



Installazione, configurazione e amministrazione.

Manuale d'uso

versione 1.0 Beta

Giugno 2020.

MiniQRNG è un marchio registrato di OpenQbit Inc., sotto licenza d'uso gratuito e commerciale. Termini e condizioni d'uso all'indirizzo: www.OpenQbit.com

Contenuto

1.	Introduzione.....	3
2.	Cos'è la programmazione Blockly?	4
3.	Che cos'è Termux?	4
4.	Cos'è il Mini QRNG?	4
5.	Configurazione della memoria all'interno di Termux.	8
6.	Installazione del server SSH (Secure Shell).	9
7.	Configurazione del server SSH su cellulare (smartphone).	10
8.	Ambientes Blockly (App Inventor, AppyBuilder y Thunkable).	16
9.	Definizione e uso dei blocchi in Mini QRNG.	17
10.	Creazione di un dispositivo "Hardware" di un QRNG.	28
11.	Allegato "OpenQbit Quantum Computing".	34
12.	Licenze e utilizzo del software.	39

1. Introduzione.

La crittografia di oggi si basa su sequenze di numeri casuali. I generatori di numeri pseudo-casuali attualmente in uso sembrano fornire sequenze di bit casuali, ma in realtà queste sequenze di bit hanno determinati schemi, quindi c'è il rischio di essere hackerati o comunque manipolati per far viaggiare le informazioni su reti pubbliche e pirata.

L'integrazione delle fonti di entropia fisica nei generatori di numeri casuali è il metodo più comune per superare questa minaccia alla sicurezza. Tuttavia, la fisica classica è causale, quindi l'imprevedibilità di una sequenza di bit generata con la fisica classica non può essere provata.

La fisica quantistica, invece, è essenzialmente casuale. I numeri generati da un generatore di numeri casuali quantistici (QRNG) non possono essere previsti: QRNG è dimostrabilmente imprevedibile. Quindi, se in un sistema di sicurezza viene usato un generatore di numeri casuali quantistici, anche un supercomputer veloce con operazioni aritmetiche veloci non può prevedere le sequenze di bit casuali usate da questo sistema di sicurezza.

La fisica quantistica utilizza metodi basati su un concetto fondamentale chiamato entropia.

TIPI DI ENTROPIA

Ci sono due tipi generali di sorgenti di entropia che possono essere misurati per generare veri numeri casuali. Il primo tipo comprende un processo fisico difficile o impossibile da misurare o troppo intenso dal punto di vista computazionale per poterlo prevedere, o entrambi. Questa è una fonte di caoticentropia. Un esempio comune noto alla maggior parte delle persone è la macchina della lotteria. Una serie di sfere numerate in sequenza sono posizionate in una camera e costantemente mescolate insieme ruotando la camera o soffiando aria attraverso la camera. Diverse palline sono lasciate cadere fuori dalla camera e i numeri segnati sulle palline rappresentano l'estrazione della lotteria. Il sorteggio è casuale a causa del gran numero di interazioni tra le palline e la telecamera, con il risultato di un rapido aumento del numero di possibili movimenti di ogni palla. Non solo la complessità di queste interazioni è estremamente elevata, ma non esiste un modo apparente per osservare o misurare con precisione tutte le variabili interne delle sfere, della telecamera e del flusso d'aria. Un secondo tipo di fonte di entropia molto diversa è la meccanica quantistica. Molte particelle o onde microscopiche, come fotoni, elettroni e protoni, hanno proprietà meccaniche quantistiche che includono rotazione, polarizzazione, posizione e quantità di moto. Data la corretta configurazione per produrre queste particelle, i valori specifici della loro rotazione o polarizzazione, per esempio, non solo sono sconosciuti e teoricamente imprevedibili, ma sono fisicamente determinati fino a quando non viene effettuata una misurazione.

2. Cos'è la programmazione Blockly?

Blockly è un **linguaggio di programmazione visuale** composto da un semplice insieme di comandi che possiamo combinare come se fossero i pezzi di un puzzle. È uno strumento molto utile per chi vuole **imparare a programmare** in modo intuitivo e semplice o per chi sa già programmare e vuole vedere le potenzialità di questo tipo di programmazione.

Blockly è una forma di programmazione in cui non è necessario alcun background in nessun tipo di linguaggio informatico, questo perché si tratta solo di unire blocchi grafici come se stessi giocando a lego o a un puzzle, basta avere un po' di logica ed è tutto!

Chiunque può creare programmi per cellulari (smartphone) senza pasticciare con quei linguaggi di programmazione difficili da capire, basta mettere insieme i blocchi in modo grafico in modo semplice, facile e veloce da creare.

3. Che cos'è Termux?

Termux è un emulatore di terminale Android e un'applicazione per l'ambiente Linux che funziona direttamente senza bisogno di routing o configurazione. Un sistema di base minimo viene installato automaticamente.

Useremo Termux per la sua stabilità e la sua facile installazione e gestione, tuttavia, è possibile utilizzare un ambiente installato di Ubuntu Linux per Android.

In questo ambiente Linux avrete il "cuore" dei processi di comunicazione del MiniQRNG.

4. Cos'è il Mini QRNG?

Mini QRNG è Software e Hardware che comprende tre soluzioni tecnologiche per creare QRNG (Quantum Random Number Generators). Classificati come segue:

- a.- API QRNG. - Generatore di numeri casuali quantistici ottenuti da server esterni.
- b.- MiniQRNG / Software. - Generatore di numeri quantici casuale ottenuto utilizzando le proprietà fisiche (quantum) della fotocamera del telefono cellulare.
- c.- MiniQRNG / Hardware. - Generatore di numeri quantistici ottenuto utilizzando hardware basato sulle proprietà fisiche quantistiche di un laser. Più tardi vi diremo come costruirlo a casa a basso costo.

1. Installazione e configurazione del terminale Termux.

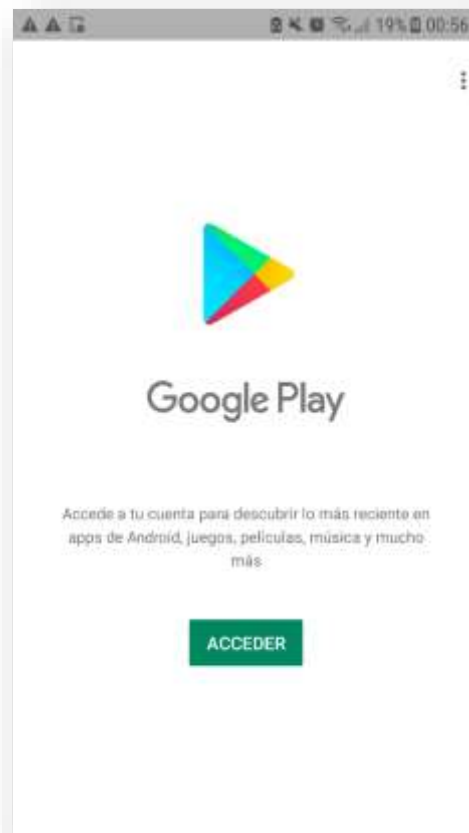
Per prima cosa abbiamo bisogno di un ambiente Linux, dato che ogni sistema Android è basato su Linux per la sicurezza e la flessibilità degli strumenti, useremo il terminale "Termux" che contiene quell'ambiente in cui installeremo gli strumenti che ci aiuteranno a creare i QRNG.

Termux è un emulatore Linux dove installeremo i pacchetti necessari per creare numeri quantici.

Uno dei principali vantaggi dell'utilizzo di Termux è la possibilità di installare programmi senza dover "ruotare" il telefono cellulare (Smartphone), il che assicura che nessuna garanzia del produttore vada persa a causa di questa installazione.

Installazione Termux.

Dal tuo cellulare, vai all'applicazione con l'icona di Google Play (play.google.com).



Ricerca per applicazione "Termux", selezionarla e avviare il processo di installazione.



Inizio dell'applicazione Termux.

Dopo l'avvio dovremo eseguire i seguenti due comandi per eseguire gli aggiornamenti dell'emulatore del sistema operativo Linux:

\$ apt aggiornamento

\$ apt aggiornamento

Confermare tutte le opzioni Y(Sì)...

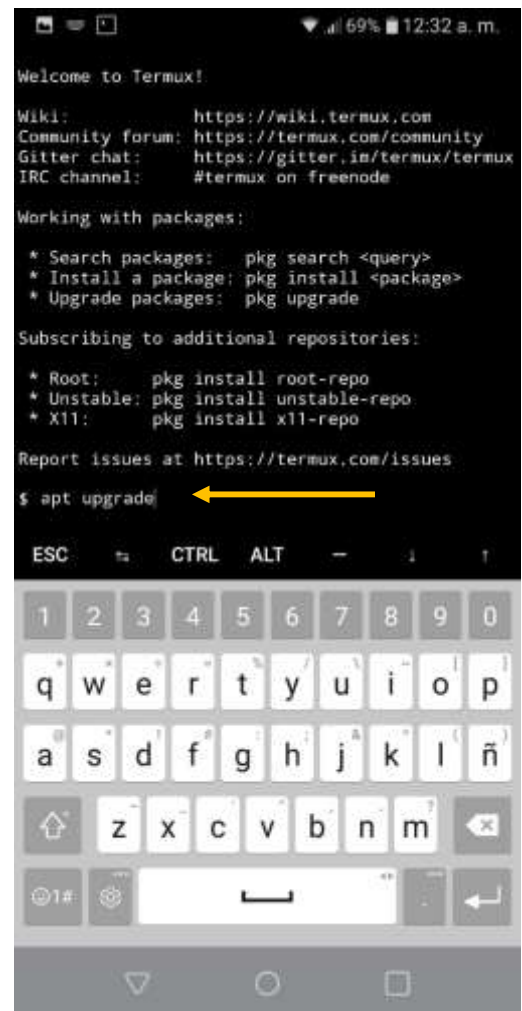
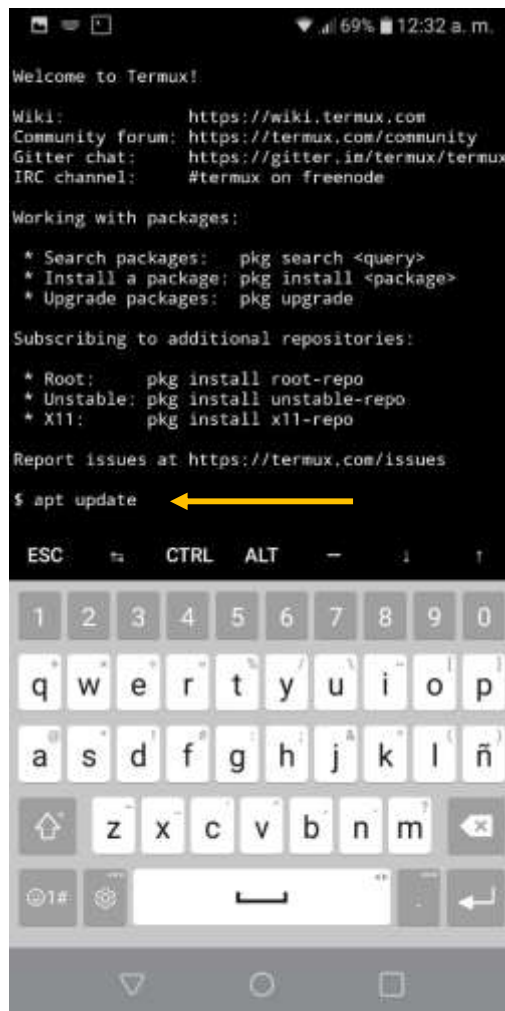
Termux

Home \$ apt aggiornamento

\$ apt



aggiornamento \$ apt



5. Configurazione della memoria all'interno di Termux.

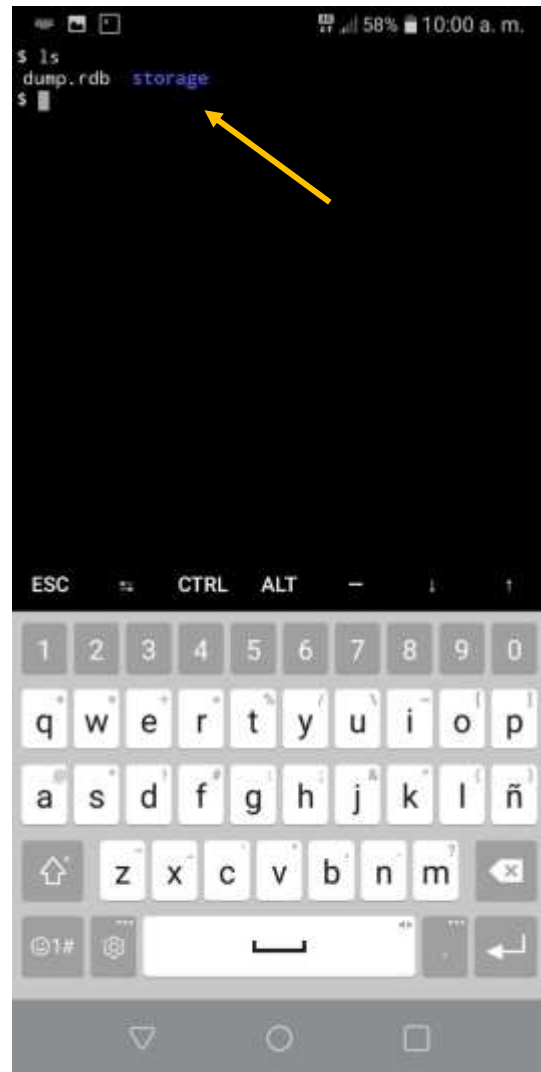
Dopo aver aggiornato e aggiornato il sistema Termux, inizieremo a configurare come visualizzare la memoria interna del telefono nel sistema Termux. Questo vi aiuterà a scambiare informazioni tra Termux e le nostre informazioni nel telefono.

Questo può essere fatto in modo semplice e veloce eseguendo il seguente comando su un terminale Termux.

\$ termux-setup-storage

Quando si esegue il comando precedente, appare una finestra che chiede di confermare la creazione di un **archivio** virtuale (directory) in Termux. Verifichiamo dando il comando:

\$ ls



6. Installazione del server SSH (Secure Shell).

\$ apt installare openssh

\$ apt installare 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 libdb libedit termux-auth
The following NEW packages will be installed:
krb5 libdns 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
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]
```

Abbiamo terminato l'installazione della rete di comunicazione per il server localhost SSH su Smartphone mobile.

7. Configurazione del server SSH su cellulare (smartphone).

Permetteremo al server SSH del cellulare di collegarsi dal nostro PC al cellulare e di lavorare in modo più veloce e confortevole, inoltre ci servirà per verificare che il servizio del server SSH nel cellulare funzioni correttamente poiché lo utilizzeremo nella comunicazione con Mini QRNG.

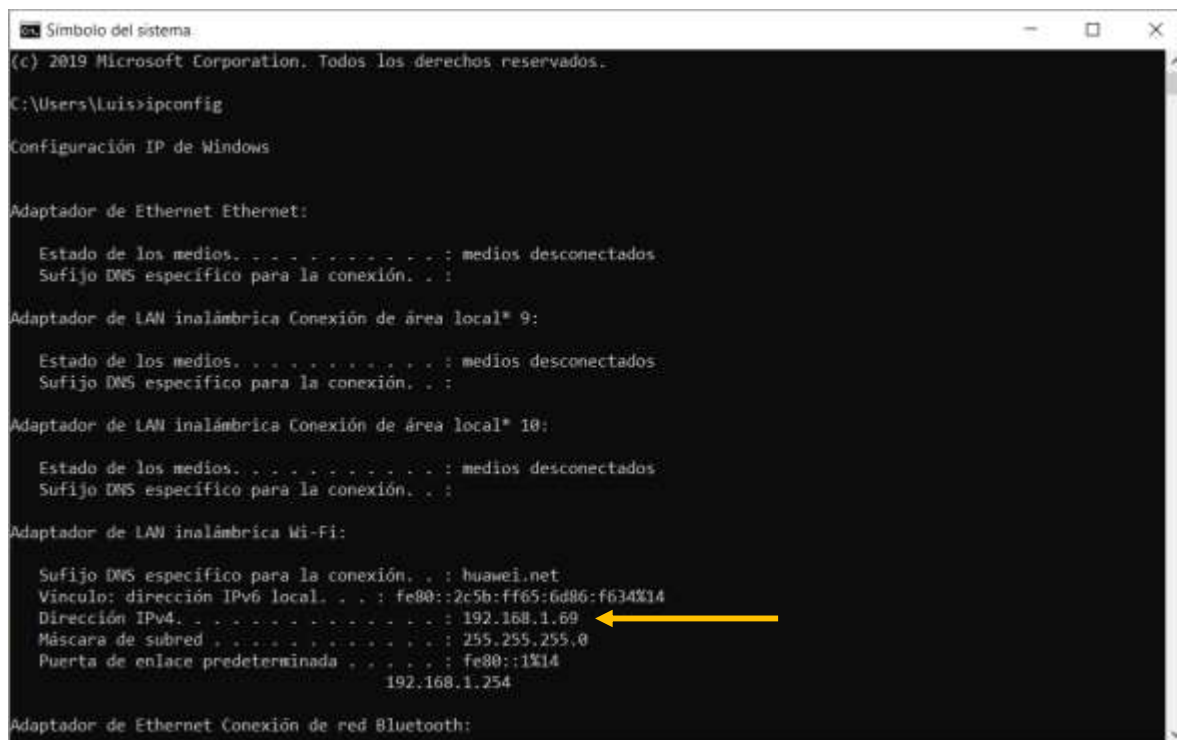
La prima cosa da fare è collegare il cellulare e il PC alla **stessa rete WiFi** in modo che possano vedersi. Gli IP o gli indirizzi devono essere simili a 192.168.XXX.XXX i valori XXX sono numeri variabili che vengono assegnati in modo casuale in ogni computer.

Questo esempio è stato testato su un cellulare LG Q6 e un PC con Windows 10 Home.

Controllare l'IP o l'indirizzo che il PC ha collegato al WiFi dobbiamo aprire un terminale in Windows.

Nel pannello inferiore dove la lente di ricerca è scritta cmd e premere il tasto Invio. Si aprirà un terminale e in esso scriveremo il comando:

C:Nome_utente> 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:ff65:6d06: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:
```

Ci mostrerà l'IP assegnato al PC in questo è 192.168.1.69 ma questo è molto probabilmente diverso in ogni caso.

NOTA: L'indirizzo dove c'è scritto "indirizzo IPv4" deve essere preso, da non confondere con il Gateway.

Ora nel caso del telefono cellulare nel terminale Termux dobbiamo digitare il seguente comando per conoscere il nome del nostro utente che useremo per connetterci al server SSH che ha il nostro telefono, eseguiamo il seguente comando:

\$ whoami

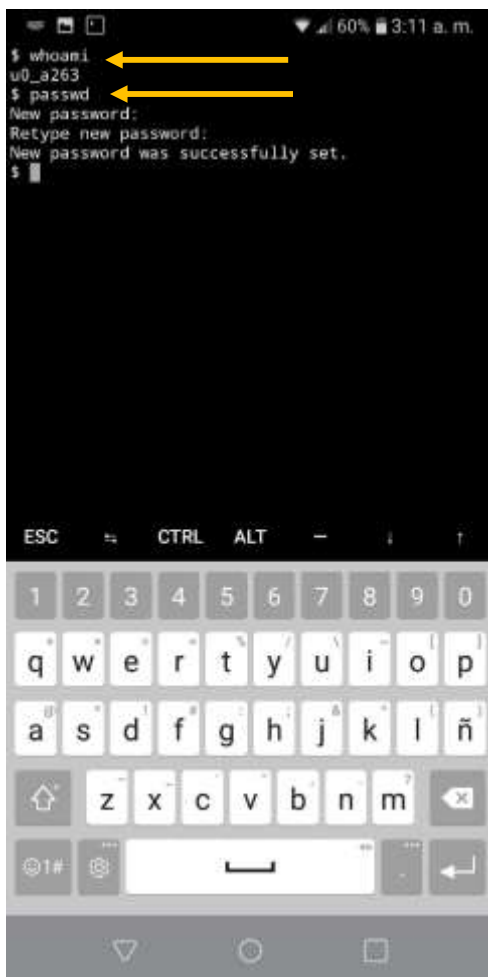
In seguito dobbiamo dare una password a questo utente, quindi dobbiamo eseguire il seguente comando:

\$ passwd

Ci chiederà di digitare una password e premere Invio, di nuovo ci chiede la password che confermiamo e premere Invio, se è stata **impostata con successo** "La nuova password è stata impostata con successo" in caso di marcatura di un errore è possibile che la password non sia stata digitata correttamente. Eseguire di nuovo la procedura.

E poi per sapere che IP abbiamo in Termux digitiamo il seguente comando, l'IP è dopo la parola "inet":

\$ ifconfig -a



```
$ whoami
u0_a263
$ passwd
New password:
Retype new password:
New password was successfully set.
$
```



```
e 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carri
er 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST
> mtu 1500
    inet 192.168.1.68 netmask 255.255.255.0
    broadcast 192.168.1.255
    inet6 fe80::257:c1ff:fee6:3051 prefixle
n 64 scopeid 0x20<link>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
    txqueuelen 1000 (UNSPEC)
    RX packets 908745 bytes 947916536 (904.
0 MiB)
    RX errors 0 dropped 0 overruns 0 fram
e 0
    TX packets 601034 bytes 93496881 (89.1
MiB)
    TX errors 0 dropped 0 overruns 0 carri
er 0 collisions 0

$
```

Ora è il momento di avviare il servizio di server SSH sul vostro telefono in modo da poter ricevere le sessioni dal vostro PC. Eseguiamo il seguente comando nel terminale Termux, questo comando non dà alcun risultato.

\$ sshd



Ora dovremo installare sul PC un programma che comunichi con il server SSH del telefono dal PC.

Dobbiamo andare su <https://www.putty.org>

Seleziona dove si trova il link "Puoi scaricare PuTTY qui".



Download PuTTY

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.

You can download PuTTY [here](#).

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](#).

Scegliete la versione a 32 bit, non importa se il vostro sistema è a 64 bit funzionerà bene.

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. You can also download the [development snapshots](#), 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](#).)

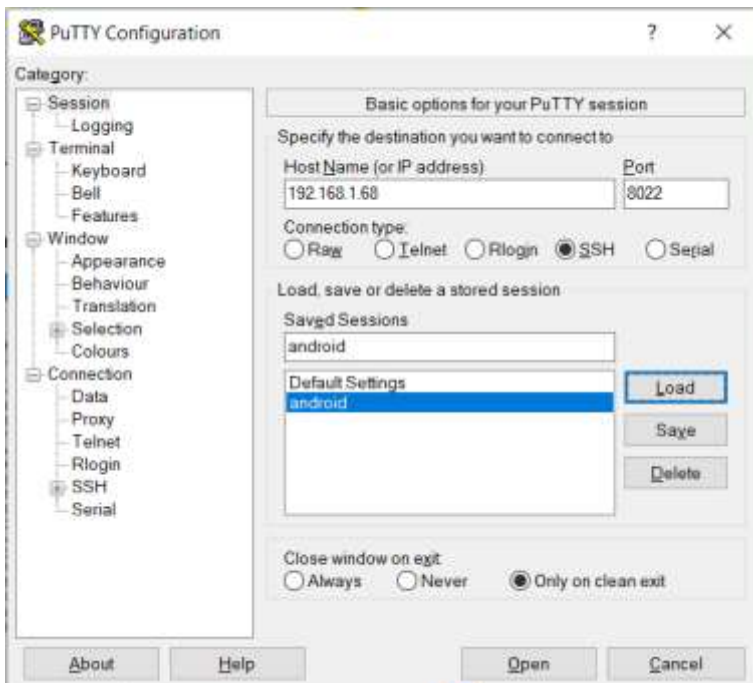
MSI ('Windows Installer')

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)
----------	-----------------------------------	-------------	-------------

Una volta scaricato sul PC, eseguirlo e installarlo con le opzioni predefinite. Poi avviare l'applicazione PuTTY.



In questa sessione inseriremo i dati del nostro server Openssh che abbiamo installato nel cellulare.

Inserire l'IP del cellulare.

HostName o indirizzo IP:

192.168.1.68 (esempio IP)

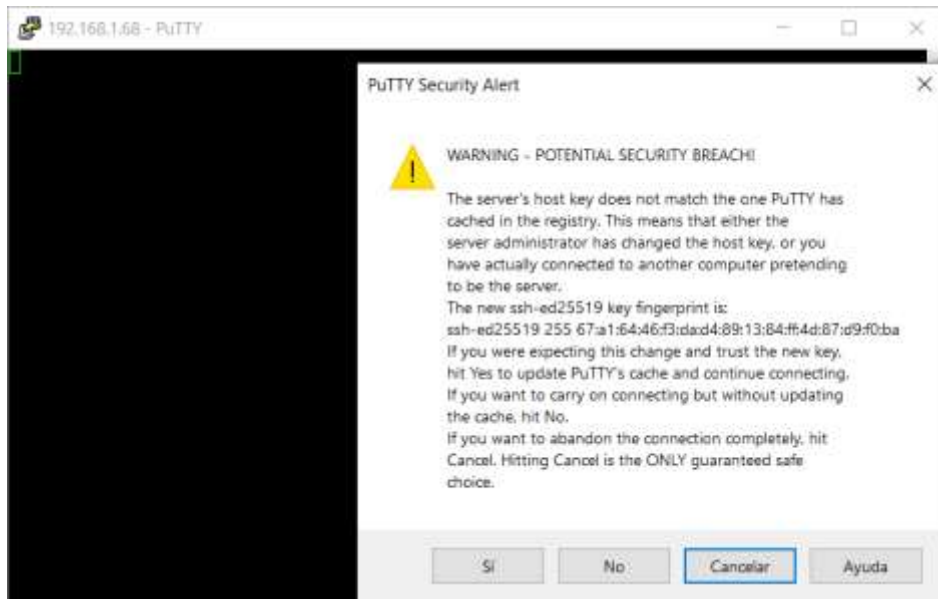
Porto:

8022 (Porta predefinita del server mobile SSH).

Possiamo dare un nome alla sessione in "Sessioni salvate" e cliccare sul pulsante Salva.

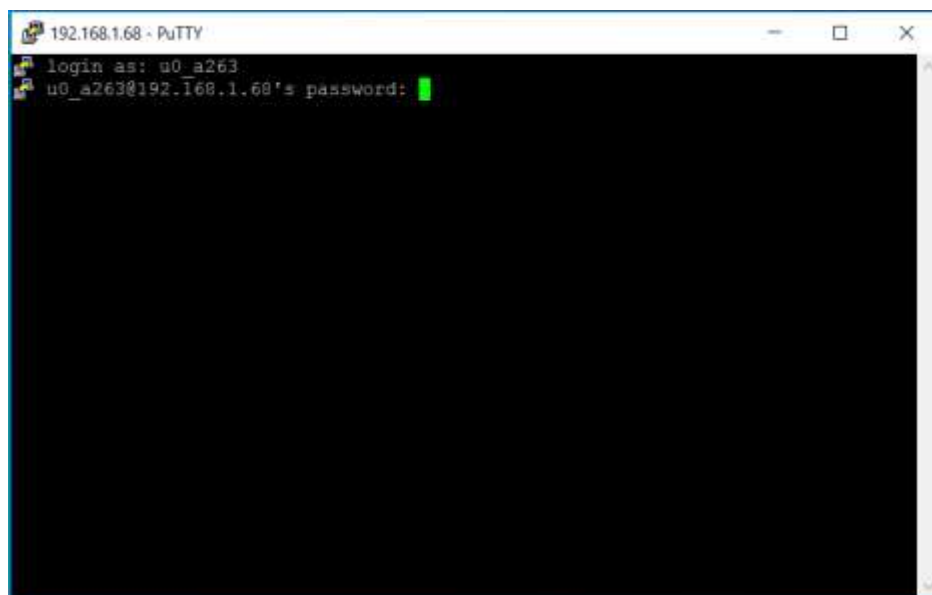
Più avanti nella parte inferiore si preme per aprire una connessione al server dando il pulsante "Apri".

Al primo collegamento sul PC vi **verrà richiesta la conferma della** chiave di cifratura delle informazioni cliccando sul tasto "Sì".

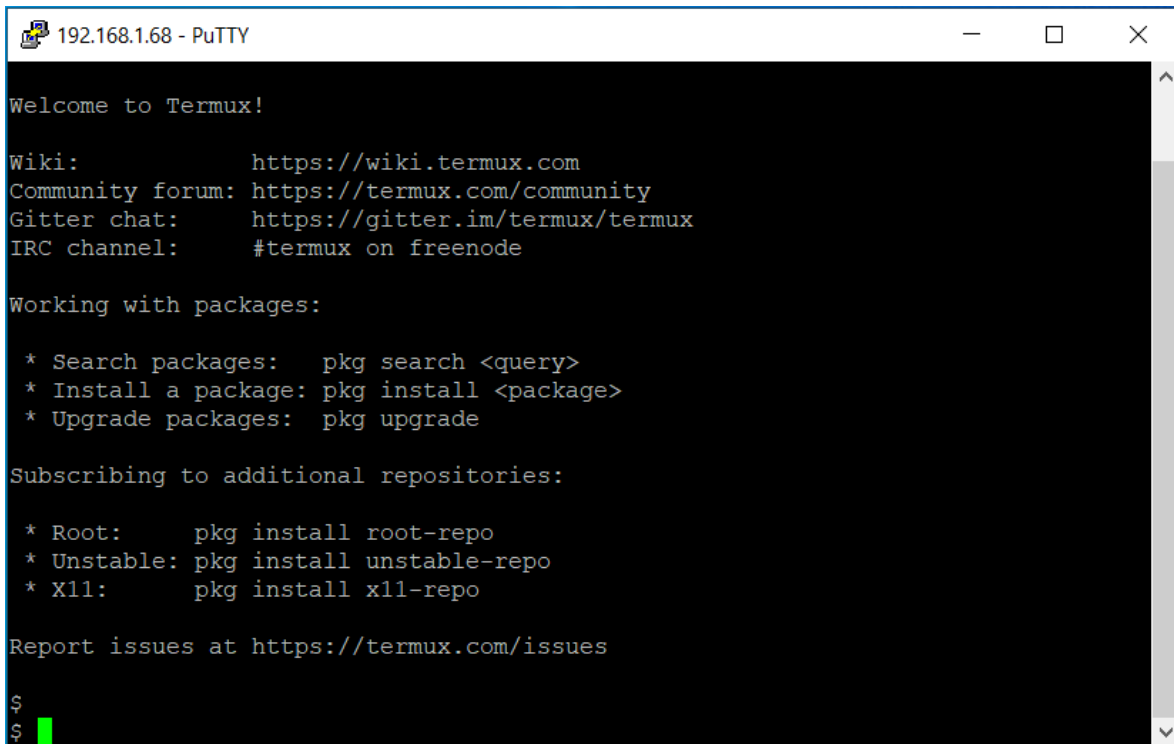


Più tardi ci verrà chiesto l'utente con cui ci collegheremo. Utilizzeremo le informazioni che abbiamo ottenuto in precedenza (utente e password).

Nel **Login come:** dobbiamo inserire il nostro utente e dare Enter, poi chiederemo di nuovo la password dando il tasto Enter.



Se i dati erano corretti, ci troveremo in una sessione SSH (Secure Shell) eseguita dal PC (Client) sul telefono (SSH Server).



```
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

$
$
```

NOTA IMPORTANTE: Ricordate che l'IP (indirizzo) del PC e l'IP (indirizzo) del telefono cellulare connesso allo stesso WiFi probabilmente cambieranno ogni volta che ci disconnettiamo e ci ricollegiamo, quindi dobbiamo controllare due volte quali indirizzi ha ogni dispositivo, questo assicurerà il successo della connessione tra i dispositivi attraverso il server SSH del telefono e il PC (Client).

8. Ambienti Blockly (App Inventor, AppyBuilder y Thunkable).

App Inventor è un ambiente di sviluppo software creato da Google Labs per costruire applicazioni per il sistema operativo Android. L'utente può, visivamente e da un insieme di strumenti di base, collegare una serie di blocchi per creare l'applicazione. Il sistema è gratuito e può essere facilmente scaricato dal web. Le applicazioni create con App Inventor sono molto facili da creare perché non è richiesta la conoscenza di alcun linguaggio di programmazione.

Tutti gli ambienti attuali che utilizzano la tecnologia di Blockly, come AppyBuilder e Thunkable, tra gli altri, hanno la loro versione gratuita, il loro modo di utilizzo può avvenire tramite internet nei loro diversi siti o può essere installato anche a casa.

I blocchi che compongono l'architettura Mini BloclyChain sono stati testati in App inventor e AppyBuilder ma a causa della loro ottimizzazione del codice dovrebbero funzionare sulle altre piattaforme.

Versioni online:

App Inventor.

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

AppyBuilder.

<http://appybuilder.com/>

Si può pensare.

<https://thunkable.com/>

Versione da installare sul vostro computer (PC):

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

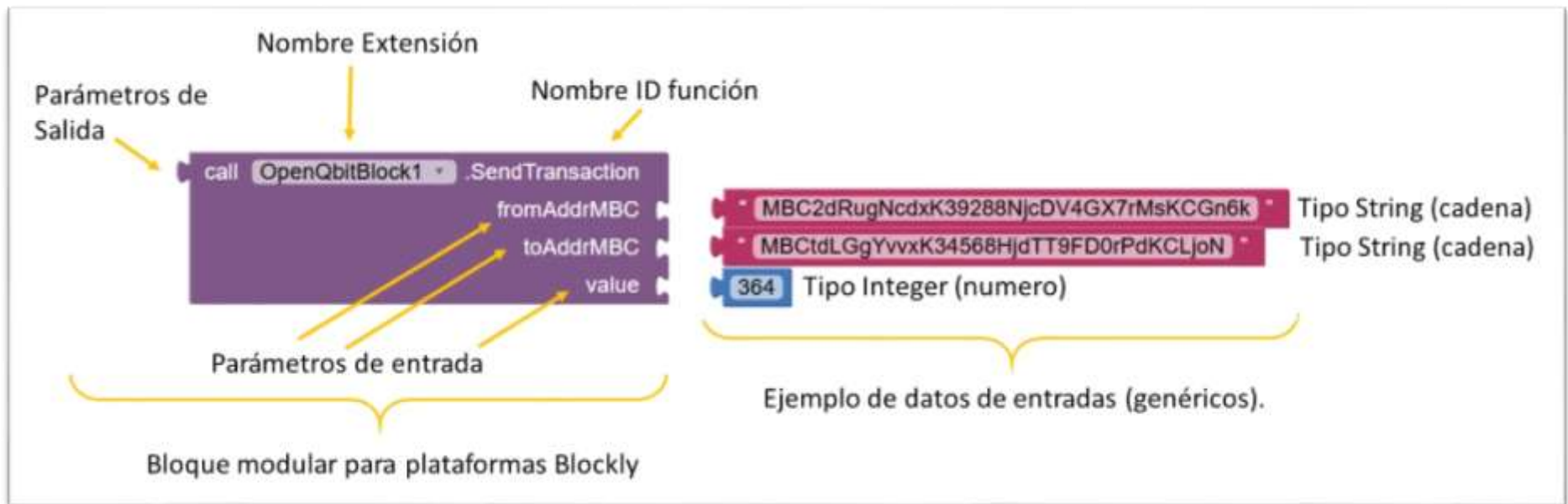
Ambiente per gli sviluppatori dei blocchi Blockly.

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

9. Definizione e uso dei blocchi in Mini QRNG.

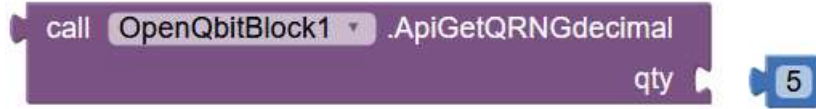
Inizieremo spiegando la distribuzione dei dati che tutti i blocchi avranno, la loro sintassi d'uso e la loro configurazione.

Nell'esempio seguente possiamo vedere un blocco modulare e i suoi parametri di ingresso e di uscita, così come i tipi di dati di ingresso, questi dati possono essere di tipo Stringa (stringa di caratteri) o Integer (intero o decimale). Mostriamo come viene utilizzato e lo configuriamo per il suo corretto funzionamento.



Ogni blocco di modulo avrà la sua descrizione e sarà nominato nel caso in cui abbia qualche dipendenza obbligatoria o opzionale di altri blocchi usati come parametri di input, il processo di integrazione sarà annunciato. Iniziamo con i blocchi dell'estensione **OpenQbitQRNG conSSH**.

Blocco per la generazione di numeri quantistici casuali decimali - (**ApiGetQRNGdecimale**)



Parametri di ingresso: **qty** <Integer>

Parametri di uscita: Dà la quantità "qty" di numeri decimali quantistici casuali inseriti nei numeri in ingresso sono all'interno dell'intervallo 0 e 1 nel formato JSON.

Esempio:

qty = 5; uscita: {"risultato": [0,5843012986202495, 0,7746497687824652, 0,05951126805960929, 0,1986079055812694, 0,036897834343989999279]}

Descrizione: Generatore di numeri casuali quantistici (QRNG) API

Blocco per la generazione di numeri quantistici casuali decimali - (**ApiGetQRNGdecimale**)



Parametri di ingresso: **qty** <Integer>, min <Integer>, max <max>

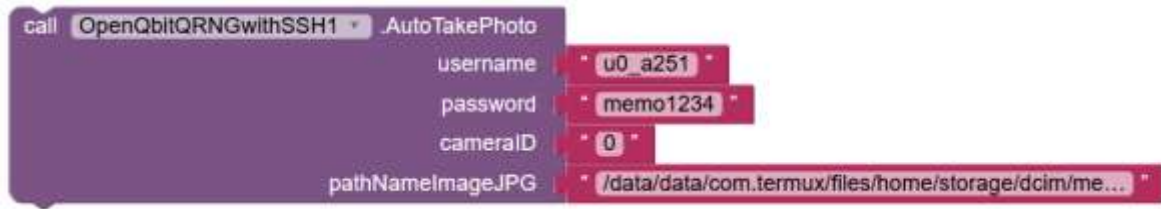
Parametri di uscita: Dà la quantità "qty" di numeri interi quantici casuali inseriti nell'ingresso i numeri si trovano nell'intervallo di min e max in formato JSON.

Esempio:

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

Descrizione: Generatore di numeri casuali quantistici (QRNG) API

Blocca per scattare una foto automaticamente - (AutoTakePhone)



Parametri di input: **username** < String>, **password** < String>, **cameraID** < String> , **pathNameImageJPG** < String>.

Dipendenza obbligatoria: per utilizzare questo blocco è necessario soddisfare due dipendenze software; installare nel terminale Termux il modulo Termux-API. Questo modulo contiene il processo per scattare automaticamente le foto e caricare il server SSH che è stato precedentemente installato.

Parametri di uscita: Fornisce una foto (immagine) in formato JPG nel percorso specificato. Nel percorso è necessario

Descrizione: Crea automaticamente una foto JPG senza l'intervento dell'utente.

Per installare il Termux-API, il seguente comando deve essere eseguito nel Termux Terminal:

\$ pkg installare 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) ...
$
```

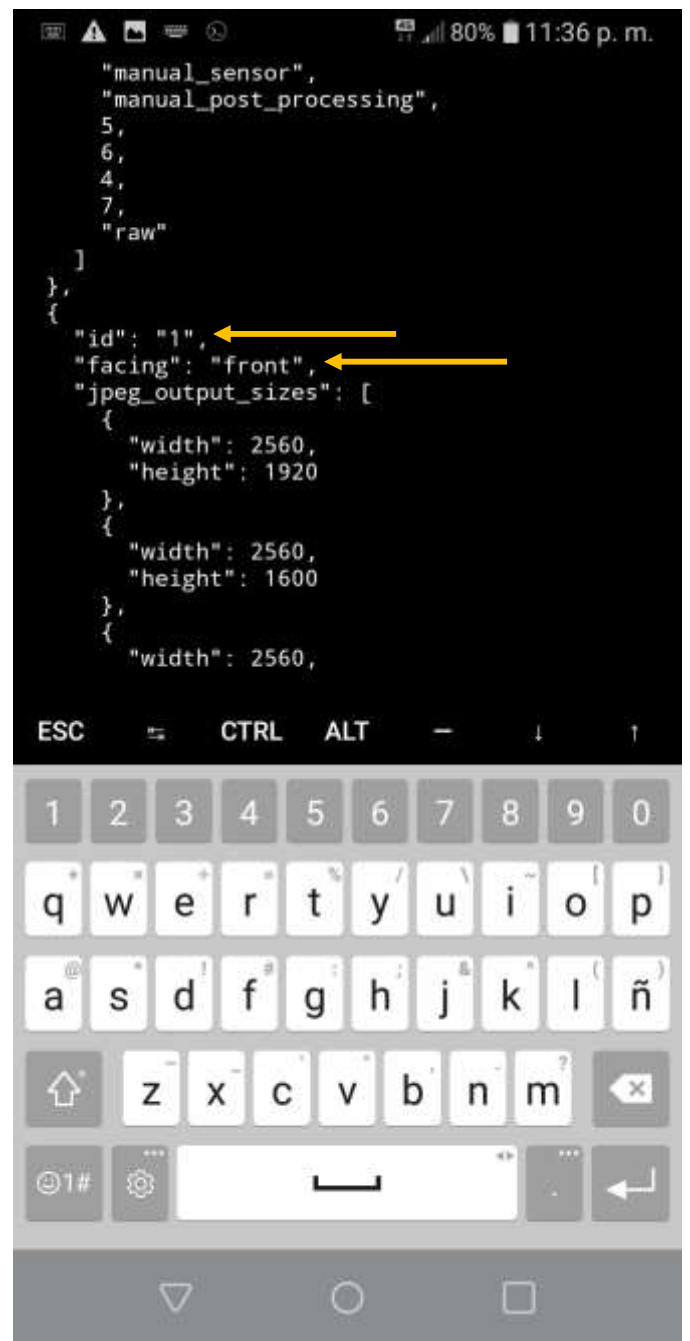
Per conoscere il numero, il numero e la posizione degli ID (identificatori di lenti fotografiche) del vostro dispositivo mobile (Smartphone), eseguite il seguente comando nel terminale Termux.

\$ 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,
```

Nel nostro esempio lo smarphone LG Q6 che abbiamo usato ha due ID "0" sul retro e "1" sul davanti.

Ora testiamo l'API per scattare una foto utilizzando l'ID "0" dell'obiettivo posteriore e diamogli un nome albitario nel nostro caso test.jpg

Ricordate che l'API fornisce solo foto in formato JPG:

\$ termux-camera-foto -c 0 test.jpg

Il comando precedente deve aver creato un file con il nome test.jpg automaticamente, se così possiamo usare il blocco (AutoTakePhoto), non dimenticate di avviare il nostro server SSH locale con il comando: **\$ sshd**

NOTA: Nel percorso variabileNameImageJPG si deve considerare il percorso all'interno del terminale Termux per accedere alla memoria dello Smartphone:

/data/dati/dati/com.termux/files/home/magazzino/dcim/esempio.jpg

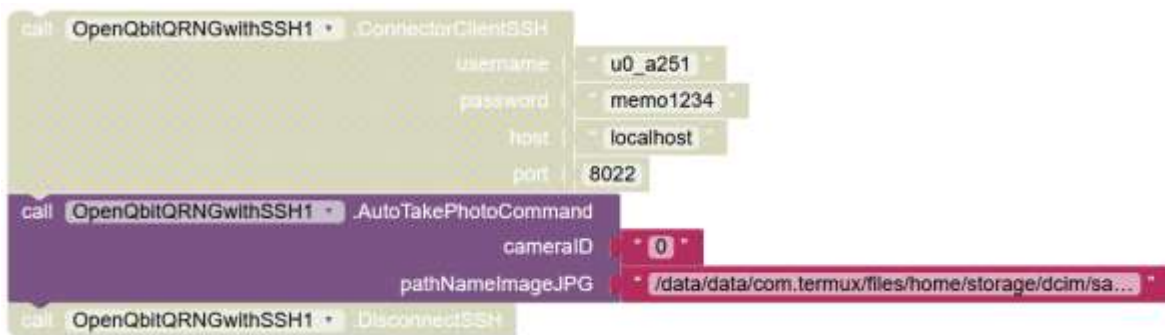
Il percorso precedente su androide sarebbe stato lo stesso:

/mnt/sdcard/dcim/example.jpg

Tuttavia, dobbiamo ricordare che nel terminale Termux il percorso valido per la visualizzazione dell'accensione del cellulare deve sempre essere considerato il percorso predefinito:

/data/dati/dati/com.termux/files/home/stoccaggio

Blocca per scattare una foto automaticamente SOLO COMANDO - (AutoTakePhoneCommand)



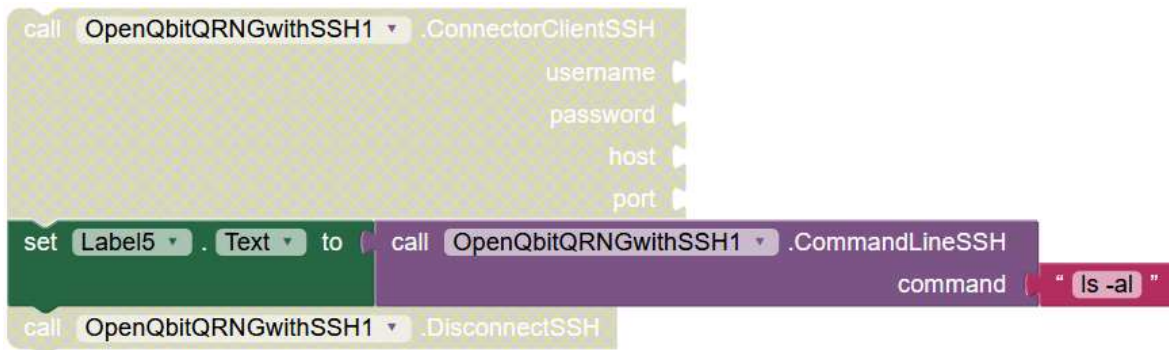
Parametri di ingresso: **cameraID** <String>, **pathNameImageJPG** <String>.

Dipendenza obbligatoria: Block (**ConnectorClientSSH**), Block (**DisconnectSSH**).

Parametri di uscita: Fornisce una foto (immagine) in formato JPG nel percorso specificato.

Descrizione: Crea automaticamente una foto JPG senza l'intervento dell'utente. Tuttavia, la differenza rispetto al blocco (AutoTakePhoto), questo blocco contiene solo il comando per creare la foto ed è necessario il blocco prima di connettersi al server SSH (**ConnectClientSSH**) e poi utilizzare il blocco (**DisconnectSSH**).

Blocco per eseguire il comando nel terminale Termux - (**CommandLineSSH**)



Parametri di ingresso: **comando** <String>

Dipendenza obbligatoria: Block (**ConnectorClientSSH**), Block (**DisconnectSSH**).

Parametri di uscita: Eseguire il comando inserito nel terminale Termux.

Descrizione: Viene eseguito un comando inserito e il blocco è necessario prima di tutto per connettersi al server SSH (**ConnectClientSSH**) e poi per utilizzare il blocco (**DisconnectSSH**).

Blocco per la connessione ad un server SSH remoto o locale - (**ConnectorClientSSH**).

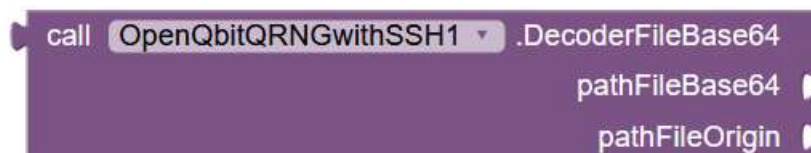


Parametri di input: **username** <string>, **password** <string>, **host** <string>, **port**<integer>, **porta**<integer>.

Parametri di uscita: Se la connessione con il server ssh del terminale Termux ha successo, ci dà un messaggio; "**Connect SSH**", se non ha successo, ci dà un messaggio **NULL**.

Descrizione: Blocco di comunicazione per collegare il server SSH scelto al terminale Termux, tramite il protocollo di comunicazione SSH (Secure Shell).

Blocco per decodificare un file con algoritmo Base64 (**DecoderFileBase64**).

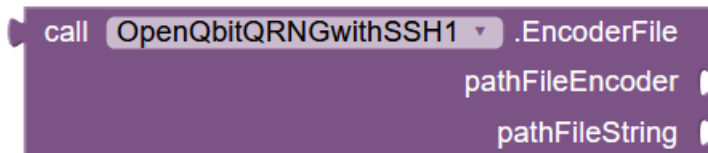


Parametri di ingresso: **pathFileBase64** <String>, **pathFileOrigin** <String>.

Parametri di uscita: File sorgente che è stato inserito nel blocco (**EncoderFileBase64**)

Descrizione: Un file Base64 viene convertito nel file originale che è stato inserito nel blocco (**EncoderFileBase64**).

Block converte un file in formato Base64 - (**EncoderFile**)

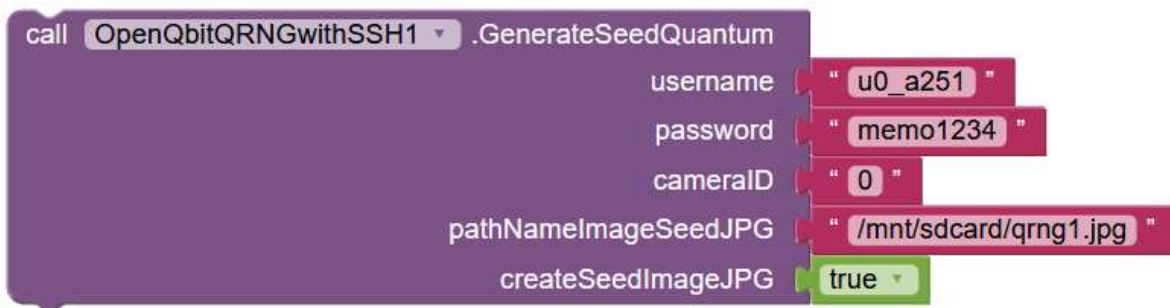


Parametri di ingresso: **pathFileOrigin** <String> , **pathFileBase64** <String> , **pathFileBase64** <String>

Parametri di uscita: File codificato Base64.

Descrizione: Converte un file sorgente di qualsiasi formato in un file Base64. I nomi dei file possono essere arbitrari e scelti dall'utente.

Blocco per generare QRNG (**Generatore** di numeri casuali quantistici) - (**GeneraSeedQuantum**)



Parametri di input: **nome utente** <Stringa>, **password** <Stringa>, **cameraID** <Stringa>, **percorsoNomeImageJPG** <Stringa>. **createSeedImageJPG** <Slean>.

Se il valore booleano è "True" quando il blocco viene eseguito ogni volta che viene creata una nuova immagine seed JPG con il nome del percorso inserito. Se il valore Boleano è "Falso" disattiviamo l'opzione di prendere un'immagine JPG (foto) e possiamo indicare manualmente dove si trova un'immagine nella nostra selezione, che può essere di qualsiasi formato.

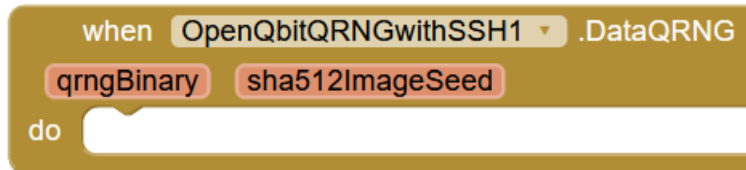
NOTA: I migliori risultati per la generazione di un QRNG si basano su un'immagine formattata "RAW". Esempio di formato DNG.

Dipendenza obbligatoria: il suddetto Termux-API deve essere installato nel blocco (AutoTakePhoto).

Parametri di uscita: L'evento viene eseguito (**DataQRNG**) e ci fornisce due valori:

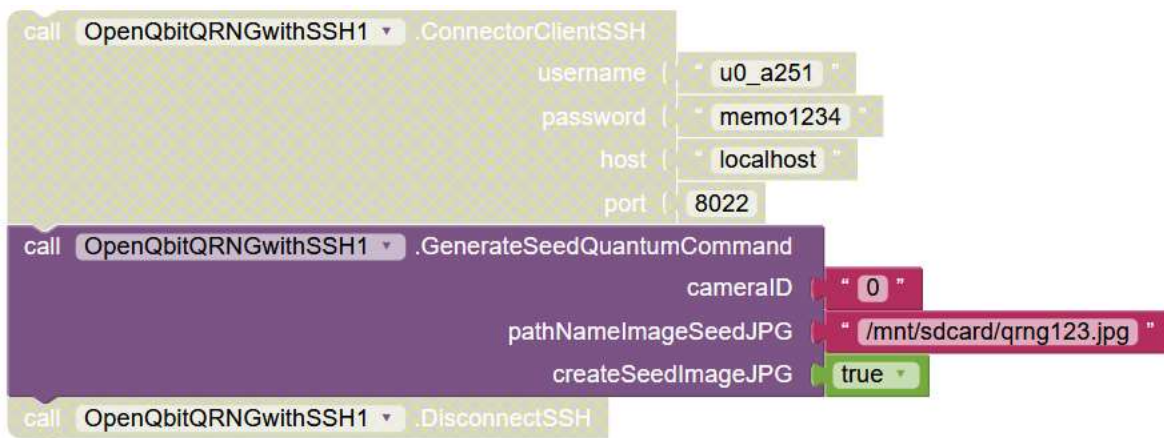
qrngBinary.- Una stringa di numeri binari casuali

Sha512ImageSeed - Sha512 dall'immagine del seme JPG da cui provengono i numeri casuali.



Descrizione: genera numeri quantistici casuali (QRNG) attraverso il sensore ottico della fotocamera del cellulare. L'algoritmo si basa sulla cattura di foto casuali e l'algoritmo viene applicato ad esso per fornire una stringa di numeri binari.

Blocco per generare QRNG (Quantum Random Number Generator) - (GenerateSeedQuantumCommand).



Parametri di ingresso: **cameraID** <String>, **pathNameImageJPG** <String>. **createSeedImageJPG** <Blean>.

Se il valore booleano è "True" quando il blocco viene eseguito ogni volta che viene creata una nuova immagine seed JPG con il nome del percorso inserito. Se il valore Boleano è "Falso" disattiviamo l'opzione di prendere un'immagine JPG (foto) e possiamo indicare manualmente dove si trova un'immagine nella nostra selezione, che può essere di qualsiasi formato.

Dipendenza obbligatoria: Block (**ConnectorClientSSH**), Block (**DisconnectSSH**).

Parametri di uscita: L'evento viene eseguito (**DataQRNG**) e ci fornisce due valori:

qrngBinary.- Una stringa di numeri binari casuali

Sha512ImageSeed - Sha512 dall'immagine del seme JPG da cui provengono i numeri casuali.



Descrizione: genera numeri quantistici casuali (QRNG) attraverso il sensore ottico della fotocamera del cellulare. L'algoritmo si basa sulla cattura di foto casuali e l'algoritmo viene applicato ad esso per fornire una stringa di numeri binari.

Tuttavia, la differenza rispetto al blocco (**GenerateSeedQuantum**), questo blocco contiene solo il comando per eseguire l'algoritmo di reazione QRNG ed è necessario il blocco prima di connettersi al server SSH (**ConnectClientSSH**) e poi utilizzare il blocco (**DisconnectSSH**).

Blocca per ottenere l'entropia di Shannon in un'immagine (foto) - (**GetShannonEntropyFile**).



Parametri di input: **nome utente** <Stringa> , **password** <Stringa> , **percorsoFileImage** <Stringa>, **pathFileImage**

Dipendenza obbligatoria: il modulo Shannon_entropy deve essere installato nel terminale Termux.

Parametri di uscita: Fornisce l'entropia di un'immagine

Esempio:

Uscita: 8.94596789873

Descrizione: Ci dà l'entropia di un'immagine. L'entropia è il parametro fondamentale per la generazione di numeri casuali di buona qualità, maggiore è l'entropia, migliori sono i risultati.

Per installare il modulo Shannon entropia, dobbiamo prima installare il pacchetto Python e poi installare il modulo Pillow e Shannon_entropy con i seguenti comandi nel terminale Termux.

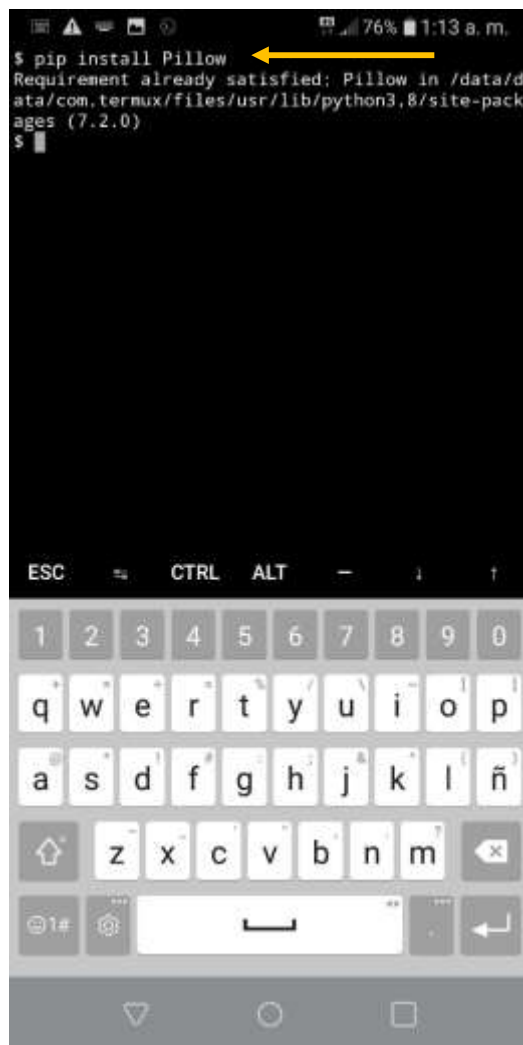
\$ apt installare Python

\$ pip installare Pillow

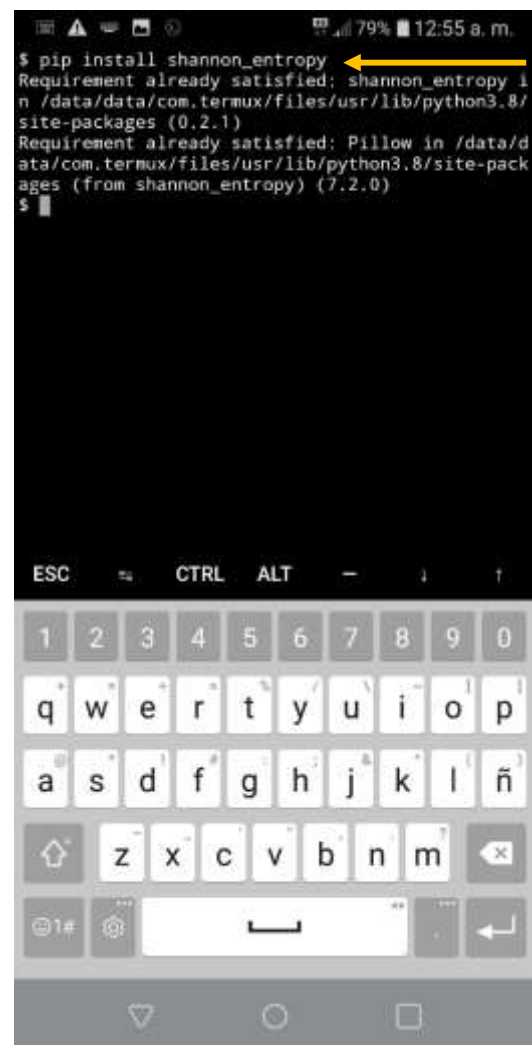
\$ pip installare Shannon_entropy



```
$ 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)
$
```

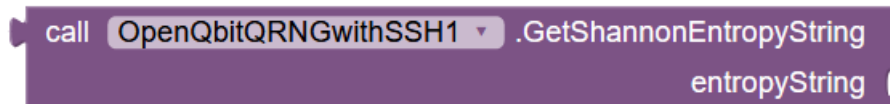
Poi dobbiamo creare nella directory "Home" di Termux il seguente file Python chiamato "entropy.py" con il seguente codice all'interno.

```
Dall'immagine di importazione del PIL  
  
Importazione di matematica  
  
Dall'importazione di Shannon_entropy *  
  
Importazione  
  
Img=Image.open(sys.argv[1])  
  
stampa(Shannon_entropia(img))
```

Salviamo il file e abbiamo il nostro ambiente da utilizzare con il blocco (GetShannonEntropyFile).

Suggerimento: Infatti, con questa installazione Python è possibile creare i propri programmi in questo linguaggio ed eseguirli attraverso il blocco (ConnectorClientSSH).

Bloccare per ottenere l'entropia di Shannon da una stringa - (GetShannonEntropyString).



Parametri di ingresso: entropyString>Stringa>Stringa

Parametri di uscita: Fornisce l'entropia di una stringa di caratteri.

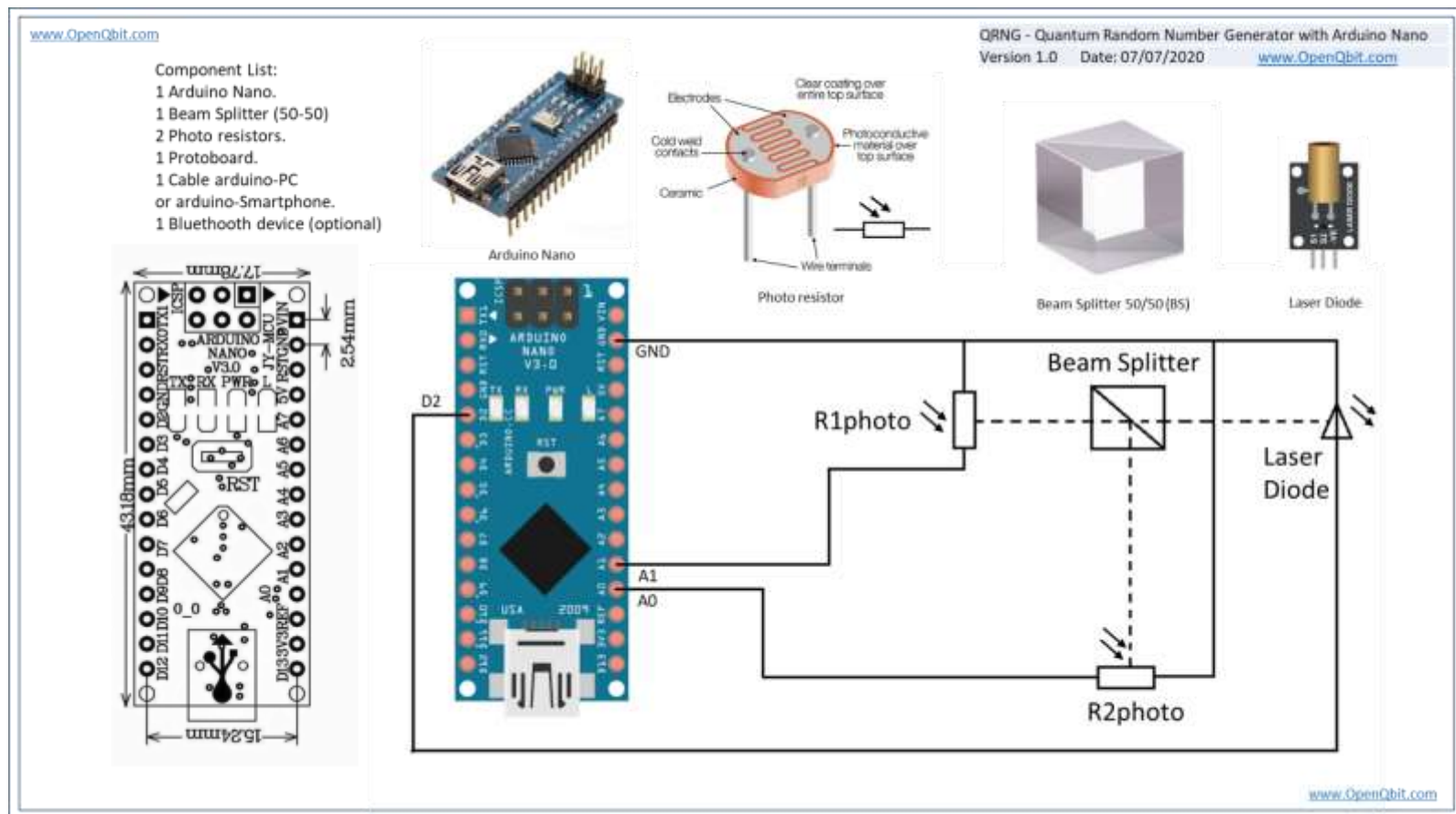
Esempio:

Uscita: 5.76002345671

Descrizione: Ci dà l'entropia di una stringa di caratteri. L'entropia è il parametro fondamentale per la generazione di numeri casuali di buona qualità, maggiore è l'entropia, migliori sono i risultati.

10. Creazione di un dispositivo "Hardware" di un QRNG.

Ora creeremo un dispositivo fisico "Hardware" per generare numeri casuali quantistici (QRNG) con componenti economici che possono essere facilmente assemblati a casa.



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.....

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

www.OpenQbit.com

Compilazione del programma QRNGv10.ino e caricamento su...



The screenshot shows the Arduino IDE interface with the file 'QRNGv10.ino' open. The code is written in C++ and includes comments in Italian. It defines pins for a quantum circuit, sets up a serial connection, and implements a random number generation function based on photon measurements. The IDE's status bar at the bottom indicates that the sketch was compiled successfully, using 2352 bytes of program memory and 198 bytes of dynamic memory.

```
QRNGv10 Arduino 1.8.10
Archivo Editar Programa Herramientas Ayuda

QRNGv10

int triggerQ = 2; // This pin will pulse our quantum circuit.
int Qu0Pin = A0; // This pin measures the horizontal polarized photons
int Qu1Pin = 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(5000);
}

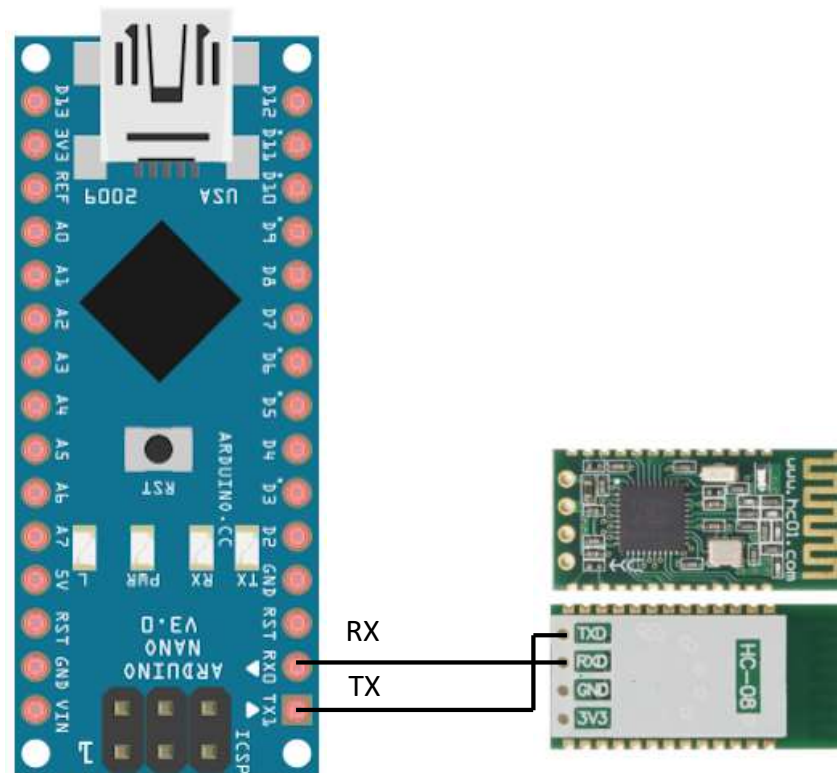
int Random() {
  // Pulse the Laser
  digitalWrite(triggerQ, HIGH);
  delay(300);
  digitalWrite(triggerQ, LOW);
  delay(300);
  // Read the photoreistors
  Qu0 = analogRead(Qu0Pin);
  Qu1 = analogRead(Qu1Pin);
  // Determine random bit
  if(Qu0>Qu1) { // More photons in the Qu0 mode, return 0
    return 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, but our
    // program will simply run the function repeatedly until
    // a random bit can be generated.
    return Random();
  }
}

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

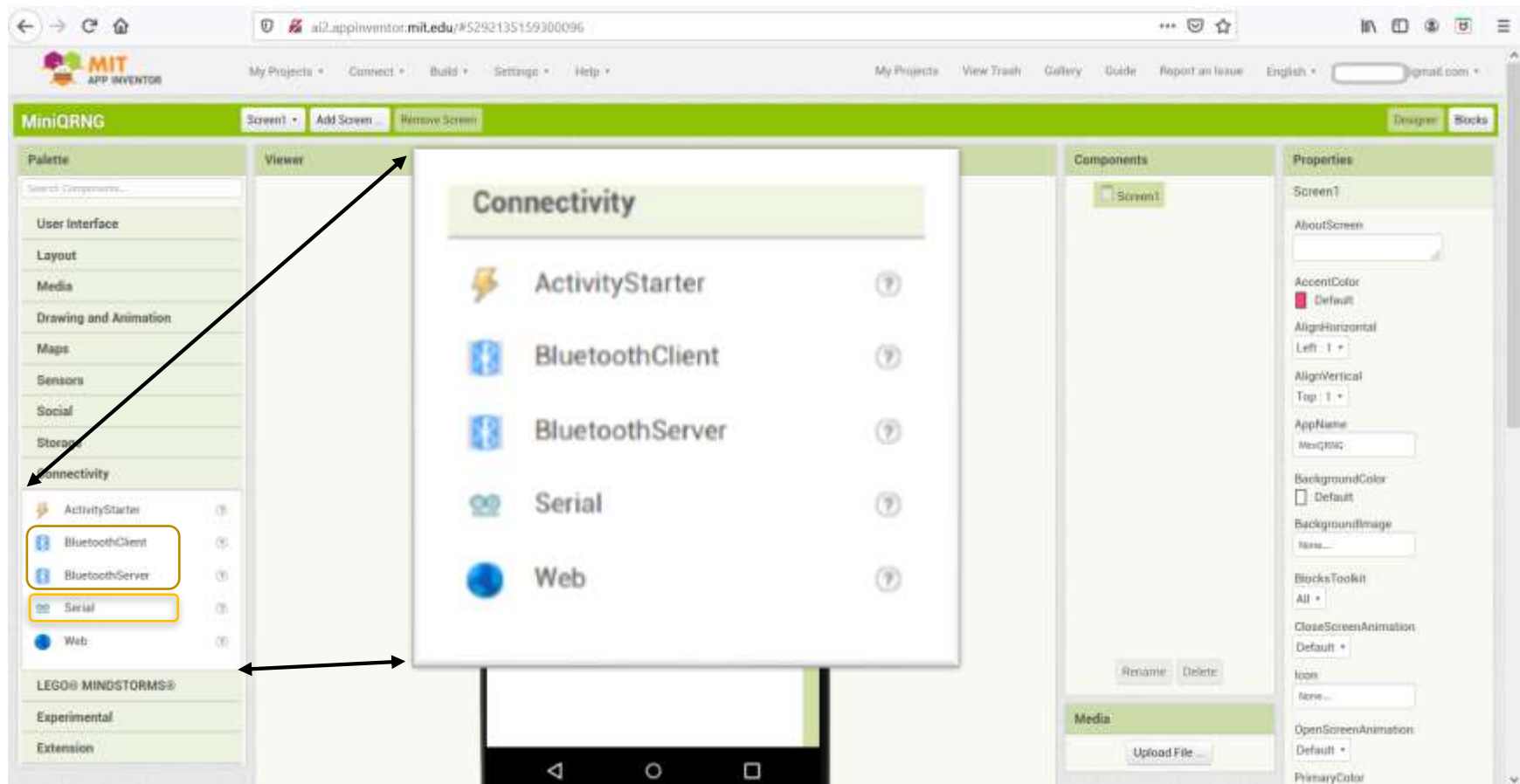
Compilado

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 COM4



I seguenti componenti seriali o Bluetooth possono essere utilizzati per collegare App Inventor ad Arduino:



Ora compilato e caricato il programma QRNGv10.ino manca solo la comunicazione con l'arduo nano per salvare i dati (numeri casuali quantistici) questi saranno in formato binario, tuttavia, i dati ottenuti possono essere facilmente passati ad un altro formato come esadecimale o decimale a seconda del requisito finale.

Infine, per vedere un esempio di come funziona la connessione seriale o Bluetooth, ecco alcuni link di riferimento.

Ricordate che tutto è attraverso la programmazione Blockly per essere testato con App Inventor questo ha già blocchi per la comunicazione con arduino seriale o altro sistema blockly può essere a tele bluetooth simili on-line.

http://kio4.com/appinventor/9A0_bluetooth_RXTX.htm

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

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

11. Allegato "OpenQbit Quantum Computing".

Come funziona il calcolo quantistico? ⁽²⁾

La trasformazione digitale sta portando il mondo a un cambiamento più rapido che mai: ci credereste che l'era digitale sta per finire? **L'alfabetizzazione digitale** è già stata identificata come un'area in cui la conoscenza aperta e le opportunità accessibili di apprendere la tecnologia sono urgenti per affrontare le lacune nello sviluppo sociale ed economico. Imparare dai concetti chiave dell'era digitale diventerà ancora più critico con l'imminente arrivo di un'altra nuova ondata tecnologica in grado di trasformare i modelli esistenti con velocità e potenza sorprendenti: **le tecnologie quantistiche**.

In questo articolo confrontiamo i concetti di base del calcolo tradizionale e del calcolo quantistico; e iniziamo anche ad esplorare la loro applicazione in altre aree correlate.

Cosa sono le tecnologie quantistiche?

Nel corso della storia, gli esseri umani hanno sviluppato la tecnologia così come hanno capito come funziona la natura attraverso la scienza. Tra il 1900 e il 1930, lo studio di alcuni fenomeni fisici non ancora ben compresi ha dato origine ad una nuova teoria fisica, la **Meccanica Quantistica**. Questa teoria descrive e spiega il funzionamento del mondo microscopico, l'habitat naturale di molecole, atomi o elettroni. Grazie a questa teoria, non solo è stato possibile spiegare questi fenomeni, ma è stato anche possibile comprendere che la realtà subatomica funziona in modo completamente contro-intuitivo, quasi magico, e che nel mondo microscopico avvengono eventi che non si verificano nel mondo macroscopico.

Queste **proprietà quantistiche** comprendono la sovrapposizione quantistica, l'entanglement quantistico e il teletrasporto quantistico.

- **La sovrapposizione quantistica** descrive come una particella può trovarsi in diversi stati contemporaneamente.
- **L'entanglement quantistico** descrive come due particelle così distanti tra loro possano essere correlate in modo tale che, interagendo con una, l'altra ne sia consapevole.
- **Il teletrasporto quantistico** usa l'entanglement quantistico per inviare informazioni da un luogo ad un altro nello spazio senza doverlo attraversare.

Le tecnologie quantistiche si basano su queste proprietà quantistiche di natura subatomica.

In questo caso, oggi la comprensione del mondo microscopico attraverso la Meccanica Quantistica ci permette di inventare e progettare tecnologie in grado di migliorare la vita delle persone. Ci sono molte e molto diverse tecnologie che utilizzano i fenomeni quantistici e alcune di esse, come i laser o la risonanza magnetica (MRI), sono con noi da più di mezzo secolo. Tuttavia, stiamo attualmente assistendo ad una rivoluzione tecnologica in settori

quali il calcolo quantistico, l'informazione quantistica, la simulazione quantistica, l'ottica quantistica, la metrologia quantistica, gli orologi quantistici o i sensori quantistici.

Cos'è il calcolo quantistico? Per prima cosa, bisogna capire l'informatica classica.




FIGURA 1.
Ejemplos de caracteres en lenguaje binario.

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

Per

capire come funzionano i computer quantistici, è conveniente spiegare prima di tutto come funzionano i computer che usiamo ogni giorno, che in questo documento chiameremo computer digitali o classici. Questi, come il resto dei dispositivi elettronici come i tablet o i telefoni cellulari, utilizzano i bit come unità fondamentali della memoria. Ciò significa che i programmi e le applicazioni sono codificati in bit, cioè in linguaggio binario di zeri e uno. Ogni volta che interagiamo con uno di questi dispositivi, ad esempio, premendo un tasto sulla tastiera, si creano, distruggono e/o modificano stringhe di zeri e uno all'interno del computer.

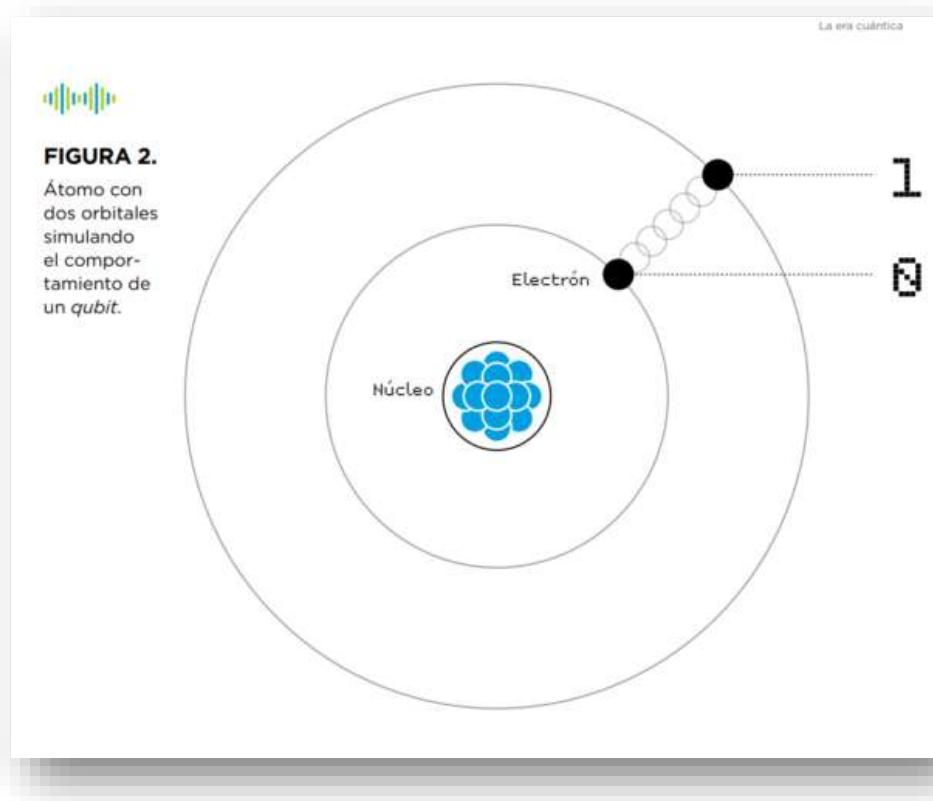
La domanda interessante è: quali sono questi zeri e quali sono fisicamente all'interno del computer? Gli stati zero e uno corrispondono alla corrente elettrica che circola, o meno, attraverso parti microscopiche chiamate transistor, che fungono da interruttori. Quando non scorre corrente, il transistor è "spento" e corrisponde al bit 0, e quando scorre è "acceso" e corrisponde al bit 1.

Più semplicemente, è come se i bit 0 e 1 corrispondessero a fori, in modo che un foro vuoto sia un bit 0 e un foro occupato da un elettrone sia un bit 1. Per questo motivo questi apparecchi sono chiamati elettronici. Come esempio, la figura 1 mostra la scrittura binaria di alcuni caratteri. Ora che abbiamo un'idea di come funzionano i computer di oggi, cerchiamo di capire come funzionano i quantum.

Da bit a qubits

L'unità fondamentale dell'informazione nel calcolo quantistico è il bit quantistico o qubit. I Qubit sono, per definizione, sistemi quantistici a due livelli - vedremo qui degli esempi - che, come i bit, possono essere a basso livello, che corrisponde ad uno stato di bassa eccitazione

o di energia definito come 0, o ad alto livello, che corrisponde ad uno stato di eccitazione superiore o definito come 1. Tuttavia, e qui sta la differenza fondamentale con il calcolo classico, i qubit possono anche trovarsi in uno qualsiasi degli infiniti stati intermedi tra 0 e 1, come ad esempio uno stato che è metà 0 e metà 1, o tre quarti di 0 e un quarto di 1.



Algoritmi quantistici, calcolo esponenzialmente più potente ed efficiente

Lo scopo dei computer quantistici è quello di sfruttare queste proprietà quantistiche dei *qubit*, come sistemi quantistici che sono, al fine di eseguire algoritmi quantistici che utilizzano la sovrapposizione e l'interleaving per fornire una potenza di elaborazione molto maggiore rispetto ai classici. È importante sottolineare che il vero cambiamento di paradigma non consiste nel fare la stessa cosa che fanno i computer digitali o classici -quelli attuali- ma più velocemente, come si può leggere in molti articoli, ma che gli algoritmi quantistici permettono di eseguire certe operazioni in un modo totalmente diverso che in molti casi si rivela più efficiente - cioè in molto meno tempo o utilizzando molto meno risorse computazionali.

Vediamo un esempio concreto di ciò che questo comporta. Immaginiamo di essere a Bogotá e vogliamo sapere il percorso migliore per arrivare a Lima tra un milione di opzioni per arrivarci ($N=1.000.000$). Per utilizzare i computer per trovare il percorso ottimale dobbiamo digitalizzare 1.000.000 di opzioni, il che implica la loro traduzione in linguaggio bit per il computer classico e in *qubit per il computer* quantistico. Mentre un computer classico

dovrebbe andare uno ad uno analizzando tutti i percorsi fino a trovare quello desiderato, un computer quantistico sfrutta il processo noto come parallelismo quantistico che gli permette di considerare tutti i percorsi contemporaneamente. Ciò implica che, mentre il computer classico ha bisogno dell'ordine di $N/2$ passi o iterazioni, cioè 500.000 tentativi, il computer quantistico troverà il percorso ottimale dopo solo \sqrt{N} operazioni sul registro, cioè 1.000 tentativi.

Nel caso precedente il vantaggio è quadratico, ma in altri casi è addirittura esponenziale, il che significa che con n *qubit* si può ottenere una capacità di calcolo equivalente a 2^n bit. Per esemplificare questo, è comune contare che con circa 270 qubit potremmo avere più stati di base in un computer quantistico - più stringhe di caratteri diversi e simultanei - rispetto al numero di atomi nell'universo, che è stimato intorno ai 280. Un altro esempio è che si stima che con un computer quantistico tra 2000 e 2500 *qubit* *potremmo* rompere praticamente tutta la crittografia utilizzata oggi (la cosiddetta crittografia a chiave pubblica).

Perché è importante conoscere la tecnologia quantistica?

Siamo in un momento di trasformazione digitale in cui diverse tecnologie emergenti come blockchain, intelligenza artificiale, droni, Internet delle cose, realtà virtuale, 5G, stampanti 3D, robot o veicoli autonomi sono sempre più presenti in molteplici campi e settori. Queste tecnologie, chiamate a migliorare la qualità della vita dell'essere umano accelerando lo sviluppo e generando un impatto sociale, avanzano oggi in modo parallelo. Solo raramente vediamo aziende che sviluppano prodotti che sfruttano combinazioni di due o più di queste tecnologie, come blockchain e IoT o droni e intelligenza artificiale. Sebbene siano destinati a convergere, generando così un impatto esponenzialmente maggiore, la fase iniziale di sviluppo in cui si trovano e la scarsità di sviluppatori e di persone con profili tecnici fanno sì che la convergenza sia ancora un compito in sospeso.

A causa del loro potenziale dirompente, ci si aspetta che le tecnologie quantistiche non solo convergano con tutte queste nuove tecnologie, ma che abbiano un'influenza trasversale su praticamente tutte. Il calcolo quantistico minaccerà l'autenticazione, lo scambio e l'archiviazione sicura dei dati, con un impatto importante sulle tecnologie in cui la crittografia ha un ruolo più rilevante, come la sicurezza informatica o la blockchain, e un impatto negativo minore, ma anche da considerare in tecnologie come il 5G, l'IoT o i droni.

Volete praticare il calcolo quantistico?

Decine di simulatori di computer quantistici sono già disponibili in rete con diversi linguaggi di programmazione già in uso come C, C++, Java, Matlab, Maxima, Python o Octave. Inoltre, nuovi linguaggi come il Q#, lanciato da Microsoft. È possibile esplorare e giocare con una macchina quantistica virtuale attraverso piattaforme come IBM e Rigetti.

Mini QRNG è creato dalla società OpenQbit.com che si concentra sullo sviluppo di tecnologie basate sul calcolo quantistico per diversi tipi di settori sia privati che pubblici.

Perché Mini QRNG è diverso dagli altri QRNG, semplicemente perché il sistema è stato creato per essere modulare e può essere facilmente assemblato in casa ad un costo abbastanza basso.

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

12. Licenze e utilizzo del software.

Androide

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

Termux

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

Nodo

<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>

Stucco SSH

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

MIT App Inventor 2 Companion e App Inventor Blockly

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

Estensioni esterne:

JSOINTOOLS

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

Per le licenze opensource e le versioni commerciali del sistema QRNG Mini si veda il sito ufficiale <http://www.openqbit.com>

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

Mini QRNG è di pubblico dominio.

Tutto il codice e la documentazione in Mini QRNG è stato dedicato al pubblico dominio dagli autori. Tutti gli autori di codici e i rappresentanti delle società per le quali lavorano hanno firmato delle dichiarazioni giurate che dedicano i loro contributi al pubblico dominio e gli originali di tali dichiarazioni sono conservati in una cassaforte presso la sede centrale di OpenQbit Mexico. Chiunque è libero di pubblicare, utilizzare o distribuire le estensioni originali Mini QRNG (OpenQbit), sia come codice sorgente che come binari compilati, per qualsiasi scopo, commerciale o non commerciale, e con qualsiasi mezzo.

Il paragrafo precedente si applica al codice e alla documentazione consegnabile in Mini QRNG quelle parti della libreria Mini QRNG che in realtà raggruppano e spediscono con un'applicazione più ampia. Alcuni script usati come parte del processo di compilazione (per esempio, script di "configurazione" generati da autoconf) possono essere inclusi in altre

licenze open source. Tuttavia, nessuno di questi script di compilazione lo rende nella libreria finale distribuibile QRNG Mini, quindi le licenze associate a questi script non dovrebbero essere un fattore di valutazione dei vostri diritti di copia e di utilizzo della libreria QRNG Mini.

Tutto il codice di consegna in Mini QRNG è stato scritto da zero. Nessun codice è stato preso da altri progetti o da internet. Ogni riga di codice può essere fatta risalire al suo autore originale, e tutti questi autori hanno dediche di pubblico dominio in archivio. Pertanto, la base di codici QRNG Mini è pulita e non contaminata da codici concessi in licenza da altri progetti open source, non da contributi aperti

Mini QRNG è open source, il che significa che potete fare tutte le copie che volete e fare quello che volete con quelle copie, senza limitazioni. Ma Mini QRNG non è open source. Per mantenere il Mini QRNG di pubblico dominio e garantire che il codice non sia contaminato da contenuti proprietari o concessi in licenza, il progetto non accetta patch di persone sconosciute. Tutto il codice in Mini QRNG è originale, in quanto è stato scritto appositamente per l'uso da parte di Mini QRNG. Nessun codice è stato copiato da fonti sconosciute su Internet.

Mini QRNG è di pubblico dominio e non richiede una licenza. Tuttavia, alcune organizzazioni vogliono una prova legale del loro diritto di utilizzare Mini QRNG. Le circostanze in cui ciò si verifica sono le seguenti:

- La vostra azienda vuole un risarcimento per le rivendicazioni di violazione del diritto d'autore.
- State utilizzando Mini QRNG in una giurisdizione che non riconosce il pubblico dominio.
- State utilizzando Mini QRNG in una giurisdizione che non riconosce il diritto di un autore di dedicare il proprio lavoro al pubblico dominio.
- Volete avere un documento legale tangibile come prova che avete il diritto legale di utilizzare e distribuire Mini QRNG.
- Il vostro ufficio legale vi dice che dovete acquistare una licenza.

Se una qualsiasi delle circostanze di cui sopra si applica a voi, OpenQbit, la società che impiega tutti gli sviluppatori di Mini QRNG, vi venderà una Mini QRNG Title Guarantee. La Title Warranty è un documento legale che afferma che i pretesi autori del Mini QRNG sono i veri autori, e che gli autori hanno il diritto legale di dedicare il Mini QRNG al pubblico dominio, e che OpenQbit si difenderà con forza contro le richieste di licenza. Tutti i proventi della vendita delle garanzie di titolo Mini QRNG sono utilizzati per finanziare il continuo miglioramento e il supporto di Mini QRNG.

Codice Contributo

Per mantenere Mini QRNG completamente gratuito e royalty free, il progetto non accetta patch. Se volete fare una modifica suggerita e includere una patch come prova del concetto, sarebbe fantastico. Tuttavia, non offendetevi se riscriviamo la vostra patch da zero. Il tipo di licenza non commerciale o opensource che la utilizza in questa modalità e alcune simili senza l'acquisto di supporto sia per uso individuale che aziendale, indipendentemente dalle dimensioni dell'azienda, sarà regolato dalle seguenti premesse legali.

Esclusione di garanzia. A meno che non sia richiesto dalla legge applicabile o concordato per iscritto, il Concessore di Licenza fornisce l'Opera (e ciascun Collaboratore fornisce i propri Contributi) "COSÌ COME È", SENZA **GARANZIE O CONDIZIONI DI QUALSIASI TIPO**, sia espresse che implicite, incluse, senza limitazione, eventuali garanzie o condizioni di TITOLO, NON VIOLAZIONE, COMMERCIALIZZABILITÀ O IDONEITÀ PER UN PARTICOLARE SCOPO. L'utente è l'unico responsabile della determinazione del corretto utilizzo o della ridistribuzione dell'Opera e dell'assunzione di eventuali rischi associati all'esercizio delle autorizzazioni previste dalla presente Licenza.

Eventuali perdite finanziarie o di altra natura derivanti dall'uso di questo software saranno a carico della parte interessata. Tutte le controversie legali le parti si sottoporranno ai tribunali solo nella giurisdizione di Città del Messico, paese Messico.

Per il supporto commerciale, l'uso e la concessione di licenze deve essere stabilito un accordo o un contratto tra OpenQbit o la sua società e la parte interessata.

I termini e le condizioni di marketing della distribuzione possono cambiare senza preavviso, si prega di andare al sito ufficiale www.openqbit.com per vedere eventuali modifiche alle clausole di supporto e di licenza non commerciali e commerciali.

Qualsiasi persona, utente, ente pubblico o privato di qualsiasi natura giuridica o di qualsiasi parte del mondo che utilizzi semplicemente il software accetta senza condizioni le clausole stabilite in questo documento e quelle che possono essere modificate in qualsiasi momento nel portale di www.openqbit.com senza preavviso e possono essere applicate a discrezione di OpenQbit in uso non commerciale o commerciale.

Tutte le domande e le informazioni su Mini QRNG devono essere indirizzate alla comunità di App Inventor o alle varie comunità del sistema Blockly così come sono: AppBuilder, Trunkable, ecc. e/o all'indirizzo e-mail opensource@openqbit.com per la richiesta di domande possono richiedere da 3 a 5 giorni lavorativi per ottenere una risposta.

Supporto con uso commerciale.

support@openqbit.com

Vendite per uso commerciale.

sales@openqbit.com

Informazioni legali e domande o dubbi sulla licenza

legal@openqbit.com

