Contents

[1. Attached file and package 2](#_Toc500397730)

[2. Enabling composite USB gadgets on the Raspberry Pi zero 2](#_Toc500397731)

[Step 0 - Configuring the SD card 2](#_Toc500397732)

[Step 1 - Kernel stuff 2](#_Toc500397733)

[Step 2 - Configuring the gadget for Mass storage 4](#_Toc500397734)

[Step3 - Create Mass storage image 6](#_Toc500397735)

[Install program for building USB Thumb Drive which can be synchronized with specified FTP server. 7](#_Toc500397736)

[Install package 7](#_Toc500397737)

[Setup Package 8](#_Toc500397738)

[Copy Main program for USB sync thumb drive and modify necessary setting 9](#_Toc500397739)

[Make a Launcher Script 10](#_Toc500397740)

[Make It Executable 11](#_Toc500397741)

[Add it Crontab 11](#_Toc500397742)

# Attached file and package

First extract my attached zip file. If so, you can look following files in directory.

/ftptool / Python package for main program

/usbsyncthumb.py Main Program for action of sync thumb drive

/InitUSBthumb Script for Setting of USB Gadget

/launcher.sh Script that allows main program to start when RPI power up

# Enabling composite USB gadgets on the Raspberry Pi zero

As you know, in order to use RPI as USB Storage device via USB OTG port, we have to use USB gadget module.

So I have attached steps for your configuration. Start point of this document, RPI zero should serves SSH via USB OTG or WIFI interface.

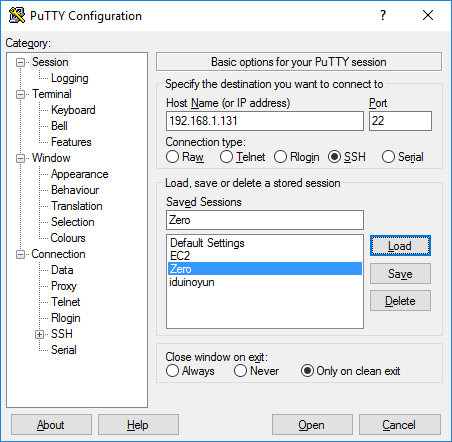
## Step 0 - Configuring the SD card

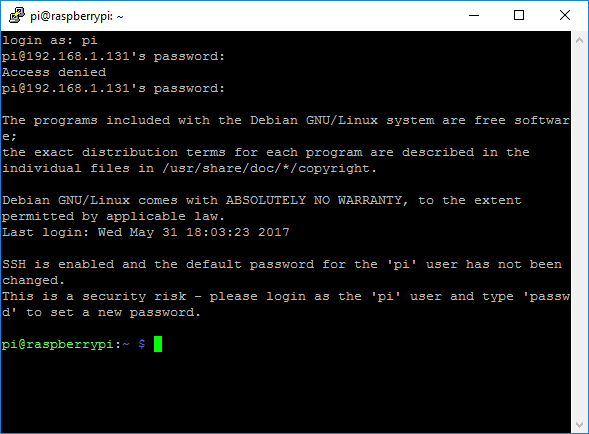
Download and install the latest Raspbian Jessie onto a suitably large SD card, and expand the root partition. This has been described enough already in Adafruit.com and raspberrypi.org.

## Step 1 - Kernel stuff

You can access into RPI using Putty or WinSCP.

you can access to your RPI zero via SSH.





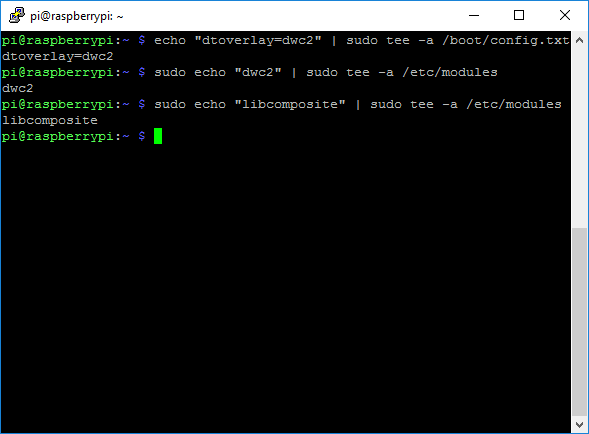
Enter following commands.

**sudo echo "dtoverlay=dwc2" | sudo tee -a /boot/config.txt**

**sudo echo "dwc2" | sudo tee -a /etc/modules**

Finally, we need to enable “libcomposite“ driver by running:

**sudo echo "libcomposite" | sudo tee -a /etc/modules**

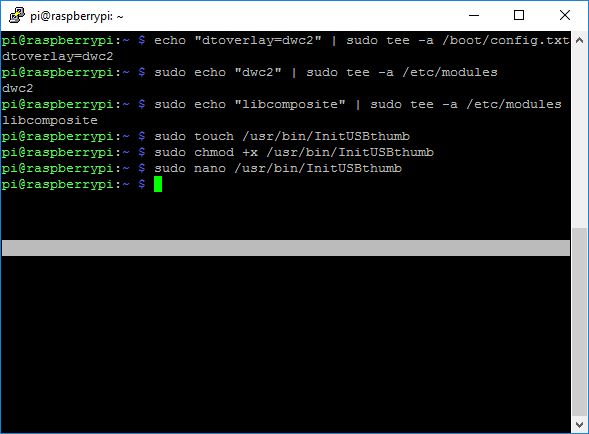


## Step 2 - Configuring the gadget for Mass storage

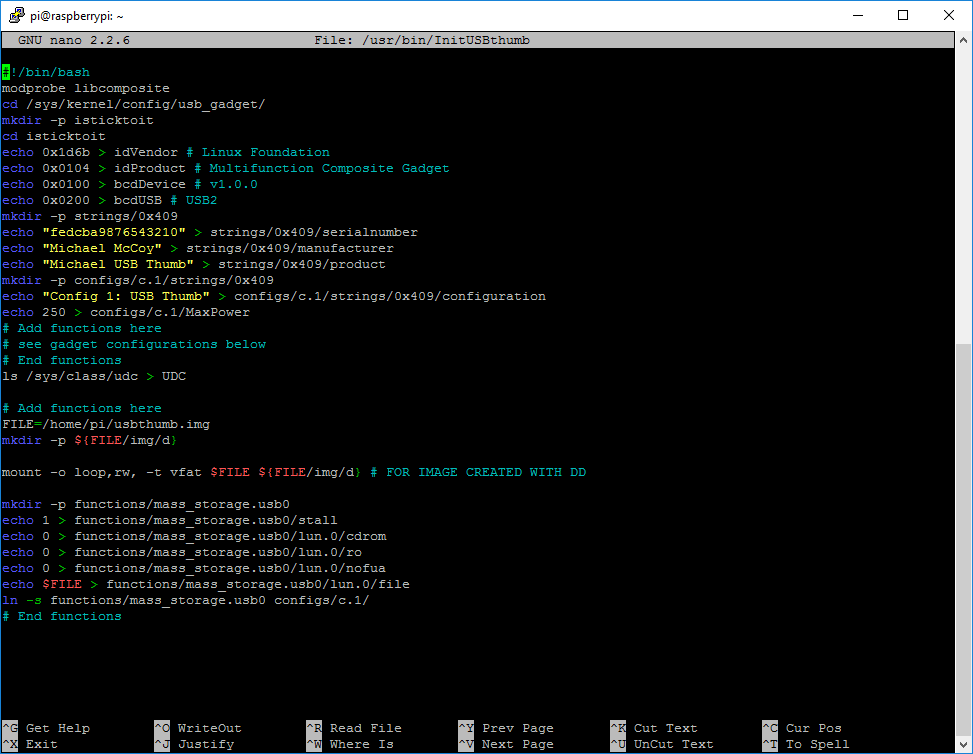
The configuration is volatile, so it must be run on each startup.  
Let’s create file **InitUSBthumb** in “/usr/bin/ “ using putty or another editor.

Type the following in putty:  
**sudo touch /usr/bin/InitUSBthumb****sudo chmod +x /usr/bin/InitUSBthumb**

**sudo nano** **/usr/bin/InitUSBthumb**



In nano editor, you have to edit startup script.



**#!/bin/bash**

**modprobe libcomposite**

**cd /sys/kernel/config/usb\_gadget/**

**mkdir -p isticktoit**

**cd isticktoit**

**echo 0x1d6b > idVendor # Linux Foundation**

**echo 0x0104 > idProduct # Multifunction Composite Gadget**

**echo 0x0100 > bcdDevice # v1.0.0**

**echo 0x0200 > bcdUSB # USB2**

**mkdir -p strings/0x409**

**echo "fedcba9876543210" > strings/0x409/serialnumber**

**echo "Michael McCoy" > strings/0x409/manufacturer**

**echo "Michael USB Thumb" > strings/0x409/product**

**mkdir -p configs/c.1/strings/0x409**

**echo "Config 1: USB Thumb" > configs/c.1/strings/0x409/configuration**

**echo 250 > configs/c.1/MaxPower**

**# Add functions here**

**# see gadget configurations below**

**# End functions**

**ls /sys/class/udc > UDC**

**# Add functions here**

**FILE=/home/pi/usbthumb.img**

**mkdir -p ${FILE/img/d}**

**mount -o loop,rw, -t vfat $FILE ${FILE/img/d} # FOR IMAGE CREATED WITH DD**

**mkdir -p functions/mass\_storage.usb0**

**echo 1 > functions/mass\_storage.usb0/stall**

**echo 0 > functions/mass\_storage.usb0/lun.0/cdrom**

**echo 0 > functions/mass\_storage.usb0/lun.0/ro**

**echo 0 > functions/mass\_storage.usb0/lun.0/nofua**

**echo $FILE > functions/mass\_storage.usb0/lun.0/file**

**ln -s functions/mass\_storage.usb0 configs/c.1/**

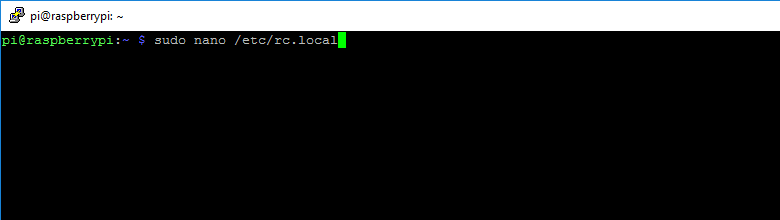
**# End functions**

After you have entered all scripts and then save and close nano editor. (Press key Ctrl+O and Ctrl+X)

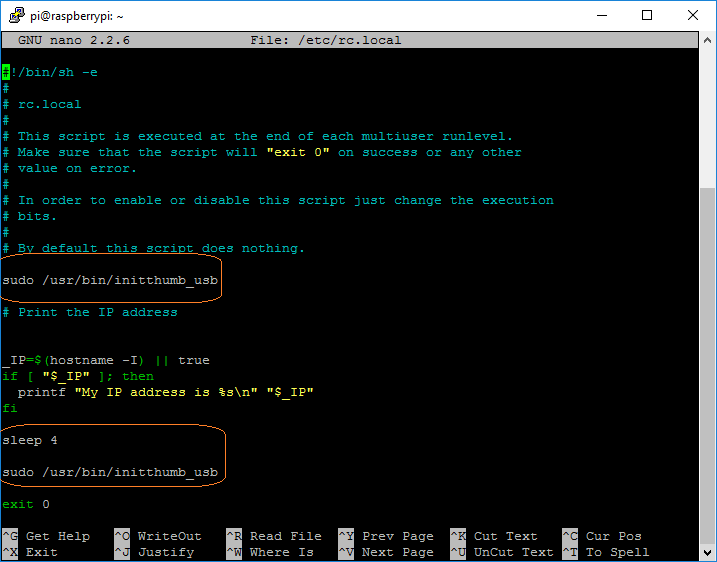
Otherwise, you can copy my script file into RPI suing WINSCP software.

Afterwards, you need to run this script automatically at startup.

**sudo nano /etc/rc.local**

****

And then add marked line

Save and Close.

## Step3 - Create Mass storage image

In putty, type following commands.

**dd if=/dev/zero of=/home/pi/usbthumb.img bs=1M count=1024****mkdosfs /home/pi/usbthumb.img**

You can change size of storage.

And then using command “sudo reboot”, lets restart your RPI zero.

So we can use RPI zero as USB Storage device.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*About “Linux USB gadget configured through configfs”, we can reference several documents and articles.

I prefer these documents.

<https://www.kernel.org/doc/Documentation/usb/gadget_configfs.txt>

<http://isticktoit.net/?p=1383>

<http://blog.gbaman.info/?p=699>

# Install program for building USB Thumb Drive which can be synchronized with specified FTP server.

Install customized ftptool package. In this package, I have defined functions to allow for programmer to sync action.

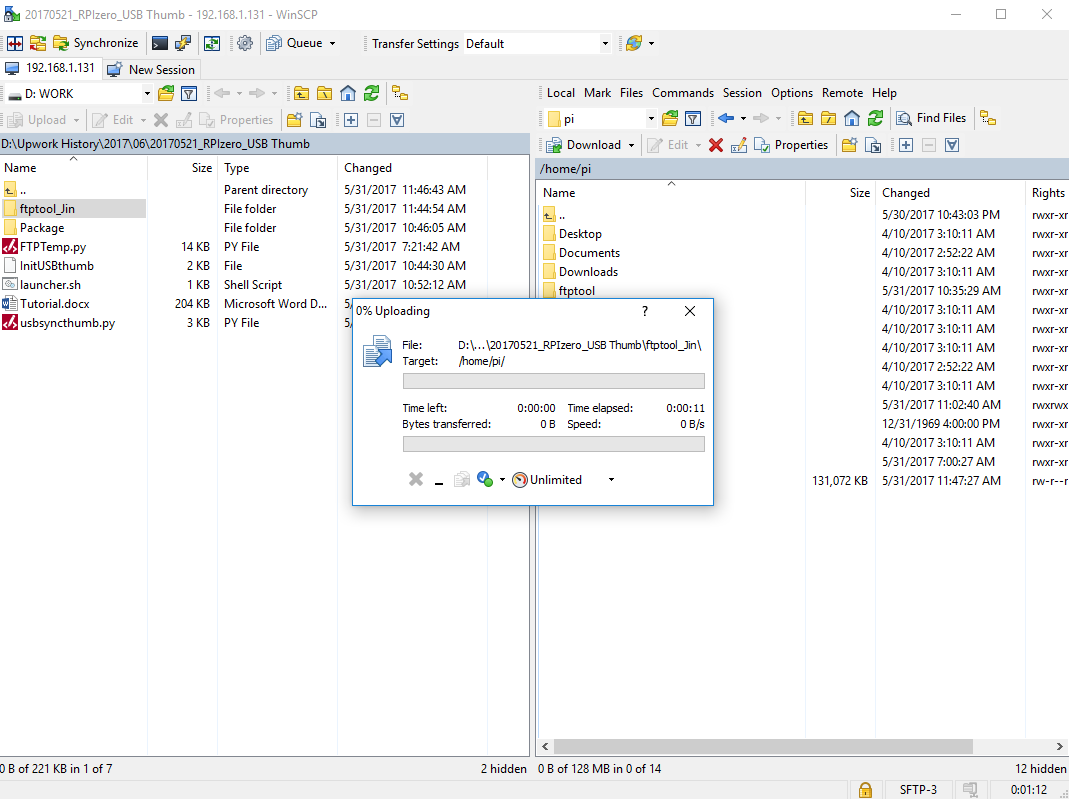
So in main program side, user don’t need consider complex directory tree.

Functions of package allows to easy implementation for your requirements.

You can understand when you review my code.

## Install package

You can copy package files(directory ftptool in attached zip file) into RPI /home/pi directory using WinSCP software. In my opinion, it is best convenient.



## Setup Package

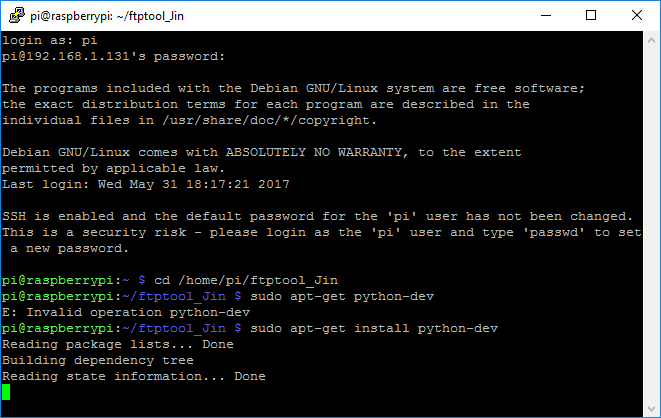
We have to build package in order to use that in code.

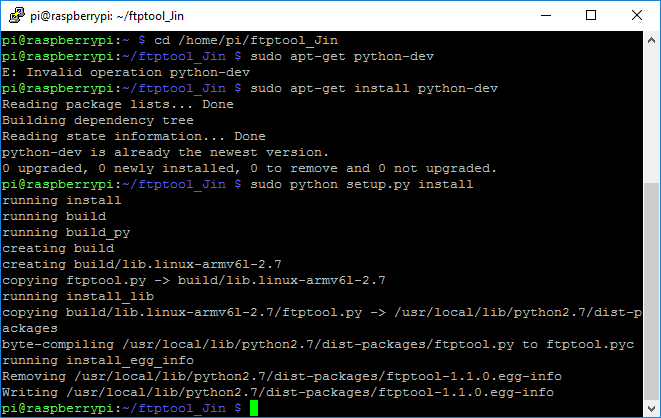
In putty,

**cd /home/pi/ftptool**

**sudo apt-get install python-dev**

**sudo python setup.py install**





## Copy Main program for USB sync thumb drive and modify necessary setting

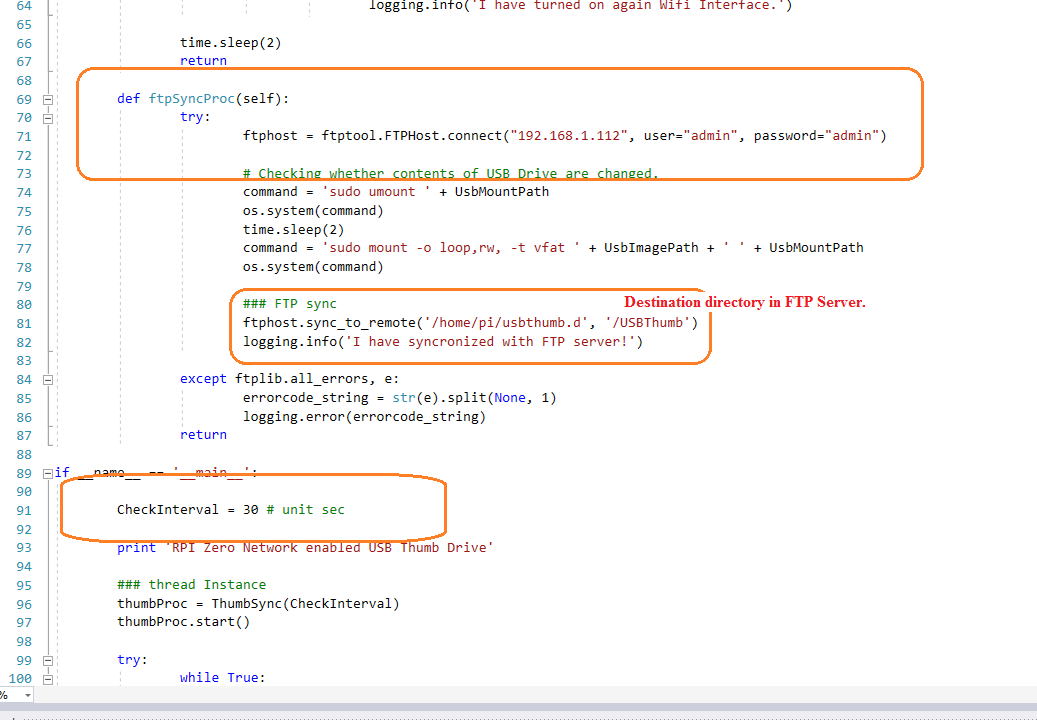
This program will check whether usb directory and files are in FTP server, and then upload files.

And also this action will be processed every specified interval. (Current setting 30 sec, you can change interval)

So when every action, it will check internet connection, and if it is disconnected from internet, again connect stored setting.

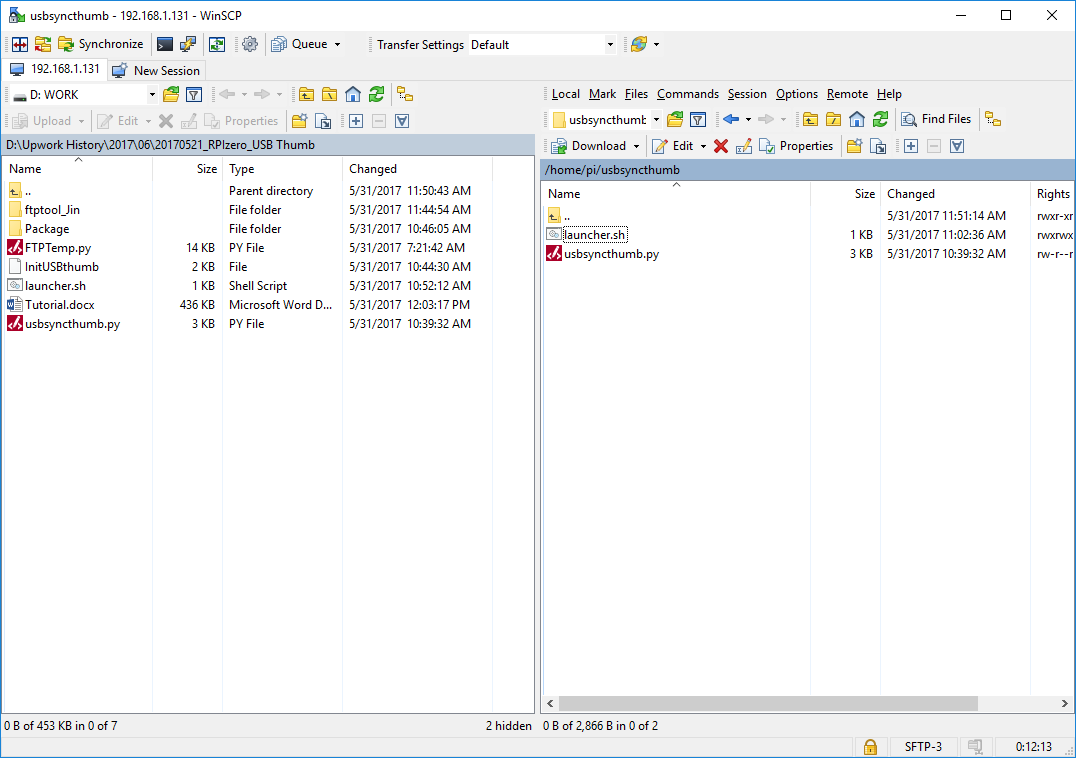
If connection is succeeded, it will sync action. (Check difference and upload files)

And also we can change easily ftp server url and user name, password, destination FTP path.



Also you can modify program in putty, too.

Copy program into RPI using WinSCP. ( /home/pi/usbsyncthumb)



Allow this python program to start when RPI boot.

In last step, we have to activate this script to launch when RPI power up.

There are several solutions for it.

I have used crontab.

## Make a Launcher Script

In putty,

sudo nano /home/pi/usbsyncthumb/launcher.sh

##############

cd /

cd /home/pi/usbsyncthumb/

sudo python usbsyncthumb.py

cd /



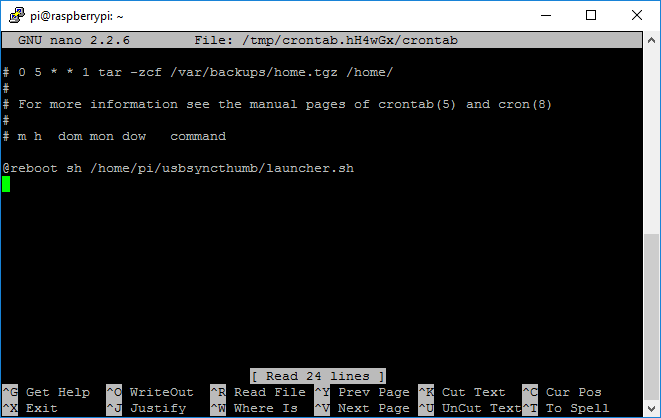
## Make It Executable

sudo chmod -R 777 /home/pi/usbsyncthumb/launcher.sh

**sudo chmod +x** /home/pi/usbsyncthumb/launcher.sh

## Add it Crontab

sudo crontab -e



Insert line at the bottom of file.

@reboot sh /home/pi/usbsyncthumb/launcher.sh

And then reboot your RPI.