

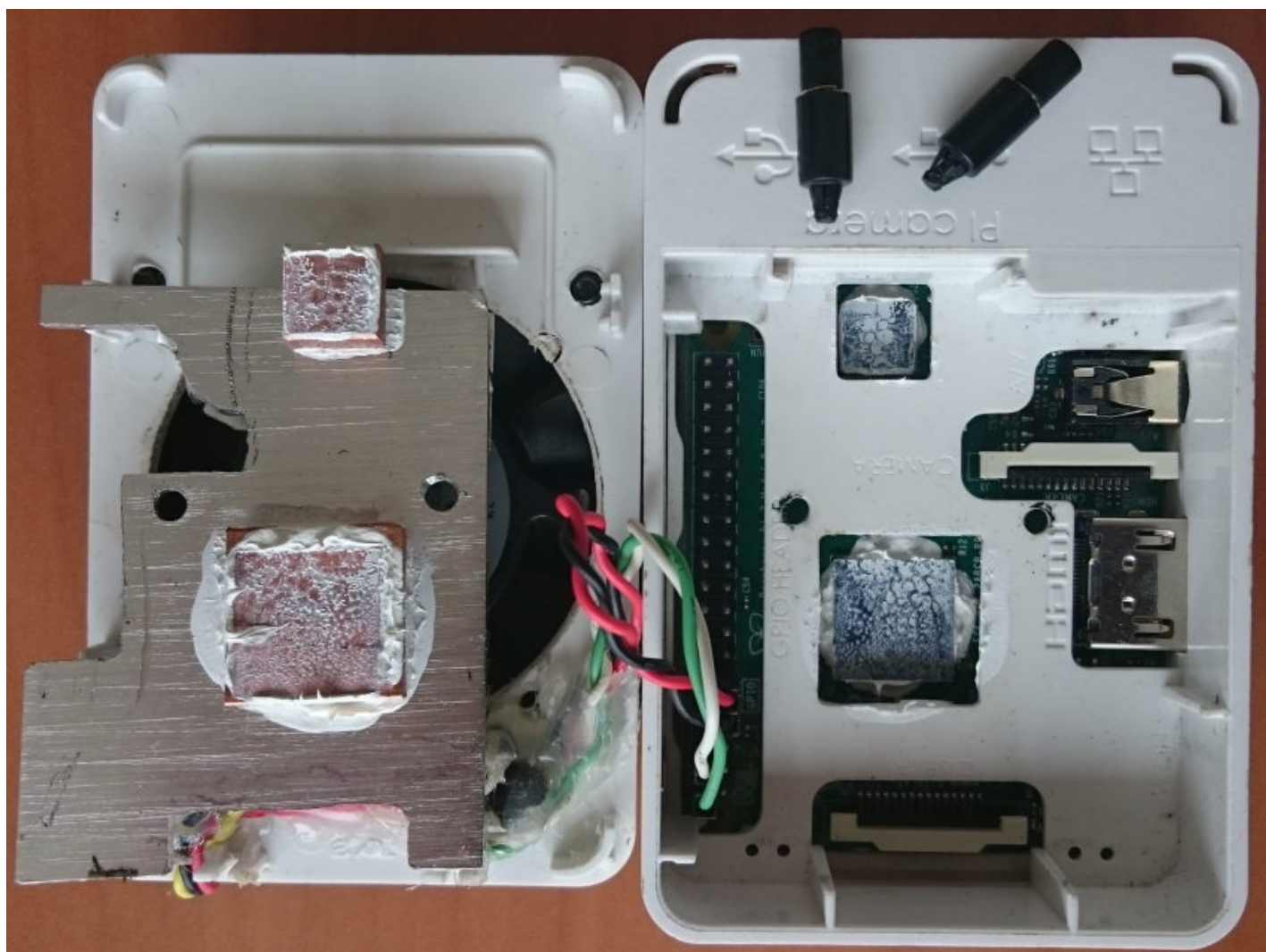
Návod na přetaktování Raspberry P3 a jeho aktivní chlazení

Nová Malina Pi3 již poskytuje dostatečný výkon pro běh serveru a nebo pracovní stanice a tak jsem se rozhodl, že na ní vyzkouším Raspbian. Po instalaci přes NOBS mě nemile překvapil fakt, že výkon systému měl potíže s plynulým přehráváním HD videa a procesor ukazoval takt 600MHz. Rozhodl jsem se jí tedy krapet poladit, což se i povedlo. Dostal jsem se až 1350MHz a 500MHz frekvence jádra. Bohužel při této rychlosti se procesor během krátké chvíle dostal k teplotě 90°C a automaticky si snížil takt na 600MHz, tudíž se opět snížil výkon. Rozhodl jsem se tedy pro pořádné pasivní chlazení, jež pomohlo trvale snížit teplotu na 75°C. Přidal jsem k tomu ještě ventilátor a už jsem byl v pohodě do 50°C a systém svištěl plynule bez jakýchkoli záškubů. Malina má ale GPIO výstupy, tak proč by měl ventilátor funět, když běžně je procesor na 35°C ? Tak jsem k tomu ještě přidal optočlen a jeden malý script do procesů, který to celé řídí.

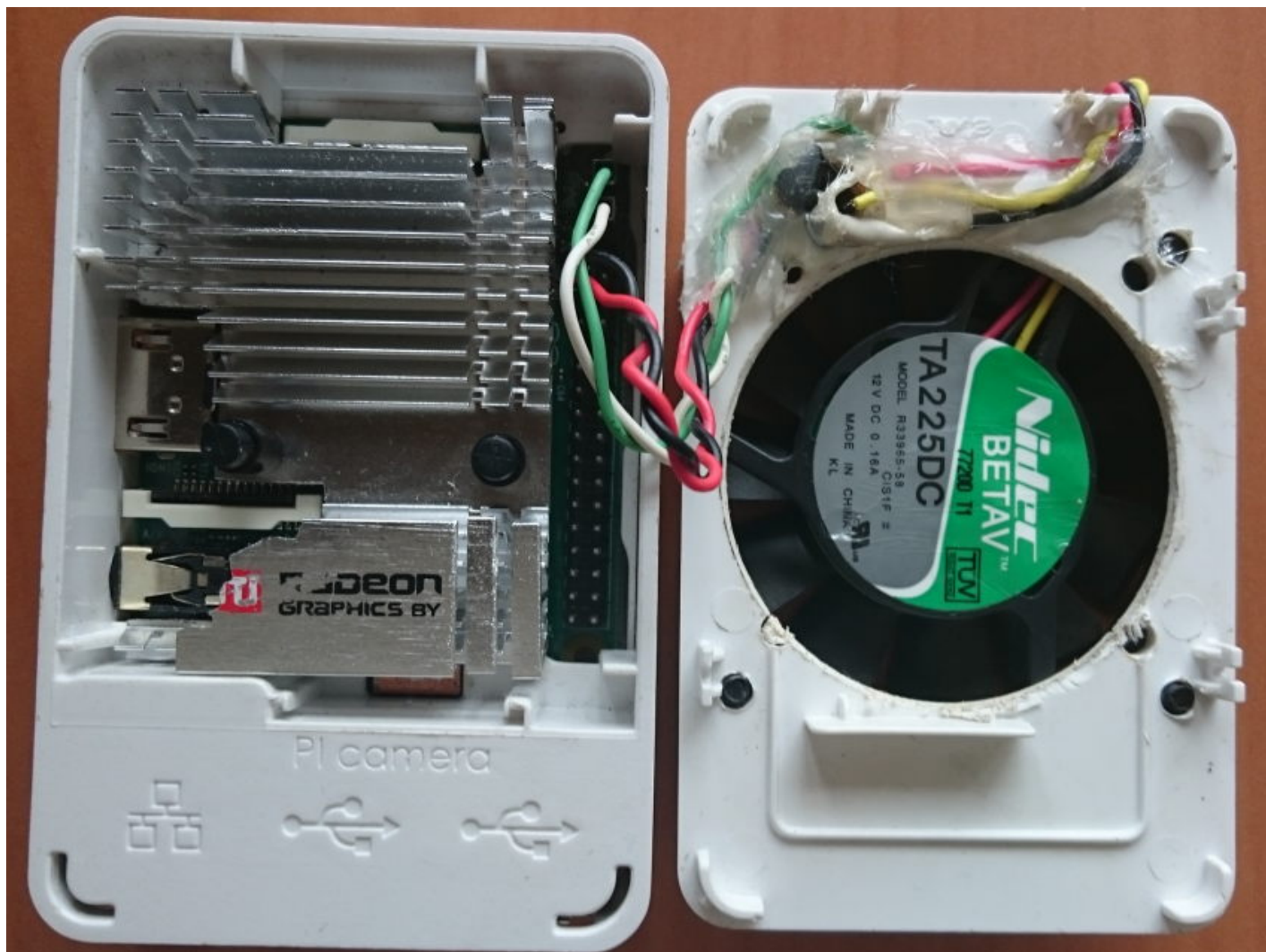
Hardwarová úprava



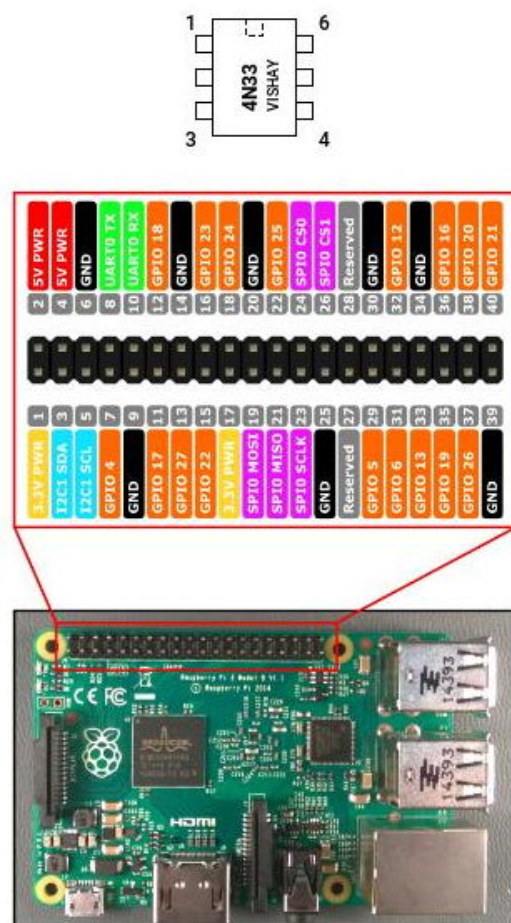
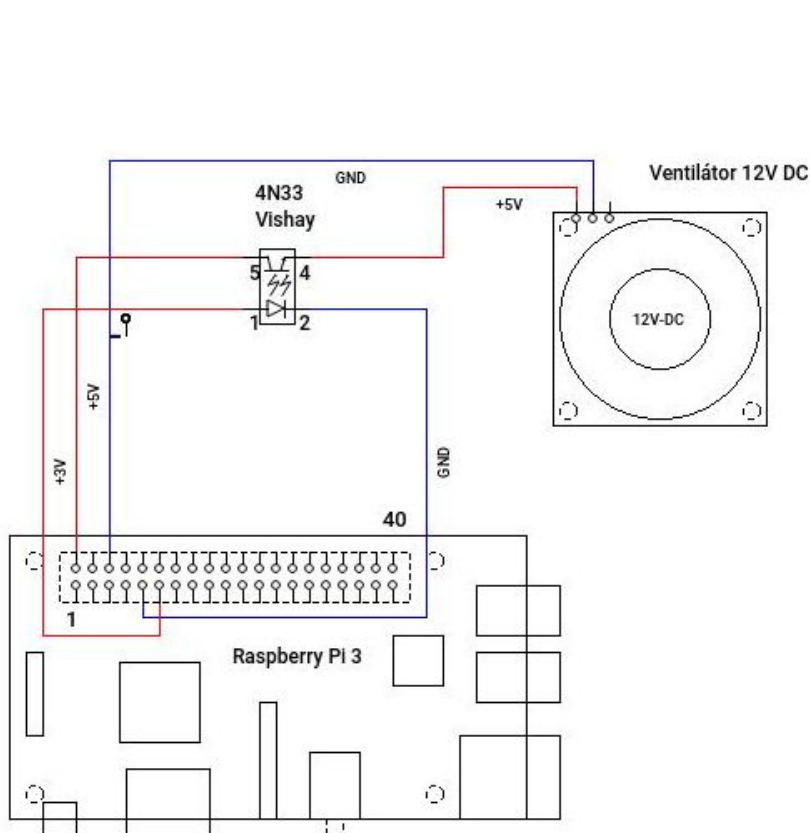
Jako pasivní chladič, jsem použil chlazení ze staré ATI grafiky, jež jsem krapet upravil, aby nepřekážel k přístupu ke konektorům. Kabele s koncovkami pro GPIO rozhraní a ventilátor jsou ze starého PCčka. Do víčka krabičky Maliny jsem vykroužil otvor pro ventilátor a odvrtal 4 díry pro uchycení.



Do středního dílu krabičky jsem vyvrtal dva otvory pro uchycení chladiče. Oba čipy Maliny jsou pod úrovní prostředního dílu krabičky, takže je potřeba vyříznout z měděné pásovinu dvě distance, aby měly čipy kontakt s chladičem.



Ventilátor je na 12V, ale 5V z Maliny bohatě stačí k jeho napájení a díky sníženým otáčkám je téměř nehlukný a oba čipy stačí uchládit. Výstup na GPIO17 má 3V, takže je potřeba zvolit optočlen s 3V diodou (většina běžných optočlenů má 5 nebo 6V na vstupu). Pro tento účel jsem zvolil optočlen 4N33, jež mají v gme.cz za 13,- Kč. Malina pak na základě vyhodnocení teploty procesoru přes optočlen spíná ventilátor, jenž dochlazuje chladič.



Software - řízení ventilátoru dle teploty

Aby se dalo z Raspberry dostat maximum, je třeba si pohrát se souborem config.txt, jež se nachází v adresáři boot . Každá Malina je ale z výroby jiná a tak je potřeba si s jednotlivými hodnotami trochu pohrát, aby byl systém stabilní při maximálním výkonu. Pokud se to přežene, bude systém zamrzat a padat. Zde můj upravený

boot/config.txt

```
#uncomment to overclock the arm. 700 MHz is the default.
```

```
force_turbo=1
dtparam=sd_overclock=100
arm_freq=1350
core_freq=500
over_voltage=6
sdram_freq=500
sdram_schmoo=0x02000020
over_voltage_sdram_p=6
over_voltage_sdram_i=4
over_voltage_sdram_c=4
v3d_freq=500
h264_freq=333
gpu_mem=256
```

Dále je potřeba vytvořit scrip, jež bude ovládat ventilátor v závislosti na teplotě procesoru. Mě se osvědčilo zapínat ventilátor, pokud teplota procesoru přesáhne 50°C a a jeho vypnutí při ochlazení pod 40°C. Lze ho napsat pod čímkoli a na tak jednoduchý program pohodlně stačí bash.

```
/home/pi/vetrak.sh
```

```
#!/bin/bash
echo "17" > /sys/class/gpio/export # aktivuje GPIO17
echo "out" > /sys/class/gpio/gpio17/direction #nastavi GPIO17 jako vystup
```

```

while [ "1" = "1" ] #nekonecna smycka
do
TEPLOTA=`vcgencmd measure_temp | cut -c 6-7` #zjistí teplotu procesoru
VETRAK=`cat /sys/class/gpio/gpio17/value`

if [ $TEPLOTA -lt 40 ] && [ $VETRAK == 1 ]; then #vypne chlazení pokud je teplota menší než 40
echo 0 > /sys/class/gpio/gpio17/value
fi

if [ $TEPLOTA -gt 50 ] && [ $VETRAK == 0 ]; then #zapne ventilátor pokud je teplota vyšší než 50
echo 1 > /sys/class/gpio/gpio17/value
fi
sleep 30s # čeká 30s
done

```

No a nakonec je potřeba zajistit, aby se scrip automaticky spouštěl po startu systému. Dá se to přidat do procesů, ale mě to přišlo přehlednější vložit do rc.local

/etc/rc.local

```

#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
printf "My IP address is %s\n" "$_IP"
fi
# autostart script
nohup /home/pi/.vetrak.sh &

exit 0

```