



**DOKUMENTACJA DRIVERA
MODUŁU KONTROLNO-POMIAROWEGO**

LC-020-3212

WERSJA 2.0

Wersja: lipiec 1995

1.	Informacje ogólne.....	3
2.	Konfiguracja sterownika.....	3
3.	Opis sterownika.....	4
3.1.	Informacje wstępne.....	4
3.2.	Standardy oznaczeń, numeracja, dane charakterystyczne.....	4
3.3.	Komunikacja ze sterownikiem.....	5
3.4.	Typy warunków startu / zakończenia operacji.....	6
4.	Funkcje sterownika.....	7
4.1.	Inicjalizacja (MODULE_INIT).....	7
4.2.	Informacja o konfiguracji ogólnej (GET_TOTAL_CONFIGURATION).....	8
4.3.	Informacja o konfiguracji modułu (GET_MODULE_INFORMATION).....	9
4.4.	Informacja o szczegółach technicznych (GET_INFO).....	9
4.4.1.	Wejścia cyfrowe.....	9
4.4.2.	Wyjścia cyfrowe.....	9
4.4.3.	Wejścia analogowe.....	10
4.4.4.	Wyjścia analogowe.....	10
4.4.5.	Kanały układów licznikowo-czasowych (CTC).....	11
4.5.	Zadeklarowanie częstotliwości zegara magistrali (SET_CLOCK).....	11
4.6.	Ustawienie zakresu napięć (SET_VOLTAGE_RANGE).....	11
4.7.	Zadeklarowanie maksymalnego czasu obsługi obcych przerw (SET_TIME).....	12
4.8.	Oczekiwanie na zakończenie operacji (WAIT_FOR_END).....	12
4.9.	Przerwanie operacji (BREAK).....	12
4.10.	Wejście cyfrowe (DIGITAL_INPUT).....	13
4.11.	Wyjście cyfrowe (DIGITAL_OUTPUT).....	14
4.12.	Zapis CTC (CTC_WRITE).....	14
4.13.	Odczyt CTC (CTC_READ).....	14
4.14.	Transmisja danych (DATA_TRANSMIT).....	15
4.15.	Przetwarzanie analogowo-cyfrowe (ANALOG_INPUT).....	16
4.16.	Przetwarzanie cyfrowo-analogowe (ANALOG_OUTPUT).....	20
4.17.	Zakończenie pracy z sterownikiem (LEAVE_DRIVER).....	20
4.18.	Obsługa przerw (INTERRUPT_SERVICE).....	21
5.	Zestawienie kodów zakończenia funkcji.....	21
6.	Projektowanie programów użytkowych.....	23
6.1.	Programowanie w języku C.....	23
6.2.	Programowanie w języku Pascal.....	24
6.3.	Programowanie w języku assemblera.....	24

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 jako program rezydentny (TSR). Może on być zainstalowany i odinstalowany w trakcie pracy z komputerem, w zależności od potrzeb. Zaletami takiego rozwiązania są: możliwość zainstalowania i odinstalowania sterownika bez potrzeby inicjalizacji komputera, możliwość zwolnienia obszaru pamięci zajmowanej przez sterownik gdy nie jest on wykorzystywany, możliwość zwolnienia lub zmiany rozmiaru bufora danych gdy nie jest on wykorzystywany. Po zainstalowaniu sterownik służy tylko do obsługi przerwania programowego, właściwego dla określonego typu modułu.

Sterowniki modułów serii LC-020 podporządkowane zostały standardowi przyjętemu przez firmę. W standardzie tym przyjęto, że w komputerze może być zainstalowanych kilka (do czterech) modułów danego rodzaju obsługiwanych przez jeden sterownik. Ponieważ standard ten został zaprojektowany do różnych typów modułów, niektóre parametry wydawać się mogą nadmiarowe.

Używać sterownika należy:

- a) dokonać konfiguracji sterownika
np: C:\AMBEX\LC2032.EXE
- b) zainstalować sterownik w pamięci komputera
np: C:\AMBEX\LC2032.EXE/i

2. Konfiguracja sterownika.

Instalacja sterownika polega na skopiowaniu zbioru LC2032.EXE (odpowiedniego dla rodzaju modułu), do wybranego katalogu na dysku twardym (np C:\AMBEX\), bądź, w przypadku pracy z komputerem nie posiadającym dysku twardego, na dyskietkę systemową.

Sterownik instaluje się w systemie poprzez wywołanie programu LC2032.EXE bez opcji. Jeśli chcemy uruchamiać sterownik z katalogu innego niż bieżący, należy wpisać dopisać ten katalog do ścieżki dostępu (zmienna PATH= w zbiorze AUTOEXEC.BAT).

np: C:\AMBEX\LC2032.EXE

Konfiguracji sterownika dokonuje się wywołując program sterownika z opcją /i,

np: C:> LC2032 /i

Po uruchomieniu programu instalacyjnego domyślnie instalacja dotyczy programu który został uruchomiony (np jeśli pliki LC2032.EXE są umieszczone w różnych katalogach to modyfikacji będzie podlegał ten który jest na ścieżce dostępu). Następnie użytkownik powinien ustalić konfigurację modułów danego typu (jednorazowy przebieg programu umożliwia instalację sterownika 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.

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-020-3212;
- b) liczba zainstalowanych modułów danego typu: od 1 do 4;
- c) typ komputera: AT;
- d) pełna nazwa ścieżki do katalogu, gdzie skopiowano zbiór LC2032.EXE (łącznie z nazwą napędu dyskowego);
- e) wybór trybu pracy: z pamięcią rozszerzoną lub bez;
- f) rozmiar pamięci rozszerzonej w kilobajtach¹⁾;

Parametry dotyczące jednego modułu:

- g) nazwa modułu: od A do D (nazwa związana jest z adresem bazowym modułu);
- h) tryb pracy modułu: master lub slave (istotne przy współpracy kilku modułów): MASTER;
- i) częstotliwość zegara modułu: 8 MHz;

j) zakres napięć dla wejść analogowych;

1) przy pracy z pamięcią rozszerzoną można używać pamięci w standardzie XMS (sterownikiem takiej pamięci jest np program HIMEM.SYS). Chcąc pracować w standardzie XMS należy zadbać o zainstalowanie programu HIMEM.SYS w pliku CONFIG.SYS.

np: C:\DOS\HIMEM.SYS

Po wprowadzeniu żądanych parametrów pracy modułu (modułów) należy uruchomić funkcję "instalacja" (F10). Program modyfikuje ciało zbioru sterownika LC2032.EXE.

Po wywołaniu programu LC2032.EXE bez opcji instalowany jest sterownik karty i sprawdzana szybkość przetwornika a/c. 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 przerywania sprzętowego oraz numer tego przerywania wykrywane są automatycznie przez sterownik karty w momencie jego instalowania. W tym celu należy w odpowiednich polach podać opcję AUTO. Jeżeli instalacja nie wykrywa tych ustawień można podać wprost zadeklarowane nastawy, zostanie sprawdzona ich obecność. Kanały DMA są tylko sprawdzane.

3. Opis sterownika.

3.1. Informacje wstępne.

Przed sterownikiem 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 upływem 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

Sterownik rozpoczyna swoją pracę w momencie zainstalowania programu TSR 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. Następnie wykonywane są testy, których celem jest sprawdzenie czy i do jakich przerwań i kanałów DMA podłączony jest każdy z zainstalowanych modułów.

Sterownik 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 sterownika 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 LC0_;
- nazwy występujące w dokumentacji są identyczne z nazwami występującymi w plikach źródłowych dla języków C, Pascal, assembler (z dokładnością do rozróżnienia małe / duże litery).

Numery przerwań programowych związanych z sterownikami poszczególnych typów modułów:

typ modułu	nazwa przerwania	numer przerwania	
		szesnastkowy	dziesiętny
LC-011-1612	LC011_16	91	145
LC-020-0812	LC020_08_2	94	148
LC-020-3212	LC020_32	95	149

Wszystkie sterowniki widziane są w systemie DOS jako urządzenia. Nazwy tych urządzeń są następujące:

typ modułu	nazwa sterownika	nazwa urządzenia
LC-011-1612	LC1116x.xxx	LC1116^^
LC-020-0812	LC2008x.xxx	LC2008^2
LC-020-3212	LC2032x.xxx	LC2032^2

Serowniki mogą być wykryte w typowy sposób: poprzez przeglądanie zainstalowanych urządzeń przez wywołanie przerwania 2F¹⁶ i poszukiwanie urządzenia o podanej nazwie.

Kodowanie numerów modułów:

moduł	nazwa kodu	wartość
A	LC0_MODA	1
B	LC0_MODB	2
C	LC0_MODC	3
D	LC0_MODAL	4

Kodowanie typów urządzeń w modułach:

urządzenie	nazwa kodu	wartość
porty cyfrowe wejściowe	LC0_DINPUT	1
porty cyfrowe wyjściowe	LC0_DOUTPUT	2
przetworniki a/c	LC0_AINPUT	3
przetworniki c/a	LC0_AOUTPUT	4
kanały CTC	LC0_CTC	5

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

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

3.3. Komunikacja ze sterownikiem.

Funkcje sterownika wywoływane są poprzez przerwanie programowe (numery przerwań - patrz p. 3.2.). Przesyłanie informacji pomiędzy programem użytkowym a sterownikiem 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 sterownika.

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

adres rekordu: DX:DI

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach

LC0_CODE określa funkcję jaka ma być wykonana przez driver a zarazem sposób interpretacji ciągu bajtów znajdujących się pod adresem DX:DI.

LC0_STATUS informuje program wywołujący driver o poprawności wykonania funkcji:

LC0_STATUS = 0: wykonanie poprawne

LC0_STATUS < 0: wykonanie błędne

LC0_STATUS > 0: wykonanie poprawne z zastrzeżeniami (ostrzeżenia)

LC0_ERR_STAT (jeżeli jest mniejszy od zera) niesie pewne dodatkowe informacje komentujące zwrócony w LC0_STATUS kod błędu. Dotyczy to dwóch sytuacji:

- błędnie podane parametry warunku startu / stopu (LC0_STATUS = LC0_ILL_START / LC0_ILL_STOP);

LC0_ERR_STAT precyzuje co zostało podane błędnie

- funkcja przerwana wywołaniem funkcji BREAK (LC0_STATUS = LC0_BROKEN); LC0_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 LC0_STATUS = 0 i LC0_ERR_STAT < 0 (patrz opis funkcji, p.4.9).

W rozdziale 5 podano tabele wszystkich kodów zwracanych przez LC0_STATUS i LC0_ERR_STAT.

3.4. Typy warunków startu / zakończenia operacji.

W driverze zaimplementowano następujące 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 (LC0_SIMMED natychmiast)	---	---	---	---	---
1 (LC0_SHARD) od sygnału sprzętowego	---	---	---	---	---
2 (LC0_SLEVEL) ¹⁾ od poziomu sygnału cyfrowego	numer modułu	numer portu	numer wejścia	poziom	---
3 (LC0_SSLOPE) ²⁾ od zbocza sygnału cyfrowego	numer modułu	numer portu	numer wejścia	zbocze	---
4 (LC0_SDIG_EQ) ³⁾ od kombinacji wejść cyfrowych warunek równości	numer modułu	numer portu	maska	wzorzec	---
5 (LC0_SDIG_NE) ³⁾ od kombinacji wejść cyfrowych warunek nierówności	numer modułu	numer portu	maska	wzorzec	---
6 (LC0_STIME) po upływie czasu określonego czasu	odcinek czasu w sekundach				---
7 (LC0_SDATE) ⁴⁾ o podanym czasie	sekunda	minuta	godzina	dzień 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)/	parametry warunku				
znaczenie	bajt 1	bajt 2	bajt 3	bajt 4	bajt 5
00 (LC0_ZSAMPLES) po zmierzeniu bloku danych	liczba próbek do zmierzenia				---

4. Funkcje sterownika.

W poniższych rozdziałach opisano wszystkie funkcje sterownika. 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 asemblera (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 (pół 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 LC0_STATUS; jeżeli punkt ten nie występuje to oznacza to, że dana funkcja nie zwraca żadnych ostrzeżeń

- błędy; lista błędów zwracanych przez funkcję w parametrze LC0_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 LC0_ERR_STAT; jeżeli punkt ten nie występuje to oznacza to, że dana funkcja nie zwraca żadnych dodatkowych informacji (LC0_ERR_STAT zawsze równe LC0_E_OK: 0)

4.1. Inicjalizacja (MODULE_INIT).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 0
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_IMODULE	1	mapa modułów

Przeznaczenie:

Funkcja powoduje zainicjalizowanie pracy (zerowanie) wyspecyfikowanych modułów.

Moduły podlegające inicjalizacji specyfikuje się na poszczególnych bitach parametru LC0_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 1C16 (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 (2C16 i 2A16). 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 1C16 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 LC0_MOD_NEWPAR (patrz opis funkcji ANALOG_INPUT i ANALOG_OUTPUT).

Ostrzeżenia (LC0_STATUS):

LC0_NON_EX_MOD - zażądano inicjalizacji nie istniejących modułów ale co najmniej 1 moduł został zainicjalizowany

Błędy (LC0_STATUS):

LC0_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
LC0_CODE	1	kod funkcji = 1
LC0_STATUS	1	kod zakończenia funkcji = LC0_OK
LC0_ERR_STAT	1	dodatkowe informacje o błędach
parametry wyjściowe:		
LC0_TONF	1	konfiguracja modułów
LC0_TIAD	1	liczba dostępnych przetworników a/c
LC0_TIDA	1	liczba dostępnych przetworników c/a
LC0_TCTC	1	liczba dostępnych kanałów CTC
LC0_TIDI	1	liczba dostępnych portów we. cyfrowych
LC0_TIDO	1	liczba dostępnych portów wy. cyfrowych
LC0_TMEMA	4	adres bufora w pamięci rozszerzonej (adres absolutny)
LC0_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 i LC-020 wszystkie moduły są typu "master")

4.3. Informacja o konfiguracji modułu (GET_MODULE_INFORMATION).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 2
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_MMODULE		numer modułu
parametry wyjściowe:		
LC0_MBASE1	2	adres bazowy modułu - pakiet 1
LC0_MBASE2 ¹⁾	2	adres bazowy modułu - pakiet 2
LC0_MIAD	1	liczba dostępnych przetworników a/c
LC0_MIDA	1	liczba dostępnych przetworników c/a
LC0_MCTC	1	liczba dostępnych kanałów CTC
LC0_MIDI	1	liczba dostępnych portów we. cyfrowych
LC0_MIDO	1	liczba dostępnych portów wy. cyfrowych
LC0_MCLOCK	2	częstotliwość zegara modułu w kHz;
LC0_MINT	1	numer przerwania (programowy)

1) dla modułów serii LC-011 i LC-020 - nieokreślone (moduły są jednopakietowe)

Przeznaczenie:

Funkcja zwraca informację o konfiguracji wyspecyfikowanego modułu.

Błędy (LC0_STATUS):

LC0_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
LC0_CODE	1	kod funkcji = 3
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_GTYPE	1	rodzaj urządzenia = LC0_DINPUT
LC0_GMODULE	1	numer modułu
LC0_GNUM	1	numer portu wejściowego
parametry wyjściowe:		
LC0_GCHAN	1	liczba bitów portu

Przeznaczenie:

Funkcja nie jest realizowana dla opisanych modułów.

4.4.2. Wyjścia cyfrowe.

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 3
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_GTYPE	1	rodzaj urządzenia = LC0_DOUTPUT
LC0_GMODULE	1	numer modułu
LC0_GNUM	1	numer portu wyjściowego

parametry wyjściowe:		
LC0_GCHAN	1	liczba bitów portu

Przeznaczenie:

Funkcja nie jest realizowana dla opisanych modułów.

4.4.3. Wejścia analogowe.

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 3
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dotkliwe informacje o błędach
LC0_GTYPE	1	rodzaj urządzenia = LC0_AINPUT
LC0_GMODULE	1	numer modułu
LC0_GNUM	1	numer przetwornika
parametry wyjściowe:		
LC0_GCHAN	1	liczba kanałów przetwornika
LC0_GRES	1	liczba bitów przetwornika
LC0_GTIME	2	czas konwersji przetwornika w ns
LC0_GMINV	1	dolna granica zakresu napięć w dziesiątych częściach wolta
LC0_GMAXV	1	górną granicę zakresu napięć w dziesiątych częściach wolta
LC0_GDMA	1	numer kanału DMA
LC0_GMINP	64	tablica minimalnych okresów próbkowania

Przeznaczenie:

Funkcja zwraca informację o konfiguracji badanego toru analogowo-cyfrowego.

Tablica LC0_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 (LC0_STATUS):

LC0_NONEX_DEV - przetwornik a/c o tym numerze nie istnieje

LC0_NO_MODULE - nie ma takiego modułu

4.4.4. Wyjścia analogowe.

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 3
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dotkliwe informacje o błędach
LC0_GTYPE	1	rodzaj urządzenia = LC0_AOUTPUT
LC0_GMODULE	1	numer modułu
LC0_GNUM	1	numer przetwornika
parametry wyjściowe:		
LC0_GCHAN	1	liczba kanałów przetwornika
LC0_GRES	1	liczba bitów przetwornika
LC0_GTIME	2	czas konwersji przetwornika w ns
LC0_GMINV	1	dolna granica zakresu napięć w dziesiątych częściach wolta
LC0_GMAXV	1	górną granicę zakresu napięć w dziesiątych częściach wolta

LC0_GDMA	1	numer kanału DMA
----------	---	------------------

Przeznaczenie:

Funkcja nie jest realizowana dla opisanych modułów.

4.4.5. Kanały układów licznikowo-czasowych (CTC).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 3
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dotatkowe informacje o błędach
LC0_GTYPE	1	rodzaj urządzenia = LC0_CTC
LC0_GMODULE	1	numer modułu
LC0_GNUM	1	numer kanału

Przeznaczenie:

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

Błędy (LC0_STATUS):

LC0_BAD_DEV_TYP - brak dostępnych kanałów CTC

LC0_NONEX_DEV - kanał CTC o tym numerze nie istnieje

LC0_NO_MODULE - nie ma takiego modułu

4.5. Zadeklarowanie częstotliwości zegara magistrali (SET_CLOCK).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 4
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dotatkowe informacje o błędach
LC0_LCLOCK	2	częstotliwość zegara w kHz

Przeznaczenie:

Funkcja nie jest realizowana dla opisanych modułów.

4.6. Ustawienie zakresu napięć (SET_VOLTAGE_RANGE).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 5
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dotatkowe informacje o błędach
LC0_VTYPE	1	rodzaj urządzenia (LC0_AINPUT lub LC0_AOUTPUT)
LC0_VMODULE	1	numer modułu
LC0_VNUM	1	numer przetwornika
LC0_VMINV	1	dolna granica zakresu napięć w dziesiątych częściach wolta
LC0_VMAXV	1	górną granicą zakresu napięć w dziesiątych częściach wolta

Przeznaczenie:

Funkcja nie jest realizowana dla opisanych modułów.

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

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 6
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_ETIME	2	maksymalny czas obsługi w dziesiątych częściach ###s

Przeznaczenie:

Funkcja nie jest realizowana dla opisanych modułów.

4.8. Oczekiwanie na zakończenie operacji (WAIT_FOR_END).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 7
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_WTYPE	1	rodzaj urządzenia (LC0_AINPUT lub LC0_AOUTPUT)
LC0_WMODULE	1	numer modułu
LC0_WNUM	1	numer przetwornika
LC0_WMODE	1	tryb pracy
parametry wyjściowe:		
LC0_WRMNUM	4	rzeczywista liczba zmierzonych / wysłanych próbek
LC0_WREMAR	2	rzeczywista długość marginesu końcowego

Przeznaczenie:

Funkcja jest realizowana tylko dla trybu pracy asynchronicznej.

4.9. Przerwanie operacji (BREAK).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 8
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_BMODE	1	tryb pracy
LC0_BPROC	4	adres procedury obsługi Ctrl-Break

Przeznaczenie:

Przerywanie pracy sterownika i modułu, instalowanie i wyinstalowywanie procedur obsługi przerwania generowanego przez klawisze Ctrl-Break.

Tryby pracy:

nazwa	wartość	znaczenie
LC0_BREAK_EXEC	0	przerwij pracę sterownika i modułu
LC0_BREAK_INST	1	zainstaluj procedurę obsługi przerwania Ctrl-Break

LC0_BREAK_UNINST	2	wyinstaluj procedurę obsługi przerwania Ctrl-Break
------------------	---	--

Przerwanie pracy - tryb LC0_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 LC0_STATUS = LC0_NO_OPER, LC0_ERR_STAT = LC0_E_OK.
2. Moduł wykonuje pod kontrolą sterownika operację synchroniczną. Ponieważ w tym przypadku należy przerwać pracę sterownika, zazwyczaj funkcja wywoływana jest w procedurze obsługi przerwania, np. Ctrl-Break. Driver zwraca LC0_STATUS = LC0_OK i LC0_ERR_STAT = LC0_E_OK.

Zainstalowanie procedury obsługi - tryb LC0_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 (LC0_BPROC, daleki adres offset-segment; jest to procedura typu interrupt) lub zlecić obsługę standardową - przez procedurę wewnętrzną sterownika (LC0_BPROC = 0:0). Procedura obsługi przerwania musi zawierać wykonanie funkcji BREAK w trybie pracy LC0_BREAK_EXEC (patrz wyżej; to też jest jedyną treścią standardowej procedury obsługi, dostarczanej przez driver).

Wyinstalowanie procedury obsługi - tryb LC0_BREAK_UNINST:

Funkcja przywraca procedurę obsługi przerwania istniejącą przed pierwszym wywołaniem funkcji BREAK w trybie LC0_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 (LC0_STATUS):

- LC0_INTR_INST - procedura obsługi Ctrl-Break jest już zainstalowana (dla żądania zainstalowania procedury)
- LC0_NO_OPER - z żadnym modulem nie jest związana żadna operacja w toku (dla trybu pracy LC0_BREAK_EXEC)
- LC0_BAD_MODE - błędny tryb pracy

Dodatkowe informacje (LC0_ERR_STAT; tylko dla trybu pracy LC0_BREAK_EXEC):

- LC0_E_BROKEN_RUN - funkcja wywołana w trakcie przetwarzania przy operacji asynchronicznej
- LC0_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
LC0_CODE	1	kod funkcji = 9
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_DMODULE	1	numer modułu
LC0_DNUM	1	numer portu
LC0_DSTST	1	typ warunku startu operacji
LC0_DVAL	1	odczytana wartość
LC0_DSTART	5	parametry warunku startu

Przeznaczenie:

Funkcja nie jest realizowana dla opisanych modułów.

4.11. Wyjście cyfrowe (DIGITAL_OUTPUT).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 10
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_OMODULE	1	numer modułu
LC0_ONUM	1	numer portu
LC0_OSTST	1	typ warunku startu operacji
LC0_OVAL	1	zapisywana wartość
LC0_OSTART	5	parametry warunków startu

Przeznaczenie:

Funkcja nie jest realizowana dla opisanych modułów.

4.12. Zapis CTC (CTC_WRITE).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 11
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_CMODULE	1	numer modułu
LC0_CMODE	1	tryb pracy funkcji
LC0_CFUN	1	tryb pracy kanału CTC
LC0_CVAL	2	wpisywana wartość licznika

Przeznaczenie:

Funkcja nie jest realizowana dla opisanych modułów.

4.13. Odczyt CTC (CTC_READ).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 12
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_UMODULE	1	numer modułu
LC0_UNUM	1	numer kanału CTC
parametry wyjściowe:		
LC0_UVAL	2	odczytana wartość licznika

Przeznaczenie:

Funkcja nie jest realizowana dla opisanych modułów.

4.14. Transmisja danych (DATA_TRANSMIT).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 13
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_RMODE	1	tryb przesyłania
LC0_RADDR	4	adres bufora (offset-segment)
LC0_RLEN	4	długość bufora
LC0_RMEAS	4	numer próbki, od której należy zacząć
LC0_RNUM	4	liczba próbek do przesłania
LC0_RMEMA	4	adres absolutny bufora w pamięci rozszerzonej
parametry wyjściowe:		
LC0_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. LC0_RMODE

Tryb pracy funkcji:

nr bitu	wartość	nazwa	znaczenie
1	1	LC0_TO_EXT_DIR	do pamięci rozszerzonej
	0	LC0_FROM_EXT_DIR	z pamięci rozszerzonej
2..8	---	---	zarezerwowane; zawsze 0

b. LC0_RADDR

Adres bufora w podstawowej pamięci komputera.

c. LC0_RLEN

Długość bufora LC0_RADDR podana w próbkach. Ten parametr nie jest związany z liczbą próbek do przesłania.

d. LC0_RMEAS

Numer pierwszej próbki do przesłania. Przesłane zostaną próbki o numerach LC0_RMEAS, LC0_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. LC0_RNUM

Całkowita liczba próbek do przesłania.

f. LC0_RMEMA

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

g. LC0_RRNUM

Rzeczywista liczba przepisanych próbek. Parametr LC0_RRNUM przybiera wartość będącą minimum z LC0_RLEN, <n> i LC0_RNUM. <n> jest tu liczbą próbek zawartych między próbka LC0_RMEAS a końcem bufora w pamięci rozszerzonej.

Ostrzeżenia (LC0_STATUS):

LC0_OTHER_LEN - przepisano mniejszą liczbę próbek niż żądano

Błędy (LC0_STATUS):

LC0_BAD_BUF_ADR - błędny adres bufora w pamięci podstawowej (odnoszący się do nieistniejącej pamięci)

LC0_BAD_BUF_LEN - błędna długość bufora w pamięci podstawowej (powodująca wyjście bufora poza pamięć itp.)

LC0_BAD_EXTMEM - błędny adres bufora w pamięci rozszerzonej

LC0_BAD_MNUM - błędny numer pierwszej próbki (np. powodujący wyjście poza jeden z buforów)

LC0_BAD_MODE - błędny tryb pracy

LC0_BROKEN - transmisja przerwana z powodu wykonania funkcji BREAK

LC0_NO_EXTMEM - brak pamięci rozszerzonej

4.15. Przetwarzanie analogowo-cyfrowe (ANALOG_INPUT).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 14
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_AMODULE	1	numer modułu
LC0_ANUM	1	numer przetwornika
LC0_AMODE	2	tryb pracy funkcji
LC0_ASTST	1	typy warunków startu / stopu operacji
LC0_APER	4	okres próbkowania
LC0_APER2 ¹⁾	2	wielokrotność okresu próbkowania dla kanałów dodatkowych
LC0_ACHAN	1	liczba kanałów / numer kanału
LC0_ACHAN2 ¹⁾	1	liczba kanałów dodatkowych
LC0_AADDR	4	adres bufora (offset-segment) w pamięci podstawowej
LC0_ALEN	4	długość bufora w pamięci podstawowej
LC0_AMEMA	4	adres absolutny bufora w pamięci rozszerzonej
LC0_ABMAR ³⁾	2	offset bufora zawierającego dane do programowania pamięci
LC0_AEMAR ³⁾	2	segment bufora zawierającego dane do programowania pamięci
LC0_AHAND ¹⁾	2	numer handlera pliku dyskowego
LC0_ASTART	5	warunki startu
LC0_ASTOP	5	warunki stopu
parametry wyjściowe:		
LC0_ARDIV1	2	rzeczywisty dzielnik zegara modułu
LC0_ARDIV2	2	- " -
LC0_ARMNUM	4	rzeczywista liczba zmierzonych próbek
LC0_ARBMAR	2	rzeczywista dł. marginesu początkowego
LC0_AREMAR ²⁾	2	rzeczywista dł. marginesu końcowego
LC0_ARLEN	4	rzeczywista liczba przepisanych próbek
LC0_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.

3) Wielkość istotna jedynie dla pracy z pamięcią sekwencji pomiarowej (LC0_MOD_RAM_SEK = 1), w przeciwnym przypadku bez znaczenia dla modułów serii LC-011 i LC-020.

Przeznaczenie:

Jest to podstawowa funkcja sterownika 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. LC0_AMODE

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

nazwa	nr bitu	uwagi
LC0_MOD_START	1	
LC0_MOD_NEW_PAR	2	
LC0_MOD_SYNCHR	3	1
LC0_MOD_INTR	4	ignorowane
LC0_MOD_INTR_TYPE	5	ignorowane
LC0_MOD_BLOCK	6	1
LC0_MOD_CYCL	7	
LC0_MOD_FILE	8	0
LC0_MOD_MEM_W	9	
LC0_MOD_EXT_CLK	10	ignorowane
LC0_MOD_EXT_MEM	11	
LC0_MOD_RAM_SEK	12	
-----	13..16	zarezerwowane; zawsze 0

LC0_MOD_START:

- start pomiarów

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

LC0_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 LC0_NO_PARAMS

LC0_MOD_SYNCHR

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

- praca synchroniczna: driver zwraca sterowanie dopiero po całkowitym zakończeniu przetwarzania.

- praca asynchroniczna: tryb pracy z przerwaniem (LC0_MOD_INTR = 1) na koniec

przetwarzania (LC0_MOD_INTR_TYPE = 0); driver zwraca sterowanie natychmiast po rozpoczęciu przetwarzania. W przypadku pomiaru przekraczającego pojedynczy blok transmisji DMA, lub przy pracy z buforem cyklicznym, należy w obsłudze przerwania z karty wywołać funkcję ANALOG_INPUT. Przy pracy w tym trybie możliwa jest praca on-line tzn. wyświetlanie i/lub analiza wyników pomiaru na bieżąco.

UWAGA: Praca asynchroniczna stanowi opcję w przygotowaniu.

LC0_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 (LC0_AMODE) powinien być ustawiony bit LC0_MOD_NEW_PAR (wówczas istotne są tylko parametry: LC0_AMODULE, LC0_ANUM, LC0_AMODE, LC0_ACHAN, LC0_AADDR, LC0_ALEN, LC0_ASTST i LC0_ASTART; nie jest analizowany okres próbkowania LC0_APER;

UWAGA: 1. 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 LC0_SIMMED, 2. moduł LC-020-3212 nie obsługuje tego trybu.

LC0_MOD_CYCL

- praca z buforem cyklicznym (1) / bez bufora (0):

praca z buforem cyklicznym polega zaprogramowaniu specjalnego trybu pracy kanału DMA, w którym po dojściu do końca bufora adres transmisji ustawiany jest na jego początek i transmisja jest kontynuowana; ze względu na ograniczenia sprzętowe układów DMA.

Jeśli ilość próbek do zmierzenia wynosi 0 pracujemy z buforem cyklicznym.

UWAGA: bufor cykliczny nigdy nie może być większy niż 64 KS i jest zawsze alokowany w obszarze pierwszego bloku pamięci pomiaru.

LC0_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 LC0_AADDR) przepisywanych jest <n> pierwszych próbek gdzie <n> jest minimum z całkowitej liczby zmierzonych próbek (LC0_ARMNUM) i rozmiaru bufora (LC0_ALEN); całkowita liczba przepisanych próbek zwracana jest w parametrze LC0_ARLEN; jeżeli natomiast bit LC0_MOD_MEM_W jest równy 0 to po zakończeniu przetwarzania dane nie są przepisywane do pamięci (wówczas

parametry LC0_AADDR i LC0_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 (LC0_MOD_EXT_MEM = 0) bit LC0_MOD_MEM_W jest ignorowany.

LC0_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 LC0_AMEMA; bufor ten musi się zawierać w buforze zadeklarowanym przy instalacji sterownika; 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 (LC0_AADDR) i długość (LC0_ALEN) bufora w pamięci podstawowej są analizowane tylko wtedy gdy jednocześnie zapalony zostanie bit LC0_MOD_MEM_W (patrz wyżej); dla pracy z pamięcią podstawową adres bufora w pamięci rozszerzonej (LC0_AMEMA) nie jest analizowany; LC0_MOD_RAM_SEK:

- praca z pamięcią sekwencji pomiarowej (1) / według parametrów standardowych(0) (zadanych przez LC0_ACHAN, ...): przy pracy z pamięcią sekwencji pomiarowej parametry sekwencji pomiarowej są umieszczone w tablicy o długości równej 2048 bajtów wskazywanej przez parametry LC0_ABMAR i LC0_AEMAR.

UWAGA: jeśli sterownik nie jest zainstalowany na pamięć rozszerzoną w standardzie XMS użytkownik musi zadbać o to, aby wykonanie pomiaru w sposób zadeklarowany w pamięci sekwencji pomiarowej dawało ilość bajtów będącą całkowitą podwielokrotnością rozmiaru strony transmisji DMA (64 KB). Powyższe ograniczenie nie dotyczy pamięci sekwencji nieprzekraczających 32 próbek.

- tryb standardowy: pomiar prowadzony jest zgodnie z zadeklarowanymi parametrami: LC0_ACHAN, ...

b. LC0_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. LC0_ASTART, LC0_ASTOP

Parametry te określają szczegółowo warunki startu i stopu operacji.

Interpretacja ich zależna jest od zadanych typów warunków startu i stopu operacji (LC0_ASTST). Szczegółowy opis - patrz p.“3.4.

d. LC0_APER

Okres próbkowania dla pracy blokowej z zegarem wewnętrznym (bit parametru LC0_AMODE LC0_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 LC0_GMINP, patrz opis funkcji).) Liczba kanałów określająca minimalny okres próbkowania wyznaczana jest następująco:

- dla pracy jednokanałowej (najstarszy bit parametru LC0_ACHAN równy 1, patrz niżej opis tego parametru) - oczywiście 1

- dla pracy wielokanałowej (najstarszy bit parametru LC0_ACHAN równy 0, patrz niżej opis tego parametru) - wartość LC0_ACHAN

Z parametrem tym związane są parametry zwrotne sterownika LC0_ARDIV1 i LC0_ARDIV2 (patrz opis poniżej).

e. LC0_ACHAN

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

b8	tryb pracy	b1..b7
0	praca wielokanałowa	liczba kanałów (2..16)
1	praca jednokanałowa	numer kanału (1..16)

f. LC0_AADDR, LC0_ALEN

Adres i długość bufora w pamięci podstawowej. Parametry te są analizowane tylko wtedy, gdy bit LC0_MOD_EXT_MEM trybu pracy (LC0_AMODE) został ustawiony na 0 (praca z pamięcią podstawową) lub - w przeciwnym razie - gdy bit LC0_MOD_MEM_W = 1 i LC0_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 LC0_MOD_MEM_W).

h. LC0_AMEMA

Parametr istotny tylko dla pracy z pamięcią rozszerzoną (LC0_MOD_EXT_MEM = 1). Oznacza absolutny, 24-bitowy adres bufora w pamięci rozszerzonej. Adres ten nie może wychodzić poza obszar bufora zadeklarowanego przy instalacji sterownika.

i. LC0_ARDIV1, LC0_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 to większy od zadanego. Parametr LC0_ARDIV1 przekazuje wartość pierwszego licznika natomiast LC0_ARDIV2 - drugiego (rzeczywisty dzielnik zegara modułu jest równy iloczynowi LC0_ARDIV1 * LC0_ARDIV2). Łącznie z częstotliwością zegara modułu (patrz funkcja GET_MODULE_INFORMATION) daje informację o faktycznej częstotliwości próbkowania.

j. LC0_ARMNUM

Rzeczywista liczba zmierzonych próbek.

k. LC0_ARBMAR

Przesunięcie początku zmierzonych próbek względem początku bufora w pamięci podstawowej zadeklarowanego przez klienta.

l. LC0_ARBUF

Początek bufora cyklicznego lub bufora pretriggeringu w segmencie danych (64 KS) przeznaczonym na ten bufor.

Błędy (LC0_STATUS):

LC0_BAD_BUF_ADR - błądny adres bufora (odnoszący się do nieistniejącej pamięci lub do pamięci rozszerzonej w przypadku transmisji programowej)

LC0_BAD_BUF_LEN - błędna długość bufora (powodująca wyjście bufora poza pamięć, przejście pomiędzy pamięcią podstawową a rozszerzoną, przekroczenie rozmiarów bufora w pamięci rozszerzonej itp.)

LC0_BAD_CHAN - numer nieistniejącego kanału

LC0_BAD_CHAN_N - zła liczba kanałów

LC0_BAD_EXTMEM - błądny adres bufora w pamięci rozszerzonej (dla pracy z pamięcią rozszerzoną, LC0_MOD_EXT_MEM = 1)

LC0_BAD_MARGIN - błędna długość marginesu początkowego (nie będąca wielokrotnością liczby kanałów)

LC0_BAD_MODE - błądny tryb pracy

LC0_BROKEN - operacja przerwana z powodu wykonania funkcji BREAK; LC0_ERR_STAT podaje, w jakim momencie operacja została przerwana

LC0_BAD_PER - za krótki okres próbkowania

LC0_DEV_BUSY - urządzenie zajęte; próba wykonania następnej funkcji ANALOG_INPUT przed zakończeniem poprzedniej

LC0_ILL_START - błędne parametry warunku startu

LC0_ILL_STOP - błędne parametry warunku stopu; w obu przypadkach sprecyzowanie błędu podane jest w LC0_ERR_STAT

LC0_ILL_START_CODE - nielegalny typ warunku startu

LC0_ILL_STOP_CODE - nielegalny typ warunku stopu;

LC0_INTR_NOT_INST - procedura obsługi przerwania nie jest zainstalowana (dla pracy z przerwaniem)

LC0_NONEX_DEV - nie istnieje przetwornik o tym numerze

LC0_NO_DMA - z danym przetwornikiem nie jest związany żaden kanał DMA (dla pracy blokowej)

LC0_NO_EXTMEM - brak pamięci rozszerzonej (dla pracy z pamięcią rozszerzoną)

LC0_NO_IRQ - z danym modułem nie jest związane żadne przerwanie (dla pracy z przerwaniem)

LC0_NO_MODULE - nie ma takiego modułu

LC0_NO_PARAMS - zgaszono bit LC0_MOD_NEW_PAR (LC0_AMODE) ale wcześniej nie ustawiono żadnych parametrów (nie było wykonania funkcji ANALOG_INPUT lub ostatnie wykonanie było niepoprawne)

LC0_NO_SECOND_FREQ - moduł nie może prowadzić przetwarzania z dwiema częstotliwościami (LC0_ACHAN2 <> 0)

LC0_OVERRUN - zakończono przetwarzanie z powodu błędu OVERRUN

LC0_TOO_LONG_MARG - suma marginesów dłuższa od bufora

Dodatkowe informacje o błędach (LC0_ERR_STAT):

LC0_E_BAD_CHAN - numer nieistniejącego kanału

LC0_E_BAD_DATE - zła specyfikacja daty

LC0_E_BAD_TIME - zły odcinek czasu

LC0_E_BROKEN_RUN - funkcja przerwana w trakcie przetwarzania

LC0_E_BROKEN_WAIT - funkcja przerwana w trakcie oczekiwania na spełnienie warunku startu
 LC0_E_NONEX_DEV - nie istnieje urządzenie o tym numerze
 LC0_E_NO_MODULE - nie ma takiego modułu

4.16. Przetwarzanie cyfrowo-analogowe (ANALOG_OUTPUT).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 15
LC0_STATUS	1	kod zakończenia funkcji
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_NMODULE	1	numer modułu
LC0_NNUM	1	numer przetwornika
LC0_NMODE	2	tryb pracy funkcji
LC0_NSTST	1	typy warunków startu / stopu operacji
LC0_NCHAN	1	liczba kanałów / numer kanału
LC0_NPER	4	okres sterowania
LC0_NADDR	4	adres bufora
LC0_NLEN	4	długość bufora
LC0_NHAND	2	numer handlera pliku dyskowego
LC0_NSTART	5	warunki startu
LC0_NSTOP	5	warunki stopu

Przeznaczenie:

Funkcja nie jest realizowana dla opisanych modułów.

4.17. Zakończenie pracy z sterownikiem (LEAVE_DRIVER).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 16
LC0_STATUS	1	kod zakończenia funkcji = LC0_OK
LC0_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 sterownika np. w sytuacji, gdy następny program żąda wykonania przetwarzania a/c wg. poprzednich parametrów (LC0_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 IC16 przez driver.

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

4.18. Obsługa przerw (INTERRUPT_SERVICE).

Rekord opisu zlecenia:

nazwa	rozmiar w bajtach	znaczenie
LC0_CODE	1	kod funkcji = 17
LC0_STATUS	1	kod zakończenia funkcji = LC0_OK
LC0_ERR_STAT	1	dodatkowe informacje o błędach
LC0_SMODULE	1	numer modułu
LC0_SPROC	4	adres procedury użytkownika
LC0_SSTAT	4	adres słowa komunikacyjnego

Przeznaczenie:

Funkcja w przygotowaniu (opcja).

5. Zestawienie kodów zakończenia funkcji.

nazwa	kod	znaczenie
LC0_OK	0	poprawne zakończenie funkcji

Błędy (LC0_STATUS):

nazwa	kod	znaczenie
LC0_UNKN_FUNC	-1	nieznany kod funkcji
LC0_NO_MODULE	-2	nie istnieje żaden z żądanych modułów; nie ma takiego modułu
LC0_BAD_DEV_TYP	-3	brak urządzeń danego typu
LC0_NONEX_DEV	-4	nie istnieje urządzenie o tym numerze
LC0_BAD_FREQ	-5	błędna częstotliwość
LC0_BAD_RANGE	-6	błędny zakres napięcie
LC0_NO_OPER	-7	z wyspecyfikowanym urządzeniem nie jest związana żadna operacja w toku
LC0_BAD_MARGIN	-8	błędna długość marginesu początkowego (nie będąca wielokrotnością liczby kanałów
LC0_BAD_BUF_ADR	-9	błędny adres bufora (odnoszący się do nieistniejącej pamięci lub do pamięci rozszerzonej w przypadku transmisji programowej)
LC0_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.)
LC0_DEV_BUSY	-11	urządzenie zajęte
LC0_BAD_PER	-12	za długi lub za krótki okres
LC0_BAD_CHAN N	-13	zła liczba kanałów
LC0_BAD_CHAN	-14	numer nieistniejącego kanału
LC0_BROKEN	-15	przetwarzanie przerwane z powodu wykonania funkcji BREAK
LC0_INTR_NOT_INST	-16	procedura obsługi przerwania nie jest zainstalowana
LC0_ILL_START_CODE	-17	nielegalny typ warunku startu
LC0_ILL_STOP_CODE	-18	nielegalny typ warunku stopu
LC0_BAD_PROC	-19	błędny adres procedury obsługi przerwania lub słowa komunikacyjnego (spoza pamięci podstawowej)
LC0_TOO_LONG_MARG	-20	suma marginesów dłuższa od bufora
LC0_ILL_START	-21	błędne parametry warunku startu
LC0_ILL_STOP	-22	błędne parametry warunku stopu
LC0_BAD_MNUM	-23	błędny numer pierwszej próbki
LC0_NOT_SUPPORTED	-24	dla danego modułu funkcja nie jest realizowana

LC0_BAD_CTC_MODE	-25	błędny tryb pracy CTC
LC0_NO_PARAMS	-26	nie podano parametrów przetwarzania a/c, c/a
LC0_OVERRUN	-27	zakończono przetwarzanie a/c z powodu błędu OVERRUN
LC0_NO_DMA	-28	z danym urządzeniem nie jest związany żaden kanał DMA
LC0_NO_IRQ	-29	z danym modulem nie jest związane żadne przerwanie lub procedura obsługi nie została zainstalowana
LC0_NOT_FULLY_SUP	-30	żądany tryb wykonania funkcji nie jest dla danego typu modułu realizowany lub jest w opracowaniu
LC0_NO_EXTMEM	-31	brak pamięci rozszerzonej
LC0_NO_SEC_FREQ	-32	moduł nie może prowadzić przetwarzania z dwiema częstotliwościami
LC0_INTR_INST	-33	procedura obsługi przerwania jest już zainstalowana
LC0_BAD_PER2 ¹⁾	-34	błędna wielokrotność okresu próbkowania (0 lub 1)
LC0_BAD_MODE	-35	błędny tryb pracy
LC0_BAD_EXTMEM	-36	błędny adres bufora w pamięci rozszerzonej
LC0_CTC_NOT_PROGRAMMED	-37	zlecono zapis wartości licznika lecz nie zaprogramowano trybu pracy kanału
LC0_REJECTED	-38	za dużo równoczesnych odwołań do sterownika

1) Nie występuje dla modułów serii LC-011 i LC-020.

Dodatkowe informacje o błędach (LC0_ERR_STAT):

nazwa	kod	znaczenie
LC0_E_OK	0	brak dodatkowych informacji
błędy w warunkach startu/stopu		
LC0_E_NO_MODULE	-1	nie ma takiego modułu
LC0_E_NONEX_DEV	-2	nie istnieje urządzenie o tym numerze
LC0_E_BAD_CHAN	-3	numer nieistniejącego kanału
LC0_E_BAD_TIME	-4	zły odcinek czasu
LC0_E_BAD_DATE	-5	zła specyfikacja daty
LC0_E_BAD_THRE	-6	błędny próg wyzwalania analogowego
moment przerwania przez funkcję BREAK:		
LC0_E_BROKEN_WAIT	-7	funkcja przerywana w trakcie oczekiwania na spełnienie warunku startu
LC0_E_BROKEN_RUN	-8	funkcja przerywana w trakcie przetwarzania

Ostrzeżenia (LC0_STATUS):

nazwa	kod	znaczenie
LC0_NON_EX_MOD	1	zażądano inicjalizacji nie istniejących modułów ale co najmniej 1 moduł został zainicjalizowany
LC0_OTHER_LEN	2	przepisano mniej próbek niż żądano
LC0_PREMATURE_END	3	przedwczesne zakończenie operacji z powodu przepełnienia / opróżnienia całego bufora
LC0_IN_PROGRESS	4	badana transmisja jeszcze trwa

6. Projektowanie programów użytkowych.

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

Wykonanie dowolnej z funkcji sterownik 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 sterownika; numery przerwania dla poszczególnych sterowników podane zostały w rozdziale 3.2.

Dobrze napisany program powinien składać się z następujących części:

- część wstępna:
 - stwierdzenie, czy sterownik jest zainstalowany; metodą wykrycia obecności sterownika w pamięci systemu w typowy sposób: poprzez przeglądanie zainstalowanych urządzeń przez wywołanie przerwania $2F^{16}$ i poszukiwanie urządzenia o nazwie określonej nazwie (patrz p.3.2.; powodzenie tej próby świadczy o zainstalowaniu sterownika, niepowodzenie - o jego braku)
 - rozpoznanie konfiguracji modułu (GET_TOTAL_CONFIGURATION - ile i jakich modułów jest zainstalowanych, GET_MODULE_CONFIGURATION - czy moduł ma zainstalowany przetwornik a/c, 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 (LC0_STATUS, LC0_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 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 sterownika.
- UWAGA: Jeżeli w swoim programie użytkowym przechwytyjemy przerwanie zegarowe 1C16 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 sterownikiem 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 sterownikiem oraz plik TEST.C zawierający przykłady wykorzystania poszczególnych funkcji sterownika.

6.2. Programowanie w języku Pascal.

Z poziomu języka Pascal komunikacja z sterownikiem 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 sterownikiem oraz plik TEST.PAS zawierający przykłady wykorzystania poszczególnych funkcji sterownika.

6.3. Programowanie w języku assemblera.

Producent dostarcza pliki źródłowe AMBEX-LC.ASM zawierający definicje wszystkich struktur danych i stałych potrzebnych do współpracy z driverem.