# USB Copier README

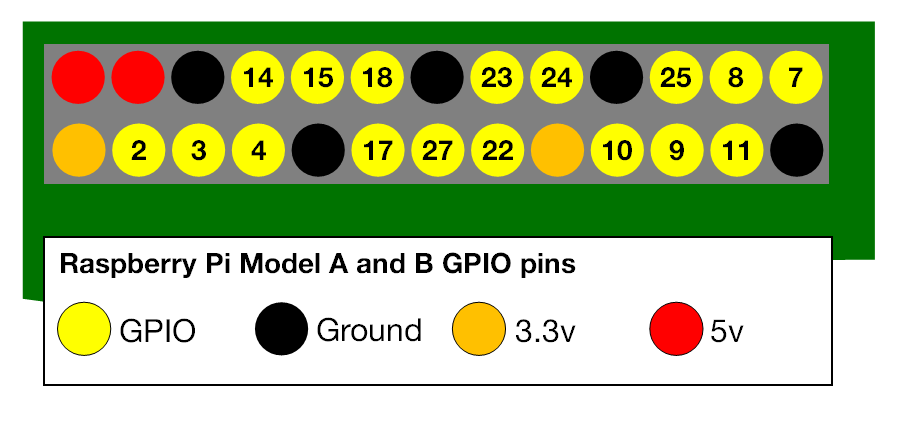
The idea ist o have a small rpi running and be able to attach a usb harddrive and when a second usb stick is attached it gets copied to the disk.

I also added a 16x2 LCD display to show the progress

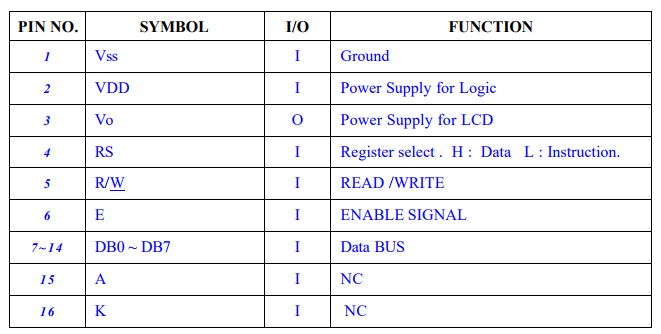
# Wiring:

From <https://pimylifeup.com/raspberry-pi-lcd-16x2/>

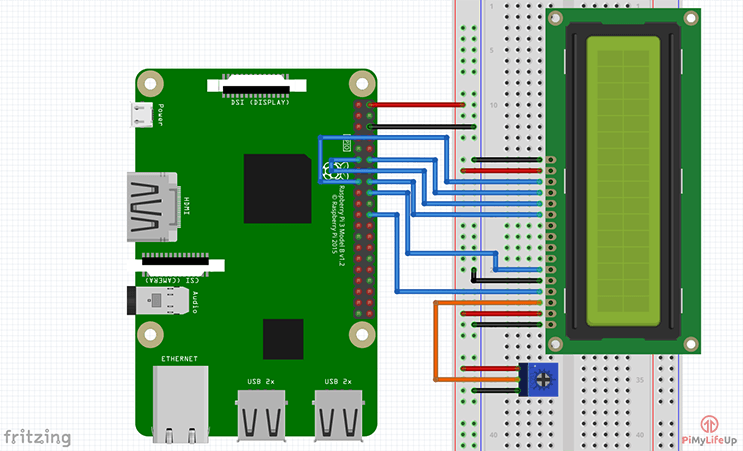
## RPI Pinout



## LCD Display 16x2 Pinout:



## Final wiring:



# Setup LCD

Get the Library:

git clone https://github.com/adafruit/Adafruit\_Python\_CharLCD.git

install it:

cd ./Adafruit\_Python\_CharLCD

sudo python setup.py install

Setup Scripts

cd ~

git clone https://github.com/lightwaver/RPI\_USBCopier.git

# USB (Auto)Mount

<https://raspberrypi.stackexchange.com/questions/41959/automount-various-usb-stick-file-systems-on-jessie-lite>

|  |  |
| --- | --- |
| down voteaccepted | After alot of research I could fix my Problem with usbmount: **Adding ntfs to usbmount**  1. Install usbmount with sudo apt-get install usbmount. 2. Install NTFS driver package **ntfs-3g** with sudo apt-get install ntfs-3g. 3. Configure usbmount to mount specified filesystems by opening the usbmount file with sudo nano /etc/usbmount/usbmount.conf.   In here there is a line called FILESYSTEMS="". Only filesystems specified in this line are mounted via usbmount, so we change it to : FILESYSTEMS="vfat ntfs fuseblk ext2 ext3 ext4 hfsplus"  If you want usbmount to mount NTFS Filesystems, be sure to add **ntfs** and **fuseblk** to the line. NTFS devices are sometimes listed as fusblk by the ntfs-3g package, so those two are the same. I don't really know why though.  Also, if you don't add **fuseblk** in the configuration, unplugged NTFS Devices will **not** automatically unmount. They will stay mounted and keep the folder they are mounted to until the device is manually unmounted with unmount /dev/sda1 --> where sda1 could be another variable.  The next important line is FS\_MOUNTOPTIONS="". Here you specify which filesystems should be mounted and how they should be mounted.  We change it to: FS\_MOUNTOPTIONS="-fstype=ntfs-3g,nls=utf8,umask=007,gid=46 -fstype=fuseblk,nls=utf8,umask=007,gid=46 -fstype=vfat,gid=1000,uid=1000,umask=007"  With this the filesystems **vfat** (fat32) **ntfs-3g** (NTFS), and **fuseblk** (NTFS again) are mounted. I think most parameters can stay the same. Again here just add ntfs-3g (**add ntfs-3g not ntfs**) and fuseblk to be able to mount all NTFS Files. For more filesystems add more lines starting with -fstype=. **Keep NTFS mounted untill device is unplugged** Here is a solution for the problem, when the mounted NTFS drive is only accessible for a few seconds. This fix comes from **Christian Weinberger**.   1. Create the file **usbmount.rules** in **/etc/udev/rules.d/** with sudo nano /etc/udev/rules.d/usbmount.rules.   This is the content:  KERNEL=="sd\*", DRIVERS=="sbp2", ACTION=="add", PROGRAM="/bin/systemd-escape -p --template=usbmount@.service $env{DEVNAME}", ENV{SYSTEMD\_WANTS}+="%c"  KERNEL=="sd\*", SUBSYSTEMS=="usb", ACTION=="add", PROGRAM="/bin/systemd-escape -p --template=usbmount@.service $env{DEVNAME}", ENV{SYSTEMD\_WANTS}+="%c"  KERNEL=="ub\*", SUBSYSTEMS=="usb", ACTION=="add", PROGRAM="/bin/systemd-escape -p --template=usbmount@.service $env{DEVNAME}", ENV{SYSTEMD\_WANTS}+="%c"  KERNEL=="sd\*", ACTION=="remove", RUN+="/usr/share/usbmount/usbmount remove"  KERNEL=="ub\*", ACTION=="remove", RUN+="/usr/share/usbmount/usbmount remove"   1. Create the file **usbmount@.service** in **/etc/systemd/system/** with sudo nano /etc/systemd/system/usbmount@.service.   This is the content:  [Unit]  BindTo=%i.device  After=%i.device  [Service]  Type=oneshot  TimeoutStartSec=0  Environment=DEVNAME=%I  ExecStart=/usr/share/usbmount/usbmount add  RemainAfterExit=yes  Now reboot and check with cat /etc/mtab to which folder the usb devices are mounted to. By Default they are mountet to **/media/usbstick0**. |

# AutoStart Script

Open crontab

sudo crontab -e

insert this line:

@reboot sh /home/pi/RPI\_USBCopier/scripts/launcher.sh /home/pi/RPI\_USBCopier/logs/startuplog.txt

Finally reboot:

sudo reboot