Program komunikacyjny wagi samoobsługowej OPTIMUS-iC Dibal K-235/255

DibalDrv Wersja 1.04 dla DOS

INSTRUKCJA OBSŁUGI

Spis treści

1 WPROWADZENIE

1.1 Informacje ogólne

Program komunikacyjny DibalDrv w wersji 1.04 do wag elektronicznych Dibal służy do programowania i odbierania danych wag Dibal z serii K (modele K235 i K255). Jest to program pracujący w środowisku DOS i może być także wykorzystywany w środowisku Windows 3.11/95/98/NT/2000/XP. Sterowanie programem odbywa się poprzez parametry wywołania (command line) a dane przekazywane są w plikach tekstowych.

Program umożliwia wykorzystanie wszystkich funkcji protokołu transmisji wagi. Uwalnia on użytkownika od zajmowania się fizycznym protokołem transmisji pozwalając skupić się jedynie na przesyłanych danych. Umożliwia więc w szybki sposób na wprowadzenie obsługi wag Dibal do istniejących lub powstających programów.

1.2 Obsługa wagi K235 we współpracy ze sterownikiem DIBALDRV.EXE

Sterownik DIBALDRV.EXE powinien działać na ustawieniach domyślnych i nie jest wymagane wprowadzanie żadnych opcji (za wyjątkiem numeru portu, który należy określić).
Programowanie wagi musi być poprzedzone wcześniejszym jej zablokowaniem. Blokowanie wagi odbywa się przez wciśnięcie sekwencji klawiszy: *** F 0 7 9 0 –**
Po zablokowaniu wagi można przesyłać dane.
Sterownik DIBALDRV.EXE „obudowuje" dane w odpowiednie ramki, aby transmisja była prawidłowa. Dane muszą być przygotowane wg opisu zawartego w pliku Regcom K series.doc i muszą być umieszczone w pliku wejściowym data_in.txt . Konieczne jest również, aby w pliku wejściowym *data_in.txt* na początku był umieszczony rejestr inicjujący transmisję a na końcu rejestr kończący transmisję. Wygląda to następująco:

KB;00;02;01;	rejestr inicjujący
28;00;1;N;4;OPTIMUS IC;	przykład rejestru z daną (wprowadzenie nagłówka)
dane…	
dane…	
……	
……	
dane…	
KB;00;02;03;	rejestr kończący transmisję

Plik wykonywalny Readall.bat służy do odbierania (wszystkich) danych z wagi (waga podłączona do portu pierwszego (COM1) i umieszcza dane w pliku *Data_out.txt.*
Kabel transmisyjny musi być typu: BK5478

1.3 Cechy programu

Oto główne cechy programu:

1. Pracuje w środowisku DOS oraz Windows 3.11/95/98/NT/2000/XP
2. Sterowanie poprzez parametry wywołania.
3. Parametry wywołania mogą być podawane w dowolnej kolejności
4. Nie wszystkie parametry wywołania muszą być podawane – zostaną użyte wartości domyślne
5. Umożliwia komunikację poprzez porty szeregowe COM1 i COM2, jak również przez porty o niestandardowych adresach I/O przy wszystkich obsługiwanych przez wagę prędkościach transmisji
6. Dane przekazywane w plikach tekstowych
7. Elastyczny format plików danych tekstowych – separowany (skrócony), sztywny

(pozycjonowany).

8. Formaty plików wejściowych i wyjściowych są niezależne
9. Rozkład danych w pliku tekstowym identyczny z rozkładem pól w fizycznych pakietach przesyłanych do wagi. Umożliwia to łatwiejsze przejście do zaimplementowania własnej obsługi komunikacji z wagą.
10. Wszystkie fizyczne aspekty transmisji z wagami Dibal są obsługiwane wewnętrznie
11. Możliwość przekodowania polskich znaków na format wagi i odwrotnie

1.4 Struktura programu

Program składa się z pliku wykonywalnego DIBALDRV.EXE, pliku danych wejściowych (domyślnie data_in.txt), pliku danych wyjściowych (domyślnie data_out.txt) i pliku błędów (domyślnie error.txt). Nazwy i lokacje plików danych i błędów mogą być dowolnie zmodyfikowane.

1.5 Lista parametrów

Do sterowania programem używamy odpowiednich parametrów wywołania. Każdy parametr podajemy w postaci IP=W, gdzie IP oznacza dwuliterowy identyfikator parametru a W znak lub ciąg znaków reprezentujący wartość danego parametru. Ten format zapisu oraz fakt, że każdy parametr posiada wartość domyślną, umożliwia elastyczne podawanie tylko potrzebnych parametrów w dowolnej kolejności.

Oto lista wszystkich obsługiwanych parametrów, zawierająca ich identyfikatory, pełne nazwy, opis, dozwolone i domyślne wartości:

ID	Nazwa	Funkcja	Dozwolone wartości	Domyślna wartość
CP	Com Port – identyfikator portu szeregowego	Określa numer portu szeregowego, do którego portu jest przyłączona waga	1, 2	1
CS	Com Speed – prędkość transmisji	Określa z jaką prędkością ma odbywać się transmisja	2400, 4800, 9600, 19200	9600
CM	Com Timeout – czas oczekiwania na dane transmisji	Określa ile czasu program ma czekać na odpowiedź kasy, podany w milisekundach	3000-20000 (3-20 sekund)	5000
CB	Com Bits	Określa liczbę bitów na znak	5-8	8
CT	Com Stop Bits	Określa, ile bitów stopu występuje w transmitowanej danej	1-2	1
CY	Com Parity	Określa sposób kontroli parzystości w transmisji danych	**N** - (none) brak kontroli, **E** - (even) parzyste, **O** - (odd) nieparzyste	N (K-235) E (K-255)

CA	Com Base Port – adres bazowy portu IO interfejsu szeregowego	Określa adres bazowy portu IO sterującego interfejsem szeregowym. Umożliwia sterowanie portem szeregowym umieszczonym na dodatkowej karcie rozszerzeń lub innym niestandardowym. Musi być podawany łącznie z parametrem CI	Poprawna wartość adresu portu IO podana heksadecymalnie (cztery cyfry w zakresie 0000-ffff), zgodna z dokumentacją i konfiguracją karty rozszerzeń. Uwaga!!! Nieprawidłowa wartość może doprowadzić do zawieszenia systemu i innych poważnych nieprawidłowości.	ffffh – parametr nieużywany, port wybierany przez parametr CP
CI	Com IRQ number – numer przerwania zgłaszanego przez interfejs szeregowy	Określa numer przerwania zgłaszanego przez interfejs szeregowy. Umożliwia sterowanie portem szeregowym umieszczonym na dodatkowej karcie rozszerzeń lub innym niestandardowym. Musi być podawany łącznie z parametrem CP	Poprawna wartość numeru przerwania interfejsue, zgodna z dokumentacją i konfiguracją karty rozszerzeń. Uwaga!!! Nieprawidłowa wartość może doprowadzić do zawieszenia systemu i innych poważnych nieprawidłowości.	-1 – parametr nieużywany, przerwanie wybierane jest automatycznie na podstawie parametru CP
CF	Com FIFO buffers – bufory FIFO portu szeregowego	Określa czy bufory FIFO interfejsu szeregowego są aktywne czy nie. Wskazane jest załączanie FIFO w systemie wielozadaniowym np. Windows. Nie wszystkie interfejsy szeregowe posiadają bufory FIFO.	0 – FIFO wyłączone, 1 – FIFO włączone.	0
CX	Com Multiplexer Channel – kanał multipleksera interfejsu szeregowego	Określa kanał multipleksera interfejsu szeregowego w systemach używających takiego urządzenia. Obecnie nie musi on być stosowany łącznie z parametrem XN	-1, 0 – n (n –ilość kanałów multipleksera) -1 – multiplekser nieużywany	-1
XN	Multiplexer Name – Nazwa typu multipleksera	Określa nazwę typu multipleksera. Możliwe wartości to FALWI (multiplekser Falwi) lub ELZAB (multiplekser Elzab). Jeżeli nie podano, multiplekser nie jest obsługiwany (połączenie bezpośrednie wagi z komputerem)	Nazwa typu multipleksera	<brak>
PS	Progress Window – okno postępu komunikacji	Powoduje wyświetlaenie podczas działania programu w lewym górnym rogu informacji o postępie komunikacji z kasą – podaje ilości wysłanych i odebranych pakietów	0 – nie pokazuj okna, 1 – pokazuj okno	0
IF	Input File – plik danych wejściowych	Określa nazwę pliku tekstowego w którym znajdują się dane, które mają być przesłane do wagi	Dowolna poprawna nazwa pliku wraz z rozszerzeniem	Data_in.txt
OF	Output File – plik danych wejściowych	Określa nazwę pliku tekstowego w którym zostaną zapisane dane, wysłane przez wagę	Dowolna poprawna nazwa pliku wraz z rozszerzeniem	Data_out.txt

EF	Error File – plik błędów	Określa nazwę pliku tekstowego w którym zostaną zapisane dane o zaistniałych błędach np. transmisji	Dowolna poprawna nazwa pliku wraz z rozszerzeniem	Error.txt
DP	Data Path – ścieżka dostępu danych	Określa ścieżkę dostępu do plików wejściowego, wyjściowego i błędów	Dowolna poprawna ścieżka dostępu	(żadna - pusty ciąg)
SL	Start Line – początkowa linia danych wejściowych	Określa numer linii w pliku danych tekstowych , która ma być wysłana jako pierwsza. Poprzednie linie zostaną zignorowane. Pierwsza linie pliku ma numer 0.	Dowolna liczba większa bądź równa 0	0
DF	Data Format – format danych wejściowych	Format danych w pliku wejściowym: T: tekstowy, separowany. B: binarny, pozycjonowany	T, B	T
AF	Answer Format – format danych wyjściowych	Format danych w pliku wyjściowym: T: tekstowy, separowany. B: binarny, pozycjonowany	T, B	B
SC	Separator Character – znak separujący	Znak, który rozdziela pola w plikach danych w formacie tekstowym	Dowolny znak (najlepiej rzadko bądź wcale nieużywany w wartościach pól np. ‘;’, ‘\’, ‘^’	;
BO	Bracket Open – nawias otwierający	Określa znak, który będzie nawiasem otwierającym dla wartości pola zawierającej w sobie znak separatora	Dowolny znak (najlepiej rzadko bądź wcale nieużywany w wartościach pól np. ‘{’, ‘[’	{
BC	Bracket Close – nawias zamykający	Określa znak, który będzie nawiasem zamykającym dla wartości pola zawierającej w sobie znak separatora	Dowolny znak (najlepiej rzadko bądź wcale nieużywany w wartościach pól np. ‘}’, ‘]’	}
BM	Bracket Mode – tryb nawiasów	Określa tryb zawierania wartości w nawiasy: S: (smart – inteligentny) – tylko wartości pól zawierające separator będą ujęte w nawiasy A: (all – wszystko) wszystkie pola będą ujęte w nawiasy	S, A	S
NL	New Line – znak nowej linii	Określa znak, który zostanie zamieniony na znak sterujący “Nowa linia” wagi	Dowolny znak (najlepiej rzadko bądź wcale nieużywany w wartościach pól np. ‘\’, ‘|’	\
DT	Device Type – rodzaj wagi	Określa typ wagi, z którą wykonywana jest komunikacja	1 – waga K-235 2 – waga K-255	1
DN	Device Number – numer urządzenia	Numer wagi, z którą należy wykonać komunikację. Dotyczy tylko modelu K-255, dla modelu K-235 parametr ten jest ignorowany	Cyfra z zakresu 1-255	0
PL	Polish Characters Conversion	strona kodowa polskich znaków w plikach danych	windows, mazovia, latin2, microvex, cyfromat, dhn, iso, csk, polonica, iea, inkaust, ventura	<brak> (konwersja nie jest wykonywana)

1.6 Wybór typu wagi

Wyboru typu wagi dokonuje się przy pomocy opcji **DT**. Podanie parametru **DT=1** spowoduje, że driver będzie używał protokołu komunikacyjnego wagi K-235, natomiast podanie **DT=2** wybierze protokół wagi K-255. Domyślnie (bez podania żadnego parametru) driver używa protokołu wagi K=235.

1.7 Parametry transmisji

Do kontroli transmisji używamy parametrów CP, CS i RT. Transmisja może odbywać się poprzez port szeregowy 1 lub 2. Standardowo wybrany jest port 1. Jeśli waga jest przyłączona do portu COM2 to należy w liście parametrów podać sekwencję CP=2.
Oto przykładowe wywołanie programu ustalające parametry transmisji na COM2, 19200,

```
DS500DRV.EXE CP=2 CS=19200 RT=0
```

Jak już wspomniano kolejność parametrów i wielkość liter jest nieistotna.

Do obsługi niestandardowych portów szeregowych używa się parametrów CA i CI. Należy ich używać ze szczególną ostrożnością gdyż nieprawidłowe wartości mogą doprowadzić do zawieszenia systemu, a nawet jego awarii. Jeśli oba te parametry są ustawione to parametr CP jest nieważny i port jest konfigurowany na podstawie właśnie CA i CI – adresu bazowego portu i numeru przerwania IRQ. Dodatkowo w tej wersji mamy możliwość złączenia buforów FIFO ([parametr CF) co może okazać się szczególnie przydatne w obciążonych systemach wielozadaniowych. Mamy także możliwość wyboru kanału multipleksera FALWI jeśli jest on używany. W tym celu używamy parametru CX. Oto przykład skonfigurowania portu szeregowego na COM2 ale definiowany parametrami CA i CI, przy prędkości 19200, z załączonymi buforami FIFO i z transmisją poprzez kanał 2 multipleksera FALWI:

```
DS500DRV.EXE CA=02f8 CI=3 CF=1 CS=19200 CX=2
```

1.8 Nazwy i położenie plików danych

Dane wysyłane i odbierane są zawarte w plikach tekstowych. Standardowo dane wejściowe są pobierane z pliku data_in.txt, dane wyjściowe (odesłane przez wagę) są zapisywane w pliku data_out.txt a informacje o zaistniałych błędach w pliku error.txt. Domyślnie pliki te są pobierane z bieżącego katalogu – program nie zmienia bieżącego katalogu. Każda nazwa pliku danych może być dowolnie zmieniona. Używamy do tego celu parametrów IF, OF, EF kolejno dla plików wejściowego, wyjściowego i błędów. Wprowadzona nazwa pliku może zawierać w sobie ścieżkę dostępu np. możemy ustalić plik wejściowy na C:\DIBALDRV\DANE_PLU.txt podając w liście parametrów IF=C:\DIBALDRV\DANE_PLU.TXT. Dodatkowym parametrem wpływającym na określanie przez program plików danych jest parametr DP. Określa on ścieżkę dostępu wspólną dla wszystkich plików danych. Standardowo jest to pusty ciąg znaków, który nie wpływa na nazwy (położenie) plików danych. Jeśli jednak zostanie temu parametrowi przypisana wartość to zostanie ona “doklejona” do nazw plików danych. Ścieżkę podajemy bez ostatniego znaku ‘\’. Jeśli np. wszystkie pliki mają standardową nazwę ale znajdują się w katalogu C:\ DIBALDRV\DATA (różnym od bieżącego) to należy wywołać program w sposób:

```
DIBALDRV.EXE DP=C:\DIBALDRV\DATA
```

1.9 Formaty plików danych - wejściowego i wyjściowego

Dodatkowymi parametrami dotyczącymi plików danych są parametry określające format danych w pliku. Ogólnie każda sekwencja danych (tzw. pakiet – patrz instrukcja protokołu

komunikacji z wagą) zajmuje jedną linię.
Każdy z pakietów rozpoczyna się trzycyfrowym kodem operacji do wykonania, po którym następują specyficzne dla danej operacji pola. Program drivera obudowuje podane dane znakami wymaganymi przez protokół komunikacji, oblicza sumę kontrolną itp., lecz dzieje się to "w tle", nie wymaga więc żadnej znajomości protokołu komunikacji.
Przykładowy pakiet programowania PLU wyglądać może następująco:

```
X1;00;M;000014;KIEŁBASA;000782;000000000000;0;000000;000000;00000;01
;000000;0000;00;0;0;0;00;00000;
```

Jeśli taki plik został wysłany do wagi to zaprogramowany zostanie artykuł (X1) z grupy 00, będzie modyfikowany (M), o kodzie nr 14, nazwie KIEŁBASA, cenie 782zł itd.

Inny format zapisu danych w pliku jest określany jako binarny-pozycjonowany. Dane pakietów są tu tak samo zapisywane jak każdy pakiet w jednej linii. W odróżnieniu od poprzedniego pola danych pakietu nie są separowane a zamiast tego wymagane jest aby miały ściśle określone rozmiary. Rozmiary poszczególnych pól są zdefiniowane w instrukcji protokołu transmisji wagi. Ten format zapisu jest ustalany poprzez parametr wywołania IF=B (dla pliku wejściowego) lub AF=B (dla pliku wyjściowego). Format B jest formatem domyślnym dla pliku danych wyjściowych.

1.10 Linia startowa pliku wejściowego

Użytkownik programu DIBALDRV ma możliwość określenia, która linia pliku wejściowego ma zostać wysłana jako pierwsza. Linie poprzedzające linię startową zostaną pominięte. Możliwość określenia linii startowej może być użyteczna przy kontynuowaniu przerwanej uprzednio transmisji. Jeśli np. w pliku wejściowym jest wiele pakietów i po wysłaniu części z nich nastąpi błąd to komunikacja zostanie przerwana. Jeśli usunięcie błędu nie wymaga ingerencji w plik wejściowy to program sterujący może wywołać DIBALDRV.EXE z parametrem SL=x, gdzie x określa linię startową, która może być odczytana z pliku error.txt zawierającego m.in. numer linii, w której wystąpił błąd. W ten sposób transmisja może być kontynuowana bez zmiany pliku wejściowego. Ten sposób jest użyteczny przy błędach fizycznego przerwania transmisji (np. wyłączenie wagi z sieci). Pierwsza linia pliku wejściowego ma numer 0.

1.11 Znak separatora

Jeśli formatem pliku danych jest format tekstowy-separowany to po każdym polu danych pakiety wstawiony jest znak separatora. Standardowo znakiem separatora jest ‘;’. Użytkownik ma możliwość zmiany znaku separatora. Np. aby ustalić znak separatora na ‘\’ należy wywołać

```
DIBALDRV.EXE SC=\
```

Znak separatora musi być dobrany bardzo uważnie. Powinien to być znak, który jak najrzadziej lub najlepiej wcale nie jest używany w wartościach parametrów. Kompletnym nieporozumieniem jest ustalenie separatora np. na ‘a’, które jest powszechnie używane np. w nazwach. Najlepiej nadają się do tego znaki: ‘;’, ‘\’, ‘|’, ‘^’, ‘~’ lub znaki graficzne o wysokich kodach ASCII. Jeśli jednak wartość pola zawiera znak separatora to powinna być ona ujęta w nawiasy co jest omówione w następnym rozdziale.

1.12 Nawiasy wartości pól pakietów danych

Jeśli wartość pola pakietu danych zawiera znak separatora (w tekstowym-separowanym formacie pliku danych) to powinna być ona ujęta w nawiasy, gdyż inaczej część tego pola zostałaby zinterpretowana jako wartość następnego pola. Np. Jeśli nazwa PLU jest “sałata 1;2” to jeśli byłaby ona wstawiona do pliku tekstowego bez nawiasów i separatorem byłby ‘;’ to 2 z nazwy potraktowane

by było jako następne pole w tym wypadku jako cena. Dla zapewnienia możliwości wprowadzania takich wartości i unikania w/w błędów DS500DRV posiada mechanizm nawiasów. Przykładowa wartość nazwy powinna być wstawiona do pliku w sposób "{sałata 1;2}". Znaki '{' i '}' są standardowymi znakami nawiasów. Program może zmienić te znaki za pomocą parametrów BO i BC. Aby np. ustalić nawiasy na '[' i ']' należy wywołać program w sposób:

```
DIBALDRV.EXE BO=[ BC=]
```

Ujmowanie w nawiasy wartości w pliku wejściowym leży po stronie programu sterującego. Jeśli program tego nie uczyni dane wejściowe mogą zostać źle zinterpretowane. Program sterujący może zaniechać wstawiania nawiasów jeśli jest pewne że separator nie wystąpi w wartościach pól (np. wybrano w ogóle nieużywany, bo np. niedozwolony w nazwach, znak separatora). Znaki nawiasów mogą być identyczne. Jeśli znak nawiasu ma być użyty w wartości to musi on być wstawiony dwukrotnie jeden po drugim. Jeśli np. nazwa PLU jest "Towar {1;}" to w pliku wejściowym powinno być "{Towar {{1;}}}.

Ujmowanie w nawiasy wartości w pliku wyjściowym leży oczywiście po stronie programu DIBALDRV.EXE. Program ten automatycznie ujmuje w nawiasy wartości, które zawierają w sobie znak separatora a format pliku wyjściowego jest ustalony na tekstowy-separowany. Pogram interpretujący dane pliku wyjściowego musi o tym pamiętać i odpowiednio interpretować znaki nawiasów. Program sterujący nie może zablokować wstawiania nawiasów, może jednak wybrać jeden z dwóch trybów wstawiania. W trybie S (smart-inteligentny) DIBALDRV obejmuje w nawiasy tylko wartości, które zawierają znak separatora. W trybie A (all-wszystko) – wszystkie wartości pól są ujęte w nawiasy. Standardowo używany jest tryb S. Do ustawiania trybu wstawiania nawiasów używamy parametru BM (Bracket mode – tryb nawiasów). Aby używać trybu A należy wywołać program w sposób:

```
DIBALDRV.EXE BM=A
```

Uwaga! Obejmowanie w nawiasy jest mechanizmem awaryjnym, mającym na celu zachowanie poprawności danych. Poprzez odpowiednie dobranie znaku separatora mechanizm pozostanie nieużywany i nie trzeba będzie się nim zajmować. Najlepiej aby program sterujący używał jako znak separatora znak niedozwolony w nazwach. Standardowe znaki separatora zostały wybrane te, które są łatwe do wprowadzenie przy ręcznym wprowadzaniu danych. Jeśli program DIBALDRV.EXE jest wywoływany przez inny program sterujący, można ustalić znak separatora na znak o kodzie ASCII np. 255, który może być zabroniony w nazwach edytowanych w tym programie sterującym. Wstawianie nawiasów i problem znaku separatora nie występuje oczywiście jeśli formatem pliku danych jest format binarny-pozycjonowany. Format binarny jest standardowym formatem pliku wyjściowego.

1.13 Znak nowej linii

Jak wiadomo kolejne pakiety danych kasy DIBALDRV są zapisane w kolejnych liniach plików danych. Znak nowej linii jest tu więc wykorzystany jako ogranicznik kolejnych pakietów. Są jednak przypadki, w których znak nowej linii może wystąpić w wartości jakiegoś pola. Dotyczy to pól tekstowych, zawierających opis, np. pole tekstu reklamowego (pakiet nr 020). Aby umożliwić zawarcie w tych treściach znaku nowej linii program DIBALDRV.EXE umożliwia zdefiniowanie jakiegoś znaku, który następnie zostanie (przy wysyłaniu) konwertowany na rzeczywisty znak nowej linii (tzn. znak DC3). Standardowo znakiem interpretowanym jako nowa linia jest '\'. Do definiowania znaku nowej linii służy parametr NL (new line – nowa linia). Znak nowej musi być dobrany bardzo uważnie. Powinien to być znak, który jak najrzadziej lub najlepiej wcale nie jest używany w wartościach parametrów. Aby np. zdefiniować znak '|' jako znak nowej linii należy wywołać program w sposób:

```
DIBALDRV.EXE NL=|
```

Uwaga! Znak nowej linii jest interpretowany tylko w polach, w których może być wykorzystany. Dotyczy to tylko pól opisu.

1.14 Podawanie parametrów wywołania.

Parametry wywołania podajemy kolejno po sobie oddzielone spacjami. Kolejność parametrów jak i wielkość liter jest nieistotna. Jeśli ustawienie jakiegoś parametru występuje dwukrotnie to drugie z kolei (ostatnie) ustawienie jest przyjęte. Wszystkie parametry mają wartości domyślne. Jeśli wartość danego parametru jest równa domyślnej to nie trzeba jej podawać.

Poniżej jest przedstawione przykładowe wywołanie programu z następującymi ustawieniami: port szeregowy COM2, prędkość transmisji 19200, plik danych wejściowych GetPLU.txt, wyjściowych PLUData.txt, znak separatora '|' i format pliku wyjściowego tekstowy:

DIBALDRV.EXE cp=2 cs=9600 if=GetPLU.txt of=PLUData.txt sc=| af=T

2 INFORMACJE O BŁĘDACH TRANSMISJI

2.1 Rezultat wykonania programu DIBALDRV.EXE

Program po wykonaniu zwraca do systemu następujące wartości:

0 – program wykonany bezbłędnie, plik błędów jest pusty, plik wyjściowy danych zawiera poprawne informacje
1 – wystąpiły błędy w inicjacji bądź transmisji, plik błędów zawiera informacje o błędach, plik wyjściowy danych zawiera poprawnie odebrane informacje.
2 – wystąpiły błędy wewnętrzne programu. Został on natychmiastowo zakończony, a zawartość pliku błędów i wyjściowego jest nieokreślona
3 – wystąpiły błędy wewnętrzne programu. Został on natychmiastowo przerwany, a zawartość pliku błędów i wyjściowego jest nieokreślona.

Rezultat 1 może wystąpić jeśli nie powiedzie się inicjacja programu (np. za mało pamięci, błędne nazwy plików itp.) lub transmisja nie przebiegnie poprawnie ze względu na niewłaściwe warunki lub dane.

Rezultat 2 i 3 występuje jeśli program nie może kontynuować swego działania ze względu np. na brak pamięci. Musi on być natychmiastowo przerwany bądź zakończony i nie może zapisać informacji o błędach w pliku błędów.

2.2 Zawartość pliku błędów.

Jeżeli program zwrócił rezultat 1 program sterujący powinien zinterpretować zawartość pliku błędów. Każdy zaistniały błąd zajmuje jedną linię tego pliku. Format tej linii jest następujący:

Kod błędu; Linia pliku wejściowego; Informacja tekstowa (w języku angielskim);

Kod błędu może przyjmować następujące wartości:

1 – Błąd inicjalizacji
2 – Błąd braku pamięci
3 – Błędne dane wejściowe

100 – Błąd transmisji

101 – Przekroczenie czasu oczekiwania
102 – Błąd sumy kontrolnej

Pole “Linia pliku wejściowego” oznacza numer linii podczas interpretacji której wystąpił błąd. Pierwsza linia ma numer 0.

UWAGA! Program DIBALDRV.EXE przerywa działanie po zaistnieniu jakiegokolwiek błędu.