrywa tylko błędy składni, a nie logiki programu

Zmienne :

W programach możemy posługiwać się zmiennymi które występują w kilku typach :

- zmienne *var* reprezentujące dane stałoprzecinkowe np. Dla odliczania ilości wykonania pętli
- zmienne *str* przechowujące teksty i cyfry np. Numer do wybrania
- zmienne *bool* przechowujące dane typu prawda/fałsz, umożliwiają realizację np. Przełączników
- zmienne *float* reprezentują dane zmiennoprzecinkowe np. Informację o cenie produktu itp.
- zmienne *date* przechowują daty

Zmienne lokalne :

var0 .. var9 - zmienna lokalna (stałoprzecinkowa)

przykład:

var1=5

str0 .. str9 - zmienna tekstowa lokalna

przykład:

str3='0618358600'

is_bool0 .. is_bool9 - zmienna bool lokalna

przykład:

is_bool9=true

float0 .. float9 - zmienna lokalna (zmiennoprzecinkowa)

przykład

float5=37.24

Zmienne globalne, widoczne przez wszystkie programy :

gl_var0 .. gl_var9 - zmienna globalna (stałoprzecinkowa)

przykład:

gl_var1=3

gl_str0 .. gl_str9 - zmienna tekstowa globalna

przykład:

gl_str1='etykieta'

gl_bool0 .. gl_bool9 - zmienna bool globalna

przykład:

gl_bool3=false

gl_date0 .. gl_date9 - globalna data

przykład:

gl_date0='20070102'

Zmienne globalne, dostęp indeksowany :

Do zmiennych globalnych indeksowanych można uzyskać dostęp wpisując indeks jako stałą lub indeksując przy pomocy zmiennej var.

gl_var{0..99} - zmienna globalna 0..99

przykład:

gl_var{1}=50

jest równoważne

gl_var1=50

czy też

var0=1
gl_var1{var0}=50

gl_str{0..99} - zmienna tekstowa globalna 0..99

przykład:

gl_str{70}='0618358600'

  gl_bool{0..99}   - przełączniki globalne 0..99

przykład:

```
var0=30
gl_bool{var0}=true
```

  gl_date{0..99}   - daty globalne 0..99

przykład:

```
gl_date{0}='20070101'
```

Zmienne specjalne (timery i stopery):

Zmienne te umożliwiają kontrolę czasu w programach.

  timer1 .. timer5   - zmienne których wartość zmniejsza się z rozdzielczością 1ms

  wait_timer1 .. wait_timer5   - instrukcja czeka do wyzerowania timera

przykład: oczekiwanie przez 5 sekundowego

```
timer1=5000
wait_timer1
```

lub

```
timer1=5000
label=petla
if (timer1>0) goto=petla
```

  stoper1 .. stoper5   - stoper podaje czas w sekundach od ostatniego zerowania

  clr_stoper1 .. clr_stoper5   - zerowanie stopera

przykład: pomiar czasu wykonania podprogramu

```
clr_stoper1
gosub=podprogram
debugprint (stoper1)
```

Zmienne specjalne (nagrywanie, odtwarzanie, zajętość linii)

  is_record   - czy nagrywanie   bool

przykład:

```
if (is_record) gosub=informacja_rozmowa_nagrywana
```

  is_play   - czy odtwarzanie   bool

przykład:

```
play_on ('Zapowiedź.wav')
label=czekaj
if (is_play) goto=czekaj
```

  is_hook   - stan linii   bool

przykład:

```
if (is_hook) hook_off
```

  gl_record{1..32}   - czy nagrywanie ale z innego kanału

przykład:

```
var0=3
if (gl_record{var0}) debugprind ('Kanał 3 nagrywa')
```

  gl_play{1..32}   - czy odtwarzanie ale z innego kanału

przykład:

```
if (gl_play{20}) debugprint ('Kanał 20 odtwarza')
```

gl_hook{1..32} - stan linii ale z innego kanału

przykład:

```
if (gl_hook{30}) goto=kanal30zajety
```

Zmienne specjalne (sygnały akustyczne i dzwonienie)

dtmf - wybrane cyfry w dtmf-ie str

przykład:

```
if (dtmf='101') goto=do_sekretariatu
```

dtmf_cnt - ilość wybranych cyfr w dtmf-ie byte

przykład:

```
label=czekaj_na_4_cyfry
if (dtmf_cnt<4) goto=czekaj_na_4_cyfry
```

clr_dtmf - kasuje bufor odbioru dtmf-u

Zmienne specjalne (stan kanału)

ring - stan dzwonienia byte

przykład:

```
if (ring<>0) goto=odbierz
restart
label=odbierz
```

level_rec - poziom sygnału chwilowy byte

zmienna zwraca prawdziwą wartość tylko jeżeli jest uruchomione nagrywanie

przykład:

```
if (level_rec>64) goto=glosno
```

level_max - poziom sygnału maksymalny byte (readonly)

zmienna zwraca prawdziwą wartość tylko jeżeli jest uruchomione nagrywanie

clr_level_max - wyzeruj poziom sygnału

przykład:

```
clr_level_max
timer1=5000
wait_timer1
if (level_max>64) goto=byl_sygnal
```

level_mid - poziom sygnału średni byte (readonly)

zmienna zwraca prawdziwą wartość tylko jeżeli jest uruchomione nagrywanie

clr_level_mid - wyzeruj poziom średniego

przykład:

```
clr_level_mid
timer1=5000
wait_timer1
if (level_mid>32) goto=sredni_sygnal_w_normie
```

Zmienne specjalne (data, czas, dzień tygodnia)

hour - aktualna godzina byte

przykład:

```
if (hour>=16) goto=firma_nieczynna
```

minute - aktualna minuta byte

przykład:

```
if (hour=16) if (minute>30) goto=jest_po_1630
```

  time - godzina w minutach word

przykład:

```
if (time<480) goto=przed_praca
if (time>960) goto=po_pracy
```

  year - aktualny rok word

  month - aktualny miesiąc byte

  day - aktualny dzień byte

przykład:

```
if (year=2008) if (month=1) if (day=1) play_on ('Witamy_w_nowym_roku.wav')
```

  day_of_week - dzień tygodnia byte
  1 - niedziela

przykład:

```
if (day_of_week=1) play_on ('W_niedziele_nie_pracujemy.wav')
```

  date - dzisiejsza data w formacie 'rrrrmmdd' str

przykład:

```
if (date='20071224') play_on ('Dzisiaj_pracujemy_do_godziny_14.wav')
if (date='20071225') play_on ('Dzisiaj_nieczynne_zapraszamy_27.wav')
if (date='20071226') play_on ('Dzisiaj_nieczynne_zapraszamy_27.wav')
if (is_play=false) play_on ('Witamy'wav')
```

Zmienne specjalne (stan portów M3 i R3)

  line_state - stan linii dla bieżącego portu z M3

przykład:

```
if (line_state=0) goto=linia_wolna
```

  port_state - stan linii dla bieżącego portu z R3

przykład:

```
if (port_state=3) goto=zmiana_polaryzacji
```

  line_state{1..32} - stan linii z M3

indeksowany odczyt stanu linii

  port_state{1..32} - stan linii z R3

indeksowany odczyt stanu portu

Zmienne pomocnicze

  chanel - numer aktualnego programu

przykład:

```
if (chanel=1) debugprint ('Śledzenie programu tylko dla kanału 1')
```

  cc4_ext - zwraca numer wewnętrzny przypisany do kanału CC4

przykład:

```
if (cc4_ext='101') debugprint ('Przypisany numer 101')
```

  cc4_con - zwraca nazwę linii miejskiej przypisaną do kanału

```
if (cc4_con='10') debugprint ('Przypisany do linii 11')
```

Zmienne specjalne (generowanie billingu)

roz_nr=? - numer wybrany

roz_ab=? - numer abonenta

roz_ko=? - kod konta

roz_tm=? - czas trwania

roz_nt=? - notatka

przykład:

```
roz_nr=dtmf
roz_ab=cc4_ext
roz_ko=''
roz_tm=stoper1
roz_nt='Dodane przez CC4'
roz_out
```

roz_nr{1..32}=? - numer wybrany

indeksowany dostęp do numeru wybranego

roz_ab{1..32}=? - numer abonenta

indeksowany dostęp do numeru abonenta

roz_ko{1..32}=? - kod konta

indeksowany dostęp do kodu konta

roz_tm{1..32}=? - czas trwania

indeksowany dostęp do czasu trwania połączenia

Zmienne specjalne (wysyłanie poczty elektornicznej)

email_r_mail=? - adres odbiorcy poczty

email_r_subj=? - temat poczty

email_r_body=? - treść poczty

email_r_atta=? - plik załącznika

email_s_mail=? - nadawca poczty

email_s_name=? - nadawca poczty

email_s_user=? - konto nadawcy

email_s_pass=? - hasło nadawcy

email_s_host=? - serwer smtp

email_s_auth=? - autoryzacja serwera

przykład:

```
email_r_mail='odbiorca@serwer.com.pl'
email_r_subj='Masz nową wiadomość'
email_r_body='Kliknij za załącznik aby odsłuchać nagranie'
email_r_atta='C:\Zestawienie.txt'
email_s_mail='cc4@system.pl'
email_s_name='cc4'
email_s_user='cc4@system.pl'
email_s_pass='hasło'
email_s_host='system.pl'
email_s_auth=true
record_off+mail
```

Sterowanie przepływem programu

label= - etykieta skoku

goto= - rozkaz skoku do etykiety

goto=str1 - rozkaz skoku pod etykietę ze zmiennej

przykład:

```
label=czekaj
if (ring=0) goto=czekaj
```

czy też

```
if (day_of_week=1) str1='niedzela'
if (day_of_week=6) str1='sobota'
if (str1<>'') goto=str1

|

label=niedziela
debugprint ('Jest niedziela')

|

label=sobota
debugprint ('Jest sobota')
```

`on_error` - do tego miejsca skok w przypadku błędu w programie

przykład:

```
on_error
debugprint ('W programie wystąpił błąd')
```

`on_usb_err` - do tego miejsca skok po błędzie usb

przykład:

```
on_usb_err
debugprint ('Sprawdź połączenie USB')
```

`on_tapi_err` - do tego miejsca skok po błędzie tapi

przykład:

```
on_tapi_err
debugprint ('Sprawdź połączenie TAPI')
```

`on_sql_err` - do tego miejsca skok po błędzie sql

przykład:

```
on_sql_err
debugprint ('Sprawdź bazę danych')
```

`restart` - uruchom program od początku

przykład:

```
if (ring>0) goto=odbierz
restart
label=odbierz
```

`gosub=` - rozkaz do podprogramu możliwe 16 poziomów podprogramów

przykład:

```
str1='Info1.wav'
gosub=odtworz_zapowiedź

|

label=odtworz_zapowiedz
play_on (str1)
wait_play
return
```

`return` - powrót z podprogramu możliwe 16 poziomów podprogramów

`restore` - powrót do programu po obsłudze błedu

przykład:

```
on_sql_err
sql_connect
restore
```

```
  break=                                          - wyskoczenie z podprogramu do etykiety
```

przykład:

```
str1='Info1.wav'
gosub=odtworz_zapowiedz

|

label=zakoncz_zapowiedz
play_off

|

label=odtworz_zapowiedz
play_on (str1)
label=odtwarzanie
if (dtmf='#') break=zakoncz_zapowiedz
if (is_play) goto=odtwarzanie
return
```

```
  if (wyrażenie warunek wyrażenie) "polecenie"      - polecenie warunkowe

                warunki  =    - równe
                         >    - większe
                         <    - mniejsze
                         <>   - różne
                         >=   - większe lub równe
                         <=   - mniejsze lub równe
                         ~    - string zawiera znak(i) (dtmf~'123')
                         ^    - string odpowiada masce (lcd^'061*')
                         %    - string zawiera string  (dtmf%'123')
```

przykład:

```
if (var1>0) goto=petla

if (ring<>0) goto=dzwonienie

if (dtmf~'#') goto=odebrano_potwierdzenie

if (dtmf^'06183586**') goto=numer_wykryty
```

Sterowanie dołączeniem do linii

```
  hook_on                                         - dołącz się do linii

  hook_off                                        - odłącz się od linii
```

przykład:

```
if (ring>0) goto=odbierz
restart
label=odbierz
hook_on
play_on ('Powitanie.wav')
wait_play
hook_off
restart
```

Sterowanie odtwarzaniem komunikatów

```
  play_on ('plik.wav')                            - uruchom odtwarzanie, plik do odtworzenia musi
                                                    znajdować się w katalogu wskazanym jako
                                                    katalog komunikatów
```

przykład:

```
play_on ('Powitanie')
```

```
  play_off                                        - zakończ odtwarzanie
```

przykład:

```
if (is_play) play_off
```

play_str (zmienna_str,'selektor języka') - uruchom odtwarzanie zawartości zmiennej

Uwaga ! W katalogu komunikatów muszą znajdować się pliki reprezentujące znaki występujące w zmiennej tekstowej

dla 1,2,3,4,a,b będzie to 1.wav, 2.wav, 3.wav, 4.wav, a.wav, b.wav

jeżeli jest podany selektor języka to pliki puszą posiadać w nazwie selektor języka

dla pl będzie to 1pl.wav, 2pl.wav, 3pl.wav, 4pl.wav, apl.wav, bpl.wav

dla en będzie to 1en.wav, 2en.wav, 3en.wav, 4en.wav, aen.wav, ben.wav

przykład:

```
str1='12ac7'
gosub=odtworz_kod_dostepu

|

play_str (str1)
wait_play
return
```

play_bin (zmienna_bin,'selektor języka') - uruchom odtwarzanie zmiennej binarnej

Uwaga ! W katalogu komunikatów muszą znajdować się pliki reprezentujące cyfry z lub bez selektora języka w zależności od potrzeb
Maksymalna czytana liczba to 9999

0.wav, 1.wav, 2.wav, 3.wav, 4.wav, 5.wav, 6.wav, 7.wav, 8.wav, 9.wav

10.wav, 11.wav, 12.wav, 13.wav, 14.wav, 15.wav, 16.wav, 17.wav, 18.wav, 19.wav

20.wav, 30.wav, 40.wav, 50.wav, 60.wav, 70.wav, 80.wav, 90.wav

100.wav, 200.wav, 300.wav, 400.wav, 500.wav, 600.wav, 700.wav, 800.wav, 900.wav

1000.wav, 2000.wav, 3000.wav, 4000.wav, 5000.wav, 6000.wav, 7000.wav, 8000.wav, 9000.wav

play_autostop - zmienna bool sterująca stanem zatrzymywania odtwarzania od wykrycia dtmf-u

przykład:

```
play_autostop=true
```

play_curr (zmienna_float) - odczytuje cenę, wymaga plików wav

Uwaga ! W katalogu komunikatów muszą znajdować się pliki reprezentujące cyfry wraz z odmianą
Maksymalna czytana wartość to 9'999'999, dokładność zawsze dwie cyfry po przecinku

PLNGa.wav – grosz
PLNGb.wav – grosze
PLNGc.wav – groszy
PLNZa.wav – złoty
PLNZb.wav – złote
PLNZc.wav – złotych
PLN1Ka.wav – tysiąc
PLN1Kb.wav – tysiące
PLN1Kc.wav – tysięcy
PLN1Ma.wav – milion
PLN1Mb.wav – miliony
PLN1Mc.wav – milionów

PLN0.wav, PLN1.wav, PLN2.wav, PLN3.wav, PLN4.wav, PLN5.wav, PLN6.wav, PLN7.wav,PLN8.wav, PLN9.wav

PLN10.wav, PLN11.wav, PLN12.wav, PLN13.wav, PLN14.wav,PLN15.wav, PLN16.wav, PLN17.wav, PLN18.wav, PLN19.wav

PLN20.wav, PLN30.wav, PLN40.wav, PLN50.wav, PLN60.wav, PLN70.wav, PLN80.wav, PLN90.wav

PLN100.wav, PLN200.wav, PLN300.wav, PLN400.wav, PLN500.wav, PLN600.wav, PLN700.wav, PLN800.wav, PLN900.wav

przykład:

```
float1=99.99
gosub=odtworz_cene_produktu

|

play_curr (float1)
wait_play
return
```

```
  play_link ('plik1.wav','plik2.wav','plik3.wav'...   )     - łączy nagrania i odtwarza razem
```

przykład:

```
label=kobieta
str1='Szanowna_Pani.wav'
goto=powitaj

label=mezczyzna
str1='Szanowny_Panie.wav'
goto=powitaj

label=powitaj
play_link (str1,'Oferta.wav')
wait_play
```

```
  dial_dtmf (zmienna_str)                          - wybierz numer w dtmf-ie ( Uwaga ! Wymaga
                                                     plików 1dtmf.wav itd. w katalogu komunikatów )
```

Uwaga ! W katalogu komunikatów muszą znajdować się pliki z próbkami dtmf-u

0dtmf.wav, 1dtmf.wav, 2dtmf.wav, 3dtmf.wav, 4dtmf.wav, 5dtmf.wav, 6dtmf.wav, 7dtmf.wav, 8dtmf.wav, 9dtmf.wav, *dtmf.wav, #dtmf.wav

przykład:

```
str1='108'
gosub=przelaczenie

|

label=przelaczenie
dial_sys (b_flash300ms)
// centrala wymaga chwili czasu, aż będzie w stanie odbierać cyfry po flashu
timer1=500
wait_timer1
dial_dtmf (str1)
wait_play
return
```

Sterowanie nagrywaniem

```
  record_on                                        - uruchom nagrywanie

  record_off                                       - zakończ nagrywanie
```

przykład:

```
record_on
timer1=10000
wait_timer1
record_off
```

```
  rec_pause_on                                     - wstrzymaj nagrywanie

  rec_pause_off                                    - uruchom nagrywanie
```

przykład:

```
clr_dtmf
record_on
label=petla
if (dtmf~'1') gosub=nagrywanie_start
if (dtmf~'0') gosub=nagrywanie_stop
if (dtmf~'#') goto=nagrywanie_koniec
goto=petla

label=nagrywanie_start
rec_pause_on
clr_dtmf
```

```
return

label=nagrywanie_stop
rec_pause_off
clr_dtmf
return
```

Komunikacja z interfejsem TAPI lub CTI centrali

Zmienne:

cti_con_type	- typ połączenia : 0 - ext<->ext 1 - ext<->co 2 - co<->co
cti_con_ext1	- numer dołączony 1
cti_con_ext2	- numer dołaczony 2
cti_con_co1	- port dołączony 1
cti_con_co2	- port dołaczony 2
cti_con_time	- aktualny czas trwania połączenia
cti_con_dir	- kierunek połączenia : 0 - wychodzące 1 - przychodzące
cc4_ext	- zwraca numer wewnętrzny przypisany do kanału CC4
cc4_con	- zwraca nazwę linii miejskiej przypisaną do kanału

Instrukcje:

cti_ext_on (nr_wewnętrzny)	- zajęcie wewnętrznego

przykład:

```
hook_on
cti_ext_on (cc4_ext)
```

cti_ext_off (nr_wewnętrzny)	- zwolnienie wewnętrznego

przykład:

```
hook_off
cti_ext_on ('101')
```

cti_ext_dial (nr_wewnętrzny,cyfry)	- wybranie cyfr

przykład:

```
hook_on
cti_ext_on (cc4_ext)
timer1=1000
wait_timer1
cti_ext_dial (cc4_ext,'9,0618358600')
timer1=60000
label=czekaj
if (timer1=0) goto=rozlacz
if (cti_ext_state {cc4_ext}<>2) goto=czekaj
play_on ('Informacja.wav')
wait_play
label=rozlacz
hook_off
cti_ext_off {cc4_ext}
```

cti_ext_state {nr_wewnętrzny}	- stan numeru wewnętrznego stany : 0 - wolny 1 - zajęty 2 - połączony 3 - dzwoni Uwaga ! w zmiennych będą dostępne dalsze informacje
cti_ext_numb	- zmienna str numer wybrany lub clip
cti_ext_ddi	- zmienna str numer ddi lub wzywany

przykład:

```
if (cti_ext_state {cc4_ext}=3) goto=odbierz
restart
label=odbierz
if (cti_ext_numb='0618358600') goto=klient_specjalny
goto=klient_zwykły
```

lub

```
if (cti_ext_state {cc4_ext}=3) goto=odbierz
restart
label=odbierz
if (cti_ext_ddi='100') goto=infolinia_0801xxx100
if (cti_ext_ddi='200') goto=infolinia_0801xxx200
if (cti_ext_ddi='300') goto=infolinia_0801xxx300
goto=infolinia_domyslna
```

```
  cti_co_state {nr_portu}                          - stan portu miejskiego
                                                     stany : 0 - wolny
                                                             1 - zajęty
                                                             2 - połączony
                                                             3 - dzwoni
```

przykład:

```
if (cti_co_state {01}<>0) info ('Linia zajeta')
```

```
  cti_new_connect                                  - odczytuje informacje o nowym połączeniu w
                                                     centrali (cti )

                                                     stany : False - brak informacji
                                                             True  - informacja w zmiennych
     aktualizuje zmienne :
cti_con_type;cti_con_ext1;cti_con_ext2;cti_con_co1;cti_con_co2;cti_con_time;cti_con_dir

  cti_ext_connect {nr_wewnętrzny}                  - odczytuje informacje o połączeniu dla
                                                     wewnętrznego

                                                     stany : False - brak informacji ( połączenia )
                                                             True  - informacja w zmiennych ( jest
                                                     połączenie )

  cti_co_connect {nr_portu}                        - odczytuje informacje o połączeniu dla portu
                                                     miejskiego

                                                     stany : False - brak informacji ( połączenia )
                                                             True  - informacja w zmiennych ( jest
                                                     połączenie )


  cti_ext_exec (nr_wewnętrzny,cyfry,akcja)         - wybranie cyfr, transfer, hold itp.

  b_drop       = 0                                 - rozłączenie po tapi
  b_ansfer     = 1                                 - odebranie po tapi
  b_dial       = 2                                 - wybranie numeru po tapi
  b_hold       = 3                                 - hold po tapi
  b_unhold     = 4                                 - unhold po tapi
  b_park       = 5                                 - park po tapi
  b_unpark     = 6                                 - unpark po tapi
  b_trans_set  = 7                                 - przygotowanie transferu po tapi
  b_trans_comp = 8                                 - zakończenie transferu po tapi
```

przykład:

```
hook_on
cti_ext_on (cc4_ext)
timer1=500
wait_timer1
cti_ext_exec (cc4_ext,'9,0618358600',b_dial)
```

Współpraca z bazą danych

Uwaga ! Parametry połączenia z serwerem MySQL są definiowane w ustawieniach R4.

```
  sql_connect                                      - dołączenie do bazy danych

  sql_create_db ('nazwa bazy')                     - utworzenie bazy danych
```

```
  sql_select_db ('nazwa bazy')                    - wybranie pazy danych

  sql_drop_db ('nazwa bazy')                      - usunięcie bazy danych

  sql_close                                       - odłączenie od bazy danych
```

przykład:

```
sql_connect
sql_select_db ('ZETKOM')
```

```
  sql_quwery: tutaj zapytanie/polecenie SQL       - wykonuje polecenie/zapytanie sql
    {var1} {str1}   - tak wpisane zmienne zostaną zamienione na wartości zmiennych

  sql_first                                       - ustawia na pierwszy wiersz odpowiedzi

  sql_next                                        - następny wiersz odpowiedzi

  sql_prior                                       - poprzedni wiersz odpowiedzi

  sql_last                                        - ostatni wiersz odpowiedzi

  is_sql                                          - czy jest połączenie z bazą danych bool

  sql_fieldscount                                 - zwraca ilość pól odpowiedzi var

  sql_rowscount                                   - zwraca ilość wierszy odpowiedzi var

  sql_field_var {numer pola}                      - zwraca pole jako zmienną var

  sql_field_str {numer pola}                      - zwraca pole jako zmienną str

  sql_field_float {numer pola}                    - zwraca pole jako zmienną float
```

przykład:

```
sql_connect
sql_select_db ('ZETKOM')
label=czekaj_na_polaczenie
if (cti_ext_state{cc4_ext}=3) goto=odbierz
goto=czekaj_na_polaczenie
label=odbierz
sql_quwery:SELECT zapowiedz FROM klienci WHERE numer='{cti_ext_numb}' LIMIT 1;
if (sql_rowscount=0) goto=zapowiedz_domyslna
play_on (sql_field_str {0})
goto=zapowiedz
label=zapowiedz_domyslna
play_on ('Powitanie standartowe')
label=zapowiedz
wait_play
```

lub

```
// łączenie na grupę
sql_quwery:SELECT numer FROM grupa WHERE zalogowany='1' AND rozmawia='0' ORDER BY rozmawial DESC
LIMT 1;
if (sql_rowscount=0) goto=grupa_zajeta
str1=sql_field_str {0}
sql_quwery:UPDATE SET rozmawial=CURRENT_TIMESTAMP(),rozmawia='1' WHERE numer='{str1}';
cti_ext_exec (cc4_ext,str1,b_trans_set)
timer1=1000
wait_timer1
cti_ext_exec (str1,'',b_trans_comp)
```