FP-GAme

Getting Started

This guide will walk you through setting up the DE10-Nano, its SD Card, and the required GPIO controller port connections to run the FP-GAme hardware. At the end of these instructions, you will have a working FP-GAme platform which runs a tech-demo game after booting.

After you complete this guide, we recommend visiting the fpgame_developers_manual.pdf guide also included within this FP-GAme User Repository. That guide offers an overview of how retro graphics rendering works on FP-GAme, as well as some instruction and best practices for programming your own game.

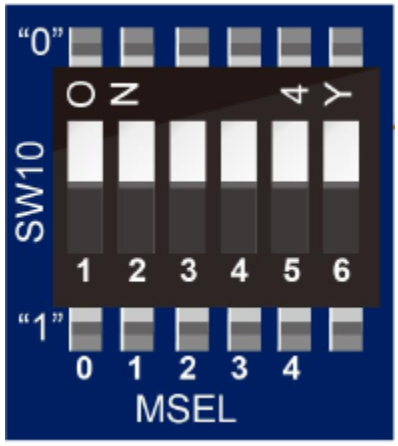
Materials Needed

- DE10-Nano Board: http://de10-nano.terasic.com
    - This board is bundled with a micro SD Card and DC adaptor.
- A computer (Linux, MacOS, Windows) with the ability to read/write the SD Card.
- FP-GAme .img File: fpgame.tar.gz
- A toothpick or similarly sized tool to push tiny switches on the DE10-Nano board.
- A modded SNES Extension Cable as per the snes_controller_mod_instructions.pdf guide also found within the FP-GAme User Repository.

Instructions

Programming the MSEL Switches on the DE10-Nano

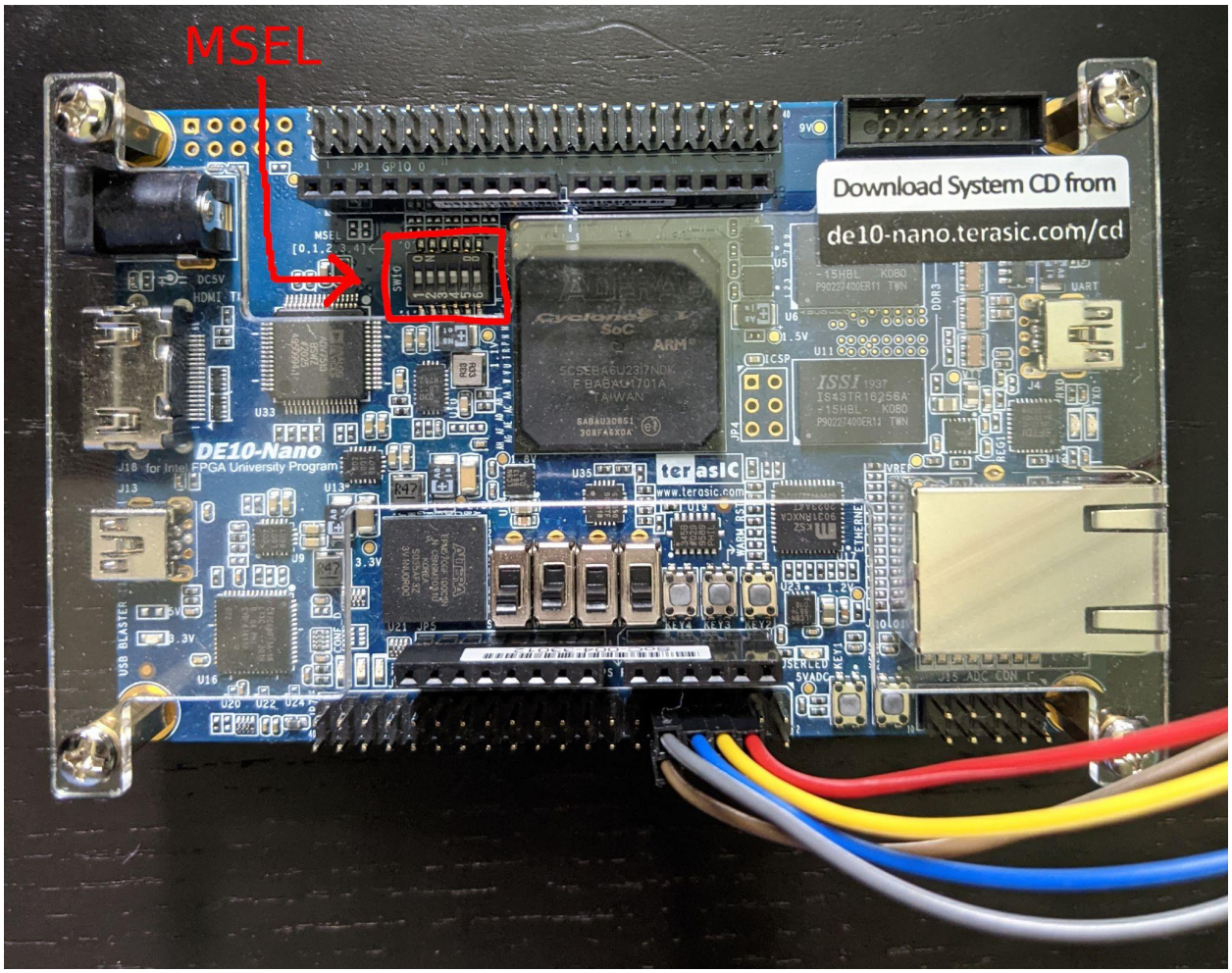
In order for the DE10-Nano to automatically program the FPGA with FP-GAme's hardware configuration from SD card, the following setting must be set on the MSEL switches on the DE10-Nano board:



Above: Required MSEL configuration for FP-GAme [0]

Your MSEL switches may be programmed like this by default. If so, you can move on to the next section. Otherwise, locate them on your board and set them all to ON.

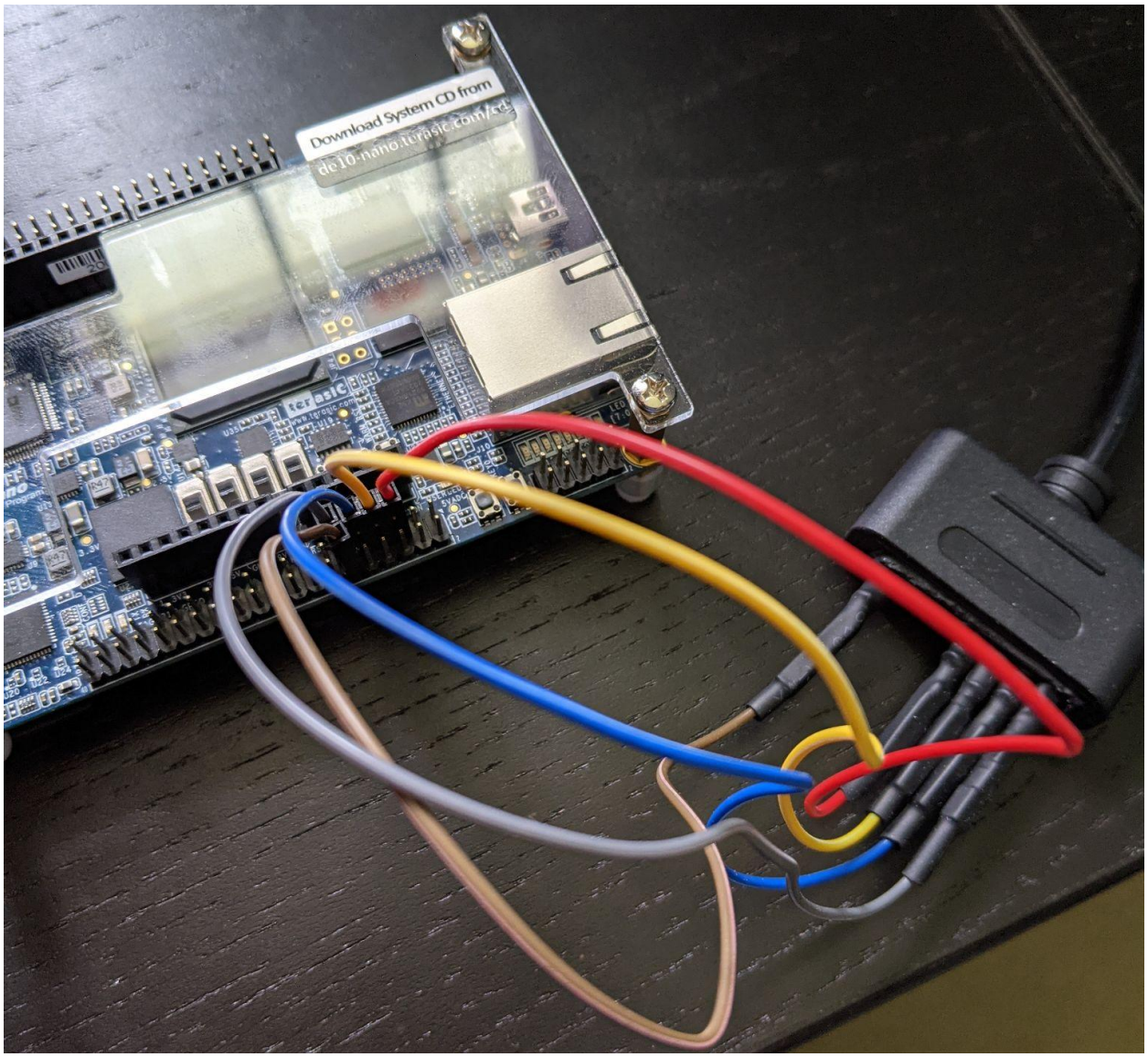
These switches are small, and can be found on the board in between the DC socket and large Cyclone V chip. See the image below for reference:



We recommend using a wooden toothpick to push these small switches.

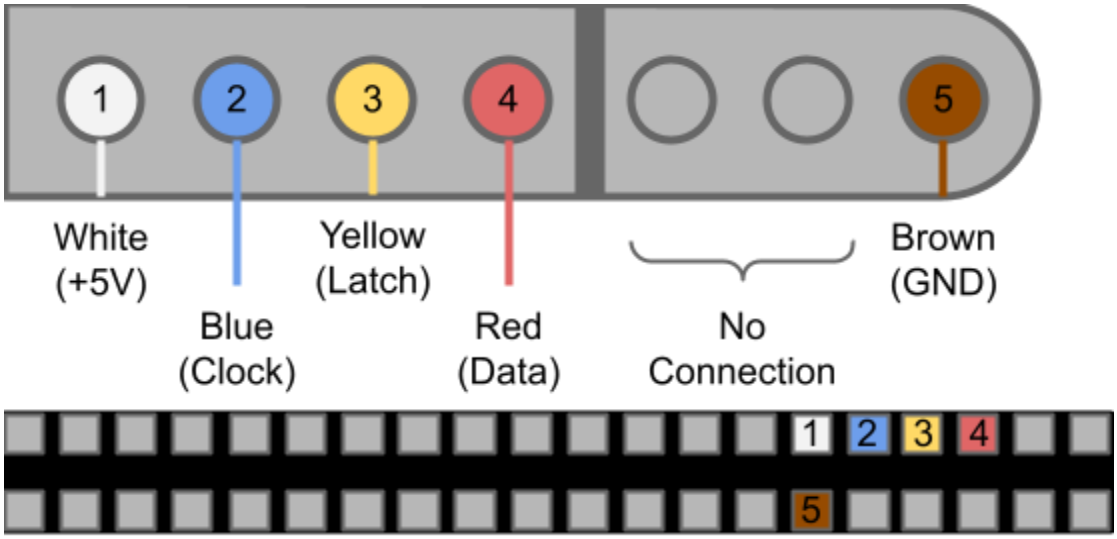
Attaching the Modded SNES Controller Extension Cable to the GPIO

We will be attaching the 5 gpio pins of the SNES Controller Extension Cable constructed in the snes_controller_mod_instructions.pdf document to the DE10-Nano's GPIO header 1.



Above: Resulting GPIO connections from the SNES controller extension cable mod

Attach the pins to their correct locations by following the color codes and numbers from the following diagram:



The first picture in this section shows our setup which uses similarly colored wires (with a grey wire in place of white). You can use this as a reference to double check that your connections look similar.

We will verify that these connections can read an SNES controller by playing the tech-demo later on.

Copying the FP-GAme Image to the DE10-Nano's SD Card

--- Linux Instructions ---

We primarily support Linux in these instructions, but brief suggestions are included for MacOS and Windows users below these instructions.

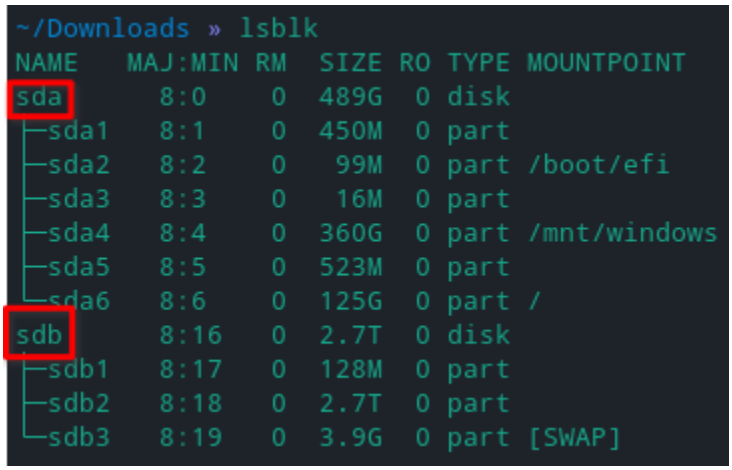
Open a terminal to the location where you downloaded the FP-GAme .img File (fpgame.tar.gz).

Run the following to decompress the .img file. Note that this file takes up 2GiB or so: Make sure you have the space.

```
tar -xvzf fpgame.tar.gz
```

In this step, we will be flashing the SD card with this image using the `dd` command. Be very careful with the of=/dev/<YOUR SD CARD> argument. This is the name of your SD card device in /dev/. If you do not get the correct device (your SD Card), you may end up wiping your system!

To ensure you have the correct device name, first unplug your SD card from your PC. Then, run `lsblk` and note down which devices are your harddrives. They will be titled something like (sda, sdb, ...etc). An example output is shown below on a PC with two harddrives:



Now plug the SD card back into your PC and rerun the `lsblk` command.

```
~/Downloads » lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0   489G  0 disk
├─sda1   8:1    0   450M  0 part
├─sda2   8:2    0    99M  0 part /boot/efi
├─sda3   8:3    0    16M  0 part
├─sda4   8:4    0   360G  0 part /mnt/windows
├─sda5   8:5    0   523M  0 part
└─sda6   8:6    0   125G  0 part /
sdb      8:16   0   2.7T  0 disk
├─sdb1   8:17   0   128M  0 part
├─sdb2   8:18   0   2.7T  0 part
└─sdb3   8:19   0   3.9G  0 part [SWAP]
sdd      8:48   1   7.3G  0 disk
├─sdd1   8:49   1 819.2M  0 part
├─sdd2   8:50   1     1G  0 part
└─sdd3   8:51   1     1M  0 part
```

You should see a new device listed. In the example above, my SD card device is “sdd”. Make note of the name of your SD card device, as it we will use it to replace <YOUR SD CARD> in the `dd` command.

Before running `dd`, double check to make sure you have not mounted the SD card. If you have, the `lsblk` command will tell you the mount point under the MOUNTPOINT column in the console output. Take note of the partition (named sdX1, sdX2, … etc.), and run the following:

```
sudo umount /dev/<YOUR SD CARD PARTITION>
```

An example is shown below where I have accidentally mounted a partition of the SD card:

```
~/Downloads » lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0   489G  0 disk
├─sda1   8:1    0   450M  0 part
├─sda2   8:2    0    99M  0 part /boot/efi
├─sda3   8:3    0    16M  0 part
├─sda4   8:4    0   360G  0 part /mnt/windows
├─sda5   8:5    0   523M  0 part
└─sda6   8:6    0   125G  0 part /
sdb      8:16   0   2.7T  0 disk
├─sdb1   8:17   0   128M  0 part
├─sdb2   8:18   0   2.7T  0 part
└─sdb3   8:19   0   3.9G  0 part [SWAP]
sdd      8:48   1   7.3G  0 disk
├─sdd1   8:49   1 819.2M  0 part /run/media/jpyankel/E0B0-FE35
├─sdd2   8:50   1     1G  0 part
└─sdd3   8:51   1     1M  0 part
```

I needed to run `sudo umount /dev/sdd1`.

Now that we have ensured the SD card is unmounted, we are finally ready to run `dd`. Assuming you are still in the same directory as the newly extracted fpgame.img. Run the following command to flash fpgame.img to your SD card, replacing <YOUR SD CARD> with the device name (sdX) you found earlier (Not sdX1, sdX2, … etc., which are partitions).

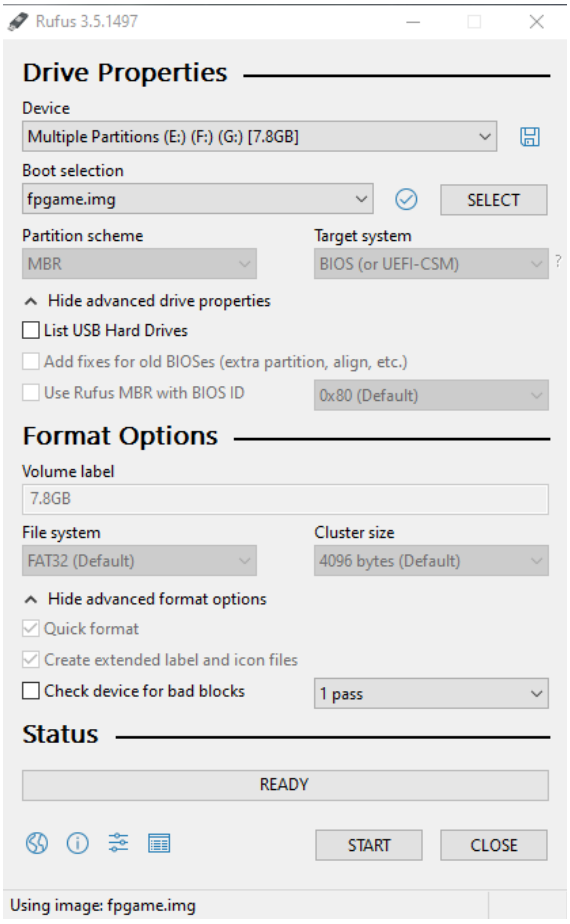
```
sudo dd bs=1M if=fpgame.img of=/dev/<YOUR SD CARD>
```

This command will likely appear frozen, as it has a lot of copying to do to a relatively slow storage device. Just let it run for a minute or two.

--- MacOS Instructions ---
The instructions are largely the same for Linux users. You will be using the Terminal application included with MacOS and inputting the same commands. There is are two exceptions:

1. The `lsblk` command should be replaced with the functionally equivalent `diskutil list`.
2. The `umount` command should be replaced with `diskutil unmountDisk`.

--- Windows Instructions ---
The easiest method is to use Rufus to flash the .img file to your SD card. Rufus can be found here: https://rufus.ie/



Running Rufus will open a window like the one shown to the left.

Under Device, select your SD card. It may have already been selected for you.

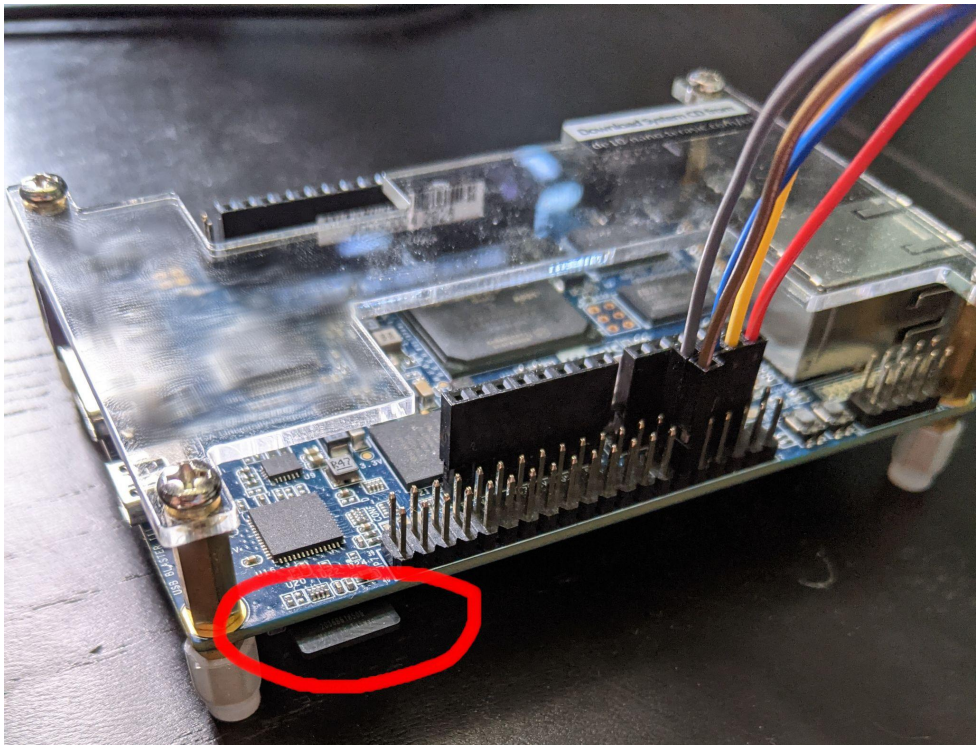
Select the extracted fpgame.img.

Then press START.

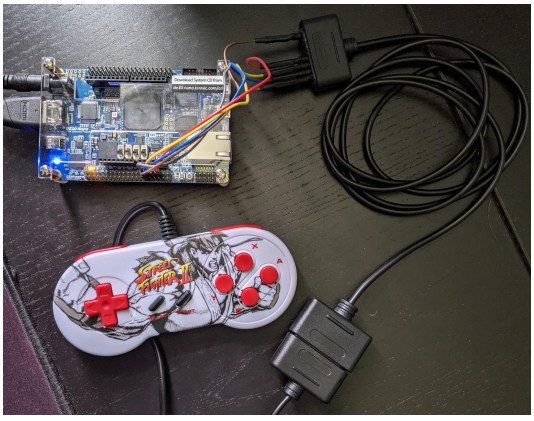
Playing the Tech Demo

You have set up the MSEL switches and the GPIO SNES controller port, and flashed the FP-GAme image to the SD card. All that left to do is play the included tech demo.

First, plug in the SD card to the DE10-Nano. The SD card slot is hidden under the board just to the left of the GPIO header 1. See the image below for reference:



Above: Location of SD card. Make sure to push the SD card into its spring-loaded slot (unlike how it is shown in the picture above).

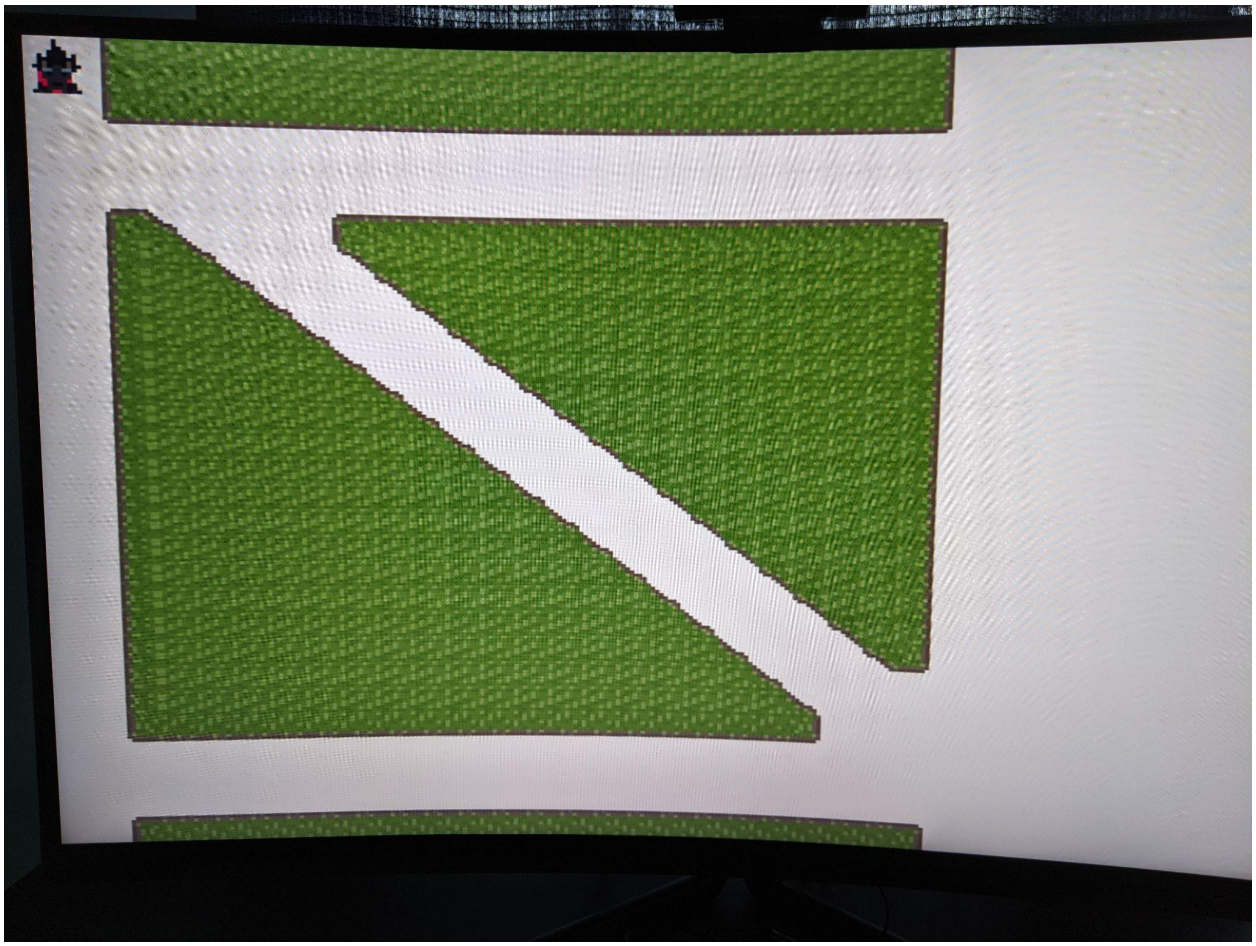


Now plug in the HDMI cable, your SNES controller, and the DC power adaptor.

If done correctly, a blue light will indicate the board is powered.

Shortly after, the board should show an orange light next to the GPIO header 1, indicating that the FPGA has been programmed.

Congratulations, the monitor should display the FP-GAme tech demo:



Test your controller by moving the Scotty Dog sprite using the D-Pad. Test audio by pressing the B-Button to have Scotty bark.

Now that you have the demo working, we recommend visiting fpgame_developers_manual.pdf, also included within this FP-GAme User Repository. This guide offers an overview of how retro graphics rendering works on FP-GAme, as well as some instruction and best practices for programming your own game.

If you are looking instead to make changes to FP-GAme hardware itself, or to view the source files, then be sure to check out the FP-GAme source repository:
https://github.com/FP-GAme/fpgame-src

References

[0] Image Source: DE10-Nano User Manual from System CD