Tutorial de Instalación de soporte para Qt-Creator y OpenCv con Raspberry PI.

**Temario**

:

* Configuración de inicio
  + Configuración de teclado
* Instalación de VNC
  + Raspberry
  + PC
* Control Remoto
  + Agregar dispositivo
  + Visualizar
* Configuración Remota
  + Habilitar descarga de archivos para desarrollo
  + Creación de directorio para QT en RPI
  + Creación de directorio para compilación cruzada en PC
  + Sincronización
  + Configuración de Raspbian en la PC

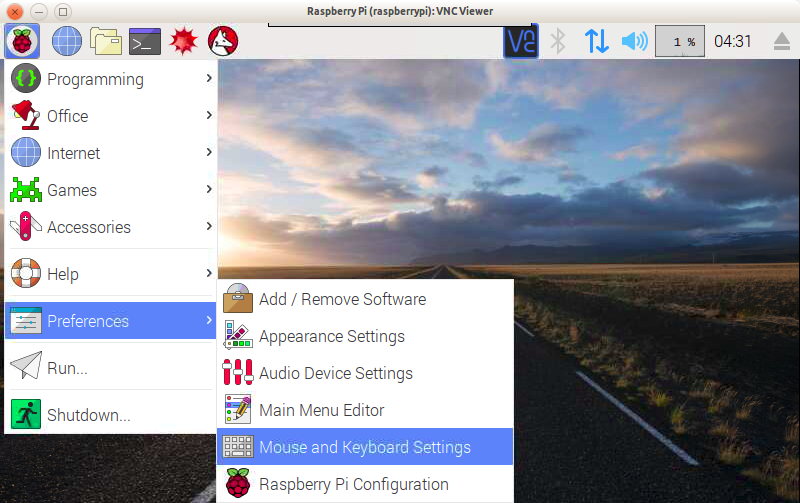
* Librerías EGL/GLES

* Configuración de QT para proyectos multiplataforma

**Configuración de inicio**

**Teclado**

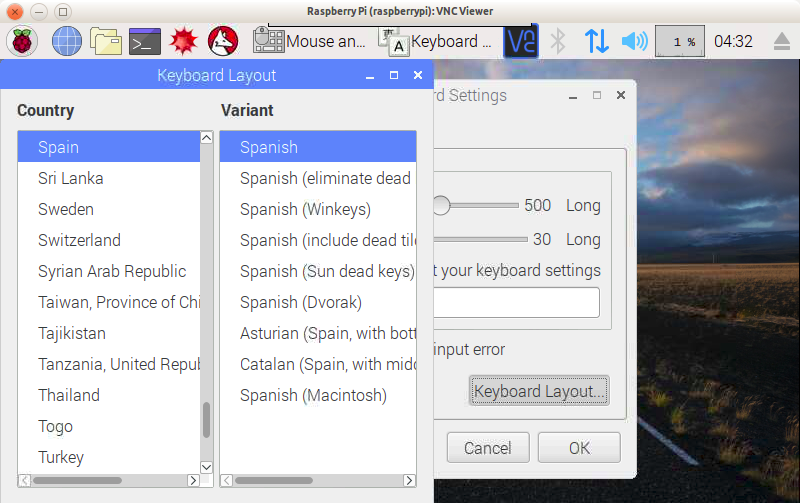
Como primer paso configuramos el idioma del teclado. Para hacerlo, nos dirigimos al menú Raspberry > Preferencias > Configuración de mouse y teclado.



En la ventana que surge, cambiamos a la pestaña del teclado y hacemos clic en diseño de teclado.



Dentro, seleccionamos España para encontrar Español. Seguido a eso, damos clic 2 veces en OK para terminar la configuración del teclado.



**Actualización**

Ya con el teclado configurado, escribimos en la terminal los siguientes comando

1. sudo apt-get update
2. sudo apt-get upgrade
3. sudo rpi-update
4. Con esto queda terminada la configuración inicial.

**Instalación de VNC**

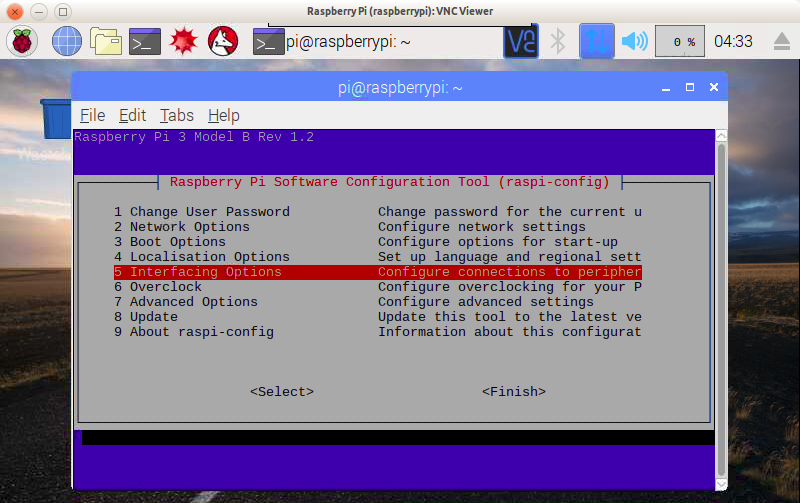
El software de VNC permite controlar en tiempo real, la Raspberry desde la PC. Por este motivo es de gran ayuda para agilizar el uso de la misma.

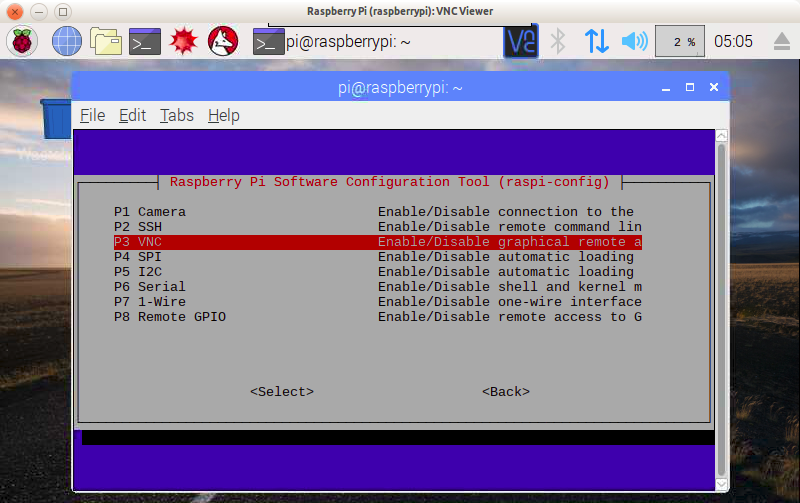
**VNC Raspberry PI**

En la terminal de raspberry ejecutamos los comandos:

1. sudo apt-get install realvnc-vnc-server
2. sudo apt-get install realvnc-vnc-viewer
3. sudo raspi-config

Navegamos con las flechas y enter hast Interfacing Options > VNC > Yes





**VNC PC**

Entrando al sitio de descarga oficial

1. <https://www.realvnc.com/es/connect/download/vnc/>

Buscamos la versión para Linux, la descargamos y después de proporcionar los permisos para ejecutar el archivo como programa ejecutable, lo instalamos en nuestra PC. Una vez instalado podemos sincronizar ambos dispositivos.

El siguiente paso es obtener la dirección IP de la Raspberry Pi. Se puede obtener de múltiples maneras: desde el explorador de Raspberry buscar sitios “¿cual es mi IP?”, escribiendo el comando ifconfig o por medio de zenmap. Para usar zenmap solo hace falta correr el comando:

1. sudo apt-get install zenmap

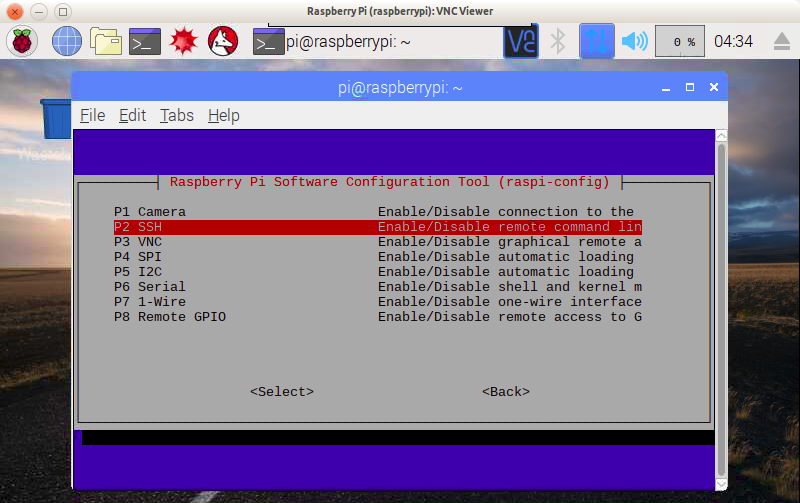
Despues corremos el siguiente comando, reemplazando los numeros en 192.168.100.0 con nuestra dirección IP.

1. sudo nmap -sP 192.168.100.0/24 | awk '/^Nmap/{ip=$NF}/B8:27:EB/{print ip}'

Seguido a esto se habilita el SSH y se actualiza Raspbian. En la terminal de comandos escribimos:

1. sudo raspi-config

Nos dirigimos en la ventana emergente a Interfacing Options > Habilitar SSH > Yes.

Al final solo hace falta actualizar Raspbian con los comandos:

1. sudo rpi-update
2. reboot

Con eso se reiniciará la Raspberry.

**Control Remoto**

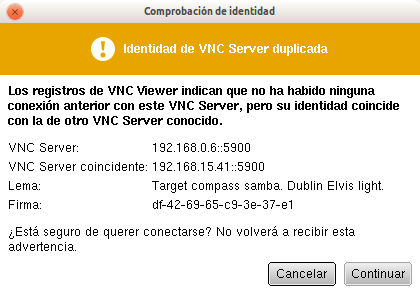
Agregar el dispositivo.

En la PC, una vez iniciada la Raspberry abriremos el VNC viewer en la PC. Seleccionamos Archivo -> Nueva conexión en el menú superior.

Escribimos la dirección IP y le ponemos un nombre a la conexión con la Raspberry.



Al ingresar por primera vez a la conexión, dependiendo de si el servidor se encuentra también registrado a la cuenta salta la siguiente ventana a la cual solo resta darle continuar. Si la conexión se realiza sin cuenta de por medio es la siguiente:

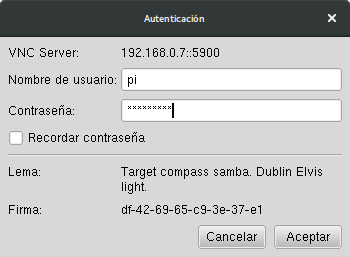


Con cuenta

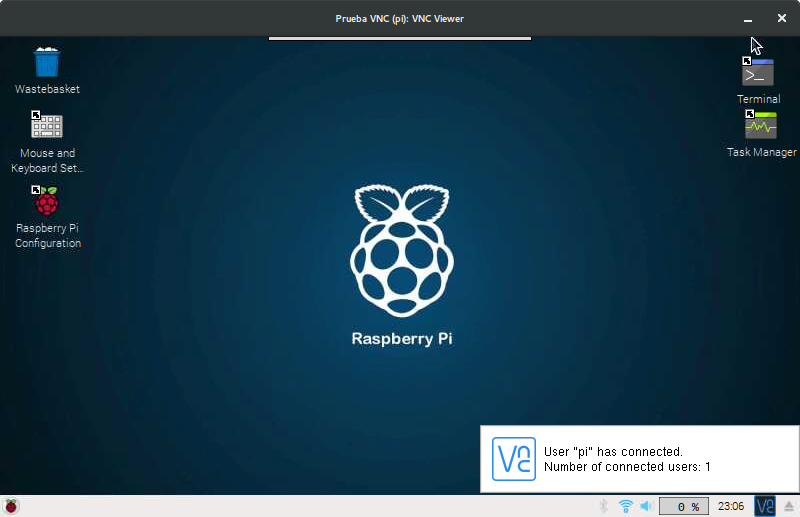


**Visualizar**

En la siguiente ventana escribimos los datos de inicio de sesión default de Raspberry. La contraseña debe ser “raspberry” sin comillas.



Con ello ya podremos controlar completamente la Raspberry de forma remota desde la PC.



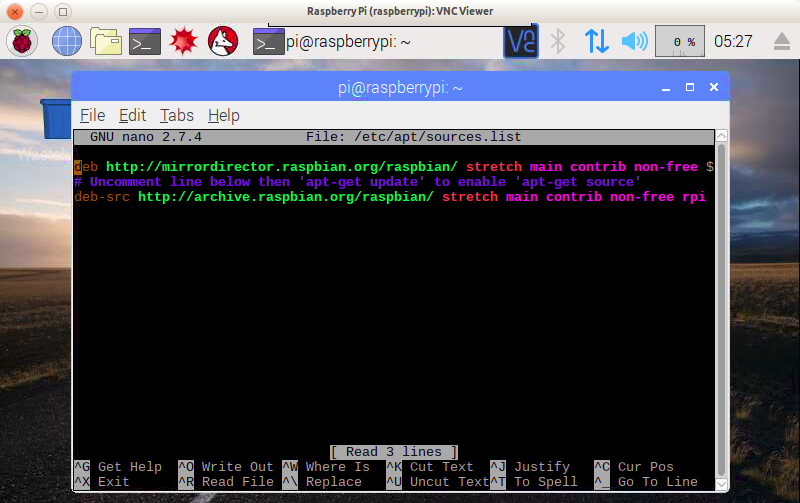
**Configuración con VNC**

**Raspberry PI**

Primero es necesario habilitar descarga de archivos para desarrollo, para ello escribimos en la terminal de la Raspberry el siguiente comando:

1. sudo nano /etc/apt/sources.list

Se abrirá un documento al que le vamos a quitar el carácter # a la línea #deb, guardamos el archivo con ctrl+X, escribimos yes y presionamos enter.



En la misma terminal ejecutamos uno a uno para instalar las librerías necesarias:

1. sudo apt-get update
2. sudo apt-get build-dep qt4-x11
3. sudo apt-get build-dep libqt5gui5
4. sudo apt-get install libudev-dev libinput-dev libts-dev libxcb-xinerama0-dev libxcb-xinerama0

Ahora, creamos el directorio para QT en la Raspberry

1. sudo mkdir /usr/local/qt5pi
2. sudo chown pi:pi /usr/local/qt5pi

Por ahora la Raspberry queda en espera.

**PC**

En la computadora abrimos la terminal y ejecutamos:

1. mkdir raspi
2. cd raspi
3. git clone <https://github.com/raspberrypi/tools>

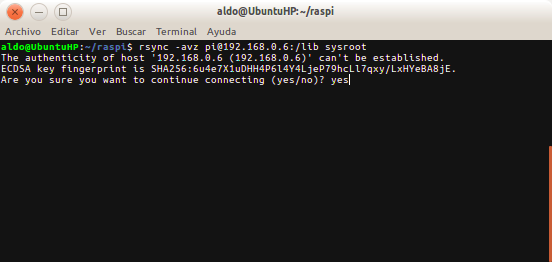
Crear árbol de directorios donde se guardaran la instalación de raspbian en la computadora.

1. mkdir sysroot sysroot/usr sysroot/opt

**Sincronización**

Para sincronizar las bibliotecas de raspbian desde la raspberry hacia la computadora para guardar dichos archivos en las carpetas que se crearon en el paso anterior ejecutaremos comandos donde es necesario SUSTITUIR raspberrypi.local por la IP de la Raspberry.

* 1. rsync -avz pi@raspberrypi.local:/lib sysroot
  2. rsync -avz pi@raspberrypi.local:/usr/include sysroot/usr
  3. rsync -avz pi@raspberrypi.local:/usr/lib sysroot/usr
  4. rsync -avz pi@raspberrypi.local:/opt/vc sysroot/opt



Ajustar los enlaces relativos es el siguiente paso:

1. wget <https://raw.githubusercontent.com/riscv/riscv-poky/priv-1.10/scripts/sysroot-relativelinks.py>
2. chmod +x sysroot-relativelinks.py
3. ./sysroot-relativelinks.py sysroot

El siguiente paso es **configurar la instalación de RASPBIAN en la PC**

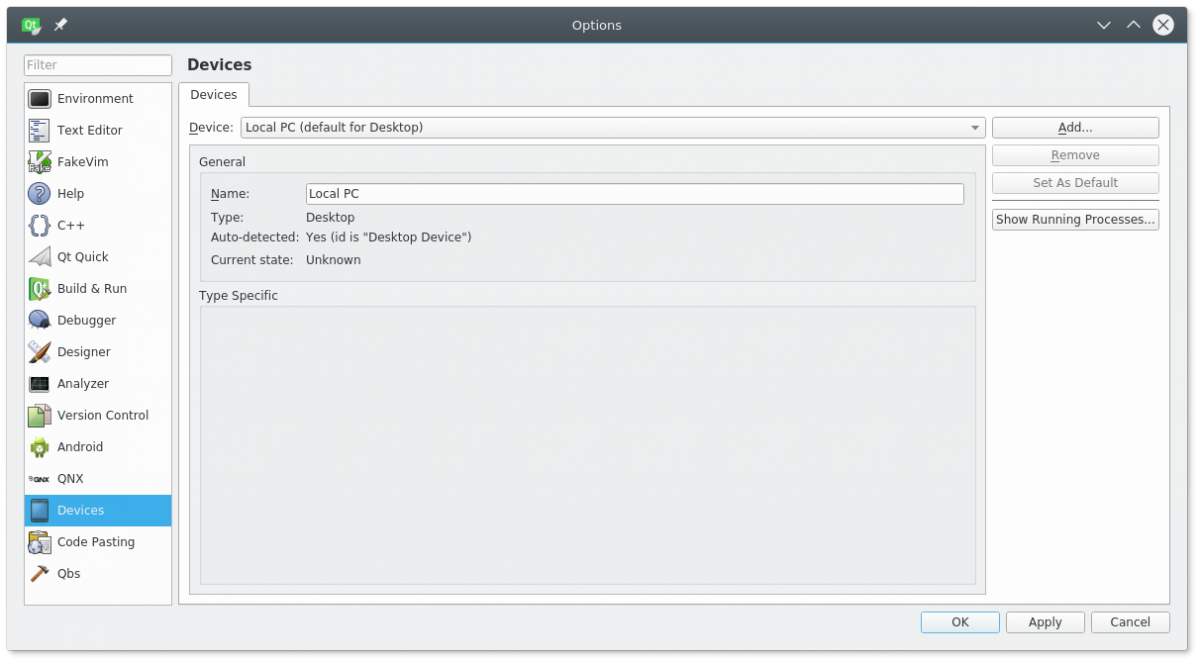
* 1. git clone git://code.qt.io/qt/qtbase.git -b 5.9
  2. cd qtbase
  3. ./configure -release -opengl es2 -device linux-rasp-pi3-g++ -device-option CROSS\_COMPILE=~/raspi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian-x64/bin/arm-linux-gnueabihf- -no-use-gold-linker -sysroot ~/raspi/sysroot -opensource -confirm-license -make libs -prefix /usr/local/qt5pi -extprefix ~/raspi/qt5pi -hostprefix ~/raspi/qt5 -v
  4. make
  5. make install
     1. Si falla, ejecutar:
     2. git clean -dfx

1. Sincronizar los cambios con la raspberry
   1. rsync -avz qt5pi pi@raspberrypi.local:/usr/local

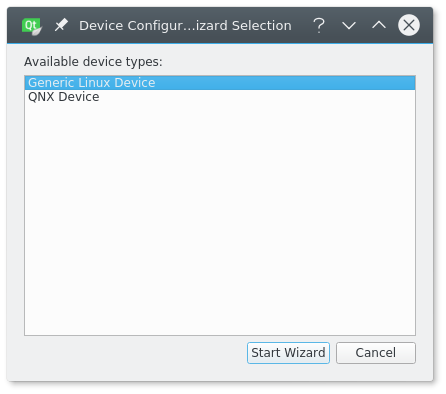
En la Raspberry,

1. **Agregar las librerías EGL/GLES**
   1. sudo mv /usr/lib/arm-linux-gnueabihf/libEGL.so.1.0.0 /usr/lib/arm-linux-gnueabihf/libEGL.so.1.0.0\_backup
   2. sudo mv /usr/lib/arm-linux-gnueabihf/libGLESv2.so.2.0.0 /usr/lib/arm-linux-gnueabihf/libGLESv2.so.2.0.0\_backup
   3. sudo ln -s /opt/vc/lib/libEGL.so /usr/lib/arm-linux-gnueabihf/libEGL.so.1.0.0
   4. sudo ln -s /opt/vc/lib/libGLESv2.so /usr/lib/arm-linux-gnueabihf/libGLESv2.so.2.0.0
   5. sudo ln -s /opt/vc/lib/libEGL.so /opt/vc/lib/libEGL.so.1
   6. sudo ln -s /opt/vc/lib/libGLESv2.so /opt/vc/lib/libGLESv2.so.2

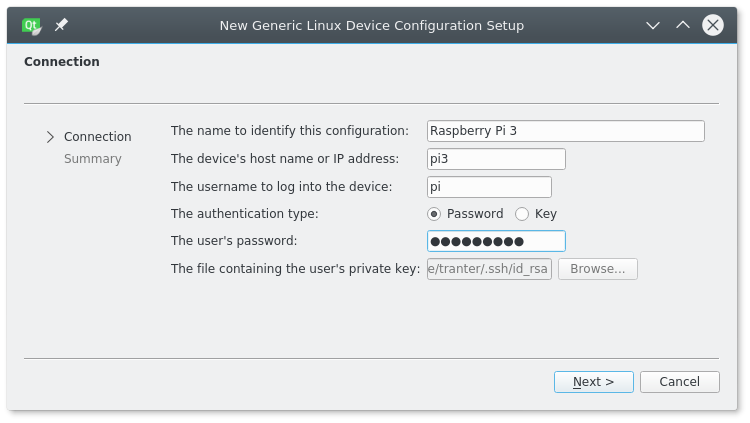
**Configuración de proyectos multiplataforma para QT-Creator**

Primero tenemos que agregar un dispositivo. Inicie Qt Creator y seleccione Herramientas / Opciones ... y luego haga clic en la pestaña Dispositivos cerca de la esquina inferior izquierda del cuadro de diálogo.  


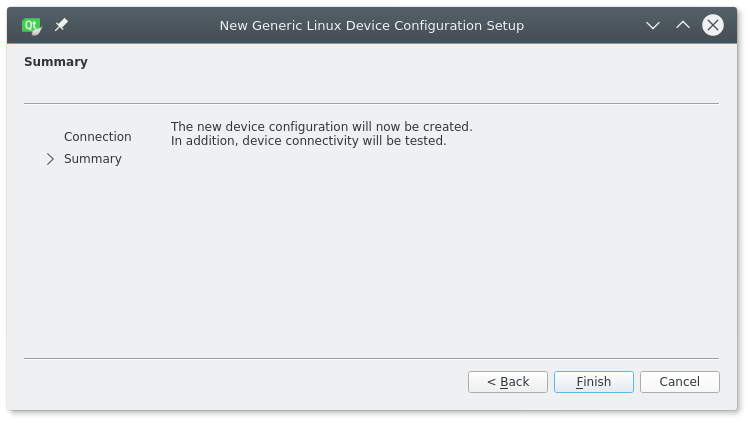
Haga clic en Agregar ... para abrir el Asistente de configuración del dispositivo. Seleccione "Dispositivo Linux genérico" y luego, inicie el asistente.



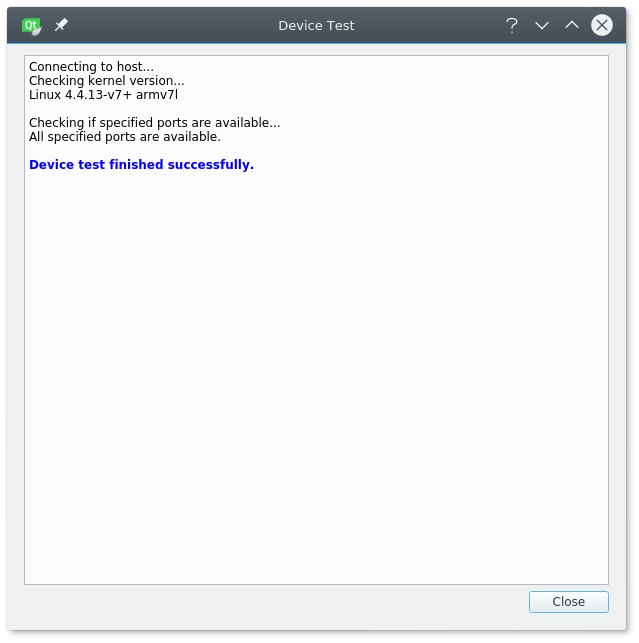
En la pantalla de Conexión, ingrese los parámetros apropiados. La captura de pantalla a continuación muestra los valores típicos. Tendrá que ingresar el nombre de host o la dirección IP apropiados para Raspberry Pi en su red.



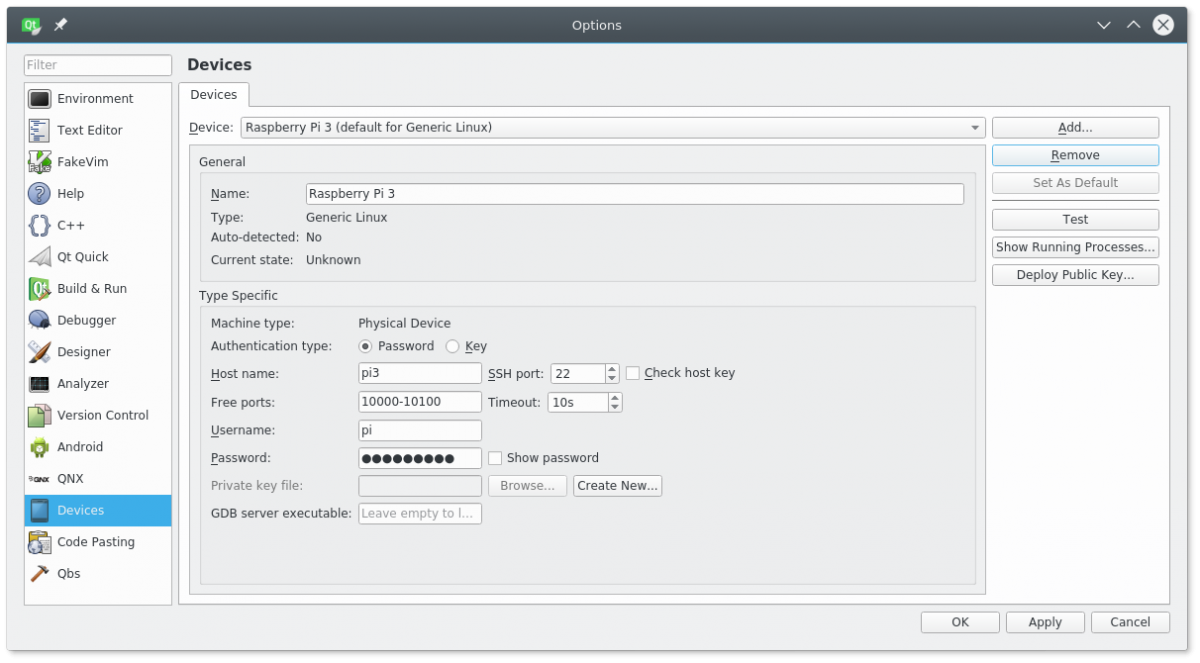
Haga clic en Siguiente> y luego en Finalizar.



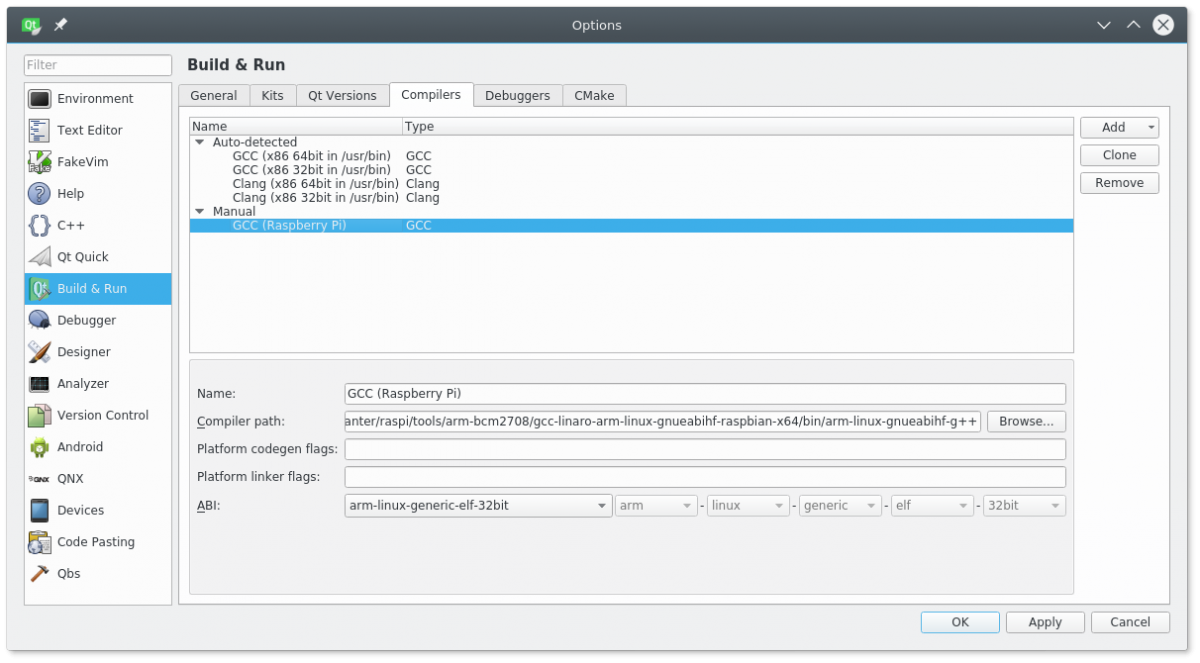
La siguiente pantalla de Prueba del dispositivo debe confirmar que Qt Creator puede comunicarse e iniciar sesión en Raspberry Pi. Si no, regresa y corrige la configuración.



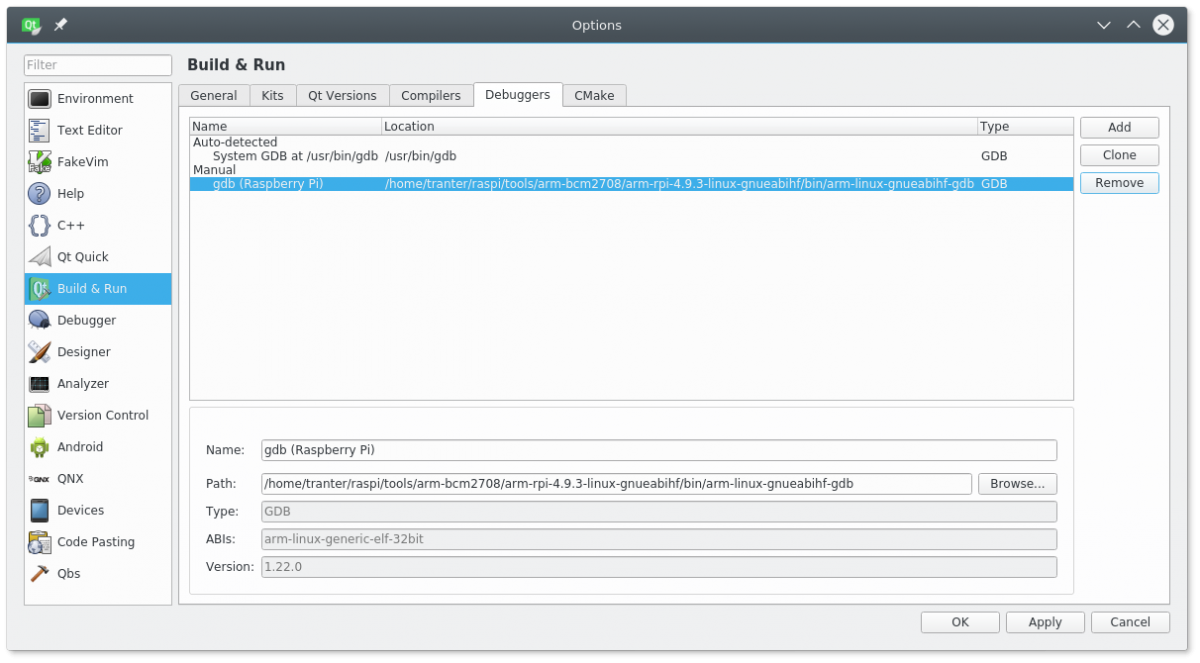
Después de hacer clic en Cerrar, debería ver la placa aparecer como un dispositivo, similar a la captura de pantalla a continuación:



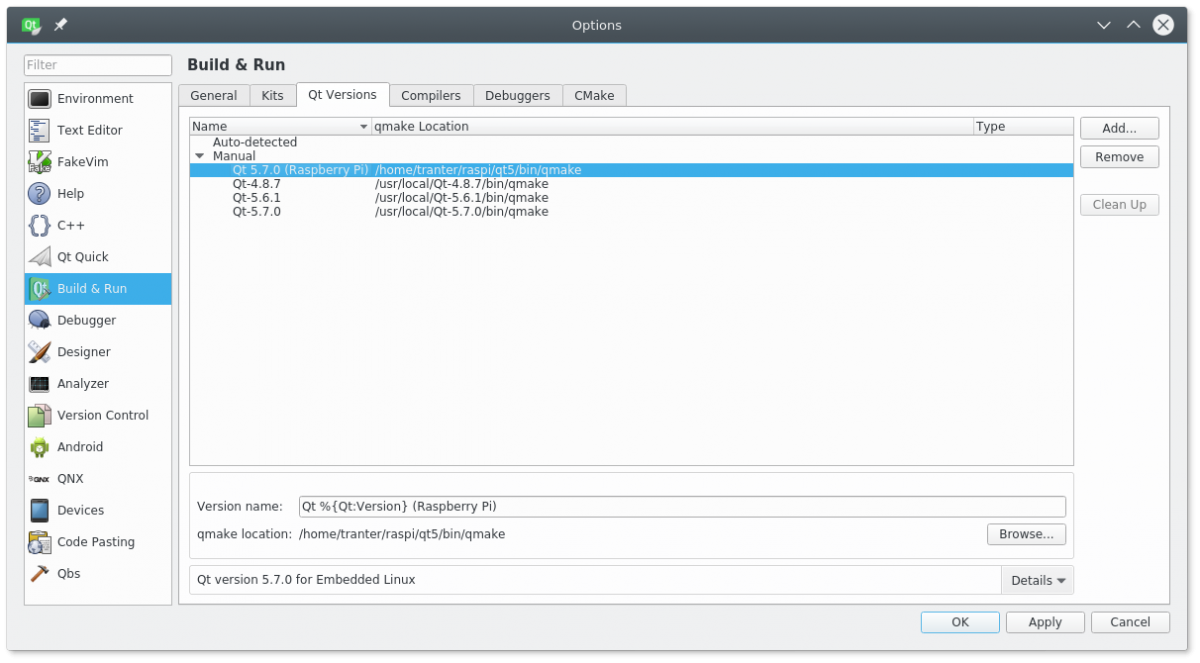
A continuación, agregue el compilador cruzado. Desde la pantalla Herramientas / Opciones ..., haga clic en la pestaña izquierda Build & Run y ​​seleccione Compilers. Haga clic en Agregar / GCC e ingrese un nuevo compilador con el nombre "GCC (Raspberry Pi)" y la ruta del compilador establecida en ~ / raspi / tools / arm-bcm2708 / gcc-linaro-arm-linux-gnueabihf-raspbian / bin / arm-linux-gnueabihf-g ++ (puede hacer clic en Examinar y navegar hasta el archivo). Debería ver el nuevo compilador en la lista, como en la siguiente captura de pantalla.



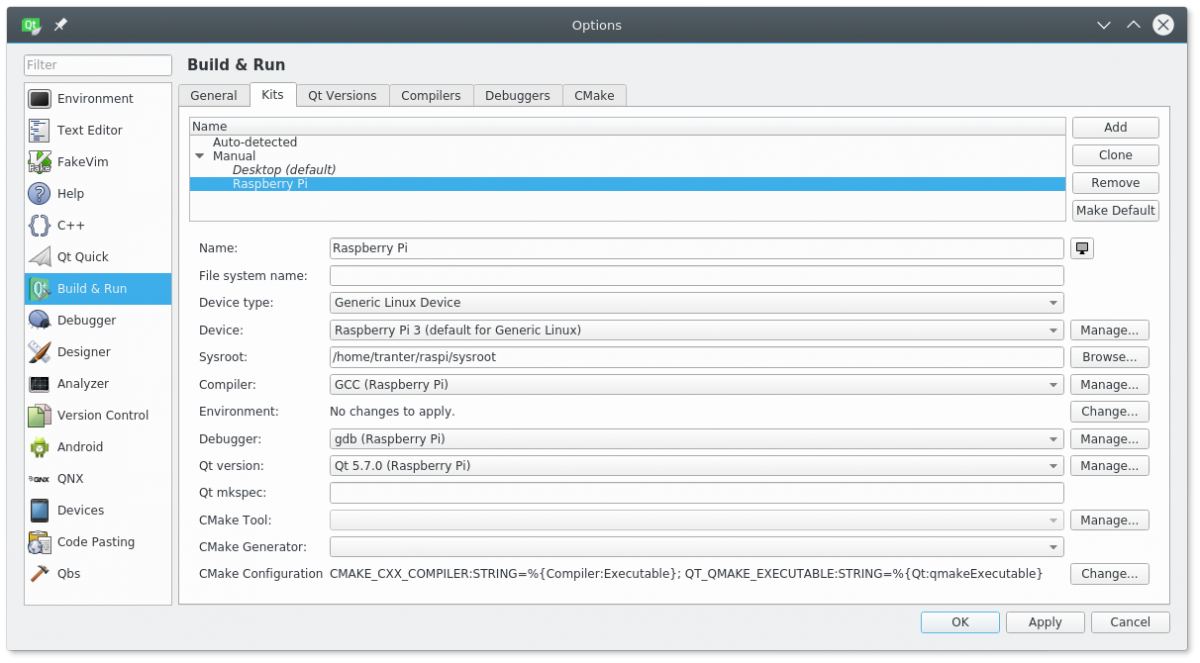
A continuación, seleccione la pestaña Depuradores y agregue un nuevo depurador. Haga clic en Agregar, ingrese un nombre adecuado, como "gdb (Raspberry Pi)", y la ruta ~ / raspi / tools / arm-bcm2708 / arm-rpi-4.9.3-linux-gnueabihf / bin / arm-linux-gnueabihf -gdb. La entrada del depurador debe ser similar a esto:



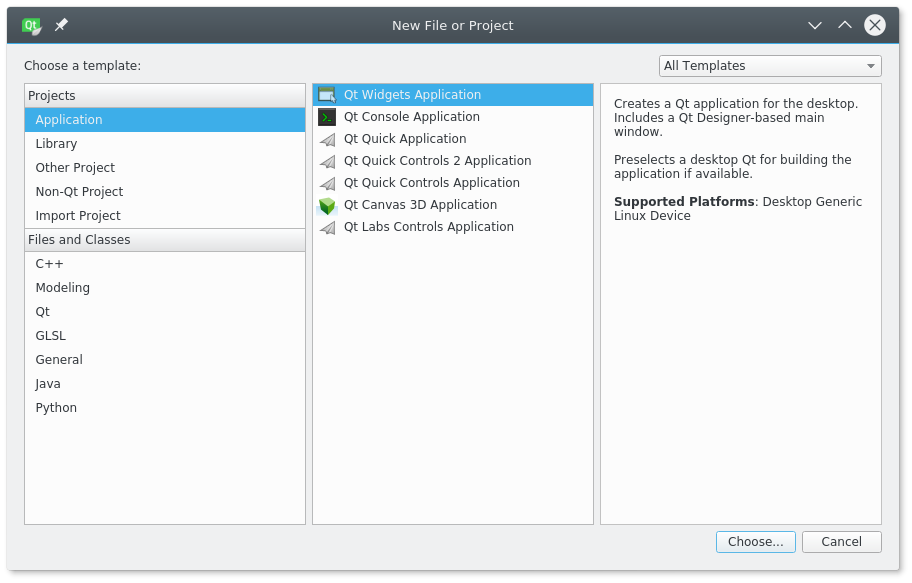
Ahora podemos ir a la pestaña Versiones de Qt de la pantalla Opciones. Compruebe si aparece una entrada con la ubicación qmake establecida en ~ / raspi / qt5 / bin / qmake. Si no, haga clic en Agregar ..., navegue hasta qhacer, y asígnele un nombre de versión adecuado. Usé "Qt 5.7.0 (Raspberry Pi)" como se muestra a continuación.



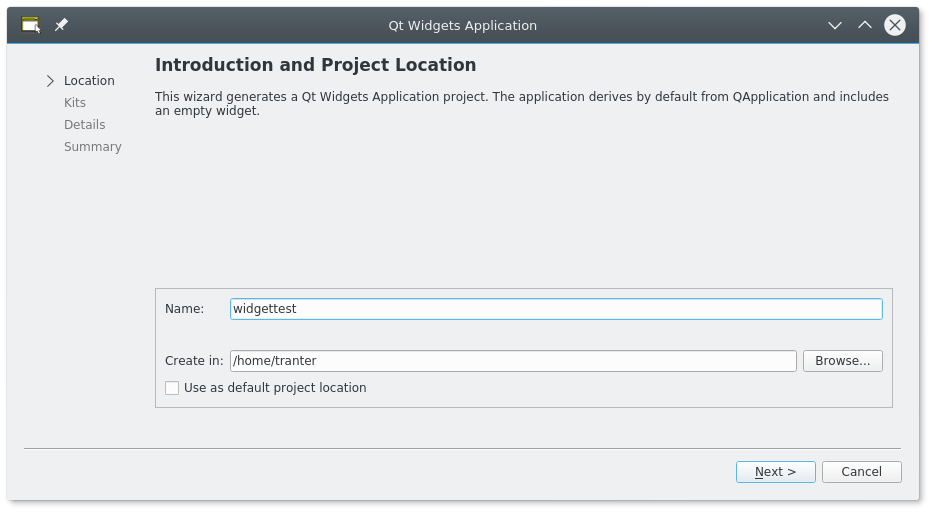
Finalmente, estamos listos para combinar estas herramientas en lo que Qt Creator llama un kit. Haga clic en la pestaña Kits de la pantalla Opciones, seleccione Agregar y cree un kit con un nombre adecuado, y el dispositivo, el compilador, el depurador y la versión Qt que acabamos de crear. El valor de sysroot debe ser ~ / raspi / sysroot y los otros campos pueden dejarse vacíos. El kit que creé se muestra a continuación.



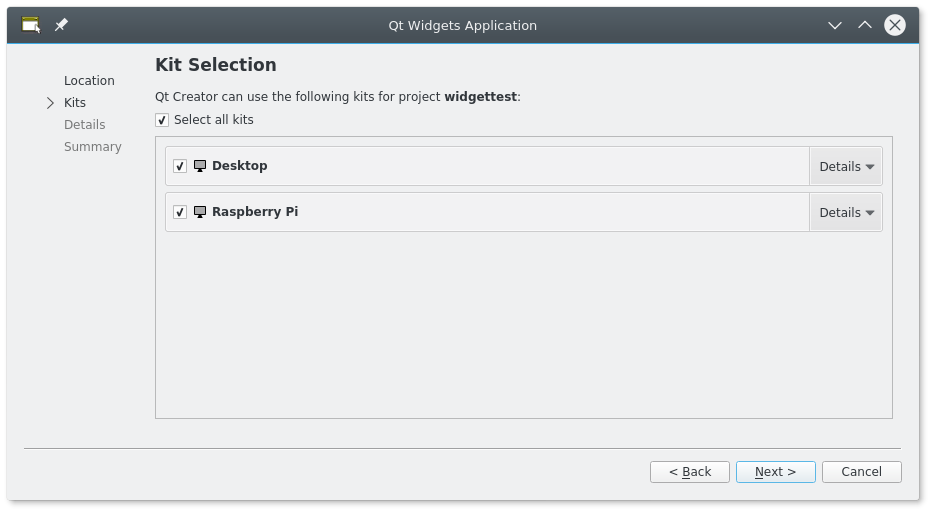
Seleccionamos Archivo / Nuevo archivo o Proyecto ... y luego seleccionamos la plantilla para la aplicación y la aplicación Qt Widgets.



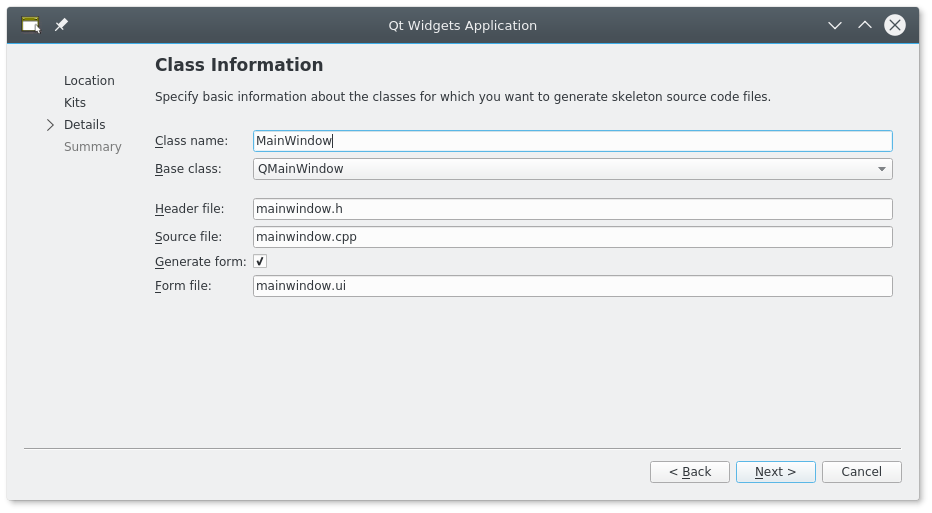
Haga clic en Elegir ... e ingrese el nombre de un proyecto o use el predeterminado.



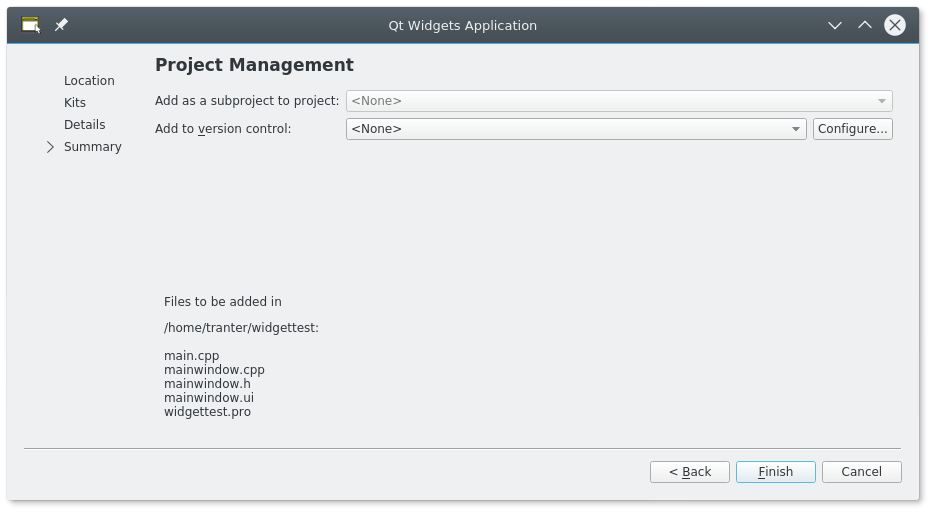
Luego haga clic en Siguiente> y seleccione los kits deseados. Seleccionemos tanto el Escritorio (o el kit que haya configurado previamente para el desarrollo nativo) como Raspberry Pi.



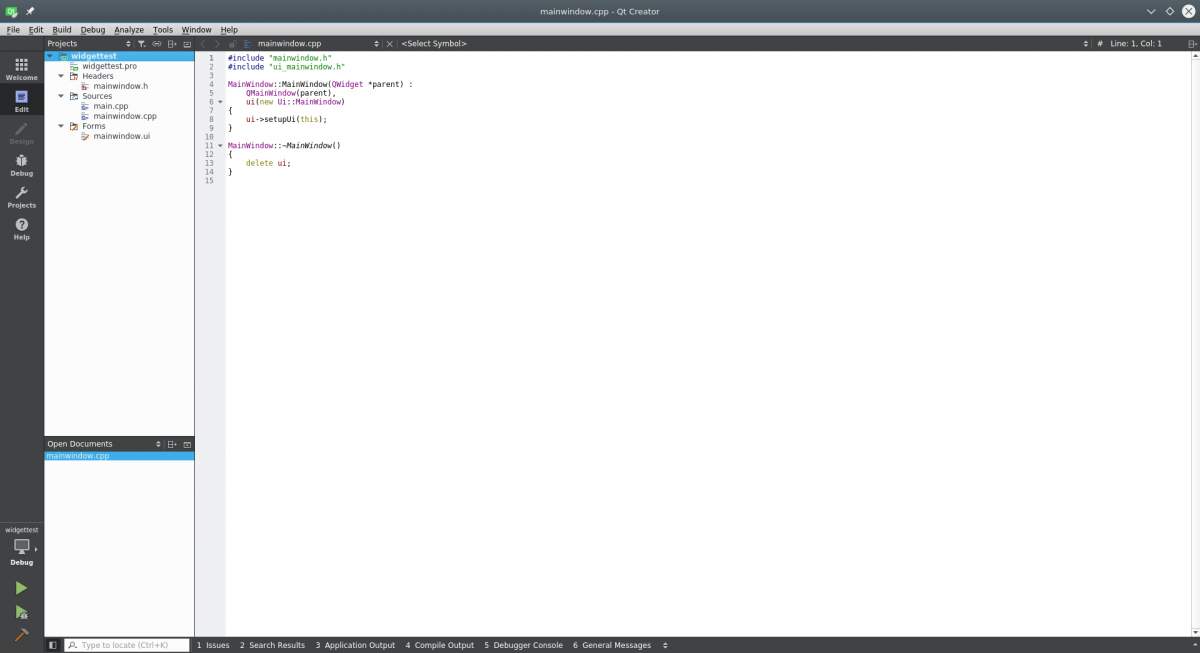
Haga clic en Siguiente> y deje la configuración de Información de clase en sus valores predeterminados.



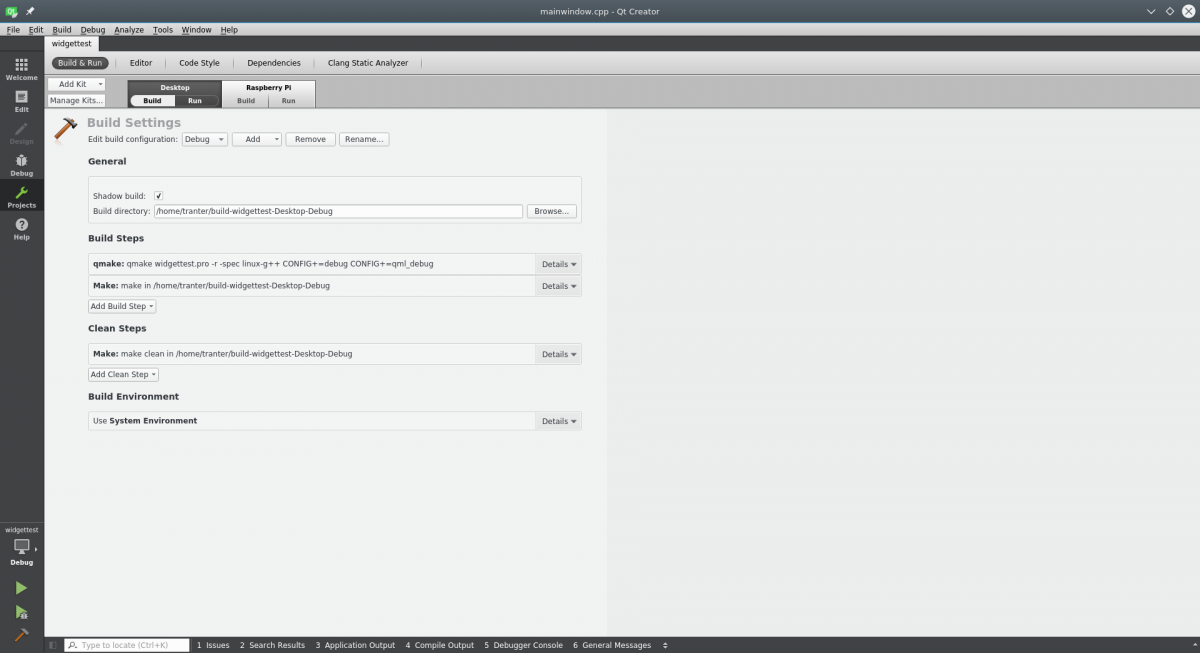
Pulse Siguiente> una vez más, use la configuración predeterminada y seleccione Finalizar.



Ahora tenemos un proyecto para probar. Le sugiero que primero lo cree y lo ejecute localmente en su escritorio, que debería ser el predeterminado. Aquí hay algo del código:



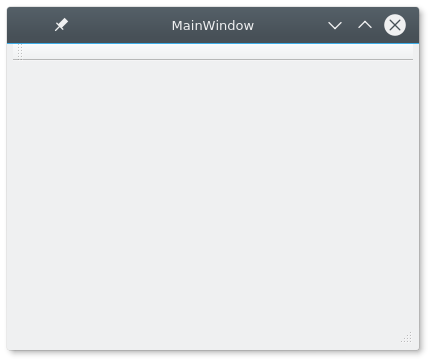
La configuración predeterminada del proyecto:



Compilando:



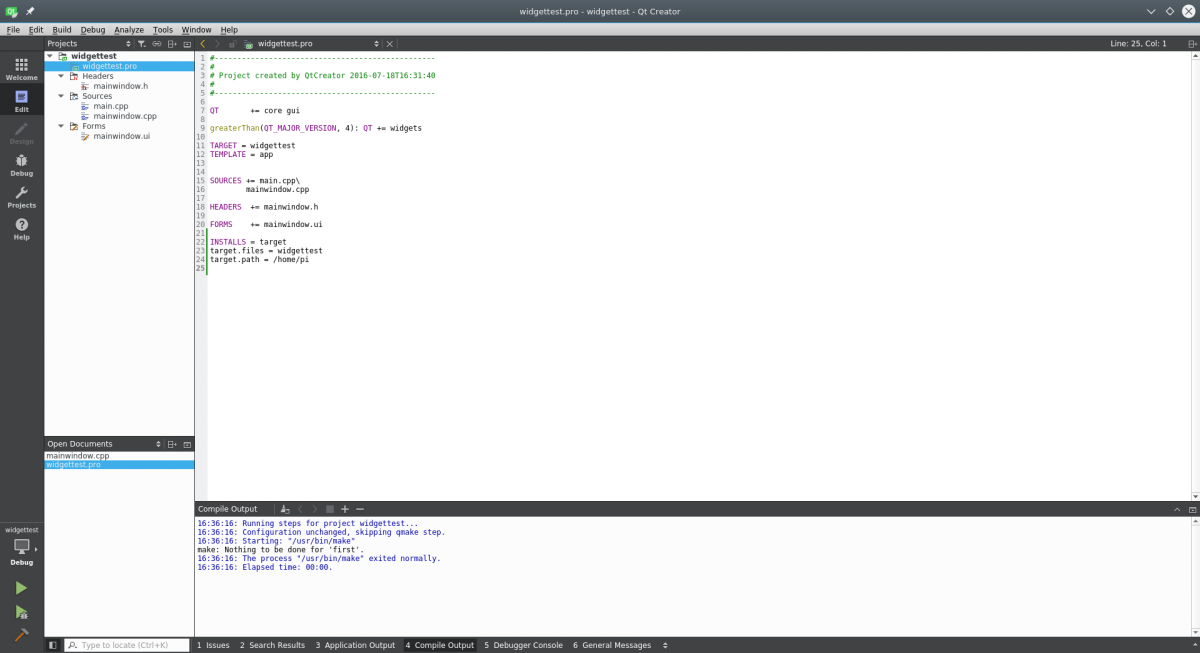
al final corriendo en el escritorio:



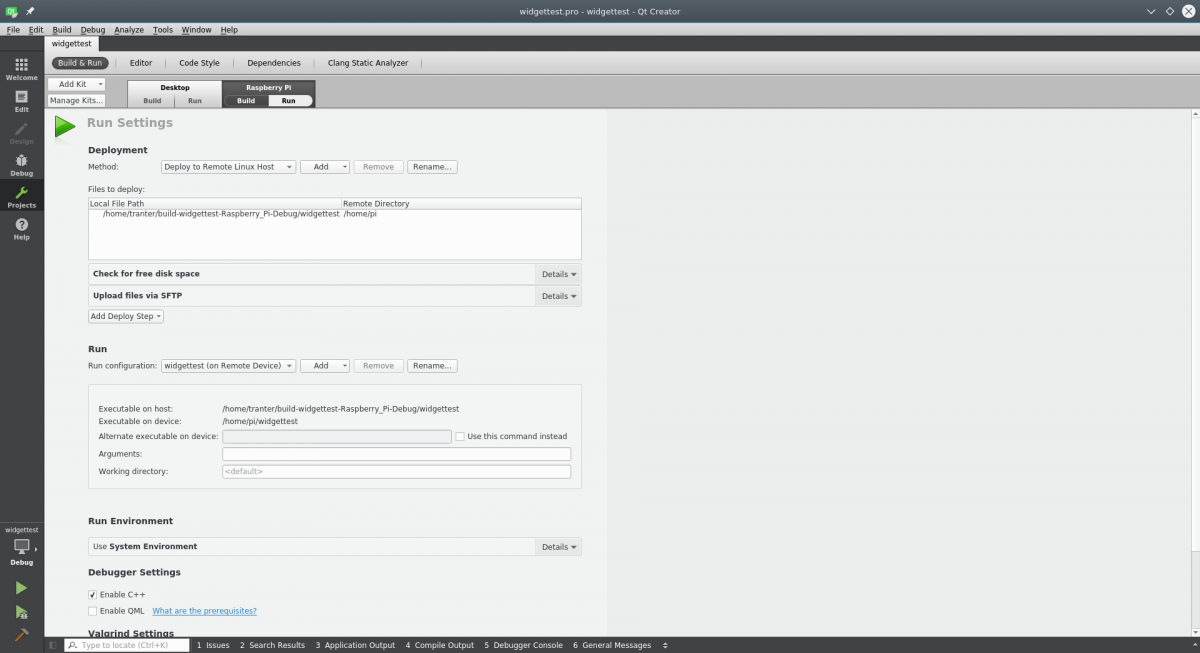
Si eso funciona, haga clic en el kit "Raspberry Pi" en la pestaña Proyectos.

Si eligió el mismo tipo de proyecto que yo, deberá agregar algunas líneas al archivo del proyecto qmake para que qmake sepa qué archivos deben implementarse en el sistema de destino para ejecutar la aplicación. Una manera fácil de hacerlo es editar (desde Qt Creator) el archivo del proyecto y agregar estas líneas en la parte inferior:

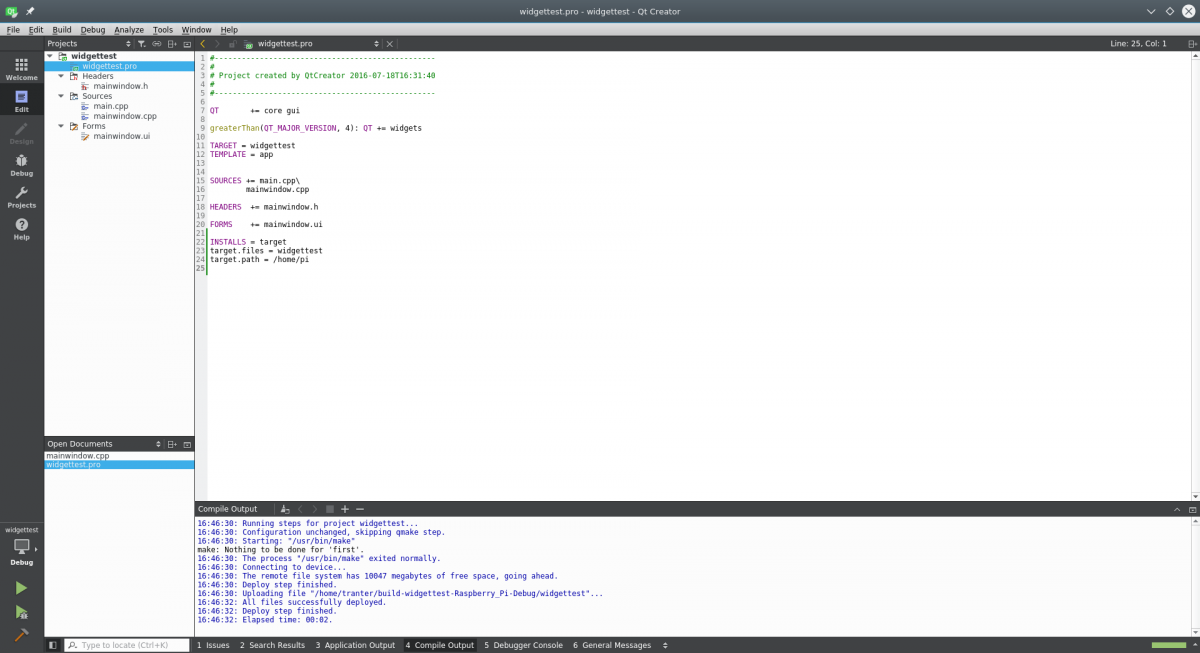
INSTALLS = target  
target.files = widgettest  
target.path = /home/pi



Si vuelve al panel Proyectos y hace clic en el botón Ejecutar en "Raspberry Pi", debería ver un archivo ahora en "Archivos para implementar:".



Si hace clic en el botón de compilación, el proyecto debe compilarse exitosamente con el kit Raspberry Pi. Luego, al hacer clic en el botón Ejecutar, se implementará y ejecutará la aplicación en Raspberry Pi. No se verá muy impresionante, solo una ventana de pantalla completa, ya que no pusimos ningún widgets en la aplicación.



Es posible que desee volver y agregar algunos widgets, como botones, a la aplicación e intente ejecutarlo de nuevo. Puede hacerlo desde Qt Designer, disponible en la pestaña Diseño.

Te sugiero que pruebes aplicaciones adicionales, ya sea que ya tengas o algunos de los ejemplos y tutoriales de Qt disponibles en la pestaña de Bienvenida en Qt Creator. Probablemente quieras probar una aplicación QML, posiblemente usando Qt Quick Controls

Todo esto debería funcionar, aunque para la mayoría de ellos necesitarás agregar algunas reglas de implementación como hicimos con el primer ejemplo.

../qt-everywhere-opensource-src-5.7.1/configure -opengl es2 -device linux-rpi3-g++ -device-option CROSS\_COMPILE=arm-linux-gnueabihf- -sysroot /opt/qt5pi/sysroot -prefix /usr/local/qt5pi -opensource -confirm-license -skip qtwebengine -skip qtscript -nomake examples -make libs -v

**Instalar OpenCV.**

* 1. <https://www.pyimagesearch.com/2016/04/18/install-guide-raspberry-pi-3-raspbian-jessie-opencv-3/>

