



wemake.cc
#makerspace
#fablab

#opencare

Workshop DIY/DIT Care

Caring with Open Source IoT: Peer-to-Peer Practices for Collaborative Upcycling in Care



www.opencare.cc
www.wemake.cc/opencare



wemake.cc
#makerspace
#fablab

Opening Technologies for Smart Communities

WeMake is an innovative enterprise based in Milan providing a series of services and training in the field of digital and traditional manufacturing and access to a fully equipped **Fab Lab in the city of Milan.**

We are fostering the development of a new model of designer-producer and agile company by facilitating the rapid iteration of **co-design solutions** and the on-demand production of smart **physical/digital artifacts**.



open care

A community working to make open and collaborative health and care solutions



université
de BORDEAUX



Horizon 2020
European Union funding
for Research & Innovation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688670.



Care AND Community: a long story

Yesterday: Families/Communities
as main providers of **informal care**

Today: Welfare

Tomorrow: ?

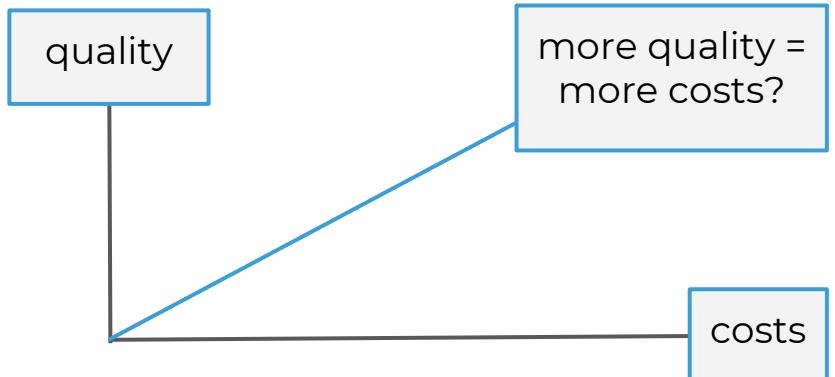


XIX century Welfare State:

Social and Health services provided from the State.

- > Professionalized care
- > One-size-fits-all approach
- > Top-down model

Crisis of Welfare State



What does open+care mean?

CARE

Care referred to a special kind of human interaction: someone (care giver) gives attention to, and takes action for, someone else (care receiver).

Care is also the products and services part of the system which, in a given time and place, enables the care activities.



OPEN

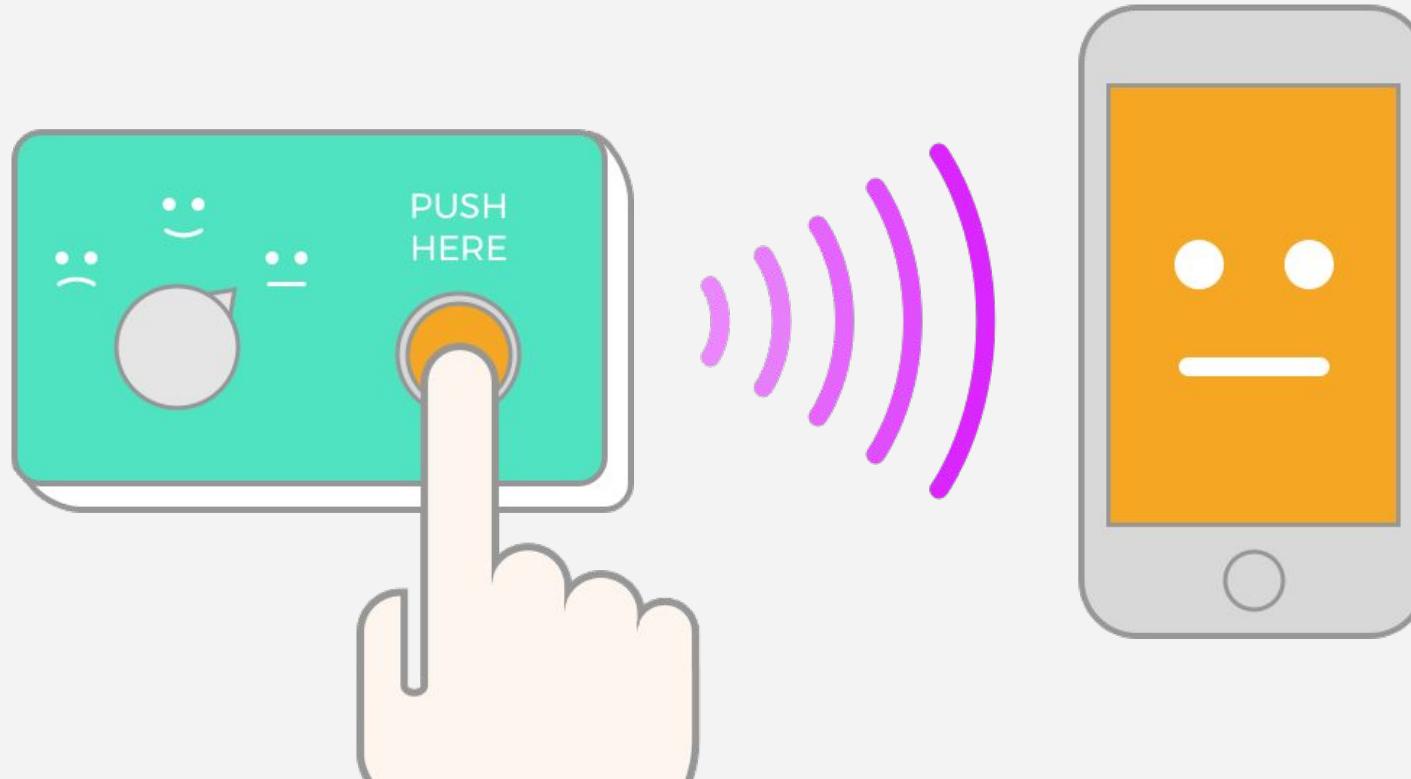
A system is **open** when different actors have the possibility to play a role. Experts and non experts.

A system is **open** when the information are totally visible, accessible and transferable to other systems.

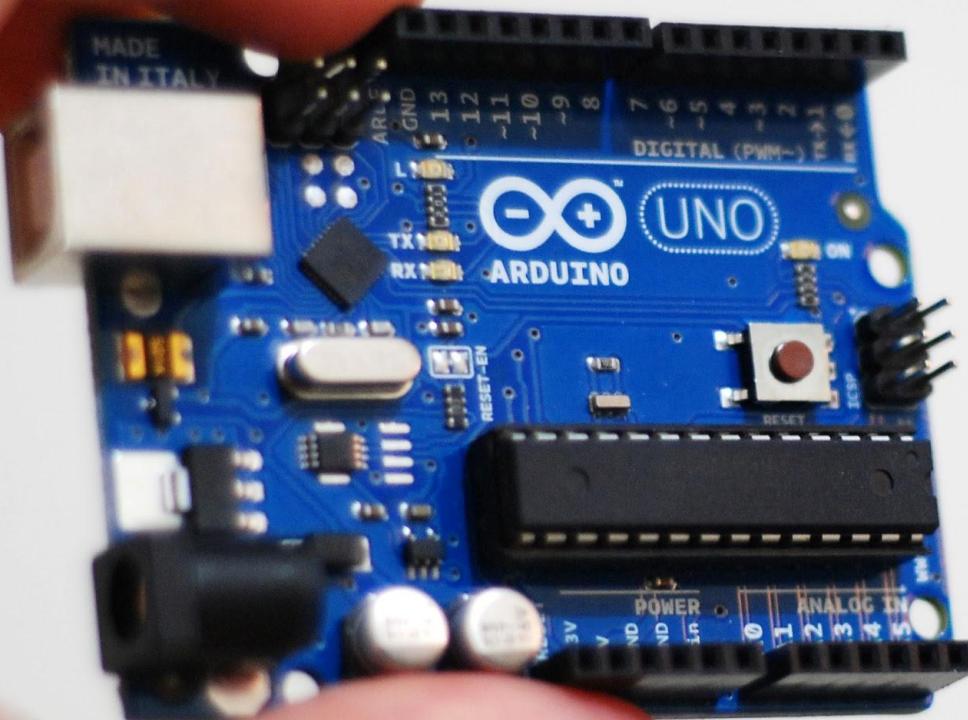
What are we gonna
do today?



A remote monitoring device...



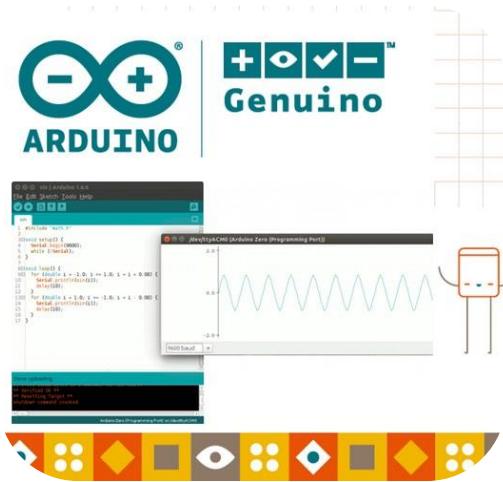
...with Arduino!



What is Arduino?



hardware



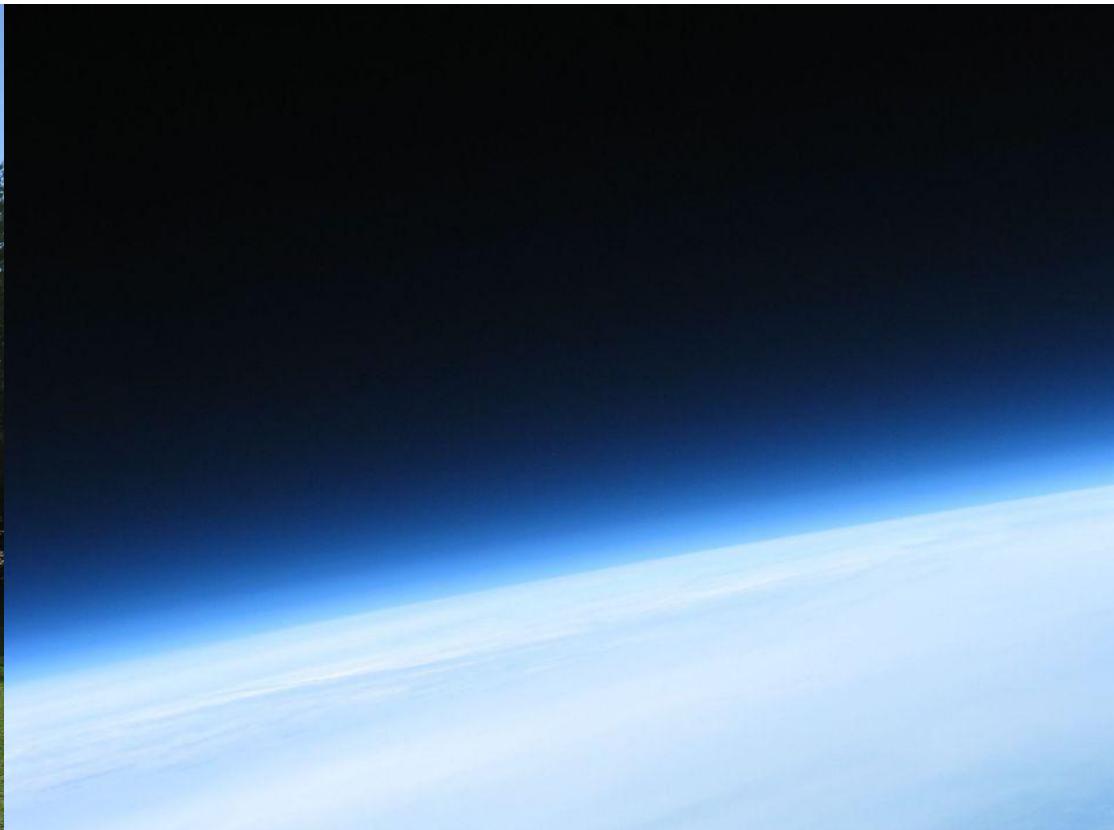
software



community



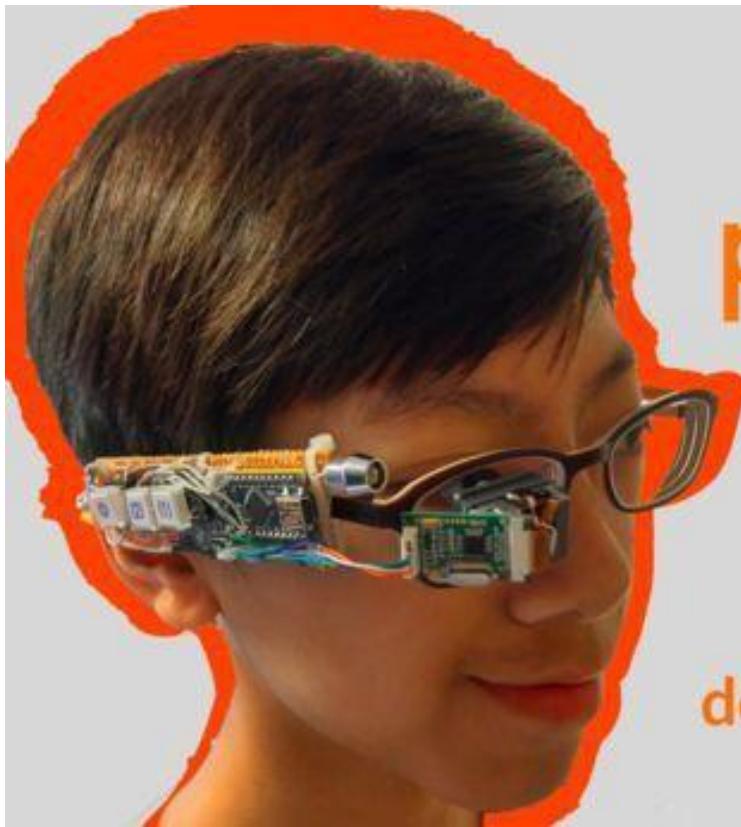
With Arduino you can... go to space!



...play and learn



...invent crazy amazing things!



pedosaglass

An Arduino-Based
Smart Glasses

developed by jordan fung



Internet of Things



Maybe you have used one of these



Or one of these



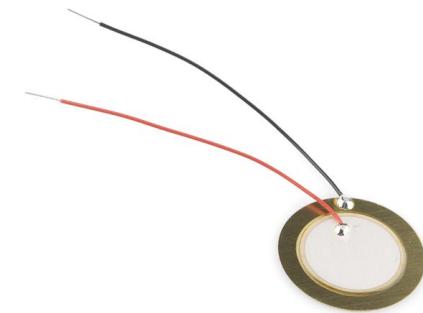
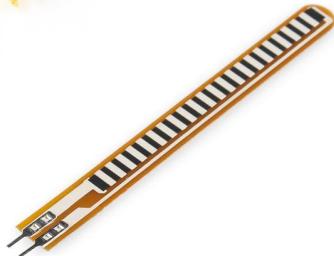
It's very powerful when applied to the care field



IoT made easy... IFTTT!



SENSORS



ACTUATORS



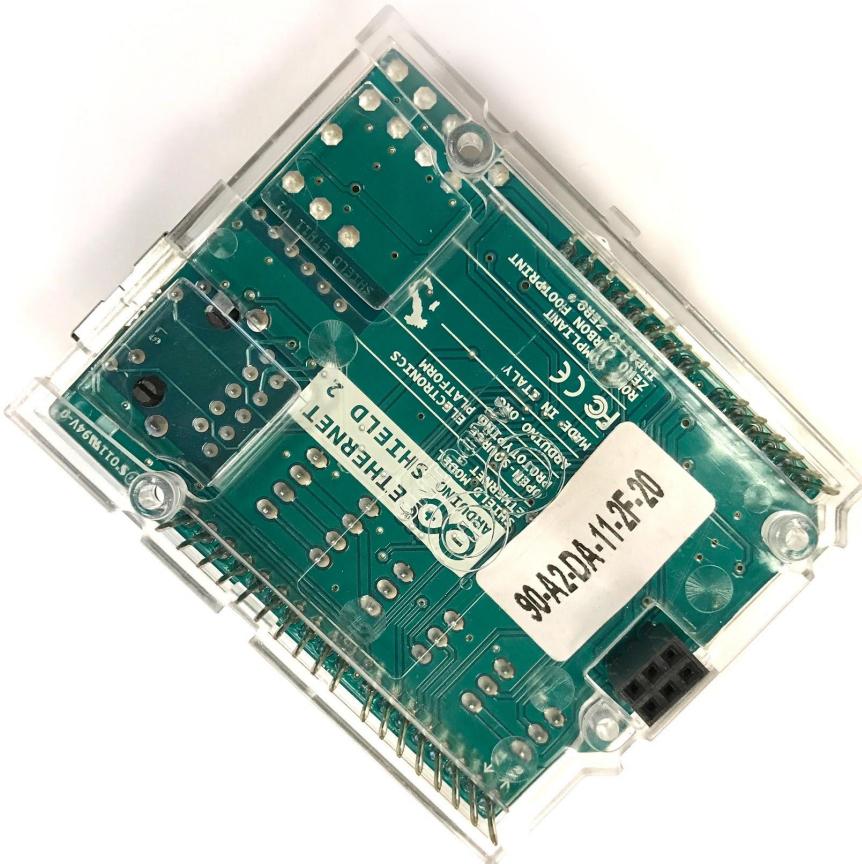
**LESS
TALK
MORE
ROCK**



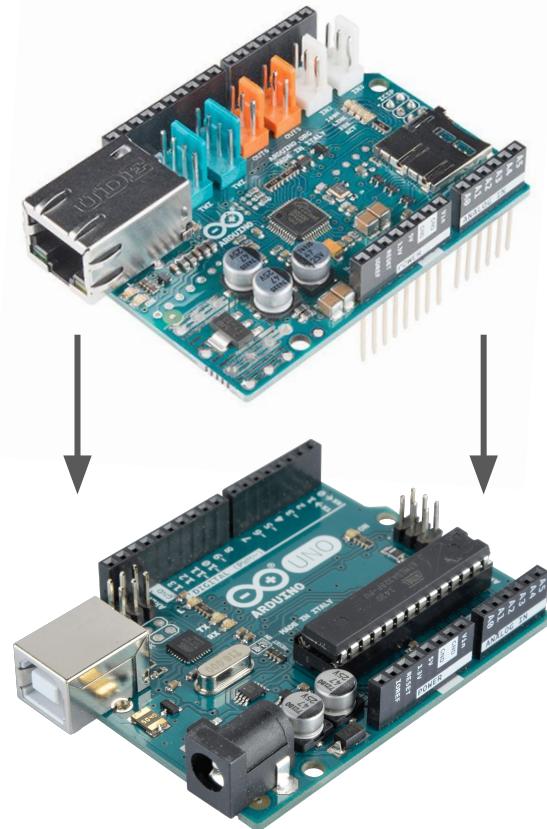
Hello World!



- > turn the shield upside down
- > and take note of the code (MAC address)



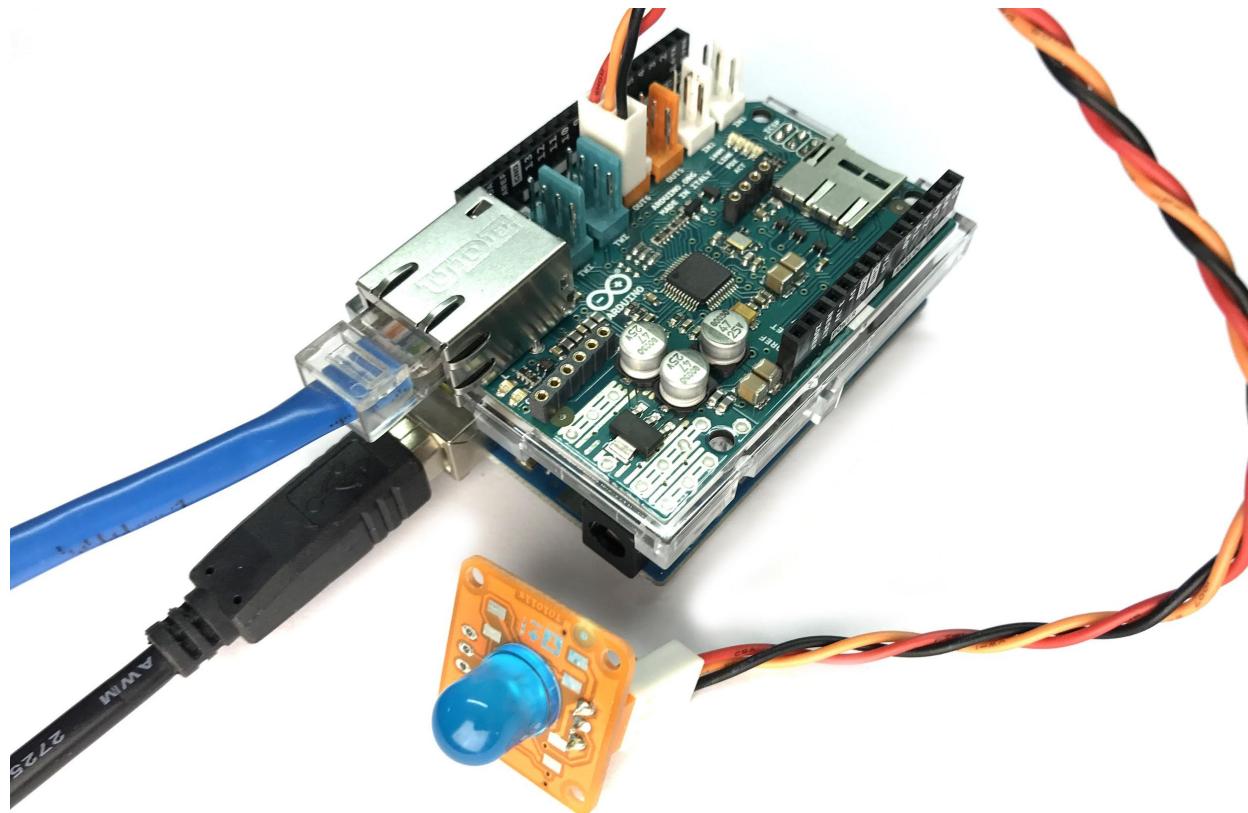
Connect Ethernet Shield to Arduino UNO



Connect LED to OUT6



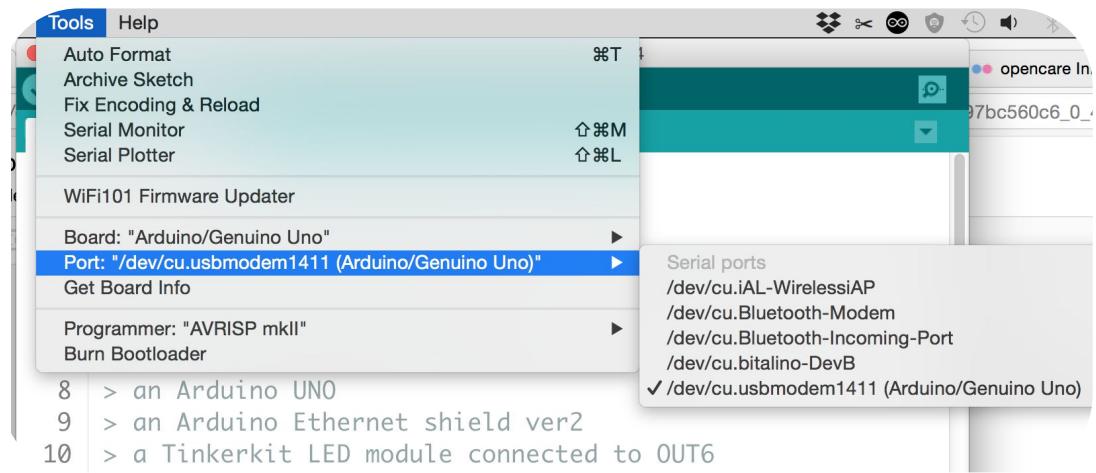
Connect Arduino to the computer and the shield to an Ethernet cable



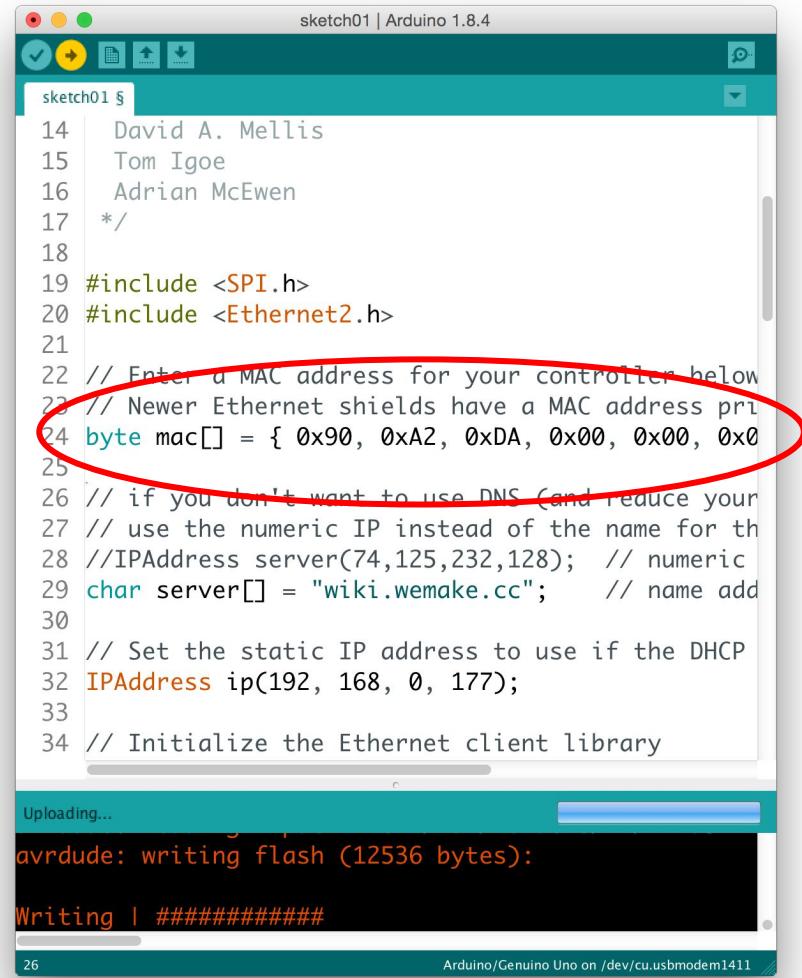
> open the Arduino IDE



> and select port



- > open Sketch01
- > add your code
(MAC address)
- > and Upload to
the board



The screenshot shows the Arduino IDE interface with the title "sketch01 | Arduino 1.8.4". The code editor contains the following sketch:

```
sketch01 §
14  David A. Mellis
15  Tom Igoe
16  Adrian McEwen
17  */
18
19 #include <SPI.h>
20 #include <Ethernet2.h>
21
22 // Enter a MAC address for your controller below
22 // Newer Ethernet shields have a MAC address pri
24 byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x00, 0x0
25
26 // if you don't want to use DNS (and reduce your
27 // use the numeric IP instead of the name for th
28 //IPAddress server(74,125,232,128); // numeric
29 char server[] = "wiki.wemake.cc"; // name add
30
31 // Set the static IP address to use if the DHCP
32 IPAddress ip(192, 168, 0, 177);
33
34 // Initialize the Ethernet client library
```

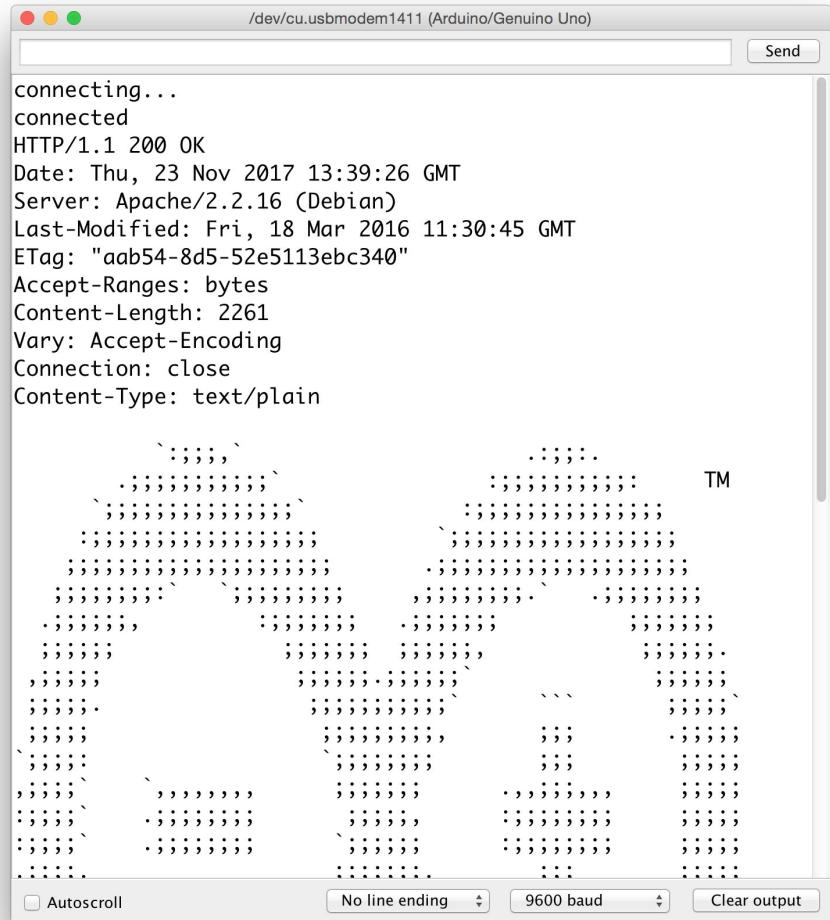
The line `byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x00, 0x0` is circled in red.

The status bar at the bottom shows "Uploading..." and the terminal window displays:

```
avrdude: writing flash (12536 bytes):
Writing | #####
```



- > open the Serial Monitor



The screenshot shows the Arduino Serial Monitor window. The title bar reads "/dev/cu.usbmodem1411 (Arduino/Genuino Uno)". The main text area displays the following output:

```
connecting...
connected
HTTP/1.1 200 OK
Date: Thu, 23 Nov 2017 13:39:26 GMT
Server: Apache/2.2.16 (Debian)
Last-Modified: Fri, 18 Mar 2016 11:30:45 GMT
ETag: "aab54-8d5-52e5113ebc340"
Accept-Ranges: bytes
Content-Length: 2261
Vary: Accept-Encoding
Connection: close
Content-Type: text/plain
```

Below the text, there is a large ASCII art logo consisting of various punctuation marks like commas, semicolons, and colons arranged in a grid pattern.

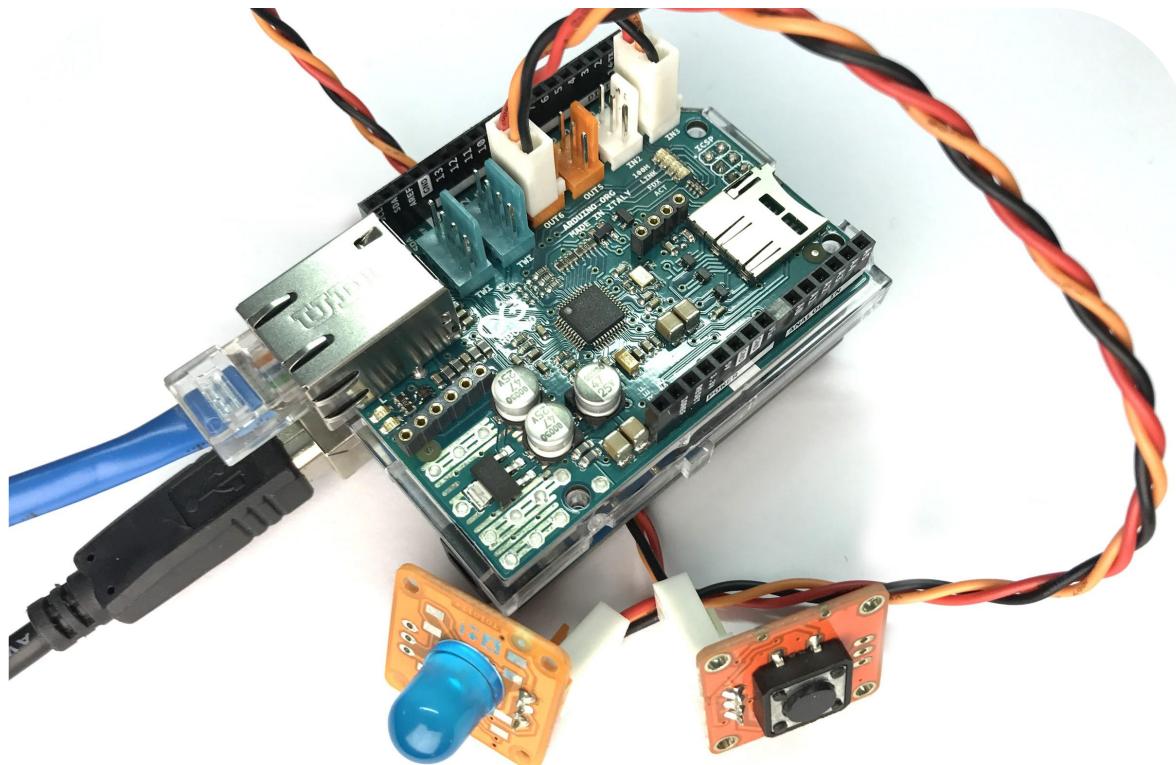
At the bottom of the window, there are several control buttons: "Autoscroll" (unchecked), "No line ending" (dropdown menu), "9600 baud" (dropdown menu), and "Clear output".



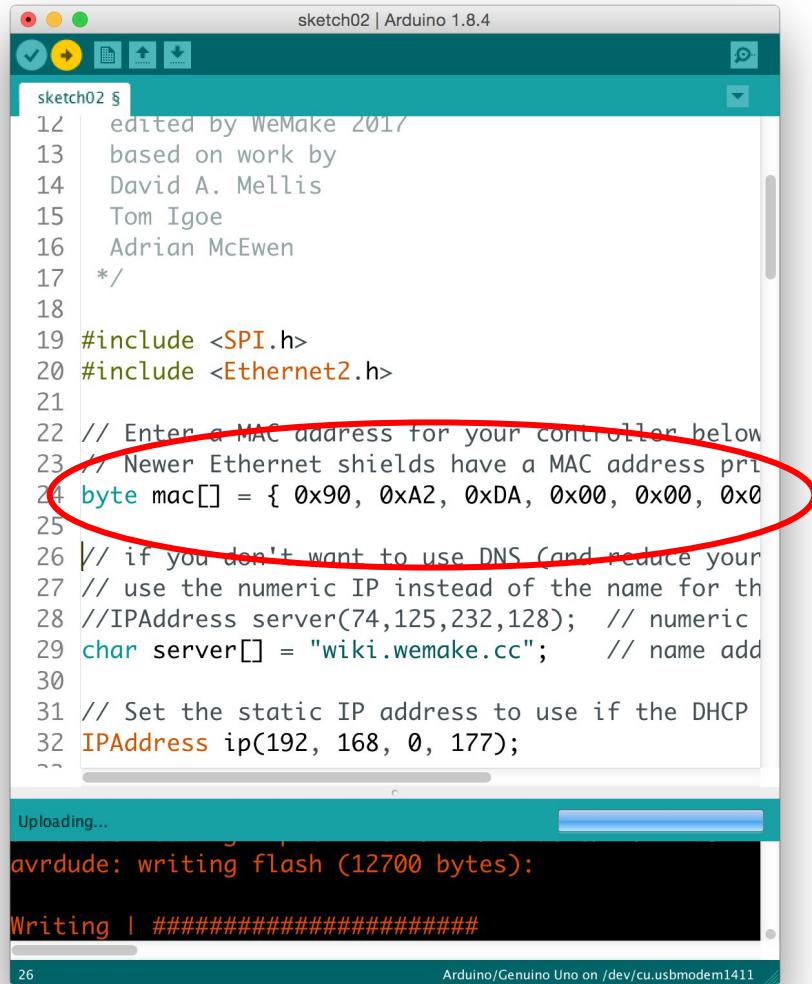
Let's add interaction!



Connect pushbutton to IN3



- > open Sketch02
- > add your code
(MAC address)
- > and Upload to
the board



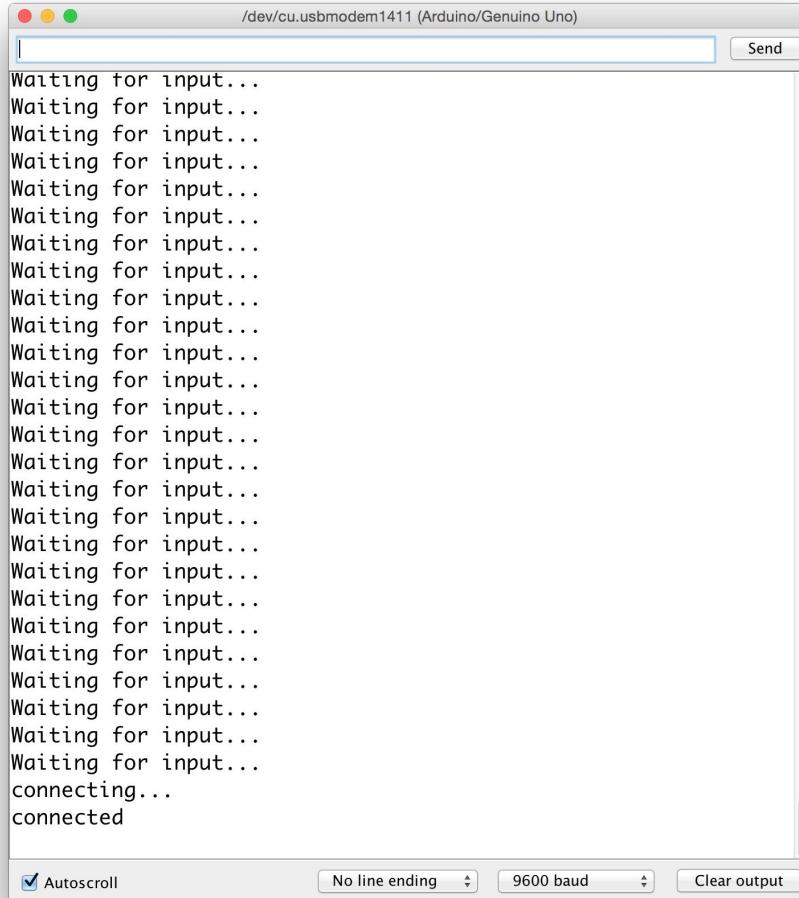
The screenshot shows the Arduino IDE interface with the title "sketch02 | Arduino 1.8.4". The code editor contains the following sketch02.ino code:

```
12 // edited by WeMake 2017
13 // based on work by
14 // David A. Mellis
15 // Tom Igoe
16 // Adrian McEwen
17 */
18
19 #include <SPI.h>
20 #include <Ethernet2.h>
21
22 // Enter a MAC address for your controller below
23 // Newer Ethernet shields have a MAC address pri
24 byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x00, 0x0
25
26 // if you don't want to use DNS (and reduce your
27 // use the numeric IP instead of the name for th
28 //IPAddress server(74,125,232,128); // numeric
29 char server[] = "wiki.wemake.cc"; // name add
30
31 // Set the static IP address to use if the DHCP
32 IPAddress ip(192, 168, 0, 177);
~~
```

The line `byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x00, 0x0` is circled in red.

The status bar at the bottom shows "Uploading..." and "avrduke: writing flash (12700 bytes)". The bottom right corner indicates "Arduino/Genuino Uno on /dev/cu.usbmodem1411".

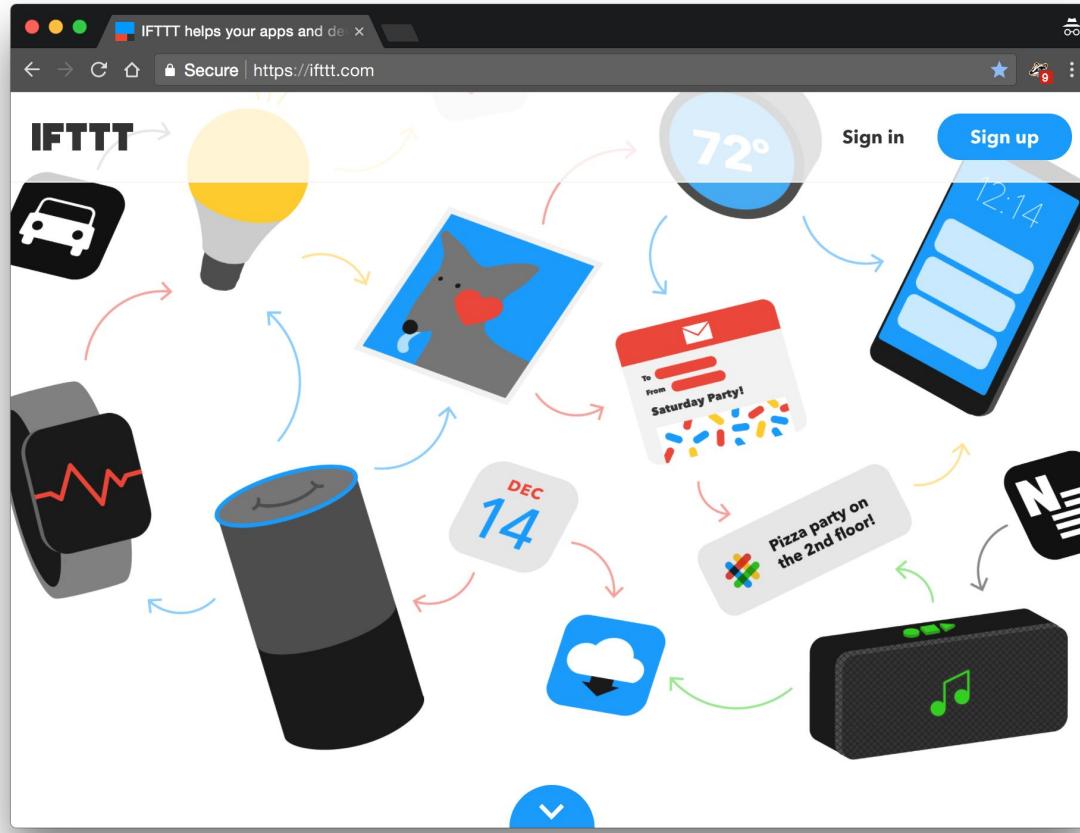
- > open the Serial Monitor
 - > push the Button!



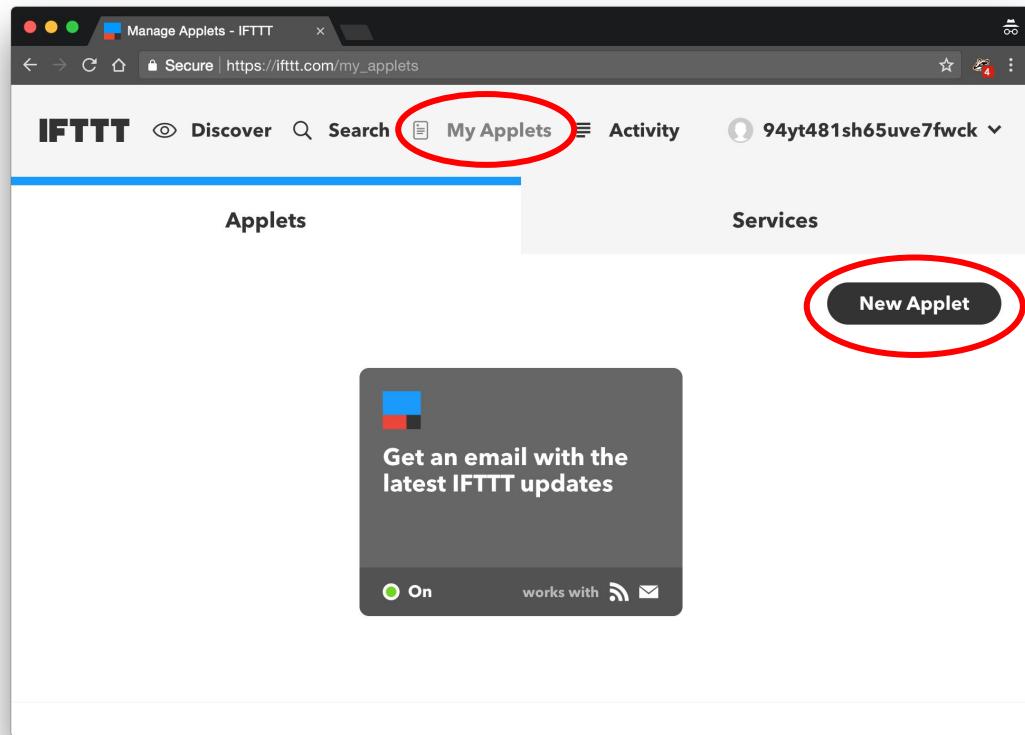
If This Then Whaaaat???



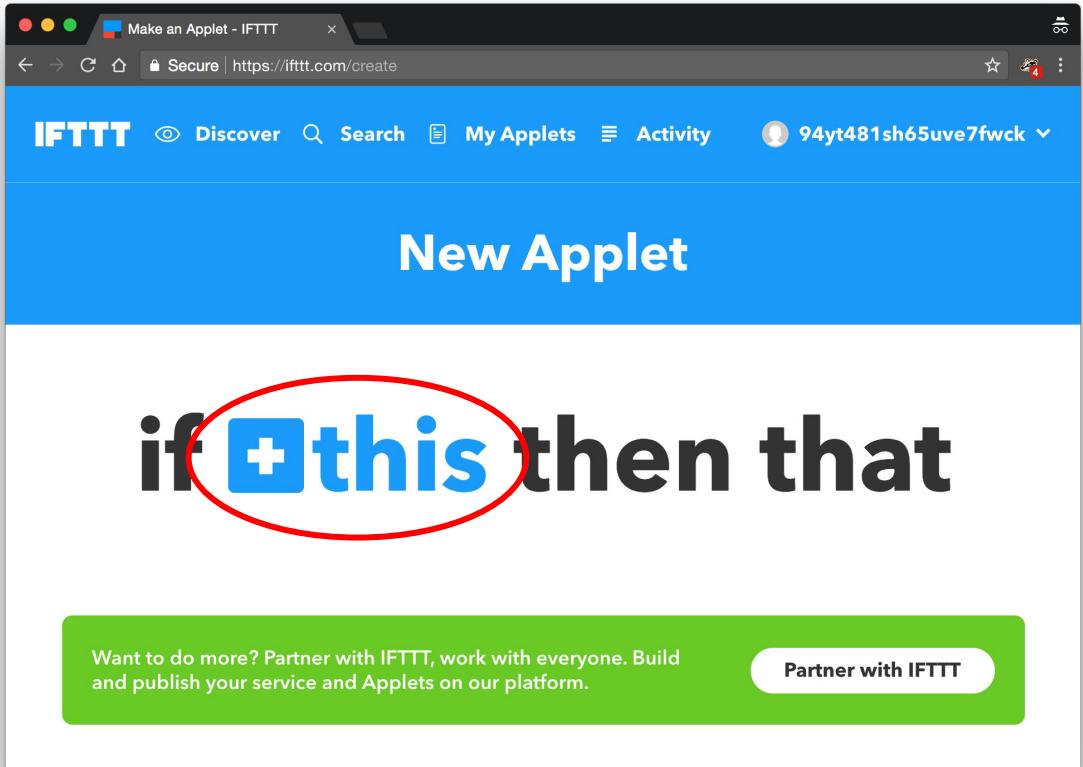
Go to ifttt.com and create a new account



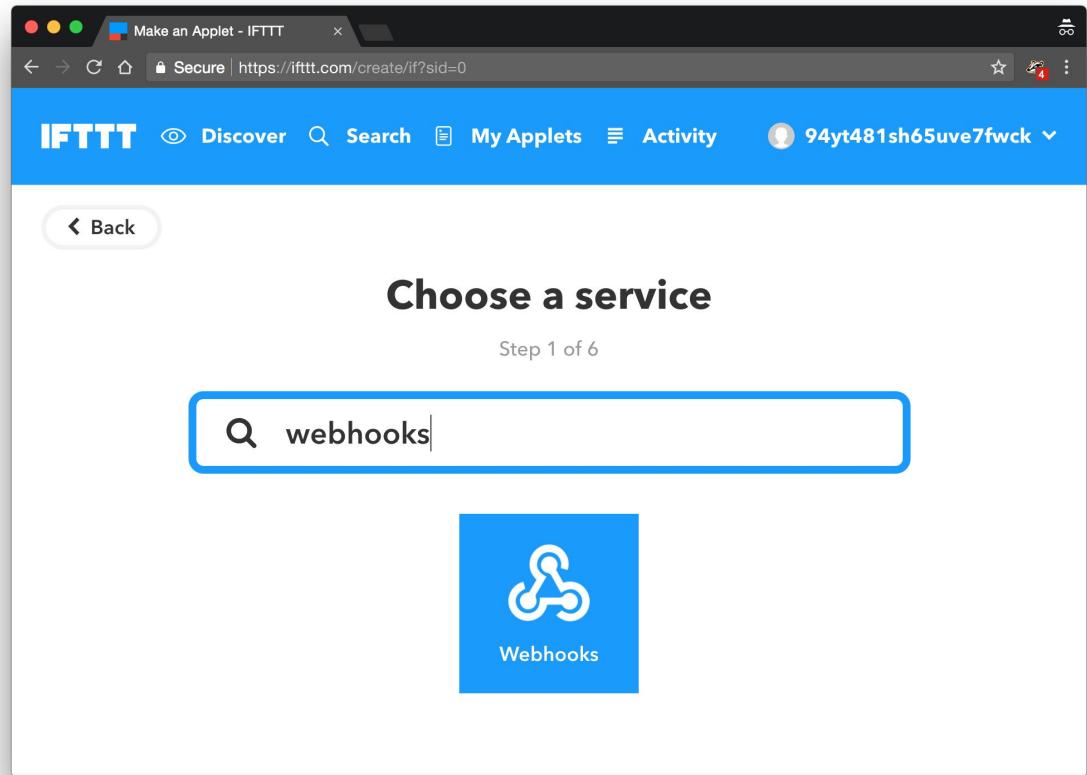
- > Click on My Applets
- > and New Applet button



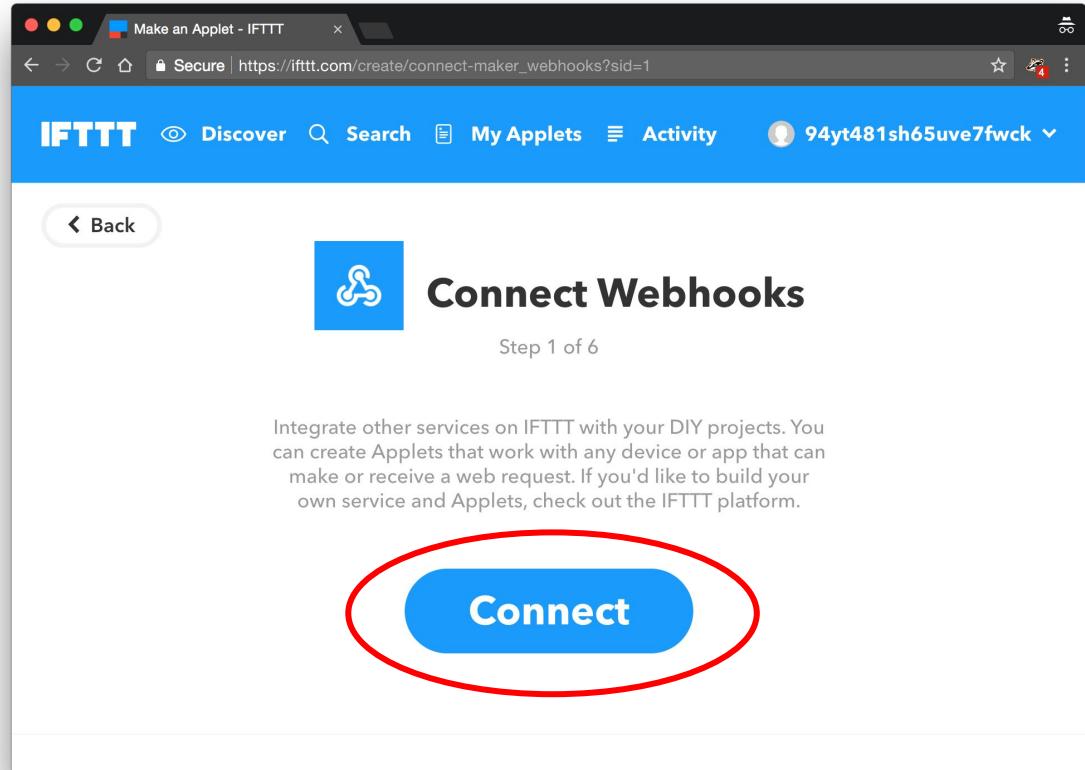
> Click on “this”



- > Search and select “webhooks”



> Connect “webhooks”



The screenshot shows the IFTTT web interface for creating a new applet. The title bar says "Make an Applet - IFTTT". The main header has tabs for "Discover", "Search", "My Applets", "Activity", and a user profile. Below the header, there's a "Back" button and a "Connect Webhooks" section. The "Connect Webhooks" section features a blue icon with three white circles and the text "Connect Webhooks" followed by "Step 1 of 6". A descriptive paragraph explains that users can integrate other services with DIY projects. At the bottom of the section is a large blue "Connect" button, which is highlighted with a thick red oval.

Secure | https://ifttt.com/create/connect-maker_webhooks?sid=1

IFTTT Discover Search My Applets Activity 94yt481sh65uve7fwck

Back

Connect Webhooks

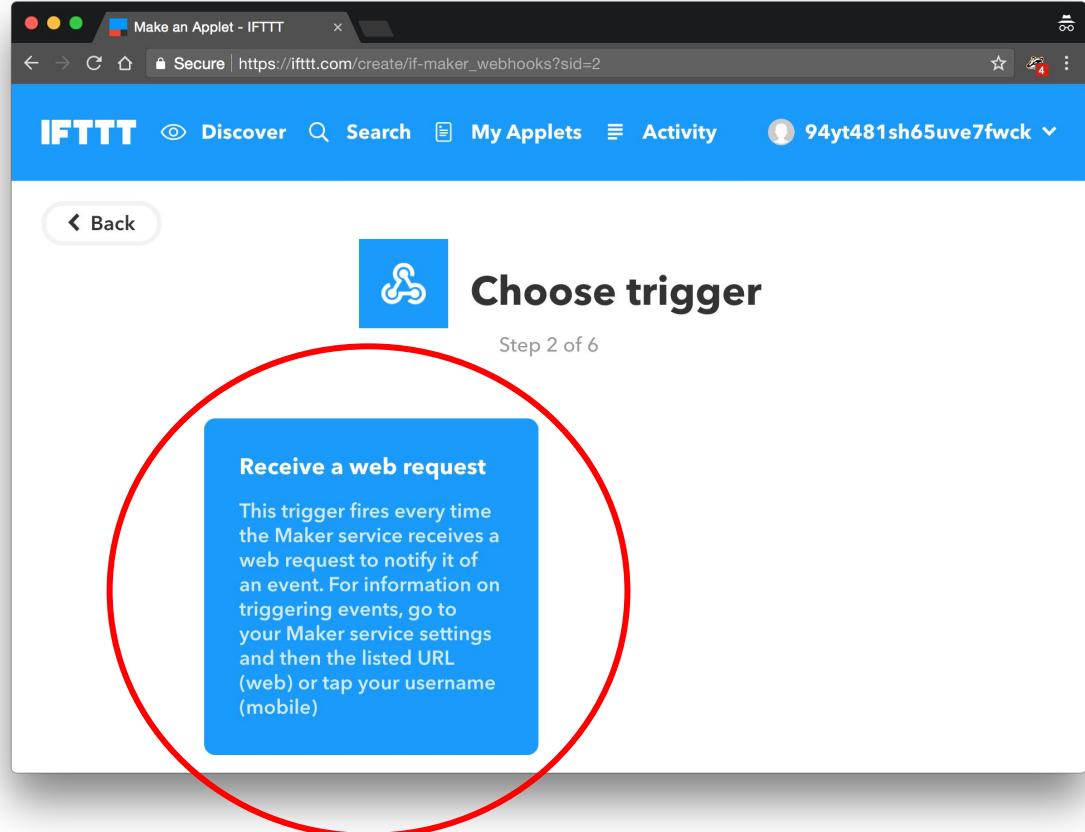
Step 1 of 6

Integrate other services on IFTTT with your DIY projects. You can create Applets that work with any device or app that can make or receive a web request. If you'd like to build your own service and Applets, check out the IFTTT platform.

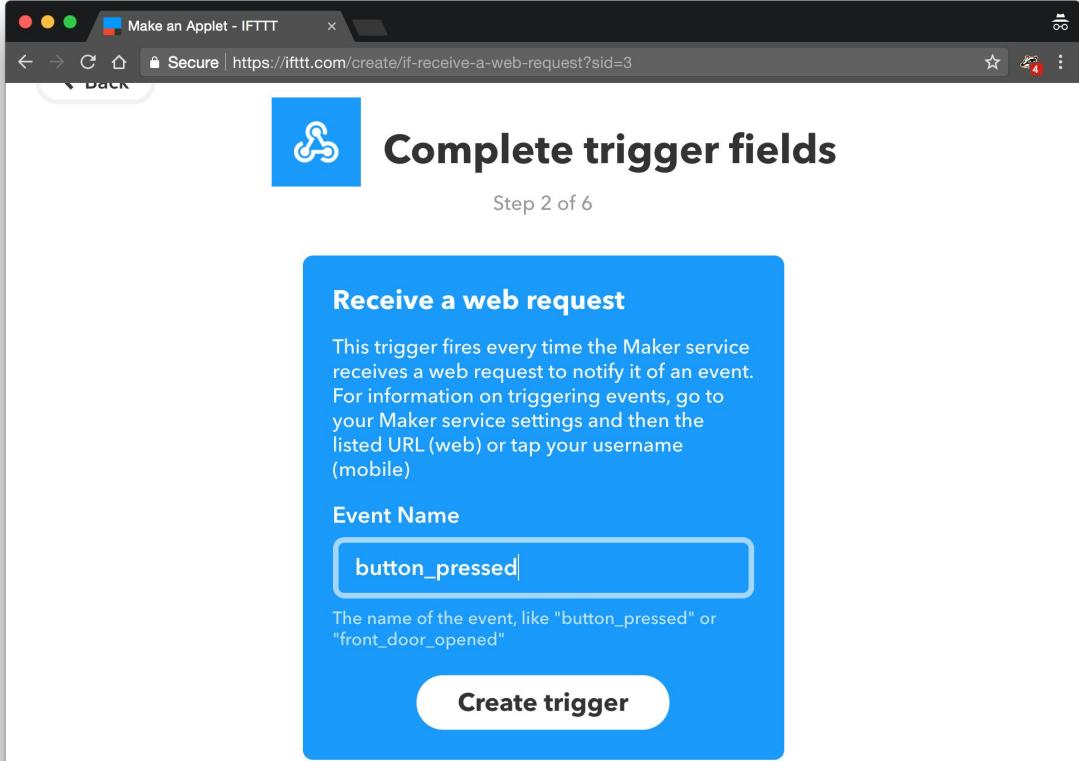
Connect



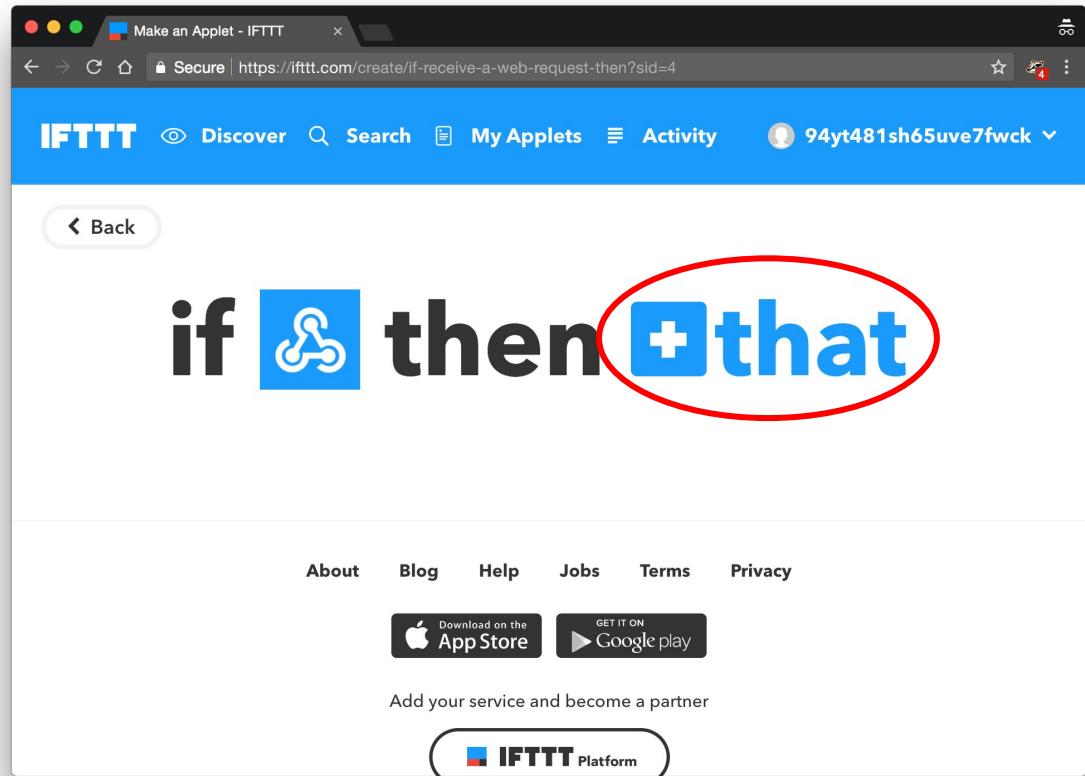
> Choose the
“Receive a web
request” trigger



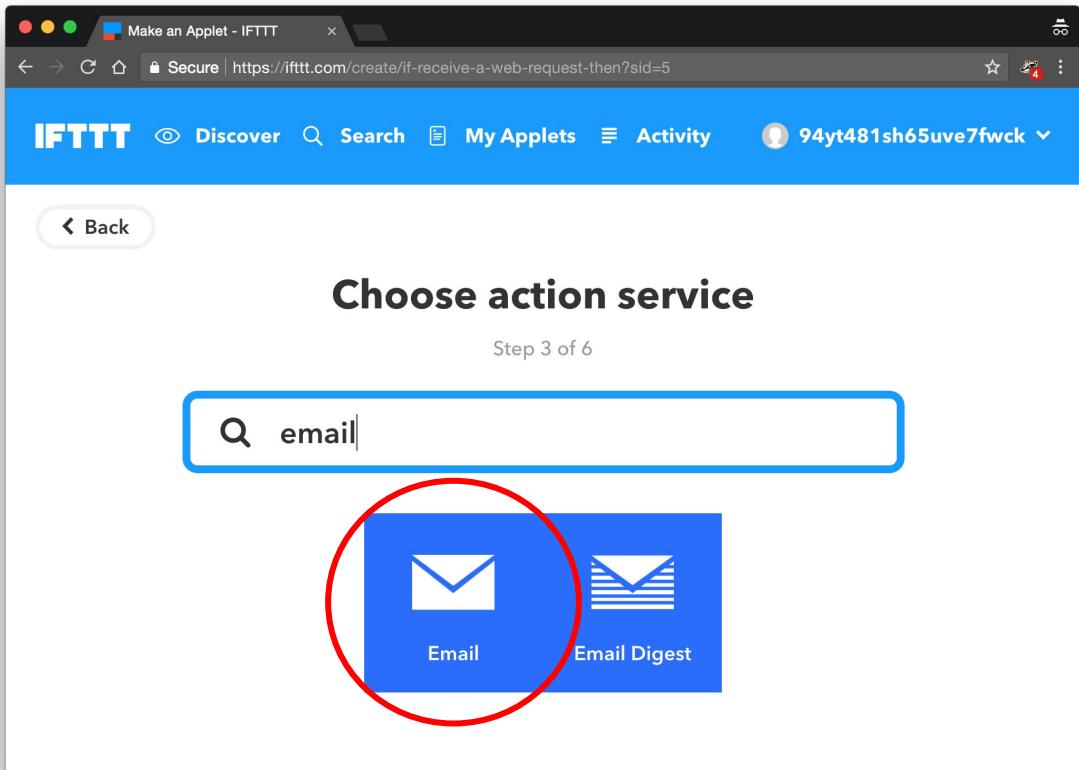
- > Give the event a name like "button_pressed"
- > and Create trigger



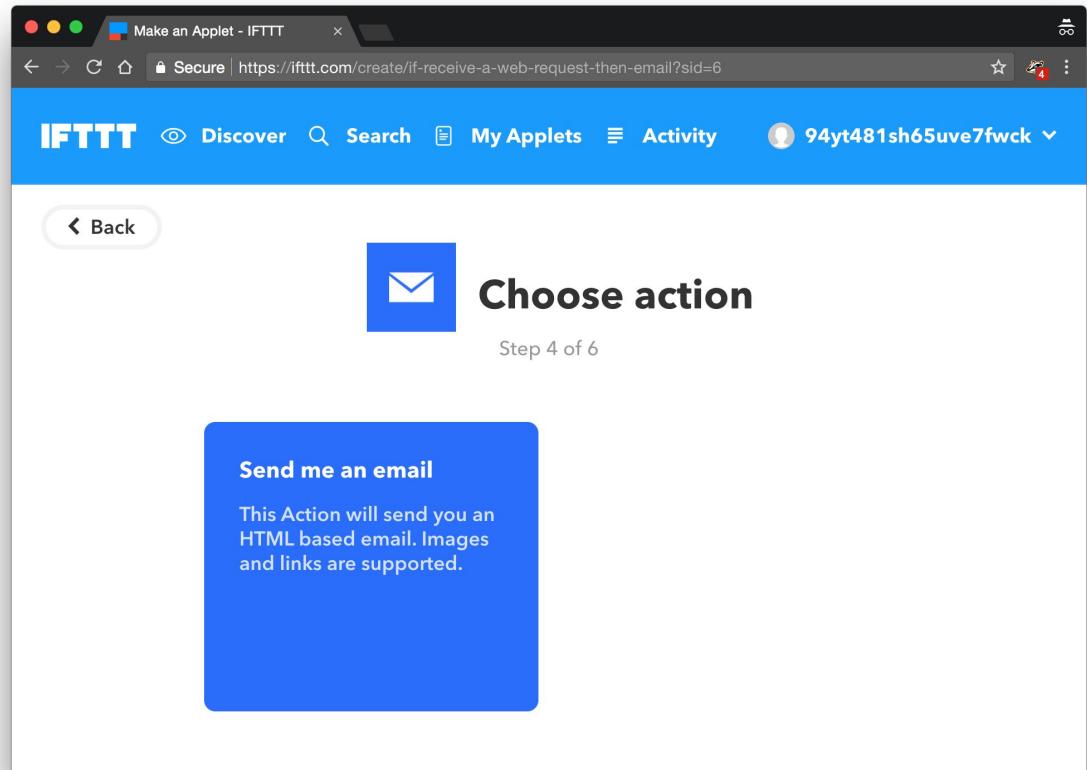
> Now click on
“that”



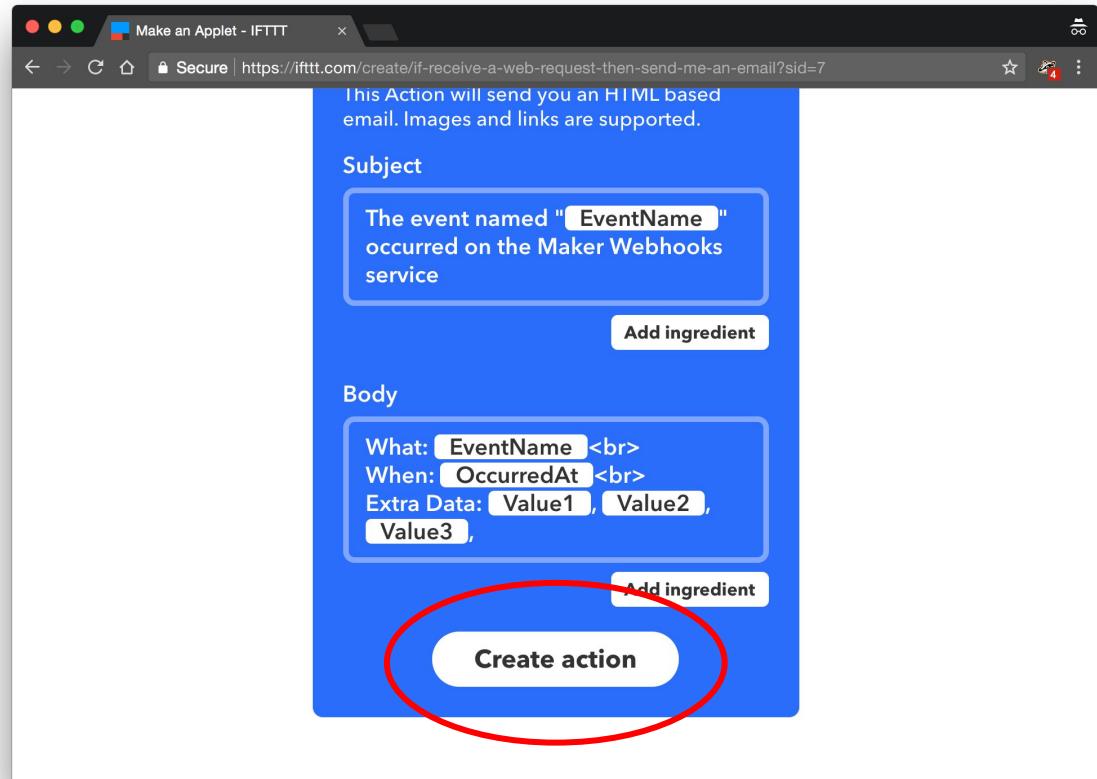
> Choose “Email”



- > Choose the “Send me an email” action

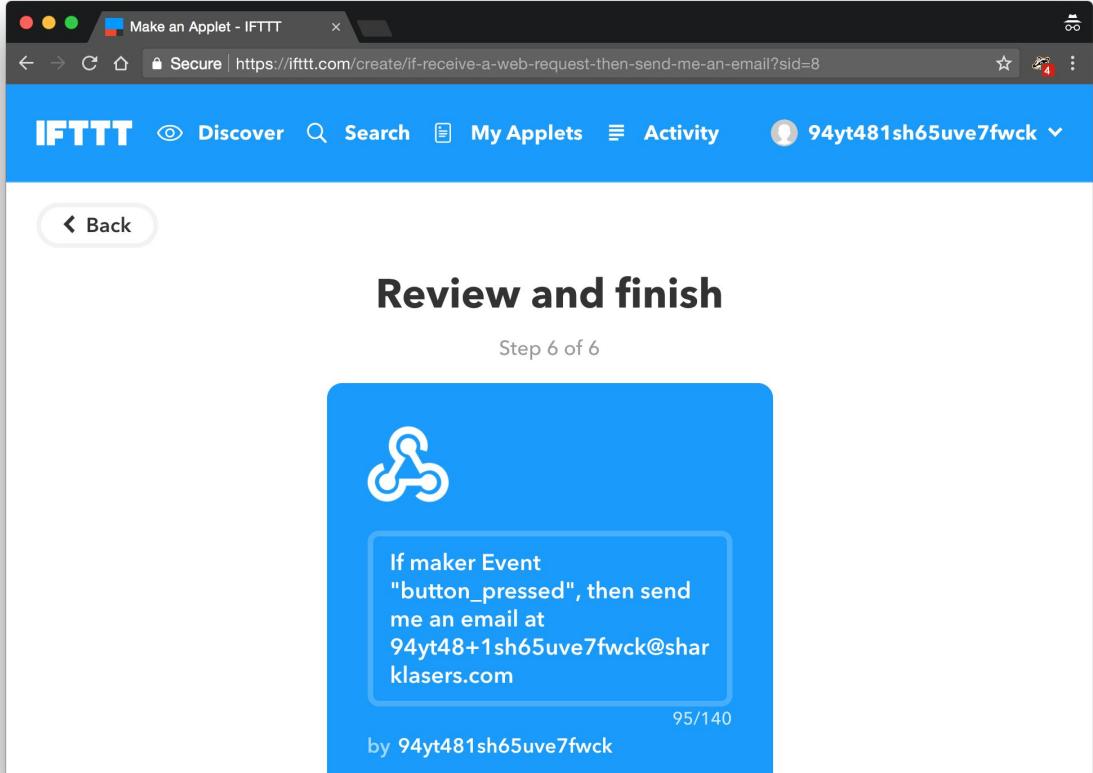


> Create action



> Finish :)

Yaaay you
just created
your first
Applet!!!



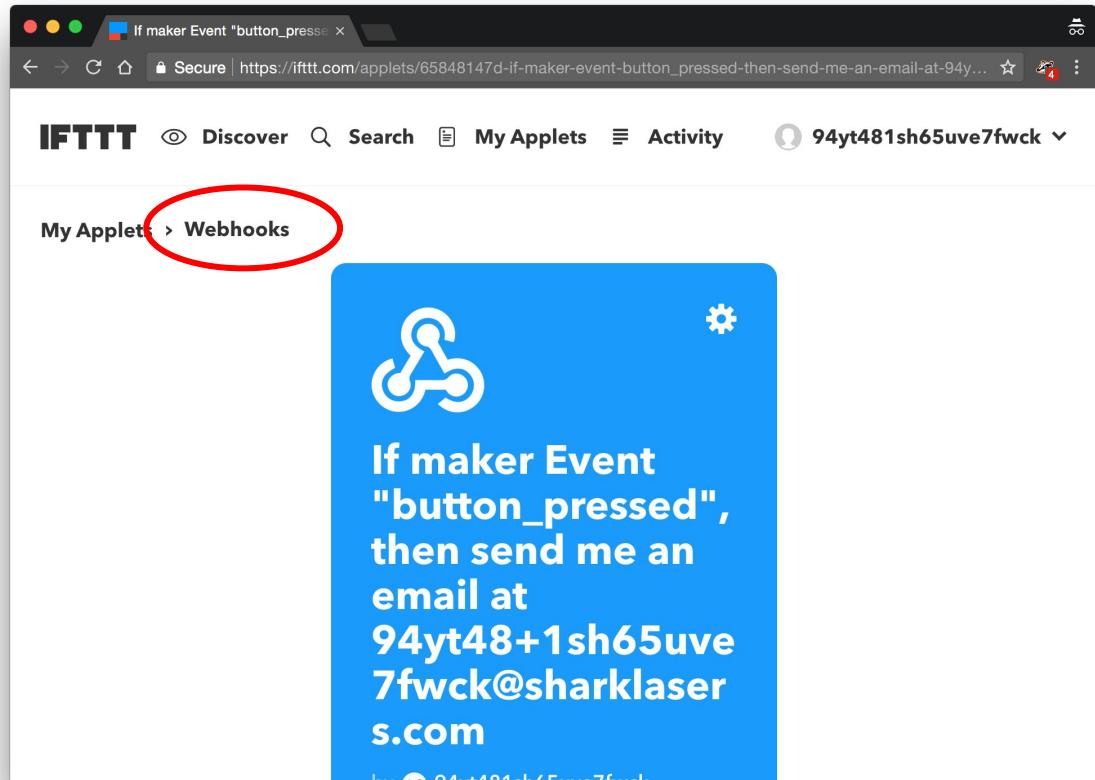
The image shows a screenshot of the IFTTT website in a web browser. The title bar reads "Make an Applet - IFTTT" and the URL is "https://ifttt.com/create/if-receive-a-web-request-then-send-me-an-email?sid=8". The IFTTT logo is at the top left, followed by "Discover", "Search", "My Applets", "Activity", and a user profile icon. Below the header, there's a "Back" button and the main content area titled "Review and finish" which says "Step 6 of 6". A large blue card displays the applet configuration: "If maker Event 'button_pressed', then send me an email at 94yt48+1sh65uve7fwck@sharklasers.com". The card also shows "95/140" and "by 94yt481sh65uve7fwck".



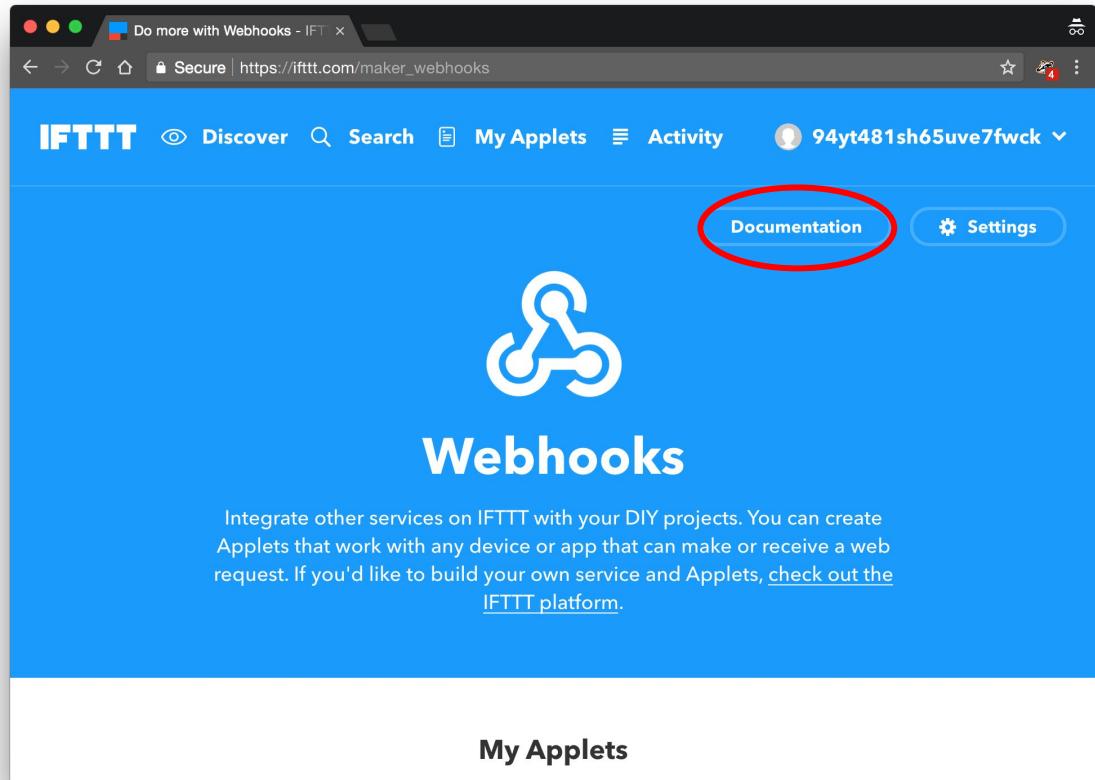
Almost there...



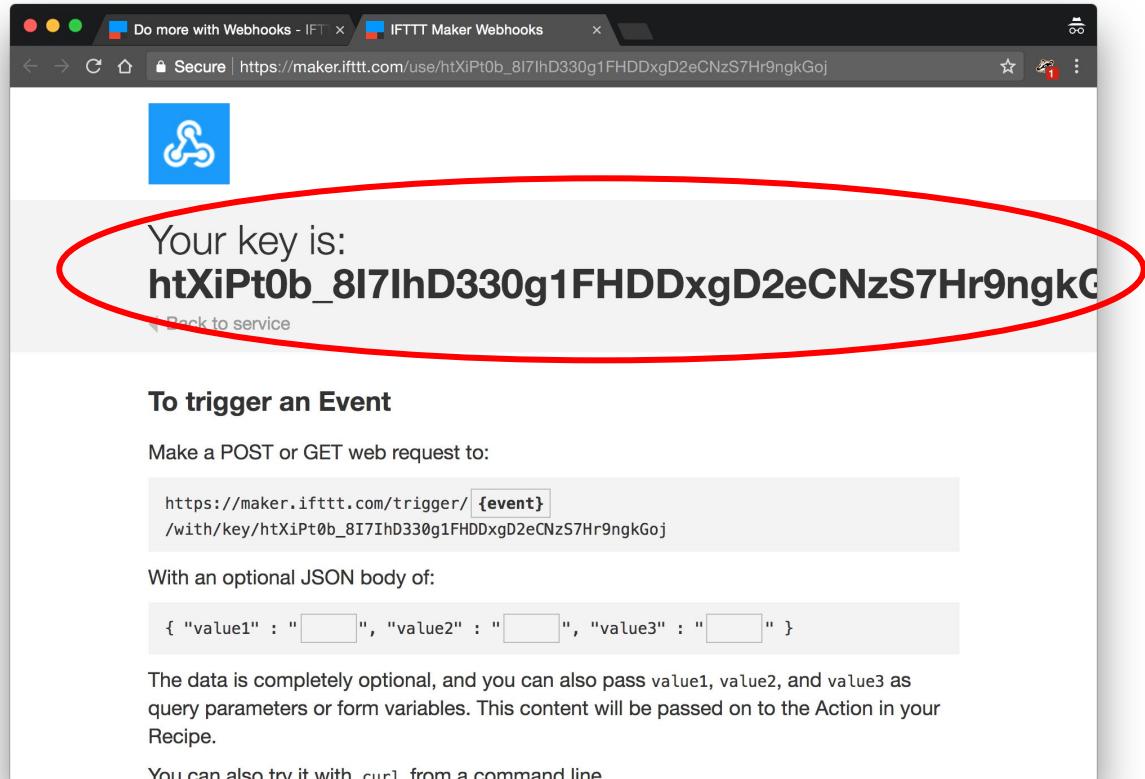
> Click on
“Webhooks”



> Click on the Documentation button



> Copy the Key



Your key is:
htXiPt0b_8I7lhD330g1FHDDxgD2eCNzS7Hr9ngkGo

To trigger an Event

Make a POST or GET web request to:

```
https://maker.ifttt.com/trigger/{event}  
/with/key/htXiPt0b_8I7lhD330g1FHDDxgD2eCNzS7Hr9ngkGo
```

With an optional JSON body of:

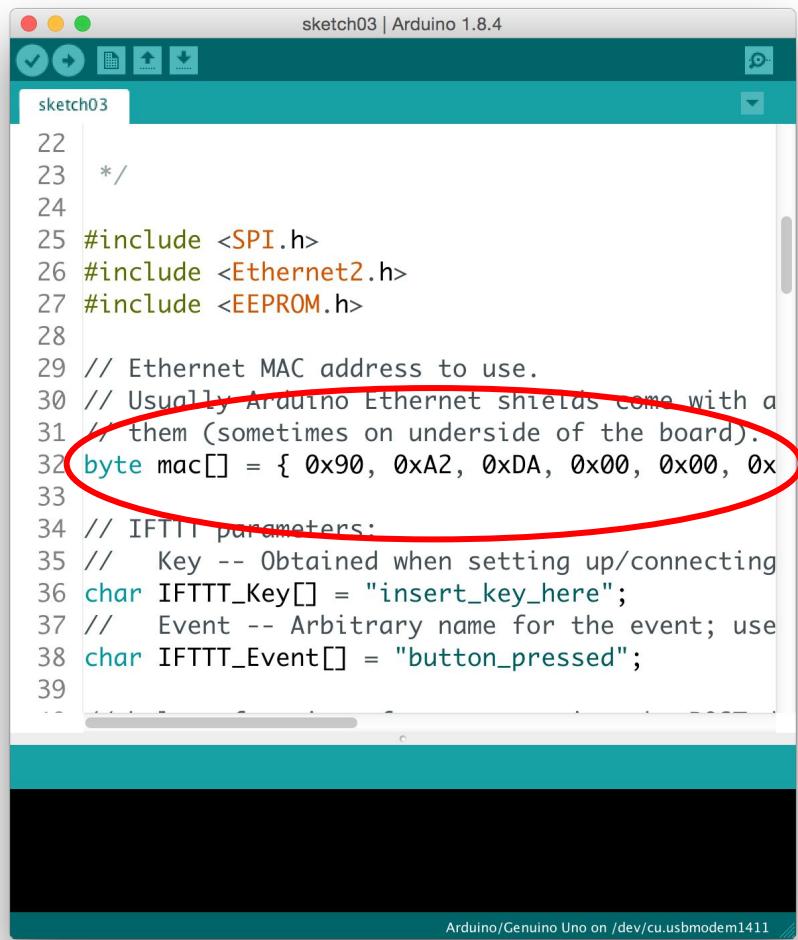
```
{ "value1" : "____", "value2" : "____", "value3" : "____" }
```

The data is completely optional, and you can also pass value1, value2, and value3 as query parameters or form variables. This content will be passed on to the Action in your Recipe.

You can also try it with `curl` from a command line.



- > open Sketch03
- > add your code
(MAC address)



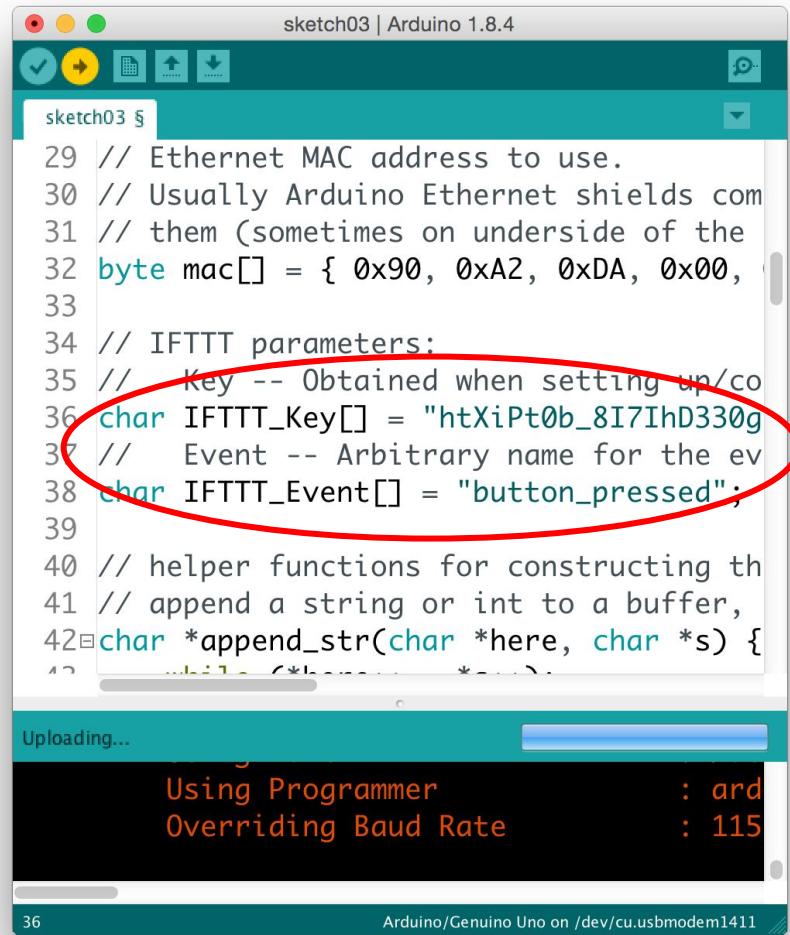
The screenshot shows the Arduino IDE interface with the title bar "sketch03 | Arduino 1.8.4". The code editor contains the following C++ code:

```
22
23  */
24
25 #include <SPI.h>
26 #include <Ethernet2.h>
27 #include <EEPROM.h>
28
29 // Ethernet MAC address to use.
30 // Usually Arduino Ethernet shields come with a
31 // them (sometimes on underside of the board).
32 byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x
33
34 // IFTTT parameters:
35 // Key -- Obtained when setting up/connecting
36 char IFTTT_Key[] = "insert_key_here";
37 // Event -- Arbitrary name for the event; use
38 char IFTTT_Event[] = "button_pressed";
39
```

A red oval highlights the line of code defining the MAC address: `byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x`. The Arduino IDE status bar at the bottom right indicates "Arduino/Genuino Uno on /dev/cu.usbmodem1411".



- > paste the KEY in IFTTT_Key
- > (eventually change IFTTT_Event to match with your Event_name)
- > Upload!



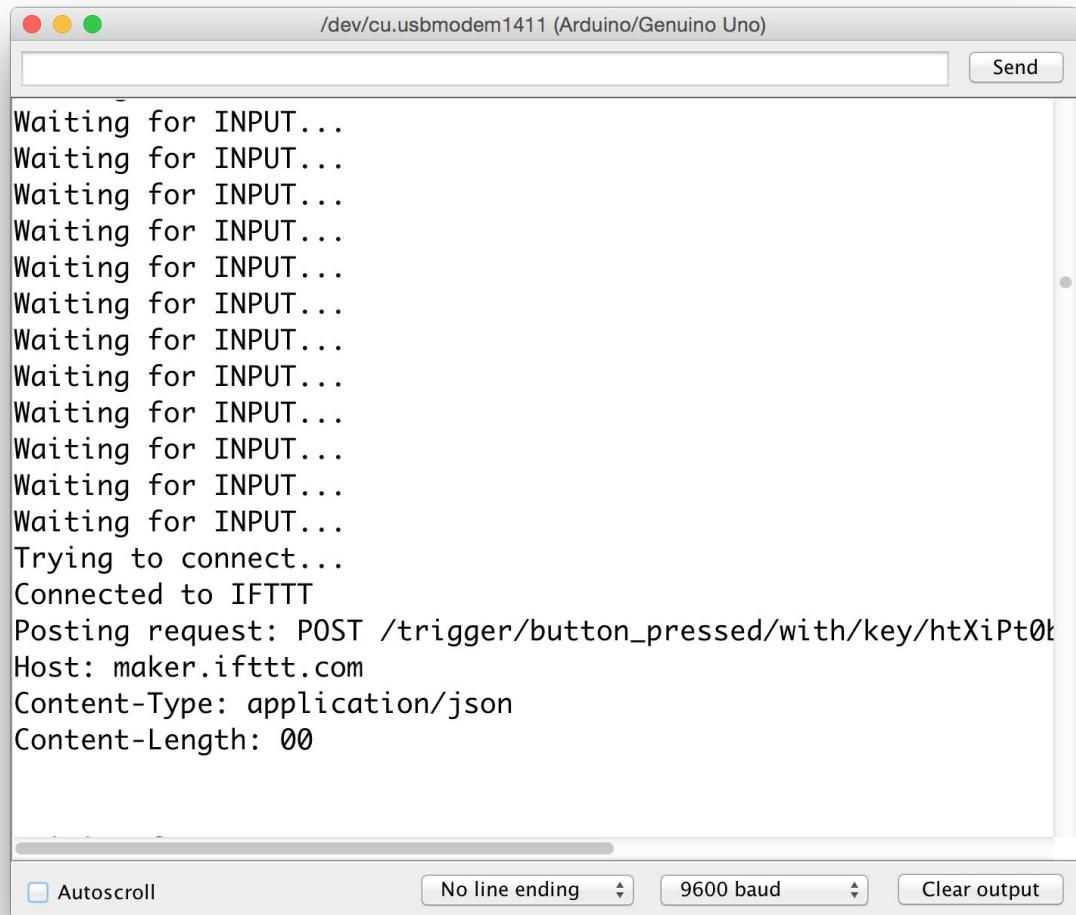
The screenshot shows the Arduino IDE interface with the title bar "sketch03 | Arduino 1.8.4". The code editor contains the following C++ code:

```
sketch03 §
29 // Ethernet MAC address to use.
30 // Usually Arduino Ethernet shields com
31 // them (sometimes on underside of the
32 byte mac[] = { 0x90, 0xA2, 0xDA, 0x00,
33
34 // IFTTT parameters:
35 // Key -- Obtained when setting up/co
36 char IFTTT_Key[] = "htXiPt0b_8I7IhD330g
37 // Event -- Arbitrary name for the ev
38 char IFTTT_Event[] = "button_pressed";
39
40 // helper functions for constructing th
41 // append a string or int to a buffer,
42 char *append_str(char *here, char *s) {
43     ...
44 }
```

A red oval highlights the line `char IFTTT_Key[] = "htXiPt0b_8I7IhD330g";`. The status bar at the bottom shows "Uploading..." and "Using Programmer : ard Overriding Baud Rate : 115".



> Open Serial Monitor



```
Waiting for INPUT...
Trying to connect...
Connected to IFTTT
Posting request: POST /trigger/button_pressed/with/key/htXiPt0k
Host: maker.ifttt.com
Content-Type: application/json
Content-Length: 00
```

Autoscroll No line ending 9600 baud Clear output



Press IT!



