

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 nagraniem znaków np. 1.wav, a.wav, dla selektora 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) - wyświetla wartość zmiennej w oknie debugowania i wstrzymuje program (wymaga włączenia przełącznika DEBUG )
```

```

clr_semaphore01 .. clr_semaphore32      - wyzeruj semafor 1..32
init_semaphore                          - wyzeruj wszystkie semafony
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

```

<code>is_bool3=?</code>		- przełącznik 3
<code>is_bool4=?</code>		- przełącznik 4
<code>is_bool5=?</code>		- przełącznik 5
<code>var1=?</code>		- zmienna lokalna 1
<code>var2=?</code>		- zmienna lokalna 2
<code>var3=?</code>		- zmienna lokalna 3
<code>var4=?</code>		- zmienna lokalna 4
<code>var5=?</code>		- zmienna lokalna 5
<code>str1=?</code>		- zmienna tekstowa lokalna 1
<code>str2=?</code>		- zmienna tekstowa lokalna 2
<code>str3=?</code>		- zmienna tekstowa lokalna 3
<code>str4=?</code>		- zmienna tekstowa lokalna 4
<code>str5=?</code>		- zmienna tekstowa lokalna 5
<code>gl_bool1=?</code>		- przełącznik globalny 1
<code>gl_bool2=?</code>		- przełącznik globalny 2
<code>gl_bool3=?</code>		- przełącznik globalny 3
<code>gl_bool4=?</code>		- przełącznik globalny 4
<code>gl_bool5=?</code>		- przełącznik globalny 5
<code>gl_var1=?</code>		- zmienna globalna 1
<code>gl_var2=?</code>		- zmienna globalna 2
<code>gl_var3=?</code>		- zmienna globalna 3
<code>gl_var4=?</code>		- zmienna globalna 4
<code>gl_var5=?</code>		- zmienna globalna 5
<code>inc (zmienna)</code>		- zwiększ zmienną var o 1
<code>inc (zmienna,wartość)</code>		- zwiększ zmienną var o wartość
<code>dec (zmienna)</code>		- zmniejsz zmienną var o 1
<code>dec (zmienna,wartość)</code>		- zmniejsz zmienną var o wartość
 { uwaga ! instrukcją dec (stoper1,5000) => można zmniejszyć wartość naliczoną przez stoper o 5 sekund }		
<code>save_str ('filename',zmienna_str)</code>		- zapisz zmienną tekstową do pliku
<code>load_str ('filename',zmienna_str)</code>		- odczytaj zmienną tekstową z pliku
<code>false</code>	<code>= 0</code>	- stała oznaczająca fałsz
<code>true</code>	<code>= 1</code>	- stała oznaczająca prawdę
<code>b_flash100ms</code>		- flash 100 ms
<code>b_flash200ms</code>		- flash 200 ms
<code>b_flash300ms</code>		- flash 300 ms
<code>b_flash400ms</code>		- flash 400 ms
<code>b_flash500ms</code>		- flash 500 ms
<code>b_flash600ms</code>		- flash 600 ms
<code>b_flash700ms</code>		- flash 700 ms

b_flash800ms	- flash 800 ms
b_flash900ms	- flash 900 ms
roz_out	- dodaj rozmowę wychodząca
roz_in	- dodaj rozmowę przychodząca
roz_outb	- dodaj rozmowę niedoszła
roz_inb	- dodaj rozmowę nieodebrana
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 wychodząca i wyślij mailem
roz_in+mail	- dodaj rozmowę przychodząca i wyślij mailem
roz_outb+mail	- dodaj rozmowę niedoszła i wyślij mailem
roz_inb+mail	- dodaj rozmowę nieodebrana i wyślij mailem
roz_kom+mail (opis)	- dodaj komunikat i wyślij mailem
roz_out+mail+link	- dodaj rozmowę wychodząca, połącz z nagraniem i wyślij mailem
roz_in+mail+link	- dodaj rozmowę przychodząca, połącz z nagraniem i wyślij mailem
roz_outb+mail+link	- dodaj rozmowę niedoszła, połącz z nagraniem i wyślij mailem
roz_inb+mail+link	- dodaj rozmowę nieodebrana, połącz z nagraniem i wyślij mailem
roz_kom+mail+link (opis)	- dodaj komunikat, połącz z nagraniem i wyślij mailem
roz_out+link	- dodaj rozmowę wychodząca i połącz z nagraniem
roz_in+link	- dodaj rozmowę przychodząca 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
```

<code>gl_str7=?</code>	- zmienna tekstowa globalna 7
<code>gl_str8=?</code>	- zmienna tekstowa globalna 8
<code>gl_str9=?</code>	- zmienna tekstowa globalna 9
<code>gl_str0=?</code>	- zmienna tekstowa globalna 0
<code>gl_str{0..99}=?</code>	- zmienna tekstowa globalna 0..99
<code>gl_bool{0..99}=?</code>	- przełączniki globalne 0..99
<code>gl_date0=?</code>	- data globalna 0
<code>gl_date1=?</code>	- data globalna 1
<code>gl_date2=?</code>	- data globalna 2
<code>gl_date3=?</code>	- data globalna 3
<code>gl_date4=?</code>	- data globalna 4
<code>gl_date5=?</code>	- data globalna 5
<code>gl_date6=?</code>	- data globalna 6
<code>gl_date7=?</code>	- data globalna 7
<code>gl_date8=?</code>	- data globalna 8
<code>gl_date9=?</code>	- data globalna 9
<code>gl_date{0..99}=?</code>	- daty globalne 0..99
<code>rec_pause_on</code>	- wstrzymaj nagrywanie
<code>rec_pause_off</code>	- uruchom nagrywanie
<code>level_mid</code>	- poziom sygnału średni byte (readonly)
<code>clr_level_mid</code>	- wyzeruj poziom średniego
<code>gosub=?</code>	- rozkaz do podprogramu możliwe 16 poziomów podprogramów
<code>return</code>	- powrót z podprogramu możliwe 16 poziomów podprogramów

Polecenia od engine 1.6 -----

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

Polecenia od engine 1.7 -----

<code>logprint (zmienna)</code>	- wysyła komunikat do logu
---------------------------------	----------------------------

<code>gl_var{0..99}=?</code>	- zmienna globalna 0..99
------------------------------	--------------------------

Wprowadzona możliwość adresowania zmiennej zmienną tak jak w przykładzie poniżej

<code>gl_var{var0}=?</code>	
-----------------------------	--

<code>chanel</code>	- numer aktualnego programu
---------------------	-----------------------------

<code>line_state{1..32}</code>	- stan linii z M3
--------------------------------	-------------------

<code>port_state{1..32}</code>	- stan linii z R3
--------------------------------	-------------------

<code>line_state</code>	- stan linii dla bieżącego portu
-------------------------	----------------------------------

<code>port_state</code>	- stan linii dla bieżącego portu
-------------------------	----------------------------------

<code>roz_nr{1..32}=?</code>	- numer wybrany
------------------------------	-----------------

<code>roz_ab{1..32}=?</code>	- numer abonenta
------------------------------	------------------

<code>roz_ko{1..32}=?</code>	- kod konta
------------------------------	-------------

<code>roz_tm{1..32}=?</code>	- czas trwania
------------------------------	----------------

<code>gl_record{1..32}</code>	- czy nagrywanie	bool
<code>gl_play{1..32}</code>	- czy odtwarzanie	bool
<code>gl_hook{1..32}</code>	- stan linii	bool
<code>isdn_clip{1..32}</code>	- ostatni clip odczytany z linii isdn	str
<code>isdn_ddi_msn{1..32}</code>	- ostatni ddi lub msn odczytany z linii isdn	str

Polecenia od engine 1.8 -----

<code>cti_ext_on (nr_wewnętrzny)</code>	- zajęcie wewnętrznego
<code>cti_ext_off (nr_wewnętrzny)</code>	- zwolnienie wewnętrznego
<code>cti_ext_dial (nr_wewnętrzny,cyfry)</code>	- wybranie cyfr
<code>cti_ext_state {nr_wewnętrzny}</code>	- stan numeru wewnętrznego stany : 0 - wolny 1 - zajęty 2 - połączony 3 - dzwoni Uwaga ! w zmiennych będą dostępne dalsze informacje
<code>cti_ext_num</code>	- zmienna str numer wybrany lub clip
<code>cti_ext_ddi</code>	- zmienna str numer ddi lub wzywany
<code>cti_co_state {nr_portu}</code>	- stan portu miejskiego stany : 0 - wolny 1 - zajęty 2 - połączony 3 - dzwoni Uwaga ! w zmiennych będą dostępne dalsze informacje
<code>cti_new_connect</code>	- odczytuje informacje o nowym połączeniu w centrali (cti) stany : False - brak informacji True - informacja w zmiennych
aktualizuje zmienne :	
<code>cti_con_type;cti_con_ext1;cti_con_ext2;cti_con_co1;cti_con_co2;cti_con_time;cti_con_dir</code>	
<code>cti_ext_connect {nr_wewnętrzny}</code>	- odczytuje informacje o połączeniu dla wewnętrznego stany : False - brak informacji (połączenia) True - informacja w zmiennych (jest połączenie)
<code>cti_co_connect {nr_portu}</code>	- odczytuje informacje o połączeniu dla portu miejskiego stany : False - brak informacji (połączenia) True - informacja w zmiennych (jest połączenie)
<code>cti_con_type</code>	- typ połączenia : 0 - ext<->ext 1 - ext<->co 2 - co<->co
<code>cti_con_ext1</code>	- numer dołączony 1
<code>cti_con_ext2</code>	- numer dołączony 2
<code>cti_con_co1</code>	- port dołączony 1
<code>cti_con_co2</code>	- port dołączony 2
<code>cti_con_time</code>	- aktualny czas trwania połączenia
<code>cti_con_dir</code>	- kierunek połączenia : 0 - wychodzące 1 - przychodzące
<code>cc4_ext</code>	- 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 bazy danych

sql_drop_db ('nazwa bazy') - usunięcie bazy danych

sql_close - odłączenie od bazy danych

sql_query: 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_fieldcount - zwraca ilość pól odpowiedzi var

sql_rowcount - 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) - wyświetla wartość zmiennej w oknie info (karta powiązania)

b_drop = 0 - rozłączenie po tapi

b_transfer = 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)

<code>float1=?</code>	- zmienna (zmiennoprzecinkowa)
<code>float2=?</code>	- zmienna (zmiennoprzecinkowa)
<code>float3=?</code>	- zmienna (zmiennoprzecinkowa)
<code>float4=?</code>	- zmienna (zmiennoprzecinkowa)
<code>float5=?</code>	- zmienna (zmiennoprzecinkowa)
<code>float6=?</code>	- zmienna (zmiennoprzecinkowa)
<code>float7=?</code>	- zmienna (zmiennoprzecinkowa)
<code>float8=?</code>	- zmienna (zmiennoprzecinkowa)
<code>float9=?</code>	- zmienna (zmiennoprzecinkowa)
<code>play_curr (zmienna_float)</code>	- odczytuje cenę, wymaga plików wav
<code>sql_field_float {numer pola}</code>	- zwraca pole jako zmienną float

Polecenia od engine 2.1 -----

<code>play_link ('plik1.wav','plik2.wav','plik3.wav'...)</code>	- łączy nagrania i odtwarza razem
<code>on_usb_err</code>	- do tego miejsca skok po błędzie usb
<code>on_tapi_err</code>	- do tego miejsca skok po błędzie tapi
<code>on_sql_err</code>	- do tego miejsca skok po błędzie sql
<code>b_clear_station = 254</code>	- wyzerowanie numeru
<code>b_clear_all = 255</code>	- wyzerowanie całego tapi
<code>restore</code>	- powrót do programu po obsłudze błędu
<code>break=</code>	- wyskoczenie z podprogramu do etykiety
<code>played</code>	- ile czasu miało odtworzone nagranie
<code>sign_rst</code>	- wyślij reset do nagrywarki
<code>peerip</code>	- numer zdalnego ip programu CRM
<code>peerport</code>	- numer portu zdalnego programu CRM
<code>peeropen</code>	- numer portu w programie CRM do komunikacji
<code>peerkey</code>	- 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_var{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 lms

```
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 elektronicznej)

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='niedziela'
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='Infol.wav'
gosub=odtworz_zapowiedz
```

```
|

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

`return`

- powrót z podprogramu możliwe 16 poziomów podprogramów

`restore`

- powrót do programu po obsłudze błędu

przykład:

```
on_sql_err
sql_connect
restore
```

`break=`

- wyskoczenie z podprogramu do etykiety

przykład:

```
str1='Infol.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 muszą 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:

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

Instrukcje:

<code>cti_ext_on (nr_wewnętrzny)</code>	- zajęcie wewnętrznego
---	------------------------

przykład:

```

hook_on
cti_ext_on (cc4_ext)

```

<code>cti_ext_off (nr_wewnętrzny)</code>	- zwolnienie wewnętrznego
--	---------------------------

przykład:

```

hook_off
cti_ext_on ('101')

```

<code>cti_ext_dial (nr_wewnętrzny,cyfry)</code>	- 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}

```

<code>cti_ext_state {nr_wewnętrzny}</code>	- stan numeru wewnętrznego stany : 0 - wolny 1 - zajęty 2 - połączony 3 - dzwoni
--	--

Uwaga ! w zmiennych będą dostępne dalsze informacje

<code>cti_ext_num</code>	- zmienna str numer wybrany lub clip
--------------------------	--------------------------------------

<code>cti_ext_ddi</code>	- 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_zwykly

```

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_col;cti_con_co2;cti_con_time;cti_con_dir

```

```

cti_ext_connect {nr_wewnetrzny}

```

- 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_wewnetrzny,cyfry,akcja)

```

- wybranie cyfr, transfer, hold itp.

```

b_drop      = 0
b_ansfer    = 1
b_dial      = 2
b_hold      = 3
b_unhold    = 4
b_park      = 5
b_unpark    = 6
b_trans_set = 7
b_trans_comp = 8

```

- rozłączenie po tapi
- odebranie po tapi
- wybranie numeru po tapi
- hold po tapi
- unhold po tapi
- park po tapi
- unpark po tapi
- przygotowanie transferu po tapi
- 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

<code>sql_select_db ('nazwa bazy')</code>	- wybranie bazy danych
<code>sql_drop_db ('nazwa bazy')</code>	- usunięcie bazy danych
<code>sql_close</code>	- odłączenie od bazy danych

przykład:

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

<code>sql_query: tutaj zapytanie/polecenie SQL</code>	- wykonuje polecenie/zapytanie sql
<code>{var1} {str1}</code>	- tak wpisane zmienne zostaną zamienione na wartości zmiennych
<code>sql_first</code>	- ustawia na pierwszy wiersz odpowiedzi
<code>sql_next</code>	- następny wiersz odpowiedzi
<code>sql_prior</code>	- poprzedni wiersz odpowiedzi
<code>sql_last</code>	- ostatni wiersz odpowiedzi
<code>is_sql</code>	- czy jest połączenie z bazą danych bool
<code>sql_fieldscount</code>	- zwraca ilość pól odpowiedzi var
<code>sql_rowscount</code>	- zwraca ilość wierszy odpowiedzi var
<code>sql_field_var {numer pola}</code>	- zwraca pole jako zmienną var
<code>sql_field_str {numer pola}</code>	- zwraca pole jako zmienną str
<code>sql_field_float {numer pola}</code>	- 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_query:SELECT zapowiedz FROM klienci WHERE numer='{cti_ext_num}' 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_query:SELECT numer FROM grupa WHERE zalogowany='1' AND rozmawia='0' ORDER BY rozmawial DESC
LIMIT 1;
if (sql_rowscount=0) goto=grupa_zajeta
str1=sql_field_str {0}
sql_query: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)
```