Chapter 1: Internet-Controlled PowerSwitch

Getting started

Arduino Ethernet Shield

Arduino UNO R3 (Front View)

Arduino Ethernet Shield R3 (Front View)

Arduino Ethernet Shield R3 (top) is stacked with Arduino UNO R3 (bottom) (Front View)

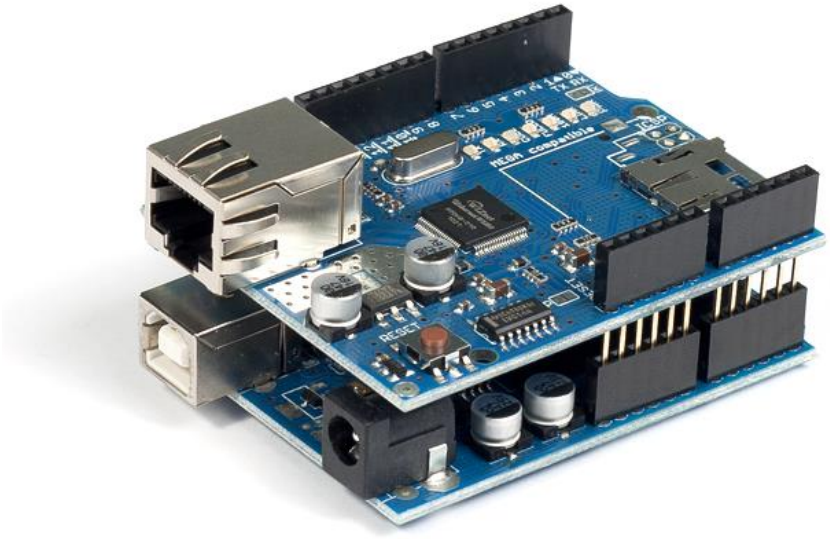
WIZnet W5100 Ethernet controller (Top View)

The Arduino Ethernet board

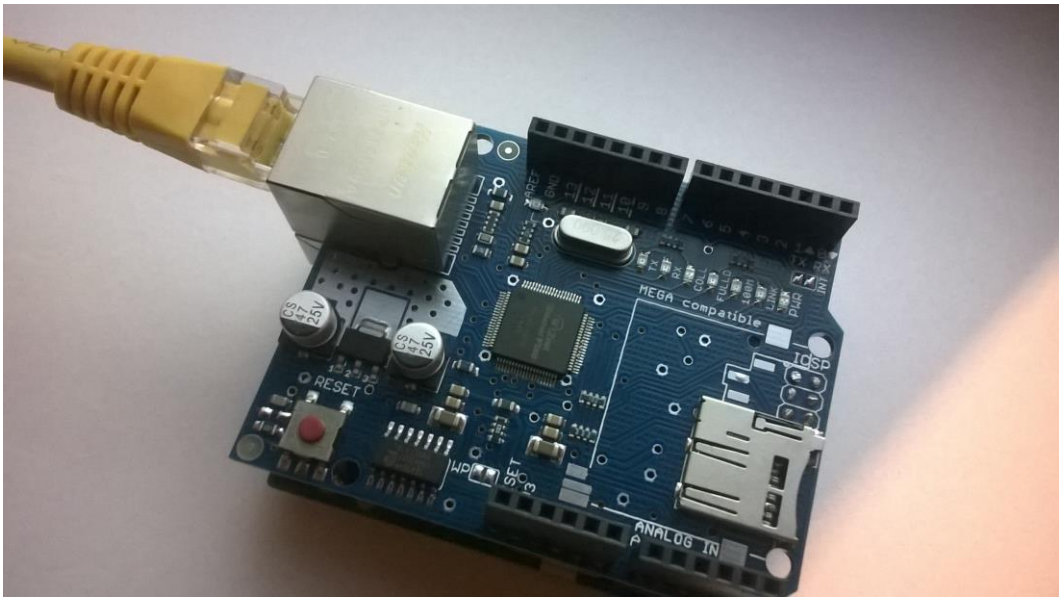
Arduino Ethernet Board (Front View)

FTDI cable 5V (Source: https://commons.wikimedia.org/wiki/File:FTDI_Cable.jpg)

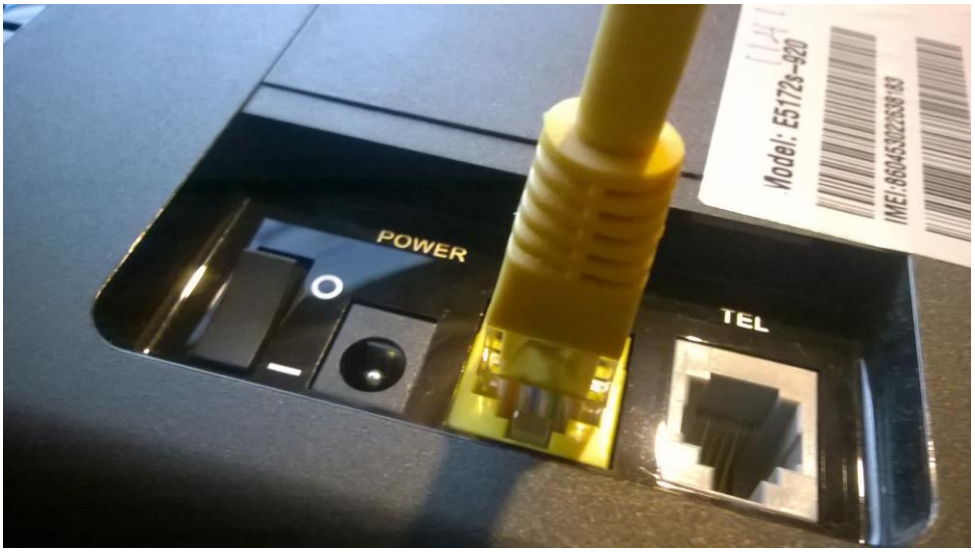
Connecting Arduino Ethernet Shield to the Internet



Fritzing representation of Arduino and Ethernet shield stack

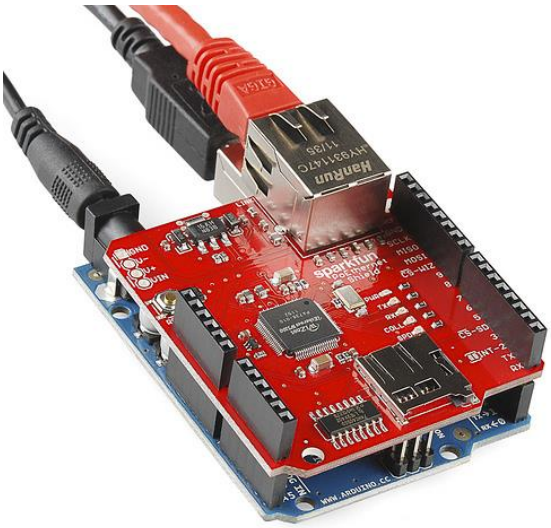


One end of the Ethernet cable is connected to the Arduino Ethernet board



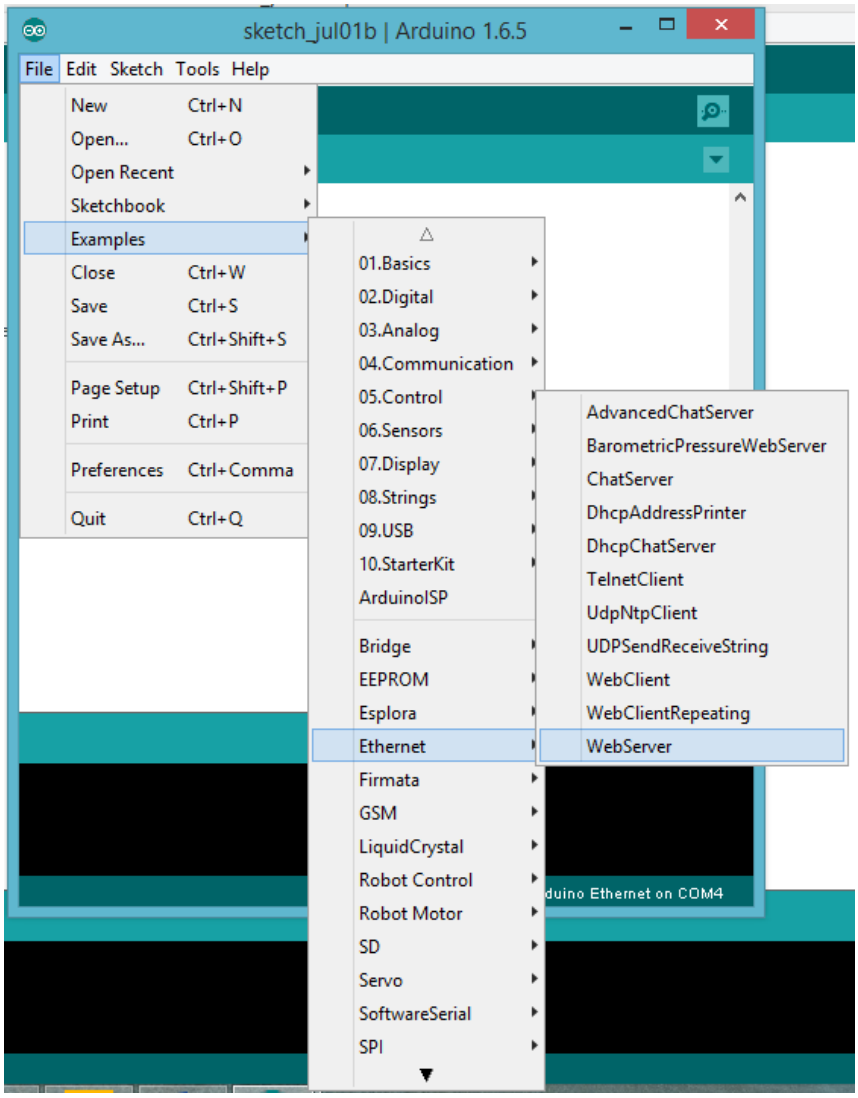
Other end of the Ethernet cable is connected to router/switch

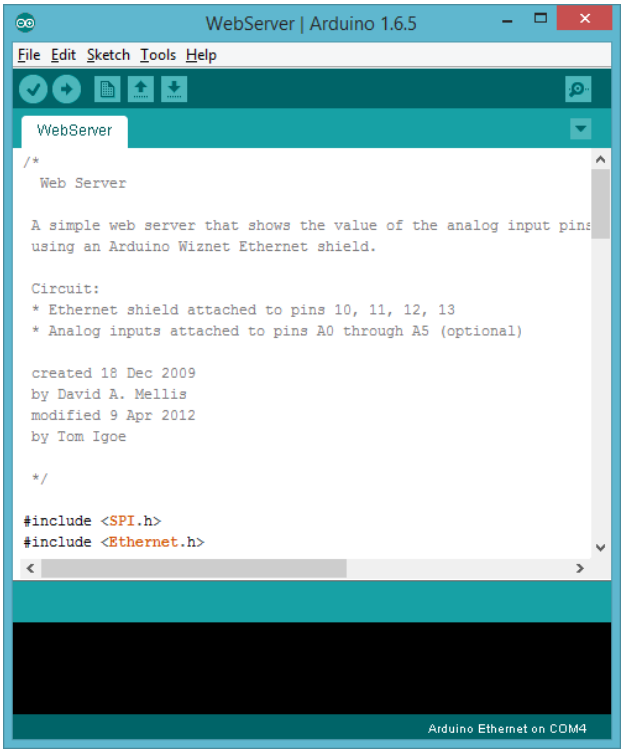
1.

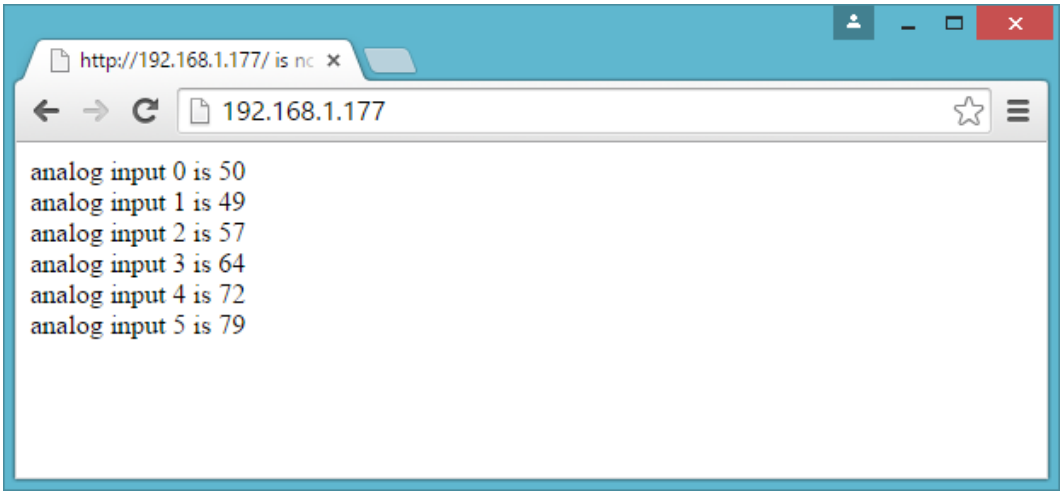


One end of the Ethernet cable is connected to the Ethernet shield (top) and the power connector and USB cable is connected to the Arduino board (bottom)

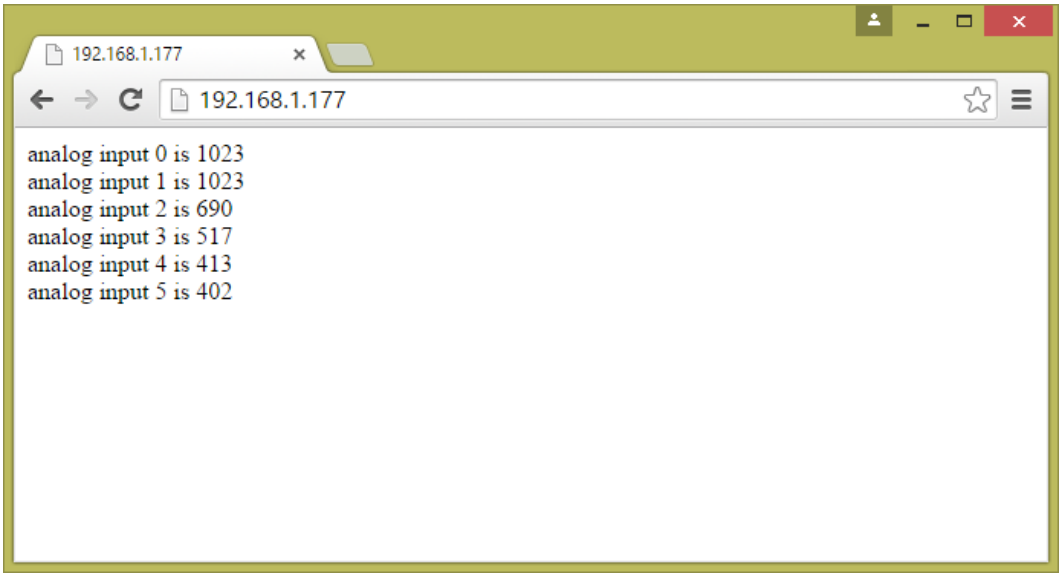
Testing your Arduino Ethernet Shield



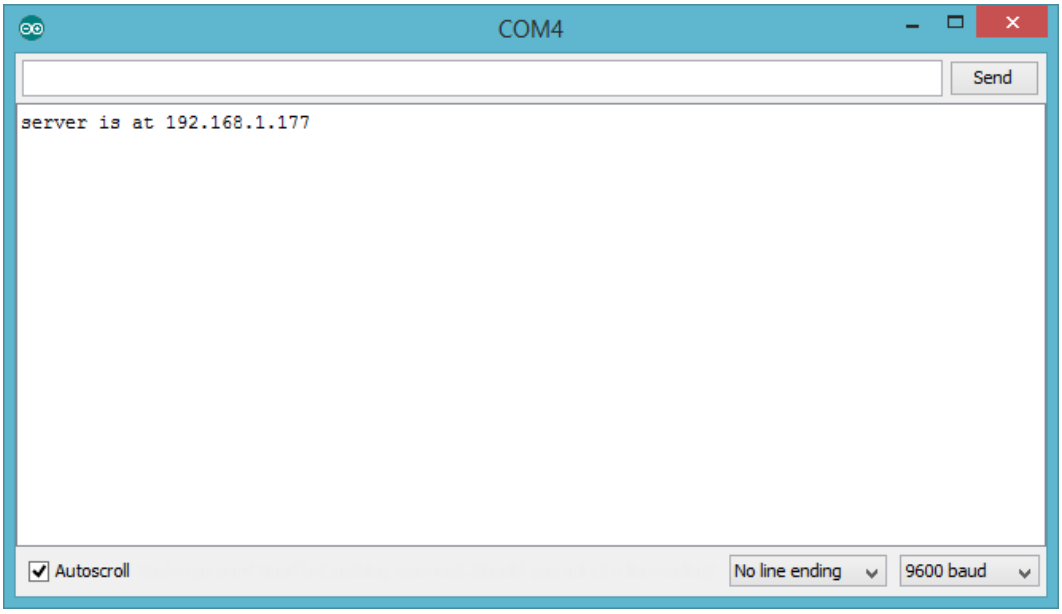




Output for Arduino Ethernet Board: Analog input values are displaying on Google Chrome browser, refreshes every 5 seconds



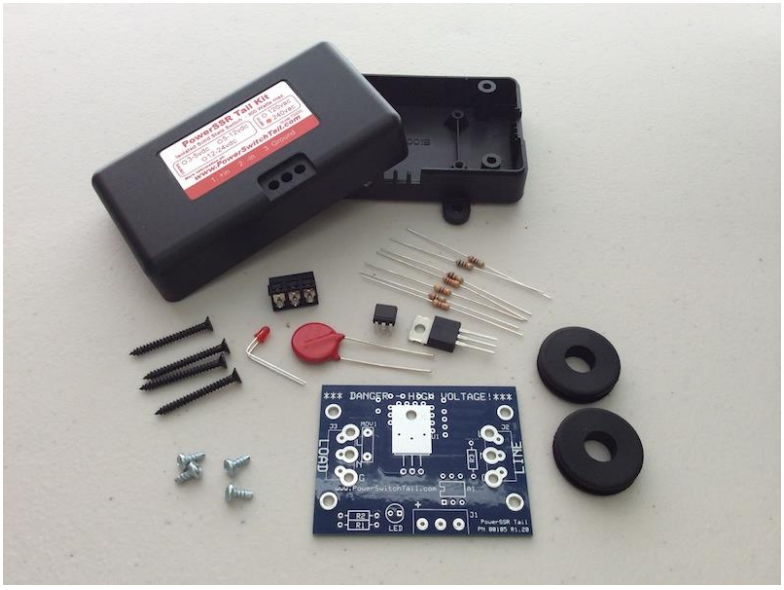
Output for Arduino UNO + Arduino Ethernet Shield: Analog input values are displaying on Google Chrome browser, refreshes every 5 seconds



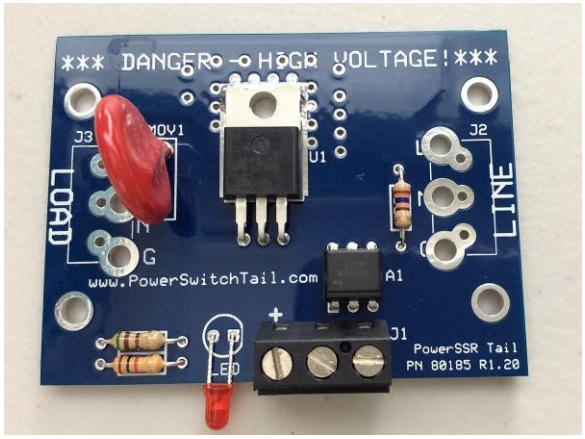
Arduino Serial Monitor Prints the static IP address of Arduino Ethernet Shield

Selecting a PowerSwitch Tail

PN PSSRKT-240



PN PSSRKT-240 Normally Open (NO) version—240V AC



PN PSSRKT-240 Normally Open (NO) version—240V

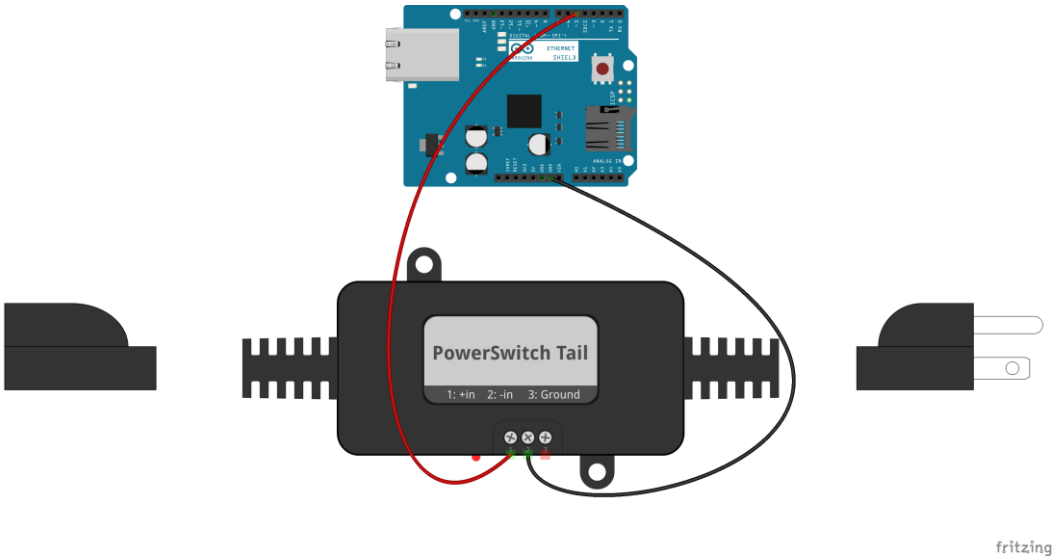
PN PSSRKT-240 Normally Open (NO) version—240V

PN80135

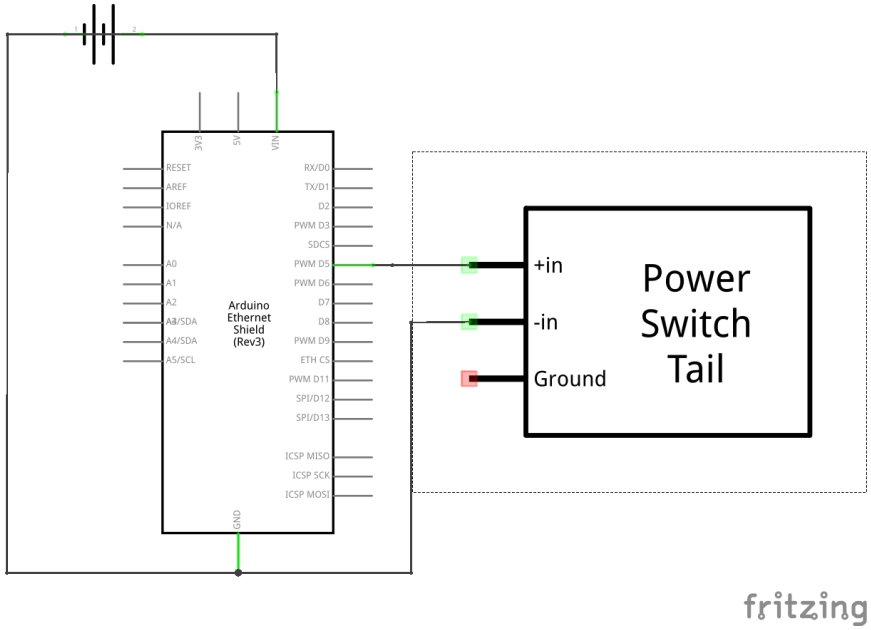
PN80135 Normally Open (NO) version—120V AC (left-hand side plug for LOAD and right-hand side plug for LINE)

Wiring PowerSwitch Tail with Arduino Ethernet Shield

Two wires from Arduino is connected to the PowerSwitch Tail



PowerSwitch Tail is connected with Ethernet Shield—Fritzing representation



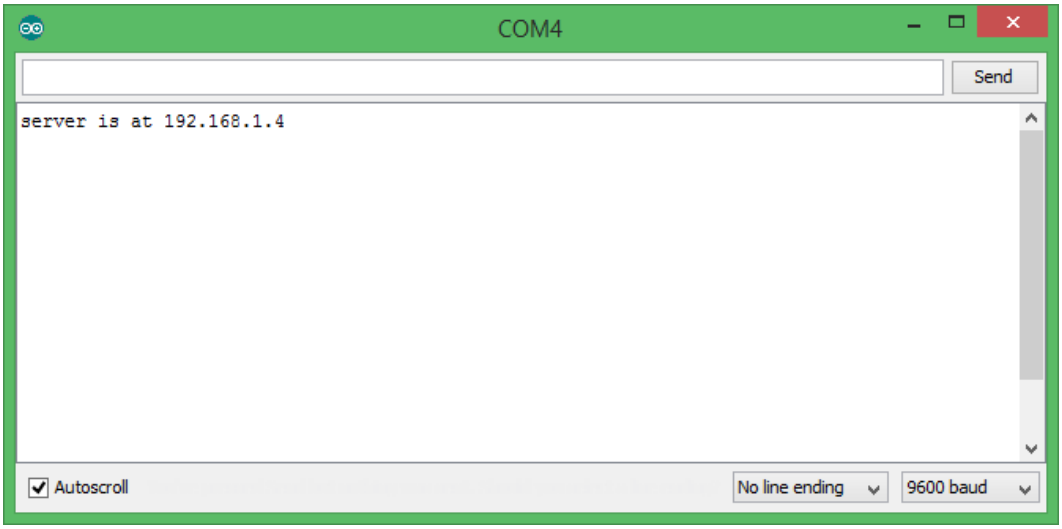
PowerSwitch Tail is connected with Ethernet Shield—Schematic

Turning PowerSwitch Tail into a simple web server

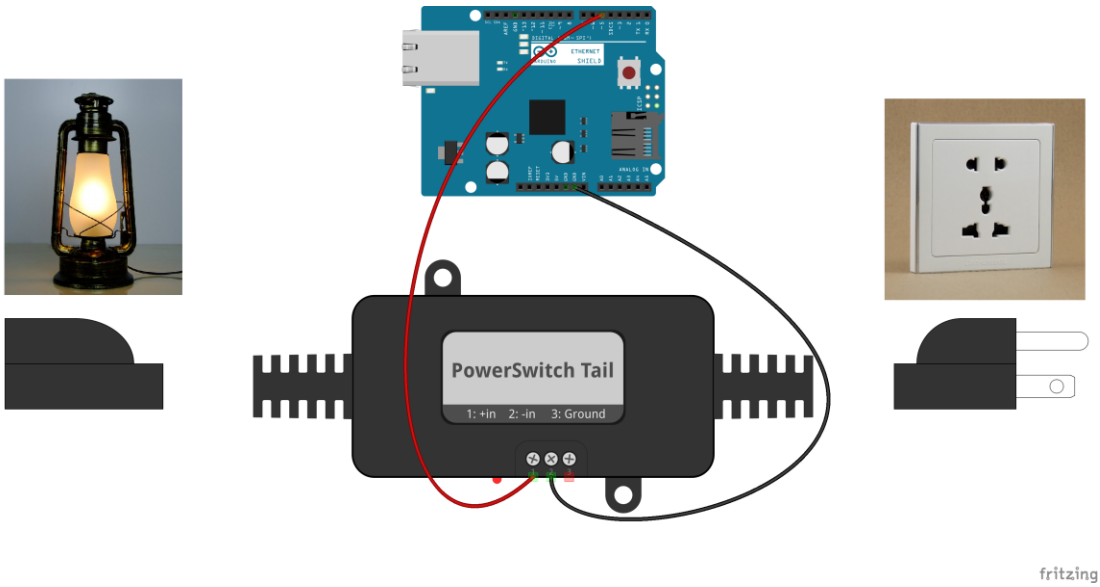
A step-by-step process of building a web-based control panel

Handling client requests by HTTP GET

.



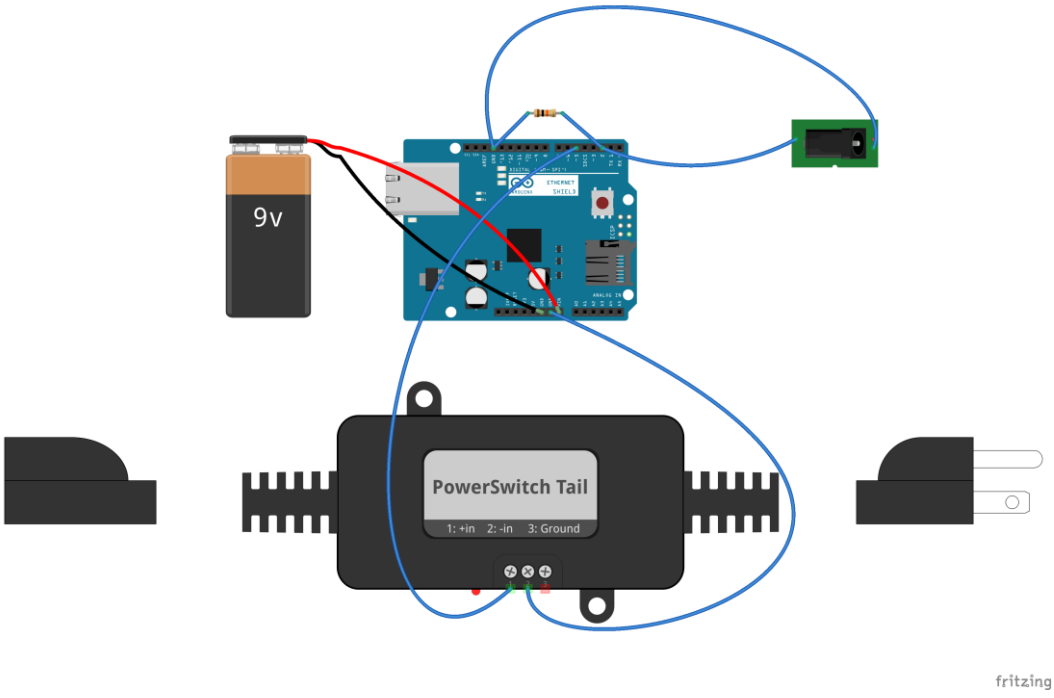
IP address assigned by the DHCP



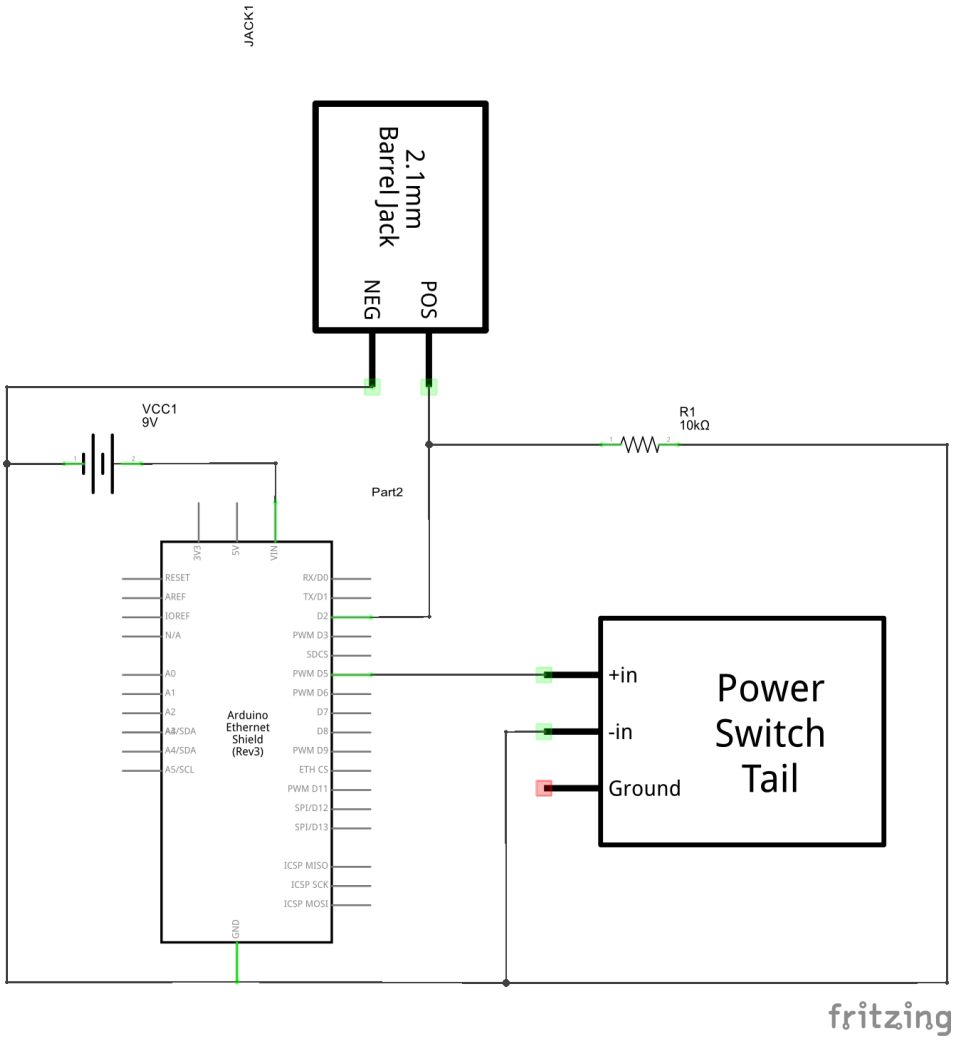
Electrical lamp is controlled by PowerSwitch Tail

PowerSwitch Tail control panel accessed by Google Chrome browser

Sensing the availability of mains electricity



Wiring diagram


JACK1
2.1mm
Barrel jack
NEG
POS
VCC1
9V
R1
10kΩ
Part2
3V3
5V
VIN
RESET
AREF
IOREF
N/A
A0
A1
A2
A3/SDA
A4/SDA
A5/SCL
RX/D0
TX/D1
D2
PWM D3
SDCS
PWM D5
PWM D6
D7
D8
PWM D9
ETH CS
PWM D11
SPI/D12
SPI/D13
ICSP MISO
ICSP SCK
ICSP MOSI
GND
Arduino
Ethernet
Shield
(Rev3)
+in
-in
Ground
Power
Switch
Tail
fritzing

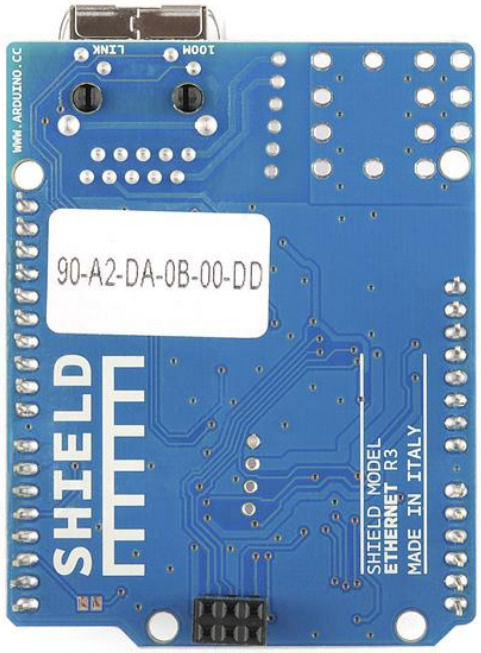
Schematic

Adding Cascade Style Sheet to the web user interface

MetroUI CSS style applied to radio buttons

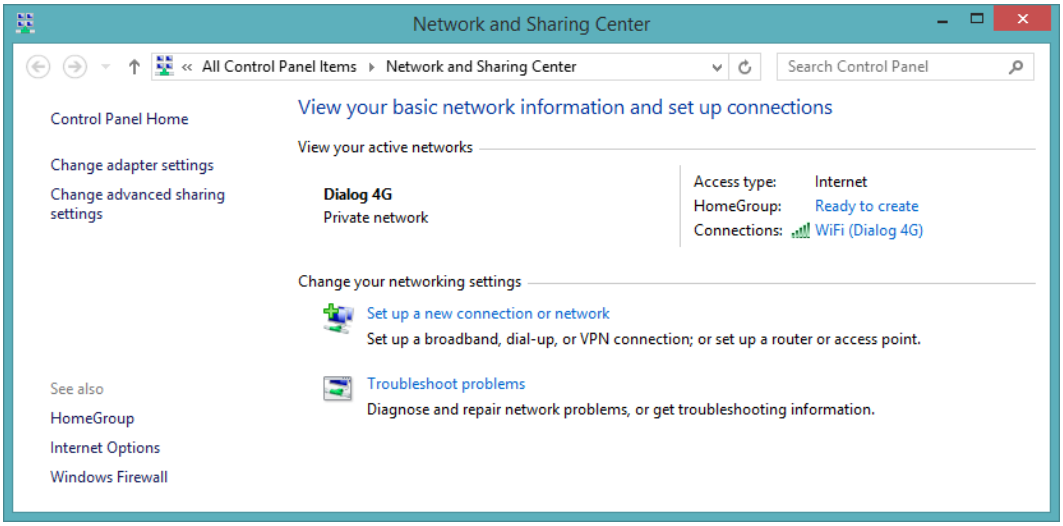
Finding the MAC Address and obtaining a valid IP address

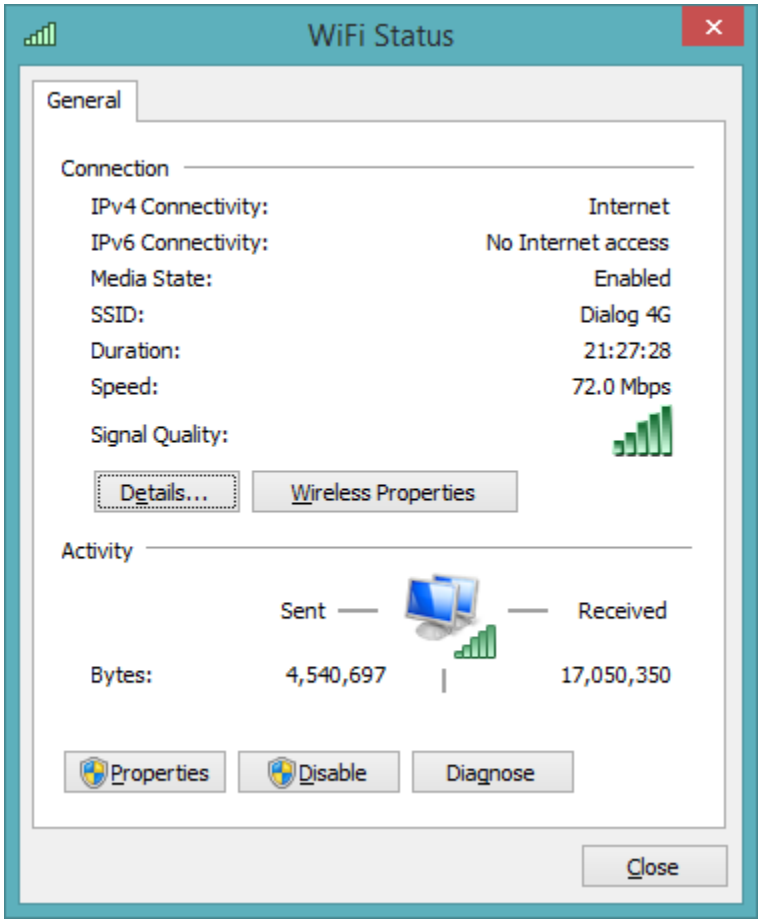
Finding the MAC address

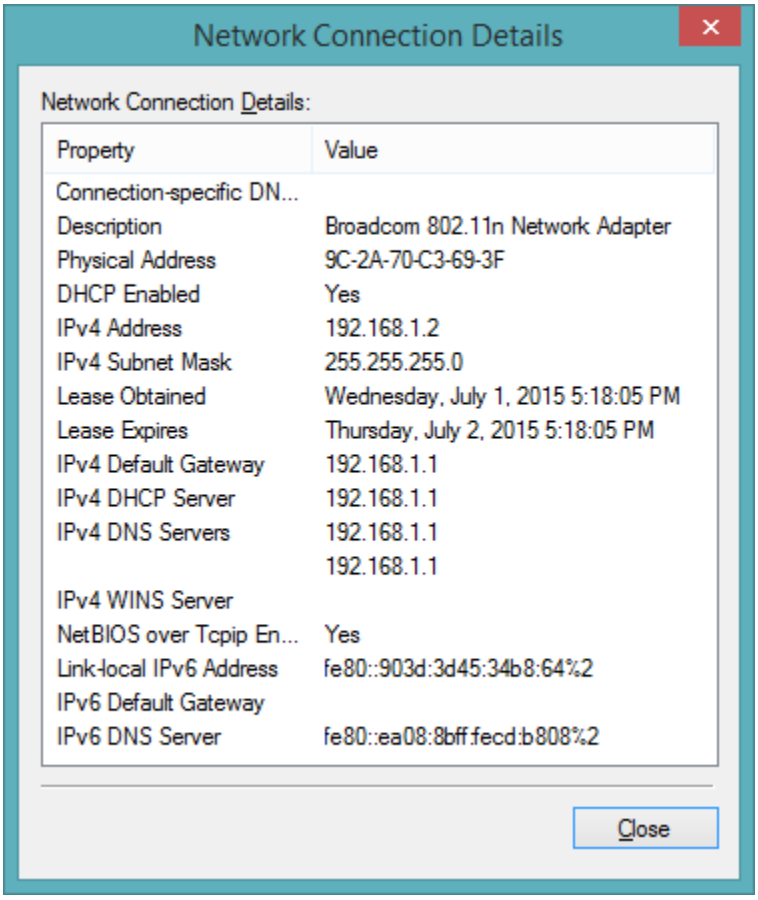


Obtaining an IP Address

Assigning a static IP address

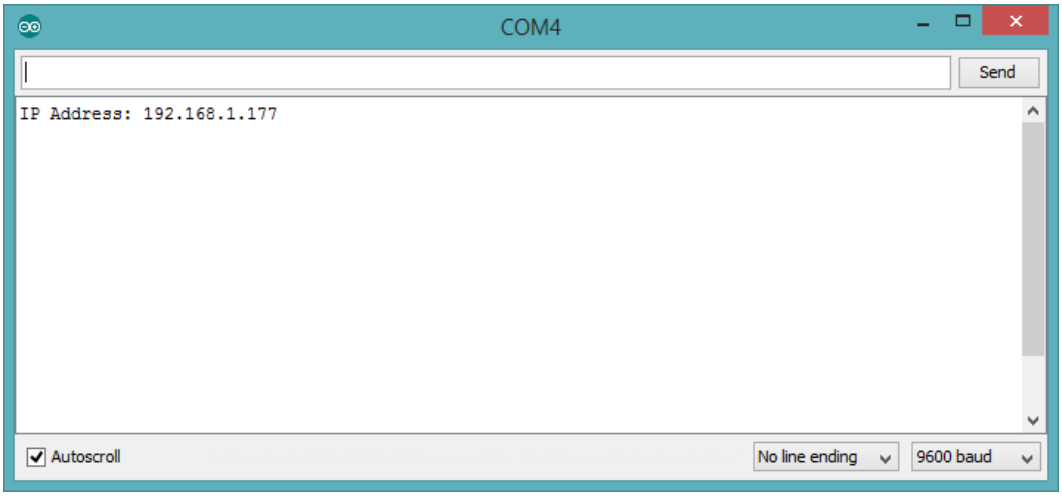



WiFi Status
General
Connection
IPv4 Connectivity:
Internet
IPv6 Connectivity:
No Internet access
Media State:
Enabled
SSID:
Dialog 4G
Duration:
21:27:28
Speed:
72.0 Mbps
Signal Quality:
Details...
Wireless Properties
Activity
Sent
Received
Bytes:
4,540,697
17,050,350
Properties
Disable
Diagnose
Close



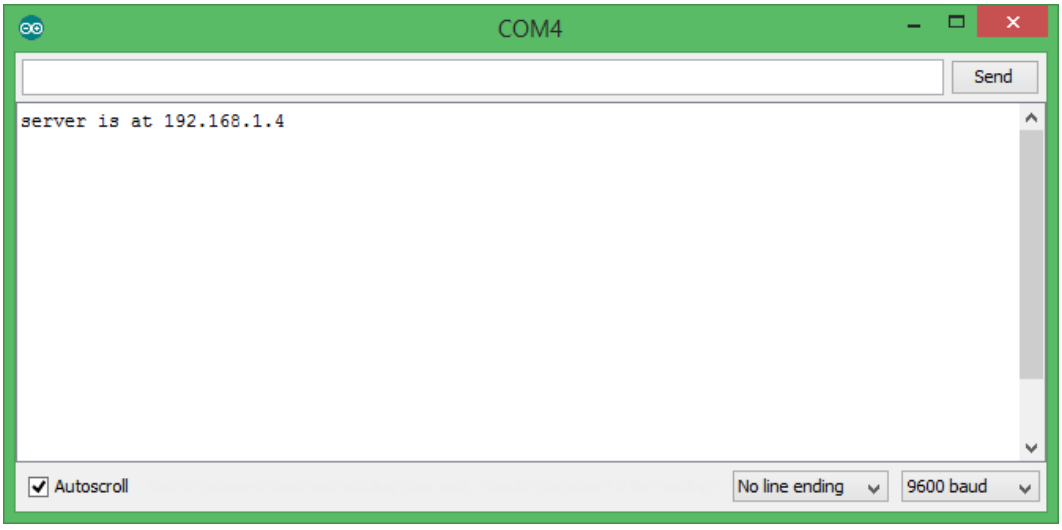
Device List

Index	Computer Name	MAC Address	IP Address	Lease Time	Status	Type	Operation
1	DELL	9C:2A:70:C3:69:3F	192.168.1.2	0 days 22 hours 58 minutes 38 seconds	Active	Wi-Fi	Kick Out
2	Windows-Phone	A8:44:81:43:AD:C4	192.168.1.3	0 days 22 hours 59 minutes 51 seconds	Active	Wi-Fi	Kick Out



Static IP address

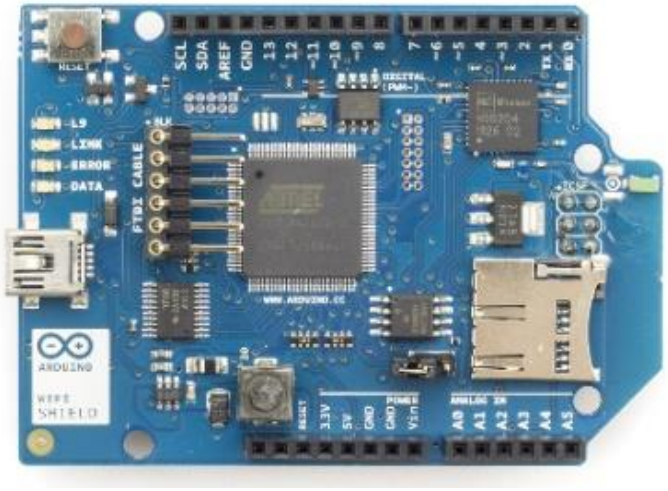
Obtaining an IP address using DHCP



DHCP assigned IP address

Chapter 2: Wi-Fi Signal Strength Reader and Haptic Feedback

Arduino WiFi Shield



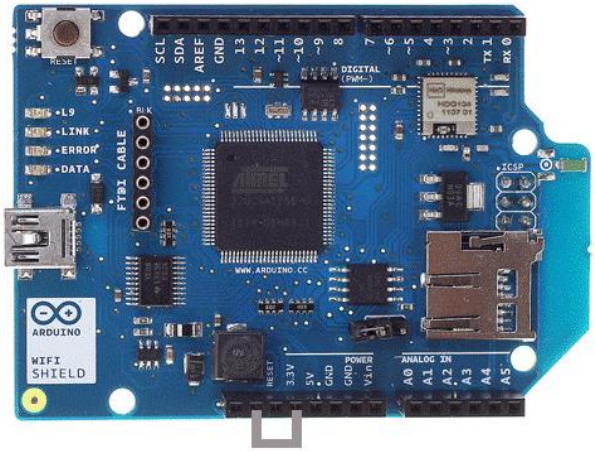
Arduino WiFi Shield (Top View)

Arduino WiFi Shield (Bottom View)

Stacking WiFi Shield with Arduino

Arduino WiFi shield is stacked with Arduino UNO

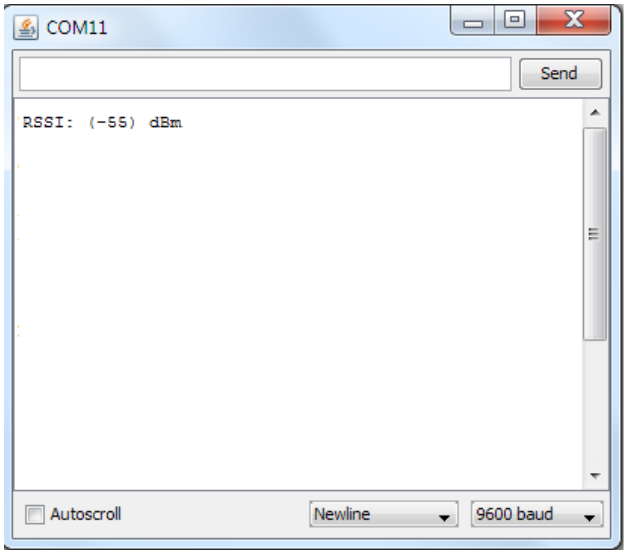
Hacking an Arduino earlier than REV3

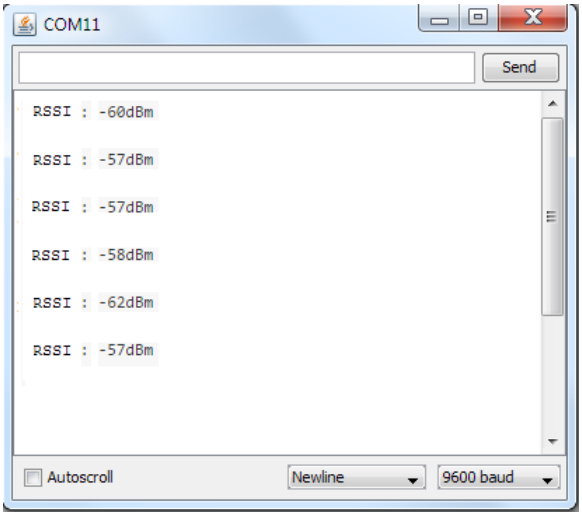


Jumper wire attached from 3.3V TO IOREF

WiFi signal strength and RSSI

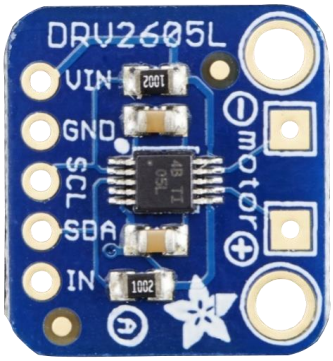
Reading the Wi-Fi signal strength





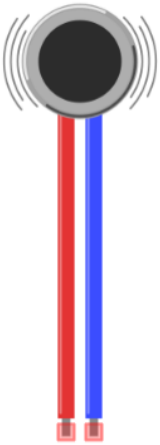
Haptic feedback and haptic motors

Getting started with the Adafruit DRV2605 haptic controller



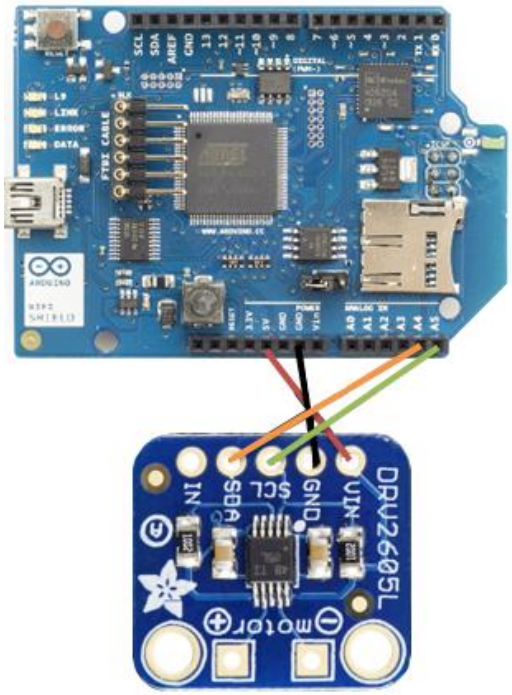
DRV2605 breakout board (Top View)

Selecting a correct vibrator

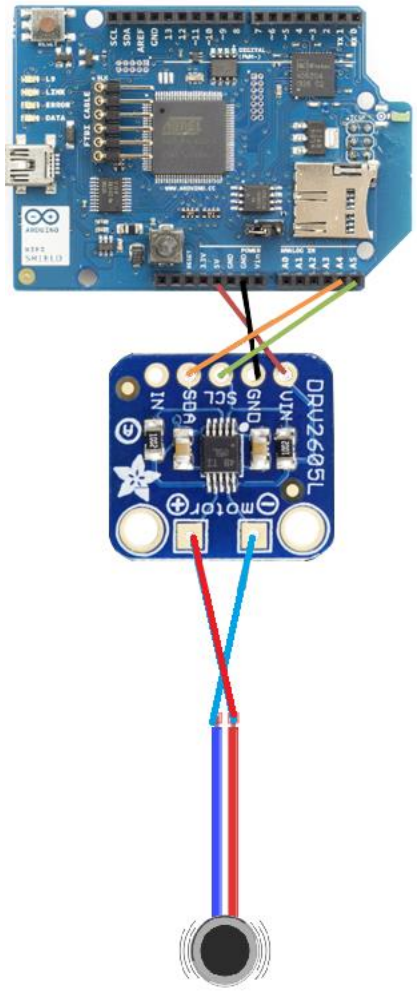


Fritzing representation of a vibrator

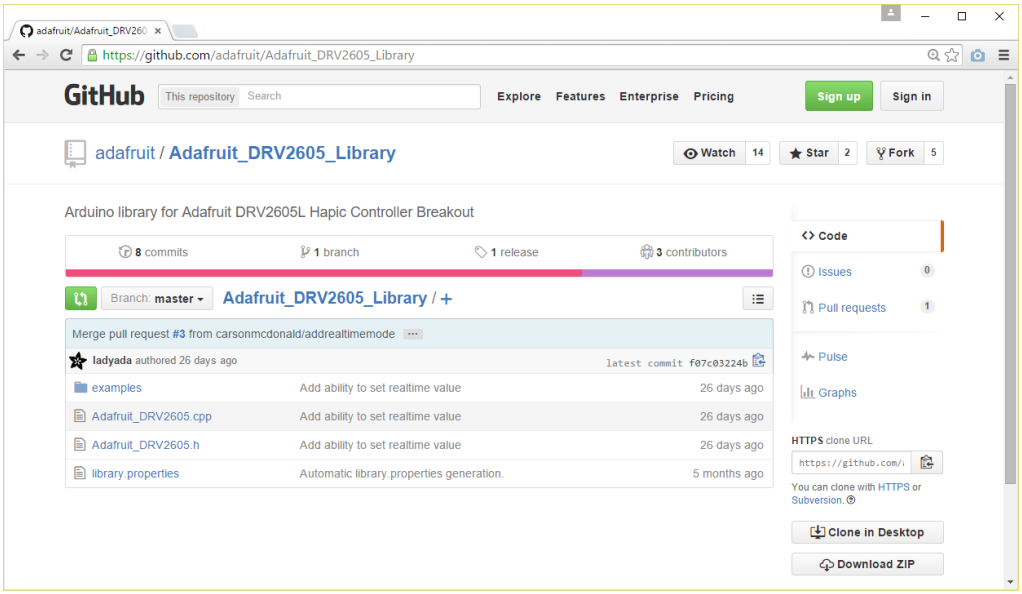
Connecting a haptic controller to Arduino WiFi Shield



Soldering a vibrator to the haptic controller breakout board



Downloading the Adafruit DRV2605 library



Adafruit DRV2605 library at GitHub

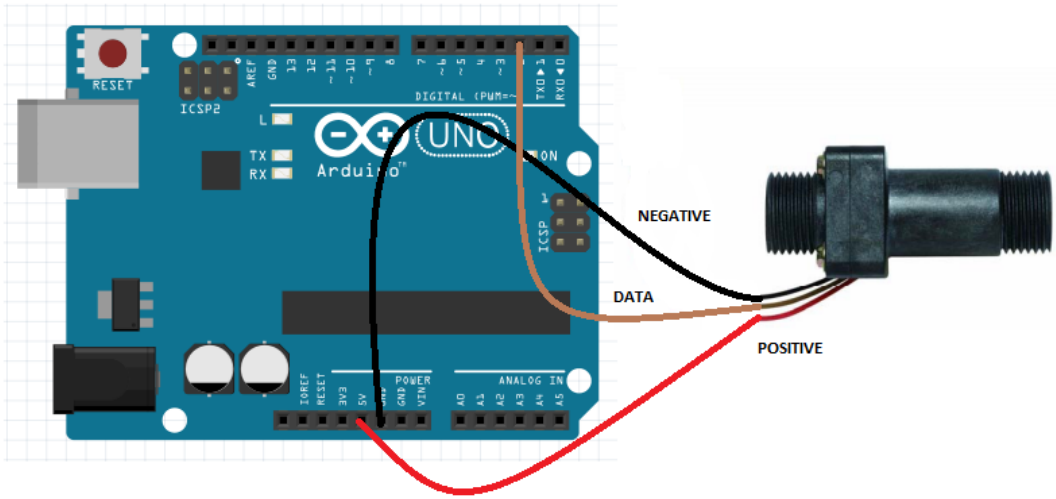
Chapter 3: Internet Connected Smart Water Meter

Water flow sensors

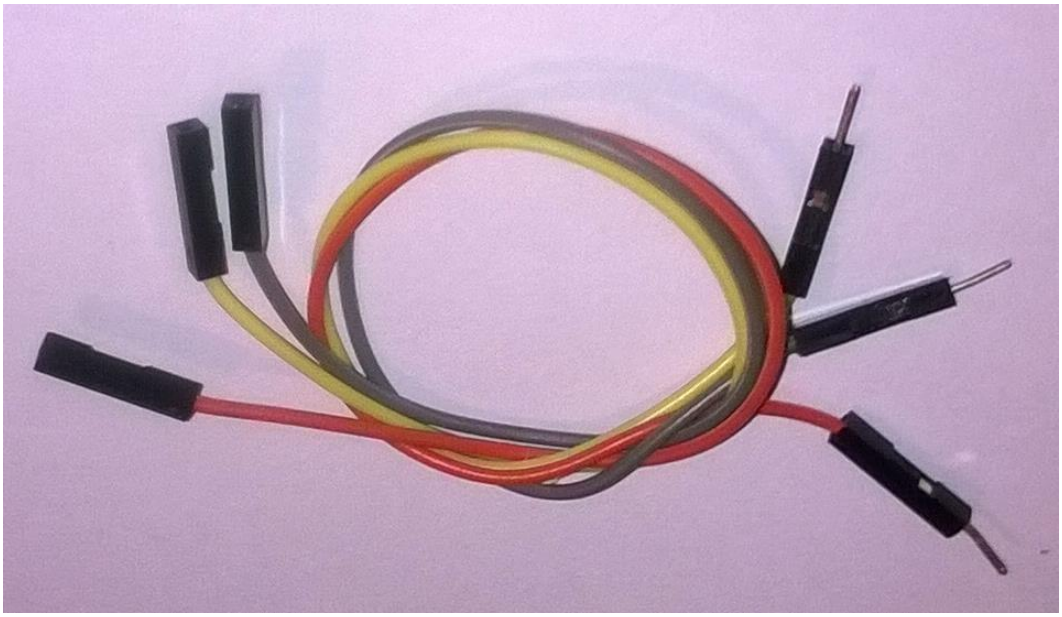
Liquid flow sensor – the flow direction is marked with an arrow

Pinwheel attached inside the water flow sensor

Wiring the water flow sensor with Arduino



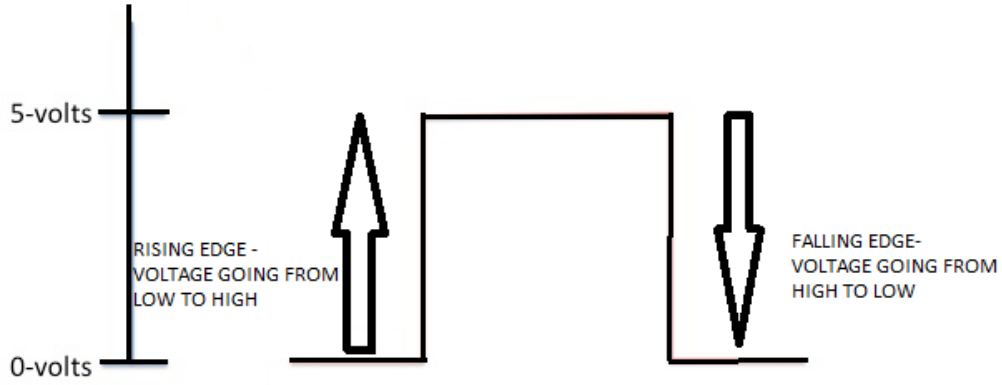
Water flow sensor connected with Arduino Ethernet Shield using three wires



Jumper wires with male and female headers

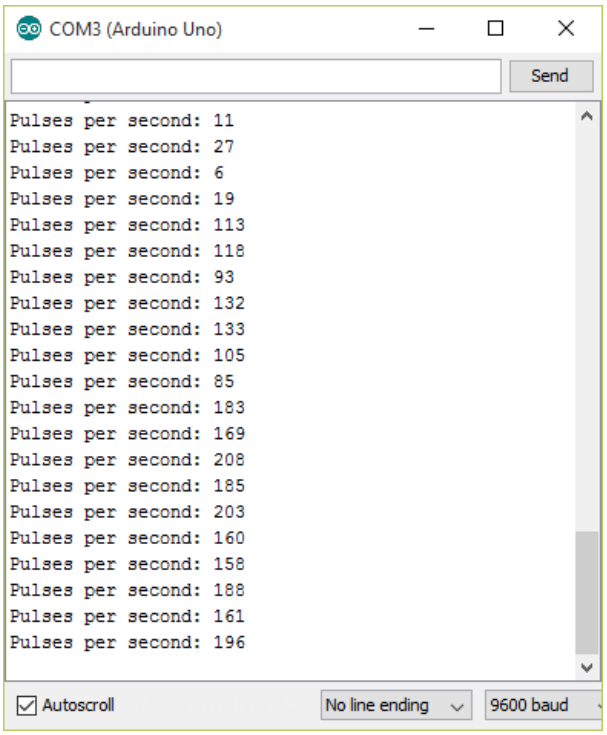
Rising edge and falling edge

.



Representation of Rising edge and Falling edge in digital signal

Reading and counting pulses with Arduino



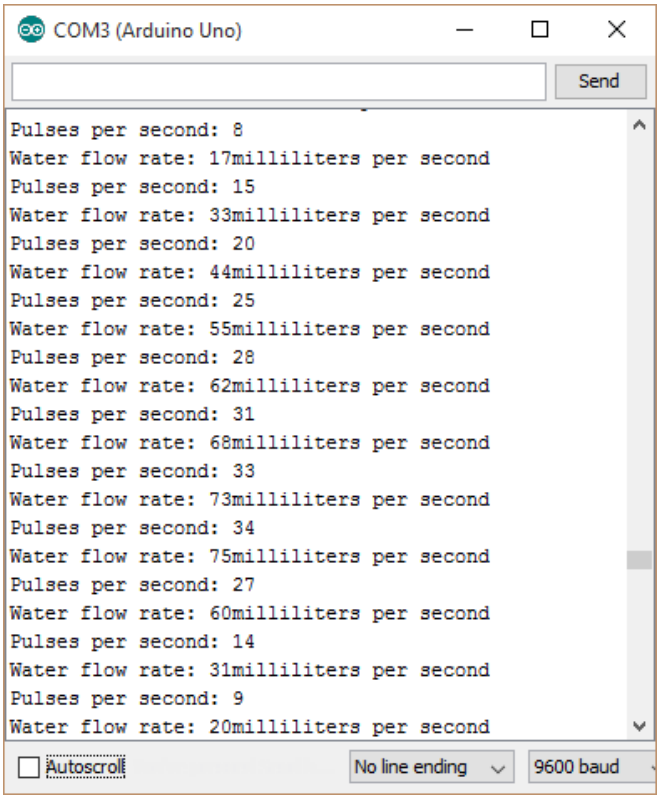
Pulses per second in each loop

Calculating the water flow rate

$$R = \frac{\text{n (pulse per second)}}{\text{m (pulse per litre)}}$$

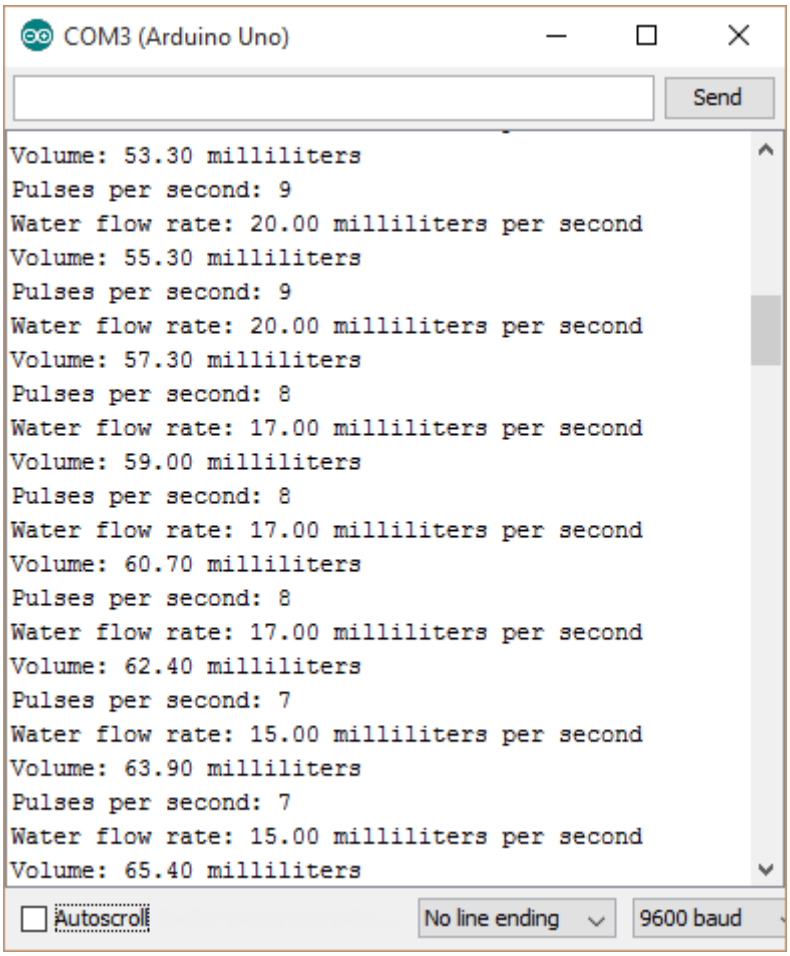
In liters per second

$$R = \frac{\text{n} * 60 \text{ (pulse per minute)}}{\text{m (pulse per litre)}}$$



Pulses per second and water flow rate in each loop

Calculating the water flow volume

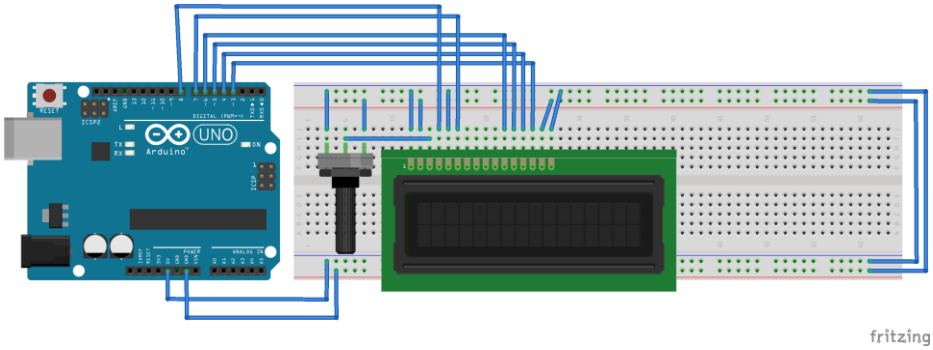


Pulses per second, water flow rate and in each loop and sum of volume

Adding an LCD screen to water meter

Hitachi HD44780 driver compatible LCD screen (16 x 2)—Top View

Hitachi HD44780 driver compatible LCD screen (16x2)—Bottom View



Fritzing representation of the circuit

LCD screen output

Converting your water meter to a web server

A little bit about plumbing

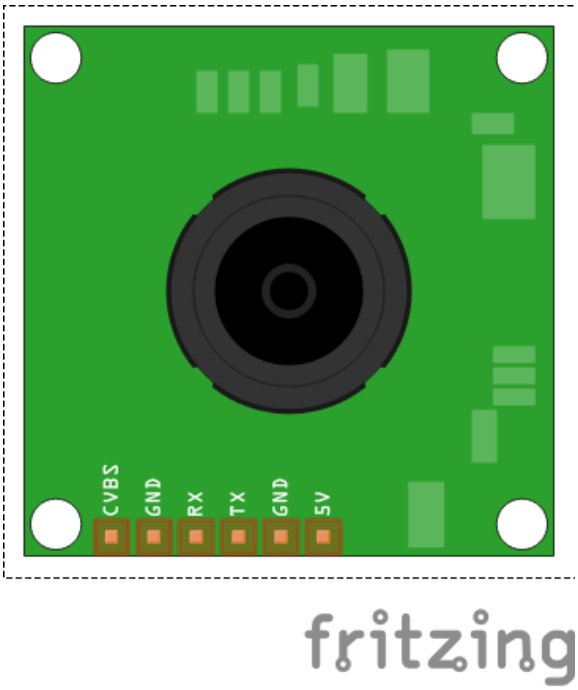
BNC pipe line connector made by PVC

Securing the connection between water flow meter and BNC pipe connector using thread seal

Image taken from https://www.flickr.com/photos/ttrimm/7355734996/

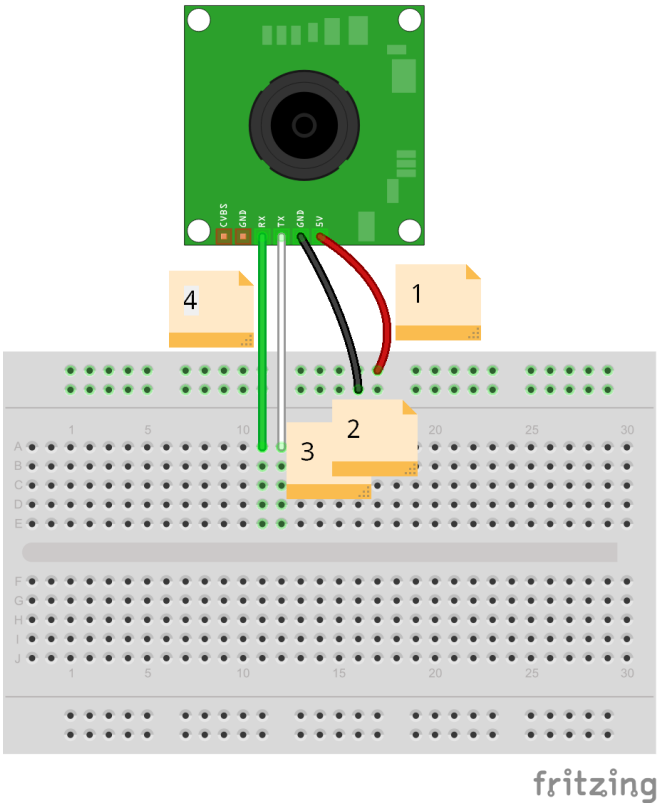
Chapter 4: Arduino Security Camera with Motion Detection

Getting started with TTL Serial Camera



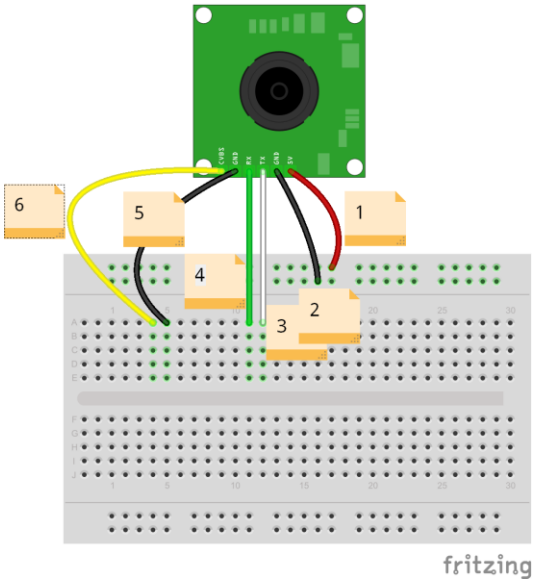
Fritzing representation of TTL Serial Camera—Top View

Wiring the TTL Serial Camera for image capturing



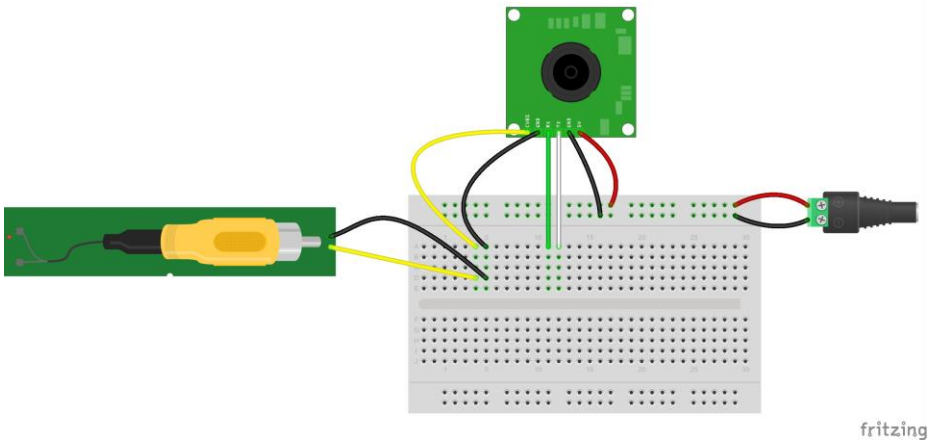
Wiring for image capturing in the JPEG format

Wiring the TTL Serial Camera for video capturing

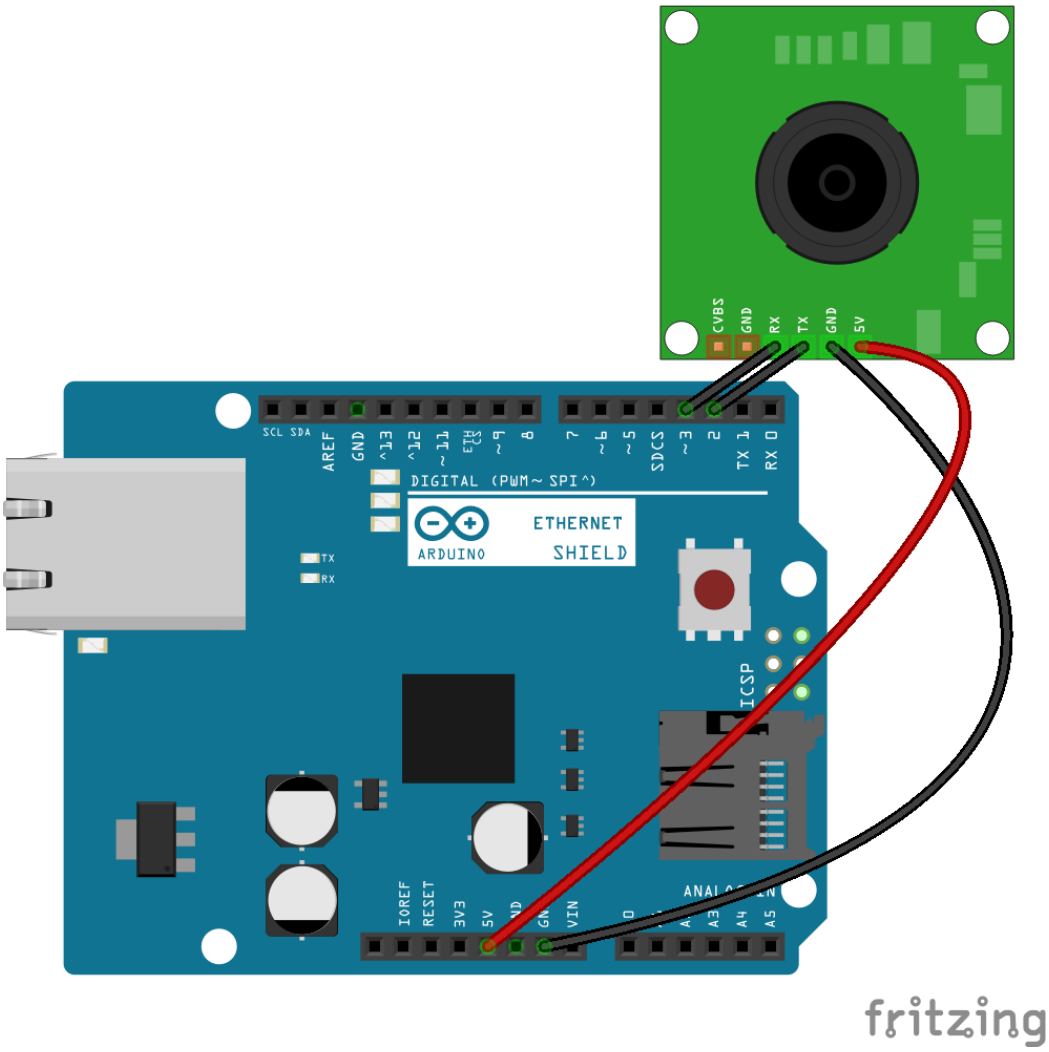


Wiring for video capturing with NTSC monochrome

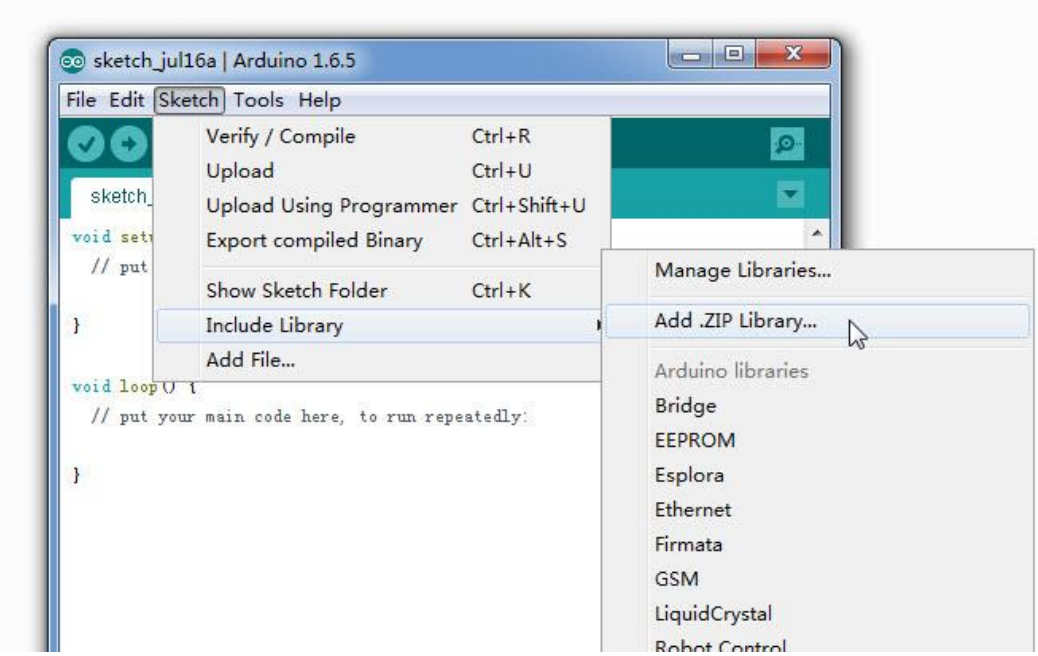
Testing NTSC video stream with video screen



Connecting TTL Serial Camera with Arduino and Ethernet Shield



Adafruit VC0706 Serial JPEG Camera is connected with Arduino Ethernet Shield

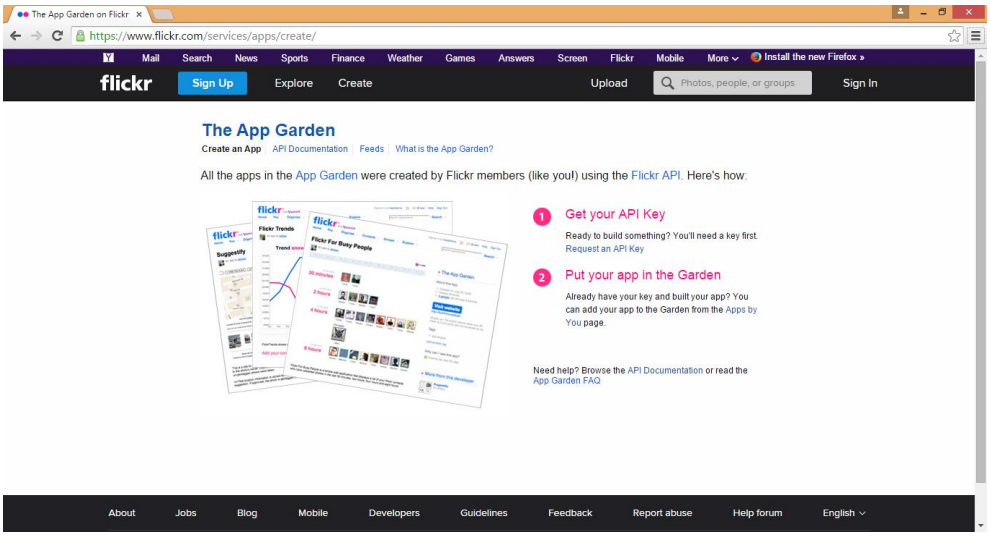


Including new library by a ZIP file

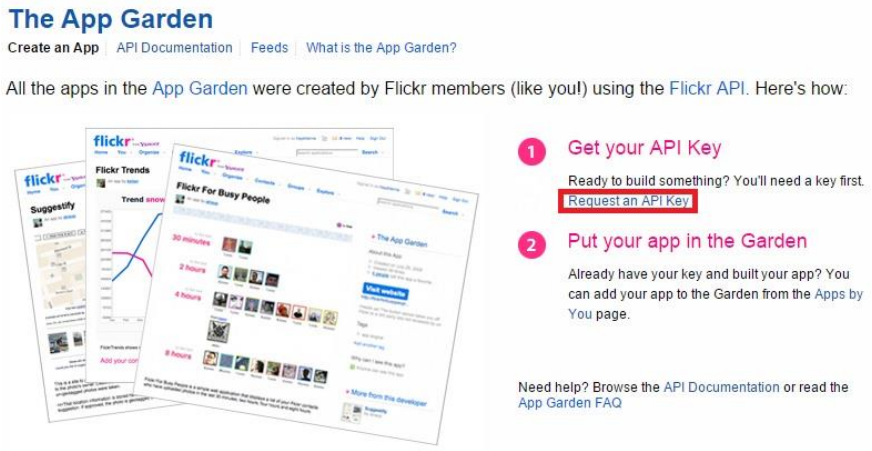
Uploading images to Flickr

Creating a Flickr account

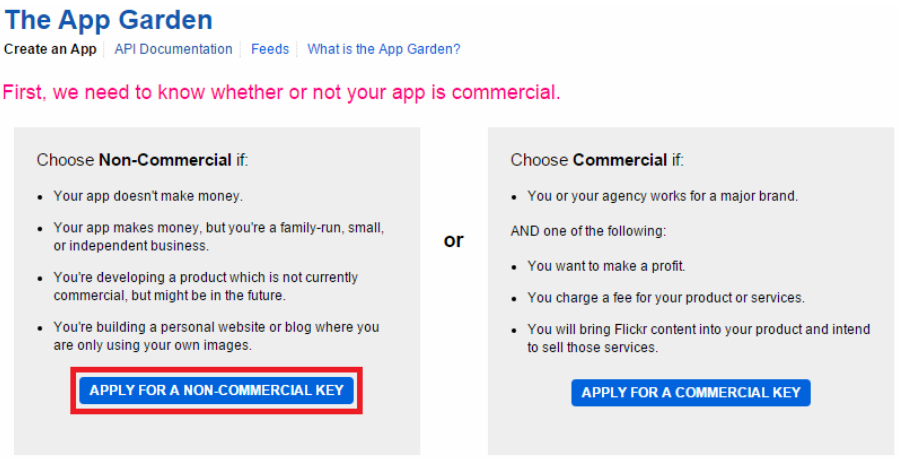
:



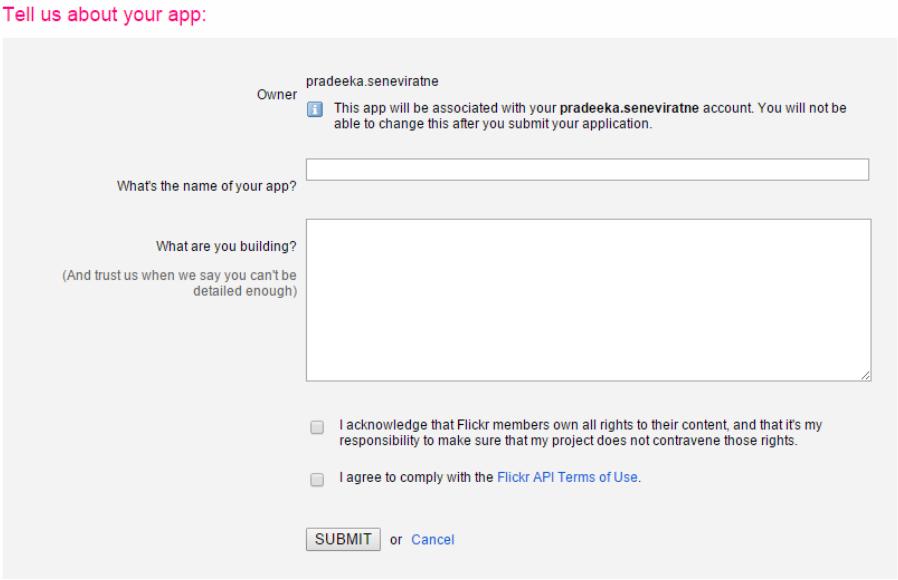
Flickr: The App Garden page



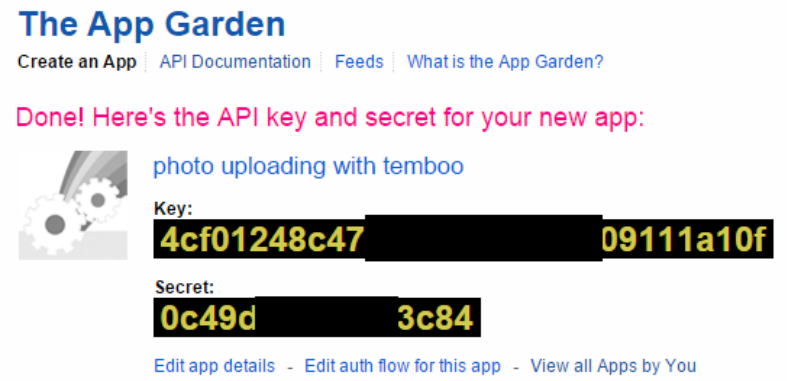
Flickr: The App Garden page



Flickr: The App Garden page

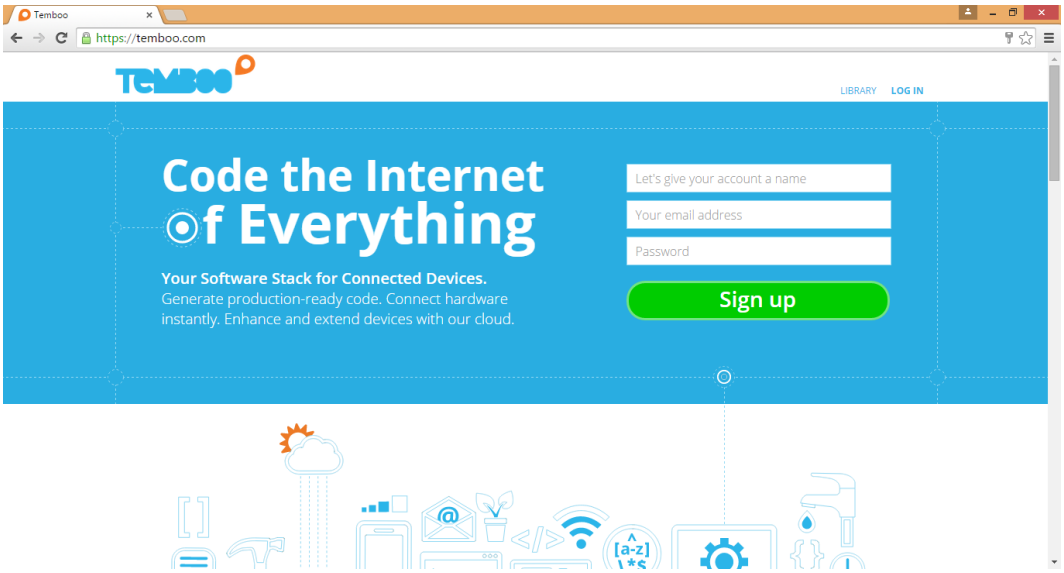


Flickr: The App Garden page

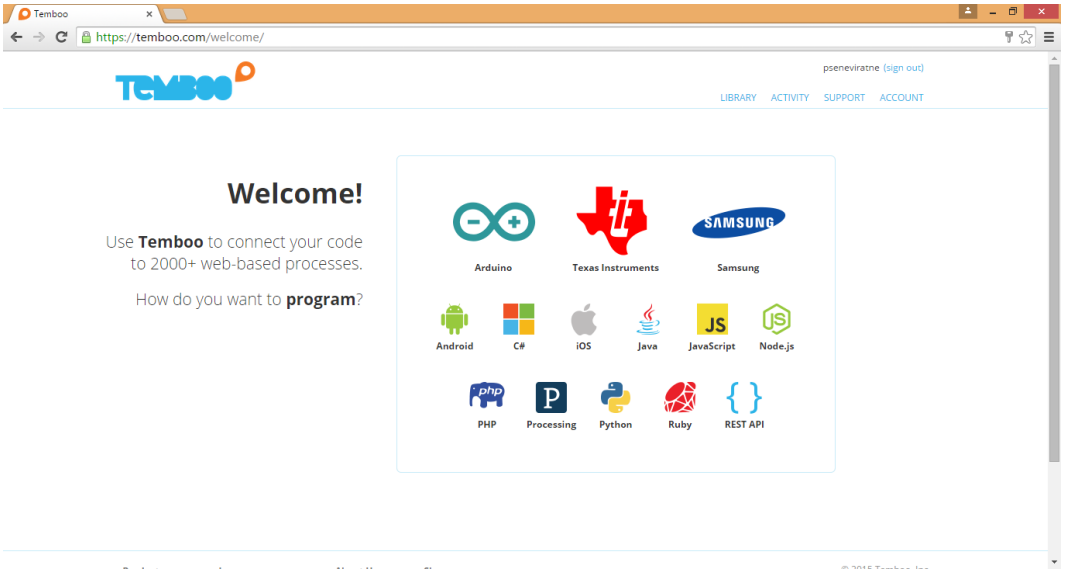


Flickr: The App Garden page

Creating a Temboo account



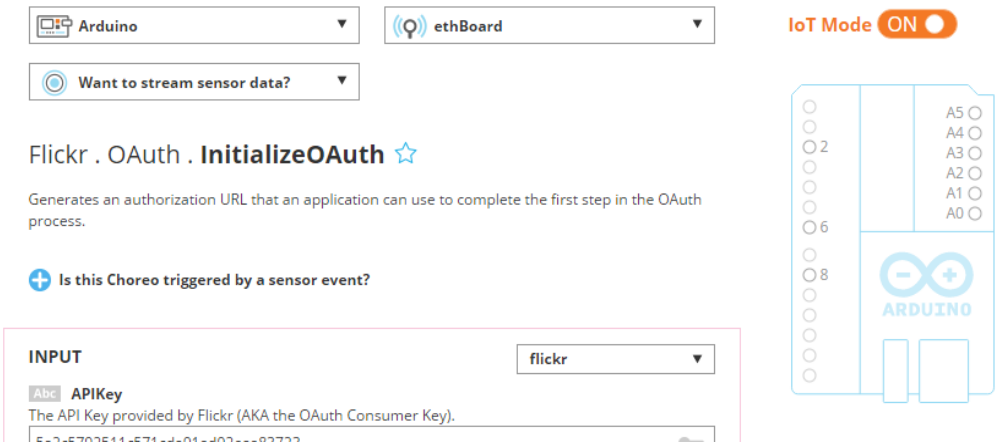
Temboo home page



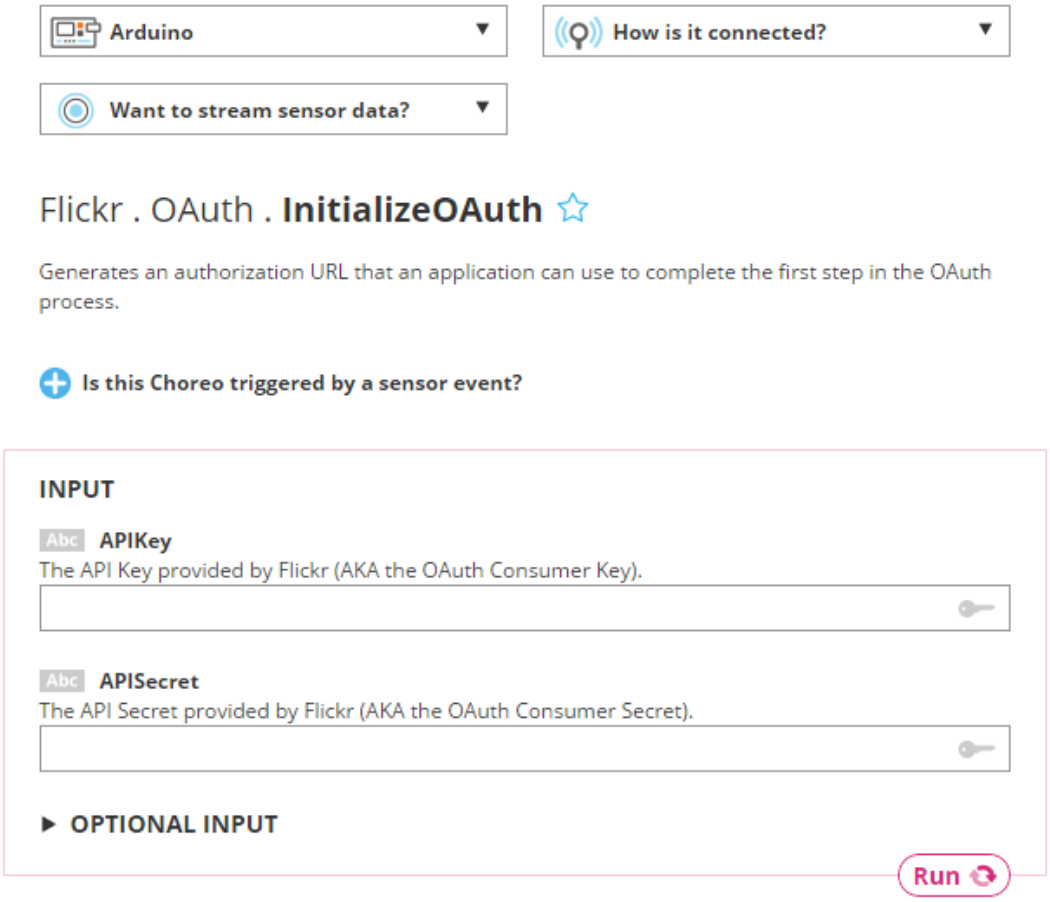
Temboo Welcome page

Creating your first Choreo

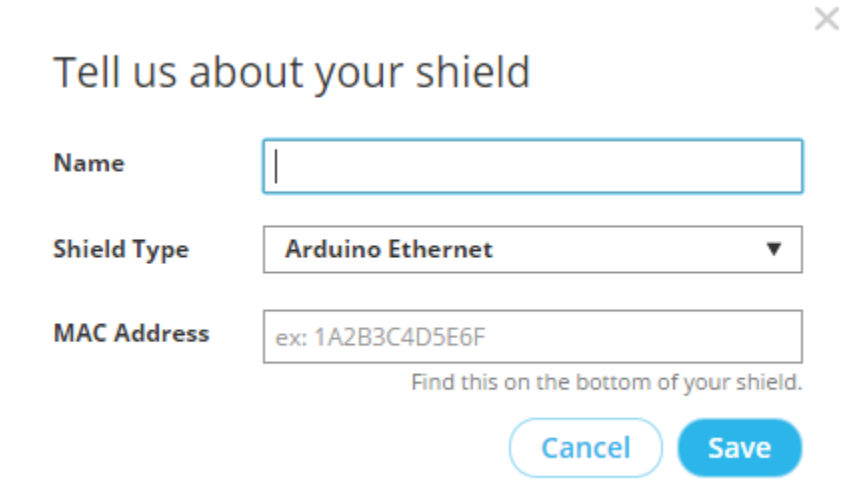
Initializing OAuth



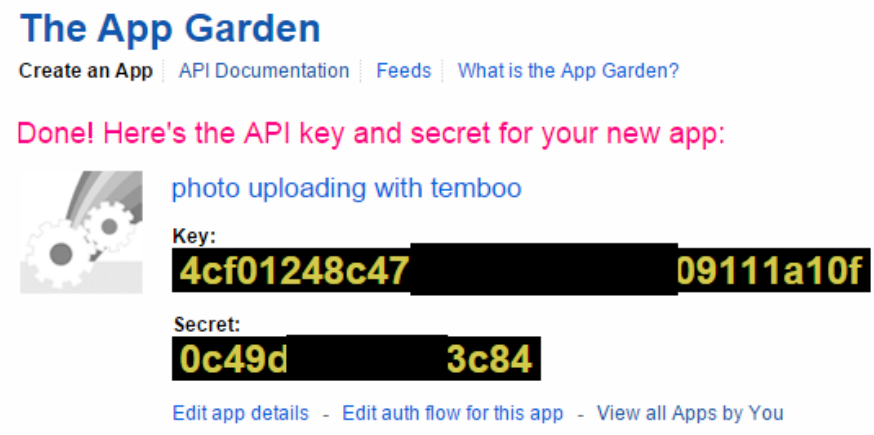
Enabling IoT mode

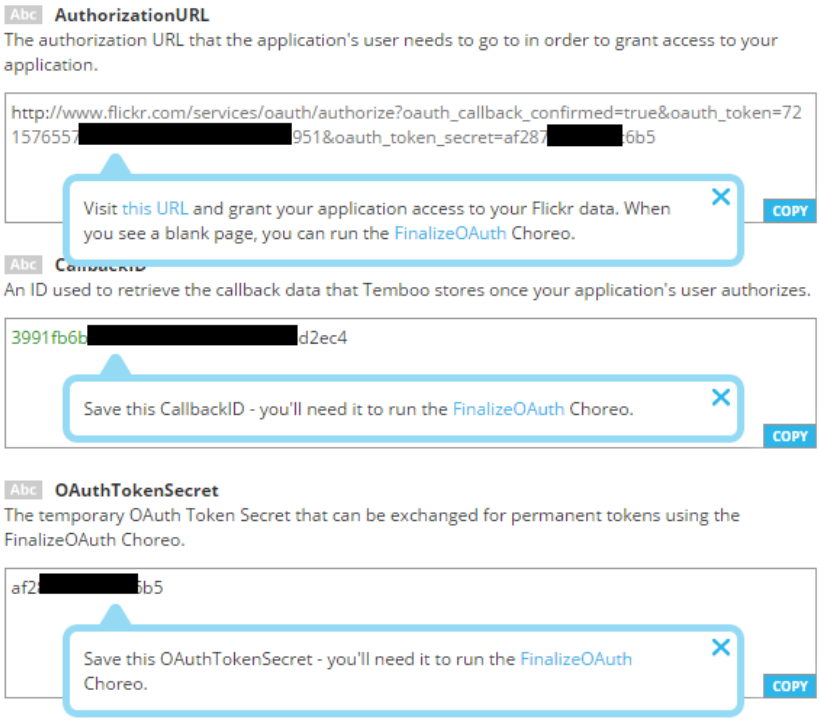


Initialize OAuth for Flicker



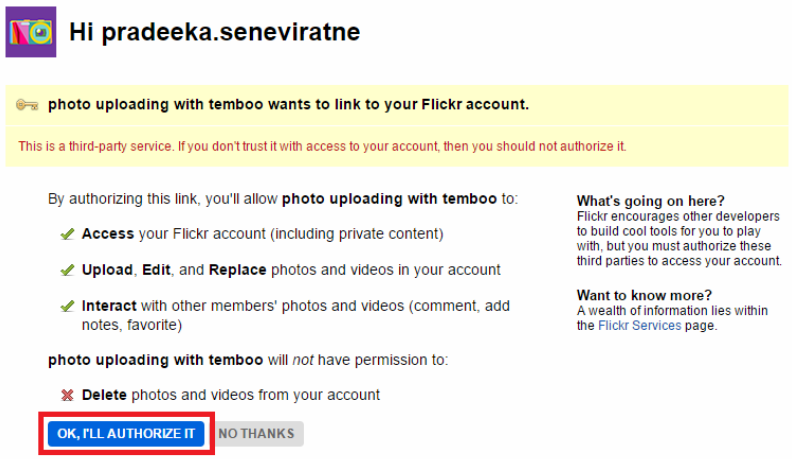
Tell us about your shield dialog box





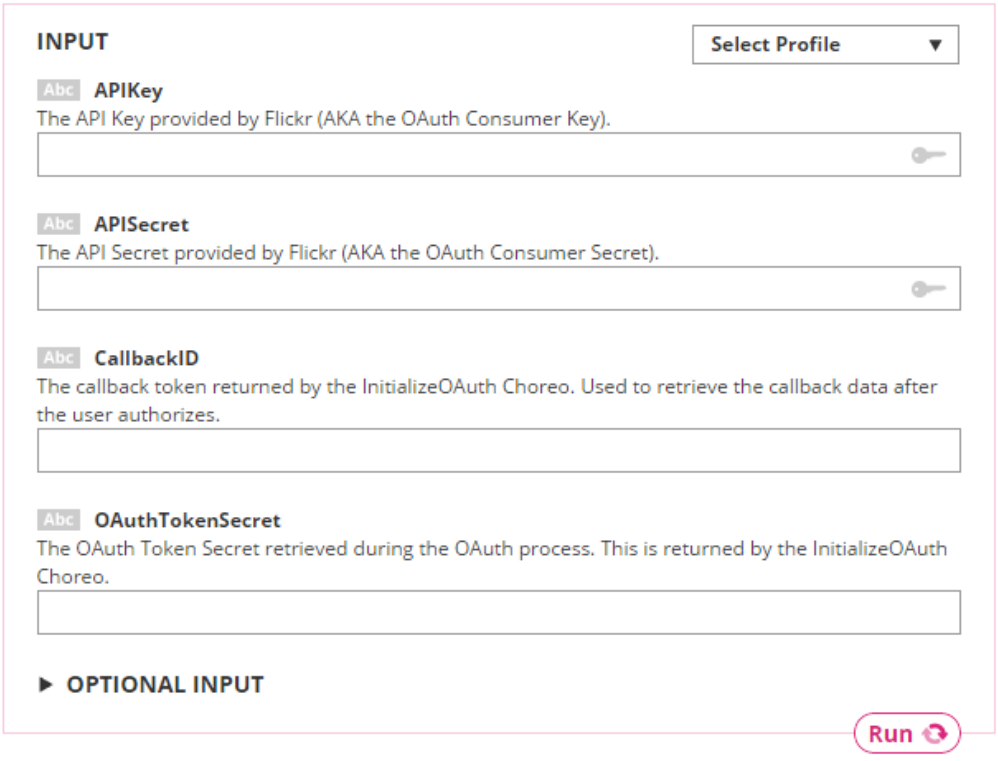
Output after the process of OAuth Initialization for Flickr

:



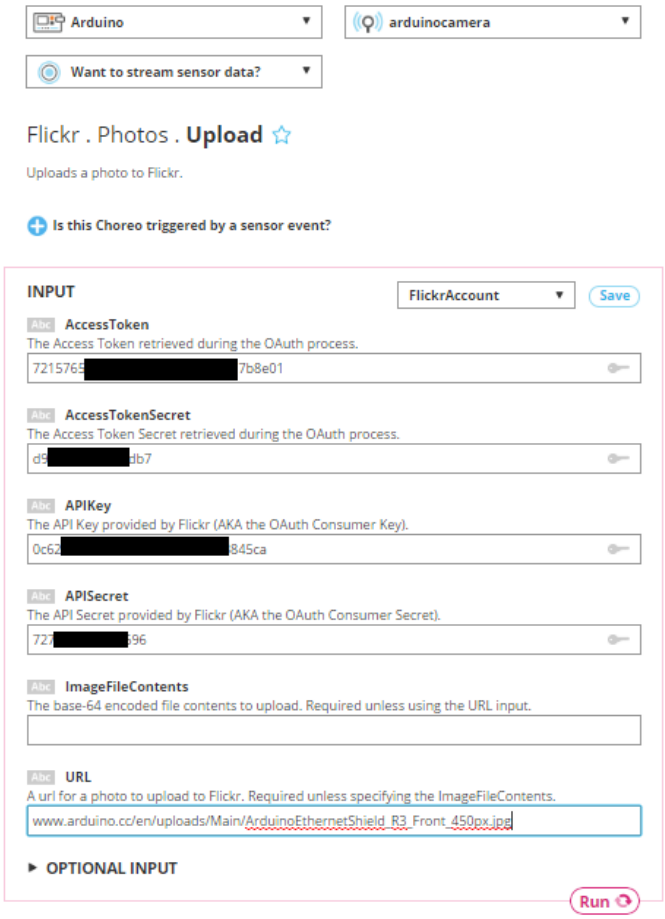
Flicker user account authorization page

Finalizing OAuth

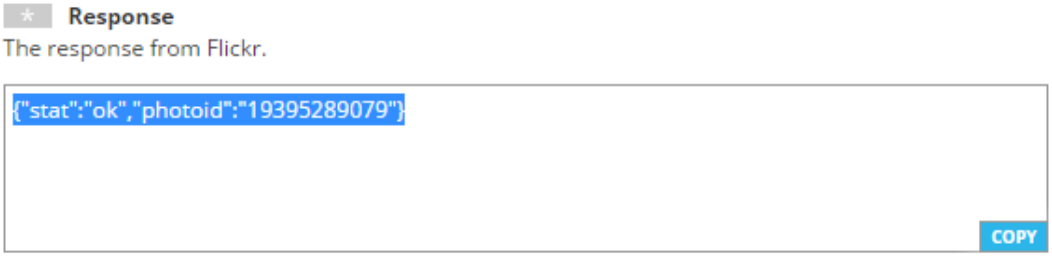


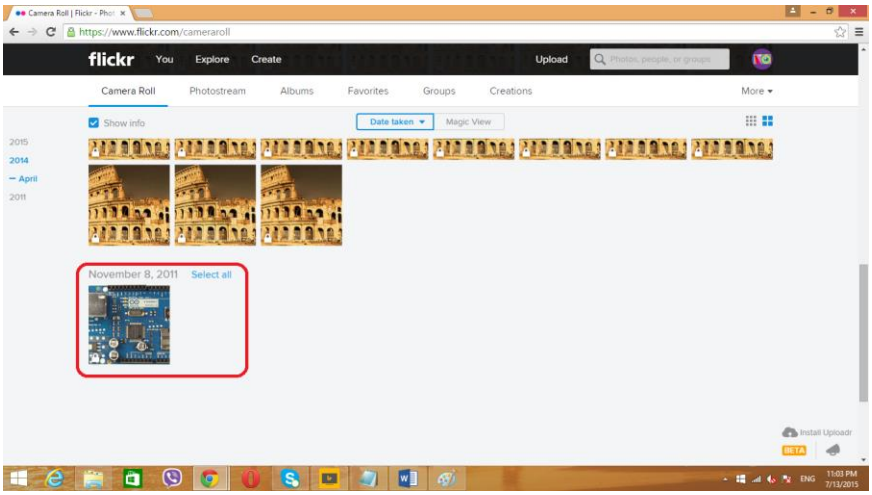
FinalizeOAuth for Flickr

Generating the photo upload sketch



Flicker photo upload Choreo





▼ CODE Download

```
/* Setup shield-specific #include statements */
#include <SPI.h>
#include <Dhcp.h>
#include <Dns.h>
#include <Ethernet.h>
#include <EthernetClient.h>
#include <Temboo.h>
#include "TembooAccount.h" // Contains Temboo account information

byte ethernetMACAddress[] = ETHERNET_SHIELD_MAC;
EthernetClient client;

int numRuns = 1;   // Execution count, so this doesn't run forever
int maxRuns = 10;   // Maximum number of times the Choreo should be executed

void setup() {
  Serial.begin(9600);
```

COPY

▼ HEADER FILE

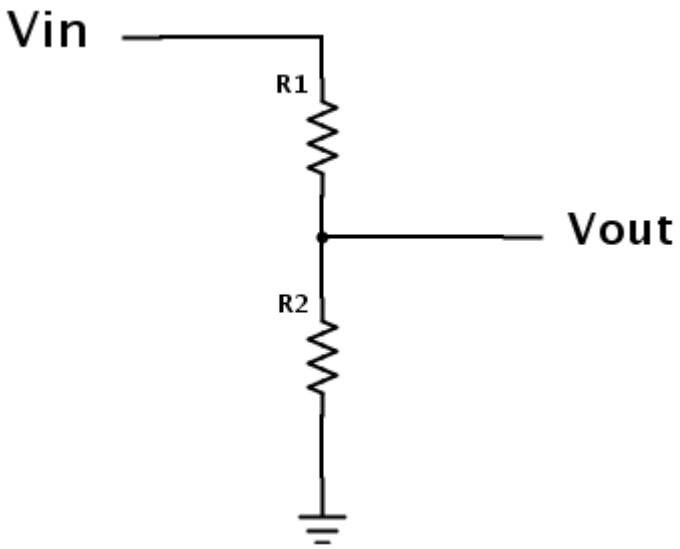
```
/*
IMPORTANT NOTE about TembooAccount.h

TembooAccount.h contains your Temboo account information and must be included
alongside your sketch. To do so, make a new tab in Arduino, call it TembooAccount.h,
and copy this content into it.
*/

#define TEMBOO_ACCOUNT "pradeeka"  // Your Temboo account name
#define TEMBOO_APP_KEY_NAME "myFirstApp"  // Your Temboo app key name
#define TEMBOO_APP_KEY "49ad3[redacted]1a03"  // Your Temboo app key

#define ETHERNET_SHIELD_MAC {[redacted]}

/*
The same TembooAccount.h file settings can be used for all Temboo SDK sketches.
Keeping your account information in a separate file means you can share the
main .ino file without worrying that you forgot to delete your credentials.
```

COPY

Chapter 5: Solar Panel Voltage Logging with NearBus Cloud Connector and Xively

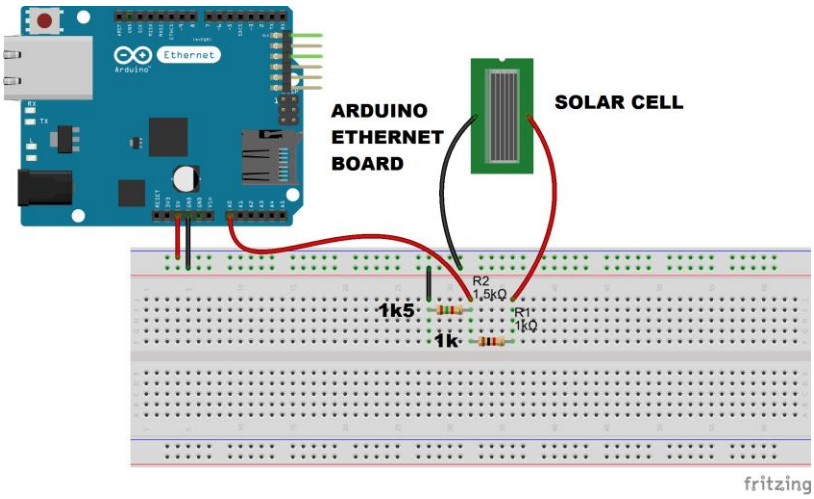
Connecting a solar cell with the Arduino Ethernet board

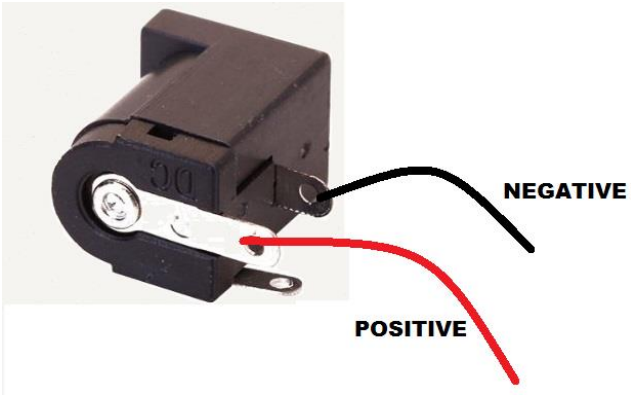
Building a voltage divider



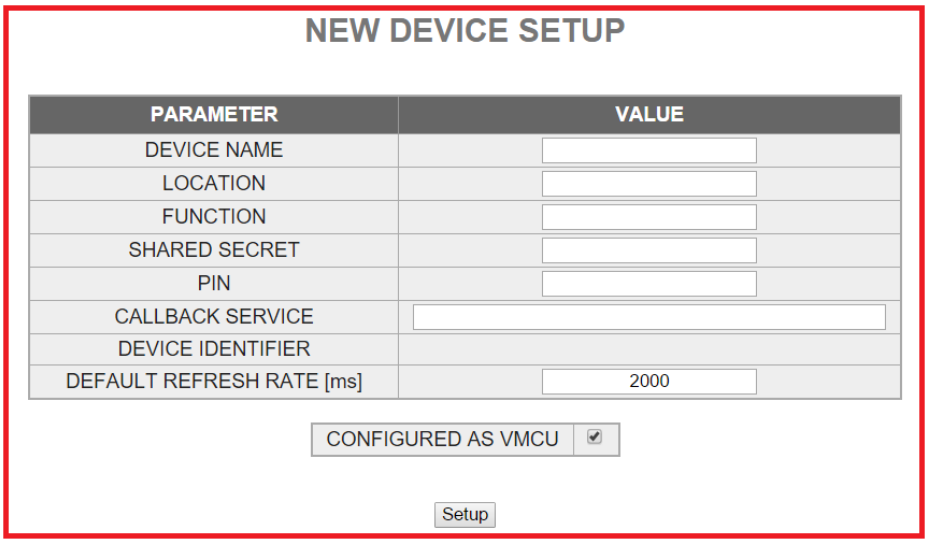
$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

Building the circuit with Arduino





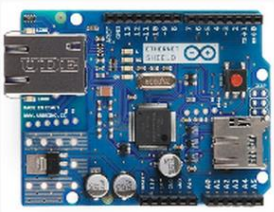
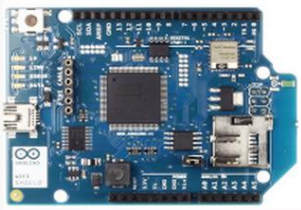
Defining a new device

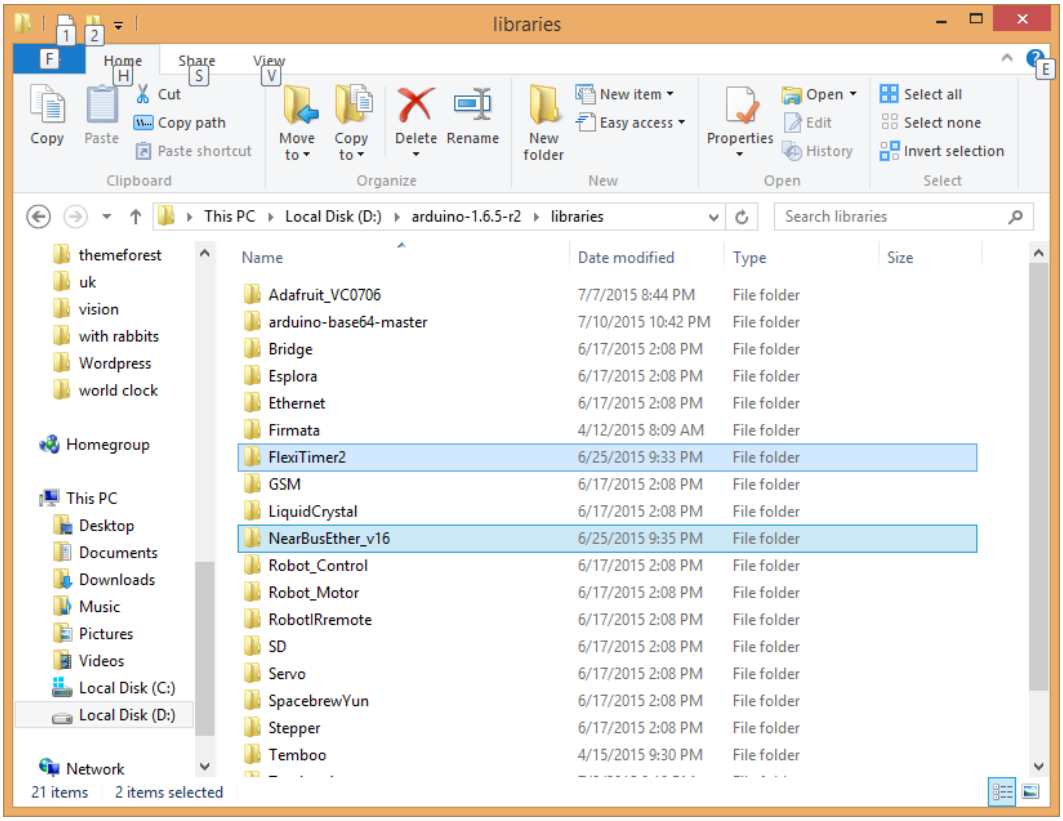


Examining the device lists

Downloading the NearBus agent

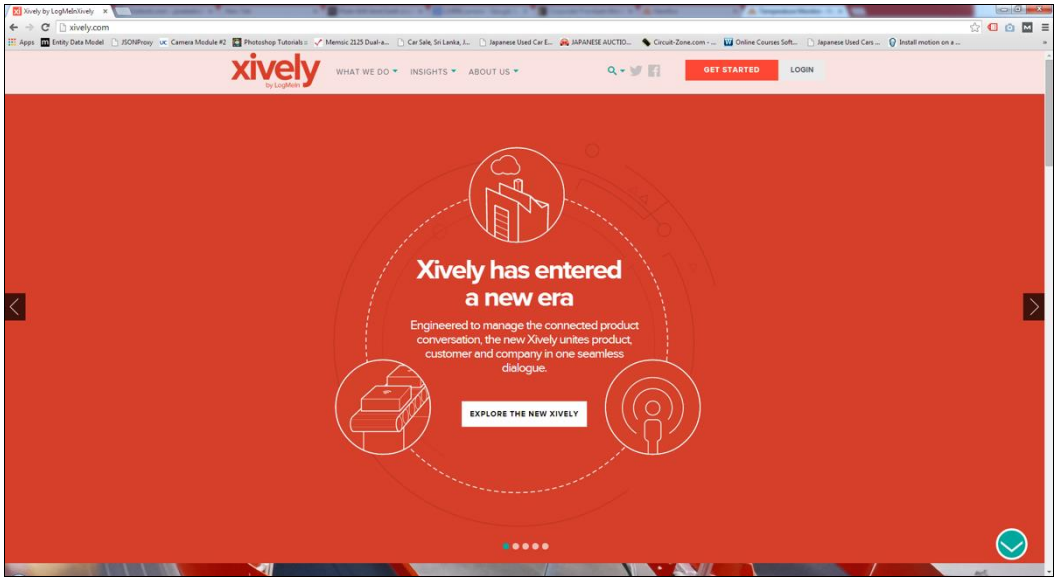
NearBus Library for Arduino - Alpha Release:

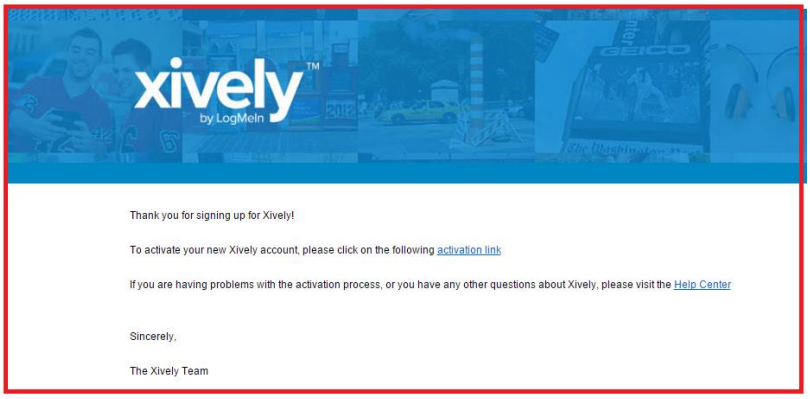
Arduino library for Ethernet - Alpha Release - (this release requires the FlexiTimer2 library installation)	NearBusEther_v16.zip	
Arduino library for WiFi - Alpha Release - (this release requires the FlexiTimer2 library installation) - IMPORTANT: It DOES NOT work with Arduino IDE 1.5.5 (Beta)	NearBusWiFi_v15.zip	

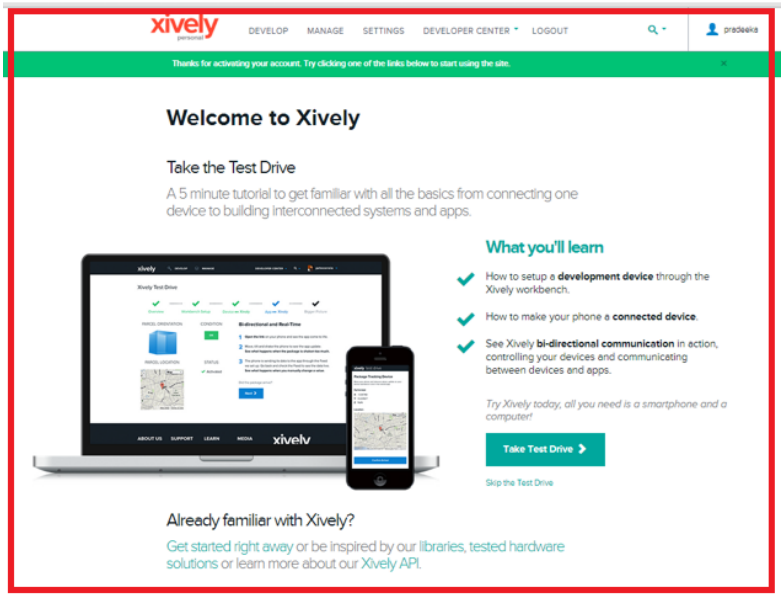

libraries
Home
Share
View
Copy
Paste
Cut
Copy path
Paste shortcut
Move to
Copy to
Delete
Rename
New folder
New item
Easy access
Properties
Open
Edit
History
Select all
Select none
Invert selection
Clipboard
Organize
New
Open
Select
This PC ▸ Local Disk (D:) ▸ arduino-1.6.5-r2 ▸ libraries
Search libraries
themeforest
uk
vision
with rabbits
Wordpress
world clock
Homegroup
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (D:)
Network
Name
Date modified
Type
Size
Adafruit_VC0706 7/7/2015 8:44 PM File folder
arduino-base64-master 7/10/2015 10:42 PM File folder
Bridge 6/17/2015 2:08 PM File folder
Esplora 6/17/2015 2:08 PM File folder
Ethernet 6/17/2015 2:08 PM File folder
Firmata 4/12/2015 8:09 AM File folder
FlexiTimer2 6/25/2015 9:33 PM File folder
GSM 6/17/2015 2:08 PM File folder
LiquidCrystal 6/17/2015 2:08 PM File folder
NearBusEther_v16 6/25/2015 9:35 PM File folder
Robot_Control 6/17/2015 2:08 PM File folder
Robot_Motor 6/17/2015 2:08 PM File folder
RobotIRremote 6/17/2015 2:08 PM File folder
SD 6/17/2015 2:08 PM File folder
Servo 6/17/2015 2:08 PM File folder
SpacebrewYun 6/17/2015 2:08 PM File folder
Stepper 6/17/2015 2:08 PM File folder
Temboo 4/15/2015 9:30 PM File folder
21 items
2 items selected

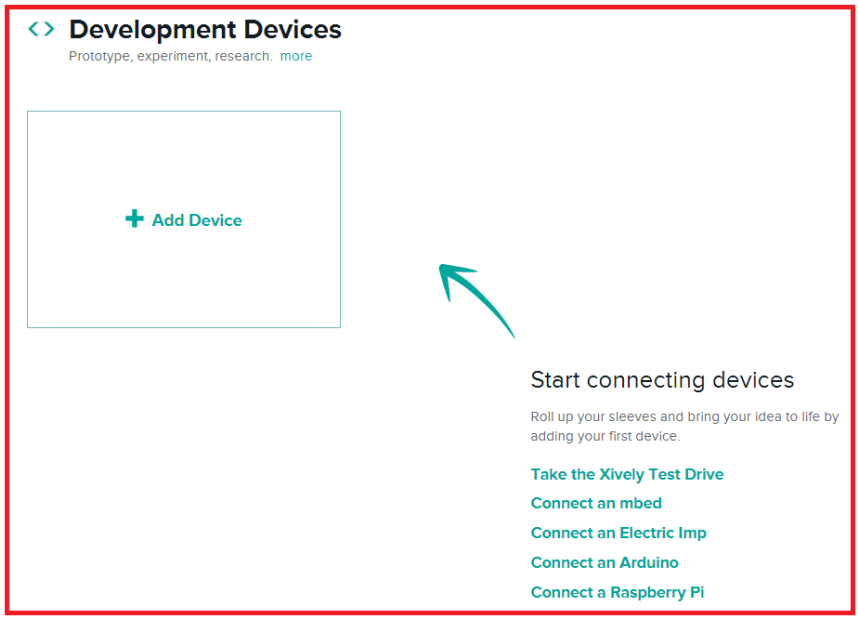
DEVICES LIST
DEVICE ID
DEVICE ALIAS
STATE
SELECT
NB101706
Arduino Ethernet
UP
CONFIG DEVICE
Setup

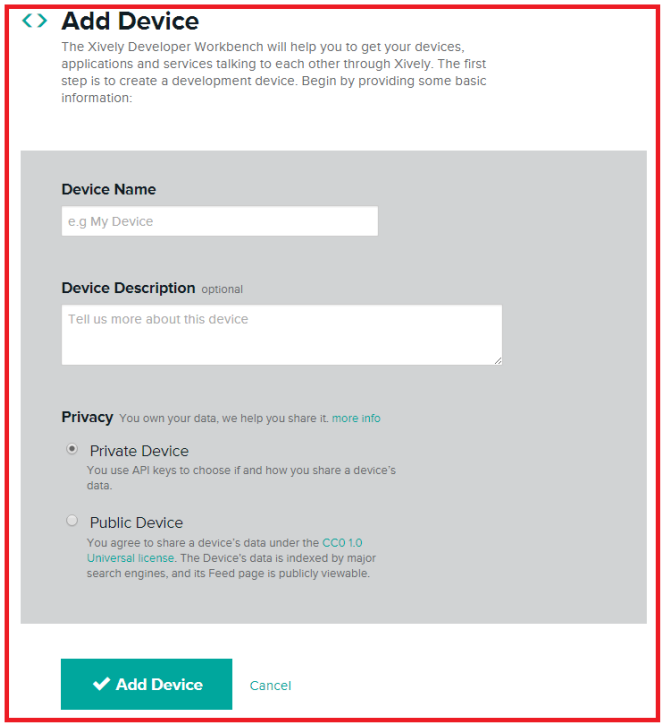
Creating and configuring a Xively account

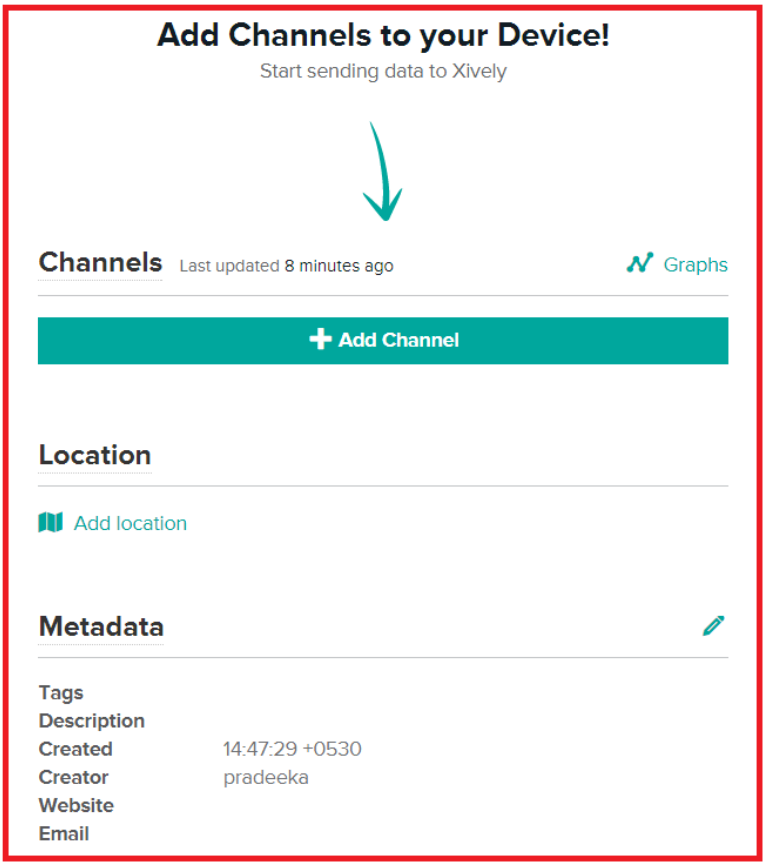


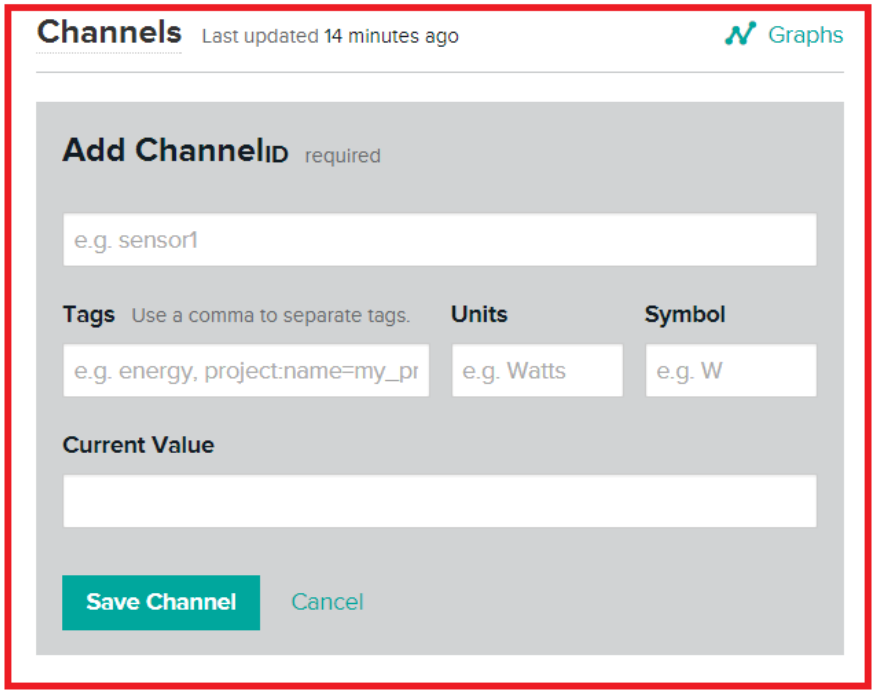


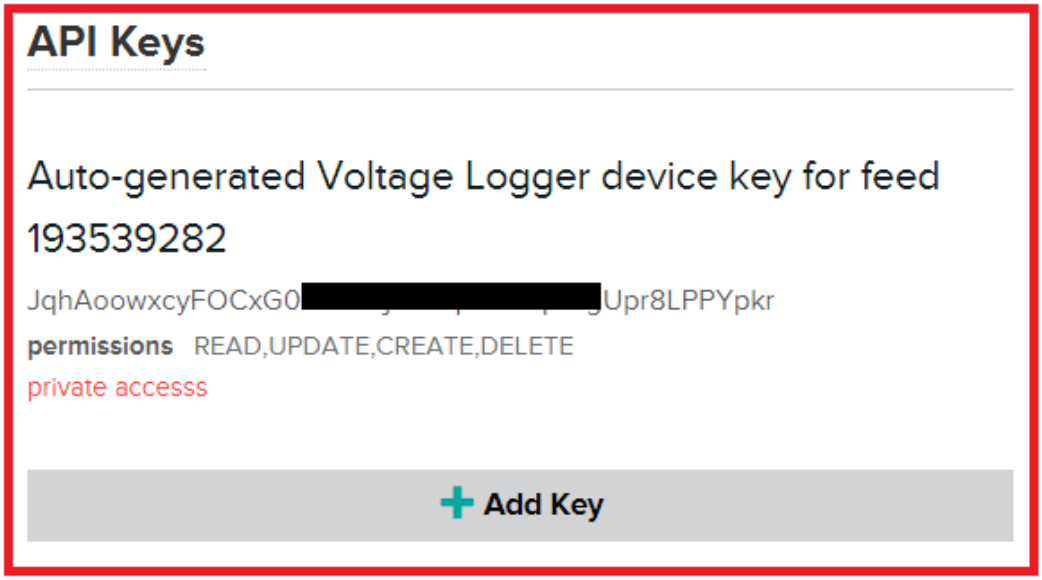

xively
DEVELOP
MANAGE
SETTINGS
DEVELOPER CENTER
LOGOUT
pradeeks
Thanks for activating your account. Try clicking one of the links below to start using the site.
Welcome to Xively
Take the Test Drive
A 5 minute tutorial to get familiar with all the basics from connecting one device to building interconnected systems and apps.
What you'll learn
How to setup a development device through the Xively workbench.
How to make your phone a connected device.
See Xively bi-directional communication in action, controlling your devices and communicating between devices and apps.
Try Xively today, all you need is a smartphone and a computer!
Take Test Drive
Skip the Test Drive
Already familiar with Xively?
Get started right away or be inspired by our libraries, tested hardware solutions or learn more about our Xively API.


Development Devices
Prototype, experiment, research. more
Add Device
Start connecting devices
Roll up your sleeves and bring your idea to life by adding your first device.
Take the Xively Test Drive
Connect an mbed
Connect an Electric Imp
Connect an Arduino
Connect a Raspberry Pi

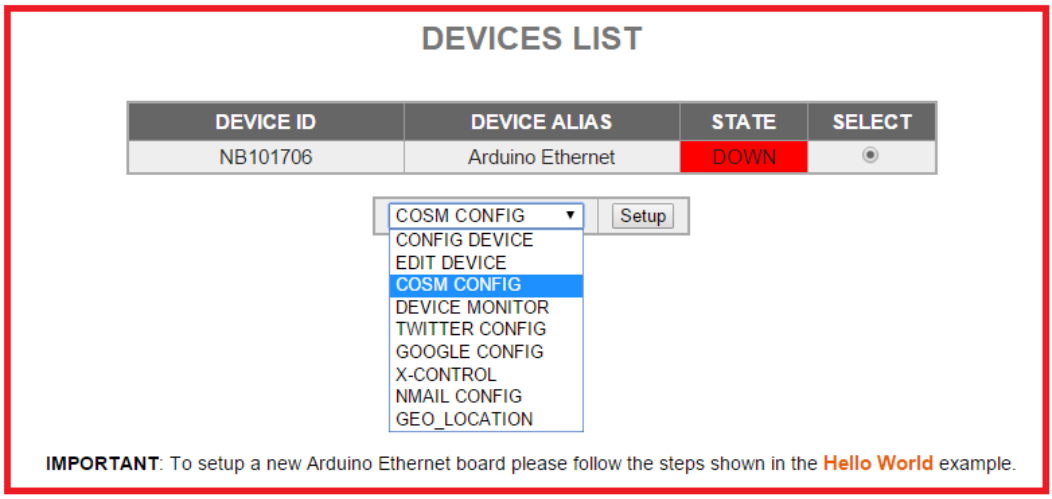

Add Device
The Xively Developer Workbench will help you to get your devices, applications and services talking to each other through Xively. The first step is to create a development device. Begin by providing some basic information:
Device Name
e.g My Device
Device Description optional
Tell us more about this device
Privacy You own your data, we help you share it. more info
Private Device
You use API keys to choose if and how you share a device's data.
Public Device
You agree to share a device's data under the CC0 1.0 Universal license. The Device's data is indexed by major search engines, and its Feed page is publicly viewable.
Add Device
Cancel


Add Channels to your Device!
Start sending data to Xively
Channels
Last updated 8 minutes ago
Graphs
Add Channel
Location
Add location
Metadata
Tags
Description
Created
14:47:29 +0530
Creator
pradeeka
Website
Email


Channels
Last updated 14 minutes ago
Graphs
Add Channel ID required
e.g. sensor1
Tags Use a comma to separate tags.
Units
Symbol
e.g. energy, project:name=my_pr
e.g. Watts
e.g. W
Current Value
Save Channel
Cancel


API Keys
Auto-generated Voltage Logger device key for feed 193539282
JqhAoowxcyFOCxG0[redacted]Upr8LPPYpkr
permissions READ,UPDATE,CREATE,DELETE
private accesss
Add Key

Configuring the NearBus connected device for Xively



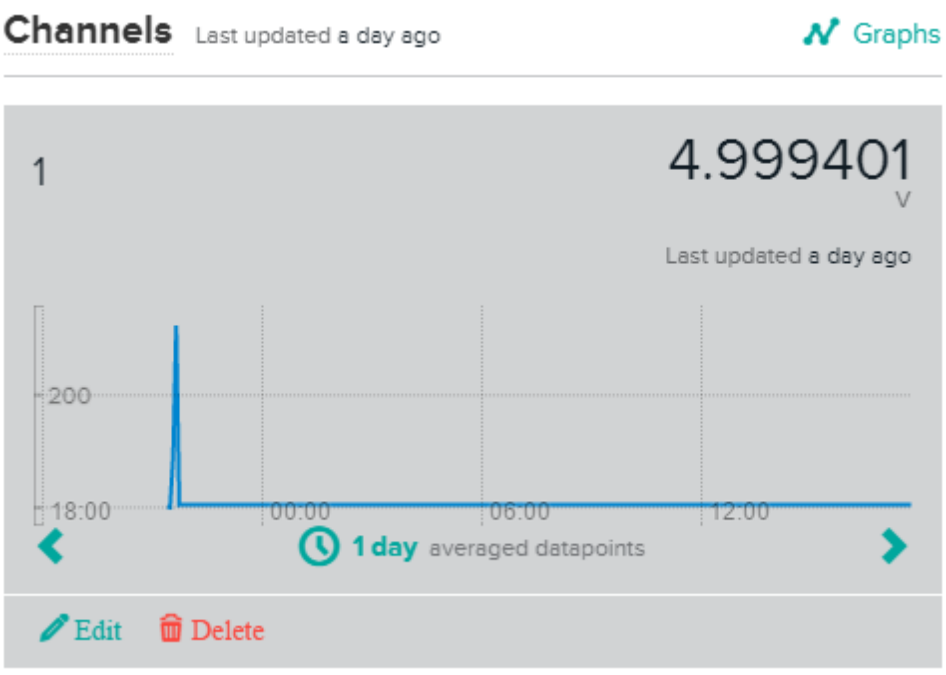
COSM CONNECTOR (xively.com)

Arduino Ethernet	DOWN
MODE	TRNSP

CHANNEL	STREAM ID	IN [A]	OUT [B]	Const [K]	Offset	ON
Channel 00	1	◉	○	0.004887	0	☑
Channel 01		○	○	1	0	☐
Channel 02		○	○	1	0	☐
Channel 03		○	○	1	0	☐
Channel 04		○	○	1	0	☐
Channel 05		○	○	1	0	☐
Channel 06		○	○	1	0	☐
Channel 07		○	○	1	0	☐

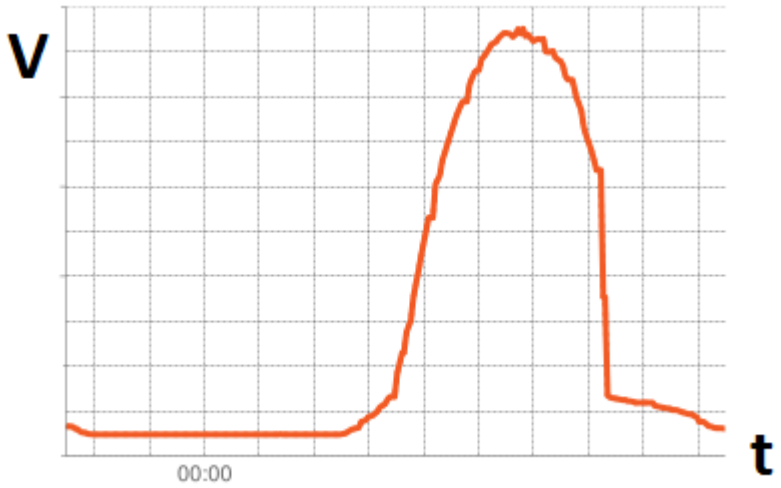
COSM FEED	1910481586
COSM API KEY	GE0sSoyHziZ3PZhtYKBP99ioKPWIHqb7adMU

Setup



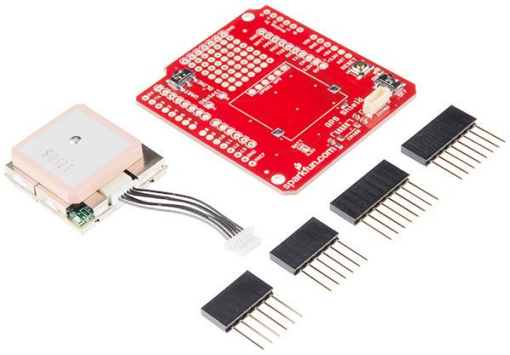
Developing a web page to display the real-time voltage values

Displaying data on a web page



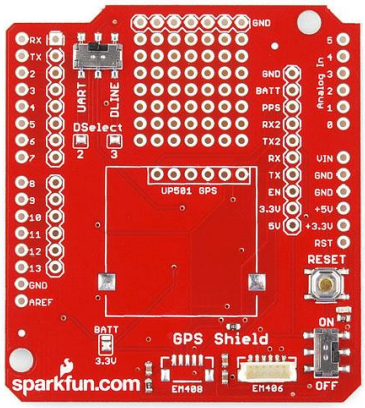
Chapter 6: GPS Location Tracker with Temboo, Twilio, and Google Maps

Getting started with Arduino GPS shield



Arduino GPS Shield Kit: Image taken from SparkFun Electronics

Connecting the Arduino GPS shield with the Arduino Ethernet board

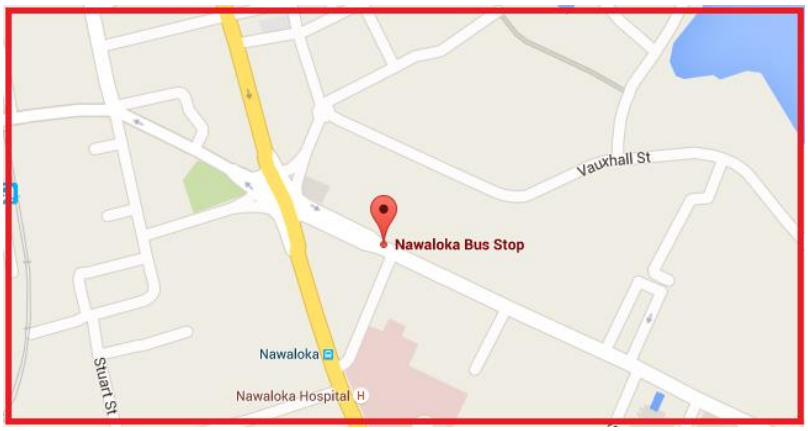


Arduino GPS Shield PCB: Image taken from SparkFun Electronics

Testing the GPS shield

```
Location: 42.120616,-88.330108  Date/Time: 2/24/2015 17:37:49.00
Location: 42.120616,-88.330108  Date/Time: 2/24/2015 17:37:50.00
Location: 42.120616,-88.330108  Date/Time: 2/24/2015 17:37:50.00
Location: 42.120616,-88.330108  Date/Time: 2/24/2015 17:37:50.00
Location: 42.120616,-88.330108  Date/Time: 2/24/2015 17:37:51.00
Location: 42.120616,-88.330108  Date/Time: 2/24/2015 17:37:51.00
Location: 42.120616,-88.330108  Date/Time: 2/24/2015 17:37:51.00
Location: 42.120616,-88.330108  Date/Time: 2/24/2015 17:37:51.00
```

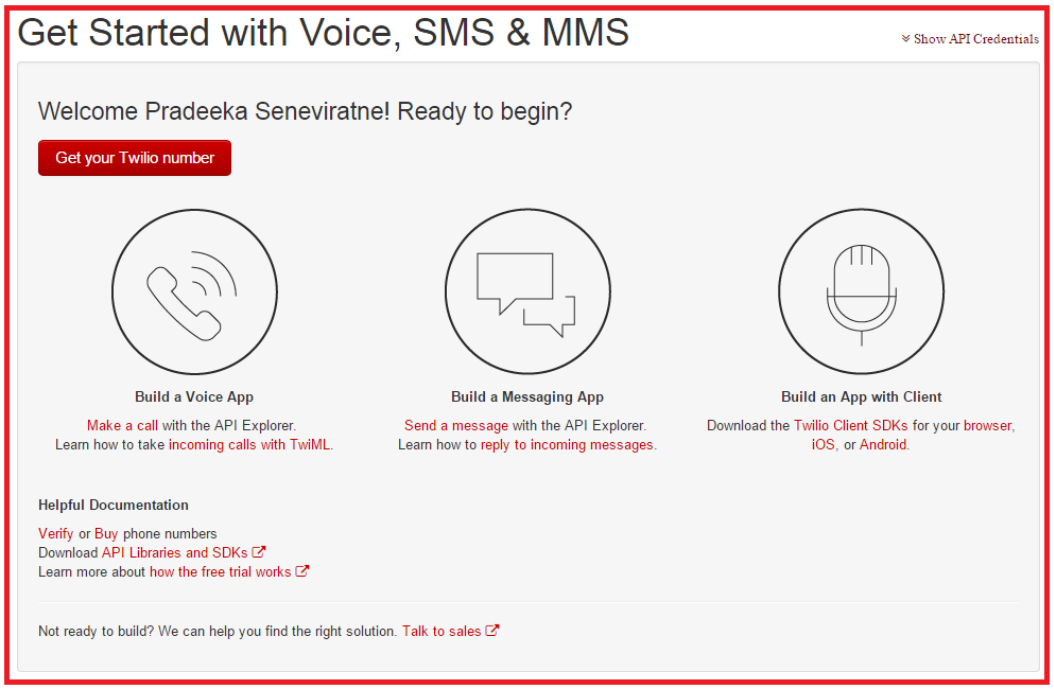
Display the current location on Google Maps



The current location of the Arduino GPS Shield is displayed on the Google Map with a marker icon

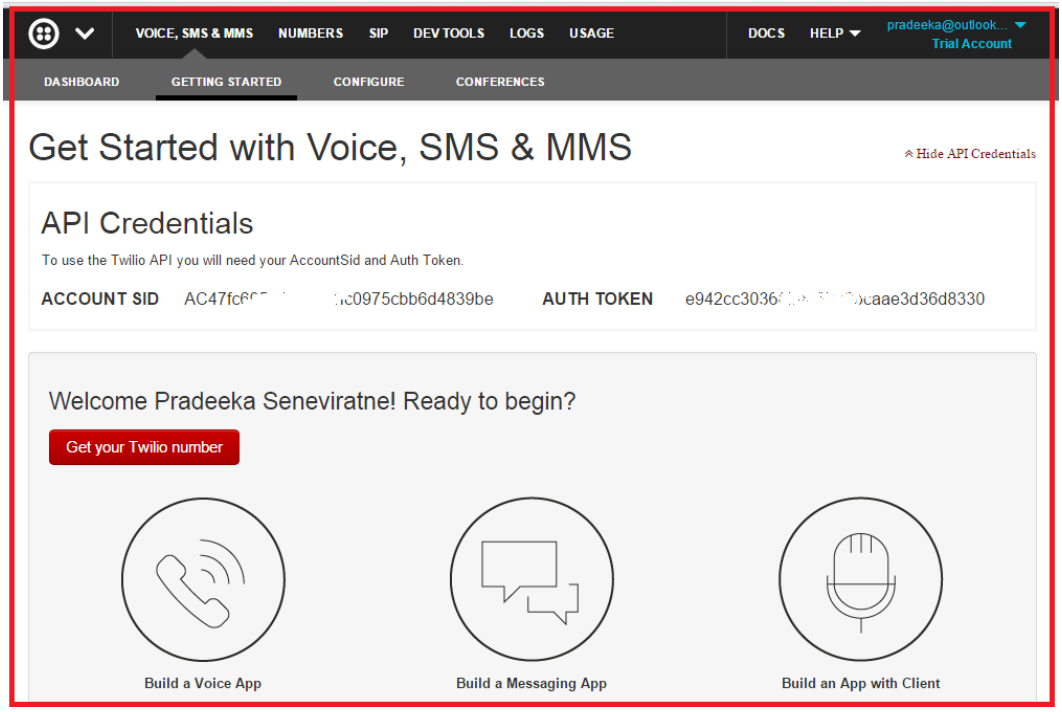
Getting started with Twilio

Creating a Twilio account



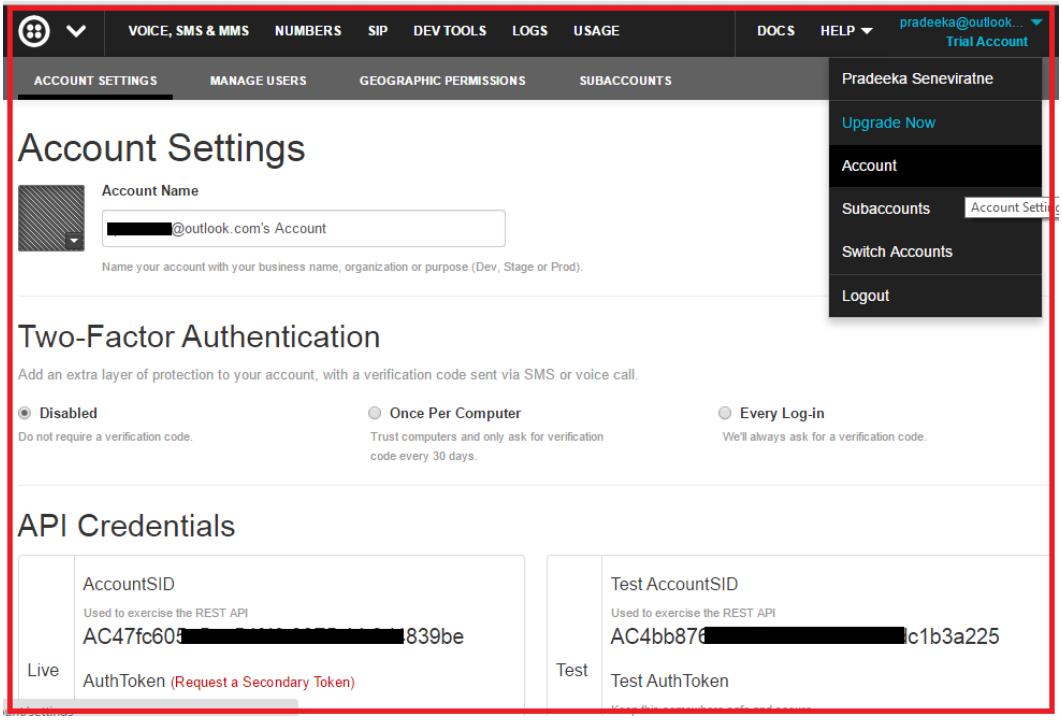
Twilio getting started page

Finding Twilio LIVE API credentials



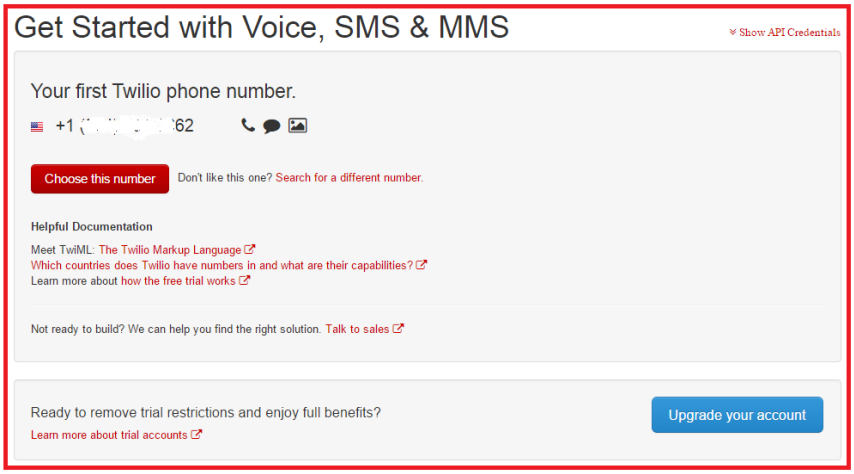
Twilio API Credentials: Account SID and Auth Token

Finding Twilio test API credentials

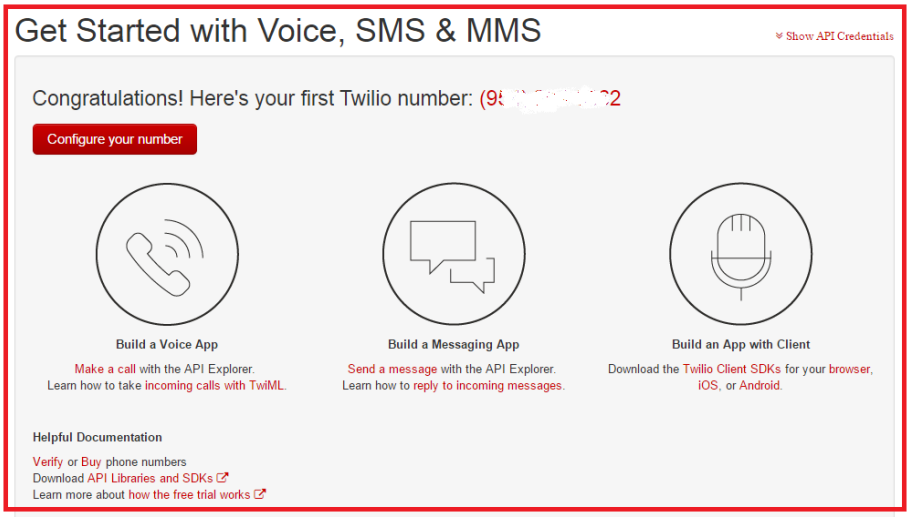


Twilio Test API Credentials: Test Account SID and Test Auth Token

Get your Twilio number



Twilio Phone number

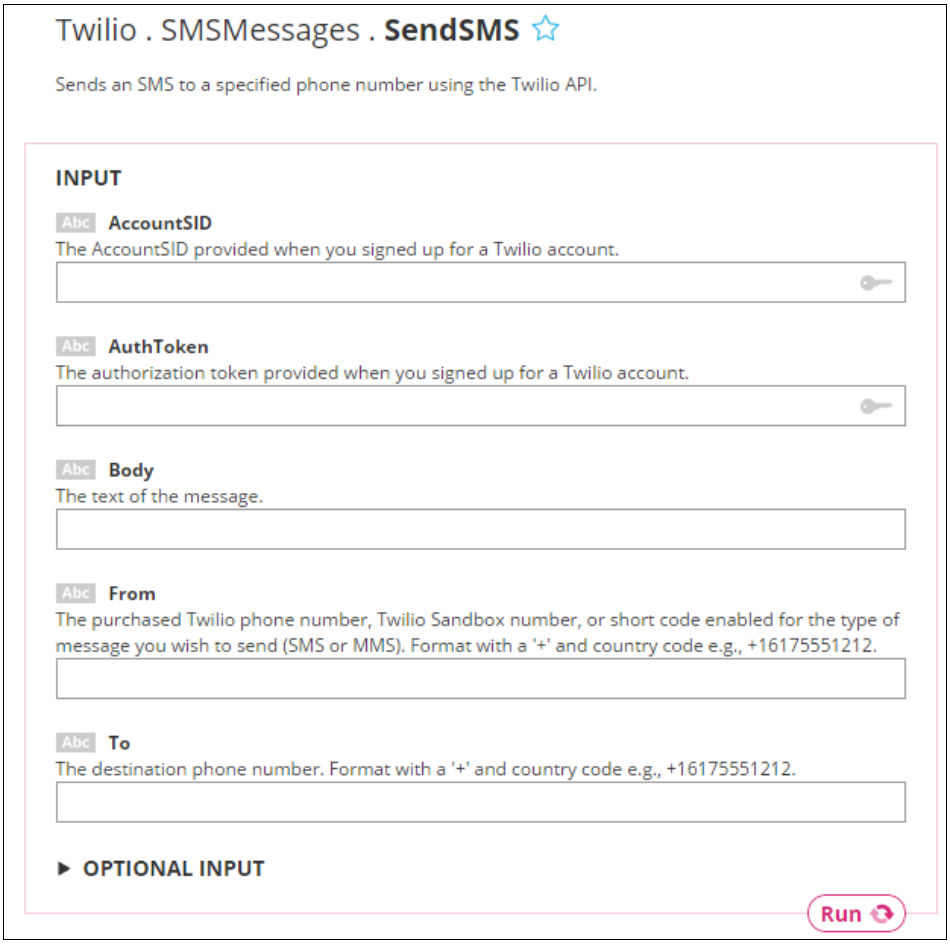


Twilio Phone number configuration page

Creating Twilio Choreo with Temboo

Sending an SMS with Twilio API

1.

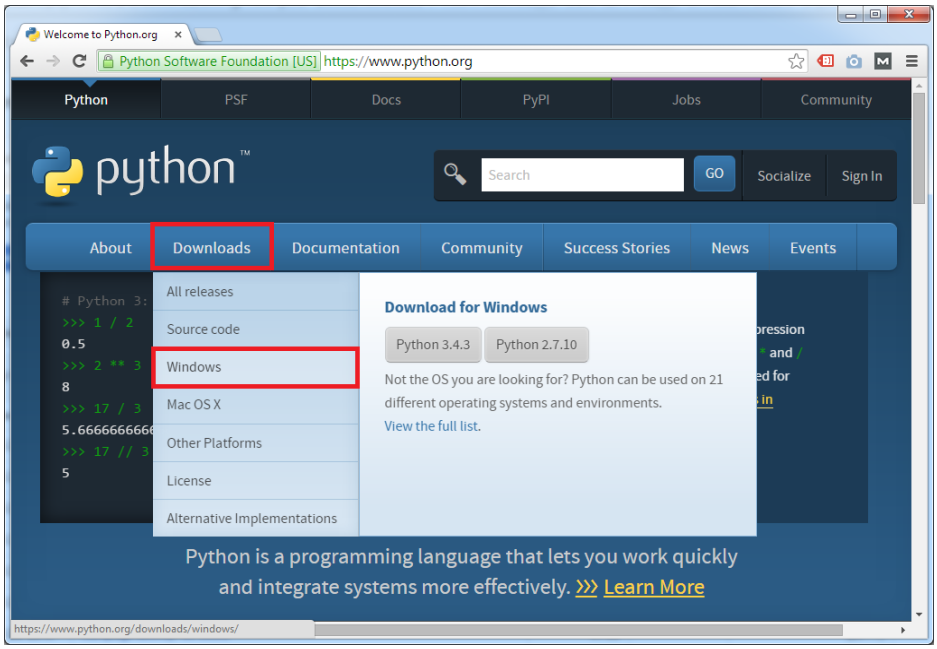


Twilio SendSMS form

Chapter 7: Tweet-a-Light – Twitter-Enabled Electric Light

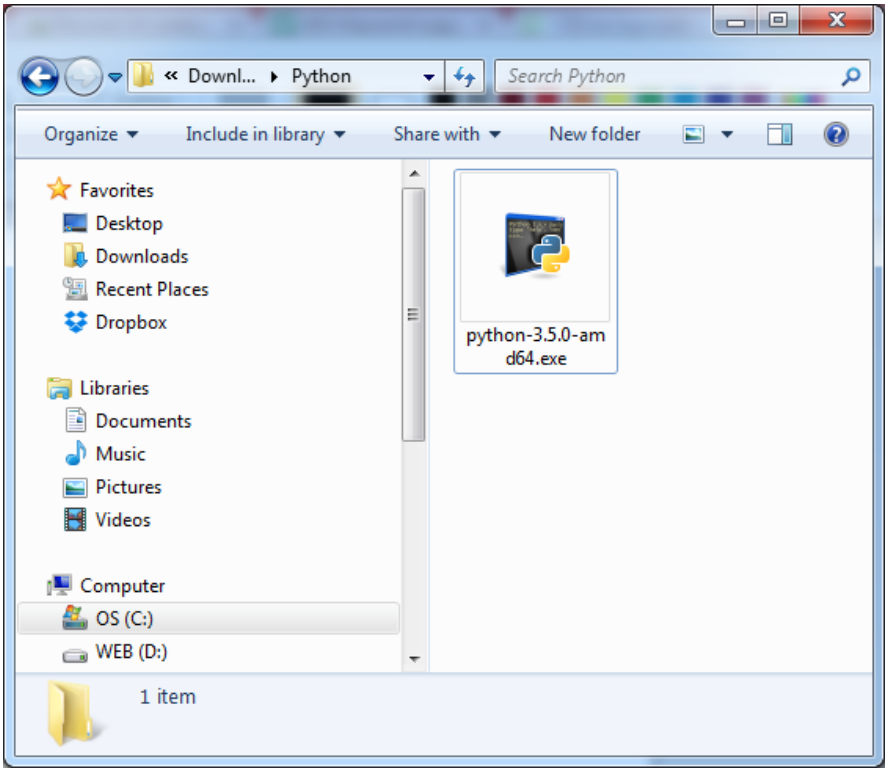
Getting started with Python

Installing Python on Windows

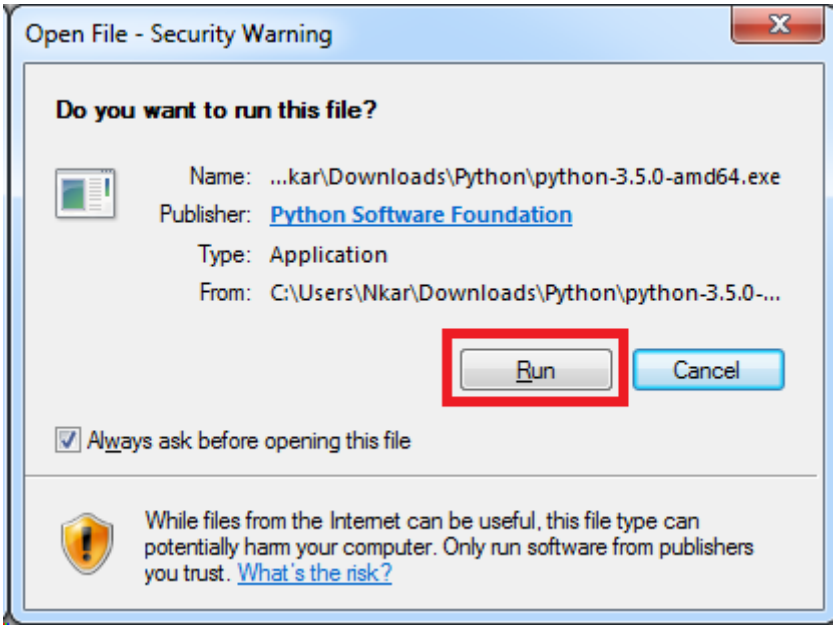


The Python home page

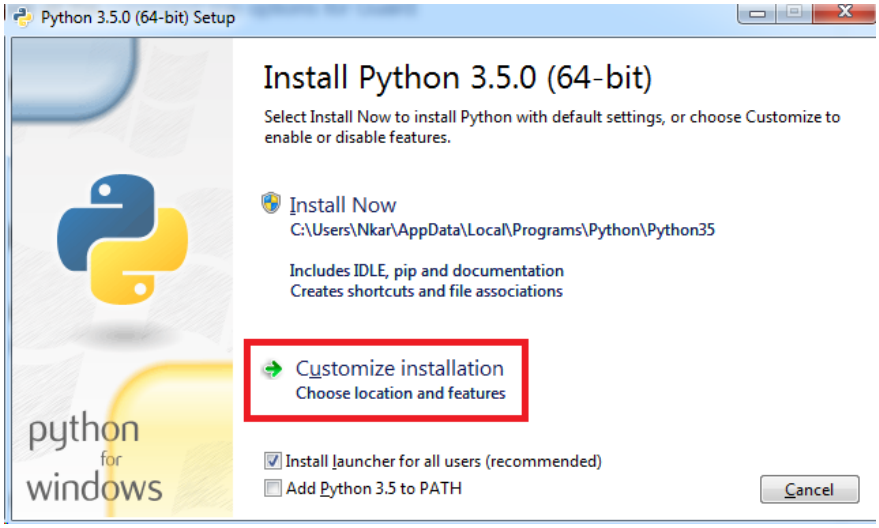
The Python download page



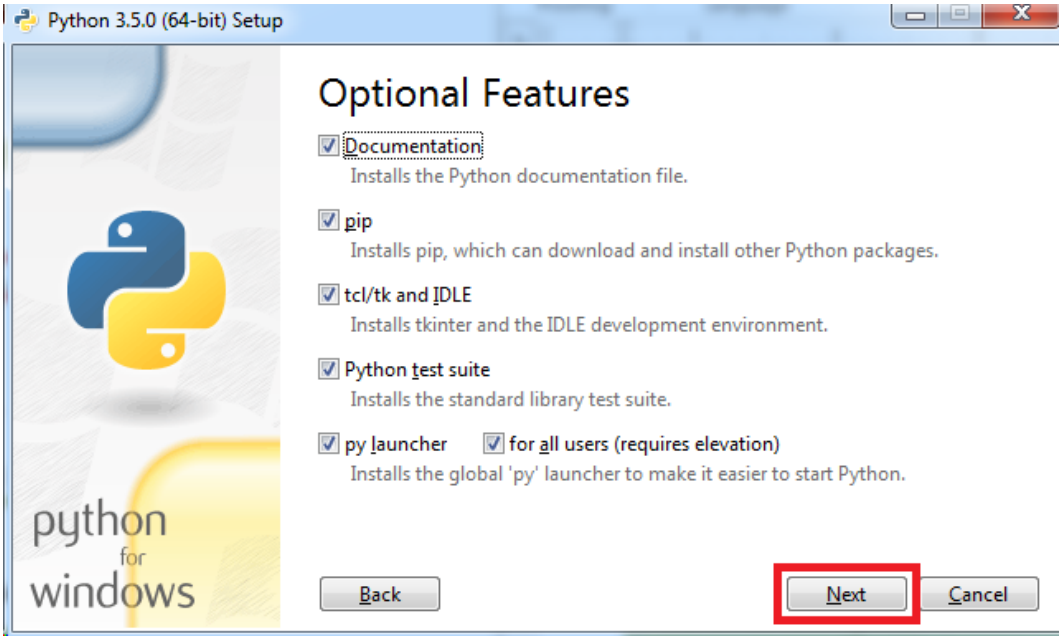
Python setup



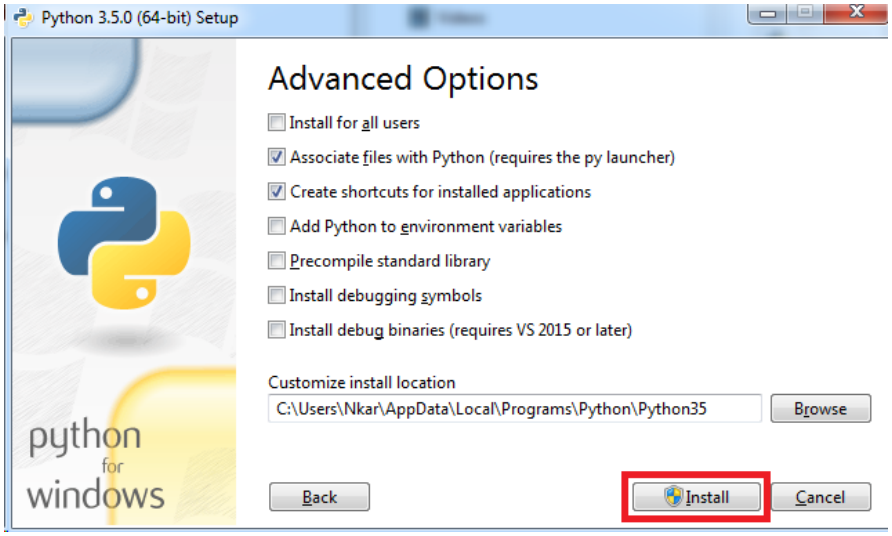
Security warning



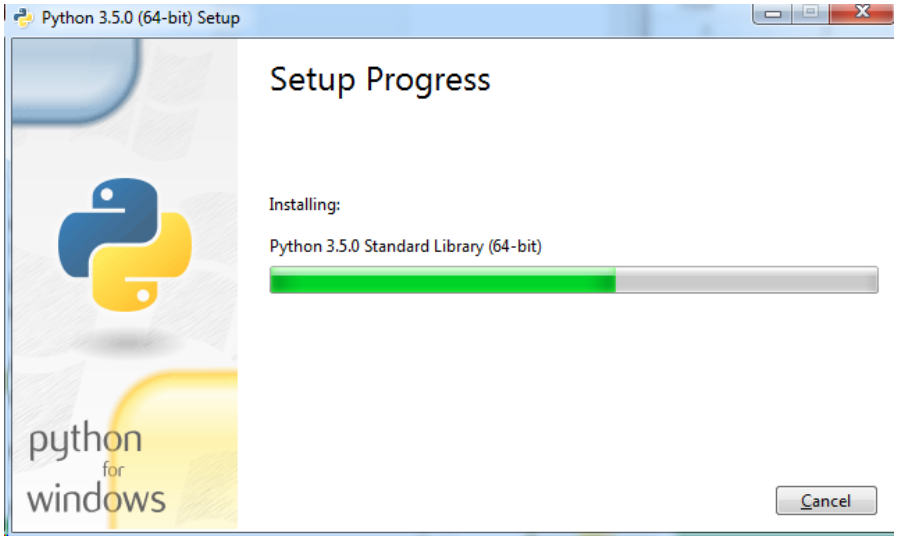
Python setup wizard—Start screen



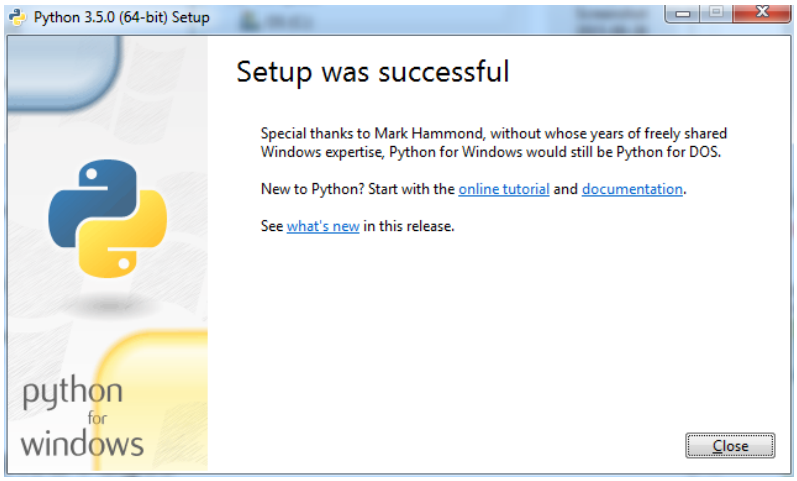
Python setup wizard—Optional Features



Python setup wizard—Advanced Options

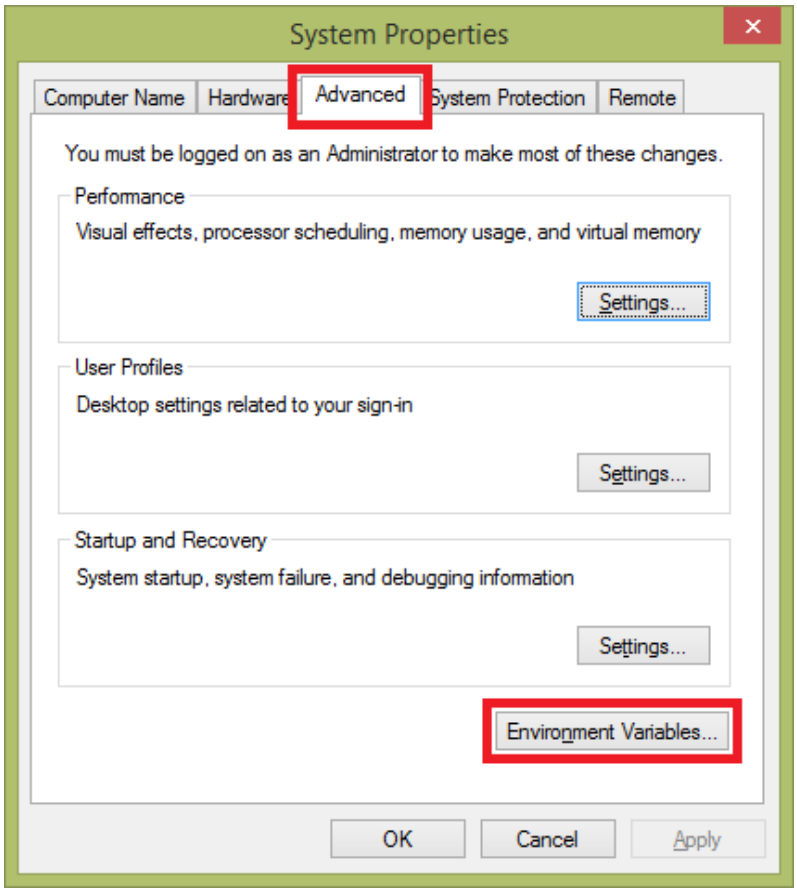


Python setup installation progress

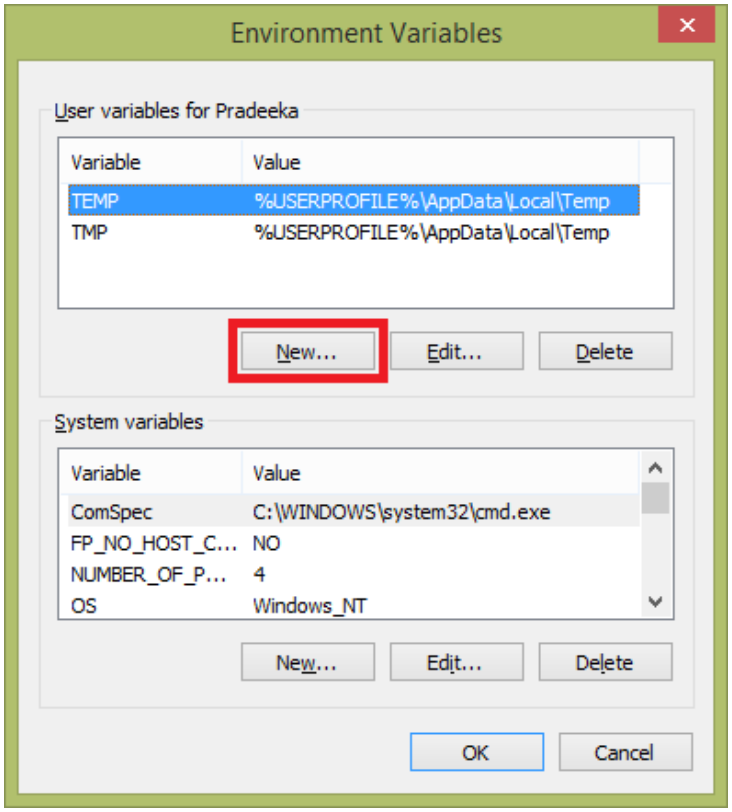


The Python setup is successful

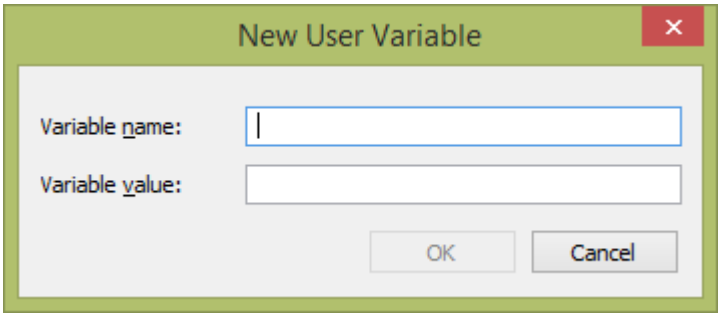
Setting environment variables for Python



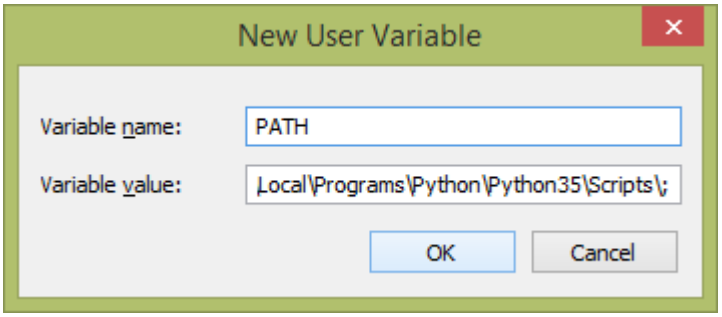
The System Properties dialog box



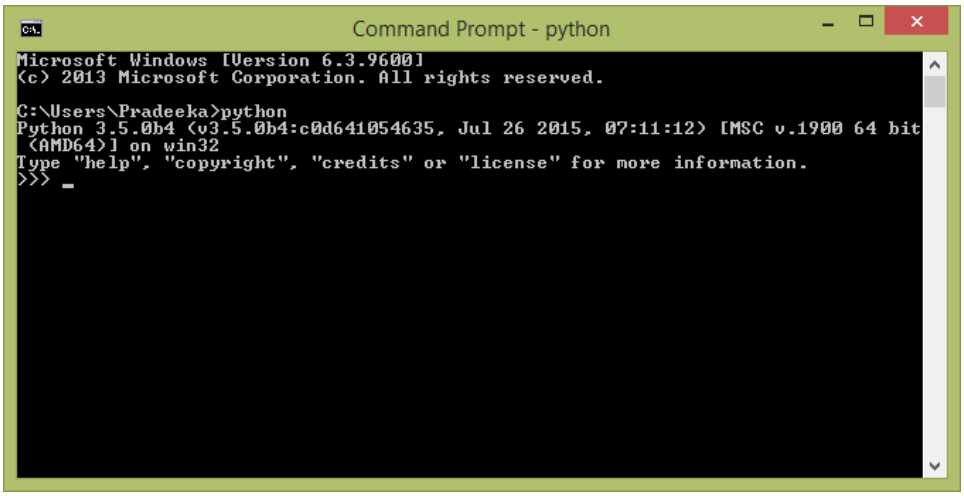
The Environment Variables dialog box



The New User Variable dialog box

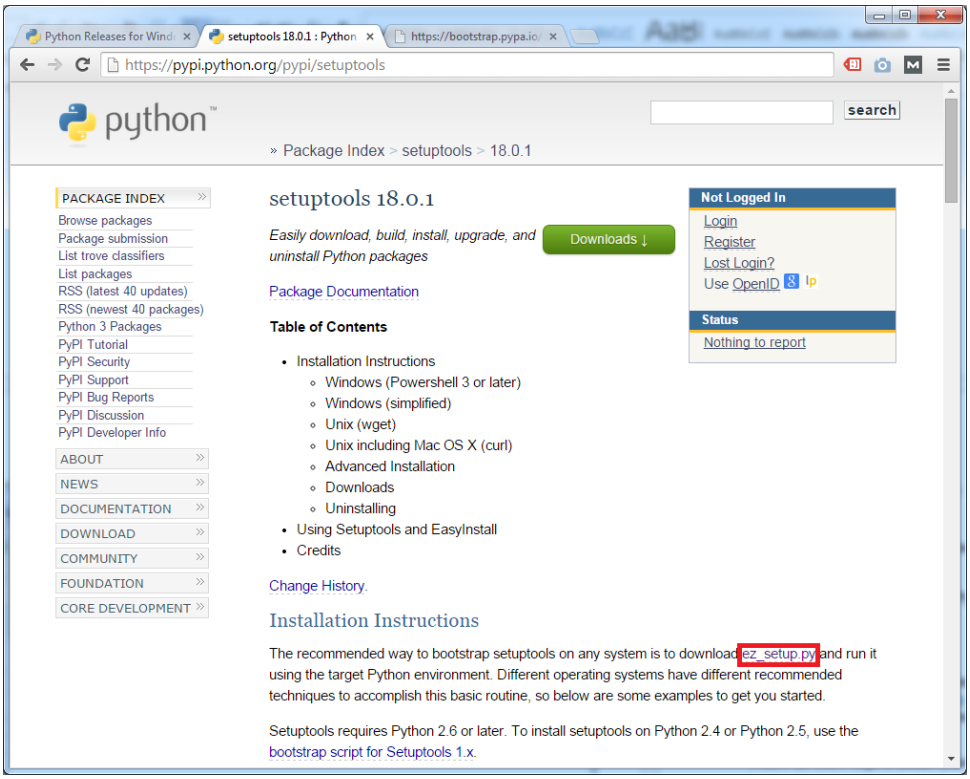


The New User Variable dialog box

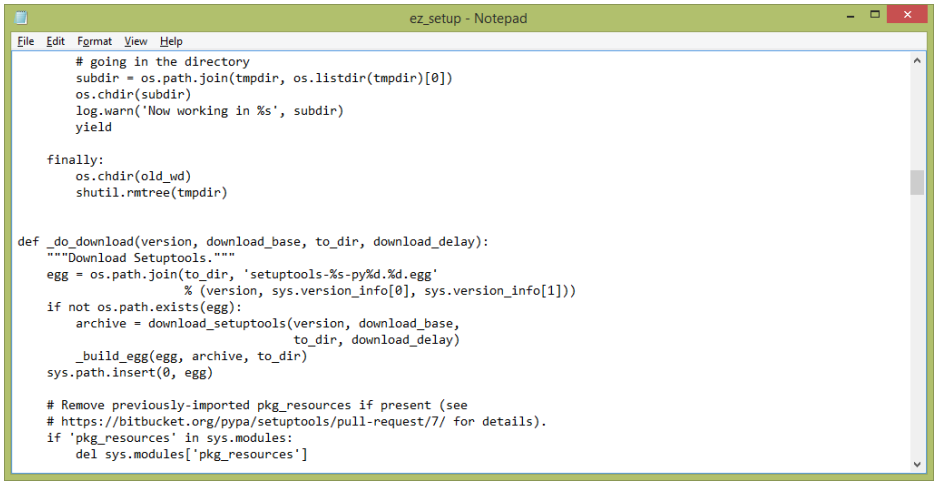


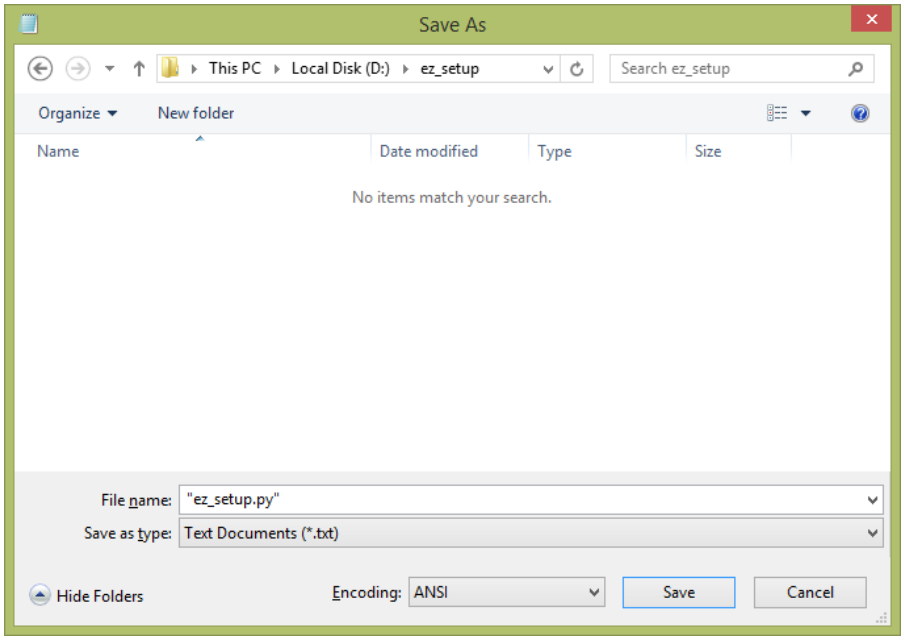
Python Command Prompt

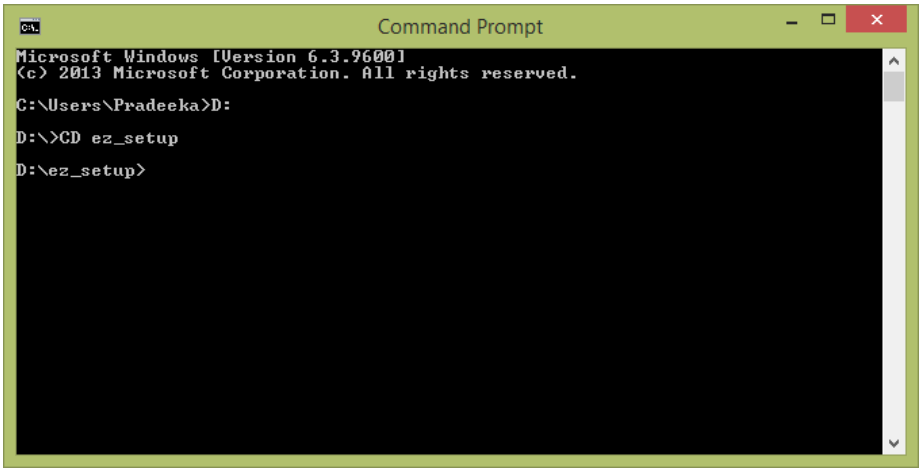
Installing the setuptools utility on Python

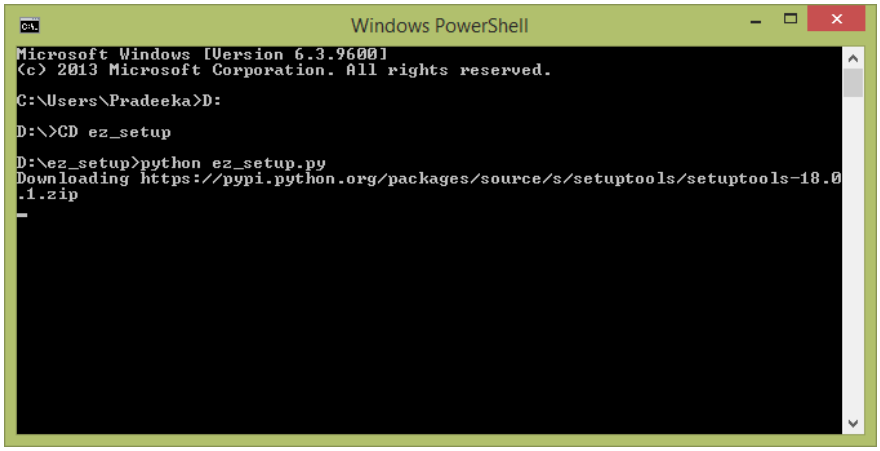


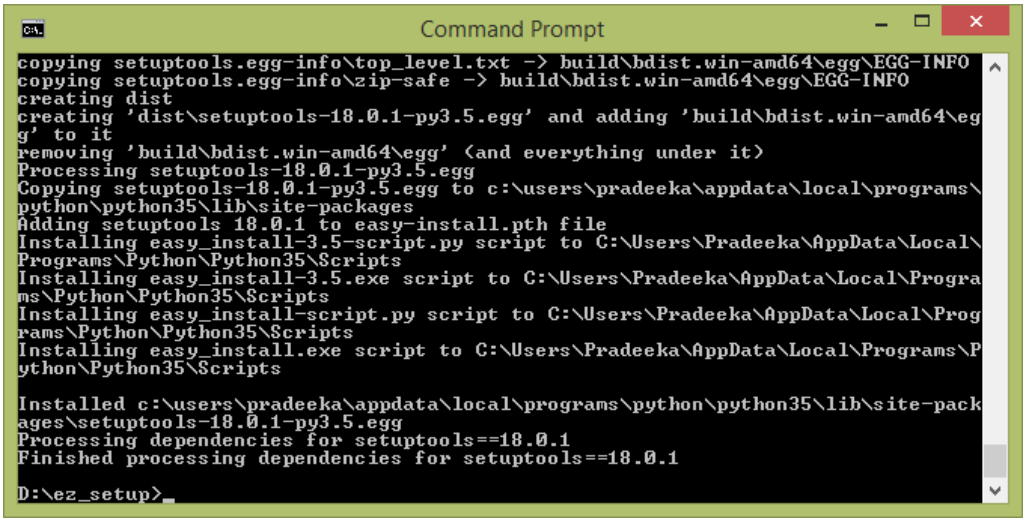
The Setuptools download page


ez_setup - Notepad
File Edit Format View Help
# going in the directory
subdir = os.path.join(tmpdir, os.listdir(tmpdir)[0])
os.chdir(subdir)
log.warn('Now working in %s', subdir)
yield
finally:
os.chdir(old_wd)
shutil.rmtree(tmpdir)
def _do_download(version, download_base, to_dir, download_delay):
"""Download Setuptools."""
egg = os.path.join(to_dir, 'setuptools-%s-py%d.%d.egg'
% (version, sys.version_info[0], sys.version_info[1]))
if not os.path.exists(egg):
archive = download_setuptools(version, download_base,
to_dir, download_delay)
_build_egg(egg, archive, to_dir)
sys.path.insert(0, egg)
# Remove previously-imported pkg_resources if present (see
# https://bitbucket.org/pypa/setuptools/pull-request/7/ for details).
if 'pkg_resources' in sys.modules:
del sys.modules['pkg_resources']

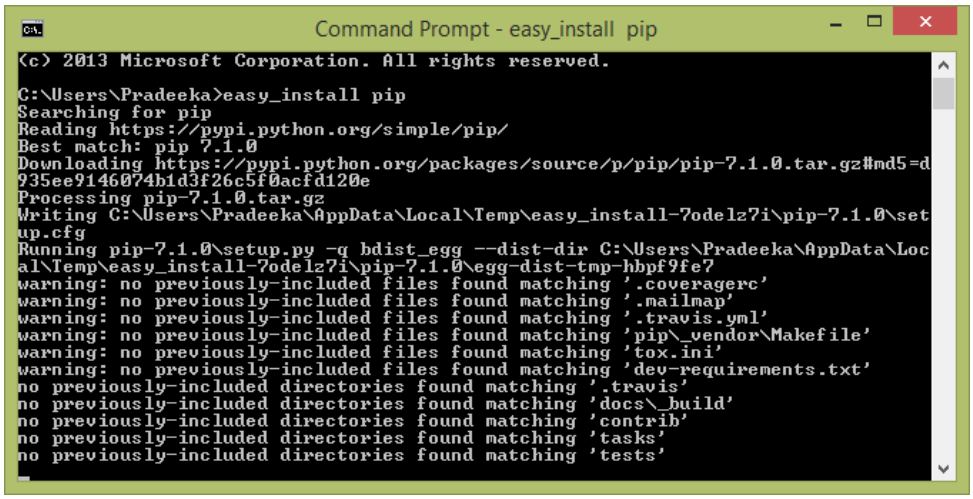

Save As
This PC
Local Disk (D:)
ez_setup
Search ez_setup
Organize
New folder
Name
Date modified
Type
Size
No items match your search.
File name:
"ez_setup.py"
Save as type:
Text Documents (*.txt)
Hide Folders
Encoding:
ANSI
Save
Cancel


Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\Pradeeka>D:
D:\>CD ez_setup
D:\ez_setup>

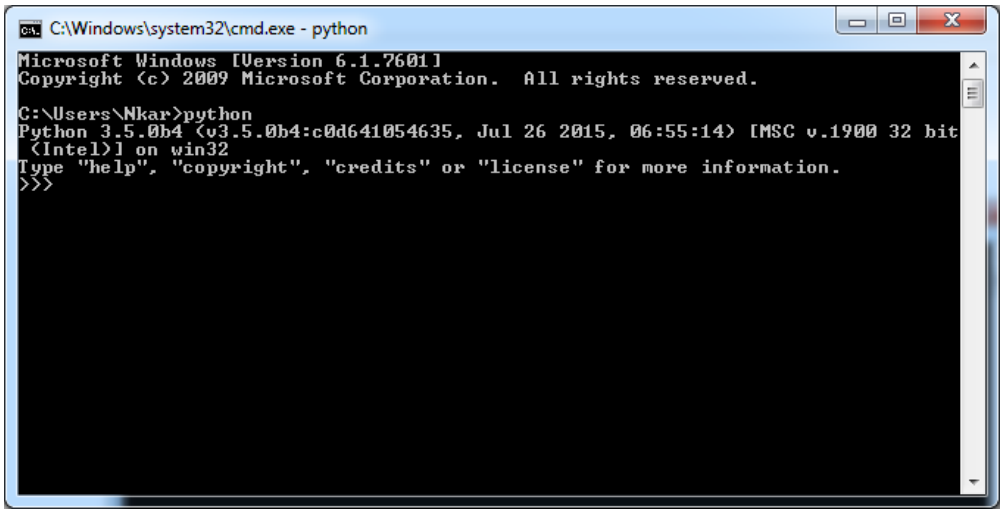

Windows PowerShell
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\Pradeeka>D:
D:\>CD ez_setup
D:\ez_setup>python ez_setup.py
Downloading https://pypi.python.org/packages/source/s/setuptools/setuptools-18.0
.1.zip


Command Prompt
copying setuptools.egg-info\top_level.txt -> build\bdist.win-amd64\egg\EGG-INFO
copying setuptools.egg-info\zip-safe -> build\bdist.win-amd64\egg\EGG-INFO
creating dist
creating 'dist\setuptools-18.0.1-py3.5.egg' and adding 'build\bdist.win-amd64\egg' to it
removing 'build\bdist.win-amd64\egg' (and everything under it)
Processing setuptools-18.0.1-py3.5.egg
Copying setuptools-18.0.1-py3.5.egg to c:\users\pradeeka\appdata\local\programs\python\python35\lib\site-packages
Adding setuptools 18.0.1 to easy-install.pth file
Installing easy_install-3.5-script.py script to C:\Users\Pradeeka\AppData\Local\Programs\Python\Python35\Scripts
Installing easy_install-3.5.exe script to C:\Users\Pradeeka\AppData\Local\Programs\Python\Python35\Scripts
Installing easy_install-script.py script to C:\Users\Pradeeka\AppData\Local\Programs\Python\Python35\Scripts
Installing easy_install.exe script to C:\Users\Pradeeka\AppData\Local\Programs\Python\Python35\Scripts
Installed c:\users\pradeeka\appdata\local\programs\python\python35\lib\site-packages\setuptools-18.0.1-py3.5.egg
Processing dependencies for setuptools==18.0.1
Finished processing dependencies for setuptools==18.0.1
D:\ez_setup>

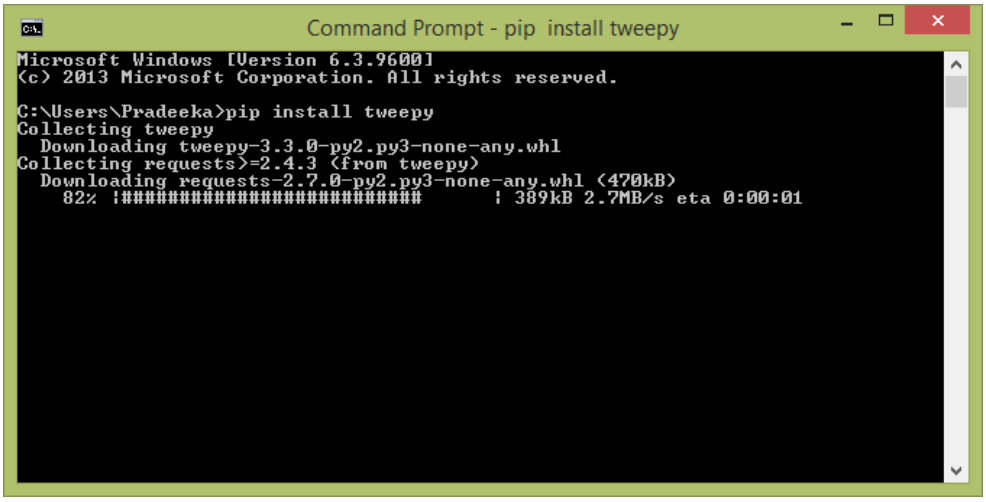
Installing the pip utility on Python



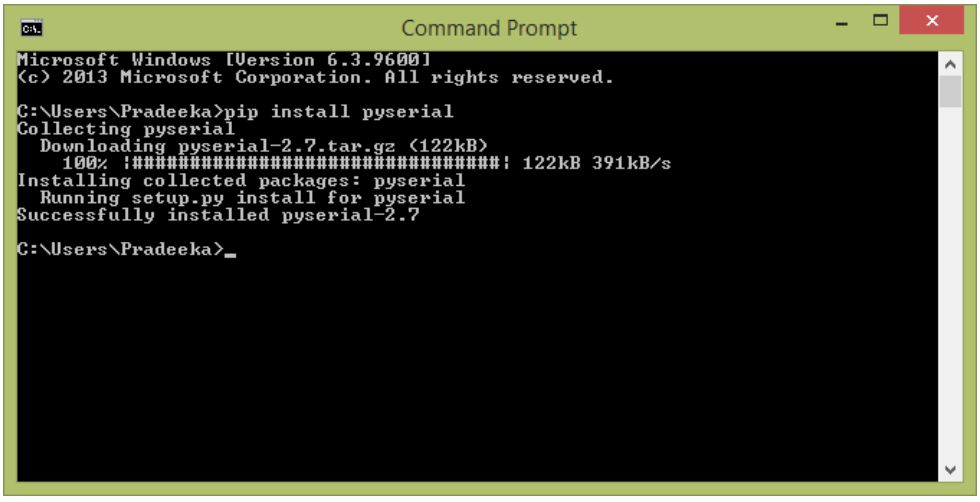
Opening the Python interpreter

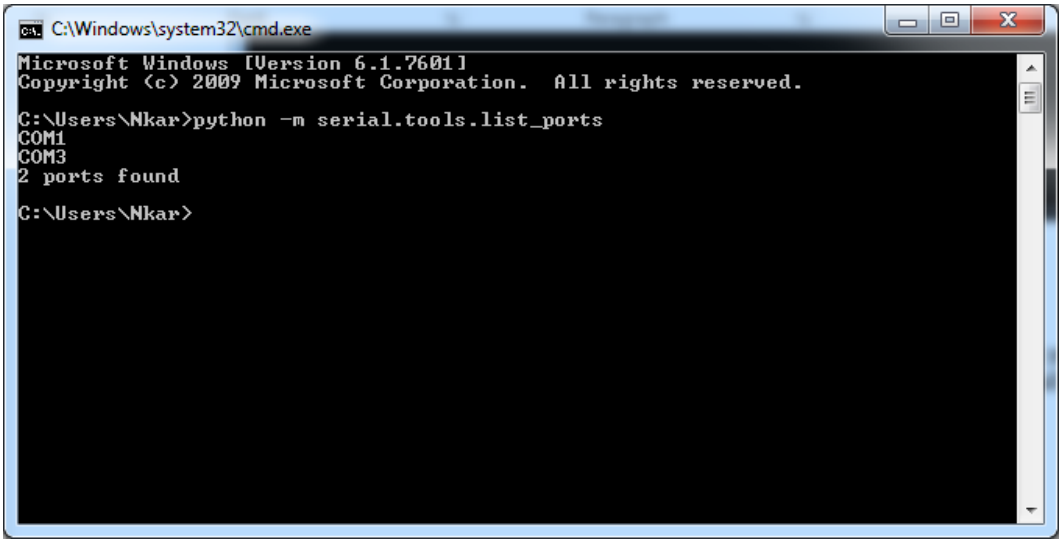


Installing the Tweepy library

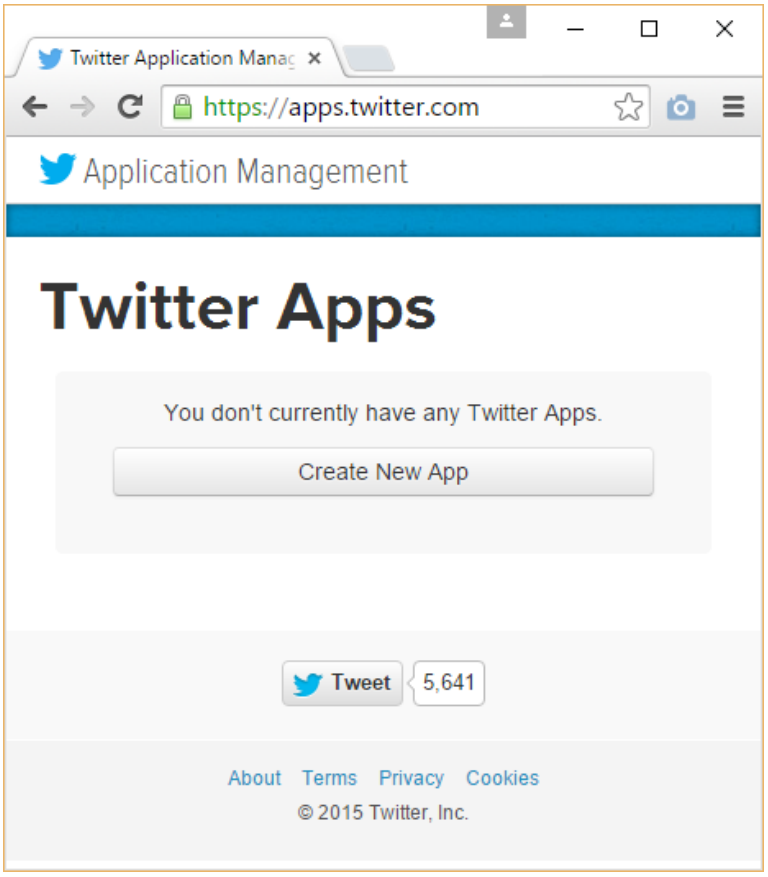


Installing pySerial

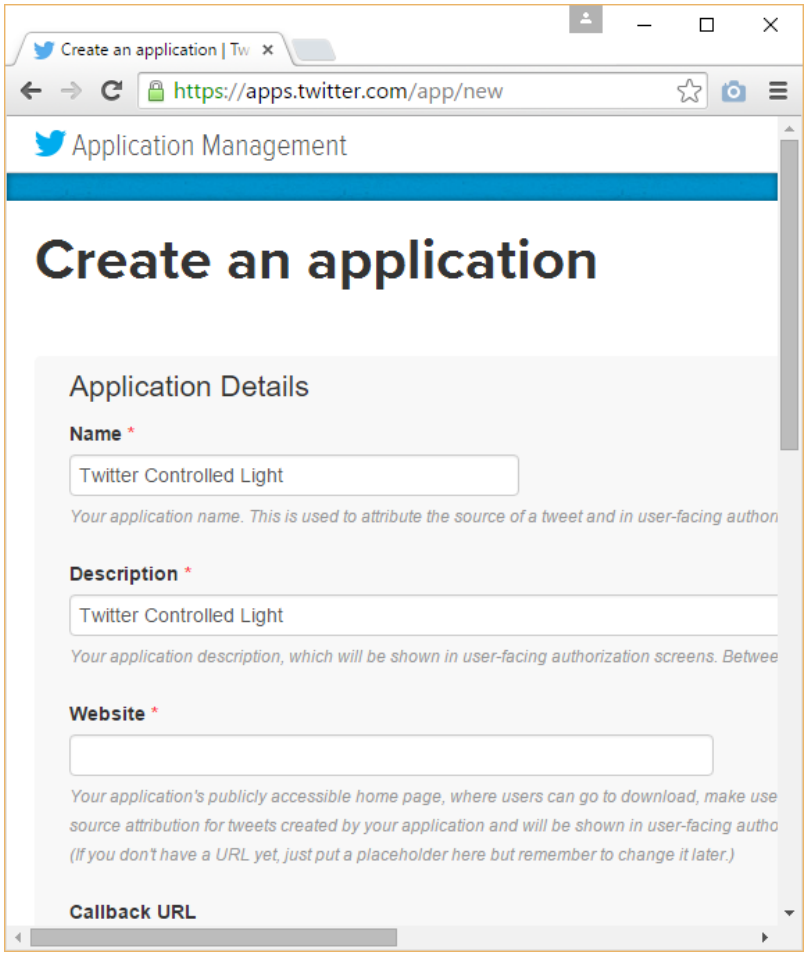



C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Nkar>python -m serial.tools.list_ports
COM1
COM3
2 ports found
C:\Users\Nkar>

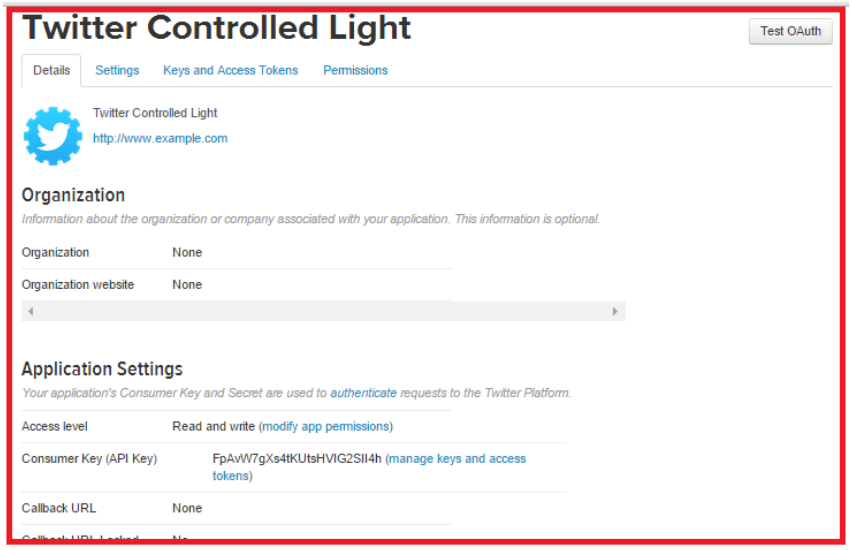
Creating a Twitter app and obtaining API keys



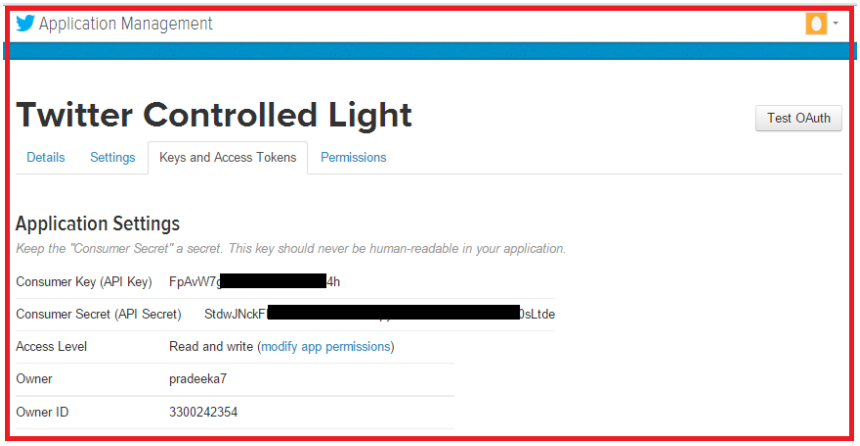
apps.twitter.com, the Application Management start page



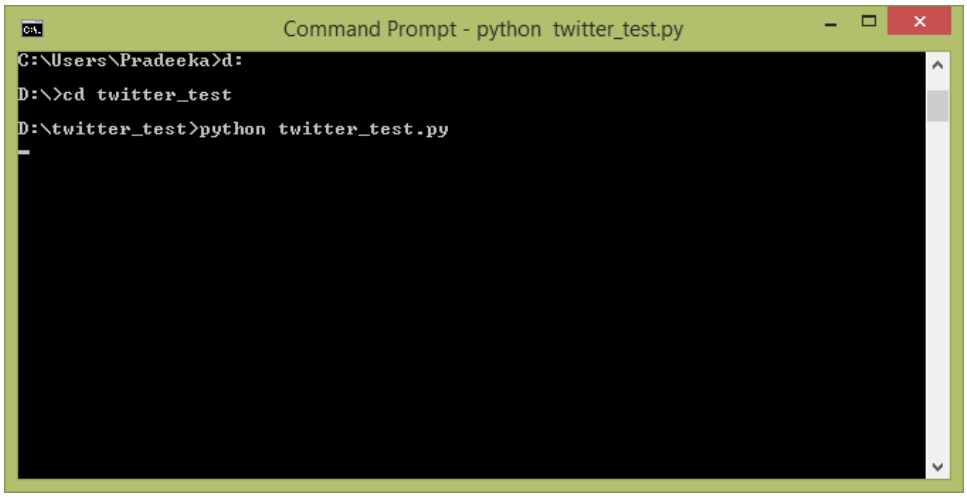
Twitter's Create an application page



The Twitter Application settings page

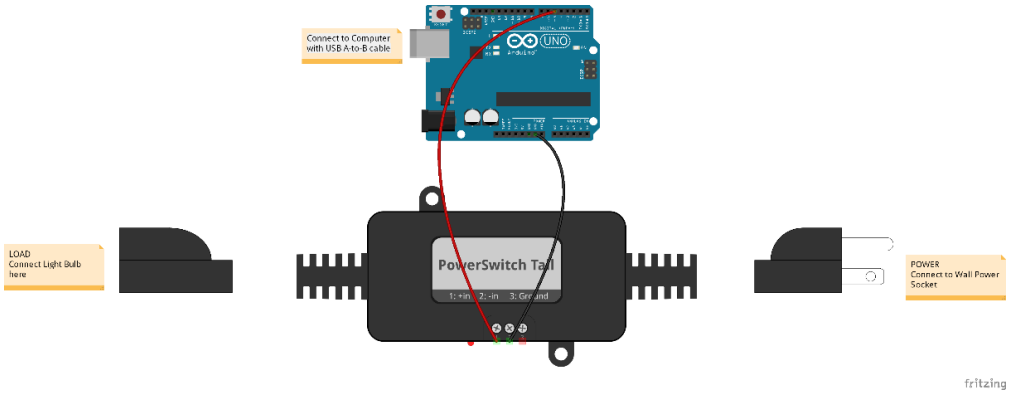


Writing a Python script to read Twitter tweets



Reading the serial data using Arduino

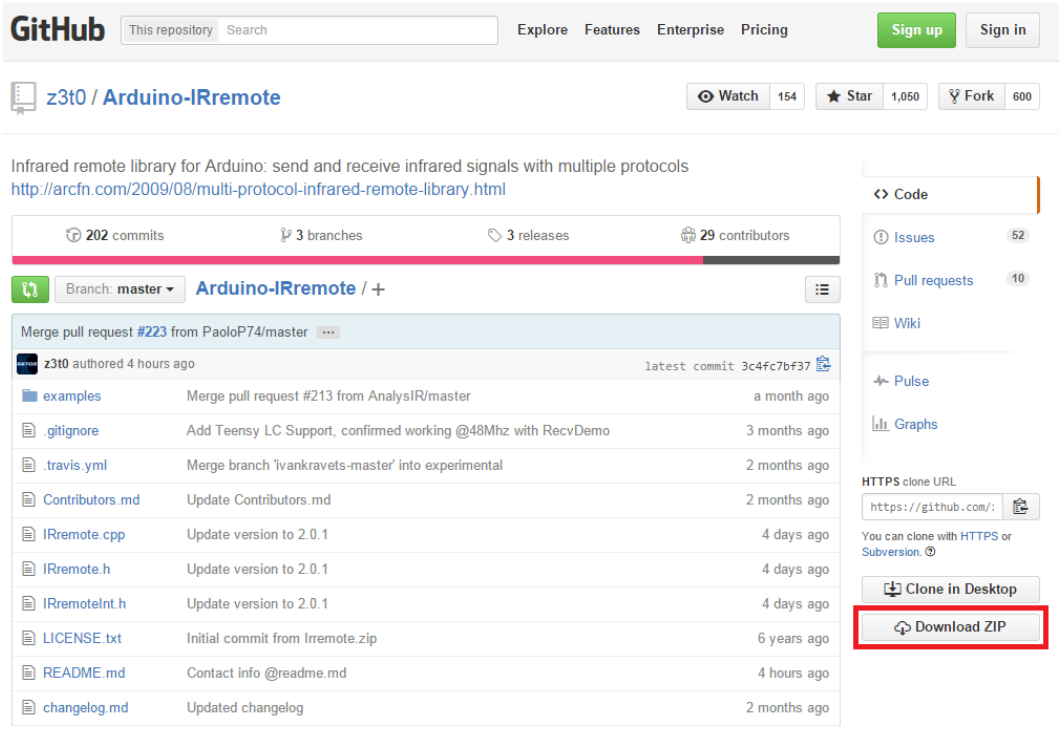
Connecting PowerSwitch Tail with Arduino



Chapter 8: Controlling Infrared Devices using IR Remote

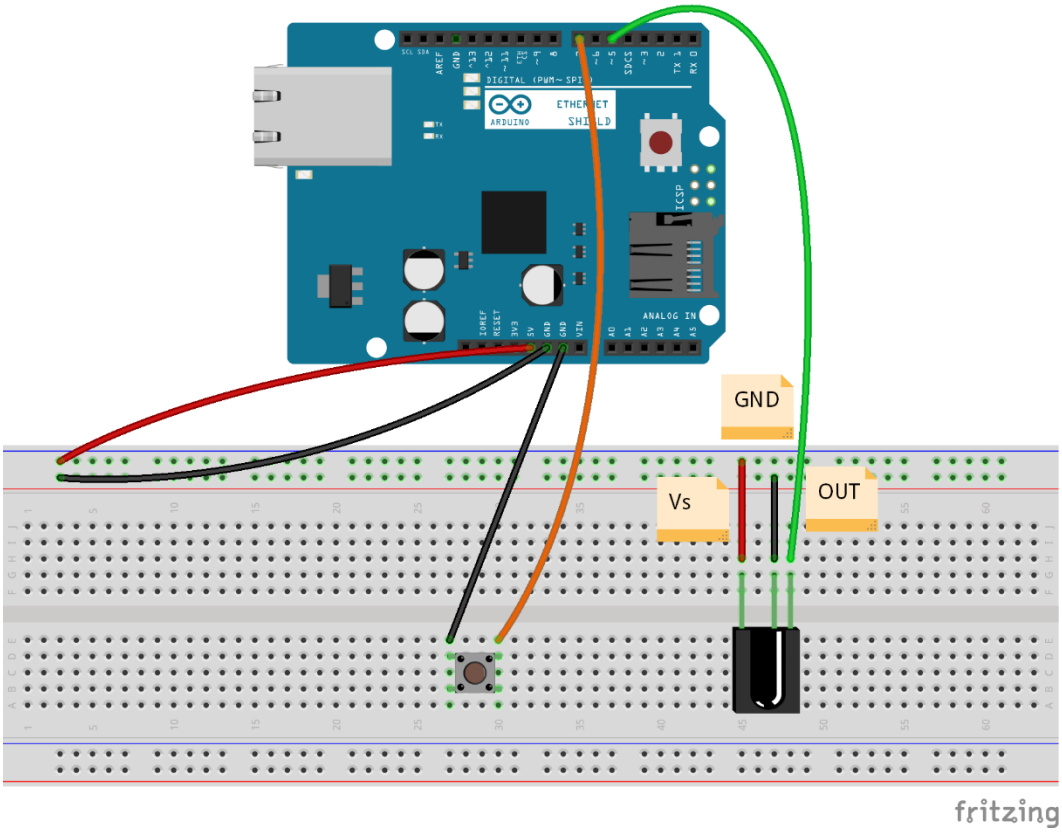
Building an Arduino infrared recorder and remote

Software

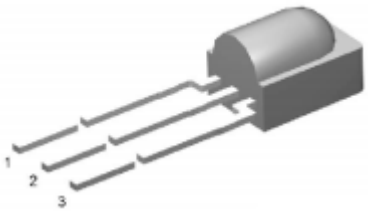


Arduino-IRremote library on GitHub

Building the IR receiver module

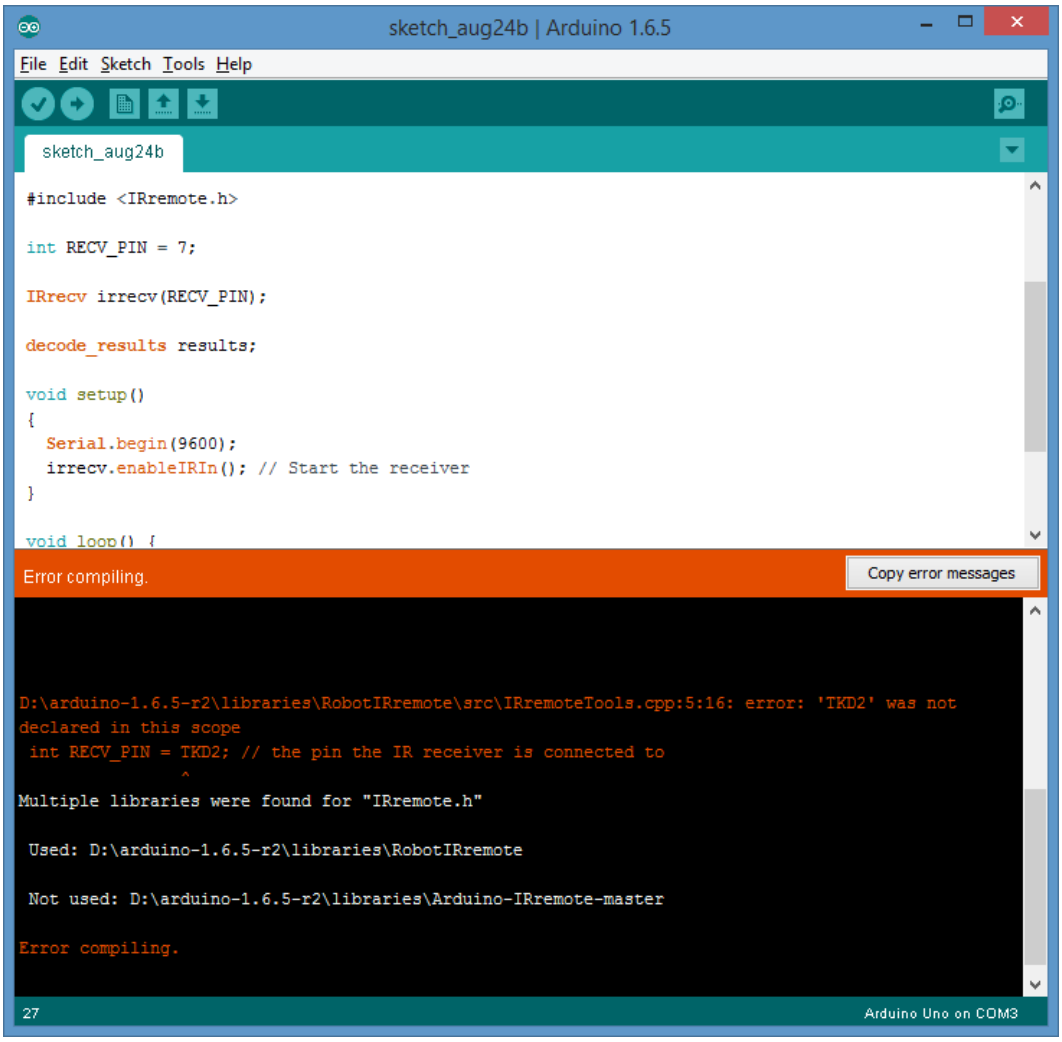


The IR Receiver: The TSOP382 IR receiver is attached to the Arduino+ Ethernet Shield - Fritzing representation

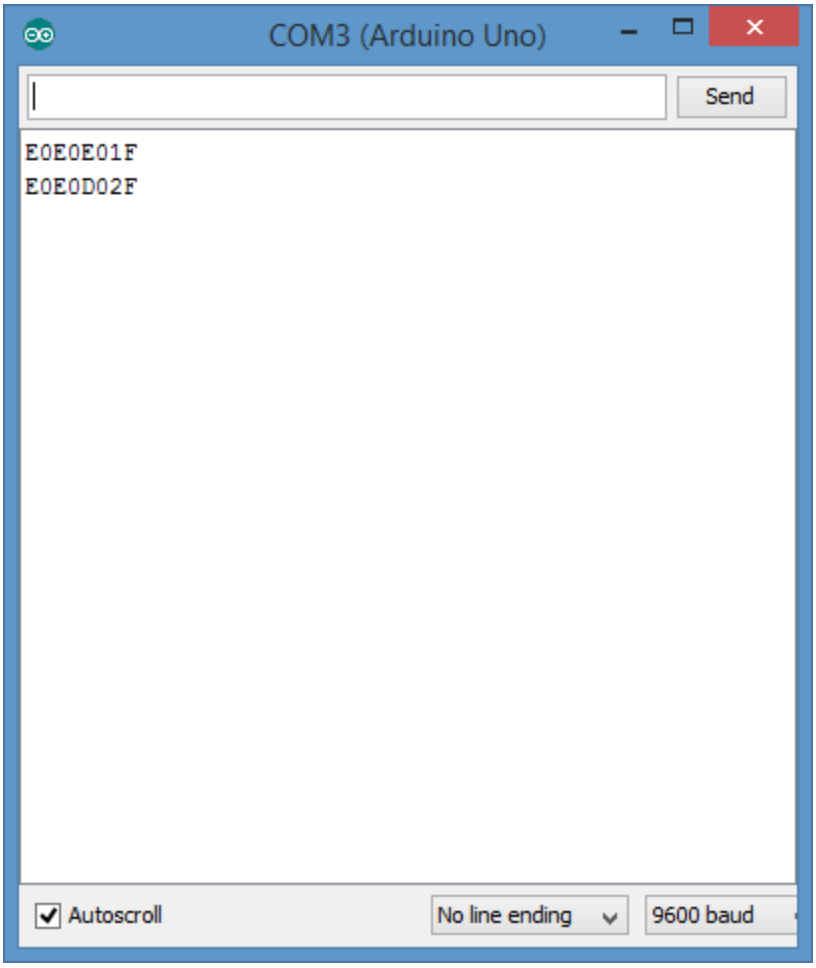


TSOP382 IR receiver diode from Vishay (http://www.vishay.com/)

Capturing IR commands in hexadecimal

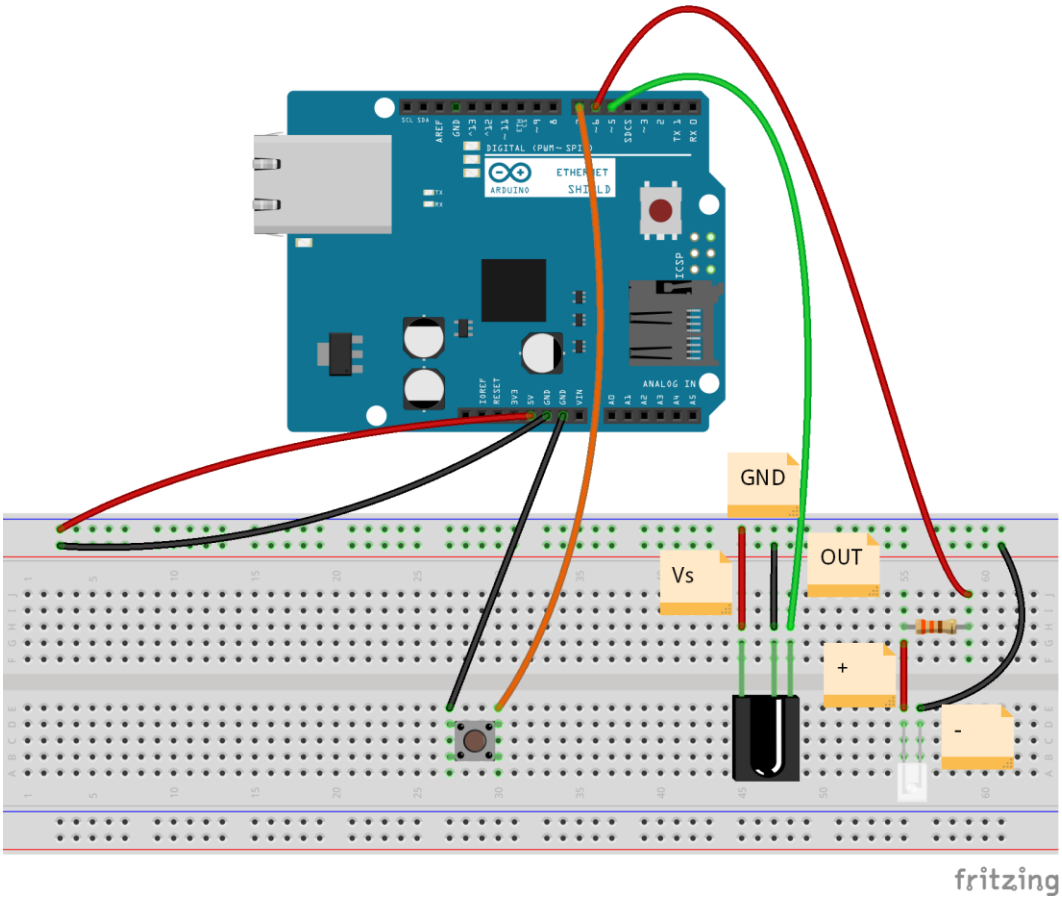


Compiler error because of the conflicting libraries



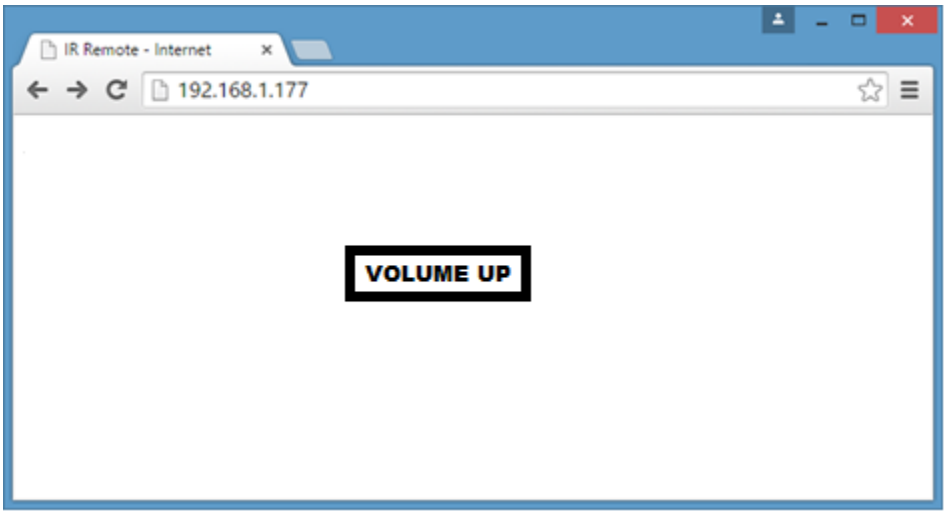
Hexadecimal values for SAMSUNG UA24H4100 TV volume up and volume down remote control buttons

Building the IR sender module



The IR sender: the infrared LED is attached to the Arduino Ethernet Shield—Fritzing Representation

Controlling through the LAN

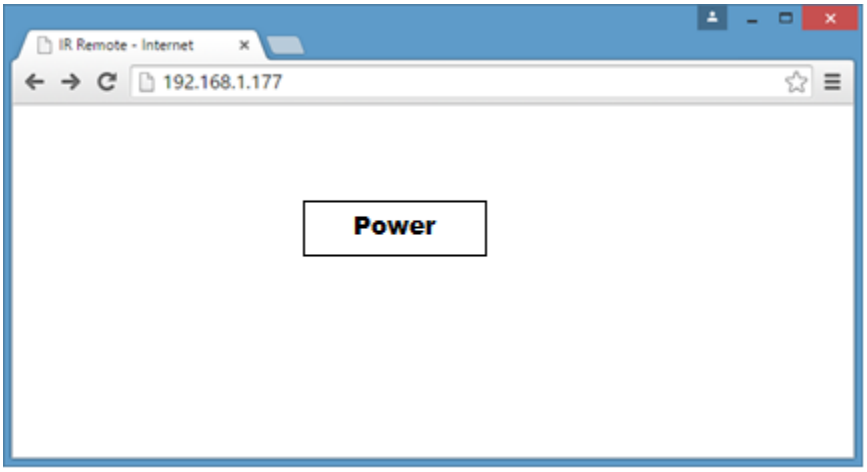


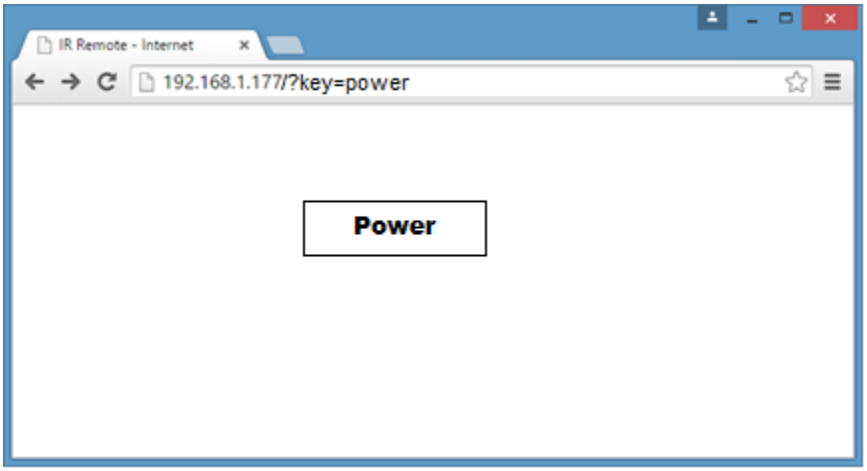
Adding an IR socket to non-IR enabled devices

Infrared Socket—Front View (image taken from eBay)

IR socket side view (image taken from eBay)

The IR remote control for The IR socket (image taken from eBay)




IR Remote - Internet
192.168.1.177/?key=power
Power