

# PiDoorbell : Home Automation with RaspberryPi and Arduino

Rupa Dachere

[rupa@codechix.org](mailto:rupa@codechix.org)



# What you will hear and see

- Problem
- Phases of my project
- Solution & Reasons
- Compare RaspberryPi, Arduino, BBB
- All the hardware I am using
- OSS and OSH that I am using
- LIVE DEMO !!
- Code + (video – maybe)
- Issues and Future Work
- Q&A

# Problem

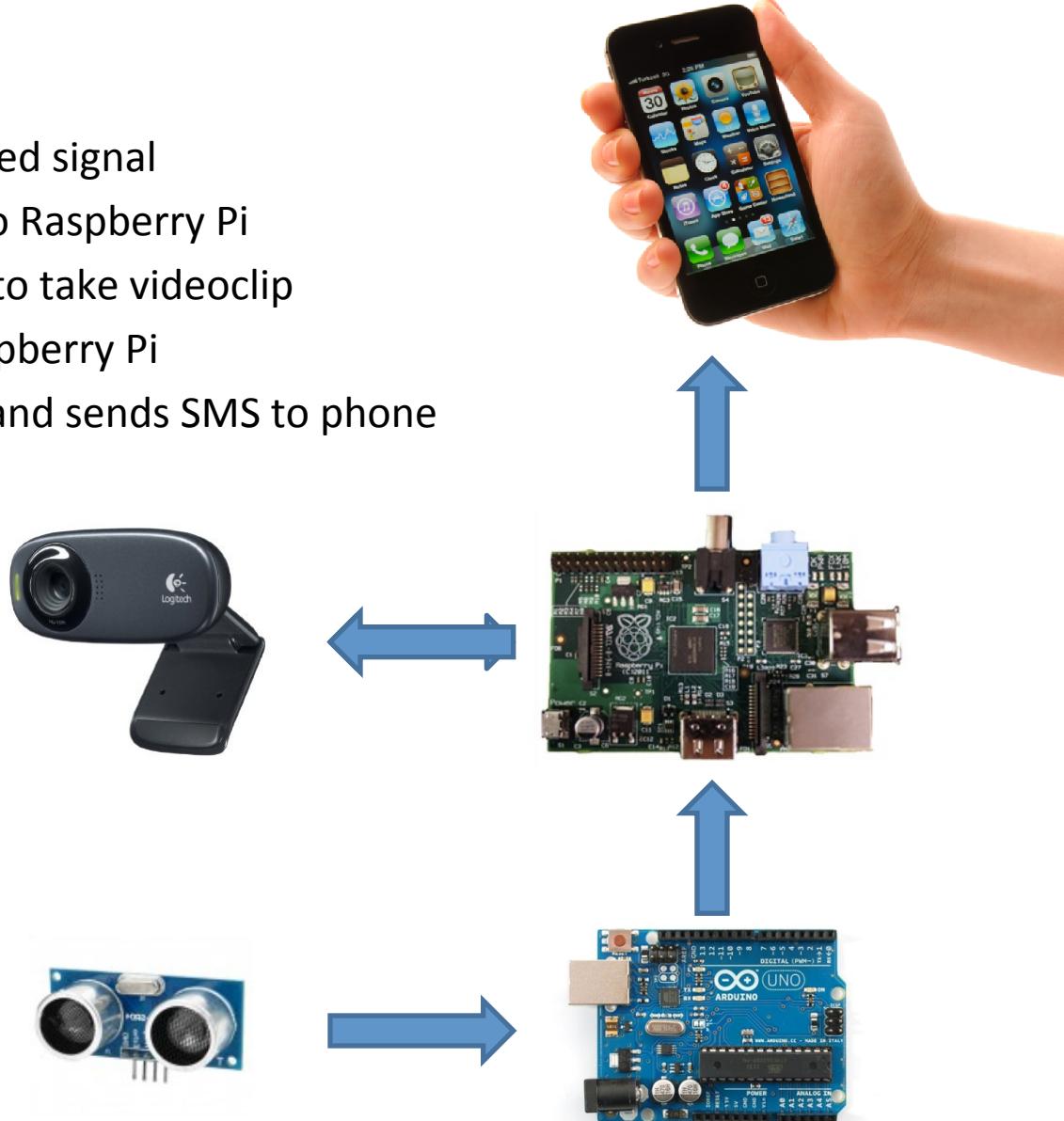
1. Wanted to know who was at my front door
2. What day and time they came
3. Wanted to be notified when they arrived

Ex. Contractors, Package Delivery, Neighbor etc.

# Phases of my project

- Phase 1:
  - Snapshot of visitor
- Phase 2:
  - Videoclip of visitor
  - Phase 2.1: Audioclip of visitor
- Phase 3:
  - Live videotostream of visitor
  - One way videotostream
  - Audio notification to visitor when they walk up
- Phase 4:
  - Two-way audio
- Phase 5:
  - Two way audio-video – maybe

- 1) Here comes a caller
- 2) Proximity sensor outputs changed signal
- 3) Arduino sends changed signal to Raspberry Pi
- 4) Raspberry Pi instructs webcam to take videoclip
- 5) Webcam sends videoclip to Raspberry Pi
- 6) Raspberry Pi uploads videoclip and sends SMS to phone



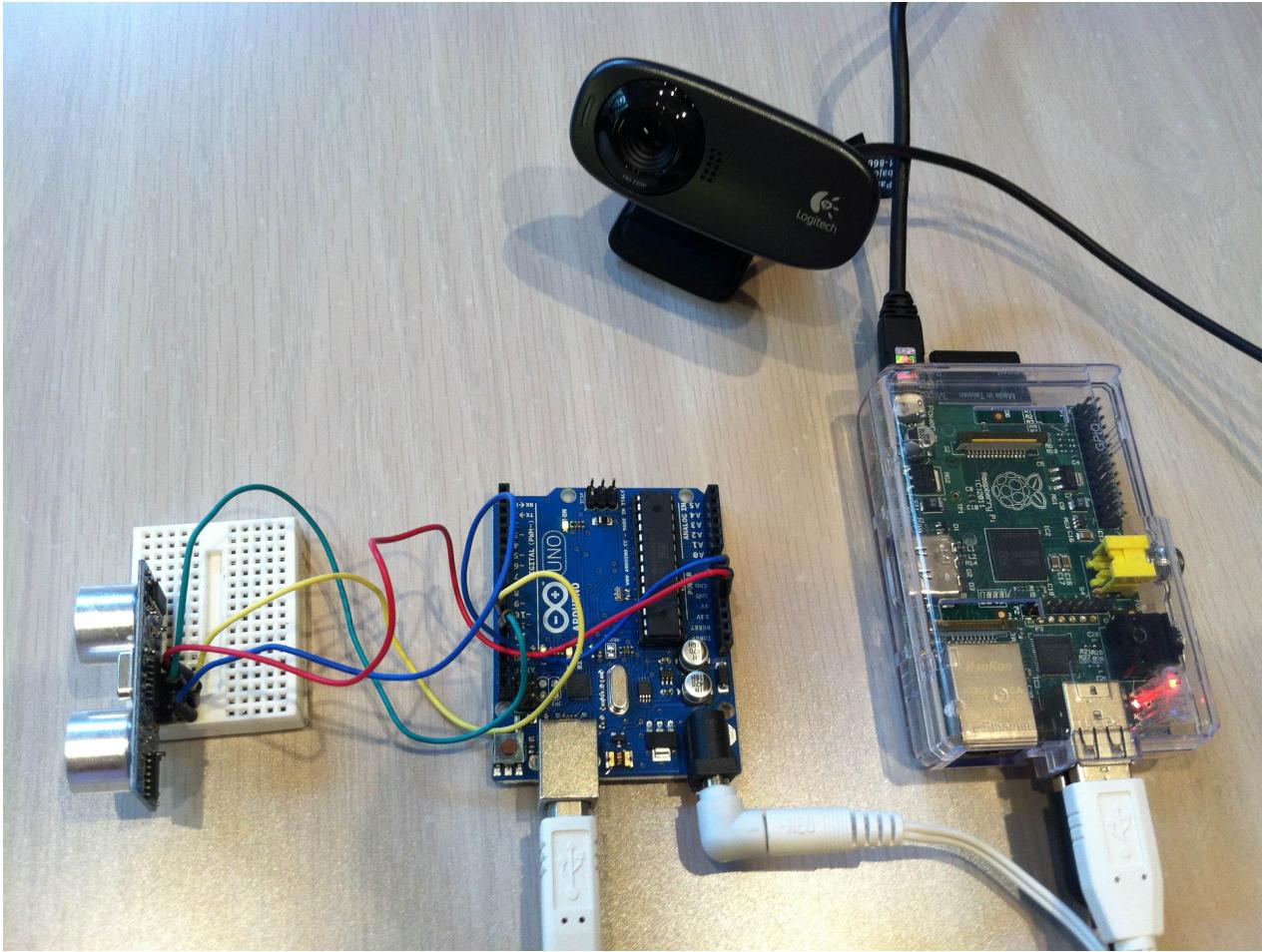
# What you will see today

- Phase 2
  - Videoclip of visitor (in this case, volunteer from the audience)
  - Audioclip of visitor
  - 10 sec capture
  - Twitter for notification (added for Australia)
  - Twilio for notification (USA)

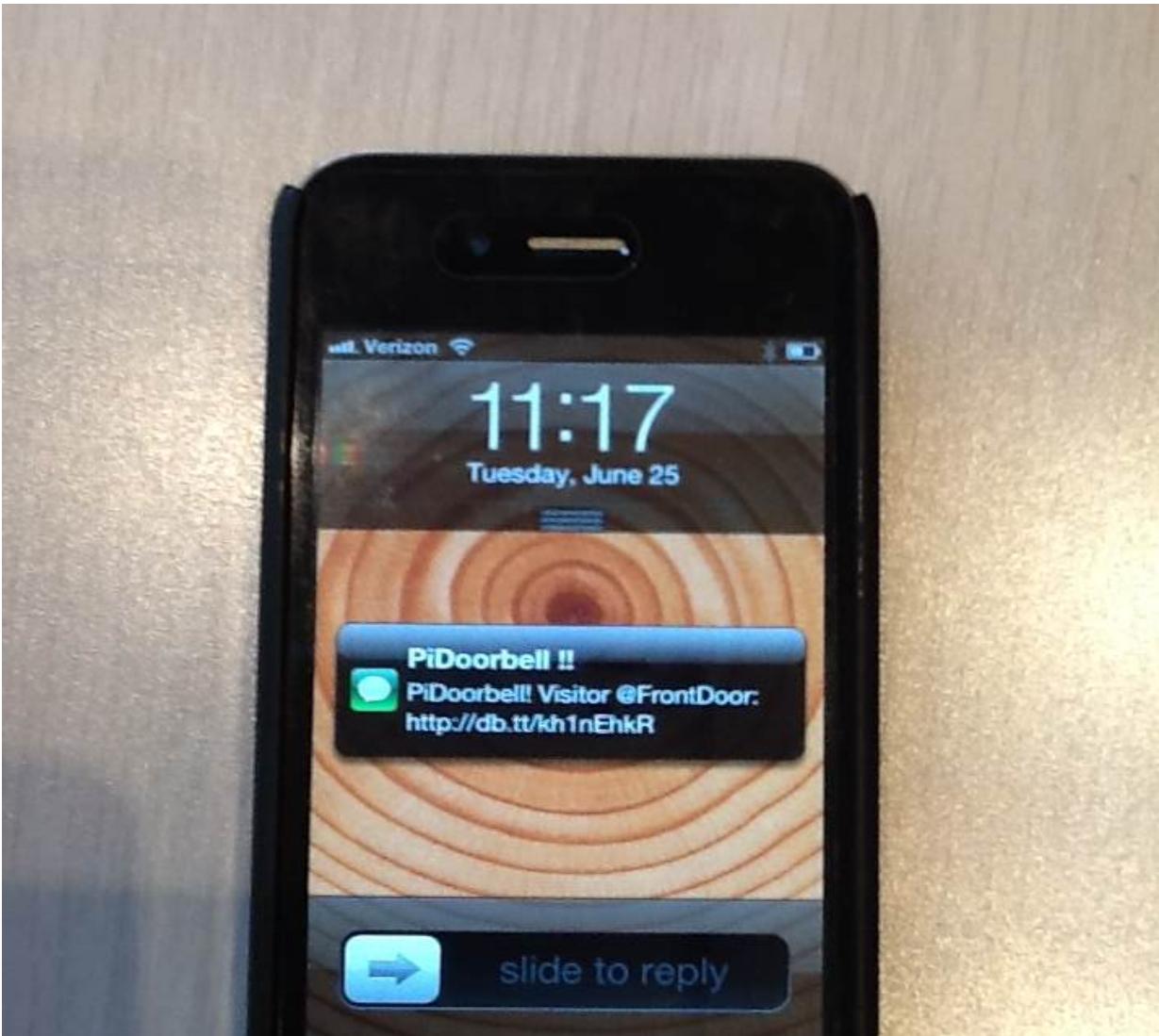
# Why this solution????

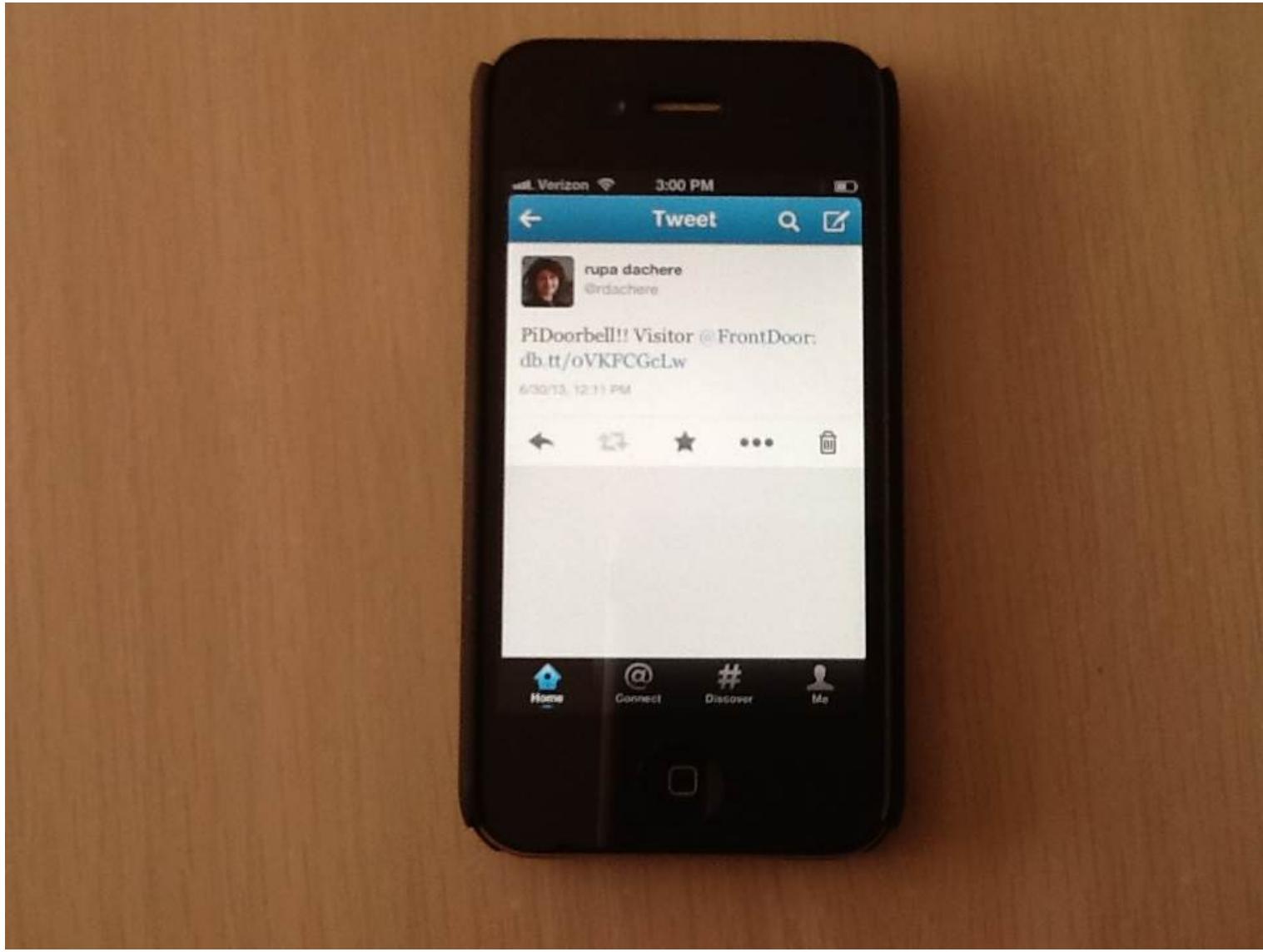
- Reasonably inexpensive
  - Around \$100
  - Nothing commercial that does this
- Learn about hardware and build my own solution
- Get to speak at conferences & Inspire others

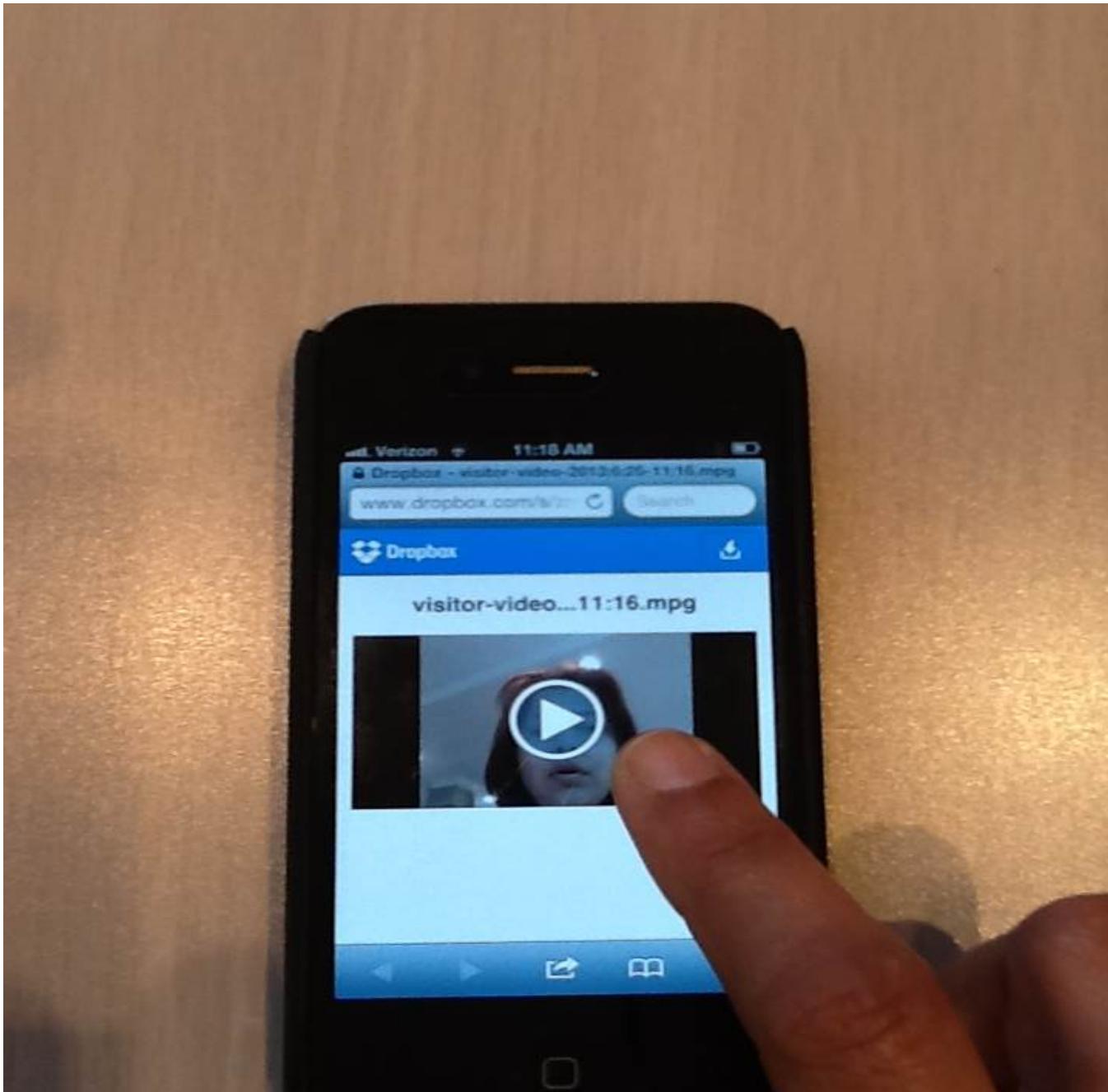
# All the hardware I used



# Notification on Mobile Device







# Arduino vs Rpi vs BeagleBone Black

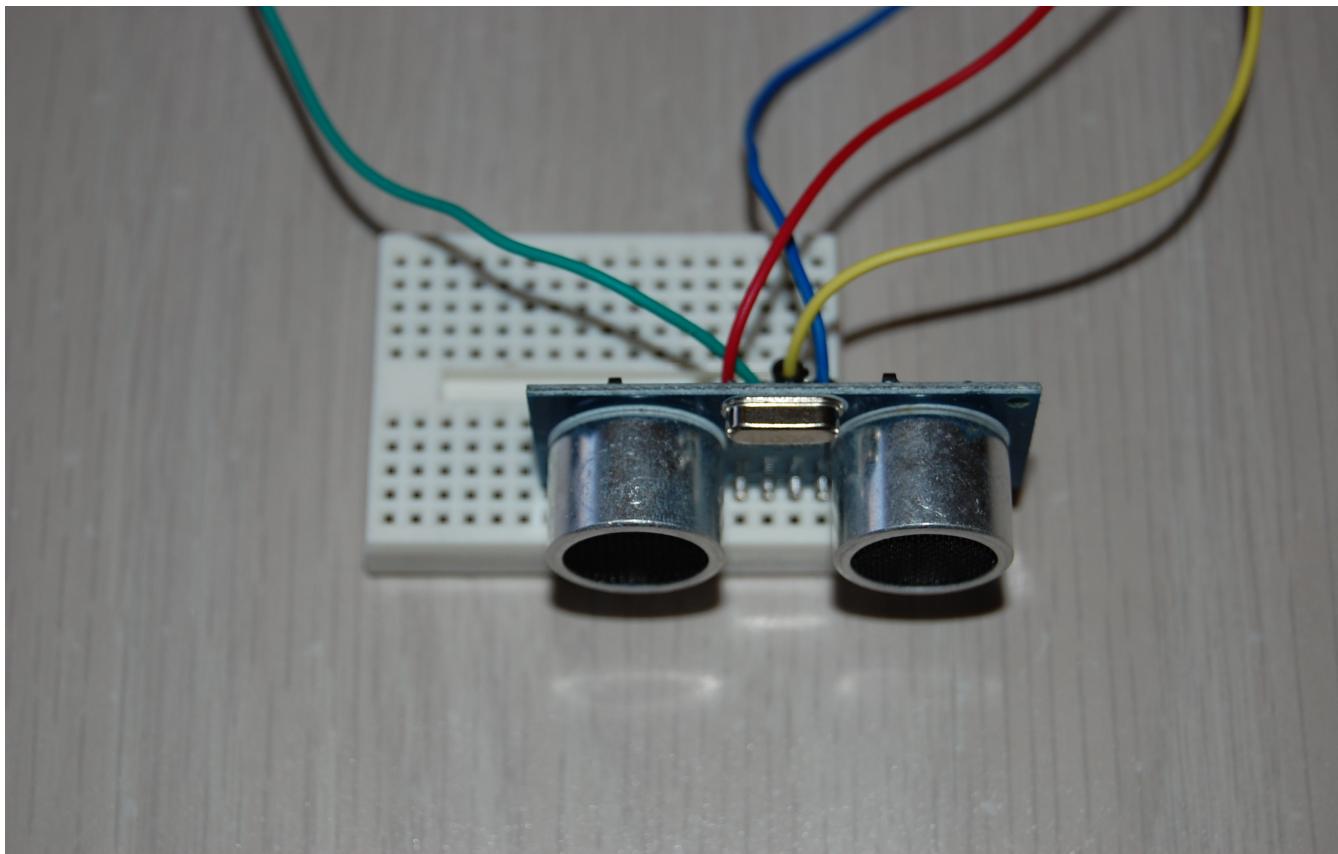
Component	Arduino Uno	Raspberry Pi	Beaglebone Black
<b>Model</b>	<b>R3</b>	<b>Model B</b>	<b>Rev A5A</b>
Processor	ATMega 328	ARM11	ARM Cortex-A8
Clock Speed	16MHz	700MHz	700MHz
<b>RAM</b>	<b>2KB</b>	<b>512MB</b>	<b>512MB</b>
<b>Flash</b>	<b>32KB</b>	<b>External SD Card</b>	<b>2GB Onboard/ optional external</b>
EEPROM	1KB		
Input Voltage	7-12V	5V	5V
Min Power	42mA	700mA	170mA
<b>Digital GPIO</b>	<b>14</b>	<b>8</b>	<b>66</b>
Analog Input	6 10-bit	N/A	7 12-bit
PWM	6	1	8
TWI/I2C	2	1	2
SPI	1	1	1
UART	1	1	5
<b>USB Master</b>	<b>N/A</b>	<b>2</b>	<b>1</b>
Ethernet	N/A	10/100	10/100
<b>Video Out</b>	<b>N/A</b>	<b>HDMI, Composite</b>	<b>microHDMI</b>
<b>Audio Out</b>	<b>N/A</b>	<b>HDMI, Analog</b>	<b>Analog</b>

# Software I wrote

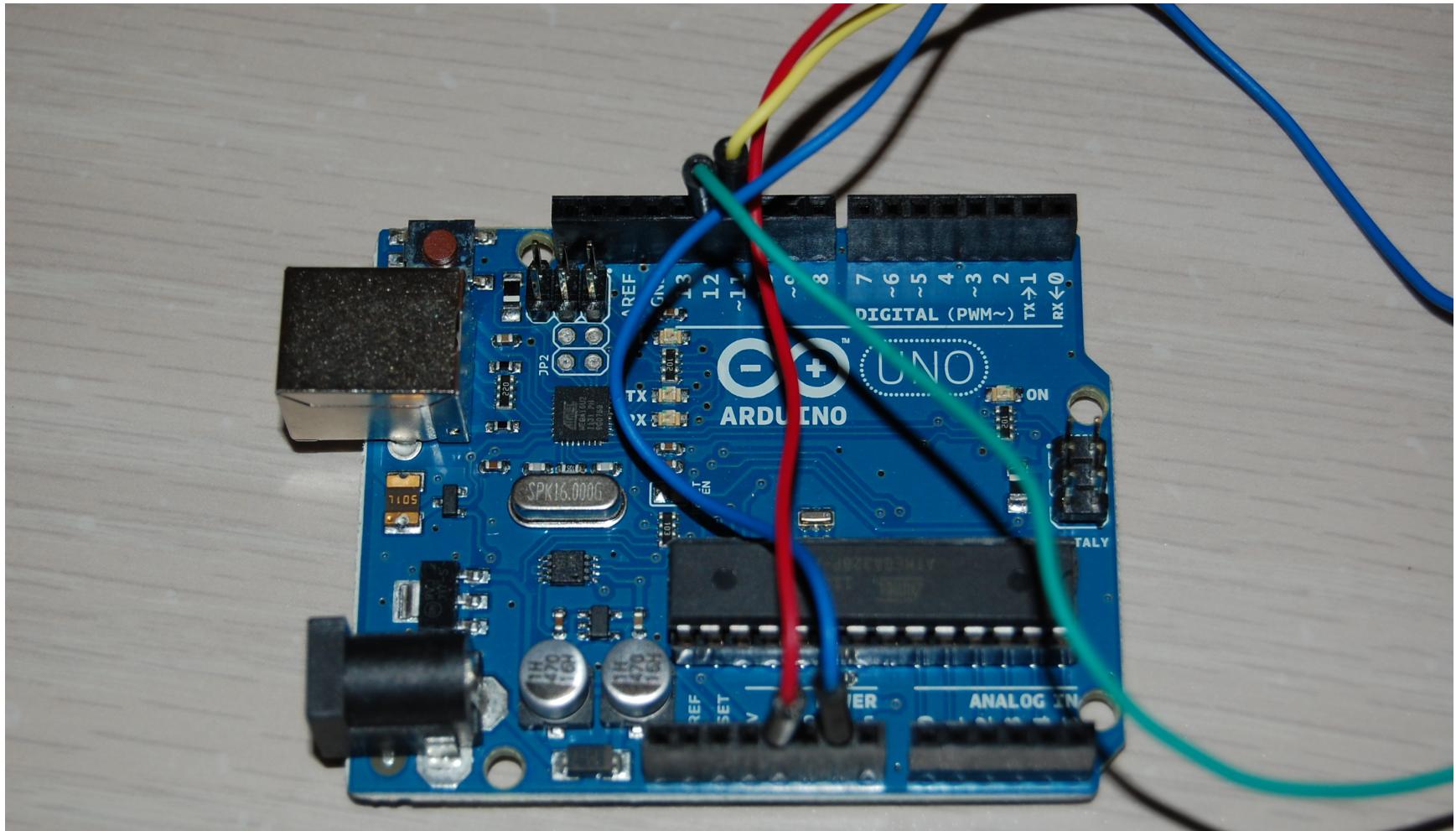
- Arduino Sketch to read output from sensor
- RaspberryPi python code – 2 scripts
  - Pidoorbell-recognizer.py
    - Read sensor output from Arduino (one thread)
    - Detect Object (5 valid reads from sensor)
    - Take videoclip by sending message to webcam (ffmpeg)
    - Upload the video to a site (Dropbox)
    - Call send\_notifications.py with URL to videoclip
  - Send\_notifications.py
    - Use Twilio/Twitter to send URL to videoclip as an SMS/Text msg
    - Standalone CLI – configurable delivery (Twitter, Twilio, etc.)

# Proximity Sensor

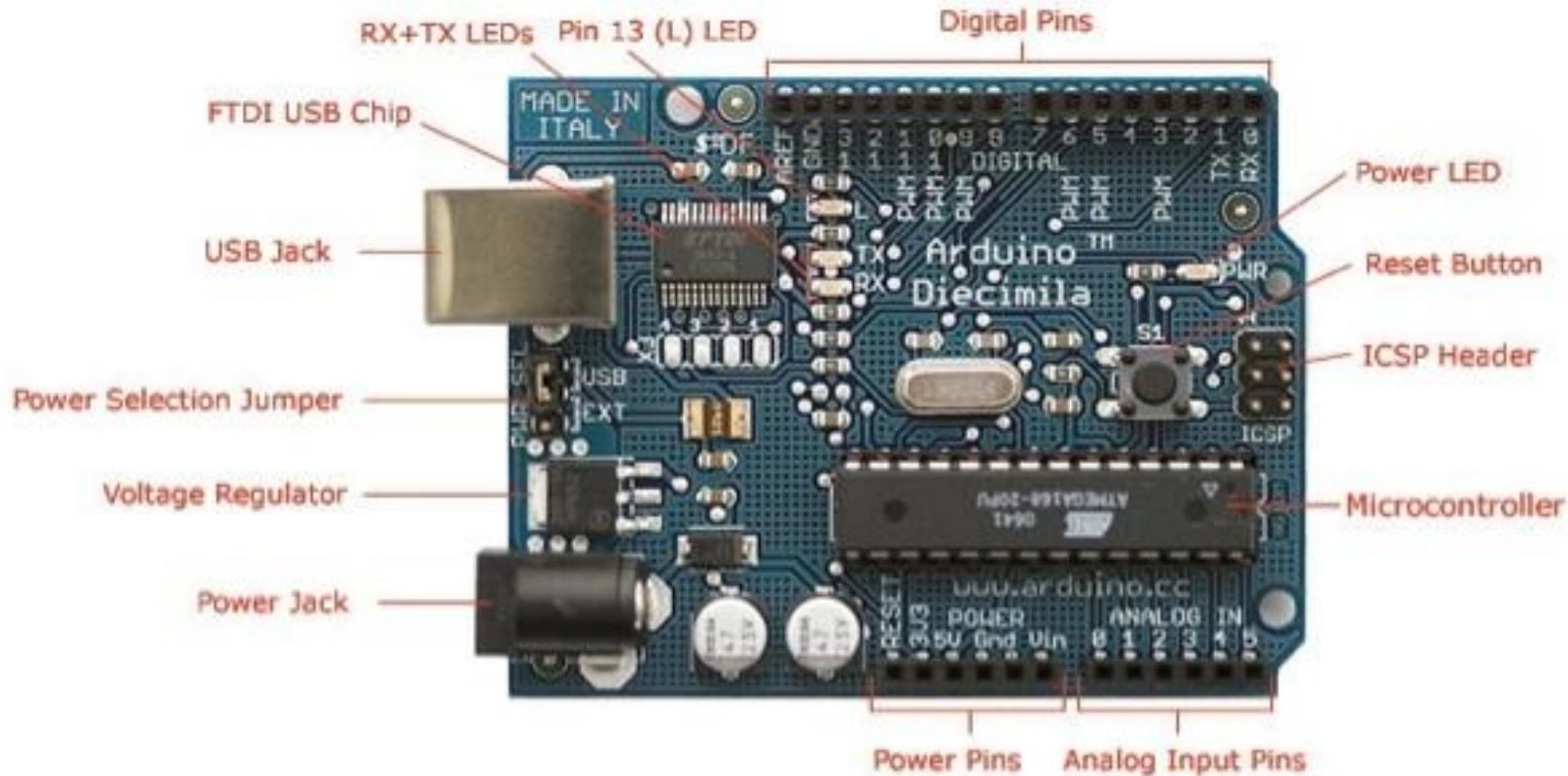
- HC-SR04



# Arduino – Uno R3

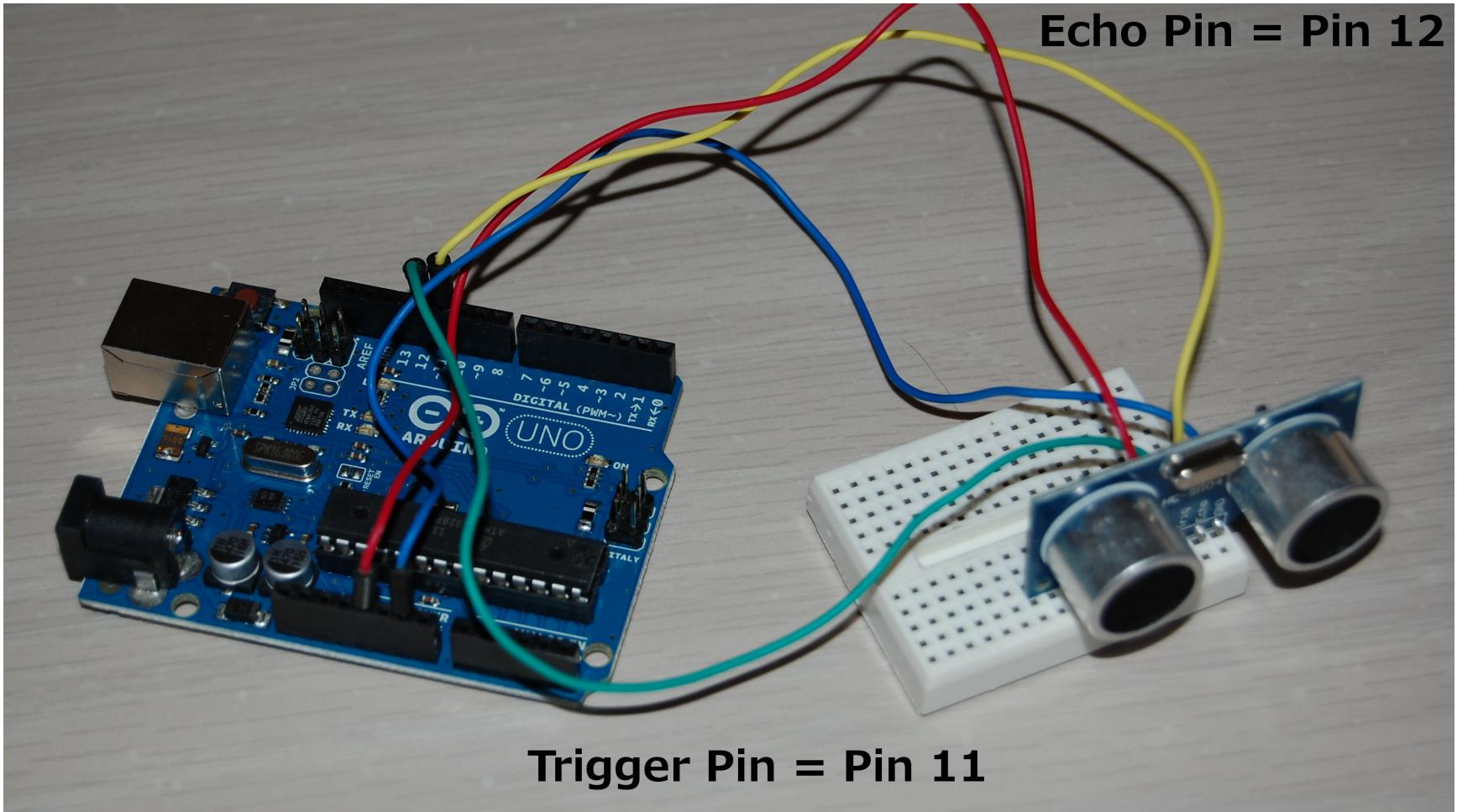


# Arduino Uno R3



Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

# Arduino + Sensor Setup

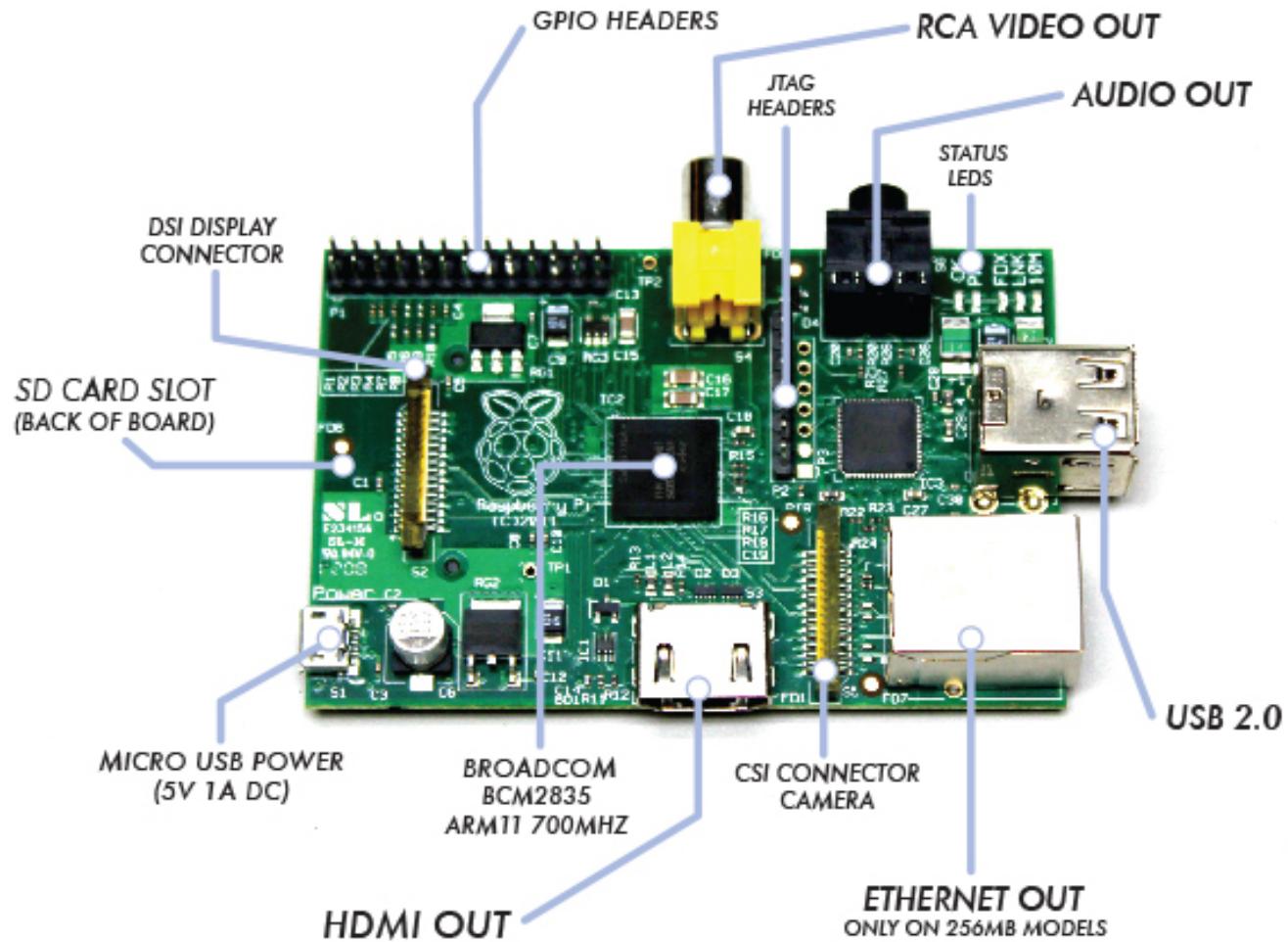


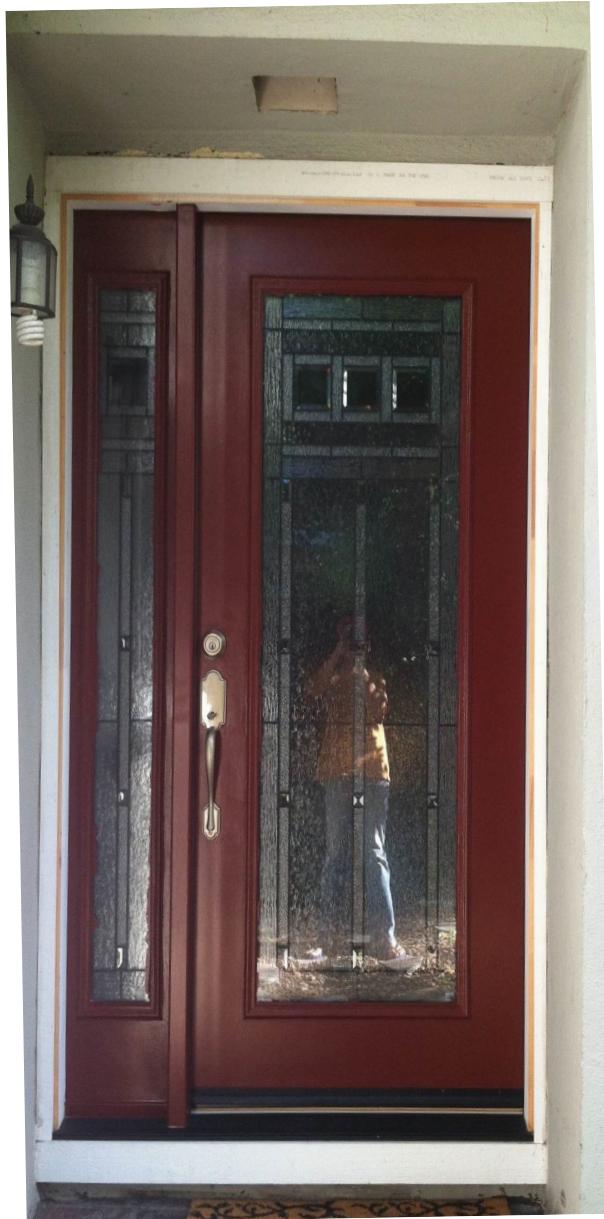
# Arduino Dev Setup

- Download dev kit – arduino.cc
- Connect via USB to Mac/PC/\*nix
- Look at example sketches – blinking lights etc.
- That's it – really \*simple\*

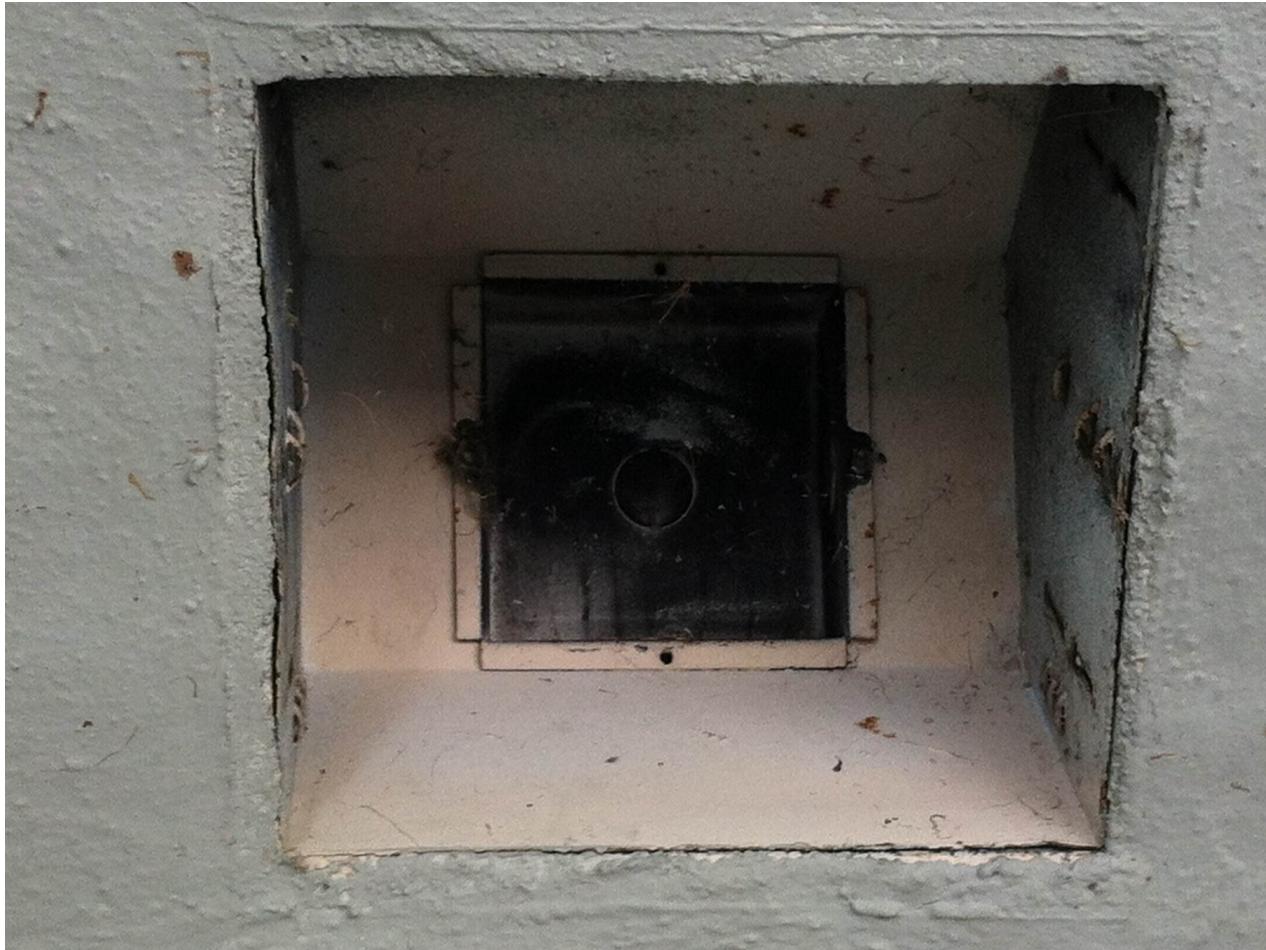
# RaspberryPi







# Mounting area – 8"x 8"x 6"



# Raspberry Pi Development Setup

- 2012-12-16-wheezy-raspbian.img
- Install pip, pyserial – apt-get pip install
- Ffmpeg – might have to compile
- Libraries for Dropbox – to do OAuth2, upload and sync files
- Libraries for Twilio Api
- Output from Arduino on /dev/ttyACM0

# Demo

- Walk in front of the proximity sensor
- See detection of foreign object
- Trigger webcam to start capturing video
- Save in file with date/timestamp
- Upload file to Dropbox and get URL to file
- Send SMS/Tweet to smartphone with URL of video on Dropbox

# Arduino Sketch Code

```
#include <NewPing.h>

#define TRIGGER_PIN 12 // Arduino pin tied to trigger pin on the ultrasonic sensor.
#define ECHO_PIN    11 // Arduino pin tied to echo pin on the ultrasonic sensor.

NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
void loop() {
    delay(1000); // Wait 1 sec between pings
    unsigned int dist_inches = sonar.ping_in();
    Serial.println(dist_inches);
}
```

# RaspberryPi Code

---

(1/7)

## Pidoorbell-recognizer.py:

```
class Arduino(threading.Thread) :  
    def run(self, interactive) :  
        # Port may vary, so look for it:  
        baseport = "/dev/ttyACM0"  
        self.ser = serial.Serial(baseport, 115200,  
        timeout=800)
```

# RaspberryPi Code

(2/7)

```
# threshold to mitigate butterflies/moths
valid_pic_count = 0

while True :
    data = self.ser.readline().strip()
    if data :
        if interactive :      #good for testing
            if int(data) < MAX_TRIGGER_DISTANCE:
                #set to a default of 30 inches
                print "Detected something -", data
```

# RaspberryPi Code

(3/7)

```
valid_pic_count += 1
```

```
if valid_pic_count == VALID_PIC_THRESHOLD:
```

```
    # get current date and timestamp  
    now = datetime.datetime.now()
```

```
#build video filename with date and timestamp  
videoclip_filename = "visitor-%d:%d:%d-%d:%d:  
                      %d.jpg" % (now.year, now.month,  
                      now.day, now.hour,  
                      now.minute, now.second)
```

# RaspberryPi Code

(4/7)

```
take_videoclip_cmd = " ffmpeg -f video4linux2 -s  
320x240 -i /dev/video0 -f alsa -ar 22050 -ac 1 -i hw:1,0  
-ab 48k -timelimit 10 <filename>"
```

```
#reset valid_pic_count  
valid_pic_count = 0
```

# RaspberryPi Code

(5/7)

```
print "UPLOADING TO DROPBOX:"
```

Uses Dropbox API:

```
sync_dropbox_cmd = "./dropbox-uploader.sh  
upload ./dropbox-pidoorbell/" + videoclip_filename  
process = subprocess.Popen(sync_dropbox_cmd, shell=True)
```

Call `send_notifications.py` with Dropbox URL

# RaspberryPi Code

(6/7)

## Get URL for Dropbox file:

1. Get Session Tokens via OAuth2
2. Use session tokens to instantiate a DropboxClient using the DropboxClient Python SDK
3. Retrieve file Metadata using the ***DropboxClient.share(path)*** method
4. Extract file URL from the metadata to pass as argument to *send\_notifications.py*

# RaspberryPi Code (7/7)

## Send\_notifications.py: Uses Twilio or Twitter API

```
parser.add_option('-u', '--sms_url', action="store", ...  
parser.add_option('-m', '--mode', action="store", dest="mode", ...
```

```
# mode == "sms" or "tweet" or "all"  
if options.mode == "sms":  
    send_sms(options.sms_url)  
elif options.mode == "tweet":  
    send_tweet(options.sms_url)  
else:  
    send_sms(options.sms_url)  
    send_tweet(options.sms_url)
```

# RaspberryPi Code (8/9)

```
def send_sms(sms_url):
```

```
    client = TwilioRestClient(sms_auth_info.account_sid,  
sms_auth_info.auth_token)
```

```
    print '\n\n ***** SENDING SMS WITH URL: ', sms_url , "*****\n\n"
```

```
    body_url = "PiDoorbell! Visitor @FrontDoor: " + sms_url
```

```
    message = client.sms.messages.create(to="+XXXYYYZZZ",  
from_="+XXXYYYZZZ", body=body_url)
```

# RaspberryPi Code (9/9)

```
def send_tweet(tweet_url):
```

```
    api = twitter.Api(consumer_key=sms_auth_info.twitter_auth_key,  
consumer_secret=sms_auth_info.twitter_auth_secret,\  
        access_token_key=sms_auth_info.twitter_access_key,  
access_token_secret=sms_auth_info.twitter_access_secret)
```

```
    api.VerifyCredentials()
```

```
    print '\n\n ***** SENDING TWEET WITH URL: ', tweet_url , "*****\n\n"
```

```
    post_message = "PiDoorbell!! Visitor @FrontDoor: " + tweet_url  
    status = api.PostUpdate(post_message)
```

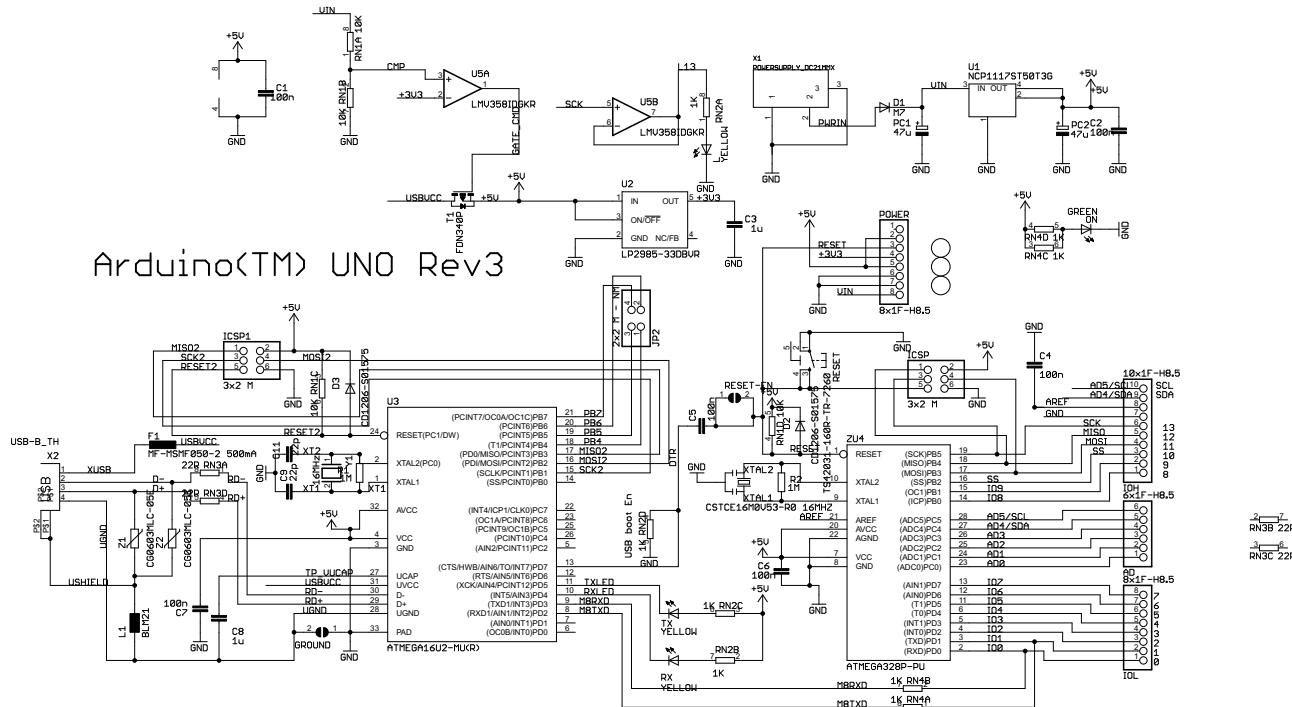
# Open Source Software

- Almost everything
  - Python
  - Ffmpeg
  - Python-twitter lib
  - Git
  - All of my code – [github.com/rdachere](https://github.com/rdachere)
- Closed source
  - Twitter API's
  - Dropbox API's

# Open Hardware

- Arduino Uno R3
- RaspberryPi Model B
- Closed Hardware
  - Proximity Sensor
  - WebCam
  - Mac, iPhone

# Arduino Uno R3 Schematic

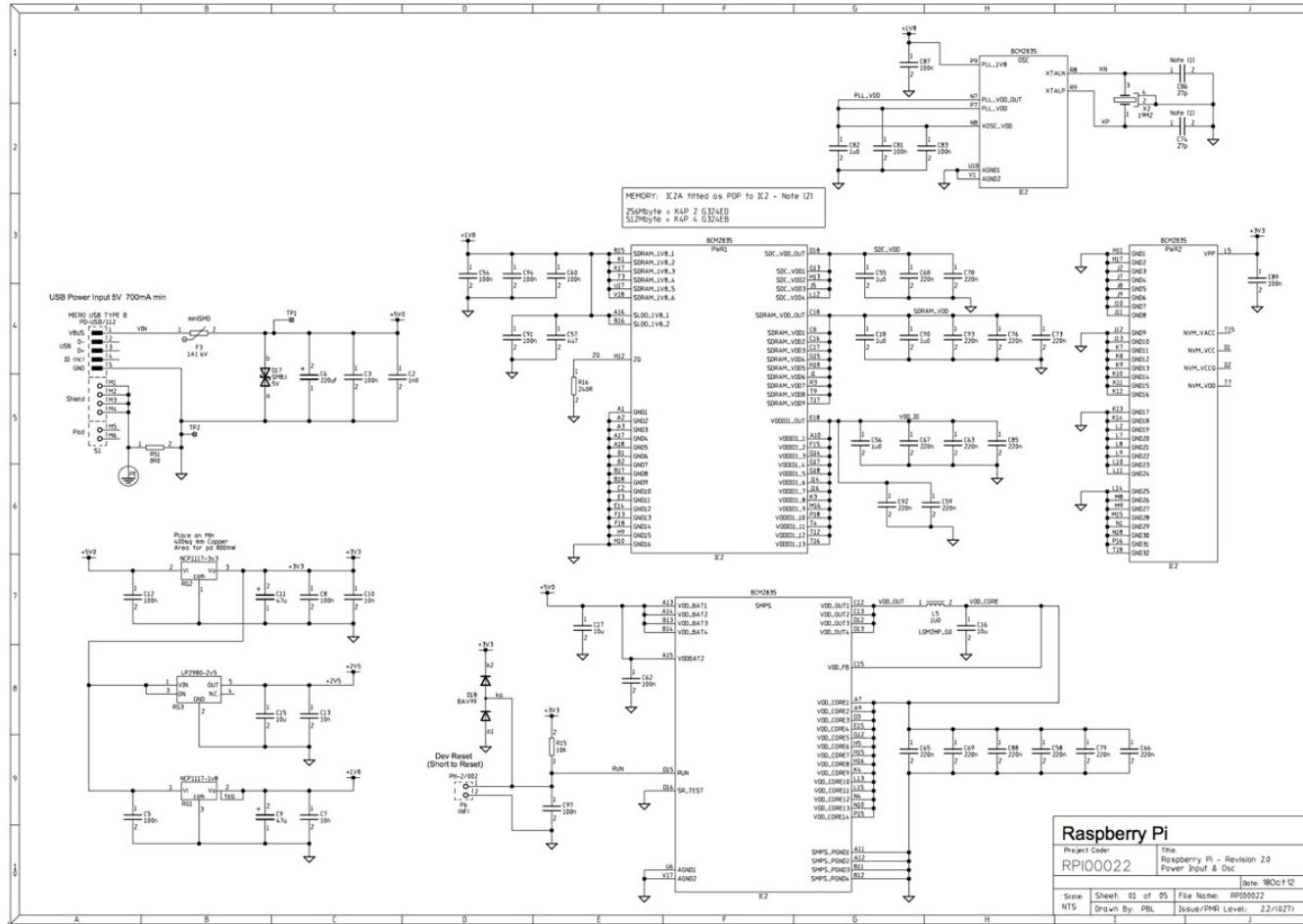


Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

ARDUINO is a registered trademark.

Use of the ARDUINO name must be compliant with <http://www.arduino.cc/en/Main/Policy>

# Raspberry Pi Model B Schematic



# Hardware & Cost

- Arduino Uno R3 = \$22
- Proximity Sensor = \$16 (now \$6?)
- Raspberry Pi = \$35
- Webcam = \$0 (free from my junk pile)
- Total = \$73 (possibly \$63?)
- Add extra for power adapters, breadboard, jumper wires, etc. <= \$100 (approx.)

# Attributions

- [www.adafruit.com](http://www.adafruit.com) / [newark.com](http://newark.com) (element14) / [sparkfun.com](http://sparkfun.com)
- [ffmpeg.org](http://ffmpeg.org)
- [dropbox.com/developers](http://dropbox.com/developers) / [twitter.com/developers](http://twitter.com/developers)
- [www.arduino.cc](http://www.arduino.cc) / [www.raspberrypi.org](http://www.raspberrypi.org)
- [code.google.com/p/python-twitter/](http://code.google.com/p/python-twitter/)
- [mcmelectronics.com](http://mcmelectronics.com) blog for information for the comparison chart
- I didn't use: Various books on Arduino and RaspberryPi – Stackoverflow is great!

# Issues and future work

- Video/Audio sync issues when recording
  - Is it the webcam, is it ffmpeg, is it Dropbox? Is it a combination of all three?
- International conferences
  - Twilio is unsupported. Fallback is Twitter – so far so good.
- Use Beaglebone Black instead
  - But need two USB ports when mounting outside front door. Could use the GPIO pins as an alternative.
  - BBB community is not as large yet
- Network latency is big issue – no way around it.
  - Have to keep tweaking delay parameters. Ideally, want heuristic driven system?
- New Rpi camera which can hook up to the ribbon connector – ordered + waiting now.

# Contact & Code/Slides

**Rupa Dachere** – [rupa@codechix.org](mailto:rupa@codechix.org)

**Twitter:** @rdachere

**Blog:** rupadachere.blogspot.com

**Site:** rupadachere.com

**Source Code & Slides (Soon):**  
Github.com/rdachere/pidoorbell

# Tips and Lessons Learned

- Lipsync and audio issues with video - ffmpeg
- RaspberryPi could not detect Arduino or Webcam – USB Hub separate PS
- Standard library for sensor didn't work
- Pins and Baudrate were different
- [http://code.google.com/p/arduino-new-ping/  
wiki/Simple NewPing Example](http://code.google.com/p/arduino-new-ping/wiki/Simple_NewPing_Example)
- Modified detection distance to inches
- Modified frequency to every second