



EX^{change}
tensions

COINsolidation。

爱沙尼亚，塔林。(电子居留)

白皮书：

1.0.0版本

2020年12月。

COINsolidation.org是COINsolidation International的注册商标，已获得免费和商业使用许可。使用条款和条件见：www.Coinsolidation.org

COINsolidation International与www.OpenQbit.com，进行基于量子力学的技术合作（Quantum Security & Quantum Computing）。此次合并后，可以使用、共享和再造OpenQbit公司开发的技术。(爱沙尼亚，电子居留权)

内容

1. 介绍：	3
2. 安全量子计算。	6
3. 创建QRNG（量子随机数发生器）的"硬件"设备。	11
4. 什么是量子证明（PQu）？	17
5. 创建综合通用地址（CUA）的算法。	20
6. 双综合地址（DAC）和（HAC）的算法。	21
由COINsolidation提供项目和解决方案。	23
7. 在15分钟内创建App CUA（综合通用地址）。	25
8. 只需15分钟就可以在Android上创建你的Ethereum加密货币交易所。	29
9. 路线图COINsolidation。	32
10. 币安代币(CUAG)--ICO发行计划。	33
11. COINsolidation代币的一般特征。	34
12. 区块链平台中应用的基本概念。	35
13. 什么是Blockly编程？	38
14. 附件"CUA算法代码"。	38
15. 条款：	38

1. 介绍：

目前，无论是经济、技术还是市场利好，兼并都是与时俱进的。

我们提出了第一个加密融合或加密代币的模型，该模型提供了两个密码单体、代币或这些混合体之间的备份，基于一种算法来创建一个在COINsolidation环境中使用和生成的合并地址。

我们创建了三种类型的合并地址。

CUA(Consolidated Universal Address)用于合并并创建一个新的(活动)令牌供用户使用。组合可以有三种类型：Cryptocurrency-加密货币、Cryptocurrency-Token或Token-Token。在CUA的情况下，它是由Token-Token关系形成的。

当我们需要合并一个关于加密货币和/或代币的地址和一个用于资产转移的正常地址时，就会使用HAC（Hibric Address Consolidated）。

DAC（Dual Address Consolidated）用于管理和整合来自同一区块链或两种不同技术的两个正常地址。

我们先来看看CUA的优势。

一个CUA地址由COINsolidation令牌地址（静态地址）和一个称为"彩币"的额外令牌（可变地址）组成。在这种情况下，我们可以看到，CUA地址将始终由某种资产（Cryptosolids或代币）组合的地址形成。

在我们的案例中，当我们将COINsolidation令牌和OAP令牌整合在一起时，我们将知道它是"CUA基因"或CUAG（Consolidated Universal Address Genesis）。

El token COINsolidation esta creado en el *blockchain Ethereum* y usa el standard ERC20（Ethereum Request for Comments 20）。

"彩币"代币是基于比特币区块链并由其创建的，并使用Open Assest Protocol（OAP）标准。

让我们开始回顾一下整合地址的潜力和好处是什么。

- I. 对于创建CUA的用户来说，可以创建一个代币（OAP），可以由创建CUA的用户自定义，用户将有可能拥有自己的代币或加密活动，这样他就可以用它来创建、支持或扩展他的业务，以一种简单和轻松的方式，他将在加密代币的世界里拥有一笔资产。
- II. 对于创建CUA的公司来说，他们可能有一个代币（OAP），他们可以在供应链中创造价值，或者根据公司资产和负债的经济支持，在流动性交易中使用该资产。
- III. 对于现有的加密单片机和代币通过创建一个CUA，他们将能够使用你的地址来识别你的资产，并通过将其与代币（OAP）合并，他们将能够通过为其当前和未来的投资者提供他们自己的用户代币来增长他们的需求。

以CUAG为例，我们有两个不同区块链的各自地址。

地址比特币-代币-（OAP）。

akXma4vqxvmEqnVAKSM953wYsnjNBhN3GM7

地址Ethereum - Token COINsolidation - （ERC20）。

0x8390f8abb8fd8ad3bf8457db59f2ed75e015d303

应用算法将之前的地址进行整合，我们得到CUA地址。

cua50d0615d303k8X3m9a04fv8qaxbvm8Efqdn8VaAdK3SbMf985435w7Ydsbn5j9NfB2heNd37G5Me70

* 关于算法的更多细节，请参阅第7节----"创建综合通用地址的算法"。

我们因此有了一个方向，代表了两个不同方向的两个不同技术整合在一个方向上。

我们以简单直接的方式反映在盈利和财务扩张领域，通过投资其中一个整合了我们CUA的代币，立即你将得到一个基于比特币区块链的代币（OAP）。

现在让我们看看我们在COINsolidation中也提出的两个准则，用于加密货币和/或代币的世界。

COINsolidation令牌是整合地址的项目，在获得定制令牌时有一个即时的支持，用于每个用户在加密货币世界的增长。该项目诞生于2018年，由一群有志于将金融和科技领域融合的工程师和金融家组成，利用投资基金的优势，占据量子计算等创新技术，给资产提供安全保障，同时也是为了使用人人都可以使用的工具。

在对几种开发可能性进行评估后，我们选择了可视化Blockly编程方法论，这种方法论是基于使用扩展或模块（java编程语言中的程序），具有简单而强大的功能，可以为任何人拓展加密资产的业务，要实现这一点，我们必须涵盖以下几点。

- ✓ a.-通过能够创造一种非有形资产（个人代币），供（CUA）的创造者和使用者独家使用，为用户、投资者和资产提供即时的财务投资回报率。
- ✓ b.-我们利用用户选择加入两个区块链的优势，利用(CUA)在加密活跃市场上发展当前和未来的投资。
- ✓ c.----通过将不同的地址合并到(发援会)中，为管理这些地址提供便利；
- ✓ d.----建立和使用基于量子计算的安全。

挑战开始于创建功能和"大小"足够模块化的扩展技术，这最后一个是COINsolidation开发团队的挑战，因为在Blockly方法论和这种类型的系统中使用的扩展（AppInventor, AppyBuilder, Thunkable, Kondular。等）通常是创建不超过100k-300k字节的扩展（程序），与限制，有在他们的大小创建扩展的任务，用于在当前的区块链几乎是不可能的，由于在其创建中使用的库超过10MB和35MB之间这些大小为当前的工具Blockly系统不功能使用它们。

该团队必须创建、调整并最大限度地减少编程方法和库，以获得具有最佳功能、安全性和尺寸的扩展。

经过近两年的开发和测试，我们已经完成了第一个使用Blockly的扩展的区块链"测试版"，包括使用量子安全的共识算法"量子证明"用于加密货币交易所。

目前，我们有一个专有的区块链，已经发布了"Beta"测试，到2021年底，我们将发布生产版本的信息发布。目前我们的COINsolidation代币是基于Ethereum和比特币区块链，后者用于为用户创建自定义代币。

2. 安全量子计算。

量子计算是如何工作的？⁽²⁾

数字化转型给世界带来的变化比以往任何时候都要快，你会相信数字时代即将结束吗？**数字扫盲**已经被确定为一个领域，在这个领域，开放知识和学习技术的机会是迫切需要的，以解决社会和经济发展中的差距。随着另一个能够以惊人的速度和力量改变现有模式的新技术浪潮的即将到来，学习数字时代的关键概念将变得更加关键：**量子技术**。

在本文中，我们比较了传统计算和量子计算的基本概念；并开始探讨它们在其他相关领域的应用。

什么是量子技术？

纵观历史，人类通过科学了解自然界的运作方式，从而发展了科技。1900年至1930年间，对一些尚不十分清楚的物理现象的研究，产生了一种新的物理理论--**量子力学**。这一理论描述和解释了微观世界的功能，即分子、原子或电子的自然栖息地。由于这个理论，不仅可以解释这些现象，而且可以理解亚原子现实以一种完全反直觉的、近乎神奇的方式运作，在微观世界中发生了宏观世界中没有发生的事件。

这些量子**特性**包括量子叠加、量子纠缠和量子传送。

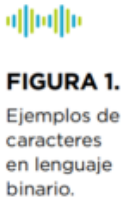
- **量子叠加**描述了一个粒子如何在同一时间处于不同的状态。
- **量子纠缠**描述了两个相距如意的粒子如何以这样的方式相关联，当与其中一个粒子相互作用时，另一个粒子会意识到它。

- **量子传送**利用量子纠缠将信息从一个地方传送到空间的另一个地方，而不需要穿越空间。

量子技术就是基于这些亚原子性质的量子特性。

在这种情况下，今天通过量子力学对微观世界的理解，使我们能够发明和设计能够改善人们生活的技术。使用量子现象的技术有很多，而且非常不同，其中一些技术，如激光或磁共振成像（MRI），已经陪伴我们半个多世纪了。然而，目前我们正在见证量子计算、量子信息、量子模拟、量子光学、量子计量、量子时钟或量子传感器等领域技术革命。

什么是量子计算？首先，你要了解经典计算。



Caracter	Bits
7	111
A	01000001
\$	00100100
:)	0011101000101001

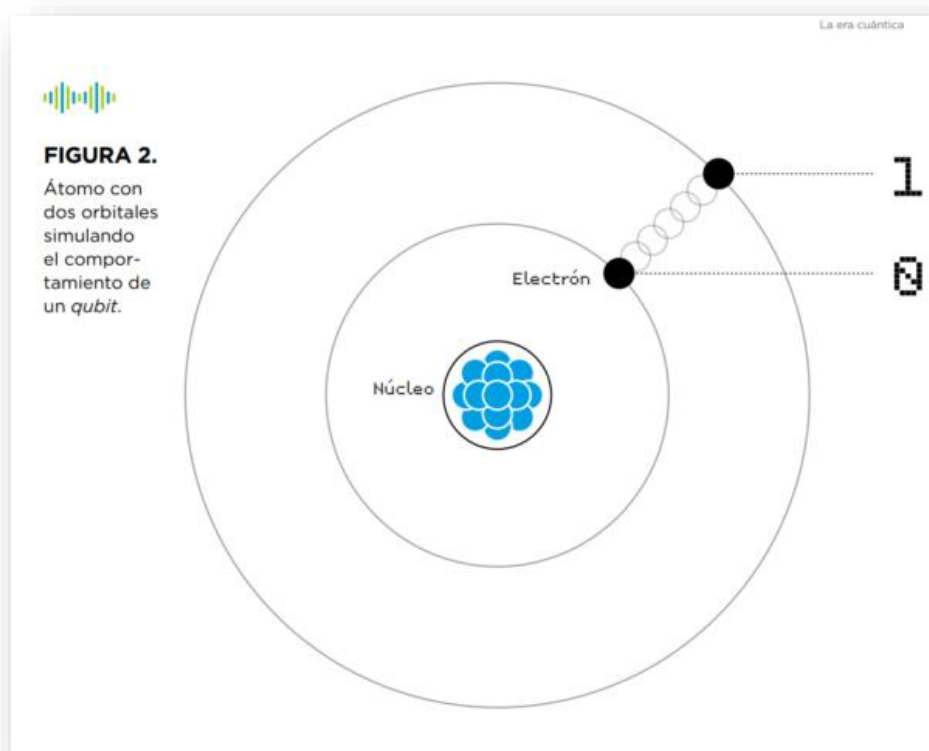
为了理解量子计算机的工作原理，我们可以方便地先解释一下我们日常使用的计算机（我们在本文中称之为数字计算机或经典计算机）的工作原理。这些和其余电子设备如平板电脑或手机一样，都是以比特为基本记忆单位。这意味着程序和应用程序是以比特为单位进行编码的，也就是以0和1的二进制语言进行编码。每当我们与这些设备进行交互时，例如按下键盘上的一个键，计算机内部就会产生、销毁和/或修改零和一的字符串。

有趣的问题是，这些0和1在物理上是计算机内部的什么？零态和一态对应的是电流，这些电流通过称为晶体管的微观部件循环或不循环，晶体管作为开关。当没有电流流过时，晶体管为"关"，对应于位0；当有电流流过时，为"开"，对应于位1。

更简单地说，就好比0位和1位对应于空穴，所以空穴是0位，电子占据的空穴是1位。这就是为什么这些设备被称为电子器件的原因。举个例子，图1显示了一些字符的二进制书写。现在我们对今天的计算机是如何工作的有了一个概念，让我们试着了解量子是如何工作的。

从比特到夸比特

量子计算中的基本信息单位是量子比特或qubit。根据定义，Qubits是两级量子系统--我们将在这里看到一些例子--它和比特一样，可以处于低级，对应于低激发状态或能量定义为0，或者处于高级，对应于高激发状态或定义为1。然而，与经典计算的根本区别就在于此，夸比特也可以处于0和1之间的任何一种无限的中间状态，比如一半0一半1的状态，或者是0的四分之三和1的四分之一。



量子算法，指数级的强大和高效计算。

量子计算机的目的是利用量子比特的这些量子特性，因为它们是量子系统，为了运行量子算法，使用重叠和交错的方式，提供比经典算法大得多的处理能力。重要的是要指出，范式的真正变化并不在于做与数字或经典计算机相同的事情--当前的计算机--但更快，正如在许多文章中可以读到的那样，而是量子算法允许以一种完全不同的方式执行某些操作，在许多情况下，这种方式被证明是更有效的--也就是说，在更短的时间内或使用更少的计算资源--。

我们来看一个具体的例子。让我们想象一下，我们在波哥大，我们想知道从一百万个选择中去利马的最佳路线（ $N=1,000,000$ ）。为了利用计算机找到最优路线，我们需要将100万个选项数字化，这意味着对经典计算机来说，要将它们翻译成比特语言，对量子计算机来说，要将它们翻译成qubits。经典的计算机需要逐一分析所有的路径，直到找到所需的路径，而量子计算机则利用了被称为量子并行的过程，使其可以同时考虑所有的路径。这意味着，经典计算机需要 $N/2$ 步或迭代的顺序，即50万次尝试，而量子计算机只需要对注册表进行 \sqrt{N} 次操作，即1000次尝试，就能找到最优路径。

在前一种情况下，其优势是二次方的，但在其他情况下，其优势甚至是指数的，也就是说，在 n 个qubits的情况下，我们可以获得相当于 2^n 位的计算能力。为了说明这一点，人们通常会计算，在一台量子计算机中，我们可以拥有更多的基态--更多不同的和同时出现的字符串--比宇宙中的原子数量还要多，据估计，宇宙中的原子数量约为 2^{80} 个。另一个例子是，据估计，如果有一台2000到2500个基态的量子计算机，我们几乎可以破解今天使用的所有密码学（所谓的公钥密码学）。

为什么要了解量子技术？

我们正处在一个数字化转型的时刻，不同的新兴技术，如区块链、人工智能、无人机、物联网、虚拟现实、5G、3D打印机、机器人或自动车等，越来越多地出现在多个领域和行业。这些技术的目的是提高人类的生活质量，加速发展并产生社会影响，如今这些技术也在同步发展。只是我们很少看到公司开发利用其中两种或多种技术组合的产

品，比如区块链和物联网或无人机和人工智能。虽然它们注定要融合，从而产生成倍的影响，但由于它们所处的发展初期阶段，以及具有技术专长的开发人员和人员的稀缺，意味着融合仍是一项有待完成的任务。

由于量子技术具有颠覆性的潜力，预计其不仅会与所有这些新技术相融合，而且会对几乎所有的新技术产生交叉影响。量子计算将威胁到数据的认证、交换和安全存储，对那些密码学有较多相关作用的技术有重大影响，如网络安全或区块链，负面影响不大，但也要考虑5G、物联网或无人机等技术。

你想练习量子计算吗？

网络上已经有几十个量子计算机模拟器，不同的编程语言已经在使用，如C、C++、Java、Matlab、Maxima、Python或Octave。另外，微软推出的Q#等新语言。你可以通过IBM和Rigetti等平台探索和玩转虚拟量子机。

约为35美元。

约为35美元。



QRNGv1.0.ino

Software
Program to arduino nano.

```
/* OpenQbitQRNG Firmware V1.0
 *Author: Guillermo Vidal
 *Copyright © 2020 OpenQbit, Inc.
 *License: MIT
 */
```

```
int triggerQ = 2; // This pin will pulse our quantum circuit
int QuA0Pin = A0; // This pin measures the horizontal polarized photons
int QuA1Pin = A1; // This pin measures the vertically polarized photons
float Qu0 = 0;
float Qu1 = 0;
```

```
void setup() {
  // Just setting up triggerPin and serial connection
  pinMode(triggerQ, OUTPUT); // sets the digital pin 2 as output
  Serial.begin(9600);
}
```

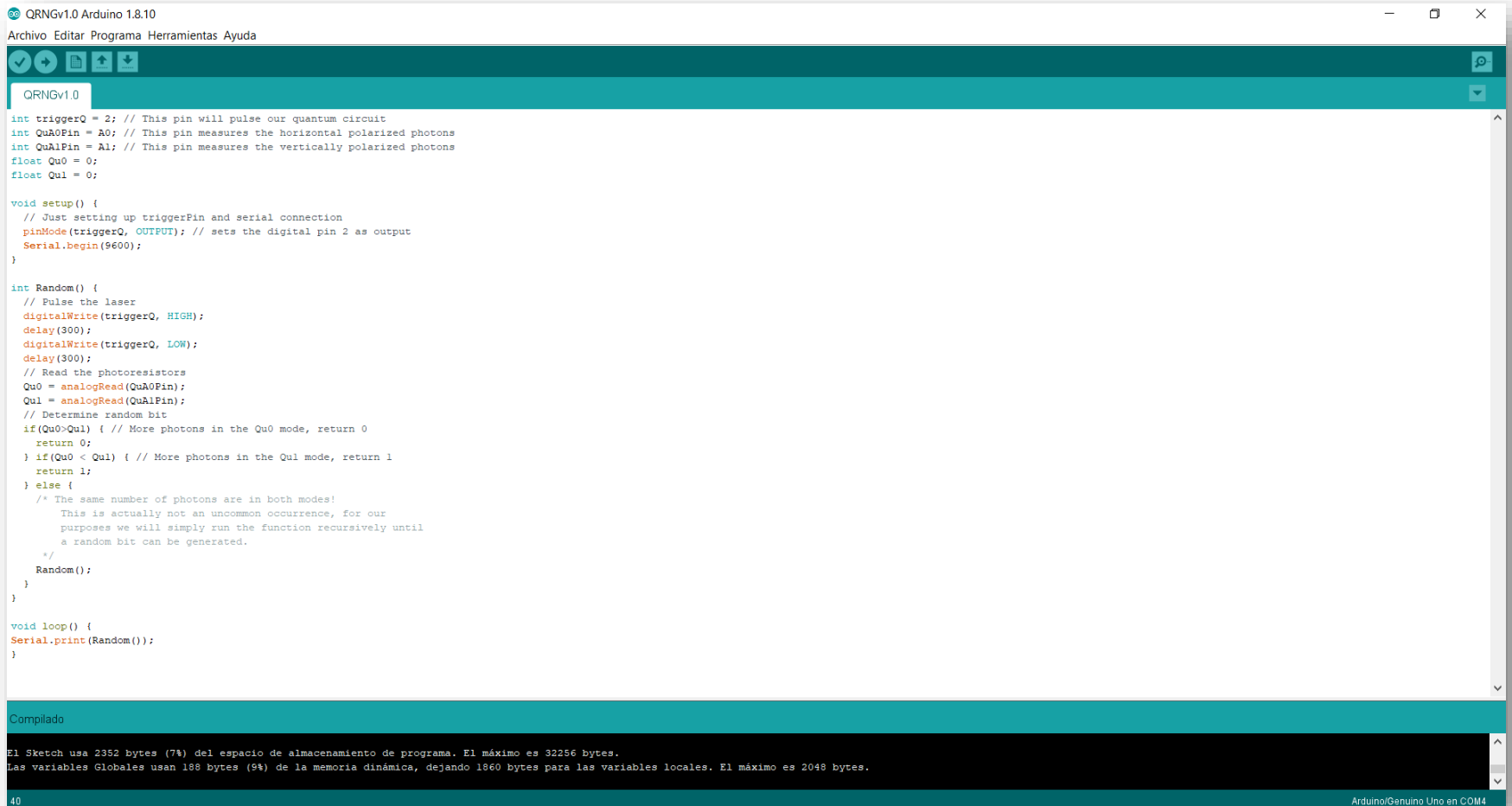
```
int Random() {
  // Pulse the laser
  digitalWrite(triggerQ, HIGH);
  delay(300);
  digitalWrite(triggerQ, LOW);
  delay(300);
  // Read the photoresistors
  Qu0 = analogRead(QuA0Pin);
  Qu1 = analogRead(QuA1Pin);
  // Determine random bit
  if(Qu0>Qu1) { // More photons in the Qu0 mode, return 0
    return 0;
  } if(Qu0 < Qu1) { // More photons in the Qu1 mode, return 1
    return 1;
  } else {
    /* The same number of photons are in both modes!
     This is actually not an uncommon occurrence, for our
     purposes we will simply run the function recursively until
     a random bit can be generated.
    */
    Random();
  }
}
```

```
void loop() {
  Serial.print(Random());
}
```

Output console

0010110101011110101011010.....

编译QRNGv10.inio程序并上传到arduino nano.....。



The screenshot shows the Arduino IDE interface with the file 'QRNGv1.0' open. The code is written in C++ and implements a quantum random number generator. It uses two analog pins (A0 and A1) to measure polarized photons and a digital pin (2) as an output. The code includes a setup function for pin modes and serial communication, a Random() function that uses the sensor data to generate random bits, and a loop function that prints the generated random bits to the serial monitor. The compilation status at the bottom indicates that the sketch uses 2352 bytes of program memory and 188 bytes of dynamic memory.

```
QRNGv1.0 Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda

QRNGv1.0

int triggerQ = 2; // This pin will pulse our quantum circuit
int QuA0Pin = A0; // This pin measures the horizontal polarized photons
int QuA1Pin = A1; // This pin measures the vertically polarized photons
float Qu0 = 0;
float Qu1 = 0;

void setup() {
  // Just setting up triggerPin and serial connection
  pinMode(triggerQ, OUTPUT); // sets the digital pin 2 as output
  Serial.begin(9600);
}

int Random() {
  // Pulse the laser
  digitalWrite(triggerQ, HIGH);
  delay(300);
  digitalWrite(triggerQ, LOW);
  delay(300);
  // Read the photoresistors
  Qu0 = analogRead(QuA0Pin);
  Qu1 = analogRead(QuA1Pin);
  // Determine random bit
  if(Qu0>Qu1) { // More photons in the Qu0 mode, return 0
    return 0;
  } if(Qu0 < Qu1) { // More photons in the Qu1 mode, return 1
    return 1;
  } else {
    /* The same number of photons are in both modes!
       This is actually not an uncommon occurrence, for our
       purposes we will simply run the function recursively until
       a random bit can be generated.
    */
    Random();
  }
}

void loop() {
  Serial.print(Random());
}
```

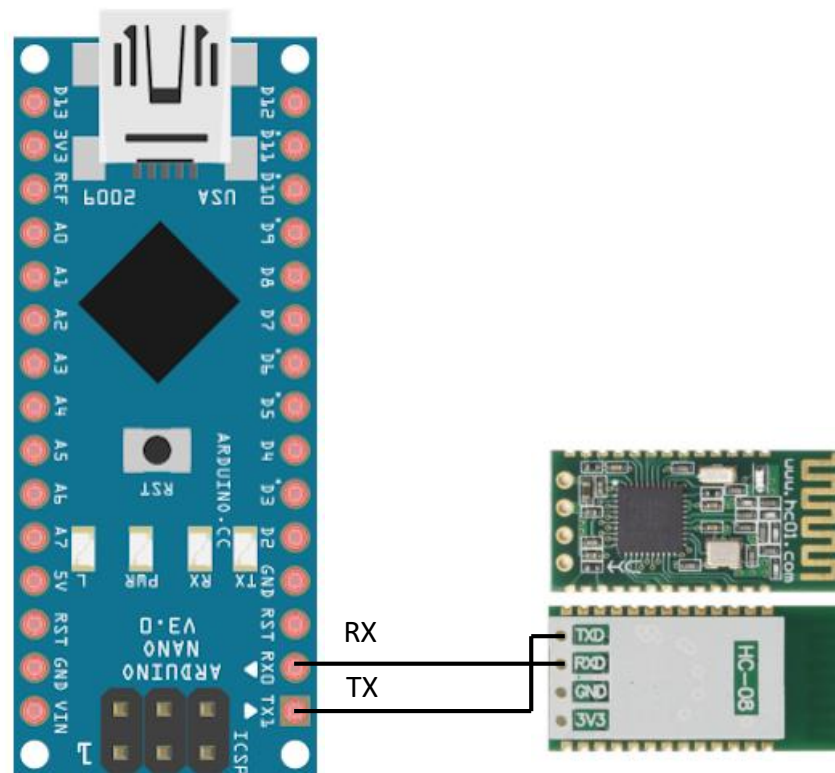
Compilado

El Sketch usa 2352 bytes (7%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables Globales usan 188 bytes (9%) de la memoria dinámica, dejando 1860 bytes para las variables locales. El máximo es 2048 bytes.

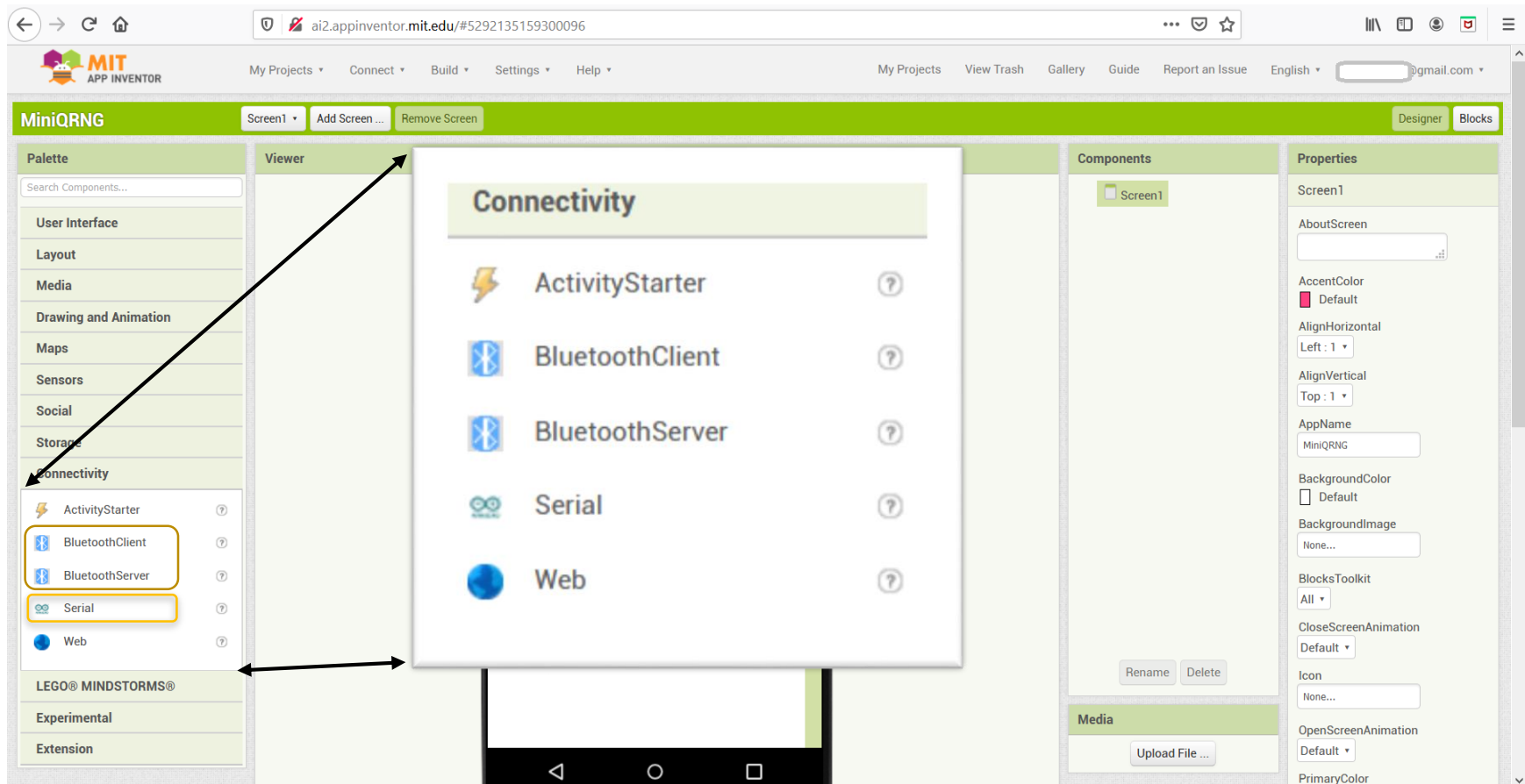
40 Arduino/Genuino Uno en COM4

与arduous nano的通信方式有两种，一种是通过Serial端口，另一种是通过蓝牙连接。

对于蓝牙的连接非常简单，我们只需要购买HC-08模块或者类似的模块，然后按照以下方式连接即可。



以下串行或蓝牙组件可用于将App Inventor连接到Arduino。





现在编译并加载程序QRNGv10.inio只缺乏与arduous nano的通信，以保存数据（量子随机数），这些将是二进制格式，然而，获得的数据可以很容易地传递到另一种格式，如十六进制或十进制，这取决于最终的要求。

最后，要想看看串口或蓝牙连接如何工作的例子，这里有一些参考链接。

请记住，一切都通过Blockly编程来测试与App Inventor这已经有块与arduino串行或其他blockly类型的系统可能是通过类似的蓝牙在线通信。

[http://kio4.com/appinventor/9A0_bluetooth_RTX.htm。](http://kio4.com/appinventor/9A0_bluetooth_RTX.htm)

[http://kio4.com/appinventor/index.htm#bluetooth。](http://kio4.com/appinventor/index.htm#bluetooth)

<https://community.appinventor.mit.edu/>

回顾QRNG（量子随机数生成器）扩展的设计和使用的整个项目。查看用户手册，地址：

<https://github.com/COINsolidation/UserGuide>



4. 什么是量子证明 (PQu) ?

PoQu。 - "量子证明"是为Mini BlocklyChain和COINsolidation开发的一种共识算法，这个测试是工作证明 (PoW) 的变种，工作原理如下。

启动时的"量子测试" (PoQu) 与"工作测试" (PoW) 的算法相同，是基于将设备 (PC、服务器、平板电脑或手机) 的处理器投入工作，以获得一串被称为"哈希"的数学难题的字符。

请记住，"哈希"是一种算法或数学过程，当引入一个短语或某种类型的数字信息，如文本文件，程序，图像，视频，声音或其他不同类型的数字信息，给我们作为一个字母数字字符，代表数字签名，代表它在一个独特的和不可重复的方式的数据。哈希算法是单向的，这意味着当你输入一个数据获得它的签名"哈希"时，它的逆向过程无法进行，有了签名"哈希"我们就不能知道获得了什么信息这个特性给我们处理我们在互联网上发送的信息带来了安全优势。想象一下，通过非安全渠道发送任何一种信息，并附上各自的"源哈希"，接收者在接收信息时，可以得到所接收信息的"哈希"，我们称之为"目的哈希"，并与"源哈希"进行核对，如果两个"哈希"相同，我们就可以确认信息在发送的渠道中没有被改变，这只是目前这种信息安全过程中使用的一个例子。

目前，有不同类型的算法或哈希过程，其安全级别不同。最常用或已知的有：MD5、SHA256和SHA512。

SHA256的例子。

我们有如下的连锁或句子。"Mini BlocklyChain是模块化的。

如果我们对前一个字符串应用一个SHA256类型的哈希，它将给我们下一个哈希。



f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8

上面的字母数字字符串是代表上面例子中的句子的签名。

更多的例子我们可以用互联网上的网站。

<https://emn178.github.io/online-tools/sha256.html>

在"测试工作"(PoW)算法的情况下，它的工作原理是利用计算能力来获得一个预定义的哈希值。

让我们想象一下，我们之前的"哈希"是从"迷你BlocklyChain是模块化的"链中提取的。

f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8

对这个"哈希"，在它的开头，我们把难度的参数，就是简单地把零"0"的开头，也就是说，如果我们说难度是4，它将有"0000"+"哈希"，我们将它称为"种子哈希"

0000 f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8

现在考虑到我们知道的输入信息是字符串："Mini BlocklyChain是模块化的"，我们在字符串的末尾添加一个从0开始的数字"0"，然后我们取出它的哈希值，我们称它为"哈希nonce"。

f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db80

我们得到了哈希nonce。

7529f3ad273fc8a9eff12183f8d6f886821900750bb6b59c1504924dfd85a7c8

然后，我们对新的"哈希nonce"和"哈希种子"进行比较，如果它们相等，首先发现相等的节点将赢得处理当前事务的执行。我们可以看到，这个过程是基于设备的概率和计算强度的，它给"工作证明"测试提供了所有节点的共识公平。

如果"种子哈希"与"哈希非奇"不一致，则难度增加1，再去掉"哈希非奇"，被增加的数字称为"非奇"数，它与"种子哈希"进行比较，直到它们一致或相同。



我们可以看到，数字"nonce"或增加是有助于获得平等的"哈希"。

在"工作测试"（PoW）算法的基础上，量子测试（PoQu）算法的基础是像PoW那样获得数字"nonce"，并使用一个。最低级别难度从1到5不等，这只是为了获得移动设备的候选权，赢得共识。

量子测试(PoQu)，当手机完成最低PoW，并赢得通关，在QRNG系统中获得一个概率数时，就会激活。

QRNG(Quantum Random Number Generator)是量子随机数生成器，这个基于量子力学生成真实随机数的系统是目前最安全的生成此类数字的系统。详见索引3中的"量子计算安全"。

COINsolidation可以实施最低PoW和PoQu两种特许经营类型。

PoQu测试是建立在获取"nonce"数字的基础上，PoQu测试中的这个数字被称为"Magic Number"，有了这个数字，Peer to Peer系统就会确认这个数字是否正确，然后用COINsolidation QRNG服务器池获取一个随机数字。这个随机数将在所有节点中登记，将创建一个包含 $((\text{节点加法})/2)+1$ 的列表，并从这个列表中选择概率最高的一个成为共识(PoQu)的赢家候选人，它将执行当前的交易队列。

PoQu算法还使用了NIST（美国国家标准与技术研究所）的测试来保证QRNG中的随机数是真正的随机数。

<https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-22r1a.pdf>

在COINsolidation中，我们实现了PoW的区块和PoQu的区块。这些区块使用哈希类型：SHA256免费使用，对于商业用途，我们有SHA512和其他需要的哈希。

关于HASH概念的更多细节请见：

https://es.wikipedia.org/wiki/Funcion_hash



注意：手机中使用的工作测试（PoW）只能使用最高难度5，因为这些设备的数学处理不像服务器或PC那样是专用的。我们只使用PoW算法来获得您进入量子随机数生成器（QRNG）系统的机会或许可，并与之执行量子随机数生成器（PoQu）算法。请看迷你BlockyChain中(PoQu)的使用。

<https://github.com/openqbit-diy/MiniBlocklyChain>

5. 创建综合通用地址（CUA）的算法。

我们的CUA（综合通用地址）地址是通过以下算法创建的。

步骤1.-从各自的地址中删除标识符，它们是识别创建区块链的地址的字母数字字符。

地址比特币-代币-（OAP）。

akXma4vqxvmEqnVAKSM953wYsnjNBhN3GM7

地址Ethereum - Token COINsolidation - （ERC20）。

0x9d08c0ac0f2fdf078c883db6fa617b15776e4b41

步骤2.-将A1中的"**a**"和A2中的"**0x**"去掉，并取每个哈希操作中用"U"符号表示的最后两个字符，得到每个地址的SHA512(地址字符串-文本)，没有初始标识符。核查号码：

$U(\text{SHA512}(\text{kXma4vqxvmEqnVAKSM953wYsnjNBhN3GM7})) = \text{50}。$

$U(\text{SHA512}(\text{9d08c0ac0f2fdf078c883db6fa617b15776e4b41})) = \text{fb}。$

步骤3.-各地址的字符从组成字符数较少的地址开始逐个连接，在字符数相同的情况下，可以从任何地址开始连接。

地址1=A10[0], A11[1], A12[2], A13[3], A14[4]。.....A1N[n], A1N+1[n+1]。

地址2=A20[0], A21[1], A22[2], A23[3], A24[4]。.....A2N[n], A2N+1[n+1]。



地址的连接。

$A_{10}[0] + A_{20}[0] + A_{10}[1] + A_{20}[1] + A_{11}[2] + A_{22}[2] + \dots + A_{1N+1}[n+1] + A_{2N+2}[n+1]$

**不能连接的最后一个字符放在字符串的开头。

6e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

步骤4.-将步骤3中可以连通的字符数加到步骤3产生的字符串的开头。

66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

步骤5.-将步骤2中每个方向的两对验证器按照A1+A2的顺序在步骤3产生的链的开头进行连接。

50fb66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

步骤6.-在步骤5中创建的地址的开头集成CUAG（综合通用地址创世）标识。

cua50fb66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

**在比特币和Ethereum地址合并的情况下，它将给出一个由82个十六进制字符组成的地址。

6. 双综合地址（DAC）和（HAC）的算法。

DAC的创建和CUA是一样的，不同的是在DAC中它们是用来整合正常的地址来接收交易的，这些地址并不代表任何加密单体或令牌。

步骤1.-从各自的地址中删除标识符，它们是识别创建区块链的地址的字母数字字符。

18gYNA9c2G9X8HZ8QxWLpLXZauAxFnsJbe(比特币地址)

0x5d2Acdb34c279Aa6d1e94a77f7b18aB938BFb2bB (Dirección Ethereum)

步骤2.-将A1中的"1"和A2中的"0x"去掉，并取每个哈希操作中用"U"符号表示的最后两个字符，得到每个地址的SHA512(地址字符串-文本)，没有初始标识符。核查号码：



$U(\text{SHA512}(\text{8gYNA9c2G9X8HZ8QxWLpLXZauAxFnsJbe})) = \text{48}。$

$U(\text{SHA512}(\text{5d2Acdb34c279Aa6d1e94a77F7b18aB938BFb2bB})) = \text{f3}。$

步骤3.-各地址的字符从组成字符数较少的地址开始逐个连接，在字符数相同的情况下，可以从任何地址开始连接。

地址1=A10[0], A11[1], A12[2], A13[3], A14[4]。.....A1N[n], A1N+1[n+1]。

地址2=A20[0], A21[1], A22[2], A23[3], A24[4]。.....A2N[n], A2N+1[n+1]。

地址的连接。

A10[0]+A20[0]+A10[1]+A20[1]+A11[2]+A22[2]+..... A1N+1 [n+1] + A2N+2 [n+1]

**不能连接的最后一个字符放在字符串的开头。

8BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3

步骤4.-将步骤3中可以连通的字符数加到步骤3产生的字符串的开头。

78BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3

步骤5.-将步骤2中每个方向的两对验证器按照A1+A2的顺序在步骤3产生的链的开头进行连接。

48f378BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3

步骤6.-DAC（双地址合并）标识整合在步骤5创建的地址开头。

dac48f378BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3

在比特币和Ethereum地址合并的情况下，它将给出一个由81个**十六进制字符组成的地址。



在HAC (Hibric Address Consolidated) 的情况下，应用在前面什么不同的是使用的地址，在这种情况下，我们将使用代表资产 (Cryptomonedas或代币) 的地址和资产转移的某种类型的区块链的正常标准地址。

注意：CUA、HAC和DAC地址的大小在每种情况下可能会有所不同，这取决于组成它们的地址。

由COINsolidation提供项目和解决方案。

目前区块链有不同类型的面向资产的不同特点，这就导致有无限多的类型地址的日常使用都要严格控制，避免在转移中出错。

另一方面，加密货币和代币的世界仅限于金融专家，或者在他们的情况下，区块链技术专家，所以普通人很难冒险创建自己的加密货币或代币。

我们通过以下几点和/或我们创造的工具，解决了前面两个COINsolidation的问题。

对于不同区块链的地址控制点，我们创建了一种算法，它将两个或多个地址以不同的组合方式进行整合（合并），从而得到一个CUA、HAC和/或DAC类型的单一地址。

有了这个解决方案，就不用再从相同或不同的区块链上发送两个地址，而只需使用一个合并地址。



对于第二个问题，我们使用了名为Blockly的编程方法论，这是一个可视化的工具，在这里不需要很高的编程知识，任何普通人或公司都可以创建自己的应用程序，而不需要投入昂贵的开发团队、时间和金钱。

我们已经创建了扩展(模块)，只需安装并使用它们来创建移动应用程序，在15分钟内。例如你自己的加密货币交易所或在几分钟内开发你自己的货币（代币）。以上所有内容都使用最先进的数据安全技术，即PQC（后量子密码学）。

只需在任何免费工具上安装扩展，如Appventor、AppyBuidler、Thunkable、Kondular或其他工具，几分钟后，你就可以进入加密单据和代币创建的世界，一切尽在掌握。

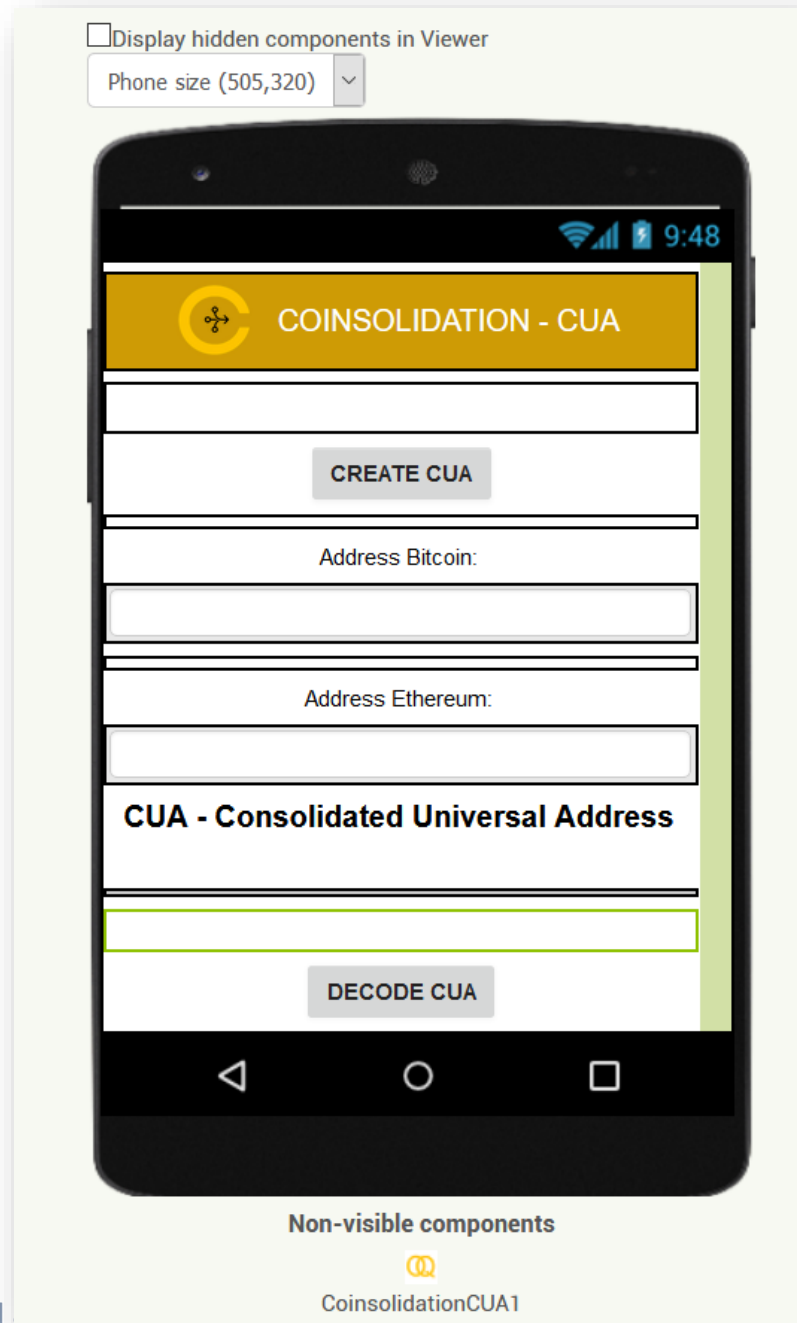
最后，COINsolidation正在创造使用低成本的量子安全（软件和硬件），可以用来保护家庭的计算机数据。目前基于量子计算和安全的技术成本较高，只有具有较高资金水平的企业才能创造和使用。但是，在COINsolidation中，我们认为新技术应该是每个人都可以使用的，区块链和量子计算的公平使用应该是每个人都可以使用的，我们创造了免费的软件（密码学）和低成本的硬件（量子安全）。



7. 在15分钟内创建App CUA（综合通用地址）。

*比特币和Ethereum币的应用程序（BTC-ETH）。

设计画面5分钟<https://appinventor.mit.edu/>





使用CoinolidatioCUA.AIX扩展名（5分钟）。

```
when GenerateCUA .Click
do
  set addressCUA . Text to call CoinsolidationCUA1 .CoinsolidationEncodeCUA_BTC_ETH
  hexAddressBitcoin InputAddressBitcoin . Text
  hexAddressEthereum InputAddressEthereum . Text

when DecodeCUA .Click
do
  call CoinsolidationCUA1 .CoinsolidationDecodeCUA_BTC_ETH
  hexAddressCUA InputAddressCUA . Text

when CoinsolidationCUA1 .OutPutAddress
  bitcoinStr ethereumStr checkBitcoin checkEthereum
do
  set addressBitcoin . Text to get bitcoinStr
  set addressEthereum . Text to get ethereumStr
  set verifyBitcoin . Text to get checkBitcoin
  set verifyEthereum . Text to get checkBitcoin
```



我们在菜单>构建>应用程序（提供.apk的二维码） - （5分钟） 中创建应用程序。

when **GenerateCUA** . Click

do

set **addressCUA** . Text to

call **CoinsolidationCUA1** .CoinsolidationEncodeCUA_BTC_ETH

hexAddressBitcoin

InputAddressBitcoin . Text

hexAddressEthereum

InputAddressEthereum . Text

when **DecodeCUA** . Click

do

call **CoinsolidationCUA1** .CoinsolidationDecodeCUA_BTC_ETH

hexAddressCUA

InputAddressCUA . Text

when **CoinsolidationCUA1** .OutPutAddress

do

bitcoinStr **ethereumStr** **checkBitcoin** **checkEthereum**

set **addressBitcoin** . Text to

get **bitcoinStr**

set **addressEthereum** . Text to

get **ethereumStr**

set **verifyBitcoin** . Text to

get **checkBitcoin**

set **verifyEthereum** . Text to

get **checkBitcoin**

0

0

Show Warnings

CUA Progress Bar

35%

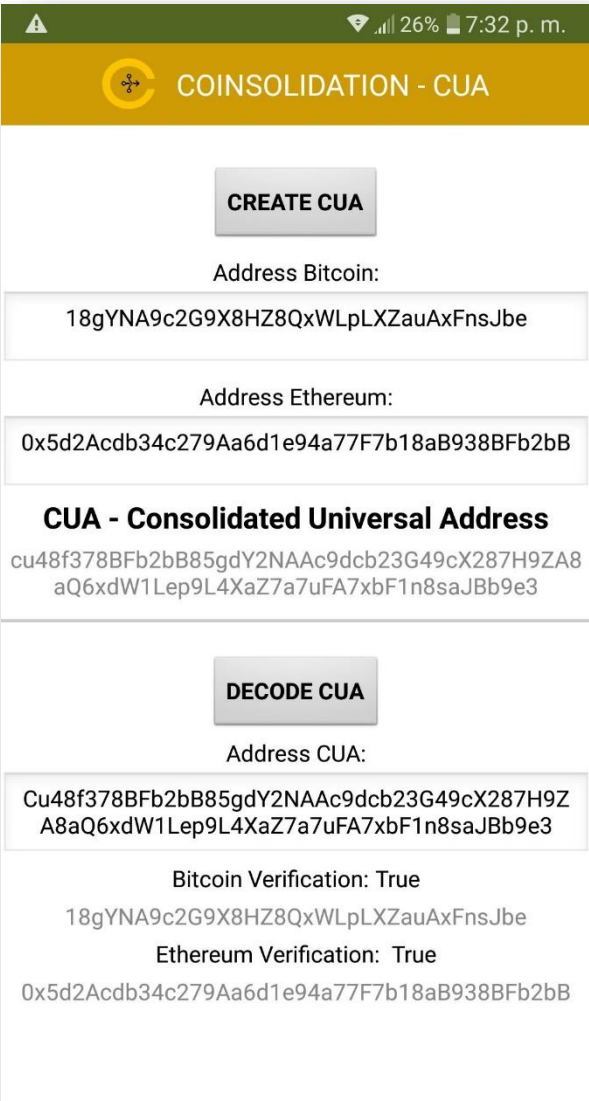
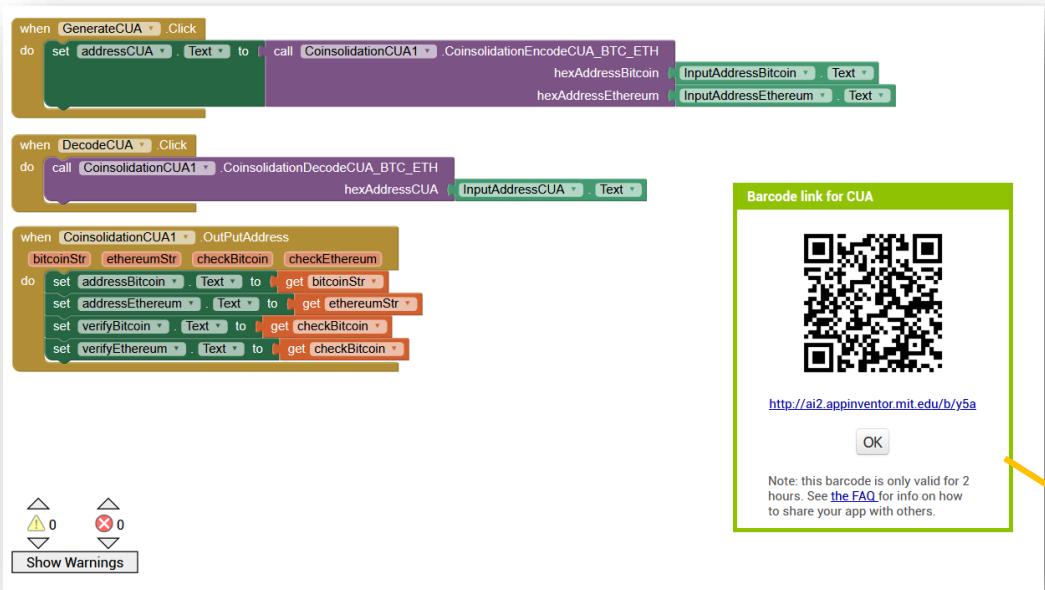
Compiling part 2 (please wait)

+

-



我们使用AppInventor的安卓应用（MIT AI2 Companion） --
<https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3>，从QR上将应用安装到手机上。

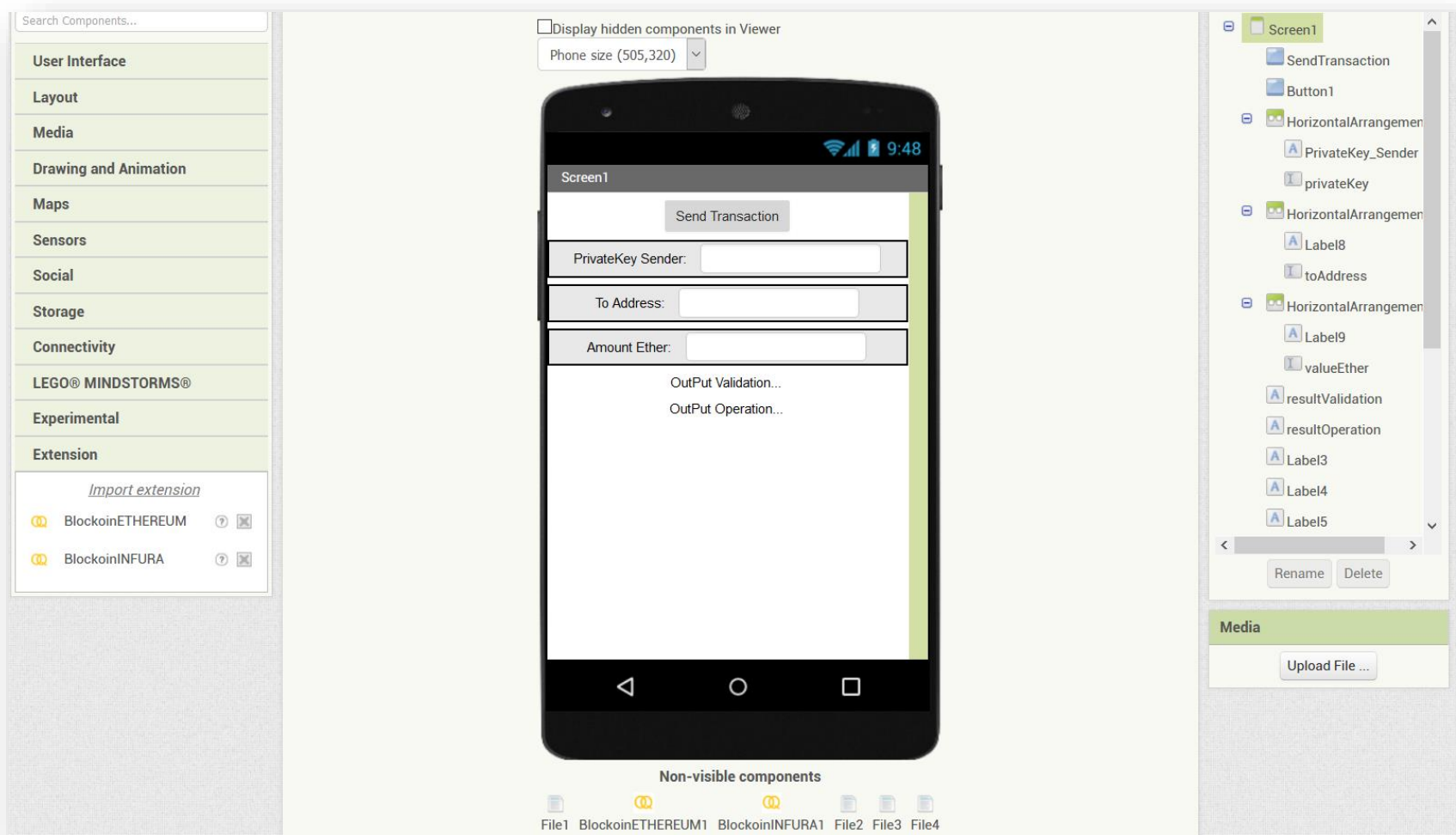


注：准备安装的APK文件应用程序位于以下资源库：<https://github.com/COINsolidation/App>。
要审查用于CUA扩展生成的Java代码，并实现综合通用地址生成算法，请审查附件"CUA算法代码"或查阅代码链接：<https://github.com/COINsolidation/source>。



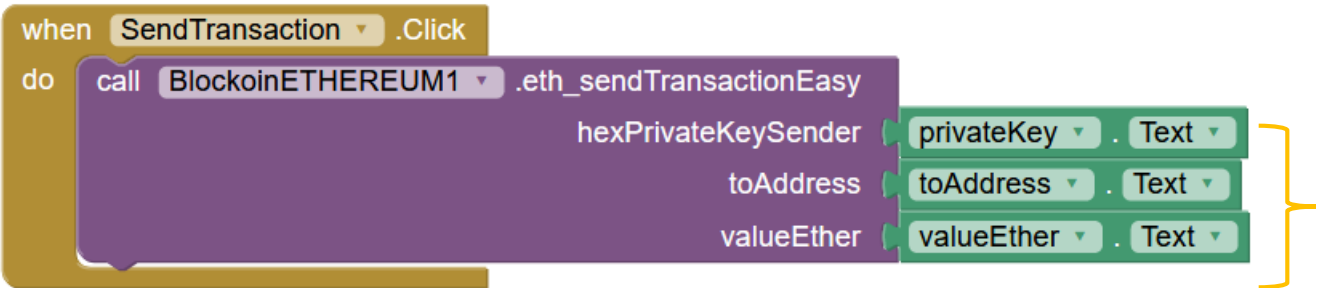
8. 只需15分钟就可以在Android上创建你的Ethereum加密货币交易所。

在App Inventor（屏幕）中设计。- 五分钟





功能块(eth_SendTransactionEasy)和事件(OutPutSendTransactionEasy)- 5分钟

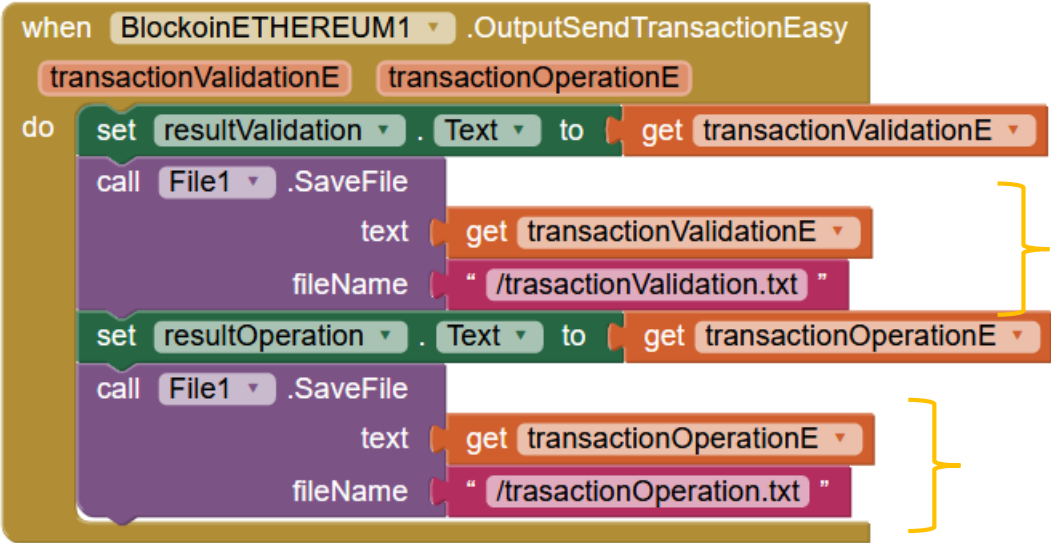


输入数据：

PrivateKey：发件人地址的主密钥。

toAddress：接收器的十六进制地址。

valueEther：给出要发送的以太币数量。



将结果保存在文本文件中。

函数File1：文件**trasactionValidation.txt**。

将结果保存在文本文件中。

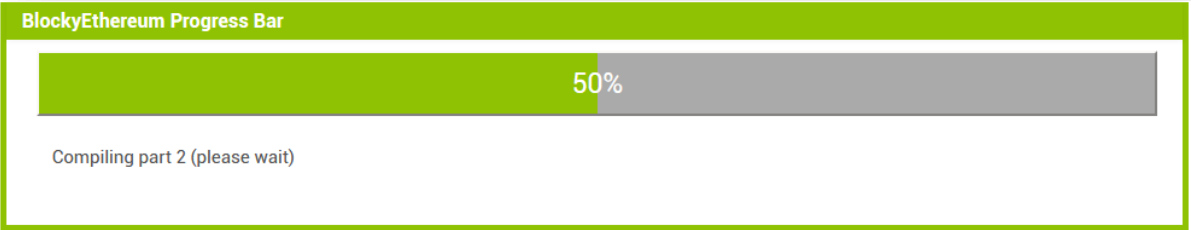
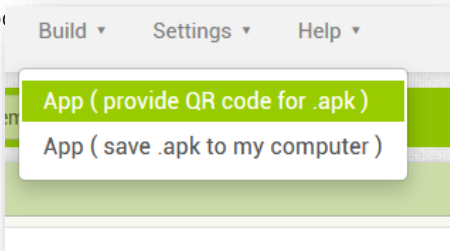
File2功能。文件 **trasactionValidation.txt**



**更多细节请参见用户指南Ethereum Exchange (EEE)扩展在版本库中：<https://github.com/COINsolidation/userguide>。

**扩展库 COINsolidation: <https://github.com/coinsolidation/Extesions-Cryptocurrencies> o OpenQbit (Blo <https://github.com/openqbit-diy>。

我们编译、生成APK文件安装到安卓设备上。- 5分钟



注意：当交易被执行时，大约需要6到8秒才能释放"发送交易"按钮。由于与Ethereum网络的连接时间。

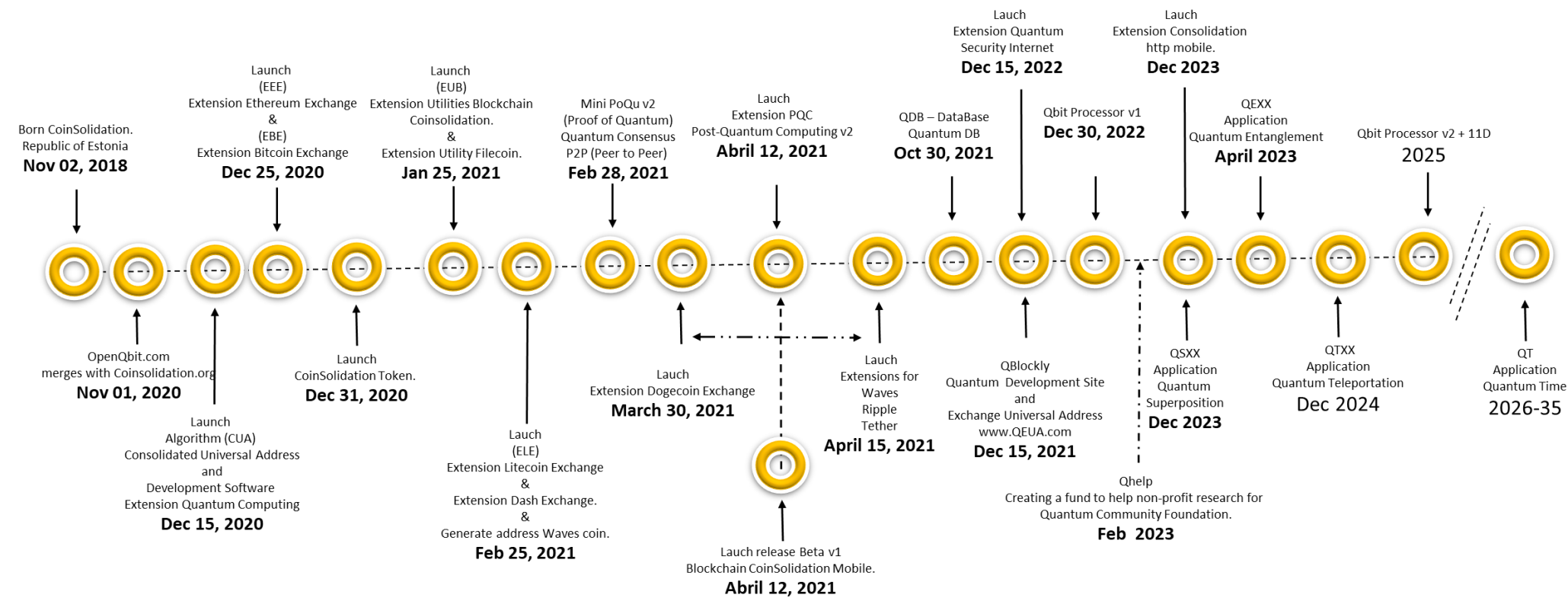
关于EEA扩展的更多细节--（Ethereum Exchange扩展）。参见链接中的EEE用户手册。

<https://github.com/COINsolidation/UserGuide>



9. 路线图COINsolidation。

ROADMAP



*OpenQbit.com与COINsolidation.org合并（2020年11月01日）/OpenQbit专注于量子计算和安全量子。

*量子处理器第1版将采用基本的量子逻辑门，供家庭使用。



10. 币安代币(CUAG)--ICO发行计划。

ICO分为三个阶段。



EXchange
tensions

The private sale	\$ 0.01 USD	(30/Dec 2020 - 30/Jan 2021)	HARD CAPITAL: \$ 280,000,000.00 USD
ICO FIRST PHASE	\$ 0.01 USD	(31/Jan 2021 - 28/Feb 2021)	SOFT CAPITAL: \$ 280,000 USD
ICO SECOND PHASE	\$ 0.05 USD	(1/Mar 2021 - 31/Mar 2021)	

CoinSolidation TOKEN DISTRIBUTION		
	%	TOKENS
TOKEN SALE	70	28,000,000,000.00
TEAM AND DEVELOPMENT	10	4,000,000,000.00
ADVISORS	5	2,000,000,000.00
PARTNERS	5	2,000,000,000.00
EXCHANGES MARKET	1.5	600,000,000.00
MARKETING	5	2,000,000,000.00
COINSOLIDATION FOUNDATION	0.5	200,000,000.00
BLOCKLY DEVELOPER COMMUNITIES	0.5	200,000,000.00
DEVELOPMENT AND RESEARCH OF QUANTUM COMPUTING	1.5	600,000,000.00
TOTAL SUPPLY 100%		40,000,000,000.00

0x4B7355FD05be6dAC458B004f54e12d6527A54A58	ICO地址
0x9d08c0ac0f2fdf078c883db6fa617b15776e4b41	协作地址



11.COINsolidation代币的一般特征。

创建人：卢沟萨马亚。

名称：COINsolidation

类型：NFT

创建的代币总数：40,000,000,000.00。

小数点的数量：18

发射国：爱沙尼亚

官方网站：www.COINsolidation.org

公司：COINsolidation International。

发射日期：2020年12月30日

共识算法：PQu(量子证明)

地址算法。综合通用地址（CUA）；

使用的安全性：基于量子计算的PQC（后量子密码学）。

技术建议：Blockly系统使用密码单体的扩展和量子安全的实现。

技术伙伴关系或协议（合并）；

公司：OpenQbit公司

行业：量子计算和PQC（后量子密码学）。

官方网站：www.OpenQbit.com





12. 区块链平台中应用的基本概念。

什么是区块链？

区块链通常与比特币和其他加密货币相关联，但这些只是冰山一角，因为它不仅用于数字货币，而且可以用于任何可能对用户和/或公司有价值的信息。这项技术起源于1991年，当时Stuart Haber和W.Scott Stornetta描述了第一个关于密码学安全区块链的工作，直到2008年，随着比特币的到来，这项技术才被人们所关注。但目前其在其他商业应用中的使用需求，预计在中期的未来，在一些市场，如金融机构或物联网、物联网等行业中，其使用量会有所增长。

区块链这个词比较好理解，它是指分布在网络中多个节点（PC、智能手机、平板电脑等电子设备）上的单一的、约定俗成的记录。就加密货币而言，我们可以把它看作是记录每一笔交易的会计账本。

如果我们去了解它的内部实现细节，它的操作可能会很复杂，但基本思路很简单。

它被存储在每个块中。

- 1.- 有效记录或交易的数量；
2. 有关该区块的资料；

因此，每个**区块在链中都有特定的、不可移动的位置**，因为每个区块都包含前一个区块的哈希信息。整条链存储在组成区块链的每个网络节点上，所以**所有网络参与者身上都存储着链的精确副本**。



区块链Ethereum平台内的地址或账户是什么？

它是Ethereum平台中42个字符的字符串，代表一个以十六进制为基础的数字，Ethereum中定义的资产将被存入或发送。在其他区块链平台中，账户或地址的字符数可以不同，例如。

`0x5d2Acdb34c279Aa6d1e94a77F7b18aB938BFb2bB`

氮金是什么？

它是一种数字或虚拟货币，旨在作为一种交易媒介。它使用密码学（数字安全）来保护和验证交易，以及控制特定密码货币的新单位的创建。

什么是令牌？

代币是可以在特定项目生态系统中使用的数字资产。

代币和加密货币的主要区别在于，前者需要另一个区块链平台（不是自己的）来操作。Ethereum是最常见的代币创建平台，主要是因为其智能合约功能。在Ethereum区块链上创建的代币一般被称为ERC-20代币，尽管还有其他更专业的代币类型，如ERC-721代币主要用于可收藏资产（卡片、电子游戏中使用、艺术品等）。

什么是交易所？

加密货币交易所是指以货币或其他加密货币为交换对象，进行加密货币交换的场所。在这些在线交易所中，市场价格的产生标志着基于供求关系的加密货币的价值。

什么是汇率？

这些是以太币或其他加密货币在各国流通货币中的价值率。例如，在本手册创建当天，以太币的美元价值为430.94美元。

什么是交易？



它是指在Ethereum系统内执行或转让某种类型的非有形资产，可以在Ethereum系统内被赋予一个预先设定的价值，并且可以在以后改变为公司或个人的有形价值。

什么是txHash？

它是一个十六进制数字，有助于跟踪每笔交易的详细结果。

有哪些交易类型？

你有两种类型，一个是交易"离线"，这创建不需要有连接到Ethereum的主网络可以存储，直到你选择连接到Ethereum的网络，并释放交易，有安全的优势，因为整个交易是离线处理，防止任何异常，可能是在网络连接。另一种交易是"在线"交易，它始终需要与互联网连接，其带来的安全优势和劣势。

什么是区块链地址？

一个地址或账户由三部分组成，地址、公钥和私钥，这两个密钥是一串十六进制格式的数字和字符，用于发送和接收（主动）或乙醚（数字货币）。

主钥匙绝对不能与任何人共享，因为它是授权释放账户中的余额（签署交易）的钥匙。

公钥是整个公众都知道的，而且是与任何人共享的，因为它是确认交易价值和发送对象正确的参考。

Ethereum网络管理组件的例子。

```
{  
  "private": "429a043ea6393b358d3542ff2aab9338b9c0ed928e35ec0aed630b93adb14a1c",  
  "public":  
    "049b4b7e72701a09d3ee09165bba460f2549494a9d9fd7a95aaac57c2827eac162fd9e105b  
    2461cd6594ca8ca6a8daf10fe982f918be1b0060c87db9cfbcd289a8",  
  "address": "88ab6dcecc3603c7042f4334fc06db8e8d7062d5"
```



}

13.什么是Blockly编程？

Blockly是一种**可视化的编程方法论**，它由一组简单的命令组成，我们可以将这些命令组合起来，就像拼图的碎片一样。对于那些想要以直观和简单的方式**学习如何编程的人来说**，或者对于那些已经知道如何编程并想要看到这种类型的编程潜力的人来说，这是一个非常有用的工具。它是基于JavaScript语言，由谷歌公司和麻省理工学院开发的。

Blockly是一种不需要任何计算机语言背景的编程形式，这是因为它只是将图形块连接起来，就像我们玩乐高或拼图一样，你只需要具备一些逻辑性，就可以了！

任何人都可以为手机（智能手机）创建程序，不用再去搞那些难懂的编程语言，只需**用图形化的方式把积木拼接起来**，以简单、方便、快捷的方式进行创建。

14.附件"CUA算法代码"。

参考Github：<https://github.com/coinsolidation/source>

15.条款：

使用条款和条件见网站www.coinsolidation.org 或 <https://github.com/coinsolidation/Terms>

支持与商业用途。

support@coinsolidation.org

销售区块链业务使用。

sales@coinsolidation.org

法律信息和许可问题或关切

legal@coinsolidation.org



社交网络。

Twitter: <https://twitter.com/ecoinsolidation>

Facebook: <https://www.facebook.com/coinsolidation>