



NYU

POLYTECHNIC SCHOOL
OF ENGINEERING

Internet of Things

Dhaval Palsana and Paul Phamduy
Advanced Mechatronics Raspberry Pi Project
May 8, 2014

Internet of Things

- Terminology is used to denote the advance connectivity of devices, systems, and services to the internet.
- They are becoming prevalent in every day objects (google glass, nest, etc.) as the cost of computing is decreasing.
- Simplifies daily tasks for the average person.





Coffee Machines

- Cooking appliances used to brew coffee has become more advanced with time. Drip brewing has become the predominate method of brewing coffee.
- 54% of Americans over the age of 18 drink coffee everyday.
- Coffee machines exist but people still wait in long lines to get coffee in the morning.





Brew Coffee over the Internet

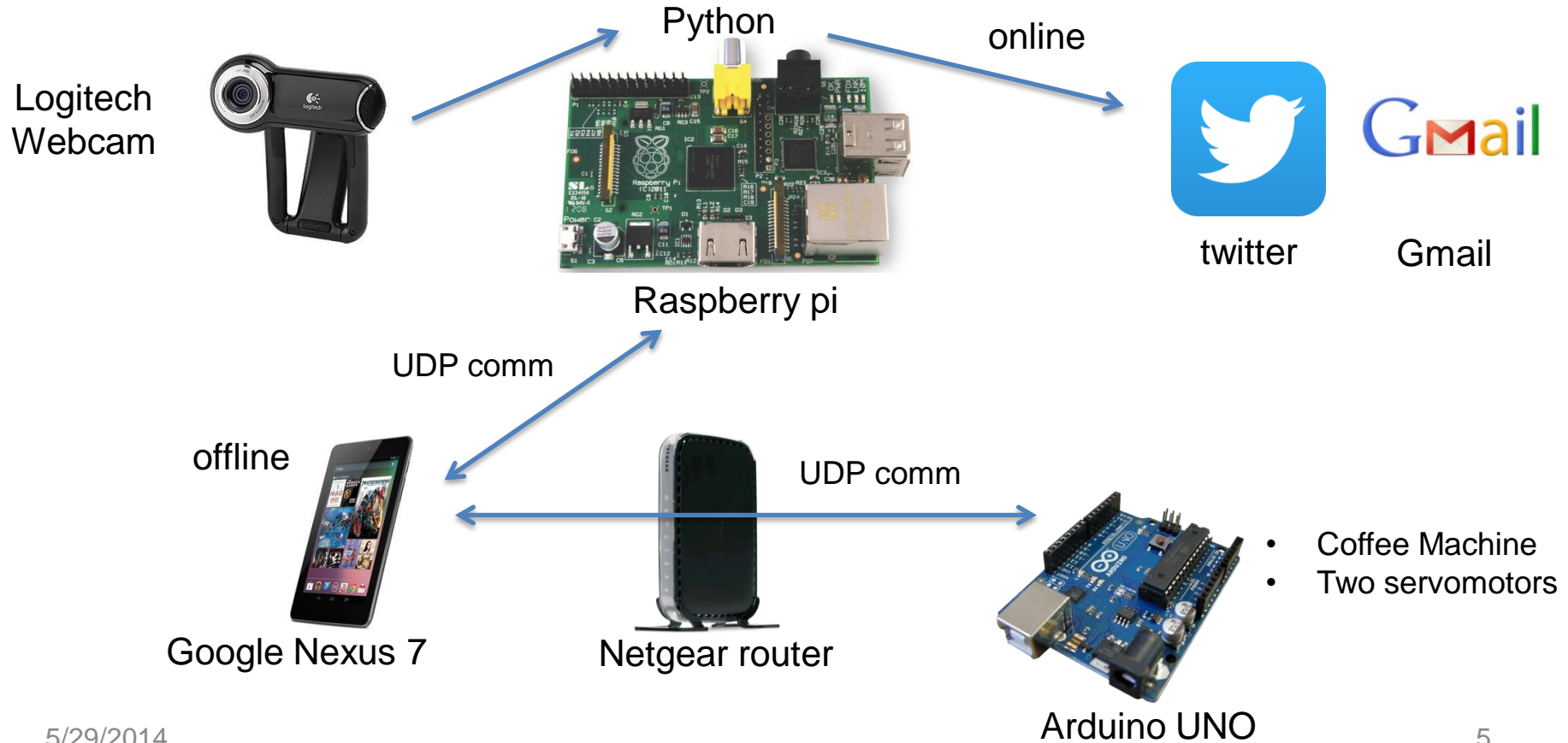
- Coffee is great any time of day, from the early mornings for people going to work to the late nights for students working on a mini-project. Very stimulating!
- If you can order food, call a taxi, and buy movie tickets over the internet, why not add brewing your own coffee to the list?
- Apply concepts in the classroom to practical knowledge and the development of an wireless controlled coffee machine. (Raspberry Pi, Android, Arduino and OpenCV)



NYU

POLYTECHNIC SCHOOL
OF ENGINEERING

Communication Protocol





Hardware

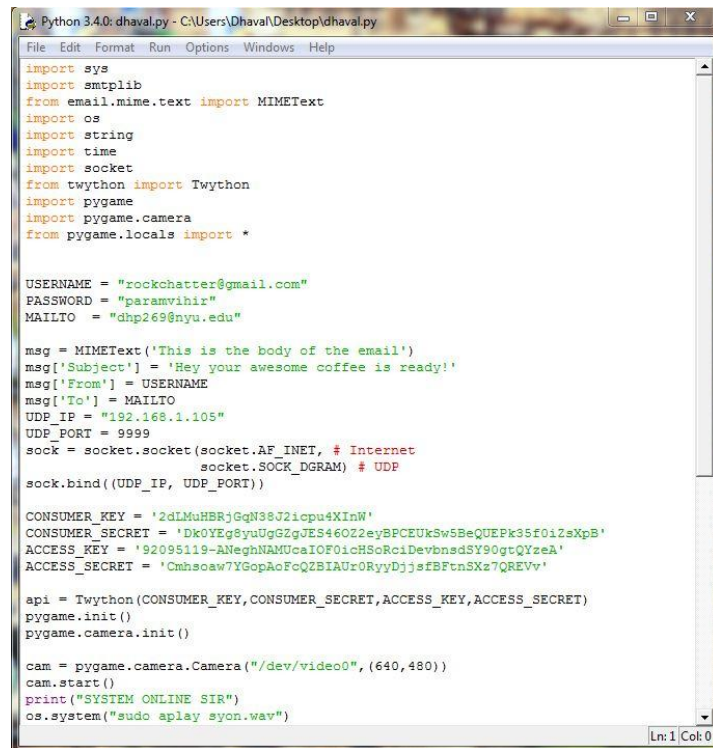
- Arduino is controlling two servomotors for the on and off switch on the coffeemaker. Ports 5 and 6.
- Android, Arduino, and Raspberry Pi are communicating over the NETGEAR router using UDP protocol.
- A web camera and monitor is attached to the Raspberry Pi to capture the image and tweet about the coffee.





Raspberry Pi

- Uses a Rasbian build.
- Uses python 3 script and UDP communication protocol
- Uses Smtplib for sending notifications over email and Twython for tweeting images as well as text.
- For tweeting it uses a Twitter authentication pipeline which allows it to use my account.
- Plays wav files when coffee is ready.



```
Python 3.4.0: dhaval.py - C:\Users\Dhaval\Desktop\dhaval.py
File Edit Format Run Options Windows Help

import sys
import smtplib
from email.mime.text import MIMEText
import os
import string
import time
import socket
from twython import Twython
import pygame
import pygame.camera
from pygame.locals import *

USERNAME = "rockchatter@gmail.com"
PASSWORD = "paramvihar"
MAILTO = "dhp269@nyu.edu"

msg = MIMEText('This is the body of the email')
msg['Subject'] = 'Hey your awesome coffee is ready!'
msg['From'] = USERNAME
msg['To'] = MAILTO
UDP_IP = "192.168.1.105"
UDP_PORT = 9999
sock = socket.socket(socket.AF_INET, # Internet
                     socket.SOCK_DGRAM) # UDP
sock.bind((UDP_IP, UDP_PORT))

CONSUMER_KEY = '2dLMuHBRjGqN38J2icpu4XInW'
CONSUMER_SECRET = 'Dk0Yeg8yuUgGZqJES460Z2ey8PCEUkSw5BeQUEFk35f012sXpB'
ACCESS_KEY = '92095119-ANeghNAMUcaIOF0icHSoRciDevbnsdSY90gtQYzeA'
ACCESS_SECRET = 'Cmhsoaw7YGopAocFqQ2BIAUroRyyDjjsfBFTnSXz7QREVv'

api = Twython(CONSUMER_KEY, CONSUMER_SECRET, ACCESS_KEY, ACCESS_SECRET)
pygame.init()
pygame.camera.init()

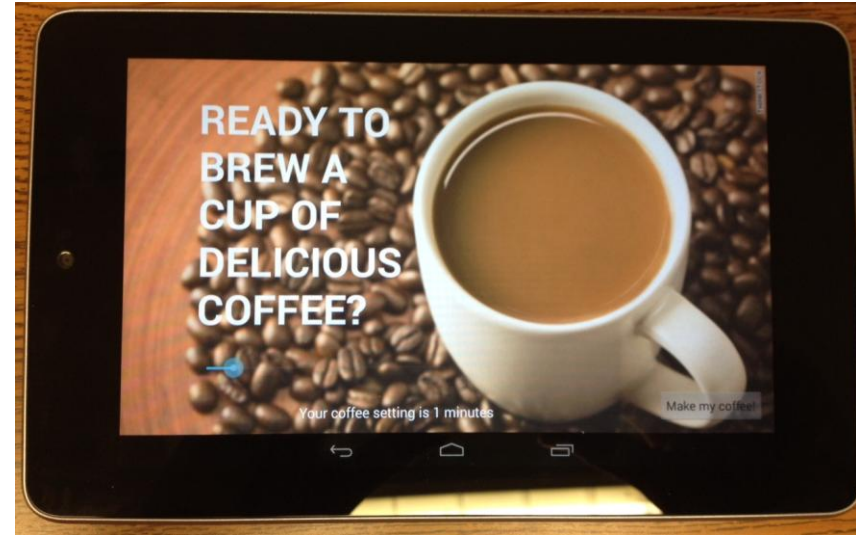
cam = pygame.camera.Camera("/dev/video0", (640, 480))
cam.start()
print("SYSTEM ONLINE SIR")
os.system("sudo aplay syon.wav")

Ln: 1 Col: 0
```



Android

- Creates a one screen interface that lets users choose the amount of coffee they want to make and a button for triggering the Arduino and Raspberry Pi
- Located on IP address 192.168.1.101
- Updates the slider bar until 'Make coffee' button is pressed.





Arduino

- Creates a UDP object and communicates at IP address 192.168.1.2 on port 9999.
- Connects two servomotors for the on and off buttons.
- Arduino is programmed to listen on the UDP channel until given the command to make coffee “Run:1P1:##”
- The Arduino presses the on button, waits for the given duration, and then presses the off button and listens for the next command.

```
by Michael Margolis

This code is in the public domain.
*/

#include <SPI.h>           // needed for Arduino versions later than 0018
#include <Ethernet.h>
#include <EthernetUdp.h>   // UDP library from: bjoern@cs.stanford.edu 12/30/2008

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 1, 2);

unsigned int localPort = 9999;      // local port to listen on

// buffers for receiving and sending data
char packetBuffer[UDP_TX_PACKET_MAX_SIZE]; //buffer to hold incoming packet,
char ReplyBuffer[] = "acknowledged";       // a string to send back

// An EthernetUDP instance to let us send and receive packets over UDP
EthernetUDP Udp;

void setup() {
  Serial.begin(9600);
  Serial.println("try begin");
  // start the Ethernet and UDP:
  Ethernet.begin(mac,ip);
  Serial.println("try udp");
  Udp.begin(localPort);
  Udp.flush();
  Serial.println("setup done");
}

void loop() {
```



Try our coffee!

**Thank you for your attention!
We would be happy to answer any
questions.**