

Eigenbau Brutautomat 2019
nach Vorlage Brutautomat.blogspot.ch

Autor: Bruno Brückmann

Frei für nichtkommerzielle Verwendung

Inhaltsverzeichnis

Vorwort.....	3
Gehäuse.....	4
Wendeeinheit.....	9
Installation des Betriebssystems Linux.....	13
Zusätzliche Software.....	14
fail2ban.....	14
ufw.....	14
Sendmail.....	14
rrdtool.....	14
Motion.....	17
Webserver.....	18
Anpassung /etc/rc.local.....	21
Installation OWFS mit DS9490R.....	21
I/O-Belegung.....	23
Crontab.....	23
Watchdog.....	23
Relais.....	23
Heizung.....	24
Feuchtigkeitsregelung.....	24
Lift.....	24
Lagerung.....	28
Eier desinfizieren.....	28
Brutstart.....	28
Lagerung Steuerdatei.....	29
Hühner Steuerdatei.....	30
Enten Steuerdatei.....	30
Programmcode.....	31
Teileliste.....	47
Lieferanten.....	49

Vorwort

Ich hatte für meinen Vater einen Janoel 12 Brüter gekauft, damit er seine Küken selbst ausbrüten kann. Leider habe ich festgestellt, dass die Angabe des chinesischen Herstellers nicht stimmt und nur max. 9 Hühnereier in den Brüter passen, wenn man die Wendeeinheit verwenden will.

Nach etwas Recherche im Internet bin ich auf die Seite Brutautomat.blogspot.ch gekommen und habe einen Nachbau angefangen. Ich habe den Brutautomat nicht neu erfunden, sondern aus den Informationen welche ich an verschiedensten Stellen im Internet gefunden habe und dem Material welches gerade Verfügbar war gebaut. Eigene Kenntnisse sind natürlich auch eingeflossen und ich hoffe sinnvolle. Ich möchte daher allen Danken, die entsprechende Informationen im Internet zur Verfügung gestellt haben und indirekt den Bau unterstützen.

Sollte sich ein Fehler eingeschlichen haben, bin ich um eine Nachricht dankbar, denn Bauen macht mir mehr Spaß als Schreiben, da kann schon mal etwas übersehen werden.

Der Brüter war schon mehrere Male erfolgreich im Einsatz, aber es gibt nichts was man nicht noch etwas verbessern kann.

Ich würde mir wünschen, dass das Projekt z.B. in einer Schule zum Einsatz kommt, hier kann man die Biologie mit Handwerk und Informatik wunderbar verbinden. Das einzige Problem besteht darin, dass jemand die Küken abnehmen muss.

Ziel des Projektes ist der Bau eines Brutautomaten für Hühner und Enten und gleichzeitig die Nutzung des freien Betriebssystems Linux unter Verwendung eines preisgünstigen Minicomputers (Raspberry PI) zu erlernen.

Der Anwender erwirbt die folgenden Kenntnisse nach Ende des Projekts:

Software

- Installation des Betriebssystems Linux
- Aktualisierung des Betriebssystems
- Aufbau einer Firewall
- Definition eines Benutzers
- Installation von Software
- Konfiguration des eigenen Routers
- Verbindung von einem entfernten Rechner zum Raspberry PI
- Aufbau eines Webservers
- Auslesen von Sensoren (Temperatur, Feuchtigkeit)
- Einfache Programmierkenntnisse (Shell, REXX)

Hardware

- Aufbau einer Erweiterungsplatine
- Verschaltung der Sensoren und Aktoren
- Aufbau eines Gehäuses incl. Zubehör (Inkubator)
- Anschluss einer Relaisplatine

Ich wünsche allen viel Spaß beim Nachbau und eigenen Modifikationen.

Gehäuse

Das Gehäuse ist aus XPS 30mm hergestellt. Das Gehäuse wurde mit Pattex PL Montagekleber geklebt. Da das Gehäuse beim Brüten innen verschmutzt, habe ich das Styrodur mit einer Mischung je 1 Teil Wasser, Wasserfester Holzleim und Acrylfarbe gestrichen.

Es ist vielleicht besser, erst alle Teile zu versiegeln und dann zusammenzukleben, dann ist die Bearbeitung leichter.

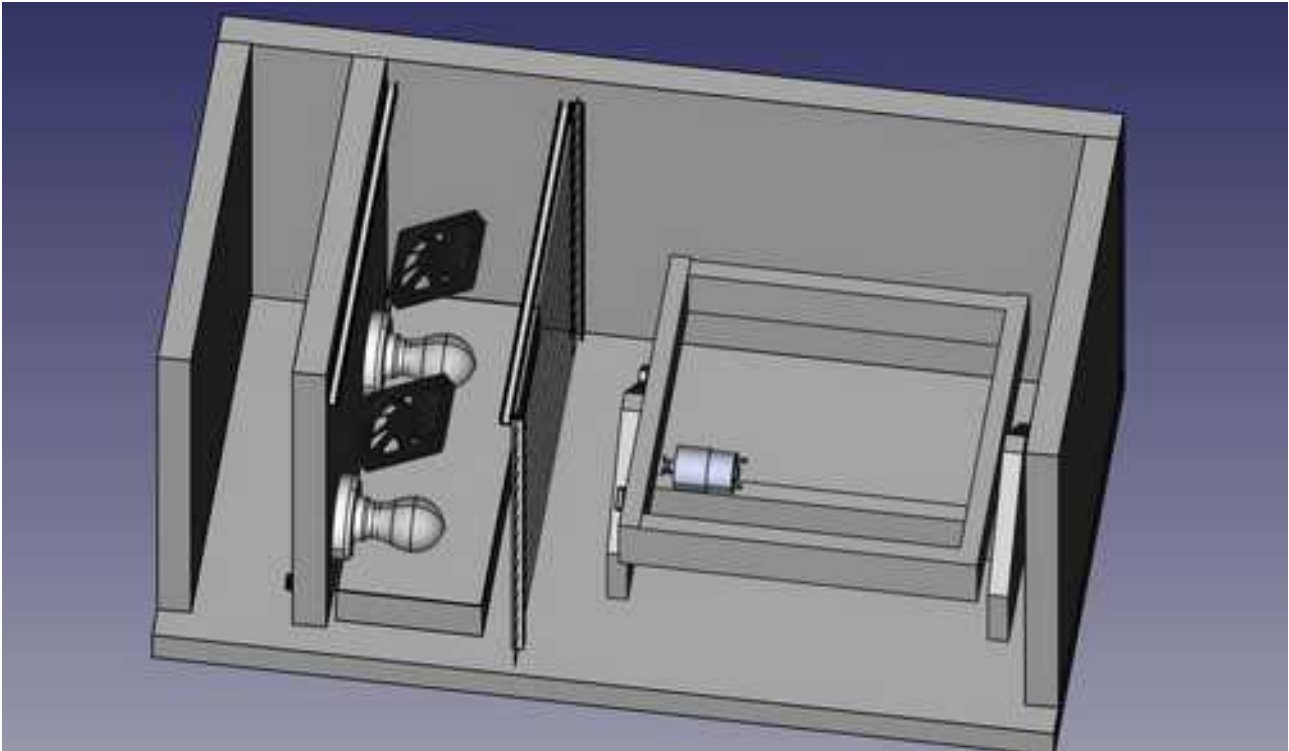


Bild 1

Im folgenden Bild sieht man die Heizung und Luftumwälzung.

Ich habe 2 E27 Fassungen verbaut, wenn eine Glühlampe ausfällt, ist die Heizung trotzdem sichergestellt. Ich verwende eine 60 W Glühlampe und einen 60 W Keramikstrahler.

Beim Schlupf ist das Blinken der 60W Lampe zwar für die Küken unangenehm, hat aber den Vorteil, das ich ein Bild mit der Kamera aufnehmen und dieses über das Internet sehen kann.

Der ganze Brüter wird über einen Raspberry Zero gesteuert und ist vom Internet überwachbar.

Für die Nachbauer möchte ich darauf Hinweisen, nur Solid State Relais für die Heizung zu verwenden. Für alle anderen Schaltfunktionen (Gleichstrom) sind normale Relais zu verwenden, da Solid State Relais nicht mit Gleichstrom funktionieren. Die Solid State Relais können zwar den Strom anschalten, aber nicht ausschalten, da der Nulldurchgang beim Gleichstrom fehlt.

Die Lampenheizung schaltet sehr häufig ein, was zu einer großen Schaltfrequenz der Relais führt.

Bei meiner letzten Brut hat das Relais geklebt und die Heizung ist ausgefallen.

Als Lüfter habe ich einen 5V USB-Tischventilator verwendet. Ich habe zuerst 2 Ventilatoren eingebaut, es hat sich aber gezeigt, das dies zu einem Orkan im Brüter führt.

Als Reflektionsfolie habe ich Alufolie aus der Küche auf Pappe geklebt und diese dann an den Brüter. Schöner wäre hier dünnes Alublech.

An dem Metalldraht in der Mitte des Bildes sind die Fühler für Temperatur und Feuchtigkeit befestigt. Das Schwarz/Rote Kabel mit der Lüsterklemme ist die Stromversorgung der Wendeeinheit.



Bild 2

Im folgenden Bild sieht man die Technik des Brüters.

Links im Bild, der Raspberry Zero mit 2-Fach Relais und einer Lochrasterplatte auf welcher das Motorwenderelais und ein 1-Wire-Anschluss vorhanden ist. Diese Platine wird gerade neu entworfen und soll dann ein 4-Fach Relais ansteuern incl. 1-Wire und IR-Sender/IR-Receiver. Die IR-Sender/Receiver sind nicht für den Brüter notwendig, aber wenn der Brüter nicht aktiv ist, kann man den Raspberry für andere Zwecke verwenden.

Auf das 4-Fach Relais will ich umstellen, da ich für das Brüten von Enteneiern noch eine Deckelöffnung einbauen möchte, um das Kühlen der Eier zu automatisieren.

In der Mitte des Bildes ist die Dosierpumpe zu sehen. Die Dosierpumpe wird über ein Relais angesteuert, wenn die Feuchtigkeit im Brüter zu gering ist. Die Pumpe läuft nur ein paar Sekunden, dann wird eine Zeit lang gewartet, ob sich die Feuchtigkeit genügend erhöht hat. Ich habe für die Pumpe Infusionsschläuche verwendet, welche mit einem 3mm Messingrohr verbunden sind. Der Silikonschlauch der Pumpe neigt zum wandern, welches leicht an den ungleich langen Enden des Schlauches an der Pumpe zu erkennen ist. Die Aufweitung des Schlauches durch das Messingrohr verhindert ein durchrutschen des Schlauches in die Pumpe.

ACHTUNG! Stellen Sie das Vorratsgefäß für das Wasser (Schüssel) niemals höher als den Boden des Brüters, sonst flutet die Schwerkraft den Brüter.

Das Schwarze rechts im Bild ist ein aktiver USB-HUB der an den Raspberry angeschlossen ist.

In der Dreifachsteckdose befinden sich ein Raspberry-Netzteil und ein USB-Netzteil für den USB-HUB. Es gibt nur 240V und 5V Verbraucher um Netzteile zu sparen.

ACHTUNG! Verwenden Sie nur das Original Raspberry Netzteil, sonst gibt es Systemausfälle wegen Spannungsschwankungen.

Über das USB-Netzteil wird die Spannung für die Wendeeinheit über das Wenderelais auf der Lochrasterplatte geführt. An den USB-HUB ist zusätzlich die Webcam angeschlossen.



Bild 3

Im folgenden Bild ist der Aufbau des Deckels zu sehen, welcher einen umlaufenden Rand zur Fixierung besitzt. Als Fenster habe ich oben und unten am Deckel Kunstglas angebracht. Über dem Glas lege ich dann meine Webcam auf und kann so den Innenraum über das Internet beobachten. Das abgeschnittene Handtuch auf dem Bild ist mein Feuchteverteiler. Ich hatte zuerst ein Gitter mit einem Küchentuch in die Wasserschale gestellt, welches sehr gut funktionierte für meine Hühnerbrut. Es hat sich aber herausgestellt, dass die Oberfläche für die Entenbrut nicht ausreicht, ich habe max. 72% Luftfeuchtigkeit erreicht. Ich habe nun 2 Handtuchstreifen an den Enden umgenäht und einen Draht alt Halter eingeschoben. Die Handtuchstreifen werden als U um die Heizelemente gelegt, wobei die Unterseite in der Wasserschale liegt. Das Wasser wird nun in die Wasserschale gepumpt und steigt an den Handtuchstreifen zur Verdunstung auf. Mit dieser Lösung komme ich nun ohne Probleme bis auf 95% (Bild 4/5). Damit sich der Infusionsschlauch in der Wasserschale hält, habe ich eine Wäscheklammer als Gewicht montiert.



Bild 4



Bild 5



Wendeeinheit

Die Wendeeinheit habe ich abweichend vom Vorbild als Wippe ausgeführt, dies war einfach zu bauen. Die Grün markierten Teile sind nicht notwendig, dies war der alte Anschlag der Schalter. Der Blau markierte Winkel verhindert das Wackeln des Fußes, da er sich etwas verzogen hat. Das rote Gitternetz unter der Eierhorde ist Putzgewebe aus dem Baumarkt.

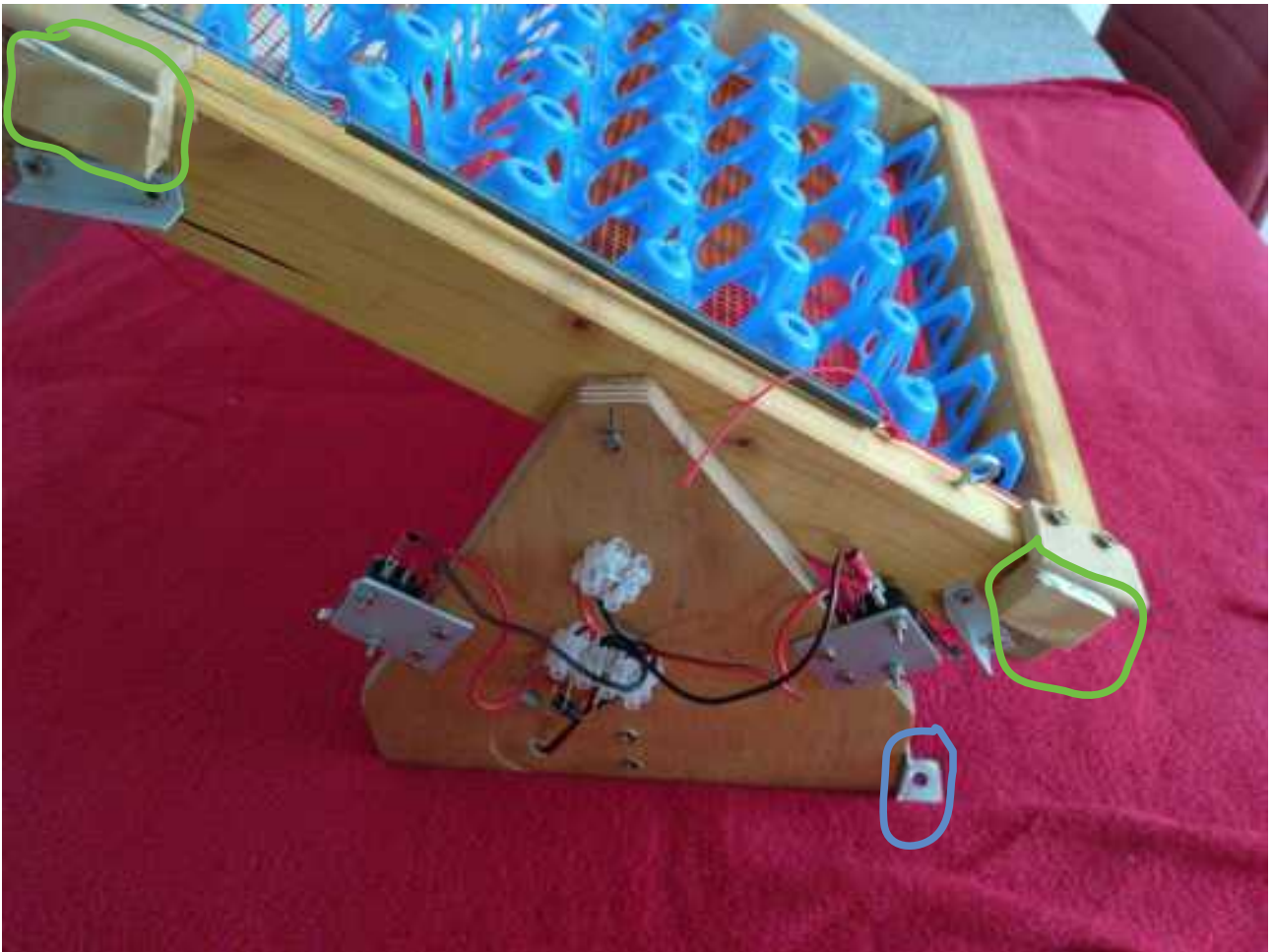


Bild 1

Als Vorlage habe ich das folgende Schaltbild (Bild 3) verwendet. Die 2 Leitungen vom Relais zum Motor und zur Diode D1 sind auf eine Lüsterklemme gelegt und können so einfach zum Schluftermin getrennt werden um die Wendeeinheit aus dem Brüter zu nehmen. Die Zuleitung vom Relais zur Wendeeinheit hat ebenfalls eine doppelte Lüsterklemme mit 2 kurzen 1,5 qm2 Drähten welche als Stecker dient (Bild 2). Es können auch Stecker aus dem Modellbau verwendet werden.



Bild 2

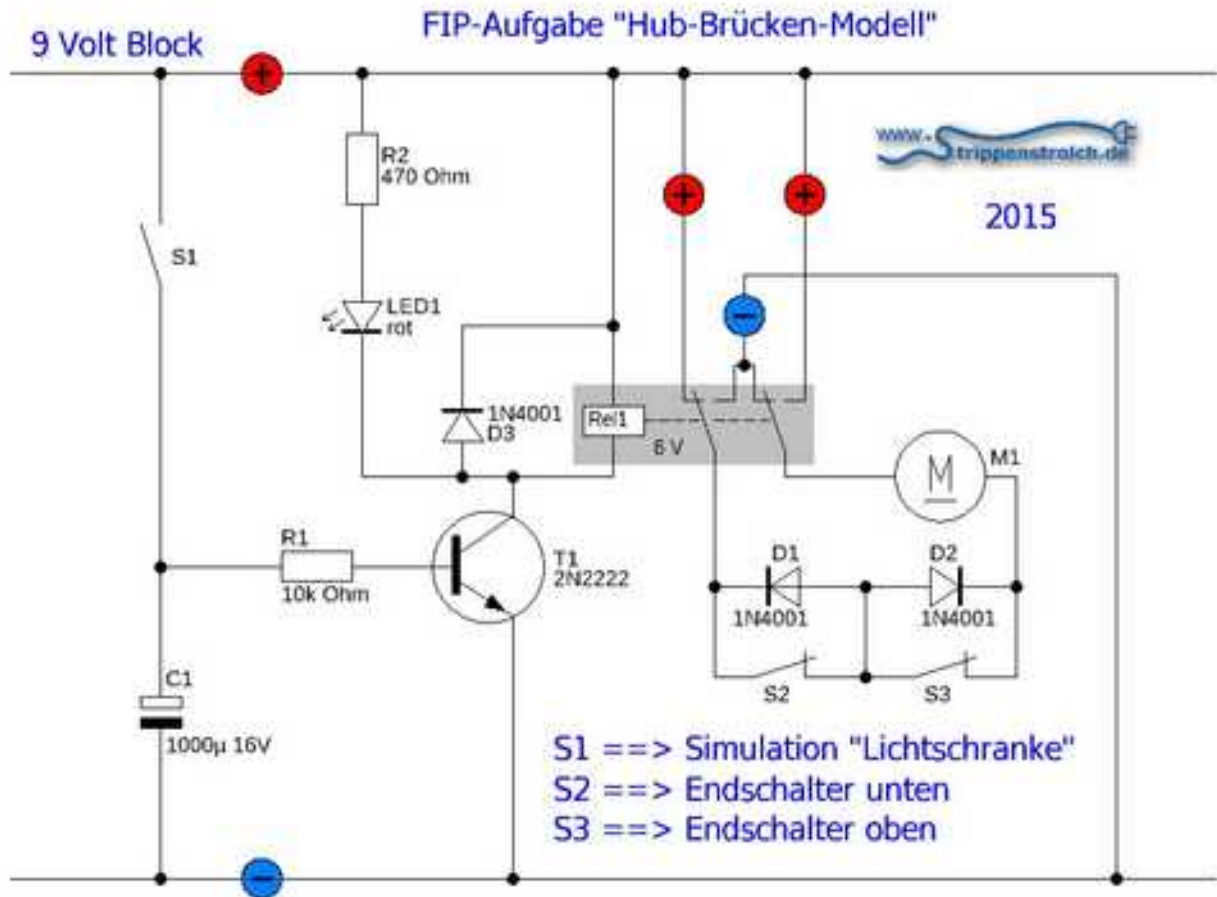
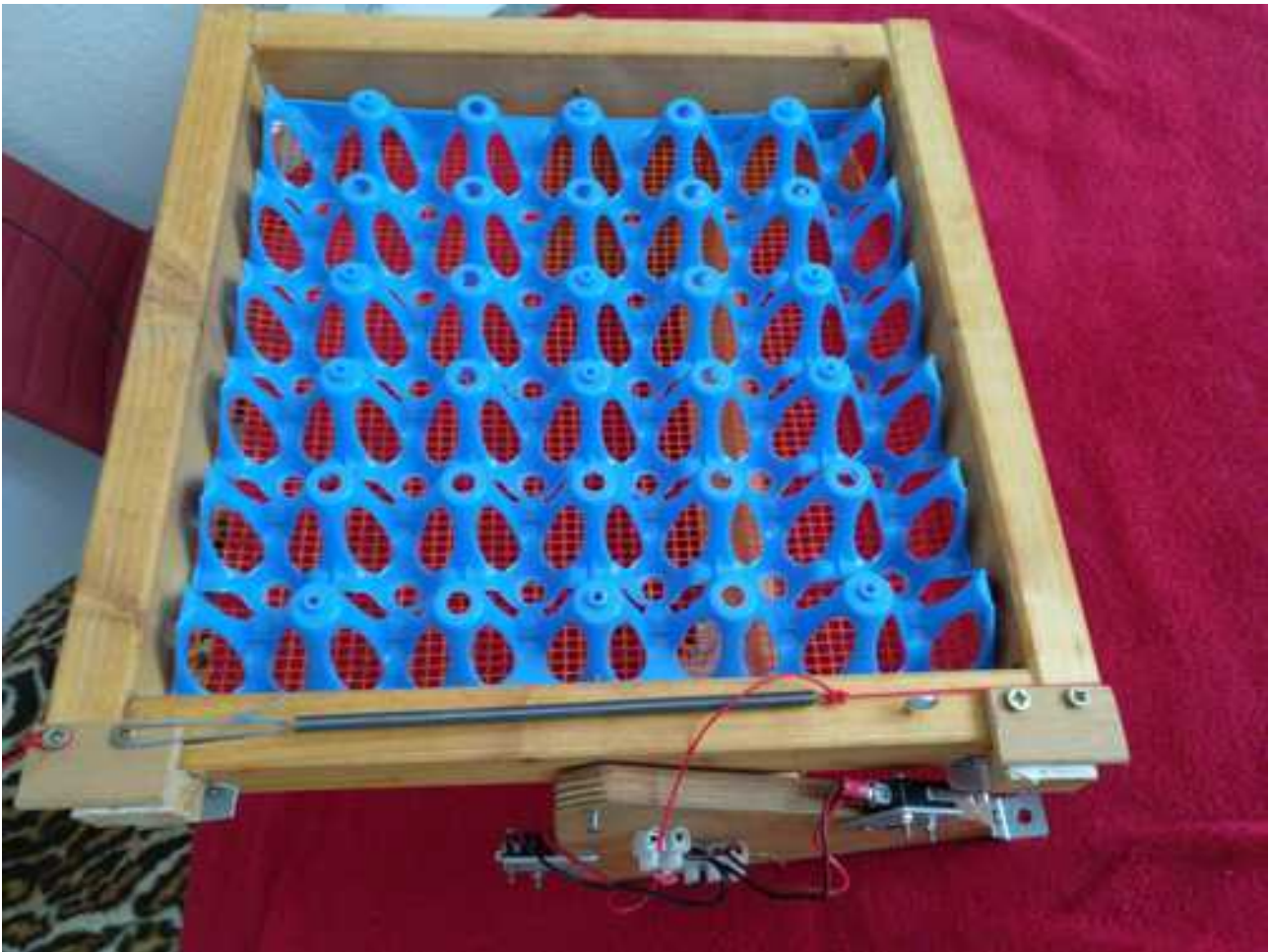


Bild 3



Bild 4



Installation des Betriebssystems Linux

```
cd /tmp
```

```
wget https://downloads.raspberrypi.org/raspbian\_lite\_latest
```

```
unzip raspbian_lite_latest
```

```
dd bs=4M if=/tmp/2018-03-13-raspbian-stretch-lite.img of=/dev/sdb conv=fsync
```

Einstecken der SD-Card in den Raspberry, anschliessen von Tastatur, Monitor und Netzteil.

Nach dem Verbinden des Netzteils mit dem Stromnetz

Geben Sie bei der Anzeige „User“ als User pi ein.

Wenn die Anzeige „password“ erscheint geben Sie raspberry ein, Achtung! Wenn Sie eine deutsche Tastatur haben, müssen Sie rasperrz eingeben. Die Standardeinstellung ist die amerikanische Tastatur.

Nach dem erfolgreichen Anmelden ändern wir sofort das Passwort mit dem Befehl
passwd .

Zusätzliche Software

fail2ban

Fail2ban ist ein Programm was einen User sperrt, wenn er eine bestimmte Anzahl fehlerhafter Anmeldeversuche auf Ihrem Raspberry durchgeführt hat. Dies verhindert, das z.B Hacker alle möglichen User/Passwörter durchprobieren können um auf Ihr System zu kommen. Verzichten Sie bitte nicht auf diesen Schutz, Sie werden schnell feststellen, wenn Sie die Logs kontrollieren, das es sehr viele Versuche gibt in Ihr System einzubrechen. Der Angreifer, weiss ja nicht, das Sie nur Eier ausbrüten und keine Bankdaten zu finden sind. Im dümmsten Fall ruiniert er Ihre Brut.

```
apt-get install fail2ban
cd /etc/fail2ban
cp fail2ban.conf fail2ban.local
nano fail2ban.local
cp jail.conf jail.local
nano /etc/fail2ban/jail.local change destemail
```

ufw

```
apt-get install ufw
ufw enable
ufw default deny
ufw allow 22
ufw allow 80
ufw allow 81
ufw allow 8081
ufw allow 443
ufw limit ssh/tcp
```

Sendmail

Installation mit:
sudo apt-get install sendmail

rrdtool

Installation mit:
sudo apt-get install rrdtool

Fügen Sie das folgende Script in die Datei `/home/pi/sensors-rrd.sh` ein:

```
#!/bin/bash
# generated script for rrd from data-Input
#
script="sensors-rrd.sh"
inlog=/home/pi/incubator.values
indir=/var/www/http/incubator.com
dsn=/home/pi/database.rrd
outdir=/var/www/http/incubator.com
interval=16 # messintervall sec
```

```

now1=$(date "+%a %x %X")
now=`echo ${now1//:/} `
timeoffset=" #need to offset the times read by file ,to the time generated the database
input=$1 #fuer Trace
Temp=0
Hum=0
Ahumr=0
Heater=0
Eggturn=0
Acool=0
case "$1" in
    create)
        # create a rrd-Database with Input-Variables
        #rrdtool create ./database.rrd --start N --step 16 \
        LINE=`head -n1 $inlog|awk '{ print $1}'`
        y=`echo "$LINE" |awk '{ print $1}'|awk 'BEGIN {FS=" "}{print $1}'|sed "s/\\/ /g"|awk
'BEGIN {FS=" "}{print $3}'`
        m=`echo "$LINE" |awk '{ print $1}'|awk 'BEGIN {FS=" "}{print $1}'|sed "s/\\/ /g"|awk
'BEGIN {FS=" "}{print $2}'`
        d=`echo "$LINE" |awk '{ print $1}'|awk 'BEGIN {FS=" "}{print $1}'|sed "s/\\/ /g"|awk
'BEGIN {FS=" "}{print $1}'`
        t='00:00:00'
        x=`date -d "$y-$m-$d $t" +%s`
        rrdtool create ./database.rrd --start $x --step 16 \
        DS:Temp:GAUGE:32:U:U \
        DS:Hum:GAUGE:32:U:U \
        DS:Ahumr:GAUGE:32:U:U \
        DS:Heater:GAUGE:32:U:U \
        DS:Eggturn:GAUGE:32:U:U \
        DS:Acool:GAUGE:32:U:U \
        RRA:AVERAGE:0.5:1:5400 \
        RRA:AVERAGE:0.5:1:5400 \
        RRA:AVERAGE:0.5:1:5400 \
        RRA:AVERAGE:0.5:1:5400 \
        RRA:AVERAGE:0.5:37:1008

        ;;
    update)
        #sleep $interval
        input=$indir/sensors_input.txt # data-input-dataset
        if [ ! -s $input ] ; then
            echo "File $input empty"
            exit 1
        fi
        if [ ! -e $input ] ; then
            echo "File $input not exists"
            exit 1
        fi
        if [ -s $input ] ; then # file not empty

```

```

cat $input | (while read LINE ;
do
    y=`echo "$LINE" |awk '{ print $1}'|awk 'BEGIN {FS=" "}{print $1}'|sed "s/\\/ /g"|awk
'BEGIN {FS=" "}{print $3}`
    m=`echo "$LINE" |awk '{ print $1}'|awk 'BEGIN {FS=" "}{print $1}'|sed "s/\\/ /g"|awk
'BEGIN {FS=" "}{print $2}`
    d=`echo "$LINE" |awk '{ print $1}'|awk 'BEGIN {FS=" "}{print $1}'|sed "s/\\/ /g"|awk
'BEGIN {FS=" "}{print $1}`
    t=`echo "$LINE" |awk '{ print $2}'|awk 'BEGIN {FS=" "}{print $1}`
    Temp=`echo "$LINE" |awk '{ print $5}'|awk 'BEGIN {FS="="}{print $2}'|sed "s/,./g"`
    Hum=`echo "$LINE" |awk '{ print $7}'|awk 'BEGIN {FS="="}{print $2}'|sed "s/,./g"`
    Ahumr=`echo "$LINE" |awk '{ print $8}'|awk 'BEGIN {FS="="}{print $2}`
    Eggturn=`echo "$LINE" |awk '{ print $10}'|awk 'BEGIN {FS="="}{print $2}`
    Ahumr=`echo $Ahumr + 2|bc`
    Heater=`echo "$LINE" |awk '{ print $9}'|awk 'BEGIN {FS="="}{print $2}`
    Heater=`echo $Heater + 4|bc`
    Acool=`echo "$LINE" |awk '{ print $11}'|awk 'BEGIN {FS="="}{print $2}`
    if [ "$Acool" = "1x10" ]; then Acool=1; fi
    if [ "$Acool" = "2x10" ]; then Acool=1; fi
    if [ "$Acool" = "2x20" ]; then Acool=1; fi
    Acool=`echo $Acool + 6|bc`
    # update rrd-Database with Input-Variables
    x=`date -d "$y-$m-$d $t" +%s`
    #dummy="echo rrdtool update $dsn N:$Temp:$Hum:$Ahumr:$Heater:$Eggturn:$Acool"
    dummy="echo rrdtool update $dsn $x:$Temp:$Hum:$Ahumr:$Heater:$Eggturn:$Acool"
    echo "$dummy"
    #`rrdtool update $dsn N:$Temp:$Hum:$Ahumr:$Heater:$Eggturn:$Acool`
    `rrdtool update $dsn $x:$Temp:$Hum:$Ahumr:$Heater:$Eggturn:$Acool`
done
)
fi
;;
graph)
# graph from a rrd-Database with Input-Variables
rrdtool graph $outdir/temp.png -a PNG \
    --height=200 --width=750 \
    -y 2:1 --vertical-label "[deg C]" \
    -s -1d -e -1s \
    DEF:Temp=$dsn:Temp:AVERAGE \
    LINE1:Temp#CC6600:"Temp" \
    GPRINT:Temp:LAST:"Temp-Last\:%8.2lf %s" \
    COMMENT:" Created at $now"
rrdtool graph $outdir/humidity.png -a PNG \
    --height=200 --width=750 \
    -y 2:1 --vertical-label "[Humidity %]" \
    -s -1d -e -1s \
    DEF:Hum=$dsn:Hum:AVERAGE \
    LINE1:Hum#1E90FF:"Hum" \
    GPRINT:Hum:LAST:"Hum-Last\:%8.2lf %s" \
    COMMENT:" Created at $now"

```

```

rrdtool graph $outdir/relais.png -a PNG \
--height=200 --width=750 \
-y 8:1 --vertical-label "[On/Off]" \
-s -1d -e -1s \
DEF:Ahumr=$dsn:Ahumr:AVERAGE \
DEF:Heater=$dsn:Heater:AVERAGE \
DEF:Eggturn=$dsn:Eggturn:AVERAGE \
DEF:Acool=$dsn:Acool:AVERAGE \
LINE1:Ahumr#CC6600:"Ahumr" \
LINE1:Heater#1E90FF:"Heater" \
LINE1:Eggturn#FF0000:"Eggturn" \
LINE1:Acool#00FF7F:"Acool" \
GPRINT:Ahumr:LAST:"Ahumr-Last\:%8.2lf %s" \
GPRINT:Heater:LAST:"Heater-Last\:%8.2lf %s" \
GPRINT:Eggturn:LAST:"Eggturn-Last\:%8.2lf %s" \
GPRINT:Acool:LAST:"Acool-Last\:%8.2lf %s" \
COMMENT:" Created at $now"
;;
start)
;;
stop)
;;
fetch)
rrdtool fetch $dsn AVERAGE -r $interval -s -1d -e -1s
;;
*)
echo "Error no generate,start,stop,update,fetch,graph or create parameter given"
exit 1
;;
esac

```

Motion

Installation mit:

```
sudo apt-get install motion fswebcam
```

Webserver

Als Webserver habe ich nginx verwendet.

Die Konfiguration in der Datei `/etc/nginx/conf.d/incubator.com.conf` sieht wie folgt aus:

incubator.com

```
server {
    rewrite_log on;
    #access_log off;
    access_log /var/log/incubator-access_log;
    error_log /var/log/incubator-error_log;
    listen 81;
    server_name incubator.com;
    root /var/www/http/incubator.com/;
    index index.php index.html;
    auth_basic "Restricted";
    auth_basic_user_file /var/www/http/incubator.com/.htpasswd;

    location ~ .php$ {
        try_files $uri =404;
        fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
        fastcgi_index index.php;
        include /etc/nginx/fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }

    location /motion {
        #proxy_pass http://192.168.1.31:8081;
        proxy_pass http://127.0.0.1:8081;
        #proxy_redirect default;
        #proxy_set_header Host $host;
        #proxy_set_header X-Real-IP $remote_addr;
        #proxy_set_header X-Forwarded-for $remote_addr;
        #port_in_redirect off;
    }

    location / {
        root /var/www/http/incubator.com/;
    }
    if ($http_user_agent ~* ^Baiduspider) {
        return 403;
    }
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root /var/www/http/incubator.com;
    }
}
```

Die Datei /var/www/http/incubator.com/index.html sieht wie folgt aus:

```
<!DOCTYPE html>
<html lang="de">
<html>
<head>
<meta charset="utf-8"/>
<title>Incubator Control Panel</title>
</head>
<META NAME="title" CONTENT="Incubator Control">
<META NAME="author" CONTENT="Dipl. Inf. Bruno Brueckmann">
<META NAME="publisher" CONTENT="Dipl. Inf. Bruno Brueckmann">
<META NAME="copyright" CONTENT="Dipl. Inf. Bruno Brueckmann">
<META NAME="revisit-after" CONTENT="10 days">
<META NAME="copyright" CONTENT="Dipl. Inf. Bruno Brueckmann">
<META NAME="revisit-after" CONTENT="10 days">
<META NAME="expires" CONTENT="31.12.2020">
<META NAME="keywords" CONTENT="Incubator Control on Raspberry ">
<META NAME="description" CONTENT="Linux">
<META NAME="abstract" CONTENT="Linux">
<META NAME="page-topic" CONTENT="Betriebssysteme">
<META NAME="audience" CONTENT=" Alle ">
<META NAME="robots" CONTENT="INDEX,FOLLOW">
<META NAME="language" CONTENT="Deutsch">
<body>
<hr>
<h2>Incubator Control</h2>
<h2>Cameras</h2>
<a href="/motion">Lifecam Motion</a><br>
<a href="webcamshot.jpg">Webcam Snapshot</a><br>
<a href="sensors_input.txt">Status as Text</a><br>
<h2>Sensordata and Status Graphics</h2>
<a href="temp.png">Temperature as Graphic</a><br>
<a href="humidity.png">Humidity as Graphic</a><br>
<a href="relais.png">Switches as Graphic</a><br>
<h2>Schalter, Achtung nur im Notfall verwenden!</h2>
<p><a href="lift-up.php">Lift hoch</a> <a href="lift-down.php">Lift runter</a></p>
<p><a href="heater-on.php">Heizung-an</a> <a href="heater-off.php">Heizung-aus</a></p>
<p><a href="pump-on.php">Pumpe-an</a> <a href="pump-off.php">Pumpe-aus</a></p>
<p><a href="turn-left.php">Wendeeinheit-Links</a> <a href="turn-right.php">Wendeeinheit-
Rechts</a></p>
<br>
<!-- <p><a href="info.php">PHP Status</a> -->
</body>
</html>
```

Die Datei /var/www/http/incubator.com/heater-off.php sieht wie folgt aus:

```
<?php
$result = shell_exec('./switch-gpio.sh 16 on 1');
echo "<pre>$result</pre>";
header('Location: ' . $_SERVER['HTTP_REFERER']);
```

```
exit;
```

```
?>
```

Die Datei `/var/www/http/incubator.com/heater-on.php` sieht wie folgt aus:

```
<?php
```

```
$result = shell_exec('./switch-gpio.sh 16 on 0');
```

```
echo "<pre>$result</pre>";
```

```
header('Location: ' . $_SERVER['HTTP_REFERER']);
```

```
exit;
```

```
?>
```

Die Datei `/var/www/http/incubator.com/lift-down.php` sieht wie folgt aus:

```
<?php
```

```
$result = shell_exec('./switch-gpio.sh 5 on 1');
```

```
echo "<pre>$result</pre>";
```

```
header('Location: ' . $_SERVER['HTTP_REFERER']);
```

```
exit;
```

```
?>
```

Die Datei `/var/www/http/incubator.com/lift-up.php` sieht wie folgt aus:

```
<?php
```

```
echo shell_exec("./switch-gpio.sh 5 on 0");
```

```
header('Location: ' . $_SERVER['HTTP_REFERER']);
```

```
exit;
```

```
?>
```

Die Datei `/var/www/http/incubator.com/pump-off.php` sieht wie folgt aus:

```
<?php
```

```
$result = shell_exec('./switch-gpio.sh 20 on 1');
```

```
echo "<pre>$result</pre>";
```

```
header('Location: ' . $_SERVER['HTTP_REFERER']);
```

```
exit;
```

```
?>
```

Die Datei `/var/www/http/incubator.com/pump-on.php` sieht wie folgt aus:

```
<?php
```

```
$result = shell_exec('./switch-gpio.sh 20 on 0');
```

```
echo "<pre>$result</pre>";
```

```
header('Location: ' . $_SERVER['HTTP_REFERER']);
```

```
exit;
```

```
?>
```

Die Datei `/var/www/http/incubator.com/turn-left.php` sieht wie folgt aus:

```
<?php
```

```
$result = shell_exec('./switch-gpio.sh 17 on 0');
```

```
echo "<pre>$result</pre>";
```

```
header('Location: ' . $_SERVER['HTTP_REFERER']);
```

```
exit;
```

```
?>
```

Die Datei `/var/www/http/incubator.com/turn-right.php` sieht wie folgt aus:

```
<?php
$result = shell_exec('./switch-gpio.sh 17 on 1');
echo "<pre>$result</pre>";
header('Location: ' . $_SERVER['HTTP_REFERER']);
exit;
?>
```

Anpassung */etc/rc.local*

insert in file

Gpio for non-root users

sudo chown root:gpio /sys/class/gpio/unexport /sys/class/gpio/export

sudo chmod g+rwX /sys/class/gpio/unexport /sys/class/gpio/export

Installation OWFS mit DS9490R

Bei der Verwendung des USB-1-Wire Adapters DS9490R



Zusätzlich werden die folgenden Adapter/Kabel benötigt:



```
sudo apt-get install owfs ow-shell
```

In die Datei `/etc/modprobe.d/raspi-blacklist.conf`
die folgenden Zeilen einfügen:

```
blacklist ds2490  
blacklist ds9490r  
blacklist wire
```

Die Datei `/etc/owfs.conf`
wie folgt ändern:

```
#server: FAKE = DS18S20,DS2405  
#  
# USB device: DS9490  
server: usb = all  
mountpoint = /mnt/1wire  
allow_other
```

Starten Sie den Server neu mit `etc/init.d/owserver restart`

I/O-Belegung

-----Nr. I/O-- Nr. USB--

Relais 1 Heizung	1	
Relais 2 Befeuchtung	1	
Pio fuer Bewegungsservo oder	1	
Relais 3 mit Wechselrelais und Endschalter		
Pio 1-Wire	1	
USB 1-Wire Feuchtigkeitssensor, Temperatur		1?
Lüfter		
Innenbeleuchtung	1	
Kamera	1	
Wlan	1?	

Gpio 26 Pin 37-> IR-Receiver
Gpio 22 Pin 15 -> IR-sender
Gpio 17 -> Relais Umschalter
Gpio 4 -> 1-Wire
Gpio 16 -> Relais 1, 2-Channel extern (möglichst Solid State) Heizung
Gpio 6 -> Relais 2, 2-Channel extern (normales Relais) Wasserpumpe
Gpio 23 -> Relais 3, 2-Channel extern (normales Relais) Frei
Gpio 24 -> Relais 4, 2-Channel extern (normales Relais) Frei
Gpio 5 -> Relais Lift

Crontab

In der Crontab des Users pi müssen die folgenden Einträge vorhanden sein:

```
@reboot /home/pi/afterboot.sh  
*/5 * * * * /home/pi/sensors-rrd.sh graph >/dev/null
```

Watchdog

Es sollte unbedingt der Watchdog des Raspberry aktiviert werden. Wenn sich das System aufhängt, wird dann über den Watchdog ein automatischer System-Restart ausgeführt. Der letzte Betriebszustand ist in einer Datei restart.txt gespeichert, so kein Eingriff erforderlich ist.

Relais

Als Schaltaktoren werden Relais verwendet.

Hierbei ist besonders zu beachten, das für die Wendeeinheit ein Umschaltrelais 2-fach 6V DC verwendet wird.

Für die Heizung welche mit Wechselstrom 240V betrieben wird empfiehlt sich ein Solid State Relais, da ein normales Relais durch das häufige Schalten der Heizung extrem belastet wird und dadurch zum Ausfall neigt. Nach der dritten Brut hat das Relais geklebt und die Heizung wurde nicht mehr ausgeschaltet, die Bruteier wurden dadurch gekocht.

Das Solid State Relais sollte ein Low Trigger haben, damit die Heizung beim Start des Raspberry ausgeschaltet ist (Sicherheit). Die weiteren Relais müssen normale Relais sein, da hiermit Gleichstrom geschaltet wird. Es gibt Relaisplatinen 4-fach mit Umschaltrelais, diese haben den Vorteil, das man den Verbraucher an den Kontakt anschliessen kann, welcher beim Einschalten des Raspberry stromlos ist (Sicherheit). Es kann ja man ein Programmfehler auftreten, da ist es gut, wenn alle Verbraucher ausgeschaltet sind.

Stehen die passenden Relais nicht zur Verfügung kann im Brutprogramm der Status der Relais entsprechend gesetzt werden, dies bedingt jedoch, das das Brutprogramm immer sicher gestartet wird. Im Brutprogramm müssen die Relaisdefinitionen mit der Hardware übereinstimmen.

Es kann das Prüfprogramm gpiotest.sh zum Relais test verwendet werden.

```
cat gpiotest.sh
#!/bin/bash
# Test GPIO Ports
gpio_set () {
    tme=15
    echo 'echo "$1" > /sys/class/gpio/export'
    echo "$1" > /sys/class/gpio/export
    echo 'echo "out" > /sys/class/gpio/gpio$1/direction'
    echo "out" > /sys/class/gpio/gpio$1/direction
    echo 'echo "0" > /sys/class/gpio/gpio$1/value'
    echo "0" > /sys/class/gpio/gpio$1/value
    echo "gpio $1 on"
    sleep $tme
    echo 'echo "1" /sys/class/gpio/gpio$1/value'
    echo "1" > /sys/class/gpio/gpio$1/value
    echo "gpio $1 off"
}
sudo chmod 222 /sys/class/gpio/export /sys/class/gpio/unexport
#
#gpio_set 17
#gpio_set 16
#gpio_set 20
#gpio_set 23
#gpio_set 24
gpio_set 5
#gpio_set 2
#gpio_set 3
#gpio_set 4
#gpio_set 6
#gpio_set 7
#gpio_set 8
#gpio_set 9
#gpio_set 10
#gpio_set 11
#gpio_set 12
#gpio_set 13
```

```
#gpio_set 14
#gpio_set 15
#gpio_set 17
#gpio_set 18
#gpio_set 19
#gpio_set 20
#gpio_set 21
#gpio_set 22
#gpio_set 25
#gpio_set 26
#gpio_set 27
exit
```

Heizung

Die Heizung wird über eine normale Glühlampe 60W und einen Keramikheizer 60W (Dunkelstrahler) realisiert. Ich habe extra 2 Heizelemente verbaut um die Ausfallsicherheit zu erhöhen. Damit der Inkubator beim Einschalten nicht überhitzt, da der Dunkelstrahler viel Masse und damit Nachheizenergie besitzt, wird über den Raspberry die Heizung nach einer gewissen Zeit pausiert, so dass es nicht zum Überspringen der Temperatur in der Anheizphase kommt. Die Relais werden stark durch Schaltvorgänge belastet, da die Temperaturregelung sehr fein reagiert. Ich rate daher das Relais beim Eigenbau in einen Sockel zu setzen (einfacher Ersatz).

Feuchtigkeitsregelung

Die Feuchtigkeit wird über verschiedene Tücher gesteuert. Die Tücher hängen in einer Wasserschale und werden bei meiner Lösung von oben über eine Dosierpumpe betropft. Es ist sicherlich auch ohne Dosierpumpe möglich, in dem die Länge der Tücher in der Wasserschale variiert wird. Da ich auch Enteneier bebrüte, kann ich mit der Dosierpumpe die Luftfeuchtigkeit regeln, indem die Pumpe einige wenige Sekunden läuft und danach abgewartet wird, wie sich der Feuchtigkeitsgehalt entwickelt. Die Wasserschale ist bei dieser Methode relativ trocken, da das von oben eingespritzte Wasser auf dem Tuch verdunstet. Achtung! Der Wasservorratsbehälter (Schüssel) darf niemals höher als der Boden des Inkubators stehen, damit das Wasser nicht von selbst mittels Schwerkraft in den Inkubator läuft und alles flutet.

Lift

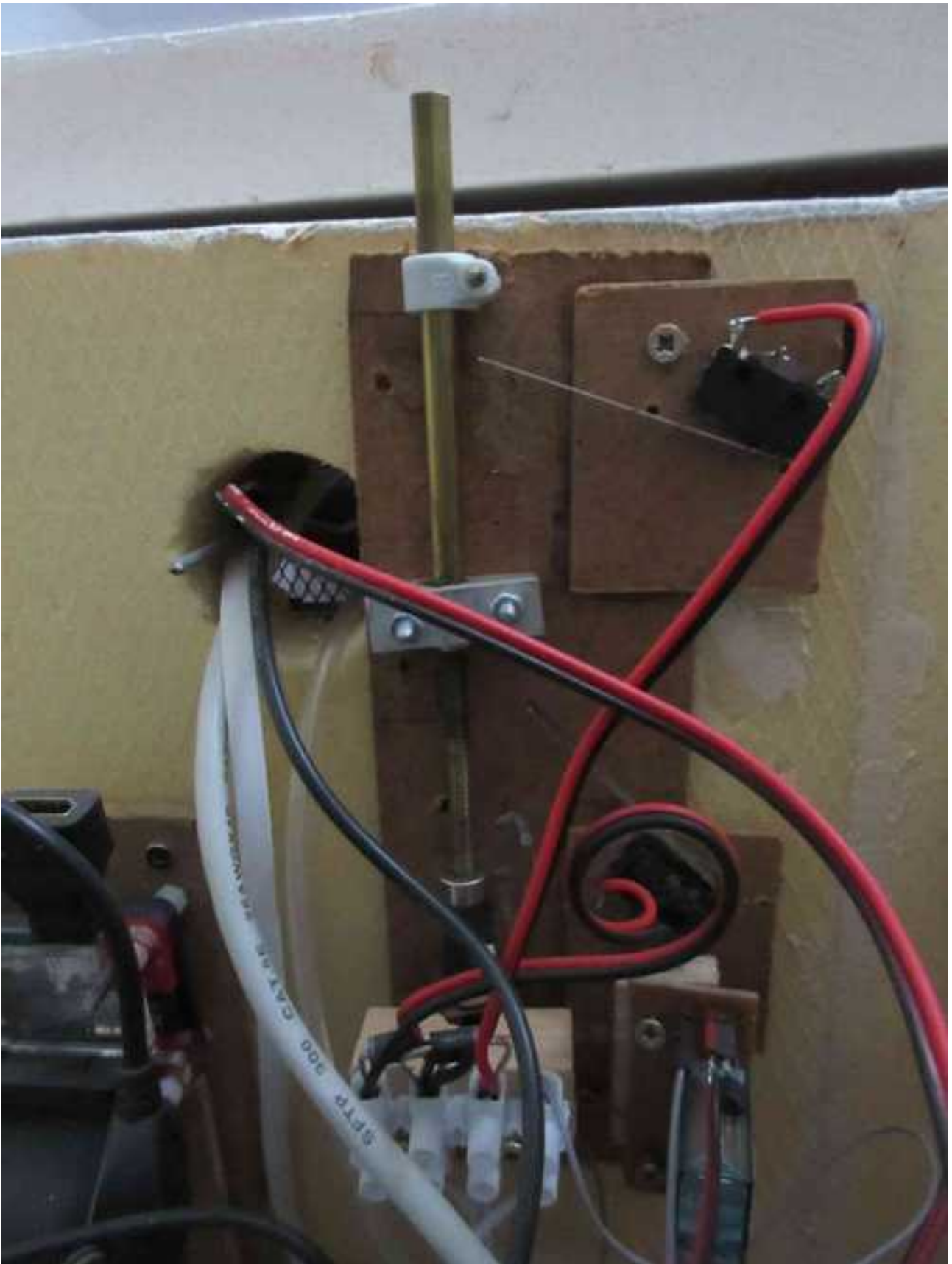
Die Idee des Lifts ist aus dem Problem geboren, das bei der Brut von Enteneier eine Kühlung notwendig ist. Ich konnte nun zum Kühlen der Eier nicht immer Zuhause sein und musste eine Alternative finden. Die Idee ist den Deckel automatisch zur Kühlung anzuheben und die Heizung abzuschalten. Ich habe dann die Seite eines Modellbauers gefunden, welcher einen mechanischen Hubzylinder für sein Bagger gebaut hat und dieses Prinzip als Lift verwendet.

Der Lift besteht mechanisch aus einem Messingrohr mit einer innenliegenden Gewindestange.

An das Messingrohr ist unten eine Mutter eingelötet in welcher die Gewindestange läuft.

Am Ende der Gewindestange ist über eine Modellbauwellenkupplung ein kleiner 6V Getriebemotor angebaut. Der Motor ist mit einem Holz auf welchem ich die Lüsterklemme für die Verdrahtung montiert habe befestigt. Das silberne Aluteil dient als Verdrehsicherung des Rohres und gleichzeitig als Anschlag für die Endschalter. Die Elektrik ist identisch mit der Wendeeinheit.

Die Abkühlung funktioniert mit dem Lift, hängt aber von der Raumtemperatur und Dauer der Öffnung ab. Durch die starke Isolation und Wannenform des Brüters fällt die Temperatur leider nicht sehr schnell ab, hier ist wohl noch etwas Optimierungsbedarf. Wenn die Ente das Nest verlässt, sind die Eier eher ungeschützt im Gegensatz zum Brüter. Ich habe schon erfolgreich Enten ausgebrütet, jedoch lag die Schlupfrate bei ca. 50 %, was natürlich auch andere Gründe haben kann. Profizüchter können mir gerne noch Tipps zukommen lassen.





Lagerung

2 x täglich Eier 45 Grad wenden(Eierkarton, Spitze nach unten) während der Sammelphase optimal bis ca. 14 Tage bei 8-15 Grad und 75% Luftfeuchtigkeit.

Eier desinfizieren

Enteneier sollten vor dem Einlegen desinfiziert werden um die Keime durch die Brut nicht zu vermehren.

Auf 5 Liter lauwarmes Wasser gebe man einen Esslöffel 18% gereinigte Salzsäure, welche für ein paar Euro in der Apotheke zu bekommen ist. Die Eier verbleiben so lange in der Lösung, bis sich über ihnen ein Ring aus feinen Bläschen, der so genannte „Heiligenschein“ gebildet hat. Man beginnt dann mit der Reinigung unter Zuhilfenahme eines Haushaltsschwammes (weiche Seite). Bitte nicht in Ungeduld verfallen und überdosieren. Die Bildung des Ringes über den Eiern kann rund 5 Minuten dauern. Man ist gut beraten das Waschen zügig durchzuführen. Es empfiehlt sich ein Probewaschen mit ein paar „unwichtigen“ Eiern. Nach dem Waschen spült man die Eier in klarem, leicht temperiertem Wasser noch einmal ab. Beschriftet mit einem Bleistift und ab geht es in den bereits laufenden und exakt eingestellten Brutapparat.

Brutstart

Eier vor dem Einlegen schieren! Ältere Eier 1/2 Tag früher einlegen.

Die Eier langsam auf Bruttemperatur bringen.

Lagerung Steuerdatei

Datei brutplan-lagerung.csv

Brutplan Lagerung bis zur Brut;;;;;

;;;;;

Tag;Temp. in Grad Celsius ;Luftfeuchte in % ;Wendung/täglich;Kühlung in Minuten ;Bemerkungen

1;10;45;3;0;

2;10;45;3;0;

3;10;45;3;0;

4;10;45;3;0;

5;10;45;3;0;

6;10;45;3;0;

7;10;45;3;0;

8;10;45;3;0;

9;10;45;3;0;

10;10;45;3;0;

11;10;45;3;0;

12;10;45;3;0;

13;10;45;3;0;

14;10;45;3;0;

15;10;45;3;0;

16;10;45;3;0;

17;10;45;3;0;

18;10;45;3;0;

19;10;45;3;0;

20;10;45;3;0;

21;10;45;3;0;

22;10;45;3;0;

23;10;45;3;0;

24;10;45;3;0;
25;10;45;0;0;
26;10;45;0;0;
27;10;45;0;0;
28;10;45;0;0;
29;10;45;0;0;
30;10;45;0;0;
31;10;45;0;0;

Hühner Steuerdatei

Datei brutplan-huehner.csv:

```
"Brutplan Huehner";;;;
;;;;
"Tag";"Temp. in Grad Celsius "; "Luftfeuchte in % "; "Wendung/täglich"; "Kühlung in Minuten"
"; "Bemerkungen"
1;37.5;45;0;0;
2;37.5;45;0;0;
3;37.5;45;0;0;
4;37.5;45;24;0;
5;37.5;45;24;0;
6;37.5;45;24;0;"Schieren"
7;37.5;45;24;0;
8;37.5;45;24;0;
9;37.5;45;24;0;
10;37.5;45;24;0;
11;37.5;45;24;0;
12;37.5;45;24;0;"Schieren"
13;37.5;45;24;0;
14;37.5;45;24;0;
15;37.5;45;24;0;
16;37.5;45;24;0;
17;37.5;45;24;0;
18;37.5;65;0;0;"Schieren, Schlupfhorde umbetten"
19;37.5;65;0;0;"Aufzucht der Hühnerküken vorbereiten"
20;37.5;65;0;0;
21;37.5;65;0;0;"Schlupf"
22;37.5;65;0;0;"Reserve"
23;37.5;65;0;0;"Reserve"
24;37.5;65;0;0;"Reserve"
```

Enten Steuerdatei

Kuehlung für Enteneier siehe Kapitel Lift.

Datei brutplan-enten.csv:

Brutplan Enten;;;;;

;;;;;

Tag;Temp. in Grad Celsius ;Luftfeuchte in % ;Wendung/täglich;Kühlung in Minuten ;Bemerkungen

1;37.7;55;3;0;

2;37.7;55;3;0;

3;37.7;55;3;0;

4;37.7;55;3;0;

5;37.7;55;3;1x10;mit Wasser bespruehen

6;37.7;55;3;1x10;mit Wasser bespruehen

7;37.7;55;3;1x10;mit Wasser bespruehen/Eier schieren

8;37.7;55;3;1x10;mit Wasser bespruehen

9;37.7;55;3;1x10;mit Wasser bespruehen

10;37.7;55;3;1x20;mit Wasser bespruehen

11;37.7;55;3;1x20;mit Wasser bespruehen

12;37.7;55;3;1x20;mit Wasser bespruehen

13;37.7;55;3;1x20;mit Wasser bespruehen

14;37.7;55;3;1x20;mit Wasser bespruehen

15;37.7;55;3;1x20;mit Wasser bespruehen

16;37.7;55;3;2x20;mit Wasser bespruehen

17;37.7;55;3;2x20;mit Wasser bespruehen

18;37.7;55;3;2x20;mit Wasser bespruehen

19;37.7;55;3;2x20;mit Wasser bespruehen

20;37.7;55;3;2x20;mit Wasser bespruehen

21;37.7;55;3;2x20;mit Wasser bespruehen

22;37.7;55;3;2x20;mit Wasser bespruehen

23;37.3;85;3;2x20;mit Wasser bespruehen

24;37.3;85;3;2x20;mit Wasser bespruehen

25;37.3;85;0;2x20;Eier schieren, in Schlupfhorde umlegen, mit Wasser bespruehen

26;37.3;85;0;2x20;mit Wasser bespruehen

27;37.3;85;0;2x20;

28;37.3;85;0;0;Schlupf

29;37.3;85;0;0;Reserve

30;37.3;85;0;0;Reserve

31;37.3;85;0;0;Reserve

Programmcode

```
#!/usr/bin/regina
```

```
/* REXX */
```

```
/* **** */
```

```
/* REXX STARTED FROM incubatorcontrol.rexx */
```

```
/* AUTOR: B. BRUECKMANN 20.01.2018 */
```

```
/* DOC: control your incubator */
```

```
/* Create controlfile with: */
```

```
/* CALL: regina incubatorcontrol.rexx Brutplan-Huehner.csv */
```

```

/* Run with:                                     */
/* CALL: regina incubatorcontrol.rexx Brutplan-Huehner.csv */
/*****/
/***** INIT VARIABLES AT TOP OF PROGRAMM *****/

TRACE off

say "Incubatorcontrol starting"

Say "Actual PATH-Value "Value( 'PATH', , 'ENVIRONMENT' )"

options arexx_bifs STDOUT_FOR_STDERR /* trace all output to file */

pid=getpid()

ADDRESS SYSTEM "echo "pid" >/home/pi/incubatorcontrol.rexx.pid"

debug="y" /* n fuer Debug and logging */

record="y" /* record sensors data for replay to check program */
replay="n" /* replay sensor data from file for testing */

homedir="/home/pi/"

webdir="/var/www/http/incubator.com/"

log=homedir"incubatorcontrol.log" /* Log */

replaypos=0 /* recordposition for replay */
checkintv=5 /* Interval sec. to check values */

heaterdelay=4 /* Number of cycles heater is off, heater ceramicbulb=5 lightbulb=0 */
heateron=6 /* number of cycles heater is on */
lastheatcnt=0 /* counter of cycles heater is on */

temptolerance=0005 /* temp tolerance for switch heater on */

filesensors=homedir"|"incubatorcontrol.check" /* Programm check with Sensor-Data from File */

if record='n' then do
    RC=XEXIST(filesensors)
    if RC=1 then do
        ADDRESS SYSTEM 'rm 'filesensors /* delete old check file */
    end
end

f=2 /* index filesensors */

htmlparm=webdir|"incubatorindex.html" /* dataset form html parameter page */
restartinfo=homedir|"restart.txt" /* last status of program */
lasthum=0 /* last humidity of interval */
lastpump="00" /* last time minute pump start*/

if record='y' then do
    ADDRESS SYSTEM 'echo "date time temp hum turn cool actcool actturner " >>'homedir"incubatorcontrol.check' /* update replay file */
end

ADDRESS SYSTEM '(sudo systemctl status watchdog.service|grep disabled) >LIFO' ; parse pull RETVAL

/* warning if watchdog is not active */

IF index(RETVAL,"disabled")>0 THEN say "Warning ! Watchdog disabled"

```

```

/***** END INIT VARIABLES *****/

parse arg parm debug
debug=lower(debug)
if parm="" then
do
say " Error no Parmfile"
say " Call with regina incubatorcontrol.rexx incubatorcontrol.conf"
exit 8
end
say "Incubatorcontrol started with "parm debug

/* Interfaces Init */

gpio_heater=16
gpio_heater_on=0
gpio_heater_off=1
gpio_eggtturn=17
gpio_eggtturn_on=0
gpio_eggtturn_off=1
gpio_humi=20
gpio_humi_on=0
gpio_humi_off=1
gpio_lift=5
gpio_lift_up=0 /* 2,5 min oeffnungszeit */
gpio_lift_down=1
gpio_reserve=24

/* 4 Relais */

if replay='n' then ADDRESS SYSTEM 'sudo chmod 222 /sys/class/gpio/export /sys/class/gpio/unexport'

sleep(2)

/* Heater Gpio */

if replay='n' then ADDRESS SYSTEM 'sudo echo ""gpio_heater"" > /sys/class/gpio/export' /* Port freischalten */

sleep(2)

if replay='n' then ADDRESS SYSTEM 'sudo echo "out" > /sys/class/gpio/gpio'gpio_heater'/direction' /* Richtung einstellen */

sleep(2)

if replay='n' then ADDRESS SYSTEM 'sudo echo "1" > /sys/class/gpio/gpio'gpio_heater'/value' /* Position */

heater_on="echo ""gpio_heater_on"" > /sys/class/gpio/gpio"gpio_heater"/value" /* set heater on */

heater_off="echo ""gpio_heater_off"" > /sys/class/gpio/gpio"gpio_heater"/value" /* set heater off */

/* Eggturner Gpio */

if replay='n' then ADDRESS SYSTEM 'sudo echo ""gpio_eggtturn"" > /sys/class/gpio/export' /* Port freischalten */

sleep(2)

if replay='n' then ADDRESS SYSTEM 'sudo echo "out" > /sys/class/gpio/gpio'gpio_eggtturn'/direction' /* Richtung einstellen */

sleep(2)

```

```

eggturner_on="echo "'gpio_eggturn_on'" > /sys/class/gpio/gpio"gpio_eggturn"/value" /* set eggturner on */
eggturner_off="echo "'gpio_eggturn_off'" > /sys/class/gpio/gpio"gpio_eggturn"/value" /* set eggturner off */
eggturner_status="cat /sys/class/gpio/gpio"gpio_eggturn"/value >LIFO' ; parse pull RETVAL" /* eggturner Status */
/* Humidity Gpio */

if replay='n' then ADDRESS SYSTEM 'sudo echo "'gpio_humi'" > /sys/class/gpio/export' /* Port freischalten */
sleep(2)

if replay='n' then ADDRESS SYSTEM 'sudo echo "out" > /sys/class/gpio/gpio"gpio_humi"/direction' /* Richtung einstellen */
sleep(2)

if replay='n' then ADDRESS SYSTEM 'sudo echo "'gpio_humi_off'" > /sys/class/gpio/gpio"gpio_humi"/value' /* Position */
humidity_on="echo "'gpio_humi_on'" > /sys/class/gpio/gpio"gpio_humi"/value" /* set humidity on */
humidity_off="echo "'gpio_humi_off'" > /sys/class/gpio/gpio"gpio_humi"/value" /* set humidity off */
/* unused relais */

if replay='n' then ADDRESS SYSTEM 'sudo echo "'gpio_lift'" > /sys/class/gpio/export' /* Port freischalten */
sleep(2)

if replay='n' then ADDRESS SYSTEM 'sudo echo "out" > /sys/class/gpio/gpio"gpio_lift"/direction' /* Richtung einstellen */
sleep(2)

if replay='n' then ADDRESS SYSTEM 'sudo echo "1" > /sys/class/gpio/gpio"gpio_lift"/value' /* Position */
sleep(2)
/* unused relais */

if replay='n' then ADDRESS SYSTEM 'sudo echo "'gpio_reserve'" > /sys/class/gpio/export' /* Port freischalten */
sleep(2)

if replay='n' then ADDRESS SYSTEM 'sudo echo "out" > /sys/class/gpio/gpio"gpio_reserve"/direction' /* Richtung einstellen */
sleep(2)

if replay='n' then ADDRESS SYSTEM 'sudo echo "1" > /sys/class/gpio/gpio"gpio_reserve"/value' /* Position */
sleep(2)
/* control Lift */

lift_up="echo "'gpio_lift_up'" > /sys/class/gpio/gpio"gpio_lift"/value" /* set lift up */
lift_down="echo "'gpio_lift_down'" > /sys/class/gpio/gpio"gpio_lift"/value" /* set lift down */
/* */

/* remove old At-Jobs */
ADDRESS SYSTEM "sudo rm /var/spool/cron/atpool/*"
ADDRESS SYSTEM "sudo rm /var/spool/cron/atjobs/*"
/* */

inp.0=5
inp.1="#/bin/sh"
inp.2=eggturner_on
inp.3="day=`cat "homedir"restart.txt |awk '{print $1}'`"
inp.4='echo "$day 1" >homedirrestart.txt'
inp.5=""
ADDRESS SYSTEM "rm "homedir"eggturn1.sh"

```

```

RC=XLINEOUT(homedir"eggturn1.sh",'inp.', '*')

inp.0=5

inp.1="#/bin/sh"

inp.2=eggturner_off

inp.3="day=`cat "homedir"restart.txt |awk '{print $1}'`"

inp.4='echo "$day 0" >'homedir'restart.txt'

inp.5=""

ADDRESS SYSTEM "rm "homedir"eggturn0.sh"

RC=XLINEOUT(homedir"eggturn0.sh",'inp.', '*')

if replay='n' then ADDRESS SYSTEM 'chmod +x 'homedir'eggturn0.sh'

if replay='n' then ADDRESS SYSTEM 'chmod +x 'homedir'eggturn1.sh'

eggturner_position='off'                /* eggturner position */

eggturner_time='00:00'                  /* time last egg turn */


sensor_temp=homedir"incubator_sensor_temp.sh" /* read temperatur sensor */

/*sensor_temp="owread /28.136756020000/temperature" */ /* read temperatur sensor Kombi-Sensor*/

/* sensor_humidity="cat /sys/bus/w1/devices/28-000002284eae/w1_slave" */ /* read humidity sensor TM3D*/

sensor_humidity=homedir"incubator_sensor_humidity.sh" /* read humidity sensor Kombi-Sensor*/

/*sensor_humidity='cat /sys/bus/w1/devices/26-0000011137d6/temperature|xargs printf "scale=2;%.2d/100\n" | bc -l' */ /* read humidity sensor
Kombi-Sensor*/

acttemp=0    /* actual temperature */

acthum=0     /* actual humidity value */

acthumrelais=0 /* actual humidity relais status */

actheater=0   /* actual heater status */

actturner=0  /* actual eggturner status */

actcool=0    /* actual cooling */

CALL LIFT actcool

/* Second Program for monitoring heater to secure hangup of this program */

ADDRESS SYSTEM "ps -ef|grep monitor.sh|grep -v grep|awk '{print $2}'|xargs kill -9"

inp.0=13

inp.1="#/bin/sh"

inp.2="# switch heater off to prevent overheating when system hangs"

inp.3="COUNTER=1"

inp.4="until [ $COUNTER -gt 1 ]; do"

inp.5=" temp="x2c(60)""homedir"incubator_sensor_temp.sh"x2c(60)

inp.6=" temp="x2c(60)"echo "x2c(22)"scale=4;$temp * 100"x2c(22)"|bc|awk 'BEGIN {FS="x2c(22)".x2c(22)"}{print $1}'""x2c(60)

inp.7=" if [ $temp -gt 3800 ]"

inp.8=" then echo '0' > /sys/class/gpio/gpio"gpio_heater"/value"

inp.9=" fi"

inp.10=" COUNTER=1"

```

```

inp.11=" sleep 60"

inp.12="done"

inp.13=""

ADDRESS SYSTEM "rm "homedir"monitor.sh"

RC=XLINEOUT(homedir"monitor.sh",'inp.', '*')

if replay='n' then ADDRESS SYSTEM 'chmod +x 'homedir'monitor.sh '

if replay='n' then ADDRESS SYSTEM 'sh 'homedir'monitor.sh &'

/*                                     */

RC=XLINEIN('sensors.',filesensors,1, '*')

RC=XLINEIN('inp.',parm,1, '*')

if inp.0=0 then do

say "no records found in input-dataset " lower(parm)

exit 8

end

/* read data from shell cmd */

/* data=""sensorcmds"sensor6.sh >LIFO' ; parse pull RETVAL" */

/* say data */

/* xx='echo exec("cat $file");' */

actdate="01/01/01" /* dummy date */

RC=XEXIST(restartinfo)

if RC=0 then

i=3 /* 3 start line */

else do

buffer=linein(restartinfo,1)

off=word(buffer,1) /* 1st word actual breed-day */

if off<3 then do

say "Error in Restart.txt"

exit 8

end

actturner=word(buffer,2) /* last position of eggturner */

if actturner=1 then ADDRESS SYSTEM eggturner_on

i=off-1 /* 3 start line + offset from restart */

end

oneaday=0

do while i<=inp.0

if replay='n' then do

acttime=substr(TIME('N'),1,8)

if actdate<>DATE('E') then do

commentsend=""

actdate=DATE('E') /* actual date */

```

```

oneaday=1 /* flag for only one action a day */

i=i+1

end

end

else do /* replay = yes */

dte=word(sensors.f,1)

if length(dte)=0 then do

say "End of Replay-file"

exit 8

end

if dte<>actdate then

i=i+1

actdate=dte

acttime=substr(word(sensors.f,2),1,8)

acttemp=word(sensors.f,3)

acthum=word(sensors.f,4)

eggturn=word(sensors.f,5)

actcool=word(sensors.f,6)

actturner=word(sensors.f,7)

say "Simulate with Check-Record="f" actdate="actdate" acttime="acttime" acttemp="acttemp" acthum="acthum" eggturn="eggturn"
actcool="actcool"

end

if debug="y" then do

say inp.i

end

data=eggturner_status

if replay='n' then do

interpret data /* status of eggturner */

actturner=RETVAl

end

buffer=i" "actturner

dummy=LINEOUT(restartinfo,buffer,1)

parse value inp.i with day ';' temp ';' humidity ';' turn ';' cool ';' comment

say "Brutplan i="i " day="day" temp="temp" humidity="humidity" turn="turn" actturner="actturner" cool="cool" comment="comment"

if comment<>"" & commentsend="" then do

say comment" "commentsend "=yes"

commentsend="yes"

ADDRESS SYSTEM 'echo "'comment'" ">'homedir'mailtext.txt' /* update replay file */

ADDRESS SYSTEM "sh "homedir"sendmailinfo.sh"

end

```

```

if oneaday=1 then do
    call EGGTURNER turn debug
    oneaday=0
end

call HEATER temp lastheatcnt heaterdelay heateron temptolerance debug

if actheater=1 then do
    if replay='n' then ADDRESS SYSTEM 'fswebcam -c 'homedir'cam.conf&'
end

/* call EGGTURNER turn debug */

debug="y"

call HUMIDITY humidity lasthum lastpump actcool debug

parse var result RC lasthum lastpump

call COOL cool debug

CALL LIFT actcool

printing=actdate" "acttime" Brutplantag="day" Solltemperatur="temp" Aktuelle-Temperatur="acttemp" Sollfeuchtigkeit="humidity" Aktuelle-
Feuchtigkeit="acthum

printing=printing" Act-Humidity-Relais="acthumrelais" Heater-ON/OFF="actheater" Turn="turn" Eggturner-ON/OFF="actturner" Cool="cool"
Aktuelles-Kuehlen="actcool

printing=printing" Comment="comment

say printing

data=' echo "'printing'" >>'homedir'incubator.values' /* write history data of incubator values */

ADDRESS SYSTEM data

data=' echo "'printing'" >'webdir'sensors_input.txt' /* write actual values for rrdtool graphics */

ADDRESS SYSTEM data

data=homedir"sensors-rrd.sh update"

ADDRESS SYSTEM data /* update rrdtool */

if replay='y' then do
    if f<sensors.0 then f=f+1
    else exit 8
end

if substr(acttime,4,2)="01" then do /* check every hour */
    e=linein(restartinfo,1) /* 1st word actual breed-day */
    actturner=word(e,2)
end

if record='y' then do
    ADDRESS SYSTEM 'echo 'actdate' 'acttime' 'acttemp' 'acthum' ' ' turn' 'cool' 'actturner' >>'homedir'incubatorcontrol.check' /* update replay file
*/

end

if replay='n' then sleep(checkintv) /* check every sec. */

end

exit 0

```

```

HEATER:

/* control heater */

ARG temp lastheatcnt heaterdelay heateron temptolerance debug
debug=lower(debug)
actheater=0
if replay='n' then do
  /* read actual temp */
  data=""sensor_temp" >LIFO' ; parse pull RETVAL"
  interpret data
  /*
  buffer=translate(RETVAL,',' '=' )
  acttemp=word(buffer,2)
  acttemp=acttemp/100    */
  acttemp=RETVAL
  if index(acttemp,"not found")>0 then actemp="0"
  do while index(actemp,"ServerRead:")>0 | acttemp= 85  /* 85 grad = error temp */
    data=""sensor_temp" >LIFO' ; parse pull RETVAL"
    interpret data
    actemp=RETVAL
  end
  if actcool=1 then do
    actheater=0
    if replay='n' then ADDRESS SYSTEM heater_off
    return 0; /* cooling in progress */
  end
end

if acttemp*100-temptolerance < temp*100 then do /* temp to low */
  actheater=1
  if lastheatcnt >= heateron then do /* prevent overheating because mass of bulbs */
    if lastheatcnt > heateron then lastheatcnt=lastheatcnt-heateron-1
    if lastheatcnt = heateron then lastheatcnt=lastheatcnt+heateron*heaterdelay
    if lastheatcnt < heateron then lastheatcnt=0
    if debug="y" then say "Prevent overheating"
    actheater=0 /* prevent heat */
  end
end

if actheater=1 then do
  if debug="y" then say "heater on"
  if replay='n' then do
    ADDRESS SYSTEM heater_on /* temp to low */

```

```

    lastheatcnt=lastheatcnt+1
end
end
else do
if debug="y" then say "heater off"
actheater=0
if replay='n' then ADDRESS SYSTEM heater_off      /* temp to high */
end

if debug="y" then say "Solltemperatur "temp" Aktuelle Temperatur "acttemp" Lastheatcnt "lastheatcnt" Heateron "heateron" Heaterdelay
"heaterdelay

return 0
EGGTURNER:
debug=lower(debug)
/* control eggturner */
ARG eggturn debug
/* eggturning */
if turn>0 then do
time=24/turn
do e=1 to turn
if time*e>substr(acttime,1,2) then do
tme=time*e":00"
if tme="24:00" then tme="23:50"
if actturner=0 then do
if replay='n' then do
ADDRESS SYSTEM 'echo 'homedir'"eggturn1.sh"|at 'tme
say "start eggturn1.sh at "tme
actturner=1
end
end
else do
if replay='n' then do
ADDRESS SYSTEM 'echo 'homedir'"eggturn0.sh"|at 'tme
say "start eggturn0.sh at "tme
actturner=0
end
end
end
end
end
/* end eggturning */

```

```
return 0
```

```
HUMIDITY:
```

```
/* control humidity */
```

```
ARG humidity lasthum lastpump actcool debug
```

```
debug=lower(debug)
```

```
l=0
```

```
if replay='n' then do
```

```
lasthum=acthum
```

```
data=""sensor_humidity" >LIFO' ; parse pull RETVAL"
```

```
interpret data
```

```
acthum=RETVAL
```

```
do while index(acthum,"ServerRead:")>0 | acthum="" | acthum>100 | acthum=0
```

```
data=""sensor_humidity" >LIFO' ; parse pull RETVAL"
```

```
interpret data
```

```
acthum=RETVAL
```

```
l=l+1
```

```
if l=10 then do
```

```
acthum=0 /* permanenter lesefehler */
```

```
return 0
```

```
end
```

```
end
```

```
end
```

```
if acthum>humidity then do
```

```
if debug="y" then say actdate" "acttime" Actual humidity "acthum" Sollhumidity "humidity" humidity to high Lasthum "lasthum" Lastpump Minute  
"lastpump
```

```
end
```

```
if acthum<humidity & actcool=0 then do
```

```
if debug="y" then say actdate" "acttime" Actual humidity "acthum" Sollhumidity "humidity" humidity to low Lasthum "lasthum" Lastpump Minute  
"lastpump
```

```
if replay='n' then do
```

```
if substr(acttime,4,2)<>lastpump then do /* prevent pumping in same minute */
```

```
lastpump=substr(acttime,4,2)
```

```
ADDRESS SYSTEM humidity_on
```

```
end
```

```
end
```

```
acthumrelais=1
```

```
sleep(1) /* wait for water flow */
```

```
if replay='n' then ADDRESS SYSTEM humidity_off
```

```
acthumrelais=0
```

```

end

return 0 lasthum lastpump

COOL:

/* control cooling */

ARG cool debug

debug=lower(debug)

cool=lower(cool)

h=substr(acttime,1,2)

m=substr(acttime,4,2)

min=h*60+m

if debug="y" then say "minutes (cool)="+min" acttime="+acttime

if cool="1x10" then do

if (min>=480 & min<490) then do /* start cooling 08:00 - 08:10 */

actcool=1 /* set cooling flag */

say "Cooling for "cool" Minutes at "acttime

end

else do

actcool=0 /* set cooling flag */

say "Cooling off"

end

end

if cool="1x20" then do

if (min>=480 & min<500) then do /* start cooling 08:00 - 08:20 */

actcool=1 /* set cooling flag */

say "Cooling for "cool" Minutes at "acttime

end

else do

actcool=0 /* set cooling flag */

say "Cooling off"

end

end

if cool="2x20" then do

if (min>=480 & min<500) && (min>=960 & min<980) then do /* start cooling 08:00-08:20, 16:00-16:20*/

actcool=1 /* set cooling flag */

say "Cooling for "cool" Minutes at "acttime

end

else do

actcool=0 /* set cooling flag */

say "Cooling off"

end

end

```

```

end

return 0

LIFT:

/* control Lift */

ARG actcool debug
debug=lower(debug)

if actcool="1" then

if replay='n' then ADDRESS SYSTEM lift_up

if actcool="0" then

if replay='n' then ADDRESS SYSTEM lift_down

return 0

/* REXX */

/*****

/* REXX STARTED FROM &SYSUID..USER.CLIST(HOSTPCIO)          **/

/*  AUTOR: B. BRUECKMANN 01/12/99                          **/

/*  DOC: REXX FUER KOMPATIBLE PC/HOST-IO                    **/

*****/

/***** INIT VARIABLES AT TOP OF PROGRAMM *****/

SYSTEM=LINUX          /* BETRIEBSSYSTEM DOS/MVS/VM */

/***** END INIT VARIABLES *****/

XEXIST:

/* FUNCTION: TEST DATASET-EXISTS          */

/* SYNTAX:                               */

/*      RC=XEXISTS(DSN)                  */

/* RC=0 DATASET NOT EXISTS                */

/* RC=1 DATASET EXISTS                    */

ARG XEXISTDS

XEXISTDS=lower(XEXISTDS)

ADDRESS SYSTEM '(ls 'XEXISTDS' >> /dev/null 2>&1 && echo yes) || echo no >LIFO' ; parse pull RETVAL

IF RETVAL = "no" THEN RETURN 0

CALL CLOSE FILE

RETURN 1

XMEM:

ARG XMEMDSN, XMEMNAME

/* FUNCTION: GET MEMBER NAMES          */

/* SYNTAX:                               */

/*      RC=XMEM(DSN,'MEM.')              */

XMEMDSN=STRIP(XMEMDSN,'T','/')

XMEMDSN=TRANSLATE(XMEMDSN,'\\','/') /* TRANSLATE FILE-SYNTAX */

DATA=""DIR 'XMEMDSN' >$XMEM.TXT"

```

```

INTERPRET DATA

XMEMI=1    /* SATZANZAHL */

/* SKIP OVER FIRST BLANK RECORD */

DATA='EOFX=LINEIN($XMEM.TXT,'XMEMI',1)'

INTERPRET DATA

XMEMI=XMEMI+1

XMEMN=1    /* TABLE-COUNTER */

EOFX=0     /* EOF FLAG */

REXXTYPE='1' /* DIFFERENT REXX-VERSIONS HAVE DIFFERENT OUTPUT */

DO WHILE EOFX<4    /* MORE THAN 4 EMPTY RECORDS =EOF */

  DATA='EOFT=LINEIN("$XMEM.TXT",'XMEMI',1)'

  INTERPRET DATA

  SELECT

    WHEN EOFT="" THEN EOFX=EOFX+1

    WHEN INDEX(TRANSLATE(EOFT),'$XMEM')>0 THEN NOP

    WHEN INDEX(TRANSLATE(EOFT),'<DIR>')>0 THEN DO

      IF WORD(EOFT,2)='<DIR>' THEN REXXTYPE='1' /*BREXX */

      IF WORD(EOFT,3)='<DIR>' THEN REXXTYPE='2' /*IBM-REXX*/

    END

    WHEN INDEX(TRANSLATE(EOFT),'BYTES')>0 THEN NOP

    WHEN INDEX(TRANSLATE(EOFT),'DATENTR')>0 THEN NOP

    WHEN INDEX(TRANSLATE(EOFT),'VOLUME IN DRIVE ')>0 THEN NOP

    WHEN INDEX(TRANSLATE(EOFT),'VOLUME SERIAL ')>0 THEN NOP

    WHEN INDEX(TRANSLATE(EOFT),'DIRECTORY OF ')>0 THEN NOP

    WHEN INDEX(TRANSLATE(EOFT),'VERZEICHNIS ')>0 THEN NOP

    WHEN INDEX(TRANSLATE(EOFT),'FILE(S)')>0 THEN NOP

    WHEN INDEX(TRANSLATE(EOFT),'DATEI(EN)')>0 THEN NOP

    WHEN INDEX(TRANSLATE(EOFT),'BYTES FREE')>0 THEN NOP

    OTHERWISE DO

      EOFX=0    /* RESET EOF */

      IF REXXTYPE='1' THEN DO

        IF WORDS(EOFT)=5 THEN /* DATASET WITH SUFFIX */

          DATA=XMEMNAME"XMEMN'=WORD(EOFT,1)".'WORD(EOFT,2)

        ELSE

          DATA=XMEMNAME"XMEMN'=WORD(EOFT,1)'

        INTERPRET DATA

        XMEMN=XMEMN+1

      END

      IF REXXTYPE='2' THEN DO

        DATA=XMEMNAME"XMEMN'=WORD(EOFT,4)'

```

```

        INTERPRET DATA

        XMEMN=XMEMN+1

    END

    END

END

XMEMI=XMEMI+1

END

DATA=XMEMNAME"0=XMEMN-1"

INTERPRET DATA

"rm $XMEM.TXT"

RETURN 0

XDEL:

/* FUNCTION: DELETE DATASET          */
/* SYNTAX:                          */
/*      RC=XDEL(DSNO)                */
/* OR      RC=XDEL("TEST.XXX")      */
ARG XDELDSN
XDELDSN=lower(XDELDSN)
SAY DELETE XDELDSN
"rm "XDELDSN""

RETURN 0

XLINEIN:

ARG DEST, NAME , LINE, COUNT
NAME=lower(NAME)
data=dest"0=0"

interpret data

/* SYNTAX: XX=LINEIN(NAME,1,COUNT) */
/*      DEST IST DIE ZIELVARIABLE */
/*      NAME IST DER DATEINAMEN */
/*      LINE IST DER RECORDOFFSET IM FILE */
/*      COUNT IST DER SATZZAEHLER */

IF COUNT=0 THEN RETURN 0
IF ARG(<>0) THEN DO
    LINEININ=1 /* SATZANZAHL */
    EOFX=0 /* EOF FLAG */
    IF COUNT='*' THEN
        DO WHILE EOFX=0 /* MORE THAN 4 EMPTY RECORDS =EOF */
            DATA='EOFT=LINEIN("'"NAME"', 'LINEININ',1)'
            INTERPRET DATA
            IF EOFT=" THEN EOFX=EOFX+1

```

```

        ELSE EOFX=0

        DATA=DEST"LINEININ'=EOFT'

        INTERPRET DATA

        /* EOF CHECK*/

        /*IF CHARS("NAME")=0 THEN RETURN 0 */

        DATA=DEST'0='LINEININ

        INTERPRET DATA

        LINEININ=LINEININ+1

        END

    ELSE DO

        DATA=DEST'=LINEIN("NAME",'LINE','COUNT')

        INTERPRET DATA

        END

        DATA=DEST'0='LINEININ-2

        INTERPRET DATA

        END

    RETURN 0

XLINEOUT:

ARG NAME , SRC, LINE

name=lower(name)

RET=0

/* SYNTAX:                */

/*   XX="TEST"            */

/*   RC=XLINEOUT(DSN,XX,1)    */

/*   NAME IST DER DSNAME      */

/*   SRC IST DIE SOURCEVARIABLE */

/*   LINE IST DER RECORDOFFSET IM FILE */

RET=0

IF LINE<>0 THEN DO

    IF ARG()<>0 THEN DO

        IF LINE='*' THEN DO /* TABLE SAVE */

            DATA="XLINEN="SRC"0"

            INTERPRET DATA

            DO XLINEI=1 TO XLINEN

                DATA="CALL LINEOUT NAME,"SRC"XLINEI";RET=RC"

                INTERPRET DATA

            END

        END

    ELSE DO

```

Schaltplan Raspberry-Erweiterungsplatine



Teileliste

Gehäuse:

- 1xBodenplatte 430mmx800mmx30mm XPS
- 1xDeckelplatte 430mmx800mmx30mm XPS
- 1xRückwand 340mmx800mmx30mm XPS
- 1xVorderwand 340mmx800mmx30mm XPS
- 3xTrenn/Endwände 370mmx340mmx30mm XPS
- 2xDeckelfalz 975mmx30mmx20mm XPS
- 2xDeckelfalz 325mmx30mmx20mm XPS
- 1x Drahtgitter 360mmx320mm
- 2x Keramikfassung E27 gerade, 14,08 Euro, www.Aqua2Terra.de
- 1x Glühbirne 60W E27
- 1x Keramikstrahler 60W E27
- 2x 300mm Alu-U-Schiene 10x10mm (Drahtgitterführung)
- 2x 370mm Alu-U-Schiene 10x10mm (Halter Befeuchtungstuch)
- 1x Wasserschale 350mmx150mm
- 2x Handtuchstreifen 130mmx650mm (Befeuchtungstuch)
- 1x Plexiglas für Deckel
- 1x Lüftungsrosette
- 1x Holzleim wasserfest
- 1x Acryllack
- 1x Pattex PL Montagekleber 10 Euro

Wendeeinheit:

- 1x Eierhorde Plastik 300x300mm
- 1x Seilrolle Alu
- 1x Maurerschnur
- 1x Feder 170mm
- 1x Gitternetz 320mmx320mm
- 1x Getriebemotor 16rpm 5V (Wendeeinheit)
- 2x Microschalter 2,15 Euro Radio Kraus
- 2x Diode 1N5408
- 1x Lüsterklemmen
- 2x Sperrholzplatte 200mmx200mmx12mm
- 2x 350mmx45mmx25mm Dachlatte
- 2x 300mmx45mmx25mm Dachlatte
- 1x 355mmx45mmx25mm Dachlatte

Elektrik

- 1x Dosierpumpe 5V (Befeuchtung)
- 1x gebrauchter Infusionsschlauch
- 2x Messingrohr 3mmx80mm
- 1x DS18B20 1-Wire Temperatursensor
- 1x Feuchtigkeitssensor 1-Wire <https://www.tm3d.de/>
- 1x Raspberry Zero W, 16 Euro
- 1x Raspberry Netzteil, 13 Euro
- 1x Raspberry Gehäuse, 6 Euro

1x Class 10 Micro-SD-Card min. 8GB
1x USB-Stick oder anderer Adapter zum Beschreiben der SD-Card
1x HDMI-Monitor und USB-Tastatur
1x 1-fach Solid State Relais Low Level Trigger + 2-3 fach Umschalt-Relais
oder
1x 4-fach Umschalt- Relais <https://www.makershop.de/module/relais/4-kanal-relais/>

1x USB Doppelnetzteil min. 2A
1x USB-Hub aktiv
1x Webcam USB
1x 3fach Verteilersteckdose mit Kabel 1,5m
1x Lüfter 5V
1x Raspberry Expansion Board (Eigenbau):
 Erweiterungsplatine Lochraster
 1-Wechslerrelais Finder 2x UM 6V/8A FIN 40.52.9 (10 Euro)
 1x Diode 1N4001 0,3 Euro
 1x Transistor BC337 0,3 Euro
 1x Widerstand 1K 0,3 Euro
 1x RJ45-Buchse 0,5 Euro
 1x Modul 039 von www.tm3d.de (Feuchtigkeitssensor) 34,50 Euro
 1x DS18B20 1,5 Euro
 1x Widerstand 4,7K 0,3 Euro
 2x Diode 1N4001 (Motorumschalter) 0,3 Euro

Deckelheber
1x Getriebemotor 16rpm 5V

Lieferanten

<https://www.sertronics-shop.de/Raspberry-PiCo/Raspberry-Pi-Zero/BoardsKits/Raspberry-Pi-Zero-W-Zusammenstellung-Essential-Starter-Kit/>

Lüftungsrosette aus Erdnussdose

Wasserbehälter aus Konservendose

https://www.amazon.de/gp/product/B008KVGWT4/ref=s9_acsd_simh_hd_bw_b2GVu91_c_x_w?pf_rd_m=A3JWKAKR8XB7XF&pf_rd_s=merchandised-search-3&pf_rd_r=TJCW10758R4WTWD92CQ9&pf_rd_t=101&pf_rd_p=af302491-99dd-5018-9da2-51debcc27332&pf_rd_i=2076291031

<https://www.pollin.de/p/kapillarrohr-regler-mit-mehrpoliger-regelung-e-g-o-55-34022-050-260725>

<http://www.ebay.at/itm/6V-DC-Peristaltikpumpe-Schlauchpumpe-Dosierpumpe-Wasserpumpe-fur-Tank-Neu-/162670028177>

<https://www.ebay.de/itm/220V-AC-Inkbird-ITC-1000-Digitale-Temperature-regler-NTC-Fuhler-sensor-heat-cool/182537538262?hash=item2a8015cad6:g:T94AAOSw4DJYlZz0#rpdCntId>

<https://www.makershop.de/module/relais/1-2-kanal-relais-set>

<https://www.makershop.de/module/relais/4-kanal-relais/>