



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-011-0812

WERSJA 1.0

Wydanie: lipiec 1992

AMBEX Spółka z o.o.					Warszawa
DZIAŁ HANDLOWY	ul. Topiel 6	8:30-16:30	tel.	(0-2)	635-87-24
SERWIS			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
komertel	(48) 39-12-07-63		telex	81-52-62 ambex pl	

- zarejestrowany znak handlowy i graficzny Z.E. AMBEX Warszawa

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 przerw (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 przerw (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 modulem 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 wyższe). 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 ładowania systemu. Po zainstalowaniu driver służy tylko jako obsługa danego przerwania programowego.

Driver modułów serii LC-011-0812 podporządkowany został standardowi przyjętemu przez firmę. W standardzie tym przyjęto, że w komputerze może być zainstalowanych kilka (do trzech) modułów danego rodzaju obsługiwanych przez jeden driver. Ponieważ standard ten został zaprojektowany do różnych typów modułów, niektóre parametry wydawać się mogą nadmiarowe.

2. Instalacja driver'a.

Instalacji driver'a dokonuje się za pomocą programu instalacyjnego - INSTALL. Przed instalacją należy skopiować:

- zbiór LC*.DRV (odpowiedni dla rodzaju komputera oraz modułu),
 - program instalacyjny INSTALL.EXE,
- 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 (jednorazowy 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-011-0812, LC-011-1612, LC-020-0812, LC-030-1612;
- b) liczba zainstalowanych modułów danego typu: od 1 do 4 *);
- c) typ komputera: XT, AT¹⁾;
- 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 (typowym adresem jest 100000 - początek pamięci rozszerzonej);

- 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) 9) .

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) 6) ;
- 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) 9) .

1) dla modułu LC-030-1612 - tylko AT.

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 multiplekser i wynosi 16 bądź 1,

- moduł LC-011-0812: liczba kanałów jest stała i wynosi 8,

- 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.

8) dla modułu LC-011-0812 maksymalna liczba modułów wynosi 3.

9) nie dotyczy pracy z modułem LC-030-1612.

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) co kończy instalację. W przypadku nieumieszczenia modułu w komputerze, bądź jego wadliwej pracy, instalacja nie jest wykonana a użytkownik jest o tym informowany stosownym komunikatem.

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 obsługiwane moduły
- rozszerzenie w sposób programowy możliwości modułów o funkcje, które nie są lub nie mogą być realizowane sprzętowo; do funkcji takich należą:
- różnorodne warunki startu operacji wejścia / wyjścia (odczyt / zapis portów cyfrowych, przetwarzanie a/c i c/a); w grę wchodzi tu warunkowanie startu operacji sygnałami cyfrowymi (poziomem, zboczem, kombinacją sygnałów), oraz wpływem czasu rzeczywistego (data i odcinek czasu);
- długie transmisje DMA - przekraczające 64kB; jak wiadomo, w komputerach klasy IBM PC układy DMA są 16-bitowe i nie są połączone bezpośrednio z tzw. rejestrem strony rozszerzającym adres transmisji do 20 (XT) lub 24 (AT) bitów; powoduje to, że jeżeli w trakcie transmisji adres transmisji ma przejść przez granicę 64kB to wymaga to śledzenia transmisji na bieżąco i odpowiedniego

przeprogramowywania układów DMA; porcja danych, którą można w danej sytuacji przesłać za pomocą pojedynczej transmisji będzie dalej nazywana blokiem transmisji DMA;

- 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 wszystkich zadeklarowanych modułów, 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 każdy z zainstalowanych modułów.

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.

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 LC-011-0812:

typ modułu	nazwa przerwania	numer przerwania	
		szesnastkowy	dziesiętny
LC-011-0812	LC011_08	90	144

Wszystkie driver'y widziane są w systemie DOS jako urządzenia. Nazwy tych urządzeń są następujące:

typ modułu	nazwa driver'a	nazwa urządzenia
LC-011-0812	LC1108?.DRV	LC1108^^

gdzie '?' oznacza typ komputera: x - 'XT',
a - 'AT' i '386'.

Kodowanie numerów modułów:

moduł	nazwa kodu	wartość
A	LCO_MODA	1
B	LCO_MODB	2
C	LCO_MODC	3

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 itp. podawane są zawsze w próbkach.

3.3. Komunikacja z driver'em.

Funkcje driver'a wywoływane są poprzez przerwanie programowe (numery 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 niezmiennicze 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	dodatkowe 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 tryby startu operacji:

- start natychmiastowy: bez czekania na spełnienie jakichkolwiek warunków
- 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;
- 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 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 zakończenia operacji zdefiniowano następujące warunki:

- przetworzenie określonej liczby próbek;

Kodowanie typu 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)	sekunda	minuta	godzina	dzień	---
o podanym czasie				miesiąca	

- 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.

Kodowanie typu warunku stopu (zatrzymania) operacji (kody podane szesnastkowo):

kod typu (szesnastkowo) / znaczenie	parametry warunku				
	bajt 1	bajt 2	bajt 3	bajt 4	bajt 5
00 (LCO_ZSAMPLES)	liczba próbek do zmierzenia				---
po zmierzeniu bloku danych					

4. Funkcje driver'a.

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 przechwycenie przerwania 1C₁σ (przerwanie usługowe, wywoływane przez procedurę obsługi przerwania zegarowego) aby w bezpieczny sposób odmierzać upływający czas astronomiczny. Po przechwyceniu przerwania następuje synchronizacja wewnętrznego zegara czasu rzeczywistego z zegarem komputera. Synchronizacja polega na przepisaniu do niego czasu i daty systemowej, pobranej za pomocą funkcji DOS'a (2C₁σ i 2A₁σ). Z tego też powodu funkcji MODULE_INIT nie należy wykonywać w procedurach obsługi przerwania - może się to skończyć zawieszeniem pracy systemu operacyjnego.

UWAGA: Inne programy wykorzystujące przerwanie 1C₁σ mają obowiązek - po przechwyceniu tego przerwania - wykonywać również poprzednią procedurę obsługi tego przerwania.

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 - zażądano 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_TONF	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	4	adres bufora w pamięci rozszerzonej (adres absolutny)
LCO_TMEML	4	długość bufora w pamięci rozszerzonej (w próbkach)

Przeznaczenie:

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

```

b8          b1
D C B A D C B A
y y y y x x x x

```

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

y: y = 1 - moduł "master", y = 0 - moduł "slave" (dla modułów serii LC-011-0812 wszystkie moduły są typu "master")

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 1
LCO_MBASE2 ¹⁾	2	adres bazowy modułu - pakiet 2
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; częstotliwość próbkowania tworzona jest przez podział 1/16 częstotliwości zegara modułu
LCO_MINT	1	numer przerwania (programowy)

¹⁾ dla modułów serii LC-011-0812 - nieokreślone (moduły są jednopaketowe)

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 wartości minimalnych okresów próbkowania w 1, 2, 3, ..., 32 kanałach. Wypełnionych jest pierwszych n pozycji, gdzie n jest maksymalną liczbą kanałów modułu. Wartości okresów dotyczą transmisji pojedynczego bloku DMA. Okresy podane są w dziesiątych częściach mikrosekundy.

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	dodatkowe 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⁾	1	numer kanału DMA

¹⁾ dla modułów serii LC-011-0812 - nieokreślone (przy przetwarzaniu C/A moduł nie współpracuje z kanałem DMA)

Przeznaczenie:

Funkcja zwraca informację o konfiguracji badanego toru cyfrowo-analogowego.

Błędy (LCO_STATUS):

LCO_BAD_DEV_TYP - brak przetworników c/a
 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	dodatkowe 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 CTC o tym numerze nie istnieje
 LCO_NO_MODULE - nie ma takiego modułu

4.5. Zadeklarowanie częstotliwości zegara magistrali (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 dla opisanego modułu.

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 lub 4)
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 nie jest realizowana dla opisanego 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 dla opisanego modułu.

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	dodatkowe informacje o błędach
LCO_WTYPE	1	rodzaj urządzenia (3 lub 4)
LCO_WMODULE	1	numer modułu
LCO_WNUM	1	numer przetwornika
LCO_WMODE	1	tryb pracy
parametry wyjściowe:		
LCO_WRMNUM	4	rzeczywista liczba zmierzonych / wysłanych próbek
LCO_WREMAR	2	rzeczywista długość marginesu końcowego

Przeznaczenie:

Funkcja nie jest realizowana dla opisanego modułu.

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
LCO_BPROC	4	adres procedury obsługi Ctrl-Break

Przeznaczenie:

Przerywanie pracy driver'a i modułu, instalowanie i wyinstalowywanie procedur obsługi przerwania 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 przerwania Ctrl-Break
LCO_BREAK_UNINST	2	wyinstaluj procedurę obsługi przerwania Ctrl-Break

Przerwanie pracy - tryb LCO_BREAK_EXEC:

Działanie funkcji polega na przerwaniu wszystkich operacji wykonywanych we wszystkich modułach. W momencie wywołania tej funkcji driver i moduł mogą znajdować się w jednym z dwóch stanów:

1. Nie jest wykonywana żadna operacja - driver zwraca błąd LCO_STATUS = LCO_NO_OPER, LCO_ERR_STAT = LCO_E_OK.
2. Moduł wykonuje pod kontrolą driver'a operację synchroniczną. Ponieważ w tym przypadku należy przerwać pracę driver'a, zazwyczaj funkcja wywoływana jest w procedurze obsługi przerwania, np. Ctrl-Break. Driver zwraca LCO_STATUS = LCO_OK i LCO_ERR_STAT = LCO_E_OK.

Zainstalowanie procedury obsługi - tryb LCO_BREAK_INST:

Funkcja instaluje obsługę Ctrl-Break przez przechwycenie przerwania 1B16 i podłożenie nowej procedury obsługi tego przerwania. 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 modułem 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 synchronicznej
- 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.

Błędy (LCO_STATUS):

LCO_ILL_START - błędne parametry trybu startu operacji
 LCO_ILL_START_CODE - błędny tryb 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_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 warunków startu

Przeznaczenie:

Wysłanie wartości na wyjścia cyfrowe podanego portu.

Błędy (LCO_STATUS):

LCO_ILL_START - błędne parametry trybu startu operacji
 LCO_ILL_START_CODE - błędny tryb 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_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⁾	1	wartość bajtu sterującego CTC
LCO_CVAL	2	wpisywana wartość licznika

¹⁾ Wartości bajtu sterującego układu 8253 opisane są w dokumentacji techniczno-ruchowej modułu.

Przeznaczenie:

Przeznaczeniem funkcji jest zapisanie trybu pracy licznika układu 8253/8254 i/lub jego nowej wartości a także uruchomienie licznika. Zależy to od ustawionych bitów trybu pracy LCO_CMODE:

nr bitu	nazwa	znaczenie
1	LCO_SET_CTC_MODE	zaprogramuj tryb pracy kanału
2	LCO_SET_COUNTER_VALUE	załaduj nową wartość licznika
3	LCO_CTC_ENABLE	zezwól na pracę licznika

Pozostałe bity są ignorowane.

Błędy:

LCO_NO_MODULE - dany moduł nie istnieje
 LCO_BAD_CTC_MODE - błędny parametr LCO_CFUN (część dotycząca trybu pracy kanału CTC)
 LCO_NONEX_DEV - w parametrze LCO_CFUN podano numer nieistniejącego kanału CTC
 LCO_CTC_NOT_PROGRAMMED - zlecono zapis wartości licznika lecz nie zaprogramowano trybu pracy kanału

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:

Funkcja odczytuje bieżącą wartość zadanego kanału CTC.

Błędy:

LCO_NO_MODULE	- dany moduł nie istnieje
LCO_NONEX_DEV	- podano numer nieistniejącego kanału CTC
LCO_CTC_NOT_PROGRAMMED	- próba odczytu wartości licznika przy niezaprogramowanym trybie pracy kanału

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	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	4	adres absolutny bufora w pamięci rozszerzonej
parametry wyjściowe:		
LCO_RRNUM	4	rzeczywista liczba przesłanych próbek

Przeznaczenie:

Funkcja przepisuje ciąg próbek między buforem w pamięci podstawowej a buforem w pamięci rozszerzonej.

Znaczenie poszczególnych parametrów:

a. LCO_RMODE

Tryb pracy funkcji:

nr bitu	wart	nazwa	znaczenie
1	1	LCO_TO_EXT_DIR	do pamięci rozszerzonej
	0	LCO_FROM_EXT_DIR	z pamięci rozszerzonej
2..8	---	-----	zarezerwowane; zawsze 0

b. LCO_RADDR

Adres bufora w podstawowej pamięci komputera.

c. LCO_RLEN

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

d. LCO_RMEAS

Numer pierwszej próbki do przesłania. Przesłane zostaną próbki o numerach LCO_RMEAS, LCO_RMEAS + 1 itd. Należy pamiętać, że pierwsza próbka ma numer 1. Jeżeli mierzonych było np. 5 kanałów to 10 próbka w pierwszym kanale ma numer 51.

e. LCO_RNUM

Całkowita liczba próbek do przesłania.

f. LCO_RMEMA

Parametr określa absolutny, 24-bitowy adres bufora w pamięci rozszerzonej. Adres ten nie może wychodzić poza obszar bufora zadeklarowanego przy instalacji driver'a.

g. LCO_RRNUM

Rzeczywista liczba przepisanych próbek. Parametr LCO_RRNUM przybiera wartość będącą minimum z LCO_RLEN, <n> i LCO_RNUM. <n> jest tu liczbą próbek zawartych między próbka LCO_RMEAS a końcem bufora w pamięci rozszerzonej.

Ostrzeżenia (LCO_STATUS):

LCO_OTHER_LEN - przepisano mniejszą liczbę próbek niż żądano

Błędy (LCO_STATUS):

LCO_BAD_BUF_ADR - błędny adres bufora w pamięci podstawowej (odnoszący się do nieistniejącej pamięci)
 LCO_BAD_BUF_LEN - błędna długość bufora w pamięci podstawowej (powodująca wyjście bufora poza pamięć itp.)
 LCO_BAD_EXTMEM - błędny adres bufora w pamięci rozszerzonej
 LCO_BAD_MNUM - błędny numer pierwszej próbki (np. powodujący wyjście poza jeden z buforów)
 LCO_BAD_MODE - błędny tryb pracy
 LCO_BROKEN - transmisja przerwana z powodu wykonania funkcji BREAK
 LCO_NO_EXTMEM - brak pamięci rozszerzonej

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_APERZ ¹⁾	2	wielokrotność okresu próbkowania dla kanałów dodatkowych
LCO_ACHAN	1	liczba kanałów / numer kanału
LCO_ACHANZ ¹⁾	1	liczba kanałów dodatkowych
LCO_AADDR	4	adres bufora (offset-segment) w pamięci podstawowej
LCO_ALEN	4	długość bufora w pamięci podstawowej
LCO_AMEMA	4	adres absolutny 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⁾ Wielkość bez znaczenia dla modułów serii LC-011 i LC-020.

²⁾ Wielkość niekreślona dla modułów serii LC-011 i LC-020.

Przeznaczenie:

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

Generalnie pomiar (w trybie blokowym, za pośrednictwem transmisji DMA) można podzielić na następujące fazy:

1. Przygotowanie toru pomiarowego do pracy.
2. Oczekiwanie na spełnienie warunku startu przetwarzania.
3. Start właściwego pomiaru.
4. Oczekiwanie na spełnienie warunku stopu przetwarzania.
6. Zakończenie pomiaru.

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	ignorowane
! LCO_MOD_INTR	4	ignorowane
! LCO_MOD_INTR_TYPE	5	ignorowane
! LCO_MOD_BLOCK	6	
! LCO_MOD_CYCL	7	ignorowane
! LCO_MOD_FILE	8	ignorowane
! LCO_MOD_MEM_W	9	
! LCO_MOD_EXT_CLK	10	ignorowane
! LCO_MOD_EXT_MEM	11	
! -----	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_BLOCK:

- tryb przetwarzania: blokowy (1) / pojedynczy (0)
- tryb blokowy: tryb podstawowy pracy modułu, w którym transmisja bloku próbek do pamięci komputera prowadzona jest za pośrednictwem kanału DMA, możliwy pomiar do bufora w pamięci rozszerzonej;
- 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; nie jest analizowany okres próbkowania LCO_APER; UWAGA: należy zwrócić uwagę, że jeżeli chcemy wykonać pomiar bloku próbek, gdzie warunek startu odnosi się do całego bloku to drugi i kolejne pomiary należy wykonywać z warunkiem startu LCO_SIMMED

LCO_MOD_MEM_W:

- przepisanie do pamięci (1) / bez przepisywania do pamięci (0) bit steruje działaniem funkcji po zakończeniu przetwarzania w trybie synchronicznym przy współpracy z pamięcią rozszerzoną; jeżeli bit jest równy 1 to po zakończeniu przetwarzania do pamięci podstawowej (pod adres LCO_AADDR) przepisywanych jest <n> pierwszych próbek gdzie <n> jest minimum z całkowitej liczby zmierzonych próbek (LCO_ARMNUM) i rozmiaru bufora (LCO_ALEN); całkowita liczba przepisanych próbek zwracana jest w parametrze LCO_ARLEN; jeżeli natomiast bit LCO_MOD_MEM_W 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 przepisanie danych do pamięci podstawowej jest możliwe poprzez wykonanie funkcji DATA_TRANSMIT; przy pracy z pamięcią podstawową (LCO_MOD_EXT_MEM = 0) bit LCO_MOD_MEM_W jest ignorowany.

LCO_MOD_EXT_MEM:

- praca z pamięcią rozszerzoną (1) / pamięcią podstawową (0): przy pracy z pamięcią rozszerzoną zmierzone dane przesyłane są do bufora w pamięci rozszerzonej pod adres LCO_AMEMA; bufor ten musi się zawierać w buforze zadeklarowanym przy instalacji driver'a; zadeklarowany bufor można wykorzystywać jako kilka mniejszych ale należy pamiętać o tym, że driver nie kontroluje zachodzenia na siebie tak utworzonych buforów; dla pracy z pamięcią rozszerzoną adres (LCO_AADDR) i długość (LCO_ALEN) bufora w pamięci podstawowej są analizowane tylko wtedy gdy jednocześnie zapalony zostanie bit LCO_MOD_MEM_W (patrz wyżej); dla pracy z pamięcią podstawową adres bufora w pamięci rozszerzonej (LCO_AMEMA) nie jest analizowany;

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.).

c. LCO_ASTART, 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 (bit parametru LCO_AMODE LCO_MOD_BLOCK = 1). Okres ten podawany jest w dziesiątych częściach mikrosekundy (np. 10000 oznacza 1 ms). Okres ten nie może być mniejszy niż minimalny okres dla danej liczby kanałów. (O minimalnych okresach próbkowania można - należy - się dowiedzieć z programu za pomocą funkcji GET_INFO (parametr LCO_GMINP, patrz opis funkcji).) 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 (najstarszy bit parametru LCO_ACHAN równy 0, patrz niżej opis tego parametru) - wartość LCO_ACHAN
- Z parametrem tym związane są parametry zwrotne driver'a LCO_ARDIV1 i LCO_ARDIV2 (patrz opis poniżej).

e. 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, 4 lub 8)	
1	praca jednokanałowa	numer kanału (1..8)	

f. LCO_AADDR, LCO_ALEN

Adres i długość bufora w pamięci podstawowej. Parametry te są analizowane tylko wtedy, gdy bit LCO_MOD_EXT_MEM trybu pracy (LCO_AMODE) został ustawiony na 0 (praca z pamięcią podstawową) lub - w przeciwnym razie - gdy bit LCO_MOD_MEM_W = 1 i LCO_MOD_SYNCHR = 1 (praca synchroniczna z pamięcią rozszerzoną z przepisaniem do pamięci podstawowej). Adres bufora jest adresem dalekim, tzn. podanym w postaci offset-segment, natomiast długość bufora podawana jest w próbkach (i jest to liczba długa tj. 32 bitowa). Długość bufora jest jednym z czynników określających liczbę próbek przepisywanych do pamięci podstawowej (patrz opis bitu LCO_MOD_MEM_W).

h. LCO_AMEMA

Parametr istotny tylko dla pracy z pamięcią rozszerzoną (LCO_MOD_EXT_MEM = 1). Oznacza absolutny, 24-bitowy adres bufora w pamięci rozszerzonej. Adres ten nie może wychodzić poza obszar bufora zadeklarowanego przy instalacji driver'a.

i. LCO_ARDIV1, LCO_ARDIV2

Częstotliwość próbkowania tworzona jest przez podział (przez całkowitą wartość) 1/16 częstotliwości generatora w module. Parametr LCO_ARDIV1 przekazuje wartość dzielnika. Łącznie z częstotliwością zegara modułu (patrz funkcja GET_MODULE_INFORMATION) daje informację o faktycznej częstotliwości próbkowania.

j. LCO_ARMNUM

Rzeczywista liczba zmierzonych próbek.

Błędy (LCO_STATUS):

LCO_BAD_BUF_ADR	- błędny adres bufora (odnoszący się do nieistniejącej pamięci lub do pamięci rozszerzonej w przypadku transmisji programowej)
LCO_BAD_BUF_LEN	- błędna długość bufora (powodująca wyjście bufora poza pamięć, przejście pomiędzy pamięcią podstawową a rozszerzoną, przekroczenie rozmiarów bufora w pamięci rozszerzonej itp.)
LCO_BAD_CHAN	- numer nieistniejącego kanału
LCO_BAD_CHAN_N	- zła liczba kanałów
LCO_BAD_EXTMEM	- błędny adres bufora w pamięci rozszerzonej (dla pracy z pamięcią rozszerzoną, LCO_MOD_EXT_MEM = 1)
LCO_BAD_MODE	- błędny tryb pracy
LCO_BROKEN	- operacja przerwana z powodu wykonania funkcji BREAK; LCO_ERR_STAT podaje, w jakim momencie operacja została przerwana
LCO_BAD_PER	- za krótki okres próbkowania
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_NONEX_DEV	- nie istnieje przetwornik o tym numerze
LCO_NO_DMA	- z danym przetwornikiem nie jest związany żaden kanał DMA (dla pracy blokowej)
LCO_NO_EXTMEM	- brak pamięci rozszerzonej (dla pracy z pamięcią rozszerzoną)
LCO_NO_IRQ	- z danym modulem nie jest związane żadne przerwanie (dla pracy z przerwaniami)
LCO_NO_MODULE	- nie ma takiego modułu
LCO_NO_PARAMS	- zgaszono bit LCO_MOD_NEW_PAR (LCO_AMODE) ale wcześniej nie ustawiono żadnych parametrów (nie było wykonania funkcji ANALOG_INPUT lub ostatnie wykonanie było niepoprawne)
LCO_OVERRUN	- zakończono przetwarzanie z powodu błędu OVERRUN

Dodatkowe informacje o błędach (LCO_ERR_STAT):

LCO_E_BAD_CHAN	- numer nieistniejącego kanału
LCO_E_BAD_DATE	- zła specyfikacja daty
LCO_E_BAD_TIME	- zły odcinek czasu
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 ⁴⁾	2	tryb pracy funkcji
LCO_NSTST	1	typy warunków startu / stopu operacji
LCO_NCHAN	1	liczba kanałów / numer kanału
LCO_NPER ⁴⁾	4	okres sterowania
LCO_NADDR	4	adres bufora
LCO_NLEN	4	długość bufora
LCO_NHAND ⁴⁾	2	numer handlera pliku dyskowego
LCO_NSTART	5	warunki startu
LCO_NSTOP ⁴⁾	5	warunki stopu

⁴⁾ Parametry ignorowane dla modułów serii LC-011-0812.

Przeznaczenie:

Funkcja służy wysyłania do danych na wyjście analogowe.

a. LCO_NSTST

Parametr określa typy warunków startu i stopu (warunek stopu ignorowany). Jest on sumą odpowiednich kodów typów warunku startu i stopu (patrz p.3.4.). Warunek startu nie może być typu LCO_SHARD.

b. LCO_NSTART

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

c. LCO_NCHAN

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)
1	praca jednokanałowa	numer kanału (1..n) gdzie n - liczba kanałów c/a w module

d. LCO_NADDR, LCO_NLEN

Parametr LCO_NADDR oznacza adres bufora w pamięci podstawowej w którym umieszczona jest wartość do wysłania na wyjście analogowe. Adres podajemy w postaci offset-segment. Parametr LCO_NLEN oznacza długość bufora w pamięci podstawowej i musi być równy 2 lub 4.

Błędy (LCO_STATUS):

LCO_BAD_BUF_ADR	- błędny adres bufora
LCO_BAD_BUF_LEN	- błędna długość bufora
LCO_BAD_CHAN	- numer nieistniejącego kanału
LCO_BROKEN	- transmisja przerwana z powodu wykonania funkcji BREAK; LCO_ERR_STAT podaje, w jakim momencie operacja została przerwana
LCO_ILL_START	- błędne parametry sposobu startu
LCO_ILL_START_CODE	- nielegalny sposób 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_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_NEW_PAR = 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 wyinstalowuje również obsługę przerwania zegarowego 1C₁₆ przez driver.

Funkcja zeruje wszystkie zainstalowane moduły oraz wyinstalowuje procedurę obsługi przerwania od Ctrl-Break zainstalowaną przez funkcję BREAK (patrz).

4.18. Obsługa przerwania (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 nie jest realizowana dla opisanego 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 lub do pamięci rozszerzonejw przypadku transmisji programowej)
LCO_BAD_BUF_LEN	-10	błędna długość bufora (powodująca wyjście bufora poza pamięć, przejście pomiędzy pamięcią podstawową a rozszerzoną, przekroczenie rozmiarów bufora w pamięci rozszerzonej 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 lub słowa komunikacyjnego (spoza pamięci podstawowej)
LCO_TOO_LONG_MARG	-20	suma marginesów dłuższa od bufora
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 modułu funkcja nie jest realizowana
LCO_BAD CTC_MODE	-25	błędny tryb pracy CTC
LCO_NO_PARAMS	-26	nie podano parametrów przetwarzania a/c, c/a
LCO_OVERRUN	-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 modułem nie jest związane żadne przerwanie lub procedura obsługi nie została zainstalowana
LCO_NOT_FULLY_SUP	-30	żądany tryb wykonania funkcji nie jest dla danego typu modułu realizowany lub jest w opracowaniu
LCO_NO_EXTMEM	-31	brak pamięci rozszerzonej
LCO_NO_SEC_FREQ	-32	moduł nie może prowadzić przetwarzania z dwiema częstotliwościami
LCO_INTR_INST	-33	procedura obsługi przerwania jest już zainstalowana
LCO_BAD_PERZ ¹⁾	-34	błędna wielokrotność okresu próbkowania (0 lub 1)
LCO_BAD_MODE	-35	błędny tryb pracy
LCO_BAD_EXTMEM	-36	błędny adres bufora w pamięci rozszerzonej
LCO CTC NOT PROGRAMMED	-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 modułów serii LC-011-0812.

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	:
: 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	: zażądano inicjalizacji nie	:
		: istniejących modułów ale co najmniej 1	:
		: moduł został zainicjalizowany	:
:LCO_OTHER_LEN	: 2	: przepisano mniej próbek niż żądano	:
:LCO_PREMATURE_END	: 3	: przedwczesne zakończenie operacji z	:
		: powodu przepełnienia / opróżnienia	:
		: całego bufora	:
:LCO_IN_PROGRESS	: 4	: badana transmisja jeszcze trwa	:

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'a 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; numery przerwania dla poszczególnych driver'ów podane zostały w rozdziale 3.2.

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 (patrz p.3.2.; powodzenie tej próby świadczy o zainstalowaniu driver'a, niepowodzenie - o jego braku
- rozpoznanie konfiguracji modułu (GET_TOTAL_CONFIGURATION - ile i jakich modułów jest zainstalowanych, GET_MODULE_CONFIGURATION - czy moduł ma zainstalowany przetwornik c/a, czy jest podłączony do któregoś z przerwania itp., GET_INFO - czy tor pomiarowy a/c jest podłączony do kanału DMA, jakie są minimalne okresy próbkowania, jakie są zakresy napięć przetworników a/c i c/a 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)

- 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 być wywołana z błędnymi parametrami 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 itp.; poza tym funkcja może zostać wywołana z omyłkowo podanymi parametrami - 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.

UWAGA: Jeżeli w swoim programie użytkowym przechytujemy przerwanie zegarowe LC, to struktura programu powinna być następująca:

- wywołanie własnej obsługi przerwania,
- wykonanie funkcji MODULE_INIT (m.in. przechwycenie przerwania zegarowego przez driver),
- program pomiarowy,
- wykonanie funkcji LEAVE_DRIVER (m.in. wyinstalowanie obsługi przerwania zegarowego przez driver),
- wyinstalowanie własnej obsługi przerwania.

6.1. Programowanie w języku C.

Z poziomu języka C komunikacja z driver'em odbywa się za pośrednictwem procedury bibliotecznej `int86`. 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 pliki źródłowe `AMBEX-LC.ASM` zawierający definicje wszystkich struktur danych i stałych potrzebnych do współpracy z driver'em. Wydruk tego pliku znajduje się w Dodatku E.