Ubuntu 18.04 Bionic-Builder

The Bionic-Builder is a All-In-One build script for Hikey970. This script will interactively put together the Ubuntu 18.04 Bionic using Debootstrap. The configuration is a part of the script so there is no need to manually edit any files to get things working. The build script can perform the following operations.

A) Downloads the Base packages needed for running Ubuntu 18.04

B) Downloads and install the ARM64 Tool-Chain needed for building the kernel.

The tool-chain used can be found here....

https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-a/downloads

gcc-arm-8.3-2019.03-x86_64-aarch64-linux-gnu

C) Downloads the Kernel Source needed for building the kernel.

This Kernel Source is configured specifically for Debain or Ubuntu Support.
The kernel source is located at...
https://github.com/Bigcountry907/linux/tree/hikey970-v4.9-Debain-Working
Be sure to use the branch hikey970-v4.9-Debain-Working if you clone the kernel not using the Bionic-Builder.

D) Compiles the kernel and installs the Kernel the Device Tree and the Modules.

Everything is installed into the Ubuntu Bionic Image.
This eliminates the need to manually install these things.

E) Interactively configures the login and wireless connection

Enter your desired username and password when prompted.
Bionic now uses netplan as a network manager.
The 01-dhcp.yaml configuration file is touchy. The spacing has to be exact.
The LAN will work on DCHP with no modification.
The WI-FI will work provided you enter the access-point name and password for your router.
This way after the first boot you will be connected to the network.

F) Includes a First-Boot Initialization script

The initialization script will finish up the Ubuntu 18 Install.
It automatically updates and upgrades the system.
It installs tasksel and runs tasksel install standard to add the standard packages to the system.
The script will automatically install the XFCE4 / Xubuntu Desktop environment if you choose.
If you choose not to install a desktop then Ubuntu will run as Ubuntu Server.

The video is working properly so upon booting provided you are using a usb keyboard and usb Mouse you can login and use the Terminal Console without UART.
The typical UART Console is used also by connecting the usb-c on the side of the board (NOT NEXT TO HDMI) !!!
Use putty in windows or ssh in linux to access the serial console. You will only see the Grub Boot Menu over the serial console.

G) The Menu allows for modification of the Rootfs and the Kernel without rebuilding all

I have set things up so that you can build each part separately. You only need to run option 1 one time.
You can edit the files in ~/Bionic-Builder/Ubuntu-SRC/Build if needed.
You can change kernel configurations or update the kernel source in ~/Bionic-Builder/Kernel-SRC/linux
Option #4 can be used after changes are made to generate a new flash-able image.

More features can be added upon request.

Instructions for using the Bionic-Builder

Note: This build script is made for running on Ubuntu or a Debain System.

The kernel is Cross-Compiled so you can not run the build script on the Hikey970 you must use a server or local Ubuntu / Debain system.

1) Install the packages require for the build.

A)

```
sudo apt-get install -y ccache python-pip build-essential kernel-package fakeroot libncurses5-dev libssl-dev gcc  git-core gnupg binfmt-support qemu qemu-user-static debootstrap simg2img
```

B) Install Git **if git is not installed yet.**

1)

```
cd ~/
sudo mkdir ~/bin
sudo nano .bashrc
```

2) {at the end of bashrc paste the below path.}

```
export PATH=~/bin:$PATH
```

Then Run

```
source .bashrc
```

3) DOWNLOAD THE REPO TOOL

```
curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin/repo
chmod a+x ~/bin/repo
```

4) SETUP YOUR GITHUB ACCOUNT

```
git config --global user.name "BigCountry907"
git config --global user.email "richjame907@gmail.com"
```

C) Clone repo and Start the Build Script

1.) `cd ~/`

2.) `git clone https://github.com/Bigcountry907/Bionic-Builder.git`

3.) `cd ~/Bionic-Builder`

4.) `sudo -s`

5.) `./BB.sh`

Welcome to the Bionic-Builder Main Menu

(1) *CREATE MINIMAL BASE ROOT FILESYSTEM*

(2) BUILD KERNEL LINUX v4.9.78

(3) COPY KERNEL & DEVICE TREE / INSTALL KERNEL MODULES in ROOTFS

(4) GENERATE FLASHABLE AND COMPRESSED IMAGES

(5) UPDATE BOTH GRUB.CFG FILES (BOARD AND BOOT IMAGE)

THE BIONIC-BUILDER MENU OPTIONS

(1) CREATE MINIMAL BASE ROOT FILESYSTEM

TO BUILD THE KERNEL ONLY SKIP TO OPTION 2. YOU DON'T HAVE TO RUN OPTION 1!

A) *Option number one uses debootstrap to download the base root filesystem. The rootfs will be located at ~/Bionic-Builder/Ubuntu-SRC/build/rootfs.*

When you choose option 1 the base rootfs is created, during the creation you will be prompted for the USERNAME and PASSWORD to log in to the Hikey 970.

I have added 3 mirrors where the packages can be downloaded from. You will be prompted to select a mirror. During the selection you can choose to input your own Mirror if you know of one.

You will also be prompted for WIFI Configuration. Enter your access point name and password for your wireless network. WIFI will then work automatically on first boot.

The default language will be set to English. After the completion of option 1 the rootfs is configured and ready for booting. At this point you never need to run option 1 again unless you want to start a complete NEW Build.

B) *You can chroot ~/Bionic-Builder/Ubuntu-SRC/build/rootfs and that will switch the root of your running system to the ~/Bionic-Builder/Ubuntu-SRC/build/rootfs.*

After running chroot you can do things like add users and install packages. Remember that the more you add in this stage the larger the system.img will be. After you are done making changes type exit to get out of the chroot.

C) *Before you can create a sparse image to flash to hikey 970 you need to build the kernel, device tree, and install the kernel modules. Options 2 and 3 will do this automatically.*

If you have a KERNEL build already that you would like to use then you can manually copy the Image-hikey970-v4.9.gz and kirin970-hikey970.dtb to ~/Bionic-Builder/Ubuntu-SRC/build/rootfs/boot/
Copy your kernel modules to: ~/Bionic-Builder/Ubuntu-SRC/build/rootfs/lib/modules/

D) *After option 1 is complete run option 2 or manually install the kernel. Once the kernel and Modules are installed you can run option 4 to generate a sparse flash-able system image.*
ALL COMPLETED FILES WILL BE FOUND IN ~/Bionic-Builder/Install/

BUILDING HIKEY 970 DEBAIN / UBUNTU KERNEL v4.9.78

(2) BUILD KERNEL LINUX v4.9.78

A) *Option number two will automatically download the ARM64 tool-chain for cross-compile. The kernel source will be downloaded as well. It will only download one time. After the tool-chain and kernel source are downloaded the KERNEL-BUILD Menu will display.*

The Kernel Source Directory

~/Bionic-Builder/Kernel-SRC/linux
GITHUB REPO
https://github.com/Bigcountry907/linux.git -b hikey970-v4.9-Debain-Working

The Tool-Chain Directory

~/Bionic-Builder/Tool-Chain/gcc-arm-8.2
Tool-Chain Source
https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-a/downloads

B) *Kernel Building Menu*

Choosing option B will automatically perform all kernel build operations.

Choosing option C will run make menuconfig and allow you to change configuration.

Choosing option O after running option B or C will run make oldconfig

C) *After the kernel build is complete you can find the kernel Image-hikey970-v4.9.gz and the device tree kirin970-hikey970.dtb in the following path*

~/Bionic-Builder/Install/kernel-install/

The modules will be compressed into file Kernel-Install.tar.gz and the script K-INST.sh will be found there also.

D) *Installing the kernel*

A) *For automatic kernel installation into the rootfs USE -->> option 3.*

B) *If you have already flashed the hikey 970 with a system image and you only need to install the kernel on the board the K-INST.sh can be used.*

#1 Copy K-INST.sh from ~/Bionic-Builder/Install to ~/ on hikey 970.
#2 Copy Kernel-Install.tar.gz from ~/Bionic-Builder/Install to ~/ on hikey 970.
#3 sudo -s   <<-- ON Hikey 970
#4 ./K-INST.sh   <<-- ON Hikey 970
After running k-inst.sh The kernel the device tree and the modules are copied to the correct locations on the hikey 970.

(3) COPY KERNEL & DEVICE TREE / INSTALL KERNEL MODULES in ROOTFS

A) *Option number THREE will copy the Image-hikey970-v4.9.gz and kirin970-hikey970.dtb to the rootfs. ~/Bionic-Builder/Ubuntu-SRC/build/rootfs/boot/*

The kernel modules will also be automatically installed to the proper directory :
~/Bionic-Builder/Ubuntu-SRC/build/rootfs/lib/modules/${uname -r}

NOTE: If you are not building the complete system image and are just building the kernel to install on the Hikey970 board follow the previous instructions. **D)** *Installing the kernel*

GENERATE ROOTFS IMAGE FOR FASTBOOT FLASH

(4) GENERATE FLASHABLE AND COMPRESSED IMAGES

A) *after you have run option #1 option #2 and option #3 you are ready to create the image that will be used for fastboot flash. Run option 4 completes the build.*

B) *If you make changes to the rootfs or to grub.cfg you can do that in the build.*
The rootfs is at ~/Bionic-Builder/Ubuntu-SRC/build/rootfs/
You can make any changes you like to the rootfs and simply create a new rootfs image for flashing by running option 4 again. There is no need to run option 1 again.

C) *You can modify and rebuild the Kernel using option 2. Then use option 3 to install the new Kernel. After using option 2 and 3 you can use option 4 to create a new image that has the updated kernel. There is no need to run option 1 again.*

D) *You can chroot ~/Bionic-Builder/Ubuntu-SRC/build/rootfs/ this will switch your running systems root to ~/Bionic-Builder/Ubuntu-SRC/build/rootfs/. Meaning any commands you run in the chroot environment are applied to your build.*
~/Bionic-Builder/Ubuntu-SRC/build/rootfs/
For example sudo apt-get update sudo apt-get upgrade.
The update and the upgrade are applied to your build and not the host machine.

NOTE: Type exit to end the chroot !!

After you have made your changes run option 4. You guessed it. !!

UPDATE BOTH GRUB.CFG FILES (BOARD AND BOOT IMAGE)

A) To update the grug.cfg you only need to edit the grub.cfg that is in ~/Bionic-Builder/Binaries
After you make changes to the grub.cfg file run option 5, option 5 will copy the grub.cfg to the rootfs /boot/grub/grub.cfg and also will mount boot-hikey970.uefi.img and copy the grub.cfg to /boot/grub.grub.cfg.

B) After running option 5 to update the grub.cfg you need to run the following command.

```
fastboot flash boot boot-hikey970.uefi.img
```

If you plan to flash the system with the new system image then there is nothing else to do.
The new system image generated by option 4 after running option 5 will have the new grub.cfg.

C) If you have already flashed the system and just want to update the grub.cfg then.

```
fastboot flash boot boot-hikey970.uefi.img
```

You will need to manually copy the updated grub.cfg from ~/Bionic-Builder/Binaries to /boot/grub/ on the hikey970 board.

INSTALLING THE COMPLETE PACKAGE

A) Set the switches on the dip block to the following.

1 = ON
2 = OFF
3 = ON
4 = OFF

B) Connect the usb-c cable to the port next to the HDMI on the hikey970.

C) Copy the entire INSTALL folder from ~/Bionic-Builder/Install to the host Machine you have the hikey970 connected to. Open terminal or command prompt and cd to the Install directory. Power on the hikey970.

D) In Windows run :

```
fastboot devices    <<-- you should see the hikey970 attached
```

Then Run update_Hikey970.bat and the entire package will be flashed.

E) In Linux run :

```
sudo fastboot devices    <<-- you should see the hikey970 attached
```

Then Run sudo ./update_Hikey970.sh and the entire package will be flashed.

If you already flashed the bootloaders and just need to flash system

```
fastboot flash system ubuntu_bionic.hikey970.V-2.0.sparse.img  <<-- windows
sudo fastboot flash system ubuntu_bionic.hikey970.V-2.0.sparse.img <<-- Linux
```

Post Install Setup / First Boot

A) *Set the switches on the dip block to the following.*

1 = ON
2 = OFF
3 = OFF
4 = OFF

NOTE: You can also set switch 1 to OFF. Then you will need to press the power button to boot after plugging the hikey970 in. This is useful if you are using NVME. You do Not want to just disconnect the power to reboot. That will corrupt the NVME.

To shut down properly type sudo poweroff and when it says rebooting in 5 seconds disconnect the power. If you are using the desktop the normal shutdown is ok.

B) *Connect the usb-c cable to the usb-c port on the side of the board, the port that is not next to the HDMI. This set of boot loaders uses the usb-c for uart both to send commands and receive commands. Use Putty in windows or screen in linux. You need the serial console to see the grub boot menu. Other than that you don't. The fb console and drm are working so after the board completely boots if you have a monitor connected to HDMI you will see the login prompt on the monitor.*

C) *I have included a post install script init.sh in /etc/.*

After the first boot run:

```
sudo -s
cd /
cd /etc
./init.sh
```

The script will test the internet connection then it will run all of the following.

```
sudo resize2fs /dev/sdd12
sudo apt-get update
sudo apt-get upgrade
sudo tasksel install standard
```

And if you choose to install a desktop.

```
sudo apt-get install xubuntu-desktop
```

D) *Booting up into the desktop..... When you first boot to the desktop you will be using either light dm or sddm. Make sure you click on the little bar to set xubuntu as the desktop to boot into and then log in. Everything should be good to go.*