Mailbox Activity

- Raspberry Pi-zero W Activity Notifications
  - Mailbox Notifications

Raspberry Pi-zero W Activity Notifications

Mailbox Notifications

https://www.youtube.com/embed/FS72ToXLkE8

Last year I started thinking about making this little gadget to let me know when the postal worker drops my mail in the box. My mailbox is about 40 ft (~13 m) from my front door, and down a hill, so my Ring doorbell doesn't always catch it when it's dropped - though it does seem to catch every other car on earth going by...

Anyway, rather than walk down to the mailbox multiple times a day looking for mail I though it would be easier if the mailbox notified me when someone opened, then closed it.

That's where my gadget comes in.

The Build

Parts:

- Raspberry Pi Zero W (though any wireless capable version should work)
- SD Card (at least 8 GB, though I suggest 15 GB)
- Battery Power Supply that can run a Pi Zero W
- Micro USB cable you can cut to expose the power wires.
- Magnetic Switch with option to run Normally Open and Normally Closed

Software you'll need

- this repository (specifically the .sh and .service files).
- Ntfy - a cli tool for sending notifications to various services.
  https://ntfy.readthedocs.io/en/latest/
- Raspberry PI OS Lite (no UI, just cli)
  https://downloads.raspberrypi.org/raspios_lite_armhf/images/raspios_lite_armhf-2021-03-25/2021-03-04-raspios-buster-armhf-lite.zip
- Optionally (Angry IP Scanner) https://angryip.org/download/#linux

The Process

1. Install Raspberry Pi OS lite on your SD card. Use a tool like Balena Etcher to do this.
2. Setup your PI to have wireless access to your network, and make sure to give yourself SSH access.

a. Wifi Access - You need to create a wpa_supplicant.conf file in the "boot" folder of the SD Card before you put it in the PI for the first time.

wpa_supplicant.conf

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=<your two character country code, e.g. UK, US, etc>

network={
     ssid="<your wireless ssid (network name)>"
     psk="<your wireless network password>"
     key_mgmt=WPA-PSK
}
```

b. SSH Access - Create an empty file simply named "SSH" in the "boot" folder of the SD Card before you use it in the PI for the first time.

3. Eject your SD Card from your computer, and put it in the pi, and boot. Give it about 3 minutes to boot, then find the IP for the Pi on your network. You can use a tool like AngryIPScanner for this, or check your Wireless router control software

4. ssh into your Pi, `ssh pi@<your local ip>`, and enter the password 'raspberry'.

5. Once logged in, you can run `sudo raspi-config` and change any configs you want from there.

6. Once logged in, type `passwd` into the terminal. Enter the current password 'raspberry', and then enter a new password twice to change the password for accessing your pi.

7. Now let's create our script.

`nano mailbox-activity-script.sh`

paste the following into that file, then save it with CTRL + O, press Enter to confirm, and exit nano with CTRL + X.

```
#!/bin/bash
```

```
# first check if wireless is connecte
while [ iwgetid = "" ]; do
    echo "No Wifi"
done

# when wifi is connected it will exit the loop above and now we'll send our message
ntfy -b telegram send "Mailbox Opened!"

# we'll give the device 30 seconds to try and get the message sent
sleep 30s

# and now shut it down til the next time the mailbox is opened
sudo shutdown -h now
```

8.Now let's also create our service for systemd to run when the pi starts up.

`sudo nano /etc/systemd/system/sendMailboxAct.service`

```
[Unit]
Description=Send Messag for Mailbox Acitivity
Wants=network.target
After=network.target

[Service]
Type=oneshot
ExecStartPre=/bin/sh -c 'until ping -c1.google.com; do sleep 1; done;'
ExecStart =/home/pi/mailbox-activity-script.sh
user=pi

[Install]
WantedBy=multi-ser.target
```

9. Save the file with CTRL + O, then press Enter to confirm, and exit the nano editor with CTRL + X

10. Install Ntfy on your pi.

11. Open the mailbox_acitivy_script.sh file and make sure to setup Ntfy to use the service you prefer (if not Telegram).

You can test the send using Ntfy by just copying and pasting the command into an empty terminal

Once working, and sending the message, you can save it to your .sh file.

12. You may need to change the permissions to make it executable `sudo chmod +x /etc/systemd/system/sendMailboxAct.service`

13. Start the service with `sudo systemctl start sendMailboxAct`

NOTE: be ready, when it starts, it will send a message and shutdown the pi.

14. Now setup your pi to work off of the magnetic switch, so when it closes (the two magnets come together) the pi starts, sends the message then shuts back down.

Support My Channel and Content

Support my Channel and ongoing efforts through Patreon:
https://www.patreon.com/awesomeopensource