



EXchange
tensions

Уплотнение COIN.

Таллинн, Эстония. (Э-резиденция)

Белая книга.

версия 1.0.0

Декабрь 2020 года.

COINsolidation.org является зарегистрированной торговой маркой COINsolidation International, на основании лицензии на свободное и коммерческое использование. Условия использования по адресу: www.Coinsolidation.org

COINsolidation International объединилась с www.OpenQbit.com для сотрудничества в области технологий, основанных на квантовой механике (Quantum Security & Quantum Computing). Это слияние позволяет использовать, обмениваться и реинжиниринговые технологии, разработанные OpenQbit Inc. (Эстония, электронная резиденция)

Содержание

1. Введение.	3
2. Security Quantum Computing.....	7
3. Создание "Аппаратного" устройства QRNG (Quantum Random Number Generator).....	12
4. Что такое Доказательство Кванта (PQu)?.....	18
5. Алгоритм создания сводного универсального адреса (CUA)	21
6. Алгоритм для двойного консолидированного адреса (DAC) и (HAC).	22
7. Проект и решение от компании COINsolidation.....	24
8. Создание App CUA (Консолидированный универсальный адрес) за 15 минут.	26
9. Создайте криптографический обмен валюты Ethereum на Android всего за 15 минут. ...	30
10. Дорожная карта консолидации COIN.	33
11. COIN-уплотнительный знак (CUAG) - ICO-дистрибьюторский план.	34
12. Общие характеристики жетона COIN-уплотнения:.....	35
13. Основные концепции, применяемые в платформах Blockchain.	36
14. Что такое блочное программирование?	39
15. Приложение "Код алгоритма КОА"	39
16. Условия.....	39

1. Введение.

В настоящее время слияния являются актуальными независимо от того, идет ли речь об экономическом, технологическом или рыночном благополучии.

Мы представляем первую модель криптосварки или криптотокен, которая предлагает резервное копирование между двумя криптомонадами, маркерами или их смесью, основанную на алгоритме создания консолидированного адреса, который используется и генерируется в среде COIN-уплотнения.

Мы создали три типа сводных адресов.

CUA (Consolidated Universal Address) используется для консолидации и создания нового (активного) маркера, который будет использоваться пользователем. Комбинация может быть трех типов: крипто-крипто-валюта, крипто-крипто-токен или токен-токен. В случае ТСЖ он образуется в виде отношений между жетонами.

НАС (Hibric Address Consolidated) используется, когда нам необходимо консолидировать адрес, относящийся к крипто-валюте и/или токenu, и обычный адрес для передачи активов.

ЦАП (Dual Address Consolidated) используется для управления и консолидации двух обычных адресов из одного и того же блок-цепочки или из двух разных технологий.

Давайте начнем с рассмотрения преимуществ ТСЖ.

Адрес CUA состоит из адреса токена объединения COIN (статический адрес) и дополнительного токена, известного как "Цветная монета" (переменный адрес). В этом случае мы видим, что адреса CUA всегда будут формироваться по адресам некоторой комбинации активов (криптосолиды или токены).

В нашем случае, когда мы консолидируем маркер COINsolidation и маркер OAP, мы будем знать его как "CUA genesis" или **CUAG (Consolidated Universal Address Genesis - консолидированный универсальный генезис адресов)**.

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

Токен "Цветная монета" основан и создан на *блок-схеме Bitcoin* и использует стандарт Open Assest Protocol (**OAP**).

Давайте начнем рассматривать, каковы потенциал и преимущества консолидации адресов.

- I. Для пользователей, которые создают CUA, можно будет создать токен (OAP), который может быть настроен пользователем, который создал CUA, пользователь будет иметь возможность иметь свой собственный токен или криптоактивы, чтобы он мог использовать его в создании, поддержке или расширении своего бизнеса(ов), простым и легким способом он будет иметь актив в мире крипто-токенов.
- II. Для компаний, которые создают TCA, они могут иметь токен (OAP), который они могут использовать для создания стоимости в своей цепочке поставок или использовать актив в операциях с ликвидностью, основанных на экономической поддержке активов и пассивов своей компании.
- III. Для существующих криптомонадов и маркеров, создав CUA, они смогут использовать Ваш адрес, который идентифицирует Ваш актив, и, объединив его с маркером (OAP), они смогут увеличить свой спрос, предлагая своим текущим и будущим инвесторам собственный маркер для своих пользователей.

Пример CUAG, у нас есть соответствующие адреса двух разных Blockchain:

Адрес Биткойн - жетон - (OAP).

akXma4vqxvmEqnVAKSM953wYsnjNBhN3GM7

Адрес Ethereum - Token COINsolidation - (ERC20).

0x8390f8abb8fd8ad3bf8457db59f2ed75e015d303

Применяя алгоритм консолидации предыдущих адресов мы получаем адрес CUA.

cua50d0615d303k8X3m9a04fv8qaxbvb8Efqdn8VaAdK3SbMf985435w7Ydsbn5j9NfB2heNd37G5Me70

* Подробнее об алгоритме см. раздел 7.- "Алгоритм создания сводного универсального адреса".

В результате мы имеем одно направление, представляющее две различные технологии из двух разных направлений, объединенные в одном направлении.

Мы отражаем это в области рентабельности и финансовой экспансии простым и понятным способом, инвестируя в один из токенов, который интегрирует наш CUA, вы сразу же получите токен, основанный на блок-схеме Bitcoin (OAP).

Теперь давайте посмотрим на два руководства, которые мы также выдвинули в COIN-уплотнении для мира криптомонтажей и/или токсов.

Токен COINsolidation - это проект по консолидации адресов и немедленной поддержке в получении пользовательского токена для использования при росте каждого пользователя в мире криптомонтажей. Проект родился в 2018 году с группой

инженеров и финансистов, заинтересованных в слиянии финансового и технологического секторов, чтобы воспользоваться инвестиционными фондами и занять инновационные технологии, такие как квантовые вычисления для обеспечения безопасности активов, а также с целью использования инструментов, которые могут быть доступны для всех.

После оценки нескольких возможностей разработки мы выбрали вариант визуальной методологии блочного программирования, основанный на использовании расширений или модулей (программ на языке программирования java) с простыми, но мощными функциональными возможностями для расширения бизнеса в криптографических активах для любого человека, для достижения этого мы должны охватить следующие пункты:

- ✓ а.- Немедленное финансовое окупаемость инвестиций для пользователей, инвесторов и активов, будучи в состоянии создать не материальный актив (персональные маркеры) для исключительного использования создателем и пользователем (CUA)
- ✓ б.- Мы используем преимущества соединения двух блок-цепочек по выбору пользователя для роста текущих и будущих инвестиций на крипто-активном рынке с использованием (CUA).
- ✓ с.- Содействие администрированию отдельных адресов путем их объединения в (ЦАП).
- ✓ d.- Создание и использование безопасности на основе Quantum Computing.

Вызов начался с создания технологии расширений достаточно модульных по функциональности и "размеру", эта последняя была вызовом команды разработчиков COINsolidation, так как расширения, которые используются в блочной методологии и системах этого типа (AppInventor, AppyBuilder, Thunkable, Kondular, и т.д.) обычно являются создаваемыми расширениями (программами), которые не превышают 100к - 300к байт, при этом ограничения, которые имеют по своему размеру задачу создания расширений для использования в текущей Блок-цепочки, были практически невозможны из-за того, что используемые при их создании библиотеки превышают от 10МБ до 35МБ эти размеры, так как текущие инструменты систем Blockly не функциональны для их использования.

Команда должна была создать, адаптировать и свести к минимуму методологию и библиотеки программирования, чтобы получить расширения с оптимальной функциональностью, безопасностью и размерами.

После почти двух лет разработки и тестирования мы завершили первый блок-бейн "бета" с использованием расширений для Blockly, включая алгоритм консенсуса "Доказательство квантования" с использованием квантовой защиты для криптообмена.

В настоящее время у нас есть проприетарный блокхейн, который был выпущен для тестирования "Beta", а к концу 2021 года мы выпустим производственную версию для распространения информации. В настоящее время наш COIN-токен основан на блок-схемах Ethereum и Bitcoin, последний предназначен для создания пользовательских токенов для пользователей.

2. Security Quantum Computing.

Как работают квантовые вычисления? ⁽²⁾

Цифровое преобразование приводит к изменениям в мире быстрее, чем когда-либо. Вы бы поверили, что цифровая эра вот-вот закончится? **Цифровая грамотность** уже была определена как область, в которой открытые знания и доступные возможности для изучения технологий являются насущной необходимостью для устранения пробелов в социальном и экономическом развитии. Обучение на основе ключевых концепций цифровой эры станет еще более критичным с неизбежным приходом еще одной новой технологической волны, способной трансформировать существующие модели с поразительной скоростью и мощностью: **квантовые технологии**.

В этой статье мы сравниваем основные понятия традиционных и квантовых вычислений, а также начинаем изучать их применение в других смежных областях.

Что такое квантовые технологии?

На протяжении всей истории люди разрабатывали технологии, понимая, как работает природа с помощью науки. Между 1900 и 1930 годами, изучение некоторых физических явлений, которые еще не были хорошо изучены, привело к созданию новой физической теории, **Квантовая механика**. Эта теория описывает и объясняет функционирование микроскопического мира, естественную среду обитания молекул, атомов или электронов. Благодаря этой теории удалось не только объяснить эти явления, но и понять, что субатомная реальность работает совершенно интуитивно, почти магически, и что в микроскопическом мире происходят события, которые не происходят в макроскопическом мире.

Эти квантовые **свойства** включают в себя квантовое суперпозиционирование, квантовую связь и квантовую телепортацию.

- **Квантовая суперпозиция** описывает, как частица может находиться в разных состояниях одновременно.
- **Квантовая связь** описывает, как две частицы, находящиеся так далеко друг от друга, могут быть соотнесены таким образом, что при взаимодействии с одной из них другая осознает это.
- **Квантовая телепортация** использует квантовую связь для передачи информации из одного места в другое в пространстве без необходимости путешествовать через нее.

Квантовые технологии основаны на этих квантовых свойствах субатомной природы.

В этом случае сегодня понимание микроскопического мира через Квантовую механику позволяет нам изобретать и проектировать технологии, способные улучшить жизнь людей. Существует много и очень разных технологий, использующих квантовые явления, и некоторые из них, такие как лазеры или магнитно-резонансная томография (МРТ), существуют у нас уже более полувека. Однако в настоящее время мы являемся свидетелями технологической революции в таких областях, как квантовые вычисления, квантовая информация, квантовое моделирование, квантовая оптика, квантовая метрология, квантовые часы или квантовые датчики.

Что такое квантовые вычисления? Во-первых, вы должны понять классические




FIGURA 1.
Ejemplos de caracteres en lenguaje binario.

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

вычисления.

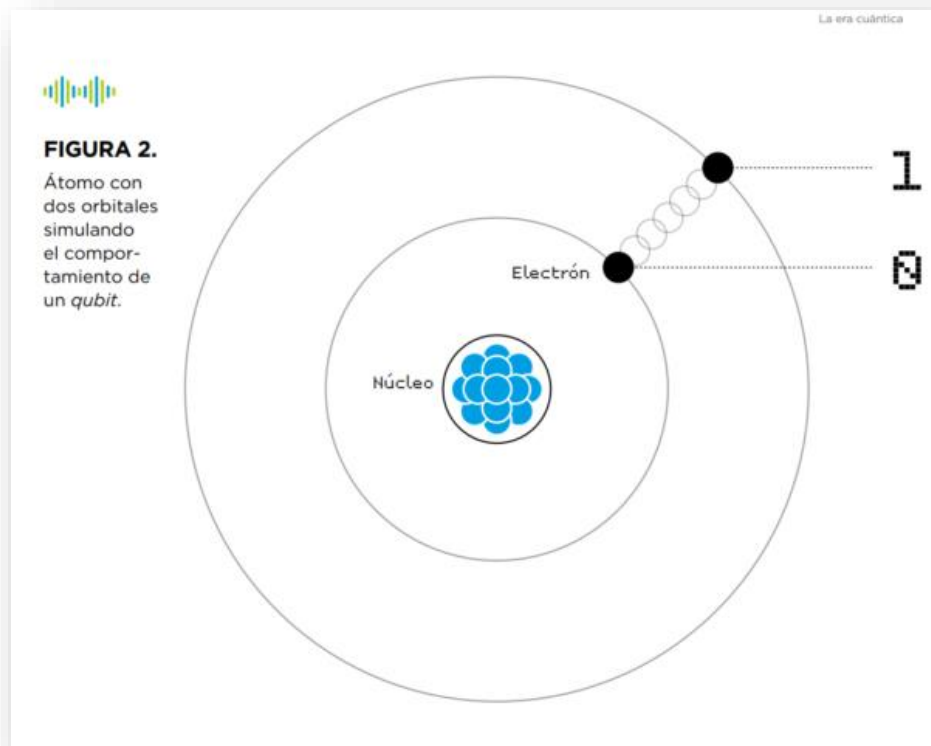
Чтобы понять, как работают квантовые компьютеры, удобно сначала объяснить, как работают компьютеры, которыми мы пользуемся каждый день и которые мы будем называть в этом документе цифровыми или классическими компьютерами. В них, как и в других электронных устройствах, таких как планшетные компьютеры или мобильные телефоны, биты используются в качестве основных блоков памяти. Это означает, что программы и приложения кодируются в битах, то есть в двоичном языке нулей и единиц. Каждый раз, когда мы взаимодействуем с любым из этих устройств, например, нажимая клавишу на клавиатуре, внутри компьютера создаются, уничтожаются и/или модифицируются строки нулей и единиц.

Интересный вопрос - что это за нули и физические нули внутри компьютера? Ноль и одно состояние соответствуют электрическому току, который циркулирует или не циркулирует через микроскопические части, называемые транзисторами, которые действуют как переключатели. Когда ток не протекает, транзистор "выключен" и соответствует биту 0, а когда он протекает, то "включен" и соответствует биту 1.

Проще говоря, биты 0 и 1 как будто соответствуют дыркам, так что пустая дырка - это бит 0, а дырка, занятая электроном, - это бит 1. Поэтому эти устройства называются электроникой. В качестве примера на рисунке 1 показана запись некоторых символов в двоичном виде. Теперь, когда у нас есть представление о том, как работают современные компьютеры, давайте попробуем понять, как работают кванты.

От битов к кубитам

Фундаментальной единицей информации в квантовых вычислениях является квантовый бит или кбит. Кубиты по определению являются двухуровневыми квантовыми системами - здесь мы увидим примеры, - которые, как и биты, могут быть на низком уровне, который соответствует состоянию низкого возбуждения или энергии, определяемой как 0, или на высоком уровне, который соответствует состоянию высокого возбуждения или определяемой как 1. Однако, и в этом заключается фундаментальное различие с классическими вычислениями, кбиты могут также находиться в любом из бесконечных промежуточных состояний между 0 и 1, например, в состоянии, которое составляет половину 0 и половину 1, или в трех четвертях 0 и четверти 1.



Квантовые алгоритмы, экспоненциально более мощные и эффективные вычисления

Целью квантовых компьютеров является использование этих квантовых свойств *квитов*, как квантовых систем, которыми они являются, для запуска квантовых алгоритмов, которые используют перекрытие и чередование, чтобы обеспечить гораздо большую вычислительную мощность, чем классика. Важно отметить, что реальное изменение парадигмы заключается не в том, чтобы делать то же самое, что делают цифровые или классические компьютеры - нынешние, но более быстрые, как это можно прочесть во многих статьях, а в том, что квантовые алгоритмы позволяют выполнять определенные операции совершенно по-другому, что во многих случаях оказывается более эффективным - то есть, за гораздо меньшее время или с использованием гораздо меньшего количества вычислительных ресурсов.

Давайте посмотрим на конкретный пример. Давайте представим, что мы находимся в Боготе и хотим знать лучший маршрут, чтобы добраться до Лимы из миллиона вариантов ($N=1,000,000$). Чтобы использовать компьютеры для поиска оптимального маршрута, необходимо оцифровать 1 000 000 вариантов, что подразумевает их перевод на битовый язык для классического компьютера и в *квиты* для квантового компьютера. В то время как классический компьютер должен будет идти один за другим, анализируя все пути до тех пор, пока не найдет нужный, квантовый компьютер использует процесс, известный как квантовый параллелизм, который позволяет ему рассматривать все пути одновременно. Это означает, что если классическому компьютеру нужен порядок $N/2$ шагов или итераций, т. е. 500 000 попыток, то квантовый компьютер найдет оптимальный путь только после операций с \sqrt{N} в реестре, т. е. 1000 попыток.

В предыдущем случае преимущество квадратично, но в других случаях оно даже экспоненциально, что означает, что при n *кубитах* мы можем получить вычислительную мощность, эквивалентную 2^n бит. В качестве примера можно привести тот факт, что на квантовом компьютере, имеющем около 270 кубитов, можно было бы иметь больше базовых состояний - более разных и одновременных символьных строк - чем количество атомов во Вселенной, которое оценивается примерно в 10^{80} . Другим примером является то, что на квантовом компьютере, имеющем от 2000 до 2500 *кубитов*, можно было бы сломать практически всю используемую сегодня криптографию (так называемая криптография с открытым ключом).

Почему важно знать о квантовых технологиях?

Мы находимся в момент цифровой трансформации, когда различные новые технологии, такие как блок-цепь, искусственный интеллект, беспилотники, интернет

вещей, виртуальная реальность, 5G, 3D-принтеры, роботы или автономные транспортные средства имеют все больше и больше присутствия в различных областях и секторах. Эти технологии, призванные повысить качество жизни человека, ускоряя развитие и создавая социальное воздействие, сегодня развиваются параллельно. Лишь в редких случаях мы видим компании, разрабатывающие продукты, использующие комбинации двух или более из этих технологий, такие как блок-цепь и IoT или беспилотные летательные аппараты и искусственный интеллект. Несмотря на то, что им суждено сойтись, тем самым создавая экспоненциально большее влияние, начальная стадия разработки, в которой они находятся, и нехватка разработчиков и людей с техническими профилями означают, что конвергенция все еще остается нерешенной задачей.

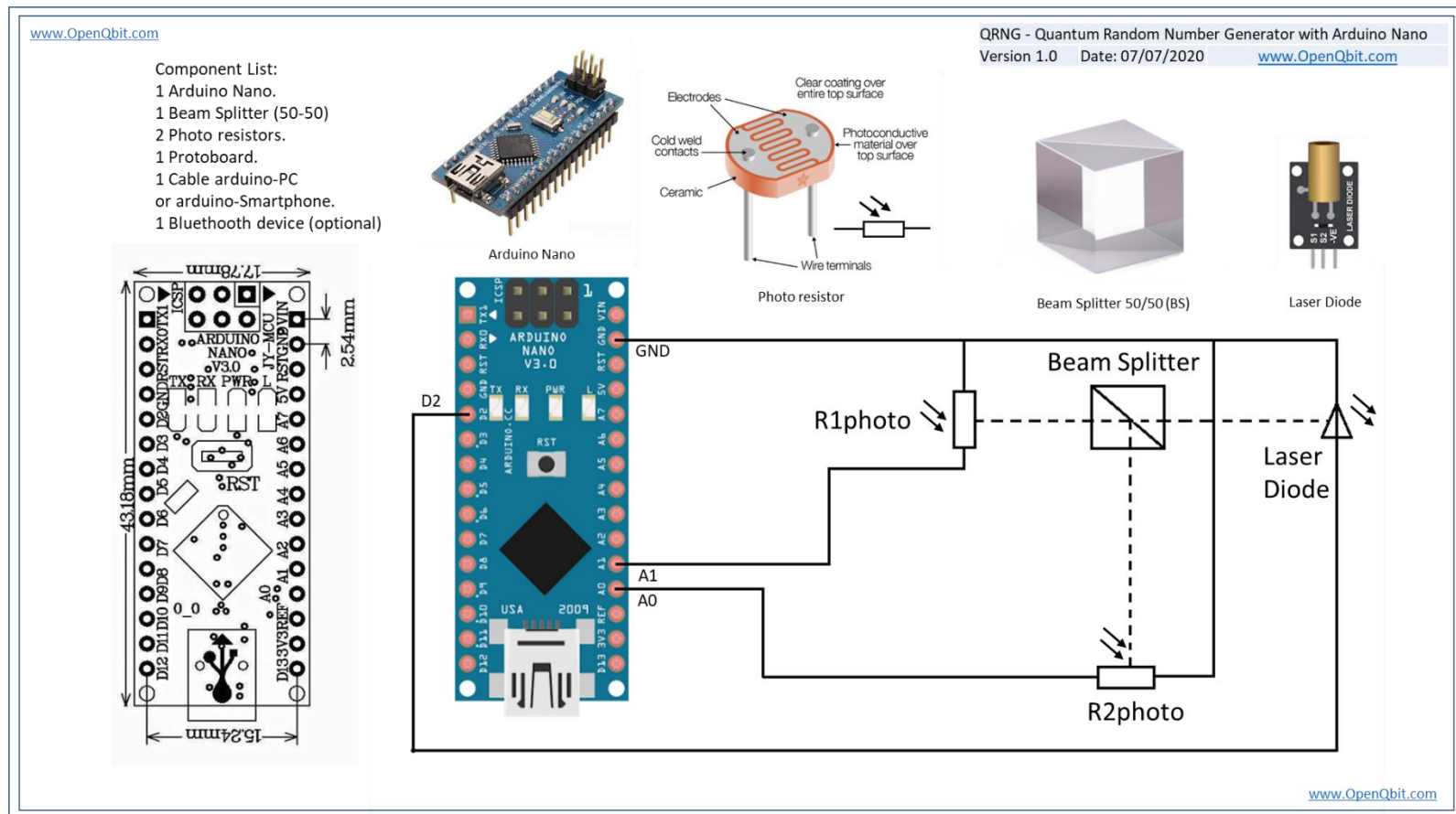
Ожидается, что в силу своего разрушительного потенциала квантовые технологии не только сблизятся со всеми этими новыми технологиями, но и окажут сквозное воздействие практически на все из них. Квантовые вычисления будут угрожать аутентификации, обмену и безопасному хранению данных, оказывая серьезное влияние на те технологии, в которых криптография играет более значимую роль, такие как кибербезопасность или блок-цепочка, а также оказывают незначительное негативное влияние, но также должны учитываться в таких технологиях, как 5G, IoT или беспилотные летательные аппараты.

Хочешь попрактиковаться в квантовых вычислениях?

Десятки квантовых компьютерных симуляторов уже доступны в сети с различными уже используемыми языками программирования, такими как C, C++, Java, Matlab, Maxima, Python или Octave. Также появились новые языки, такие как Q#, запущенные Microsoft. Вы можете исследовать и играть с виртуальной квантовой машиной через такие платформы, как IBM и Rigetti.

3. Создание "Аппаратного" устройства QRNG (Quantum Random Number Generator).

Теперь мы создадим физическое "аппаратное" устройство для генерации квантовых случайных чисел (QRNG) с недорогими компонентами, которые можно легко собрать дома и которые стоят примерно 35 долларов США.



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);
}
```

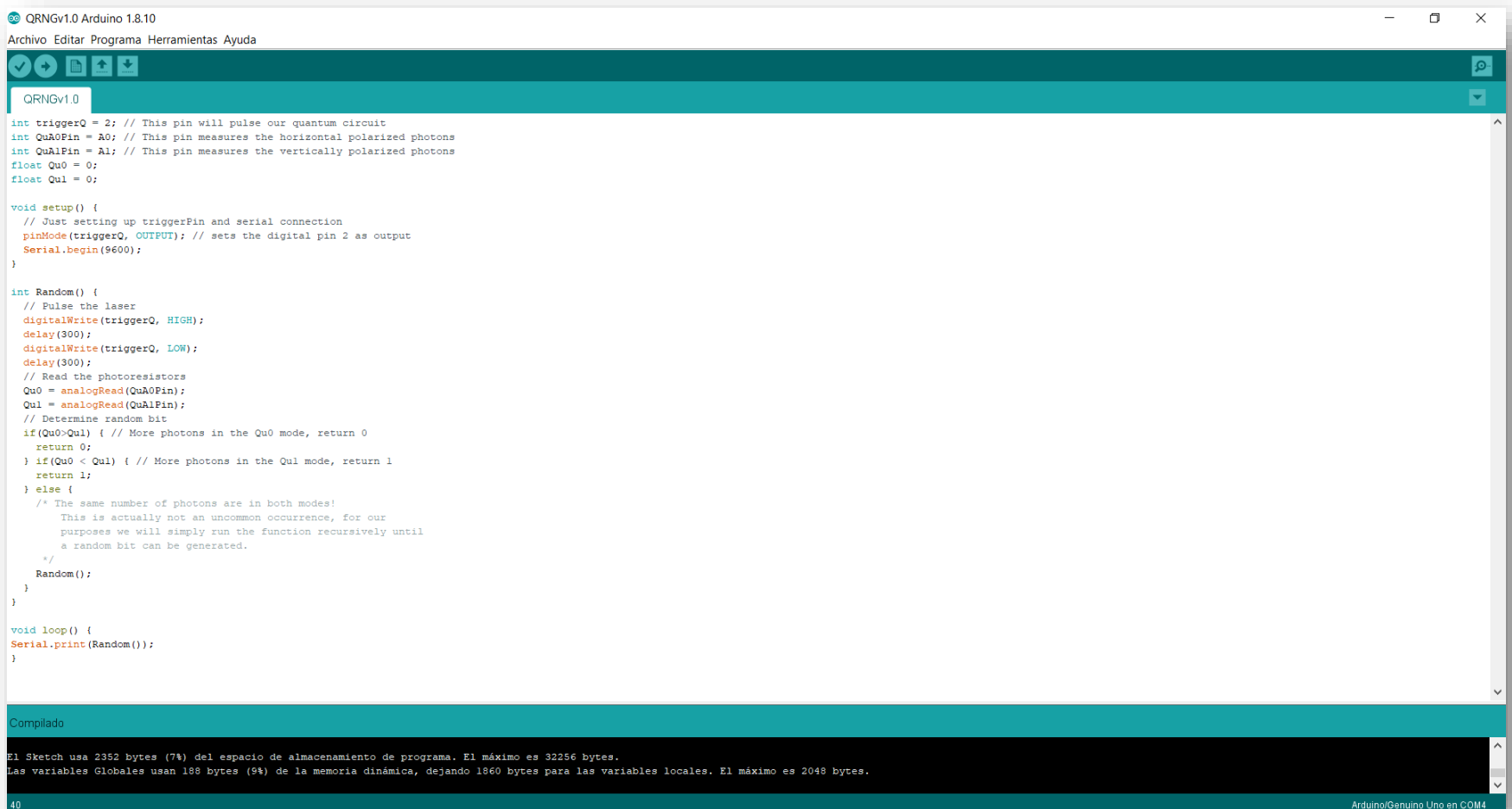
```
int Random() {
  // Pulse the laser
  digitalWrite(triggerQ, HIGH);
  delay(300);
  digitalWrite(triggerQ, LOW);
  delay(300);
  // Read the photoresistors
  Qu0 = analogRead(QuA0Pin);
  Qu1 = analogRead(QuA1Pin);
  // Determine random bit
  if(Qu0>Qu1) { // More photons in the Qu0 mode, return 0
    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.....

Компиляция программы QRNGv10.ino и загрузка на arduino nano.....



```
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());
}
```

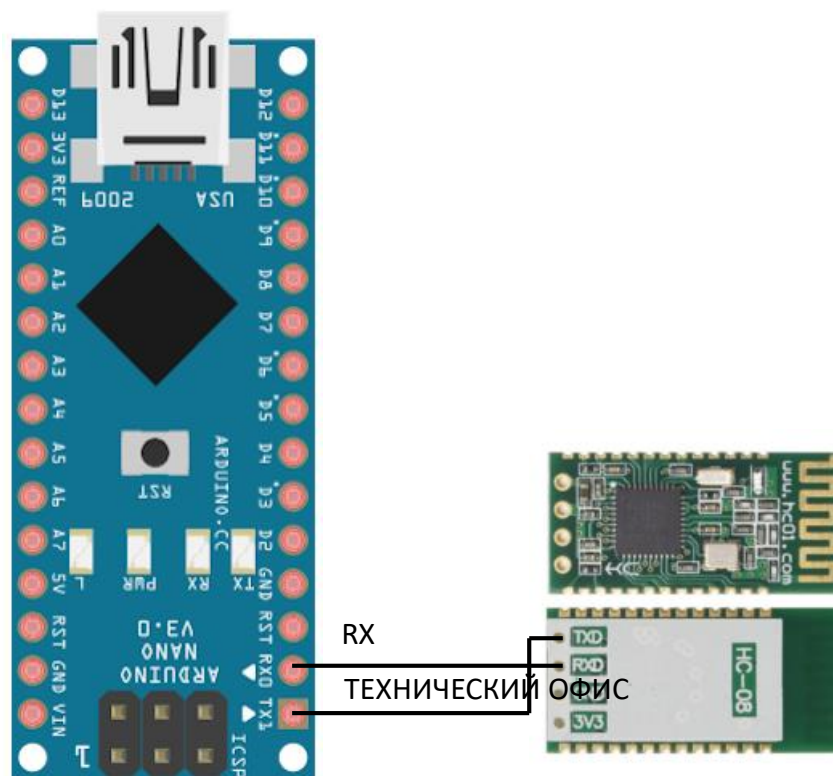
Compilado

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.

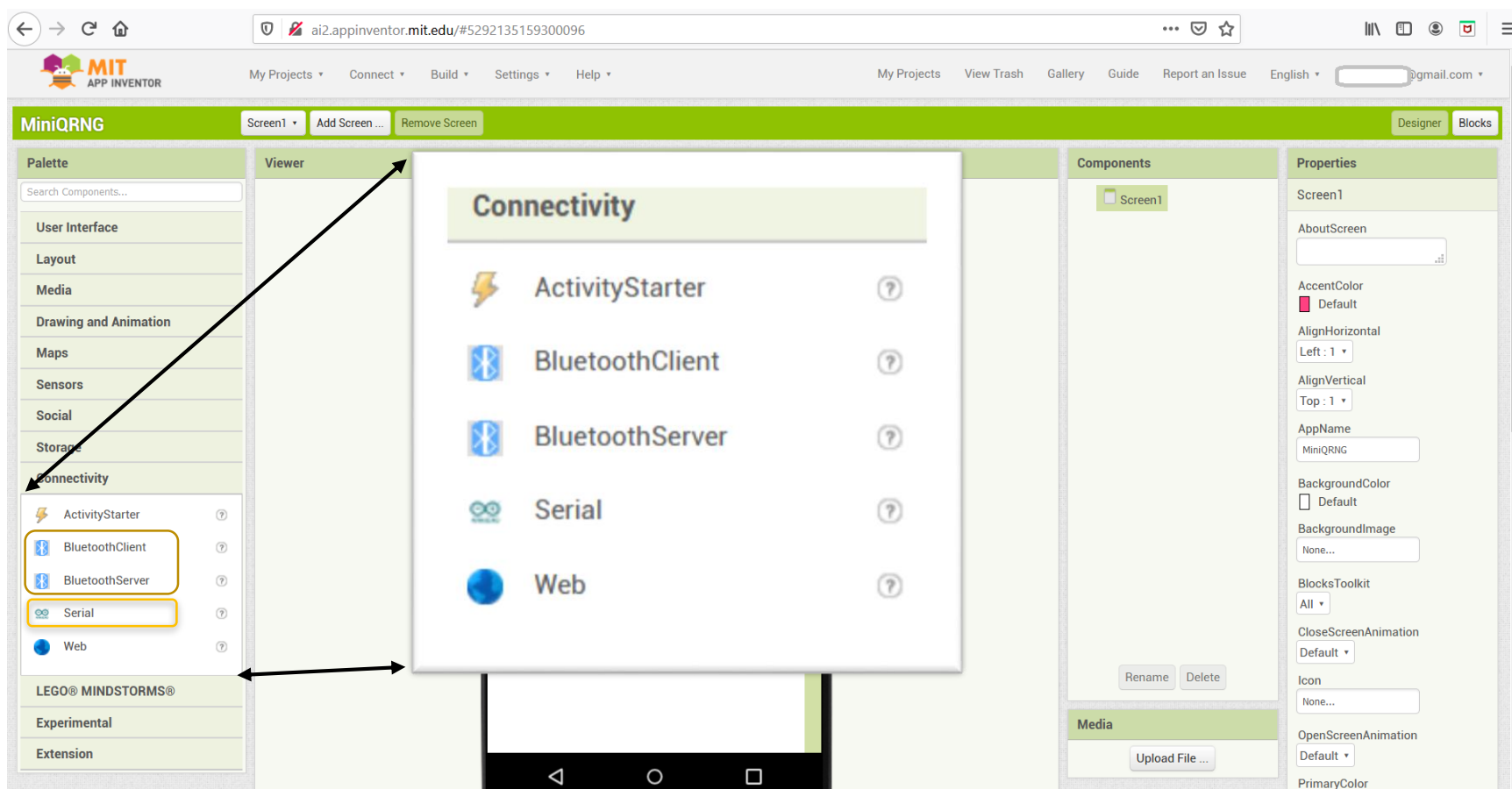
40 Arduino/Genuino Uno en COM4

Есть два способа связи с напряженным нано, один - через последовательный порт, другой - через соединение Bluetooth.

Для очень простого Bluetooth-соединения нам достаточно купить модуль HC-08 или аналогичный и подключить его следующим образом:



Следующие компоненты последовательного интерфейса или Bluetooth могут быть использованы для подключения App Inventor к Arduino:





Теперь скомпилированная и загруженная программа QRNGv10.ino только не общается со сложными нано для сохранения данных (квантовые случайные числа) они будут в двоичном формате, однако, полученные данные легко могут быть переданы в другой формат, такой как шестнадцатеричный или десятичный в зависимости от конечного требования.

Наконец, чтобы посмотреть пример того, как работает последовательное или Bluetooth-соединение, вот несколько ссылок.

Помните, что все через блочное программирование для тестирования с App Inventor это уже есть блоки для связи с arduino серийный или другой блочный тип системы может быть через аналогичный bluetooth онлайн.

<http://kio4.com/appinventor/9A0BluetoothRXTX.htm>

<http://kio4.com/appinventor/index.htm#bluetooth>

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

Рассмотреть весь проект разработки и использования расширений QRNG (Quantum Random Number Generator). Просмотрите инструкцию по эксплуатации:

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



4. Что такое Доказательство Кванта (PQu)?

Поку. - "Доказательство Кванта" - это алгоритм консенсуса, разработанный для Mini BlocklyChain и COINsolidation, данный тест является вариантом Доказательства работы (PoW), который работает следующим образом.

Тест кванта (PoQu) при запуске выполняется по тому же алгоритму, что и "Тест работы" (PoW), основанный на постановке процессора устройства (ПК, сервера, планшета или мобильного телефона) на работу для получения строки символов, представляющей собой математическую головоломку, называемую "хэш".

Помните, что "хэш" - это алгоритм или математический процесс, который при введении фразы или какого-то вида цифровой информации, такой как текстовые файлы, программа, изображение, видео, звук или другой разнородный вид цифровой информации, дает нам в результате буквенно-цифровой символ, представляющий собой цифровую подпись, которая представляет собой уникальную и неповторяемую форму данных, алгоритм хэширования является однонаправленным, это означает, что при вводе данных для получения их "хэша" подписи, обратный процесс не может быть выполнен, имея "хэш" подписи, мы не можем знать, какая информация была получена этим свойством, что дает нам преимущество в безопасности при обработке информации, которую мы посылаем через Интернет. Как это работает? Представьте себе, что посылая любую информацию по небезопасным каналам и сопровождая ее соответствующим "исходным хэшем", получатель при получении информации может получить "хэш" полученной информации, мы назовем его "целевой хэш" и проверим его "исходным хэшем", если оба "хэша" одинаковы, мы можем подтвердить, что информация не была изменена в канале, который был отправлен, это всего лишь пример, где этот тип процесса информационной безопасности используется в настоящее время.

В настоящее время существуют различные типы алгоритмов или хэш-процессов, которые отличаются уровнем безопасности. Наиболее используемые или известные: MD5, SHA256 и SHA512.

Пример SHA256:

У нас есть следующая цепочка или предложение: "Мини БлоклиЦеня" модульная.



Если мы применим хэш SHA256 к предыдущей строке, то получим следующий хэш.

f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8

Вышеуказанная буквенно-цифровая строка является подписью, которая представляет предложение в вышеприведенном примере

Для более подробного примера можно воспользоваться сайтом в интернете:

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

В случае алгоритма "Test Work" (PoW), он работает, используя вычислительную мощность, для получения предопределенного хэша.

Давайте представим, что у нас есть предыдущий "хэш", который мы взяли из цепи "Mini BlocklyChain is modular".

f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8

К этому "хэшу" в его начале мы добавляем параметр сложности, то есть просто ставим нули "0" в начале, то есть если мы скажем, что сложность равна 4, то он будет иметь "0000" + "хэш", то мы назовем его "семенной хэш".

0000 f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db8

Теперь, учитывая, что мы знаем входную информацию, которая является строкой: "Mini BlocklyChain является модульной", мы добавляем в конце строки число, начинающееся с нуля "0", и вынимаем его хэш к этому мы будем называть его "хэш нонсе":

f41af7e61c3b02fdd5e5c612302b62a2dd52fcb38f9de97cb2afd827e8804db80

У нас есть гашиш нонс:

7529f3ad273fc8a9eff12183f8d6f886821900750bb6b59c1504924dfd85a7c8

Затем выполняется сравнение нового "хэша нонсе" с "семенем хэша", если они равны узлу, который первым найдет равенство, то выиграет выполнение обработки текущей транзакции. Как видим, этот процесс основан на вероятности и вычислительной силе устройства, которое дает тест "Доказательство работоспособности" равенство консенсуса для всех узлов.

Если "семенной хэш" не совпадает с "хэшем нонсе", сложность увеличивается на единицу и "хэш нонсе" снова удаляется, увеличиваемое число называется "nonce", оно



сравнивается с "семенным хэшем" до тех пор, пока они не совпадут или не станут одинаковыми.

Как видим, число "nonce" или увеличение - это число, которое поможет получить "хэш" равенства.

Основанный на алгоритме "Test of Work" (PoW), алгоритм квантового тестирования (PoQu) основан на получении числа "nonce", как это делает PoW, с использованием а минимальный уровень сложности от 1 до 5, это служит только для того, чтобы получить право мобильного устройства быть кандидатом на выигрыш консенсуса.

Квантовый тест (PoQu), активируется, когда мобильный телефон заканчивает минимальный PoW и выигрывает проход для получения вероятностного номера в системе QRNG.

QRNG (Quantum Random Number Generator - Генератор Квантовых Случайных чисел) - это Генератор Квантовых Случайных чисел, эта система основана на генерации истинных случайных чисел на основе квантовой механики и на сегодняшний день является самой безопасной системой для генерации таких чисел. Подробнее см. раздел "Безопасность квантовых вычислений" в индексе 3.

COIN-уплотнение может реализовать как минимальный тип концессии PoW, так и тип концессии PoQu.

Тест PoQu основан на получении "nonce" числа, которое в тесте PoQu известно как "Магическое число" и при этом система Peer to Peer подтвердит правильность числа, а затем случайное число будет получено с пулом серверов COINsolidation QRNG. Это случайное число будет зарегистрировано во всех узлах, будет создан список, содержащий $((\text{Node Addition}) / 2) + 1$ и из этого списка будет выбран тот, который с наибольшей вероятностью станет кандидатом-победителем консенсуса (PoQu), и он выполнит текущую очередь транзакций.

Алгоритм PoQu также использует тестирование **NIST** (Национального института стандартов и технологий), чтобы убедиться, что случайные числа в QRNG действительно являются случайными числами.

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

В COINsolidation мы реализовали блок для PoW и блок для PoQu. Эти блоки используют хэш-тип: SHA256 для свободного использования, для коммерческого использования у нас есть SHA512 и другие хэши по мере необходимости.



Более подробная информация о концепции HASH приведена ниже:

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

ПРИМЕЧАНИЕ: Тест работы (PoW), используемый в мобильных телефонах, может использовать только максимальную трудность 5, так как математическая обработка этих устройств не выделена, как серверы или ПК. Мы используем алгоритм PoW только для того, чтобы получить Ваш пропуск или разрешение на вход в систему Генератора Квантовых Случайных Номеров (QRNG) и с его помощью выполнить алгоритм Генератора Квантовых Случайных Номеров (PoQu). См. использование (PoQu) в Mini BlocklyChain:

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

5. Алгоритм создания сводного универсального адреса (СУА)

Наши адреса СУА (Консолидированный универсальный адрес) создаются по следующему алгоритму:

Шаг 1.- Идентификаторы удаляются с соответствующих адресов, это буквенно-цифровые символы, которые идентифицируют адрес, с которого была создана цепочка блоков.

Адрес Биткойн - жетон - (OAP).

akXma4vqxvmEqnVAKSM953wYsnjNBhN3GM7

Адрес Ethereum - Token COINsolidation - (ERC20).

0x9d08c0ac0f2fdf078c883db6fa617b15776e4b41

Шаг 2.- SHA512(address String-Text) каждого адреса без его начального идентификатора получается путем удаления "a" из A1 и "0x" из A2 и взятия последних двух символов каждой операции хэширования, символизируемых буквой "U". Номера проверки.

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

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

Шаг 3.- Символы каждого адреса складываются один за другим, начиная с адреса, который имеет меньшее количество символов, чем тот, который его составляет, в



случае, если количество символов одинаково, складывание может начинаться с любого адреса.

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

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

Концентрация адресов:

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

****Последние символы, которые не могут быть сцеплены, помещаются в начало строки.**

6e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

Шаг 4.- Количество символов, которые могут быть скомпонованы в шаге 3, добавляется в начало строки, полученной в результате шага 3.

66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

Шаг 5.- Две пары верификаторов с шага 2 каждого направления собраны в начале цепочки, полученной с шага 3, в том же порядке A1 + A2.

50fb66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

Шаг 6.- Идентификация **CUAG** (Consolidated Universal Address Genesis) интегрирована в начало адреса, созданного на шаге 5.

cua50fb66e4b41k9Xdm0a84cv0qaxcv0mfE2qfndVfA0K7S8Mc985833wdYbs6nfjaN6B1h7Nb31G5M777

****В случае консолидации адресов Bitcoin и Ethereum, он даст адрес, состоящий из 82 шестнадцатеричных символов.**

6. Алгоритм для двойного консолидированного адреса (DAC) и (HAC).

Создание ЦАПов такое же, как и CUA, разница в том, что в ЦАПах они используются для консолидации обычных адресов для получения транзакций, эти адреса не представляют собой криптовалюту или токен.

Шаг 1.- Идентификаторы удаляются с соответствующих адресов, это буквенно-цифровые символы, которые идентифицируют адрес, с которого была создана цепочка блоков.

18gYNA9c2G9X8HZ8QxWLpLXZauAxFnsJbe (Биткойн-адрес)



0x5d2Acdb34c279Aa6d1e94a77F7b18aB938BFb2bB (Dirección Ethereum)

Шаг 2.- SHA512(address String-Text) каждого адреса без его начального идентификатора получается путем удаления "1" с A1 и "0x" с A2 и взятия последних двух символов каждой операции хэширования, символизируемых буквой "U". Номера проверки.

U(SHA512 (8gYNA9c2G9X8HZ8QxWLpLXZauAxFnsJbe)) = **48**

U(SHA512(5d2Acdb34c279Aa6d1e94a77F7b18aB938BFb2bB)) = **f3**

Шаг 3.- Символы каждого адреса складываются один за другим, начиная с адреса, который имеет меньшее количество символов, чем тот, который его составляет, в случае, если количество символов одинаково, складывание может начинаться с любого адреса.

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

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

Концентрация адресов:

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

****Последние символы, которые не могут быть сцеплены, помещаются в начало строки.**

8BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3

Шаг 4.- Количество символов, которые могут быть скомпонованы в шаге 3, добавляется в начало строки, полученной в результате шага 3.

78BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3

Шаг 5.- Две пары верификаторов с шага 2 каждого направления собраны в начале цепочки, полученной с шага 3, в том же порядке A1 + A2.

48f378BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3

Шаг 6.- Идентификация **DAC** (Dual Address Consolidated) интегрирована в начало адреса, созданного на шаге 5.

dac48f378BFb2bB85gdY2NAAc9dcb23G49cX287H9ZA8aQ6xdW1Lep9L4XaZ7a7uFA7xbF1n8saJBb9e3



******В случае консолидации адресов Bitcoin и Ethereum, он даст адрес, состоящий из **81** шестнадцатеричного символа.

В случае НАС (Hibric Address Consolidated) применяется в предыдущем, что варьируется адреса, которые используются, в этом случае мы будем использовать адрес, который представляет собой актив (Cryptomonedas или токен) и обычный стандартный адрес передачи активов какого-либо типа блок-цепочки.

ПРИМЕЧАНИЕ: Размеры адресов CUA, НАС и DAC могут варьироваться в каждом случае в зависимости от того, из каких адресов они состоят.

7. Проект и решение от компании COINsolidation.

В настоящее время существуют различные типы Блок-цепочки, ориентированные на активы с различными характеристиками, что приводит к тому, что бесконечное количество типов адресов ежедневного использования должны держать под жестким контролем, чтобы избежать ошибок при передаче.

С другой стороны, мир криптовалют и жетонов ограничен финансовыми экспертами или, в их случае, специалистами по технологии блок-цепей, поэтому среднестатистическому человеку трудно отважиться на создание собственного криптовалюты или жетона.

Мы решили две предыдущие проблемы в COIN-уплотнении, сделав следующие пункты и/или инструменты, которые мы создали.

Для точек контроля адресов различных блоков мы создали алгоритм, в котором он объединяет (соединяет) два или более адреса в их различные комбинации, давая в результате один адрес типа CUA, НАС и/или DAC.

При таком решении вместо отправки двух адресов из одного или разных блоков будет использоваться только один объединенный адрес.

Для второй проблемы мы использовали методологию программирования под названием Blockly, это визуальный инструмент, где не требуется больших знаний в программировании, и любой обычный человек или компания сможет создавать свои собственные приложения без необходимости вкладывать дорогостоящие команды разработчиков, время и деньги.



Мы создали расширения (модули), чтобы просто установить и использовать их для создания мобильных приложений, за 15 минут. Пример собственного обмена криптографической валюты или разработайте свою собственную валюту (токен) за считанные минуты. Все вышеперечисленное с использованием современных средств защиты данных, называемых PQC (Post-Quantum Cryptography - постквантовая криптография).

Просто установите расширения на любой бесплатный инструмент, такой как Appventor, AppyBuidler, Thunkable, Kondular или другие, и через несколько минут вы сможете войти в мир криптовалют и создания жетонов все на ладони.

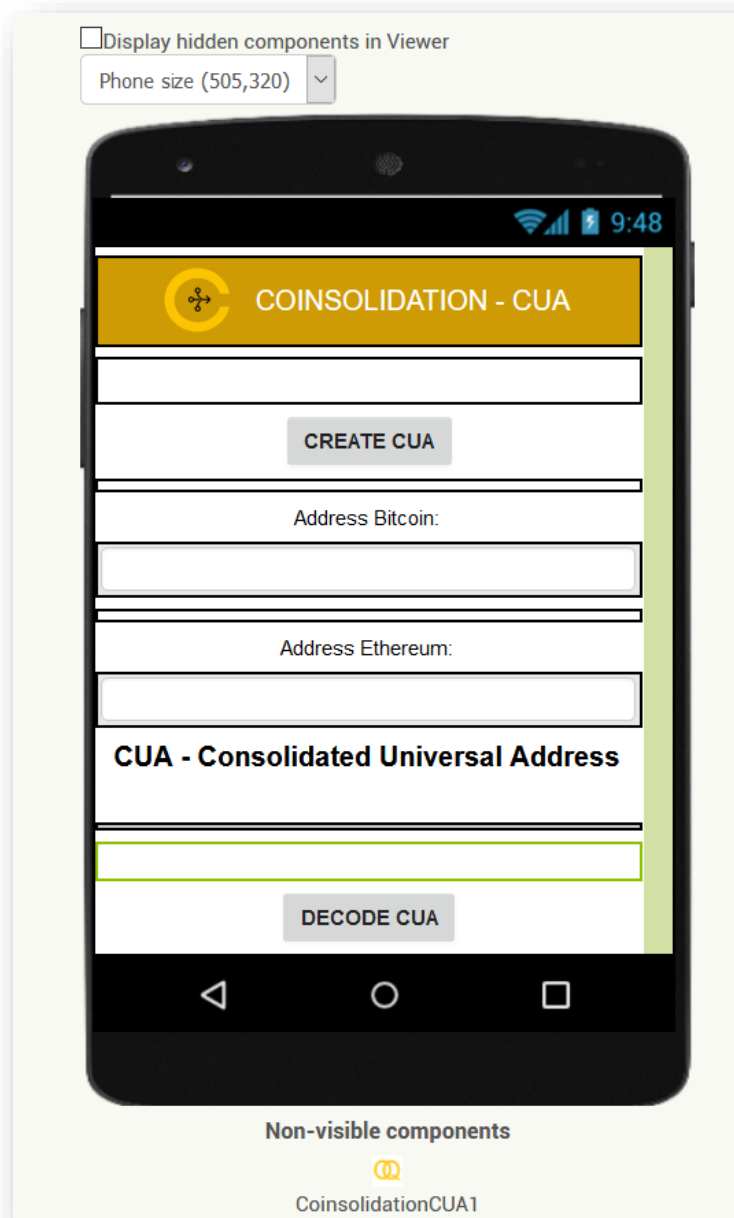
Наконец, COINsolidation создает использование недорогой квантовой защиты (программное и аппаратное обеспечение), которая может быть использована для защиты компьютерных данных в домашних условиях. В настоящее время технологии, основанные на квантовых вычислениях и безопасности, имеют высокую стоимость, создавать и использовать их может только корпорация с высоким финансовым уровнем. Тем не менее, в COIN-солидации мы считаем, что новые технологии должны быть доступны для всех, справедливость использования Блок-цепочки и Квантовый Компьютинг должны быть для всех, мы создаем свободное программное обеспечение (cryptomonics) и недорогое оборудование (квантовая безопасность).



8. Создание App CUA (Консолидированный универсальный адрес) за 15 минут.

*Приложение для монет Bitcoin и Ethereum (BTC-ETH)

Дизайн экрана 5 минут на [сайте https://appinventor.mit.edu/](https://appinventor.mit.edu/)





Использование расширения **CoinsolidatioCUA.AIX** (5 минут).

```
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
```



Мы создаем приложение в Меню > Сборка > Приложение (предоставьте QR-код для .apk) - (5 минут).

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

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)

🎯

+

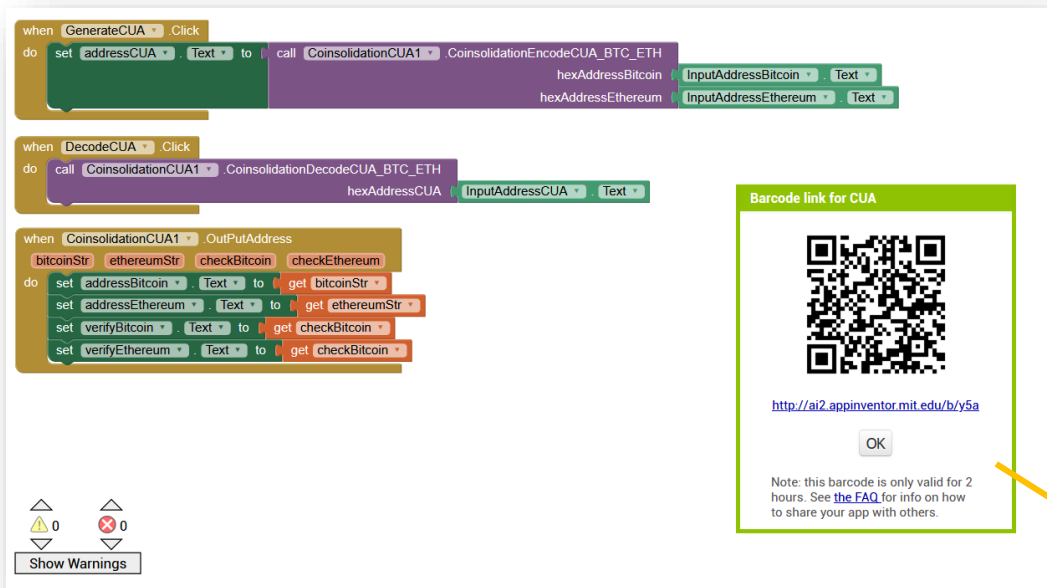
-

COINsolidation.org

Страница 28 | 40

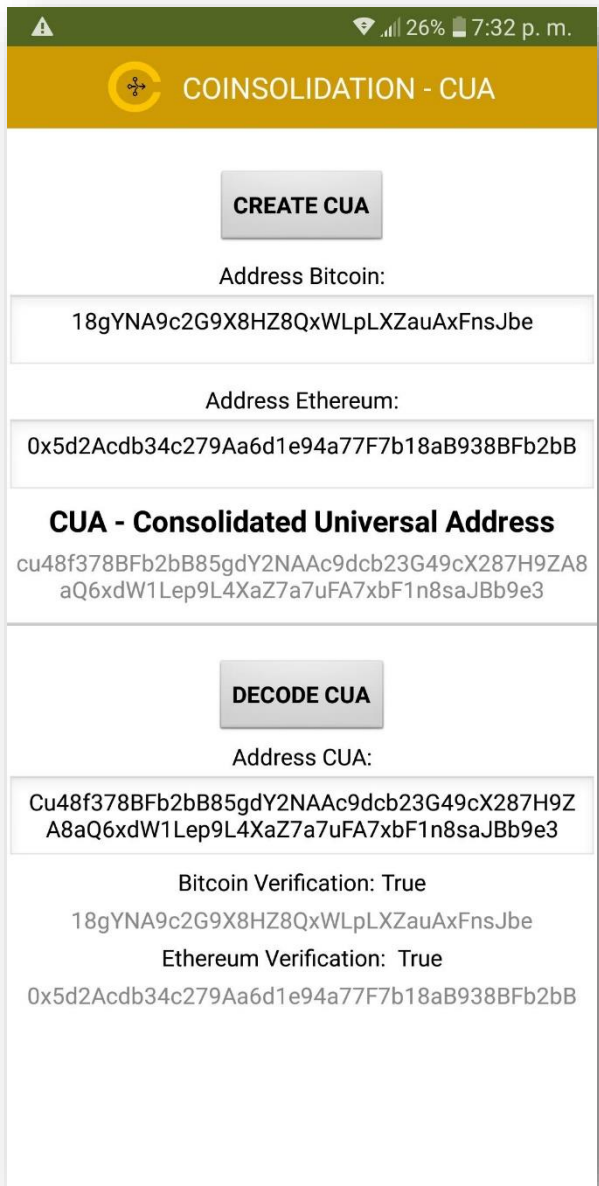


Мы установили приложение на мобильный телефон с QR с помощью приложения AppInventor для Android (MIT AI2 Companion)
- <https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3>.



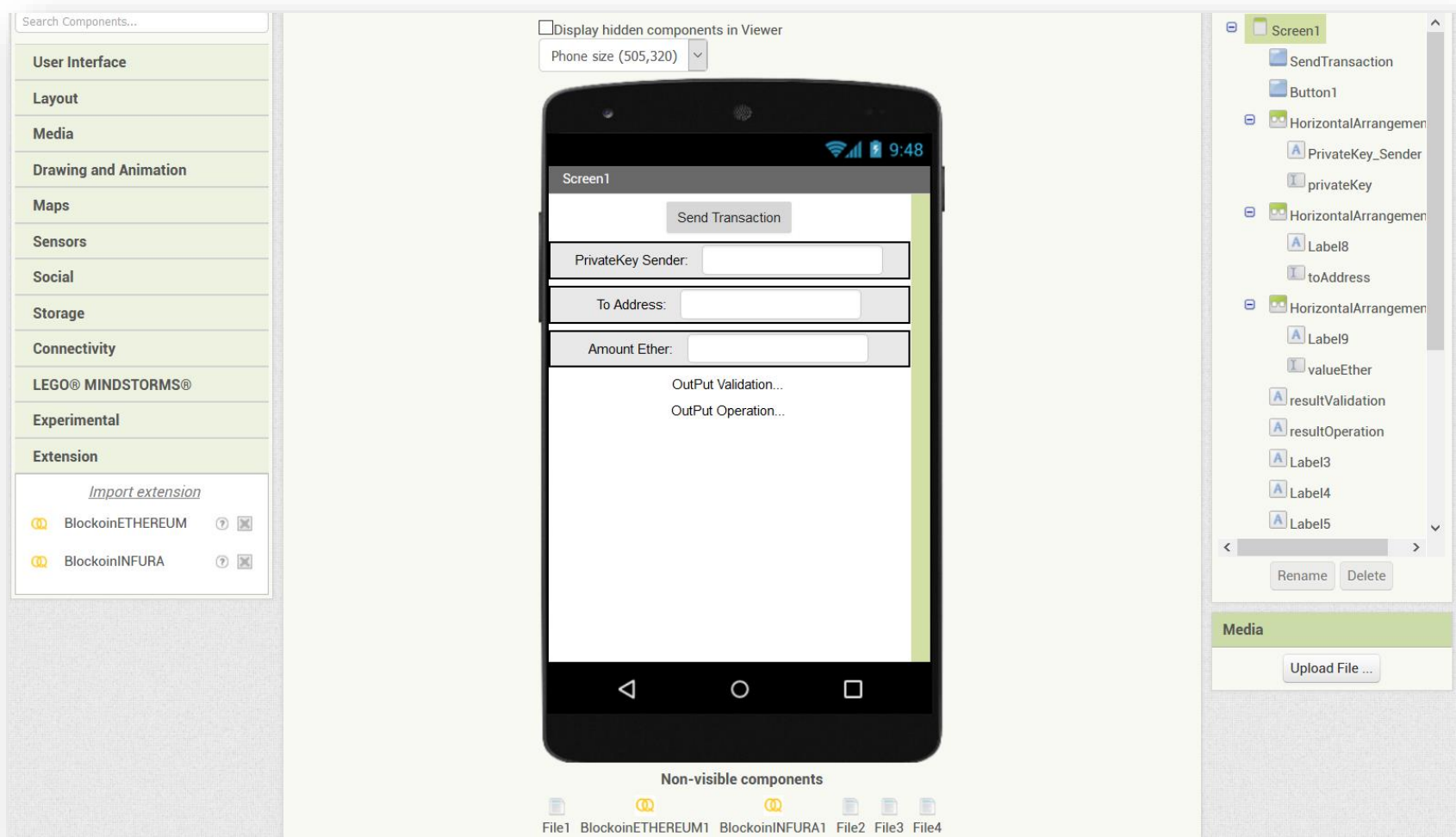
ПРИМЕЧАНИЕ: Готовый к установке файл APK приложения находится в следующем репозитории: <https://github.com/COINsolidation/App>.

Для просмотра Java-кода для генерации расширений КУА и реализации сводного универсального алгоритма генерации адресов ознакомьтесь с приложением "Код алгоритма КУА" или пройдите по ссылке с кодом: <https://github.com/COINsolidation/source>.





9. Создайте криптографический обмен валюты Ethereum на Android всего за 15 минут.
Дизайн в App Inventor (Экран). - 5 минут.





Функциональные блоки (eth_SendTransactionEasy) и событие (OutPutSendTransactionEasy) - 5 минут

```
when SendTransaction .Click
do
  call BlockoinETHEREUM1 .eth_sendTransactionEasy
    hexPrivateKeySender privateKey .Text
    toAddress toAddress .Text
    valueEther valueEther .Text
```

Вводите данные:

Первичный ключ к адресу отправителя.

Адрес: шестнадцатеричный адрес получателя.

valueEther: Дайте количество эфира, которое будет отправлено.

```
when BlockoinETHEREUM1 .OutputSendTransactionEasy
  transactionValidationE transactionOperationE
do
  set resultValidation .Text to get transactionValidationE
  call File1 .SaveFile
    text get transactionValidationE
    fileName "/trasactionValidation.txt"
  set resultOperation .Text to get transactionOperationE
  call File1 .SaveFile
    text get transactionOperationE
    fileName "/trasactionOperation.txt"
```

Сохраните результаты в текстовых файлах:

Функция Файл1: Файл **трассацииValidation.txt**

Сохраните результаты в текстовых файлах:

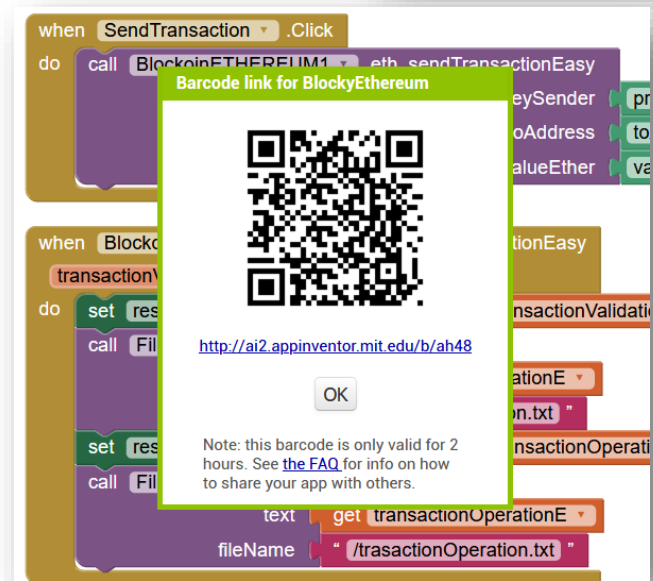
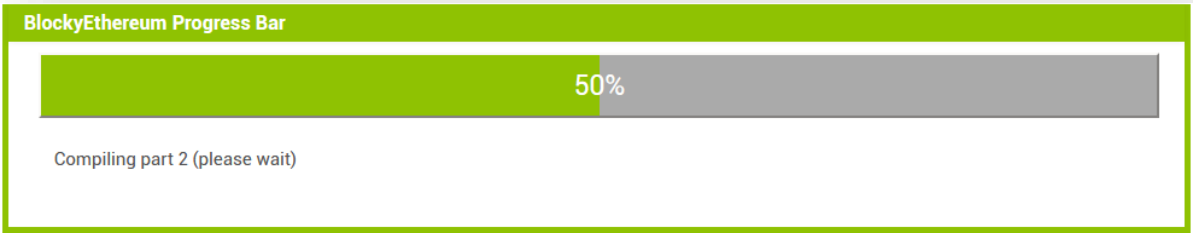
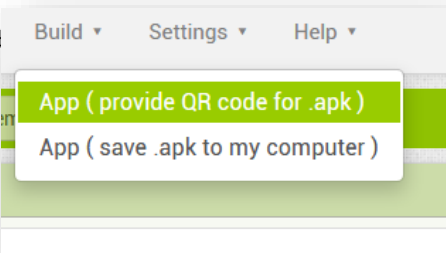
Функция File2: **Трассация файловОценка.txt**

******Больше подробностей смотрите в руководстве User Guide Ethereum Exchange (EEE) Extension (Расширение Ethereum Exchange (EEE)) в репозитории:
<https://github.com/COINsolidation/userguide>.



**Репозиторий расширений COINsolidation: <https://github.com/coinsolidation/Extesions-Cryptocurrencies> о OpenQI <https://github.com/openqbit-diy>.

Мы компилируем, генерируем APK файл, чтобы установить его на Android устройство. - 5 минут



ПРИМЕЧАНИЕ: Когда транзакция будет выполнена, потребуется примерно от 6 до 8 секунд, чтобы отпустить кнопку "Отправить транзакцию". Из-за времени подключения к сети Ethereum.

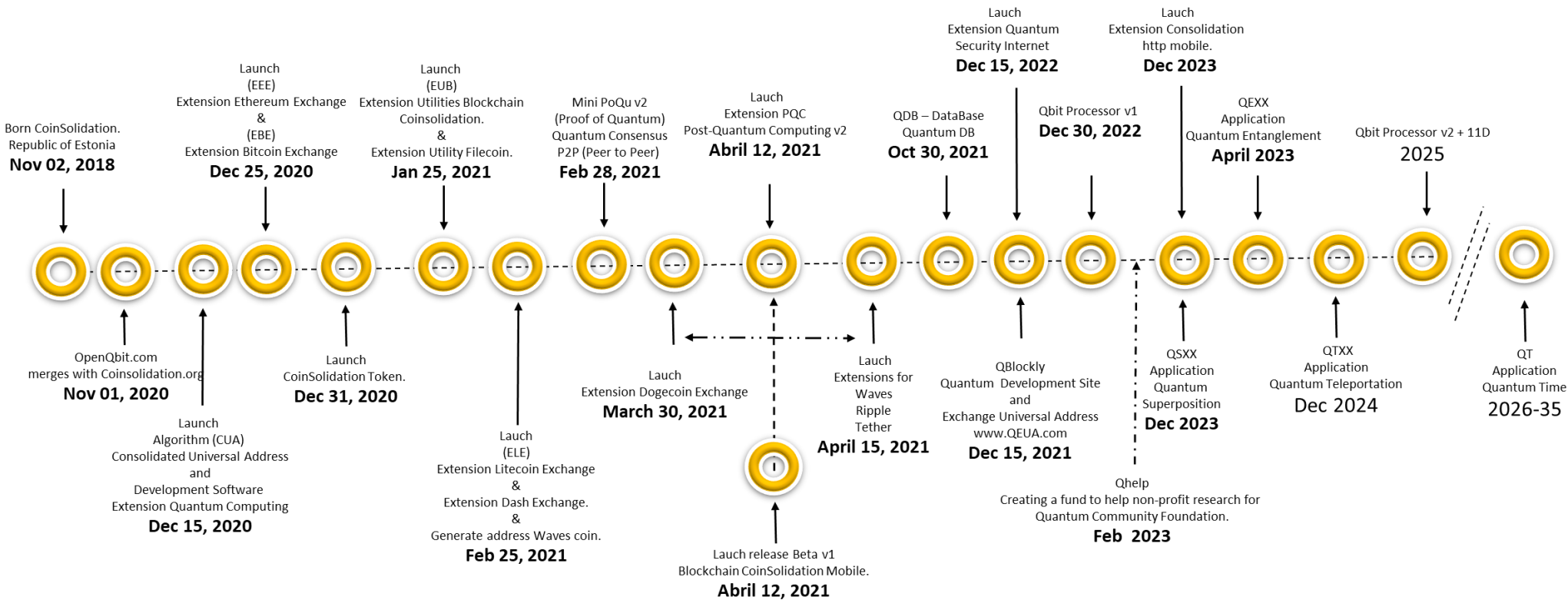
Для более подробной информации о расширении ЕЕА - (расширение Ethereum Exchange). См. руководство пользователя ЕЕЕ в ссылке:

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



10. Дорожная карта консолидации COIN.

ROADMAP



*OpenQbit.com сливается с COINsolidation.org (01 ноября 2020 г.) / OpenQbit специализируется на Квантовом вычислении и Квантовом обеспечении безопасности.

*Квантовый процессор версии 1 будет использовать основные квантовые логические ворота для домашнего использования.



11.COIN-уплотнительный знак (CUAG) - ICO-дистрибьюторский план.



ICO разделен на три этапа:

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	ИКО АДРЕСС
0x9d08c0ac0f2fdf078c883db6fa617b15776e4b41	АДРЕС СОВМЕСТНОЙ КОНСОЛИДАЦИИ



12. Общие характеристики жетона COIN-уплотнения:

Создано: Лугу Самая.

Название: COIN-уплотнение

Символ: CUAG - (Консолидированный универсальный генезис адресов).

Тип: NFT

Общее количество созданных жетонов: 40,000,000,000.00

Количество десятичных знаков: 18

Страна запуска: Эстония

Официальный сайт: www.COINsolidation.org

Компания: COINsolidation International.

Дата запуска: 30 декабря 2020 г.

Алгоритм консенсуса: PQc (Доказательство Кванта)

Алгоритм адресации: Консолидированный универсальный адрес (CUA).

Используемая безопасность: PQC (Post-Quantum Cryptography - пост-квантовая криптография) на основе квантовых вычислений.

Технологическое предложение: Расширение систем Blockly для использования криптомонадов и реализация квантовой защиты.

Технологические партнерства или соглашения (слияние):

Компания: OpenQbit Inc.

Промышленность: Квантовые вычисления и PQC (постквантовая криптография).

Официальный сайт: www.OpenQbit.com





13. Основные концепции, применяемые в платформах Blockchain.

Что такое блок-цепь?

Блокчейн обычно ассоциируется с Bitcoin и другими крипто-валютами, но это лишь верхушка айсберга, так как он используется не только для цифровых денег, но и может быть использован для любой информации, которая может иметь ценность для пользователей и/или компаний. Эта технология, которая берет своё начало в 1991 году, когда Стюарт Хабер и В. Скотт Сторнетта описали первую работу над цепочкой криптографически защищённых блоков, не была замечена до 2008 года, когда она стала популярной с приходом bitcoin. Но в настоящее время его использование востребовано в других коммерческих приложениях и, по прогнозам, в среднесрочной перспективе будет расти на нескольких рынках, таких как финансовые учреждения или Интернет вещей IoT среди других секторов.

Блок-схема, более известная как блок-схема, представляет собой единую, согласованную запись, распределённую по нескольким узлам (электронным устройствам, таким как ПК, смартфоны, планшеты и т.д.) в сети. В случае с крипто-валютами, мы можем думать об этом как о бухгалтерской книге, в которой регистрируется каждая из транзакций.

Его работа может быть сложной для понимания, если вникнуть во внутренние детали его реализации, но основная идея проста для понимания.

Он хранится в каждом блоке:

- 1.- количество действительных записей или операций,
- 2.- информация, касающаяся этого блока,
- 3.- его связь с предыдущим и следующим блоком через хэш каждого блока — уникальный код, который был бы похож на отпечаток пальца блока.

Поэтому **каждый блок имеет определенное и неподвижное место в цепи**, так как каждый блок содержит информацию из хэша предыдущего блока. Вся цепочка хранится на каждом узле сети, составляющем блок-цепочку, поэтому **точная копия цепочки хранится у всех участников сети**.

Что такое адрес или учетная запись в платформе Ethereum блок-цепочки?



Это строка из 42 символов в платформе Ethereum, представляющая собой число в шестнадцатеричной базе, где активы, определенные в Ethereum будут депонированы или отправлены. На других платформах блок-цепочки количество символов учетной записи или адреса может отличаться, например:

0x5d2Acdb34c279Aa6d1e94a77F7b18aB938BFb2bB

Что такое криптомони?

Это цифровая или виртуальная валюта, предназначенная для функционирования в качестве средства обмена. Она использует криптографию (цифровую безопасность) для обеспечения безопасности и проверки транзакций, а также для контроля за созданием новых блоков конкретного криптомонета.

Что такое жетон?

Жетоны - это цифровые активы, которые могут быть использованы в рамках данной экосистемы проекта.

Основное различие между маркерами и крипто-валютами заключается в том, что для работы первых требуется другая платформа блок-цепочки (не их собственная). Ethereum является наиболее распространенной платформой для создания токенов, в основном благодаря своей функции "умного контракта". Токены, созданные на блок-схеме Ethereum, обычно называются ERC-20, хотя существуют и другие, более специализированные типы токенов, такие как ERC-721, используемые в основном для коллекционных ценностей (карты, использование в видеоиграх, произведения искусства и т.д.).

Что такое Обмен?

Обмен крипто-валют является местом встречи, где происходит обмен крипто-валют в обмен на фиат-денег или других крипто-валют. В этих онлайн-биржах генерируется рыночная цена, которая обозначает стоимость криптомонет на основе спроса и предложения.

Что такое обменные курсы?

Это курсы стоимости эфира или другой криптографической валюты в оборотной валюте каждой страны. Например, в день создания настоящего руководства стоимость эфира в долларах США составляла 430,94 долл.



Что такое сделка?

Именно исполнение или передача какого-то не материального актива может быть присвоена заранее установленная стоимость в системе Ethereum, которая впоследствии может быть изменена на материальную стоимость для компании или лица.

Что такое txHash?

Это шестнадцатеричное число помогает отслеживать результат в деталях каждой транзакции.

Какие существуют виды сделок?

У вас есть два типа, один из которых транзакция "оффлайн", которая создается без необходимости подключения к основной сети Ethereum может храниться до тех пор, пока вы не решите подключиться к сети Ethereum и выпустить транзакцию, имеют преимущество безопасности, потому что вся транзакция обрабатывается в автономном режиме, что предотвращает любые аномалии, которые могут быть в сетевом подключении. Другая сделка - "онлайн", которая всегда должна быть подключена к Интернету с преимуществами и недостатками безопасности, которые она приносит.

Что такое адрес блок-цепочки?

Адрес или счет состоит из трех частей, адрес, открытый ключ и частный ключ, эти два ключа представляют собой строку цифр и символов в шестнадцатеричном формате, которые используются для отправки и получения (активный) или эфир (цифровая валюта).

Первичный ключ никогда никому не должен передаваться, так как именно он санкционирует разблокирование остатка (подписывает сделки), хранящегося на счете.

Публичный ключ известен всей общественности и передается любому, так как он является ссылкой, подтверждающей правильность сделки как с точки зрения стоимости, так и того, кому он передается.

Примеры компонентов управления сетью Ethereum:

```
{  
  "private": "429a043ea6393b358d3542ff2aab9338b9c0ed928e35ec0aed630b93adb14a1c",
```



```
"public":  
"049b4b7e72701a09d3ee09165bba460f2549494a9d9fd7a95aaac57c2827eac162fd9e105b  
2461cd6594ca8ca6a8daf10fe982f918be1b0060c87db9cfbcd289a8",  
"address": "88ab6dcecc3603c7042f4334fc06db8e8d7062d5"  
}
```

14. Что такое блочное программирование?

Blockly - это **визуальная методология программирования**, состоящая из простого набора команд, которые мы можем комбинировать, как кусочки головоломки. Это очень полезный инструмент для тех, кто хочет **научиться программировать** интуитивно понятным и простым способом, или для тех, кто уже умеет программировать и хочет увидеть потенциал этого вида программирования. Он основан на языке JavaScript и разработан компанией Google и MIT.

Блоки - это форма программирования, где не требуется никакого фона ни на каком компьютерном языке, это потому, что это просто соединение графических блоков, как если бы мы играли в лего или головоломку, вам просто нужна какая-то логика и все!

Любой человек может создавать программы для мобильных телефонов (смартфонов), не связываясь с теми языками программирования, которые трудно понять, просто сложите блоки в графическом виде простым, легким и быстрым способом.

15. Приложение "Код алгоритма КОА".

Ссылка на Github: <https://github.com/coinsolidation/source>

16. Условия.

Условия использования см. на сайте www.coinsolidation.org или <https://github.com/coinsolidation/Terms>.

Поддержка с коммерческим использованием.
support@coinsolidation.org

Продажа блок-цепей для бизнеса.
sales@coinsolidation.org

Правовая информация и вопросы или проблемы лицензирования
legal@coinsolidation.org



Социальная сеть:

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

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