



Informacji na temat kart produkowanych dawniej przez Ambex udziela firma Egmont Instruments.

Pod adresem <http://www.ambex.com.pl> powstaje archiwum instrukcji i oprogramowania do kart komputerowych produkowanych dawniej przez Ambex, a obecnie jeszcze w części oferty przez Egmont Instruments. Archiwum to jest systematycznie uzupełniane. Jeśli poszukują Państwo informacji do starych lub aktualnych wyrobów, prosimy kierować się właśnie pod powyższy adres w Internecie. Jeśli nie znajdą tam Państwo potrzebnej informacji, prosimy o bezpośredni kontakt z nami.

Strony <http://www.ambex.com.pl> są prowadzone bezpośrednio przez firmę Egmont Instruments.



DOKUMENTACJA DRIVERA
MODUŁU KONTROLNO-POMIAROWEGO

LC-030-1612

WERSJA 3.00

Wydanie: listopad 1991

AMBEX Spółka z o.o.

Warszawa

DZIAŁ HANDLOWY	ul. Topiel 6	8:30-16:30	tel. (0-2) 635-87-24 tel. (0-2) 635-04-76 fax / tel. (0-2) 635-91-51
SKLEP ELEKTRONICZNY	ul. Topiel 15b	9:00-17:00	tel. (0-2) 635-04-05
SERWIS	ul. Zabińskiego 7	8:00-16:00	tel.

Spis treści:

1. Informacje ogólne.
2. Instalacja driver'a.
3. Opis driver'a.
 - 3.1. Informacje wstępne.
 - 3.2. Standardy oznaczeń, numeracja, dane charakterystyczne.
 - 3.3. Komunikacja z driver'em.
 - 3.4. Typy warunków startu / zakończenia operacji.
4. Funkcje driver'a.
 - 4.1. Inicjalizacja (MODULE_INIT).
 - 4.2. Informacja o konfiguracji ogólnej (GET_TOTAL_CONFIGURATION).
 - 4.3. Informacja o konfiguracji modułu (GET_MODULE_CONFIGURATION).
 - 4.4. Informacja o szczegółach technicznych (GET_INFO).
 - 4.4.1. Wejścia cyfrowe.
 - 4.4.2. Wyjścia cyfrowe.
 - 4.4.3. Wejścia analogowe.
 - 4.4.4. Wyjścia analogowe.
 - 4.4.5. Układy licznikowo-czasowe (CTC).
 - 4.5. Zadeklarowanie częstotliwości zegara magistrali (SET_CLOCK).
 - 4.6. Ustawienie zakresu napięć (SET_VOLTAGE_RANGE).
 - 4.7. Zadeklarowanie maksymalnego czasu obsługi obcych przerwań (SET_TIME).
 - 4.8. Oczekiwanie na zakończenie operacji (WAIT_FOR_END).
 - 4.9. Przerwanie operacji (BREAK).
 - 4.10. Wejście cyfrowe (DIGITAL_INPUT).
 - 4.11. Wyjście cyfrowe (DIGITAL_OUTPUT).
 - 4.12. Zapis CTC (CTC_WRITE).
 - 4.13. Odczyt CTC (CTC_READ).
 - 4.14. Transmisja danych (DATA_TRANSMIT).
 - 4.15. Przetwarzanie analogowo-cyfrowe (ANALOG_INPUT).
 - 4.16. Przetwarzanie cyfrowo-analogowe (ANALOG_OUTPUT).
 - 4.17. Zakończenie pracy z driver'em (LEAVE_DRIVER).
 - 4.18. Obsługa przerwań (INTERRUPT_SERVICE).
5. Zestawienie kodów zakończenia funkcji.
6. Projektowanie programów użytkowych.
 - 6.1. Programowanie w języku C.
 - 6.2. Programowanie w języku Pascal.
 - 6.3. Programowanie w języku assemblera.

1. Informacje ogólne.

Przy tworzeniu oprogramowania modułów przyjęto zasadę, że cała komunikacja z modułem prowadzona jest za pośrednictwem rezydentnego programu dostępnego dla programów użytkowych poprzez przerwanie programowe. Takie rozwiązanie ma następujące zalety:

- użytkownik jest zwolniony ze znajomości szczegółów technicznych tak modułu jak i używanego komputera,
- rozwiązanie to jest niezależne od używanej implementacji języka wyższego poziomu.

Program obsługi został napisany w standardzie driver'ów systemu operacyjnego MS-DOS (wersja 3.1. i wyzsze). Główną przyczyną wyboru takiego rozwiązania jest umożliwienie prostego badania obecności driver'a w systemie. Jediną wykorzystywaną standardową funkcją driver'a jest funkcja inicjalizacji wykonywana w trakcie jego instalacji. Po zainstalowaniu driver służy tylko jako obsługa danego przerwania programowego.

Driver modułu LC-030-1612 jest jednym z serii driver'ów obsługujących poszczególne moduły firmy AMBEX. W związku z tym podporządkowany został standardowi przyjętemu przez firmę. W standardzie tym przyjęto, że w komputerze może być zainstalowanych kilka (do czterech) modułów danego rodzaju obsługiwanych przez jeden driver. Moduł LC-030-1612 jest jednak modułem nietypowym w tym sensie, że można zainstalować w komputerze tylko jeden taki moduł. W związku z tym niektóre parametry wydawać się mogą nadmiarowe.

2. Instalacja driver'a.

Instalacji driver'a dokonuje się za pomocą programu instalacyjnego - INSTALL. Przed instalacją należy skopiować (odpowiedni dla rodzaju komputera oraz modułu) zbiór LC*.DRV (driver) do wybranego katalogu na dysku twardym, bądź, w przypadku pracy z komputerem nie posiadającym dysku twardego, na dyskietkę systemową.

Po uruchomieniu programu instalacyjnego zleceniem INSTALL wybieramy opcję instalacji: dysk twardy bądź dyskietka (musi być to dyskietka systemowa). Następnie użytkownik powinien ustalić konfigurację modułów danego typu (jedorazowy przebieg programu umożliwia instalację driver'a dla modułów tylko jednego typu). Bieżący parametr wskazywany jest podświetlonym paskiem. Zmiana wartości parametru następuje zawsze po wskazaniu go i naciśnięciu klawisza Enter. W zależności od typu parametru następuje wówczas nadanie parametrowi kolejnej wartości z listy wartości dopuszczalnych lub na ekranie pojawia się ramka, w której użytkownik powinien wpisać nową wartość. W tym drugim przypadku zaakceptowanie nowej wartości następuje po naciśnięciu klawisza Enter, zaś jej odrzucenie - po naciśnięciu klawisz Esc: parametr zachowuje wówczas swoją poprzednią wartość.

Program instalacyjny INSTALL jest wspólny dla wszystkich rodzajów modułów analogowych produkowanych przez firmę AMBEX. Na ekranie pojawiają się tylko parametry istotne dla wybranego typu modułu; w zależności od typu modułu są one w różny sposób ograniczone. Poniżej podany zostanie skrótowy opis wszystkich parametrów.

Parametry ogólne:

- a) typ modułu: LC-010-1612, LC-011-1612, LC-020-0812, LC-020-3212, LC-030-1612;
- b) liczba zainstalowanych modułów danego typu: od 1 do 4 ^{*)} ;
- c) typ komputera: XT, AT, 386 ^{*)} ;
- d) pełna nazwa ścieżki do katalogu, gdzie skopiowano zbiór LC*.DRV (łącznie z nazwą napędu dyskowego);
- e) wybór trybu pracy: z pamięcią rozszerzoną lub bez ^{2) 3)} ;
- f) adres pamięci rozszerzonej w postaci szesnastkowej;
- g) rozmiar pamięci rozszerzonej w kilobajtach;
- h) maksymalny czas obsługi przerwania, które może mieć miejsce w trakcie trwania długich pomiarów - podawany w mikrosekundach (dodatnia liczba mieszcząca się na 16 bitach) ^{3) 8)} .

Parametry dotyczące jednego modułu ⁴⁾:

- i) nazwa modułu: od A do D (nazwa związana jest z adresem bazowym modułu);
- j) tryb pracy modułu: master lub slave (istotne przy współpracy kilku modułów) ⁸⁾;
- k) częstotliwość zegara modułu: 4 lub 8 MHz ⁹⁾;
- l) zakres napięć dla wejść analogowych;
- m) liczba kanałów wejść analogowych ⁵⁾;
- n) wyposażenie modułu we wzmacniacz Sample&Hold ⁶⁾;
- o) zakres napięć dla wyjścia analogowego - kanał 1 ⁷⁾;
- p) zakres napięć dla wyjścia analogowego - kanał 2 ³⁾ ⁷⁾.

¹⁾ dla modułu LC-030-1612 - tylko AT lub 386.

²⁾ nie dotyczy pracy z komputerem typu XT.

³⁾ nie dotyczy pracy z modułem LC-010-1612.

⁴⁾ przy instalacji kilku modułów jednego typu parametry dotyczące każdego z modułów różnią się między sobą np. moduły muszą różnić się nazwą.

⁵⁾ moduł LC-010-1612: liczba kanałów zależna jest od tego czy na module zainstalowano multiplexer i wynosi 16 bądź 1, moduły LC-011-1612 i LC-030-1612: liczba kanałów jest stała i wynosi 16, moduły LC-020-0812 i LC-020-3212: liczba kanałów jest zależna od liczby zainstalowanych wzmacniaczy Sample&Hold

⁶⁾ dotyczy tylko modułu LC-010-1612.

⁷⁾ nie dotyczy pracy z modułem LC-020-3212.

Po wprowadzeniu żądanych parametrów pracy modułu (modułów) należy uruchomić funkcję - instalacja (F10). Program modyfikuje zbiór CONFIG.SYS lub w przypadku jego braku - tworzy zbiór o tej nazwie. Następnie program inicjalizuje system operacyjny (reboot).

Uwaga: podłączenie modułu do przerwania sprzętowego oraz numer tego przerwania wykrywane są automatycznie przez driver w momencie ładowania systemu. To samo dotyczy kanałów DMA.

3. Opis driver'a.

3.1. Informacje wstępne.

Przed driver'em postawione zostały następujące zadania:

- pełne wykorzystanie możliwości sprzętowych oferowanych przez moduł LC-030-1612
- rozszerzenie w sposób programowy możliwości modułu o funkcje, które nie są lub nie mogą być realizowane sprzętowo; do funkcji takich należą: różnorodne warunki startu i zatrzymania (stopu) operacji wejścia / wyjścia (odczyt / zapis portów cyfrowych, przetwarzanie a/c i c/a); w grę wchodzi tu warunkowanie startu / stopu operacji sygnałami cyfrowymi (poziomem, zboczem, kombinacją sygnałów) oraz wpływem czasu rzeczywistego (data i odcinek czasu);
- realizacja pewnych funkcji, dzięki którym możliwe jest pisanie uniwersalnych programów, niezależnych od instalacji konkretnego modułu

Driver rozpoczyna swoją pracę już w momencie ładowania systemu. Wówczas to pobiera i analizuje parametry instalacji podane w poleceniu "DEVICE=..." z pliku CONFIG.SYS - dzięki tym informacjom możliwe jest pisanie programów niezwiązanych z konkretną instalacją modułu. Następnie wykonywane jest tzw. twarde zerowanie modułu, w trakcie którego wykonywane jest wstępne programowanie - m.in. ustawianie wyjść analogowych na 0V a wyjść cyfrowych - na 1. Następnie wykonywane są testy, których celem jest sprawdzenie czy i do jakich przerwań i kanałów DMA podłączony jest moduł. Na koniec driver przechwytuje przerwanie 1C16 (przerwanie usługowe, wywoływane przez procedurę obsługi przerwania zegarowego) aby w bezpieczny sposób odmierzać upływający czas astronomiczny (synchronizacja wewnętrznego zegara driver'a z zegarem komputera dokonywana jest w momencie wywoływania funkcji MODULE_INIT). Inne programy wykorzystujące przerwanie 1C16 mają obowiązek - po przechwyceniu tego przerwania - wykonywać również poprzednią procedurę obsługi tego przerwania.

Driver'y modułów analogowych produkowanych przez firmę AMBEX zostały zaprojektowane w sposób jednolity. Dzięki temu, jeżeli tylko program nie korzysta jawnie z cech czy funkcji modułu specyficznych tylko dla niego, to może być bez zmiany wykorzystywany do współpracy z różnymi typami modułów. Oczywiście z powodu takich założeń pewne funkcje czy tryby pracy driver'a są dostępne dla jednych typów modułów, dla innych - nie.

Driver jest do pewnego stopnia wielowejsciowy (re-entrant). Oznacza to, że w trakcie wykonywania jednej z funkcji driver'a można wykonać inną. W chwili obecnej liczba takich równoległych wejść nie może przekroczyć 3, w przeciwnym razie driver zwraca błąd LCO_REJECTED.

3.2. Standardy oznaczeń, numeracja, dane charakterystyczne.

Przy pisaniu tak oprogramowania jak i niniejszej dokumentacji przyjęto następujące zasady:

- wszystkie nazwy pól rekordów, stałych itp. (z wyjątkiem nazw funkcji) opatrzone są przedrostkiem LCO_;
- nazwy występujące w dokumentacji są identyczne z nazwami występującymi w plikach źródłowych dla języków C, Pascal, assembler (z dokładnością do rozróżnienia małe / duże litery).

Numer przerwania programowego związanego z driver'em modułu:

typ modułu	nazwa przerwania	numer przerwania szesnastkowy	numer przerwania dziesiętny
LC-030-1612	LC030_16	96	150

Driver widziane jest w systemie DOS jako urządzenie. Nazwa tego urządzenia jest następująca:

typ modułu	nazwa driver'a	nazwa urządzenia
LC-030-1612	LC3016A.DRV	LC3016^^

Kodowanie numerów modułów:

moduł	nazwa kodu	wartość
A	LCO_MODA	1
B	LCO_MODB	2

Kodowanie typów urządzeń w modułach:

urządzenie	nazwa kodu	wartość
porty cyfrowe wejściowe	LCO_DINPUT	1
porty cyfrowe wyjściowe	LCO_DOUTPUT	2
przetworniki a/c	LCO_AINPUT	3
przetworniki c/a	LCO_AOUTPUT	4
kanały CTC	LCO_CTC	5

Wszystkie wejścia, wyjścia, kanały itp. numerowane są od 1.

Długości buforów, pomiarów, marginesów itp. podawane są zawsze w próbkach.

3.3. Komunikacja z driver'em.

Funkcje driver'a wywoływane są poprzez przerwanie programowe (numer przerwania - patrz p. 3.2.). Przesyłanie informacji pomiędzy programem użytkowym a driver'em odbywa się poprzez rekord opisu zlecenia, którego adres przekazywany jest w rejestrach DX:DI. Rekord ten służy do przekazywania informacji zarówno do jak i od driver'a.

Rekord opisu zlecenia ma strukturę zależną od rodzaju zlecenia. Jedynie trzy pierwsze pola są niezmiennie i mają następujące znaczenie:

adres rekordu: DX:DI

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dotkliwe informacje o błędach

LCO_CODE określa funkcję jaka ma być wykonana przez driver a zarazem sposób interpretacji ciągu bajtów znajdujących się pod adresem DX:DI.

LCO_STATUS informuje program wywołujący driver o poprawności wykonania funkcji:

LCO_STATUS = 0: wykonanie poprawne

LCO_STATUS < 0: wykonanie błędne

LCO_STATUS > 0: wykonanie poprawne z zastrzeżeniami (ostrzeżenia)

LCO_ERR_STAT (jeżeli jest mniejszy od zera) niesie pewne dodatkowe informacje komentujące zwrócony w LCO_STATUS kod błędu. Dotyczy to dwóch sytuacji:

- błędnie podane parametry warunku startu / stopu (LCO_STATUS = LCO_ILL_START / LCO_ILL_STOP); LCO_ERR_STAT precyzuje co zostało podane błędnie
- funkcja przerwana wywołaniem funkcji BREAK (LCO_STATUS = LCO_BROKEN); LCO_ERR_STAT określa wtedy, czy funkcja została przerwana w trakcie oczekiwania na spełnienie warunku startu czy już w trakcie przetwarzania.

Istnieje jeden wyjątek od tej zasady: funkcja BREAK może zwrócić LCO_STATUS = 0 i LCO_ERR_STAT <> 0 (patrz opis funkcji, p.4.9).

W rozdziale 5 podano tabele wszystkich kodów zwracanych przez LCO_STATUS i LCO_ERR_STAT.

3.4. Typy warunków startu / zakończenia operacji.

W driverze zaimplementowano następujące warunki startu i zakończenia operacji:

- poziom sygnału cyfrowego: warunek jest spełniony, gdy sygnał cyfrowy ma zadaną wartość;
- zbocze sygnału cyfrowego: warunek jest spełniony, gdy sygnał cyfrowy zmieni swoją wartość w określony sposób;
- kombinacja sygnałów cyfrowych: warunek jest spełniony, gdy kombinacja sygnałów cyfrowych jest równa (różna) zadanej; w pierwszym przypadku wszystkie zadeklarowane sygnały muszą mieć zadany poziom, w drugim - wystarczy, że jeden z sygnałów ma poziom różny od zadanego;
- poziom sygnału analogowego: warunek jest spełniony, gdy badany sygnał analogowy przekroczy podaną wartość graniczną w określonym kierunku (mniejszych lub większych wartości)

Warunki startu:

- data: warunek jest spełniony, gdy bieżąca data zrówna się z zadaną;
- czas: warunek jest spełniony po upływie zadanego odcinka czasu.

Należy pamiętać o tym, że powyżej opisane warunki startu / stopu realizowane są programowo w związku z czym początek / koniec pomiaru jest zawsze nieco opóźniony względem momentu spełnienia warunku. Opóźnienie to (rzędu mikrosekund) zależne

jest od szybkości komputera.

Warunki startu związane z pomiarem czasu realizowane są w oparciu o czas systemowy. Z tego powodu czas podawany jest z dokładnością do sekundy.

Dla startu operacji zdefiniowano dodatkowo następujące warunki:

- start natychmiastowy: bez czekania na spełnienie jakichkolwiek warunków
- tylko dla przetwarzania a/c: sygnał sprzętowy; warunek jest spełniony, gdy na odpowiednie wejście na łączówce modułu zostanie podany sygnał cyfrowy o wartości 0; warunek ten tym różni się od pozostałych, że jest realizowany sprzętowo, natomiast pozostałe - programowo.

Dla zakończenia operacji zdefiniowano dodatkowo następujący warunek:

- przetworzenie określonej liczby próbek.

Kodowanie typów warunku startu operacji:

kod typu / znaczenie	parametry warunku				
	bajt 1	bajt 2	bajt 3	bajt 4	bajt 5
0 (LCO_SIMMED) natychmiast	----	----	----	----	----
1 (LCO_SHARD) od sygnału sprzętowego	----	----	----	----	----
2 (LCO_SLEVEL) ¹⁾ od poziomu sygnału cyfrowego	numer modułu	numer portu	numer wejścia	poziom	----
3 (LCO_SSLOPE) ²⁾ od zbocza sygnału cyfrowego	numer modułu	numer portu	numer wejścia	zbocze	----
4 (LCO_SDIG_EQ) ³⁾ od kombinacji wejść cyfrowych - warunek równości	numer modułu	numer portu	maska	wzorzec	----
5 (LCO_SDIG_NE) ³⁾ od kombinacji wejść cyfrowych - warunek nierówności	numer modułu	numer portu	maska	wzorzec	----
6 (LCO_STIME) po upłygnięciu określonego czasu	odcinek czasu w sekundach				----
7 (LCO_SDATE) ⁴⁾ o podanym czasie	sekunda	minuta	godzina	dzień miesiąca	----
8 (LCO_SANALOG) ⁵⁾ po przekroczeniu zadanej wartości analogowej	numer modułu	numer przet- wornika	numer kanału + kierunek przechr.	próg wyzwiania [mV]	

- 1) "Poziom" wskazuje oczekiwany stan wejścia cyfrowego (0/1). Wszystkie wartości różne od zera traktowane są jak "1".
- 2) Oczekiwane zbocze wejścia cyfrowego kodowane jest następująco:
0 - zbocze opadające, 1 - zbocze narastające.
- 3) Bajt maski wskazuje, które bity portu wejściowego brane są pod uwagę przy badaniu warunku: 1 wskazuje bit badany, 0 - ignorowany. 4) "Dzień miesiąca" dotyczy bieżącego miesiąca. Jeżeli numer dnia jest mniejszy niż bieżący - następnego miesiąca.
- 5) Numer kanału zapisywany jest na bitach $b_4..b_7$ natomiast najstarszy bit (b_8) określa kierunek przekroczenia progu wyzwalania:

b_8	kierunek
0	w dół
1	w górę

Dla modułów serii LC-030 numer przetwornika i numer kanału są ignorowane (wyzwalanie analogowe realizowane jest sprzętowo przez 8-bitowy przetwornik c/a przeznaczony tylko do tego celu).

Kodowanie typów warunku zatrzymania operacji:

kod typu (szesnastkowo) / znaczenie	parametry warunku				
	bajt 1	bajt 2	bajt 3	bajt 4	bajt 5
00 (LCO_ZSAMPLES) po zmierzeniu bloku danych	liczba próbek do zmierzenia				----
10 (LCO_ZBREAK) 1) programowo - po wykonaniu f. BREAK	----	----	----	----	----
20 (LCO_ZLEVEL) 1) od poziomu sygnału cyfrowego	numer modułu	numer portu	numer wejścia	poziom	----
30 (LCO_ZSLOPE) 1) od zbocza sygnału cyfrowego	numer modułu	numer portu	numer wejścia	zbocze	----
40 (LCO_ZDIG_EQ) 1) od kombinacji wejść cyfrowych - warunek równości	numer modułu	numer portu	maska	wzorzec	----
50 (LCO_ZDIG_NE) 1) od kombinacji wejść cyfrowych - warunek nierówności	numer modułu	numer portu	maska	wzorzec	----
60 (LCO_ZTIME) 1) po upłygnięciu określonego czasu	odcinek czasu w sekundach				----
70 (LCO_ZDATE) 1) o podanym czasie	sekunda	minuta	godzina	dzień	----
				miesiąca	

80 (LCO_ZANALOG) ¹⁾	numer	numer	numer	próg wyzwania
po przekroczeniu	modułu	przet-	kanalu +	[mV]
zadanej wartości		wornika	kierunek	
analogowej			przechr.	

¹⁾ W modułach serii LC-030 nie realizowany.

4. Funkcje drivera.

W poniższych rozdziałach opisano wszystkie funkcje driver'a. Każdy rozdział ma następującą strukturę:

- tabela zawierająca strukturę rekordu opisu zlecenia; w tabeli tej opisano każde pole rekordu w sposób następujący:
- nazwa pola; nazwa ta używana jest konsekwentnie w plikach źródłowych dotyczących języka C, Pascal i assemblera (patrz rozdział 6)
- rozmiar w bajtach; typ danej reprezentowanej przez to pole (np. czy jest to liczba ze znakiem czy bez) wynika ze znaczenia pola; w razie wątpliwości należy porównać z odpowiednim dla danego języka plikiem źródłowym deklarującym struktury danych (dodatki A, C i E)
- znaczenie
- przeznaczenie funkcji
- szczegółowy opis parametrów funkcji (pól rekordu opisu zlecenia); ten punkt został zamieszczony tylko wtedy, gdy uznano, że znaczenie parametru podane w tabeli jest niewystarczająco oczywiste
- ostrzeżenia; lista ostrzeżeń zwracanych przez funkcję w parametrze LCO_STATUS; jeżeli punkt ten nie występuje to oznacza to, że dana funkcja nie zwraca żadnych ostrzeżeń
- błędy; lista błędów zwracanych przez funkcję w parametrze LCO_STATUS; jeżeli punkt ten nie występuje to oznacza to, że dana funkcja nie zwraca żadnych błędów
- dodatkowe informacje o błędach; lista dodatkowych informacji zwracanych przez funkcję w parametrze LCO_ERR_STAT; jeżeli punkt ten nie występuje to oznacza to, że dana funkcja nie zwraca żadnych dodatkowych informacji (LCO_ERR_STAT zawsze równe LCO_E_OK: 0)

4.1. Inicjalizacja (MODULE_INIT).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 0
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_IMODULE	1	mapa modułów

Przeznaczenie:

Funkcja powoduje zainicjalizowanie pracy (zerowanie) wyspecyfikowanych modułów. Moduły podlegające inicjalizacji specyfikuje się na poszczególnych bitach parametru LCO_IMODULE:

```

b8           b1
- - - - D C B A
0 0 0 0 x x x x
x = 1 - zeruj moduł, x = 0 - nie zeruj modułu
    
```

Dodatkową czynnością wykonywaną przez funkcję jest synchronizacja wewnętrznego zegara czasu rzeczywistego z zegarem komputera. Synchronizowany zegar posługuje się przerwaniem $1C_1\sigma$ przechwyconym w trakcie rozpoczynania pracy przez driver (patrz p.3.1.). Synchronizacja polega na przepisaniu do niego czasu i daty systemowej, pobranej za pomocą funkcji DOS'a ($2C_1\sigma$ i $2A_1\sigma$). Z tego też powodu funkcji MODULE_INIT nie należy wykonywać w procedurach obsługi przerwania - może się to skończyć zawieszeniem pracy systemu operacyjnego.

Po wykonaniu funkcji nadal są pamiętane parametry ostatniego przetwarzania a/c i c/a, w związku z czym po inicjalizacji modułu można wykonać przetwarzanie ze zgaszonym bitem trybu pracy LCO_MOD_NEWPAR (patrz opis funkcji ANALOG_INPUT i ANALOG_OUTPUT).

Ostrzeżenia (LCO_STATUS):

LCO_NON_EX_MOD -- zarządzano inicjalizacji nie istniejących modułów ale co najmniej 1 moduł został zainicjalizowany

Błędy (LCO_STATUS):

LCO_NO_MODULE - nie istnieje żaden z żądanych modułów

4.2. Informacja o konfiguracji ogólnej (GET_TOTAL_CONFIGURATION).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 1
LCO_STATUS	1	kod zakończenia funkcji = LCO_OK
LCO_ERR_STAT	1	dodatkowe informacje o błędach
parametry wyjściowe:		
LCO_TONE	1	konfiguracja modułów
LCO_TIAD	1	liczba dostępnych przetworników a/c
LCO_TIDA	1	liczba dostępnych przetworników c/a
LCO_TCTC	1	liczba dostępnych kanałów CTC
LCO_TIDI	1	liczba dostępnych portów we. cyfrowych
LCO_TIDO	1	liczba dostępnych portów wy. cyfrowych
LCO_TMEMA ¹⁾	4	adres bufora w pamięci rozszerzonej
LCO_TMEML ¹⁾	4	długość bufora w pamięci rozszerzonej

¹⁾ Dla modułów serii LC-030 - nieokreślone (moduł nie współpracuje z pamięcią rozszerzoną).

Przeznaczenie:

Funkcja zwraca informację o sumarycznej konfiguracji wszystkich modułów danego typu. Bajt konfiguracji modułów ma następujący format:

```

b8           b1
- - - - D C B A
- - - - x x x x

```

x = 1 - moduł zainstalowany, x = 0 - modułu nie ma

4.3. Informacja o konfiguracji modułu (GET_MODULE_INFORMATION).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 2
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_MMODULE	1	numer modułu
parametry wyjściowe:		
LCO_MBASE1	2	adres bazowy modułu - pakiet sterujący
LCO_MBASE2	2	adres bazowy modułu - pakiet analogowy
LCO_MIAD	1	liczba dostępnych przetworników a/c
LCO_MIDA	1	liczba dostępnych przetworników c/a
LCO_MCTC	1	liczba dostępnych kanałów CTC
LCO_MIDI	1	liczba dostępnych portów we. cyfrowych
LCO_MIDO	1	liczba dostępnych portów wy. cyfrowych
LCO_MCLOCK	2	częstotliwość zegara modułu w kHz
LCO_MINT	1	numer przerwania (programowy)

Przeznaczenie:

Funkcja zwraca informację o konfiguracji wyspecyfikowanego modułu.
Błędy (LCO_STATUS):

LCO_NO_MODULE - nie ma takiego modułu

4.4. Informacja o szczegółach technicznych (GET_INFO).

4.4.1. Wejścia cyfrowe.

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 3
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_GTYPE	1	rodzaj urządzenia = 1
LCO_GMODULE	1	numer modułu
LCO_GNUM	1	numer portu wejściowego
parametry wyjściowe:		
LCO_GCHAN	1	liczba bitów portu

Przeznaczenie:

Funkcja zwraca informację o liczbie bitów badanego portu wejściowego.

Błędy (LCO_STATUS):

LCO_NONEX_DEV - port wejściowy o tym numerze nie istnieje
LCO_NO_MODULE - nie ma takiego modułu

4.4.2. Wyjścia cyfrowe.

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 3
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_GTYPE	1	rodzaj urządzenia = 2
LCO_GMODULE	1	numer modułu
LCO_GNUM	1	numer portu wyjściowego
parametry wyjściowe:		
LCO_GCHAN	1	liczba bitów portu

Przeznaczenie:

Funkcja zwraca informację o liczbie bitów badanego portu wyjściowego.

Błędy (LCO_STATUS):

LCO_NONEX_DEV - port wyjściowy o tym numerze nie istnieje
 LCO_NO_MODULE - nie ma takiego modułu

4.4.3. Wejścia analogowe.

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 3
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_GTYPE	1	rodzaj urządzenia = 3
LCO_GMODULE	1	numer modułu
LCO_GNUM	1	numer przetwornika
parametry wyjściowe:		
LCO_GCHAN	1	liczba kanałów przetwornika
LCO_GRES	1	liczba bitów przetwornika
LCO_GTIME	2	czas konwersji przetwornika w ns
LCO_GMINV	1	dolna granica zakresu napięć w dziesiątych częściach wolta
LCO_GMAXV	1	górną granicę zakresu napięć w dziesiątych częściach wolta
LCO_GDMA	1	numer kanału DMA
LCO_GMINP	64	tablica minimalnych okresów próbkowania

Przeznaczenie:

Funkcja zwraca informację o konfiguracji badanego toru analogowo-cyfrowego.

Tablica LCO_GMINP zawiera (2-bajtowe) wartości minimalnych okresów próbkowania w 1, 2, 3, ..., 16 kanałach. Okresy podane są w dziesiątych częściach mikrosekundy. Rozmiar tablicy jest większy (dla 32 kanałów) dla zgodności z driver'ami innych modułów serii LC-...

Błędy (LCO_STATUS):

LCO_NONEX_DEV - przetwornik a/c o tym numerze nie istnieje
 LCO_NO_MODULE - nie ma takiego modułu

4.4.4. Wyjścia analogowe.

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 3
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dotatkowe informacje o błędach
LCO_GTYPE	1	rodzaj urządzenia = 4
LCO_GMODULE	1	numer modułu
LCO_GNUM	1	numer przetwornika
parametry wyjściowe:		
LCO_GCHAN	1	liczba kanałów przetwornika
LCO_GRES	1	liczba bitów przetwornika
LCO_GTIME	2	czas konwersji przetwornika w ns
LCO_GMINV	1	dolna granica zakresu napięć w dziesiątych częściach wolta
LCO_GMAXV	1	górną granicę zakresu napięć w dziesiątych częściach wolta
LCO_GDMA	1	numer kanału DMA

Przeznaczenie:

Funkcja zwraca informację o konfiguracji badanego toru cyfrowo-analogowego.

Błędy (LCO_STATUS):

LCO_NONEX_DEV - przetwornik c/a o tym numerze nie istnieje
 LCO_NO_MODULE - nie ma takiego modułu

4.4.5. Kanały układów licznikowo-czasowych (CTC).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 3
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dotatkowe informacje o błędach
LCO_GTYPE	1	rodzaj urządzenia = 5
LCO_GMODULE	1	numer modułu
LCO_GNUM	1	numer kanału

Przeznaczenie:

Funkcja informuje czy badany kanał CTC istnieje (przez kod odpowiedzi).

Błędy (LCO_STATUS):

LCO_BAD_DEV_TYP - brak dostępnych kanałów CTC
 LCO_NONEX_DEV - kanał o tym numerze nie istnieje lub jest niedostępny
 LCO_NO_MODULE - nie ma takiego modułu

Uwaga: moduł LC-030-1612 nie zawiera kanałów CTC dostępnych dla użytkownika.

4.4. Zdeklarowanie częstotliwości zegara (SET_CLOCK).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 4
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_LCLOCK	2	częstotliwość zegara w kHz

Przeznaczenie:

Funkcja nie jest realizowana.

4.6. Ustawienie zakresu napięć (SET_VOLTAGE_RANGE).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 5
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_VTYPE	1	rodzaj urządzenia (3)
LCO_VMODULE	1	numer modułu
LCO_VNUM	1	numer przetwornika
LCO_VMINV	1	dolna granica zakresu napięć w dziesiątych częściach wolta
LCO_VMAXV	1	górną granicę zakresu napięć w dziesiątych częściach wolta

Przeznaczenie:

Funkcja zmienia zakres napięć pracy przetwornika a/c. Zakres powinien być jednym z czterech opisanych w dokumentacji techniczno-ruchowej modułu (0..10, 0..5, +/-5, +/-2.5, tzn. parametry LCO_VMINV i LCO_VMAXV powinny mieć wartości: 0/100, 0/50, -50/50 lub -25/25).

Błędy (LCO_STATUS):

LCO_BAD_RANGE - błędny zakres napięć
 LCO_NONEX_DEV - przetwornik danego typu o tym numerze nie istnieje
 LCO_NO_MODULE - nie ma takiego modułu

4.7. Zadeklarowanie maksymalnego czasu obsługi obcych przerw (SET_TIME).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 6
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_ETIME	2	maksymalny czas obsługi w dziesiątych częściach μ s

Przeznaczenie:

Funkcja nie jest realizowana.

4.8. Oczekiwanie na zakończenie operacji (WAIT_FOR_END).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 7
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dotatkowe informacje o błędach
LCO_WTYPE	1	rodzaj urządzenia (3)
LCO_WMODULE	1	numer modułu
LCO_WNUM	1	numer przetwornika
LCO_WMODE	1	tryb pracy funkcji
parametry wyjściowe:		
LCO_WRMNUM	4	rzeczywista liczba zmierzonych próbek
LCO_WREMAR ¹⁾	2	rzeczywista długość marginesu końcowego

¹⁾ Dla modułów serii LC-030 parametr ten jest zawsze zerowy.

Przeznaczenie:

tryb	nazwa	znaczenie
0	LCO_W_WAIT	Oczekiwanie na zakończenie operacji asynchronicznej w podanym urządzeniu (patrz opis funkcji ANALOG_INPUT)
1	LCO_W_TEST	Sprawdzenie czy operacja asynchroniczna w podanym urządzeniu zakończyła się (patrz opis funkcji ANALOG_INPUT); jeżeli tak to zwracany jest status LCO_OK w przeciwnym razie - ostrzeżenie LCO_IN_PROGRESS
2	LCO_W_FINISHED	zasygnalizowanie driver'owi zakończenia transmisji; musi wystąpić przy pracy asynchronicznej z przerwaniem

Ostrzeżenia (LCO_STATUS):

LCO_IN_PROGRESS - transmisja trwa; *de facto* nie jest to ostrzeżenie

Błędy (LCO_STATUS):

LCO_BAD_DEV_TYP - brak przetworników danego typu

LCO_BAD_MODE - błędny tryb pracy funkcji

LCO_BROKEN - operacja przerwana przez funkcję BREAK

LCO_NONEX_DEV - przetwornik o tym numerze nie istnieje

LCO_NO_MODULE - nie ma takiego modułu

LCO_NO_OPER - z wyspecyfikowanym urządzeniem nie jest związana żadna operacja asynchroniczna odbywająca się samodzielnie (nie można oczekiwać na operacje odbywające się pod kontrolą driver'a)

Dodatkowe informacje (LCO_ERR_STAT):

LCO_E_BROKEN_WAIT - funkcja przzerwana w trakcie oczekiwania na spełnienie warunku startu

LCO_E_BROKEN_RUN - funkcja przzerwana w trakcie przetwarzania

4.9. Przerwanie operacji (BREAK).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 8
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_BMODE	1	tryb pracy funkcji
LCO_BPROC	4	adres procedury obsługi Ctrl-Break

Przeznaczenie:

Przerywanie pracy driver'a i modułu, instalowanie i wyinstalowywanie procedur obsługi przzerwania generowanego przez klawisze Ctrl-Break.

Tryby pracy:

nazwa	wartość	znaczenie
LCO_BREAK_EXEC	0	przerwij pracę driver'a i modułu
LCO_BREAK_INST	1	zainstaluj procedurę obsługi przzerwania Ctrl-Break
LCO_BREAK_UNINST	2	wyinstaluj procedurę obsługi przzerwania Ctrl-Break

Przerwanie pracy - tryb LCO_BREAK_EXEC:

Działanie funkcji polega na przzerwaniu wszystkich operacji wykonywanych w module. W momencie wywołania tej funkcji driver i moduł mogą znajdować się w jednym z trzech stanów:

1. Nie jest wykonywana żadna operacja - driver zwraca błąd LCO_STATUS = LCO_NO_OPER, LCO_ERR_STAT = LCO_E_OK.
2. Moduł wykonuje operację pod kontrolą driver'a: początkową część operacji asynchronicznej (patrz opis funkcji ANALOG_INPUT) lub operację synchroniczną. Ponieważ w tym przypadku należy przerwać pracę driver'a, zazwyczaj funkcja wywoływana jest w procedurze obsługi przzerwania, np. Ctrl-Break. Driver zwraca LCO_STATUS = LCO_OK i LCO_ERR_STAT = LCO_E_OK.
3. Moduł wykonuje samodzielnie operację asynchroniczną (bez kontroli driver'a; patrz opis funkcji ANALOG_INPUT). Funkcja może być wywoływana zarówno przez program główny (np. gdy program sam stwierdzi, że wykonanie operacji trwa za długo) jak i przez procedurę obsługi przzerwania (np. po interwencji operatora Ctrl-Break). Driver zwraca LCO_STATUS = LCO_OK natomiast LCO_ERR_STAT zależy od stanu przetwarzania: LCO_E_BROKEN_WAIT (oczekiwanie na spełnienie warunku startu - start od sygnału sprzętowego LCO_SHARD) lub LCO_E_BROKEN_RUN (przetwarzanie w toku).

Zainstalowanie procedury obsługi - tryb LCO_BREAK_INST:

Funkcja instaluje obsługę Ctrl-Break przez przechwycenie przzerwania 1B16 i podłożenie nowej procedury obsługi tego przzerwania. Użytkownik może podać adres

własnej procedury obsługi przerwania (LCO_BPROC, daleki adres offset-segment; jest to procedura typu interrupt) lub zlecić obsługę standardową - przez procedurę wewnętrzną driver'a (LCO_BPROC = 0:0). Procedura obsługi przerwania musi zawierać wykonanie funkcji BREAK w trybie pracy LCO_BREAK_EXEC (patrz wyżej; to też jest jedyną treścią standardowej procedury obsługi, dostarczanej przez driver).

Wyinstalowanie procedury obsługi - tryb LCO_BREAK_UNINST:

Funkcja przywraca procedurę obsługi przerwania istniejącą przed pierwszym wywołaniem funkcji BREAK w trybie LCO_BREAK_INST. Istotne jest więc aby zadbać o wyinstalowanie obsługi Ctrl-Break przed zakończeniem programu!

Procedura obsługi wyinstalowywana jest również przez funkcję LEAVE_DRIVER (patrz).

Błędy (LCO_STATUS):

LCO_INTR_INST - procedura obsługi Ctrl-Break jest już zainstalowana (dla żądania zainstalowania procedury)
 LCO_NO_OPER - z żadnym modulem nie jest związana żadna operacja w toku (dla trybu pracy LCO_BREAK_EXEC)
 LCO_BAD_MODE - błędny tryb pracy

Dodatkowe informacje (LCO_ERR_STAT; tylko dla trybu pracy LCO_BREAK_EXEC):

LCO_E_BROKEN_RUN - funkcja wywołana w trakcie przetwarzania przy operacji asynchronicznej
 LCO_E_BROKEN_WAIT - funkcja wywołana w trakcie oczekiwania na spełnienie warunku startu przy operacji asynchronicznej

4.10. Wejście cyfrowe (DIGITAL_INPUT).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 9
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_DMODULE	1	numer modułu
LCO_DNUM	1	numer portu
LCO_DSTST	1	typ warunku startu operacji
LCO_DVAL	1	odczytana wartość
LCO_DSTART	5	parametry warunku startu

Przeznaczenie:

Funkcja odczytuje stan wejść cyfrowych podanego portu. Warunkiem startu operacji nie może być LCO_SHARD (sygnał sprzętowy).

Błędy (LCO_STATUS):

LCO_ILL_START - błędne parametry warunku startu operacji
 LCO_ILL_START_CODE - błędny typ warunku startu operacji
 LCO_NONEX_DEV - port wejściowy o tym numerze nie istnieje
 LCO_NO_MODULE - nie ma takiego modułu

Dodatkowe informacje o błędnych parametrach warunku startu operacji (LCO_ERR_STAT):

LCO_E_BAD_CHAN - numer nieistniejącego kanału
 LCO_E_BAD_DATE - zła specyfikacja daty
 LCO_E_BAD_THRE - błędny próg wyzwiania analogowego

LCO_E_BAD_TIME - zły odcinek czasu
 LCO_E_NONEX_DEV - nie istnieje urządzenie o tym numerze
 LCO_E_NO_MODULE - nie ma takiego modułu

4.11. Wyjście cyfrowe (DIGITAL_OUTPUT).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 10
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_OMODULE	1	numer modułu
LCO_ONUM	1	numer portu
LCO_OSTST	1	typ warunku startu operacji
LCO_OVAL	1	zapisywana wartość
LCO_OSTART	5	parametry warunku startu

Przeznaczenie:

Wysłanie wartości na wyjścia cyfrowe podanego portu. Warunkiem startu operacji nie może być LCO_SHARD (sygnał sprzętowy).

Błędy (LCO_STATUS):

LCO_ILL_START - błędne parametry warunku startu operacji
 LCO_ILL_START_CODE - błędny typ warunku startu operacji
 LCO_NONEX_DEV - port wyjściowy o tym numerze nie istnieje
 LCO_NO_MODULE - nie ma takiego modułu

Dodatkowe informacje o błędnych parametrach warunku startu operacji (LCO_ERR_STAT):

LCO_E_BAD_CHAN - numer nieistniejącego kanału
 LCO_E_BAD_DATE - zła specyfikacja daty
 LCO_E_BAD_THRE - błędny próg wyzwalania analogowego
 LCO_E_BAD_TIME - zły odcinek czasu
 LCO_E_NONEX_DEV - nie istnieje urządzenie o tym numerze
 LCO_E_NO_MODULE - nie ma takiego modułu

4.12. Zapis CTC (CTC_WRITE).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 11
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_CMODULE	1	numer modułu
LCO_CMODE	1	tryb pracy funkcji
LCO_CFUN	1	tryb pracy kanału CTC
LCO_CVAL	2	wpisywana wartość licznika

Przeznaczenie:

Zaprogramowanie licznika CTC. W module LC-030-1612 nie ma dostępnych kanałów CTC w związku z tym funkcja zwraca zawsze błąd:

LCO_BAD_DEV_TYP - brak układów CTC

4.13. Odczyt CTC (CTC_READ).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 12
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_UMODULE	1	numer modułu
LCO_UNUM	1	numer kanału CTC
parametry wyjściowe:		
LCO_UVAL	2	odczytana wartość licznika

Przeznaczenie:

Odczyt licznika CTC. W module LC-030-1612 nie ma dostępnych kanałów CTC w związku z tym funkcja zwraca zawsze błąd:

LCO_BAD_DEV_TYP - brak układów CTC

4.14. Transmisja danych (DATA_TRANSMIT).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 13
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_RMODE ¹⁾	1	tryb przesyłania
LCO_RADDR	4	adres bufora (offset-segment)
LCO_RLEN	4	długość bufora
LCO_RMEAS	4	numer próbki, od której należy zacząć
LCO_RNUM	4	liczba próbek do przesłania
LCO_RMEMA ²⁾	4	adres bufora w pamięci rozszerzonej
parametry wyjściowe:		
LCO_RRNUM	4	rzeczywista liczba przesłanych próbek

¹⁾ Dla modułu LC-030-1612 musi być równy 0.

²⁾ Dla modułu LC-030-1612 ignorowany.

Przeznaczenie:

Funkcja służy do przesyłania ciągu zmierzonych próbek z pamięci modułu do bufora w pamięci komputera.

Znaczenie poszczególnych parametrów:

a. LCO_RADDR

Adres bufora w podstawowej pamięci komputera. Do niego przesłane zostaną próbki z pamięci modułu.

b. LCO_RLEN

Długość bufora LCO_RADDR podana w próbkach. Ten parametr nie jest związany z liczbą próbek do przesłania.

c. LCO_RMEAS

Numer pierwszej próbki do przesłania. Przesłane zostaną próbki o numerach LCO_RMEAS, LCO_RMEAS + 1 itd. Należy pamiętać, że pierwsza zmierzona próbka ma numer 1. Jeżeli mierzonych było np. 5 kanałów to 10 próbka w pierwszym kanale ma

numer 51. Uwaga: LCO_RMEAS musi wskazywać na pierwszą próbkę w serii pomiarowej, w przeciwnym razie driver zwraca błąd LCO_BAD_MNUM. W podanym przykładzie błędnymi byłyby numery: 2, 3, 4, 5.

d. LCO_RNUM

Całkowita liczba próbek do przesłania.

e. LCO_RRNUM

Rzeczywista liczba przepisanych próbek. Parametr LCO_RRNUM przybiera wartość będącą minimum z LCO_RLEN, LCO_RNUM i liczby próbek ostatnio zmierzonych. Dodatkowo LCO_RRNUM jest obcinane do takiej wartości, aby przepisane zostały całkowite serie pomiarowe.

Uwaga: zarówno w pamięci własnej modułu jak i w buforze w pamięci podstawowej komputera próbki ułożone są w kolejności od najstarszego kanału do najmłodszego (patrz dokumentacja techniczno-ruchowa);

Ostrzeżenia (LCO_STATUS):

LCO_OTHER_LEN - przepisano mniejszą liczbę próbek niż żądano

Błędy (LCO_STATUS):

LCO_BAD_BUF_ADR - błędny adres bufora (nie odnoszący się do pamięci podstawowej)
 LCO_BAD_BUF_LEN - błędna długość bufora (powodująca wyjście bufora poza pamięć itp.)
 LCO_BAD_MNUM - błędny numer pierwszego pomiaru (np. większy niż liczba zmierzonych próbek, wypadający wewnątrz serii pomiarowej)
 LCO_BAD_MODE - błędny tryb pracy
 LCO_BROKEN - transmisja przerwana z powodu wykonania funkcji BREAK

4.15. Przetwarzanie analogowo-cyfrowe (ANALOG_INPUT).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 14
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_AMODULE	1	numer modułu
LCO_ANUM	1	numer przetwornika
LCO_AMODE	2	tryb pracy funkcji
LCO_ASTST	1	typy warunków startu / stopu operacji
LCO_APER	4	okres próbkowania
LCO_APER2	2	wielokrotność okresu próbkowania dla kanałów dodatkowych
LCO_ACHAN	1	liczba kanałów podstawowych / numer kanału
LCO_ACHAN2	1	liczba kanałów dodatkowych
LCO_AADDR	4	adres bufora (offset-segment)
LCO_ALEN	4	długość bufora
LCO_AMEMA ¹⁾	4	adres bufora w pamięci rozszerzonej
LCO_ABMAR ¹⁾	2	długość marginesu początkowego
LCO_AEMAR ¹⁾	2	długość marginesu końcowego
LCO_AHAND ¹⁾	2	numer handlera pliku dyskowego
LCO_ASTART	5	warunki startu
LCO_ASTOP	5	warunki stopu
parametry wyjściowe:		
LCO_ARDIV1	2	rzeczywisty dzielnik zegara modułu
LCO_ARDIV2	2	- " -

LCO_ARMNUM	4	rzeczywista liczba zmierzonych próbek
LCO_ARBMAR ²⁾	2	rzeczywista dł. marginesu początkowego
LCO_AREMAR ²⁾	2	rzeczywista dł. marginesu końcowego
LCO_ARLEN	4	rzeczywista liczba przepisanych próbek
LCO_ARBUF ²⁾	2	początek bufora cyklicznego

¹⁾ Dla modułu LC-030-1612 parametry te są ignorowane.

²⁾ Dla modułu LC-030-1612 parametry te są zawsze zerowe.

Przeznaczenie:

Jest to podstawowa funkcja driver'a sterująca główną częścią modułu - torem przetwarzania analogowo - cyfrowego.

Generalnie pomiar (w trybie blokowym) można podzielić na następujące fazy:

1. Przygotowanie toru pomiarowego do pracy.
2. Oczekiwanie na spełnienie warunków startu pomiaru.
3. Start właściwego pomiaru.
4. Oczekiwanie na spełnienie warunków stopu pomiaru.
5. Zakończenie pomiaru - w szczególności przetransmitowanie żądanej części zmierzonych próbek do pamięci komputera.

Znaczenie poszczególnych parametrów:

a. LCO_AMODE

Parametr określa tryb pracy funkcji przy czym znaczenie mają kolejne bity parametru:

nazwa	nr bitu	uwagi
LCO_MOD_START	1	
LCO_MOD_NEW_PAR	2	
LCO_MOD_SYNCHR	3	
LCO_MOD_INTR	4	
LCO_MOD_INTR_TYPE	5	zarezerwowane; zawsze 0
LCO_MOD_BLOCK	6	
LCO_MOD_CYCL	7	zarezerwowane; zawsze 0
LCO_MOD_FILE	8	zarezerwowane; zawsze 0
LCO_MOD_MEM_W	9	
LCO_MOD_EXT_CLK	10	
LCO_MOD_EXT_MEM	11	zarezerwowane; zawsze 0
-----	12..16	zarezerwowane; zawsze 0

LCO_MOD_START:

- start pomiarów

ustawienie tego bitu na 1 oznacza żądanie wykonania przetwarzania; wartość 0 powoduje jedynie analizę poprawności i zapamiętanie parametrów funkcji

LCO_MOD_NEW_PAR:

- ustawienie nowych parametrów

1 oznacza, że parametry przetwarzania pobierane będą z rekordu opisu zlecenia; 0 oznacza, że parametry przetwarzania będą identyczne jak poprzednio - jeżeli do tej pory nie było wykonania funkcji ANALOG_INPUT lub ostatnie było niepoprawne to driver zgłosi błąd LCO_NO_PARAMS

LCO_MOD_SYNCHR:

- rodzaj pracy: synchroniczna (1) / asynchroniczna (0):

- praca synchroniczna: driver zwraca sterowanie dopiero po całkowitym zakończeniu przetwarzania; uwaga: ustawienie bitu LCO_MOD_INTR na 1 wymusza pracę asynchroniczną;

- praca asynchroniczna: driver zwraca sterowanie natychmiast po zaprogramowaniu

modułu (warunki startu: LCO_SHARD, LCO_SANALOG) lub po wystartowaniu przetwarzania (pozostałe warunki startu); dalsze działanie zależne jest od bitu LCO_MOD_INTR

LCO_MOD_INTR:

- rodzaj pracy: z przerwaniem (1) / bez przerwania (0):
- praca z przerwaniem: użytkownik dostarcza procedurę wywoływaną w obsłudze przerwania (patrz funkcji INTERRUPT_SERVICE); przerwanie zgłaszane jest dwukrotnie: po wystartowaniu pomiarów i po ich zakończeniu, tj. po zmierzeniu zadanej liczby próbek; rozróżnienie tych przerw jest możliwe dzięki słowu komunikacyjnemu procedury obsługi przerwania: po starcie słowo to ma ustawiony bit LCO_IS_START, po zakończeniu - LCO_IS_END; zażądanie pracy z przerwaniem wymusza pracę w trybie asynchronicznym (patrz wyżej);
- praca bez przerwania: jeżeli funkcja wykonywana jest w trybie synchronicznym to daje to razem najprostszyspósb przeprowadzania pomiaru; jeżeli natomiast zażądano pracy w trybie asynchronicznym to po oddaniu sterowania przez driver programowi nadrzędnemu moduł kontynuuje pomiar (lub oczekiwanie na spełnienie warunku startu, patrz opis bitu LCO_MOD_SYNCHR) przy czym koniec przetwarzania nie jest sygnalizowany przerwaniem; program nadrzędny może dowiedzieć się o stan przetwarzania wykonując funkcję WAIT_FOR_END w trybie LCO_W_TEST lub zlecić oczekiwanie na koniec wykonując tę samą funkcję w trybie LCO_W_WAIT

LCO_MOD_BLOCK:

- tryb przetwarzania: blokowy (1) / pojedynczy (0)
- tryb blokowy: podstawowy tryb pracy modułu, w którym do pamięci własnej modułu wpisywany jest blok zmierzonych wartości;
uwaga: zarówno w pamięci własnej modułu jak i w buforze w pamięci podstawowej komputera próbki ułożone są w kolejności od najstarszego kanału do najmłodszego (patrz dokumentacja techniczno-ruchowa);
- tryb pojedynczy: pomiar prowadzony jest bezpośrednio do bufora określonego przez użytkownika w pamięci podstawowej komputera; mierzone jest tylko po jednej próbce z każdego zadeklarowanego kanału; przy pierwszym wykonaniu funkcji w słowie trybu pracy (LCO_AMODE) powinien być ustawiony bit LCO_MOD_NEW_PAR (wówczas istotne są tylko parametry: LCO_AMODULE, LCO_ANUM, LCO_AMODE, LCO_ACHAN, LCO_AADDR, LCO_ALEN, LCO_ASTST i LCO_ASTART; parametr LCO_ACHAN2 traktowany jest tak, jakby był równy zero; nie są analizowany okres próbkowania LCO_APER ani warunki stopu przetwarzania (część LCO_ASTST oraz LCO_ASTOP) jako nie mające w tym kontekście sensu); przy następnych wykonaniach funkcji bit ten może być (choć nie jest to konieczne) zgaszony i wówczas istotne będą parametry: LCO_AMODULE, LCO_ANUM, LCO_AMODE i LCO_AADDR; wymagane jest zadeklarowanie adresu bufora z pamięci podstawowej; w słowie trybu pracy (LCO_AMODE) analizowane są tylko bity LCO_MOD_START i LCO_MOD_NEW_PAR (nie licząc opisywanego!), pozostałe - ignorowane
uwaga: należy zwrócić uwagę, że jeżeli chcemy w trybie pracy pojedynczej wykonać pomiar bloku próbek, gdzie warunek startu odnosi się do całego bloku to drugi i kolejne pomiary należy wykonywać z warunkiem startu LCO_SIMMED, niezależnie od warunku startu bloku

LCO_MOD_MEM_W:

- przepisanie do pamięci (1) / bez przepisywania do pamięci (0) bit steruje działaniem funkcji po zakończeniu przetwarzania w trybie synchronicznym; jeżeli bit jest równy 1 to po zakończeniu przetwarzania do pamięci komputera (pod adres LCO_AADDR) przepisywane jest <n> pierwszych próbek gdzie <n> jest minimum z całkowitej liczby zmierzonych próbek (LCO_ARMNUM) i rozmiaru bufora (LCO_ALEN); <n> jest obcinane do takiej wartości, aby przepisane zostały całkowite serie pomiarowe; całkowita liczba przepisanych próbek zwracana jest w parametrze LCO_ARLEN; jeżeli natomiast bit jest równy 0 to po zakończeniu przetwarzania dane nie są przepisywane do pamięci (wówczas parametry LCO_AADDR i LCO_ALEN nie są analizowane); zarówno w jednym jak i drugim przypadku transmisja danych jest możliwa poprzez wykonanie funkcji DATA_TRANSMIT

LCO_MOD_EXT_CLK:

- zadeklarowanie źródła sygnału taktującego pracę modułu w trybie blokowym; 0 oznacza taktowanie zegarem wewnętrznym modułu; jeżeli natomiast bit ten zostanie ustawiony na 1 to moduł będzie taktowany zewnętrznym sygnałem zegarowym (musi to być sygnał w standardzie TTL, prostokątny; czas trwania poziomu 0 musi być mniejszy niż minimalny okres próbkowania dla mierzonej liczby kanałów - patrz dokumentacja techniczno-ruchowa); dla pracy z zegarem zewnętrznym okres próbkowania (parametr LCO_APER) jest ignorowany i nie można deklarować typów warunku stopu: LCO_ZTIME i LCO_ZDATE

b. LCO_ASTST

Parametr określa typy warunków startu i stopu (ten drugi tylko dla pracy blokowej) pomiaru. Jest on sumą odpowiednich kodów typów warunku startu i stopu (patrz p.3.4.). 0 ile dopuszczalne są wszystkie opisane typy warunku startu to jako typ warunku stopu można podać:

- LCO_ZSAMPLES - po zmierzeniu podanej liczby próbek
- LCO_ZTIME i LCO_ZDATE - po upływie określonej liczby sekund lub o podanej dacie; te typy warunku stopu dozwolone są tylko przy pracy z zegarem wewnętrznym (bit LCO_MOD_EXTCLK w parametrze LCO_AMODE ustawiony na 0); dodatkowo typ warunku stopu LCO_ZDATE dozwolony jest tylko dla typów warunku startu: LCO_SIMMED, LCO_STIME lub LCO_SDATE.

c. LCO_ASTART i LCO_ASTOP

Parametry te określają szczegółowo warunki startu i stopu operacji. Interpretacja ich zależna jest od zadanych typów warunków startu i stopu operacji (LCO_ASTST). Szczegółowy opis - patrz p.13.4.

d. LCO_APER

Okres próbkowania dla pracy blokowej z zegarem wewnętrznym (bity parametru LCO_AMODE LCO_MOD_BLOCK = 1, LCO_MOD_EXTCLK = 0). Okres ten podawany jest w dziesiątych częściach mikrosekundy (np. 10000 oznacza 1 ms). Okres ten nie może być mniejszy niż minimalny okres podawany przez dokumentację dla danej liczby kanałów. Liczba kanałów określająca minimalny okres próbkowania wyznaczana jest następująco:

- dla pracy jednokanałowej (najstarszy bit parametru LCO_ACHAN równy 1, patrz niżej opis tego parametru) - oczywiście 1
- dla pracy wielokanałowej z jedną częstotliwością (najstarszy bit parametru LCO_ACHAN równy 0, parametr LCO_ACHAN2 = 0, patrz opis tego parametru) - wartość LCO_ACHAN
- dla pracy wielokanałowej z dwoma częstotliwościami (najstarszy bit parametru LCO_ACHAN równy 0, parametr LCO_ACHAN2 różny 0) - LCO_ACHAN + LCO_ACHAN2

O minimalnych okresach próbkowania można - należy - się dowiedzieć z programu za pomocą funkcji GET_INFO (parametr LCO_GMINP, patrz opis funkcji).

Z parametrem LCO_APER związane są parametry zwrotne driver'a LCO_ARDIV1 i LCO_ARDIV2 (patrz opis poniżej).

e. LCO_APER2

Parametr istotny tylko dla pracy blokowej, wielokanałowej, z dwoma częstotliwościami. Określa wielokrotność okresu próbkowania kanałów podstawowych. Oznacza to, że o ile standardowo próbkowanych jest LCO_ACHAN kanałów (z okresem LCO_APER) to co LCO_APER2 takich serii pomiarowych próbkowanych jest LCO_ACHAN + LCO_ACHAN2 kanałów. Parametr ten musi być większy od 1.

f. LCO_ACHAN

Parametr określający tryb pracy modułu: jednokanałowo / wielokanałowo oraz (odpowiednio): liczbę kanałów / numer kanału:

b ₈	tryb pracy	b ₁ ..b ₇
0	praca wielokanałowa	liczba kanałów (2..16)
1	praca jednokanałowa	numer kanału (1..16)

g. LCO_ACHAN2

Liczba kanałów dodatkowych w pracy z dwoma częstotliwościami. Analizowany jest tylko w przypadku pracy blokowej wielokanałowej - wówczas musi mieć wartość 0..(161-LCO_ACHAN). Jeżeli ma wartość 0 to moduł pracuje z jedną częstotliwością. Przy pracy w trybie pojedynczym lub przy pracy jednokanałowej jest ignorowany. Patrz opis parametru LCO_APER2.

h. LCO_AADDR, LCO_ALEN

Adres i długość bufora w pamięci podstawowej. Parametry te są analizowane tylko wtedy, gdy bit LCO_MOD_MEM_W trybu pracy (LCO_AMODE) został ustawiony na 1. Adres bufora jest adresem dalekim, tzn. podanym w postaci offset-segment, natomiast długość bufora podawana jest w próbkach. Długość bufora jest jednym z czynników określających liczbę próbek przepisywanych do pamięci komputera (patrz opis bitu LCO_MOD_MEM_W).

i. LCO_ARDIV1, LCO_ARDIV2

Częstotliwość próbkowania tworzona jest przez podział (przez całkowitą wartość) częstotliwości generatora w module. Poza tym dzielnik ten jest faktycznie iloczynem dwóch liczb (dwa połączone liczniki 16-bitowe). W związku z tym okres próbkowania zadany przez użytkownika nie zawsze jest osiągalny. W takim przypadku wybierany jest najbliższy możliwy okres, początkowo mniejszy a jeżeli jest to niemożliwe - większy od zadanego. Parametr LCO_ARDIV1 przekazuje wartość jednego z liczników natomiast LCO_ARDIV2 - drugiego (rzeczywisty dzielnik zegara modułu jest równy iloczynowi LCO_ARDIV1 * LCO_ARDIV2). Łącznie z częstotliwością zegara modułu (patrz funkcja GET_MODULE_INFORMATION, parametr LCO_MCLOCK) daje informację o faktycznej częstotliwości próbkowania.

j. LCO_ARMNUM

Rzeczywista liczba zmierzonych próbek (we wszystkich kanałach). Parametr ten jest istotny tylko dla typów warunku stopu: LCO_ZTIME i LCO_ZDATE, dla typu LCO_ZSAMPLES będzie równy zadanej liczbie próbek.

k. LCO_ARLEN

Rzeczywista liczba próbek przepisanych do pamięci komputera. Patrz opis bitu LCO_MOD_MEM_W w parametrze LCO_AMODE.

Błędy (LCO_STATUS):

LCO_BAD_BUF_ADR - błędny adres bufora (nie odnoszący się do pamięci podstawowej; dla pracy blokowej z przepisaniem do pamięci)
 LCO_BAD_BUF_LEN - błędna długość bufora (powodująca wyjście bufora poza pamięć; dla pracy blokowej z przepisaniem do pamięci)
 LCO_BAD_CHAN - numer nieistniejącego kanału (< 1 lub > 16)
 LCO_BAD_CHAN_N - zła liczba kanałów przy pracy wielokanałowej (LCO_ACHAN < 2 lub LCO_ACHAN + LCO_ACHAN2 > 16)
 LCO_BAD_MODE - błędny tryb pracy funkcji
 LCO_BAD_PER - za krótki okres próbkowania
 LCO_BAD_PER2 - błędna wielokrotność okresu próbkowania (0 lub 1; dla pracy z dwoma częstotliwościami, LCO_ACHAN2 <> 0)
 LCO_BROKEN - pomiar przerwany z powodu wykonania funkcji BREAK;
 LCO_ERR_STAT podaje, w jakim momencie operacja została przerwana
 LCO_DEV_BUSY - urządzenie zajęte (próba wykonania funkcji ANALOG_INPUT w momencie, gdy nie zakończono jeszcze poprzedniego przetwarzania)
 LCO_ILL_START - błędne parametry warunku startu

LCO_ILL_STOP - błędne parametry warunku stopu; w obu przypadkach sprecyzowanie błędu podane jest w LCO_ERR_STAT
 LCO_ILL_START_CODE - nielegalny typ warunku startu
 LCO_ILL_STOP_CODE - nielegalny typ warunku stopu
 LCO_INTR_NOT_INST - procedura obsługi przerwania nie jest zainstalowana (dla pracy z przerwaniem)
 LCO_NONEX_DEV - nie istnieje przetwornik o tym numerze
 LCO_NO_DMA - z danym urządzeniem nie jest związany żaden kanał DMA (dla pracy blokowej)
 LCO_NO_IRQ - z danym modułem nie jest związane żadne przerwanie (dla pracy z przerwaniem)
 LCO_NO_MODULE - nie ma takiego modułu
 LCO_NO_PARAMS - zgaszono bit LCO_MOD_NEWPAR (LCO_AMODE) ale wcześniej nie ustawiono żadnych parametrów (nie było wykonania funkcji ANALOG_INPUT lub ostatnie wykonanie było niepoprawne)

Dodatkowe informacje o błędach (LCO_ERR_STAT):

LCO_E_BAD_CHAN - numer nieistniejącego kanału
 LCO_E_BAD_DATE - zła specyfikacja daty
 LCO_E_BAD_THRE - błędny próg wyzwalania analogowego
 LCO_E_BAD_TIME - zły odcinek czasu
 LCO_E_BAD_LEN - zła długość pomiaru (więcej niż pamięć modułu)
 LCO_E_BROKEN_RUN - funkcja przerwana w trakcie przetwarzania
 LCO_E_BROKEN_WAIT - funkcja przerwana w trakcie oczekiwania na spełnienie warunku startu
 LCO_E_NONEX_DEV - nie istnieje urządzenie o tym numerze
 LCO_E_NO_MODULE - nie ma takiego modułu

4.16. Przetwarzanie cyfrowo-analogowe (ANALOG_OUTPUT).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 15
LCO_STATUS	1	kod zakończenia funkcji
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_NMODULE	1	numer modułu
LCO_NNUM	1	numer przetwornika
LCO_NMODE	2	tryb pracy funkcji
LCO_NSTST	1	typ warunku startu operacji
LCO_NCHAN	1	liczba kanałów
LCO_NPER ¹⁾	4	okres sterowania
LCO_NADDR	4	adres bufora (offset-segment)
LCO_NLEN	4	długość bufora
LCO_NSTART	5	warunki startu
LCO_NSTOP ¹⁾	5	warunki stopu

¹⁾ Dla modułu LC-030-1612 - ignorowane.

Przeznaczenie:

Funkcja służy do wysłania wartości na wyjście analogowe.

Znaczenie poszczególnych parametrów:

a. LCO_NMODE

Parametr określa tryb pracy funkcji przy czym znaczenie mają kolejne bity parametru:

nazwa	nr bitu	uwagi
LCO_MOD_START	1	
LCO_MOD_NEW_PAR	2	
LCO_MOD_SYNCHR	3	zarezerwowane; zawsze 0
LCO_MOD_INTR	4	zarezerwowane; zawsze 0
LCO_MOD_INTR_TYPE	5	zarezerwowane; zawsze 0
LCO_MOD_BLOCK	6	
LCO_MOD_CYCL	7	zarezerwowane; zawsze 0
LCO_MOD_FILE	8	zarezerwowane; zawsze 0
LCO_MOD_MEM_W	9	zarezerwowane; zawsze 0
LCO_MOD_EXT_CLK	10	zarezerwowane; zawsze 0
LCO_MOD_EXT_MEM	11	zarezerwowane; zawsze 0
-----	12..16	zarezerwowane; zawsze 0

LCO_MOD_START:

- start pomiarów

ustawienie tego bitu na 1 oznacza żądanie wykonania przetwarzania; wartość 0 powoduje jedynie analizę poprawności i zapamiętanie parametrów funkcji

LCO_MOD_NEW_PAR:

- ustawienie nowych parametrów

1 oznacza, że parametry przetwarzania pobierane będą z rekordu opisu zlecenia; 0 oznacza, że parametry przetwarzania będą identyczne jak poprzednio - jeżeli do tej pory nie było wykonania funkcji ANALOG_OUTPUT lub ostatnie było niepoprawne to driver zgłosi błąd LCO_NO_PARAMS

LCO_MOD_BLOCK:

- tryb przetwarzania: blokowy (1) / pojedynczy (0)

dla modułu LC-030-1612 jedynym dozwolonym trybem przetwarzania jest tryb pojedynczy, w którym każde wywołanie funkcji powoduje wysłanie pojedynczej wartości na wyjście analogowe; przy pierwszym wykonaniu funkcji w słowie trybu pracy (LCO_NMODE) powinien być ustawiony bit LCO_MOD_NEW_PAR (wówczas istotne są tylko parametry: LCO_NMODULE, LCO_NNUM, LCO_NMODE, LCO_NADDR, LCO_NLEN, LCO_NSTST i LCO_NSTART; nie jest analizowany okres próbkowania LCO_NPER ani warunki stopu przetwarzania (część LCO_NSTST oraz LCO_NSTOP) jako nie mające w tym kontekście sensu); przy następnych wykonaniach funkcji bit ten może być (choć nie jest to konieczne) zgaszony i wówczas istotne będą parametry: LCO_NMODULE, LCO_NNUM, LCO_NMODE i LCO_NADDR; wymagane jest zadeklarowanie adresu bufora z pamięci podstawowej; w słowie trybu pracy (LCO_NMODE) analizowane są tylko bity LCO_MOD_START i LCO_MOD_NEW_PAR (nie licząc opisywanego!), pozostałe - ignorowane uwaga: należy zwrócić uwagę, że jeżeli chcemy wykonać wysłanie bloku próbek, gdzie warunek startu odnosi się do całego bloku to drugi i kolejne wysłania należy wykonywać z warunkiem startu LCO_SIMMED

b. LCO_NSTST

Parametr określa typ warunku startu pomiaru (patrz p.3.4.). Warunkiem tym nie może być start od sygnału sprzętowego (LCO_SHARD).

c. LCO_NSTART.

Parametr ten określa szczegółowo warunek startu operacji. Interpretacja jego zależna jest od zadanego typu warunku startu operacji (LCO_NSTST). Szczegółowy opis - patrz p. 3.4.

d. LCO_NCHAN

Parametr określający tryb pracy modułu: jednokanałowo / wielokanałowo oraz (odpowiednio): liczbę kanałów / numer kanału:

b ₈	tryb pracy	b _{1..b₇}
0	praca wielokanałowa	liczba kanałów (2)
1	praca jednokanałowa	numer kanału (1..2)

Jeżeli w module jest zainstalowany tylko jeden przetwornik c/a to nie jest możliwa praca wielokanałowa.

e. LCO_AADDR, LCO_ALEN

Adres i długość bufora w pamięci podstawowej, z którego pobierana jest wartość do wysłania. Wartość ta powinna być podana w kodzie przetwornika a nie w woltach (czyli dla przetwornika 12 bitowego powinna być liczbą naturalną z zakresu 0..4095). Adres bufora jest adresem dalekim, tzn. podanym w postaci offset-segment, natomiast długość bufora podawana jest w próbkach i nie może być zerowa.

Błędy (LCO_STATUS):

LCO_BAD_BUF_ADR - błędny adres bufora (odnoszący się do nieistniejącej pamięci)
 LCO_BAD_BUF_LEN - błędna długość bufora (powodująca wyjście bufora poza pamięć).
 LCO_BAD_CHAN - błędny numer kanału - różny od 1 i 2
 LCO_BAD_CHAN_N - błędna liczba kanałów - różna od 2
 LCO_BAD_MODE - błędny tryb pracy funkcji
 LCO_BROKEN - operacja przerwana z powodu wykonania funkcji BREAK;
 LCO_ERR_STAT podaje, w jakim momencie operacja została przerwana
 LCO_ILL_START - błędne parametry warunku startu; sprecyzowanie błędu podane jest w LCO_ERR_STAT
 LCO_ILL_START_CODE - nielegalny typ warunku startu
 LCO_NONEX_DEV - nie istnieje przetwornik o tym numerze
 LCO_NO_MODULE - nie ma takiego modułu
 LCO_NO_PARAMS - zgaszono bit LCO_MOD_NEWPAR (LCO_NMODE) ale wcześniej nie ustawiono żadnych parametrów (nie było wykonania funkcji ANALOG_OUTPUT lub ostatnie wykonanie było niepoprawne)

Dodatkowe informacje o błędach (LCO_ERR_STAT):

LCO_E_BAD_CHAN - numer nieistniejącego kanału
 LCO_E_BAD_DATE - zła specyfikacja daty
 LCO_E_BAD_THRE - błędny próg wyzwalania analogowego
 LCO_E_BAD_TIME - zły odcinek czasu
 LCO_E_BROKEN_WAIT - funkcja przerwana w trakcie oczekiwania na spełnienie warunku startu
 LCO_E_NONEX_DEV - nie istnieje urządzenie o tym numerze
 LCO_E_NO_MODULE - nie ma takiego modułu

4.17. Zakończenie pracy z driver'em (LEAVE_DRIVER).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 16
LCO_STATUS	1	kod zakończenia funkcji = LCO_OK
LCO_ERR_STAT	1	dodatkowe informacje o błędach

Przeznaczenie:

Funkcja, którą należy wywołać przed zakończeniem programu. Powoduje "zapomnienie" przez driver wszystkiego co zostało mu przekazane w trakcie pracy programu. Zapobiega to błędnemu działaniu driver'a np. w sytuacji, gdy następny program żąda wykonania przetwarzania a/c wg. poprzednich parametrów (LCO_MOD_NEWPAR = 0) - a żadnych parametrów nie podał. Wykonanie funkcji w takiej sytuacji spowodowałoby np. przetransmitowanie danych na obszar ciała programu (adres bufora odziedziczony po poprzednim programie).

Funkcja zeruje moduł oraz wyinstalowuje procedurę obsługi przerwania od Ctrl-Break zainstalowaną przez funkcję BREAK (patrz) a także procedurę obsługi przerwania z modułu zainstalowaną przez funkcję ANALOG_INPUT (patrz).

4.18. Obsługa przerw (INTERRUPT_SERVICE).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LCO_CODE	1	kod funkcji = 17
LCO_STATUS	1	kod zakończenia funkcji = LCO_OK
LCO_ERR_STAT	1	dodatkowe informacje o błędach
LCO_SMODULE	1	numer modułu
LCO_SPROC	4	adres procedury użytkownika
LCO_SSTAT	4	adres słowa komunikacyjnego

Przeznaczenie:

Funkcja służy do instalowania i wyinstalowywania procedur obsługi przerwania przychodzącego z danego modułu. Działanie funkcji zależy od wartości parametru LCO_SPROC:

LCO_SPROC	działanie
0:0	wyinstalowanie procedury obsługi przerwania przychodzącego z modułu i zakazanie tego przerwania w kontrolerze przerw
inne	zainstalowanie procedury obsługi przerwania i zezwolenie na przerwanie w kontrolerze przerw; procedura ta jest procedurą wewnętrzną driver'a

Instalowana procedura jest procedurą wewnętrzną driver'a; jej zadaniem jest:

- odłożenie na stos rejestrów,
- wpisanie do rejestru DS właściwej wartości (jest to wartość, jaką miał ten rejestr w momencie wywoływania przez program użytkowy funkcji INTERRUPT_SERVICE),
- zbadanie przyczyny przerwania i wpisania jej pod adres LCO_SSTAT;
- wywołanie procedury o adresie LCO_SPROC,
- powiadomienie kontrolera przerw o obsłudze przerwania,
- przywrócenie wartości rejestrów

Parametr LCO_SPROC musi być tzw. dalekim adresem, tzn. podanym w postaci offset-segment. Procedura o podanym adresie musi być procedurą bez parametrów. Obowiązana jest wykonać funkcję driver'a WAIT_FOR_END w trybie LCO_W_FINISHED (aby powiadomić driver o zakończeniu przetwarzania). Procedura ta dostaje od driver'a stos o długości 128 bajtów. Jeżeli to nie wystarcza to powinna przełączać się na stos własny. Z tego też względu (jeżeli nie jest pisana w assemblerze) powinna być kompilowana z wyłączonym badaniem stosu).

Słowo komunikacyjne służy do przekazywania procedurze LCO_SPROC przyczyny przerwania, kodowanej na poszczególnych bitach:

nr bitu	nazwa	znaczenie
1	LCO_IS_START	przerwanie wystąpiło z powodu rozpoczęcia pomiaru
2	LCO_IS_ONE	nieużywane (przerwanie nastąpiło z powodu zmierzenia kolejnej serii pomiarowej)
3	LCO_IS_END_ADC	przerwanie nastąpiło z powodu zakończenia przetwarzania a/c
4	LCO_IS_END_DAC	nieużywane (przerwanie nastąpiło z powodu zakończenia przetwarzania c/a)
5	LCO_IS_BROKEN	miar został przerwany przez funkcję BREAK; ten bit ustawiany jest przez funkcję BREAK a nie przez procedurę obsługi przerwania!

Procedura obsługi przerwania wyinstalowywana jest również przez funkcję LEAVE_DRIVER (patrz).

Błędy:

LCO_BAD_PROC - błędny adres procedury obsługi przerwania lub słowa komunikacyjnego (spoza pamięci podstawowej)
 LCO_INTR_INST - procedura obsługi już jest zainstalowana
 LCO_NO_IRQ - z danym modulem nie jest związane żadne przerwanie
 LCO_NO_MODULE - nie ma takiego modułu

5. Zestawienie kodów zakończenia funkcji.

nazwa	kod	znaczenie
LCO_OK	0	poprawne zakończenie funkcji

Błędy (LCO_STATUS):

nazwa	kod	znaczenie
LCO_UNKN_FUNC	-1	nieznany kod funkcji
LCO_NO_MODULE	-2	nie istnieje żaden z żądanych modułów; nie ma takiego modułu
LCO_BAD_DEV_TYP	-3	brak urządzeń danego typu
LCO_NONEX_DEV	-4	nie istnieje urządzenie o tym numerze
LCO_BAD_FREQ	-5	błędna częstotliwość
LCO_BAD_RANGE	-6	błędny zakres napięć
LCO_NO_OPER	-7	z wyspecyfikowanym urządzeniem nie jest związana żadna operacja w toku
LCO_BAD_MARGIN*)	-8	błędna długość marginesu początkowego (nie będąca wielokrotnością liczby kanałów
LCO_BAD_BUF_ADR	-9	błędny adres bufora (odnoszący się do nieistniejącej pamięci)

LCO_BAD_BUF_LEN	-10	błędna długość bufora (powodująca wyjście bufora poza pamięć, przejście pomiędzy pamięcią podstawową a rozszerzoną, przekroczenie rozmiarów bufora w pamięci dodatkowej itp.)
LCO_DEV_BUSY	-11	urządzenie zajęte
LCO_BAD_PER	-12	za długi lub za krótki okres
LCO_BAD_CHAN_N	-13	zła liczba kanałów
LCO_BAD_CHAN	-14	numer nieistniejącego kanału
LCO_BROKEN	-15	przetwarzanie przerwane z powodu wykonania funkcji BREAK
LCO_INTR_NOT_INST	-16	procedura obsługi przerwania nie jest zainstalowana
LCO_ILL_START_CODE	-17	nielegalny typ warunku startu
LCO_ILL_STOP_CODE	-18	nielegalny typ warunku stopu
LCO_BAD_PROC	-19	błędny adres procedury obsługi przerwania (spoza pamięci podstawowej)
LCO_TOO_LONG_MARG	-20	za długi margines
LCO_ILL_START	-21	błędne parametry warunku startu
LCO_ILL_STOP	-22	błędne parametry warunku stopu
LCO_BAD_MNUM	-23	błędny numer pierwszej próbki
LCO_NOT_SUPPORTED	-24	dla danego typu modułu funkcja nie jest realizowana
LCO_BAD_CTC_MODE ¹⁾	-25	błędny tryb pracy CTC
LCO_NO_PARAMS	-26	nie podano parametrów przetwarzania a/c, c/a
LCO_OVERRUN ¹⁾	-27	zakończono przetwarzanie a/c z powodu błędu OVERRUN
LCO_NO_DMA	-28	z danym urządzeniem nie jest związany żaden kanał DMA
LCO_NO_IRQ	-29	z danym modulem nie jest związane żadne przerwanie
LCO_NOT_FULLY_SUP	-30	żądany tryb wykonania funkcji nie jest dla danego typu modułu realizowany lub jest w opracowaniu
LCO_NO_EXTMEM ¹⁾	-31	brak pamięci dodatkowej
LCO_NO_SEC_FREQ ¹⁾	-32	moduł nie może prowadzić przetwarzania z dwiema częstotliwościami
LCO_INTR_INST	-33	procedura obsługi przerwania jest już zainstalowana
LCO_BAD_PER2	-34	błędna wielokrotność okresu próbkowania (0 lub 1)
LCO_BAD_MODE	-35	błędny tryb pracy
LCO_BAD_EXTMEM ¹⁾	-36	błędny adres bufora w pamięci rozszerzonej
LCO_CTC_NOT_PROGRAMMED ¹⁾	-37	zlecono zapis wartości licznika lecz nie zaprogramowano trybu pracy kanału
LCO_REJECTED	-38	za dużo równoczesnych odwołań do driver'a

¹⁾ Nie występuje dla modułu LC-030-1612.

Dodatkowe informacje o błędach (LCO_ERR_STAT):

nazwa	kod	znaczenie
LCO_E_OK	0	brak dodatkowych informacji

; błędy w warunkach startu / stopu:

LCO_E_NO_MODULE	-1	nie ma takiego modułu
LCO_E_NONEX_DEV	-2	nie istnieje urządzenie o tym numerze
LCO_E_BAD_CHAN	-3	numer nieistniejącego kanału
LCO_E_BAD_TIME	-4	zły odcinek czasu
LCO_E_BAD_DATE	-5	zła specyfikacja daty
LCO_E_BAD_THRE	-6	błędny próg wyzwalania analogowego
LCO_E_BAD_LEN	-9	zła długość pomiaru (więcej niż pamięć modułu)

; moment przerwania przez funkcję BREAK:

LCO_E_BROKEN_WAIT	-7	funkcja przerwana w trakcie oczekiwania na spełnienie warunku startu
LCO_E_BROKEN_RUN	-8	funkcja przerwana w trakcie przetwarzania

Ostrzeżenia (LCO_STATUS):

nazwa	kod	znaczenie
LCO_NON_EX_MOD	1	zaządzano inicjalizacji nie istniejących modułów ale co najmniej 1 moduł został zainicjalizowany
LCO_OTHER_LEN	2	przepisano mniej próbek niż żądano
LCO_PREMATURE_END ¹⁾	3	przedczesne zakończenie operacji z powodu przepełnienia / opróżnienia całego bufora
LCO_IN_PROGRESS	4	badana transmisja jeszcze trwa

¹⁾ Nie występuje dla modułów serii LC-030.

6. Projektowanie programów użytkowych.

W poniższym rozdziale zostanie omówiony sposób komunikacji programów użytkowych z driver'em. Na wstępie opisane zostaną ogólne zasady komunikacji a w dalszych podrozdziałach - komunikacja z driver'em z poziomu programów napisanych w C, Pascalu i assemblerze.

Wykonanie dowolnej z funkcji driver' wymaga następujących czynności:

- wypełnienie odpowiedniego rekordu opisu zlecenia; odpowiednie struktury danych dostarczane są przez producenta w postaci zbiorów źródłowych
- wpisanie adresu rekordu do rejestrów DX i DI
- wykonanie odpowiedniego przerwania programowego; numer przerwania zależy od konkretnego driver'a; w przypadku driver'a LC3016A.DRV dla modułu LC-030-1612 jest to numer 96₁₆ (150₁₀).

Dobrze napisany program powinien składać się z następujących części:

- część wstępna:
 - stwierdzenie, czy driver jest zainstalowany; Metodą wykrycia obecności driver'a w pamięci systemu jest próba otwarcia urządzenia o nazwie określonej przez driver (LC3016^^); powodzenie tej próby świadczy o zainstalowaniu driver'a, niepowodzenie - o jego braku
 - rozpoznanie konfiguracji modułu (GET_TOTAL_CONFIGURATION - czy moduł jest zainstalowany jako A czy B, GET_MODULE_CONFIGURATION - czy moduł ma zainstalowany przetwornik c/a, czy jest podłączony do któregoś z przerwań, GET_INFO - czy tor pomiarowy a/c jest podłączony do kanału DMA, jakie są minimalne okresy próbkowania itp.); ten etap jest szczególnie ważny, gdy projektowany jest program uniwersalny,

mający operować na kilku rodzajach modułów

- inicjalizacja modułu; jest konieczna zwłaszcza w sytuacji, gdy przewidywane jest używanie czasowych warunków startu operacji - inicjalizacja powoduje (między innymi) synchronizację programowego zegara czasu rzeczywistego zawartego w driverze z zegarem komputera (który musi być prawidłowo ustawiony!)
- instalacja procedury obsługi przerwania generowanego przez klawisz Ctrl-Break (jeżeli przewiduje się jego użycie)
- ustawienie zakresu pracy przetwornika a/c (jeżeli ma być stały w czasie pracy programu)
- część wykonawcza: tu powinny się znaleźć funkcje wykonujące właściwe operacje modułu jak `ANALOG_INPUT`, `DATA_TRANSMIT`, `ANALOG_OUTPUT`, `DIGITAL_INPUT`, `DIGITAL_OUTPUT` itp.; należy zwrócić uwagę na dwie rzeczy:
 - każda funkcja może dostać błędne parametry i zasygnalizować to w kodzie odpowiedzi (`LCO_STATUS`, `LCO_ERR_STAT`); należy koniecznie sprawdzać tę odpowiedź, szczególnie w dwóch sytuacjach: gdy program jest na etapie uruchamiania i gdy parametry funkcji są dostarczane interakcyjnie przez użytkownika
 - moduł może ulec uszkodzeniu - wówczas niektóre funkcje nie mogą się zakończyć (przetwarzanie a/c, oczekiwanie na spełnienie warunków startu związanych z wejściami cyfrowymi czy sygnałem analogowym); poza tym funkcja może dostać parametry błędne mimo że formalnie poprawne (np. gdy operator poda dziesięciokrotnie mniejszy okres próbkowania albo - jako warunek startu - dzień następnny zamiast bieżącego); należy - przewidując taką sytuację - albo umożliwić operatorowi przerwanie takiej funkcji przez naciśnięcie klawisza Ctrl-Break (i wykonanie w procedurze obsługi przerwania funkcji `BREAK`) albo samodzielnie odmierzać czas wykonania operacji i po przekroczeniu oszacowanego wcześniej limitu - automatycznie wykonywać funkcję `BREAK`; jest to istotne o tyle, że w przeciwnym razie operator zmuszony będzie powtórnie załadować system co może się wiązać ze stratą zmierzonych uprzednio - i być może unikatowych - danych
- część końcowa:
 - przed zakończeniem programu należy koniecznie wykonać funkcję `LEAVE_DRIVER`; jest to szczególnie ważne wtedy, gdy na komputerze wykonywanych jest kilka programów korzystających z tego samego driver'a.

6.1. Programowanie w języku C.

Z poziomu języka C komunikacja z driver'em odbywa się za pośrednictwem procedury bibliotecznej `int86x`. W przypadku implementacji firmy Borland (Turbo C) można korzystać również z pseudozmiennych `_DX` i `_DI` oraz makroinstrukcji `geninterrupt` (`dos.h`).

Producent dostarcza dwa pliki źródłowe: plik `AMBEX-LC.H` zawierający definicje wszystkich struktur danych i stałych potrzebnych do współpracy z driver'em oraz plik `TEST.C` zawierający przykłady wykorzystania poszczególnych funkcji driver'a.

W Dodatku A znajduje się wydruk pliku `AMBEX-LC.H`, natomiast w Dodatku B - wydruk pliku `TEST.C`.

6.2. Programowanie w języku Pascal.

Z poziomu języka Pascal komunikacja z driver'em odbywa się za pośrednictwem procedury bibliotecznej `intr`.

Producent dostarcza dwa pliki źródłowe: plik `AMBEX-LC.PAS` zawierający definicje wszystkich struktur danych i stałych potrzebnych do współpracy z driver'em oraz plik `TEST.PAS` zawierający przykłady wykorzystania poszczególnych funkcji driver'a.

W Dodatku C znajduje się wydruk pliku `AMBEX-LC.PAS`, natomiast w Dodatku D - wydruk pliku `TEST.PAS`.

6.3. Programowanie w języku assemblera.

Producent dostarcza plik źródłowy `AMBEX-LC.ASM` zawierający definicje wszystkich /struktur danych i stałych potrzebnych do współpracy z driver'em. Wydruk tego pliku znajduje się w Dodatku E.