

Intel Innovation Week

University of Johannesburg (23 – 27 Jun 2014)

Connectivity



What will you make?

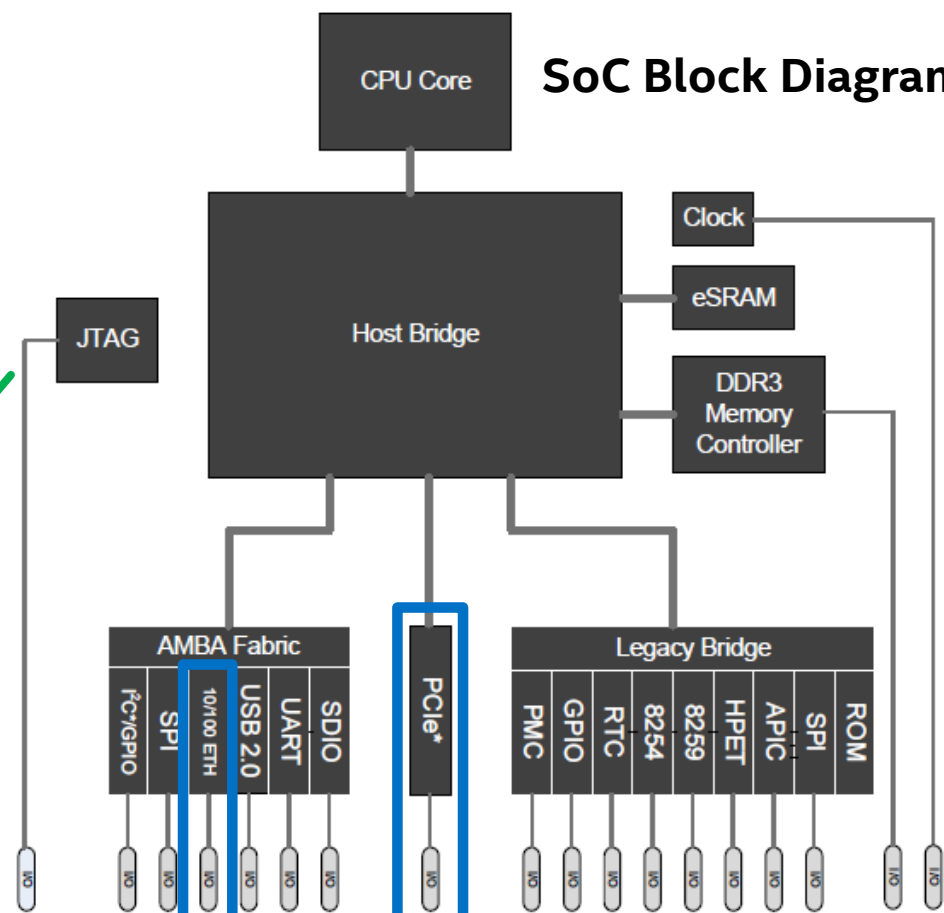


Connectivity - Ethernet/Wifi Introduction

Ethernet - hardware

- SoC has 2 * 10/100 Mbps Ethernet
- Galileo exposes 1 Ethernet port
- No Ethernet shield needed
- Use existing Arduino sketches

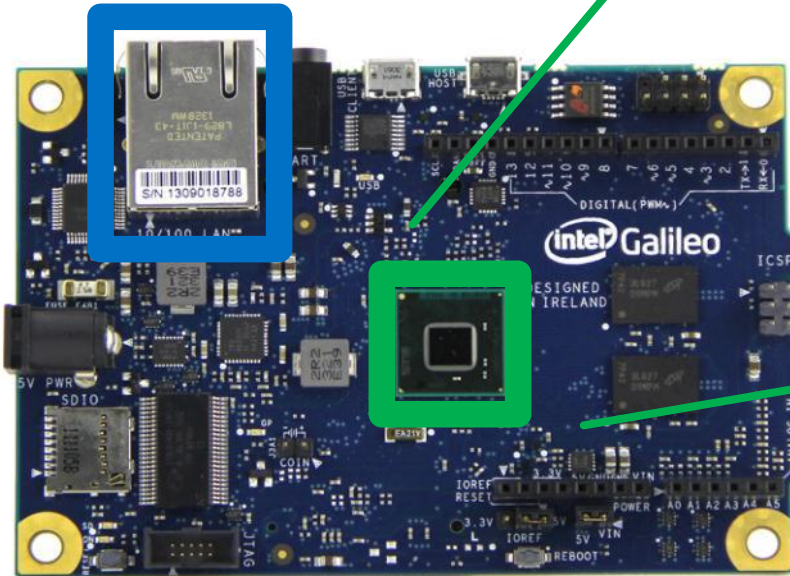
SoC Block Diagram



2 SoC Ethernet Controllers

2 SoC PCIe lanes

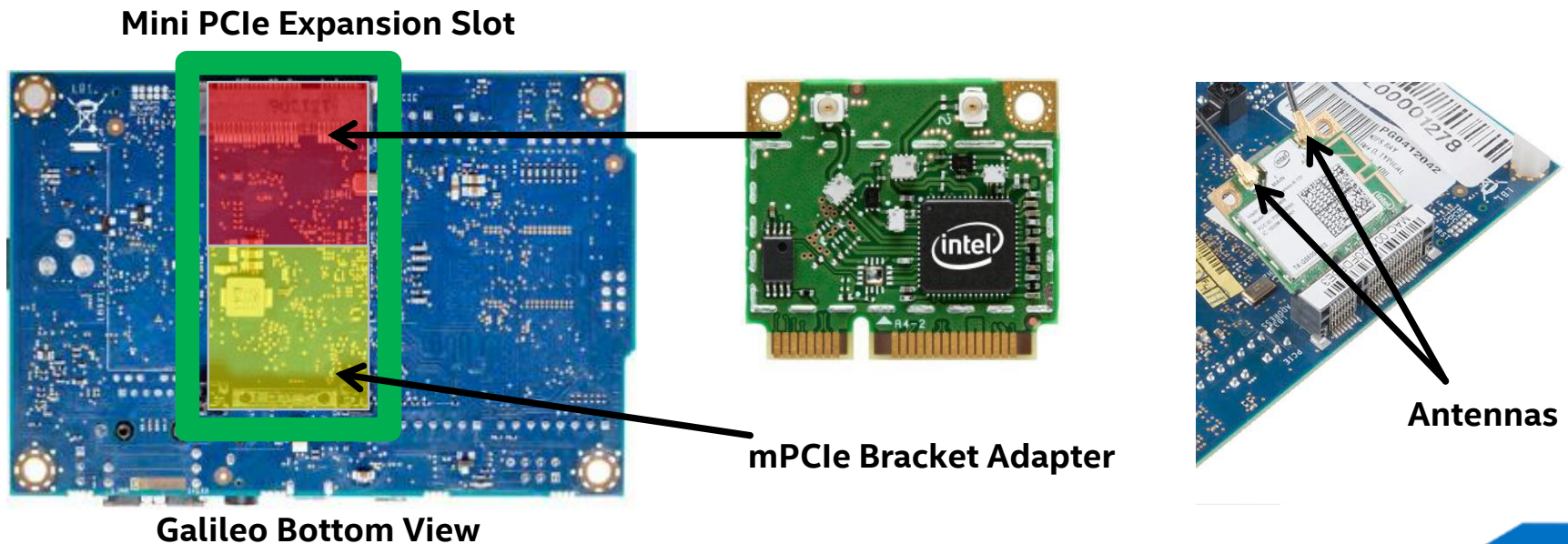
Ethernet port



Galileo Top View

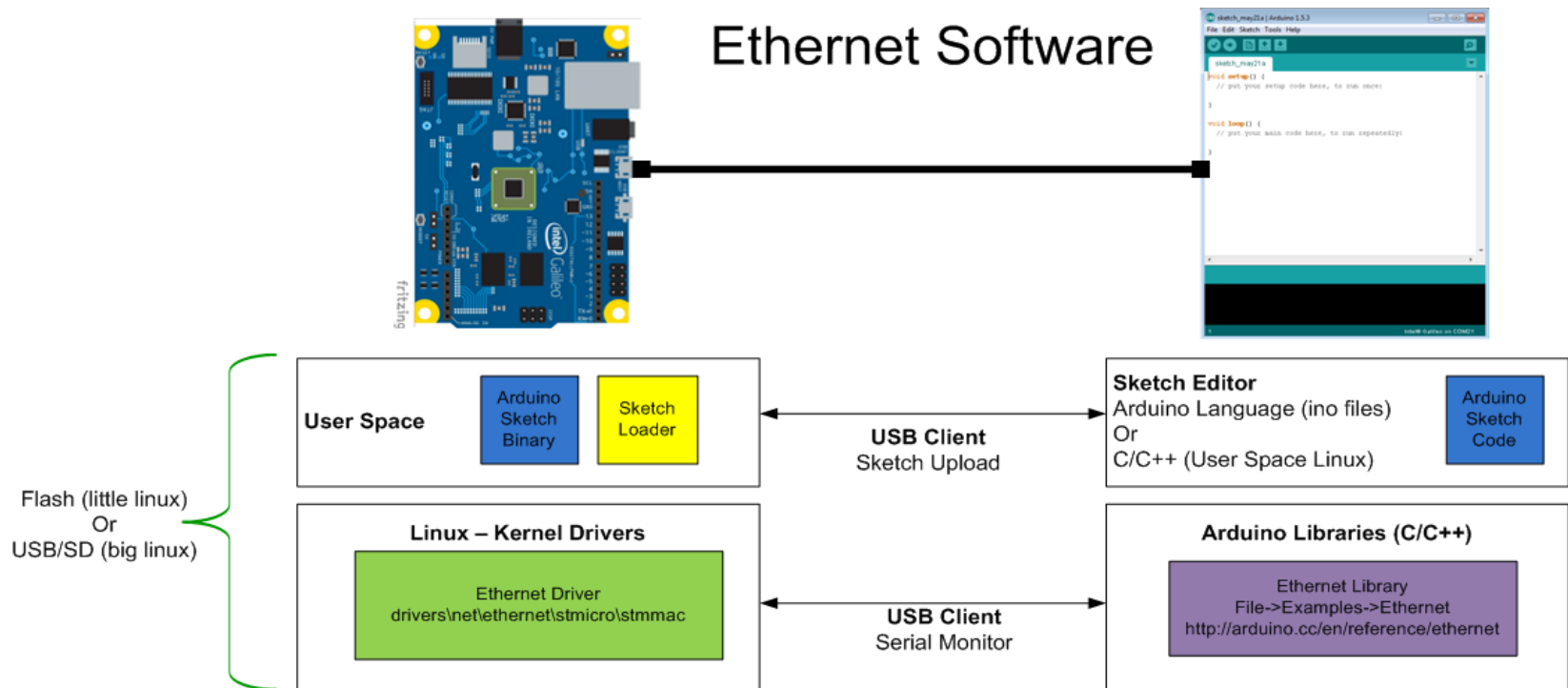
WiFi - hardware

- SoC has 2 PCI Express* ports
- Galileo exposes 1 on the bottom of the board through a Mini PCIe slot
- Plug in card N-135 (Wifi+Bluetooth) or (N-6025) (Wifi only)
- Also screw in mPCIe bracket adapter to hold mPCIe card in place
- Plug in your Antennas
- You need to use the Linux Image for SD for Wifi to work



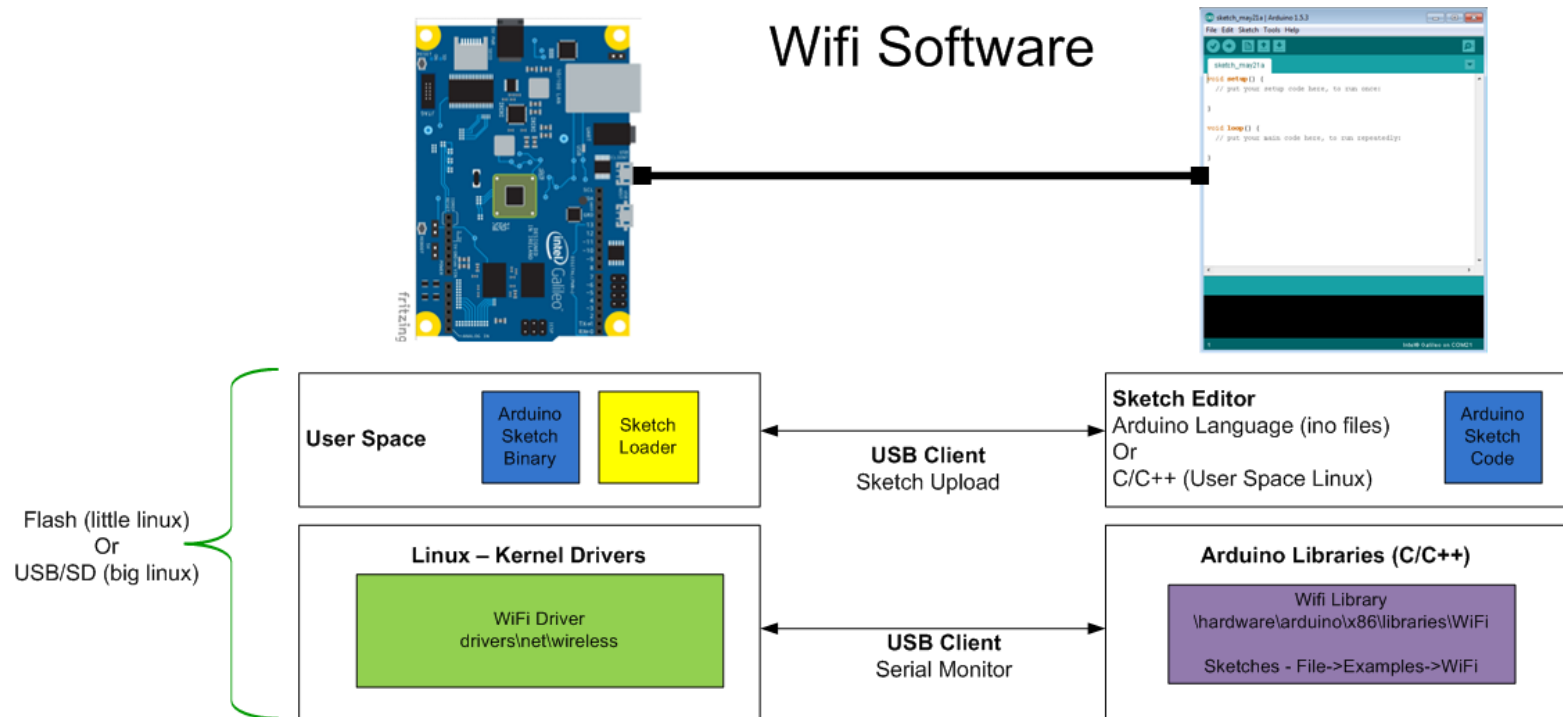
Ethernet - Software

- Galileo Ethernet libraries are used by default for the Ethernet port
- You **DONT** need to use the Linux Image for SD for Ethernet to work (Flash or Mass storage works)
- Reference - <http://arduino.cc/en/reference/ethernet>



Wifi - Software

- Galileo Wifi libraries are used by default for the Wifi card
- You **DO** need to use the Linux Image for SD for Wifi to work (Flash or Mass storage works)
- Reference - <http://arduino.cc/en/Reference/WiFi>



Setup - Ethernet

- Open putty serial console to Galileo using serial cable
- Type the Linux command “ifconfig”
- eth0 is the Ethernet port and its IP address is displayed
- ping an address or website to ensure you are properly connected
- Take note of your mac address, you will need it later !

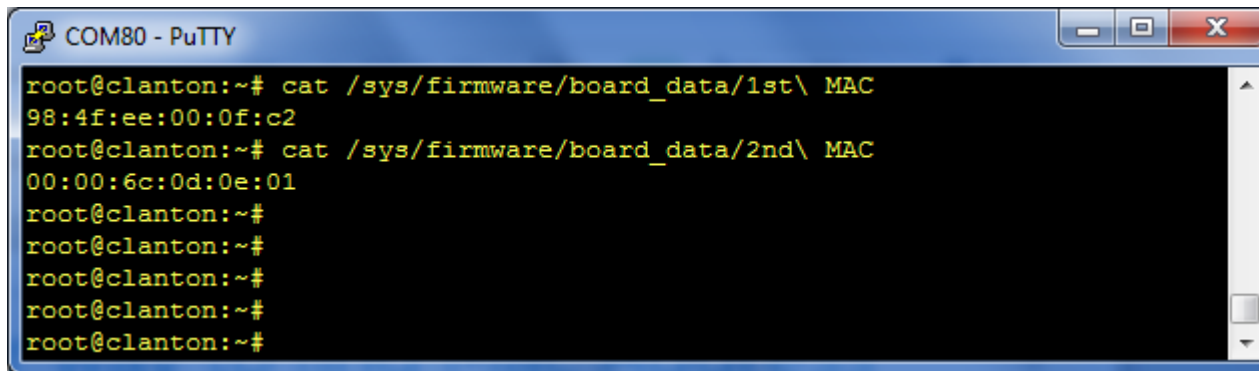
```
root@clanton:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 98:4F:EE:00:0F:C2
          inet addr:192.168.1.18  Bcast:0.0.0.0  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:104 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9089 (8.8 KiB)  TX bytes:1958 (1.9 KiB)
          Interrupt:41 Base address:0x4000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@clanton:~#
root@clanton:~#
root@clanton:~# ping www.google.com
PING www.google.com (74.125.24.103): 56 data bytes
64 bytes from 74.125.24.103: seq=0 ttl=50 time=10.304 ms
64 bytes from 74.125.24.103: seq=1 ttl=50 time=9.341 ms
```

Setup – How to see board specific data

- Open PuTTY serial console to Galileo using serial cable
- Board specific data from your Flash memory can be viewed on the Linux file system located at `/sys/firmware/board_data/`
- Eg Mac address's are unique to your board so they can be viewed by entering the commands below

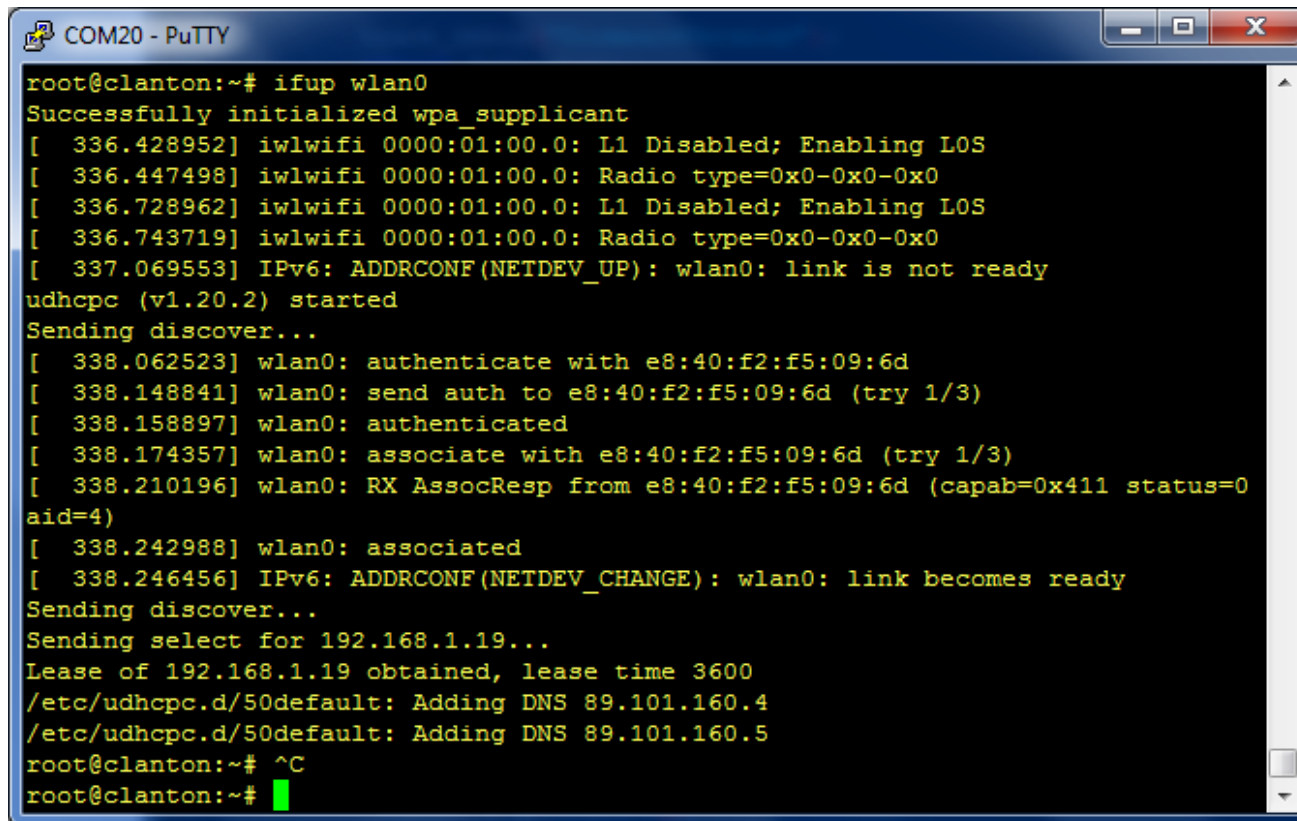


A screenshot of a PuTTY terminal window titled "COM80 - PuTTY". The terminal shows a Linux shell prompt "root@clanton:~#" and two commands being executed. The first command is "cat /sys/firmware/board_data/1st\ MAC" and the output is "98:4f:ee:00:0f:c2". The second command is "cat /sys/firmware/board_data/2nd\ MAC" and the output is "00:00:6c:0d:0e:01". Below these, there are four more prompts without output.

```
root@clanton:~# cat /sys/firmware/board_data/1st\ MAC
98:4f:ee:00:0f:c2
root@clanton:~# cat /sys/firmware/board_data/2nd\ MAC
00:00:6c:0d:0e:01
root@clanton:~#
root@clanton:~#
root@clanton:~#
root@clanton:~#
root@clanton:~#
```


Setup - Wifi

- Open serial console to Galileo using serial cable
- Ensure you are booting from “Mass Storage” and not “Flash”
- Type the Linux command “ifup wlan0” – you should see similar to below



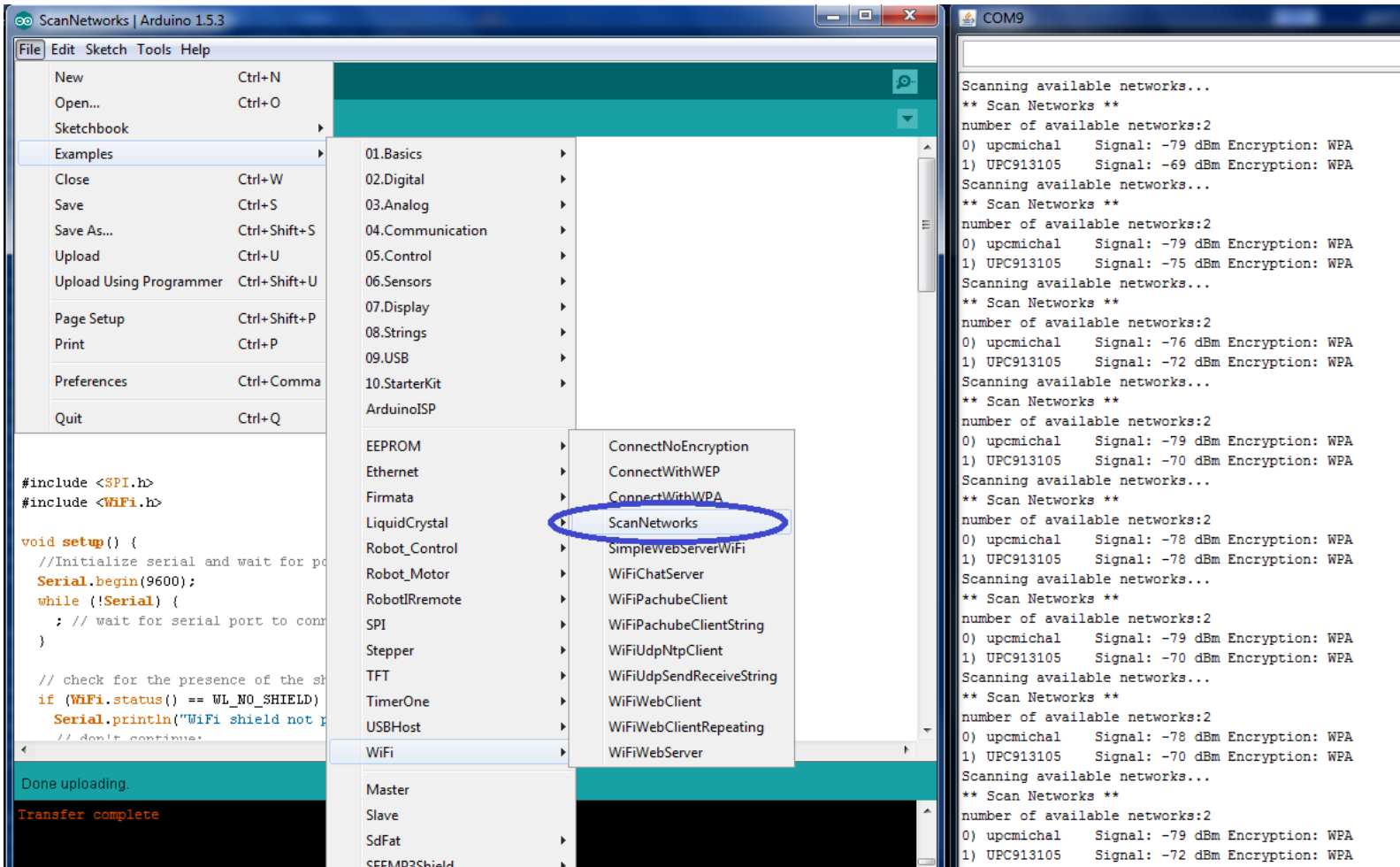
```
COM20 - PuTTY
root@clanton:~# ifup wlan0
Successfully initialized wpa_supplicant
[ 336.428952] iwlwifi 0000:01:00.0: L1 Disabled; Enabling LOS
[ 336.447498] iwlwifi 0000:01:00.0: Radio type=0x0-0x0-0x0
[ 336.728962] iwlwifi 0000:01:00.0: L1 Disabled; Enabling LOS
[ 336.743719] iwlwifi 0000:01:00.0: Radio type=0x0-0x0-0x0
[ 337.069553] IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
udhcpc (v1.20.2) started
Sending discover...
[ 338.062523] wlan0: authenticate with e8:40:f2:f5:09:6d
[ 338.148841] wlan0: send auth to e8:40:f2:f5:09:6d (try 1/3)
[ 338.158897] wlan0: authenticated
[ 338.174357] wlan0: associate with e8:40:f2:f5:09:6d (try 1/3)
[ 338.210196] wlan0: RX AssocResp from e8:40:f2:f5:09:6d (capab=0x411 status=0
aid=4)
[ 338.242988] wlan0: associated
[ 338.246456] IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
Sending discover...
Sending select for 192.168.1.19...
Lease of 192.168.1.19 obtained, lease time 3600
/etc/udhcpc.d/50default: Adding DNS 89.101.160.4
/etc/udhcpc.d/50default: Adding DNS 89.101.160.5
root@clanton:~# ^C
root@clanton:~#
```

Setup - Wifi

- Now run “ifconfig” and see wlan0 with associated IP address
- ping an address or website to ensure you are properly connected

Setup - Wifi

Verify your WiFi module works by loading the ScanNetworks sketch



Connectivity – Simple Web Client

Simple Web Client - Ethernet Library

- Reference <http://arduino.cc/en/Tutorial/WebClient>
- Open sketch File -> Examples -> Ethernet -> WebClient
- **No** need to Modify mac address for Client as Galileo knows to use its own mac address id you call **Ethernet.begin(mac);**
- In summary the sketch returns a Google search term for the word "Arduino". The results of this search are viewable as HTML through your Arduino's serial monitor.
- Upload sketch and open serial monitor quickly (ctrl+shift+m)
- Now start looking at the sketch code

Simple Web Client – Sketch Explained

- What is a Class/Object? (see Ethernet class, Server class etc)
- See Classes here - <http://arduino.cc/en/reference/ethernet>
- Galileo replaces the Ethernet shield with its own Ethernet port
- Eg...

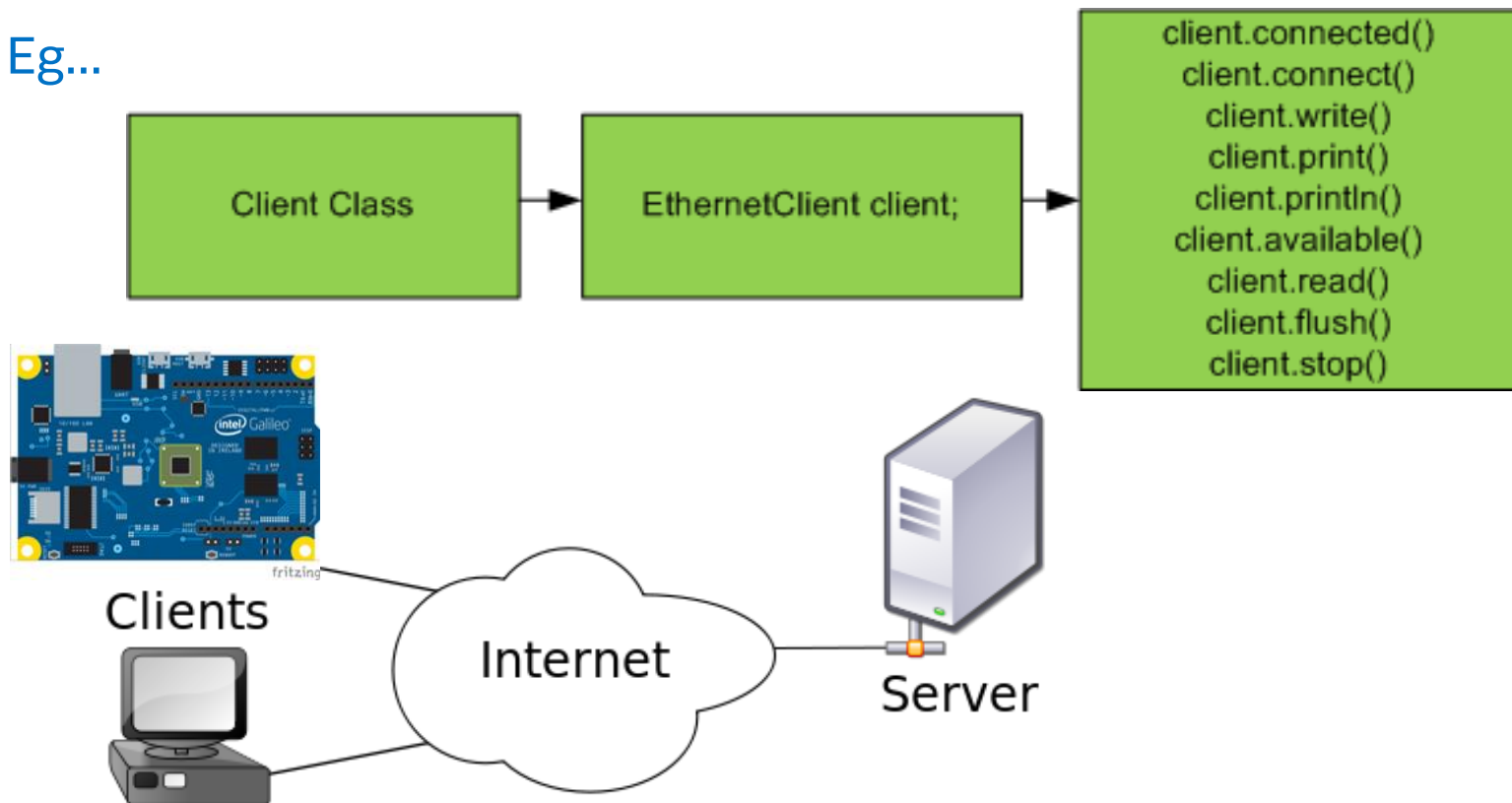


Image Source: http://en.wikipedia.org/wiki/Client%E2%80%93server_model

Simple Web Client – Wifi

Using Wifi is almost identical to Ethernet just do the following

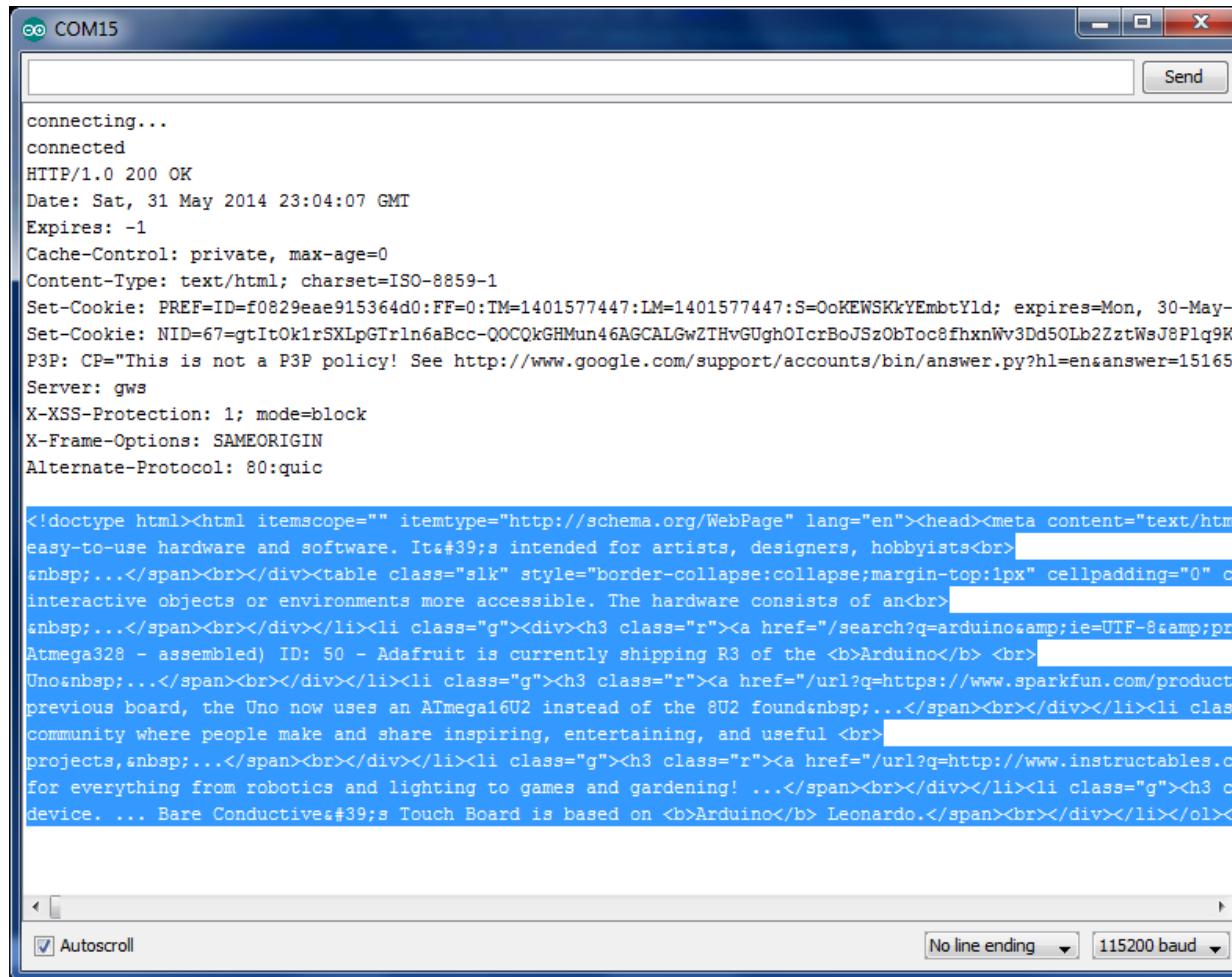
1. Use sketch File -> Examples -> WiFi -> WiFiWebClient
2. Modify the security to match your network on lines 27/28
`char ssid[] = "yourNetwork"; // your network SSID (name)`
`char pass[] = "secretPassword"; // your network password (use for WPA, or use as key for WEP)`

NOTE:

For any Wifi sketches. If your Wifi router has open security and has no password then just replace `WiFi.begin(ssid, pass);` -> `WiFi.begin(ssid);`

Simple Web Client – Serial Monitor Output

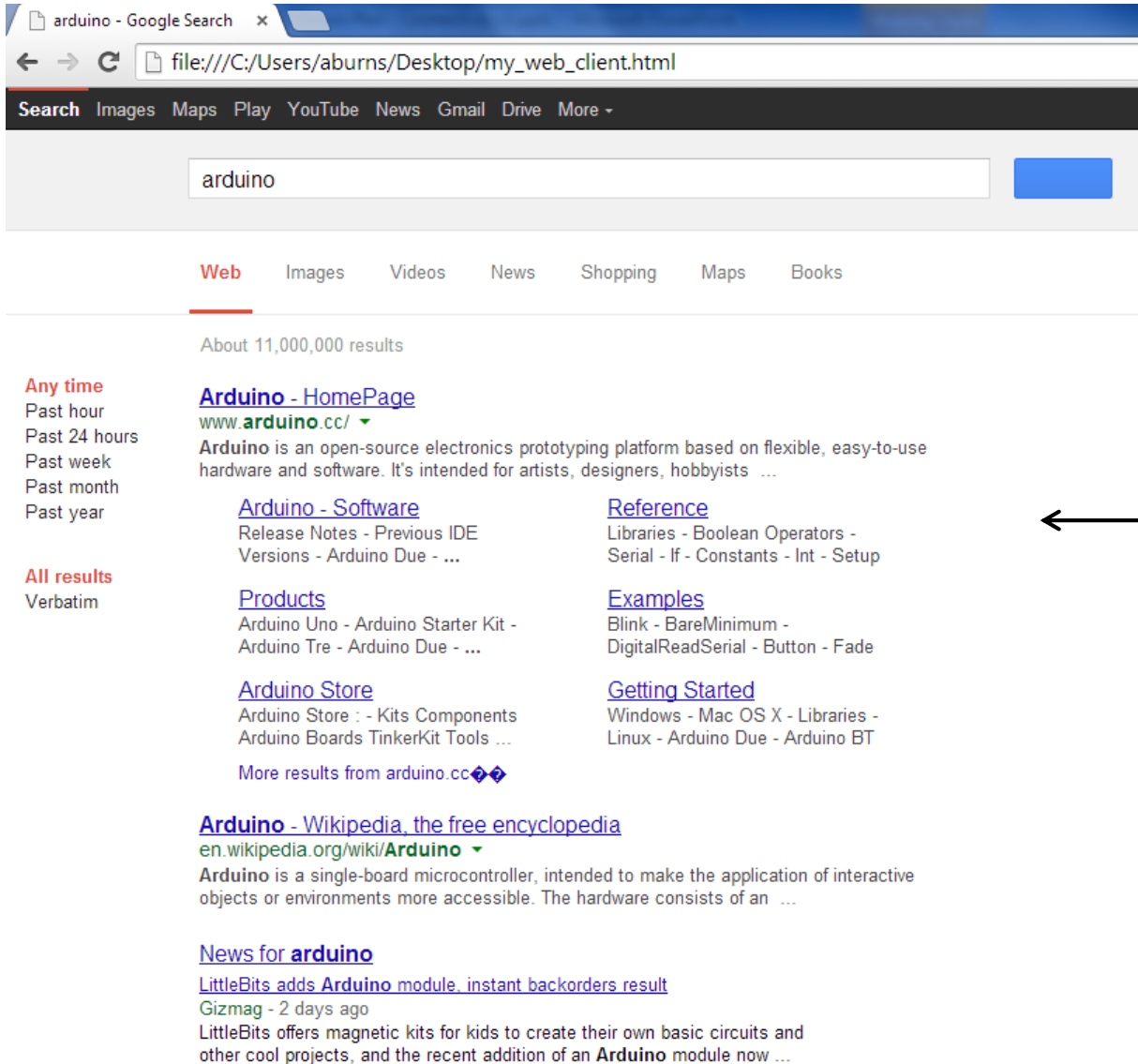
- Copy the text below – the HTML web page data
- Save it in a text editor as a *.html file then open in a web browser



```
COM15
connecting...
connected
HTTP/1.0 200 OK
Date: Sat, 31 May 2014 23:04:07 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
Set-Cookie: PREF=ID=f0829eae915364d0:FF=0:TM=1401577447:LM=1401577447:S=OoKEWSKkYEmbtYld; expires=Mon, 30-May-
Set-Cookie: NID=67=gtItOk1rSXLpGTrln6aBcc-QOCQkGHMun46AGCALGwZTHvGUghOIcrBoJSzObToc8fhxnWv3Dd50Lb2ZztWsJ8Plq9K
P3P: CP="This is not a P3P policy! See http://www.google.com/support/accounts/bin/answer.py?hl=en&answer=15165
Server: gws
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Alternate-Protocol: 80:quic

<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en"><head><meta content="text/html
easy-to-use hardware and software. It's intended for artists, designers, hobbyists
<br>...</span><br></div><table class="slk" style="border-collapse:collapse;margin-top:1px" cellpadding="0" c
interactive objects or environments more accessible. The hardware consists of an<br>
<br>...</span><br></div></li><li class="g"><div><h3 class="r"><a href="/search?q=arduino&ie=UTF-8&pr
Atmega328 - assembled) ID: 50 - Adafruit is currently shipping R3 of the <b>Arduino</b> <br>
Uno<br>...</span><br></div></li><li class="g"><h3 class="r"><a href="/url?q=https://www.sparkfun.com/product
previous board, the Uno now uses an ATmega16U2 instead of the 8U2 found<br>...</span><br></div></li><li clas
community where people make and share inspiring, entertaining, and useful <br>
projects,<br>...</span><br></div></li><li class="g"><h3 class="r"><a href="/url?q=http://www.instructables.c
for everything from robotics and lighting to games and gardening! ...</span><br></div></li><li class="g"><h3 c
device. ... Bare Conductive's Touch Board is based on <b>Arduino</b> Leonardo.</span><br></div></li></ol>
```


Simple Web Client - View Webpage






The screenshot shows a web browser window with the address bar displaying `file:///C:/Users/aburns/Desktop/my_web_client.html`. The browser's search bar contains the word "arduino". Below the search bar, the "Web" tab is selected, showing search results for "arduino".

On the left side of the search results, there are filters for "Any time" (Past hour, Past 24 hours, Past week, Past month, Past year) and "All results" (Verbatim).

The main search results include:

- Arduino - HomePage**
www.arduino.cc/ 
 Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists ...
- Arduino - Software**
 Release Notes - Previous IDE
 Versions - Arduino Due - ...
- Products**
 Arduino Uno - Arduino Starter Kit -
 Arduino Tre - Arduino Due - ...
- Arduino Store**
 Arduino Store : - Kits Components
 Arduino Boards TinkerKit Tools ...
- Reference**
 Libraries - Boolean Operators -
 Serial - If - Constants - Int - Setup
- Examples**
 Blink - BareMinimum -
 DigitalReadSerial - Button - Fade
- Getting Started**
 Windows - Mac OS X - Libraries -
 Linux - Arduino Due - Arduino BT

More results from arduino.cc  

Arduino - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Arduino 
 Arduino is a single-board microcontroller, intended to make the application of interactive objects or environments more accessible. The hardware consists of an ...

News for arduino
[LittleBits adds Arduino module, instant backorders result](#)
 Gizmag - 2 days ago
 LittleBits offers magnetic kits for kids to create their own basic circuits and other cool projects, and the recent addition of an **Arduino** module now ...

This is the webpage that we have just downloaded from Galileo following the google.com search. Now go to your web browser and search for "arduino" to see the same results downloaded.



Simple Web Client - Engineering Challenge



Challenge 1 (Everyone)

- File -> Save As... and save new sketch called WebClientRepeat.ino
- Make sketch Download 2 web pages instead of 1
- Hint, create 2 new functions called `connectToServer()` and `getWebData()` and put the correct code in them
- Leave sketch configurable as to
 - a) The number of webpages to download and
 - b) The time in between each download

Project Files:

Ethernet Code: Lesson4-Connectivity\Section2-SimpleWebClient\WebClientRepeat

Wifi Code: Lesson4-Connectivity\Section2-SimpleWebClient\WiFiWebClientRepeat

Connectivity – Simple Web Server

Simple Web Server – Sketch Explained

- Reference <http://arduino.cc/en/Tutorial/WebServer>
- Open sketch File -> Examples -> Ethernet -> WebServer
- Now start looking at the sketch code but don't do lab yet
- Galileo creates a simple Web server to answer a HTTP request. Then open a browser and navigate to your Galileo's Ethernet IP address. Galileo will respond with just enough HTML for a browser to display the input values from the six analog pins.

Simple Web Server – Sketch Explained

- What is a Class/Object? (see Ethernet class, Server class etc)
- See Classes here - <http://arduino.cc/en/reference/ethernet>
- Galileo replaces the Ethernet shield with its own Ethernet port
- Eg...

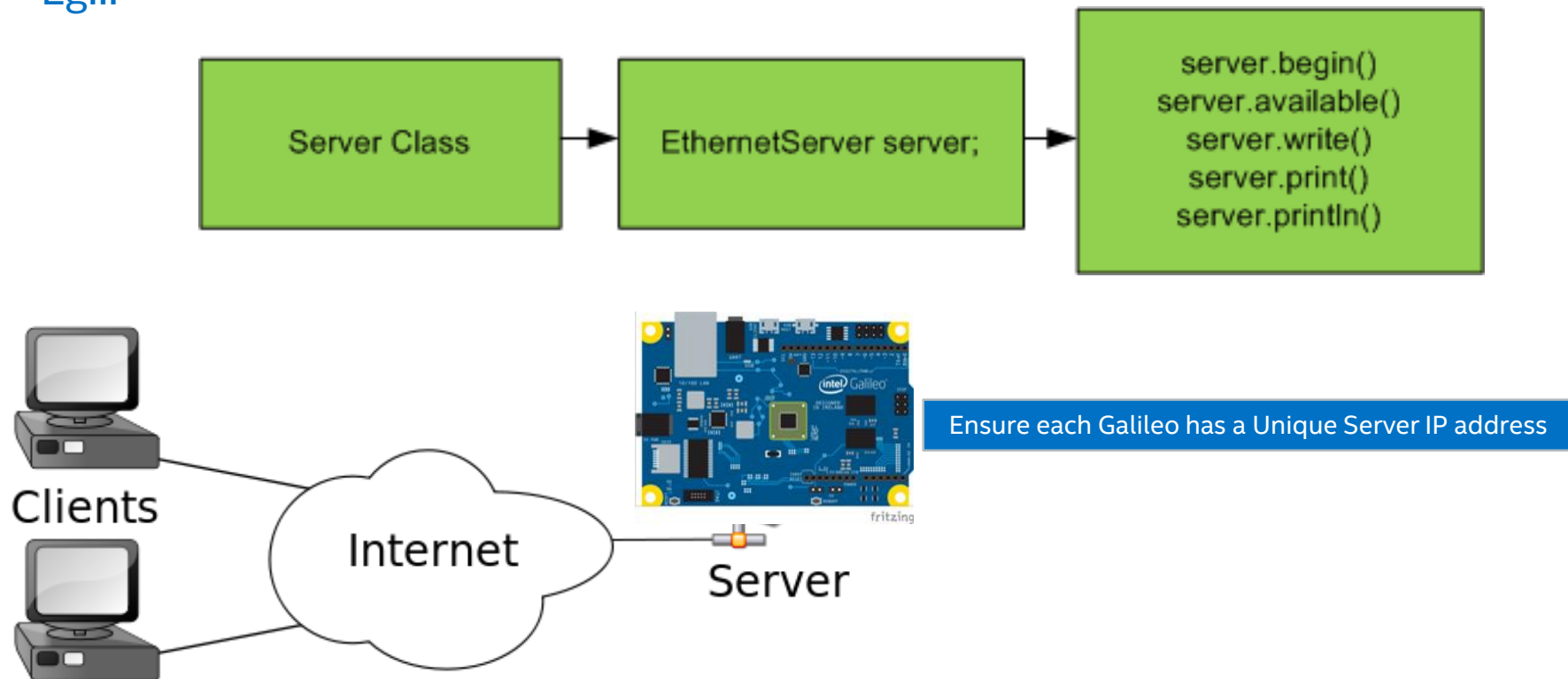
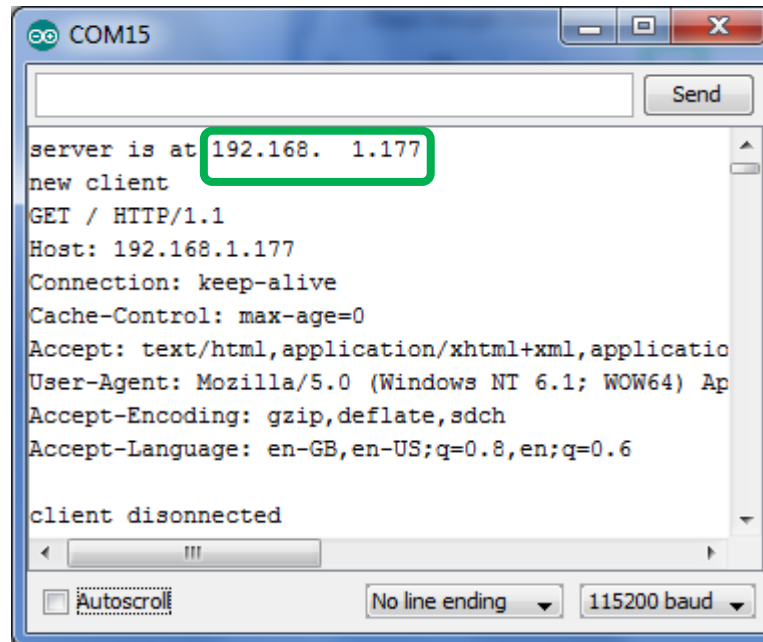


Image Source: http://en.wikipedia.org/wiki/Client%E2%80%93server_model

Simple Web Server – Ethernet Example

- In sketch: File -> Examples -> Ethernet -> WebServer
- Each device on a network must have a unique mac address (line 24)
- Change the server IP address (line 25) to a unique number 192.168.1.XXX and Ensure you use a unique number for XXX
- Upload sketch and open serial monitor quickly (ctrl+shift+m)
- Now change the values coming into the Analog pins A0-A5

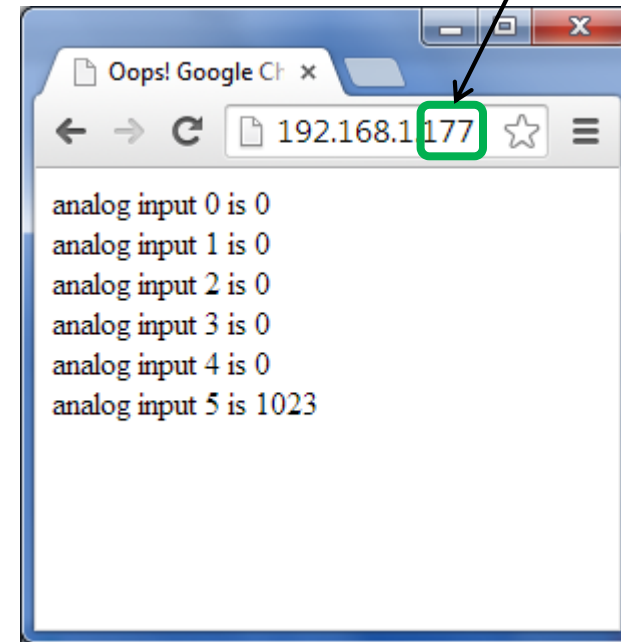


COM15

```
server is at 192.168.1.177
new client
GET / HTTP/1.1
Host: 192.168.1.177
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/javascript;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/31.0.2069.64 Safari/537.36
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-GB,en-US;q=0.8,en;q=0.6

client disconnected
```

Autoscroll No line ending 115200 baud



Oops! Google Chrome

192.168.1.177

```
analog input 0 is 0
analog input 1 is 0
analog input 2 is 0
analog input 3 is 0
analog input 4 is 0
analog input 5 is 1023
```

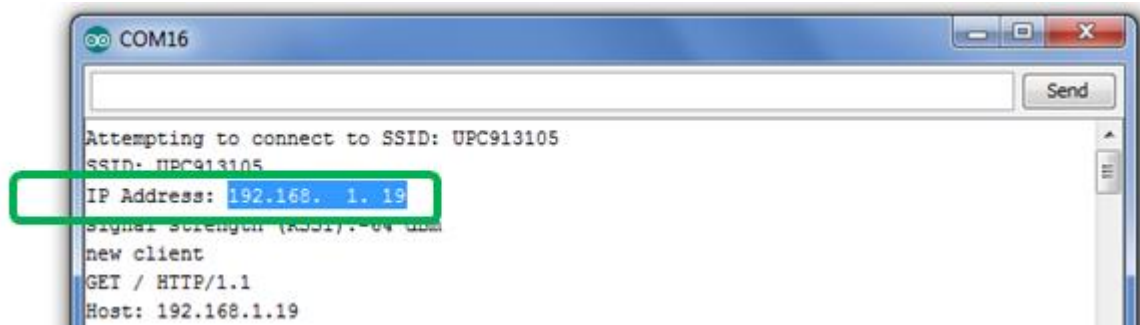

Simple Web Server – Wifi Example

Using Wifi is almost identical to Ethernet just do the following

1. Use sketch File -> Examples -> WiFi -> WiFiWebServer
2. Modify the security to match your network
`char ssid[] = "yourNetwork"; // your network SSID (name)`
`char pass[] = "secretPassword"; // your network password (use for WPA, or use as key for WEP)`

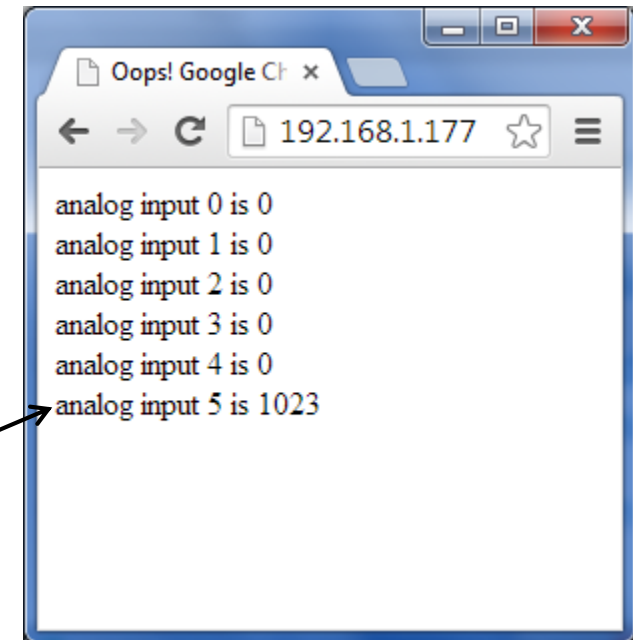
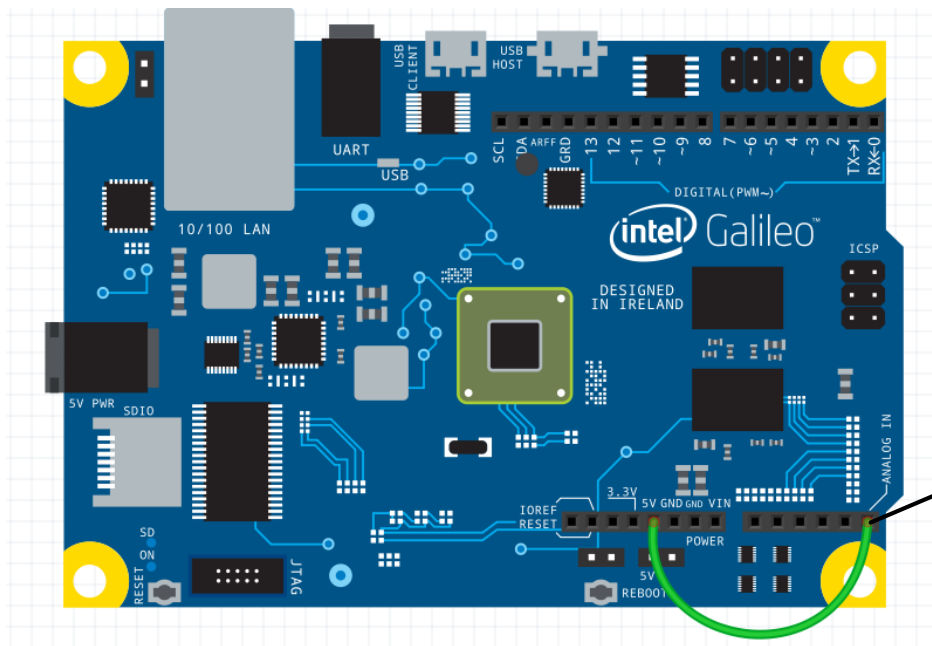
NOTE:

For any Wifi sketches. If your Wifi router has open security and has no password then just replace `WiFi.begin(ssid, pass);` -> `WiFi.begin(ssid);`
Use IP address provided by Wifi for your server, eg below is 192.168.1.19



Simple Web Server – Test Sketch

- Connect 5V to the pin A5 and we should see the value increase.
- Then switch jumper to all analog pins to test they are working
- Refresh the web page as needed to see changes on the server



Simple Web Server – Engineering Challenge 1

1 (for everyone)

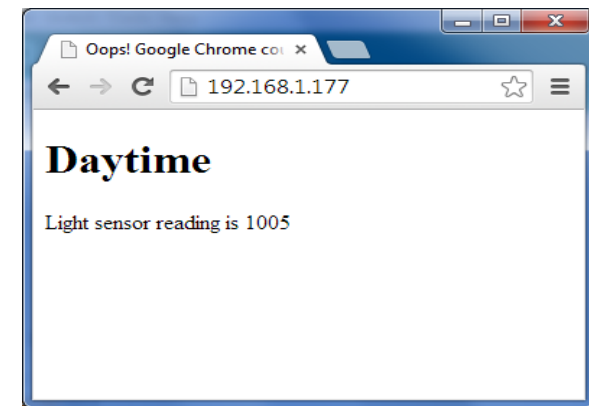
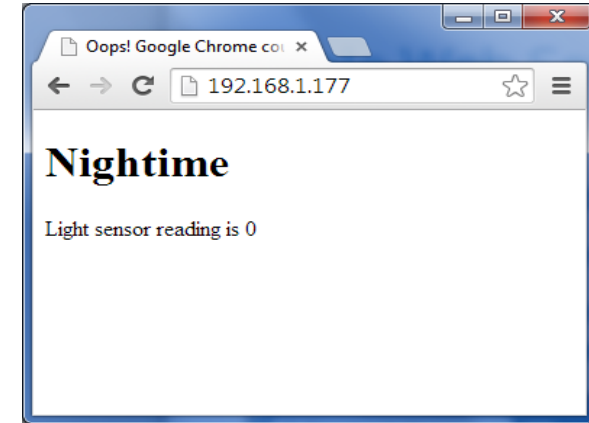
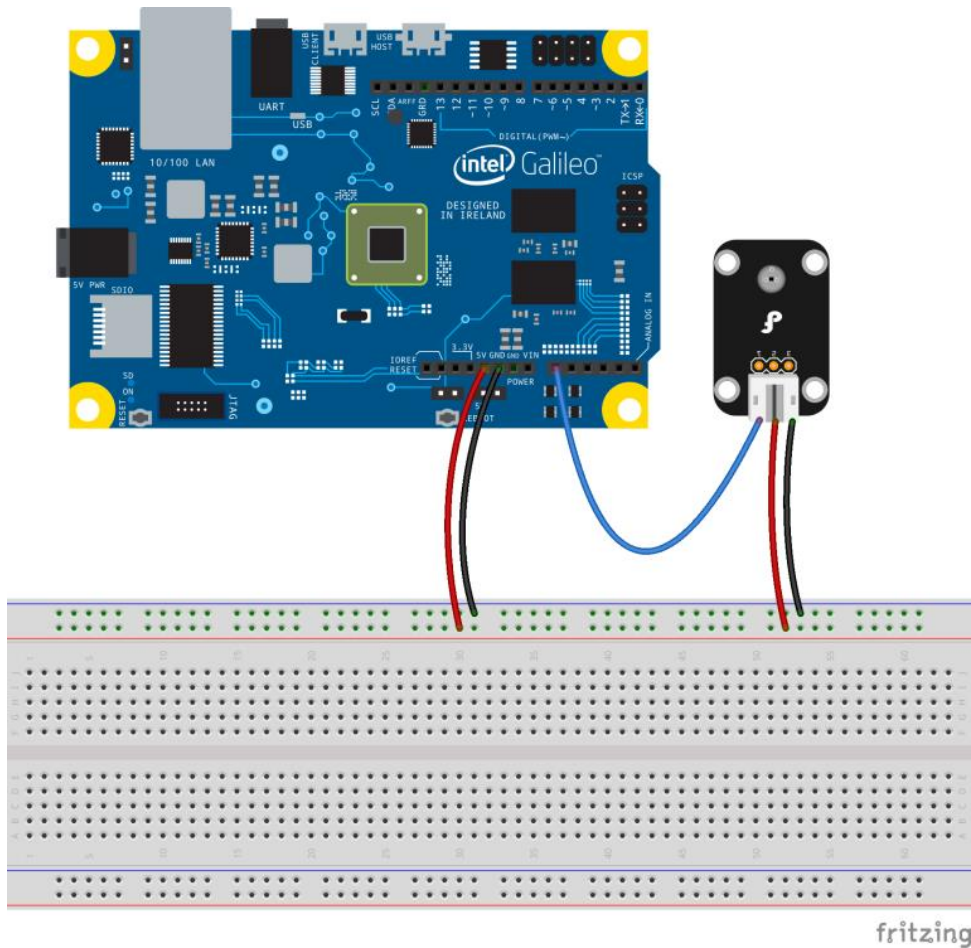
- File -> Save As... and make a new sketch called WebServerLightSensor.ino
- Refer/Refresh on Lesson “Sensors - Light Dependent Resistor”
- Change sketch to display in the browser whether it is night or day based in the Light sensor reading like next page

Troubleshooting:

I have found that sometimes I need to reload the sketch more than once for the web browser to connect to the server.

Simple Web Server – Engineering Challenge 1 ^{LAB}

Light Sensor Wiring Diagram



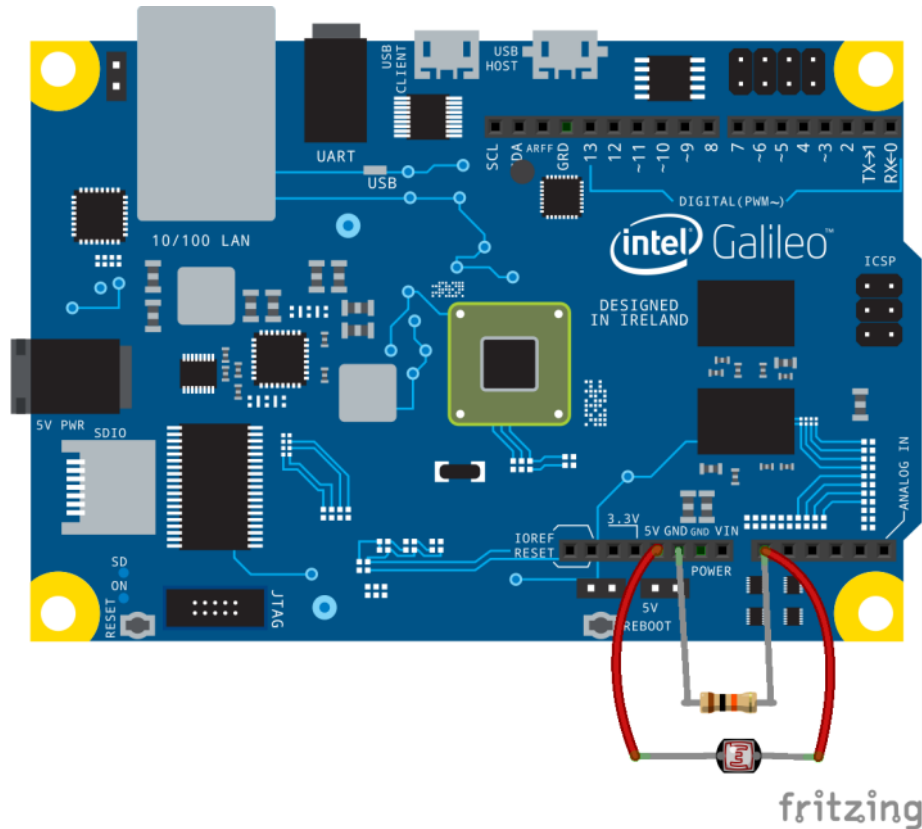
Project Files:

Ethernet Code: Lesson4-Connectivity\Section3-SimpleWebServer\WebServerLightSensor

Wifi Code: Lesson4-Connectivity\Section3-SimpleWebServer\WiFiWebServerLightSensor

Simple Web Server – LDR Sensor

- Alternative Light sensor wiring diagram



- LDR
- 220 Ohm Resistor

Simple Web Server – Engineering Challenge 2

2 (if you have time)

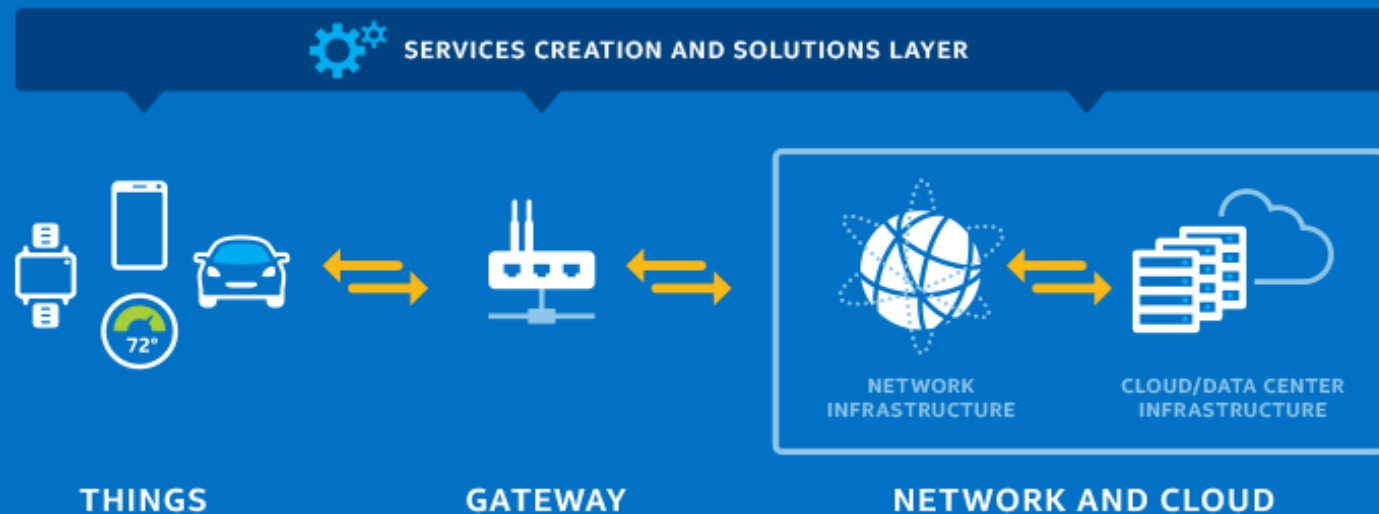
- Add your favourite sensor from the sensors lessons and label it appropriately

3 (if you have time)

- Explore the tutorials here and add a button to your webpage to switch LEDs on/off from the Galileo server
- <http://startingelectronics.com/tutorials/arduino/ethernet-shield-web-server-tutorial/>

Connectivity – Sensor IoT – Cloud

Internet of Things



Examples



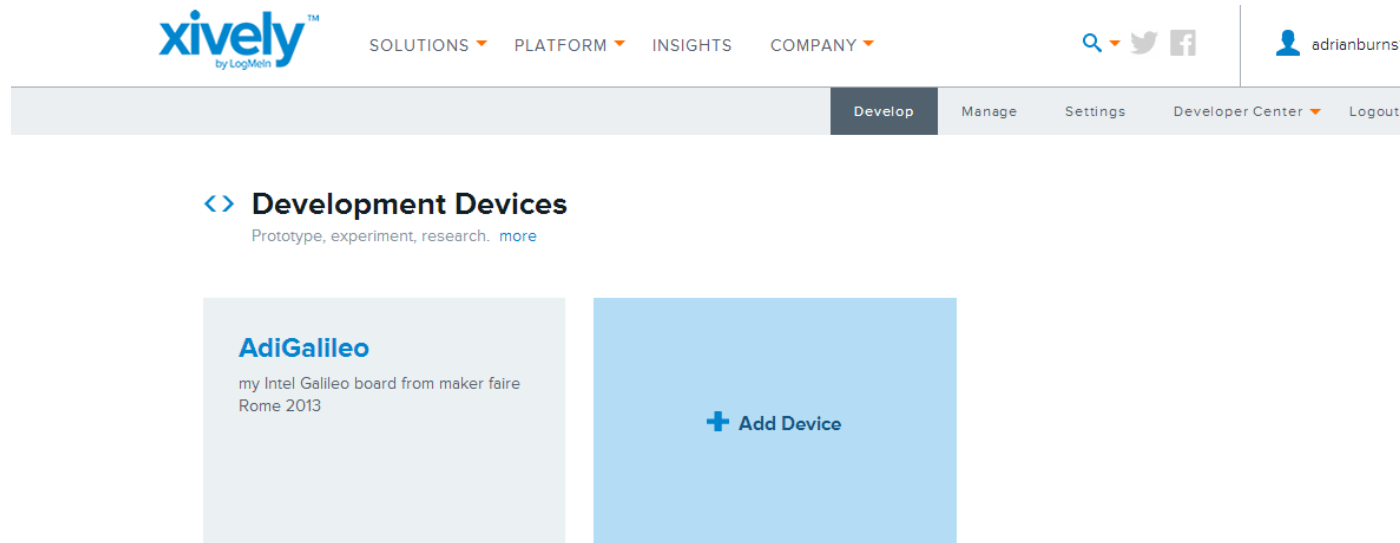
Sensor IoT – Upload Data to Cloud using Xively

- **Xively** (formerly known as Cosm and Pachube) offers an Internet of Things (IoT) platform as a service <http://en.wikipedia.org/wiki/Xively>
- Open sketch File -> Examples -> Ethernet -> PachubeClient
or
- Open sketch File -> Examples -> WiFi -> WiFiPachubeClient
- Reference <http://arduino.cc/en/Tutorial/PachubeClient>
- Now start looking at the sketch code but don't run sketch yet
- In this sketch Galileo will then connect to your account and upload sensor data every 10 seconds.

Sensor IoT – Setup Xively Account

Go to xivley.com and create a new account

Once in the Develop tab, press the 'Add Device' button.



You will be taken to the newly created Xively Workbench for your device, fill this out with the name of your device eg. **JohnGalileo**.

Then click on “Add device”

Sensor IoT – Add your device

1 Add your device like this

2 Then click

✓ Add Device

<> 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

3

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.

3 Add a Channel for analog pin 0

So just 1 stream for now

Call it "Light"

+ Add Channel

Sensor IoT – Modify Sketch



SOLUTIONS

PLATFORM

INSIGHTS

COMPANY



Develop

Manage

Settings

Developer Center

AdiGalileo

Public Device

Product ID A1MEI0uCsB9EGgioJ5PT
Product Secret ed0289e355dce6afd9a9683721e775c47b85cdd3
Serial Number PRTJYQXPENQQ
Activation Code 264dff44f286b1a28c9cd95f5185910b76fd7fc6

[Learn about the Develop stage](#)

Activated [Deactivate](#)

at 01-04-2014 15:07:22

Deploy

Feed ID 1704385900

[Feed URL](#) <https://api.xively.com/v2/feeds/1704385900>

[API Endpoint](#) <https://api.xively.com/v2/feeds/1704385900>

Channels

Last updated 3 hours ago

Graphs

sensor1

0

general data

Last updated 3 hours ago



Edit Delete

+ Add Channel

Request Log

API



Waiting for requests

Your requests will appear here as soon as we get them, you debug by clicking each individual request.

API Keys

Auto-generated AdiGalileo device key for feed 1704385900

LLf8g2trDHdFyxOLcUUtqQI5UykpPGTSqT4fb56nFJMBMtP

1

2

Sensor IoT – Modify Sketch

- Get below data from your Xively account and change sketch to match

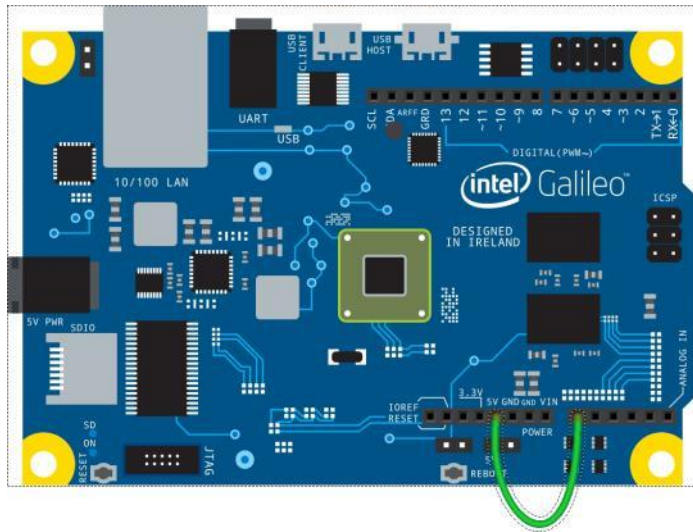
```
#define APIKEY "LLf8g2treddFyxOLciUTqtQl5UykpPGTSqT4fb56nFJMBMtP"
```

```
#define FEEDID 1204385900
```

- Each device on a network must have a unique mac address (line 38)
- Important Remove **“!client.connected() &&”** from line 93
- Line 93 now reads **“if((millis() - lastConnectionTime > postingInterval)) {“**
- Upload sketch
- Click on your device and view live data
- Data will refresh every 10 seconds

Sensor IoT – Test out

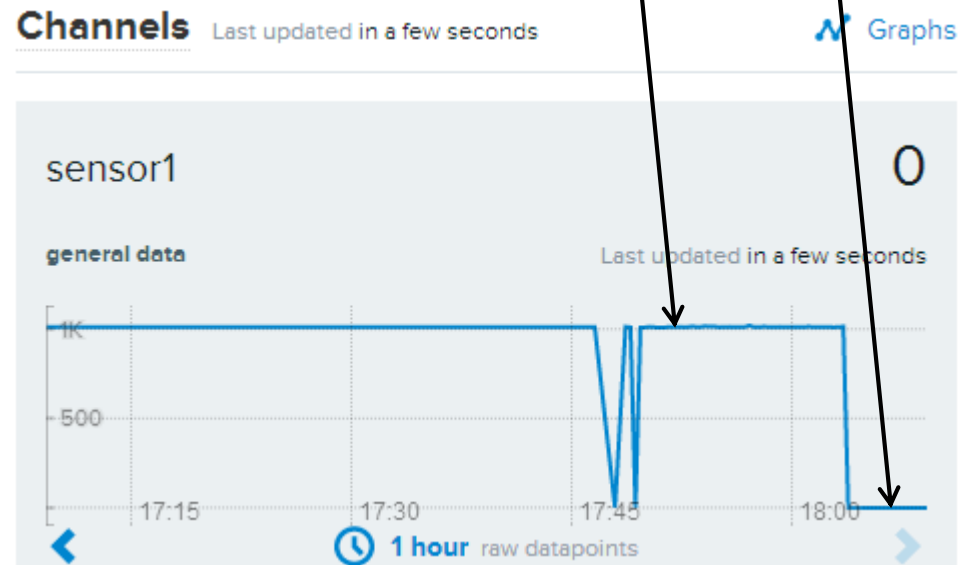
Use a Jumper wire to switch A0 from 0V to 5V and test that the data is being refreshed in the cloud.



fritzing

5V = 1024 on A0

0V = 0 on A0



**Swap jumper wire from 0v to 5V
To test data is refreshed on Xively**

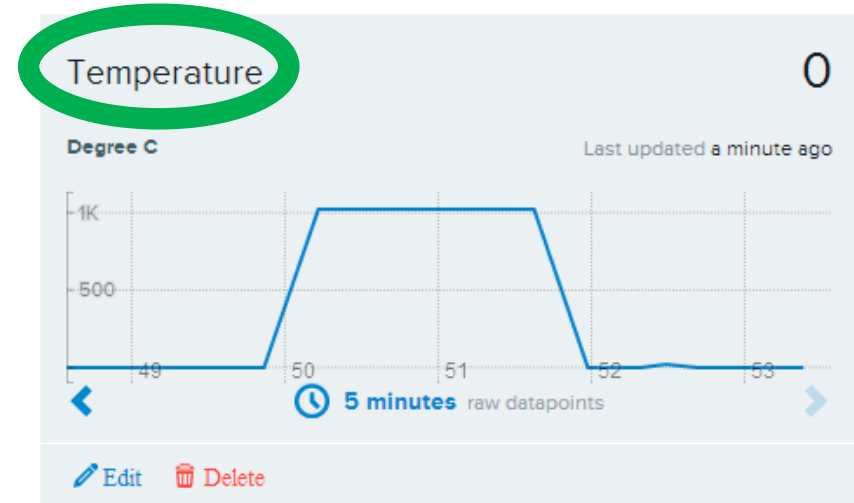
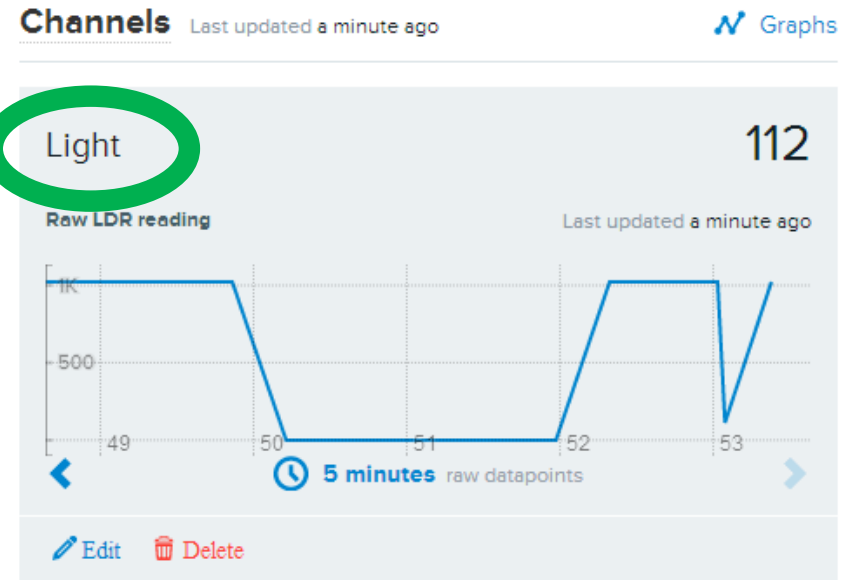
All other names and brands may be claimed as properties of others.

Sensor IoT – Engineering Challenge 1

- Extend existing sketch to upload Light data to the Cloud
- Refer back to the Sensors lesson on “Light Sensing” if needs be
- Save your work

Sensor IoT – Engineering Challenge 2

- Extend existing sketch
- Upload Light and Temperature data
- Light = Analog0
- Temperature = Analog1
- Now add more data channels !

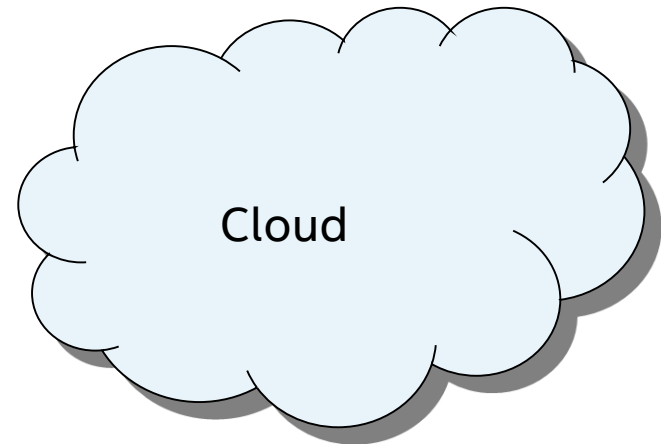
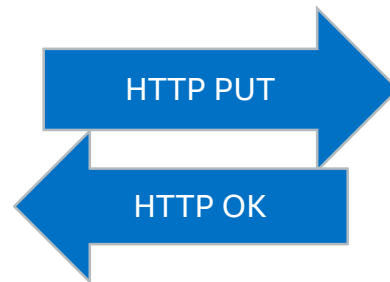
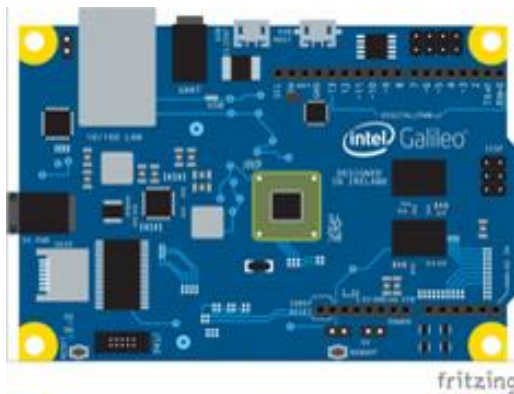




Ethernet Code: Lesson4-Connectivity\Section4-SensorIoT\IoTXively\Xively
Wifi Code: Lesson4-Connectivity\Section4-SensorIoT\WiFiloTXively\WiFiXively

Sensor IoT – Upload Data to Cloud using Xively

You are now uploading sensor data in realtime to the cloud and can view it from anywhere in the world



Half Duplex Data Transfer

Sensor IoT – Control Digital IO via Cloud

- Add a new data channel in Xively to your Galileo in the usual way

+ Add Channel

- This time we will control the on board Galileo LED via Pin 13
- Below (c_13_led) is what you call your new xively channel

Edit ChannelID

Tags	Units	Symbol
<input type="text" value="c_13_led"/>	<input type="text" value="e.g. Watts"/>	<input type="text" value="e.g. W"/>

Current Value

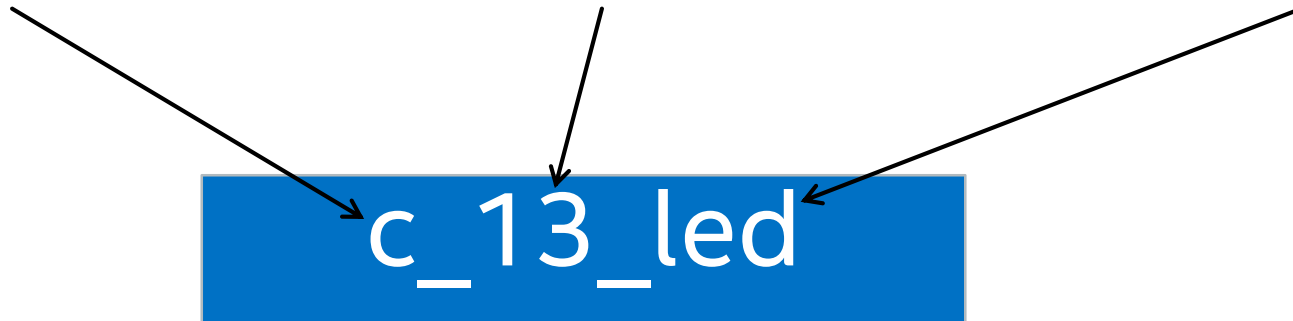
Save Channel Cancel

Sensor IoT – Control Digital IO via Cloud

Control IO

Digital IO number

Data label



- Turn LED on/off from xively and then add an external led and control with PWM (change value below from 0 to 255)
- If value is 0 or 1 digitalWrite() or if greater than 1 analogWrite()

A screenshot of a web interface. At the top, the text 'Current Value' is displayed. Below it is a text input field containing the number '0'. The input field is highlighted with a green border. Below the input field are two buttons: 'Save Channel' (in blue) and 'Cancel' (in grey).

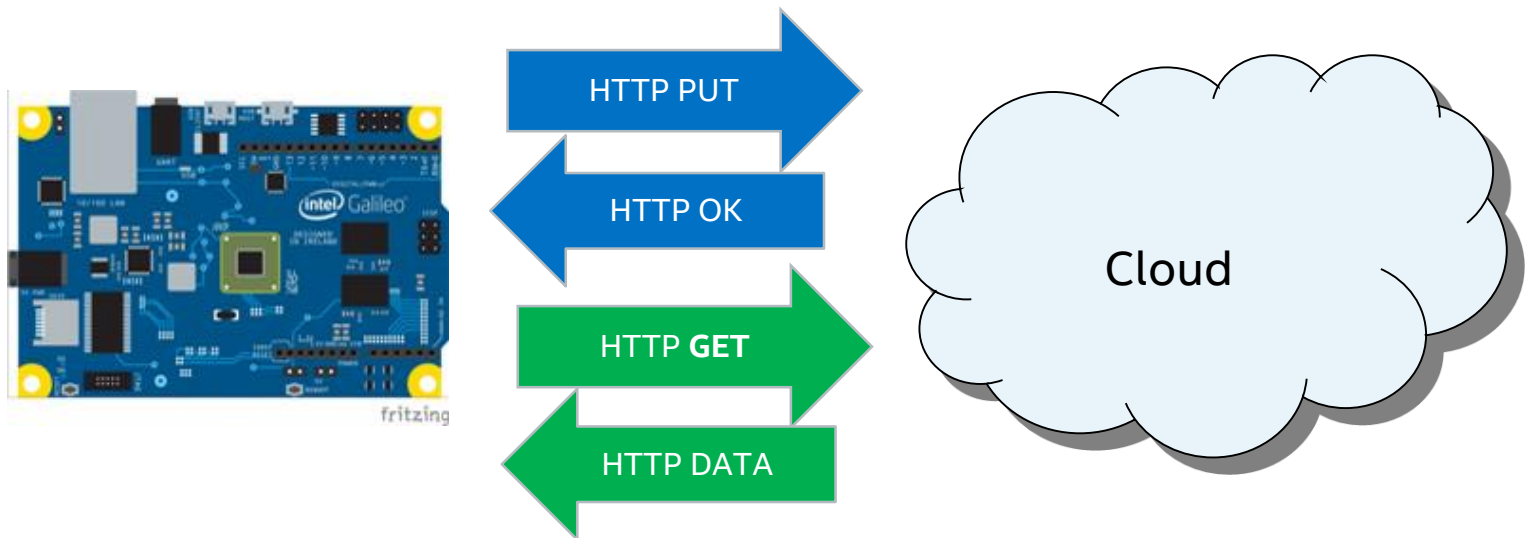
Project Files:

Ethernet Code: Lesson4-Connectivity\Section4-SensorIoT\IoTXively\XivelyLightTemp

Wifi Code: Lesson4-Connectivity\Section4-SensorIoT\WiFiloTXively\WiFiXivelyLightTemp

Sensor IoT – Control Galileo via Cloud

We used a HTTP PUT message to send Light/Temp data to the cloud and now we can use HTTP GET get control data from cloud

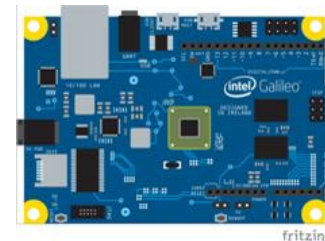
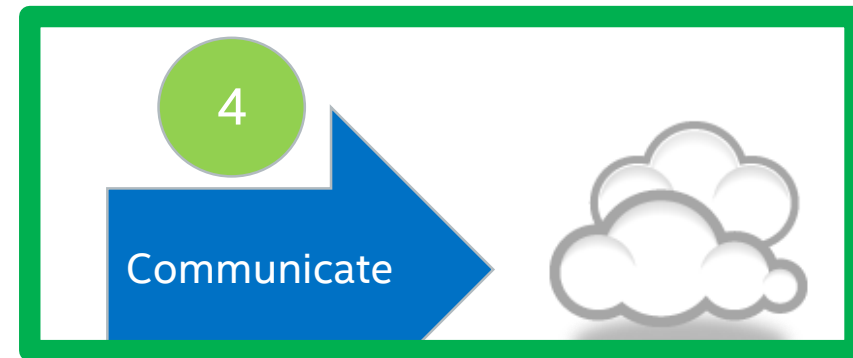
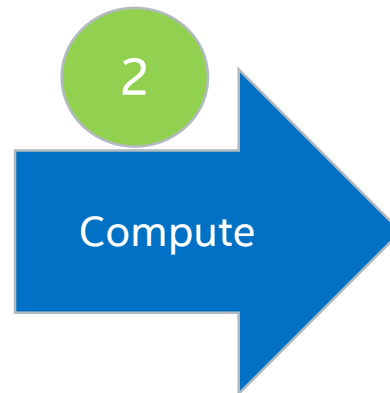
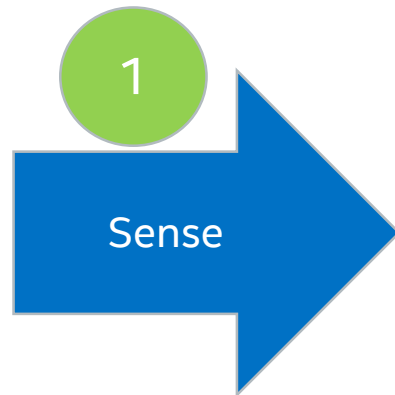
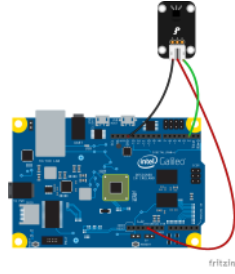


Full Duplex Data Transfer

Connectivity – Sensor IoT Email/SMS

Sensor IoT – Trigger Events - Reminder

Sensing Motion



Sensor IoT – Trigger Apps from Galileo Events

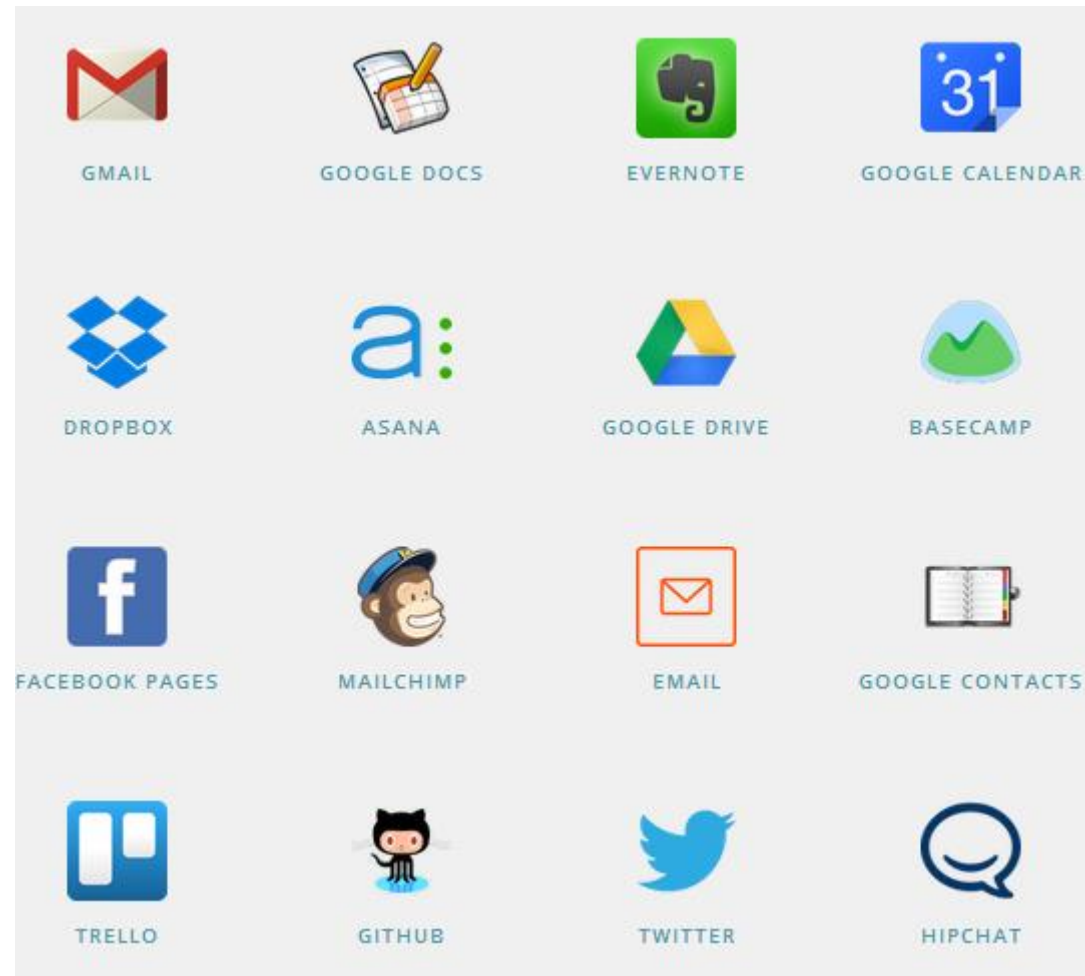
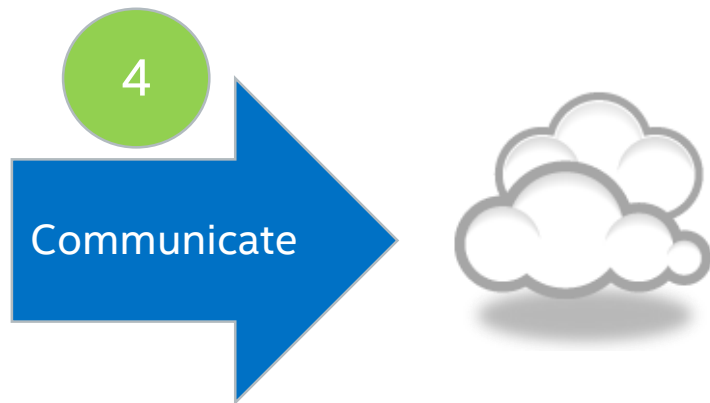


Image Source: <http://www.zapier.com>

All other names and brands may be claimed as properties of others.

Sensor IoT – Trigger Apps from Galileo Events

- How about being told when something happens on Galileo?
- Don't reinvent the wheel
- Read the following for Zapier use right from Xively

<https://xively.com/dev/tutorials/zapier/>

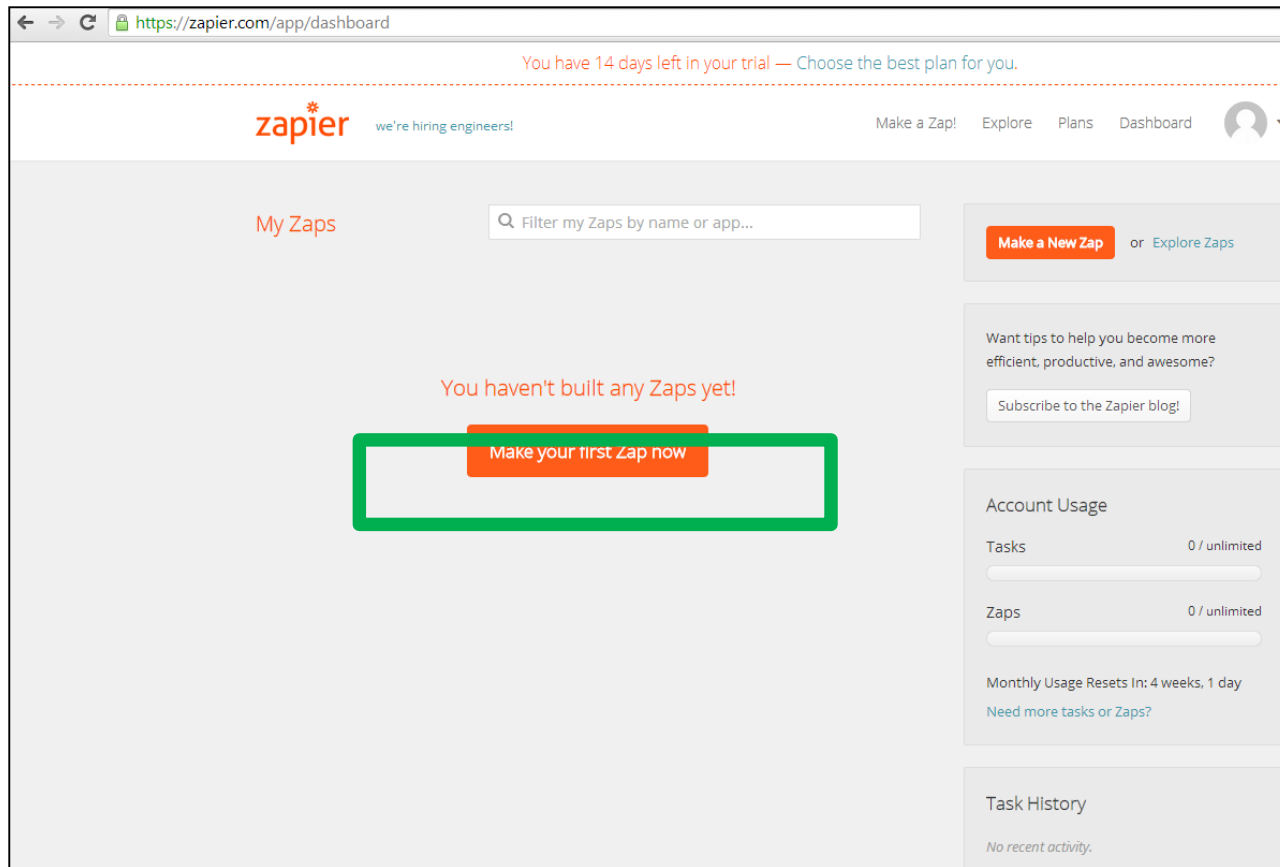
- Read the following for ideas on great ways to use Zapier

<http://zapier-downloads.s3.amazonaws.com/ebook-101-ways-use-zapier.pdf>

Engineering Challenge 1 - Send Email

Every time your A0 sensor pin goes below 200, email yourself

1. Open zapier.com and create an account
2. Click Make your first zap now



All other names and brands may be claimed as properties of others.

Engineering Challenge 1 - Send Email


3. Choose the setting below if you use Gmail (if not setup a gmail a/c)
4. Select continue to get the webhook URL

The screenshot shows the Zapier app editor interface. At the top, the URL is <https://zapier.com/app/editor/1282784>. A banner at the top right says "You have 14 days left in your trial — Choose the best plan for you." The Zapier logo and "we're hiring engineers!" are on the left. Navigation links "Make a Zap!", "Explore", "Plans", and "Dashboard" are on the right. The main section is titled "1 Choose a trigger and action". Below this, it says "Need Inspiration? See existing Zap templates or explore the Zapbook." There are two rows of dropdown menus. The first row has "Web Hook" selected for the trigger and "Gmail" selected for the action. The second row has "Catch Hook" selected for the trigger and "Send Email" selected for the action. Handwritten red text "WHEN THIS HAPPENS ..." with an arrow points to the "Web Hook" dropdown. Another handwritten red text "... DO THIS" with an arrow points to the "Send Email" dropdown. A red "Continue" button is at the bottom left.

Engineering Challenge 1 - Send Email

5. Copy the webhook URL below

2

Select a  Web Hook account

We've generated a custom webhook URL for you to send requests to.

When data is sent via requests to this webhook URL, we will attempt to interpret various serialization formats including form-encoded, JSON and XML. [Learn more here.](#)


Use this Webhook URL:


Copy to clipboard

Continue

6. Select your Gmail account, click continue

3

Select a  Gmail account

 adrianburns1@gmail.com's Gmail account

▼

✓ Account is working.

Continue

Engineering Challenge 1 - Send Email

7. Construct the automatic email to say what you want, click continue

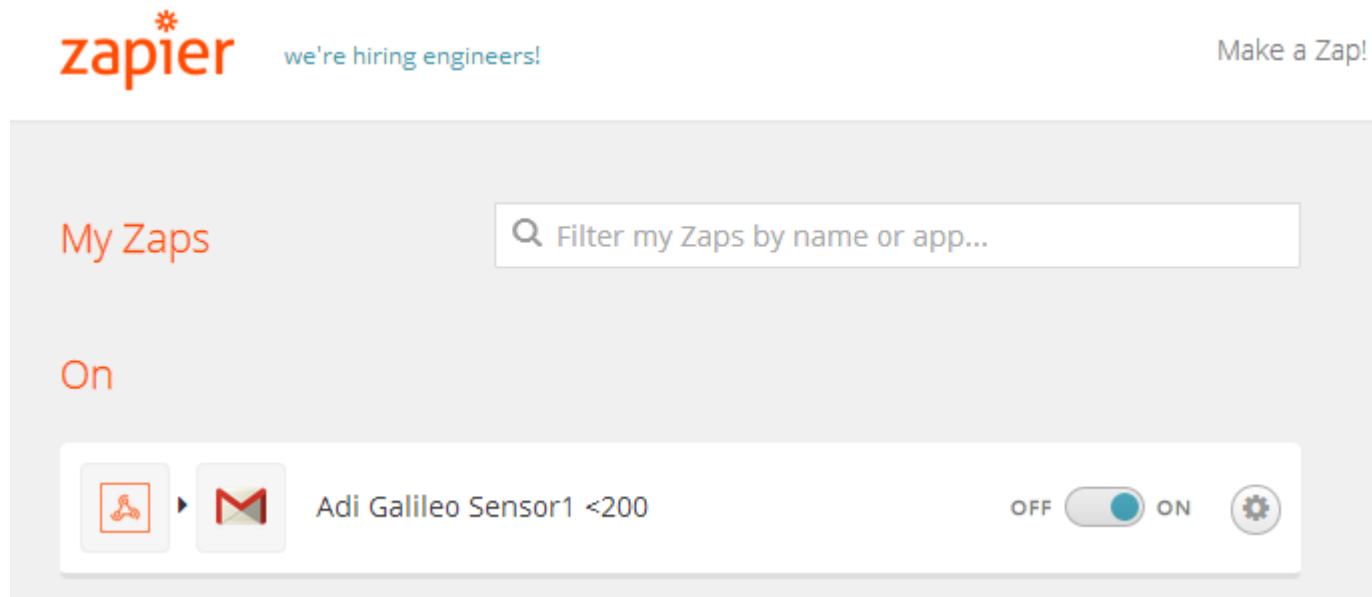
The screenshot shows the Zapier email editor interface for a Zap titled "Match up Web Hook Hook to Gmail Email". The interface is divided into several sections for configuring the email:

- To (required):** "Who will this email be sent to?" The field contains "adrian.burns@intel.com".
- Cc (optional):** "Who should be cc'd on this email?" The field contains "adrianburns1@gmail.com".
- Bcc (optional):** "Who should be bcc'd on this email?" The field is empty.
- From Name (optional):** "This will be your 'display name' when sending." The field contains "My Galileo Sensor is < 200".
- From Email (optional):** "From email address (if empty, we will use the default). Please note, to use an address besides the default you will need to add and verify it first." The field is empty.

Each field has an "Insert fields" button and a plus/minus icon to the right.


Engineering Challenge 1 - Send Email

8. Ensure your zap is switched on like below



Engineering Challenge 1 - Send Email

9. Construct the automatic email to say what you want, click continue
10. Now go back to xively and add the webhook URL and save trigger

2 Select a  Web Hook account

We've generated a custom webhook URL for you to send requests to.

When data is sent via requests to this webhook URL, we will attempt to serialize in various formats including form-encoded, JSON and XML. [Learn more](#)

Use this Webhook URL:

`https://zapier.com/hooks/catch/cscnc/`

[Continue](#)

Triggers

Edit Trigger

Channel Where the trigger will be attached

sensor1 ▼

Condition When to fire the trigger

< ▼ 200

HTTP POST URL

`https://zapier.com/hooks/catch/cscnc/`

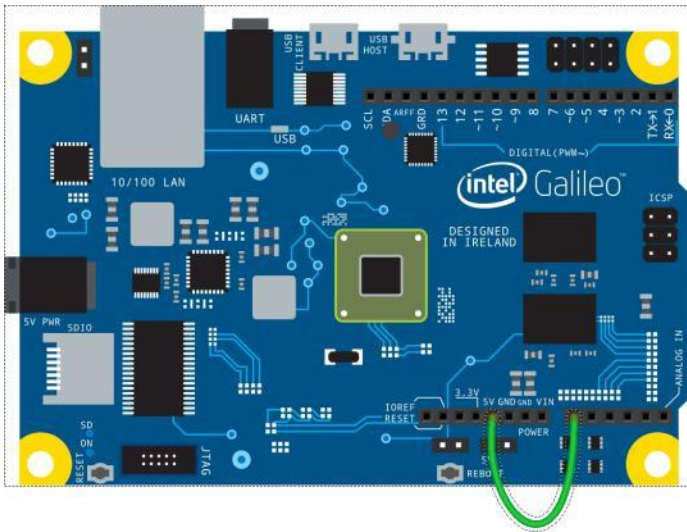
✓ Save Trigger

Cancel

+ Add Trigger

Engineering Challenge 1 - Send Email

11. Now switch your A0 to 5V and leave it for a few minutes
12. Now pull the jumper wire to bring A0 value from 1023 to 0
13. This bring vale to below 200 so you should receive email



fritzing

**Swap jumper wire from 0v to 5V
To test data is refreshed on Xively**

Reply to Adi's Galileo generated email

Inbox x

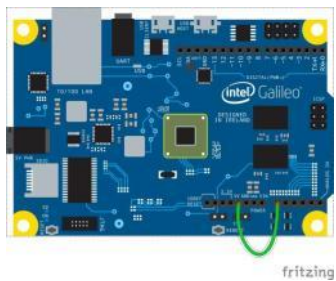


My Galileo sensor1 signal is < youremail @gmail.com>
to someone.else@gmail.com

Body of reply

Engineering Challenge 1 - Conclusion

Sensing A0 Value



Send Events
To Cloud

zapier

+



Image Source: <http://www.xively.com>

Image Source: <http://www.zapier.com>

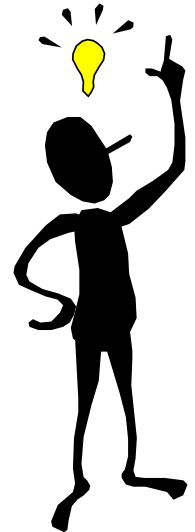
Email



SMS

Facebook

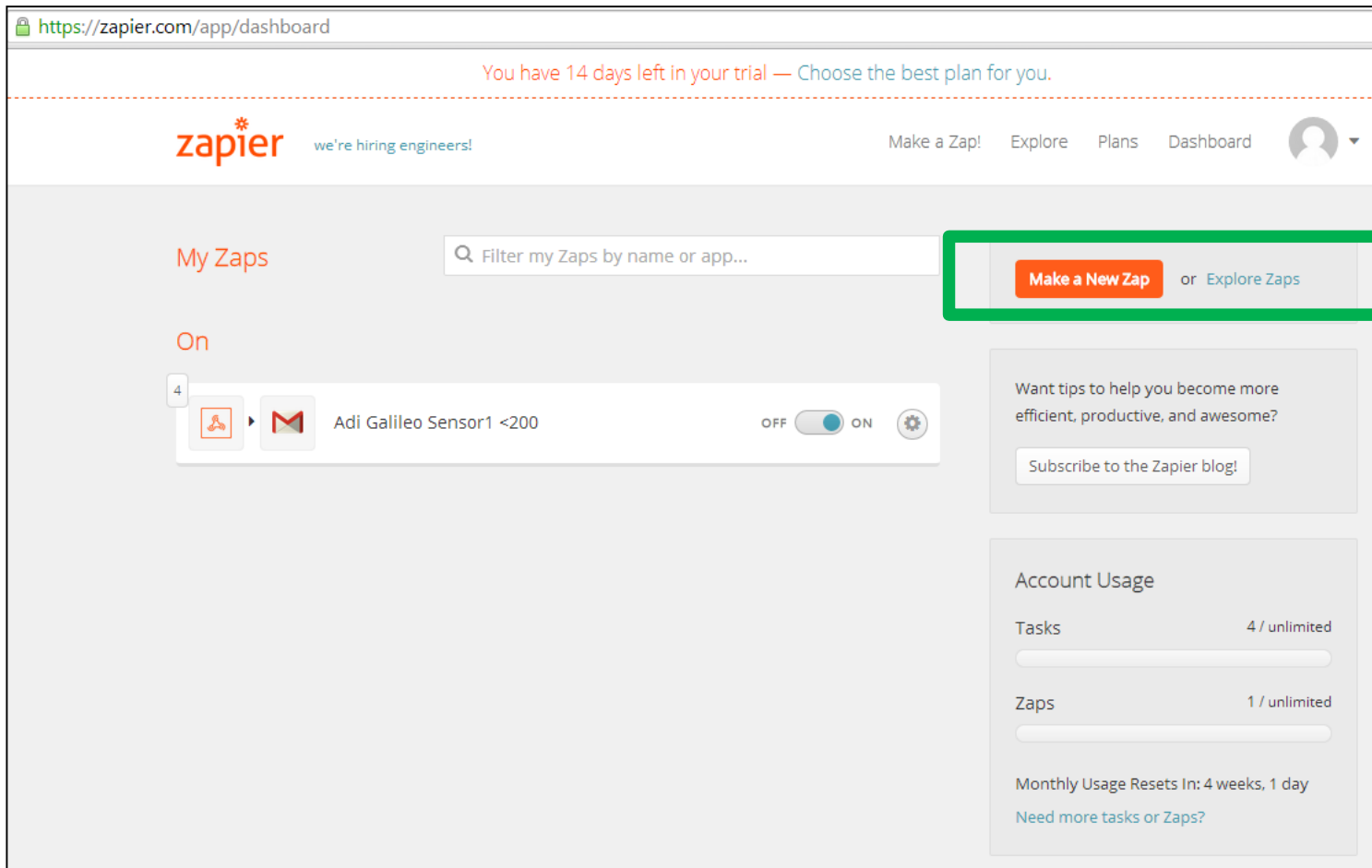
Twitter



Engineering Challenge 2 – Send SMS

Every time your A0 sensor pin goes above 1000, SMS yourself

1. Open zapier.com and make a new zap

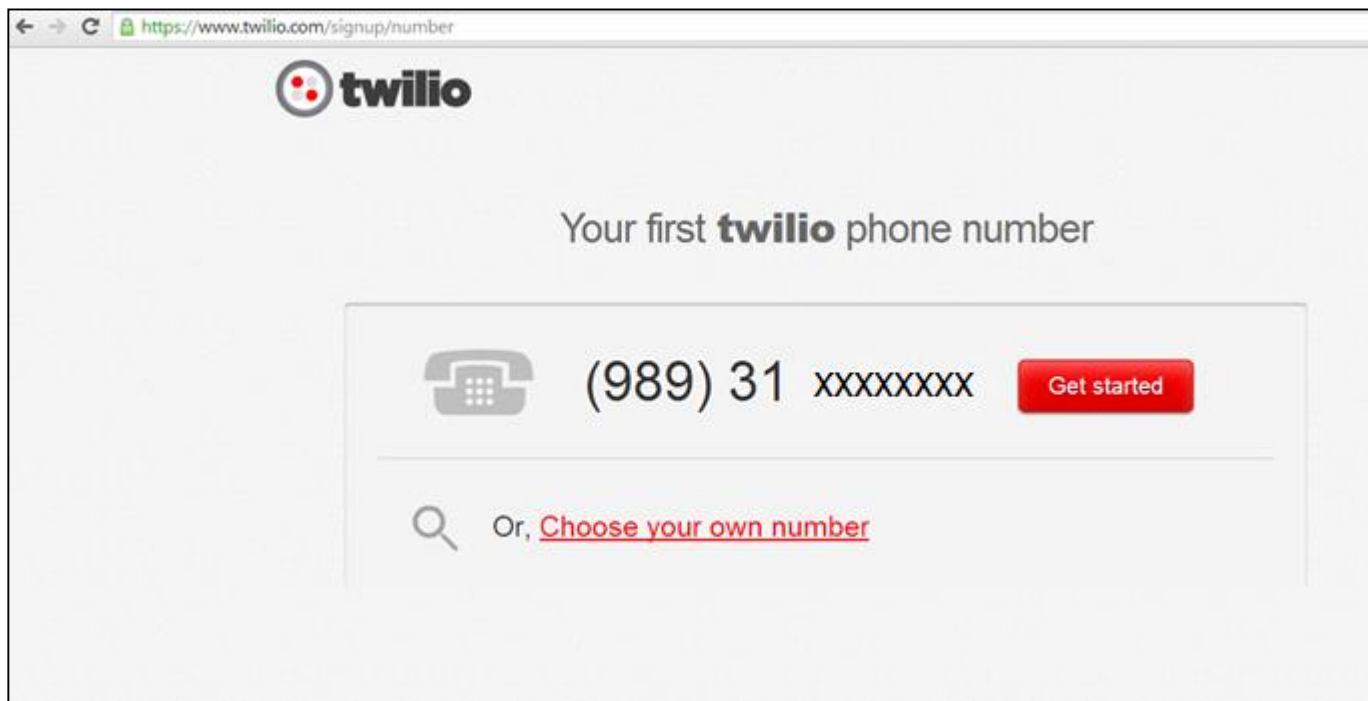


All other names and brands may be claimed as properties of others.

Engineering Challenge 2 – Send SMS

Every time your A0 sensor pin goes above 1000, SMS yourself

2. This time choose Twilio instead of an Email app
3. Go to [twilio.com](https://www.twilio.com), create an account and get your number like below



Engineering Challenge 2 – Send SMS

4. Now go back to zapier.com and create a new webhook like below

1 Choose a trigger and action

Need inspiration? [See existing Zap templates](#) or [explore the Zapbook](#).

Web Hook

Catch Hook

Twilio

Send SMS

WHEN THIS HAPPENS ...

... DO THIS

This screenshot shows the first step of creating a Zap on Zapier. It has a title '1 Choose a trigger and action' and a link for inspiration. There are two rows of dropdown menus. The first row has 'Web Hook' selected for the trigger and 'Twilio' for the action. The second row has 'Catch Hook' for the trigger and 'Send SMS' for the action. Handwritten orange arrows point from the text 'WHEN THIS HAPPENS ...' to the 'Web Hook' dropdown, and from '... DO THIS' to the 'Send SMS' dropdown.

5. Match it up with your newly created Twilio account

3 Select a Twilio account



adrianburns1@gmail.com's Twilio account

Test this Account

This screenshot shows the third step of creating a Zap on Zapier. It has a title '3 Select a Twilio account'. Below the title is a dropdown menu showing 'adrianburns1@gmail.com's Twilio account' and a button labeled 'Test this Account'.

Engineering Challenge 2 – Send SMS

6. Now fill in your Twilio number (From number) and your own phone number (To Number) and test it out like below, save and continue

5 Match up  Web Hook Hook to  Twilio SMS

From Number (required)

(989) 31XXXXXX

To Number (required)

Required number format: 15554443333 or 44111222333. Can send to multiple numbers with commas:
15554443333,44111222333

35386881XXXXXX

Message (required)

Text that Galileo sensor1 pin is >1000


Split Message? ☒ (optional)

Should we split the message and send multiple SMS if the message is more than 160 characters?
Otherwise we will truncate and only send one message.

no

Engineering Challenge 2 – Send SMS

7. Now go back to Xively and add the webhook URL and save trigger
8. Now switch your A0 to 0V and leave it for a minute
9. Now put in the jumper wire to bring A0 value 0 to 1023
10. You should get your SMS alert ??

2 Select a  Web Hook account

We've generated a custom webhook URL for you to send requests to.

When data is sent via requests to this webhook URL, we will attempt to serialize in various formats including form-encoded, JSON and XML. [Learn more](#)

Use this Webhook URL:

`https://zapier.com/hooks/catch/cscnc/`

[Continue](#)

Triggers

Edit Trigger

Channel Where the trigger will be attached

Condition When to fire the trigger

HTTP POST URL

[✓ Save Trigger](#) [Cancel](#)

[+ Add Trigger](#)

Engineering Challenge 2 - Conclusion

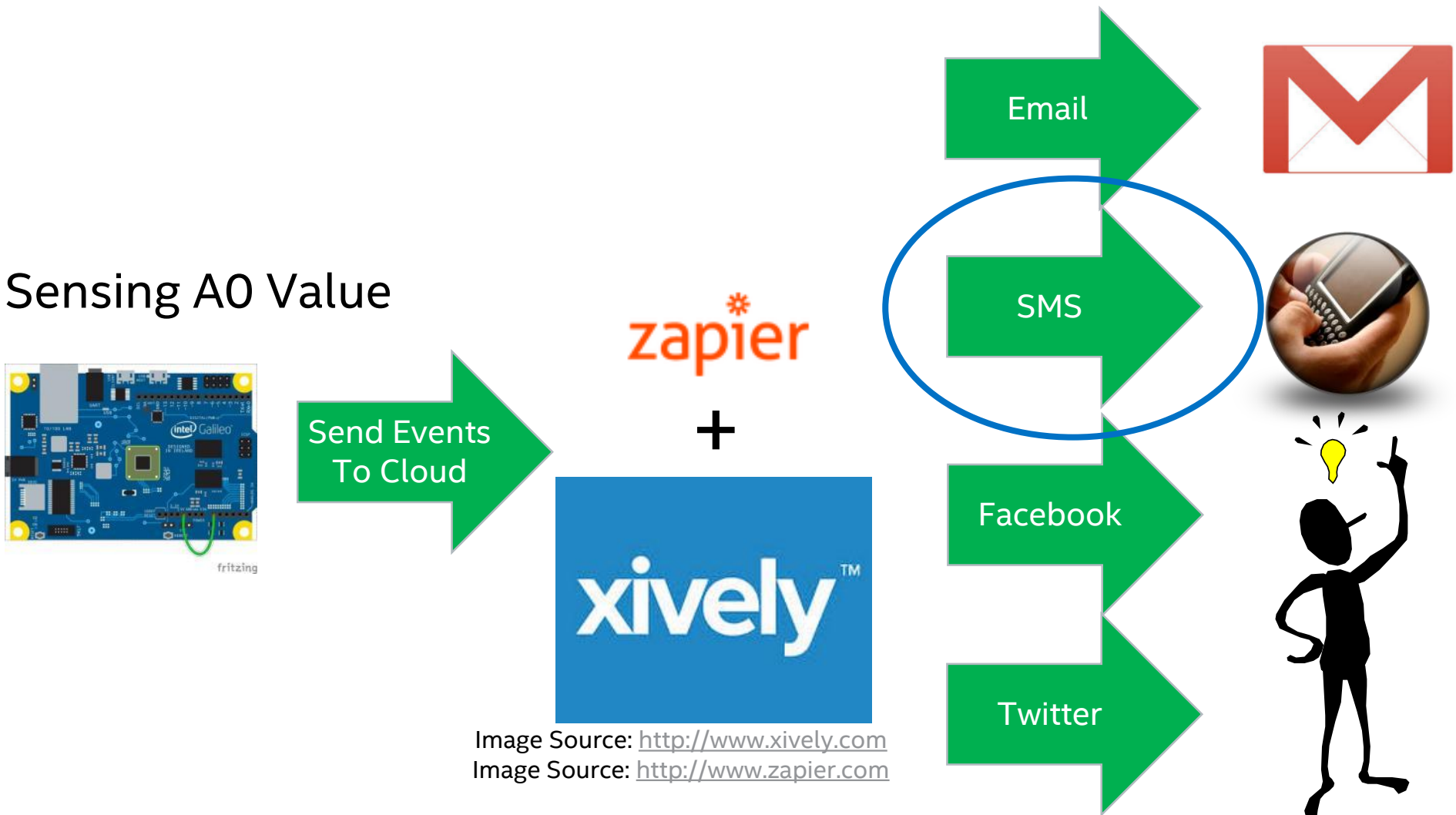


Image Source: <http://www.xively.com>

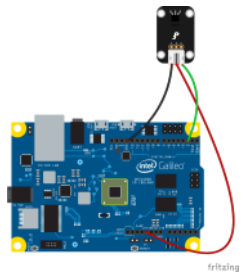
Image Source: <http://www.zapier.com>

Now you can send Emails/SMS
based on what Galileo is doing
😊

Sensors – Home Security System

Send Email/SMS on Motion Detection

Sensing Motion



Send Events
To Cloud

zapier
+



Image Source: <http://www.xively.com>

Image Source: <http://www.zapier.com>

Email



SMS



Facebook

Twitter

Home Security System 1 – Touch enabled

Using your own circuit design and Arduino sketch, design a solution that solves the following challenges.

Use previous Lab example as needed for reference

Challenge:

Build a motion detection system that is enabled/disabled through the touch sensor. After its enabled it will email or SMS you when a motion event is received rather than sending them to the SD card like the Sensor Logger.

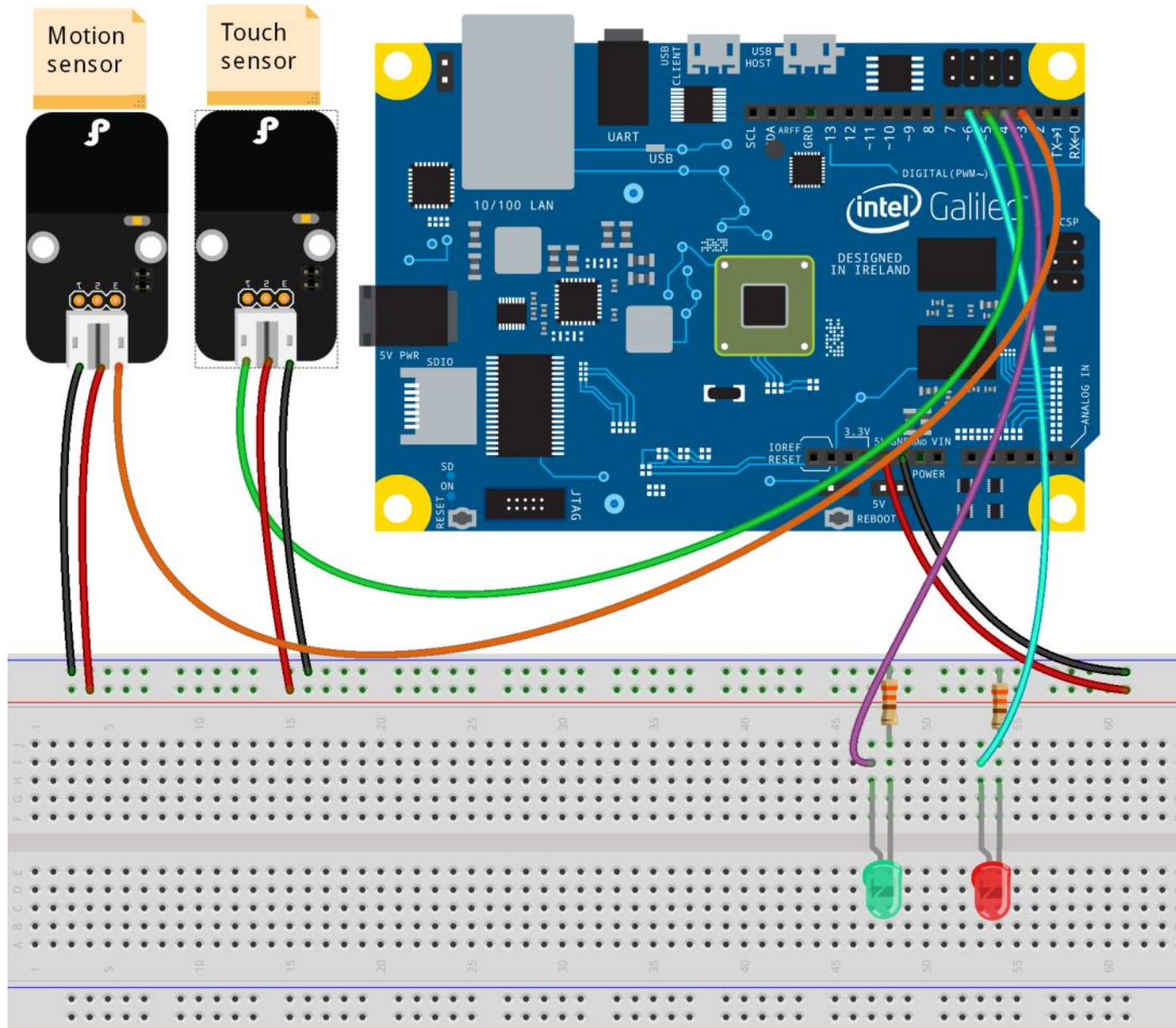
Home Security System 2 – Enabled via Web

Using your own circuit design and Arduino sketch, design a solution that solves the following challenges.

Use previous Lab example as needed for reference

Challenge:

Build on the previous section but this time the Security System is remotely enable (web). After its enabled it will email or SMS you when a motion event is received.



Congratulations 😊

Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: <http://www.intel.com/design/literature.htm> Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, visit Intel Performance Benchmark Limitations

All products, computer systems, dates and figures specified are preliminary based on current expectations, and are subject to change without notice.

Celeron, Intel, Intel logo, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel SpeedStep, Intel XScale, Itanium, Pentium, Pentium Inside, VTune, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Intel® Active Management Technology requires the platform to have an Intel® AMT-enabled chipset, network hardware and software, as well as connection with a power source and a corporate network connection. With regard to notebooks, Intel AMT may not be available or certain capabilities may be limited over a host OS-based VPN or when connecting wirelessly, on battery power, sleeping, hibernating or powered off. For more information, see <http://www.intel.com/technology/iamt>.

64-bit computing on Intel architecture requires a computer system with a processor, chipset, BIOS, operating system, device drivers and applications enabled for Intel® 64 architecture. Performance will vary depending on your hardware and software configurations. Consult with your system vendor for more information.

No computer system can provide absolute security under all conditions. Intel® Trusted Execution Technology is a security technology under development by Intel and requires for operation a computer system with Intel® Virtualization Technology, an Intel Trusted Execution Technology-enabled processor, chipset, BIOS, Authenticated Code Modules, and an Intel or other compatible measured virtual machine monitor. In addition, Intel Trusted Execution Technology requires the system to contain a TPMv1.2 as defined by the Trusted Computing Group and specific software for some uses. See <http://www.intel.com/technology/security/> for more information.

†Hyper-Threading Technology (HT Technology) requires a computer system with an Intel® Pentium® 4 Processor supporting HT Technology and an HT Technology-enabled chipset, BIOS, and operating system. Performance will vary depending on the specific hardware and software you use. See www.intel.com/products/ht/hyperthreading_more.htm for more information including details on which processors support HT Technology.

Intel® Virtualization Technology requires a computer system with an enabled Intel® processor, BIOS, virtual machine monitor (VMM) and, for some uses, certain platform software enabled for it. Functionality, performance or other benefits will vary depending on hardware and software configurations and may require a BIOS update. Software applications may not be compatible with all operating systems. Please check with your application vendor.

* Other names and brands may be claimed as the property of others.

Other vendors are listed by Intel as a convenience to Intel's general customer base, but Intel does not make any representations or warranties whatsoever regarding quality, reliability, functionality, or compatibility of these devices. This list and/or these devices may be subject to change without notice.

Copyright © 2014, Intel Corporation. All rights reserved.



Backup Section