

Fuglaborð

Efnisyfirlit:

[Leiðbeiningar við uppsetningu á RPi.](#)

[Í fartölvu:](#)

[Í RaspberryPi:](#)

[Lýsing á fuglaborðinu:](#)

[Forritið og staðsetning:](#)

[Láta sjálfkrafa kveikja á forriti þegar RPi er ræst \(boot up\).](#)

[Græja sjálfvirkja endurræsinguna einu sinni í viku.](#)

[Ræsa forrit.](#)

[Tengjast við skjálausa \(headless\) RPi með router og netsnúru.](#)

[Tengjast við skjálausa \(headless\) RPi með LAN-snúru.](#)

[Finna IP-tölu á skjálausri \(headless\) RPi.](#)

[Færa hluti á milli tölva með "SSH".](#)

["GUI" \(Graphical User Interface\) over "SSH" \(X-forwarding\).](#)

[Sækja hljóð af netinu.](#)

[Almennar upplýsingar um RPi smá tölvurnar.](#)

[RaspberryPi tegundir og hvernig þú finnur þína.](#)

[Upplýsingar um "GPIO" pinnana.](#)

[Fuglabord_Launcher.sh v1.1](#)

[Fuglabord_main.py v1.1](#)

Örvar Sigþórsson
866-1428

Vatnajökulspjóðgarður
Gljúfrastofa

Efni:

- RaspberryPi tölvu. (RPI - 1, rev2)
- SDHC Memory card, Class:10.
- 5V Spennubreytir fyrir RPi. (Micro-USB, 800mA eða meira)
- 8 stk. 12VDC relay og straumskinnu.
- 12V Spennubreytir fyrir relay og skynjara.
- Vírar og tengi fyrir GPIO pinna á RPi.
- 230V Hátalara með AUX-snúru.
- Raðtengi fyrir plús og mínus.
- DIN-skinnu.

Þarf að gera:

- Setja upp stýrikerfið á RPi. Stýrikerfið sem er notað er Linux-distribution sem heitir Raspbian.
- Setja upp Python3 og PIP.
- Setja upp VLC-module með PIP.
- Sækja RPi.GPIO module fyrir Python3.
- Finna fugla hljóðin.
- Búa til Python forrit sem les fuglaborðið og spilar hljóð.
- Sjá til þess að Python forritið fari sjálfkrafa í gang.
- Halda utan um hvað er að gerast í "log" fælum.
- Nota Crontab til að endurræsa RPI einu sinni í viku.
- Tengja 12VDC skynjara við relay.
- Tengja relay við 5VDC GPIO pinnana á RPi.

Kunnátta:

- Kannast við RaspberryPi smá tölvurnar og GPIO pinnana.
- Kannast við Linux og hvernig terminal-ið er notað.
- Kunna aðins að forrita, mikill kostur að þekkja Python3 og Bash.
- Skilja rafmagnsfræði eða rafeindafræði.

Leiðbeiningar við uppsetningu á RPi.

Þessi tegund af RPi (RaspberryPi) sem ég er að nota er:

“Raspberry Pi-1 rev2, 26-pin (China) 000f.”

(Það eru upplýsingar um tegundir neðar)

Stýrikerfið sem ég nota fyrir RPi heitir:

“Raspbian”

Í fartölvu:

Það þarf aðra tölvu til að setja upp RPi, ég er að nota “Linux” fartölvu.

Í fartölvunni: fara inná: <https://www.raspberrypi.org/downloads/> og niðurhala útgáfunni af “Raspbian” sem hefur sem mest fyrirfram installað eða innbyggt, núna árið 2020 mæli ég með:

“Raspbian Buster with desktop and recommended software”.

Þegar niðurröðunin er búin gæti fællinn verið “zip”aður, “un-zip”aðu honum eða “extract from archive” eins og það kallast í Linux. Þá ertu vonandi kominn með “img” file eða einhverskonar “image” file.

Fyrir RPi tölvuna sem ég er að notaði þarf “SDHC Memory card”, að minnsta kosti “Class:10” eða eitthvað nógu hraðvirkt kort til að setja stýrikerfið upp á.

Tengdu kortið við fartölvuna og formattaðu kortið sem “EXT4”(eða “FAT”) og **brenndu** (ekki copy-a) svo “image” file-inn inn á það.

Ég notaði forrit sem heita: “USB Stick Formatter” og “USB Image Writer”.

“USB Stick Formatter” is part of the “mintStick” package available by default in “Linux Mint” and it can easily be installed in “Ubuntu”. The “mintStick” package also includes “USB Image Writer”, an utility to create a bootable USB stick from ISO or IMG files.

To install: >> sudo apt-get install mintstick

Þá er komið “bootable” fresh-install af “Raspbian” inn á SD-kortið.

Svo bara setja SD-kortið í RPi, tengja lyklaborð, mús, skjá og netsnúru beint við internet router og svo endá á 5 volta Micro-USB spennugjafanum (hann þarf að vera 800mA eða meira). Þá ræsir RPi sig og keyrir up “Raspbian” í fyrsta skipti, það getur tekið **10 min** og skjárin verður svartur á meðan, svo þolinmæði!

Svo er annar möguleiki, það er að nota “ssh” og tengjast við RPi úr annari tölvu í gegnum net eða LAN tengingu. Þá þarf ekki að tengja skjá, lyklaborð né mús við RPi. Þetta er útskýrt hér fyrir neðan í köflum sem heita:

“Tengjast við skjálausu (headless) RPi með router og netsnúru”.

“Tengjast við skjálausu (headless) RPi með LAN-snúru”.

Í RaspberryPi:

Þegar RPi er búin að ræsa sig, færum við okkur yfir í hana og vinnum í henni, hér eru smá upplýsingar:

Default username: pi

Default password: raspberry

Það gæti verið að það þurfi að stilla á íslenskt lyklaborð þegar RPi er alveg búin að ræsa sig, það er undir “Preferences”>”Keyboard and Mouse”>”Keyboard Layout”>”Layout” og velja “Icelandic”.

Það er mjög mikilvægt að skipta um password, sérstaklega ef RPi verður tengd við internetið í lengri tíma og “SSH” er virkjað. Til að skipta um password þarf að opna “terminal”ið (Ýttu samtímis á ‘Ctrl’+‘Alt’+‘T’) og skrifaðu eftirfarandi skipanir:

```
>> sudo passwd -d pi
```

og svo:

```
>> sudo passwd
```

Ég valdi passwordið: “**okaypi**”, þú getur haft það eitthvað annað en ef það glatast og gleymist er stýrikerfið á RPi sama og ónýtt.

Næsta verk að uppfæra hugbúnaðinn, opna “terminal”ið og skrifaðu:

```
>> sudo apt-get update
```

og svo:

```
>> sudo apt-get upgrade
```

Þetta uppfærir allan hugbúnaðinn í tölvuni.

Svo þarf að virkja “SSH”, það er undir “Preferences”>”Raspberry Pi Configuration”>”Interfaces” og velja “Enabled” við “SSH”. Þar þarf að endurræsa RPi til að virkja þessa breytingu.

(Það eru upplýsingar um “SSH” neðar.)

Forritunar tungumálið “Python3” á að vera fyrirfram installað á RPi, til að athuga það:

```
>> python3 -V
```

og:

```
>> pip -V
```

Ef við fáum eitthvað svipað og: “Python 3.7.3” og “pip 18.1 from /usr/l...” er allt í góðum málum.

Svo þarf að sækja “VLC” module fyrir “Python3” með “pip3” sem heitir “python-vlc”:

```
>> sudo pip3 install python-vlc
```

Muna að stilla hljóð á “Analog” enn ekki “HDMI” í RPi. (hægrismella á hátalara icon uppi í hægra horni.)

Svo þurfum við “library” sem gerir okkur kleift að stjórna “GPIO”-pinnunum með Python kóða. Til að installa því með terminal-inu:

```
>> sudo apt-get install python-rpi.gpio
```

Þetta er “module” sem heitir “RPi.GPIO” í Python kóða.

Lýsing á fuglaborðinu:

Nú er komið að forritinu okkar, hér er smá lýsing um hvað forritið á að gera:

Forritið byggist á Python kóða sem les stöðuna á “GPIO”-pinnunum á RPi. “GPIO”-pinnarnir eru tengdir við átta “**relay**”(spólurofa) sem fá boð frá átta skynjurunum sem lesa stöðuna á fjórum hjólum, á hjólunum eru myndir og getur fólk snúið hjólunum. Þegar öll hjólin passa sama við einn af fjórum fuglum, spilast viðeigandi fugla-hljóð.

Forritið og staðsetning:

Það er hægt að finna Python kóðann og tæknilegar upplýsingar um hann á “Github”:

<https://github.com/xxORVARxx/Fuglabord>

Python forritið er í fæl sem heitir “Fuglabord_main.py”.

Það er hægt að ‘copy’ og ‘paste’ innihaldið af Github yfir í nýjan fæl. Bara að passa að fyrsta línan í fælnum sé: “**#!/usr/bin/env python3**” og að hann heiti “Fuglabord_main.py”. Það þarf líka að gera hann “executable”, það er hægt í terminal-inu með þessari skipun:

```
>> sudo chmod +x path/to/the_file
```

Þegar við erum komin með Python kóðann og hann er í “executable” fæl sem heitir “Fuglabord_main.py” skulum við staðsetja hann í RPi. Ég mæli með að setja hann hér: “~/Project/Fuglabord” þetta er sami staður og “/home/pi/Project/Fuglabord”

Við getum notað terminal-ið til að búa til þessar möppur:

```
>> mkdir -p ~/Project/Fuglabord
```

Fuglahljóðin fara í sömu möppu, Þetta eru þá fælarnir sem þarf:

```
/home/pi/Project/Fuglabord/audnutittlingur.ogg  
/home/pi/Project/Fuglabord/falki.ogg  
/home/pi/Project/Fuglabord/Fuglabord_main.py  
/home/pi/Project/Fuglabord/stelkur.ogg  
/home/pi/Project/Fuglabord/stokkond.ogg  
/home/pi/Project/Fuglabord/Fuglabord_Launcher.sh ← Þessi fæll er útskýrður hér fyrir neðan.
```

Það er mjög mikilvægt að fugla hljóðin heiti réttum nöfnum, sem eru:

“audnutittlingur.ogg”, “falki.ogg”, “telkur.ogg” og “stokkond.ogg”

NOTE: Í Linux skipta stórir og litlir stafir miklu máli! ‘fálki’ og ‘Fálki’ eru ekki sami fælinn.

Til að útvega fuglahljóðin, gæti kafli hér fyrir neðan hjálpað til:

“Sækja hljóð af netinu”.

Inn á Github er svo annar fæll sem heitir “Fuglabord_Launcher.sh”. Þetta er lítið “Bash script” sem sér um að kveikja á Python forritinu(Fuglabord_main.py) og halda því gangandi. (Python forritið er einfaldlega inní endalausri loopu).

Eins og með Python fælinn er hægt að ‘copy’ og ‘paste’ innihaldið af Github og gera hann “executable”. Nema núna þarf fællinn að byrja á “**#!/bin/bash**”.

Þetta “Bash script” sér líka um að vera “logger” og halda uppi “log” fælunum sem munu verða staðsettir hér: “/home/pi/Project/Fuglabord/_logger_m_y.txt”.

Láta sjálfkrafa kveikja á forriti þegar RPi er ræst (boot up).

Ég nota forrit sem heitir '**crontab**' og er oftast innbyggt í Linux.

*NOTE: Þegar maður er að vinna með "**crontab**" er ekki gott að finna "**cron**" fælinn sjálfur og opna hann með einhverjum standard "**texteditor**" eins og "**Emacs**" eða "**Nano**" til að vinna á honum, það þarf að nota "**crontab**" terminal skipanirnar og opna fælinn í gegnum þær.*

Með "**crontab**" er hægt er gera allavega sjálfvirk tímaplön og eins getur maður stjórna hvað gerist við ræsinguna (boot up). Þetta er terminal skipunin til að byrja:

```
>> sudo crontab -u pi -e
```

(*'pi'* er user-name, *-u* er sudo, *-e* er edit)

Þú gætir verið beðinn um að velja þér "**texteditor**" eins og "**Emacs**" eða "**Nano**", þegar þú ert búinn þá opnast "**cron**" fælinn. '**Copy**' og '**Paste**'aðu eftirfarandi inní fælinn og vistaðu:

```
# To Run Commands On Boot Up, Use: @reboot <command to be executed>
```

```
directory="/home/pi/Project/Fuglabord/"
```

```
@reboot ${directory}Fuglabord_Launcher.sh ${directory}
```

```
# Kveikum á forriti við ræsing.
```

```
# To Run Schedule Based Job:
```

```
# .----- minute (0 - 59)
```

```
# | .----- hour (0 - 23)
```

```
# | | .----- day of month (1 - 31)
```

```
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
```

```
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7)
```

```
# | | | | |
```

```
# * * * * * <command to be executed>
```

```
30 3 * * 7 sudo reboot
```

```
# Klukkan 3:30 eftir miðnætti á sunnudögum 'reboot'um við.
```

```
# NOTE: Ef það væri '*' fyrir mínútur, myndi reboot-ast stanslaust
```

```
# í klukkutíma sem er líklega ekki ætlunin!!
```

Græja sjálfvirkja endurræsinguna einu sinni í viku.

Það er líka gert með "**crontab**" og er seinni hlutinn í gráa textanum hér fyrir ofan sem sér um það.

Ræsa forrit.

Eins og ég setti þetta upp sér "**crontab**" um að ræsa forritið og allt er sjálfvirk, en ef þú vilt ræsa forritið handvirk geturðu notað þessa skipun:

```
>> sudo /home/pi/Project/Fuglabord/Fuglabord_Launcher.sh "/home/pi/Project/Fuglabord/"
```

og ef þú vilt bara ræsa Python hlutann:

```
>> sudo /home/pi/Project/Fuglabord/Fuglabord_main.py "/home/pi/Project/Fuglabord/"
```

Tengjast við skjálausa (headless) RPi með router og netsnúru.

Ef við viljum tengjast RPi í gegnum router og netsnúru og stjórna henni þannig:

Tengja fartölvu við sama router, ég nota Linux fartölvu og forrit sem heitir “SSH”.

Fyrst þarf að finna IP-töluna á RPi, það er útskýrt hér fyrir neðan í kafla sem heitir:

“Finna IP-tölu á skjálausri (headless) RPi”.

Þegar við erum komin með IP-töluna og tilbúin að tengjast við RPi, opnum terminal-ið og skrifum: (skipta út x fyrir IP-tölu á RPi)

```
>> sudo ssh pi@x.x.x.x
```

Í fyrsta skipti fáum við *security/authenticity* viðvörðun og ‘yes’ ‘no’ spurningu, skrifaðu ‘yes’.

Svo er beðið um fartölvu password-ið eða RPi password-ið, það er: “raspberrypi” eða “okaypi” eða það sem þú bjóst til.

Ef IP-talan og password-ið er rétt ætti að standa í terminal-inu:

```
pi@raspberrypi:~
```

Það þýðir að þetta er terminal-ið í RPi enn ekki í fartölvunni. Til að fara til baka í fartölvuna ýttu samtímis á: ‘Ctrl’+‘D’ og ‘Ctrl’+‘C’.

Tengjast við skjálausa (headless) RPi með LAN-snúru.

<https://www.youtube.com/watch?v=8qleH35Kgjk>

Ef við erum ekki með router getum við tengt netsnúru/LANsnúru bara beint á milli RPi og fartölvunar, en fyrst þurfum við að búa til nýja LAN-ethernet tengingu til að nota.

Finndu net stillingar í fartölvunni, sem sagt lista yfir allar net tengingarnar sem eru í boði. Heitir oft “Connection editor” eða “Network Connections”, og finnur hann oft með því að hægri-klikka á internet iconið á desktop-inu.

Þegar við erum komin þangað, klikkum þá á plúsinn ‘+’ eða ‘Add’ og veljum svo “Wired Ethernet” eða “Ethernet”. Byrjum á að breyta nafninu í eitthvað gáfulegt eins og: “RPi-LAN”, stillum svo undir “IPv4” eitthvað sem er oft kallað “Method” í “Share to other computers”. Förmum svo yfir í “IPv6” og breytum þar “Method” í “Ignored”.

Núna getum við tengt LAN-snúruna á milli RPi og fartölvunar, stillum svo á nýja ‘RPi-LAN’ networkið okkar ef það gerist ekki að sjálfum sér.

Svo þarf að finna IP-töluna á RPi, það er útskýrt hér fyrir neðan í kafla sem heitir:

“Finna IP-tölu á skjálausri (headless) RPi”.

Þegar við erum komin með IP-töluna og tilbúin að tengjast við RPi, opnum terminal-ið og skrifum: (skipta út x fyrir IP-tölu á RPi)

```
>> sudo ssh pi@x.x.x.x
```

Í fyrsta skipti fáum við *security/authenticity* viðvörðun og ‘yes’ ‘no’ spurningu, skrifaðu ‘yes’.

Svo er beðið um fartölvu password-ið eða RPi password-ið, það er: “raspberrypi” eða “okaypi” eða það sem þú bjóst til.

Ef IP-talan og password-ið er rétt ætti að standa í terminal-inu:

```
pi@raspberrypi:~
```

Það þýðir að þetta er terminal-ið í RPi enn ekki í fartölvunni. Til að fara til baka í fartölvuna ýttu samtímis á: 'Ctrl'+ 'D' og 'Ctrl'+ 'C'.

Finna IP-tölu á skjálausri (headless) RPi.

Tengja RPi við router eða aðra tölvu með netsnúru.

Tengja fartölvu við sama router, ég nota Linux fartölvu og forrit sem heitir "**nmap**". Til að installa:

```
>> sudo apt install nmap
```

Okay, ip-tölur eru fjórar tölur aðskildar með púnkti, til dæmis: "**192.168.1.5**"

RPi er með sömu þrjár fyrstu tölur og fartölvun, sem sagt: "**192.168.1.?**"

Svo við verðum að finna síðustu töluna í IP-töluni á RPi.

Við byrjum á að finna IP-töluna á fartölvunni, til að gera það, opna terminal og skrifa:

```
>> hostname -I
```

Það gætu birst tvær IP-tölur, ein fyrir LAN (td: **10.42.0.1**) og önnur fyrir internet (td: **192.168.1.5**).

Svo leitum við á öðruhvorum network-inu af öllum IP-tölum sem byrja á því sama: (skiptu út x fyrir fartölu IP-töluna í skrefinu fyrir ofan)

```
>> sudo nmap -sn x.x.x.0/24
```

Við fáum lista af öllum tækjum sem eru á þessu networki, RaspberryPi ætti að vera í listanum og IP-talan sem okkur vantar þar fyrir **ofan**.

Til að vera viss um að við séum með IP-töluna á RPi, getum við sett hana í "**nmap**" og við ættum að sjá nafnið RaspberryPi í útkomunni: (skipta út x fyrir IP-tölu á RPi)

```
>> sudo nmap x.x.x.x
```

Færa hluti á milli tölvu með "SSH".

Að finna IP-töluna á RPi er útskýrt í kafla sem heitir:

"Finna IP-tölu á skjálausri (headless) RPi".

Syntax: **scp <source> <destination>**

To copy a file from 'B' to 'RPi' while logged into 'B':

```
>> scp -rp /path/to/file pi@x.x.x.x:/home/pi/destination
```

To copy a file from 'B' to 'RPi' while logged into 'RPi':

```
>> scp -rp username@b:/path/to/file /path/to/destination
```


“GUI” (Graphical User Interface) over “SSH” (X-forwarding).

Að finna IP-töluna á RPi er útskýrt í kafla sem heitir:

“Finna IP-tölu á skjálausri (headless) RPi”.

You can forward your X-session over SSH, to allow the use of graphical applications, by using the -Y flag:

```
>> ssh -Y pi@x.x.x.x &
```

Now you are on the command line (or terminal) as before, but you have the ability to open up graphical windows, for example:

```
>> emacs path/to/file.py
```

or:

```
>> thonny path/to/file.py
```

Will open up the editor in a graphical window.

Sækja hljóð af netinu.

Opna netsíðu og spila hljóðið.

Hægri-klikka og velja “Inspect” -> “Network”, Reload-a svo síðunni og þá ættu hlutir að týnast inn, ef hljóðið er að spilast þá ætti það að vera í þessum lista, það þarf bara að finna það.

Líklega “mp3” eða “ogg” fæll eða eitthvað svoleiðis.

Þegar fællinn er fundin í listanum, hægri-klikka á hann og velja, “Open in new tab”, fara svo yfir í þann tap og þar er líklega hægt að hægri-klikka og velja “Save as..”.

Almennar upplýsingar um RPi smá tölvurnar.

https://elinux.org/RPi_Low-level_peripherals

Power:

Pin 1 provides 3.3V and is limited to 50mA max. Pin 2 provides 5V and draws current directly from your microUSB supply, so it can use whatever is left over after the board has taken its share, around 700mA. For example, a 1A(1000mA) power supply could supply up to 300mA to pin-2 once the board itself has drawn 700mA.

NOTE: Most of the pins in the header go directly to the Broadcom chip. It is important to carefully design the components you attach to them as there is a risk you will permanently damage your Pi. No short circuits and wiring mistakes!

Tengja LED Ljós:

<https://www.raspberrypi-spy.co.uk/2012/06/control-led-using-gpio-output-pin/>

RaspberryPi tegundir og hvernig þú finnur þína.

Finding your Pi Revision Number is a useful technique to identify what model of Pi you have and, in some cases, where it was manufactured. The variants currently available are :

Model and PCB Revision:	RAM:	Hardware Revision Code from cpuinfo:
Model B Rev 1	256MB	2
Model B Rev 1 ECN0001 (no fuses, D14 removed)	256MB	3
Model B Rev 2	256MB	0004, 0005, 0006
Model A	256MB	0007, 0008, 0009
Model B Rev 2	512MB	000d, 000e, 000f
Model B+	512MB	0010, 0013, 900032
Compute Module	512MB	11
Compute Module	512MB	0014 (Embest, China)
Model A+	256MB	12
Model A+	256MB	0015 (Embest, China)
Model A+	512MB	0015 (Embest, China)
Pi 2 Model B v1.1	1GB	a01041 (Sony, UK)
Pi 2 Model B v1.1	1GB	a21041 (Embest, China)
Pi 2 Model B v1.2	1GB	a22042
Pi Zero v1.2	512MB	900092
Pi Zero v1.3	512MB	900093
Pi Zero W	512MB	9000C1
Pi 3 Model B	1GB	a02082 (Sony, UK)
Pi 3 Model B	1GB	a22082 (Embest, China)
Pi 3 Model B+	1GB	a020d3 (Sony, UK)
Pi 4	1GB	a03111 (Sony, UK)
Pi 4	2GB	b03111 (Sony, UK)
Pi 4	4GB	c03111 (Sony, UK)

Methods for Finding RaspberryPi Revision Number:

The Pinout Utility: **>> pinout**

The “cpuinfo” File: **>> cat /proc/cpuinfo**

Finding Raspberry Pi Model Information: **>> cat /proc/device-tree/model**

NOTE: The Revision number given in “cpuinfo” file is the hardware revision number. This is not the same as the Raspberry Pi Rev number sometimes used to describe updated PCBs. In this example I have a Model B “Rev 1.0” with a hardware revision code of 0002.

Upplýsingar um “GPIO” pinnana.

Það er tvær aðferðir til að tilgreina hvaða “GPIO” pinna átt er við. Önnur aðferðin notast við “Pin-number” sem er einfaldlega raunveruleg staðsetning pinnana á RPi prentplötunni, ef prentplatan er skoðuð sést að fyrsti pinninn er merktur annað hvort með litlum kassa og nálægt sjást stafirnir ‘P1’ eða með því að við pinna-eitt vantar hornið á hvítu línuna sem rammur inn “GPIO” pinna.

Hin aðferðin er að nota “GPIO” númerin, þau stjórna af því við hvaða þessi tiltekni pinni tengist við innan prentplötunar og “GPIO” númeri þess pinna getur breyst milli útgáfa á RPi. Í Python kóðanum er hægt að velja hvor aðferðina þú vilt notast við þegar þú ert að vinna með “GPIO”-pinnana, ég ákvað nota “Pin-number” því þá skiptir ekki máli hvaða útgáfu af RPi er notað því víringin á “GPIO” pinnunum er alltaf bara eins, sama hvaða RPi er notuð.

RPI 1, A/B Rev-1 P1-GPIO				
		Pin No.		
Power	3,3V	1 2	5V	Power
I ² C	GPIO0	3 4	5V	Power
I ² C	GPIO1	5 6	GND	Ground
GPIO	GPIO4	7 8	GPIO14	UART
Ground	GND	9 10	GPIO15	UART
GPIO	GPIO17	11 12	GPIO18	GPIO
GPIO	GPIO21	13 14	GND	Ground
GPIO	GPIO22	15 16	GPIO23	GPIO
Power	3,3V	17 18	GPIO24	GPIO
SPI	GPIO10	19 20	GND	Ground
SPI	GPIO9	21 22	GPIO25	GPIO
SPI	GPIO11	23 24	GPIO8	SPI
Ground	GND	25 26	GPIO7	SPI

Pin 1 provides 3.3V and 50mA max.

Þetta er útgáfan af RPi sem ég notaði:

RPi 1 A/B, Rev-2 P1-GPIO					
		Pin No.			
Power	3,3V	1	2	5V	Power
I ² C	GPIO2	3	4	5V	Power
I ² C	GPIO3	5	6	GND	Ground
GPIO	GPIO4	7	8	GPIO14	UART
Ground	GND	9	10	GPIO15	UART
GPIO	GPIO17	11	12	GPIO18	GPIO
GPIO	GPIO27	13	14	GND	Ground
GPIO	GPIO22	15	16	GPIO23	GPIO
Power	3,3V	17	18	GPIO24	GPIO
SPI	GPIO10	19	20	GND	Ground
SPI	GPIO9	21	22	GPIO25	GPIO
SPI	GPIO11	23	24	GPIO8	SPI
Ground	GND	25	26	GPIO7	SPI

Pin 1 provides 3.3V and 50mA max.

RPi 1, B+ J8-GPIO				
		Pin No.		
Power	3,3V	1 2	5V	Power
I ² C	GPIO2	3 4	5V	Power
I ² C	GPIO3	5 6	GND	Ground
GPIO	GPIO4	7 8	GPIO14	UART
Ground	GND	9 10	GPIO15	UART
GPIO	GPIO17	11 12	GPIO18	GPIO
GPIO	GPIO27	13 14	GND	Ground
GPIO	GPIO22	15 16	GPIO23	GPIO
Power	3,3V	17 18	GPIO24	GPIO
SPI	GPIO10	19 20	GND	Ground
SPI	GPIO9	21 22	GPIO25	GPIO
SPI	GPIO11	23 24	GPIO8	SPI
Ground	GND	25 26	GPIO7	SPI
DNC		27 28		DNC
GPIO	GPIO5	29 30	GND	Ground
GPIO	GPIO6	31 32	GPIO12	GPIO
GPIO	GPIO13	33 34	GND	Ground
GPIO	GPIO19	35 36	GPIO16	GPIO
GPIO	GPIO26	37 38	GPIO20	GPIO
Ground	GND	39 40	GPIO21	GPIO

“DNC” (Do Not Connect).

The I2C, SPI and UART interfaces can also be used as general purpose I/O pins when not being used in their bus modes. “DNC” (Do Not Connect) .

Giving a total of 17 I/O pins on the P1-connector (plus 4 more on the P5-connector on a Rev-2 Pi) and 28 I/O pins on the B+

By default the I/O pins are all configured as inputs except GPIO 14 & 15.

Each GPIO pin in RPi has Internal software configurable “pull-up” and “pull-down” resistors.

Pull-up is 50 kOhm - 65 kOhm

Pull-down is 50 kOhm - 60 kOhm

Fuglabord_Launcher.sh v1.1

```
#!/bin/bash
#
# http://mywiki.woledge.org/ProcessManagement
# Relaunch an executable if it dies:
#
python_file="Fuglabord_main.py" # <----- Python file name.
logger_file="_logger_$(date +%m_%y').txt" # <-- Logger file name.

script=`basename $0`
directory="$1"
echo "$(date +%d.%m.%Y-%H:%M') | $script: Allt Gott! Kveiki á Python forriti." >> ${directory}/${logger_file}
while ;; do
    sleep 5
    sudo python3 "${directory}/${python_file}" "$directory" >> "${directory}/${logger_file}" 2>&1
    sleep 15
    echo "$(date +%d.%m.%Y-%H:%M') | $script: Eitthvað fór úrskeiðis! Endurræsi Python forrit..." >>
    ${directory}/${logger_file}
done
echo "$(date +%d.%m.%Y-%H:%M') | $script: Eitthvað fór úrskeiðis! slökkt á Python forriti" >>
${directory}/${logger_file}
```

Fuglabord_main.py v1.1

```
#!/usr/bin/env python3

### Til að kveikja á forrit gerðu:
### python3 ./Fuglabord_main.py "~/path/to/program/directory/"
### Eða:
### ./Fuglabord_Launcher.sh "~/path/to/program/directory/"

### Import Modules:
import sys
import time
# Import the python-vlc module:
# https://www.olivieraubert.net/vlc/python-ctypes/doc/
import vlc
# Import the GPIO module and check to see if it is successful:
# https://sourceforge.net/p/raspberry-gpio-python/wiki/Examples/
try:
    import RPi.GPIO as GPIO
except RuntimeError:
    time_string = time.strftime("%d.%m.%Y-%H:%M", time.localtime())
    print(time_string, ""| Python: Error importing RPi.GPIO! This
          is probably because you need superuser privileges.
          You can achieve this by using 'sudo' to run
          your script"", flush=True)

# Command-line argument:
directory = "./"
try:
    directory = str(sys.argv[1])
except:
    time_string = time.strftime("%d.%m.%Y-%H:%M", time.localtime())
    print(time_string, ""| Python: Aðvörun! Vantar "command-line argument"
          sem á að inni halda staðsetninguna á forritinu. Til að
          kveikja á forrit:
          python3 ./Fuglabord_main.py "~/path/to/program/" "",
          flush=True)

# Using the BOARD numbering system. This refers to the pin numbers on
# the P1 header of the Raspberry Pi board.
GPIO.setmode(GPIO.BOARD)
```

```

def Stilla_vlc(fugla_hljod):
    vlc_spilarar = []
    for hljod in fugla_hljod:
        time_string = time.strftime("%d.%m.%Y-%H:%M", time.localtime())
        print(time_string, '| Python: Sæki hljóðfæl: "' + hljod + "'", flush=True)
        print("          | Python: VLC: Villur og Viðvaranir:",
              vlc_spilarar.append(vlc.MediaPlayer(hljod)), flush=True)
    for spilari in vlc_spilarar:
        spilari.audio_set_volume(100)
    return vlc_spilarar

def Prenta_innganga(innganga_listi):
    stada_skynjara = []
    for inngangur in innganga_listi:
        stada_skynjara.append(GPIO.input(inngangur))
    time_string = time.strftime("%d.%m.%Y-%H:%M", time.localtime())
    print(time_string, "| Python: Stada Skynjara:", stada_skynjara, flush=True)

def Bera_saman_fugla_og_skynjara(innganga_listi, fugla_listi):
    stada_skynjara = []
    for inngangur in innganga_listi:
        stada_skynjara.append(GPIO.input(inngangur))
    fugl_nr = 0
    for fugl in fugla_listi:
        if fugl == stada_skynjara:
            #print("Spila Fuglahljóð nr:", fugl_nr)
            return fugl_nr
        break
    fugl_nr += 1
    return -1

def Spila_hljod(spilari, fugl):
    if not spilari.is_playing():
        spilari.stop()
        spilari.play()
    time_string = time.strftime("%d.%m.%Y-%H:%M", time.localtime())
    print(time_string, "| Python: Spila Hljóð:", fugl, flush=True)

innganga_listi = [16, 18, 19, 21, 22, 23, 24, 26]
GPIO.setup(innganga_listi, GPIO.IN, pull_up_down=GPIO.PUD_UP)
# The GPIO-pins NEED to have 'pull-ups'!
# The sensors are wired in such a way that when it sense something,
# it will connect the GPIO-pin to ground(GND).
# NO 3V3 OR 5V ARE NEED FOR THE SENSORS AND SHOULD NOT BE CONNECTED!

```



```

fugla_listi = [[0,0,0,0,0,0,0], # Fálki.
               [0,1,0,1,0,1,0,1], # Stokkönd.
               [1,0,1,0,1,0,1,0], # Auðnutittlingur.
               [1,1,1,1,1,1,1,1]] # Stelkur.
fugla_hljod = [directory + "falki.ogg",
               directory + "stokkond.ogg",
               directory + "audnutittlingur.ogg",
               directory + "stelkur.ogg"]
vlc_spilarar = Stilla_vlc(fugla_hljod)

time.sleep(5)
var_ad_spila = -1
try:
    time_string = time.strftime("%d.%m.%Y-%H:%M", time.localtime())
    print(time_string, "| Python: Main-loop Byrjar.", flush=True)
    while(True):
        #Prenta_innganga(innganga_listi)
        fugl_nr = Bera_saman_fugla_og_skynjara(innganga_listi, fugla_listi)
        if fugl_nr >= 0:
            if fugl_nr != var_ad_spila:
                #print("Spila Fuglahljóð nr:", fugl_nr)
                Spila_hljod(vlc_spilarar[fugl_nr], fugla_hljod[fugl_nr])
                var_ad_spila = fugl_nr
            else:
                var_ad_spila = -1
            time.sleep(0.2)
except Exception as e:
    time_string = time.strftime("%d.%m.%Y-%H:%M", time.localtime())
    print(time_string, "| Python: VILLA!", e, flush=True)
finally:
    GPIO.cleanup()
    time_string = time.strftime("%d.%m.%Y-%H:%M", time.localtime())
    print(time_string, "| Python: GPIO Cleanup done.", flush=True)

```
