

Документация API библиотеки libpioneer3.dll

Версия библиотеки 3.9.53

31 марта 2022 г.

Оглавление

I Общие положения	3
II Команды библиотеки	5
1 Команды общего назначения	6
1.1 Получение версии библиотеки	6
1.2 Инициализация сессии работы с ККТ	6
1.3 Закрытие сессии работы с ККТ	8
1.4 Перезагрузка ККТ	8
1.5 Получить техническую информацию о ККТ	9
1.6 Получить состояние ККТ	10
1.7 Записать товары в номенклатуру ККТ	12
1.8 Прочитать товары из номенклатуры ККТ	15
1.9 Удалить товары из номенклатуры ККТ	17
2 Команды режима ФР	19
2.1 Вход в режим ФР	20
2.2 Выход из режима ФР	21
2.3 Запросить статус выполнения задачи в ККТ	22
2.4 Сброс состояния ФР к начальным условиям	29
2.5 Открытие смены	30
2.6 Закрытие смены	32
2.7 Регистрация/перерегистрация ФН	34
2.8 Закрытие архива ФН	37
2.9 Формирование отчета о состоянии расчетов	39
2.10 Открыть документ	40
2.11 Отмена документа	43
2.12 Добавить позицию в документ	44
2.13 Добавить текстовую информацию в печатную форму чека	47
2.14 Добавить текстовую информацию в подвал печатной формы чека	49
2.15 Установить признак агента	51
2.16 Установить отраслевые реквизиты	53
2.17 Установить данные покупателя	54
2.18 Отправить запрос о коде маркировки	55

2.19	Заккрыть документ	57
2.20	Х-отчет (сменный отчет без гашения)	59
2.21	Добавить текстовую информацию в печатную форму чека	60
2.22	Запрос суммы наличных в кассе ККТ	61
2.23	Печать произвольной строки на принтере ККТ	62
2.24	Печать копии фискального документа из ФН по номеру	63
2.25	Посылка пакета keep-alive в режиме ФР	64
2.26	Установка даты/времени в ККТ	65
3	Команды работы с ФН	66
3.1	Получить архивный документ ФН по номеру	67
3.2	Получить реквизит документа ФН по номеру и тегу	69
3.3	Получить состояние ФН	71
III	Примеры	73
IV	Таблицы	79

Часть I

Общие положения

Библиотека предназначена для работы с ККТ Пионер-114Ф в режиме фискального регистратора (ФР). Библиотека реализует следующие функции: открытие смены, формирование торговых документов (расход, возврат расхода), внесение и выплата наличных, X-отчет, закрытие смены (Z-отчет), получение текущей суммы наличности в кассе, печать произвольного текста.

Библиотека работает с ККТ по протоколу TCP/IP, используя по умолчанию порт 8080. Начиная с версии ККТ 2.60.87 порт подключения может быть изменен на ККТ, в этом случае при подключении необходимо указывать его после IP-адреса. После удачной инициализации сессии работы с ККТ, все команды (кроме входа/выхода в режим ФР, а также получения суммы наличности в кассе) выполняются по следующей схеме:

1. отправка команды в ККТ и получение в ответ номера задачи
2. опрос ККТ через произвольные интервалы времени до получения статуса задачи «выполнено» или «ошибка».

При инициализации сессии библиотека запоминает заводской номер ККТ, и в дальнейшем проверяет его при каждом обращении к ККТ (во избежание внезапной смены IP адреса ККТ в процессе работы).

В каждой команде возможно задать время таймаута (в мс), в течении которого библиотека должна получить ответ от ККТ. Все команды получают параметры и возвращают ответ в виде null-терминированных строк. Память под строку с ответом выделяет и освобождает библиотека, поэтому нельзя освобождать указатель на строку с ответом, а наоборот, для дальнейшего разбора рекомендуется скопировать строку ответа, выделив память в пользовательской программе.

Параметры и результаты выполнения оформлены в виде формате JSON, все строковые данные должны быть в UTF-8.

Каждая команда возвращает непосредственно код ошибки (или удачного выполнения). Этот код относится непосредственно к протоколу обмена библиотеки и ККТ, а не к выполнению задачи в ККТ. Коды результатов выполнения задачи в ККТ возвращаются в строке ответа.

Формат ответа с результатом выполнения команды следующий:

```
{
    "response": integer,
    "extended": integer,
    "data": json
}
```

Все команды имеют спецификацию вызова вида:

```
extern "C" __declspec(dllimport) _cdecl
```

В дальнейшей документации при описании формата команд эта спецификация опущена.

Для удобства разработчиков к библиотеке приложен интерфейс на языке Go (в папке golang). (Экспериментальный!).

Часть II

Команды библиотеки

Глава 1

Команды общего назначения

1.1 Получение версии библиотеки

```
void pioneer3_version(int *major, int *minor, int *micro, int *build);
```

1.2 Инициализация сессии работы с ККТ

```
int pioneer3_open(const char *host, uint16 timeout, const char* &response);
```

Параметры:

host	IP-адрес ККТ (и опционально :port ^a)
timeout	таймаут ожидания ответа
response	строка с результатом выполнения

^a начиная с версии ККТ 2.60.87

Возвращает:

отрицательное число	код ошибки
положительное число	идентификатор сессии работы с ККТ, в дальнейшем используется во всех командах.

Блок данных в строке ответа:

```
{
    "serial": string,
    "active_goods" : integer,
    "flash_bank": integer,
    "flash_size": integer,
    "goods_count": integer,
    "boot_version": integer,
    "line_length": integer,
    "version": string
}
```

serial	заводской номер ККТ
active_goods ^a	использованное число позиций в справочнике номенклатуры
flash_bank ^a	активный банк с прошивкой (1..2)
flash_size ^a	размер флешпамяти ККТ (в кбайтах)
goods_count ^a	максимальное число позиций в справочнике номенклатуры
boot_version ^a	версия загрузчика
line_length ^a	текущая ширина чековой ленты (в символах)
version ^a	версия прошивки ККТ (параметр ФС)

^a начиная с версии ККТ 2.58.70

Версия загрузчика представлена в виде 32-битного LE числа, каждый байт которого расшифровывается:

первый байт - 0, второй байт - build, третий байт - minor, четвертый байт - major

Версия загрузчика = major.minor.build.0

Пример строки ответа:

```
{
  "response": 0,
  "extended": 0,
  "data": {
    "active_goods": 1824,
    "boot_version": 1638913,
    "flash_bank": 1,
    "flash_size": 1024,
    "goods_count": 1968,
    "line_length": 27,
    "serial": "57000012",
    "version": "2.59.77d wifi"
  }
}
```

Версия загрузчика из примера: 1.2.25.0

1.3 Закрытие сессии работы с ККТ

int pioneer3_close(**int** handle)

Параметры:

handle	полученное значение при инициализации сессии
--------	--

Возвращает: код ошибки

1.4 Перезагрузка ККТ¹

int pioneer3_reset(**int** handle, uint16 timeout)

Входные параметры: нет

Выходные параметры: нет

¹начиная с версии ККТ 2.57.68

1.5 Получить техническую информацию о ККТ²

```
int pioneer3_get_info(int handle, uint16 timeout, const char*  
    &response)
```

Входные параметры: нет

Блок данных в строке ответа:

```
{  
    "serial": string,  
    "active_goods": integer,  
    "flash_bank": integer,  
    "flash_size": integer,  
    "goods_count": integer,  
    "boot_version": integer,  
    "line_length": integer,  
    "version": string  
}
```

serial	заводской номер ККТ
active_goods	использованное число позиций в справочнике номенклатуры
flash_bank	активный банк с прошивкой (1..2)
flash_size	размер флешпамяти ККТ (в кбайтах)
goods_count	максимальное число позиций в справочнике номенклатуры
boot_version	версия загрузчика
line_length	текущая ширина чековой ленты (в символах)
version	версия прошивки ККТ (параметр ФС)

Особенности:

Команда не требует опроса статуса выполнения, запрошенные данные возвращаются сразу в строке ответа.

²начиная с версии ККТ 2.58.70

1.6 Получить состояние ККТ³

```
int pioneer3_get_state(int handle, uint16 timeout, const char*  
    &response)
```

Входные параметры: нет

Блок данных в строке ответа:

```
{  
    "dev_context": integer,  
    "fd_mode": integer,  
    "fd_ext_mode": integer,  
    "taxcode": integer,  
    "agents": integer,  
    "fd_serial": string,  
    "regnum": string,  
    "company_name": string,  
    "company_inn": string,  
    "sale_address": string,  
    "sale_location": string,  
    "ofd_name": string,  
    "ofd_inn": string,  
    "fns_site": string,  
    "sender_email": string,  
    "automat_number": string,  
    "date_time": integer  
}
```

³начиная с версии ККТ 2.58.70

dev_context	текущий контекст ККТ, битовая маска
fd_mode	режимы работы ФН, битовая маска
fd_ext_mode	режимы работы ФН (расширенные), битовая маска
taxcode	зарегистрированные системы налогообложения, битовая маска
agents	зарегистрированные признаки агентов, битовая маска
fd_serial	заводской номер ФН
regnum	регистрационный номер ККТ
company_name	название организации
company_inn	ИНН организации
sale_address	адрес организации
sale_location	место торговли
ofd_name	название ОФД
ofd_inn	ИНН ОФД
fns_site	адрес сайта ФНС
sender_email	email отправителя чеков
automat_number	номер автомата
date_time	текущие дата/время ККТ (unixtime) ^a

^a начиная с версии ККТ 2.59.83

Особенности:

Команда не требует опроса статуса выполнения, запрошенные данные возвращаются сразу в строке ответа.

Значения флагов для параметра dev_context приведены в "Флаги контекста ККТ".

Значения флагов для параметра fd_mode приведены в "Режимы работы ККТ и ФН".

Значения флагов для параметра fd_ext_mode приведены в "Расширенные режимы работы ККТ и ФН".

Значения флагов для параметра taxcode приведены в "Коды системы налогообложения".

Значения флагов для параметра agents приведены в "Виды агентов (битовая маска)".

Значения для названия и адреса организации, ОФД запрашиваются в памяти ККТ и могут отличаться от содержимого ФН, о чем ККТ предупреждает при загрузке.

1.7 Записать товары в номенклатуру ККТ⁴

```
int pioneer3_write_goods(int handle, uint16 timeout, const char
    *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
    "list" : [
        {
            "goods_id" : integer,
            "dept_it" : integer,
            "intcode" : string,
            "name" : string,
            "price" : integer,
            "excise" : integer,
            "unit" : string,
            "barcode" : string,
            "item_attr" : integer,
            "item_attr_ext" : integer
        }
    ]
    "delete_before" : integer
}
```

list	массив номенклатурных позиций для записи в ККТ
delete_before	флаг очистки всей номенклатуры перед записью (0 - не очищать, иначе очищать)

⁴начиная с версии ККТ 2.59.77

Таблица 1.1: Структура номенклатурной позиции

goods_id	позиция в справочнике номенклатуры (1..макс.номер позиции)
dept_id	номер отдела (1..16)
intcode	внутренний код (макс. 5 символов)
name	наименование (макс. 84 символа)
price	цена за единицу
excise	акциз за единицу
unit	ед.измерения (макс. 3 символа)
barcode	штрихкод (макс. 14 символов, EAN8, EAN12, EAN13, ITF14)
item_attr	признак предмета расчета
item_attr_ext	расширенный признак предмета расчета

Возвращает: код ошибки

Блок данных в строке ответа:

```
[
    {
        "goods_id" : integer,
        "response" : integer
    }
]
```

goods_id	позиция в справочнике номенклатуры
response	результат выполнения записи номенклатурной позиции

Особенности:

Максимальный номер позиции для ККТ можно узнать из выдачи команды 1.5 "Получить техническую информацию о ККТ"(параметр goods_count).

Признаки предмета расчета перечислены в [1].

Признаки предмета расчета (расширенные), может принимать значения 0 (нет признаков) и 1 (маркированный предмет расчета).

Каждая позиция записывается в ККТ отдельной командой, поэтому в блоке данных в строке ответа указан результат выполнения для каждой позиции (коды аналогичны соответствующим кодам результата выполнения обычных задач).

Примеры:

```
{
  "delete_before": 1,
  "list": [
    {
      "barcode": "978020137962",
      "dept_id": 1,
      "goods_id": 1,
      "intcode": "10050",
      "item_attr": 1,
      "name": "Вотка северная",
      "price": 138100,
      "excise": 20000
    },
    {
      "dept_id": 1,
      "goods_id": 7,
      "intcode": "150",
      "item_attr": 1,
      "name": "Мишка на сервере",
      "price": 1230000,
      "unit": "шт."
    }
  ]
}
```

1.8 Прочитать товары из номенклатуры ККТ⁵

```
int pioneer3_read_goods(int handle, uint16 timeout, const char
    *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params (список позиций):

```
{
    "list" : [
        {
            "goods_id" : integer
        }
    ]
}
```

list	список позиций для чтения из ККТ
goods_id	номер позиции для чтения из ККТ

Формат параметра params (диапазон позиций):

```
{
    "range" : {
        "goods_id" : integer,
        "goods_count" : integer
    }
}
```

range	структура с начальным номером и количеством позиций для чтения
goods_id	начальный номер для чтения
goods_count	количество позиций для чтения

Возвращает: код ошибки

⁵начиная с версии ККТ 2.59.77

Блок данных в строке ответа:

```
[
  {
    "goods_id" : integer,
    "dept_it" : integer,
    "intcode" : string,
    "name" : string,
    "price" : integer,
    "excise" : integer,
    "unit" : string,
    "barcode" : string,
    "item_attr" : integer,
    "item_attr_ext" : integer
  },
  {
    "goods_id" : integer,
    "response" : integer
  }
]
```

Структура номенклатурной позиции

goods_id	позиция в справочнике номенклатуры
dept_id	номер отдела
intcode	внутренний код
name	наименование
price	цена за единицу
excise	акциз за единицу
unit	ед.измерения
barcode	штрихкод
item_attr	признак предмета расчета
item_attr_ext	расширенный признак предмета расчета

Случай ошибки при считывании позиции

goods_id	позиция в справочнике номенклатуры
response	код ошибки

1.9 Удалить товары из номенклатуры ККТ⁶

```
int pioneer3_delete_goods(int handle, uint16 timeout, const char
    *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params (список позиций):

```
{
    "list" : [
        {
            "goods_id" : integer
        }
    ]
}
```

list	список позиций для удаления из ККТ
goods_id	номер позиции для удаления из ККТ

Формат параметра params (диапазон позиций):

```
{
    "range" : {
        "goods_id" : integer,
        "goods_count" : integer
    }
}
```

range	структура с начальным номером и количеством позиций для удаления
goods_id	начальный номер для удаления
goods_count	количество позиций для удаления

В случае указания диапазона позиций для удаления можно не указывать параметр "count" и будут удалены все позиции от стартовой до конца номенклатуры.

⁶начиная с версии ККТ 2.59.77

Возвращает: код ошибки

Блок данных в строке ответа:

```
[
    {
        "goods_id" : integer,
        "response" : integer
    }
]
```

goods_id	удаленная позиция в справочнике номенклатуры
response	результат удаления номенклатурной позиции

Глава 2

Команды режима ФР

2.1 Вход в режим ФР

int pioneer3_fremu_enter(**int** handle, uint16 timeout, **const char*** &response)

Параметры:

handle	идентификатор сессии (полученный при инициализации сессии)
timeout	таймаут ожидания ответа
response	строка с результатом выполнения

Возвращает: код ошибки

Блок данных в строке ответа: нет

2.2 Выход из режима ФР

int pioneer3_fremu_exit(**int** handle, uint16 timeout, **const char*** &response)

Параметры:

handle	идентификатор сессии (полученный при инициализации сессии)
timeout	таймаут ожидания ответа
response	строка с результатом выполнения

Возвращает: код ошибки

Блок данных в строке ответа: нет

2.3 Запросить статус выполнения задачи в ККТ

```
int pioneer3_fremu_query(int handle, uint16 timeout, uint32 task,  
    const char* &response)
```

Параметры:

handle	идентификатор сессии (полученный при инициализации сессии)
timeout	таймаут ожидания ответа
task	номер задачи, полученный ранее при постановке задачи в ККТ
response	строка с результатом выполнения

Возвращает: код ошибки

Блок данных в строке ответа^a:

```
{
  "fd_date": integer,
  "fd_number": integer,
  "fd_sign": integer,
  "shift_id": integer,
  "fd_error": integer,
  "fd_fail": integer,
  "fd_overflow": integer,
  "fd_shift_state": integer,
  "shift_cash": integer,
  "fd_waiting": integer,
  "fd_waiting_date": integer,
  "fd_waiting_document": integer,
  "sales" : {
    "check_count": integer,
    "check_total": integer
  },
  "buys" : {
    "check_count": integer,
    "check_total": integer
  },
  "sales_return" : {
    "check_count": integer,
    "check_total": integer
  },
  "buys_return" : {
    "check_count": integer,
    "check_total": integer
  },
  "sales_correction" : {
    "check_count": integer,
    "check_total": integer
  },
  "buys_correction" : {
    "check_count": integer,
    "check_total": integer
  },
  "mark_validation_status": integer,
  "mark_result_code": integer,
  "mark_online_type": integer,
  "mark_status": integer,
  "mark_handle_code": integer
}
```

^aначиная с версии ККТ 2.57.60

Параметры в блоке данных возвращаются, если текущая задача завершает формирование документа: чека, открытия/закрытия смены, внесения/выплаты. Параметры `fd_number` и `fd_sign` возвращаются в случае, если документ фискальный (т.е. вышеуказанные, кроме внесения/выплаты). Если ККТ не фискализирована, то для всех документов будет возвращен только параметр `fd_date`.

fd_date	дата документа/операции, в формате unixtime
fd_number	номер фискального документа
fd_sign	фискальный признак документа
shift_id	номер смены
fd_error ^a	флаг критической ошибки ФН
fd_fail ^a	флаг ошибки ФН
fd_overflow ^a	флаг переполнения ФН
fd_shift_state ^a	состояние смены (1 - закрыта, 2 - открыта)
shift_cash ^b	сумма наличных в кассе за смену, в формате Currency
fd_waiting ^b	количество документов, ожидающих отправки в ОФД
fd_waiting_date ^b	дата первого документа, ожидающего отправки в ОФД в формате unixtime
fd_waiting_document ^b	номер первого документа, ожидающего отправки в ОФД
sales ^b	счетчики прихода (вложенная JSON структура)
buys ^b	счетчики расхода (вложенная JSON структура)
sales_return ^b	счетчики возврата прихода (вложенная JSON структура)
buys_return ^b	счетчики возврата расхода (вложенная JSON структура)
sales_correction ^b	счетчики коррекции прихода (вложенная JSON структура)
buys_correction ^b	счетчики коррекции расхода (вложенная JSON структура)
mark_validation_status ^c	статус проверки КМ
mark_result_code ^c	результат проверки КМ
mark_online_type ^c	тип КМ, определенный при проверке
mark_status ^c	присвоенный статус МТ
mark_handle_code ^c	режим обработки КМ

^a параметр возвращается только для документов открытия и закрытия смены

^b параметр возвращается только для документов закрытия смены

^c параметр возвращается только для запроса о коде маркировки

Таблица 2.1: Структура счетчиков чеков

check_count	количество чеков
check_total	сумма чеков, в формате Currency

Особенности:

Если в JSON структуре ответа поля response и extended имеют нулевые значения, значит задача завершилась успешно. Иначе в response будет код 3, а в extended - код расширенной диагностики.

Список ошибок можно посмотреть в Коды ответа ККТ (поле "response" в JSON-ответе) и Расширенные коды ответа ККТ (поле "extended" в JSON-ответе).

Этот код ошибки может быть числом типа uint32, в этом случае это внутренний код ошибки, который используется для режима отладки.

Значения поля mark_validation_status указаны в Статус проверки КМ (битовая маска).

Значения поля mark_result_code указаны в [1], табл.110

Значения поля mark_online_type указаны в [1], табл.104

Значения поля mark_status указаны в [1], табл.105,107

Значения поля mark_handle_code указаны в [1], табл.132

Примеры:

Успешное завершение задачи

```
{
  "response": 0,
  "extended": 0
}
```

Успешное завершение открытия смены

```
{
  "response": 0,
  "extended": 0,
  "data": {
    "fd_date": 1591467060,
    "fd_error": 0,
    "fd_fail": 0,
    "fd_number": 7,
    "fd_overflow": 0,
    "fd_shift_state": 2,
    "fd_sign": 185439481,
    "shift_id": 3
  }
}
```

Успешное завершение закрытия смены

```
{
  "response": 0,
  "extended": 0,
  "data": {
    "fd_date": 1591467060,
    "fd_error": 0,
    "fd_fail": 0,
    "fd_number": 9,
    "fd_overflow": 0,
    "fd_shift_state": 1,
    "fd_sign": 3150062908,
    "fd_waiting": 9,
    "fd_waiting_date": 1589828460,
    "fd_waiting_document": 1,
    "sales": {
      "check_count": 1,
      "check_total": 11935200
    },
    "shift_cash": 0,
    "shift_id": 3
  }
}
```

Успешное завершение формирования фискального документа

```
{
  "response": 0,
  "extended": 0,
  "data": {
    "fd_date": 1558460032,
    "fd_number": 1234,
    "fd_sign": 345007834,
    "shift_id": 3
  }
}
```

Задача еще не завершена

```
{
  "response": 3,
  "extended": 103
}
```

Ошибка выполнения задачи, неправильная сумма к оплате

```
{  
  "response": 3,  
  "extended": 109  
}
```

2.4 Сброс состояния ФР к начальным условиям¹

int pioneer3_fremu_reset(**int** handle, uint16 timeout)

Параметры:

handle	идентификатор сессии (полученный при инициализации сессии)
timeout	таймаут ожидания ответа

Возвращает: код ошибки

Блок данных в строке ответа: нет

¹Начиная с версии ККТ 2.59.77

2.5 Открытие смены

```
int pioneer3_fremu_open_shift(int handle, uint16 timeout, uint32
    &task, const char *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
    "user_id": integer,
    "user_name": string,
    "user_inn": string,
    "opt_attr": string,
    "opt_data": string,
    "force_print": bool,
    "forbid_print": bool
}
```

user_id	код пользователя в ККТ (1..16)	обяз.
user_name	полное имя кассира	обяз.
user_inn	ИНН кассира	обяз.
opt_attr ^c	доп.реквизит ООС	
opt_data ^c	доп.данные ООС (в формате base64)	
force_print ^a	принудительная печать чека (в режиме Интернет)	
forbid_print ^b	запрет печати чека	

^a начиная с версии ККТ 2.57.62

^b начиная с версии ККТ 2.63.96

^c начиная с версии ККТ 3.64.201, для ФФД 1.2

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

Должен быть задан параметр `user_id` или `user_name`. Если заданы оба, то `user_id` игнорируется.

Дата открытия смены, номер смены, номер документа формируются в ККТ и не передаются в библиотеку.

При задании одновременно флагов `force_print` и `forbid_print` используется `forbid_print`.

Примеры:

Задаем пользователя по id в ККТ

```
{  
  "user_id": 1  
}
```

Задаем пользователя по имени и ИНН

```
{  
  "user_name": "Кассир Иванов И.И.",  
  "user_inn": "771234567890"  
}
```


2.6 Заккрытие смены

int pioneer3_fremu_close_shift(**int** handle, uint16 timeout, uint32 &task, **const char** *params, **const char*** &response)

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
    "user_id": integer,
    "user_name": string,
    "user_inn": string,
    "opt_attr": string,
    "opt_data": string,
    "force_print": bool,
    "forbid_print": bool
}
```

user_id	код пользователя в ККТ (1..16)	обяз.
user_name	полное имя кассира	обяз.
user_inn	ИНН кассира	обяз.
opt_attr ^c	доп.реквизит ОЗС	
opt_data ^c	доп.данные ОЗС (в формате base64)	
force_print ^a	принудительная печать чека (в режиме Интернет)	
forbid_print ^b	запрет печати чека	

^a начиная с версии ККТ 2.57.62

^b начиная с версии ККТ 2.63.96

^c начиная с версии ККТ 3.64.201, для ФФД 1.2

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

Должен быть задан параметр user_id или user_name. Если заданы оба, то user_id игнорируется.

Дата закрытия смены, номер смены, номер документа формируются в ККТ и не передаются в библиотеку.

При задании одновременно флагов force_print и forbid_print используется forbid_print.

Примеры:

Задаем пользователя по id в ККТ

```
{  
  "user_id": 1  
}
```

Задаем пользователя по имени и ИНН

```
{  
  "user_name": "Кассир Иванов И.И.",  
  "user_inn": "771234567890"  
}
```

2.7 Регистрация/перерегистрация ФН²

int pioneer3_fremu_registration(**int** handle, uint16 timeout, **const char** *params, **const char*** &response)

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
    "regnum": string,
    "fd_serial": string,
    "taxcode": integer,
    "fd_mode": integer,
    "fd_ext_mode": integer,
    "reg_reason": integer,
    "agents": integer,
    "company_name": string,
    "company_inn": string,
    "sale_address": string,
    "sale_location": string,
    "ofd_name": string,
    "ofd_inn": string,
    "fns_site": string,
    "sender_email": string,
    "automat_number": string
}
```

²начиная с версии ККТ 2.58.70

regnum	регистрационный номер ККТ	обяз.
fd_serial	заводской номер ФН	обяз.
taxcode	системы налогообложения (битовая маска)	обяз.
fd_mode	режимы работы ФН, битовая маска	обяз.
fd_ext_mode	режимы работы ФН (расширенные), битовая маска	
reg_reason	причина перерегистрации	обяз.
agents	признаки агентов (битовая маска)	
automat_number	номер автомата	
company_name	название организации	обяз.
company_inn	ИНН организации	обяз.
sale_address	адрес организации	обяз.
sale_location	место торговли	обяз.
ofd_name	название ОФД	обяз.
ofd_inn	ИНН ОФД	обяз.
fns_site	адрес сайта ФНС	
sender_email	email отправителя чеков	
automat_number	номер автомата	
reg_reason_ext ^a	причины изм. сведений о ККТ, битовая маска	

^a начиная с версии ККТ 3.64.201, ФФД 1.2

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

Команда выполняет большой объем работ в ККТ и требует достаточно большого таймаута.

Значения флагов для параметра fd_mode приведены в "Режимы работы ККТ и ФН".

Значения флагов для параметра fd_ext_mode приведены в "Расширенные режимы работы ККТ и ФН".

Значения флагов для параметра taxcode приведены в "Коды системы налогообложения".

Значения флагов для параметра agents приведены в "Виды агентов (битовая маска)".

Допустимые причины перерегистрации указаны в "Причины перерегистрации".

Значения для названия и адреса организации, ОФД записываются в память ККТ даже в случае неудачного выполнения непосредственной команды регистрации.

Таблица с причинами изменения сведений о ККТ см. [1], табл. 93

2.8 Заккрытие архива ФН³

```
int pioneer3_fremu_close_archive(int handle, uint16 timeout, uint32
    &task, const char *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
    "user_id": integer,
    "user_name": string,
    "user_inn": string
}
```

user_id	код пользователя в ККТ (1..16)	обяз.
user_name	полное имя кассира	обяз.
user_inn	ИНН кассира	обяз.

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

Успешное выполнение команды необратимо меняет статус ФН и оформлять далее документы на этом ФН будет невозможно.

Должен быть задан параметр user_id или user_name. Если заданы оба, то user_id игнорируется.

Дата закрытия архива, номер документа формируются в ККТ и не передаются в библиотеку.

³начиная с версии ККТ 2.58.70

Примеры:

Задаем пользователя по id в ККТ

```
{  
  "user_id": 1  
}
```

Задаем пользователя по имени и ИНН

```
{  
  "user_name": "Кассир Иванов И.И.",  
  "user_inn": "771234567890"  
}
```

2.9 Формирование отчета о состоянии расчетов⁴

```
int pioneer3_fremu_report_state(int handle, uint16 timeout, uint32  
    &task, const char* &response)
```

Параметры:

handle	идентификатор сессии (полученный при инициализации сессии)
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
response	строка с результатом выполнения

Возвращает: код ошибки

Блок данных в строке ответа: нет

⁴Начиная с версии ККТ 2.58.70

2.10 Открыть документ

```
int pioneer3_fremu_open_doc(int handle, uint16 timeout, uint32 &task,  
    const char *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{  
    "user_id": integer,  
    "user_name": string,  
    "user_inn": string,  
    "force_print": bool,  
    "doc_kind": integer,  
    "taxcode": integer,  
    "client_contact": string,  
    "client_inn": string,  
    "client_name": string,  
    "correction_kind": integer,  
    "correction_date": integer,  
    "correction_document": string,  
    "correction_description": string  
}
```

user_id	код пользователя в ККТ (1..16)	обяз.
user_name	полное имя кассира	обяз.
user_inn	ИНН кассира	обяз.
doc_kind	вид документа	обяз.
taxcode	система налогообложения	
client_contact ^a	контактные данные клиента (телефон/email)	
force_print ^b	принудительная печать чека (в режиме Интернет)	
client_inn ^c	ИНН клиента	
client_name ^c	наименование клиента	
correction_kind ^d	вид коррекции	
correction_date ^d	дата корруемой операции в формате unixtime	
correction_document ^d	номер предписания налогового органа	
correction_description ^d	описание коррекции	

^a начиная с ККТ версии 2.57.60

^b начиная с ККТ версии 2.57.62

^c начиная с ККТ версии 2.57.63

^d начиная с ККТ версии 2.58.70

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

После команды открытия документа необходимо подать одну или несколько команд "Добавить позицию в документ"(при необходимости), а затем подать команду "Заккрыть документ".

Формирование документа на ККТ начинается именно по команде "Заккрыть документ". Также можно в любой момент прервать заполнение документа командой "Отмена документа".

Коды видов документов и систем налогообложения находятся в "Виды документов"и "Коды системы налогообложения".

Виды коррекции находятся в "Виды коррекции".

Параметр client_contact должен содержать или телефон или email; это требование формата фискальных документов.

Примеры:

Задаем пользователя по id в ККТ, возврат прихода, УСН доход

```
{  
  "user_id": 1,  
  "doc_kind": 1,  
  "taxcode": 2  
}
```

Задаем пользователя по id в ККТ, коррекция прихода, УСН доход, с реквизитами контрагента

```
{  
  "client_contact": "client@mail.ru",  
  "client_inn": "7712345678",  
  "client_name": "Иванопуло",  
  "correction_date": 1591442853,  
  "correction_kind": 0,  
  "doc_kind": 7,  
  "taxcode": 2,  
  "user_id": 1  
}
```

Задаем пользователя по имени и ИНН, внесение наличных

```
{  
  "user_name": "Кассир Иванов И.И.",  
  "user_inn": "771234567890",  
  "doc_kind": 2  
}
```

2.11 Отмена документа

```
int pioneer3_fremu_cancel_doc(int handle, uint16 timeout, uint32  
    &task, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
response	строка с результатом выполнения

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

Команда предназначена для выполнения в штатном режиме ФР, то есть вне состояния ошибки в работе ФР. Это значит, что команда отработает, как положено, если предыдущие команды (открытия документа, добавления позиции, ...) завершились успешно, либо не поставились в очередь на выполнение (добавление позиции, например, не удалось вследствие неправильных параметров).

В случае сбоев по причине связи с ККТ, таймаутов получения ответа от ККТ и других подобных причин, команда отмены документа может не отработать. В этих случаях рекомендуется сначала сбросить состояние ФР командой "Сброс состояния ФР", а потом давать команду "Отмена документа".

2.12 Добавить позицию в документ

```
int pioneer3_fremu_add_item(int handle, uint16 timeout, uint32 &task,  
    const char *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
params	строка с параметрами задачи
response	строка с результатом выполнения

Добавление позиций допускается только в документы вида 0,1,5,6,7,8,10,11; только эти документы имеют табличную часть.

Формат параметра params:

```
{  
    "dept_id": integer,  
    "tax_id": integer,  
    "price": integer,  
    "quant": integer,  
    "excise": integer,  
    "intcode": string,  
    "name": string,  
    "item_disc_abs": integer,  
    "payment_method": integer,  
    "item_attr": integer,  
    "item_attr_ext": integer,  
    "mark_stamp": string,  
    "unit_id": integer,  
    "country": integer,  
    "gtd": string,  
    "mark_status": integer,  
    "mark_online_type": integer,  
    "quant_numerator": integer,  
    "quant_denominator": integer,  
    "timeout": integer  
}
```

dept_id	код отдела в ККТ (1..16)	обяз.
tax_id	код ставки НДС в ККТ (1..6)	обяз.
price	цена	обяз.
quant	цена	обяз.
excise ^c	величина акциза за единицу	
intcode	внутренний код (артикул)	обяз.
name	наименование	обяз.
item_disc_abs	скидка по абсолютной величине	
payment_method ^a	признак способа расчета (1..7)	
item_attr ^b	признак предмета расчета	
item_attr_ext ^b	признак предмета расчета (расширенный)	
mark_stamp ^b	код маркировки	
unit_id ^d	код единицы измерения	
country ^d	код страны происхождения	
gtd ^d	номер ГТД	
mark_status ^e	планируемый (или итоговый) статус товара	
mark_online_type ^e	тип КМ, определенный при проверке	
quant_numerator ^e	числитель дробного количества	
quant_denominator ^e	знаменатель дробного количества	
timeout ^e	таймаут ожидания ответа от ОИСМ (в мс)	

^a начиная с версии ККТ 2.57.68

^b начиная с версии ККТ 2.58.70

^c начиная с версии ККТ 2.59.77

^d начиная с версии ККТ 3.64.201

^e начиная с версии ККТ 3.64.204

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

Если dept_id = 0, или отсутствует необходимо задать tax_id.

Коды ставки НДС находятся в "Коды ставки НДС".

Коды признака способа расчета находятся в "Признаки способа расчета".

Признаки предмета расчета перечислены в [1], таблица 29.

Признаки предмета расчета (расширенные), может принимать значения 0 (нет признаков) и 1 (маркированный предмет расчета).

В случае неустановленного признака способа расчета принимается признак с кодом 4 (полный расчет).

Поля price, quant, item_disc_abs задаются в формате Currency (int64, с 4 десятичными разрядами после запятой).

Поле mark_stamp должно содержать строку символов, которая содержится в QR-коде акцизной марки (либо в линейном коде, если нет QR-кода). Обратите внимание, что для корректного разбора эта строка должна содержать **все** символы (в том числе служебный символ <GS>, который кодируется 0x1d)! Поле mark_stamp должно идти в паре с расширенным признаком предмета расчета 1 (маркированный товар).

Коды единиц измерения см. [1], таблица 114.

Коды стран см. в Общероссийском классификаторе стран мира

Для версии ККТ 2.64.204 (ФФД 1.2): Если задано поле timeout, то при правильно заполненных полях mark_stamp, mark_status при добавлении позиции товара будет сделан запрос в ОИСМ о коде маркировки. Если не задано поле timeout и добавляется маркированный товар, то необходимо выполнить команду Отправить запрос о коде маркировки непосредственно перед выполнением этой команды и подставить полученные значения для полей mark_status, mark_online_type.

Значения поля mark_status указываются по [1], табл. 105.

Таймаут ожидания ответа рекомендуется задавать не меньше 5000 (5 сек.)

Поле quant должно иметь значение 10000 (1 ед.), если установлены числитель и знаменатель дробного количества

Примеры:

Отдел 1 в ККТ, НДС 10/110, цена 123р.45к., количество 2.5 кг, скидка 14р.55к.

```
{
  "dept_id": 1,
  "tax_id": 3,
  "price": 1234500,
  "quant": 25000,
  "intcode": "004",
  "name": "Курица, филе",
  "item_disc_abs": 145500
}
```

2.13 Добавить текстовую информацию в печатную форму чека⁵

int pioneer3_fremu_add_text(**int** handle, uint16 timeout, uint32 &task, **const char** *params, **const char*** &response)

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
    "text": string,
    "text_type": integer
}
```

text	строка символов для печати	обяз.
text_type	способ представления строки в печатной форме (текст, штрихкод, QR-код)	обяз.

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

Способы представления указаны в "Способы представления текста на печати". Если способ представления не указан, будет напечатан просто текст.

При печати Code39 длина строки ограничена, максимальная длина строки зависит от различных условий. Если заданная строка превысит максимальную длину для Code39, то команда вернет в коде возврата ошибку "Некорректные параметры команды".

Все представления кроме текста печатаются по центру чековой ленты.

⁵начиная с версии ККТ 2.58.70

Пример:

Добавление строки в печатную форму чека

```
{  
  "text": "Акция: Купи слона и получи второго в подарок!"  
}
```

2.14 Добавить текстовую информацию в подвал печатной формы чека⁶

int pioneer3_fremu_add_footer(**int** handle, uint16 timeout, uint32 &task, **const char** *params, **const char*** &response)

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
    "text": string,
    "text_type": integer
}
```

text	строка символов для печати	обяз.
text_type	способ представления строки в печатной форме (текст, штрихкод, QR-код)	обяз.

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

Способы представления указаны в "Способы представления текста на печати". Если способ представления не указан, будет напечатан просто текст.

При печати Code39 длина строки ограничена, максимальная длина строки зависит от различных условий. Если заданная строка превысит максимальную длину для Code39, то команда вернет в коде возврата ошибку "Некорректные параметры команды".

Все представления кроме текста печатаются по центру чековой ленты.

Текст подвала, заданный в ККТ не печатается, если пользователь добавил текст в подвал, используя эту функцию.

⁶начиная с версии ККТ 2.60.87

Пример:

Добавление строки в подвал печатной формы чека

```
{  
  "text": "Акция: Купи слона и получи второго в подарок!"  
}
```

2.15 Установить признак агента⁷

```
int pioneer3_fremu_set_agent(int handle, uint16 timeout, uint32 &task,  
    const char *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
params	строка с параметрами задачи
response	строка с результатом выполнения

Признак агента устанавливается для последней добавленной позиции.
Формат параметра params:

```
{  
    "agent": integer,  
    "transfer_op_name": string,  
    "transfer_op_addr": string,  
    "transfer_op_inn": string,  
    "transfer_op_phone": string,  
    "payment_op_phone": string,  
    "payment_op_action": string,  
    "payment_reception_op_phone": string,  
    "provider_name": string,  
    "provider_phone": string,  
    "provider_inn": string  
}
```

⁷начиная с версии ККТ 2.57.68

agent	вид агента (1..7)	
transfer_op_name	наименование оператора перевода (макс. 64 символа)	для агентов 1, 2
transfer_op_addr	адрес оператора перевода (макс. 256 символов)	для агентов 1, 2
transfer_op_inn	ИНН оператора перевода	для агентов 1, 2
transfer_op_phone	телефон оператора перевода (макс. 19 символов)	для агентов 1, 2, 3, 4
payment_op_phone	телефон платежного агента (макс. 19 символов)	для агентов 1, 2, 3, 4
payment_op_action	операция платежного агента (макс. 24 символа)	для агентов 1, 2
payment_reception_op_phone	телефон оператора по приему платежей	для агентов 3, 4
provider_name	наименование поставщика (макс. 64 символа)	для всех агентов
provider_phone	телефон поставщика (макс. 19 символов)	для всех агентов
provider_inn	ИНН поставщика	для всех агентов

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

Виды агентов находятся в "Виды агентов-[1].

Каждый вид агента требует свой набор обязательных полей; все остальные поля, присутствующие в параметрах, игнорируются при передаче в ККТ.

2.16 Установить отраслевые реквизиты⁸

```
int pioneer3_fremu_set_industry_props(int handle, uint16 timeout,  
uint32 &task, const char *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
params	строка с параметрами задачи
response	строка с результатом выполнения

Если команда вызывается при пустом открытом документе, то реквизиты применяются к документу. Иначе реквизиты устанавливаются для последней добавленной позиции.

Формат параметра params:

```
{  
    "foiv_ident": integer,  
    "ind_prop_docdate": integer,  
    "ind_prop_docnumber": string,  
    "ind_prop_value": string  
}
```

foiv_ident	идентификатор ФОИВ	обяз.
ind_prop_docdate	дата документа основания (unixtime, учитывается только дата)	обяз.
ind_prop_docnumber	номер документа основания	
ind_prop_value	значение отраслевого реквизита	

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

Идентификаторы ФОИВ указаны в [1].

⁸начиная с версии ККТ 3.64.201, для ФФД 1.2

2.17 Установить данные покупателя⁹

int pioneer3_fremu_set_client_info(**int** handle, uint16 timeout, uint32 &task, **const char** *params, **const char*** &response)

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
  "birth_date": integer,
  "citizenship": integer,
  "client_ident_type": integer,
  "client_ident_data": string,
  "client_address": string,
  "client_inn": string,
  "client_name": string
}
```

birth_date	дата рождения (unixtime)	
citizenship	гражданство	
client_ident_type	вид документа удостоверяющего личность	
client_ident_data	данные документа удостоверяющего личность	
client_address	адрес	
client_inn	ИНН	
client_name	наименование	

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

Гражданство - код страны по общероссийскому классификатору стран мира

Вид документа удостоверяющего личность см. [1], табл. 116

⁹начиная с версии ККТ 2.64.201

2.18 Отправить запрос о коде маркировки¹⁰

```
int pioneer3_fremu_query_marking(int handle, uint16 timeout, uint32
    &task, const char *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
  "mark_stamp": string,
  "mark_status": integer,
  "quant": integer,
  "unit_id": integer,
  "quant_numerator": integer,
  "quant_denominator": integer,
  "timeout": integer
}
```

mark_stamp	код маркировки	обяз.
mark_status	планируемый статус товара	обяз.
quant	количество	обяз.
unit_id	код единицы измерения	
quant_numerator	числитель дробного количества	
quant_denominator	знаменатель дробного количества	
timeout	таймаут ожидания ответа от ОИСМ (в мс)	обяз.

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

¹⁰начиная с версии ККТ 2.64.204

Поле mark_stamp должно содержать строку символов, которая содержится в QR-коде акцизной марки (либо в линейном коде, если нет QR-кода). Обратите внимание, что для корректного разбора эта строка должна содержать **все** символы (в том числе служебный символ <GS>, который кодируется 0x1d)!

Значения поля mark_status указываются по [1], табл. 105.

Поле quant должно иметь значение 10000 (1 ед.), если установлены числитель и знаменатель дробного количества

Таймаут ожидания ответа рекомендуется задавать не меньше 5000 (5 сек.)

2.19 Закреть документ

```
int pioneer3_fremu_close_doc(int handle, uint16 timeout, uint32 &task,
    const char *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
    "cash": integer,
    "credit": integer,
    "postpay": integer,
    "prepay": integer,
    "otherpay": integer,
    "forbid_print": bool
}
```

cash	сумма оплаты наличными
credit	сумма оплаты электронными (картой)
postpay	сумма оплаты кредитом
prepay	сумма аванса
otherpay	другие виды оплаты
forbid_print ^a	запрет печати чека

^a начиная с версии ККТ 2.59.85

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

Сумма всех видов оплаты должна быть не меньше суммы товарной части документа.

Разница между суммой товарной части документа и суммой всех видов оплаты не может превышать суммы оплаты наличными.

Поля задаются в формате Currency (int64, с 4 десятичными разрядами после запятой).

Примеры:

Наличные 37р.12к., Электронными 2000р.

```
{  
  "cash": 371200,  
  "credit": 20000000  
}
```

2.20 X-отчет (сменный отчет без гашения)

```
int pioneer3_fremu_report_x(int handle, uint16 timeout, uint32 &task,  
    const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
response	строка с результатом выполнения

Возвращает: код ошибки

Блок данных в строке ответа: нет

2.21 X-отчет (сменный отчет без гашения) расширенная версия¹¹

int pioneer3_fremu_report_xe(**int** handle, uint16 timeout, uint32 &task, **const char*** &response)

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
  "user_id": integer,
  "user_name": string,
  "user_inn": string
}
```

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

Должен быть задан параметр user_id или user_name. Если заданы оба, то user_id игнорируется.

¹¹начиная с версии ККТ 2.63.95

2.22 Запрос суммы наличных в кассе ККТ

int pioneer3_fremu_get_money(**int** handle, uint16 timeout, **const char*** &response)

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
response	строка с результатом выполнения

Возвращает: код ошибки

Блок данных в строке ответа:

```
{
    "shift_cash": integer
}
```

В параметре "shift_cash" возвращается сумма наличных в кассе ККТ

Особенности:

Команда не требует опроса статуса выполнения, запрошенные данные возвращаются сразу в строке ответа.

Сумма возвращается в формате Currency (int64, с 4 десятичными разрядами после запятой).

Пример строки ответа:

Сумма наличности в кассе 3440р.18к.

```
{
    "response": 0,
    "extended": 0,
    "data": {
        "shift_cash": 34401800
    }
}
```

2.23 Печать произвольной строки на принтере ККТ

```
int pioneer3_fremu_print(int handle, uint16 timeout, uint32 &task,  
    const char *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
task	номер задачи, используется в дальнейшем для получения статуса выполнения задачи
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{  
    "text": string,  
    "text_type": integer  
}
```

text	строка символов для печати	обяз.
text_type ^a	способ представления строки в печатной форме (текст, штрихкод, QR-код)	обяз.

^a начиная с версии ККТ 2.58.70

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

Способы представления указаны в "Способы представления текста на печати". Если способ представления не указан, будет напечатан просто текст.

При печати Code39 длина строки ограничена, максимальная длина строки зависит от различных условий. Если заданная строка превысит максимальную длину для Code39, то команда вернет в коде возврата ошибку "Некорректные параметры команды".

Все представления кроме текста печатаются по центру чековой ленты.

Пример:

Печать строки на принтере ККТ

```
{  
    "text": "Акция: Купи слона и получи второго в подарок!"  
}
```

2.24 Печать копии фискального документа из ФН по номеру¹²

```
int pioneer3_fremu_print_doc(int handle, uint16 timeout, const char
    *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
    "fd_number": integer
}
```

В параметре fd_number передается номер документа.

Возвращает: код ошибки

Блок данных в строке ответа: нет

¹²начиная с версии ККТ 2.58.70

2.25 Посылка пакета keep-alive в режиме ФР¹³

```
int pioneer3_fremu_keep_alive(int handle, uint16 timeout, const char
    *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
    "timeout": integer
}
```

timeout	таймаут, в миллисекундах
---------	--------------------------

Возвращает: код ошибки

Блок данных в строке ответа: нет

Особенности:

После подачи команды в ККТ с ненулевым параметром timeout, ККТ ожидает получить следующую команду keep-alive или любую другую команду режима ФР до истечения таймаута.

При получении команды таймер отсчета таймаута начинает отсчет заново до установленного значения таймаута. Если получена команда keep-alive, то значение таймаута устанавливается в указанное в команде значение. При передаче timeout = 0 режим keep-alive отключается.

Если до истечения таймаута не пришла ни одна команда режима ФР, ККТ перезагружается.

Команда предназначена для использования на ККТ с модулем WiFi, которые имеют внутренний дефект и могут перестать отвечать на запросы по сети через продолжительное время неактивности. Перезагрузка помогает восстановить работоспособность интерфейса ККТ.

¹³начиная с версии ККТ 2.59.82

2.26 Установка даты/времени в ККТ¹⁴

```
int pioneer3_set_datetime(int handle, uint16 timeout, const char
    *params, const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
    "date_time": integer
}
```

date_time	дата/время (unixtime)	обяз.
-----------	-----------------------	-------

Возвращает: код ошибки

Блок данных в строке ответа: нет

¹⁴начиная с версии ККТ 2.59.83

Глава 3

Команды работы с ФН

3.1 Получить архивный документ ФН по номеру¹

```
int pioneer3_get_fd(int handle, uint16 timeout, const char *params,
const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
    "fd_doc_id": integer
}
```

В параметре fd_doc_id передается номер документа.

Возвращает: код ошибки

Блок данных в строке ответа:

```
{
    "fd_doc_type": integer,
    "fd_date": integer,
    "fd_number": integer,
    "fd_sign": integer,
    "inn": string,
    "kkt_regnum": string,
    "taxcode": integer,
    "mode": integer,
    "reason": integer,
    "shift_number": integer,
    "wait_count": integer,
    "first_waiting_date": integer,
    "operation": integer,
    "summa": integer
}
```

Выходные параметры:

¹начиная с версии ККТ 2.57.68

fd_doc_type*	вид документа
fd_date*	дата документа в формате unixtime
fd_number*	фискальный номер документа
fd_sign*	фискальный признак документа
inn	ИНН организации
kkt_regnum	регистрационный номер ККТ в ФНС
taxcode	маска используемых систем налогообложения
mode	маска режимов работы ККТ и ФН
reason	причина перерегистрации
shift_number	номер смены
wait_count	количество документов, ожидающих отправки в ОФД
first_waiting_date	дата первого документа, ожидающего отправки в ОФД в формате unixtime
operation	код операции
summa	сумма чека в формате Currency (int64, с 4 десятичными разрядами после запятой)

Параметры (*) присутствуют во всех видах документов, наличие остальных параметров в ответе зависит от вида документа.

Особенности:

Команда не требует опроса статуса выполнения, запрошенные данные возвращаются сразу в строке ответа.

Виды документов перечислены в "Виды фискальных документов-[?].

Биты маски системы налогообложения указаны в "Коды системы налогообложения-[1].

Биты маски режимы работы ККТ и ФН указаны в "Режимы работы ККТ и ФН".

Список причин перерегистрации указаны в "Причины перерегистрации".

3.2 Получить реквизит документа ФН по номеру и тегу²

Обратите внимание! Начиная с версии библиотеки 3.6.29 изменились имена входных параметров в JSON-структуре!

```
int pioneer3_get_fdtag(int handle, uint16 timeout, const char *params,
    const char* &response)
```

Параметры:

handle	полученное значение при инициализации сессии
timeout	таймаут ожидания ответа
params	строка с параметрами задачи
response	строка с результатом выполнения

Формат параметра params:

```
{
    "fd_doc_id": integer,
    "fd_tag": integer
}
```

Параметры:

fd_doc_id	номер документа	обяз.
fd_tag	тег (код) реквизита документа	обяз.

Возвращает: код ошибки

Блок данных в строке ответа:

```
{
    "base64": string
}
```

В параметре "base64" возвращается содержимое реквизита, упакованное в base64.

Особенности:

Команда не требует опроса статуса выполнения, запрошенные данные возвращаются сразу в строке ответа.

Список тегов указаны в [1].

²начиная с версии ККТ 2.57.68

Реквизит возвращается в формате TLV (tag-length-value), в том виде, как он хранится в ФН.

Порядок байт для значений tag, length - LE (little-endian).

tag	length	value
тег (uint16)	длина (uint16)	значение (формат зависит от тега, длина задана полем "длина")

3.3 Получить состояние ФН³

Обратите внимание! Начиная с версии библиотеки 3.6.29 изменились имена выходных параметров в JSON-структуре!

```
int pioneer3_get_fdstate(int handle, uint16 timeout, const char*
    &response)
```

Входные параметры: нет

Блок данных в строке ответа:

```
{
    "fd_last_date": integer,
    "fd_last_doc": integer,
    "fd_serial": string,
    "fd_flags": integer,
    "fd_phaze": integer,
    "shift_id": integer,
    "shift_doc_number": integer,
    "fd_shift_state": integer,
    "fd_waiting": integer,
    "fd_waiting_date": integer,
    "fd_waiting_document": integer,
    "fd_expired": integer,
    "fd_reg_left": integer,
    "fd_reg_count": integer,
    "fd_reg_document": integer,
    "fd_reg_date": integer,
    "ffd_version": integer,
    "kkt_ffd_version": integer
}
```

³начиная с версии ККТ 2.57.68

fd_last_date	дата последнего документа в формате unixtime
fd_last_doc	фискальный номер последнего документа
fd_serial	заводской номер ФН
fd_flags	флаги состояния ФН (предупреждения, битовая маска)
fd_phaze	фаза работы ФН
shift_id	номер текущей смены
shift_doc_number	номер текущего документа в смене
fd_shift_state	состояние текущей смены (0 - закрыта, 1 - открыта)
fd_waiting	количество документов, ожидающих отправки в ОФД
fd_waiting_date	дата первого документа, ожидающего отправки в ОФД в формате unixtime
fd_waiting_document	номер первого документа, ожидающего отправки в ОФД
fd_expired	дата окончания срока действия ФН в формате unixtime
fd_reg_left	количество оставшихся перерегистраций
fd_reg_count ^a	количество сделанных регистраций/перерегистраций
fd_reg_document ^a	номер документа регистрации/перерегистрации
fd_reg_date ^a	дата документа регистрации/перерегистрации
ffd_version ^a	версия формата фискальных документов
kkt_ffd_version ^a	версия формата фискальных документов ККТ

^a начиная с версии ККТ 2.58.70

Особенности:

Команда не требует опроса статуса выполнения, запрошенные данные возвращаются сразу в строке ответа.

Флаги состояния ФН перечислены в "Флаги состояния ФН".

Возможные значения фаз работы ФН перечислены в "Фазы жизни ФН".

Возможные значения версий формата фискальных документов указаны в "Версии форматов фискальных документов".

Часть III

Примеры

Часть программы на C++ для работы с ККТ

```
#include "libpioneer3_api.h"

// разбор строки json, возврат результатов выполнения задачи
void parse_response(const char *resp, int &response, int &extended) {
    // реализация разбора json строки
}

// процедура входа в режим ФР
int enter_fr(int handle, uint16_t timeout) {
    const char *resp;
    int res = pioneer3_fremu_enter(handle, timeout, resp);
    if(res!=0) {
        pioneer3_close(handle);
        return -1;
    }
    int response, extended;
    // разбор json в строке resp
    // заполнение переменных response и extended
    parse_response(resp, response, extended);
    if(response!=0) {
        pioneer3_close(handle);
        return -1;
    }
    return 0;
}

// процедура опроса ККТ о завершении задачи
int query(int handle, uint16_t timeout, uint32_t task, int count) {
    int res;
    const char *resp;
    while(count-->0) {
        res = pioneer3_query(handle, timeout, task, resp);
        if(res!=0) break;
        int response, extended;
        // разбор json в строке resp
        // заполнение переменных response и extended
        parse_response(resp, response, extended);
        if(response==0) break;
        if(!(response==3 && extended==103)) break;
        // формируем небольшую задержку, не менее 200 мс
        // sleep(200);
    }
    if(res!=0) {
        pioneer3_close(handle);
    }
}
```

```

        return -1;    // ошибка
    }
    if(response!=0) {
        pioneer3_close(handle);
        return -1;    // ошибка
    }
    return 0;
}

// процедура открытия смены
int open_shift(int handle, uint16_t timeout, const char *params) {
    uint32_t task;
    int res = pioneer3_open_shift(handle, timeout, task, params, resp);
    if(res!=0) {
        pioneer3_close(handle);
        return -1;    // ошибка
    }
    int response, extended;
    // разбор json в строке resp
    // заполнение переменных response и extended
    parse_response(resp, response, extended);
    if(response!=0) {
        pioneer3_close(handle);
        return -1;    // ошибка
    }
    res = query(handle, 5000, task, 100);
    return res;
}

// открытие документа
int open_doc(int handle, uint16_t timeout, const char *params) {
    uint32_t task;
    int res = pioneer3_open_document(handle, timeout, task, params,
        resp);
    if(res!=0) {
        pioneer3_close(handle);
        return -1;    // ошибка
    }
    int response, extended;
    // разбор json в строке resp
    // заполнение переменных response и extended
    parse_response(resp, response, extended);
    if(response!=0) {
        pioneer3_close(handle);
        return -1;    // ошибка
    }
}

```

```

        res = query(handle, 5000, task, 100);
        return res;
    }

    // добавление товара в документ
    int open_doc(int handle, uint16_t timeout, const char *params) {
        uint32_t task;
        int res = pioneer3_add_item(handle, timeout, task, params, resp);
        if(res!=0) {
            pioneer3_close(handle);
            return -1;    // ошибка
        }
        int response, extended;
        // разбор json в строке resp
        // заполнение переменных response и extended
        parse_response(resp, response, extended);
        if(response!=0) {
            pioneer3_close(handle);
            return -1;    // ошибка
        }
        res = query(handle, 5000, task, 100);
        return res;
    }

    // закрытие документа
    int close_doc(int handle, uint16_t timeout, const char *params) {
        uint32_t task;
        int res = pioneer3_close_document(handle, timeout, task, params,
            resp);
        if(res!=0) {
            pioneer3_close(handle);
            return -1;    // ошибка
        }
        int response, extended;
        // разбор json в строке resp
        // заполнение переменных response и extended
        parse_response(resp, response, extended);
        if(response!=0) {
            pioneer3_close(handle);
            return -1;    // ошибка
        }
        res = query(handle, 5000, task, 100);
        return res;
    }

    // закрытие смены

```

```

int close_shift(int handle, uint16_t timeout, const char *params) {
    uint32_t task;
    int res =pioneer3_close_shift(handle, timeout, task, params, resp);
    if(res!=0) {
        pioneer3_close(handle);
        return -1;    // ошибка
    }
    int response, extended;
    // разбор json в строке resp
    // заполнение переменных response u extended
    parse_response(resp, response, extended);
    if(response!=0) {
        pioneer3_close(handle);
        return -1;    // ошибка
    }
    res = query(handle, 5000, task, 100);
    return res;
}

// процедура выхода из режима ФР
int exit_fr(int handle, uint16_t timeout) {
    const char *resp;
    int res = pioneer3_fremu_exit(handle, timeout, resp);
    if(res!=0) {
        pioneer3_close(handle);
        return -1;
    }
    int response, extended;
    // разбор json в строке resp
    // заполнение переменных response u extended
    parse_response(resp, response, extended);
    if(response!=0) {
        pioneer3_close(handle);
        return -1;
    }
    return 0;
}

const char *resp;
// подключаемся к ККТ по адресу в строке ipaddr
int handle = pioneer3_open(ipaddr, timeout, resp);
if(handle<=0) return handle;    // ошибка
// входим в режим ФР
int res = enter_fr(handle, 10000);
if(res!=0) return res;    // ошибка
// открываем смену

```

```

char *params;
// формирование json в строке params
res = open_shift(handle, 10000, params);
if(res!=0) return res; // ошибка
// открываем документ
// формирование json в строке params
res = open_doc(handle, 10000, params);
if(res!=0) return res;
// добавляем позицию товара
// формирование json в строке params
res = add_item(handle, 10000, params);
if(res!=0) return res;
// закрываем документ
// формирование json в строке params
res = close_doc(handle, 10000, params);
if(res!=0) return res;
// закрываем смену
// формирование json в строке params
res = close_shift(handle, 10000, params);
if(res!=0) return res;
// выходим из режима ФР
res = exit_fr(handle, 10000);
if(res!=0) return res;
// завершаем сессию
pioneer3_close(handle);

```

Часть IV

Таблицы

Таблица 3.1: Коды возврата из функций библиотеки

Код	Значение
-1	Неправильный handle
-2	Ошибка подключения к ККТ
-3	Ошибка передачи данных в ККТ
-4	Таймаут получения ответа из ККТ
-5	Ошибка в размере пакета данных
-6	Неправильная контрольная сумма пакета данных
-7	Неправильный заголовок пакета данных
-8	Ошибка в параметрах команды (при разборе JSON)
-9	Некорректно сформированный JSON
-10	Некорректно сформированный адрес подключения к ККТ

Таблица 3.2: Коды ответа ККТ (поле "response" в JSON-ответе)

Код	Значение
1	Ошибка разбора пакета данных в ККТ
2	Ошибка в значениях параметров команды (JSON). Например неуставленное значение для обязательного параметра, или значение выходит за пределы допустимого диапазона.
3	Ошибка выполнения задачи. Возникает в процессе выполнения поставленной задачи. Детализацию ошибки можно посмотреть в параметре "extended".
4	Неправильный заводской номер ККТ. Возникает в случае, если сессию открывали для ККТ с одним заводским номером, а при постановке задачи отвечает ККТ с другим номером (произошла смена IP-адреса).

Таблица 3.3: Расширенные коды ответа ККТ (поле "extended" в JSON-ответе)

Код	Значение
29	Сработал датчик конца бумаги.
101	Некорректное состояние режима ФР. Возникает при постановке задачи, если ФР не находится в одном из двух режимов - ожидание или занят (см. ошибку 103). Рекомендуется дать команду "Сброс состояния ФР к начальным условиям".
103	ККТ занята выполнением задачи. Попытка поставить на выполнение задачу, при том, что предыдущая еще не завершилась.
104	Невозможно поставить на выполнение задачу. Недостаточно ресурсов для выполнения задачи. Данная ошибка практически не должна возникать.
106	Неправильное значение параметра команды. Величина параметра может удовлетворять граничным условиям, но в конкретном состоянии ККТ это значение неправильное. Например, если формировать документ с СНО, которая не зарегистрирована в ФН.
107	Смена уже открыта. Попытка открыть уже открытую смену.
108	Продолжительность смены больше 24 часов. Возникает при открытии документа.
109	Неправильная сумма оплаты. Например, попытка выплаты из кассы большей суммы, чем есть наличности.
110	Внутренняя ошибка. Логическая или аппаратная ошибка при выполнении задачи.
111	Невозможно начать выполнение задачи. Возникает при попытке открыть документ, при уже открытом документе.
112	Недостаточно денег. Сумма оплаты при закрытии документа меньше суммы документа, или оформление возврата товара на сумму, превышающую наличность в кассе ККТ.
113	Запрещено выполнение задачи. Обычно связана с правами доступа и конкретным состоянием ККТ, практически не должна возникать.
114	Смена уже закрыта. Попытка закрыть неоткрытую смену.
115	Неправильный номер задачи. Возникает при попытке запросить статус задачи, которая в данный момент не выполняется (команда «Запросить статус задачи»).
116	Данные не найдены. Возникает в запросах на импорт данных из ККТ в случае, если нет данных, соответствующих запросу.
117	Режим фискального регистратора выключен, либо попытка повторно включить режим фискального регистратора.

Таблица 3.4: Виды документов

Код	Значение
0	Документ прихода
1	Документ возврата прихода
2	Внесение (наличных в кассу)
3	Выплата (наличных из кассы)
5 ¹	Документ расхода
6 ¹	Документ возврата расхода
7 ¹	Документ коррекции прихода
8 ¹	Документ коррекции расхода
10 ²	Документ коррекции возврата прихода
11 ²	Документ коррекции возврата расхода

¹ начиная с версии ККТ 2.58.70

² начиная с версии ККТ 2.64.201

Таблица 3.5: Коды системы налогообложения

Код	Значение
1	ОСН
2	УСН доход
4	УСН доход-расход
8	ЕНВД
16	ЕСХН
32	Патент

Таблица 3.6: Коды ставки НДС

Код	Значение
1	Без налога
2	НДС 10%
3	НДС 20%
4	НДС 10/110
5	НДС 20/120
6	НДС 0%

Таблица 3.7: Виды агентов

Код	Значение
1	Банковский платежный агент
2	Банковский платежный субагент
3	Платежный агент
4	Платежный субагент
5	Поверенный
6	Комиссионер
7	Агент

Таблица 3.8: Виды агентов (битовая маска)

Код	Значение
1	Банковский платежный агент
2	Банковский платежный субагент
4	Платежный агент
8	Платежный субагент
16	Поверенный
32	Комиссионер
64	Агент

Таблица 3.9: Признаки способа расчета

Код	Значение
1	предоплата 100%
2	предоплата
3	аванс
4	полный расчет
5	частичный расчет и кредит
6	передача в кредит
7	оплата кредита

Таблица 3.10: Виды фискальных документов

Код	Значение
1	Отчет о регистрации
2	Отчет об открытии смены
3	Кассовый чек
4	Бланк строгой отчетности (БСО)
5	Отчет о закрытии смены
6	Отчет о закрытии фискального накопителя
11	Отчет об изменении параметров регистрации
21	Отчет о текущем состоянии расчетов
31	Кассовый чек коррекции
41	БСО коррекции

Таблица 3.11: Режимы работы ККТ и ФН

Код	Значение
1	Шифрование
2	Автономный режим
4	Автоматический режим
8	Применение в сфере услуг
16	Режим БСО
32	Применение в Интернет

Таблица 3.12: Расширенные режимы работы ККТ и ФН

Код	Значение
1	Принтер в автомате
2	Работа с акцизными товарами
4	Применение в сфере азартных игр
8	Применение в сфере лотерей
16	Режим АС БСО
32	Работа с маркированными товарами
64	Применение в ломбардной деятельности
128	Применение в страховой деятельности

Таблица 3.13: Причины перерегистрации

Код	Значение
1	Замена ФН
2	Смена ОФД
3	Смена реквизитов пользователя
4	Смена настроек ККТ

Таблица 3.14: Фазы жизни ФН

Код	Значение
0	Настройка
1	Готовность к фискализации
3	Открыт фискальный режим
7	Постфискальный режим (архив ФН закрыт)
15	Закончена передача фискальных данных в ОФД

Таблица 3.15: Флаги состояния ФН

Код	Значение
1	До окончания срока действия 3 дня
2	До окончания срока действия 30 дней
4	Переполнение памяти ФН (архив заполнен на 90%)
8	Превышено время ожидания ответа ОФД
128	Критическая ошибка ФН

Таблица 3.16: Виды коррекции

Код	Значение
0	Самостоятельная операция
1	По предписанию налогового органа

Таблица 3.17: Способы представления текста на печати

Код	Значение
1	Текст
2	EAN8
3	EAN13
4	Code39
5	QR-код

Таблица 3.18: Версии форматов фискальных документов

Код	Значение
1	версия 1.0
2	версия 1.05
3	версия 1.1

Таблица 3.19: Флаги контекста ККТ

Код	Значение
2	смена открыта
4	фискальное состояние ОК
8	состояние флешпамяти ОК
16	архив ФН закрыт
32	архив ФН открыт (рабочее состояние)
64	подключена ФН, незарегистрированная в ККТ
128	архив ФН закрыт и все документы переданы в ОФД
256	фискальный режим
512	некорректные версии ФФД ФН и/или ККТ
1024	критическая ошибка ККТ

Таблица 3.20: Причины перерегистрации

Код	Значение
1	Замена ФН
2	Смена ОФД
3	Смена реквизитов пользователя
4	Смена настроек ККТ

Таблица 3.21: Статус проверки КМ (битовая маска)

Код	Значение
1	КМ проверена в ОИСМ и прошла проверку
2	КМ проверена в ФН и прошла проверку
4	КМ проверена в ФН, но не прошла проверку

Литература

[1] Приложение N 2 к приказу ФНС России от 14.09.2020 г. N ЕД-7-20/662