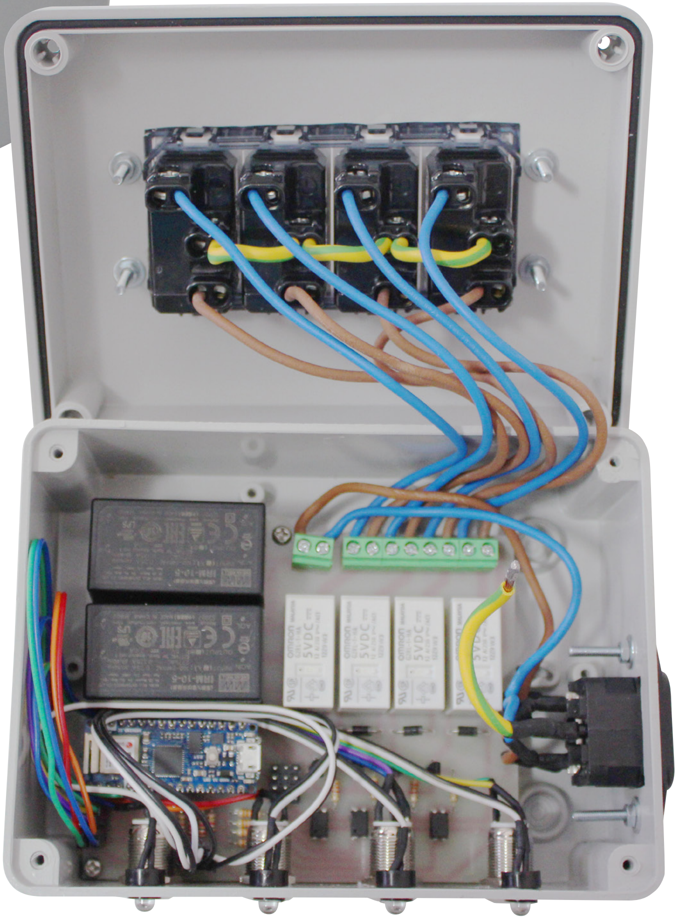
DIYODE

smartphone + hero styling tba



Smart Plug

IoT Relays with Voice Control

CHRISTAN IANNELLA

28 year old Electronics Enthusiast, Italy.

@raspuino

By using an Arduino Nano 33 IoT board and the Arduino IoT Cloud, Christian is able to turn on mains appliances in his home from his Smartphone and voice commands.

Thanks to the popularity of voice controlled devices, such as Google Assistant and Amazon Alexa, we can now turn lights and devices on and off in the home with a voice command. Many types of WiFi Smart plugs are commercially available nowadays, but maker, Christian Iannella chose to make his own.

We caught up with Christian to find out more about his four way home-made power point and how he made it.

First, it is very important that we point out that Christian's project involves mains wiring. It is dangerous and can be lethal to handle mains power. In Australia, you need to be a q ualified electrician to handle any voltages above 50VAC. For this reason, we describe Christian's project as inspiration only, and recommend that you only use low DC voltages if you want to build something similar.

Before we discuss your project Christian, please tell our readers a little about yourself.

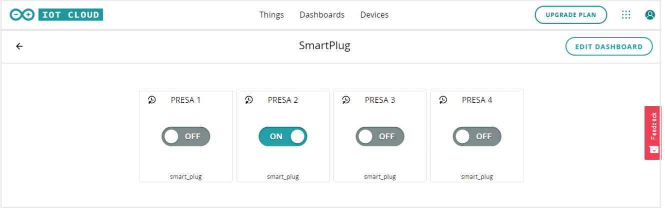
I'm 28, live in Rome, and I am passionate about electronics and programming. I am completely self-taught and since I discovered Arduino a few years ago I have not been able to do without it.

Thank you for submitting your Smart Plug project via our website to be featured in our mag. What motivated you to make it and how does it work?

I wanted to demonstrate to myself and others the simplicity of working with Arduino and the Arduino IoT Cloud (you will see that the project is managed by very few lines of code) and how easy it is to associate everything with Amazon Alexa, thanks to the Arduino Skill for Alexa. In fact, with a quick configuration the game is done.

So summarising the project, thanks to the Arduino Nano 33 IoT board, it connects to the WiFi network that we indicate in our code.

Once the connection is established, it will try to connect with the Cloud to which we would previously have associated our Arduino board. Once this is done, we can activate our sockets from the cloud dashboard.



SmartPlug

PRESA 1 OFF smart_plug

PRESA 2 ON smart_plug

PRESA 3 OFF smart_plug

PRESA 4 OFF smart_plug

To do this, we install the Arduino Skill on Alexa, do a search, and when Alexa has found our Smartplug give it a name and we're done.

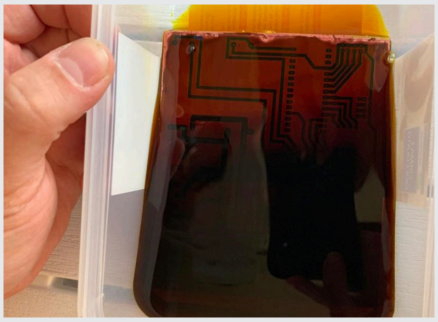
Now we could simply say "Alexa please turn on socket 1" and the corresponding relay will activate. In fact, this system allows us not only to tell what to do, but also to monitor its status. We can ask Alexa if they are on or off.

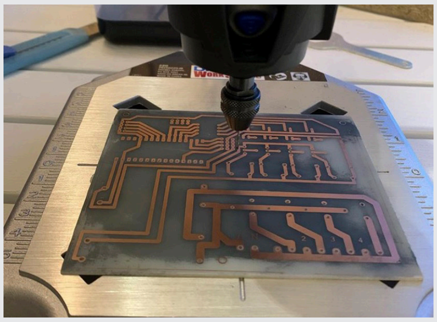
Great! We also notice you have pushbuttons to manually trigger the relays. Is your project limited to just Alexa?

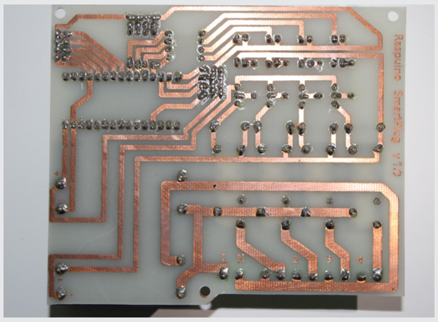
It is possible to interface with Google Assistant, IFTTT, Adafruit IoT, etc. Of all the Clouds I've tried, the Arduino Cloud seems to me the most robust.

The Internet is full of examples of smart plugs that work with various assistants but I decided to do a slightly different project, I didn't want the simple switching on of a relay. I wanted something functional, almost a commercial product.

For this reason, I have worked on an update to implement a small HTML page in the code which we will access with our smartphone. At first or after a reset it will ask us to enter the SSID and PASSWORD of the network that we would like to use in so as not to be bound to the network set in the code. It is possible to do this because the Nano 33 IoT can work as an access point, we just need to search for the network called Smartplug among the available networks. Using a browser, we go to the address of our Nano 33 IoT, enter the data and restart the Smartplug. Upon restart, it will no longer behave as an access point but will attempt to connect to the network we have set.


Etching the circuit board


Drilling the circuit board


Populated and soldered circuit board

What made you choose the Nano 33 IoT board and could the same be done with other boards?

As for the choice of the board, it fell on the Nano 33 IoT because it is the smallest and cheapest (only 16.00 Euro in Italy) among those with WiFi connectivity. I have used many for other projects such as MKR WiFi 1010 or the Yun but the Nano 33 IoT is undoubtedly the most versatile.

We notice you made your own circuit board as well. Tell us more about that.

The PCB was made with the Press n Peel technique. For those who do not know, it is a technique that allows us, once we have obtained the design of the printed circuit, to print it in a mirrored way on a PnP sheet, and with an iron, transfer it on the base copper. After which, it will only be necessary to eliminate the excess copper with acid and then drill it.

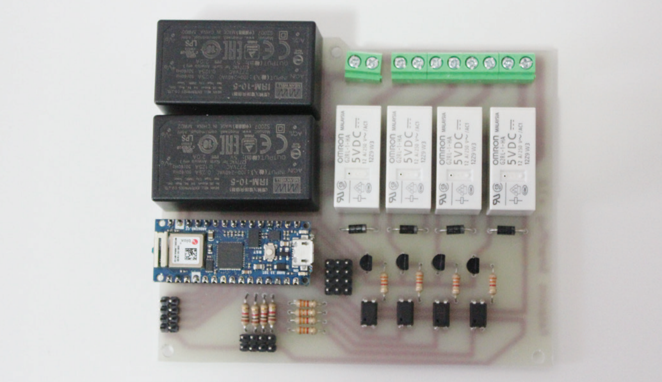
I made the drawing with FidoCADJ, a free and extremely simple to use software.

Once the components have been soldered, the circuit dimensions will be approximately 10cm x 10cm x 2cm so it will be easy to connect in an electrical panel or similar.

It's great to be able to make PCBs at home. We've published some tutorials on using Press n Peel and how to etch a PCB (Issue 36). Run us through how the circuitry works and what components you use?

It's nothing too complicated. You will immediately notice the two Mean Well 5v 2A power supplies (I did things big).

I wanted to separate the logic part from the power part to protect the Arduino board from possible discharges so the two blocks are then coupled via optocouplers (PC817).



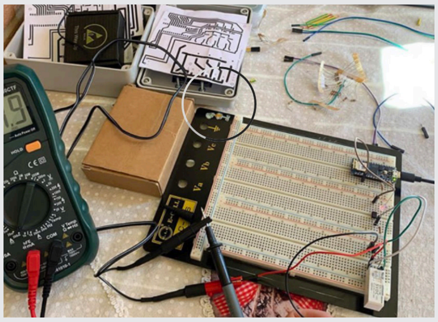
Both the logic and the power part operate at 5v. Obviously it is possible to power everything with any energy source, not necessarily the domestic voltage (230VAC in Italy).

The relays also work at 5v and are operated via a PN2222a (NPN) transistor which is capable of driving up to 600mA but, in any case, much less is needed. In turn, the transistor is driven by a 3.3k resistor from a PC817 optocoupler which is the meeting point between the logic part and the power part. It will be the one that interrupts any power discharges.

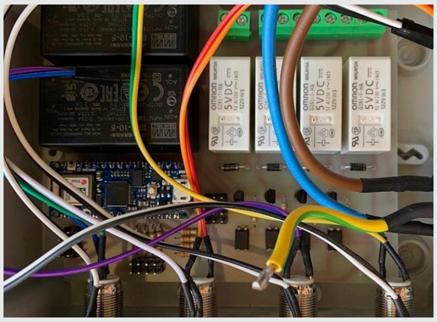
Note the diode across the relay serves to protect our transistor when the relative relay is deactivated.

We then come to the logic part. First of all, we have four pushbuttons connected to D2, D3, D4, D5 of the Arduino. You will notice that there are no pull-up or pull-down resistors because the pull-up is done via software (Arduino Nano 33 IoT allows it). With these buttons we will be able to control the relative sockets. One press turns the related relay on, pressing it again turns it off, and so on.

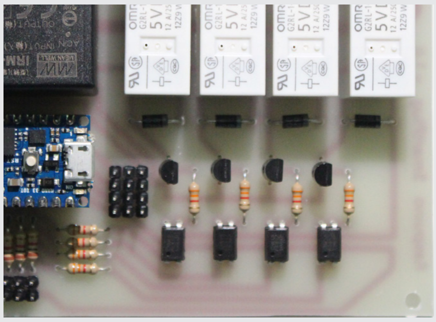
The pins D6, D7, D8, D9 are set as Dout also in our code. These are connected to four LEDs through 1kΩ resistors, and the four photocouplers through 3.3kΩ resistors.



Early prototyping



Heatshrink on all wiring



Optoouplers

As you can see on the PCB, there is a connector next to the Arduino board. I wanted to leave it for any updates such as adding a temperature sensor, a gas detector, or other device. It is connected to D10, D11, and D12.

PARTS REQUIRED:
1 x Arduino Nano 33 IoT
4 x Pushbuttons
4 x LEDs
8 x 3.3kΩ Resistors
4 x 1kΩ Resistors
4 x PN2222a NPN Transistors
4 x PC817 Optocouplers
4 x 1N4004 Diodes
4 x Relays
2 x AC-DC (5v 2A output) Power Modules

Great! From what you just explained, makers could use your circuit for lower voltage applications if they want to avoid mains power too?

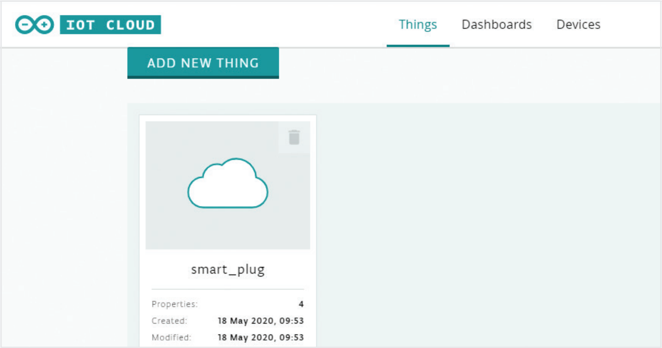
Yes. The power source can come from any source greater than or equal to 5v. You just need to eliminate the two AC/DC power supplies and put a connector for an external power supply that provides 5V or DC/DC transformers if we have a transformer more than 5V.

Tell us how you went about coding the Arduino Nano 33.

The program is very simple, and if we proceed with the Arduino online editor it will generate almost all of the code.

We must enter the name and password of the network we want the Smartplug to access and write the code to manage the buttons.

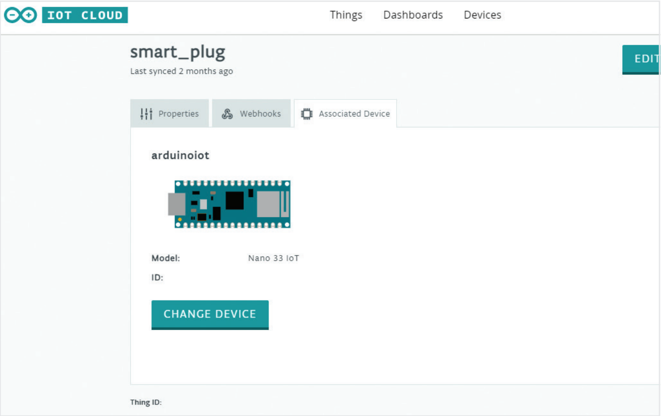
To begin with, you need to create a free profile on Arduino Cloud, and then create a new Thing, which we called 'smart plug'. After that, we added four smartplug type properties with read-and-write permission.



smart_plug

Properties: 4
Created: 18 May 2020, 09:53
Modified: 18 May 2020, 09:53

The last step was to associate a device (our Arduino Nano 33 IoT) to the Thing. We just followed the wizard, which helped make our project safe and immune from attacks as it assigns a unique key to the card and Thing.



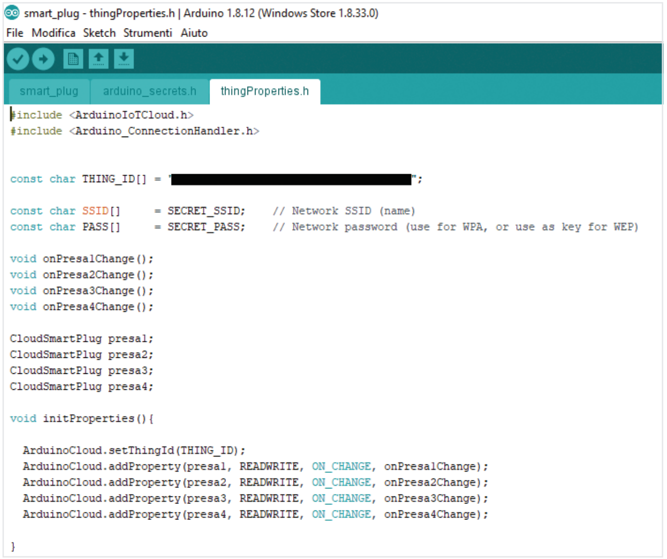
smart_plug
Last synced 2 months ago

arduinoiot

Model: Nano 33 IoT
ID:

Thing ID:

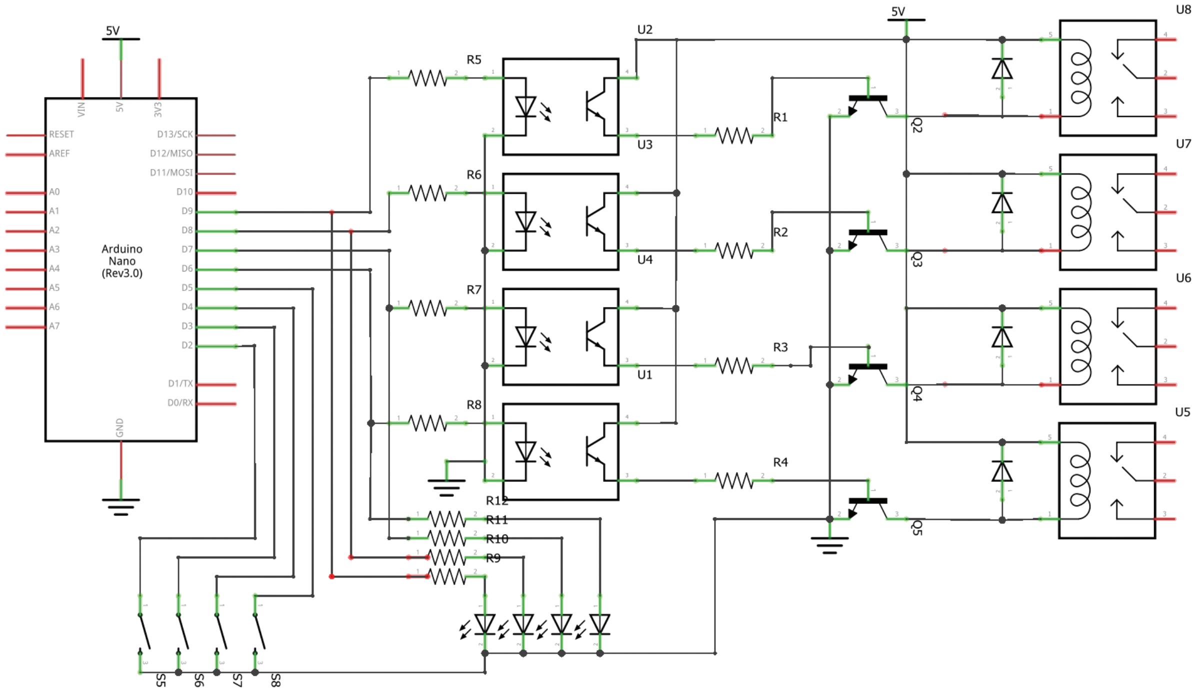
Once done, we select "Edit Sketch" which opens the online editor. That will have also generated the four variables, representing the four properties added in the Thing, as well as our four relays.


smart_plug - thingProperties.h | Arduino 1.8.12 (Windows Store 1.8.33.0)
File Modifica Sketch Strumenti Aiuto
smart_plug
arduino_secrets.h
thingProperties.h

```
#include <ArduinoIoTCloud.h>
#include <Arduino_ConnectionHandler.h>


const char THING_ID[] = "[redacted]";

const char SSID[]     = SECRET_SSID;    // Network SSID (name)
const char PASS[]     = SECRET_PASS;    // Network password (use for WPA, or use as key for WEP)

void onPresa1Change();
void onPresa2Change();
void onPresa3Change();
void onPresa4Change();

CloudSmartPlug presa1;
CloudSmartPlug presa2;
CloudSmartPlug presa3;
CloudSmartPlug presa4;

void initProperties(){

  ArduinoCloud.setThingId(THING_ID);
  ArduinoCloud.addProperty(presa1, READWRITE, ON_CHANGE, onPresa1Change);
  ArduinoCloud.addProperty(presa2, READWRITE, ON_CHANGE, onPresa2Change);
  ArduinoCloud.addProperty(presa3, READWRITE, ON_CHANGE, onPresa3Change);
  ArduinoCloud.addProperty(presa4, READWRITE, ON_CHANGE, onPresa4Change);

}
```

Then it was up to us to decide what to do when an event, linked to one or more of these variables, is triggered. In our case, we send one or more digital gates (D6, D7, D8, D9) to high level. Rest assured, by reading our code, everything becomes much clearer.

The last step is to set up Alexa. First, we create an Amazon profile and access Alexa via the Skill and Games section. Look for Arduino Skill and install it.

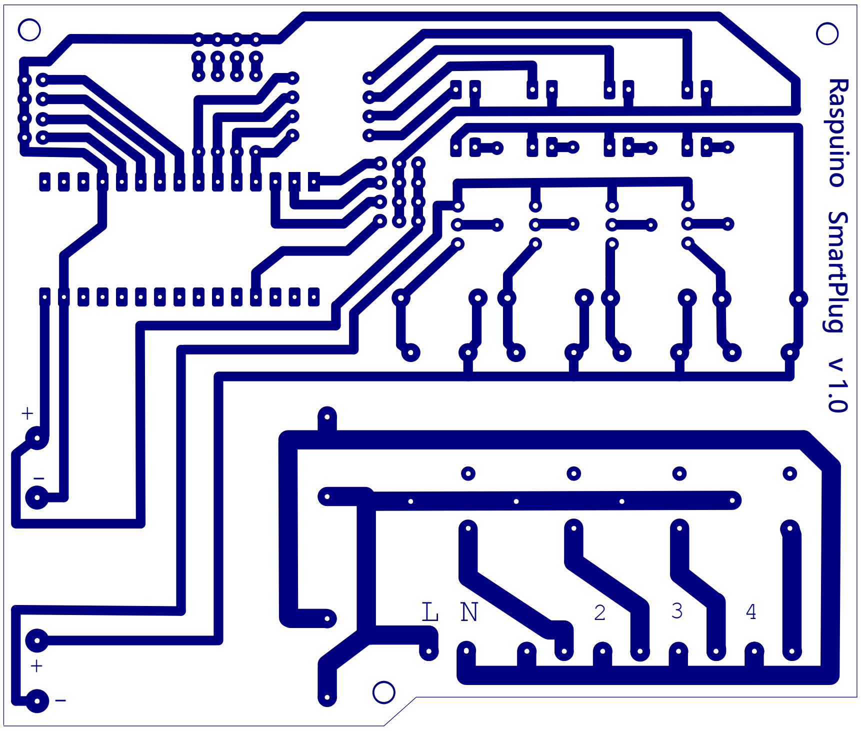


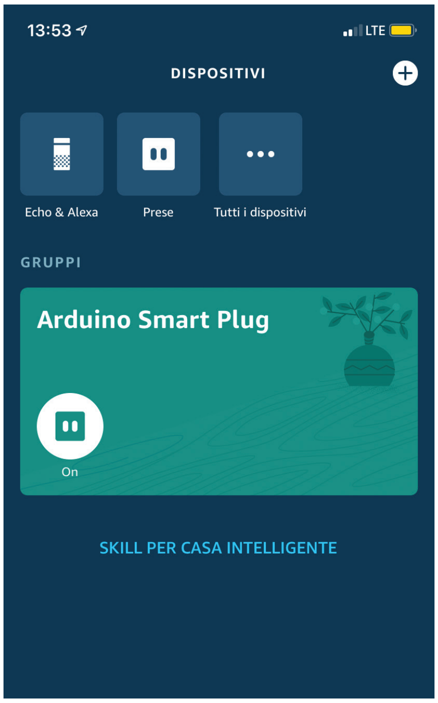
FidoCadJ Design Software

Christian designed his circuit board using FidoCadJ, which is a free vector-based graphical editor for electronics (and more). It works on macOS, Windows, Linus and even Android.

It has a simple schematic capture with a built-in set of electrical symbols and component footprints.

http://darwinne.github.io/FidoCadJ/





13:53 LTE

DISPOSITIVI

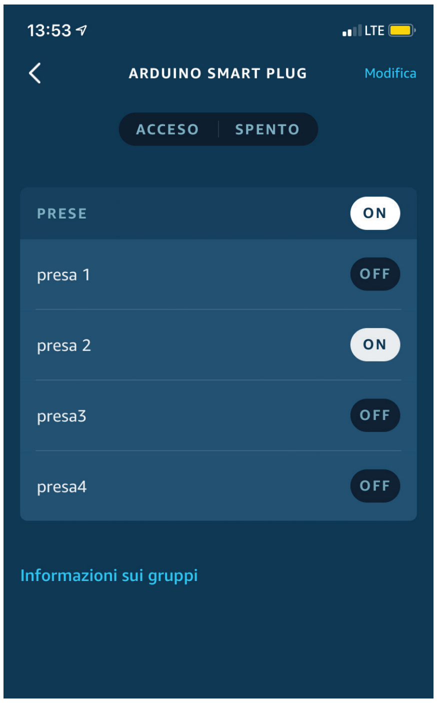
Echo & Alexa | Prese | Tutti i dispositivi

GRUPPI

Arduino Smart Plug

On

SKILL PER CASA INTELLIGENTE



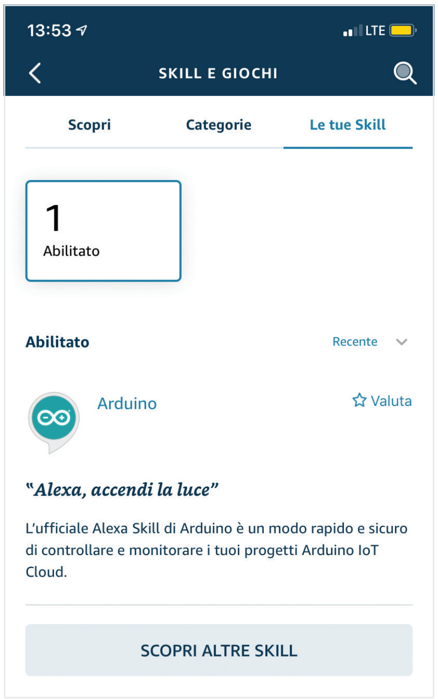
13:53 LTE

ARDUINO SMART PLUG | Modifica

ACCESO | SPENTO

PRESE	ON
presa 1	OFF
presa 2	ON
presa3	OFF
presa4	OFF

Informazioni sui gruppi



13:53 LTE

SKILL E GIOCHI

Scopri | Categorie | Le tue Skill

1 Abilitato

Abilitato | Recente

Arduino | ☆ Valuta

"Alexa, accendi la luce"

L'ufficiale Alexa Skill di Arduino è un modo rapido e sicuro di controllare e monitorare i tuoi progetti Arduino IoT Cloud.

SCOPRI ALTRE SKILL

With the Smartplug on and connected to the WiFi network, we go to the "Add Device" section, select "Sockets" then "Other" and start the search. At the end of the search, if we have done everything right we should have found a new device (our Smartplug). At this point we just have to give a name to the sockets and start using it.

You make the setup sound straightforward. We see that there are also tutorials and articles on the Arduino website too if any of our readers are interested.

WANT MORE?

For the FidoCADJ PCB, Fritzing, and code files, or to discuss this feature, visit:
https://diyode.io/00?xxxx

Reading & Resources:

- **ARDUINO IOT TUTORIALS AND ARTICLES**
  https://www.arduino.cc/en/IoT/HomePage
- **ALEXA SKILLS & ARDUINO**
  https://www.amazon.com/Arduino-LLC/dp/B07ZT2PK2H