Headless Raspberry PI Projection Setup Manual

Spencer Granlund, Matthew McMahon

Introduction:

This manual is meant to familiarize the reader with the headless Raspberry PI projection system, and also show a step-by-step process on how to set up this system in a laboratory setting. This system was created to solve the following problem.

In order to complete multi-agent SUAS-ground vehicle experiments, the WPI Autonomous Aerial Mobility Lab needed to be able to project large Gaussian threat field on the laboratory floor while maintaining the following:

* Cost effectiveness
* Cable management
* Simplicity

The system has three main components the projectors, the Raspberry PI and the control desk computer. The system works by having a program on the control desk computer generate the required Gaussian threat field, and then use a UDP (User Datagram Protocol) connection to transmit these files to projectors. The projectors already in the lab are not network connected so they needed a way to interface with local Wi-Fi network, this is why the Raspberry PIs were chosen. The Raspberry PIs will act as a connection/input for the projectors, receiving the messages from the control desk and inputting them to the projectors. This is a beneficial solution because it almost entirely removes cables from the system, the PIs are inexpensive and they require simple python scripts to run.

Setting up the Raspberry PIs & Control Computer

Raspberry PIs are popular microcomputers that have the same functionality as their larger counterparts, but a fraction of the power and cost, making them ideal for scenarios like this. The raspberry PIs are meant to be run in a headless format, meaning there will be no monitor to control them. To use the PIs in this way the free program NoMachine will be used to remotely access the PIs from another computer and configure them.

**Step 1: Formatting the Raspberry PI SD card**

To begin this step, you will need the Raspberry PI, a micro SD card, a monitor, and a computer with a micro SD card port.

Start by going to [Raspberry Pi OS – Raspberry Pi](https://www.raspberrypi.com/software/), and downloading the Raspberry PI flasher, a program that formats the SD card for a PI

Next insert the SD card into the computer and use the Raspberry PI flasher to install the recommended 32bit version of Raspian (Operating System)

Now put the SD card into the PI, plug the PI into a monitor and then power the PI. Follow the instruction on the screen to finish setting up the PI

**Step 2: Installing NoMachine**

Using the Raspberry PI, navigate to nomachine.com and find the download page for Raspberry PIs and download the correct .tar version for the Raspberry PI. To check what ARM your CPU uses, open a terminal and type “cat /proc/cpuinfo”. Download the .tar.gx file version.

Next move the file from downloads directory to the main “usr” directory, then execute these commands

“tar -xvf ‘package\_name’.tar.gz”

“cd ‘package\_name’”

“sudo ./nxserver –install”

**Step 3: Run Required Scripts on the PI**

Before running the scripts on the PI you must install and upgrade the python packages that came pre-installed on the PI. Execute the two following commands in the terminal:

“pip install matplotlib==2.2.4”

“pip3 install --upgrade numpy”

“sudo apt-get install libatlas-base-dev”

“sudo apt update”

“sudo apt install python3-opencv”

Using the web browser on the PI, sign into your email and email the PI the required UDP\_PI\_1\_Script file. Open the file and run it in the pre-installed python editor Thonny.

If the PI is set up heedlessly with the projector already, use NoMachine on the control computer to access the PIs.

**Step 4: Run the Required Scripts on the Control Computer**

For the PIs to receive the necessary projection images, the ground control station scripts must be ran by the control computer. This is as simple as downloading an python IDE (Integrated Development Environment ) such as VScode ([Download Visual Studio Code - Mac, Linux, Windows](https://code.visualstudio.com/download)).

After setting up the IDE open the GCS\_script and run it on the IDE, change the IP addresses within the code so that it transmits an image to each of the four raspberry PIs. To check the IP address of each PI, open a terminal in each PI and use the below command, then use the IPV4 address as the IP in the code.

“ifconfig”

The code is currently set up to use a png called “projection” as the image that is split into fourths and distributed, for the experiment, place the desired png in the directory where the code is located on the control computer, and then replace the “projection” term in the code with the name of the desired image.

**Final Notes:**

Once the scripts are running on the PIs and the control system simultaneously, the PIs should be displaying whichever image is defined in the code. If you are having issues with the code itself, feel free to reach out to and ask questions. ([srgranlund@wpi.edu](mailto:srgranlund@wpi.edu) or [mkmcmahon@wpi.edu](mailto:mkmcmahon@wpi.edu) )