ISS Picture Frame

Andrew Karch

May 14, 2016

DESCRIPTION

A simple picture frame built around a Raspberry Pi Model B and a PiTFT. The stream will auto start, can switch between streams from the International Space Station every 5 minutes, and reconnect if it loses connection. Due to the small size of the stream (240p and 284p) it doesn't use much bandwidth, so there shouldn't be very much network slowdown. The International Space Station is often out of communication with the receivers, there are times where the streams are not available or is on the dark side of the Earth.

The International Space Station Orbits the earth about ever 90 minutes, so there are about 16 sunrises you can see everyday!

The build is pretty simple and straight forward. A basic knowledge of soldering and Linux are all that are required.

You can expand on it by adding buttons to the front to add a power button and to switch between streams.

SUMMARY

1. Buy the necessary parts
2. Assemble the PiTFT Screen
3. Download and install Raspbian Lite to an SD card
4. Boot up the Raspberry Pi, Login as “pi” password: raspberry
5. Run Raspberry Pi Configuration to expand the file system and change the keyboard layout (If desired)
6. If desired, set up Wi-Fi
7. Disable Overscan and000 set the Framebuffer width and height to 320/240 respectively
8. Install PiTFT Kernel
9. Install python-pip, cmake, git, omxplayer, and FBCP
10. Create Script
11. Edit rc.local to disable the cursor, set FBCP and the run script to start on boot

Parts

REQUIRED PARTS

Item	Cost
Raspberry Pi Model B - The B is the only model compatible with the PiTFT Screen	$39.95
PiTFT - Capacitive or Resistive 320x240 2.8" TFT + touchscreen for Raspberry Pi - Capacitive has a better screen, while resistive is cheaper	$44.95
PiTFT Enclosure for Raspberry Pi Model B	$9.95
Kingston 8 GB microSDHC Class 4 - You don't need very much space	$3.92
Edimax Wi-Fi USB Adaptor - I recommend this one as some cheaper ones don't have the bandwidth needed to support the video.	$9.99
Total	**$108.76**

OPTIONAL PARTS

Item	Cost
Addicore Raspberry Pi Heatsink Set (Set of 3 Aluminum Heat Sinks) - Optional, the Pi shouldn't get too hot	$4.55
Black Shortening microSD adapter for Raspberry Pi & Macbooks - Optional if you don't want the SD card to stick over the edge.	$5.95
12 Watt USB Power Supply - I used an iPad Charger	$9.99
Tactile Switch Buttons (6mm slim) x 20 pack - Optional, if you want an on/off switch.	$4.95
USB Micro-B Male Right Angle to Female Extension Cable - To route the code through the back	$7.98
JBtek Raspberry Pi Micro USB Cable with ON / OFF Switch - adds an on/off switch	$5.98
Total	**$39.40**

TOOLS

Item	Cost
Soldering iron and Solder - Required if you don't have one and didn't get the assembled PiTFT	$22.00
USB Keyboard - Any USB Keyboard will work. There is no need for a USB mouse since this guide only uses the terminal.	$14.95
Total	**$36.95**

Instructions

First step is to solder the PiTFT screen together, if you didn't get the assembled version of the PiTFT. The most up to date instructions on how to assemble the PiTFT can be found on Adafruit's website.

Go to the Raspberry Pi Foundation's website and download the latest Raspbian Lite image. I recommend Raspbian Lite over the Adafruit PiTFT Lite image, once the PiTFT software is installed it is difficult to get the pi to output to HDMI for easier configuration. Using the lite version, will save on resources since this project doesn't need a graphical user interface or have many dependencies. Alternatively, you can download the official Adafruit Jessie image, this will allow you to skip setting up the PiTFT later, but limit your ability to use external displays during set up.

Regardless of which image you chose in the previous step, install Raspbian to the SD card following these instructions.

Once the image is installed, attach the PiTFT to the Raspberry Pi, plugin the USB keyboard, Wi-Fi adaptor (or an ethernet cable), connect the pi to an external display such as a TV or monitor (If you didn't use Adafruit's image), and power on the pi.

Once your pi is powered on login using:

username: pi
password: raspberry

First thing you will need to do with your new installation is expand the file system and change the keyboard layout to match the keyboard you're using. You can do this by using:

```
raspi-config
```

From the menu expand the filesystem (Option 1) and change the keyboard layout (Option 4) if you're not using a UK Keyboard.

If you're planning on using Wi-Fi, connect your Raspberry Pi to Wi-Fi using these instructions. At this point you can run `hostname -I` to get your pi's IP Address and ssh in to it using the terminal or PuTTY. Sometimes the Wi-Fi will not work immediately and may require a reboot.

Next step is to disable overscan, to do this open the /boot/config,txt with `sudo nano /boot/config.txt` (Make sure you add sudo, otherwise you won't be able to save your changes). In the config.txt:

1. Uncomment out this line: disable_overscan=1
2. Change the framebuffer_width to 320: framebuffer_width=320
3. Change the framebuffer_height to 240: framebuffer_height=240
4. CTRL-X to close, then press Y to write changes.

Once connected to the internet update your pi and install necessary dependences:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install python-pip cmake git omxplayer
sudo pip install livestreamer
```

Now, you need to set up the PiTFT screen. The latest instructions can be found on the Adafruit website, once installed reboot. Now the console will appear on the PiTFT display.

To get the video to output to the TFT screen we need to install FBCP. Omxplayer outputs to a frame buffer not used by the TFT screen. FBCP will duplicate the output across the two buffers showing the video on the PiTFT display. This installation is a bit more complicated as it has to be downloaded from Github and compiled manually. To compile and build FBCP:

```
git clone https://github.com/tasanakorn/rpi-fbcp
mv rpi-fbcp/* /home/pi/
mkdir build
cd build
cmake ..
make
sudo mv fbcp /bin/
clear
```

This will install FBCP, however FBCP needs to be ran, it doesn’t auto start.

Everything is now set up and we just need to make the script that will run. This can be done by making at file located at /home/pi/ named iss-streamer.sh. There are multiple versions of this file you can make, depending on the streams that you want. The two primary stream a High Definition Stream and a Standard Definition Stream you will need to pick if you want to stream the HD, SD, or both.

This script will switch between both streams every 5 minutes. To change the time it stays on each stream change "sleep 300" to how long you want the stream to play in secs. This script is in the Github repo as both-stream.sh

```
#!/bin/bash
while true
do
( cmdpid=$BASHPID; (sleep 300; kill $cmdpid) & exec livestreamer http://
ustream.tv/channel/iss-hdev-payload mobile_240p --player omxplayer --fifo --
player-args "--win \"0 0 720 480\" {filename}”)
sleep 5
( cmdpid=$BASHPID; (sleep 300; kill $cmdpid) & exec livestreamer http://
ustream.tv/channel/live-iss-stream mobile_284p --player omxplayer --fifo --
player-args "--win \"0 0 720 480\" {filename}")
done
```

To play the HD Version only use this script. This is usually focused down on the earth or on the horizon. This script is in the Github repo as hd-stream.sh

```
#!/bin/bash
while true
do
livestreamer http://ustream.tv/channel/iss-hdev-payload mobile_240p --player
omxplayer --fifo --player-args "--win \"0 0 720 480\" {filename}”
done
```

To play the SD version only use this script. It has more of the ISS in frame. This script is in the Github repo as sd-stream.sh

```
#!/bin/bash
while true
do
http://ustream.tv/channel/live-iss-stream mobile_284p --player omxplayer --
fifo --player-args "--win \"0 0 720 480\" {filename}"
done
```

Copy the script you want to use, or download it from the Github repo, and save it as /home/pi/iss-streamer.sh and make it executable:

```
chmod +x iss-streamer.sh
```

Last step is to set everything to autostart. To do this edit /etc/rc.local (`sudo nano /etc/rc.local`) and edit it to this:

```
#!/bin/sh -e
    setterm -cursor off
    fbcp &
    sudo -u pi /home/pi/iss-streamer.sh &
    exit 0
```

Reboot and it should automatically start streaming. If the screen is blue then that means you’re currently streaming the SD stream and it is currently no sending signals. The SD stream has no image for when it is out of range.

LINKS

http://www.ustream.tv/channel/iss-hdev-payload
http://www.ustream.tv/channel/live-iss-stream
http://blog.miguelgrinberg.com/post/watch-live-video-of-earth-on-your-raspberry-pi
http://eol.jsc.nasa.gov/HDEV/