



Installation, configuration et administration.

# Manuel d'utilisation

version 1.0 Beta

Juin 2020.

MiniQRNG est une marque déposée d'OpenQbit Inc, sous licence d'utilisation libre et commerciale. Conditions d'utilisation à l'adresse suivante : [www.OpenQbit.com](http://www.OpenQbit.com)

## Contenu

1. Introduction. ....	3
2. Qu'est-ce que la programmation en bloc ? .....	4
3. Qu'est-ce que Termux ? .....	4
4. Qu'est-ce que le Mini QRNG ? .....	4
5. Configuration du stockage dans Termux. ....	9
6. Installation d'un serveur SSH (Secure Shell). ....	10
7. Configuration du serveur SSH sur le téléphone mobile (smartphone). ....	11
8. Ambientes Blockly (App Inventor, AppyBuilder y Thunkable). ....	18
9. Définition et utilisation des blocs dans le Mini QRNG. ....	19
10. Création du dispositif "Hardware" d'un QRNG. ....	31
11. Annexe "Informatique quantique à OpenQbit" .....	37
12. Licences et utilisation des logiciels. ....	42

## 1. Introduction.

La cryptographie actuelle est basée sur des séquences de nombres aléatoires. Les générateurs de nombres pseudo-aléatoires actuellement utilisés semblent fournir des séquences de bits aléatoires, mais en réalité, ces séquences de bits présentent certains schémas, de sorte qu'il existe un risque de piratage ou de toute autre manipulation pour faire circuler l'information sur les réseaux publics et piratés.

L'intégration de sources d'entropie physique dans des générateurs de nombres aléatoires est la méthode la plus courante pour surmonter cette menace pour la sécurité. Cependant, la physique classique étant causale, l'imprévisibilité d'une séquence de bits générée avec la physique classique ne peut être prouvée.

La physique quantique, en revanche, est essentiellement aléatoire. Les nombres générés par un générateur quantique de nombres aléatoires (QRNG) ne peuvent pas être prédits : le QRNG est manifestement imprévisible. Ainsi, si un générateur de nombres aléatoires quantiques est utilisé dans un système de sécurité, même un superordinateur rapide avec une opération arithmétique rapide ne peut pas prévoir les séquences de bits aléatoires utilisées par ce système de sécurité.

La physique quantique utilise des méthodes basées sur un concept fondamental appelé entropie.

### LES TYPES D'ENTROPIE

Il existe deux types généraux de sources d'entropie qui peuvent être mesurées pour générer de véritables nombres aléatoires. Le premier type comprend un processus physique qui est difficile ou impossible à mesurer ou trop exigeant en calcul pour être prédit, ou les deux. C'est une source de caoticentropie. Un exemple commun connu de la plupart des gens est celui d'un appareil de loterie. Un ensemble de billes numérotées séquentiellement est placé dans une chambre et constamment mélangé en faisant tourner la chambre ou en soufflant de l'air à travers la chambre. Plusieurs des boules peuvent tomber de la chambre et les numéros marqués sur les boules représentent le tirage de la loterie. Le tirage au sort est aléatoire en raison du grand nombre d'interactions entre les balles et la caméra, ce qui entraîne une augmentation rapide du nombre de mouvements possibles de chaque balle. Non seulement la complexité de ces interactions est extrêmement élevée, mais il n'existe apparemment aucun moyen d'observer ou de mesurer avec précision toutes les variables internes des billes, de la caméra et du flux d'air. Un deuxième type de source d'entropie, très différent, est la mécanique quantique. De nombreuses particules ou ondes microscopiques, telles que les photons, les électrons et les protons, ont des propriétés mécaniques quantiques qui incluent la rotation, la polarisation, la position et le moment. Étant donné la configuration adéquate pour produire ces particules, les valeurs spécifiques de leur rotation

ou de leur polarisation, par exemple, sont non seulement inconnues et théoriquement imprévisibles, mais sont physiquement déterminées jusqu'à ce qu'une mesure soit effectuée.

## 2. Qu'est-ce que la programmation en bloc ?

**Blockly** est un **langage de programmation visuel** composé d'un simple ensemble de commandes que nous pouvons combiner comme s'il s'agissait des pièces d'un puzzle. C'est un outil très utile pour ceux qui veulent **apprendre à programmer** de manière intuitive et simple ou pour ceux qui savent déjà programmer et qui veulent voir le potentiel de ce type de programmation.

Blockly est une forme de programmation où il n'est pas nécessaire d'avoir des connaissances dans un quelconque langage informatique, c'est parce qu'il s'agit simplement de joindre des blocs graphiques comme si nous jouions au lego ou à un puzzle, il suffit d'avoir un peu de logique et c'est tout !

Tout le monde peut créer des programmes pour les téléphones mobiles (smartphones) sans se frotter aux langages de programmation difficiles à comprendre, il suffit d'assembler des blocs de manière graphique, de façon simple, facile et rapide à créer.

## 3. Qu'est-ce que Termux ?

Termux est un émulateur de terminal Android et une application en environnement Linux qui fonctionne directement sans avoir besoin de routage ou de configuration. Un système de base minimal est automatiquement installé.

Nous utiliserons Termux pour sa stabilité et sa facilité d'installation et de gestion, cependant, vous pouvez utiliser un environnement installé de Ubuntu Linux pour Android.

Dans cet environnement Linux, vous aurez le "noyau" des processus de communication du MiniQRNG.

## 4. Qu'est-ce que le Mini QRNG ?

Mini QRNG est un logiciel et du matériel qui comprend trois solutions technologiques pour créer des QRNG (Quantum Random Number Generators). Classé comme suit :

- a.- QRNG API. - Générateur de nombres aléatoires quantiques obtenu à partir de serveurs externes.
- b.- MiniQRNG / Logiciel. - Générateur de nombres quantiques aléatoires obtenu en utilisant les propriétés physiques (quantiques) de l'appareil photo du téléphone portable.

c.- MiniQRNG / Matériel. - Générateur de nombres quantiques obtenu à l'aide d'un matériel basé sur les propriétés de physique quantique d'un laser. Nous vous expliquerons plus tard comment le construire chez vous à faible coût.

#### 1. Installation et configuration du terminal Termux.

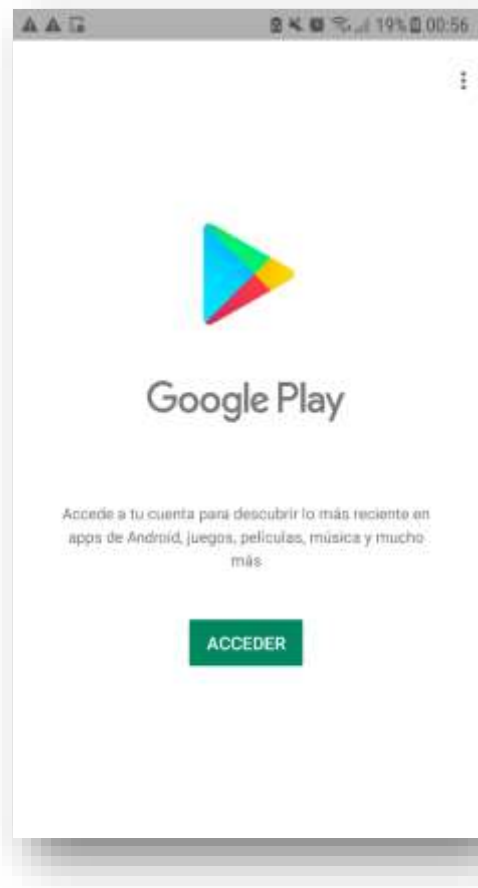
Tout d'abord, nous avons besoin d'un environnement Linux. Comme tout système Android est basé sur Linux pour la sécurité et la flexibilité des outils, nous utiliserons le terminal "Termux" qui contient cet environnement où nous installerons le ou les outils qui nous aideront à créer des QRNG.

Termux est un émulateur de Linux dans lequel nous allons installer les paquets nécessaires pour créer des nombres quantiques.

L'un des principaux avantages de l'utilisation de Termux est que vous pouvez installer des programmes sans avoir à "faire tourner" le téléphone portable (Smartphone), ce qui garantit qu'aucune garantie du fabricant n'est perdue à cause de cette installation.

Installation de Termux.

Depuis votre mobile, rendez-vous sur l'application Google Play ([play.google.com](https://play.google.com)).



Recherchez par l'application "Termux", sélectionnez-la et lancez le processus d'installation.



## Début de l'application Termux.

Après le démarrage, nous devons exécuter les deux commandes suivantes pour effectuer les mises à jour de l'émulateur du système d'exploitation Linux :

Mise à jour de \$ apt

Mise à niveau appropriée

Confirmez toutes les options O(Oui)...

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:

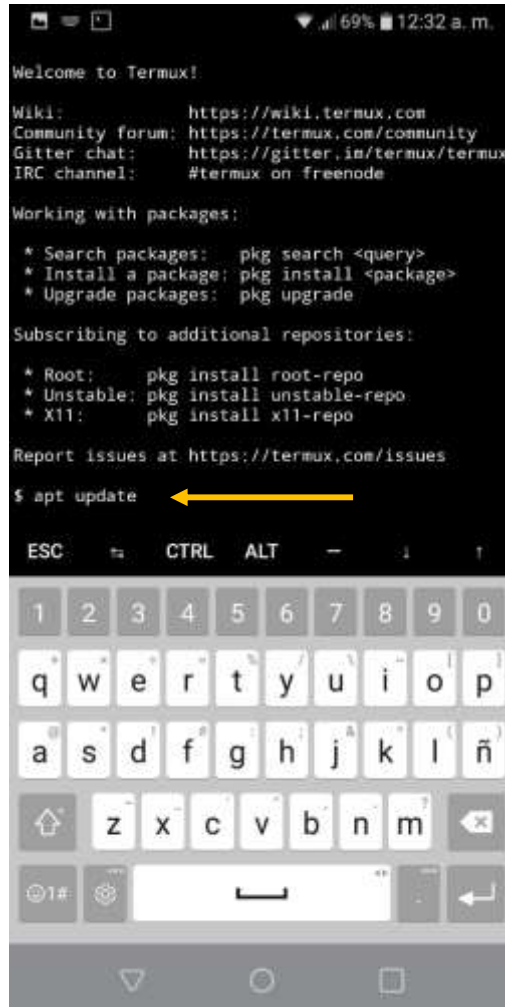
* Search packages:  pkg search <query>
* Install a package: pkg install <package>
* Upgrade packages:  pkg upgrade

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. Configuration du stockage dans Termux.

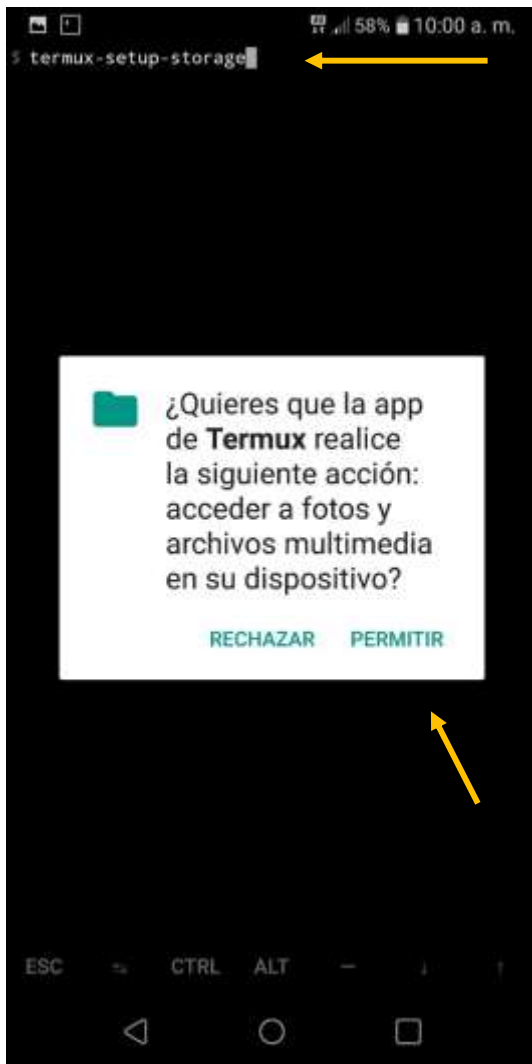
Une fois que vous aurez mis à jour et amélioré le système Termux, nous commencerons à configurer la façon de visualiser le stockage interne du téléphone dans le système Termux. Cela vous aidera à pouvoir échanger des informations entre Termux et nos informations dans le téléphone.

Cela peut être fait simplement et rapidement en exécutant la commande suivante sur un terminal Termux.

**\$ termux-setup-storage**

Lorsque vous exécutez la commande précédente, une fenêtre apparaît pour vous demander de confirmer la création d'un **stockage** virtuel (répertoire) dans Termux. Nous vérifions en donnant l'ordre :

ls



## 6. Installation d'un serveur SSH (Secure Shell).

\$ apt install openssh

\$ apt install sshpass

\$ apt install openssh

\$

apt install sshpass

```

$ apt install openssh
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  krb5 libdns libldb libedit termux-auth
The following NEW packages will be installed:
  krb5 libdns libldb 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 libldb 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
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]
  
```

Nous avons terminé l'installation du réseau de communication pour le serveur SSH localhost sur Smartphone mobile.

## 7. Configuration du serveur SSH sur le téléphone mobile (smartphone).

Nous allons permettre au serveur SSH du téléphone portable de se connecter de notre PC au téléphone portable et de travailler de manière plus rapide et plus confortable. Cela nous permettra également de vérifier que le service du serveur SSH du téléphone portable fonctionne correctement puisque nous l'utiliserons pour la communication avec le Mini QRNG.

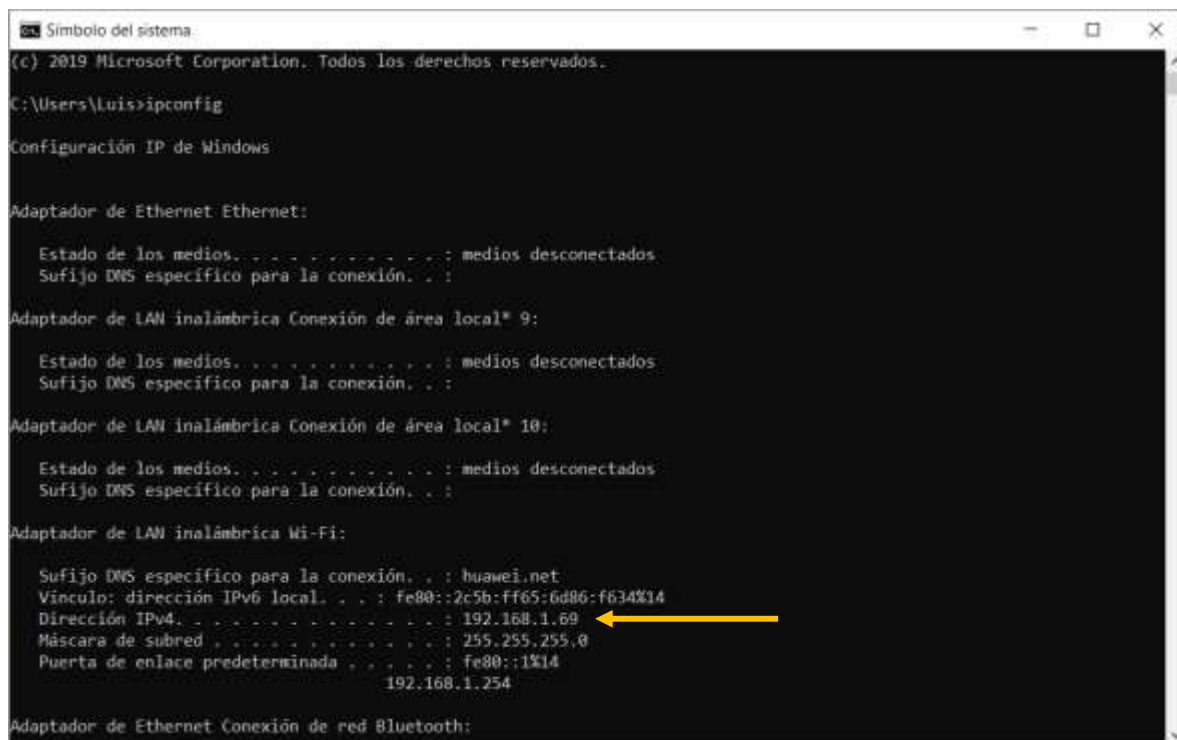
La première chose à faire est de connecter le mobile et le PC au **même réseau WiFi** pour qu'ils puissent se voir. Les IP ou adresses doivent être similaires à 192.168.XXX.XXX ; les valeurs XXX sont des nombres variables qui sont attribués de manière aléatoire dans chaque ordinateur.

Cet exemple a été testé sur un téléphone portable LG Q6 et un PC avec Windows 10 Home.

Vérifiez l'IP ou l'adresse que le PC a connecté au WiFi ; il faut ouvrir un terminal dans Windows.

Dans le panneau inférieur où se trouve la loupe de recherche, écrivez cmd et appuyez sur la touche Entrée. Un terminal s'ouvre et on y écrit la commande :

```
C:\Nom_de_l'utilisateur> ipconfig
```



```
(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:
```

Il nous montrera l'adresse IP attribuée au PC dans ce cas est 192.168.1.69 mais il est fort probable qu'elle soit différente dans chaque cas.

REMARQUE : l'adresse où il est indiqué "adresse IPv4" doit être prise, à ne pas confondre avec la passerelle.

Maintenant, dans le cas du téléphone portable dans le terminal Termux, nous devons taper la commande suivante pour connaître le nom de notre utilisateur que nous utiliserons pour nous connecter au serveur SSH qui a notre téléphone, nous exécutons la commande suivante :

**\$ whoami**

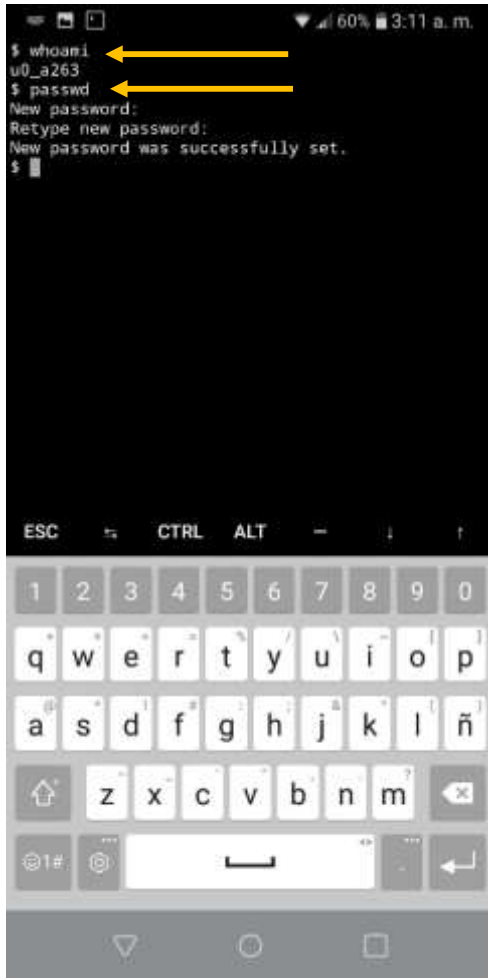
Plus tard, nous devons donner un mot de passe à cet utilisateur et nous devons donc exécuter la commande suivante :

**\$ passwd**

Il nous demandera de taper un mot de passe et d'appuyer sur Entrée, il nous demande à nouveau le mot de passe que nous confirmons et appuyons sur Entrée, s'il a été correctement **"Nouveau mot de passe a été défini avec succès"** en cas de marquage d'une erreur est possible que le mot de passe n'ait pas été tapé correctement. Répétez la procédure.

Et puis pour savoir quelle IP nous avons dans Termux nous tapons la commande suivante, l'IP est après le mot **"inet"** :

**\$ ifconfig -a**



Il est

maintenant temps de lancer le service de serveur SSH sur votre téléphone pour que vous puissiez recevoir des sessions depuis votre PC. Nous exécutons la commande suivante dans le terminal Termux, cette commande ne donne aucun résultat.

\$ sshd



Nous allons maintenant devoir installer sur le PC un programme qui communiquera avec le serveur SSH du téléphone à partir du PC.

Il faut aller sur <https://www.putty.org>

Sélectionnez où se trouve le lien "Vous pouvez télécharger PuTTY ici".



The screenshot shows the PuTTY website homepage. At the top is a navigation bar with a search icon and the text "Search for a new PuTTY version". Below this is a large heading "Download PuTTY". To the left of the heading is a small image of the PuTTY application window. To the right of the heading is a paragraph of text: "PuTTY is an SSH and telnet client, developed originally by Simon Tatham for the... with source code and is developed and supported by a group of volunteers." Below this paragraph is a line of text: "You can download PuTTY [here](#)." A yellow arrow points from the word "here" to the right. Below the main content area is a horizontal line, and below that is a paragraph: "Below suggestions are independent of the authors of PuTTY. They are *not* to be seen a".



#### Bitvise SSH Client

Bitvise SSH Client is an SSH and SFTP client for Windows. It is developed and supported prof supports all features supported by PuTTY, as well as the following:

- graphical SFTP file transfer;
- single-click Remote Desktop tunneling;
- auto-reconnecting capability;
- dynamic port forwarding through an integrated proxy;
- an FTP-to-SFTP protocol bridge.

Bitvise SSH Client is **free to use**. You can [download it here](#).

Choisissez la version 32 bits, peu importe que votre système soit en 64 bits, il fonctionnera parfaitement.

## Download PuTTY: latest release

[Home](#) | [FAQ](#) | [Feedback](#) | [Licence](#) | [Updates](#) | [Mirror](#)  
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')

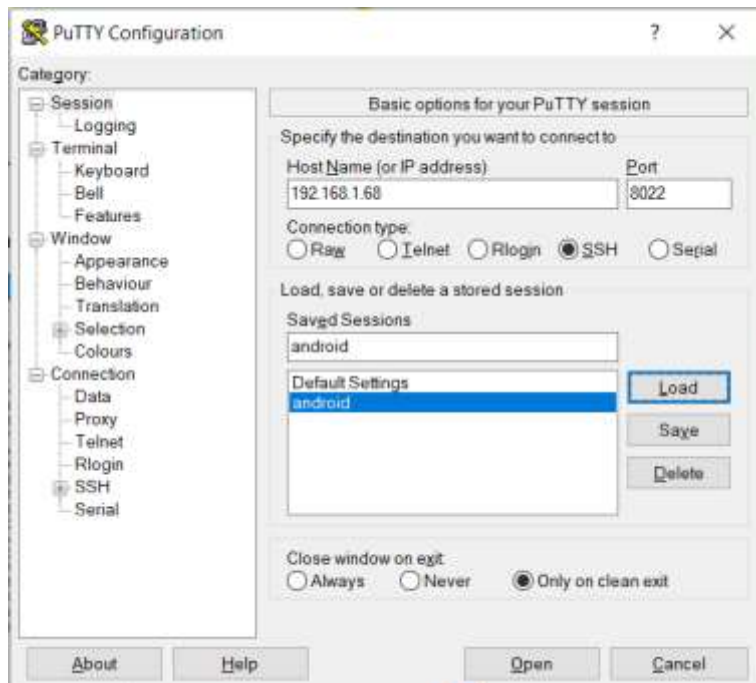
32-bit: [putty-0.73-installer.msi](#) (or by FTP) (signature)  
64-bit: [putty-64bit-0.73-installer.msi](#) (or by FTP) (signature)

#### Unix source archive

.tar.gz: [putty-0.73.tar.gz](#) (or by FTP) (signature)

Une fois qu'il a été téléchargé sur votre PC, lancez-le et installez-le avec les options par défaut. Ensuite, lancez l'application PuTTY.

Dans cette session, nous allons entrer les données de notre serveur Openssh que nous avons installé dans le téléphone portable.



Entrez l'adresse IP du téléphone portable.

Nom d'hôte ou adresse IP :

**192.168.1.68** (exemple IP)

Port :

**8022** (port par défaut du serveur SSH mobile).

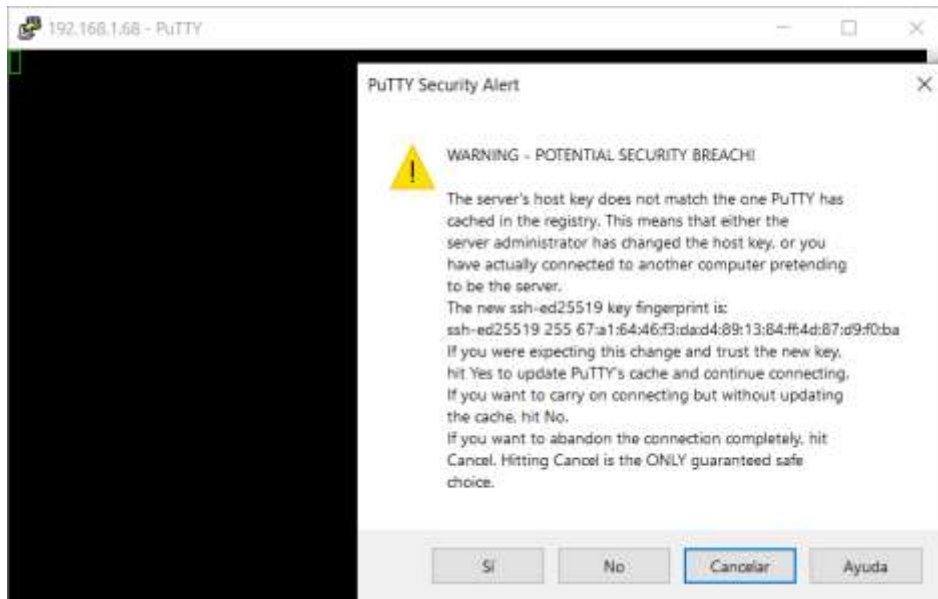
Nous pouvons donner un nom à la session dans "Sessions

enregistrées" et cliquer sur le bouton "Enregistrer".



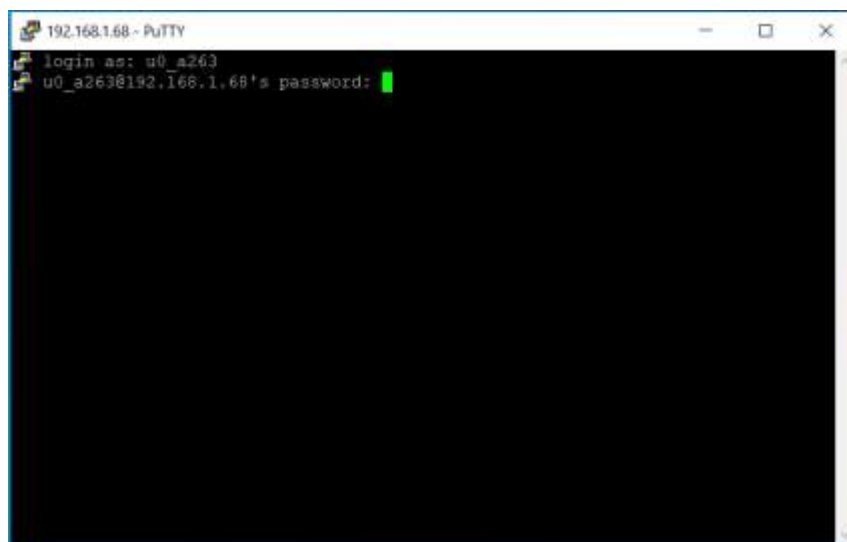
Plus tard, dans la partie inférieure, nous appuyons sur le bouton "Ouvrir" pour ouvrir une connexion au serveur.

Sur le PC, lorsque vous vous connectez pour la première fois, il vous sera **demandé de confirmer** la clé de cryptage des informations en cliquant sur le bouton "Oui".



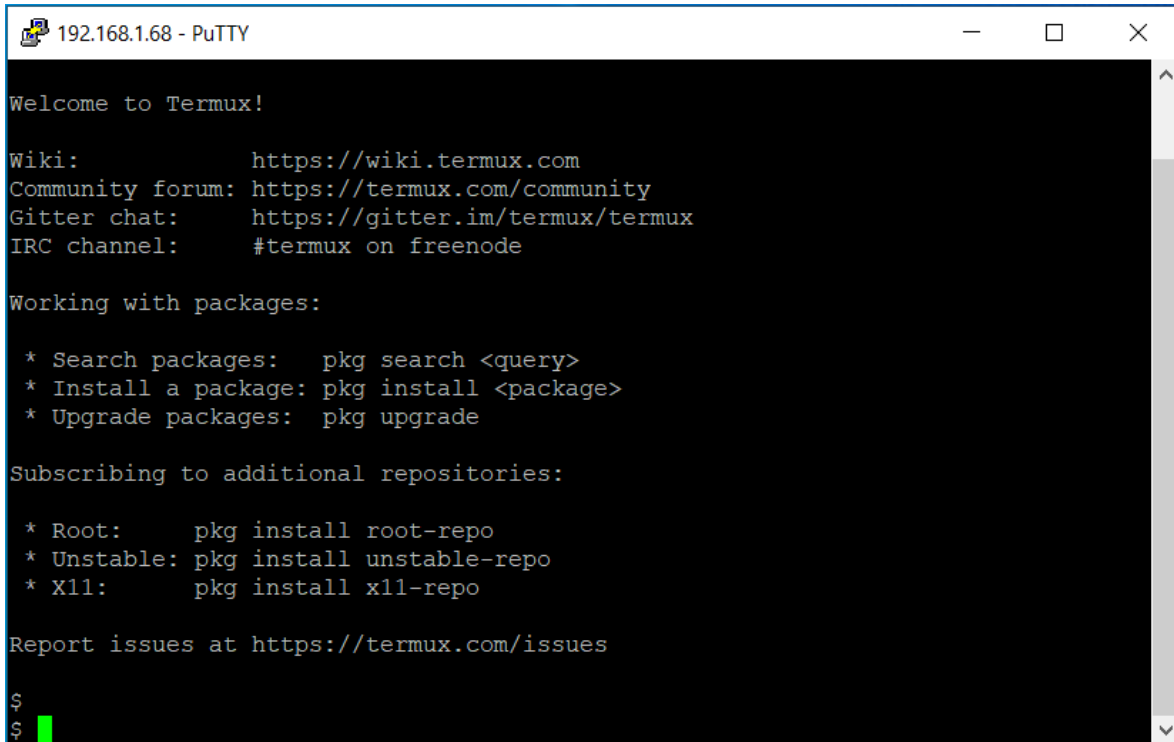
Plus tard, on nous demandera le nom de l'utilisateur avec lequel nous allons nous connecter. Nous utiliserons les informations que nous avons obtenues précédemment (utilisateur et mot de passe).

Dans le **Login comme** : nous devons entrer notre utilisateur et donner Entrée, puis nous demanderons le mot de passe à nouveau donner le bouton Entrée.





Si les données étaient correctes, nous serions dans une session SSH (Secure Shell) effectuée depuis le PC (Client) sur le téléphone (Serveur SSH).



```
192.168.1.68 - PuTTY
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

$
$
```

**NOTE IMPORTANTE :** N'oubliez pas que l'IP (adresse) du PC et l'IP (adresse) du téléphone mobile connecté au même WiFi changeront probablement chaque fois que nous nous déconnecterons et nous reconnecterons. Nous devons donc vérifier les adresses de chaque appareil, ce qui garantira le succès de la connexion entre les appareils via le serveur SSH du téléphone et le PC (Client).

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

App Inventor est un environnement de développement logiciel créé par Google Labs pour créer des applications pour le système d'exploitation Android. L'utilisateur peut, visuellement et à partir d'un ensemble d'outils de base, relier une série de blocs pour créer l'application. Le système est gratuit et peut être facilement téléchargé sur le web. Les applications créées avec App Inventor sont très faciles à créer car aucune connaissance d'un langage de programmation n'est requise.

Tous les environnements actuels qui utilisent la technologie de Blockly tels que AppyBuilder et Thunkable entre autres ont leur version gratuite, leur mode d'utilisation peut se faire via Internet sur leurs différents sites ou il peut également être installé à la maison.

Les blocs qui composent l'architecture Mini BloclyChain ont été testés dans App inventor et AppyBuilder mais, en raison de l'optimisation de leur code, ils devraient fonctionner sur les autres plateformes.

Versions en ligne :

App Inventor.

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

AppyBuilder.

<http://appybuilder.com/>

Possédable.

<https://thunkable.com/>

Version à installer sur votre ordinateur (PC) :

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

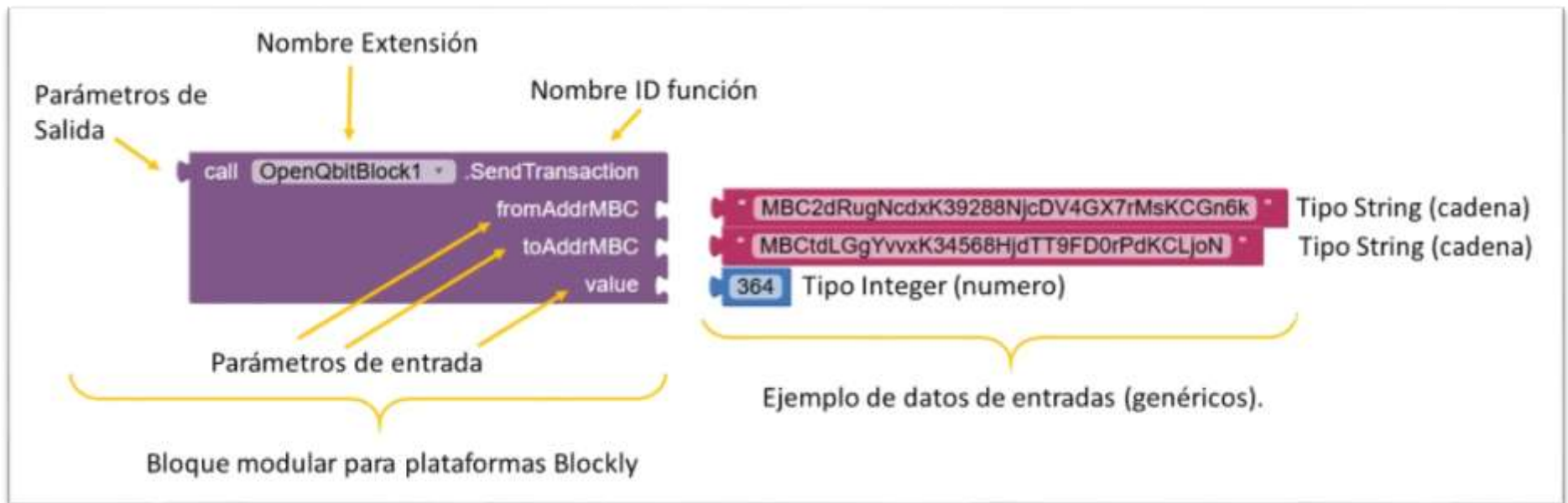
Environnement pour les développeurs de blocs Blockly.

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

## 9. Définition et utilisation des blocs dans le Mini QRNG.

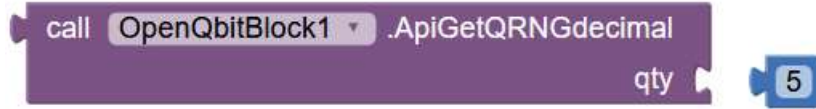
Nous commencerons par expliquer la répartition des données que tous les blocs auront, leur syntaxe d'utilisation et leur configuration.

Dans l'exemple suivant, nous pouvons voir un bloc modulaire et ses paramètres d'entrée et de sortie, ainsi que les types de données d'entrée, ces données peuvent être de type String (chaîne de caractères) ou Integer (entier ou décimal). Nous montrons comment il est utilisé et le configurons pour son bon fonctionnement.



Chaque bloc de module aura sa description et sera nommé au cas où il aurait une ou plusieurs dépendances obligatoires ou facultatives d'autres blocs utilisés comme paramètres d'entrée, le processus d'intégration sera annoncé. Commençons par les blocs de l'extension OpenQbitQRNGwithSSH.

Bloc pour la génération de nombres quantiques aléatoires décimaux - (**ApiGetQRNGdécimal**)



Paramètres d'entrée : **qté** <Intégrer>

Paramètres de sortie : donne la quantité "qté" de nombres décimaux quantiques aléatoires entrés dans les numéros d'entrée sont dans la gamme de 0 et 1 dans le format JSON.

Exemple :

qté = 5 ; sortie : {"résultat" : [0,5843012986202495, 0,7746497687824652, 0,05951126805960929, 0,1986079055812694, 0,03689783439899279]}

Description : API du générateur quantique de nombres aléatoires (QRNG)

Bloc pour la génération de nombres quantiques aléatoires décimaux - (**ApiGetQRNGdécimal**)



Paramètres d'entrée : **qty** <Intérieur>, min <Intérieur>, max <max>

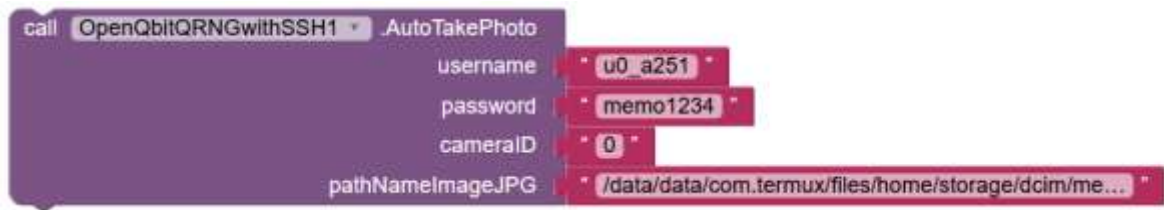
Paramètres de sortie : donne la quantité "qté" d'entiers quantiques aléatoires entrés dans l'entrée ; les nombres sont compris entre min et max dans le format JSON.

Exemple :

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

Description : API du générateur quantique de nombres aléatoires (QRNG)

Bloquer pour prendre une photo automatiquement - (**AutoTakePhone**)



Paramètres de saisie : **nom d'utilisateur** <String>, **mot de passe** <String>, **cameraID** <String>, **pathNameImageJPG** <String>.

Dépendance obligatoire : pour utiliser ce bloc, vous devez remplir deux dépendances logicielles : installer dans le terminal Termux le module Termux-API. Ce module contient le processus permettant de prendre automatiquement des photos et de télécharger le serveur SSH qui a été installé précédemment.

Paramètres de sortie : Il délivre une photo (image) au format JPG dans le chemin spécifié. Sur le chemin, vous devez

Description : crée une photo JPG automatiquement sans intervention de l'utilisateur.

Pour installer le Termux-API, la commande suivante doit être exécutée dans le Terminal Termux :

\$ 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) ...
$

```

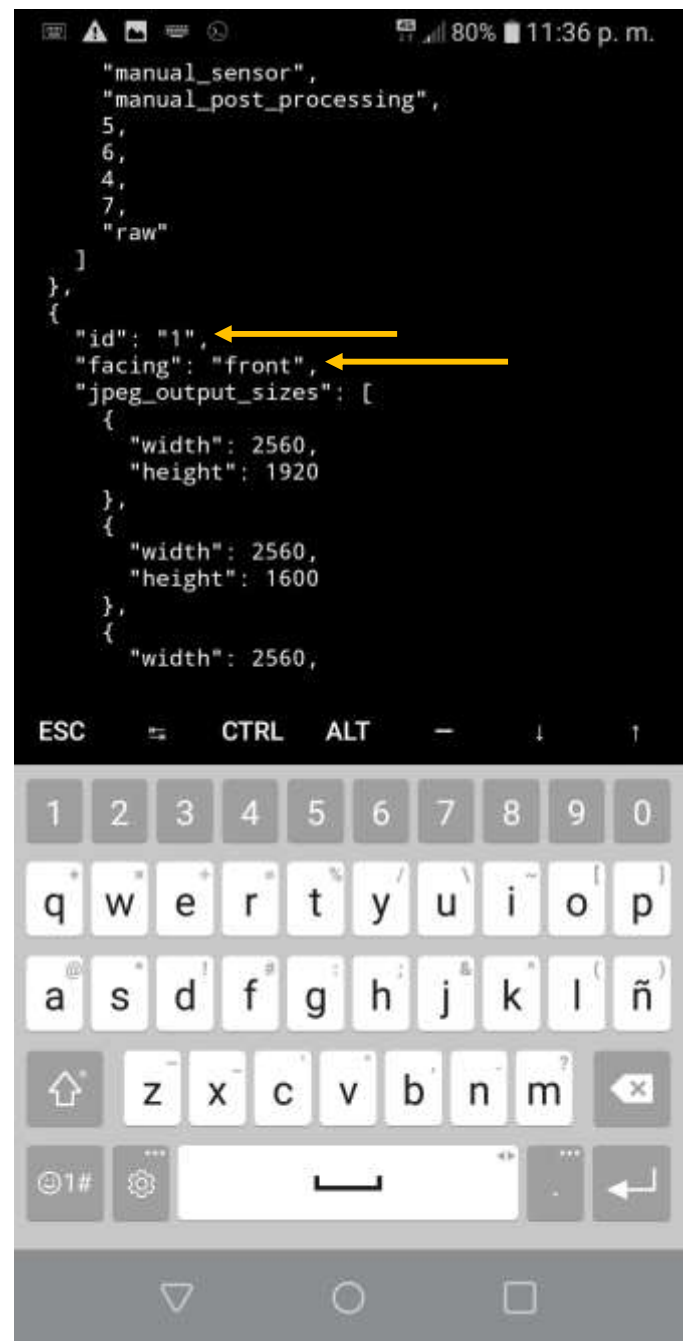
Pour connaître le nombre, le nombre et la position des ID (photo lens identifiers) dont dispose votre appareil mobile (Smartphone), exécutez la commande suivante dans le terminal Termux.

\$ 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,
```

Dans notre exemple, le smarphone LG Q6 que nous avons utilisé a deux identifiants "0" à l'arrière et "1" à l'avant.

Maintenant, testons l'API pour prendre une photo en utilisant l'ID "0" de la lentille arrière et donnons-lui un nom aléatoire dans notre cas test.jpg

N'oubliez pas que l'API ne fournit que des photos au format JPG :

**\$ termux-camera-photo -c 0 test.jpg**

La commande précédente doit avoir créé un fichier avec le nom test.jpg automatiquement, si c'est le cas nous pouvons utiliser le bloc (AutoTakePhoto), n'oubliez pas de démarrer notre serveur SSH local avec la commande : **\$ sshd**

**NOTE :** Dans la variable pathNameImageJPG, il faut considérer que le chemin à l'intérieur du terminal Termux pour accéder au stockage du Smartphone doit être utilisé :

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

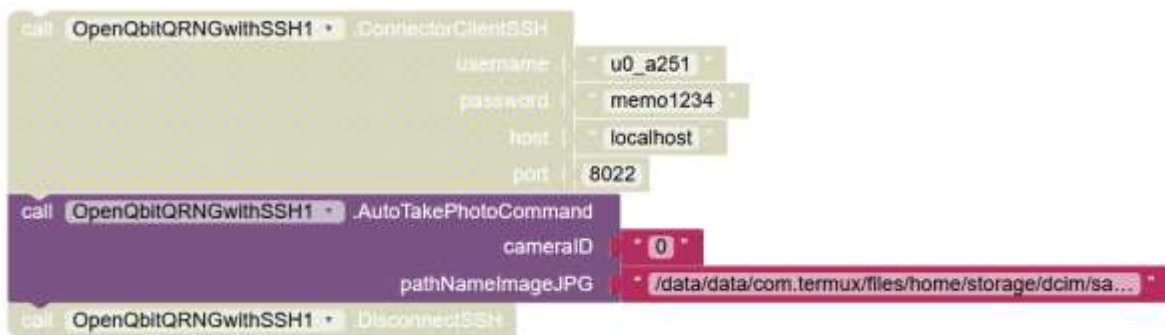
L'itinéraire précédent sur l'androïde serait le même que :

/mnt/sdcard/dcim/exemple.jpg

Cependant, il faut se rappeler que dans le terminal Termux, la route valide pour visualiser la mise sous tension du téléphone portable doit toujours être considérée comme la route par défaut :

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

Blocage pour prendre une photo automatiquement COMMANDE SEULEMENT - (AutoTakePhoneCommand)



Paramètres d'entrée : **cameraID <String>**, **pathNameImageJPG <String>**.

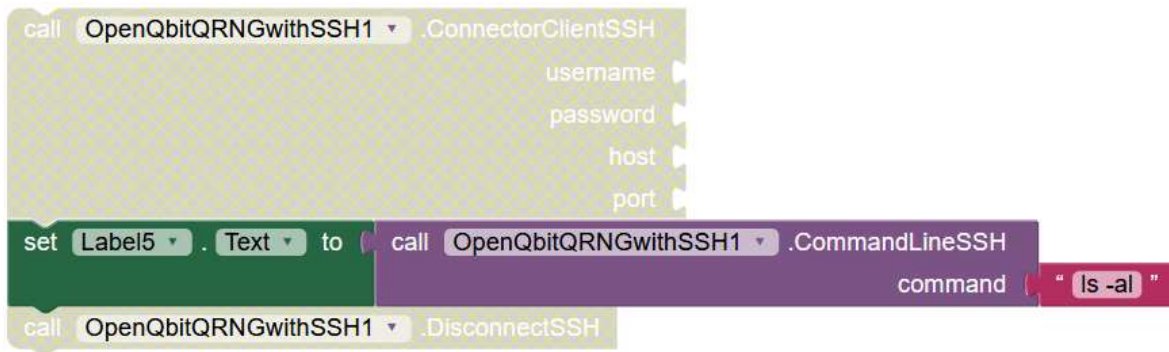
Dépendance obligatoire : Bloquer (**ConnectorClientSSH**), Bloquer (**DisconnectSSH**).

Paramètres de sortie : Il délivre une photo (image) au format JPG dans le chemin spécifié.

Description : crée une photo JPG automatiquement sans intervention de l'utilisateur. Cependant, la différence par rapport au bloc (AutoTakePhoto), ce bloc ne contient que la commande pour créer la photo et vous avez besoin du bloc d'abord pour vous connecter au serveur SSH (**ConnectClientSSH**) et ensuite utiliser le bloc (**DisconnectSSH**).



Bloc pour exécuter une commande dans le terminal Termux - (CommandLineSSH)



Paramètres d'entrée : **commande** <String>

Dépendance obligatoire : Bloquer (ConnectorClientSSH), Bloquer (DisconnectSSH).

Paramètres de sortie : Exécute la commande saisie dans le terminal Termux.

Description : une commande saisie est exécutée et le bloc est nécessaire d'abord pour se connecter au serveur SSH (ConnectorClientSSH) et ensuite pour utiliser le bloc (DisconnectSSH).

Bloc pour se connecter à un serveur SSH distant ou local - (ConnectorClientSSH).

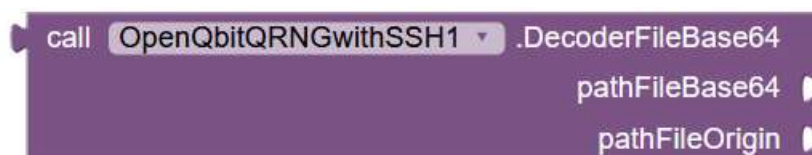


Paramètres d'entrée : **nom d'utilisateur** <string>, **mot de passe** <string>, **hôte** <string>, **port**<integer>

Paramètres de sortie : Si la connexion avec le serveur ssh du terminal Termux est réussie, il nous donne un message ; **"Connect SSH"**, si elle n'est pas réussie, il nous donne un message **NULL**.

Description : Bloc de communication pour connecter le serveur SSH choisi au terminal Termux, via le protocole de communication SSH (Secure Shell).

Bloc pour décoder un fichier avec l'algorithme Base64 (DecoderFileBase64).



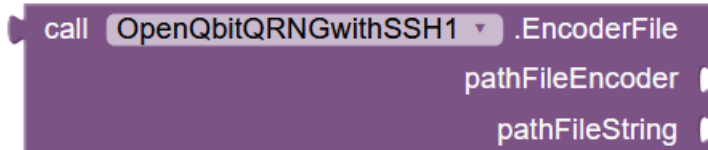


Paramètres d'entrée : **pathFileBase64** <String>, pathFileOrigin <String>

Paramètres de sortie : Fichier source qui a été saisi dans le bloc (**EncoderFileBase64**)

Description : un fichier Base64 est converti en fichier original qui a été inséré dans le bloc (**EncoderFileBase64**).

Block convertit un fichier au format Base64 - (**EncoderFile**)

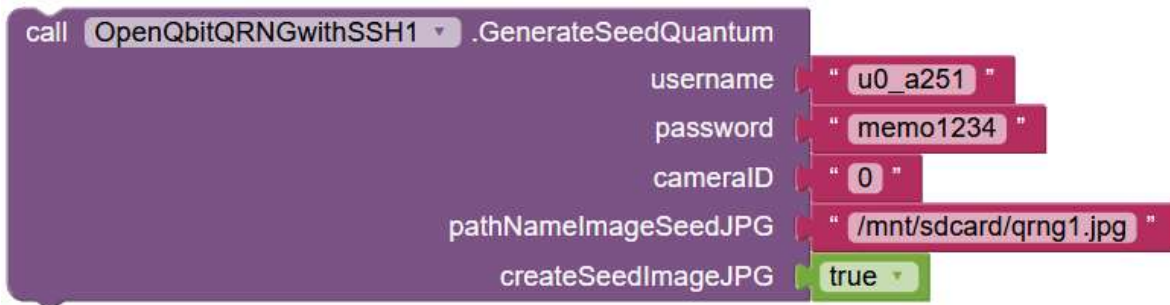


Paramètres d'entrée : **pathFileOrigin** <String> , pathFileBase64 <String>

Paramètres de sortie : Fichier encodé en Base64.

Description : Convertit un fichier source de n'importe quel format en un fichier Base64. Les noms de fichiers peuvent être arbitraires et choisis par l'utilisateur.

Bloc pour générer le QRNG (Quantum Random Number Generator) - (**GenerateSeedQuantum**)



Paramètres de saisie : **nom d'utilisateur** <String>, **mot de passe** <String>, **cameraID** <String>, **pathNameImageJPG** <String>. **createSeedImageJPG** <Boolean>

Si la valeur booléenne est "True" lorsque le bloc est exécuté à chaque fois qu'une nouvelle image JPG de départ sera créée avec le nom du chemin saisi. Si la valeur de Boolean est "False", nous désactivons l'option de prendre une image JPG (photo) et nous pouvons indiquer manuellement où se trouve une image dans notre sélection, qui peut être de n'importe quel format.

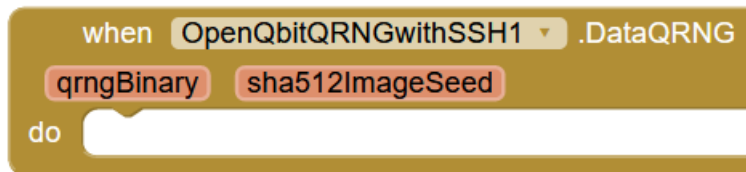
NOTE : Les meilleurs résultats pour la génération d'un QRNG sont basés sur une image au format "RAW". Exemple de format DNG.

Dépendance obligatoire : L'API Termux susmentionnée doit être installée dans le bloc (AutoTakePhoto).

Paramètres de sortie : L'événement est exécuté (**DataQRNG**) et nous donne deux valeurs :

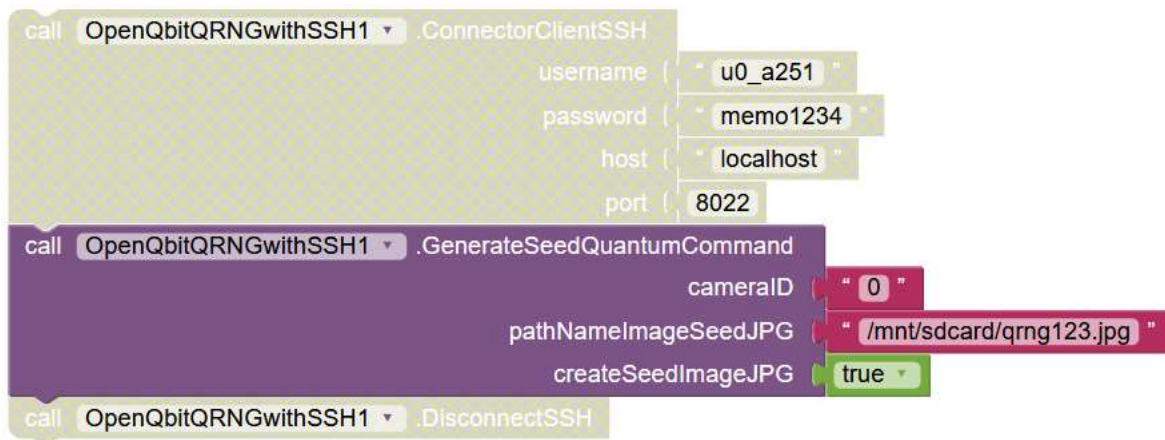
qrngBinary - Une chaîne de nombres binaires aléatoires

Sha512ImageSeed - Sha512 de l'image JPG d'où proviennent les nombres aléatoires.



Description : il génère des nombres aléatoires quantiques (QRNG) par le biais du capteur optique de l'appareil photo du téléphone portable. L'algorithme est basé sur la capture de photos aléatoires et l'algorithme lui est appliqué pour fournir une chaîne de nombres binaires.

Bloc pour générer le QRNG (Quantum Random Number Generator) - (GenerateSeedQuantumCommand).



Paramètres d'entrée : **cameraID** <String>, **pathNameImageJPG** <String>. **createSeedImageJPG** <Boolean>

Si la valeur booléenne est "True" lorsque le bloc est exécuté à chaque fois qu'une nouvelle image JPG de départ sera créée avec le nom du chemin saisi. Si la valeur de Boolean est "False", nous désactivons l'option de prendre une image JPG (photo) et nous pouvons indiquer manuellement où se trouve une image dans notre sélection, qui peut être de n'importe quel format.

Dépendance obligatoire : Bloquer (**ConnectorClientSSH**), Bloquer (**DisconnectSSH**).

Paramètres de sortie : L'événement est exécuté (**DataQRNG**) et nous donne deux valeurs :

qrngBinary - Une chaîne de nombres binaires aléatoires

Sha512ImageSeed - Sha512 de l'image JPG d'où proviennent les nombres aléatoires.



Description : il génère des nombres aléatoires quantiques (QRNG) par le biais du capteur optique de l'appareil photo du téléphone portable. L'algorithme est basé sur la capture de photos aléatoires et l'algorithme lui est appliqué pour fournir une chaîne de nombres binaires.

Cependant, la différence par rapport au bloc (**GenerateSeedQuantum**), ce bloc ne contient que la commande d'exécution de l'algorithme de réaction QRNG et vous avez besoin du bloc d'abord pour vous connecter au serveur SSH (**ConnectClientSSH**) et ensuite utiliser le bloc (**DisconnectSSH**).

Bloc pour obtenir l'entropie de Shannon dans une image (photo) - (**GetShannonEntropyFile**).



Paramètres de saisie : nom d'utilisateur <String> , mot de passe <String>, pathFileImage <String>

Dépendance obligatoire : Le module Shannon\_entropy doit être installé dans le terminal Termux.

Paramètres de sortie : Fournit l'entropie d'une image

Exemple :

Résultat : 8.94596789873

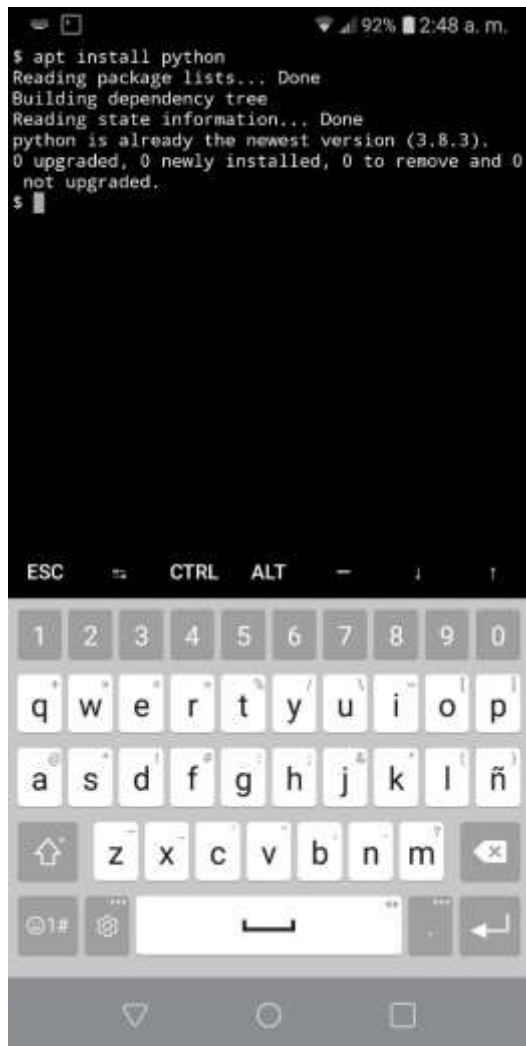
Description : Il nous donne l'entropie d'une image. L'entropie est le paramètre fondamental pour la génération de nombres aléatoires de bonne qualité, plus l'entropie est élevée, meilleurs sont les résultats.

Pour installer le module Shannon entropy, nous devons d'abord installer le paquet Python, puis installer le module Pillow et Shannon\_entropy avec les commandes suivantes dans le terminal Termux.

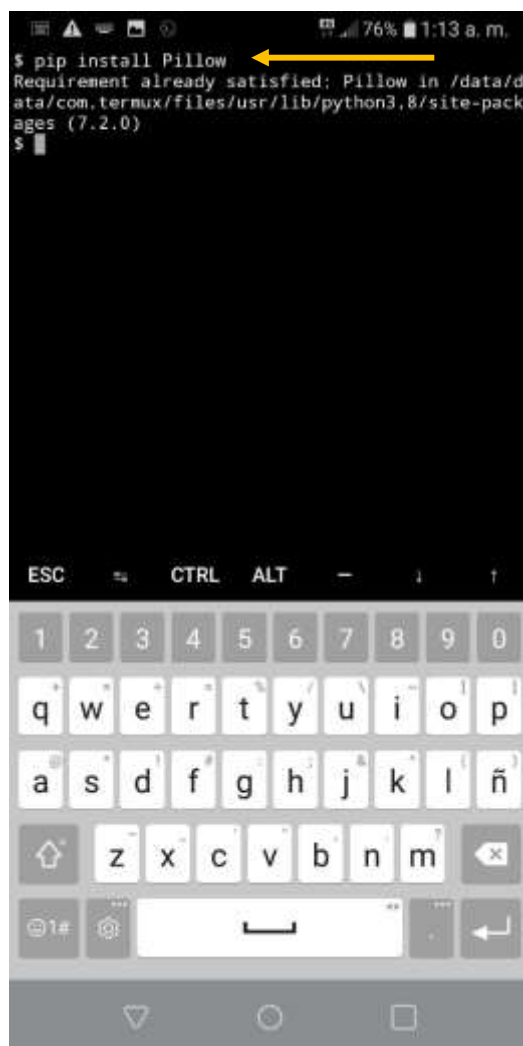
\$ apt installent Python

\$ pip install Pillow

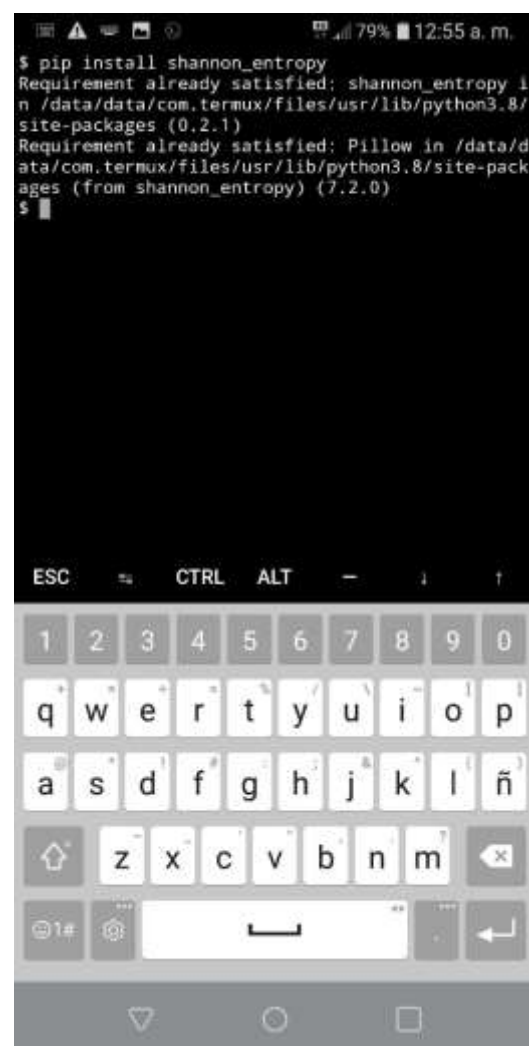
\$ pip install Shannon\_entropy



```
$ 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/data/com.termux/files/usr/lib/python3.8/site-packages (7.2.0)
$
```



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

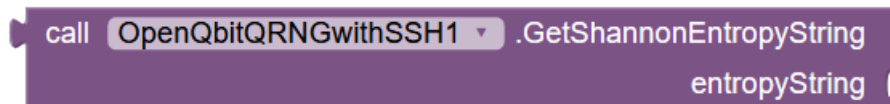
Ensuite, nous devons créer dans le répertoire "Home" de Termux le fichier Python suivant appelé "entropy.py" avec le code suivant à l'intérieur.

```
De l'image de l'importation PIL
Mathématiques d'importation
De Shannon_entropy import *
Importation sys
Img=Image.open(sys.argv[1])
print(Shannon_entropy(img))
```

Nous sauvegardons le fichier et avons notre environnement à utiliser avec le bloc (GetShannonEntropyFile).

**Astuce :** En fait, avec cette installation Python, vous pouvez créer vos propres programmes dans ce langage et les exécuter à travers le bloc (ConnectorClientSSH).

Bloc pour obtenir l'entropie de Shannon à partir d'une corde - (GetShannonEntropyString).



Paramètres d'entrée : entropyString>String>

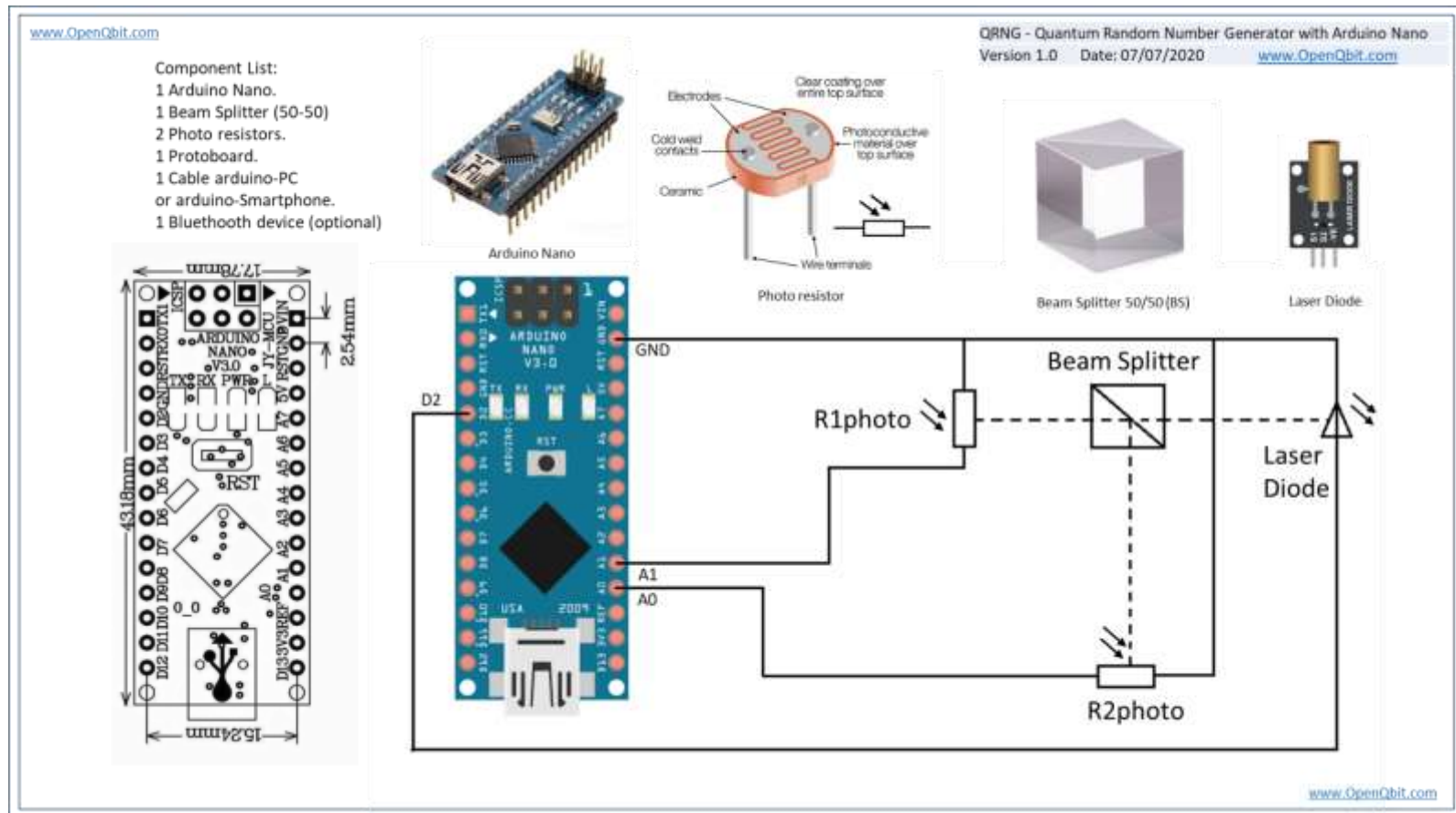
Paramètres de sortie : Fournit l'entropie d'une chaîne de caractères.

Exemple :

Production : 5.76002345671

Description : Il nous donne l'entropie d'une chaîne de caractères. L'entropie est le paramètre fondamental pour la génération de nombres aléatoires de bonne qualité, plus l'entropie est élevée, meilleurs sont les résultats.





[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)



Compilation du programme QRNGv10.ino et téléchargement vers le nano...



```
QRNGv10 Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda

QRNGv10

int triggerQ = 2; // This pin will pulse our quantum circuit.
int Qa0Pin = A0; // This pin measures the horizontal polarised photons
int Qa1Pin = A1; // This pin measures the vertically polarised photons
float Qa0 = 0;
float Qa1 = 0;

void setup() {
  // Just setting up triggerPin and serial connection
  pinMode(triggerQ, OUTPUT); // sets the digital pin 2 as output
  Serial.begin(5000);
}

int Random() {
  // Pulse the laser
  digitalWrite(triggerQ, HIGH);
  delay(300);
  digitalWrite(triggerQ, LOW);
  delay(300);
  // Read the photoreistors
  Qa0 = analogRead(Qa0Pin);
  Qa1 = analogRead(Qa1Pin);
  // Determine random bit
  if(Qa0 > Qa1) { // More photons in the Qa0 mode, return 0
    return 0;
  } else if(Qa0 < Qa1) { // More photons in the Qa1 mode, return 1
    return 1;
  } else {
    // The same number of photons are in both modes!
    // This is actually not an uncommon occurrence, but our
    // purpose is still easily run the function sequentially until
    // a random bit can be generated.
    return Random();
  }
}

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

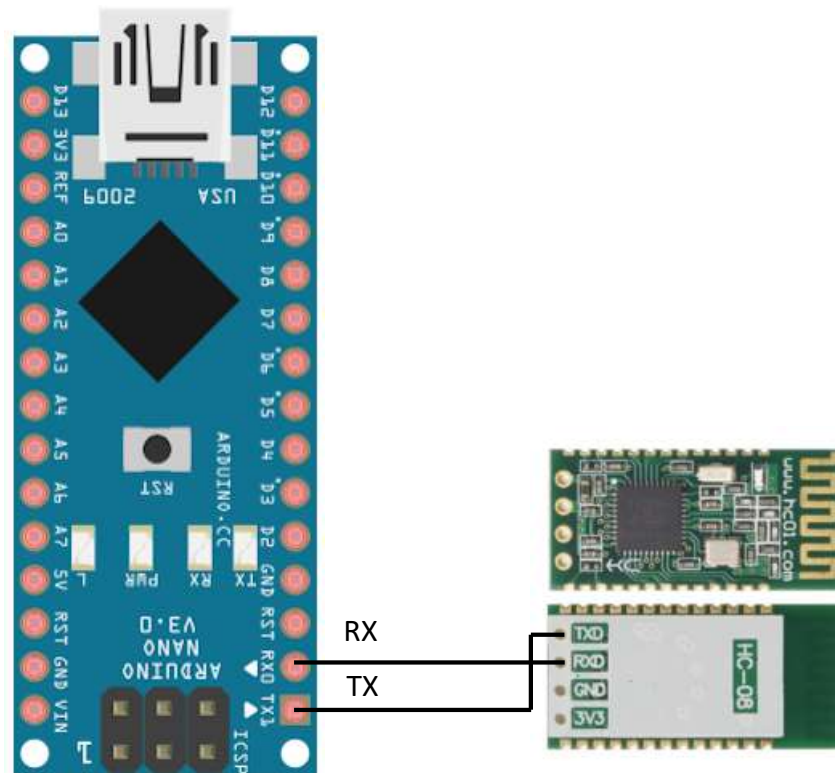
Compilado

El Sketch usa 2352 bytes (7%) del espacio de almacenamiento de programa. El máximo es 32256 bytes.  
Las variables Globales usan 198 bytes (1%) de la memoria dinámica, dejando 1840 bytes para las variables locales. El máximo es 2048 bytes.

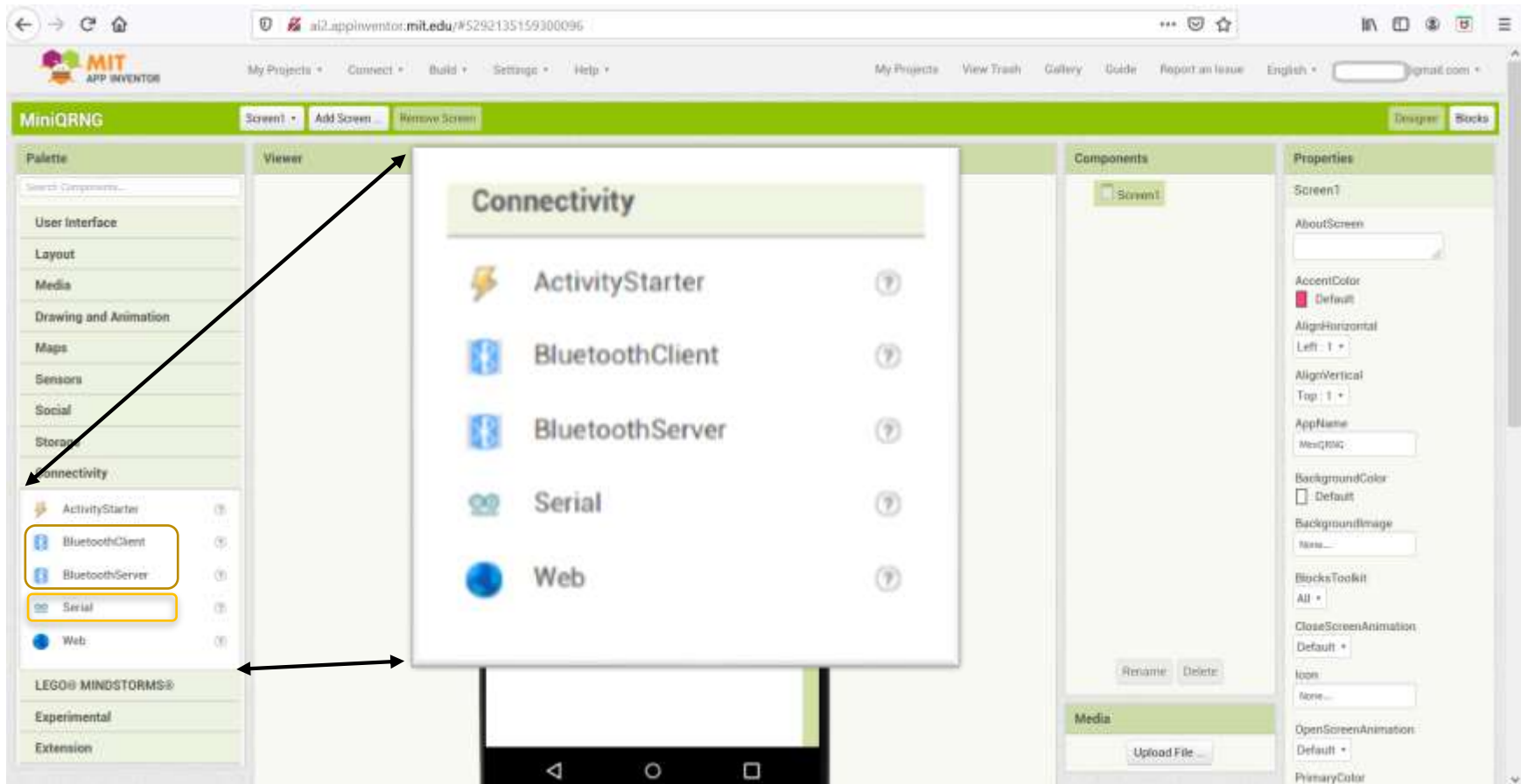
40 ArduinoGetters Ltd en CDM

Il y a deux façons de communiquer avec le nano difficile, l'une via le port série et l'autre via une connexion Bluetooth.

Pour la connexion bluetooth est très simple, il suffit d'acheter le module HC-08 ou un module similaire et de le connecter comme suit :



Les composants série ou Bluetooth suivants peuvent être utilisés pour connecter App Inventor à Arduino :



Maintenant compilé et chargé le programme QRNGv10.ino ne manque que de communiquer avec la nano ardue pour sauvegarder les données (nombres aléatoires quantiques) celles-ci seront en format binaire, cependant, les données obtenues peuvent être facilement passées à un autre format tel que l'hexadécimal ou le décimal en fonction de l'exigence finale.

Enfin, pour voir un exemple du fonctionnement de la connexion série ou Bluetooth, voici quelques liens de référence.

Rappelez-vous que tout est par la programmation Blockly à être testé avec App Inventor qui a déjà des blocs pour la communication avec arduino série ou autre système blockly peut être à des tarves bluetooth similaires en ligne.

[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. Annexe "Informatique quantique à OpenQbit".

### Comment fonctionne l'informatique quantique ? <sup>(2)</sup>

La transformation numérique entraîne des changements dans le monde plus rapides que jamais. Croyez-vous que l'ère numérique est sur le point de prendre fin ? La **culture numérique** a déjà été identifiée comme un domaine dans lequel il est urgent de disposer de connaissances ouvertes et de possibilités accessibles d'apprentissage de la technologie afin de combler les lacunes en matière de développement social et économique. L'apprentissage des concepts clés de l'ère numérique deviendra encore plus crucial avec l'arrivée imminente d'une autre nouvelle vague technologique capable de transformer les modèles existants avec une rapidité et une puissance étonnantes : les **technologies quantiques**.

Dans cet article, nous comparons les concepts de base de l'informatique traditionnelle et de l'informatique quantique ; et nous commençons également à explorer leur application dans d'autres domaines connexes.

Que sont les technologies quantiques ?

Tout au long de l'histoire, les êtres humains ont développé la technologie à mesure qu'ils ont compris comment la nature fonctionne grâce à la science. Entre 1900 et 1930, l'étude de certains phénomènes physiques encore mal compris a donné naissance à une nouvelle théorie physique, la **mécanique quantique**. Cette théorie décrit et explique le fonctionnement du monde microscopique, l'habitat naturel des molécules, des atomes ou des électrons. Grâce à cette théorie, non seulement il a été possible d'expliquer ces phénomènes, mais il a également été possible de comprendre que la réalité subatomique fonctionne d'une manière totalement contre-intuitive, presque magique, et que dans le monde microscopique se produisent des événements qui ne se produisent pas dans le monde macroscopique.


Ces **propriétés quantiques** comprennent la superposition quantique, l'intrication quantique et la téléportation quantique.

- **La superposition quantique** décrit comment une particule peut se trouver dans différents états en même temps.
- **L'intrication quantique** décrit comment deux particules aussi éloignées l'une de l'autre que souhaité peuvent être corrélées de telle manière que, lors de leur interaction avec l'une d'entre elles, l'autre en soit consciente.
- **La téléportation** quantique utilise l'enchevêtrement quantique pour envoyer des informations d'un endroit à un autre dans l'espace sans avoir à le traverser.

Les technologies quantiques sont basées sur ces propriétés quantiques de nature subatomique.

Dans ce cas, la compréhension du monde microscopique par la mécanique quantique nous permet aujourd'hui d'inventer et de concevoir des technologies capables d'améliorer la vie des gens. Il existe de nombreuses et très différentes technologies qui utilisent des phénomènes quantiques et certaines d'entre elles, comme les lasers ou l'imagerie par résonance magnétique (IRM), sont présentes depuis plus d'un demi-siècle. Cependant, nous assistons actuellement à une révolution technologique dans des domaines tels que l'informatique quantique, l'information quantique, la simulation quantique, l'optique quantique, la métrologie quantique, les horloges ou les capteurs quantiques.

Qu'est-ce que l'informatique quantique ? Tout d'abord, il faut comprendre l'informatique



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

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

classique.

Pour comprendre le fonctionnement des ordinateurs quantiques, il convient d'abord d'expliquer comment fonctionnent les ordinateurs que nous utilisons tous les jours, que nous appellerons dans ce document ordinateurs numériques ou classiques. Ceux-ci, comme le reste des appareils électroniques tels que les tablettes ou les téléphones portables, utilisent les bits comme unités fondamentales de la mémoire. Cela signifie que les programmes et les applications sont encodés en bits, c'est-à-dire en langage binaire de zéros et de uns. Chaque fois que nous interagissons avec l'un de ces dispositifs, par exemple en appuyant sur une touche du clavier, des chaînes de zéros et de uns sont créées, détruites et/ou modifiées à l'intérieur de l'ordinateur.

La question intéressante est de savoir ce que sont ces zéros et ces uns physiquement à l'intérieur de l'ordinateur. Les états zéro et un correspondent à un courant électrique qui circule, ou non, à travers des parties microscopiques appelées transistors, qui agissent comme des interrupteurs. Lorsqu'aucun courant ne circule, le transistor est "off" et correspond au bit 0, et lorsqu'il circule, il est "on" et correspond au bit 1.

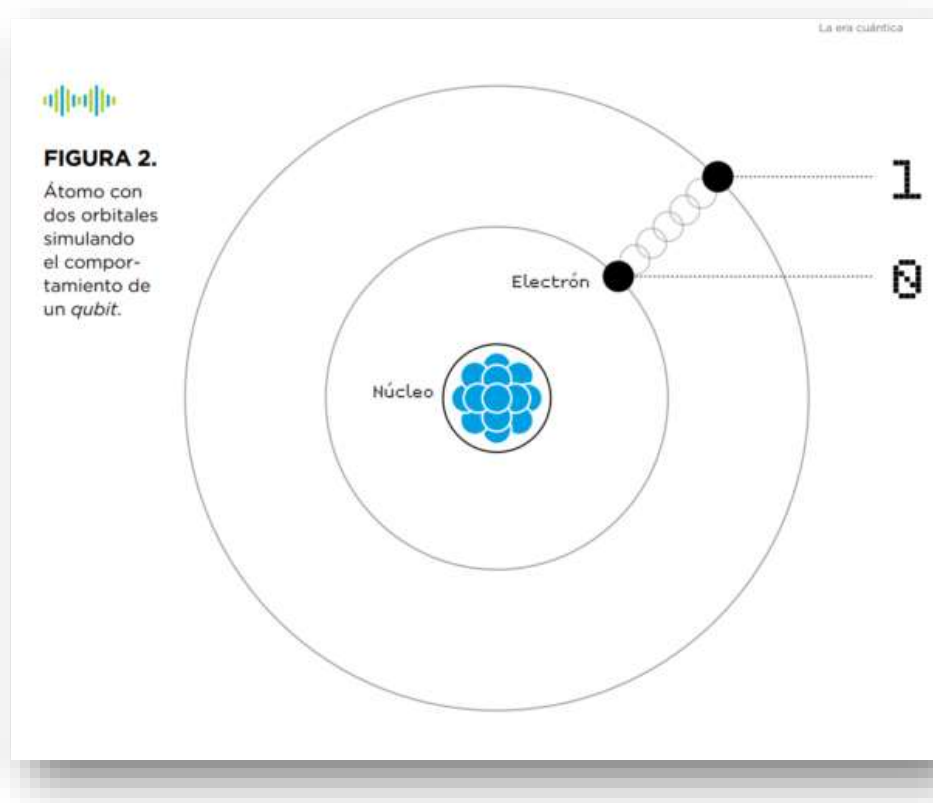
Plus simplement, c'est comme si les bits 0 et 1 correspondaient à des trous, de sorte qu'un trou vide est un bit 0 et un trou occupé par un électron est un bit 1. C'est pourquoi ces

appareils sont appelés électroniques. À titre d'exemple, la figure 1 montre l'écriture binaire de certains caractères. Maintenant que nous avons une idée du fonctionnement des ordinateurs d'aujourd'hui, essayons de comprendre comment fonctionnent les quanta.

## Des bits aux qubits

L'unité fondamentale de l'information dans l'informatique quantique est le bit ou qubit quantique. Les qubits sont, par définition, des systèmes quantiques à deux niveaux - nous en verrons des exemples ici - qui, comme les bits, peuvent être au niveau bas, qui correspond à un état de faible excitation ou d'énergie défini comme 0, ou au niveau haut, qui correspond à un état d'excitation plus élevé ou défini comme 1. Cependant, et c'est là que réside la différence fondamentale avec le calcul classique, les qubits peuvent également se trouver dans n'importe lequel des états intermédiaires infinis entre 0 et 1, comme un état qui est la moitié de 0 et la moitié de 1, ou les trois quarts de 0 et un quart de 1.

Les



algorithmes quantiques, un calcul exponentiellement plus puissant et plus efficace

L'objectif des ordinateurs quantiques est de tirer parti de ces propriétés quantiques des *qubits*, en tant que systèmes quantiques, afin d'exécuter des algorithmes quantiques qui utilisent le chevauchement et l'entrelacement pour fournir une puissance de traitement beaucoup plus importante que les classiques. Il est important de souligner que le véritable changement de paradigme ne consiste pas à faire la même chose que les ordinateurs

numériques ou classiques - les actuels - mais plus rapidement, comme on peut le lire dans de nombreux articles, mais que les algorithmes quantiques permettent d'effectuer certaines opérations d'une manière totalement différente qui, dans de nombreux cas, s'avère plus efficace - c'est-à-dire en beaucoup moins de temps ou en utilisant beaucoup moins de ressources informatiques -.

Voyons un exemple concret de ce que cela implique. Imaginons que nous soyons à Bogota et que nous voulions connaître le meilleur itinéraire pour se rendre à Lima parmi un million de possibilités pour y arriver ( $N=1\,000\,000$ ). Afin d'utiliser les ordinateurs pour trouver la voie optimale, nous devons numériser  $1\,000\,000$  d'options, ce qui implique de les traduire en langage binaire pour l'ordinateur classique et en *qubits* pour l'ordinateur quantique. Alors qu'un ordinateur classique devrait analyser un par un tous les chemins jusqu'à trouver celui qu'il souhaite, un ordinateur quantique tire parti du processus connu sous le nom de parallélisme quantique qui lui permet de considérer tous les chemins en même temps. Cela implique que, alors que l'ordinateur classique a besoin de l'ordre de  $N/2$  étapes ou itérations, c'est-à-dire  $500\,000$  tentatives, l'ordinateur quantique trouvera le chemin optimal après seulement des opérations  $\sqrt{N}$  sur le registre, soit  $1\,000$  tentatives.

Dans le cas précédent, l'avantage est quadratique, mais dans d'autres cas, il est même exponentiel, ce qui signifie qu'avec  $n$  *qubits*, nous pouvons obtenir une capacité de calcul équivalente à  $2^n$  bits. Pour illustrer ce point, il est courant de compter qu'avec environ  $270$  qubits, nous pourrions avoir plus d'états de base dans un ordinateur quantique - plus de chaînes de caractères différentes et simultanées - que le nombre d'atomes dans l'univers, qui est estimé à environ  $280$ . Un autre exemple est qu'on estime qu'avec un ordinateur quantique de  $2000$  à  $2500$  *qubits*, nous pourrions casser pratiquement toute la cryptographie utilisée aujourd'hui (la cryptographie dite à clé publique).

Pourquoi est-il important de connaître la technologie quantique ?

Nous sommes dans un moment de transformation numérique où les différentes technologies émergentes telles que la chaîne de blocs, l'intelligence artificielle, les drones, l'Internet des objets, la réalité virtuelle, les imprimantes 5G, 3D, les robots ou les véhicules autonomes sont de plus en plus présentes dans de multiples domaines et secteurs. Ces technologies, appelées à améliorer la qualité de vie de l'être humain en accélérant le développement et en générant un impact social, progressent aujourd'hui de manière parallèle. Il est rare de voir des entreprises développer des produits qui exploitent des combinaisons de deux ou plusieurs de ces technologies, comme la chaîne de blocs et l'IdO ou les drones et l'intelligence artificielle. Bien qu'elles soient destinées à converger, générant ainsi un impact exponentiellement plus important, le stade initial de développement dans lequel elles se trouvent et la rareté des développeurs et des personnes ayant un profil technique font que la convergence est encore une tâche en suspens.



En raison de leur potentiel perturbateur, les technologies quantiques devraient non seulement converger avec toutes ces nouvelles technologies, mais aussi avoir une influence transversale sur pratiquement toutes d'entre elles. L'informatique quantique menacera l'authentification, l'échange et le stockage sécurisé des données, ce qui aura un impact majeur sur les technologies où la cryptographie joue un rôle plus important, comme la cybersécurité ou la chaîne de blocage, et un impact négatif mineur, mais qui doit également être pris en compte dans les technologies telles que la 5G, l'IdO ou les drones.

Vous voulez pratiquer l'informatique quantique ?

Des dizaines de simulateurs d'ordinateurs quantiques sont déjà disponibles sur le net avec différents langages de programmation déjà utilisés tels que C, C++, Java, Matlab, Maxima, Python ou Octave. De plus, de nouveaux langages comme Q#, lancé par Microsoft. Vous pouvez explorer et jouer avec une machine quantique virtuelle grâce à des plateformes telles qu'IBM et Rigetti.

Mini QRNG est créé par la société OpenQbit.com qui se concentre sur le développement de technologies basées sur l'informatique quantique pour différents types de secteurs, tant privés que publics.

**Pourquoi le Mini QRNG est différent des autres QRNG, simplement parce que le système a été créé pour être modulaire et peut être facilement assemblé à la maison à un coût assez faible.**

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

## 12. Licences et utilisation des logiciels.

Android

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

Termux

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

Nœud

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

Python

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

OpenSSH

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

Mastic SSH

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

MIT App Inventor 2 Companion et App Inventor Blockly

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

Extensions externes :

JSOINTOOLS

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

L'octroi de licences pour les versions open source et commerciales du système QRNG Mini est disponible sur le site officiel <http://www.openqbit.com>

Mini QRNG, Mini BlocklyChain, MiniBlockly, BlocklyCode, MiniBlockMiniChain, QBlockly son marcas registradas por OpenQbit.

Le mini QRNG est du domaine public.

Tous les codes et la documentation du Mini QRNG ont été mis dans le domaine public par les auteurs. Tous les auteurs de codes et les représentants des entreprises pour lesquelles ils travaillent ont signé des déclarations sous serment consacrant leurs contributions au domaine public et les originaux de ces déclarations sont conservés dans un coffre-fort au siège d'OpenQbit Mexico. Tout le monde est libre de publier, d'utiliser ou de distribuer les extensions originales Mini QRNG (OpenQbit), soit sous forme de code source, soit sous forme

de binaires compilés, à n'importe quelle fin, commerciale ou non, et par n'importe quel moyen.

Le paragraphe précédent s'applique au code et à la documentation délivrés dans le Mini QRNG aux parties de la bibliothèque du Mini QRNG qui regroupent et expédient effectivement une application plus importante. Certains scripts utilisés dans le cadre du processus de compilation (par exemple, les scripts de "configuration" générés par autoconf) peuvent être inclus dans d'autres licences open source. Toutefois, aucun de ces scripts de compilation ne fait partie de la mini-bibliothèque QRNG finale, de sorte que les licences associées à ces scripts ne devraient pas être un facteur d'évaluation de vos droits de copie et d'utilisation de la mini-bibliothèque QRNG.

Tout le code livrable dans le Mini QRNG a été écrit à partir de zéro. Aucun code n'a été repris d'autres projets ou de l'internet ouvert. Chaque ligne de code peut être retracée jusqu'à son auteur d'origine, et tous ces auteurs ont des dédicaces du domaine public dans leur dossier. Par conséquent, la base de mini-codes du QRNG est propre et n'est pas contaminée par du code provenant d'autres projets à source ouverte, et non par une contribution ouverte

Mini QRNG est un logiciel libre, ce qui signifie que vous pouvez faire autant de copies que vous le souhaitez et en faire ce que vous voulez, sans limitation. Mais Mini QRNG n'est pas un logiciel libre. Afin de maintenir le Mini QRNG dans le domaine public et de garantir que le code n'est pas contaminé par des contenus propriétaires ou sous licence, le projet n'accepte pas les patches de personnes inconnues. Tout le code dans Mini QRNG est original, car il a été écrit spécifiquement pour être utilisé par Mini QRNG. Aucun code n'a été copié à partir de sources inconnues sur Internet.

Le mini QRNG est dans le domaine public et ne nécessite pas de licence. Cependant, certaines organisations veulent une preuve légale de leur droit à utiliser le Mini QRNG. Les circonstances dans lesquelles cela se produit sont notamment les suivantes :

- Votre entreprise veut être indemnisée pour des plaintes pour violation de droits d'auteur.
- Vous utilisez le Mini QRNG dans une juridiction qui ne reconnaît pas le domaine public.
- Vous utilisez le Mini QRNG dans une juridiction qui ne reconnaît pas à un auteur le droit de consacrer son œuvre au domaine public.
- Vous voulez avoir un document juridique tangible comme preuve que vous avez le droit légal d'utiliser et de distribuer le Mini QRNG.
- Votre service juridique vous dit que vous devez acheter une licence.

Si l'une des circonstances ci-dessus s'applique à vous, OpenQbit, la société qui emploie tous les développeurs de Mini QRNG, vous vendra une garantie de titre Mini QRNG. Une garantie de titre est un document juridique qui déclare que les auteurs revendiqués du Mini QRNG

sont les véritables auteurs, et que les auteurs ont le droit légal de dédier le Mini QRNG au domaine public, et qu'OpenQbit se défendra vigoureusement contre les revendications de licence. Tous les produits de la vente des garanties de titre de Mini QRNG sont utilisés pour financer l'amélioration et le soutien continu de Mini QRNG.

#### Code de contribution

Pour que le Mini QRNG reste totalement gratuit et libre de droits, le projet n'accepte pas les patches. Si vous souhaitez faire une suggestion de changement et inclure un patch comme preuve de concept, ce serait bien. Cependant, ne soyez pas offensé si nous réécrivons votre patch à partir de zéro. Le type de licence non commerciale ou open source qui l'utilise dans cette modalité et certaines autres similaires sans achat de support, qu'il s'agisse d'une utilisation individuelle ou collective, quelle que soit la taille de l'entreprise, sera régi par les principes juridiques suivants.

Exclusion de garantie. Sauf si la loi applicable l'exige ou si cela est convenu par écrit, le Concédant fournit l'œuvre (et chaque Contributeur fournit ses Contributions) "TEL QUEL", **SANS GARANTIES OU CONDITIONS DE QUELQUE NATURE QUE CE SOIT**, expresses ou implicites, y compris, sans limitation, toute garantie ou **condition de TITRE**, d'ABSENCE DE CONTREFAÇON, de VALEUR MARCHANDE OU d'ADAPTATION À UN USAGE PARTICULIER. Vous êtes seul responsable de déterminer l'utilisation ou la redistribution correcte de l'œuvre et d'assumer les risques associés à l'exercice des droits que vous confère la présente licence.

Toute perte financière ou autre subie par l'utilisation de ce logiciel sera supportée par la partie concernée. Tous les litiges juridiques que les parties soumettront à la juridiction des tribunaux de la ville de Mexico, au Mexique.

Pour le support commercial, l'utilisation et la licence, un accord ou un contrat doit être établi entre OpenQbit ou sa société et la partie intéressée.

Les conditions de distribution et de commercialisation peuvent être modifiées sans préavis. Veuillez consulter le site officiel [www.openqbit.com](http://www.openqbit.com) pour voir les modifications apportées aux clauses de soutien et de licence non commerciales et commerciales.

Toute personne, utilisateur, entité privée ou publique de toute nature juridique ou de toute partie du monde qui utilise simplement le logiciel accepte sans conditions les clauses établies dans ce document et celles qui peuvent être modifiées à tout moment dans le portail de [www.openqbit.com](http://www.openqbit.com) sans avis préalable et peuvent être appliquées à la discrétion d'OpenQbit dans un usage non commercial ou commercial.

Toutes les questions et informations concernant le Mini QRNG doivent être adressées à la communauté App Inventor ou aux différentes communautés du système Blockly telles qu'elles sont : AppBuilder, Trunkable, etc. et/ou à l'adresse électronique [opensource@openqbit.com](mailto:opensource@openqbit.com) pour la demande de questions peut prendre de 3 à 5 jours ouvrables pour être répondu.

Soutien à l'utilisation commerciale.

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

Vente à des fins commerciales.

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

Informations juridiques et questions ou préoccupations concernant les licences

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

