



インストール、設定、管理。

# ユーザーマニュアル

バージョン1.0ベータ

2020年6月に

MiniQRNGはOpenQbit Inc.の登録商標です。利用規約と条件：[www.OpenQbit.com](http://www.OpenQbit.com)

オープンキュービットドットコム  
ジ

## 内容

|   |    |
|---|----|
| 1. 序章です。 .....  | 3  |
| 2. Blocklyプログラミングとは? .....  | 4  |
| 3. タームックスとは? .....  | 4  |
| 4. ミニQRNGとは? .....  | 5  |
| 5. Termux内のストレージ構成。 .....   | 9  |
| 6. SSH (Secure Shell) サーバのインストール。 .....                             | 10 |
| 7. 携帯電話 (スマホ) でのSSHサーバの設定。 .....                                    | 12 |
| 8. Ambientes Blockly (App Inventor, AppyBuilder y Thunkable)。 ..... | 20 |
| 9. ミニQRNGのブロックの定義と使い方 .....   | 21 |
| 10. QRNGの「ハードウェア」デバイスの作成 .....                                      | 34 |
| 11. 付録「OpenQbit Quantum Computing」 。 .....                          | 40 |
| 12. ソフトウェアのライセンスと使用。 .....  | 45 |

## 1. 序章です。

今日の暗号は乱数列に基づいています。現在使用されている疑似乱数発生器は、ランダムなビット列を提供しているように見えますが、実際にはこれらのビット列は特定のパターンを持っているため、ハッキングされたり、その他の方法で操作されて、情報が公共ネットワークや海賊版ネットワーク上を移動する危険性があります。

乱数発生器における物理的エントロピーの発生源の統合は、このセキュリティ上の脅威を克服するための最も一般的な方法です。しかし、古典物理学は因果関係があるので、古典物理学で生成されたビット列の予測不可能性を証明することはできません。

一方、量子物理学は、本質的にランダムです。量子乱数発生器(QRNG)で生成された数字を予測することはできません。そのため、セキュリティシステムに量子乱数発生器が使用されている場合、演算処理の速いスーパーコンピュータでも、このセキュリティシステムで使用されている乱数ビット列を予測することはできない。

量子物理学では、エントロピーと呼ばれる基本的な概念に基づいた手法を用いています。

### エントロピーの種類

真の乱数を生成するために測定できるエントロピー源には、一般的に2つのタイプがあります。最初のタイプには、測定が困難または不可能な物理プロセス、または予測するには計算量が多すぎる物理プロセス、またはその両方が含まれます。これがcaoticentropyの元になっています。ほとんどの人に知られている一般的な例は、宝くじの機械です。チャンバー内に連番のボールのセットが配置され、チャンバーを回転させたり、チャンバー内に空気を吹き込んだりして常に混合されます。複数の遊技球が転落することが許容されており、遊技球に記された数字が抽選を表しています。抽選は、ボールとカメラの間の多数の相互作用により、各ボールの可能な動きの数が急速に増加しているため、ランダムに行われます。これらの相互作用の複雑性が非常に高いだけでなく、ボール、カメラ、気流のすべての内部変数を正確に観察したり、測定したりする方法は明らかにありません。第二の、非常に異なるタイプのエントロピーの源は、量子力学である。光子、電子、

陽子などの多くの微小な粒子や波動は、回転、偏光、位置、運動量などの量子力学的な性質を持っています。これらの粒子を生成するための適切な構成があれば、例えば、その回転や偏光の具体的な値は、未知で理論的に予測できないだけでなく、測定が行われるまでは物理的に決定される。

## 2. Blocklyプログラミングとは？

Blocklyは、パズルのピースであるかのように組み合わせることができるシンプルなコマンドのセットで構成された**視覚的なプログラミング言語**です。直感的でシンプルな方法で**プログラミングを学びたい人**や、すでにプログラミングの方法を知っていて、このタイプのプログラミングの可能性を見たい人にはとても便利なツールです。

Blocklyは、コンピュータ言語のいずれかの種類のバックグラウンドを必要としないプログラミングの形態です。

誰もが理解するのが難しいそれらのプログラミング言語をいじらずに携帯電話（スマートフォン）用のプログラムを作成することができます、ちょうど作成するためのシンプルで簡単かつ迅速な方法でグラフィカルな方法でブロックをまとめてください。

## 3. タームックスとは？

Termuxは、Android端末エミュレータであり、ルーティングや設定を必要とせずに直接動作するLinux環境のアプリケーションです。最低限の基本システムが自動的にインストールされています。

安定性やインストール・管理のしやすさからTermuxを使用しますが、Ubuntu Linux for Androidのインストール環境を使用しても問題ありません。

このLinux環境では、MiniQRNGの通信プロセスの"コア"となるものを手に入れることができます。

## 4. ミニQRNGとは？

ミニQRNGは、QRNG（量子乱数発生器）を作成するための3つの技術的なソリューションを含むソフトウェアとハードウェアです。以下のように分類されます。

- a. - QRNG APIです。- 外部サーバーから取得した量子乱数発生器
- b.- MiniQRNG/ソフトウェア。- 携帯電話のカメラの物理的性質（量子）を利用して得られるラウンド量子数発生器。
- c.- MiniQRNG/ハードウェア。- レーザーの量子物理学的性質に基づいたハードウェアを用いて得られる量子数発生器。後日、自宅で安く建てる方法をお伝えします。

### 1. Termuxターミナルのインストールと設定

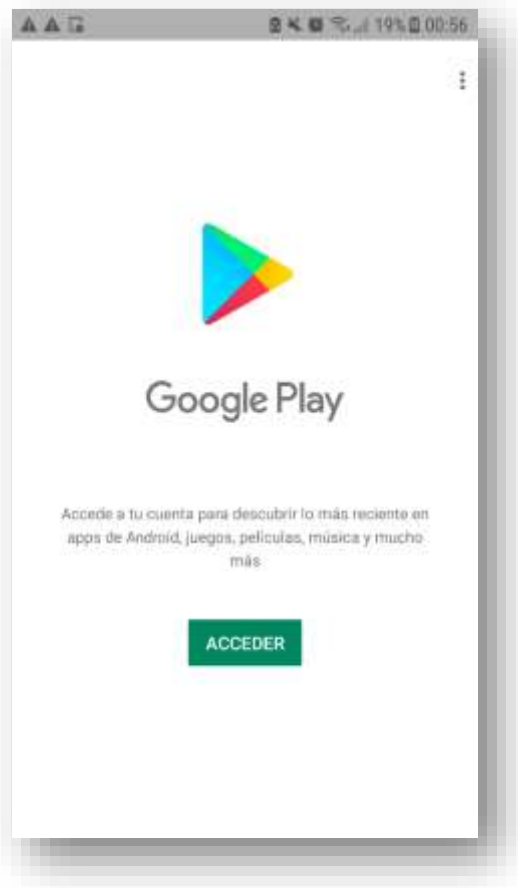
Androidシステムはセキュリティとツールの柔軟性のためにLinuxをベースにしているので、まずLinux環境が必要です。

TermuxはLinuxエミュレータで、量子数を作成するために必要なパッケージをインストールします。

Termuxを使用する大きなメリットの一つは、携帯電話（スマートフォン）を「回転」させずにプログラムをインストールできることです。これにより、このインストールによってメーカー保証が失われることはありません。

Termuxのインストール。

モバイルからGoogle Playのアイコンアプリ（play.google.com）にアクセスします。



アプリ「Termux」で検索し、選択してインストール作業を開始します。



## Termuxアプリケーションの起動。

起動後、Linuxオペレーティングシステムエミュレータのアップデートを実行するために、以下の2つのコマンドを実行する必要があります。

`apt update`

`aptアップグレード`

すべてのオプションを確認する Y(はい)



Termux

Home \$ apt update

\$

apt upgrade



```

Welcome to Termux!

Wiki:           https://wiki.termux.com
Community forum: https://termux.com/community
Gitter chat:    https://gitter.im/termux/termux
IRC channel:    #termux on freenode

Working with packages:

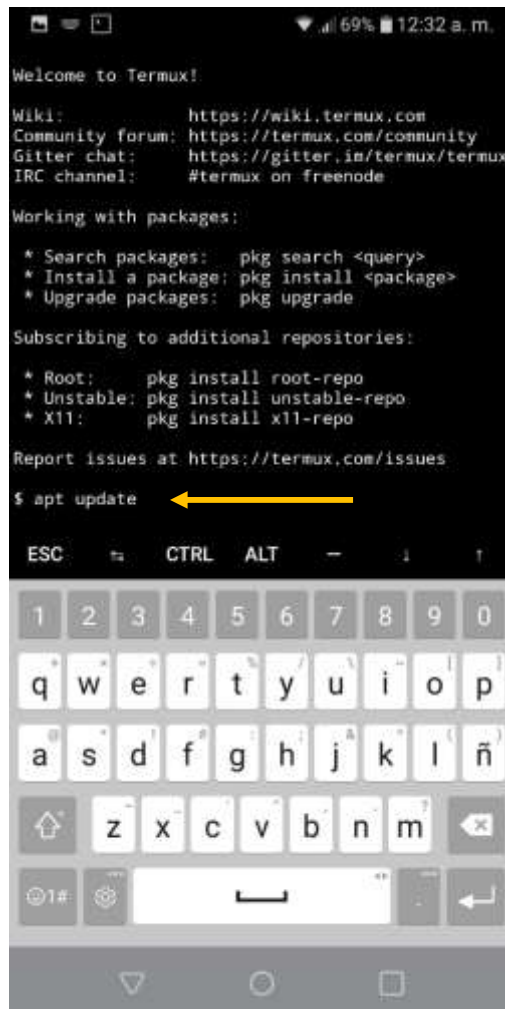
* Search packages:  pkg search <query>
* Install a package: pkg install <package>
* Upgrade packages:  pkg upgrade

Subscribing to additional repositories:

* Root:   pkg install root-repo
* Unstable: pkg install unstable-repo
* X11:    pkg install x11-repo

Report issues at https://termux.com/issues

$
  
```



```

Welcome to Termux!

Wiki:           https://wiki.termux.com
Community forum: https://termux.com/community
Gitter chat:    https://gitter.im/termux/termux
IRC channel:    #termux on freenode

Working with packages:

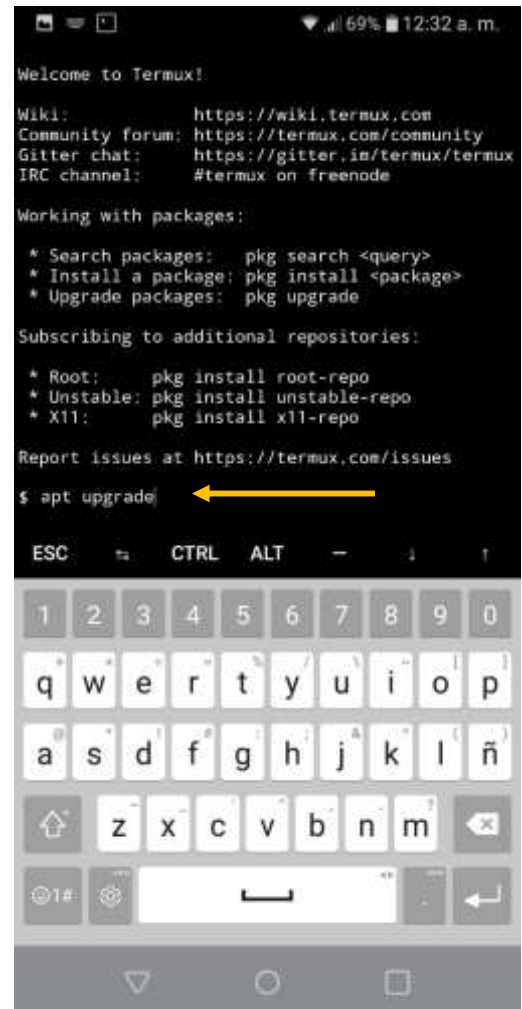
* Search packages:  pkg search <query>
* Install a package: pkg install <package>
* Upgrade packages:  pkg upgrade

Subscribing to additional repositories:

* Root:   pkg install root-repo
* Unstable: pkg install unstable-repo
* X11:    pkg install x11-repo

Report issues at https://termux.com/issues

$ apt update
  
```



```

Welcome to Termux!

Wiki:           https://wiki.termux.com
Community forum: https://termux.com/community
Gitter chat:    https://gitter.im/termux/termux
IRC channel:    #termux on freenode

Working with packages:

* Search packages:  pkg search <query>
* Install a package: pkg install <package>
* Upgrade packages:  pkg upgrade

Subscribing to additional repositories:

* Root:   pkg install root-repo
* Unstable: pkg install unstable-repo
* X11:    pkg install x11-repo

Report issues at https://termux.com/issues

$ apt upgrade
  
```

## 5. Termux内のストレージ構成。

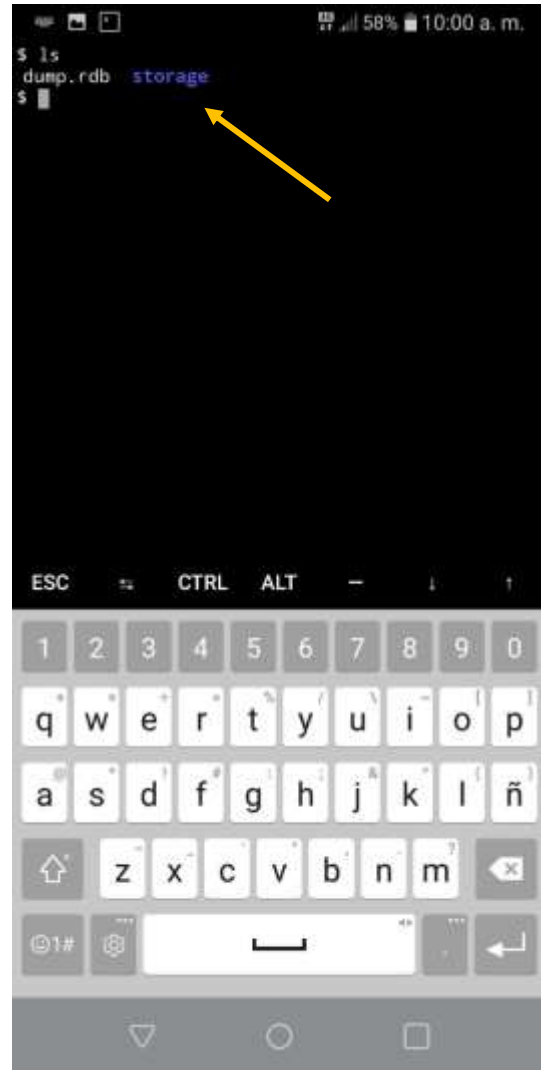
Termuxシステムをアップデートしてアップグレードした後、Termuxシステムで電話機の内部ストレージを見る方法を設定していきます。 これにより、Termuxと電話機内の当社の情報をやり取りできるようになります。

これは、Termuxターミナルで以下のコマンドを実行することで、簡単かつ迅速に行うことができます

。

\$ termux-setup-storage

先ほどのコマンドを実行すると、Termuxでの仮想ストレージ（ディレクトリ）の作成を確認するウィンドウが表示されます。コマンドを与えて検証します。ls



## 6. SSH（Secure Shell）サーバのインストール。

\$ apt install openssh

\$ apt install sshpass

\$ apt install openssh

```

$ apt install openssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  krb5 ldns libdb libedit termux-auth
The following NEW packages will be installed:
  krb5 ldns libdb libedit openssh termux-auth
0 upgraded, 6 newly installed, 0 to remove and 0
not upgraded.
Need to get 2255 kB of archives.
After this operation, 11.9 MB of additional disk
space will be used.
Do you want to continue? [Y/n] Y
Get:1 https://dl.bintray.com/termux/termux-packa
ges-24 stable/main arm libdb arm 18.1.32-4 [465
kB]
Get:2 https://dl.bintray.com/termux/termux-packa
ges-24 stable/main arm krb5 arm 1.18.1 [839 kB]
24% [2 krb5 131 kB/839 kB 16%]

```

\$ apt install sshpass

```

$ apt install sshpass
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  sshpass
0 upgraded, 1 newly installed, 0 to remove and 0
not upgraded.
Need to get 7158 B of archives.
After this operation, 57.3 kB of additional disk
space will be used.
0% [Working]

```

モバイルスマートフォンにlocalhost SSHサーバ用の通信ネットワークを設置して終了しました。

## 7. 携帯電話（スマホ）でのSSHサーバの設定。

携帯電話内のSSHサーバを利用することで、PCから携帯電話への接続が可能になり、より速く快適に作業ができるようになりますし、携帯電話内のSSHサーバのサービスが正常に動作するかどうかの確認にも役立ちます。

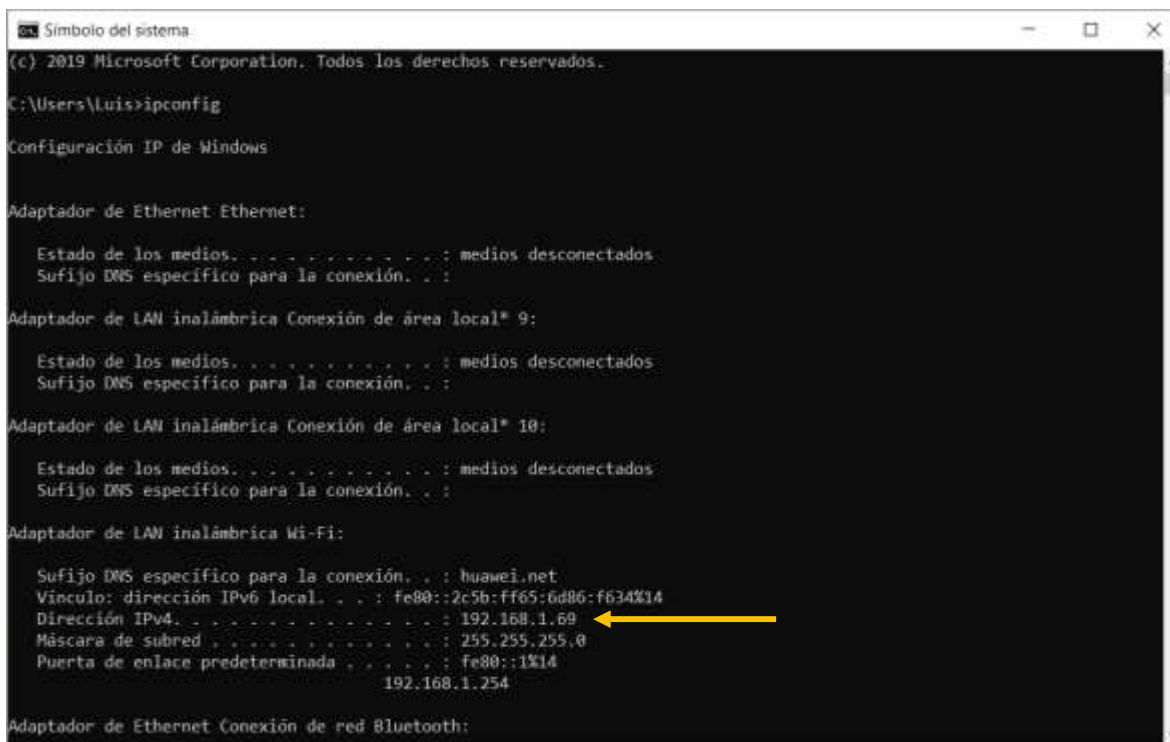
まずはモバイルとPCを**同じWiFiネットワーク**に接続して、お互いに顔が見えるようにすることから始めます。IPまたはアドレスは、192.168.XXX.XXXに似ている必要があります。XXXの値は、各コンピュータでランダムに割り当てられる可変数字です。

この例は、LG Q6携帯電話とWindows 10 Homeを搭載したPCでテストしました。

PCがWiFiに接続しているIPやアドレスを確認するには、Windowsで端末を開く必要があります。

検索拡大鏡がある下のパネルでcmdと書いてEnterキーを押します。ターミナルが開きますので、その中にコマンドを書きます。

```
C:\User_Name> ipconfig
```



```
Símbolo del sistema.
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Luis>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de LAN inalámbrica Conexión de área local* 9:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de LAN inalámbrica Conexión de área local* 10:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de LAN inalámbrica Wi-Fi:

    Sufijo DNS específico para la conexión. . . : huawei.net
    Vínculo: dirección IPv6 local. . . : fe80::2c5b:ff65:6d86:f634%14
    Dirección IPv4. . . . . : 192.168.1.69
    Máscara de subred. . . . . : 255.255.255.0
    Puerta de enlace predeterminada. . . : fe80::1%14
                                     192.168.1.254

Adaptador de Ethernet Conexión de red Bluetooth:
```

これでPCに割り当てられているIPが192.168.1.69であることがわかりますが、これはそれぞれのケースで異なる可能性が高いです。

注意：「IPv4アドレス」と書かれているアドレスは、ゲートウェイと混同しないように取る必要があります。

携帯電話の場合は、次のコマンドを入力して、携帯電話を持っているSSHサーバに接続するためのユーザ名を知る必要があります。

## 誰もが

後日、このユーザーにパスワードを与えなければならないので、以下のコマンドを実行しなければなりません。

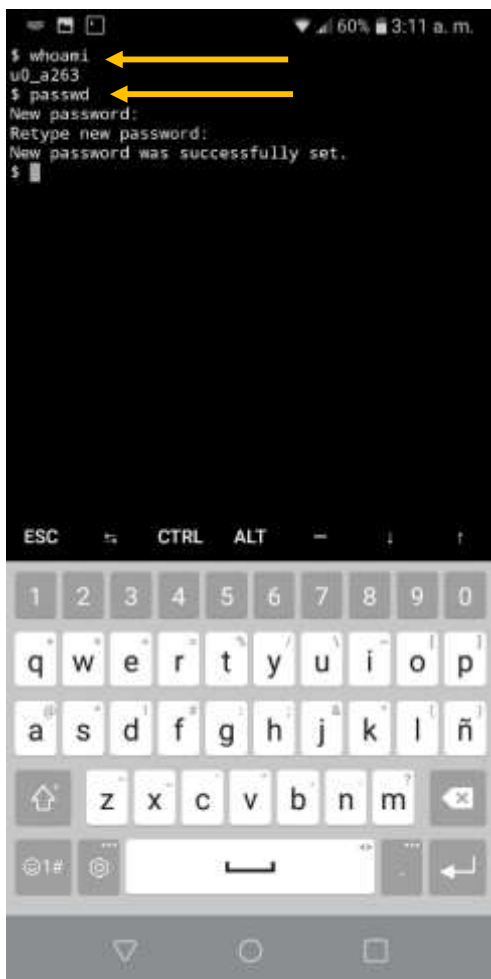
## パスワードを指定する

それは私たちにパスワードを入力してEnterを押すように求められますが、再びそれは私たちがそれを確認し、Enterを押して、それが正常にされている場合は、**"新しいパスワードが正常に設定され**

ました"エラーをマーキングの場合には、パスワードが正しく入力されていない可能性があることを確認してください。再度手順を実行してください。

そして、TermuxでどのようなIPを持っているかを知るために、次のコマンドを入力します。

ifconfig -a



あとは

スマホのSSHサーバサービスを起動して、PCからのセッションを受信できるようにするだけです。

Termuxターミナルで以下のコマンドを実行してみましたが、何の結果も得られませんでした。

sshd

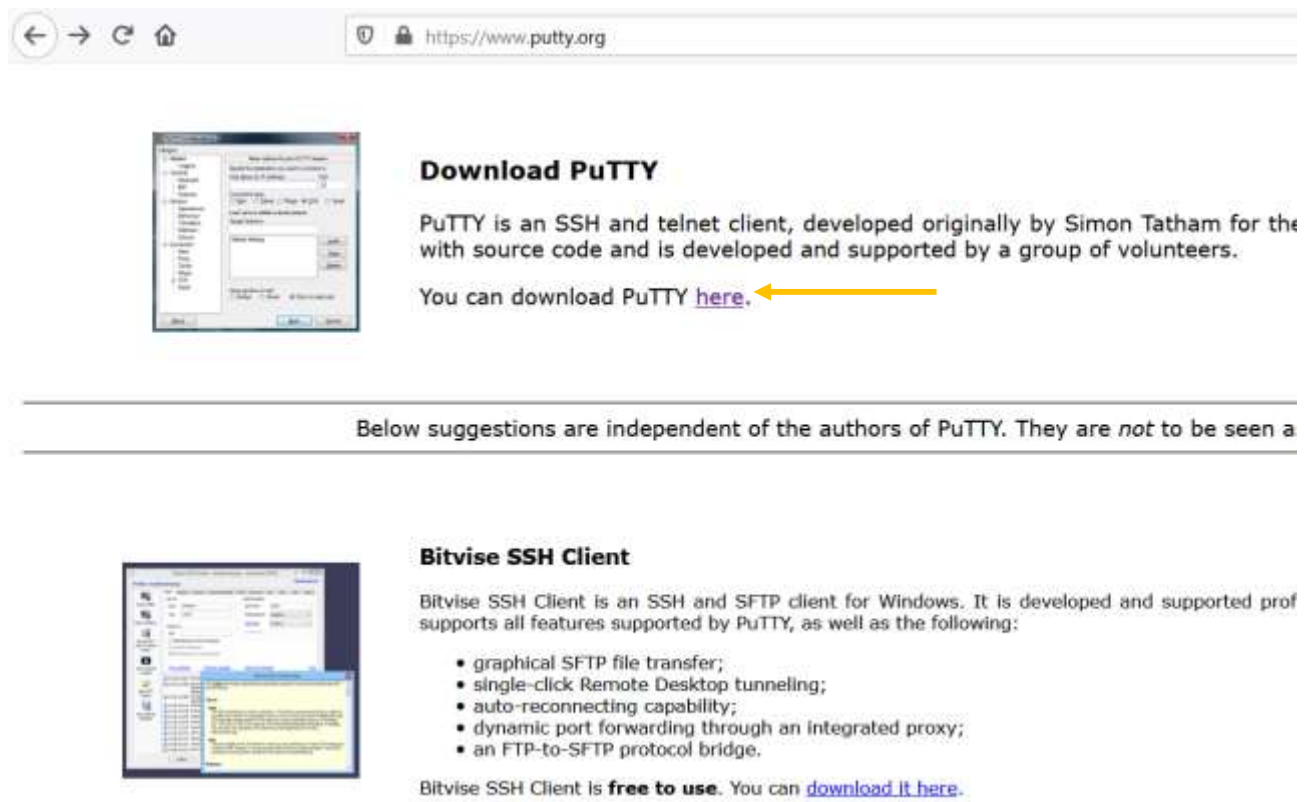




あとは、PCから電話のSSHサーバと通信するプログラムをPCにインストールする必要があります。

<https://www.putty.org>

ここからPuTTYをダウンロードできます」というリンクがある場所を選択してください。



32ビット版を選択してください、それはあなたのシステムが64ビットであるかどうかは問題ではありませんうまく動作します。

## Download PuTTY: latest release

[Home](#) | [FAQ](#) | [Feedback](#) | [Licence](#) | [Updates](#) | [Mirrors](#)  
Download: [Stable](#) | [Snapshot](#) | [Docs](#) | [Contact](#)

This page contains download links for the latest released version of PuTTY. Currently this is 0.73, released on 2019-09-29.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternative

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date, to see if the problem has already been fixed in those versions.

**Package files**

You probably want one of these. They include versions of all the PuTTY utilities.

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

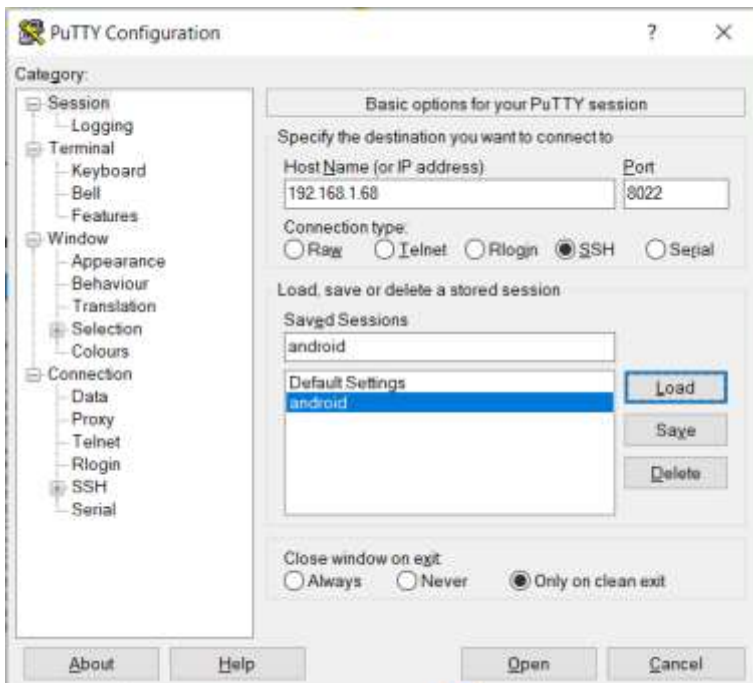
**MSI ('Windows Installer')**

|         |  |             |             |
|---------|--|-------------|-------------|
| 32-bit: | <a href="#">putty-0.73-installer.msi</a>       | (or by FTP) | (signature) |
| 64-bit: | <a href="#">putty-64bit-0.73-installer.msi</a> | (or by FTP) | (signature) |

**Unix source archive**

|          |                                   |             |             |
|----------|-----------------------------------|-------------|-------------|
| .tar.gz: | <a href="#">putty-0.73.tar.gz</a> | (or by FTP) | (signature) |
|----------|-----------------------------------|-------------|-------------|

PCにダウンロードされたら、実行してデフォルトのオプションでインストールします。その後、PuTTYアプリケーションを起動します。



このセッションでは、携帯電話にインストールしたOpensshサーバーのデータを入力していきます。

携帯電話のIPを入力します。

ホスト名またはIPアドレス。

192.168.1.68 (IP例)

港だ

8022 (モバイルSSHサーバのデフォルトポート)。

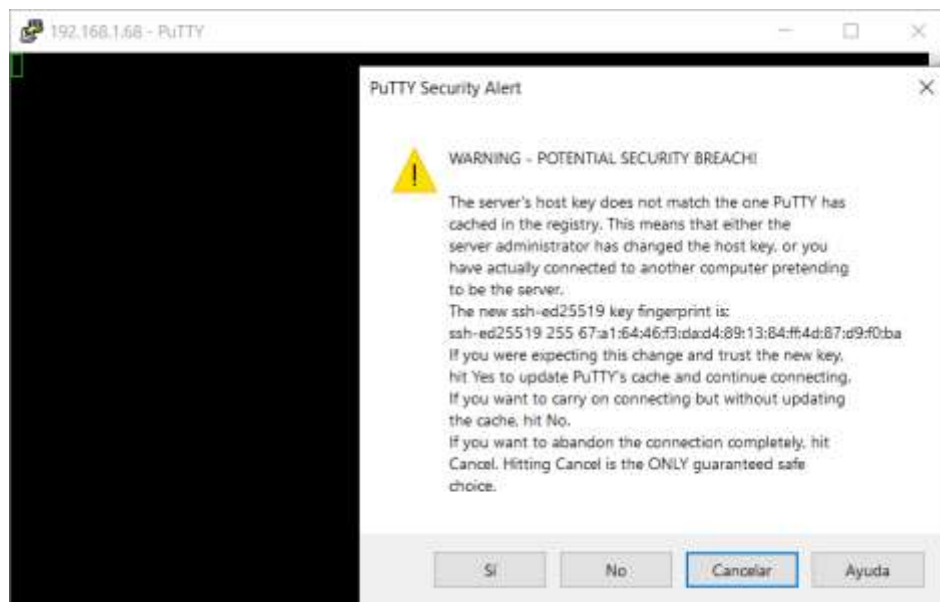


保存されたセッション」でセッションに名前をつけて保存ボタンをクリックします。

後で下の部分で我々は"開く"ボタンを与えるサーバーへの接続を開くために押す。

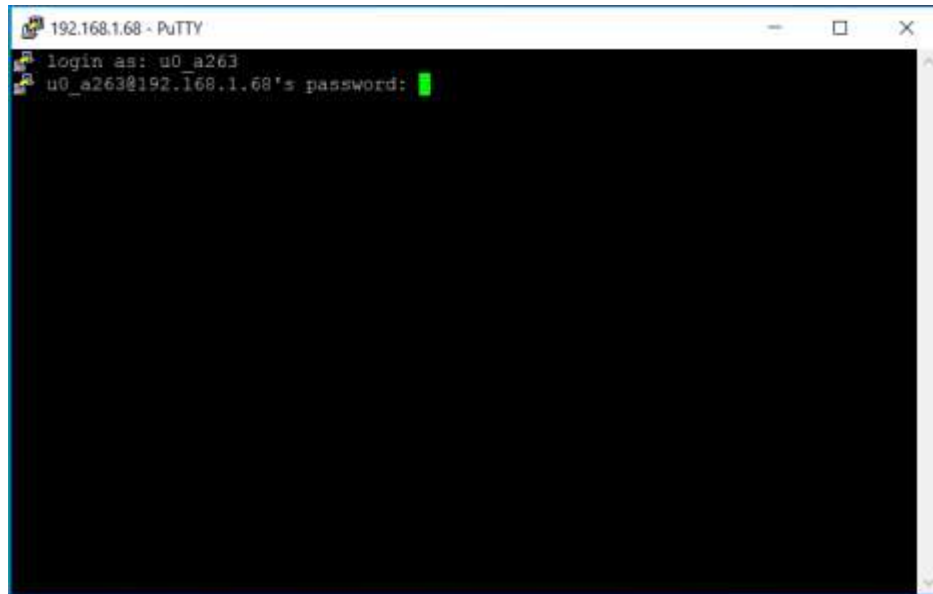
初めて接続したパソコンでは、「はい」ボタンをクリックして情報暗号化キーの確認を求められます

。

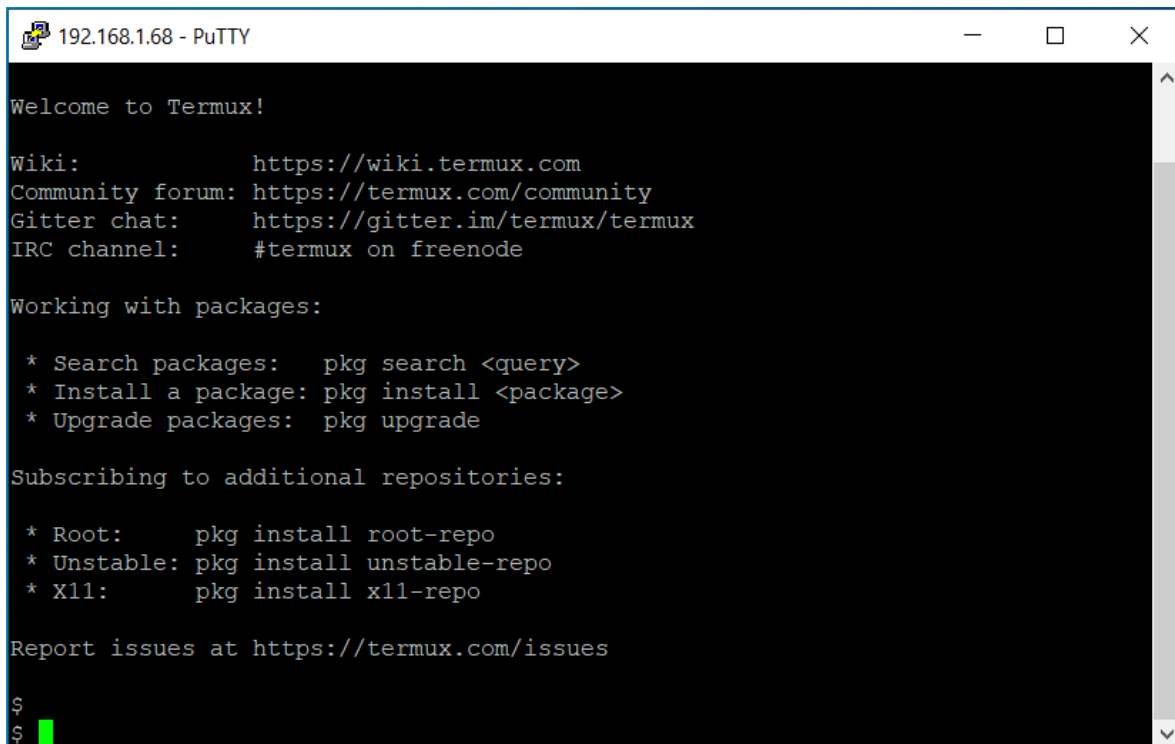


後日、接続するユーザーを聞かれることになります。以前に取得した情報（ユーザーとパスワード）を利用させていただきます。

としてログインでは：私たちは私たちのユーザーを入力し、Enterを与える必要がありますが、その後、我々は再びEnterボタンを与えるパスワードを求められます。



データに間違いがなければ、PC（クライアント）から電話機（SSHサーバ）上で行われるSSH（Secure Shell）セッションになります。



**重要な注意:** 同じWiFiに接続されているPCのIP(アドレス)と携帯電話のIP(アドレス)は、おそらく私たちが切断して再接続するたびに変更されることを覚えているので、我々は各デバイスが持っているアドレスをダブルチェックする必要があります, これは、携帯電話のSSHサーバとPC(クライアント)を介してデバイス間の接続の成功を保証します.

## 8. Ambientes Blockly (App Inventor, AppyBuilder y Thunkable)。

App Inventorは、Google Labsが作成したAndroid OS用のアプリケーションを構築するためのソフトウェア開発環境です。ユーザーは、視覚的に、そして基本的なツールのセットから、一連のブロックをリンクしてアプリケーションを作成することができます。システムは無料で、Webから簡単にダウンロードすることができます。App Inventorで作成されたアプリケーションは、プログラミング言語の知識が不要なため、非常に簡単に作成することができます。

AppyBuilderやThunkableなど、Blocklyの技術を使用している現在の環境はすべて無料版があり、その使用方法は、インターネットを介してさまざまなサイトで使用することができますし、自宅にインストールすることもできます。

Mini BloclChain アーキテクチャを構成するブロックは、App inventor と AppyBuilder でテストされていますが、コードの最適化のため、他のプラットフォームでも動作するはずですが。

オンライン版。

App Inventor。

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

AppyBuilderです。

<http://appybuilder.com/>

サンカブル。

<https://thunkable.com/>

お使いのパソコン（PC）にインストールするバージョン。

<https://sites.google.com/site/aprendeappinventor/instala-app-inventor>

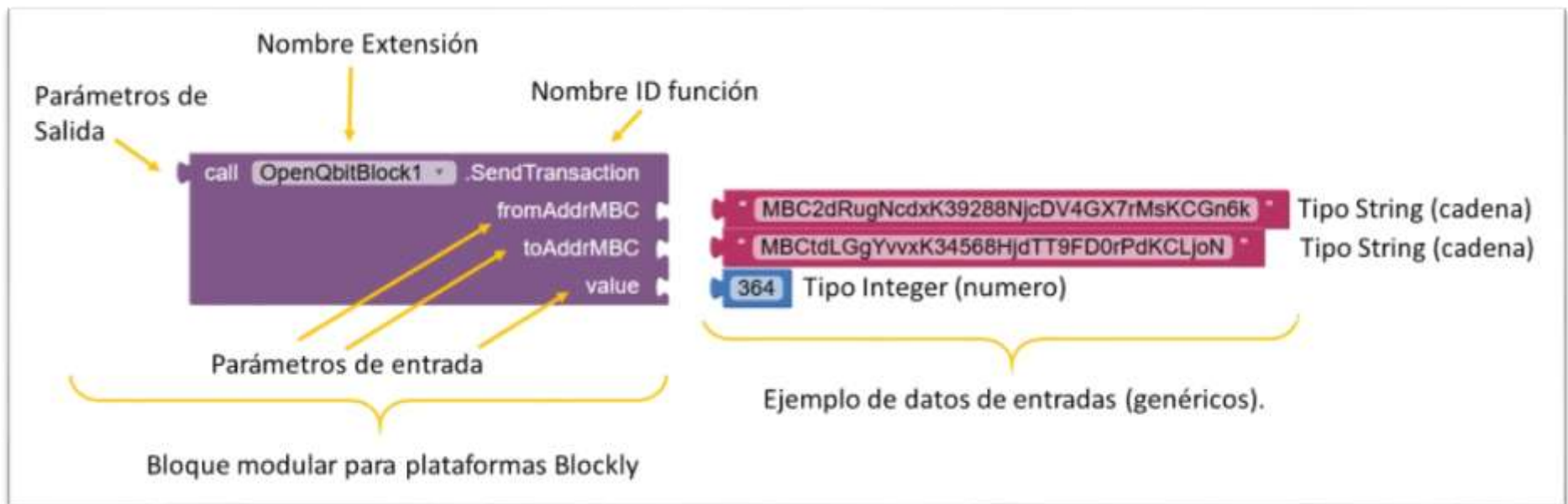
Blocklyブロックの開発者のための環境。

<https://editor.appybuilder.com/login.php>

## 9. ミニQRNGのブロックの定義と使い方

まずは、すべてのブロックが持つであろうデータの分布、使用の構文、構成について説明します。

次の例では、入力データのタイプと同様に、モジュールブロックとその入出力パラメータを見ることができ、これらのデータのタイプはString（文字列）またはInteger（整数または10進数）です。どのように使用されているかを示し、正しく機能するように設定します。



各モジュールブロックはその説明を持ち、入力パラメータとして使用される他のブロックの必須または任意の依存関係がある場合には名前が付けられ、統合プロセスが発表されます。まずは OpenQbitQRNGwithSSH 拡張モジュールのブロックから見ていきましょう。

## 10進乱数量子数生成ブロック - (ApiGetQRNGdecimal)



入力パラメータ: qty <整数>

出力パラメータ: 入力されたランダムな量子十進数の量"qty"をJSON形式で0と1の範囲内で与えます。

例。

qty = 5; output: {"result": [0.5843012986202495, 0.7746497687824652, 0.05951126805960929, 0.1986079055812694, 0.03689783439899279]}。

説明: 量子乱数発生器 (QRNG) API

## 10進乱数量子数生成ブロック - (ApiGetQRNGinteger)



入力パラメータ: qty <整数>, min <整数>, max <max>

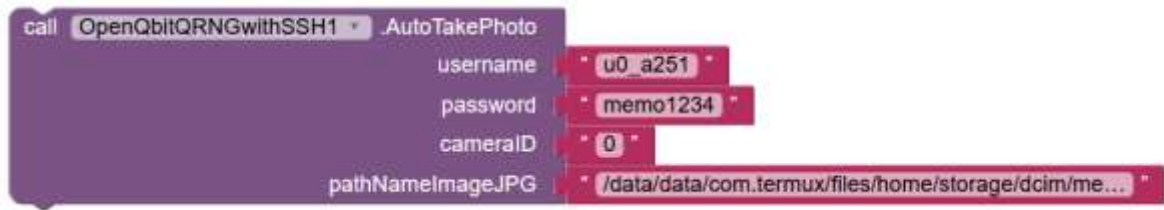
出力パラメータ: 入力されたランダムな量子整数の量"qty"を与えます。

例。

qty = 8, min = 1, max = 100; output: {"result": [3, 53, 11, 2, 66, 44, 9, 78]}。

説明: 量子乱数発生器 (QRNG) API

## 自動撮影をブロックする - (AutoTakePhone)



入力パラメータ: username <文字列>, password <文字列>, cameraID <文字列>, pathNameImageJPG <文字列>

必須の依存関係: このブロックを使用するには、2つのソフトウェアの依存関係を満たす必要があります。Termux-APIモジュールをTermuxターミナルにインストールしてください。このモジュールには、自動的に写真を撮影し、以前にインストールされたSSHサーバーをアップロードするプロセスが含まれています。

出力パラメータ: 指定されたパスでJPG形式の写真（画像）を配信します。あなたがしなければならない道の中で

説明: ユーザーの介入なしで自動的にJPG写真を作成します。

Termux-APIをインストールするには、Termuxターミナルで以下のコマンドを実行する必要があります。

pkg install termux-api

```

$ pkg install termux-api
Ign:2 https://dl.bintray.com/grimler/game-packag
es-24 games InRelease
Ign:3 https://dl.bintray.com/grimler/science-pac
kages-24 science InRelease
Ign:1 https://dl.bintray.com/termux/termux-packa
ges-24 stable InRelease
Get:5 https://dl.bintray.com/grimler/game-packag
es-24 games Release [5344 B]
Get:6 https://dl.bintray.com/grimler/science-pac
kages-24 science Release [6191 B]
Get:4 https://dl.bintray.com/termux/termux-packa
ges-24 stable Release [8255 B]
Get:7 https://dl.bintray.com/grimler/game-packag
es-24 games Release.gpg [475 B]
Get:8 https://dl.bintray.com/grimler/science-pac
kages-24 science Release.gpg [475 B]
Get:9 https://dl.bintray.com/termux/termux-packa
ges-24 stable Release.gpg [821 B]
0% [8 Release.gpg gpgv 6191 B]

```

```

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  termux-api
1 upgraded, 0 newly installed, 0 to remove and 7
4 not upgraded.
Need to get 21.2 kB of archives.
After this operation, 4096 B of additional disk
space will be used.
Get:1 https://dl.bintray.com/termux/termux-packa
ges-24 stable/main arm termux-api arm 0.50-1 [21
.2 kB]
Fetched 21.2 kB in 1s (18.7 kB/s)
(Reading database ... 25317 files and directorie
s currently installed.)
Preparing to unpack .../termux-api_0.50-1_arm.de
b ...
Unpacking termux-api (0.50-1) over (0.50) ...
Setting up termux-api (0.50-1) ...
$

```

自分のモバイル端末（スマートフォン）が持っているID（フォトレンズ識別子）の数、数、位置を調べるには、Termux端末で以下のコマンドを実行します。

\$ termux-camera-info



```

$ termux-camera-info
[
  {
    "id": "0",
    "facing": "back",
    "jpeg_output_sizes": [
      {
        "width": 4160,
        "height": 3120
      },
      {
        "width": 4160,
        "height": 2340
      },
      {
        "width": 4160,
        "height": 2080
      },
      {
        "width": 3264,
        "height": 2448
      }
    ]
  }
]

```

```

    "manual_sensor",
    "manual_post_processing",
    5,
    6,
    4,
    7,
    "raw"
  ],
  {
    "id": "1",
    "facing": "front",
    "jpeg_output_sizes": [
      {
        "width": 2560,
        "height": 1920
      },
      {
        "width": 2560,
        "height": 1600
      },
      {
        "width": 2560,

```

今回の例では、使用したLG Q6 smarphoneには、背面に「0」、前面に「1」という2つのIDが付いています。

さて、リアレンズのID "0"を使って写真を撮るためのAPIをテストしてみましょう、私たちの場合はtest.jpgでそれにアルビタリー名を与えます

APIはJPG形式の写真のみを配信していることを覚えておいてください。

```
$ termux-camera-photo -c 0 test.jpg
```

前のコマンドは、名前test.jpgのファイルを自動的に作成している必要がありますが、そうであれば、我々はブロック（AutoTakePhoto）を使用することができます、コマンドを使用してローカルSSHサーバを起動することを忘れないでください： `$ sshd`

**注意：**変数pathNameImageJPGでは、スマートフォンのストレージにアクセスするためのTermux端末内のパスを使用することを考慮する必要があります。

```
/data/data/com.termux/files/home/storage/dcim/example.jpg
```

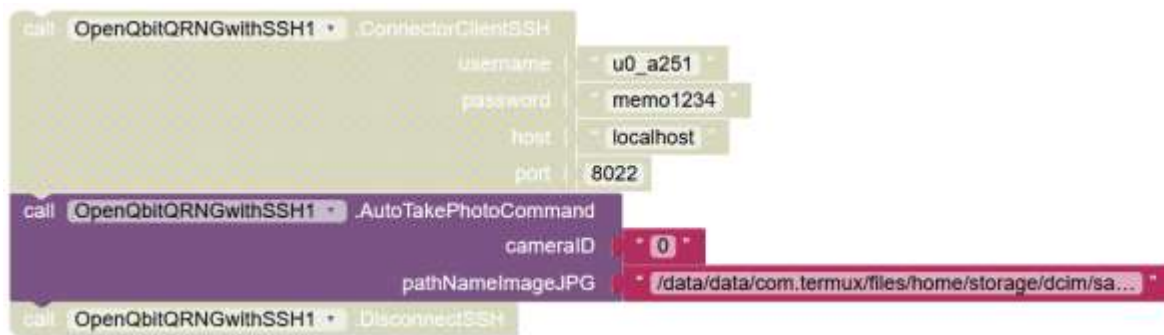
アンドロイドでの前のルートもそうだろう。

```
/mnt/sdcard/dcim/example.jpg
```

ただし、Termux端末では、携帯電話のパワーアップを見るための有効なルートは、常にデフォルトルートと考えなければならないことを覚えておく必要があります。

```
/data/data/com.termux/files/home/storage
```

自動的に写真を撮ることをブロックする ONLY COMMAND - (AutoTakePhoneCommand)



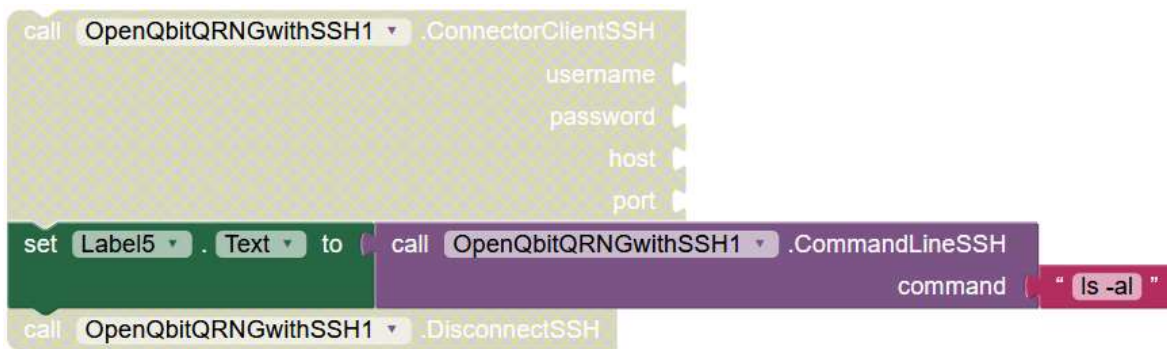
入力パラメータ: cameraID <String>, pathNameImageJPG <String>.

必須の依存関係: Block (ConnectorClientSSH)、Block (DisconnectSSH)。

出力パラメータ：指定されたパスでJPG形式の写真（画像）を配信します。

説明: ユーザーの介入なしで自動的にJPG写真を作成します。ただし、ブロック(AutoTakePhoto)との違いは、このブロックには写真を作成するコマンドが入っているだけで、まずSSHサーバに接続(ConnectClientSSH)してからブロック(DisconnectSSH)を使う必要があります。

Termux端末でコマンドを実行するためのブロック - (CommandLineSSH)



入力パラメータ: コマンド<文字列>

必須の依存関係: Block (ConnectorClientSSH)、Block (DisconnectSSH)。

出力パラメータ: Termux 端末で入力したコマンドを実行します。

説明: 入力されたコマンドを実行し、まずSSHサーバに接続(ConnectClientSSH)し、次にブロック(DisconnectSSH)を使用するためのブロックが必要です。

リモートまたはローカルのSSHサーバに接続するためのブロック - (ConnectorClientSSH)。



入力パラメータ: username <string>, password <string>, host <string>, port<integer>.

出力パラメータ: Termux 端末の ssh サーバとの接続に成功した場合は "Connect SSH" というメッセージを、失敗した場合は NULL というメッセージを出力します。

説明: SSH (Secure Shell) 通信プロトコルを使用して、選択した SSH サーバを Termux 端末に接続するための通信ブロックです。

Base64アルゴリズム(DecoderFileBase64)でファイルをデコードするブロック。

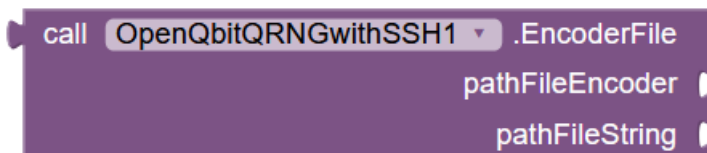


入力パラメータ: pathFileBase64 <String>, pathFileOrigin <String>

出力パラメータ: ブロックに入力されたソースファイル (EncoderFileBase64)

説明: Base64 ファイルをブロック内に挿入された元のファイル (EncoderFileBase64) に変換します。

ブロックは、ファイルを Base64 形式に変換します - (EncoderFile)

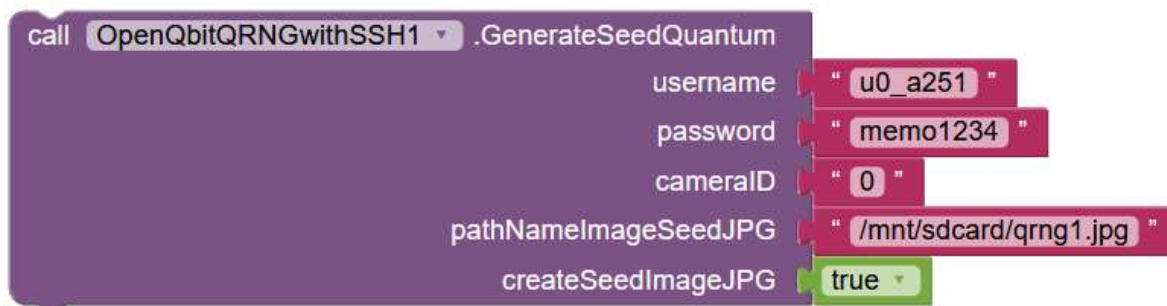


入力パラメータ: pathFileOrigin <String> , pathFileString <String>

出力パラメータ: Base64エンコードされたファイル

説明: 任意の形式のソースファイルをBase64ファイルに変換します。ファイル名は任意で、ユーザーが選択することができます。

QRNG(Quantum Random Number Generator)を生成するブロック - (GenerateSeedQuantum)



入力パラメータ: username <String>, password <String>, cameraID <String>, pathNameImageJPG <String>. createSeedImageJPG <Boolean>.

ブロックが実行されるたびに、入力されたパス名で新しいシードJPG画像が作成されるときにブール値が"True"の場合。ブーリアン値が "False" の場合は JPG 画像 (写真) を撮影するオプションを無効にし、選択範囲内のどこに画像があるかを手動で指定します。

注：QRNGを生成するための最良の結果は、"RAW"フォーマットされた画像に基づいています。DNG形式の例。

必須依存関係：前述のTermux-APIはブロック（AutoTakePhoto）にインストールされている必要があります。

出力パラメータ：イベントが実行され（DataQRNG）、2つの値が与えられます。

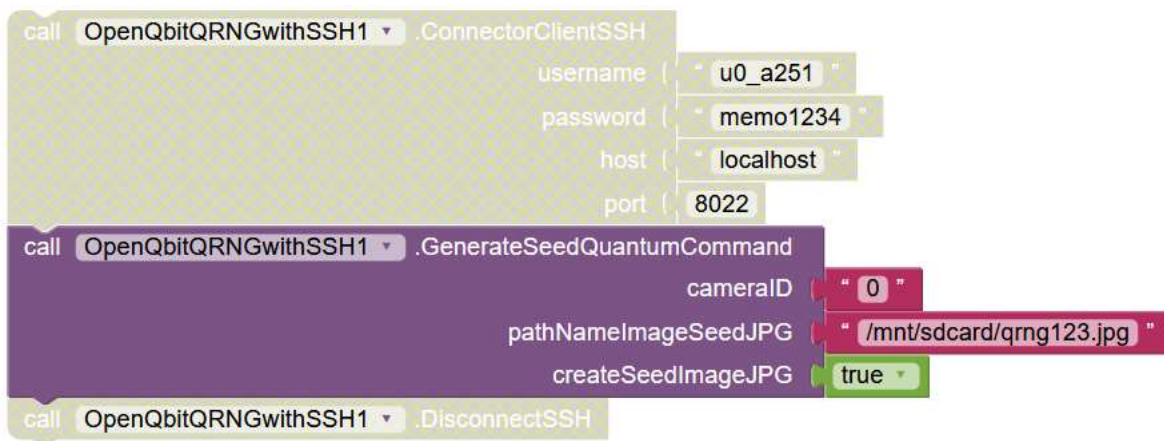
qrngBinary. - ランダムな 2 進数の文字列.

Sha512ImageSeed - 乱数が発生したJPGのシード画像からSha512を作成しました。



説明：携帯電話のカメラの光学センサーを介して量子乱数（QRNG）を生成します。ランダムな写真を撮影し、それにアルゴリズムを適用して2進数の文字列を配信します。

QRNG(Quantum Random Number Generator)を生成するブロック (GenerateSeedQuantumCommand)。



入力パラメータ: cameraID <String>, pathNameImageJPG <String>. createSeedImageJPG <Boolean>.

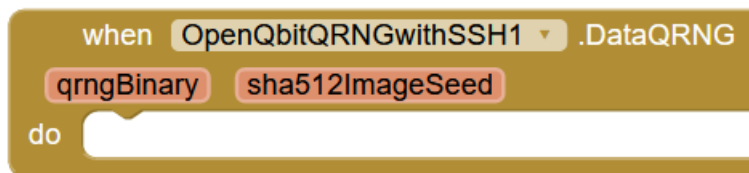
ブロックが実行されるたびに、入力されたパス名で新しいシードJPG画像が作成されるときにブール値が"True"の場合。ポリアン値が "False" の場合は JPG 画像 (写真) を撮影するオプションを無効にし、選択範囲内のどこに画像があるかを手動で指定します。

必須の依存関係: Block (ConnectorClientSSH)、Block (DisconnectSSH)。

出力パラメータ: イベントが実行され (DataQRNG)、2つの値が与えられます。

qrngBinary. - ランダムな 2 進数の文字列.

Sha512ImageSeed - 乱数が発生したJPGのシード画像からSha512を作成しました。



説明: 携帯電話のカメラの光学センサーを介して量子乱数 (QRNG) を生成します。 ランダムな写真を撮影し、それにアルゴリズムを適用して2進数の文字列を配信します。



ただし、ブロック(GenerateSeedQuantum)との違いは、このブロックにはQRNG反応アルゴリズムを実行するためのコマンドが含まれているだけで、まずSSHサーバに接続(ConnectClientSSH)してからブロック(DisconnectSSH)を利用する必要があります。

画像（写真）のシャノンエントロピーを取得するブロック - (GetShannonEntropyFile)



入力パラメータ: ユーザ名 <String>, パスワード <String>, pathFileImage <String>

必須依存関係: Shannon\_entropyモジュールはTermuxターミナルにインストールする必要があります。

出力パラメータ: 画像のエントロピーを提供します。

例。

出力: 8.94596789873

説明: イメージのエントロピーを与えてくれます。エントロピーは良質な乱数を生成するための基本的なパラメータであり、エントロピーが高いほど良い結果が得られます。

Shannon entropyモジュールをインストールするには、まずPythonパッケージをインストールし、Termuxターミナルで以下のコマンドでPillowとShannon\_entropyモジュールをインストールする必要があります。

```
apt install Python
```

```
pip install pillow
```

```
pip install Shannon_entropy
```

```
$ apt install python
Reading package lists... Done
Building dependency tree
Reading state information... Done
python is already the newest version (3.8.3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
$
```

```
$ pip install Pillow
Requirement already satisfied: Pillow in /data/d
ata/com.termux/files/usr/lib/python3.8/site-pack
ages (7.2.0)
$
```

```
$ pip install shannon_entropy
Requirement already satisfied: shannon_entropy i
n /data/data/com.termux/files/usr/lib/python3.8/
site-packages (0.2.1)
Requirement already satisfied: Pillow in /data/d
ata/com.termux/files/usr/lib/python3.8/site-pack
ages (from shannon_entropy) (7.2.0)
$
```



次に、Termuxの"Home"ディレクトリに、以下のコードを含む"entropy.py"というPythonファイルを作成します。

PILから画像をインポート

インポート数学

Shannon\_entropyからのインポート



## システムのインポート

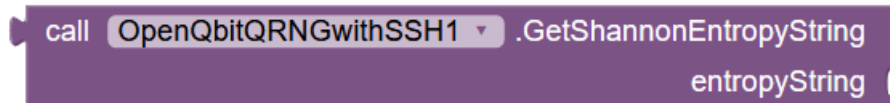
```
Img=Image.open(sys.argv[1])

print(Shannon_entropy(img))
```

ファイルを保存して、ブロック（GetShannonEntropyFile）で使用する環境を用意しています。

**Tip:** 実際、このPythonのインストールでは、この言語で独自のプログラムを作成し、ブロック (ConnectorClientSSH)を介して実行することができます。

文字列からシャノンエントロピーを取得するブロック - (GetShannonEntropyString)



入力パラメータ： entropyString>String

出力パラメータ: 文字列のエントロピーを提供します。

例。

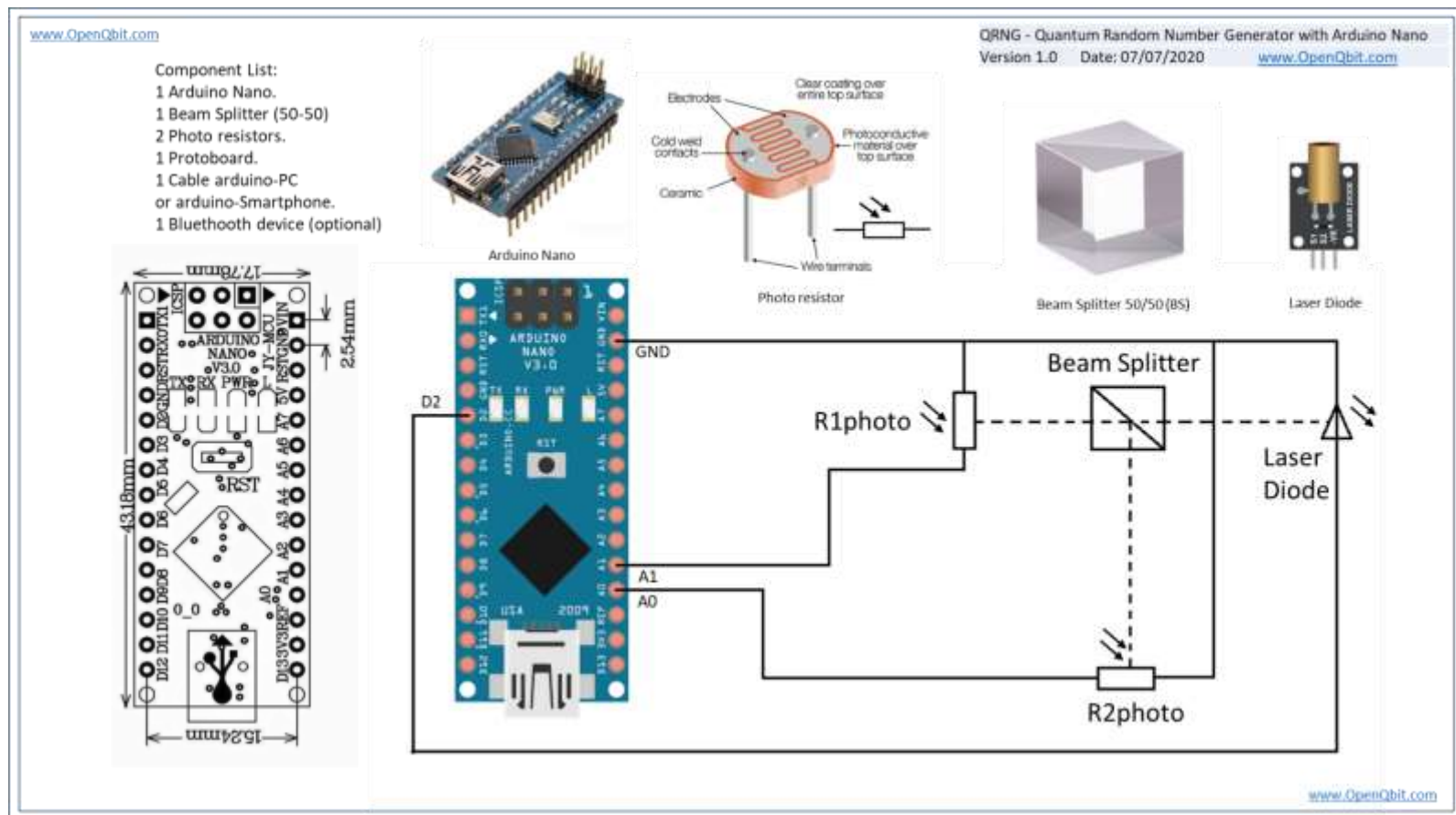
出力： 5.76002345671

説明: 文字列のエントロピーを与えてくれます。エントロピーは良質な乱数を生成するための基本的なパラメータであり、エントロピーが高いほど良い結果が得られます。

## 10. QRNGの「ハードウェア」デバイスの作成

これからは、自宅で簡単に組み立てられる安価な部品で量子乱数（QRNG）を生成する物理的な「ハードウェア」デバイスを作成していきます

。



[www.OpenQbit.com](http://www.OpenQbit.com)

## 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.....

[www.OpenQbit.com](http://www.OpenQbit.com)

[www.OpenQbit.com](http://www.OpenQbit.com)

QRNG - Quantum Random Number Generator with Arduino Nano  
Version 1.0    Date: 07/07/2020    [www.OpenQbit.com](http://www.OpenQbit.com)

QRNGv10.inoプログラムをコンパイルして、arduous nanoにアップロードすると...

```

QRNGv1.0 Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda

QRNGv1.0

// triggerPin = 2; // This pin will pulse the quantum circuit
// Qa0Pin = A0; // This pin measures the horizontal polarized photons
// Qa1Pin = A1; // This pin measures the vertically polarized photons
// Qa0 = 0;
// Qa1 = 0;

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

int Random() {
  // Pulse the laser
  digitalWrite(triggerPin, HIGH);
  delay(300);
  digitalWrite(triggerPin, LOW);
  delay(300);
  // Read the photoreceptors
  Qa0 = analogRead(Qa0Pin);
  Qa1 = analogRead(Qa1Pin);
  // Determine random bit
  if(Qa0 > Qa1) { // More photons in the Qa0 mode, return 0
    return 0;
  } else if (Qa1 > Qa0) { // More photons in the Qa1 mode, return 1
    return 1;
  } else {
    // The same number of photons are in both modes
    // This is actually an impossible condition, but for
    // purposes we will simply run the function repeatedly until
    // a random bit can be generated.
  }
  Random();
}

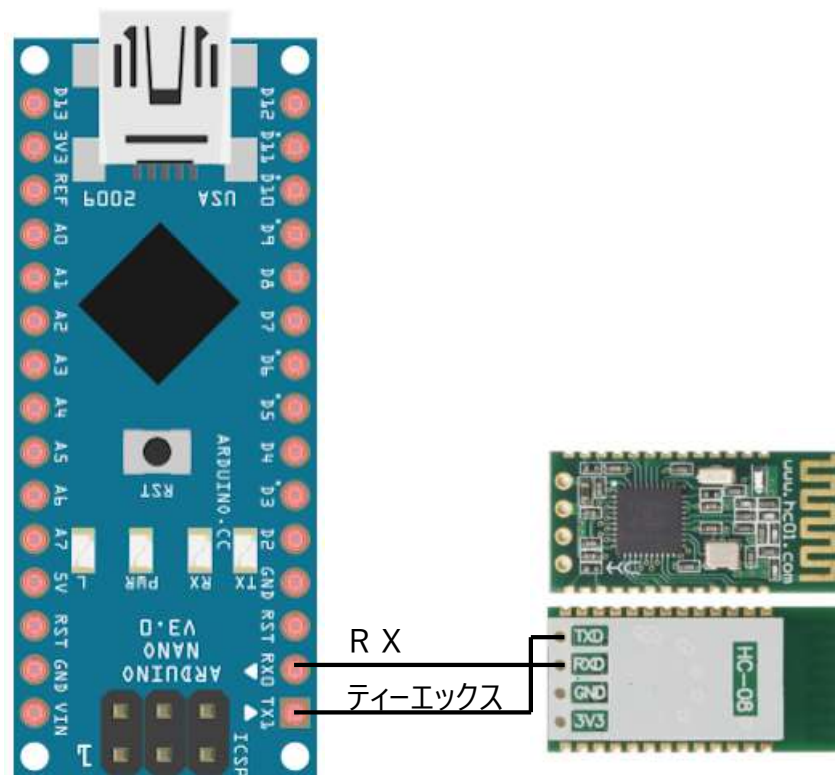
void loop() {
  Serial.print(Random());
}

Compilado
El sketch usa 2362 bytes (74%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.
Las variables Globales usan 120 bytes (94%) de la memoria dinámica, dejando 1860 bytes para las variables locales. El máximo es 2048 bytes.

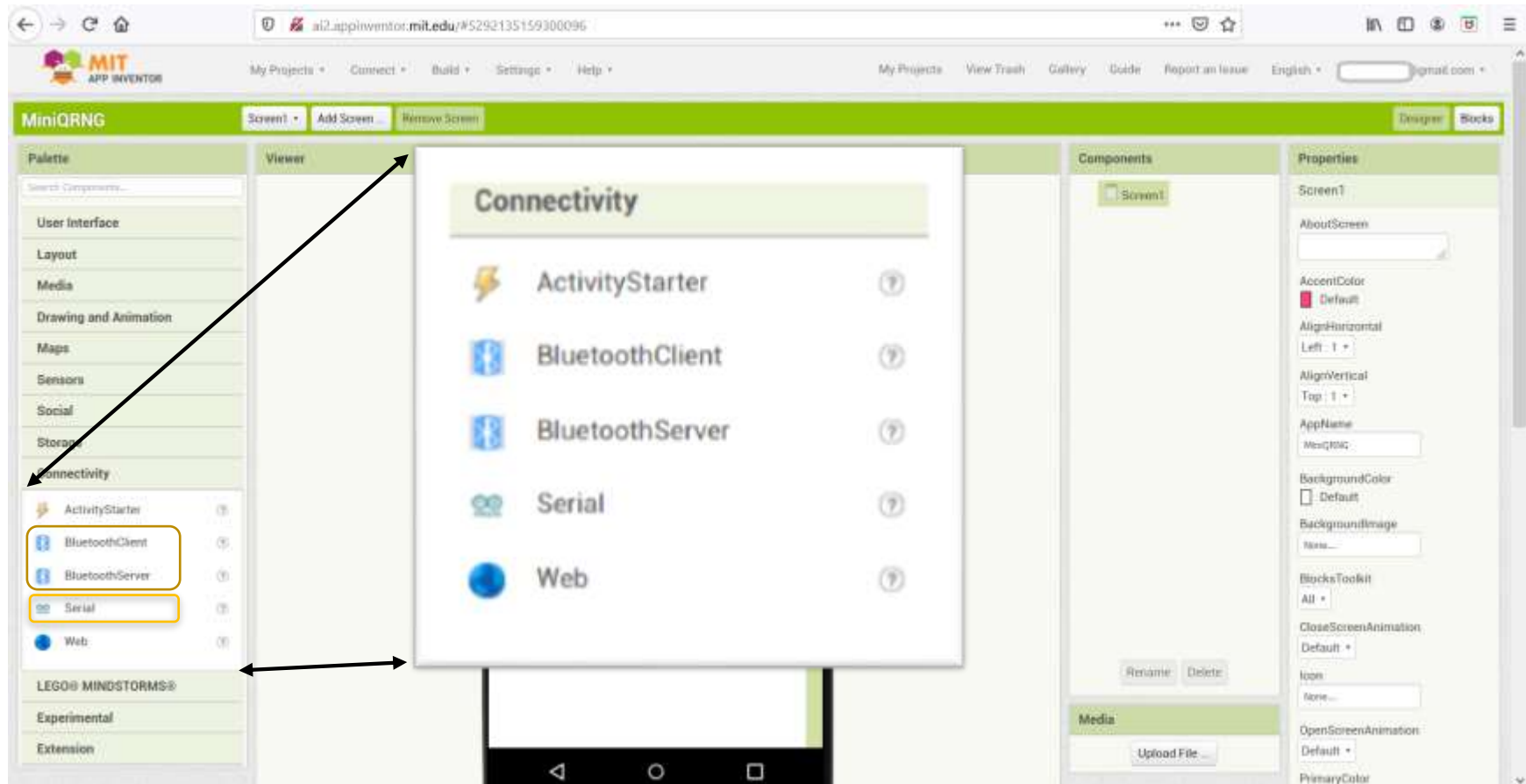
```

arduous nanoとの通信方法は2通りあり、1つはシリアルポート経由で、もう1つはBluetooth接続での通信です。

ブルートゥース接続のために我々はちょうどHC-08モジュールまたは同様のものを購入し、次のようにそれを接続する必要があります非常に簡単です。



以下のシリアルまたはBluetoothコンポーネントを使用して、App InventorをArduinoに接続することができます。



今すぐコンパイルされ、プログラムQRNGv10.inoの唯一の欠如は、データ（量子乱数）を保存するために困難なナノと通信してロードされ、これらはバイナリ形式になりますが、しかし、得られたデータは簡単に16進数や10進数などの最終的な要件に応じて、別の形式に渡すことができます。

最後に、シリアル接続やBluetooth接続の動作例を見るために、いくつかの参考リンクをご紹介します。

すべてのものがApp InventorでテストするBlocklyプログラミングを介していることを覚えておいて、これはすでにArduinoシリアルまたは他のBlocklyシステムとの通信のためのブロックを持っている同様のbluetoothのtarvesをオンラインにすることができます。

[http://kio4.com/appinventor/9A0\\_bluetooth\\_RXTX.htm](http://kio4.com/appinventor/9A0_bluetooth_RXTX.htm)

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

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

## 11. 付録「OpenQbit Quantum Computing」。

### 量子コンピューティングはどのように機能するのでしょうか？<sup>(2)</sup>

デジタルトランスフォーメーションは、これまでにないスピードで世界に変化をもたらしています。デジタル時代が終わろうとしていると思いますか？**デジタルリテラシー**は、社会的・経済的発展のギャップに対処するために、オープンな知識と技術を学ぶためのアクセス可能な機会が急務となっている分野としてすでに指摘されています。デジタル時代のキーコンセプトから学ぶことは、既存のモデルを驚異的なスピードとパワーで変革することができるもう一つの新しい技術の波、すなわち**量子技術の到来が間近に迫っているため、より一層重要になってきます**。

この記事では、従来のコンピューティングと量子コンピューティングの基本的な概念を比較し、他の関連分野への応用を探ります。

量子技術とは何か？

人類は歴史の中で、科学を通して自然の仕組みを理解し、技術を発展させてきました。1900年から1930年の間に、まだ十分に理解されていないいくつかの物理現象の研究は、新しい物理理論である**量子力学**を生み出しました。この理論は、分子や原子、電子などの自然界に生息するミクロの世界の働きを説明し、説明しています。この理論のおかげで、これらの現象を説明することができるようになっただけでなく、素粒子の現実が全く逆に不可解な、ほとんど魔法のような方法で働いていることや、ミクロの世界ではマクロの世界では起こらないような出来事が起こることを理解することができるようになりました。

これらの**量子特性**には、量子重ね合わせ、量子もつれ、量子テレポーテーションなどがあります。


- **量子重ね合わせ**は、粒子が同時に異なる状態になる方法を説明しています。
- **量子もつれ**は、2つの粒子がどのように相関しているかを説明しています。
- **量子テレポーテーション**は、量子もつれを利用して、宇宙空間を移動することなく、ある場所から別の場所へ情報を送ることができます。

量子技術は、このような素粒子の量子的性質に基づいています。



この場合、今日では、量子力学でミクロの世界を理解することで、人々の生活を向上させる技術を発明したり、設計したりすることができるようになっています。量子現象を利用した技術は数多くあり、その中にはレーザーや磁気共鳴イメージング（MRI）のように半世紀以上も前からあるものもあります。しかし、現在、量子コンピューティング、量子情報、量子シミュレーション、量子光学、量子計測、量子時計、量子センサーなどの分野で技術革命が起きています。

量子コンピューティングとは？まず、古典的なコンピューティングを理解する必要があります。



**FIGURA 1.**  
Ejemplos de caracteres en lenguaje binario.

| Character | Bits             |
|-----------|------------------|
| 7         | 111              |
| A         | 01000001         |
| \$        | 00100100         |
| :)        | 0011101000101001 |

量子コンピュータの仕組みを理解するためには、まず、私たちが日常的に使用しているコンピュータ（ここではデジタルコンピュータやクラシックコンピュータと呼ぶ）がどのように動くのかを説明するのが便利です。これらは、タブレットや携帯電話などの他の電子機器と同様に、メモリの基本単位としてビットを使用しています。これは、プログラムやアプリケーションがビット、つまり0と1のバイナリ言語で符号化されていることを意味します。私たちがこれらのデバイスのいずれかと対話するたびに、例えばキーボードのキーを押すことで、0と1の文字列が作成され、破壊され、コンピュータ内部で変更されます。

興味深いのは、この0と1が物理的にコンピュータの中にあるのは何なのかということです。ゼロ状態と1状態は、スイッチの役割を果たすトランジスタと呼ばれる微細な部品を介して循環する電流の有無に対応しています。電流が流れていないときはトランジスタは"オフ"でビット0に対応し、電流が流れているときは"オン"でビット1に対応します。

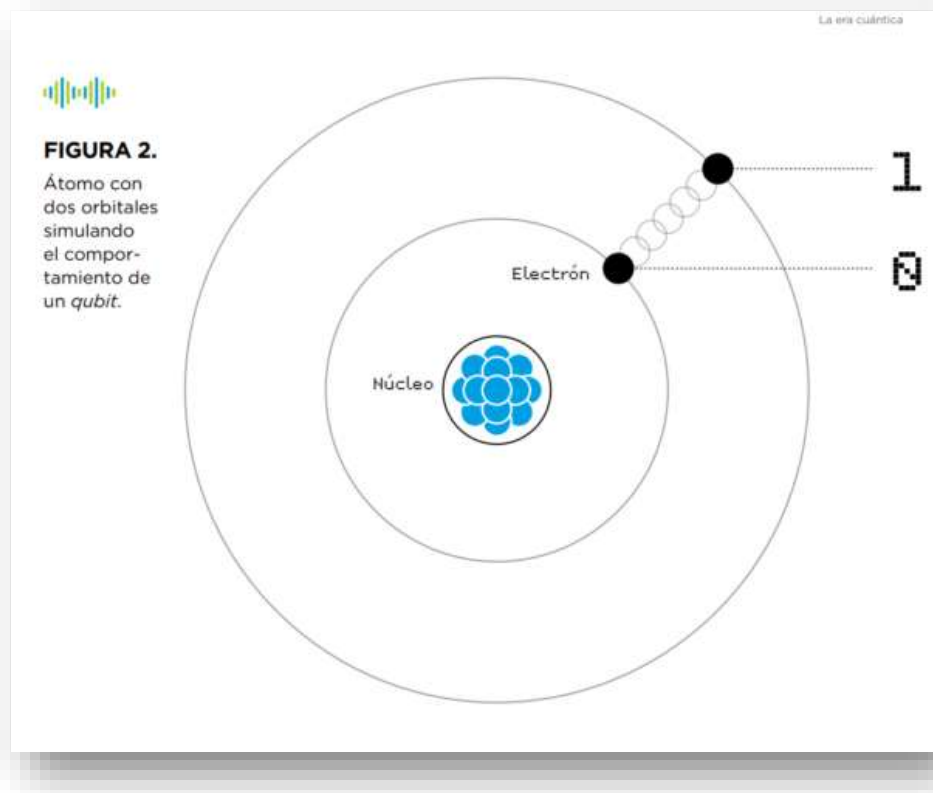
より簡単に言えば、ビット0とビット1が正孔に対応しているようなもので、空の正孔がビット0、電子が占有している正孔がビット1となります。このため、これらの装置はエレクトロニクスと呼ばれて

おり、一例として、図 1 は、ある文字の二進法での書き込みを示している。今日のコンピュータの仕組みがわかったところで、量子の仕組みを理解してみましょう。

## ビットからクビットへ

量子コンピューティングにおける情報の基本的な単位は、量子ビットまたは量子ビットです。キュービットは、定義上、2レベルの量子系であり、ビットと同様に、低レベルの励起またはエネルギーが0と定義された状態に対応する低レベル、または高レベルの励起または1と定義された状態に対応する高レベルの量子系であることができます。しかし、ここに古典的なコンピューティングとの根本的な違いがありますが、クビットは0と1の間の無限の中間状態、例えば0と1の半分、または0の4分の3と1の4分の1の状態のような状態のいずれかにもなります。

量子アルゴリズム、指数数に力効的なユーティング



関的強で率コンピ

量子コンピュータの目的は、量子システムとしての量子ビットのこれらの量子的性質を利用して、オーバーラップとインターリーブを利用した量子アルゴリズムを実行することで、古典よりもはるかに大きな処理能力を提供することにあります。重要なのは、パラダイムの真の変化は、デジタルコン

ピュータや古典的なコンピュータが現在行っていることと同じことを行うのではなく、多くの記事で紹介されているように、量子アルゴリズムによって、ある操作を全く異なる方法で実行することができ、多くの場合、より効率的であることが判明します。

具体的な例を見てみましょう。私たちはボゴタにいと想像してみましょう、私たちはそこに到達するための100万のオプションの中からリマに到達するための最良のルートを知りたい（ $N=1,000,000,000$ ）。コンピュータを使って最適なルートを見つけるためには、100万個のオプションをデジタル化する必要がありますが、これは古典的なコンピュータではビット言語に、量子コンピュータでは量子ビットに翻訳することを意味します。従来のコンピュータでは、目的のパスを見つけるまですべてのパスを1つずつ解析する必要がありましたが、量子コンピュータでは、量子並列性として知られるプロセスを利用して、一度にすべてのパスを考慮することができます。このことは、古典的なコンピュータが $N/2$ ステップまたは反復の順序、つまり50万回の試行を必要とするのに対し、量子コンピュータはレジストリに対する $VN$ 回の操作、つまり1,000回の試行の後に最適なパスを見つけることを意味しています。

前のケースでは2次関数的な利点がありますが、他のケースでは指数関数的な利点もあり、 $n$ 個のクビットで $2^n$ ビットに相当する計算能力が得られます。これを例示するために、約270量子ビットの量子コンピュータでは、宇宙の原子の数（約 $280 \times 10^{21}$ 個と推定される）よりも多くの基底状態を持つことができます。もう一つの例として、2000から2500量子ビットの量子コンピュータでは、現在使用されている暗号（いわゆる公開鍵暗号）を実質的にすべて破ることができると推定されています。

なぜ量子技術を知ることが重要なのか？

私たちは、ブロックチェーン、人工知能、ドローン、モノのインターネット、バーチャルリアリティ、5G、3Dプリンター、ロボット、自律走行車などのさまざまな新興技術が、複数の分野やセクターでますます存在感を増しているデジタルトランスフォーメーションの瞬間にいます。これらの技術は、人間の生活の質を向上させ、開発を加速させ、社会的インパクトを生み出すと呼ばれ、現代では並行して進歩しています。ブロックチェーンとIoT、ドローンと人工知能など、これらの技術を2つ以上組み合わせた製品を開発している企業を目にすることはほとんどありません。しかし、開発の初

期段階では、開発者や技術的なプロフィールを持った人材が不足しているため、収束はまだ保留されているのが現状です。

その破壊的な可能性から、量子技術は、これらすべての新技術に収束するだけでなく、事実上すべての新技術に横断的な影響を及ぼすことが期待されています。量子コンピューティングは、認証、交換、データの安全な保存を脅かすことになり、サイバーセキュリティやブロックチェーンなど、暗号がより関連性の高い役割を持つ技術には大きな影響を与え、マイナーなネガティブな影響はあるが、5G、IoT、ドローンなどの技術にも考慮される。

量子コンピューティングを実践してみませんか？

C, C++, Java, Matlab, Maxima, Python, Octaveなどの様々なプログラミング言語を使用した数十種類の量子コンピュータシミュレータがすでにネット上で利用可能です。また、マイクロソフトが立ち上げたQ#のような新しい言語も。IBMやリグゼッティなどのプラットフォームを介して、仮想量子マシンを探索したり、遊んだりすることができます。

ミニQRNGは、民間と公共の両方のセクターのさまざまなタイプのための量子コンピューティングベースの技術の開発に焦点を当ててOpenQbit.com会社によって作成されます。

**なぜミニQRNGが他のQRNGと違うのかというと、単純にシステムがモジュール化されていて、かなり低コストで自宅で簡単に組み立てられるようになっているからです。**

- (1) <https://blogs.iadb.org/conocimiento-abierto/es/como-funciona-la-computacion-cuantica/>

## 12. ソフトウェアのライセンスと使用。

アンドロイド

<https://source.android.com/setup/start/licenses>

タームックス

<https://github.com/termux/termux-app/blob/master/LICENSE.md>

節

<https://raw.githubusercontent.com/nodejs/node/master/LICENSE>

パイソン

<https://www.python.org/download/releases/2.7/license/>

オープンSSH

<https://www.openssh.com/features.html>

パテ SSH

<https://www.chiark.greenend.org.uk/~sgtatham/putty/licence.html>

MIT App Inventor 2 コンパニオンとApp Inventor Blockly

<https://appinventor.mit.edu/about/termservice>

外部拡張子。

JSOONTOOLS

<https://thunkableblocks.blogspot.com/2017/07/jsontools-extension.html>

QRNG Miniシステムのオープンソース版および商用版のライセンスについては、公式ウェブサイト

<http://www.openqbit.com> を参照してください。

Mini QRNG、Mini BlocklyChain、MiniBlockly、BlocklyCode、MiniBlockMiniChain、QBlocklyはOpenQbitで登録されている商標です。

ミニQRNGはパブリックドメインです。

Mini QRNGのすべてのコードとドキュメントは、作者によってパブリックドメインに捧げられています。すべてのコード作者と彼らが働く企業の代表者は、自分たちの貢献をパブリックドメインに捧げる宣誓供述書に署名しており、それらの宣誓供述書の原本はOpenQbit Mexicoの本社の金庫に保管されています。オリジナルのMini QRNG (OpenQbit)拡張機能をソースコードとして、あるいは

コンパイル済みバイナリとして、商用・非商用を問わず、どのような目的でも、どのような手段でも、誰でも自由に公開、使用、配布することができます。

前の段落では、ミニ QRNG のコードとドキュメントの納品物について説明しましたが、ミニ QRNG ライブラリで実際にグループ化され、より大きなアプリケーションで出荷されている部分にも適用されます。コンパイルプロセスの一部として使用されるスクリプト(例えば、autoconfによって生成される「設定」スクリプト)は、他のオープンソースライセンスに含まれている可能性があります。ただし、これらのコンパイル・スクリプトはいずれも最終的な QRNG Mini 納品可能ライブラリには含まれていないため、これらのスクリプトに関連するライセンスは、QRNG Mini ライブラリをコピーして使用する権利を評価する際の要因にはなりません。

ミニQRNGの納品コードはすべて一から書き直しています。他のプロジェクトやオープンインターネットからのコードは一切使用していません。コードの各行を元の作者まで遡ることができ、それらの作者はすべてファイルにパブリックドメインの献辞を持っています。そのため、QRNG Mini のコードベースはクリーンで、他のオープンソースプロジェクトからライセンスされたコードで汚染されていません。

ミニQRNGはオープンソースなので、好きなだけコピーを作って、そのコピーで好きなようにすることができます。でもミニQRNGはオープンソースではありません。Mini QRNG をパブリックドメインに保ち、コードがプロプライエタリなコンテンツやライセンスを受けたコンテンツに汚染されないようにするために、プロジェクトでは未知の人からのパッチを受け付けていません。ミニQRNGのコードは、ミニQRNGで使用するために特別に書かれたものなので、すべてオリジナルです。インターネット上の未知のソースからコピーされたコードはありません。

ミニQRNGはパブリックドメインであり、ライセンスは必要ありません。しかし、一部の団体はミニQRNGの使用権を法的に証明することを求めています。このような状況になった場合には、以下のような状況が考えられます。

- あなたの会社は、著作権侵害のクレームに対して補償を求めています。
- パブリックドメインを認めていない法域でミニQRNGを使用しています。
- あなたは、著作者の著作物をパブリックドメインに捧げる権利を認めていない法域でミニQRNGを使用しています。



- ミニQRNGを使用して配布する権利を持っている証拠として、有形の法律文書を持っていると思います。
- 法務部は免許証を買わなければならないと言っています。

上記のいずれかに該当する場合は、ミニQRNG開発者全員を採用しているOpenQbitがミニQRNGタイトル保証を販売します。タイトルワランティとは、Mini QRNGの主張された作者が真の作者であり、作者がMini QRNGをパブリックドメインに捧げる法的権利を持っていること、OpenQbitがライセンスの主張に対して精力的に自己防衛することを示す法的文書です。ミニQRNGのタイトル保証書の販売収益はすべて、ミニQRNGの継続的な改善とサポートのための資金として使用されます。

### 寄託されたコード

ミニQRNGを完全に無料でロイヤリティフリーに保つために、プロジェクトではパッチを受け付けていません。提案された変更を行い、コンセプトの証明としてパッチを入れていただけると助かります。ただし、パッチを一から書き換えても怒らないでください。このモダリティといくつかの類似のサポートの購入なしでそれを使用する非商用またはオープンソースのライセンスの種類は、企業の規模に関係なく、個人または企業の使用は、次の法的な前提条件によって支配されます。

保証の免責事項。許諾者は、適用法によって要求されるか、または書面で合意されない限り、『作品』（および各投稿者はその投稿を提供する）を「現状のまま」提供し、明示または黙示を問わず、**いかなる種類の保証または条件**（タイトル、非侵害、市販性、特定目的への適合性の保証または条件を含むがこれに限定されない）もなく、「現状のまま」で提供しています。あなたは、作品の正しい使用または再配布を決定し、本ライセンスの下での許可の行使に関連するリスクを引き受けるために単独で責任を負うものとします。

本ソフトウェアの使用により発生した金銭的・その他の損失は、当該当事者が負担するものとします。すべての法的紛争は、当事者がメキシコシティ、国メキシコの管轄下でのみ裁判所に提出します。

商業的なサポート、使用およびライセンスのためには、OpenQbitまたはその企業と利害関係のある当事者との間で契約または契約を締結する必要があります。

配信マーケティングの規約は予告なく変更される場合がありますので、サポート条項やライセンス条項の非商用・商用への変更は公式サイト [www.openqbit.com](http://www.openqbit.com) でご確認ください。

任意の法的性質の任意の個人、ユーザー、民間または公共団体、または世界の任意の部分から、単にソフトウェアを使用している人は、この文書に確立された条項と事前の通知なしに [www.openqbit.com](http://www.openqbit.com) のポータルでいつでも変更することができ、非商業的または商業的な使用でOpenQbitの裁量で適用することができるものを無条件で受け入れます。

Mini QRNGに関するご質問や情報は、App Inventorコミュニティ、または各種Blocklyシステムのコミュニティに直接お問い合わせください。AppBuilder、Trunkableなど、および/または質問の需要のための電子メール [opensource@openqbit.com](mailto:opensource@openqbit.com) には、回答されるまでに3〜5営業日かかることがあります。

商用利用で対応。

[support@openqbit.com](mailto:support@openqbit.com)

業務用販売。

[sales@openqbit.com](mailto:sales@openqbit.com)

法的情報とライセンスに関する質問や懸念事項

[legal@openqbit.com](mailto:legal@openqbit.com)

