

KCY1 Token - Пълно описание на функционалността

Какво прави токенът KCY1?

KCY1 е **deflationary ERC-20 токен** на Binance Smart Chain с вградени защити срещу ботове, манипулация и злоупотреби. Токенът автоматично изгаря част от себе си при всяка транзакция, което го прави по-рядък с времето.

ОСНОВНИ ХАРАКТЕРИСТИКИ

Параметри при създаване:

- **Име:** KCY1
- **Символ:** KCY1
- **Total Supply:** 1,000,000 токена (започва с 1M, намалява с времето)
- **Decimals:** 18 (стандарт за крипто)
- **Blockchain:** Binance Smart Chain (BSC)

Разпределение при deploy:

Owner (ти):	600,000 токена (60%)
Contract (резерв):	400,000 токена (40%)

КАК РАБОТЯТ ТАКСИТЕ (Tokenomics)

При всяка обикновена транзакция:

Изпращаш:	1000 токена
└─	3% (30 токена) → ИЗГАРЯ СЕ (burn) - намалява totalSupply
└─	5% (50 токена) → ОТИВА при Owner
└─	92% (920 токена) → ПОЛУЧАВА адресатът

Exempt адреси (преференциални):

Exempt адрес изпраща:	1000 токена
└─	100% (1000 токена) → ДИРЕКТНО при получателя (БЕЗ такси!)

Кои са exempt адреси?

- Owner (ти)
 - Contract адресът
 - PancakeSwap Router/Factory (за търговия)
 - До 5 адреса, които ти избереш (exemptAddress1-5)
-

🛡️ ЗАЩИТНИ МЕХАНИЗМИ

1. 48 часа Trading Lock при старт

Какво прави:

- Първите 48 часа след deploy НИКОЙ не може да търгува
- Само Owner може да прехвърля токени

Защо е нужно:

- Ботовете не могат да изкупят всичко за първата секунда
- Имаш време да настроиш exempt адресите
- Имаш време да добавиш liquidity на PancakeSwap

Как работи:

```
javascript

// Първи 48 часа:
token.transfer() → FAIL "Trading locked for 48h"

// След 48 часа:
token.transfer() → SUCCESS ✅
```

2. Max Transaction Limit (1000 токена)

Какво прави:

- Един адрес не може да изпрати повече от 1000 токена наведнъж

Защо е нужно:

- Предотвратява "whale dump" (кит продава всичко и руши цената)
- Забавя масови продажби

Как работи:

javascript

```
// Опит за изпращане на 1001 токена:  
token.transfer(addr, 1001) → FAIL "Exceeds max transaction (1000 tokens)"  
  
// Изпращане на 1000 токена:  
token.transfer(addr, 1000) → SUCCESS ✓
```

Изключения: Exempt адресите НЯМАТ лимит

3. Max Wallet Limit (20,000 токена)

Какво прави:

- Един портфейл не може да държи повече от 20,000 токена

Зашо е нужно:

- Предотвратява китове да контролират токена
- Разпределя токените между повече хора

Как работи:

javascript

```
// Портфейлът има 19,500 токена  
// Опит да получи още 600:  
token.transfer(addr, 600) → FAIL "Recipient would exceed max wallet (20,000 tokens)"  
  
// Опит да получи 400:  
token.transfer(addr, 400) → SUCCESS ✓ (общо 19,900)
```

Изключения: Exempt адресите НЯМАТ лимит

4. Cooldown Period (2 часа между транзакции)

Какво прави:

- След като изпратиш токени, трябва да изчакаш 2 часа за следваща транзакция

Зашо е нужно:

- Предотвратява bot trading (множество бързи сделки)
- Намалява price manipulation

Как работи:

```
javascript
```

```
// 10:00 часа - първи трансфер:  
token.transfer(addr1, 500) → SUCCESS ✓
```

```
// 10:30 часа - втори трансфер (30 минути по-късно):  
token.transfer(addr2, 500) → FAIL "Must wait 2 hours between transactions"
```

```
// 12:01 часа - трети трансфер (2+ часа по-късно):  
token.transfer(addr2, 500) → SUCCESS ✓
```

Изключения: Exempt адресите НЯМАТ cooldown

5. Emergency Pause (48 часа блокиране)

Какво прави:

- Owner може да блокира ВСИЧКИ трансфери за 48 часа
- След 48 часа автоматично се отблокира

Защо е нужно:

- При hack или критичен бъг
- При масова bot атака
- Emergency stop button

Как работи:

```
javascript
```

```
// Owner активира pause:  
token.pause() → SUCCESS  
  
// Всички трансфери са блокирани:  
token.transfer() → FAIL "Contract is paused"  
  
// След 48 часа автоматично се активира:  
token.transfer() → SUCCESS ✓
```

⚠ ВАЖНО: НЕ МОЖЕ да се отмени предсрочно! Ако активираш pause, ще трае пълните 48 часа.

6. Blacklist System (блокиране на измамници)

Какво прави:

- Owner може да блокира адреси от изпращане/получаване на токени

Защо е нужно:

- Спиране на scammers
- Блокиране на засечени ботове
- Защита на community-то

Как работи:

javascript

```
// Единичен blacklist:  
token.setBlacklist("0xScammerAddress", true) → Блокиран ✓
```

```
// Ограничение за трансфер от blacklist адрес:  
token.transfer() → FAIL "Sender is blacklisted"
```

```
// Масов blacklist (много адреси наведнъж):  
token.setBlacklistBatch([addr1, addr2, addr3], true) → Всички блокирани ✓
```

```
// Премахване от blacklist:  
token.setBlacklist("0xScammerAddress", false) → Отблокиран
```

⚠️ Ограничения:

- НЕ МОЖЕ да blacklist-неш Owner
- НЕ МОЖЕ да blacklist-неш contract адреса

🎯 EXEMPT АДРЕСИ (Преференциални условия)

Какво означава да си "Exempt"?

Exempt адрес има ПЪЛНА СВОБОДА:

- ✓ БЕЗ такси (0% burn, 0% owner fee)
- ✓ БЕЗ max transaction limit
- ✓ БЕЗ max wallet limit
- ✓ БЕЗ cooldown период
- ✓ Може да търгува дори първите 48 часа

Кои са автоматично exempt?

1. Owner - ти като създател

2. Contract - самият токен contract

Кои могат да станат exempt?

До 7 допълнителни адреса:

- `exemptAddress1` - Избиращ ти
- `exemptAddress2` - Избиращ ти
- `exemptAddress3` - Избиращ ти
- `exemptAddress4` - Избиращ ти
- `exemptAddress5` - Избиращ ти
- `pancakeswapRouter` - PancakeSwap Router
- `pancakeswapFactory` - PancakeSwap Factory

Защо PancakeSwap е exempt?

PancakeSwap е децентрализирана борса (DEX) където хората купуват/продават токена.

Ако НЕ беше exempt:

Човек иска да купи 1000 KCY1 за 0.1 BNB

- └ PancakeSwap изпраща 1000 токена
- └ 3% се изгаря (30 токена)
- └ 5% отива при Owner (50 токена)
- └ Човекът получава само 920 токена 

Понеже PancakeSwap Е exempt:

Човек иска да купи 1000 KCY1 за 0.1 BNB

- └ Човекът получава пълните 1000 токена 

LOCK МЕХАНИЗЪМ ЗА EXEMPT АДРЕСИТЕ

Защо е нужен Lock?

Проблем без Lock:

- Owner може да промени exempt адресите по всяко време
- Community няма доверие - "Какво ако махне PancakeSwap от exempt утре?"
- Токенът не е наистина децентрализиран

Решение с Lock:

- След натискане на "Lock" НИКОЙ не може да променя exempt адресите
- Дори Owner не може
- Токенът става напълно децентрализиран

Как работи?

ФАЗА 1: Unlocked (настройка)

javascript

// Можеши да променяши exempt адресите КОЛКОТО ПЪТИ ИСКАШ:

```
setExemptAddresses(
  [addr1, addr2, addr3, 0x0, 0x0],
  pancakeRouter,
  pancakeFactory
) → SUCCESS ✓
```

// Тестваш...

// Нещо не е наред? Променяши!

```
setExemptAddresses(
  [NEW_addr1, NEW_addr2, NEW_addr3, 0x0, 0x0],
  pancakeRouter,
  pancakeFactory
) → SUCCESS ✓
```

// Можеши да правиш това БЕЗКРАЙНО пъти

ФАЗА 2: Locked (production)

javascript

// Когато си 100% сигурен в адресите:

lockExemptAddresses() → LOCK АКТИВИРАН 

// Опит за промяна:

setExemptAddresses([...]) → FAIL "Exempt addresses are locked forever"

// Дори Owner не може да промени вече!

КРИТИЧНО ВАЖНО:

Lock е НЕОБРАТИМ и ПЕРМАНЕНТЕН!

- След натискане на `lockExemptAddresses()` НЕ МОЖЕ да се върне назад
- Exempt адресите са фиксирани ЗАВИНАГИ
- Дори ти като Owner не можеш да ги промениш
- Това е едностранна врата - няма връщане назад

Кога да активираш Lock?

- Тествал си всичко многократно
- Exempt адресите са 100% правилни
- PancakeSwap адресите са верни
- Готов си за production launch
- Искаш да покажеш на community-то че токенът е децентрализиран

ВСИЧКИ ФУНКЦИИ

За всички потребители (public):

Функция	Параметри	Описание
<code>transfer(to, amount)</code>	адрес, количество	Изпраща токени
<code>transferFrom(from, to, amount)</code>	от адрес, до адрес, количество	Изпраща от одобрен адрес
<code>approve(spender, amount)</code>	адрес, количество	Одобрява изразходване
<code>balanceOf(address)</code>	адрес	Проверява баланс
<code>allowance(owner, spender)</code>	собственик, харчещ	Проверява одобрение

Само за Owner:

Функция	Параметри	Описание
setExemptAddresses()	[адреси], router, factory	Задава exempt адреси
lockExemptAddresses()	-	Блокира exempt адресите ЗАВИНАГИ
pause()	-	Активира 48ч пауза
setBlacklist(addr, status)	адрес, true/false	Blacklist на адрес
setBlacklistBatch(addrs, status)	[адреси], true/false	Масов blacklist
burn(amount)	количество	Ръчно изгаряне
withdrawCirculationTokens(amount)	количество	Изтегля от contract
rescueTokens(token, amount)	токен адрес, количество	Спасява чужди токени
withdrawBNB()	-	Изтегля BNB от contract

View функции (само четене, бесплатни):

Функция	Връща	Описание
name()	string	Връща "KCY1"
symbol()	string	Връща "KCY1"
decimals()	uint8	Връща 18
totalSupply()	uint256	Общо токени в обръщение
owner()	address	Адресът на собственика
isExemptAddress(addr)	bool	Дали адресът е exempt
isTradingEnabled()	bool	Дали търговията е активна
timeUntilTradingEnabled()	uint256	Секунди до активиране
isPaused()	bool	Дали е в пауза
timeUntilUnpaused()	uint256	Секунди до край на пауза
exemptAddressesLocked()	bool	Дали exempt адресите са lock-нати
getExemptAddresses()	tuple	Всички exempt адреси + lock статус
isBlacklisted(addr)	bool	Дали адресът е blacklist-нат
lastTransactionTime(addr)	uint256	Кога е била последната транзакция

ТИПИЧЕН РАБОТЕН ПРОЦЕС

День 0 - Deploy:

1. Deploy contract-а на BSC
2. Owner получава 600,000 токена
3. Contract държи 400,000 токена
4. Trading е БЛОКИРАНА за 48 часа
5. Exempt адресите са празни

Първи 48 часа - Настройка:

1. Задаваш exempt адресите:
setExemptAddresses([твой адрес 1, твой адрес 2, 0x0, 0x0, 0x0], router, factory)
2. Добавяш liquidity на PancakeSwap:
 - 100,000 KCY1
 - 1 BNB
3. Тестваш функциите:
 - Transfer между exempt адреси → Без такси
 - Transfer между обикновени → С такси
 - Max limits → Работят
 - Cooldown → Работи
4. Ако нещо не е наред → променяш exempt адресите отново
5. Всичко е OK → LOCK exempt адресите

След 48 часа - Launch:

1. Trading се активира автоматично
2. Хората могат да купуват на PancakeSwap
3. Мониторинг за ботове → blacklist ако трябва
4. При нужда от emergency → pause()

Ongoing - Управление:

- Проверка за suspicious transactions
- Blacklist на scammers
- Изгаряне на токени (ако искаш да намалиш supply)
- Мониторинг на health на токена

💡 ПОЛЕЗНИ СЦЕНАРИИ

Сценарий 1: Масова bot атака

Проблем: 50 bot адреса купуват токени едновременно

Решение:

1. Забележи подозрителната активност
2. Запиши bot адресите
3. Извикай: setBlacklistBatch([bot1, bot2, bot3, ...], true)
4. Ботовете са блокирани 

Сценарий 2: Security breach

Проблем: Открит критичен бъг в contract-a

Решение:

1. Извикай: pause()
2. Всички трансфери спират за 48 часа
3. Имаш време да комуникираш с community
4. След 48 часа автоматично се възстановява

Сценарий 3: Whale dump

Проблем: Кит иска да продаде 50,000 токена

Заштита:

- Max transaction = 1000 токена
- Cooldown = 2 часа
- Резултат: Трябват му 50 транзакции x 2 часа = 100 часа (4+ дни)
- Има време пазарът да се адаптира 

Сценарий 4: Грешно изпратени токени

Проблем: Някой изпраща 5000 USDT на contract адреса по грешка

Решение:

1. Извикай: rescueTokens(USDT_address, 5000)
2. USDT се връщат в Owner портфейла
3. Можеш да ги върнеш на изпращача

⚠ ОГРАНИЧЕНИЯ И РИСКОВЕ

Това което токенът НЕ МОЖЕ:

- ✗ НЕ МОЖЕ да добавя нови токени (no minting)
- ✗ НЕ МОЖЕ да променя Owner (immutable)
- ✗ НЕ МОЖЕ да отменя pause предсрочно
- ✗ НЕ МОЖЕ да променя exempt адреси след lock
- ✗ НЕ МОЖЕ да променя такси след deploy (3% + 5% са фиксираны)
- ✗ НЕ МОЖЕ да променя лимити след deploy (1000/20,000 са фиксираны)

Потенциални рискове:

- ⚠ Ако загубиш Owner private key → не можеш да управляваш токена
 - ⚠ Ако lock-неш грешни exempt адреси → не можеш да ги поправиш
 - ⚠ Ако зададеш грешен PancakeSwap router → DEX няма да работи правилно
 - ⚠ Pause блокира ВСИЧКИ (дори Owner) за 48 часа
-

ДЪЛГОСРОЧНА ВИЗИЯ

Месец 1: Launch

- Deploy и настройка
- Добавяне на liquidity
- Първи потребители
- Тестване на защитите

Месец 2-3: Стабилизация

- Мониторинг и blacklist на ботове
- Lock на exempt адресите (децентрализация)
- Growing community

Месец 6+: Mature токен

- Напълно децентрализиран (exempt са lock-нати)
 - Deflationary ефект се усеща (supply намалява)
 - Стабилна търговия на PancakeSwap
 - Community ownership
-

ТЕРМИНОЛОГИЯ

- **Deflationary:** Токен, който намалява с времето (изгаря се)
 - **Burn:** Изгаряне - токените се изтриват завинаги
 - **Exempt:** Адрес без такси и ограничения
 - **Whale:** Адрес с много токени (потенциална заплаха)
 - **DEX:** Decentralized Exchange (PancakeSwap)
 - **Liquidity Pool:** Резервоар с токени за търговия
 - **Lock:** Перманентно блокиране на промени
 - **Cooldown:** Период на изчакване между действия
 - **BSC:** Binance Smart Chain
-

Версия: 1.0

Дата: 2024

Статус: Production Ready 