# 以太坊erc20Token合约发布(上币)教程

## MetaMask钱包下载及配置

* MetaMask钱包插件下载,chrome或firefox浏览器应用商店下载安装

|  |
| --- |
| Chrome metamask 插件    Firefox metamask 插件 |

* MetaMask的配置

|  |
| --- |
| 打开metamask插件,一直点Accept就可以了，设置登录密码,然后点create创建账户  IMG_256  创建好之后，MetaMask会默认为用户创建12个英文助记词，这些助记词一定要保存好，建议复制保存到安全的地方，这个是确认钱包账户所有者的凭证，在其他钱包导入这个新创建的账户的时候或者修改免的时候有可能需要用到这些助记词。  IMG_256  MetaMask会自动为用户创建了一个钱包地址    点击钱包首页左上角绿色的“Main Network”，可以选择钱包使用的网络。(在此处我们选择以太坊的测试网络Ropsten)如果你没有余额请点击购买buy，进入的网站可以送一些测试以太币给你。 |

## 开发代币

* ERC20 Token

|  |
| --- |
| 也许你经常看到ERC20和代币一同出现， ERC20是以太坊定义的一个代币标准。  要求我们在实现代币的时候必须要遵守的协议，如指定代币名称、总量、实现代币交易函数等，只有支持了协议才能被以太坊钱包支持 |

* 编写代币合约代码

|  |
| --- |
| pragma solidity ^0.4.16;  interface tokenRecipient {  function receiveApproval(address \_from, uint256 \_value, address \_token, bytes \_extraData) public;  }  contract TokenERC20 {  string public name;  string public symbol;  uint8 public decimals = 18; // 18 是建议的默认值  uint256 public totalSupply;  mapping (address => uint256) public balanceOf;  mapping (address => mapping (address => uint256)) public allowance;  event Transfer(address indexed from, address indexed to, uint256 value);  event Burn(address indexed from, uint256 value);  function TokenERC20(uint256 initialSupply, string tokenName, string tokenSymbol) public {  totalSupply = initialSupply \* 10 \*\* uint256(decimals);  balanceOf[msg.sender] = totalSupply;  name = tokenName;  symbol = tokenSymbol;  }  function \_transfer(address \_from, address \_to, uint \_value) internal {  require(\_to != 0x0);  require(balanceOf[\_from] >= \_value);  require(balanceOf[\_to] + \_value > balanceOf[\_to]);  uint previousBalances = balanceOf[\_from] + balanceOf[\_to];  balanceOf[\_from] -= \_value; balanceOf[\_to] += \_value;  Transfer(\_from, \_to, \_value);  assert(balanceOf[\_from] + balanceOf[\_to] == previousBalances);  }  function transfer(address \_to, uint256 \_value) public {  \_transfer(msg.sender, \_to, \_value);  }  function transferFrom(address \_from, address \_to, uint256 \_value) public returns (bool success) {  require(\_value <= allowance[\_from][msg.sender]); // Check allowance  allowance[\_from][msg.sender] -= \_value;  \_transfer(\_from, \_to, \_value);  return true;  }  function approve(address \_spender, uint256 \_value) public returns (bool success) {  allowance[msg.sender][\_spender] = \_value;  return true;  }  function approveAndCall(address \_spender, uint256 \_value, bytes \_extraData) public returns (bool success) {  tokenRecipient spender = tokenRecipient(\_spender);  if (approve(\_spender, \_value)) {  spender.receiveApproval(msg.sender, \_value, this, \_extraData);  return true;  }  }  function burn(uint256 \_value) public returns (bool success) {  require(balanceOf[msg.sender] >= \_value);  balanceOf[msg.sender] -= \_value;  totalSupply -= \_value;  Burn(msg.sender, \_value);  return true;  }  function burnFrom(address \_from, uint256 \_value) public returns (bool success) {  require(balanceOf[\_from] >= \_value);  require(\_value <= allowance[\_from][msg.sender]);  balanceOf[\_from] -= \_value;  allowance[\_from][msg.sender] -= \_value;  totalSupply -= \_value;  Burn(\_from, \_value);  return true;  }  } |

* 打开Remix网页编辑器,部署合约(以下代币合约部署在ropsten测试链)

|  |
| --- |
| 访问https://remix.ethereum.org/#optimize=false&version=soljson-v0.4.20+commit.3155dd80.js将合约代码复制粘贴到Remix网页编辑器中,然后点击右上角的Run，写上发行币的总量=2100000000，名称=瑞银币，代号=RYH，然后点击transact.    这时MetaMask会弹出一个交易确认框，点confirm，完成合约部署    在ropsten.etherscan.io上可查看部署交易已打包完成,其中代币合约地址=  0x00e2854c36421b802b291b0f4d84a3a54b53575c    这样可以在etherscan(ropsten测试网)查询到我们刚才部署的代币了,查看到代币名称=瑞银币，发行总量=2,100,000,000 RYH |

* 使用Metamask钱包添加代币

|  |
| --- |
| 选择metamask菜单,进入钱包帐户信息界面,点击Add Token    输入token地址,metamask钱包自动加载token信息,点击NEXT    点击ADD TOKENS,完成metamask钱包添加代币 |

* Metamask代币交易(从account1转1000RYH到account2)

|  |
| --- |
| Account1的钱包界面上点击send    转token界面上,To选择account2账号地址,填入转出的RYH数量=1000，点击NEXT    弹出转币确认信息,其中需要支付手续费=0.000079eth(~0.02美金),点击confirm,完成转币交易    Metamask显示正在打包交易区块    大约10几秒后,完成交易区块打包，钱包account1 token数量= 2099999000 RYH减少了1000RYH,account2 token数量从0变为1000RYH      同时可以在etherscan查看此次的交易记录 |