

## 关于3M系统开发？3M互助系统开发源码Demo详情

|      |                           |
|------|---------------------------|
| 产品名称 | 关于3M系统开发？3M互助系统开发源码Demo详情 |
| 公司名称 | 广州杰肯狸网络科技有限公司             |
| 价格   | .00/件                     |
| 规格参数 |                           |
| 公司地址 | 广州天河区中山大道                 |
| 联系电话 | 18125913365 19927739756   |

### 产品详情

关于超级MMM（3M）互助系统，它是位于以太坊\*\*\*上一个特定地址的一系列代码（函数）和数据状态。通过这种智能合约，用户可以在\*\*\*上实现安全、透明和高效的投资和收益分配，提高了【系统176搭建·0206-可电可微5616-】互助和投资活动的可靠性和公正性。3M互助是一种三倍收益的互助模式，用户可以将资金投入智能合约中，通过三倍的复利机制来获取收益。3M互助和MMMBSC互助复利智能合约是一种基于\*\*\*技术的智能合约，可以实现复利互助和收益分配的功能。

通过这种智能合约，用户可以在\*\*\*上安全地进行互助和投资活动，获得更高的收益和更可靠的保障。

通过与MMMBSC代币进行交互，实现投资和收益分配的功能。在构造函数中，我们需要传入MMMBSC代币的地址和受益人地址。在invest函数中，用户可以将MMMBSC代币转入智能合约中进行投资。投资金额必须大于0，投资金额将被添加到智能合约中的余额中，同时计算出10%的奖励，并将奖励转账给受益人地址。\*后，将原始投资金额和奖励金额合并，再将总金额转入智能合约中。如果转账成功，则将原余额转账给受益人。

MMMBSC互助示例：

```
scssCopy codepragma solidity ^0.8.0;
```

```
interface MMMBSC {

function transferFrom(address sender, address recipient, uint256 amount) external returns (bool);

}

uint256 totalAmount = balance + amount;

uint256 reward = (totalAmount * 10) / 100;

mmmBsc.transferFrom(msg.sender, address(this), amount);

contract MMMBSCInvestment {

MMMBSC public mmmBsc;

address payable public beneficiary;

constructor(address _mmmbscAddress, address payable _beneficiary) {

mmmBsc = MMMBSC(_mmmbscAddress);

beneficiary = _beneficiary;

}
```

```
function invest(uint256 amount) public {

require(amount > 0, "Investment amount must be greater than 0");

uint256 balance = mmmBsc.balanceOf(address(this));

function balanceOf(address account) external view returns (uint256);

mmmBsc.transfer(beneficiary, reward);

uint256 newBalance = mmmBsc.balanceOf(address(this));

require(newBalance >= totalAmount, "Investment failed");

beneficiary.transfer(balance);

}

}
```