

Общество с ограниченной ответственностью «Ремонт и обслуживание  
контрольно-кассовой техники»

УТВЕРЖДАЮ  
Генеральный директор  
ООО «РОККаТ»

А.И. Фокин

**КОНТРОЛЬНО-КАССОВАЯ ТЕХНИКА  
МЕЩЕРА-01-Ф**

РУКОВОДСТВО ПРОГРАММИСТА

## Содержание

1. Общие положения.....	5
2. Нижний уровень.....	6
3. Верхний уровень.....	25
Приложение А Пример передачи числа в формате VCD.....	51
Приложение Б Коды ошибок ККТ.....	52
Приложение В Коды ошибок ОФД.....	56
Приложение Г Коды ошибок ФН.....	57
Приложение Д Формат регистрационного номера ККТ.....	58

## 1. Общие положения

Контрольно-кассовая техника (далее — ККТ) МЕЩЕРА-01-Ф предназначена для работы при осуществлении расчетов на территории Российской Федерации в целях обеспечения интересов граждан и организаций, защиты прав потребителей, а также обеспечения установленного порядка осуществления расчетов, полноты учета выручки в организациях и у индивидуальных предпринимателей. Обмен данными между компьютером и ККТ ведётся по интерфейсу RS – 232 со скоростью 1200, 2400, 4800, 9600, 14400, 38400, 58600 и 115200 бод и с физическим кадром:

- 1 стартовый бит;
- 8 битов данных;
- 1 стоповый бит;
- без проверки на четность;
- без управления потоком;
- 3 линии (TXD, RXD, GND).

Документ состоит из двух основных частей: «Нижний уровень» и «Верхний уровень».

Под командой понимается посылка, состоящая из кода команды и данных (они могут отсутствовать), которые определяют тип действия для ККТ. На каждую команду ККТ присылает ответ, содержащий код ошибки и данные (могут отсутствовать) в зависимости от команды.

Все команды и ответы передаются по единому протоколу нижнего уровня, который обеспечивает достоверность передачи данных с помощью механизма контрольных сумм и повторов. Таким образом, в разделе «Нижний уровень» описан общий для всех команд и ответов алгоритм их передачи и приема.

В разделе «Верхний уровень» описаны конкретные коды команд, передаваемые в них данные и форматы ответов на команды, а также возможные коды ошибок и их описание.

## 2. Нижний уровень

Нижний уровень условно разделен на две части, для реализации каждой из которых используется механизм, реализуемый другой его частью:

- транспортная часть отвечает за обмен пакетами по физическому уровню;
- буфер заданий осуществляет организацию очереди получаемых заданий.

При описании будем использовать условные обозначения:

- "Name" – имя константы (значения констант приведены в подразделе «Константы»).
- Name – имя переменной или параметра. При таком формате указания имени параметра учитывается только его наличие и положение относительно других параметров, а не его значение, размерность и т.п.
- Name [X] – имя переменной или параметра с указанием его фиксированной длины в байтах. Все параметры двоичные (беззнаковые), если в описании не указано иного. Если в скобках ничего не указано (пустые скобки), то считается, что длина параметра не предопределена (определяется другими параметрами).

Все передаваемые многобайтовые значения передаются по правилу «младший байт первым».

Константы

Транспортный уровень

Обозначение	Код
STX	0xFE
ESC	0xFD
TSTX	0xEE
TESC	0xED

Ошибки

Обозначение	Код
E_Overflow	0xB1
E_AlreadyExists	0xB2
E_NotFound	0xB3
E_IllegalValue	0xB4

Статус задания

Обозначение	Код
Pending	0xA1
InProgress	0xA2
Result	0xA3

Error	0xA4
Stopped	0xA5
AsyncResult	0xA6
AsyncError	0xA7
Waiting	0xA8

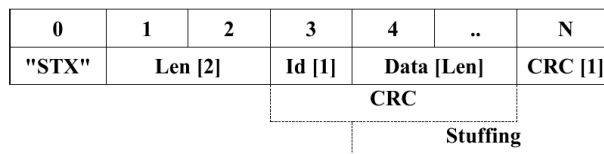
### Команды буфера

Обозначение	Код
Add	0xC1
Ack	0xC2
Req	0xC3
Abort	0xC4
AckAdd	0xC5

### Транспортная часть

#### Структура пакета данных при обмене

Обмен в обоих направлениях производится пакетами с единой структурой:



Обозначение	Описание
Byte stuffing	<p>Для того чтобы байт "STX" не встречался в полях Data и CRC, при передаче байтов этих полей применяется маскирование (Byte stuffing): если очередной байт не равен "STX" или "ESC", то он передается как есть. Если байт равен "STX", то вместо него передается пара байтов: "ESC" и "TSTX". Если байт равен "ESC", то вместо него передается пара байтов: "ESC" и "TESC". Следует заметить, что байты, равные "TSTX" и "TESC", передаются как есть (без дополнения лишними байтами). Получение сочетания "ESC" + XX, где XX не равен "TSTX" или "TESC", считать нарушением обмена – пакет отбрасывать.</p>

Обозначение	Описание
Len	<p>Количество информационных байтов в поле Data. При передаче производится маскирование, поэтому количество байтов реально передаваемых по физическому каналу может быть больше указанного значения. Чтобы само поле Len не требовало маскирования нужно учитывать:</p> <ul style="list-style-type: none"> <li>– допустимые значения 0x0000..0x7E7F,</li> <li>– в младшем байте следует передавать младшие 7! (а не 8) битов значения длины (бит 7 содержит 0);</li> <li>– а в старшем передавать остальные 8 бит.</li> </ul> <pre> unsigned char* Buf; Buf [1] = (unsigned char) Len &amp; 0x7F; Buf [2] = Len &gt;&gt; 7; //младший байт длины //старший байт длины </pre>
Id	<p>По номеру Id можно определить, ответом на какой из пакетов данных ПК является этот пакет. То есть ПК нумерует отсылаемые пакеты по своему усмотрению, а ККТ в ответных пакетах указывает, что «это ответ на такой-то пакет ПК». Допустимые значения для id, используемых ПК, 0x00..0xDF. Остальные значения зарезервированы.</p>
Data	<p>Это информационная часть пакета, которую транспортная часть протокола никак не интерпретирует (это делает вторая часть протокола – буфер заданий, подробнее смотрите в разделе «Буфер заданий»).</p>
CRC	<p>Контрольная сумма CRC8 по полям Id и Data (с учетом маскирования). Начальное значение: 0xFF (Полином: <math>x^8+x^5+x^4+1</math> / 0x31 / “CRC-8-Dallas/Maxim”).</p>

#### Логика работы

1. Первым по каналу передается байт с индексом 0 (то есть "STX").
2. ПК при формировании пакета может назначить любое допустимое значение Id.  
Рекомендуется – монотонно нарастающее.
3. ККТ, отвечая на пакет ПК, использует тот же Id.
4. Если CRC не совпадает, то пакет отбрасывается.
5. Если в середине пакета встретилось "STX", то предыдущая полученная часть пакета игнорируется (отбрасывается) и начинается получение нового пакета.
6. Если Id или Len имеют некорректное значение или ПК прислал неизвестную команду, то ККТ шлет пакет без информационной части ("STX", 0x0000, Id, CRC) для

сигнализации

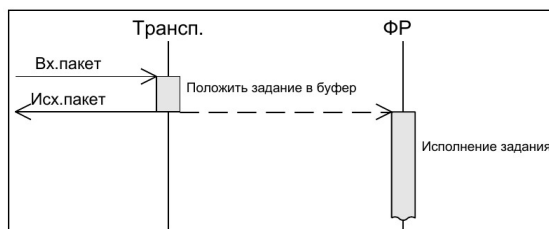
об ошибке.

7. Транспортный уровень получает пакеты в прерывании, там же их обрабатывает (не путать с «исполняет команды ККТ»).

8. Так как команда верхнего уровня в ККТ может выполняться достаточно длительное время, то во время исполнения одной команды можно передать следующую команду. Реализация транспортного уровня ведется в прерываниях (UART), а исполнение команд верхнего уровня – в основном цикле MCU ККТ. То есть считается, что в ККТ есть два потока: транспортный уровень (прием пакетов) и верхний уровень (поток ККТ).

9. Назовем команды верхнего уровня, доставленные в пакете транспортным уровнем из ПК, заданием.

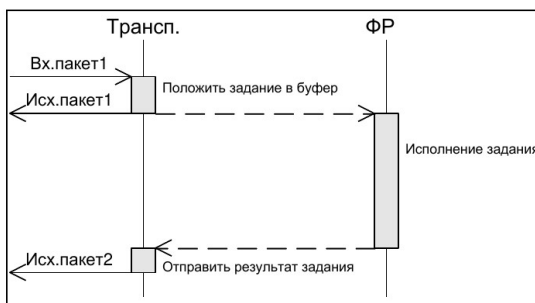
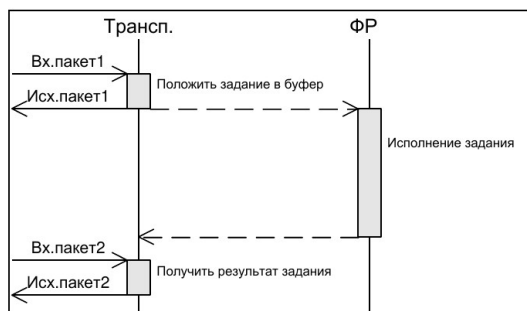
10. Транспортный уровень имеет команды Добавить задание в буфер, Получить результат исполнения задания, Очистить буфер заданий, которые исполняет в прерывании и сразу же отправляет ответный пакет (синхронно с точки зрения ПК).



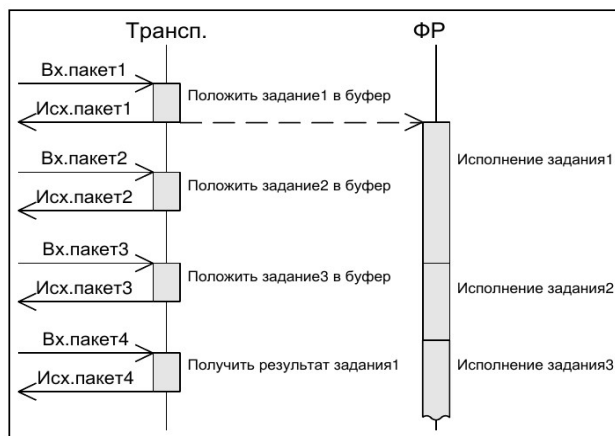
11. Полученные задания транспортный уровень складывает в буфер FIFO (в прерывании).

12. Основной поток MCU ККТ, реализующий функции ККТ, берет очередное задание из буфера, исполняет команду ККТ и складывает результат ее исполнения на место задания в буфере.

13. ПК может командой (по транспортному уровню) запросить результат исполнения задания или указать, чтобы ККТ сразу после исполнения задания отправил в ПК (асинхронно с точки зрения ПК) пакет с результатом задания.

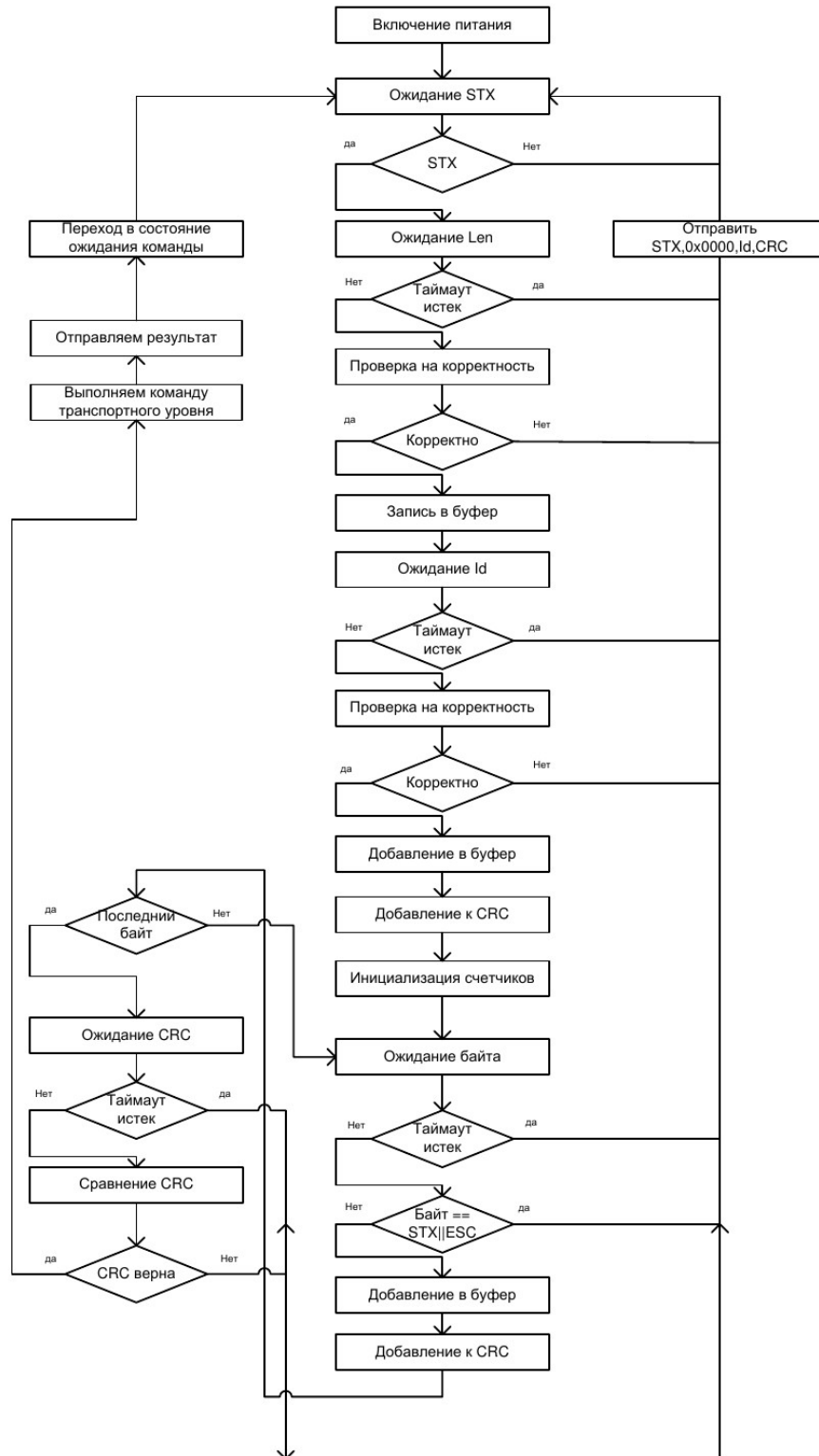


Транспортный уровень позволяет настроить «конвейер»:

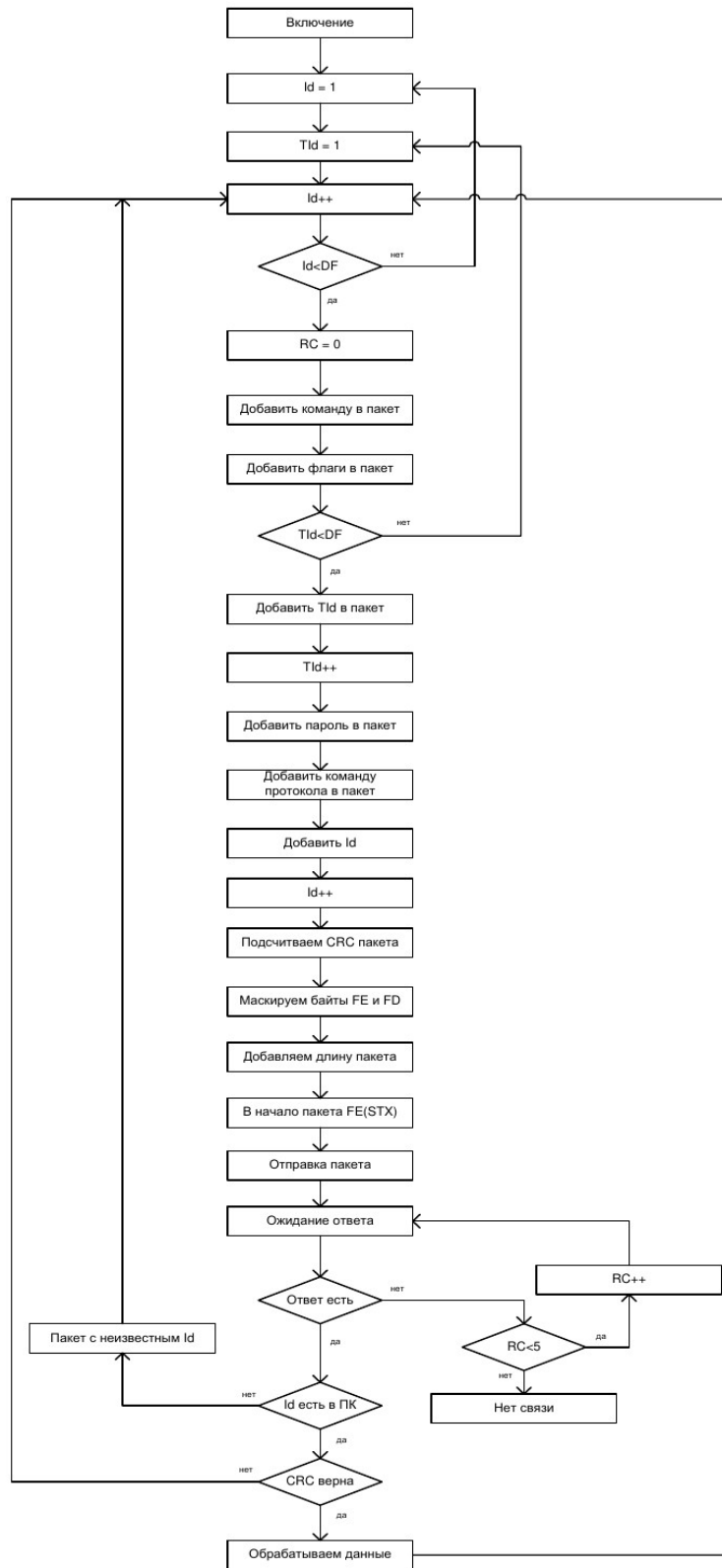




# Диаграммы состояний обмена транспортным уровнем Со стороны ККТ



# Со стороны ПК



## Буфер заданий

1. В ККТ есть буфер заданий FIFO. Его емкость не регламентируется (зависит от модели

ККТ, динамического распределения памяти ККТ и т.п.).

2. Буфер может быть в одном из состояний:

- "Complete" – в буфере нет заданий или все задания из буфера исполнены потоком ККТ.
- "Running" – одно из заданий, находящихся в буфере, исполняется потоком ККТ.
- "Error" – при исполнении одного из заданий возникла ошибка.

3. Все получаемые транспортным уровнем задания помещаются в буфер.

4. Исполнение заданий из буфера производится потоком ККТ до возникновения первой ошибки. При возникновении ошибки буфер переходит в режим "Error", исполнение заданий прекращается. Из состояния Error в состояние Complete буфер можно вывести только командой полной очистки буфера.

У задания есть признак «игнорировать ошибку». При возникновении ошибки в процессе исполнения такого задания, работа буфера не прекращается (ошибка игнорируется).

5. Задание в буфере может иметь статус:

- "Pending" – задание помещено в буфер, но до него еще не дошла очередь.
- "InProgress" – задание исполняется потоком ККТ в данный момент.
- "Waiting" – задание исполняется в фоновом режиме. ККТ ожидает данные от внешнего устройства.
- "Result" – задание исполнено потоком ККТ без ошибки.
- "Error" – задание исполнено потоком ККТ с ошибкой.
- "Stopped" – задание было в состоянии Pending на момент возникновения ошибки при исполнении одного из предшествующих в буфере заданий.

6. При добавлении задания в буфер ПК передает с заданием его однобайтовый идентификатор TId (не путать с Id транспортного уровня). По этому идентификатору ПК различает задания в буфере (при запросе статуса задания и т.п.). Допустимые значения для TId 0x00..0xDF.

7. Если задание еще не исполнилось ККТ (состояния "Pending", "InProgress", "Waiting" или "Stopped"), то в буфере хранится само задание (пакет с командой верхнего уровня).

8. Когда задание исполнилась ККТ (состояния "Result" или "Error"), то в буфере в задании уже вместо пакета с командой верхнего уровня хранится пакет с ответом верхнего уровня.

9. ПК имеет возможность запросить статус любого задания, находящегося в буфере (по идентификатору задания TId). Если задание исполнено ("Result" или "Error"), то вместе со статусом в ПК отсылается результат исполнения задания, сформированный ККТ (ответ верхнего уровня).

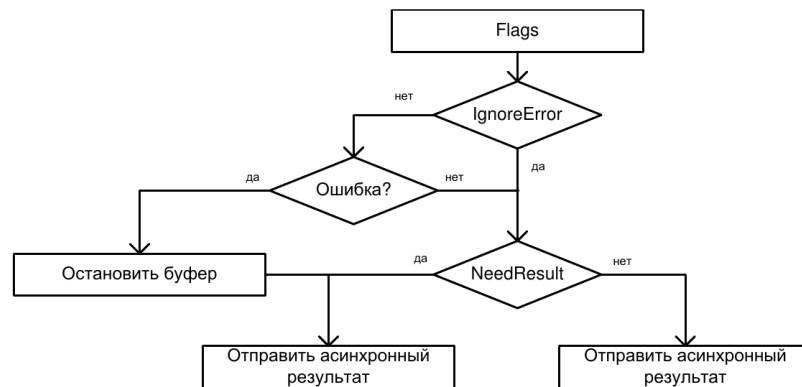
10. Каждое задание в буфере имеет флаг NeedResult. После исполнения задания с возведенным этим флагом поток ККТ отсылает в ПК асинхронно

пакет с идентификатором задания и его результатом.

11. Каждое задание в буфере имеет флаг IgnoreError. Если этот флаг установлен и задание исполнилось с ошибкой, то буфер на это реагирует так же, как если бы задание исполнилось без ошибки.

12. Каждое задание в буфере имеет флаг WaitAsyncData. Если флаг установлен, то задание может находиться в буфере в состоянии Waiting сколько угодно долго. При добавлении нового задания, задание с установленным флагом WaitAsyncData в начале буфера не препятствует добавлению нового задания (подробнее смотрите описание команды Add). В ККТ вместо хранения флага для всех заданий в буфере можно реализовать хранение задания с установленным флагом WaitAsyncData в отдельной переменной – это несколько упрощает реализацию.

13. Если при исполнении задания ККТ возникает ошибка, то ККТ отправляет в ПК асинхронно пакет с идентификатором этого задания TId и его результатом вне зависимости от флага NeedResult.



14. Все команды в первую очередь проверяют корректность значений параметров (команд работы с буфером, но не параметров самого задания). Если значение параметра недопустимое, то в ответ на команду отправляется код ошибки "E\_IllegalValue" и индекс параметра, в котором обнаружена ошибка (параметры нумеруются, начиная с 0).

Добавить задание в буфер

Команда добавления нового задания в буфер. Команда имеет битовое поле

Флаги – битовая маска:

0-й бит – флаг NeedResult, настройка передачи результата задания (0 – результат не передается, 1 – результат передается);

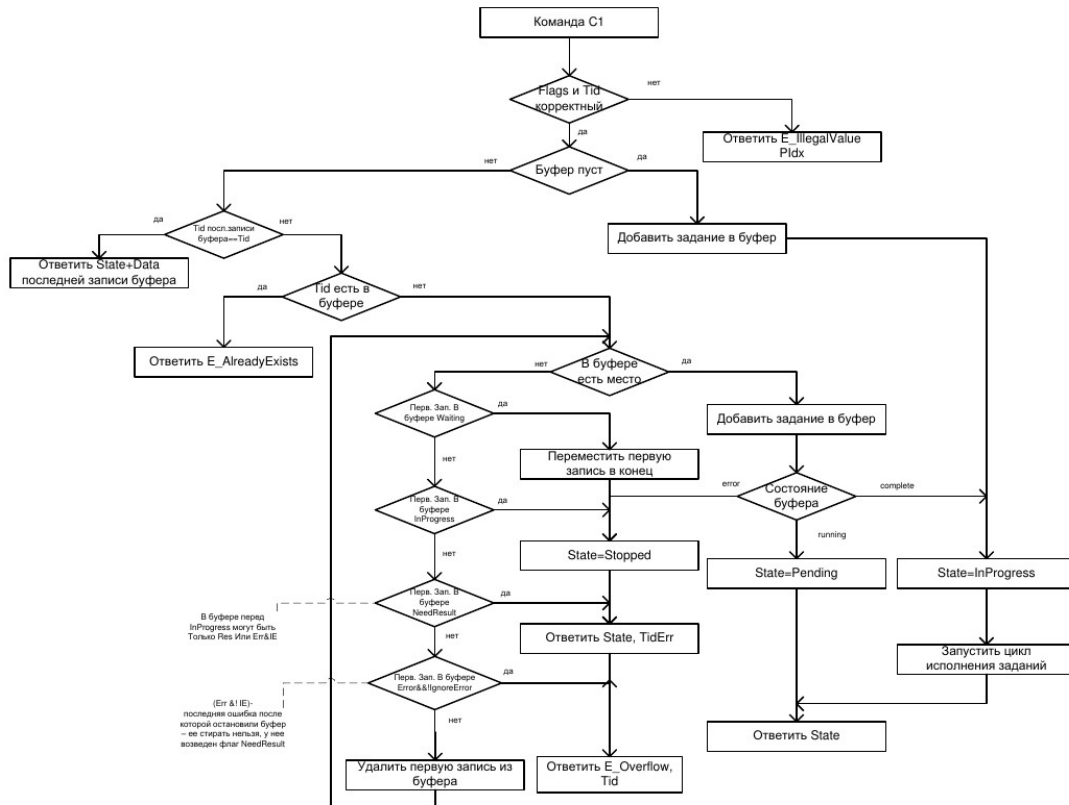
1-й бит – флаг IgnoreError, настройка работы с ошибками (0 – не игнорировать, 1 – игнорировать);

2-й бит – флаг WaitAsyncData, настройка ожидания выполнения задания (0 – сразу выполнять задание, 1 – ожидать выполнения задания сколько угодно долго, не препятствовать добавлению нового задания).

Биты 3..7 зарезервированы, для совместимости будущими версиями должны содержать 0, в случае если бит3=бит4=..= бит7≠0, то вернется "E\_IllegalValue".

Команда		Ответ	Описание
"Add", Flags [1], Tid [1], Data[]	=>		
	<=	"Pending"	Задание помещено в буфер, но пока не исполняется.
	<=	"InProgress"	Задание помещено в буфер и уже исполняется.
	<=	"Waiting"	Задание помещено в буфер и уже ожидает данных от внешнего устройства.
	<=	"Result", Data []	Задание уже находилось в буфере и ККТ успела его успешно выполнить. Data – результат, возвращенный ККТ.
	<=	"Error", Data []	Задание уже находилось в буфере, и ККТ успела его выполнить, но возникла ошибка. Data – результат, возвращенный ККТ.
	<=	"Stopped", TidErr [1]	Задание добавлено в буфер, но буфер находится в состоянии "Error" из-за ошибки, возникшей при исполнении ККТ задания TidErr.
	<=	"E_IllegalValue", PIdx [1]	Задание не может быть помещено в буфер – недопустимое значение параметра. Pidx= 0 (для Flags), PIdx = 1 (для Tid).
	<=	"E_AlreadyExists"	Задание не может быть помещено в буфер – в буфере уже есть задание с таким Tid (и это не может быть повтором).
	<=	"E_Overflow", Tid2 [1]	Задание не может быть помещено в буфер – не достаточно места в буфере, так как не удается удалить задание с Tid2 (можно

Команда	Ответ	Описание
		затереть только удачно завершённое задание с не установленным NeedResult).



## Логика работы

1. Задание добавляется в конец буфера.
2. Если нет свободного места, то затираются уже исполненные задания со сброшенным флагом NeedResult (для ПК результат не важен).
3. Затирать исполненные задания с установленным флагом NeedResult нельзя (для ПК важен результат).
4. Задание, при исполнении которого произошла ошибка, если у него установлен флаг IgnoreError (игнорировать ошибку), затирать можно. В случае если флаг не установлен, затирать нельзя (ПК может понадобиться код возникшей ошибки). Правило «Если NeedResult установлен, то затирать нельзя» тут тоже действует (см. пункт 2).
5. Затирать исполняющееся задание нельзя (возможно, что в процессе его исполнения возникнет ошибка, а Tid стерт).
6. Если в начале буфера находится задание в состоянии Waiting, то оно перемещается в конец буфера. Поскольку буфер кольцевой, это означает только изменение указателя (или указателей). В случае хранения задания с установленным флагом WaitAsyncData в отдельной переменной, данный механизм не требуется.
7. Понятия "Complete" и "Running" очень похожи. С точки зрения WinAPI

"Complete" эквивалентно WaitForSingleObject в «живом» потоке.

Механизм переповторов базируется на том факте, что при нарушении обмена в момент добавления задания *i* ПК не начинает добавлять задание *i+1* до тех пор, пока он не завершит работу с результатом добавления задания *i*.

Дополнительно существует возможность запросить состояние задания по TId

(если у задания «есть» статус, то это задание уже добавилось в буфер). Но при этом нужно учитывать то, что если будет потерян ответ E\_AlreadyExists, то запрос статуса с этим TId будет неверен. Рекомендуются уделять особое внимание назначению TId и «неполучению» ответа.

Нужно учитывать, что добавление задания в буфер, находящийся в состоянии "Error", не имеет практического смысла (это задание никогда не будет исполнено), но служит для информирования ПК о возникновении ошибки при исполнении другого задания с указанием, какого именно.

Получить состояние задания

Запрашивает статус задания и, если есть, результат его исполнения ККТ.

Команда		Ответ	Описание
"Req", TId [1]	=>		
	<=	"Pending"	Задание еще не начали исполнять.
	<=	"InProgress"	Задание в данный момент исполняется.
	<=	"Waiting"	Задание ожидает данных от внешнего устройства.
	<=	"Result", Data []	Задание исполнено успешно. Data – результат, возвращенный ККТ.
	<=	"Error", Data []	Задание исполнено с ошибкой. Data – результат, возвращенный ККТ.
	<=	"Stopped", TIdErr [1]	Задание отменено из-за ошибки, возникшей при исполнении задания TIdErr.
	<=	"E_IllegalValue", PIdx [1]	Недопустимое значение TId (PIdx = 0).
	<=	"E_NotFound"	Задания с таким TId нет в буфере.

Если задание *i* исполнено (статус "Result" или "Error") и у него установлен флаг NeedResult, то флаг NeedResult у задания *i* сбрасывается. Это позволит при следующих добавлениях заданий в буфер затереть это задание. При этом нужно учитывать ограничение: не получив (потеряв) ответ на команду "Req", нельзя

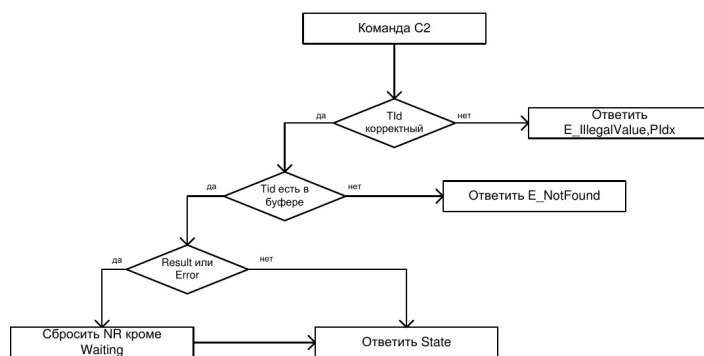
подать команду "Add", а потом опять пытаться получить "Req" этого задания – задание может быть уже стерто командой "Add".

Подтвердить получение результата

Этой командой ПК информирует ККТ о том, что ККТ может более не запрашивать данные о сохранности результатов текущего задания и всех ему предшествующих в буфере, кроме запросов с установленным флагом WaitAsyncData (нужно снять флаг NeedResult для текущего задания и всех ему предшествующих в буфере, кроме запросов с установленным флагом WaitAsyncData).

Команда		Ответ	Описание
"Ack", TId [1]	=>		
	<=	"Pending"	Задание еще не начали исполнять – команда не выполнена.
	<=	"InProgress"	Задание в данный момент исполняется – команда не выполнена.
	<=	"Waiting"	ККТ сняла флаг для задания и прекращает ждать данные от внешнего устройства. Состояние задание меняется на Result или Error на усмотрение прошивки.
	<=	"Result"	ККТ сняла флаг для успешно исполненного задания.
	<=	"Error"	ККТ сняла флаг для исполненного с ошибкой задания.
	<=	"Stopped"	Команда не выполнена, так как задание отменено из-за ошибки, возникшей при исполнении другого задания.
	<=	"E_IllegalValue", PIdx [1]	Недопустимое значение TId (PIdx = 0).
	<=	"E_NotFound"	Задания с таким TId нет в буфере.





Если указанное задание не исполнено (состояние "Pending", "InProgress", "Stopped"), то флаг не сбрасывается ни у одного из заданий, даже если предыдущие уже исполнены. Так же флаг не сбрасывается, если указан неверный или несуществующий в буфере Tid. Команда, фактически возвращает текущий статус указанного задания, но не возвращает данные (при "Result" и "Error") или Tid (при "Stopped"), как это делают "Add" и "Req".

### Очистить буфер

Команда не просто чистит буфер, но и пытается прервать текущее исполняющееся задание. Команда удаляет все задания, вне зависимости от того, успешно оно или нет и установлен ли флаг NeedResult.

Команда		Ответ	Описание
"Abort"	=>		
	<=	"Result"	Все задания удалены и прерваны.
	<=	"InProgress", Tid [1]	Все задания удалены, кроме того Tid, которое исполняется и не может быть прервано.

При текущей реализации верхнего уровня можно говорить, что "Abort" стирает из буфера все задания, имеющие любой статус, кроме "InProgress" (то есть будут стерты все выполненные и ожидающие исполнения задания, но останется исполняемое в данный момент).

### Асинхронный ответ

Пакет посылается на транспортном уровне с Id = 0xF0.

Команда		Ответ	Описание
Отсутствует			
	<=	"AsyncResult", Tid [1], Data	Задание Tid исполнено успешно. Data – результат, возвращенный ККТ.

Команда		Ответ	Описание
		[]	
	<=	"AsyncError", Tid [1], Data []	Задание Tid исполнено с ошибкой. Data – результат, возвращенный ККТ.

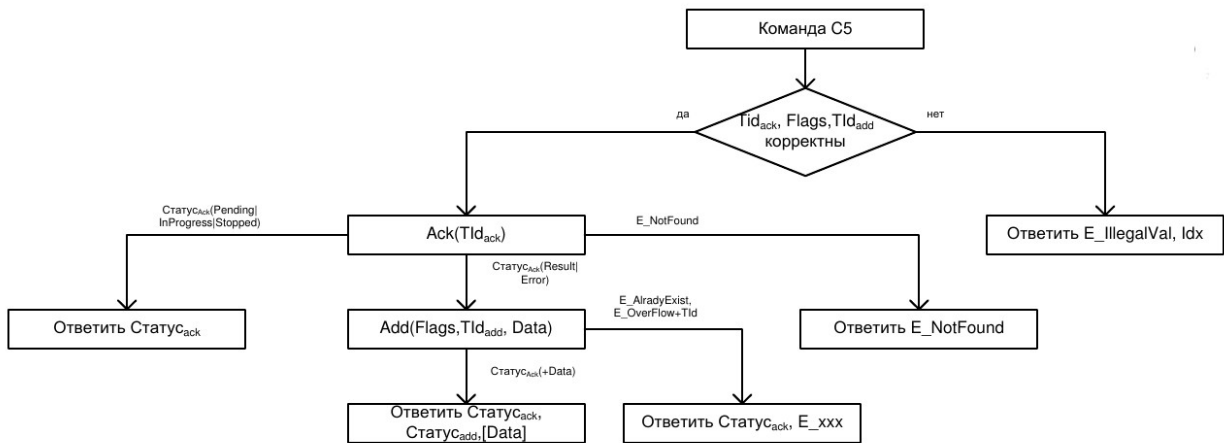
ККТ не ждет никаких пакетов от ПК в ответ на этот пакет. То есть, если ПК не получил этот пакет в течение времени ожидания, то ПК должен сам запросить статус интересующих заданий командой "Req".

Добавить задание в буфер с одновременным подтверждением

Это комбинация двух команд: "Ack" и "Add". То есть одним пакетом ПК подтверждает ККТ получение результатов выполнения предыдущих команд (чтобы ККТ больше не запрашивала результат) и добавляет новое задание в буфер.

Команда		Ответ	Описание
"AckAdd", TidAck [1], FlagsTidAdd [1], Data []	=>		
	<=	"E_IllegalValue", PIdx [1]	Ack и Add не выполнены – недопустимое значение параметра. PIdx = 0 (для TidAck), PIdx = 1 (для Flags), PIdx = 2 (для TidAdd).
	<=	"E_NotFound"	Ack не выполнен (задания с таким TidAck нет в буфере). Добавление тоже не проводилось.
	<=	"Pending"	Ack не выполнен, задание не помещено в буфер.
	<=	"InProgress"	Ack не выполнен, задание не помещено в буфер.
	<=	"Stopped"	Ack не выполнен, задание не помещено в буфер.
	<=	StateAck [1], "Pending"	StateAck = {"Result"   "Error"}. Ack выполнен удачно, задание помещено в буфер, но пока не исполняется.
	<=	StateAck [1], "InProgress"	StateAck = {"Result"   "Error"}. Ack выполнен удачно, задание помещено в буфер и уже исполняется.
	<=	StateAck [1],	StateAck = {"Result"   "Error"}. Ack

Команда	Ответ	Описание
	"Waiting"	выполнен удачно, задание помещено в буфер и ожидает данные от внешнего устройства.
	<= StateAck [1], "Result", Data []	StateAck = {"Result"   "Error"}. Ack выполнен удачно, задание уже находилось в буфере и ККТ успела его успешно выполнить. Data – результат, возвращенный ККТ.
	<= StateAck [1], "Error", Data []	StateAck = {"Result"   "Error"}. Ack выполнен удачно, задание уже находилось в буфере и ККТ успела его выполнить, но возникла ошибка. Data – результат, возвращенный ККТ.
	<= StateAck [1], "Stopped", TIdErr [1]	StateAck = {"Result"   "Error"}. Ack выполнен удачно, задание добавлено в буфер, но буфер находится в состоянии "Error" из-за ошибки, возникшей при исполнении ККТ задания TIdErr.
	<= StateAck [1], "E_AlreadyExists"	StateAck = {"Result"   "Error"}. Ack выполнен удачно, задание не может быть помещено в буфер – в буфере уже есть задание с таким TId (и это не может быть повтором).
	<= StateAck [1], "E_Overflow", TId2 [1]	StateAck = {"Result"   "Error"}. Ack выполнен удачно, задание не может быть помещено в буфер – недостаточно места в буфере, так как не удастся удалить задание с TId2 (можно затереть только удачно завершенное задание со снятым флагом NeedResult).



Ответ команды – комбинация ответов команд Ack и Add. Если ошибка возникла при выполнении Ack, то Add даже не начинает исполняться и ее ответ отсутствует, если же Ack выполнен удачно, то его ответ всегда передается вместе с результатом исполнения Add (пусть это будет даже ошибка).

Команда удобна при получении асинхронных данных от внешнего устройства (установлен флаг WaitAsyncData). То есть ПК одной командой подтверждает получение данных и сообщает ККТ, что она должна продолжать ждать.

### Типичные сценарии

#### Успешное выполнение команды

Команда		Ответ	Примечание
"Add", 0, 1, PrintString A	→		
	←	"InProgress"	
"Add", 0, 2, PrintString B	→		
	←	"Pending"	
...			
"Add", 0, N, PrintString Z	→		
	←	"Pending"	
"Add", 1, N+1, Close			Установлен флаг NeedResult
	←	"Pending"	
Ждем пакет			
	←	"AsyncResult", N+1, Data []	

### При выполнении команды возникла ошибка

Команда		Ответ	Примечание
"Add", 0, 1, PrintString A	→		
	←	"InProgress"	
"Add", 0, 2, PrintString B	→		
	←	"Pending"	
...			
"Add", 0, N, PrintString Z	→		
	←	"Stopped", 5	Ошибка в задании 5
"Req", 5			Получить код ошибки
	←	"Error", Data []	

### Переполнение очереди

Команда		Ответ	Примечание
"Add", 0, 1, PrintString A	→		
	←	"InProgress"	
"Add", 0, 2, PrintString B	→		
	←	"Pending"	
...			
"Add", 0, N, PrintString Z	→		
	←	"E_Overflow", 1	Задание 1 еще не исполнено
Ожидание T ms			
"Add", 0, N, PrintString Z	→		
	←	"E_Overflow", 1	
...			
Ожидание Tms			
"Add", 0, N, PrintString Z	→		Задание 1 исполнено, его затерли
	←	"Pending"	
...			

### Утерян пакет с данными об успешном исполнении команды

Команда		Ответ	Примечание
"Add", 1, N, Data []			Установлен флаг NeedResult
	←	"Pending"	
Ждем пакет T ms			
	X←	"AsyncResult", N, Data []	Потерялся пакет об успешном исполнении
Истекло T ms			
"Req", N	→		
	←	"Result", Data []	

### 3. Верхний уровень

#### 3.1 Общие положения

Все строки посылаются и принимаются в кодировке ср866.

Все денежные величины — целые числа, указанные в минимальной денежной единице (копейка).

Количество — целое число, указанное в граммах. При этом количество 1 единица товара (услуги) считается равной 1 килограмму (соответствует 1000).

Все команды передаются с паролем доступа — 4BCD символа.

Все числовые единицы передаются в двоично-десятичном формате, если не указано другое. Первым передаётся старший байт, последним — младший.

#### 3.2 Поддерживаемые команды

Код команды (HEX)	Команда
3Fh	Запрос состояния ККТ
43h	Скидка на чек
45h	Запрос состояния ККТ
46h	Чтение таблицы
48h	Выход из текущего режима
49h	Внесение денег
4Ah	Закрыть чек (со сдачей)
4Bh	Программирование времени
4Ch	Печать строки
4Dh	Запрос наличных
50h	Программирование таблицы
52h	Регистрация
56h	Вход в режим
57h	Возврат
58h	Получение последнего сменного итога
59h	Аннулирование всего чека
5Ah	Снятие отчёта о закрытии смены
61h	Ввод заводского номера
64h	Программирование даты
67h	Начало снятия отчета без гашения

6Bh	Технологическое обнуление ККТ
6Ch	Печать клише чека
71h	Инициализация таблиц начальными значениями
73h	Печать нижней части чека
75h	Отрезать чек
77h	Общее гашение
78h	Отмена последней скидки/надбавки
91h	Считать регистр
92h	Открыть чек
99h	Расчет по чеку
9Ah	Открыть смену
9Dh	Получение версии
A4h:	Команда обмена данными с ФН
A4h 33h	<ul style="list-style-type: none"> <li>Запрос версий ФН</li> </ul>
A4h 41h	<ul style="list-style-type: none"> <li>Запрос с печатью квитанции по номеру документа</li> </ul>
A4h 42h	<ul style="list-style-type: none"> <li>запрос (с печатью) количества документов, на которые нет квитанции</li> </ul>
A4h 43h	<ul style="list-style-type: none"> <li>печать итогов фискализации</li> </ul>
A4h 44h	<ul style="list-style-type: none"> <li>печать реквизита итогов фискализации</li> </ul>
A6h	Активизация ФН
A7h	Закрытие архива ФН
A8h	Печать итогов регистрации/перерегистрации ККТ
AVh	Печать документа по номеру из ФН
B3h	Получить последний код ошибки
B6h	Печать документа по номеру
BAh	Регистрация скидки/надбавки
BFh	Формирование реквизита
E6h	Регистрация позиции
E8h	Запись реквизита
E9h	Чтение реквизита

### 3.2 Ответ по умолчанию

Большинство команд возвращают код ошибки в формате:

<Код ответа> <Код Ошибки (1)> <0> ,



но есть несколько команд, возвращающих данные не в таком формате или не возвращающих ответа вовсе. Особенность такова, что данное отличие заметно только, когда ошибок нет. Если возникают какие-либо ошибки, то ККТ возвращает код ошибки в стандартном формате:  
<Код ответа> <Код Ошибки (1)> <0>.

### 3.3 Описание команд

После включения питания ККТ находится в состоянии «Выбор» из которого можно перейти в один из режимов:

- регистрации;
- отчетов о состоянии счетчиков ККТ;
- отчетов о закрытии смены;
- программирования;
- ввода заводского номера;
- доступа к фискальному накопителю (ФН).

Из состояния «Выбор» можно перейти в требуемый режим командой

#### **Вход в режим**

Команда:

<56h> <Режим(1)><Пароль(4)>

Ответ:

<55h><Код Ошибки(1)><0>

Команда предназначена для входа в режим работы ККТ.

Режим. Устанавливаемый режим (двоично-десятичный формат):

- Режим регистрации.
- Режим отчетов без гашения.
- Режим отчетов о закрытии смены.
- Режим программирования.
- Режим ввода заводского номера.
- Режим доступа к ФН.

Пароль. 8 двоично-десятичных символов, пароль для входа в указанный режим (все пароли программируются в таблице паролей ККТ).

Вернуться из текущего режима в состояние «Выбор» можно по команде Выход из режима.

#### **Выход из текущего режима**

Команда:

<48h>

Ответ:

<55h><Код Ошибки(1)><0>

Команда выхода из текущего режима в «надрежим», которая отменяет любое

начатое на ККТ действие (кроме открытого чека). Например, если ККТ находится в состоянии перевода времени при переходе на летнее / зимнее время, в режиме подтверждения ввода даты и т.д.

Структура режимов выглядит так:

Режим «Выбор» – Режим i – Подрежим j,

по этой причине выход из подрежима сразу в режим «Выбор» недопустим.

Данной командой нельзя выйти из состояний 1.4 (режим приема платежей по чеку) и 7.11.

### Печать строки

Команда: <4Ch><Печатаемые символы (X)>.

Ответ:

<55h><Код Ошибки(1)><(0)>.

Команда предназначена для печати последовательности символов на ЧЛ.

Таблица «Настройка соединения с ОФД»

Ряд	Поле	Назначение	Размер	Допустимые значения
1	1	Адрес ОФД	64 BIN	DEF: 0.0.0.0 Дополнено символами «пробел» до 64
	2	Порт	2 BIN	DEF: 0
	3	DNS ОФД	4 BIN	DEF: 0
	4	Канал обмена с ОФД	1 BIN	0 – Команды протокола 1 – EthernetOverUsb 2 – Ethernet 3 – Wifi 4 – GSM модем DEF: 0
	5	Интервал опроса ФН на наличие сообщений для передачи в ОФД	2 BIN	100 мс .. 60 сек DEF: 100 мс
	6	Начальное значение Url-ссылки	64 BIN	DEF: http(s)://URL/par

Чтобы измененные настройки вступили в силу нужно выйти из режима программирования.

Информацию о реквизитах для обмена с ОФД предоставляется ОФД, с которым заключен договор.

### Налоги

Для начисления налогов и передачи информации по налогам оператору фискальных

данных используются теги налогов (Поле 3 в каждом из рядов):

- 1102 – НДС итога чека со ставкой 18% значение реквизита – Сумма рассчитанного налога;
- 1103 – НДС итога чека со ставкой 10% – Сумма рассчитанного налога;
- 1104 – НДС итога чека со ставкой 0% – Сумма (часть итога) к которой применяется ставка 0;
- 1105 – НДС не облагается – Сумма (часть итога) к которой не применяется НДС;
- 1106 – НДС итога чека с рассчитанной ставкой 18% значение реквизита – Сумма рассчитанного налога;
- 1107 – НДС итога чека с рассчитанной ставкой 10% – Сумма рассчитанного налога.

### **Отрезать чек**

Команда: <75h>.

Ответ: <55h><Код Ошибки(1)><0>.

При подаче команды ККТ производит отрезку чековой ленты.

Выполнение данной команды при открытом служебном отчете невозможно.

### **Запрос состояния ККТ**

Команда: <3Fh>

Ответ: <44h><Кассир(1)> <Номер\_в\_зале(1)> <Дата\_YMD(3)>  
<Время\_HMS(3)><Флаги(1)> <Заводской\_номер(4)> <Модель(1)>  
<Версия\_ПО\_ККТ(2)><Режим\_работы(1)> <Номер\_чека(2)>  
<Номер\_смены(2)><Состояние\_чека(1)> <Сумма\_чека(5)>  
<Десятичная\_точка(1)> <Порт(1)>.

В ответ на команду ККТ возвращает информацию о текущем состоянии.

### **Запрос наличных**

Команда: <4Dh>.

Ответ: <4Dh><Сумма(7)>.

Команда запрашивает сумму наличных денег на текущий момент работы ККТ. После выполнения данной команды, ответ с кодом ошибки вернется, только когда возникла ошибка при выполнении команды, иначе возвращается ответ в указанном формате.

Сумма – сумма наличных денег в денежном ящике ККТ, число формата BCD в пределах 0000000000000000.. 99999999999999 мде.

### **Получение последнего сменного итога**

Команда: <58h>

Ответ: <55h><Код Ошибки(1)><Сменный Итог Продаж (7)><Сменный Итог Покупок (7)>

Команда запрашивает последний сменный итог продаж на текущий момент работы ККТ.

Сменный Итог Продаж – сумма текущего сменного итога продаж, число формата BCD в пределах 0000000000000000 .. 99999999999999 мде.

Сменный Итог Покупок – сумма текущего сменного итога покупок, число формата BCD в пределах 0000000000000000 .. 99999999999999 мде.

Получение версии

Команда: <9Dh>.

Ответ: <55h><Код\_ошибки (1)><Версия (1)><ПодВерсия (1)><КодЯзыка (1)><Сборка (2)>.

Команда запрашивает информацию о версии ПО источника, номере сборки и коде языка, который поддерживает данная ККТ.

Версия. Формат BCD, 00..99. Номер версии ПО Источника.

ПодВерсия. Формат BCD, 00..99. Номер подверсии ПО Источника.

КодЯзыка. Формат BCD, 00..99. Номер поддерживаемого языка (0 – русский).

Сборка. Формат VCD, 0000..9999. Номер сборки ПО Источника.  
Версия указывается в формате «А.В.С.Д.», где А – версия, В – подверсия, С – сборка, D –литерное обозначение поддерживаемого языка (например, «рус.»).  
По результатам работы этой команды можно однозначно определить с устройством какой версии осуществляется обмен.

### **Считать регистр**

Команда: <91h><Регистр (1)><Параметр1 (1)><Параметр2 (1)>.

Ответ: <55h><Код ошибки (1)><Значение (X)>.

Команда предназначена для получения значений основных операционных и денежных регистров в любой момент, а также для получения основных параметров ККТ (дата, время, версия, номер документа, сквозной номер документа и т.д.).

Регистр. Двоичное число. Допустимые значения 1..38. Значения 37..255 – недопустимые (зарезервированы для будущих версий).

Параметр1. Двоичное число (0 .. 255). Если поле Регистр не требует параметров, то поле Параметр1 должно передаваться и содержать 0.

Параметр2. Двоичное число (0 .. 255). Если поле Регистр не требует параметров, то поле Параметр2 должно передаваться и содержать 0.

Значение. В зависимости от регистра ККТ возвращает различное число байт в качестве значения регистра. Также интерпретация поля «Значение» (формат и тип поля) возлагается на хост в соответствии с Таблицей регистров.

При описании регистров использовались следующие обозначения:

ТЧ – тип чека (1 – приход, 2 – возврат прихода, 4 – расход, 5 – возврат расхода, 7 – чек коррекции: приход, , 9 – чек коррекции: расход).

ТО – тип оплаты (01h – наличные, 02h – тип оплаты 2, ... 09h – тип оплаты 9, 0Ah – тип оплаты 10).

ТСЧ – тип счетчика (1 – постоянный, 2 – сбрасываемый при технологическом обнулении).

ТС – тип скидки/надбавки, может принимать значения от 1 до 16 включительно.

ТН – тип налога, может принимать значения от 1 до 6 включительно.

ТИ – тип итога смены (0 – итог по продажам, 1 – итог по покупкам, 2 – итог по возвратам продаж, 3 – итог по возвратам покупок).

ТИП – тип источника питания (0 – внешний блок питания, 1 – батарея часов).

ТСШ – тип счетчика шагов ШД (0 – общее количество шагов ШД, 1 – количество шагов ШД в направлении «вперед»).

РС – размерность счетчика (0 – 7 VCD, 1 – 9 VCD).

НС – номер секции (1 – секция 1, 2 – секция 2, ... 16 – секция 16). Цх – целое число из диапазона от 0 до  $10^x - 1$ .

Дх.у – дробное число из диапазона 0 до  $10^x - 10^{-y}$ .

Регистр (HEX)	Параметр		Комментарий		
	1	2	Описание	Формат	Примечание
01h	ТЧ		Сумма регистраций	6 BCD (Д10.2)	
02h	ТЧ		Сумма сторно	6 BCD (Д10.2)	
03h	ТЧ	ТО	Сумма платежей	6 BCD (Д10.2)	
04h			Сумма внесений	6 BCD (Д10.2)	
05h			Сумма выплат	6 BCD (Д10.2)	
06h	ТЧ		Количество регистраций	2 BCD (Ц4)	
07h	ТЧ		Количество сторно	2 BCD (Ц4)	
08h			Количество внесений	2 BCD (Ц4)	
09h			Количество выплат	2 BCD (Ц4)	
0Ah			Наличность в кассе	7 BCD (Д12.2)	
0Bh			Знак выручки Выручка	1 BCD (Ц1) 6 BCD (Д10.2)	0-положительная, 1- отрицательная
0Ch	ТИ		Сменный итог	7 BCD (Д12.2)	
0Dh- 10h			зарезервированы		
11h			Текущая дата Текущее время	3 BCD (ДДММГГ) 3 BCD (ЧЧММСС)	
12h	ТИ		Смена открыта Дата закрытия смены Время закрытия смены	1 Bin (Ц1) 3 BCD (ДДММГГ) 3 BCD (ЧЧММСС)	
13h			Режим работы Состояние чека	1 Bin (Ц2) 1 Bin (Ц1)	0 – чек закрыт 1 – чек прихода 2 – чек возврата прихода 3 – не используется 4 – чек расхода 5 – чек возврата расхода

Регистр (HEX)	Параметр		Комментарий		
			Номер чека	2 BCD (Ц4)	6 – не используется 7 – чек коррекции: приход 9 – чек коррекции: расход Номер последнего закрытого + 1 Номер последнего закрытого + 1
			Сквозной номер документа (в рамках одной смены)	4 BCD (Ц8)	
14h			Сумма чека	5 BCD (Д8.2)	
			Остаток чека	5 BCD (Д8.2)	
			Сдача	5 BCD (Д8.2)	
15h			Номер смены	2 BCD (Ц4)	Номер последней закрытой смены
16h			Заводской номер	4 BCD (Ц7)	
17h			Модель ККТ	1 Bin (Ц2)	Фикс. 200 ASCII-формат ASCII-формат
			Версия ККТ	1 BCD (Ц2)	
			Подверсия ККТ	1 BCD (Ц2)	
18h			Не используется		
	1	2	Описание	Формат	Примечание
19h			Максимальное количество передаваемых байт пакета	2 BCD (Ц4)	
1Ah			Номер текущей перерегистрации ФН	1 BCD (Ц2)	
			Количество оставшихся перерегистраций ФН	1 BCD (Ц2)	
1Bh			Дата регистрации/ перерегистрации	3 BCD (ДДММГГ)	
1Ch			Номер ФН	5 BCD (Ц10)	
1Dh- 1Fh			Зарезервированы		
20h	ТС		Суммовой счетчик	6 BCD (Д10.2)	

Регистр (HEX)	Параметр		Комментарий		
			зарегистрированны х скидок/надбавок за смену		
21h-27h			Зарезервированы		
28h	ТЧ		Необнуляемая сумма	7 BCD (Д12.2)	
29h	ТЧ	НС	Сменный оборот налога по секциям Сумма налогов по секциям	6 BCD 6 BCD	
2Ah	ТЧ		Сумма скидок за смену Сумма надбавок за смену	6BCD 6BCD	
2Bh			Код ошибки обмена ОФД	3BCD	
2Ch			Количество не отправленных документов в ФН	3BCD	
2Dh			Дата самого раннего не отправленного документа в ФН	5BCD	Формат ДД.ММ.ГГ.ЧЧ.М М
2Eh	ТЧ		Сумма аннулирований в смене	6 BCD (Д10.2)	1 – приход, 2 – возврат прихода, 4 – расход, 5 – возврат расхода Параметр ТЧ для чеков коррекции не реализован.
2Fh			Номер ФН	16 ASCII	
30h			Дата окончания срока действия ФН	3 BCD (ДДММГГ)	



## Запрос статуса ФН

Команда: <A4h><30h>

Ответ: <55h><Состояние фазы жизни (1)><Текущий документ (1)>

<Данные документа (1)><Состояние смены (1)><Флаги предупреждения (1)>

<Дата и время (5)><Номер ФН (6)><Номер последнего ФД (4)>

Команда запрашивает текущий статус ФН.

Состояние фазы жизни. Битовый формат. Поле принимает значения, которые соответствуют статусам жизни ФН:

0 – настройка ФН;

1 – готовность к активизации;

3 – фискальный режим ФН;

7 – постфискальный режим (производится передача фискальных документов ОФД);

15 – доступ к архиву ФН.

Текущий документ. Информация о текущем документе, параметр может принимать значения:

00h – нет открытого документа;

01h – отчёт о регистрации ККТ;

02h – отчёт об открытии смены;

04h – кассовый чек;

08h – отчёт о закрытии смены;

10h – отчёт о закрытии фискального режима;

11h – бланк строкой отчетности (БСО);

12h – отчет об изменении параметров регистрации ККТ в связи с заменой ФН;

13h – отчет об изменении параметров регистрации ККТ;

14h – кассовый чек коррекции;

15h – БСО коррекции;

17h – отчет о текущем состоянии расчетов.

Данные документа: 0 – нет данных документа, 1 – получены данные документа.

Состояние смены: 0 – смена закрыта, 1 – смена открыта.

*Флаги предупреждения.* Битовое поле. Используется для информирования пользователя о ресурсах ФН:

0-й бит: 0 – замена ФН не требуется, 1 – требуется срочная замена ФН;

1-й бит: 0 – ресурс КС не исчерпан, 1 – исчерпан ресурс КС (до окончания срока действия осталось 30 дней);

2-й бит: 0 – память ФН не переполнена, 1 – память ФН переполнена (архив ФН заполнен на 90%);

3-й бит: 0 – время ожидания ответа ОФД не превышено (в режиме ожидания ответа от ОФД составляет 5 минут), 1 – превышено время ожидания ответа от ОФД;

7-й бит: 0 – нет критической ошибки ФН, 1 – критическая ошибка ФН.

Остальные биты не используются, и должны содержать 0.

*Дата и время.* Текущие дата и время.

*Номер ФН.* Номер фискального накопителя, установленного в ККТ.

Номер последнего ФД. Номер последнего фискального документа.

### **Запрос кода состояния ККТ**

Команда: <45h>.

Ответ: <55h><Режим работы(1)><Флаги (1)>.

Команда возвращает код состояния ККТ, в котором она сейчас находится.

Режим работы. Двоичное число (00h .. FFh). Младшая тетрада – режим, старшая – подрежим (формат «Подрежим.Режим»).

Флаги. 0-й (младший) бит: 0 – бумага есть, 1 – нет бумаги (в принтере чеков).

1-й бит: 0 – связь с принтером установлена, 1 – нет связи с принтером.

2-й бит: бит зарезервирован, при считывании возвращает 0.

3-й бит: 0 – нет ошибок отрезчика, 1 – ошибка отрезчика (для ККТ ошибка отрезчика возможна в случае отключения отрезчика).

4-й бит: 0 – нет ошибок принтера, 1 – восстанавливаемая ошибка принтера.

5-й и 6-й биты зарезервированы и не используются.

### **Получить последний код ошибки**

В ККТ команда выполняется в любом режиме работы, но имеет смысл (то есть коды ошибок отличны от нуля) только в режиме 7.15.

Команда: <B3h>

Ответ: <55h><Код ошибки (1)><Расширенный код ошибки (1)>

Команда является дополнительным средством диагностики ККТ. Результат сохраняется до начала печати нового документа (т.е. не сбрасывается после запроса).

### **Программирование даты**

Команда: <64h> <День(1)><Месяц(1)><Год(1)>

Ответ: <55h> <Код Ошибки(1)><0>.

Команда предназначена для ввода даты.

Если вводимая дата превышает текущую дату ККТ хотя бы на один день, то ККТ ожидает подтверждения ввода даты. Для подтверждения ввода даты необходимо повторно подать команду Программирование даты.

### **Программирование времени**

Команда: <4Bh><Час(1)><Минута(1)><Секунда(1)>

Ответ: <55h> <Код Ошибки(1)><0>.

Команда предназначена для ввода времени и может быть выполнена только при закрытой смене и только в том случае, если текущая дата ККТ меньше 01.01.2090 и больше 31.12.1999.

### **Запись реквизита**

Команда <E8h><Флаги (1)><Количество блоков (1)><Номер блока (1)>  
<Данные реквизита (X)>

Ответ: <55h><Код ошибки (1)>

Команда позволяет передать реквизит в ФН, при оформлении фискальных документов.

Существует три режима использования команды:

- Программирование реквизита фискализации (смена должна быть закрыта) на нефискальной кассе и в случае, если ФН не фискализирован, разрешено программирование всех реквизитов фискализации.
- Программирование/изменение реквизита для формирования отчета об изменении параметров регистрации без замены ФН. Если ФН фискализирован, допускается программирование некоторых разрешенных к изменению реквизитов фискализации. При изменении реквизитов необходимо обязательно перепрограммировать значение реквизита – тега 1101 (код причины перерегистрации) в соответствии с измененными данными.
- Запись реквизита в ФН при открытом чеке.

Флаги. Битовое поле: 0-й бит (младший): 0 – не выводить на печать, 1 – выводить на печать; Остальные биты не используются и должны содержать ноль.

Параметр Флаги имеет значение только при отправке первого блока данных реквизита. При последующих отправках блоков, флаг не учитывается и может содержать любое значение. Флаг печати учитывается только при записи реквизитов, которые допускается не выводить на печать – для реквизитов «Дополнительный реквизит» (тег 1084) и «Сообщение оператору» (тег 1069). Количество блоков. Количество блоков в реквизите, которое необходимо передать.

Номер блока. Бинарный формат. Порядковый номер очередного блока, нумерация блоков начинается с 0.

Данные реквизита. Данные реквизита в формате TLV. Если реквизит содержит несколько блоков, заголовков TLV (тип и длина данных) передается один раз в первом блоке. Для строковых значений используется кодировка CP 866.

Нужно обратить внимание, что для передачи реквизитов «Дополнительный реквизит» (тег 1084) и «Сообщение оператору» (тег 1069), в параметре Данные реквизита необходимо передавать структуру TLV.

### **Чтение реквизита**

Команда: <E9h><Номер реквизита (2)><Номер блока (1)>

Ответ: <55h><Код ошибки (1)><Количество блоков (1)><Данные реквизита(X)>

Команда разрешена к выполнению в любом режиме работы ККТ.

Номер реквизита. Битовый формат. Номер реквизита, равен тегу по Формату ФД, передается сначала младший байт, потом старший.

Номер блока. Битовый формат. Порядковый номер очередного блока, нумерация блоков начинается 0.

Количество блоков. Количество блоков в реквизите, которое необходимо принять ПК.

Данные реквизита. Данные реквизита в формате TLV. Если реквизит содержит несколько блоков, заголовков TLV (тип и длина данных) передается один раз в первом блоке. Для строковых значений используется кодировка CP 866. Команду нужно отправлять столько раз, сколько будет указано в параметре Количество блоков.

### **Технологическое обнуление ККТ**

Команда: <6Vh>

Ответ: <55h> <Код Ошибки(1)><0>

Команда инициализирует таблицы начальными значениями, производит общее гашение, обнуляет счетчик общих гашений и производит гашение ЭЖ (для ККТ в исполнении с ЭЖ).

Команда НЕ инициализирует ячейки таблиц 9 и 10. Чек общего гашения не печатается. Пароль доступа к ККТ при подаче этой команды не проверяется (т.е. можно передать любой пароль).

Последовательности выполнения:

Технологическое обнуление ККТ.

Цикл команд Запрос кода состояния ККТ, пока Состояние = 7.1.

Если Состояние = 0.0, то удачное завершение, но это может означать и неверное завершение (выключили и включили ККТ, а инициализация не завершилась).

### **Открыть смену**

Команда: <9Ah><Флаги (1)><Текст (X)>

Ответ: <55h><Код ошибки (1)><0>

Команда открывает смену в ККТ и работает в режиме 1.0, только если чек и смена закрыты.

Состояние ККТ и чека не меняется, а смена открывается.

Флаги. Битовое поле. ККТ проверяет только младший бит: 0 – выполнить операцию, 1 – режим проверки операции. Остальные биты не используются и должны содержать ноль.

Текст (X). Строка произвольного текста в кодировке MS-DOS 866.

Если поле имеет длину 0 (байты не переданы), то строка не печатается.

### **Открыть чек**

Команда: <92h><Флаги (1)><Тип чека (1)>

Ответ: <55h><Код ошибки (1)><0>

Команда открывает чек и работает только в режиме 1.0 при состоянии чека «Закрыт».

Флаги. Битовое поле: 0-й бит: 0 – выполнить операцию, 1 – режим проверки операции.

2-й бит: 0 – печатать чек; 1 – не печатать чек. При бит 2 !1 формируется только электронный чек, который передается в ОФД и на печать чек не выводится. Для оформления чека без печати на ЧЛ после команды 92h предварительно должен быть запрограммирован реквизит 1008 («адрес покупателя») – команда Запись

реквизита (E8h).

Остальные биты не используются и должны содержать ноль.

Тип чека: 1 – чек прихода, 2 – чек возврата продажи, 4 – чек расхода, 5 – чек возврата покупки, 7 – чек коррекции: приход, 8 – чек коррекции: возврат прихода, 9 – чек коррекции: расход, 10 – чек коррекции: возврат расхода.

Остальные значения зарезервированы.

При успешном выполнении команды состояние ККТ не меняется, а состояние чека меняется на «Открыт чек прихода / возврата прихода / расхода / возврата расхода/чек коррекции...». Команда не меняет суммы открываемого чека – она остается нулевой.

При открытии чека печатается тип чека.

При оформлении чека коррекции сумма коррекции передается с помощью команды регистрации - команда Регистрация позиции (E6h) или Регистрация (52h). Можно отменить чек или закрыть чек командой Закрыть чек (4Ah), выбрав только один вид оплаты. Повтор операции E6h или 52h, а также любые другие команды режима регистрации блокируются.

### **Регистрация**

Команда: <52h><Флаги(1)><Цена(5)><Количество(5)><Секция(1)>.

Ответ: <55h> <Код Ошибки(1)><0>.

Команда предназначена для проведения регистрации товара определенного количества по определенной цене, причем перед подачей команды регистрации нужно вывести на печать наименование товара с использованием одной из команд Печать строки 4Ch или Печать поля 87h, в которой передается наименование позиции. Если после команды 4Ch или 87h выполняется новая команда 4Ch или 87h, или команда закрытия чека 4Ah, данные напечатанной до этого строки передаются в ФН в виде дополнительного реквизита. st  
Запись в ФН данных о предыдущей регистрации выполняется в ФН при поступлении новой команды 4Ch или 87h, либо по команде закрытия чека 4Ah. Вследствие чего, после команды 4Ch или 87h становятся недоступны скидки/надбавки, отмена скидки/надбавки на предыдущую позицию.

Флаги. Битовое поле:

0-й (младший) бит: 0 – выполнить операцию, 1 – режим проверки операции (см. стр. 108);

1-й бит: 0 – проверять денежную наличность, 1 – не проверять (см. подраздел «О контроле наличности» и примечания к нему).

Остальные биты не используются и должны содержать ноль.

Цена. Цена регистрируемого товара, двоично-десятичное число, 2 знака после запятой. Диапазон равен 000000000..4294967295 мде (два знака после запятой): 0,01...42949672,95.

Количество. Регистрируемое количество товара, двоично-десятичное число 00000001..99999999 (3 знака после запятой: 0,001 ... 99999,999).

Секция. Двоично-десятичное число 00 .. 30 – секция, в которую осуществляется

регистрация.

Если Секция = 0, то регистрация произведется в 1-ю секцию, но на чеке и контрольной ленте не будут напечатаны номер и название секции.

В ККТ команда выполняется при следующих условиях:

- Чек закрыт или открыт чек прихода / расхода / возврат прихода / возврат расхода.
- Сумма чека + Цена \* Количество не более 4294967295мде.

Команда Регистрация (52h) может выполнять функцию команды возврата. С помощью этой команды возврат можно производить в различные секции, в то время как с помощью команды Возврат (57h) - возврат осуществляется только в первую секцию. Если чек закрыт, то при подаче команды Регистрация (52h) открывается чек продажи.

### **Аннулирование всего чека**

Команда: <59h>

Ответ: <55h><Код Ошибки(1)><0>

Команда аннулирует (отменяет) текущий открытый чек.

### **Внесение денег**

Команда: <49h><Флаги(1)><Сумма(5)>.

Ответ: <55h><Код Ошибки(1)><0>.

Команда предназначена учета внесенных наличных денег.

Флаги. Проверяется только младший бит: 0 – выполнить операцию, 1 – режим проверки операции (описание параметра смотрите на странице 108). Остальные биты не используются и должны содержать ноль.

Сумма. Вносимая сумма двоично-десятичное число из диапазона 000000001..4294967295 мде.

### **Выплата денег**

Команда: <4Fh><Флаги(1)><Сумма(5)>.

Ответ: <55h><Код Ошибки(1)><0>.

Команда предназначена для учета выплаченных наличных денег.

Флаги. Проверяется только младший бит: 0 – выполнить операцию, 1 – режим проверки операции. Остальные биты не используются и должны содержать ноль.

Сумма. Выплачиваемая из ККТ сумма, двоично-десятичное число: 000000001..4294967295 мде.

### **Возврат**

Команда: <57h ><Флаги(1)><Цена(5)><Количество(5)>.

Ответ: <55h> <Код Ошибки(1)><0>.

Команда предназначена для учета возврата продажи или возврата покупки.

Флаги. Битовые флаги:

0-й (младший) бит: 0 – выполнить операцию, 1 – режим проверки операции;

1-й бит: 0 – проверять денежную наличность, 1 – не проверять.

Остальные биты не используются и должны содержать ноль.

Цена. Цена возвращаемого товара, двоично-десятичное число, два знака после запятой.

Диапазон 000000000..4294967295 (2 знака после запятой: 0,01 ... 42949672,95).

Количество. Возвращаемое количество товара, двоично-десятичное число, три знака после запятой – 00000001 .. 99999999 (0,001.. 99999,999).

### **Скидка**

Команда: <43h><Флаги(1)><Область(1)><Тип(1)><Знак(1)><Размер(X)>.

Ответ: <55h> <Код Ошибки(1)><0>.

С помощью команды производится начисление скидки с различными параметрами.

Флаги. Проверяется только младший бит: 0 – выполнить операцию, 1 – режим проверки операции. Остальные биты не используются и должны содержать ноль.

Область. Проверяется только младший бит: 0 – на весь чек.

Остальные биты не используются и должны содержать ноль.

Тип. Проверяется только младший бит: 0 – процентная, 1 – суммовая.

Остальные биты не используются и должны содержать ноль.

Знак. Проверяется только младший бит: 0 – скидка. Остальные биты не используются и должны содержать ноль.

Размер. Размер скидки двоично-десятичный;

- для процентной операции  $X = 3$  (000001 .. 010000, интерпретируется как 00,01 .. 100,00%);
- для суммовой операции  $X = 5$  (0000000001.. 4294967295 мде).

Скидка «на весь чек» может быть только последней операцией в чеке, то есть после нее из команд режима регистраций можно подавать только команды

Отмена последней скидки/надбавки, Расчет по чеку, Закрытие чека,

Регистрация налога (область на весь чек), Аннулирование всего чека.

После команды Скидка «на весь чек» из режима регистрации можно подавать только Отмена последней скидки, Регистрация скидки (область на весь чек), Расчет по чеку, Закрытие чека или Аннулирование всего чека.

### **Регистрация скидки/надбавки**

Команда: ь <VAh><Флаги(1) > <Область(1) > <Тип(1) > <Сумма(5) >.

Ответ: <55h> <Код ошибки(1) >.

Команда регистрирует скидку/надбавку с различными параметрами и выполняется только при открытом регистрационном документе. Вызвать регистрацию скидки/надбавки возможно только до вызова регистрации налога (для области на регистрацию – правило действует в пределах одной регистрации).

Флаги. Проверяется только младший бит: 0 – выполнить операцию, 1 – режим проверки операции. Остальные биты не используются и должны быть равны

нулю.

Область. Используется только младший бит: 0 – на весь чек, 1 – на регистрацию.

Тип. Формат двоичный, номер типа скидки/надбавки, должен быть в пределах от 1 до 16 включительно.

Сумма. Величина скидки/надбавки, двоично-десятичный формат, может принимать значение от 0000000001 до 4294967295 мде.

Скидки/надбавки зарегистрированные таким образом в чеке должны изменять уже существующие регистры по аналогии с существующей протокольной командой Скидка/Надбавка.

Использование команды регистрации скидки/надбавки с областью на весь чек возможно только последней операцией в чеке, т.е. после неё из команд режима регистрации можно подавать только команды Регистрация Налога (область - на весь чек), Скидка/Надбавка (область – на весь чек), Регистрация скидки/надбавки (область - на весь чек), Отмена последней скидки/надбавки, Расчёт по чеку, Закрытие чека или Аннулирование всего чека.

Сумма скидки не должна превышать: в случае с областью на регистрацию – величину регистрации и текущих надбавок; в случае с областью на весь чек – сумму по чеку с учетом надбавок на весь чек.

### **Отмена последней скидки/надбавки**

Команда: <78h>

Ответ: <55h> <Код Ошибки(1)><0>.

Команда отменяет последнюю скидку/надбавку.

Команда отменяет последнюю начисленную скидку/надбавку в текущем чеке (до закрытия).

Для остальных ККТ команда отменяет последнюю начисленную скидку/надбавку в текущем чеке (до закрытия) и до выполнения любой другой операции, то есть может быть выполнена только сразу же после начисления скидки/надбавки.

### **Расчет по чеку**

Команда: <99h><Флаги (1)><Тип оплаты (1)><Сумма (5)>.

Ответ: <55h><Код ошибки (1)><Остаток (5)><Сдача (5)>.

Команда предназначена для расчета по чеку с учетом типов оплаты.

Флаги. Битовое поле. 0-й (младший) бит: 0 – выполнить операцию, 1 – режим проверки операции. Остальные биты не используются и должны содержать ноль.

Тип оплаты. Формат BCD. 01 – Наличными, 02..10 – Типом оплаты i.

Сумма. Формат BCD, 0000000000.. 4294967295. Сумма в мде, полученная указанным типом оплаты.

Остаток. Формат BCD, 0000000000.. 4294967295. Неоплаченная сумма чека в мде. Если данное поле

имеет значение ноль, то чек считается полностью оплаченным и в дальнейшем



может быть закрыт.

Сдача. Формат BCD, 0000000000.. 4294967295. Сумма сдачи в мде, начисленная в результате всех платежей по текущему чеку (с учетом этого платежа).

При возникновении какой-либо ошибки ККТ не передает остаток чека и сумму сдачи (не посылает байты полей Остаток и Сдача). Ответ в этом случае имеет вид: <55h ><Код ошибки (1)><0>.

### **Закреть чек (со сдачей)**

Команда: <4Ah> <Флаги(1)><Тип оплаты(1)><Внесенная сумма(5)>.

Ответ: <55h> <Код Ошибки(1)><0>.

Команда закрывает чек с автоматическим расчетом сдачи.

Флаги. Проверяется только младший бит: 0 – выполнить операцию, 1 – режим проверки операции. Остальные биты не используются и должны содержать ноль.

Тип оплаты. Формат BCD. 01 – Наличными, 02..10 – Типом оплаты I (наименование типов оплаты

запрограммировано в Таблице 12 «Наименование типов оплаты»).

Внесенная сумма. Формат BCD, 0000000000.. 4294967295 мде. Для чеков покупки, возврата всегда должна быть равна 0.

В случае если после закрытия чека при печати клише закончилась чековая лента, ККТ не возвращает ошибки об окончании бумаги. Необходимо проверить состояние изделия и наличие ЧЛ.

Логика работы команды зависит от режима ККТ:

Режим ККТ = 1.0 (в данном чеке не использовалась команда Расчет по чеку):

Если Внесенная сумма = 0, то сдача не начисляется (получена сумма, равная сумме чека).

Внесенная сумма не может быть меньше суммы чека (если Внесенная сумма ≠ 0).

Для начисления сдачи необходимо передать ненулевую Внесенную сумму (не менее суммы чека) и указать Тип оплаты = 1 (при оплате «нечислыми» сдача начисляться не может).

Режим ККТ = 1.4 (в данном чеке использовалась команда Расчет по чеку):

Сумма платежей, проведенных по данному чеку командами Расчет по чеку (с учетом Сторно расчета по чеку), должна быть не меньше суммы чека (Остаток = 0).

Внесенная сумма (команда Закреть чек (со сдачей)) должна быть равна 0.

Поле Тип оплаты команды Закреть чек (со сдачей) игнорируется.

### **Регистрация позиции**

Команда: <E6h><Флаги (1)><Наименование товара (64)><Цена (6)><Количество (5)><Тип (1)><Знак (1)><Размер(6)><Налог (1)><Секция (1)><ШК (16)><Резерв (1)>

Ответ: <55h><Код Ошибки(1)><0>

Команда предназначена для проведения регистрации товара определенного

количества по определенной цене.

Флаги. Битовое поле:

0-й (младший) бит: 0 – выполнить операцию, 1 – режим проверки операции;

1-й бит: 0 – проверять денежную наличность, 1 – не проверять.

Остальные биты не используются и должны содержать ноль.

Наименование товара. Строка символов в кодировке CP 866 длиной 64 .

Неиспользуемые байты заполняются бинарными нулями.

Цена. Цена регистрируемого товара, двоично-десятичное число, 2 знака после запятой. Диапазон 000000000000.. 999999999999 мде (два знака после запятой: 0,01... 9999999999,99)

Количество. Регистрируемое количество товара, двоично-десятичное число 0000000001..9999999999 (3 знака после запятой: 0,001 ... 9999999,999).

Тип. Проверяется только младший бит: 0 – процентная, 1 – суммовая.

Остальные биты не

используются и должны содержать ноль.

Знак. Проверяется только младший бит: 0 – скидка, 1 – надбавка. Остальные биты не используются и должны содержать ноль.

Размер. Размер скидки/надбавки двоично-десятичный;

- для процентной операции (000000000000..000000010000, интерпретируется как 00,00..100,00%);
- для суммовой операции (000000000000.. 999999999999 мде).

Налог. Номер ряда в Таблице «Налоговые ставки». Если значение 0, то номер налога определяется по номеру секции – номер ряда Таблицы, соответствующий данной секции, предварительно должен быть запрограммирован .

Секция. Двоично-десятичное число 00 .. 30 – секция, в которую осуществляется регистрация. Если Секция = 0, то регистрация произведется в 1-ю секцию, но на чеке не будут напечатаны номер и название секции.

ШК. Строка 16 символов в кодировке ASCII, формат EAN13. Если ШК нет, то в поле устанавливается значение 00h (бинарные нули). - не используется в текущей версии.

### **Начало снятия отчета**

Команда: <67h><Тип Отчета(1)>

Ответ: <55h> <Код Ошибки(1)><0>

Команда предназначена для печати на ЧЛ отчетов без гашения различных типов.

Тип Отчета. Формат BCD:

1 – отчет о состоянии счетчиков,

7 – отчет количеств,

2 – отчет по секциям,

3 – отчет по кассирам,

8 – служебный отчет,

5 – почасовой отчет,

9 – отчет о текущем состоянии расчетов – выполняется только при закрытой смене, электронная версия отчета отправляется ОФД.

При печати отчета по секциям (Тип Отчета = 2) поля «СКИДКИ» и «НАДБАВКИ» отражают сумму скидок и надбавок, начисленных на ВЕСЬ ЧЕК, по всем чекам. Скидки и надбавки, начисленные на последнюю операцию, учитываются в сумме по секции, к которой относилась операция.

Последовательности выполнения:

Начало снятия отчета без гашения.

Цикл команд Запрос кода состояния ККТ, пока Состояние = 2.2 (рекомендуемая частота опроса – 2 раза / сек.).

Если Состояние = 2.0,

то если бит 0 поля Флаги = 1,

то ошибка «Нет бумаги» (на остатке ленты ККТ автоматически печатается «Чек аннулирован» и отчет прерывается),

иначе если бит 1 поля Флаги = 1

то ошибка «Нет связи с принтером чека»,

иначе – удачное завершение,

иначе если бит 2 поля Флаги = 1

то ошибка «Механическая ошибка печатающего устройства»,

иначе (биты 0, 1 и 2 поля Флаги = 0) ошибка «Снятие отчета прервалось».

Если Состояние ≠ 2.0 – ошибка «Снятие отчета прервалось» и печать «Чек аннулирован» (например, выключили и включили питание ККТ (наиболее вероятно при большом периоде опроса)).

После удачного выполнения данной команды с Типом Отчета = 8 (служебный документ) ККТ должна находиться в режиме 2.3 «Формирование служебного документа».

Последовательность снятия служебного отчета отличается от описанного выше алгоритма.

Содержимое отчета формируется командами Печать строки и Печать поля.

Завершение отчета производится вызовом команды протокола Печать нижней части чека, которая дополнительно увеличивает порядковый номер служебного отчета на единицу. ККТ после этого находится в режиме 2.0. Прервать печать служебного документа возможно выполнением команды Выход из текущего режима, при этом на ЧЛ будет напечатано «Документ аннулирован».

### **Печать нижней части чека**

Команда: <73h>

Ответ:

<55h> <Код Ошибки(1)><0>

Команда предназначена для печати нижней части чека.

### **Снятие отчета о закрытии смены**

Команда: <5Ah>.

Ответ: <55h> <Код Ошибки(1)><0>.

Команда выводит на печать отчет о закрытии смены.

Последовательности выполнения:

Снятие отчета о закрытии смены.

Цикл команд Запрос кода состояния ККТ, пока Состояние = 3.2 (рекомендуемая частота опроса – 2 раза / сек.).

Если Состояние != 7.1,

    то если бит 0 поля Флаги = 1,

        то ошибка «Нет бумаги» (на остатке ленты ККТ автоматически печатается «Чек аннулирован» и отчет прерывается),

    иначе если бит 1 поля Флаги = 1

        то ошибка «Нет связи с принтером чека»,

    иначе (биты 0 и 1 поля Флаги = 0) ошибка «Снятие отчета прервалось»,

    иначе если бит 2 поля Флаги = 1

        то ошибка «Механическая ошибка печатающего устройства»,

    иначе (биты 0, 1 и 2 поля Флаги = 0) ошибка «Снятие отчета прервалось».

Цикл команд Запрос кода состояния ККТ, пока Состояние = 7.1 (рекомендуемая частота опроса – 2 раза / сек.).

После изменения состояния с 7.1 на любое другое – удачное завершение.

Как только состояние сменилось с 3.2 на 7.1 (начали гаситься операционные регистры) ККТ переходит в состояние, после которого гашение закончится, даже после выключения-включения питания или обрыва бумаги. Однако лучше дождаться изменения состояния с 7.1 на иное – ККТ закончила гашение, иначе потенциально возможна подача следующей команды до окончания гашения ККТ.

Если ФН переполнен, то ККТ не переходит в состояние 7.1: после состояния 3.2 идет состояние 3.0. при большом количестве сменных записей в ФН время снятия отчета может увеличиваться.

### **Общее гашение**

Команда: <77h>

Ответ: <55h> <Код Ошибки(1)><0>.

Команда производит обнуление счетчиков ККТ, включая счетчики отчетов, сквозного номера документа и денежную наличность, но счетчик общих гашений не обнуляется, а увеличивается на единицу. На фискализированной ККТ счетчик отчетов о закрытии смены не обнуляется.

### **Печатать документ по номеру**

Команда: <B6h><Номер(3)>

Ответ: <55h><Код Ошибки(1)><0>

Команда предназначена для печати фискального документа по его номеру. При отсутствии запрашиваемого номера документа возвращает ошибку с кодом 172 (ACh).

Номер. Формат BCD. Номер чека, который необходимо распечатать.

### **Ввод заводского номера**

Команда: <61h><Заводской номер(4)>

Ответ: <55h><Код Ошибки(1)><0>

Команда предназначена для ввода заводского номера ККТ.

Заводской номер. Заводской номер ККТ в формате BCD. Допустимы номера: 00000000..00999999 .

### **Программирование таблицы**

Команда: <50h><Таблица(1)><Ряд(2)><Поле(1)><Значение (X)>

Ответ: <55h><Код Ошибки(1)><0>

Команда предназначена для программирования таблиц – присваивания указанному полю таблицы нужного значения.

Таблица. Двоичное число, номер таблицы в ККТ.

Ряд. Двоичное число (сначала старший байт, потом младший), номер ряда в указанной таблице.

Поле. Двоичное число, номер поля в ряду таблицы.

Количество байт в параметре Значение (X) должно строго соответствовать размеру поля таблицы. При программировании строковых полей символы следует передавать в кодировке MS DOS 866 (1 байт = 1 символ, формат: двоичный).

### **Чтение таблицы**

Команда: <46h><Таблица(1)><Ряд(2)><Поле(1)>.

Ответ: <55h> <Код Ошибки(1)><Значение (X)>.

Команда предназначена для считывания данных, которые запрограммированы в таблице.

Таблица. Двоичное число, номер таблицы в ККТ.

Ряд. Двоичное число (сначала старший байт, потом младший), номер ряда в указанной таблице.

Поле. Двоичное число, номер поля в ряду таблицы.

Таблица состояний

Режим	Описание	Состояние (режим.подреж им)
Выбор	Выбор	0.0
Регистрация	Ожидание команды	1.0
	Ввод пароля	1.1
	Ожидание ввода секции	1.2
	Прием платежей	1.4
Отчеты о состоянии счетчиков	Ожидание команды	2.0
	ввод пароля	2.1
	Идёт печать отчёта о состоянии счетчиков	2.2.
	ККТ	2.3
	Формирование служебного отчета	
Отчёт о закрытии смены	Ожидание команды	3.0
	Ввод пароля	3.1
	Идёт печать отчёта о закрытии смены	3.2
	Подтверждение гашения счётчиков	3.3
	Ввод даты	3.4
	Ожидание подтверждения общего гашения счетчиков	3.5 3.6
	Идёт общее гашение	
Программирова ние	Ожидание команды	4.0
	Ввод пароля	4.1
Ввод ЗН	Ожидание команды	5.0
	Ввод пароля	5.1
	–	5.2
	Ввод данных	5.3
	Подтверждение ввода	5.4
Доступ к ФН	Ожидание команды	6.0
	Ввод пароля	6.1
	Ввод данных	6.2
	Идёт печать отчёта	6.3
Дополнительны й	Идет обнуление таблиц и гашение операционных регистров	7.1
	Выполняется етстовый прогон	7.2
	Режим ввода времени	7.3
	Режим тестов	7.4
	Ввод даты после сбоя часов	7.5
	Ввод времени после сбоя часов	7.6

Режим	Описание	Состояние (режим.подрезж им)
	Начальная инициализация ККТ	7.7
	Ожидание подтверждения обнуления таблиц	7.8
	Разные накопители памяти	7.9
	ККТ не инициализирована	7.10
	ККТ заблокирована при вводе даты, меньше даты последней записи ФН	7.11
	Блокировка при ошибке ФН	7.14

### **Активизация ФН**

Команда: <A6h>

Ответ: <55h><Код Ошибки(0)><0>

Команда вызывает активизацию ФН. После активизации ФН в составе данной ККТ, ККТ не может работать без наличия ФН.

### **Закрытие архива ФН**

Команда: <A7h>

Ответ: <55h><Код Ошибки(1)><0>.

Команда закрывает архив ФН. После закрытия архива ФН доступна только печать документов по их номеру ФН. Если на данный ФД получена квитанция ОФД, то печатается также квитанция ОФД.

Если ФД в формате TLV уже недоступен в ФН, то по данной команде печатается фискальные данные документа из архива ФН.

### **Печать итогов активизации**

Команда: <A8h>

Ответ: <55h> <Код Ошибки(1)><0>

Команда выполняет печать всех отчетов об итогах регистрации и перерегистрации ККТ, которые были выполнены на данном ФН. При этом все отчеты нумеруются, начиная с 1. Если на ФН был сформирован отчет о регистрации, первым напечатается отчет о регистрации. Если на ФН был сформирован отчет об изменении параметров регистрации в связи с заменой ФН, то отчета о регистрации не будет.

### **Печать документа по номеру**

Команда: <AVh><Номер документа (4)>.

Ответ: <55h> <Код Ошибки(1)><0>.

Команда вызывает печать документа с указанным номером фискального документа.

Номер документа. Формат BCD.



## Приложение А

### Пример передачи числа в формате BCD

1. Пусть необходимо передать число 1234567 и длина посылки — 5 BCD.
2. Дополняем число до 10 разрядов (5 байт по 2 разряда в байте): 0001234567
3. Разбиваем последовательность на пары: 00 01 23 45 67.
4. Записываем каждую пару в байты: 00h 01h 23h 45h 67h.
5. Передаем байты, начиная с 00h, затем 01h, 23h, 45h и последним 67h.

## Приложение Б

### Коды ошибок ККТ

Код (HEX)	Описание
00h	Ошибок нет
08h	Неверная цена(сумма)
0Ah	Неверное количество
0Bh	Переполнение счетчика наличности
0Ch	Невозможно сторно последней операции
0Dh	Сторно по коду невозможно
0Eh	Невозможен повтор последней операции
0Fh	Повторная скидка на операцию невозможна
10h	Скидка/надбавка на предыдущую операцию невозможна
11h	Неверный код товара
13h	Неверный формат
14h	Неверная длина
15h	ККТ заблокирована в режиме ввода даты
16h	Требуется подтверждение ввода даты
18h	Нет больше данных для передачи ПО ККТ
19h	Нет подтверждения или отмены продажи
1Ah	Отчет о закрытии смены прерван. Вход в режим невозможен.
1Eh	Вход в режим заблокирован
1Fh	Проверьте дату и время
20h	Дата и время в ККТ меньше чем в ФН
21h	Невозможно закрыть архив
3Dh	Товар не найден
3Fh	Переполнение буфера чека
40h	Недостаточное количество товара
41h	Сторируемое количество больше проданного
42h	Заблокированные товар не найден в буфере чека
43h	Данный товар не продавался в чеке, сторно невозможно
66h	Команда не реализуется в данном режиме ККТ

Код (HEX)	Описание
67h	Нет бумаги
68h	Нет связи с принтером
69h	Механическая ошибка печатающего устройства
6Ah	Неверный тип чека
6Ch	Неверный номер регистра
70h	Сумма сторно больше, чем было получено данным типом оплаты
71h	Сумма не наличных платежей превышает сумму чека
72h	Сумма платежей меньше суммы чека
75h	Переполнение суммы платежей
7Bh	Неверная величина скидки/надбавки
7Ch	Операция после скидки/надбавки невозможна
7Dh	Неверная секция
7Eh	Неверный вид оплаты
7Fh	Переполнение при умножении
80h	Операция запрещена
81h	Переполнение итога чека
82h	Открыт чек аннулирования — операция невозможна
86h	Вносимая клиентом сумма меньше суммы чека
87h	Открыт чек возврата — операция невозможна
88h	Смена превысила 24 часа
89h	Открыт чек продажи — операция невозможна
8Ah	Переполнение ФП
8Ch	Неверный пароль
8Fh	Обнуленная касса (повторное гашение невозможно)
91h	Неверный номер таблицы
92h	Неверный номер ряда
93h	Неверный номер поля
94h	Неверная дата
95h	Неверное время
96h	Сумма чека по секции меньше суммы сторно
97h	Подсчёт суммы сдачи невозможен

Код (HEX)	Описание
98h	В ККТ нет денег для выплаты
9Ah	Чек закрыт — операция невозможна
9Bh	Чек открыт — операция невозможна
9Eh	Заводской номер уже указан
A2h	Неверный номер смены
A3h	Неверный тип отчёта
A4h	Недопустимый пароль
A5h	Недопустимый заводской номер ККТ
A6h	Недопустимый регистрационный номер
A7h	Недопустимый ИНН
A8h	ККТ не фискализирована
A9h	Не задан городской номер
AAh	Нет отчетов
ABh	Режим не активирован
AFh	Отсутствуют данные в буфере ККТ
B0h	Требуется выполнение общего гашения
B2h	Невозможна отмена скидки/надбавки
B3h	Невозможно закрыть чек данным типом оплаты
BAh	Ошибка обмена с фискальным модулем
BFh	Неверные номера смен в ККТ и ФН
CBh	Больше одной регистрации в чеке
D0h	Активизация данного ФН в составе данной ККТ невозможна
D3h	Ошибка формата передачи ФН
D4h	Неверное состояние ФН
D5h	Неисправимая ошибка ФН
D6h	Ошибка КС ФН
D7h	Закончен срок эксплуатации ФН
D8h	Архив ФН переполнен
D9h	В ФН переданы неверная дата и время
DAh	В ФН нет запрошенных данных
DBh	Переполнение ФН (итог чека)
DCh	Буфер переполнен

Код (HEX)	Описание
DDh	Невозможно напечатать вторую фискальную копию
DFh	Сумма налога больше суммы регистраций по чеку и/или итога и больше суммы регистрации
E0h	Начисление налога на последнюю операцию невозможно
E1h	Неверный номер ФН
E4h	Сумма сторно налога больше суммы зарезервированного налога данного типа
E6h	Операция невозможна, недостаточно питания
E7h	Некорректное значение параметров команды ФН
E8h	Превышение размеров TLV данных ФН
E9h	Нет транспортного соединения ФН
EAh	Исчерпан ресурс КС ФН
EBh	Исчерпан ресурс хранения ФН
ECh	Сообщение от ОФД не может быть принято ФН
EFh	В ФН есть неотправленные ФД
F1h	Ошибочные флаги агента (Неправильно указан тип агента(например, касса зарегистрирована для Платежного агента а в чеке указали Банковского агента))
F2h	Не хватает реквизитов для формирования документа (если не все реквизиты переданы)
F3h	Начисление налога на последнюю операцию невозможно или неправильно передан тип налоговой системы

## Приложение В

### Коды ошибок ОФД

Код (HEX)	Описание
00h	Нет ошибок
01h	Неверный фискальный признак
02h	Неверный формат сообщения оператора
03h	Неверный номер ФД
04h	Неверный номер ФН
0Bh	Неисправимая ошибка
0Eh	Ошибка обработки документа

## Приложение Г

### Коды ошибок ФН

Код (HEX)	Описание
00h	Нет ошибок
01h	Неизвестная команда
02h	Неверное стояние ФН
03h	Ошибка ФН
04h	Ошибка КС
05h	Неисправимая ошибка
06h	Архив ФН переполнен
08h	Нет запрошенных данных
10h	Превышен размер TLV
11h	Нет транспортного соединения
12h	Исчерпан ресурс КС
20h	Сообщение от ОФД не может быть принято

## Приложение Д

### Формат регистрационного номера ККТ

Формат регистрационного номера ККТ должен:

состоять из 16 цифр (10 + 6), XXXXXXXXXXXXYYYYYY, где:

XXXXXXXXXX – порядковый номер зарегистрированной ККТ, состоящий из 10 цифр (от 0 до 9);

YYYYYY – контрольное число для проверки регистрационного номера ККТ число, состоящее из 6 цифр (от 0 до 9). Если порядковый номер регистрируемой ККТ содержит менее чем из 10 цифр, то он дополняется лидирующими нулями до длины строки в 10 цифр; обеспечивать вычисление контрольного числа регистрационного номера ККТ по алгоритму расчета контрольной суммы CRC16-CCITT (дополняется лидирующими нулями до длины строки в 6 цифр), используя следующие параметры алгоритма CRC16-CCITT:

Width = 16 bits;

Truncated polynomial = 0x1021;

Initial value = 0xFFFF;

No XOR is performed on the output CRC.

Примечание: на вход алгоритма CRC16-CCITT вводятся:

- а) регистрационный номер ККТ (дополняется лидирующими нулями до длины в 10 цифр);
- б) ИНН пользователя ККТ (дополняется лидирующими нулями до длины в 12 цифр);
- в) заводской номер ККТ (дополняется лидирующими нулями до длины в 20 цифр).