

**Смарт контракт
ICO
для блокчейна TON**

Декабрь 2019

1.0 Предназначение.

Данный смарт контракт предназначен для использования в блокчейн сети TON и позволяет провести мероприятие по сбору средств Initial Coin Offering (ICO) для различных проектов. Сбор осуществляется в криптовалюте Gram. Взамен предлагаются токены или жетоны. Если сбор прошел успешно и была собрана минимальная сумма для проекта, токены будут переведены на кошельки пользователей. Если сбор средств не удался, то пользователи смогут получить свои средства обратно.

2.0 Основные особенности смарт контракта:

2.1 SoftCup.

Смарт контракт позволяет установить минимальное количество средств, необходимое для осуществления проекта. Если softcup не будет собран, пользователи смогут вернуть свои средства обратно.

2.2. Начисление баунти стейков на кошельки пользователей с последующим обменом их на токены.

Инициатор проведения ICO может воспользоваться помощью энтузиастов еще до начала проведения ICO, и чтобы их как-то заинтересовать, может предложить баунти стейки за определенную работу - написать статью, разместить видео и т.п.

2.3 Управление смарт контрактом.

Смарт контракт управляется с помощью специальных команд, отправляемых с кошелька владельца, указанного в параметре owner_wallet на этапе настройки смарт контракта. Реализованы следующие команды:

Команда	Параметры	Действие
<i>enroll_bounty</i>	<i>адрес кошелька, стейк</i>	Зачисляет указанное количество стейков на баланс пользователя в смарт контракте
<i>finalize_ico</i>		Завершает ICO. Команда осуществляет начисление токенов на кошельки пользователей, в т.ч. владельцев баунти стейков. После чего все собранные средства переводятся на целевой кошелек, указанный в

		параметре dest_wallet при настройке смарт контракта.
--	--	--

3.0 Принцип работы:

3.1 Настройка смарт контракта.

Перед началом загрузки смарт контракта в сеть TON, его необходимо инициализировать соответствующими настройками. Для этого используются .fif скрипты в папке fif. В настройках указываются такие параметры как даты начала и окончания ICO, размер SoftCap, адрес целевого кошелька, куда будут переведены все собранные средства, а также адрес владельца смарт контракта. С этого кошелька будут производиться действия по управлению смарт контрактом, такие как начисление баунти и завершение ICO. Процедура завершения ICO осуществляет перевод токенов на адреса инвесторов, затем переводит все собранные средства на кошелек указанный в параметре dest_wallet. Обратите внимание, что к этому моменту желательно чтобы кошелек dest_wallet был инициализирован и в нем был загружен смарт контракт кошелька, например, кошелек с мульти подписью.

3.2. Основной принцип работы.

После загрузки смарт контракта в сеть блокчейн TON, он ожидает даты начала ICO. До этого момента любые входящие транзакции будут отвергнуты.

При наступлении даты начала ICO смарт контракт входит в первый этап распродажи.

На каждом этапе действуют свои скидки и предлагается ограниченное количество токенов для продажи. Чем ближе к концу распродажи, тем меньше скидки.

Вот как может выглядеть таблица этапов ICO:

Этап	Количество токенов	Цена токена в нанограмах
1	500000	30000
2	500000	40000
3	500000	60000
4	500000	70000

При покупке всех токенов, выделенных для конкретного этапа, смарт контракт автоматически переходит на следующий этап, что позволит закончить проведение ICO досрочно.

3.2.1 Поведение смарт контракта между этапами.

Когда сумма платежа превышает количество токенов, доступное для покупки на данном этапе, смарт контракт может действовать двумя способами, в зависимости от параметра *any_stage*, переданного в сообщении. Если *any_stage=1*, то смарт контракт перейдет на следующий этап и докупит токены на оставшиеся средства по цене нового этапа. Это позволит завершить ICO досрочно. Если параметр *any_stage=0*, то смарт контракт осуществит покупку всех токенов данного этапа, а неизрасходованные средства будут возвращены покупателю. По умолчанию, *any_stage = 1*, это значит, что можно сделать простой перевод средств на адрес смарт контракта, без какого-либо сообщения. На всю сумму будут зачислены токены. Если была израсходована не вся сумма, остаток будет возвращен на счет покупателя.

Данный смарт контракт позволяет осуществлять владельцем начисление баунти стейков на различные кошельки. После успешного проведения ICO (SoftCup собран) на эти кошельки будет перечислено соответствующее количество токенов. Количество токенов, причитающихся на каждый стейк станет известно по окончании ICO, когда станет известно общее количество проданных токенов. Общее количество баунти токенов определяется как процент от общего количества проданных токенов и указывается в параметрах при создании смарт контракта. Обратите внимание, что параметр *bounty_percent* представляет собой процент умноженный на 100. Т.е. если на баунти необходимо выделить 12,33% от количества проданных токенов, то в параметр *bounty_percent* нужно указать как 1233 при настройке смарт контракта.

По завершению всех этапов, либо достижении даты окончания ICO, смарт контракт переходит в финальную часть.

В финальной части, если SoftCup был собран, то все купленные токены зачисляются на счета инвесторов и собранная в ходе проведения ICO сумма, переводится на кошелек, указанный в параметре *dest_wallet*.

Если SoftCup не был собран к дате окончания ICO, то все инвесторы могут вернуть свои средства обратно. Для этого нужно просто отправить с того же кошелька небольшое количество грам, скажем 1 грам, на смарт контракт. Смарт контракт вернет средства инвестора и этот 1 грам, за вычетом платы за газ.

4.0 Вызов методов смарт контракта.

Смарт контракт позволяет вызывать определенные методы для получения текущего состояния смарт контракта любым пользователем. Для этого нужно в консольной программе *lite-client* ввести команду *runmethod*, затем указать адрес смарт контракта, название метода и, если необходимо, параметры. Поддерживаются следующие методы:

<i>Метод</i>	<i>Параметры</i>	<i>Описание</i>
current_stage	-	Возвращает информацию о текущем этапе в виде [количество токенов, цена в нанограммах] Например, [500000 1500]
start_end_date	-	Даты начала и окончания ICO в unixtimestamp Например, [1576972800 1579651200]
can_buy	-	Возвращает -1 если покупка токенов возможна и 0 если ICO не активно.
balance_of	hash кошелька	Отображает количество купленных токенов и сумму затраченных средств в нанограммах. Например, [833333 999999500] В данном случае были куплены 833333 токена из них 500000 по цене 1000 нг, и 333333 по цене 1500 нг. Покупатель перевел 1000000000 нг. Смарт контракт выполнил покупку токенов на данную сумму и остаток 500 нг вернул обратно инвестору.
bounty_stake	hash кошелька	Отображает количество баунти стейков. По завершению ICO эти стейки будут переведены в токены и зачислены на кошелек пользователя.
get_settings	-	Возвращает текущие настройки. Параметры: [token_id, [softcup, collected_funds, tokens_sold][owner_wallet, dest_wallet]bounty_percent] Например, [239 [3000000000 3500000000 2000000] [CS{Cell{00438009fbc3d615ce8c1398a8ab27ed3fe0f847d869768848e878e605dc1627a7b28d90} bits: 0..267; refs: 0..0} CS{Cell{0043800db20d7944cbeebe0a85fb64f664086403063b2b89d9fac26e8a435c28c2ccc9d0} bits: 0..267; refs: 0..0}] 3333]
get_flags	-	Возвращает флаги can_buy, is_finished, [date_start, date_end]

Преобразовать дату в unixtimestamp можно например на сайте -

<https://www.unixtimestamp.com/>

Для вызова методов `balance_of` и `bounty_stake` в качестве параметра нужно указать хеш кошелька. Используйте скрипт `addr_to_h_addr.fif` в папке `fif` для получения хеша кошелька.

5.0 Технические подробности:

При поступлении платежа, данный смарт контракт запоминает адрес, с которого был произведен платеж, сумму и количество купленных токенов. Токены переводятся на счета инвесторов по окончании ICO, если был собран SoftCup. Зачисление токенов осуществляется владельцем смарт контракта специальной транзакцией с командой `finalize_ico`. Эта транзакция также переводит все собранные смарт контрактом граммы на кошелек с мульти подписью, указанный в параметре `dest_wallet`.

Для функционирования этого смарт контракта необходимо с помощью специальной транзакции к `masterchain` зарегистрировать новую криптовалюту/токен (`config param 7`) с определенным `currency_id`, создать необходимое кол-во токенов и отправить их на адрес данного смарт контракта.