**GRUB based Automated Flashing for Continous Integration and Validation**

The purpose of this project is to introduce a method to automate the flashing of android-ia to Intel boards supporting GRUB.

**Motivation :** Sometimes the regular flashing methods like Kernelflinger, fastboot do not work due to missing support or BIOS dependencies. GRUB is one of the generic way which is widely available on almost all high-end Intel boards and can be used an alternative way of flashing. But, as of now, there is no automation available for that.

**Solution :** This method automates this whole flashing process for GRUB by emulating the user actions :

* Plug/unplug USB storage
* Write images directly from build system to the USB device and deploy it to DUT (Device Under Test)s

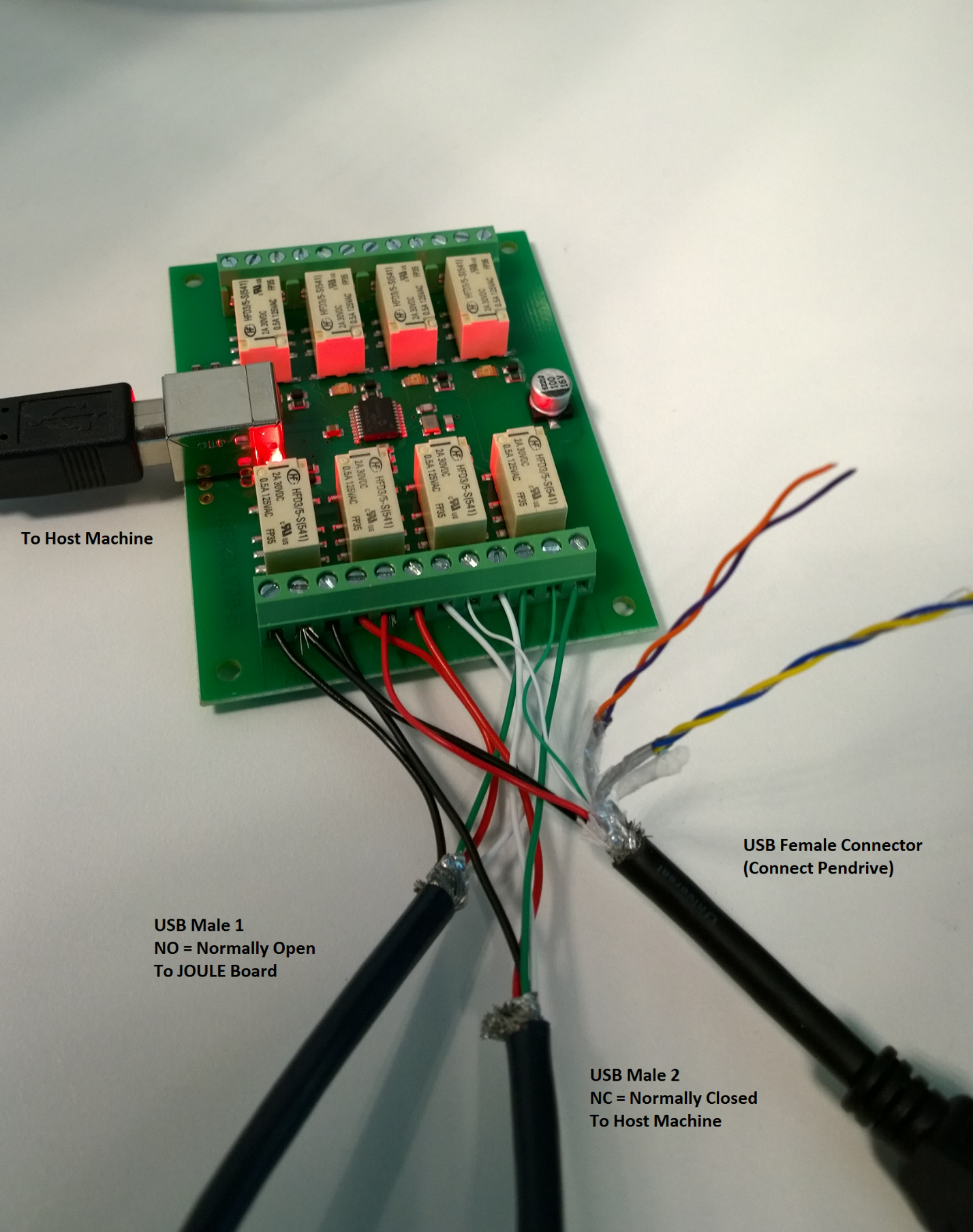
The idea is to share the USB storage device (containing build images) between host machine and DUT. Initially, host machine writes the images from build system to the storage device (makes it bootable). Once done, the USB device is disconnected and routed to the DUT using relay. DUT can then boot through this device for installation.

Attached is the wiring diagram for the setup. It makes use of the following hardware items :

1. Relay Board
2. 2 Male USB heads
3. 1 Female USB head (to connect USB storage device)
4. USB storage device

You need to give the USB as highest priority in the boot order.

[Note : One fallback is that the OS installed may again change the boot order and you may not be able to boot through USB again. You may need to overwrite the boot sectors in order to make the OS itself as non-bootable on next reboot, so that BIOS can fall back to USB boot option again]



Sample bash code :

function flashGrub {

echo "Flashing the $1 GRUB image"

############### Flashing the build ###############

echo "Going to remove android bootloader"

timeout 180 adb wait-for-device

errorAndExit $? "Board not responding, probably broken from previous flash... Need manual intervention"

adb shell dd if=/dev/zero of=/dev/block/sda1

sync

#Connect the USB to host machine

echo "Connecting the USB to this machine"

sudo echo s > /dev/ttyACM0

sudo echo t > /dev/ttyACM0

sudo echo u > /dev/ttyACM0

sudo echo v > /dev/ttyACM0

echo "USB plugged in to this machine"

#Make the USB bootable

sleep 5 #waiting for the USB to be mounted

echo "Writing images to USB / Making USB bootable"

sudo dd if=archive/grub/$1/${IMAGE\_BASE}.img of=/dev/sdb

sync

umount /dev/sdb1

echo "USB made bootable"

#Plug the USB to the board

echo "Connecting the USB to the board"

sudo echo i > /dev/ttyACM0

sudo echo j > /dev/ttyACM0

sudo echo k > /dev/ttyACM0

sudo echo l > /dev/ttyACM0

echo "USB plugged in to the board"

#Reboot the board, flash will automatically start as board should boot through USB now

adb reboot

#Waiting for device to come up

echo "Device being flashed.... Waiting for device to come up"

timeout 300 adb wait-for-device

errorAndExit $? "Board didn't boot up.... something went wrong. Marking this build as failed"

echo "Flashing done successfully and device is online"

}

#calling the function to flash

flashGrub "eng"