



instructables

SmartFridge



by BerendV

Did you ever have the problem, that you didn't really know what was left in your fridge? Or did you ever take something out of your fridge and forgot to close it? Well I most certainly did. That's why i created this project.

This SmartFridge notifies you with a beeping sound if you left the door open, and remembers which products are inside of your fridge and the amount of them.

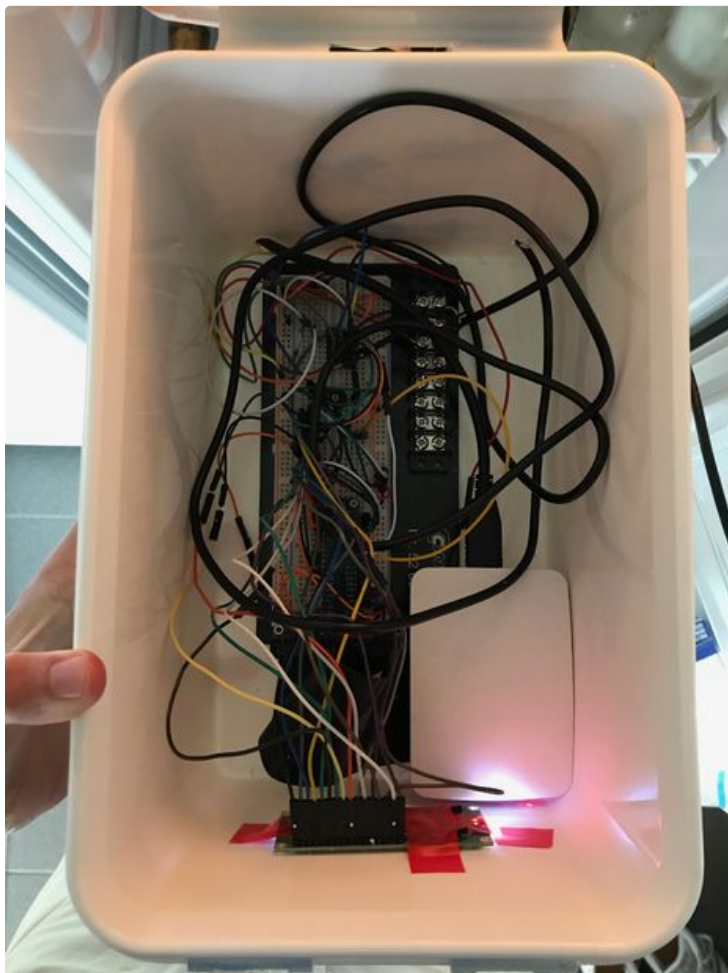
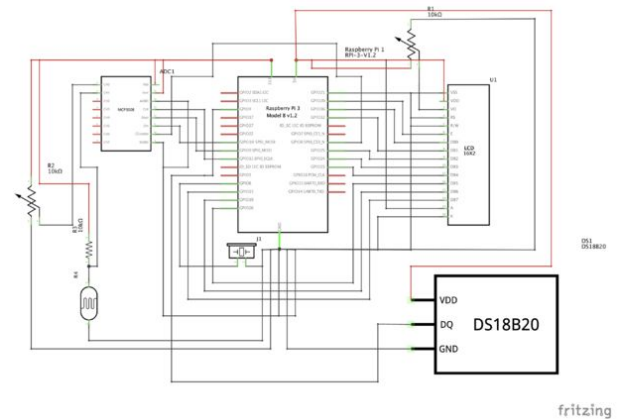
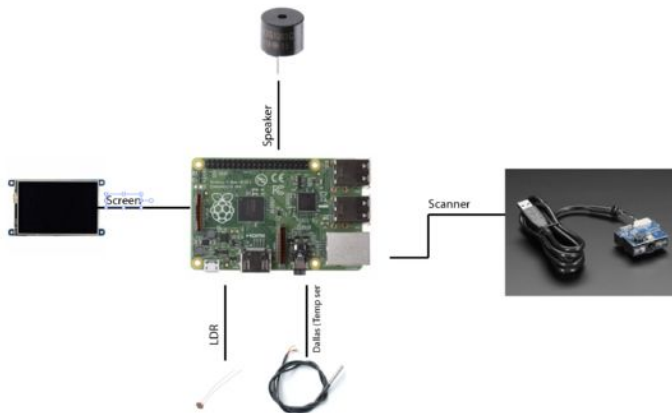


Step 1: The Electrical Circuit

First you'll need to put the circuit together, I provided a Fritzing schema to do this.

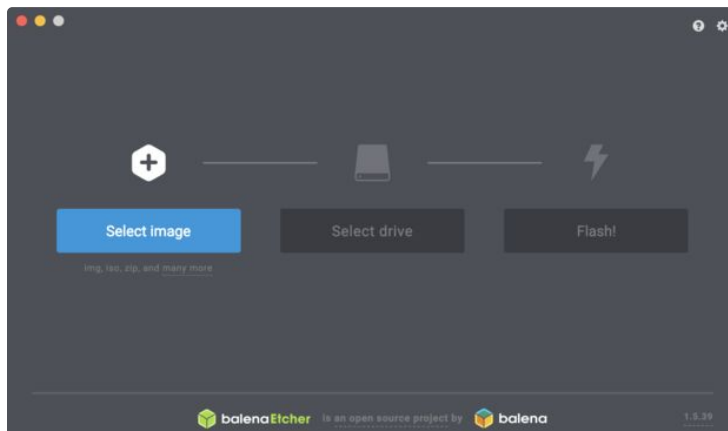
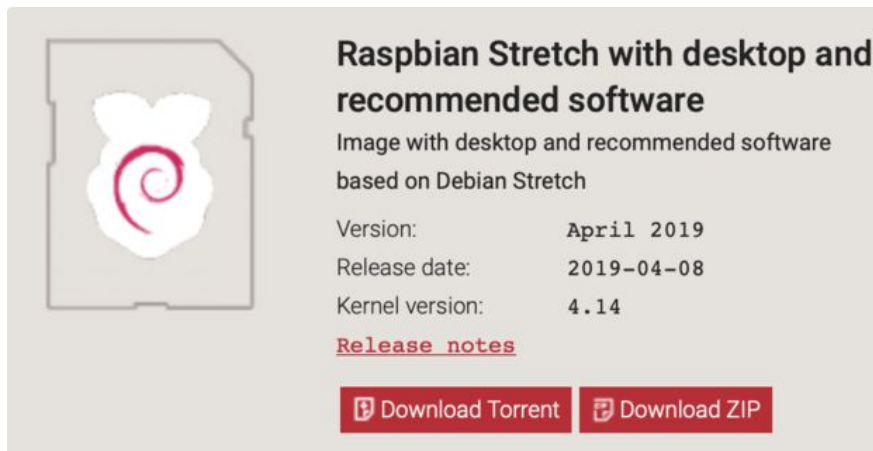
If you will use my code I provided at the step read sensor data Its important you wire te sensors and LED to the same GPIO pins, You could also just change the pins in the code.

Don't forget to connect the barcode scanner to 1 of the USB ports of the pi



Step 2: Setup Raspberry Pi

- Download the raspbian image from this website: <https://www.raspberrypi.org>
- Choose the **Raspbian Stretch with desktop and recommended software** option.
- Install Balena Etcher: <https://www.balena.io/etcher/>
- Plug-in your micro SD card to your computer
- Open Etcher & select the iso file you just downloaded from raspberrypi.org
- Select your micro SD card
- Click Flash! & wait a minute



Step 3: Change Raspbian Settings

When booted in raspbian you should change some settings:

- First connect to your wifi network
- Open a terminal window and type:
sudo raspi-config
- Go to boot options > Desktop / CLI
Choose Console
- Press enter
- Go to interfacing options
Turn on Serial and 1-wire
- Press finish and let the pi reboot

After the reboot type in the console:

- `sudo apt-get update`
- `sudo apt-get upgrade`
- `y`



Step 4: Read Sensor Data

You will use the programming language Python to read data from the sensors & send data to the LCD display.

You can find all of my code on github: <https://github.com>

You will see I use different classes to read the data from my sensors.

- The **barcode scanner** is connected to the pi via USB and uses the serial interface, As standard the scanner has been setup as a keyboard. So we will have to change this:
Use this [manual](#) and scan these codes:
 - USB COM Port Emulation
 - None (page 12 of the manual)
 - EnableEAN-8* (page 24)
 - EnableEAN-13* (page 25)
- The analog values from the **LDR** are converted using the MCP3008, then transformed from a value to a percentage.

If you want to use more then 1 while True loop you will need to import threading (this is already done in the provided code)

```
1 from RPi import GPIO
2 import time
3
4 class lcd:
5     def __init__(self, E=28, RS=21, D0=16, D1=12, D2=25, D3=24, D4=23, D5=26, D6=13, D7=13):
6         self.__pin = [D0, D1, D2, D3, D4, D5, D6, D7]
7         self.__e = E
8         self.__rs = RS
9         self.init_GPIO()
10        self.init_LCD()
11
12    def init_GPIO(self):
13        GPIO.setmode(GPIO.BCM)
14        GPIO.setup(self.__pin, GPIO.OUT)
15        GPIO.setup([self.__e, self.__rs], GPIO.OUT)
16
17    def init_LCD(self):
18        self.send_instruction(0b00110000)
19        self.send_instruction(0b00001111)
20        self.send_instruction(0b00000001)
21        self.send_instruction(0b0000001100)
22
23    def set_data_bits(self, byte):
24        for i in range(8):
25            bit = (byte >> i) & 1
26            GPIO.output(self.__pin[i], bit)
27
28    def send_instruction(self, data_byte):
29        GPIO.output(self.__e, 1)
30        GPIO.output(self.__rs, 0)
31        self.set_data_bits(data_byte)
32        GPIO.output(self.__e, 0)
33        time.sleep(0.01)
34
35    def send_character(self, data_byte):
36        GPIO.output(self.__e, 1)
37        GPIO.output(self.__rs, 1)
38        self.set_data_bits(data_byte)
39        GPIO.output(self.__e, 0)
40        time.sleep(0.01)
41
42    def write_message(self, message):
43        for k in message:
44            self.send_character(ord(k))
45
46    def second_row(self):
47        self.send_instruction(0b10000000)
```

Step 5: The MySQL Database

We will store our products in the fridge and temperature in our database.

To manage the database i'm using mysqlworkbench, In mysqlworkbench you will need to setup a connection with your raspberry pi. But for this to work we'll need to install mariaDB on our raspberry pi.

Here is how you do it:

```
sudo apt-get install mariadb-server
```

Then setup mariaDB with:

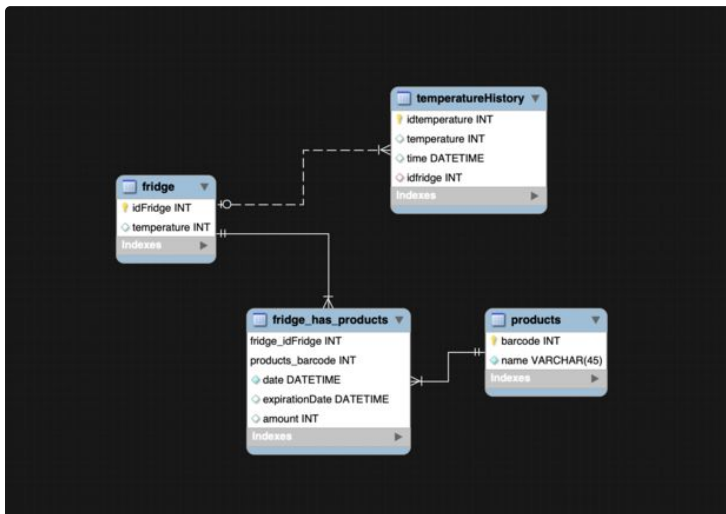
```
sudo mysql_secure_installation<br>Enter current password for root (enter for none): root
Change the root password? [Y/n] Y
New password: root123
Remove anonymous users? [Y/n] y
Disallow root login remotely? [Y/n] y
Remove test database and access to it? [Y/n] y
Reload privilege tables now? [Y/n] y
```

Connect with mysqlworkbench to the mariaDB server with the user **root** and the password you chose above on your pi's ip address.

You can find your pi's ip address by typing the following in the terminal:

```
ip a
```

Look at the ip address under wlan0 it will probably look somewhat like 192.168.X.X



<https://www.instructabl...>

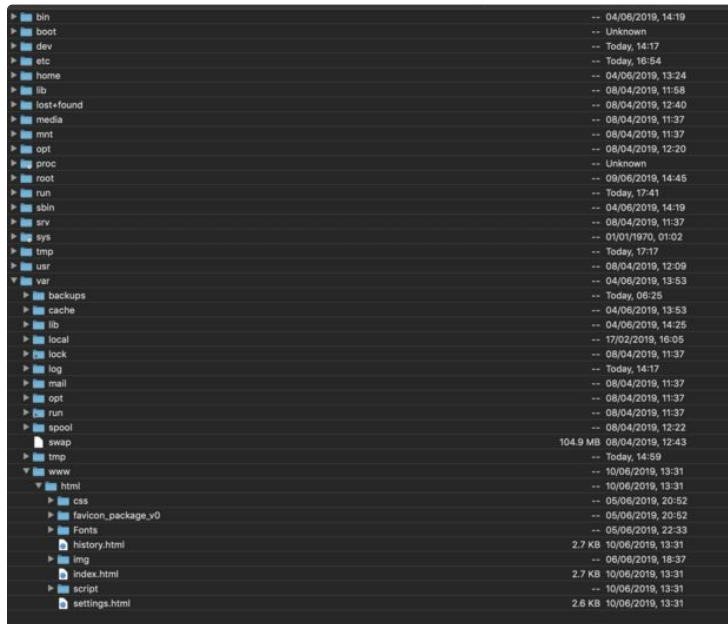
Download

Step 6: Install the Apache Webserver

Now we will go ahead and install the apache webserver by typing the following in the terminal:

```
sudo apt-get install apache2  
sudo reboot
```

Now copy the files from the directory below to the folder **/var/www/html** via SFTP using a program like Cyberduck or Filezilla on the ip address from the step before



<https://www.instructabl...>

Download

Step 7: Upload the Python Code to Your Pi

Also using cyberduck or filezilla you will need to upload the python code to your pi under **/home/user_name**
The files are provided below or on my github

Then run the script using this command:

```
python /home/<em>user_name</em>/back/app.py
```

Check if everything is working by typing in your raspberry pi's ip address in your browser.
You should see a webpage with the temperature and an empty table

Now your smartfridge should be working perfectly and we can go ahead and make it "pretty"

Note: change user_name above always with the name of the user you created on your pi or just use the user pi

SmartFridge

Instellingen

Home

7.6 °C De temperatuur in de koelkast wordt te warm!

Op dit moment in je koelkast:

Naam	Invoerbdatum	Vervaldatum	Aantal
Yonke Peeling Smeltbol	Don, 13 Jun 2018 14:51:04 GMT	naft	1
Everglades Fritjesbark	Fri, 14 Jun 2018 15:04:05 GMT	naft	10
Bom broodjeskastjes	Fri, 14 Jun 2018 15:07:29 GMT	naft	2
Comrade kippen enk kastjes	Fri, 14 Jun 2018 15:07:30 GMT	naft	2
Doritos	Fri, 14 Jun 2018 15:07:34 GMT	naft	4
Stella Aardis	Thu, 13 Jun 2018 17:09:45 GMT	naft	8
Vlaamsehe i reus	Thu, 13 Jun 2018 17:09:45 GMT	naft	1
Everglades broodjes enk	Thu, 13 Jun 2018 18:43:29 GMT	naft	2

https://www.instructabl...

Download

Step 8: The Housing

The housing that i've made is made of a plastic box you can close with clips, but you can make it in any way you want. Just don't forget to make sure the light sensor can actually see the light.

- spray paint the box white
- Drill 2 holes of 8mm diameter at the back, one for the temperature sensor and 1 for the barcode scanner
- Drill 1 hole on the left side of the box for the powercable of your pi
- Make a rectangle hole in the front of the box for the LCD display (**Don't use a knife I already tried it and cut myself**)
- Use some double sided tape to make sure your breadboard stays where it is.



Step 9: Place the Housing in the Fridge

Place your box inside the fridge

- Drill a hole through the bottom of your fridge for an extension cable.
- Fill the hole back up with some styrofoam
- Use silicon to fill the rest of the gap
- Make it smooth with a spoon with some soap on the back (so your fingers aren't sticky)
- Use some cable trucking to hide the extension cable

Now you can power your raspberry pi inside the fridge

- Use 1 of those plastic shelves from the fridge door to attach your barcode scanner to
- Make sure it doesn't move with some double sided tape
- Use some cable trucking to hide the cable



Step 10: Portforwarding

If you want to use the website to check what's inside your fridge from outside your house you will need to forward your ports in the router.

You can do this by surfing to your routers ip address in your browser, find the port forwarding page, setup a rule for your raspberry pi's ip address open port 80 to 80 and 5000 to 5000 with the TCP protocol and save the settings.

Note: To figure out your router's ip on windows, type cmd in the search bar, open the commandprompt and type: ipconfig. The default gateway will be your router ip.

On mac you go to the apple menu/System preferences/Network/advanced/TCPIP, your router ip will be router