

Software:

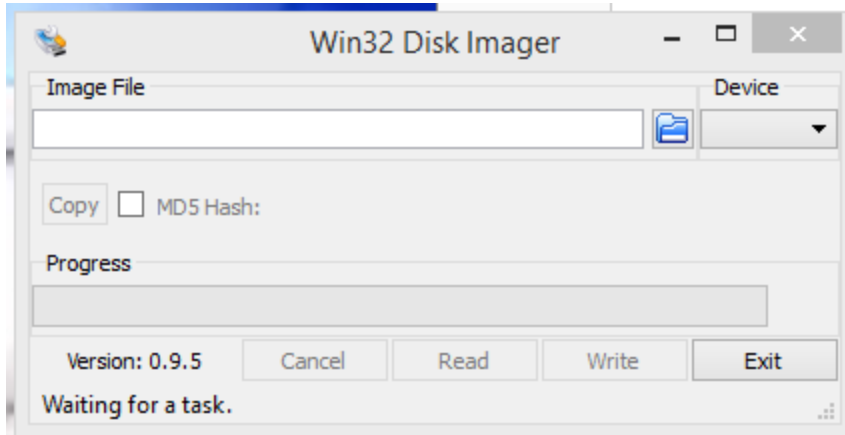
The password for GMAIL had to be changed to doorbell2

Raspberry Pi image:

<https://mega.nz/file/YJk3RDzT#dFZdAtWbs6ZBAu65Gv6DJ6aC3KF5G0JBc2lXXgwoSC8>

This was developed on a PI 2, the image may not work on the PI 3.

It is very simple to make this project, just flash the SD card using Win32 Disk imager, load the image file and press write and that's all the software done!



RUN LCD3.py!!

Here is the code in **Git**: <https://github.com/in06khattab/Doorbell-6--final->

Hardware:

I used this for the speaker

http://www.ebay.co.uk/itm/PAM8403-5V-DC-Audio-Amplifier-Board-2-Channel-3W-2-Volume-Control-USB-Power-Hot-/281516168118?pt=UK_BOI_Electrical_Components_Supplies_ET&var=&hash=item418babd3b6

This is the doorbell that I took apart:

http://www.ebay.co.uk/itm/321664088801?_trksid=p2060353.m2749.l2649&ssPageName=STRK%3AMEBIDX%3AIT

I used this camera:

http://www.ebay.co.uk/itm/271702042292?_trksid=p2060353.m2749.l2649&ssPageName=STRK%3AMEBIDX%3AIT

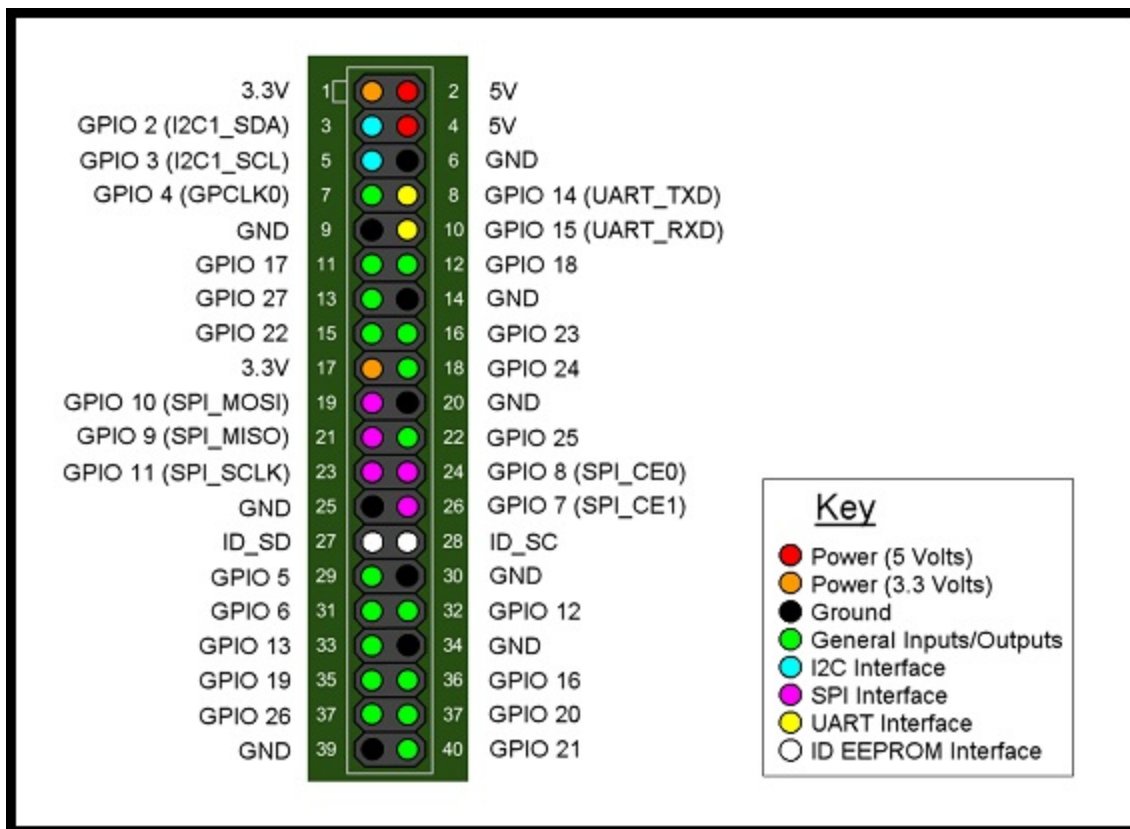
Audio breakout:

http://www.ebay.co.uk/itm/141445037537?_trksid=p2060353.m2749.l2649&ssPageName=STRK%3AMEBIDX%3AIT

Connect a Raspberry Pi camera to the Camera connector:



Open up a wireless doorbell and connect the speaker wires to gpio 7 (pin 26) and ground (14) (through a 10K ohm resistor) to the Raspberry Pi pins. Connect the battery pins of the doorbell circuit to the 3.3 V and ground. Test this by pushing the wireless doorbell switch and you should hear the “ding dong”



Connect a USB microphone to the Raspberry Pi.

Connect the LCD to the Raspberry Pi and make sure there are no short circuits. Insert the newly flashed SD card into the Raspberry Pi and turn it on.

Connect the Raspberry Pi to the local network either by wifi or ethernet.

Stream locally, run LCD3.py and press the left button and go to the RPi's ip address:8080. Press the right button to stop.

To stream to youtube, perform the following commands in terminal:

```
cd ./arm/bin
```

```
raspivid -o - -t 0 -w 1270 -h 720 -fps 25 -b 600000 -g 50 | ./ffmpeg -re -ar 44100 -ac 2 -acodec pcm_s16le -f s16le -ac 2 -i /dev/zero -f h264 -i - -vcodec copy -acodec aac -ab 128k -g 50 -strict experimental -f flv rtmp://a.rtmp.youtube.com/live2/DoorBellDing.aueb-jj84-1hcv-93k4
```

Then go to your doorbell's channel and view the stream.

```
Local streaming: raspivid -t 999999 -w 1280 -h 720 -fps 10 -b 2000000 -o - | gst-launch-1.0 -e -vvv fdsrc ! h264parse ! rtph264pay pt=96 config-interval=5 ! udpsink host=192.168.0.19 port=5000  
raspivid -t 0 -b 2000000 -n -o - | gst-launch-1.0 -e -vvvv fdsrc ! h264parse ! flvmux ! rtmpsink location=rtmp://localhost/rtmp/live
```

Use <https://ngrok.com/> to view the Raspberry Pi stream.

(<https://learn.adafruit.com/monitor-your-home-with-the-raspberry-pi-b-plus/access-your-pi-from-anywhere>)

Login details for the doorbell is below. Use it to add yourself as a contact for hangouts access and send a test message to yourself. Keep using the old style chat as it works:

doorbellding

doorbell1

Connect an external speaker:

I bought this:

http://www.ebay.co.uk/itm/PAM8403-5V-DC-Audio-Amplifier-2-Channel-3W-2-Volume-Control-USB-Power-New-Board-/371183702252?pt=LH_DefaultDomain_3&hash=item566c45d0ec

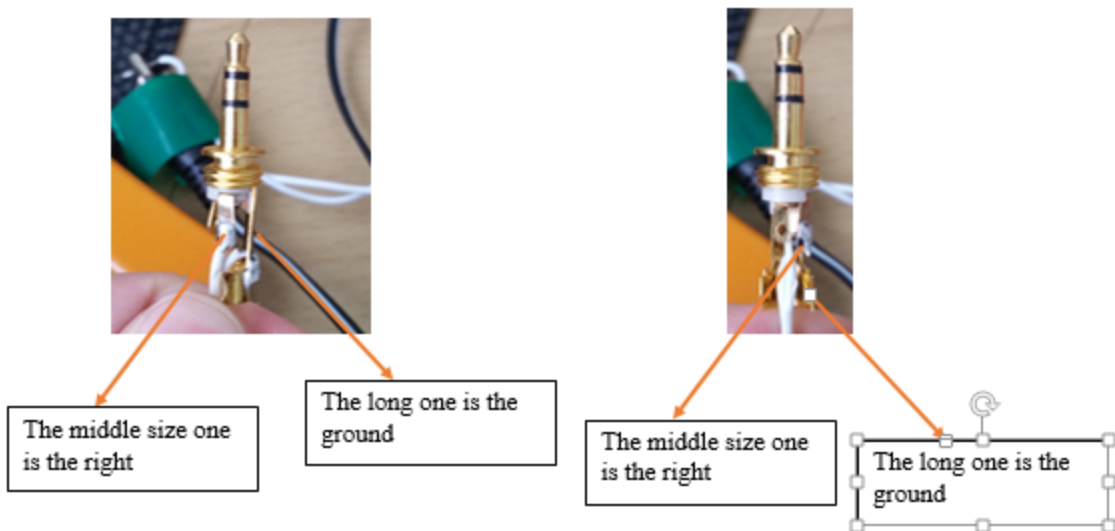
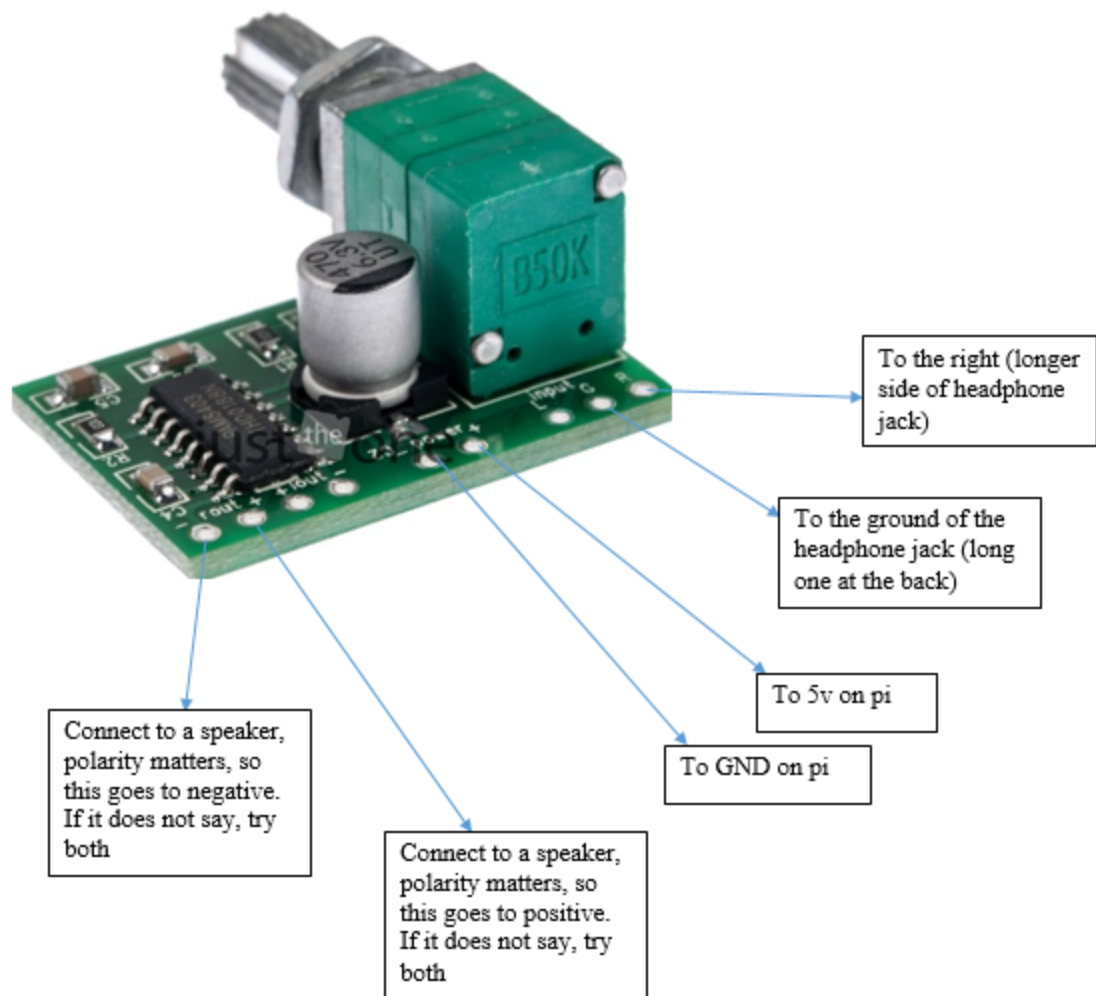
And a replacement headphone jack like this:

http://www.ebay.co.uk/itm/141445037537?_trksid=p2060778.m2749.l2648&ssPageName=STRK%3AMEBIDX%3AIT

Connect as shown in the image below and test it out with the following command

“omxplayer test.mp3” in terminal. Rather than connect to 5v, connect to the high pin of the LED (it stops the pi restarting when switched on)

Music should play from the speaker. Any speaker should work. I used USB power as the PI kept restarting when the switch was turned.



Changing settings

From any PC, go to the Raspberry Pi's ip address.

Go to `/my-pi-projects/Doorbell`

Intercom:

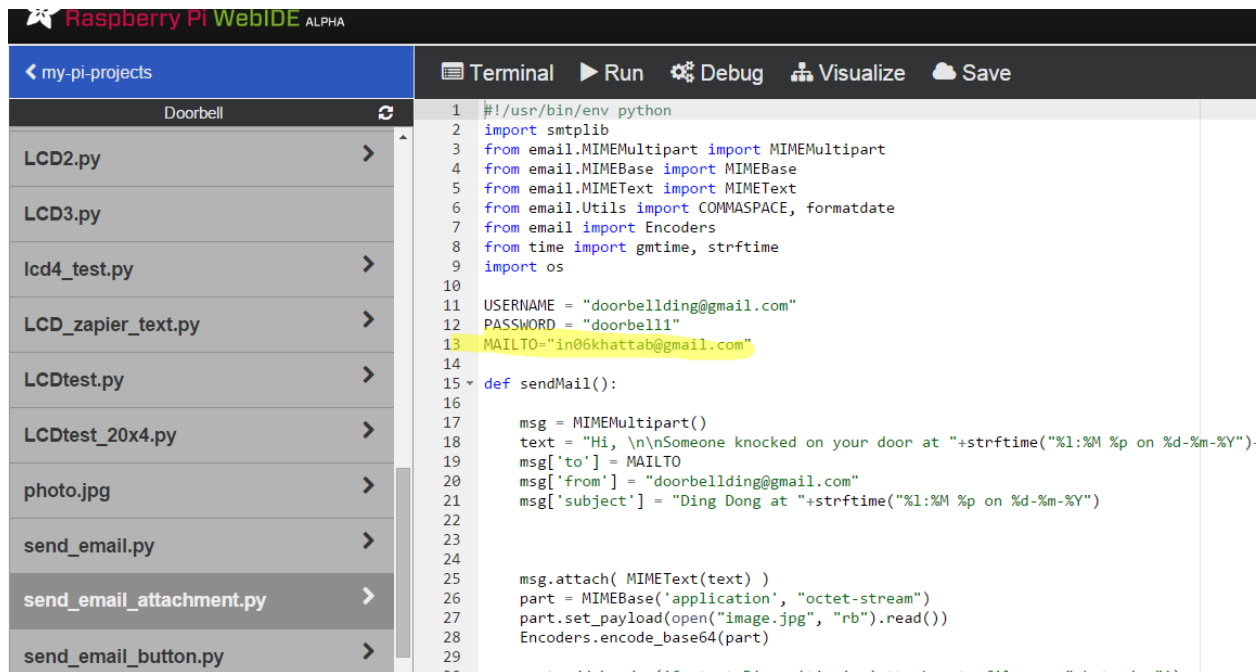
Create a free Sip account from Ekiga.net and change the email address in the call.sh file:

The screenshot displays the Adafruit WebIDE environment. On the left, a sidebar shows a project named "my-pi-projects". Inside this project, there's a folder named "Doorbell" which contains several files: ".call.sh.swm", ".call.sh.swo", ".call.sh.swp", "Adafruit_CharLCDPlate.py", "Adafruit_I2C.py", "Adafruit_MCP230xx.py", "button.py", "call.py", "call.sh", "camera.py", and "client_secret_935106472194-ntv...". Below the file list are three buttons: "+ Create New File", "+ Create New Folder", and "+ Upload File".

The main area on the right is a terminal window titled "Terminal" with a "Save" icon. It shows a shell prompt where the user has entered a complex command to run a script called "mkfifo" and execute it. The output shows the script running successfully, printing "sip:in06khattab@ekiga.net" and then exiting. Below the terminal output, there's a red error message indicating a registration failure: "sip, registration failed: 476 Unresolvable destination (476/TM)". This is followed by a traceback from the "lcd.py" file, specifically from the "buttonPressed" method, which calls "readU8" on the I2C bus.

Email:

Go to the `send_email_attachment.py` and change the MAILTO address to your email address.

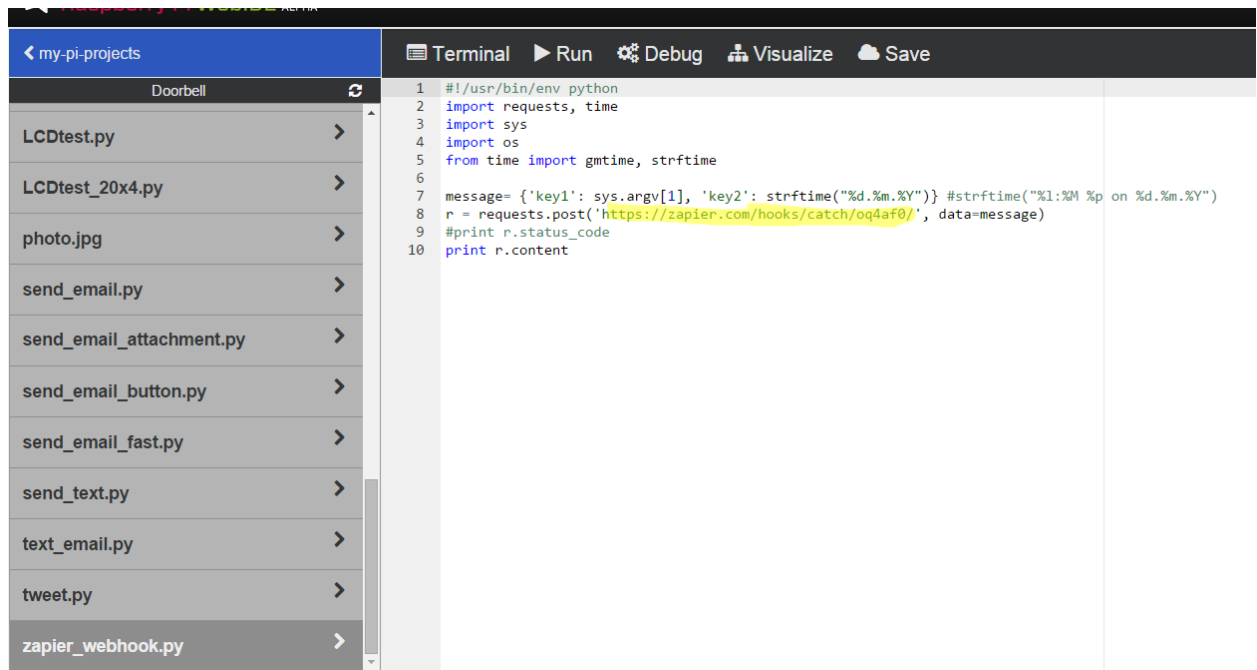


The screenshot shows the Raspberry Pi WebIDE interface. On the left, a file explorer lists files in the 'my-pi-projects' directory, including 'Doorbell', 'LCD2.py', 'LCD3.py', 'lcd4_test.py', 'LCD_zapier_text.py', 'LCDtest.py', 'LCDtest_20x4.py', 'photo.jpg', 'send_email.py', 'send_email_attachment.py', and 'send_email_button.py'. The 'Doorbell' file is selected. The main editor displays a Python script for sending an email via SMTP. The script includes imports for smtplib, email.MIMEMultipart, email.MIMEBase, email.MIMEText, email.Utils, and time. It defines a 'sendMail()' function that constructs a MIME message with a text body and an attachment (photo.jpg). The message is then sent to the email address specified in the MAILTO variable.

```
1 #!/usr/bin/env python
2 import smtplib
3 from email.MIMEMultipart import MIMEMultipart
4 from email.MIMEBase import MIMEBase
5 from email.MIMEText import MIMEText
6 from email.Utils import COMMASPACE, formatdate
7 from email import Encoders
8 from time import gmtime, strftime
9 import os
10
11 USERNAME = "doorbellding@gmail.com"
12 PASSWORD = "doorbell11"
13 MAILTO="in06khattab@gmail.com"
14
15 def sendMail():
16
17     msg = MIMEMultipart()
18     text = "Hi, \n\nSomeone knocked on your door at "+strftime("%l:%M %p on %d-%m-%Y")
19     msg['to'] = MAILTO
20     msg['from'] = "doorbellding@gmail.com"
21     msg['subject'] = "Ding Dong at "+strftime("%l:%M %p on %d-%m-%Y")
22
23
24
25     msg.attach( MIMEText(text) )
26     part = MIMEBase('application', "octet-stream")
27     part.set_payload(open("image.jpg", "rb").read())
28     Encoders.encode_base64(part)
29
30     part.add_header('Content-Disposition', 'attachment; filename="photo.jpg"')
```

Text message:

Create a free zapier account <https://zapier.com> and create a trigger with a webhook to a phone SMS. Insert the URL into step 2 of the "Select a Web Hook account". Paste this into the zapier_webhook.py file.



The screenshot shows the Raspberry Pi WebIDE interface. On the left, a file explorer lists files in the 'my-pi-projects' directory, including 'Doorbell', 'LCDtest.py', 'LCDtest_20x4.py', 'photo.jpg', 'send_email.py', 'send_email_attachment.py', 'send_email_button.py', 'send_email_fast.py', 'send_text.py', 'text_email.py', 'tweet.py', and 'zapier_webhook.py'. The 'zapier_webhook.py' file is selected. The main editor displays a Python script that uses the requests library to send a POST request to a Zapier webhook URL. The request body contains a JSON object with 'key1' (the first command-line argument) and 'key2' (the current date and time).

```
1 #!/usr/bin/env python
2 import requests, time
3 import sys
4 import os
5 from time import gmtime, strftime
6
7 message= {'key1': sys.argv[1], 'key2': strftime("%d.%m.%Y")} #strftime("%l:%M %p on %d.%m.%Y")
8 r = requests.post('https://zapier.com/hooks/catch/oq4af0/', data=message)
9 #print r.status_code
10 print r.content
```

Enter your phone number into step 3 of Zapier. Run this code and then match up the keys as follows to customise your text message. Key 1 is the time and Key 2 is the date.

5 Match up  Web Hook Hook to  Phone SMS

Message (required)

Hi,

Someone knocked on your door at **Key1** on **Key2**.

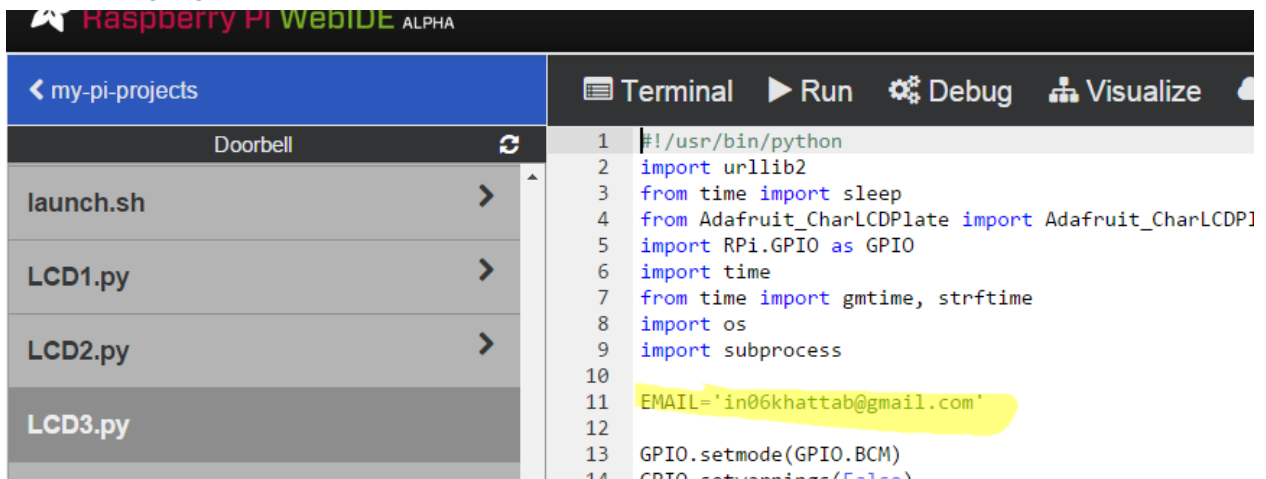
Have a great day!

Insert  fields

Finally, give it a name and turn it on.

Hangouts:

Change this email address in LCD3.py and send a sample hangouts message to doorbellding@gmail.com.



```

1  #!/usr/bin/python
2  import urllib2
3  from time import sleep
4  from Adafruit_CharLCDPlate import Adafruit_CharLCDPlate
5  import RPi.GPIO as GPIO
6  import time
7  from time import gmtime, strftime
8  import os
9  import subprocess
10
11  EMAIL='in06khattab@gmail.com'
12
13  GPIO.setmode(GPIO.BCM)
14  GPIO.setup(GPIO, GPIO.OUT)

```

Twitter:

Go to tweet.py and change this to your Twitter username.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import tweepy, time, sys
from tweepy import OAuthHandler
from time import gmtime, strftime

#enter the corresponding information from your Twitter application:
CONSUMER_KEY = 'dS3RgnaO2UMbYSXFd3aGWdlog'#keep the quotes, replace this with
CONSUMER_SECRET = 'zWmGQLiV5ZYPn3cKgFx1etmVx5tZxNYYb9mbO8ZA85p2OwaPDA'#keep th
ACCESS_KEY = '3001952026-sm160Kzd2CAshTGJlHQ6XjayzvDScKSjgMsZ8uU'#keep the quc
ACCESS_SECRET = 'pcp4WSS9Xbgl2BeAzqMsmiOsh45SSMeQbrVUHw3iXIS1d'#keep the quote
auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)

try:
    api = tweepy.API(auth)
    line="@in06khattab Ding Dong at "+strftime("%l:%M %p on %d-%m-%Y")

    api.update_status(status=line)
except tweepy.TweepError as e:
    print e.message[0]['code'] # prints 34
    print e.args[0][0]['code'] # prints 34
```

```
cd /home/webide/repositories/my-pi-projects/Doorbell/
sudo python LCD3.py
```

Password for hangouts is here: open sd card image and search file contents with
<https://www.paragon-software.com/home/linuxfs-windows/>
 Replace doorbell1 with doorbell2. Probably in /etc/sendhangouts or sendxmpprc

```
End twinkle
sudo pkill -9 twinkle
```