



Dokumentacja techniczna biblioteki fiskalnej MFBO dla drukarek Elzab: Omega oraz Mera

Wersja: 1.1.0

UPOS System sp. z o.o.

44-190 Knurów,

ul. Sienkiewicza 13

tel.

(032) 33 88 200

fax

(032) 33 88 211

e-mail

firma@upos.com.pl

Internet

www.upos.com.pl

Historia zmian:

Data	Wersja	Autor	Komentarz
30.04.2008	1.0.0	Dariusz Pawelec	Opracowanie dokumentu
23.04.2009	1.1.0	Dariusz Pawelec	Dodano funkcje: <ul style="list-style-type: none">• mfParShiftReport,• mfBgShiftReport,• mfLineShiftReport,• mfEndShiftReport

Spis treści

1. Specyfikacja.....	5
2. Instalacja.....	5
3. Wyszczególnienie i opis zaimplementowanych funkcji.....	6
3.1 Funkcje informacyjne.....	6
3.1.1 Odczyt czasu zegara wewnętrznego drukarki fiskalnej	6
3.1.2 Odczyt daty zegara wewnętrznego drukarki fiskalnej.....	6
3.1.3 Odczyt wersji drukarki fiskalnej (wersji firmware'u).....	6
3.1.4 Współczynnik wolnych zasobów.....	6
3.1.5 Odczyt numeru ostatniego paragonu (faktury paragonowej).....	6
3.1.6 Odczyt stawek podatkowych.....	6
3.2 Funkcje informacyjne stałe.....	7
3.2.1 Odczyt maksymalnej liczby zmian w ciągu doby.....	7
3.2.2 Odczyt rodzaju biblioteki (niefiskalna / fiskalna).....	7
3.2.3 Odczyt wersji biblioteki.....	7
3.2.4 Odczyt treści komunikatu.....	7
3.2.5 Odczyt parametrów danych o towarze.....	7
3.2.6 Odczyt parametrów dotyczących formatu nagłówka na paragonach.....	7
3.2.7 Odczyt danych dotyczących parametrów wydruku paragonu.....	7
3.4 Funkcje inicjujące i definiujące.....	8
3.4.2 Definicja stopki paragonu	8
3.4.3 Definicja numeru kasy i kasjera oraz nazwy kasjera.....	8
3.4.4 Ustawienie czasu zegara wewnętrznego drukarki fiskalnej.....	8
3.4.5 Ustawienie stawek VAT.....	9
3.4.6 Ustawienie i odczyt definiowanych napisów.....	9
3.5 Funkcje obsługujące paragon fiskalny (fakturę paragonową).....	10
3.5.1 Definiowanie nagłówka.....	10
3.5.2 Rozpoczęcie wydruku paragonu.....	10
3.5.3 Wydruk linii paragonu.....	10
3.5.4 Unieważnienie linii paragonu.....	10
3.5.5 Unieważnienie paragonu.....	10
3.5.6 Zakończenie wydruku paragonu.....	10
3.5.7 Zamknięcie doby fiskalnej.....	10
3.5.8 Rozszerzony wydruk linii paragonu.....	10
3.5.9 Unieważnienie rozszerzonego wydruku linii paragonu.....	11
3.5.10 Zakończenie rozszerzonego wydruku paragonu.....	11
3.5.11 Odczyt danych dotyczących parametrów wydruku faktury paragonowej.....	11
3.5.12 Rozpoczęcie faktury paragonowej.....	11
3.5.13 Zakończenie faktury paragonowej.....	11
3.6 Funkcje obsługujące raporty fiskalne.....	12
3.6.1 Raport okresowy szczegółowy.....	12
3.6.2 Raport okresowy sumaryczny.....	12
3.6.3 Wydruk stanu modułu fiskalnego.....	12
3.6.4 Pobranie informacji o numerach rekordów dobowych.....	12
3.6.5 Pobranie rekordu dobowego.....	12
3.7 Funkcje obsługujące paragon нефiskalny.....	14

3.7.1 Wydruk kwitu wpłaty do kasy KP.....	14
3.7.2 Wydruk kwitu wypłaty z kasy KW.....	14
3.7.3 Wydruk kwitu (bonu) zamówienia w kuchni.....	14
3.7.4 Wydruk dokumentu zwrotu towarów.....	14
3.7.5 Wydruk raportu rozliczenia kasjera.....	15
3.7.6 Wydruk raportu rozliczenia zmiany.....	15
3.7.7 Unieważnienie dokumentu.....	16
3.7.8 Wydruk kwitu karty płatniczej.....	16
3.7.9 Rozszerzony wydruk kwitu transakcji kartą płatniczą.....	16
3.7.10 Wydruk kwitu transakcji eCash.....	18
3.7.11 Wydruk doładowania konta PREPAID.....	20
3.8 Funkcje obsługujące szufladę.....	22
3.8.1 Definicja parametrów (rodzaju) szuflady.....	22
3.8.2 Otwarcie szuflady.....	22
3.8.3 Sprawdzenie stanu szuflady (otwarta / zamknięta).....	22
3.9 Funkcje obsługujące wyświetlacz.....	23
3.9.1 Odczyt parametrów wyświetlacza zainstalowanego w DF.....	23
3.9.2 Wyświetlenie POZYCJI SPRZEDAŻY.....	23
3.9.3 Wyświetlenie SUMY.....	23
3.9.4 Wyświetlenie PODSUMOWANIA.....	23
3.9.5 Wyświetlenie LOGO UŻYTKOWNIKA.....	23
3.9.6 Graficzny wygaszacz ekranu (logo).....	23
3.10 Funkcje statusowe.....	24
3.10.1 Tryb szkoleniowy / fiskalny.....	24
3.10.2 Waluta przed denominacją / po denominacji.....	24
3.10.3 Doba fiskalna zamknięta / otwarta.....	24
3.10.4 Paragon zamknięty / otwarty.....	24
3.10.5 Dokument nie drukowany / drukowany.....	24
3.10.6 Brak papieru.....	24
3.11 Pozostałe funkcje.....	25
3.11.1 Bezpośrednia komunikacja z drukarką.....	25
3.11.2 Funkcja umożliwia odczyt kodu błędu.....	25
3.11.3 Funkcja umożliwia odczyt numeru seryjnego drukarki.....	25
3.11.4 Odczyt numeru kompilacji wersji biblioteki.....	25
3.11.5 Funkcja zwraca czy dla danej drukarki jest konieczne obsługiwane EJ.....	25
3.11.6 Ustawianie poziomu logowania.....	25
3.11.7 Zwraca max liczbę towarów które można zwrócić w jednym zwrocie.....	25
4 Indeks funkcji MFBO.....	26

1. Specyfikacja

Biblioteka obsługuje drukarki fiskalne FP-600, Omega oraz Mera firmy Elzab, współpracuje z programem EuroPOS na systemie MS Windows oraz DOS, oraz każdym innym programem opartym na interfejsie MFBO. Podczas pisania biblioteki oparto się na dokumentacji UPOS: „MF2 – Uniwersalna biblioteka funkcji do obsługi modułów fiskalnych dla drukarek fiskalnych UPOS wersja 1.3.0”.

Zakres zaimplementowanych funkcji znajduje się w dalszej części dokumentacji.

2. Instalacja

Instalacja biblioteki polega na skopiowaniu jej do katalogu EuroPOS’a oraz ustawienie wartości parametru MFBODLL z pliku EUROPOS2.INI na nazwę biblioteki. Następnie należy ustawić parametr BillPort z sekcji BILL_PRINTER na nazwę portu do którego podłączona jest drukarka. Biblioteka ignoruje parametry portu szeregowego zdefiniowane w pliku EUROPOS2.INI i ustawia je samodzielnie na następujące:

- szybkość 9600 b/s
- parzystość parzysta (even)
- liczba bitów 8
- liczba bitów stopu 1

3. Wyszczególnienie i opis zaimplementowanych funkcji

W rozdziale znajduje się wyszczególnienie zaimplementowanych funkcji. Opisywania biblioteka zachowuje sposób implementacji dotychczasowej biblioteki dla drukarek FP-600 i Omega. W przypadku jeśli implementacja nowych funkcji odbiega od standardu MFBO, albo działanie wymaga dodatkowego opisu, dodany jest odpowiedni komentarz z opisem zmian, dodatkowych wymagań itp.

3.1 Funkcje informacyjne

3.1.1 Odczyt czasu zegara wewnętrznego drukarki fiskalnej

int mfGetTime (int *hour, int *minute)

3.1.2 Odczyt daty zegara wewnętrznego drukarki fiskalnej

int mfGetDate (int *year, int *month, int *day)

3.1.3 Odczyt wersji drukarki fiskalnej (wersji firmware'u)

int mfGetVersion (int *version, int *subVersion)

int mfGetFirmwareVersion(int *insideNumber, int *compilationNumber) – niezaimplementowane (zwraca mfeIMPLEMENTATION)

3.1.4 Współczynnik wolnych zasobów

int mfGetCoefficient (int *recCoefficient, int *unRecCoefficient) - niezaimplementowane (zwraca mfeIMPLEMENTATION)

Odczyt liczby wolnych rekordów do zapisu:

a) SFI określająca czy skojarzona FP jest dostępna w danej DF

int mfParGetFreeRec(void) - niezaimplementowane (zwraca mfeASSOCIATION)

b) FP odczytująca liczbę wolnych rekordów do zapisu

int mfGetFreeRec(int *freeRec) - niezaimplementowane (zwraca mfeIMPLEMENTATION)

3.1.5 Odczyt numeru ostatniego paragonu (faktury paragonowej)

int mfGetLastTicketNo (long *lastTicketNo)

int mfGetLastInvoiceNo (long *lastInvoiceNo) - niezaimplementowane (zwraca mfeIMPLEMENTATION)

3.1.6 Odczyt stawek podatkowych

int mfGetTax (int *taxA, int *taxB, int *taxC, int *taxD, int *taxE, int *taxF)

Odczyt stawek podatkowych rekordu:

a) SFI określająca czy skojarzona FP jest dostępna w danej DF

int mfParGetTaxRec(void) - niezaimplementowane (zwraca mfeASSOCIATION)

b) FP odczytująca stawki podatkowe rekordu

int mfGetTaxRec(int recNo, int *recStatus, int *taxA, int *taxB, int *taxC, int *taxD, int *taxE, int *taxF) - niezaimplementowane (zwraca mfeIMPLEMENTATION)

3.2 Funkcje informacyjne stałe

3.2.1 Odczyt maksymalnej liczby zmian w ciągu doby

int mfParMaxShiftNo (int *maxShiftNo)

3.2.2 Odczyt rodzaju biblioteki (niefiskalna / fiskalna)

int mfParFiscal (int *flag)

3.2.3 Odczyt wersji biblioteki

int mfParLibVersion (int *version, int *subVersion, char *name)

3.2.4 Odczyt treści komunikatu

int mfParMessage (int code, char *groupMsg, char *detailMsg)

3.2.5 Odczyt parametrów danych o towarze

int mfParPLU (long *maxCode, int *length, char *setOfChar)

3.2.6 Odczyt parametrów dotyczących formatu nagłówka na paragonach

int mfParHeader (int *flag, int *maxLines, int *length, char *setOfChar)

Definiowanie nagłówka przez bibliotekę nie jest możliwe (tylko przez program serwisowy Elzab)

3.2.7 Odczyt danych dotyczących parametrów wydruku paragonu

int mfParTicket (int *discountDecimalPoint, int *flag)

3.4 Funkcje inicjujące i definiujące

3.4.1 Dopuszczenie aplikacji

```
int mfStart ( const char *ident, int mfPortNo )
int mfStart2( const char *ident,
               int mfPortType,
               int mfPortNo,
               int column,
               int emptyLine,
               const char *endLine,
               const char *cutPaper,
               int taxA, int taxB, int taxC, int taxD, int taxE, int taxF,
               int sBauds, int sParity, int sHandshake,
               int codePage,
               const char *initString )
int mfStop( void )
```

Wszystkie parametry oprócz *mfPortNo* są ignorowane.

3.4.2 Definicja stopki paragonu

a) SFI zwracająca dane dotyczące formatu stopki drukowanej na paragonach

```
int mfParFooter ( int *maxLines, int *length, char *setOfChar )
```

b) FP definiująca stopkę

```
int mfDefFooter ( const char *footer )
```

Definiowanie stopki jest możliwe tylko dla drukarki Mera, tylko jeden raz w ciągu doby fiskalnej. EuroPOS podczas każdego uruchomienia próbuje definiować stopkę, więc jeśli stopka będzie zmieniana więcej niż jeden raz to podczas kolejnych uruchomień będzie zgłaszał błąd "Inny błąd".

3.4.3 Definicja numeru kasy i kasjera oraz nazwy kasjera

a) SFI zwracająca dopuszczalny zakres parametrów numeracji kasy i kasjera oraz nazwy kasjera

```
int mfParPOS ( int *maxCashNo, int *maxCashierNo, int *lengthName, char *setOfChar )
```

b) FP definiująca numer kasy i kasjera oraz nazwę kasjera

```
int mfDefPOS ( int cashNo, int cashierNo, const char *cashierName )
```

3.4.4 Ustawienie czasu zegara wewnętrznego drukarki fiskalnej

a) SFI sprawdzająca czy ustawienie czasu w zegarze jest wykonalne (w danej chwili)

```
int mfParTime ( void )
```

b) FP ustawiająca czas zegara wewnętrznego DF

```
int mfDefTime ( int hour, int minute )
```


Ustawienie czasu możliwe tylko jeden raz na dobę, po raporcie dobowym fiskalnym, kiedy licznik paragonów (mfGetLastTicketNo) jest równy '0' (zero), o maksymalnie godzinę do przodu lub do tyłu.

3.4.5 Ustawienie stawek VAT

a) SFI sprawdzająca czy ustawienie stawek VAT jest wykonalne (w danej chwili)

int mfParTax (void)

b) FP ustawiająca stawki VAT

int mfDefTax (int taxA, int taxB, int taxC, int taxD, int taxE, int taxF

3.4.6 Ustawienie i odczyt definiowanych napisów

a) SFI odczytująca napis i jego maksymalną długość

int mfParString(int stringID, char *string, int *maxLen)

b) SFI redefiniująca napis

int mfDefString(int stringID, const char *string)

3.5 Funkcje obsługujące paragon fiskalny (fakturę paragonową)

3.5.1 Definiowanie nagłówka

int mfBeginShift (const char *header) - funkcja niezaimplementowana (zwraca mfeIMPLEMENTATION)

3.5.2 Rozpoczęcie wydruku paragonu

int mfBeginTicket (long total)

3.5.3 Wydruk linii paragonu

int mfLineTicket (long no, const char *name, char taxCode,
const char *unitStr, long price, long quantity, long value)

3.5.4 Unieważnienie linii paragonu

int mfStornoLineTicket (long no, const char *name, char taxCode,
const char *unitStr, long price, long quantity, long value)

3.5.5 Unieważnienie paragonu

int mfStornoTicket (void)

3.5.6 Zakończenie wydruku paragonu

int mfEndTicket (int percentDiscount, long valueDiscount, long transactionNo,
long paym1, const char *paymStr1,
long paym2, const char *paymStr2,
long paym3, const char *paymStr3,
long change, const char *changeStr,
long pawn, const char *pawnStr,
const char *tableStr,
const char *custStr)

Parametr valueDiscount jest ignorowany.

3.5.7 Zamknięcie doby fiskalnej

int mfShiftEnd (void)

3.5.8 Rozszerzony wydruk linii paragonu

a) SFI sprawdzająca czy DF udostępnia funkcję rozszerzonego wydruku linii paragonu

int mfParLineTicketExt (void) - zwraca :

- mfeOK dla drukarki Mera
- mfeASSOCIATION dla innych

b) FP realizująca rozszerzonego wydruk linii paragonu

int mfLineTicketExt (long no, const char *name, char taxCode,
const char *unitStr, long price, long quantity, long value) - tylko Mera, dla
innych drukarek zwraca mfeIMPLEMENTATION

Wydruk rabatu pozycyjnego możliwy jest tylko do ostatniej pozycji sprzedaży. Biblioteka sprawdza

czy nazwa i kod podatku są takie same, oraz czy wartość jest mniejsza niż dla ostatniej pozycji wysłanej przez mfLineTicket. Jeśli te warunki są spełnione to wysyłany jest rozkaz do drukarki, w przeciwnym przypadku zwracane jest mfeIMPLEMENTATION.

3.5.9 Unieważnienie rozszerzonego wydruku linii paragonu

a) SFI sprawdzająca czy DF udostępnia funkcję unieważnienia rozszerzonego wydruku linii paragonu

int mfParStornoLineTicketExt (void) - zwraca mfeASSOCIATION

b) FP realizująca unieważnienie rozszerzonego wydruku linii paragonu

int mfStornoLineTicketExt (long no, const char *name, char taxCode,
const char *unitStr, long price, long quantity, long value) -
niezaimplementowane (zwraca mfeIMPLEMENTATION)

3.5.10 Zakończenie rozszerzonego wydruku paragonu

int mfEndTicketExt (int percentDiscount, long valueDiscount,
long transactionNo,
long paym1, const char *paymStr1,
long paym2, const char *paymStr2,
long paym3, const char *paymStr3,
long change, const char *changeStr,
long pawn, const char *pawnStr,
const char *tableStr,
const char *custStr,
int loyaltyNo, const char * loyaltyParamStr,
int reserveNo, const char * reserveParamStr) - niezaimplementowane (zwraca
mfeIMPLEMENTATION)

3.5.11 Odczyt danych dotyczących parametrów wydruku faktury paragonowej

int mfParInvoice (int *maxItems,
int *custNameLineLen,
int *custNameNrOfLines,
int *custNIPLen) - zwraca mfeASSOCIATION

3.5.12 Rozpoczęcie faktury paragonowej

int mfBeginInvoice (const char * number) - niezaimplementowane (zwraca
mfeIMPLEMENTATION)

3.5.13 Zakończenie faktury paragonowej

int mfEndInvoice (int percentDiscount,
long valueDiscount,
long transactionNo,
long paym1, const char * paymStr1,
long paym2, const char * paymStr2,
long paym3, const char * paymStr3,

```
long change, const char * changeStr,  
long pawn, const char * pawnStr,  
const char * tableStr,  
const char * custStr,  
const char * custNIP, const char * custName ) - niezaimplementowane (zwraca  
mfeIMPLEMENTATION)
```

3.6 Funkcje obsługujące raporty fiskalne

3.6.1 Raport okresowy szczegółowy

```
int mfPrDetailReport ( int startYear, startMonth, startDay,  
int stopYear, stopMonth, stopDay )
```

3.6.2 Raport okresowy sumaryczny

```
int mfPrGlobalReport ( int startYear, startMonth, startDay,  
int stopYear, stopMonth, stopDay )
```

3.6.3 Wydruk stanu modułu fiskalnego

a) SFI sprawdzająca czy FP jest wykonalna

```
int mfParPrState ( void )
```

b) FP drukująca stan modułu fiskalnego

```
int mfPrState( void )
```

3.6.4 Pobranie informacji o numerach rekordów dobowych

a) SFI sprawdzająca czy FP jest wykonalna

```
int mfParGetFiscalRecRange ( void ) - zwraca mfeASSOCIATION
```

b) FP pobierająca informacje o numerach rekordów dobowych

```
int mfGetFiscalRecRange( int startYear, int startMonth, int startDay,  
int startHour, int startMinute,  
int stopYear, int stopMonth, int stopDay,  
int stopHour, int stopMinute,  
int *recFrom, int *recTo ) - niezaimplementowane (zwraca mfeIMPLEMENTATION)
```

3.6.5 Pobranie rekordu dobowego

a) SFI sprawdzająca czy FP jest wykonalna

```
int mfParGetFiscalRecRange ( void ) - zwraca mfeASSOCIATION
```

b) FP pobierająca rekord dobowy

```
int mfGetFiscalRec( int recNo,  
int *recStatus,  
int *reportNo,  
int *closeYear,  
int *closeMonth,  
int *closeDay,  
int *closeHour,
```

```
int *closeMinute,  
int *taxRecId,  
long *totalA,  
long *totalB,  
long *totalC,  
long *totalD,  
long *totalE,  
long *totalF,  
long *totalG,  
long *sumTax,  
int *nrOfTickets,  
long *sumCancelled,  
long *nrOfCancelled ) - niezaimplementowane (zwraca mfeIMPLEMENTATION)
```

3.7 Funkcje obsługujące paragon niefiskalny

3.7.1 Wydruk kwitu wpłaty do kasy KP

a) SFI określająca czy skojarzona FP jest dostępna w danej DF

int mfParPayIn (void)

b) FP drukująca dokument

int mfPrPayIn (const char *no, const char *personName, const char *payInName,
const char *amount)

3.7.2 Wydruk kwitu wypłaty z kasy KW

a) SFI określająca czy skojarzona FP jest dostępna w danej DF

int mfParPayOut (void)

b) FP drukująca dokument

int mfPrPayOut (const char *no, const char *personName, const char *payOutName,
const char *amount)

3.7.3 Wydruk kwitu (bonu) zamówienia w kuchni

a) SFI określająca czy skojarzona FP jest dostępna w danej DF

int mfParOrder (void)

b) FP rozpoczynająca wydruk

int mfBgOrder (const char *title, const char *name)

c) FP drukująca jedną linię dokumentu

int mfLineOrder (const char *name, const char *modifier, const char *quant)

d) FP kończąca wydruk

int mfEndOrder (void)

3.7.4 Wydruk dokumentu zwrotu towarów

a) SFI określająca czy skojarzona FP jest dostępna w danej DF

int mfParReturn (void)

b) FP rozpoczynająca wydruk

int mfBgReturn (const char *no, const char *personName, const char *value)

c) FP drukująca jedną linię dokumentu

int mfLineReturn (const char *name, const char *quant, const char *total)

int mfParLineReturnExt (void) - zwraca mfeASSOCIATION

int mfLineReturnExt (long no, const char *name, char taxCode,

const char *unitStr, long price, long quantity, long value) - niezaimplementowane (zwraca
mfeIMPLEMENTATION)

d) FP kończąca wydruk
 int mfEndReturn (void)

Możliwy jest zwrot tylko jednego towaru (jednej pozycji).

3.7.5 Wydruk raportu rozliczenia kasjera

a) SFI określająca czy skojarzona FP jest dostępna w danej DF
 int mfParCashierReport (void) - zwraca mfeASSOCIATION

b) FP rozpoczynająca wydruk
 int mfBgCashierReport (const char *no, const char *name) - niezaimplementowane (zwraca mfeIMPLEMENTATION)

c) FP drukująca jedną linię dokumentu
 int mfLineCashierReport (int typeNo, const char *name, const char *counted,
 const char *cash, const char *difference) - niezaimplementowane (zwraca mfeIMPLEMENTATION)

d) FP kończąca wydruk
 int mfEndCashierReport (void) - niezaimplementowane (zwraca mfeIMPLEMENTATION)

3.7.6 Wydruk raportu rozliczenia zmiany

a) SFI określająca czy skojarzona FP jest dostępna w danej DF
 int mfParShiftReport (void) - zwraca mfeOK

b) FP rozpoczynająca wydruk
 int mfBgShiftReport (const char *no,
 const char *startDate, const char *endDate, const char *personName)

Parametry:

1. no - numer zmiany (max 10 znaków).
2. startDate - data i godzina rozpoczęcia zmiany (max 16 znaków): do drukarki przekazywane jest pierwsze pięć znaków, a następnie od znaku jedenastego do końca, np. z łańcucha „17/04/2009 15:10” do drukarki zostanie przekazany łańcuch „17/04 15:10”
3. endDate - data i godzina zakończenia zmiany (max 16 znaków).
4. personName - nazwisko osoby drukującej raport (max 20 znaków).

c) FP drukująca jedną linię dokumentu
 int mfLineShiftReport (int typeNo, const char *name, const char *counted,
 const char *cash, const char *difference)

Funkcja wywoływana wielokrotnie - rozpoznaje i zapamiętuje następujące zestawienia parametrów:

- typeNo=4, name="Opakowania przyj." - wartość z pola 'cash' umieszczana jest w polu "12" formatki „33” drukarki,
- typeNo=4, name="Opakowania wydane" - wartość z pola 'cash' umieszczana jest w polu "03" formatki „33” drukarki, wartość z pola 'cash' odejmowana jest od wartości w polu „21” formatki „33” drukarki,
- typeNo=4, name="SPRZEDA” FISKALNA" - wartość z pola 'cash' dodawana jest do pola "21" formatki „33” drukarki,

–typeNo=2, name="Gotówka" – wartości z pól 'cash', 'counted', 'difference' umieszczane są odpowiednio w polach „44”, „54” oraz „74” formatki „33” drukarki z nazwą „Gotówka PLN” w polu „34” drukarki
 –maksymalnie 10 pozycji z typeNo=2 oraz 'name' różnym od "Gotówka" zostaje zapamiętanych oraz wydrukowanych w polach „34”, „44”, „54”, „74” o wartościach odpowiednio 'name', 'cash', 'counted', 'difference',
 –typeNo=3, name="WpłATA DO KASY" jeśli wystąpi po pozycji typeNo=2, name="Gotówka", oraz przed inną pozycją z typeNo=2, zostaje wydrukowana w polu „41” drukarki, oraz dodana do wartości pola „32”
 –typeNo=3, name="WYPłATA Z KASY" jeśli wystąpi po pozycji typeNo=2, name="Gotówka", oraz przed inną pozycją z typeNo=2, zostaje wydrukowana w polu „82” drukarki, oraz dodana do wartości pola „93”

d) FP kończąca wydruk

int mfEndShiftReport (void)

Dopiero wywołanie tej funkcji powoduje podliczenie i wysłanie na drukarkę wszystkich danych zebranych w funkcji mfBgShiftReport, oraz wielokrotnym wywołaniu funkcji mfLineShiftReport.

3.7.7 Unieważnienie dokumentu

int mfPrVoid (void)

3.7.8 Wydruk kwitu karty płatniczej

a) SFI określająca czy skojarzona FP jest dostępna w danej DF

int mfParVISA (void)

b) FP drukująca dokument

int mfPrVISA (int language,

Int signature,

const char *date,

const char *transactionNo,

const char *ticketNo,

const char *salesPointNo,

const char *TID,

const char *UID,

const char *PosID,

const char *cashierName,

const char *cardName,

const char *cardNumber,

const char *expiryDate,

const char *authorizationCode,

const char *salesAmount,

const char *returnAmount,

const char *salesSays1,

const char *salesSays234)

3.7.9 Rozszerzony wydruk kwitu transakcji kartą płatniczą

a) SFI określająca czy skojarzona FP jest dostępna w danej DF

int mfParEFT (void)

b) FP realizujące rozszerzony wydruk kwitu transakcji kartą płatniczą

Kwit autoryzacji transakcji:

```
int mfPrEFT_Transaction (const int i_bCopy,
const int i_eReceiptType,
const int i_eTransactionType,
const int i_eReceiptVariant,
const int i_eCardHolderVerification,
const char* i_szDateTimeTransaction,
const char* i_szCardAcceptorTerminalIdentification,
const char* i_szCardAcceptorIdentificationCodeMID,
const long i_nTransactionNumber,
const long i_nBatchNumber,
const long i_nSTAN,
const long i_nPOSNumber,
const long i_nPOSTransactionNumber,
const char* i_szCashierIdentification,
const char* i_szCardName,
const char* i_szCommercialCode,
const char i_cCardDataInputModeIndicator,
const char* i_szPrimaryAccountNo_Receipt,
const char* i_szDateExpiration,
const long i_nAmountTransaction,
const char i_cCardHolderVerificationIndicator,
const char* i_szApprovalCode,
const int i_eAuthorizationSource,
const int i_eAcquirerResponseReason,
const char* i_szMessage_Receipt,
const int i_nEnglishAgreement )
```

Kwit unieważnienia transakcji:

```
int mfPrEFT_Reversal (const int i_bCopy,
const int i_eReceiptType,
const int i_eTransactionType,
const int i_eReceiptVariant,
const int i_eCardHolderVerification,
const char* i_szDateTimeTransaction,
const char* i_szCardAcceptorTerminalIdentification,
const char* i_szCardAcceptorIdentificationCodeMID,
const long i_nTransactionNumber,
const long i_nBatchNumber,
const long i_nSTAN,
const long i_nPOSNumber,
const long i_nPOSTransactionNumber,
const char* i_szCashierIdentification,
const char* i_szCardName,
const char i_cCardDataInputModeIndicator,
const char* i_szPrimaryAccountNo_Receipt,
const char* i_szDateExpiration,
```

```
const long i_nAmountTransaction,
const char* i_szApprovalCode,
const int i_eAuthorizationSource,
const int i_eAcquirerResponseReason,
const char* i_szMessage_Receipt )
```

Kwit rejestracji zewnętrznej transakcji:

```
int mfPrEFT_RegisterOffline (const int i_bCopy,
const int i_eReceiptType,
const int i_eTransactionType,
const int i_eCardHolderVerification,
const char* i_szDateTimeTransaction,
const char* i_szCardAcceptorTerminalIdentification,
const char* i_szCardAcceptorIdentificationCodeMID,
const long i_nTransactionNumber,
const long i_nBatchNumber,
const long i_nPOSNumber,
const long i_nPOSTransactionNumber,
const char* i_szCashierIdentificator,
const char* i_szCardName,
const char i_cCardDataInputModeIndicator,
const char* i_szPrimaryAccountNo_Receipt,
const char* i_szDateExpiration,
const long i_nAmountTransaction,
const char* i_szApprovalCode,
const int i_eAuthorizationSource,
const char* i_szMessage_Receipt )
```

3.7.10 Wydruk kwitu transakcji eCash

a) SFI określająca czy skojarzona FP jest dostępna w danej DF

```
int mfPar_eCash ( void ) - zwraca mfeASSOCIATION
int mfPar_eCash2 ( void ) - zwraca mfeASSOCIATION
```

b) FP realizujące wydruk kwitu transakcji eCash

```
int mfPr_eCash ( const int i_eReceiptType,
const int i_eTransactionType,
const char* i_szDateTimeTransaction,
const char* i_szMerchantIdentification,
const char* i_szMerchantTerminalIdentification,
const long i_nTransactionNumber,
const long i_nBatchNumber,
const long i_nSTAN,
const long i_nPOSNumber,
const long i_nPOSTransactionNumber,
const int i_nCashierNumber,
const char* i_szCashierIdentification,
const char* i_szControlNumber,
const char* i_szBankNumber,
const char* i_szAccountNumber,
```

```

const char* i_szBillIssuerName,
const char* i_szBillIssuerIdentification,
const char* i_szBillSequenceNumber,
const char* i_szCustomerIdentification,
const char* i_szBillDueDate,
const long i_nBillAmount,
const long i_nCustomerAmount,
const long i_nFineAmount,
const long i_nCommissionAmount,
const long i_nTotalAmount,
const char* i_szBillCode,
const char* i_szApprovalCode,
const char* i_szTransactionSignature,
const int i_nAcquirerResponseCode,
const char* i_szMessage_Receipt,
const char* i_szMessage_Receipt2 ) - niezaimplementowane (zwraca mfeIMPLEMENTATION)

```

```

int mfPr_eCash2 ( const int i_eAcceptorType,
const int i_eApplicationMode,
const int i_eReceiptType,
const int i_bReceiptOriginal,
const int i_eTransactionType,
const char* i_szDateTimeTransaction,
const char* i_szMerchantIdentification,
const char* i_szMerchantTerminalIdentification,
const long i_nTransactionNumber,
const long i_nBatchNumber,
const long i_nSTAN,
const long i_nPOSNumber,
const long i_nPOSTransactionNumber,
const int i_nCashierNumber,
const char* i_szCashierIdentification,
const char* i_szControlNumber,
const char* i_szBankNumber,
const char* i_szAccountNumber,
const char* i_szBillBarCode,
const char* i_szBillIssuerName,
const char* i_szBillIssuerIdentification,
const char* i_szBillSequenceNumber,
const char* i_szCustomerIdentification,
const char* i_szBillDueDate,
const long i_nBillAmount,
const long i_nCustomerAmount,
const long i_nPromotionAmount,
const long i_nFineAmount,
const long i_nCommissionAmount,
const long i_nTotalAmount,
const char* i_szBillCode,

```

```
const char* i_szApprovalCode,  
const char* i_szTransactionSignature,  
const int i_nAcquirerResponseCode,  
const char* i_szMessage_Receipt,  
const char* i_szMessage_Receipt2 ) - niezaimplementowane (zwraca mfeIMPLEMENTATION)
```

3.7.11 Wydruk doładowania konta PREPAID

```
MFBO_INT mfParPrePaid_Voucher (__MFVPARAM ); - zwraca mfeOK  
MFBO_INT mfPrPrePaid_Voucher(  
const int i_eReceiptType, // dopuszczalne wartości wg EReceiptType  
const char* i_szDateTimeTransaction, // format YYYYMMDDhhmm  
const char* i_szMerchantId, // 15 znaków  
const char* i_szTerminalId, // 8 znaków  
const long i_nPOSTransactionNumber, // jeśli > -1  
const long i_nTransactionId, // jeśli > -1  
const long i_nBatchNumber, // jeśli > -1  
const long i_nVoucherValue,  
const char* i_szVoucherName, // 20 znaków  
const char* i_szVoucherSerialNumber, // 16 znaków  
const char* i_szVoucherCode, // 24 znaki  
const char* i_szVoucherExpireDate, // łańcuch  
const char* i_szIssuerPhoneNumber, // jeśli nie pusty  
const int i_nResponseCode, // dla eReceiptType_Decline i eReceiptType_Error  
const char* i_szMessage1, // jeśli nie pusty  
const char* i_szMessage2 // jeśli nie pusty  
__MFPARAM  
);
```

```
typedef enum
{
    eReceiptType_Approve = 1, // (transakcja zaaprobowana)
    eReceiptType_Decline = 2, // (odmowa)
    eReceiptType_Error = 3, // (błąd kasy/serwera)
    eReceiptType_Cancel = 4, // (transakcja przerwana)
    eReceiptType_Reversal = 5 // (transakcja anulowana)
} EReceiptType;
```

Parametry:

- *i_eReceiptType*, - dopuszczalne wartości wg EReceiptType, rodzaj wydruku
- *i_szDateTimeTransaction*, - data i czas transakcji
- *i_szMerchantId*, - identyfikator sklepu
- *i_szTerminalId*, - numer terminala
- *i_nPOSTransactionNumber*, - numer transakcji na kasie
- *i_nTransactionId*, - numer transakcji PrePaid
- *i_nBatchNumber*, - numer paczki
- *i_nVoucherValue*, - wartość doładowania
- *i_szVoucherName*, - identyfikator doładowania (np. TAKTAK 50)
- *i_szVoucherSerialNumber*, - nr seryjny doładowania
- *i_szVoucherCode*, - kod doładowania
- *i_szVoucherExpireDate*, - data ważności doładowania
- *i_szIssuerPhoneNumber*, - numer tel. dostawcy (sieci komórkowej)
- *i_nResponseCode*, // dla eReceiptType_Decline i eReceiptType_Error, nie używane
- *i_szMessage1*, // komunikat na wydruk
- *i_szMessage2*, // komunikat na wydruk

3.8 Funkcje obsługujące szufladę

3.8.1 Definicja parametrów (rodzaju) szuflady

a) SFI zwracająca nazwy zaimplementowanych szuflad
`int mfParDrawer(int Item, char *name, int *maxNo)`

b) FP definiująca typ szuflady
`int mfDefDrawer(int item, int portNo)`

3.8.2 Otwarcie szuflady

`int mfDrawerOpen (int no)`

3.8.3 Sprawdzenie stanu szuflady (otwarta / zamknięta)

`int mfDrawerIsOpen (int no, int *flag)`

3.9 Funkcje obsługujące wyświetlacz

3.9.1 Odczyt parametrów wyświetlacza zainstalowanego w DF

a) SFI zwracająca parametry zaimplementowanych w danej wersji biblioteki typów wyświetlaczy fiskalnych (dotyczy DF) lub niestandardowych

```
int mfParFiscDisp ( int item, char *name, int *column, int *row, char *setOfChar,  
int *numericFlag, int *pointFlag )
```

b) FP definiująca wyświetlacz dostępny z listy (fiskalny lub niestandardowy)

```
int mfDefFiscalDisp ( int item, int portNo )
```

3.9.2 Wyświetlenie POZYCJI SPRZEDAŻY

```
int mfDispTicketLine ( const char *name, long price, long quantity, long value )
```

3.9.3 Wyświetlenie SUMY

```
int mfDispSum ( const char *sumName, long sumValue )
```

3.9.4 Wyświetlenie PODSUMOWANIA

```
int mfDispSummary( const char *payName, long payValue,  
const char *remainderName, long remainderValue )
```

3.9.5 Wyświetlenie LOGO UŻYTKOWNIKA

```
int mfDispConsumerLOGO ( const char *lineLOGO1, const char *lineLOGO2 )
```

3.9.6 Graficzny wygaszacz ekranu (logo)

Funkcja graficznego wygaszacza ekranu dostępna jest tylko dla drukarki Mera. Logo wyświetla się po wydrukowaniu paragonu lub raportu dobowego (po upływie minimum 46 sekund od zakończenia wydruku). Po zmianie logo jest również jednokrotnie drukowane na następnym raporcie dobowym.

Ustawienie wygaszacza ekranu następuje poprzez przekazanie do biblioteki ścieżki dostępu do pliku tekstowego. Każdy znak w pliku tekstowy odpowiada jednemu pikselowi na ekranie, a ponieważ ekran drukarki Mera zawiera 192 piksele w poziomie, oraz 64 piksele w pionie, to definicja wygaszacza w pliku tekstowym też zawiera 64 linie po 192 znaki, licząc od linii nr 2 (pierwsza linia w pliku jest pomijana). Wstawienie znaku 'X' (duża litera) powoduje zapalenie piksela, każdy inny znak to piksel wygaszony.

Przekazanie ścieżki do pliku z wygaszaczem musi wystąpić przed wywołaniem mfStart za pomocą następujących funkcji:

```
MFBO_INT mfParConfigParam( __MFVPARAM ); - zawsze zwraca mfOK
```

```
MFBO_INT mfDefConfigParam( int i_nType, long i_lParam, const char *i_pParam  
__MFPARAM);
```

gdzie:

- i_nType: #define mfeConfigScreenSaverPicture 11

- mfeConfigScreenSaverPicture - i_lParam zawiera ścieżkę z nazwą pliku do wyświetlenia na wyświetlaczu

3.10 Funkcje statusowe

3.10.1 Tryb szkoleniowy / fiskalny

int mfStatMode (int *flag)

3.10.2 Waluta przed denominacją / po denominacji

int mfStatDenom(int *flag)

3.10.3 Doba fiskalna zamknięta / otwarta

int mfStatShift (int *flag)

3.10.4 Paragon zamknięty / otwarty

int mfStatReceipt (int *flag)

3.10.5 Dokument nie drukowany / drukowany

int mfStatDocument (int *flag)

3.10.6 Brak papieru

int mfStatNoPaper (int *flag)

3.11 Pozostałe funkcje

3.11.1 Bezpośrednia komunikacja z drukarką

int mfAnyCommand (int outputL,const char *output,int inputL,char *input,int *resultL,long timeOut)

3.11.2 Funkcja umożliwia odczyt kodu błędu.

int mfError (void)

3.11.3 Funkcja umożliwia odczyt numeru seryjnego drukarki.

int mfGetSerNo (char *serNo)

3.11.4 Odczyt numeru kompilacji wersji biblioteki

int mfParLibBuild (int *build)

3.11.5 Funkcja zwraca czy dla danej drukarki jest konieczne obsługiwane EJ

int mfejDumpNecessary (int *pnEjType)

3.11.6 Ustawianie poziomu logowania

int mfSetDebugLevel(const int i_nDebugLevel)

3.11.7 Zwraca max liczbę towarów które można zwrócić w jednym zwrocie

int mfParMaxLinesReturn (int *nMaxLines)

4 Indeks funkcji MFBO

Indeks alfabetyczny

mfAnyCommand	25
mfBeginInvoice.....	11
mfBeginShift	10
mfBeginTicket.....	10
mfBgCashierReport.....	15
mfBgOrder.....	14
mfBgReturn.....	14
mfBgShiftReport.....	15
mfDefConfigParam.....	23
mfDefDrawer.....	22
mfDefFiscalDisp.....	23
mfDefFooter.....	8
mfDefPOS.....	8
mfDefString.....	9
mfDefTax.....	9
mfDefTime.....	8
mfDispConsumerLOGO	23
mfDispSum.....	23
mfDispSummary.....	23
mfDispTicketLine.....	23
mfDrawerIsOpen	22
mfDrawerOpen.....	22
mfejDumpNecessary.....	25
mfEndCashierReport	15
mfEndInvoice.....	11
mfEndOrder.....	14
mfEndReturn.....	15
mfEndShiftReport.....	16
mfEndTicket.....	10
mfEndTicketExt.....	11
mfError.....	25
mfGetCoefficient.....	6
mfGetDate.....	6
mfGetFirmwareVersion.....	6
mfGetFiscalRec.....	12
mfGetFiscalRecRange.....	12
mfGetFreeRec.....	6
mfGetLastInvoiceNo.....	6
mfGetLastTicketNo	6
mfGetSerNo	25
mfGetTax	6
mfGetTaxRec.....	6
mfGetTime.....	6

mfGetVersion.....	6
mfLineCashierReport	15
mfLineOrder.....	14
mfLineReturn	14
mfLineReturnExt	14
mfLineShiftReport.....	15
mfLineTicket.....	10
mfLineTicketExt.....	10
mfPar_eCash.....	18
mfPar_eCash2.....	18
mfParCashierReport.....	15
mfParConfigParam.....	23
mfParDrawer.....	22
mfParEFT.....	16
mfParFiscal	7
mfParFiscDisp.....	23
mfParFooter.....	8
mfParGetFiscalRecRange.....	12
mfParGetFreeRec.....	6
mfParGetTaxRec.....	6
mfParHeader	7
mfParInvoice.....	11
mfParLibBuild	25
mfParLibVersion.....	7
mfParLineReturnExt.....	14
mfParLineTicketExt	10
mfParMaxLinesReturn	25
mfParMaxShiftNo.....	7
mfParMessage	7
mfParOrder.....	14
mfParPayIn.....	14
mfParPayOut.....	14
mfParPLU.....	7
mfParPOS	8
mfParPrePaid_Voucher.....	20
mfParPrState	12
mfParReturn.....	14
mfParShiftReport.....	15
mfParStornoLineTicketExt	11
mfParString.....	9
mfParTax.....	9
mfParTicket	7
mfParTime	8
mfParVISA.....	16
mfPr_eCash	18
mfPr_eCash2.....	19
mfPrDetailReport.....	12
mfPrEFT_RegisterOffline.....	18



mfPrEFT_Reversal.....	17
mfPrEFT_Transaction.....	17
mfPrGlobalReport.....	12
mfPrPayIn.....	14
mfPrPayOut.....	14
mfPrPrePaid_Voucher.....	20
mfPrState.....	12
mfPrVISA.....	16
mfPrVoid.....	16
mfSetDebugLevel.....	25
mfShiftEnd.....	10
mfStart.....	8
mfStart2.....	8
mfStatDenom.....	24
mfStatDocument.....	24
mfStatMode.....	24
mfStatNoPaper	24
mfStatReceipt	24
mfStatShift.....	24
mfStop.....	8
mfStornoLineTicket.....	10
mfStornoLineTicketExt.....	11
mfStornoTicket.....	10