

1. Use Raspberry Pi Imager to flash the 0.250 image from BYOAC forum onto a 16gb or larger SD card. Once done, insert SD card into a Raspberry Pi 4B, connect a display and keyboard, and power on.
-

2. NOTE: the username for the system is “pi” and the password is “raspberrry”.
-

3. On first boot, log in and enter this command:

```
./expand-data-partition.sh
```

4. Once that command has finished, reboot the system by sending this command:

```
sudo reboot
```

5. Once rebooted, log in again and enter this command to view the README file:

```
nano README.txt
```

Once done reading, exit the file by pressing CTRL+X.

6. Next, safely shut down the Pi by sending this command:

```
sudo shutdown -h now
```

7. Remove the SD card and configure settings for Wifi and SSH by following this tutorial:
https://linuxhint.com/raspberry_pi_wifi_wpa_supplicant/
-

8. Once those files have been added, reinsert the SD card into Pi and boot.
-

9. Once loaded, the current IP address of the system will be listed in the top left corner. If it shows 127.0.0.1, the Pi is not connected to your network and you will need to troubleshoot. You can also log in and issue the following command to obtain the IP address (note the capitalized i):

```
hostname -I
```

10. From this point it is recommended to continue setup on a separate computer. Use PuTTY to connect to the Pi's command line and FileZilla to connect to the file system, both using the IP address your Pi provided as the host name.
-

11. To make any system change, you need to put the system in read/write mode. By default, to preserve the lifespan of the SD card, the system is kept in read-only mode. Every time you logout from an interactive shell session or if you reboot, the system is put back in read-only mode. To switch the system to read/write mode, send the following command:

```
rw
```

The prompt suffix will change from (ro) to (rw). You can then make any persistent change to the root filesystem of Linux. If you logout or reboot, the system will automatically be restored to read-only mode. To manually switch the system back to read-only mode, send the following command:

```
ro
```

The prompt suffix will change from (rw) to (ro).

12. Now is a good time to configure the system's audio output. Make sure the system is in read/write mode (rw) and then send the following command:

```
sudo raspi-config
```

Navigate to System Options, then Audio, and then select which sound device you want to use. Also take note of the number assigned to each audio device for a later step. Once done, save and exit raspi-config, then issue the following commands, in order:

```
ro  
sudo reboot
```

Once rebooted, make sure the system is in read/write mode (rw) and then adjust the system's volume levels by issuing the following command:

```
alsamixer
```

Once loaded, use the left and right arrows to change between sound devices (HDMI or 3.5mm jack) and use the up arrow to bring the dB gain to ~0 for each one. You can increase this number later if the audio is still too quiet. Once done, exit using the ESC key and issue the following commands, in order:

```
sudo alsactl store  
ro  
sudo reboot
```

Once rebooted, make sure the system is in read/write mode (rw) and then check the following config script by issuing the following command:

```
sudo nano /etc/asound.conf
```

Once the script opens, check that the code displayed matches the following:

```
pcm.!default {  
  type hw  
  card 0  
}  
  
ctl.!default {  
  type hw  
  card 0  
}
```

The number after 'card' represents what sound device the system is using. Ensure it matches the number of the device you want to use from earlier (For example, 0 might be HDMI and 1 might be the 3.5mm jack). Use CTRL+X to exit, saving and overwriting the file if needed. Now, reboot the system by issuing the following command:

```
sudo reboot
```

-
13. If audio is not playing after step 12, make sure the system is in read/write mode (rw) and then adjust the boot/config.txt file by issuing the following command:

```
sudo nano /boot/config.txt
```

Ensure that the following lines are present. If they are not present, add them at the end of the file:

```
dtparam=audio=on  
dtoverlay=vc4-fkms-v3d
```

Use CTRL+X to exit, saving and overwriting the file if needed. Now, reboot the system by issuing the following command:

```
sudo reboot
```

-
14. Using FileZilla, connect to the Pi's file system and navigate to `\data\mame\roms`. This is where you should put your arcade game ROMs. Make sure the system is in read/write mode (rw) and then use FileZilla to transfer your selected ROMs to this folder. Once done, reboot the system by issuing the following command:

```
sudo reboot
```

15. The system has 2 different operating modes: Service Mode, for ROM management and maintenance, and Arcade Mode, for playing games. To toggle between the two modes, make sure the system is in read/write mode (rw) and then issue one of the following commands:

```
servicemode
```

OR

```
arcademode
```

After sending the chosen command, you must reboot the system by issuing the following command:

```
sudo reboot
```

If you need to toggle the system to Service mode but are unable to do it while the Raspberry Pi is running, there are 2 methods you can try:

Method #1:

1. Shutdown/quit MAME to initiate a clean shutdown
2. Remove the SD card from the Raspberry Pi
3. Put the SD card in a card reader connected to a Linux host system (or Mac OS X)
4. Remove/delete these 2 files from the partition named "rootfs" on the SD card:
 - `etc/systemd/system/multi-user.target.wants/mame-autostart.service`
 - `etc/systemd/system/multi-user.target.wants/shutdown.service`
5. Unmount the SD card, put it back in your Raspberry Pi, and power it on
6. The system will boot in Service mode

Method #2:

1. Shutdown/quit MAME to initiate a clean shutdown
2. Remove the SD card from the Raspberry Pi
3. Put the SD card in a card reader connected to a Linux host system (or Windows or Mac OS X)
4. From the "boot" partition, add an empty file called "service" (no file extension)
5. Unmount the SD card, put it back in your Raspberry Pi, and power it on
6. The system will boot in Service mode (it may take up to 2 reboots to take effect)

-
16. To add a button that safely shuts down the Pi before removing power, make sure the system is in read/write mode (rw) and then adjust the boot/config.txt file by issuing the following command:

```
sudo nano /boot/config.txt
```

Once loaded, add these two lines at the end of the file:

```
#shutdown button
dtoverlay=gpio-shutdown
```

Use CTRL+X to exit, saving and overwriting the file when prompted. Now, reboot the system by issuing the following command:

```
sudo reboot
```

Once rebooted, you can add a push button between GPIO3 and Ground (pins 5 and 6) that will safely shut down the Pi before removing power.

17. To add a button that controls the volume level of the system's audio, make sure the system is in read/write mode (rw) and then issue the following command:

```
nano ~/scripts/mame-volume-ctrl.sh
```

Once the script opens, paste the following code into the file, overwriting any existing code:

```
#!/bin/bash
#file:/home/pi/scripts/mame-vol.sh

function cleanup {
    # Deactivate GPIO pin
    echo $GPIOPIN >/sys/class/gpio/unexport 2>/dev/null
    exit 0
}

function waitStateChange {
    while [ $(cat /sys/class/gpio/gpio${GPIOPIN}/value) != $1 ]; do
        inotifywait -q -e modify /sys/class/gpio/gpio${GPIOPIN}/value >
/dev/null
    done
}

# Clean up when exit
trap cleanup EXIT
```

```

trap cleanup SIGHUP
trap cleanup SIGQUIT
trap cleanup SIGINT
trap cleanup SIGTERM

HIGH='95%'
MED='85%'
LOW='75%'
MUTE='0%'

GPIOPIN=4    # We use GPIO 4

# Initialize GPIO pin
[ -d /sys/class/gpio/gpio${GPIOPIN} ] && echo $GPIOPIN >
/sys/class/gpio/unexport # Deactivate GPIO pin
echo $GPIOPIN > /sys/class/gpio/export    # Activate GPIO pin
sleep 0.1                                # A small delay is required so
that the system has time                  # to properly create and set the
                                           # file's permission
echo in > /sys/class/gpio/gpio${GPIOPIN}/direction # Input signal
echo both > /sys/class/gpio/gpio${GPIOPIN}/edge    # We use the interrupt
controller to avoid a CPU spin loop

waitStateChange 1    # Wait for button release
while true; do
    waitStateChange 0    # Wait for button press
    # Volume button pressed
    CHECK_VOL=$(amixer | awk -F'[\\[\\]]' '/Mono: Playback/ {print $2}')

    case $CHECK_VOL in
        $HIGH)
            # Setting volume to medium
            amixer -q cset numid=1 $MED ;;
        $MED)
            # Setting volume to low
            amixer -q cset numid=1 $LOW ;;
        $LOW)
            # Setting volume to mute
            amixer -q cset numid=1 "$MUTE" ;;
        $MUTE)
            # Setting volume to high
            amixer -q cset numid=1 $HIGH ;;
    esac
done

```

```

*)
    # Setting volume to medium (default)
    amixer -q cset numid=1 $MED ;;
esac
waitStateChange 1 # Wait for button release
done

```

Use CTRL+X to exit, saving and overwriting the file when prompted. Next, adjust the file permissions so the file can run by sending this command:

```
chmod +x ~/scripts/mame-volume-ctrl.sh
```

Next, send the following command to create a service so the volume control script can run in the background:

```
sudo systemctl edit mame-volume-ctrl.service --force --full
```

Once the file opens, paste the following code into the file, overwriting any existing code:

```

[Unit]
Description=MAME Audio Volume Controller service
Before=mame-autostart.service
After=alsa-restore.service

[Service]
Type=simple
User=pi
Group=pi
ExecStart=/bin/bash /home/pi/scripts/mame-volume-ctrl.sh
KillSignal=SIGINT

[Install]
WantedBy=multi-user.target

```

Next, run the following commands one at a time to enable and start the service:

```

sudo systemctl enable mame-volume-ctrl.service
sudo systemctl start mame-volume-ctrl.service

```

Now, reboot the system by issuing the following command:

```
sudo reboot
```

Once rebooted, you can add a push button between GPIO4 and Ground that will adjust the system volume when pressed. The button will toggle between High, Medium, Low, and Mute, cycling through each setting one at a time.

-
18. To add custom boot and shutdown images, use FileZilla to connect to the Pi's file system and navigate to `\home\pi\splash`. Replace `mame-boot.jpg` and

mame-shutdown.jpg with your own custom images. The replacement files must be named exactly the same as the original files.