بسم الله الرحمان الرحیم

عنوان

عنوان کنترل لوازم خانگی IR به وسیله ی اینترنت از طریق تلفن همراه هوشمند

صورت جلسه دفاع

احتراما این جانب علی مهرابی و علی نریمیسا ورودی سال 94 و 95 در مقطع کارشناسی رشته ی برق که در تاریخ ............. پایان نامه خود را تحت عنوان کنترل لوازم خانگی IR به وسیله ی اینترنت از طریق تلفن همراه دفاع نمودم

امضا علی مهرابی و علی نریمیسا

علی مهرابی

و

علی نری میسا

استاد راهنما : دکتر سید مجتبی موسوی

با تشکر و قدردانی از جناب آقای دکتر موسوی که در این راه کمکمان کردند

جهاد دانشگاهی خوزستان

و خانواده هایمان که دلسوزانه همراهمان بودند

فرم حق انتشار

**تعهد اصالت اثر**

اینجانبان علی مهرابی و علی نریمیسا متعهد میشویم که این پروژه در اینترنت و مجلات اعم از الکترونیک و غیر الکترونیک وجود نداشته و تمام این اثار حاصل دسترنج ما در یک کار گروهی میباشد

**تقدیم به شهدای جهاد دانشگاهی**

فهرست

**فصل اول(معرفی)..................................................................................................4**

1.2بیان مسئله........................................................................................4

1.3 اهمیت و ضرورت...........................................................................4

1.3معرفی قطعات........................................................................................5

1.4توضیحات قطعات و سایت.................................................................10

**فصل دوم(برنامه نویسی ماژول).....................................................................**21

2.1توضیح کد برنامه و کتاب خانه ها و راه اندازی ماژول و کد ...............21

2.2تحریک فیلد ها..............................................................41

**فصل سوم (برنامه نویسی وب)..........................................48**

برنامه نویسی HTML....................................................49

برنامه نویسی جاوا اسکریپت......................................................50

برنامه نویسی Css....................................................................54

تهیه هاست و اپلیکیشن ....................................................57

**فصل چهارم) شماتیک (**

نتیجه..........................................................................55

**هدف ما این است بخشی از زندگی را که فکر میکردیم به آخرت تکامل خود رسیده وارد فاز جدیدی از شکوفایی کنیم تا ثابت کنیم تکامل و شکوفایی برای بشریت حد و مرز ندارد**

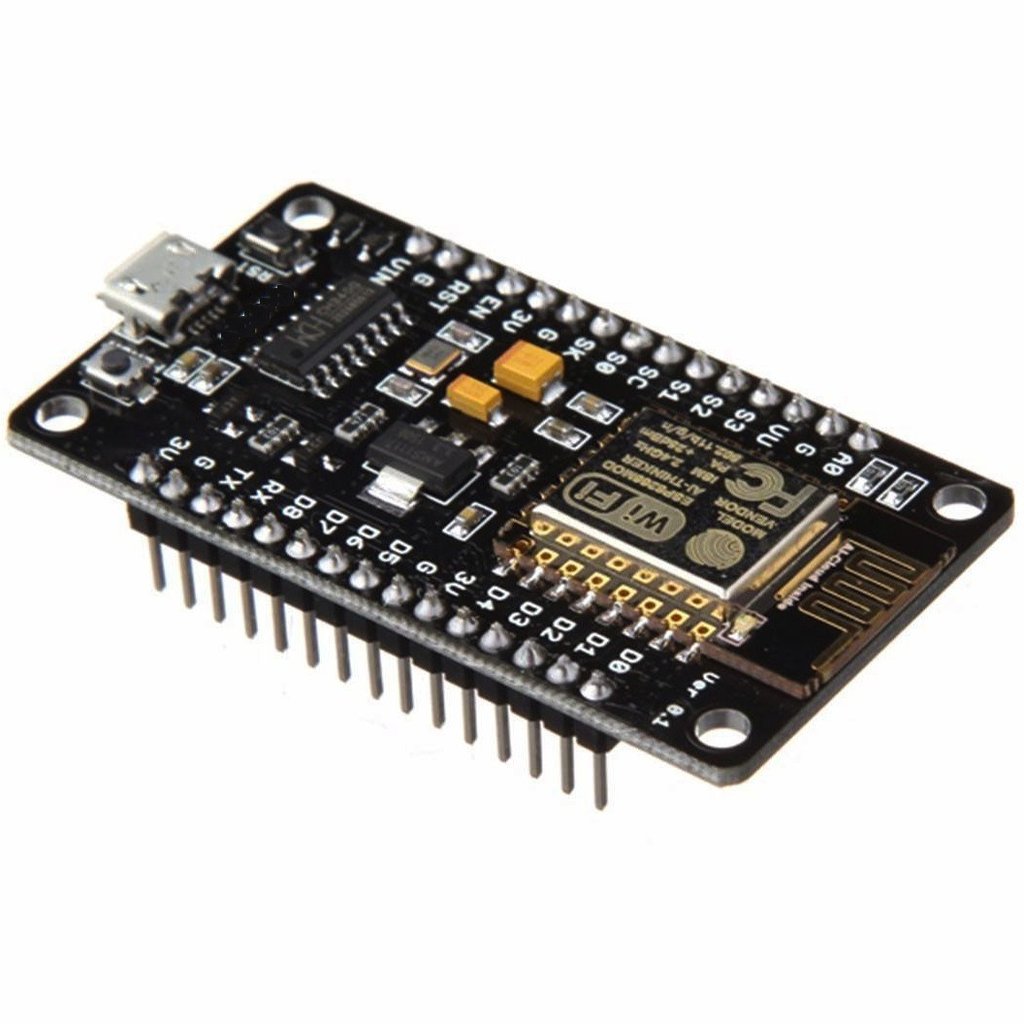
**ALIS IOT PROJECT:**

با رشد روز افزون و جهانگیر اینترنت در زندگی انسان و ورود تکنولوژی های هوشمند بستری با نام IOT به معنای اینترنت اشیا یا internet of things فراهم شده و میتوان وسایل هوشمند را به شبکه ی جهانی اینترنت وصل کرد . در این پروژه تلاش شده که وسایل خانگی که توسط فرستنده مادون قرمز کنترل میشوند را با استفاده از اینترنت و از راه دور با استفاده از دستگاهی که میسازیم کنترل کنیم. امیدواریم این پروژه سر آغاز اتفاقی بزرگ در راه هوشمند سازی خانه ها و محل زندگی ما باشد.

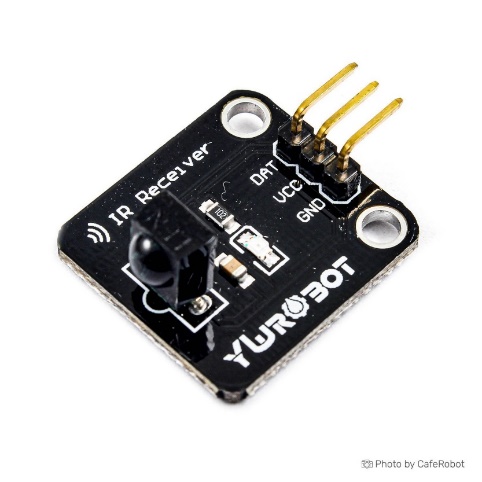
این دستگاه با ماژول esp 8266 به شبکه ی جهانی متصل میگردد و با یک اپلیکیشن میتوان به آن دستور داد

این پروژه دارای :

Esp8266



ماژول گیرنده



Push button



LED RED



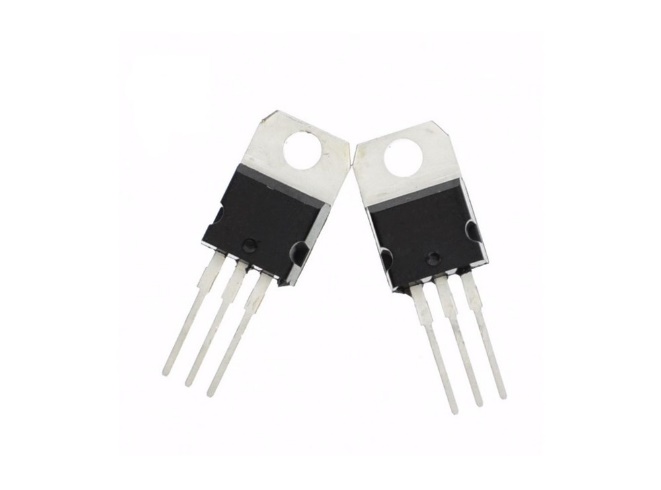
مقاومت ده کیلو و 680 اهم



ترانزیستور npn BC337



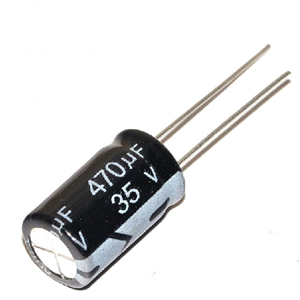
رگلاتور 5 ولت 7805



سینک رگلاتور



خازن 470Uf و 1uF

سوکت تغذیه ماده



صفحه مسی مدار خام یک رو 10\*10



رنگ محافظ مدار سبز (دلخواه)



فرستنده ی IR



هدر پین



و دیگر وسایلی است که به دلیل دم دستی و مشهور بودن از آنها یاد نمیشود

**wemos-esp8266** چیست:

**برد wemos-esp8266 مبتنی بر  Esperessif ESP8266-12E به همراه هسته وای فای است . این برد مجهز به سیستم عامل متن باز و مبتنی بر Lua است. برای پروژه های IoT که به اتصال بی سیم نیاز داریم بسیار مناسب است. این تراشه ارتباط بسیار زیادی با Arduino دارد . هر دوی آنها میکروکنترلری هستند که می شود با استفاده از Arduino IDE  آنها را برنامه ریزی کرد.**

**با توجه به اینکه ESP8266 نسخه جدیدتری نسبت به Arduino است ، تعجب آور نیست که مشخصات قوی تری داشته باشد. یک پردازنده ۳۲ بیتی RISC با سرعت ۸۰ مگاهرتز ، یک RAM  مکمل و پشتیبانی از حداکثر ۱۶ مگابایت حافظه فلش خارجی وجود دارد. این دستگاه به دلیل اندازه کوچکش و پشتیبانی از WiFi برای پروژه های IoT  بسیار مفید است.**

**Thingspeaks.com:**

سایتی است که ما برای پروژه استفاده کردیم در این سایت ماژول ESP ما میتواند data را از آن بخواند و مقادریر data توسط دستور get عوض میشنود:

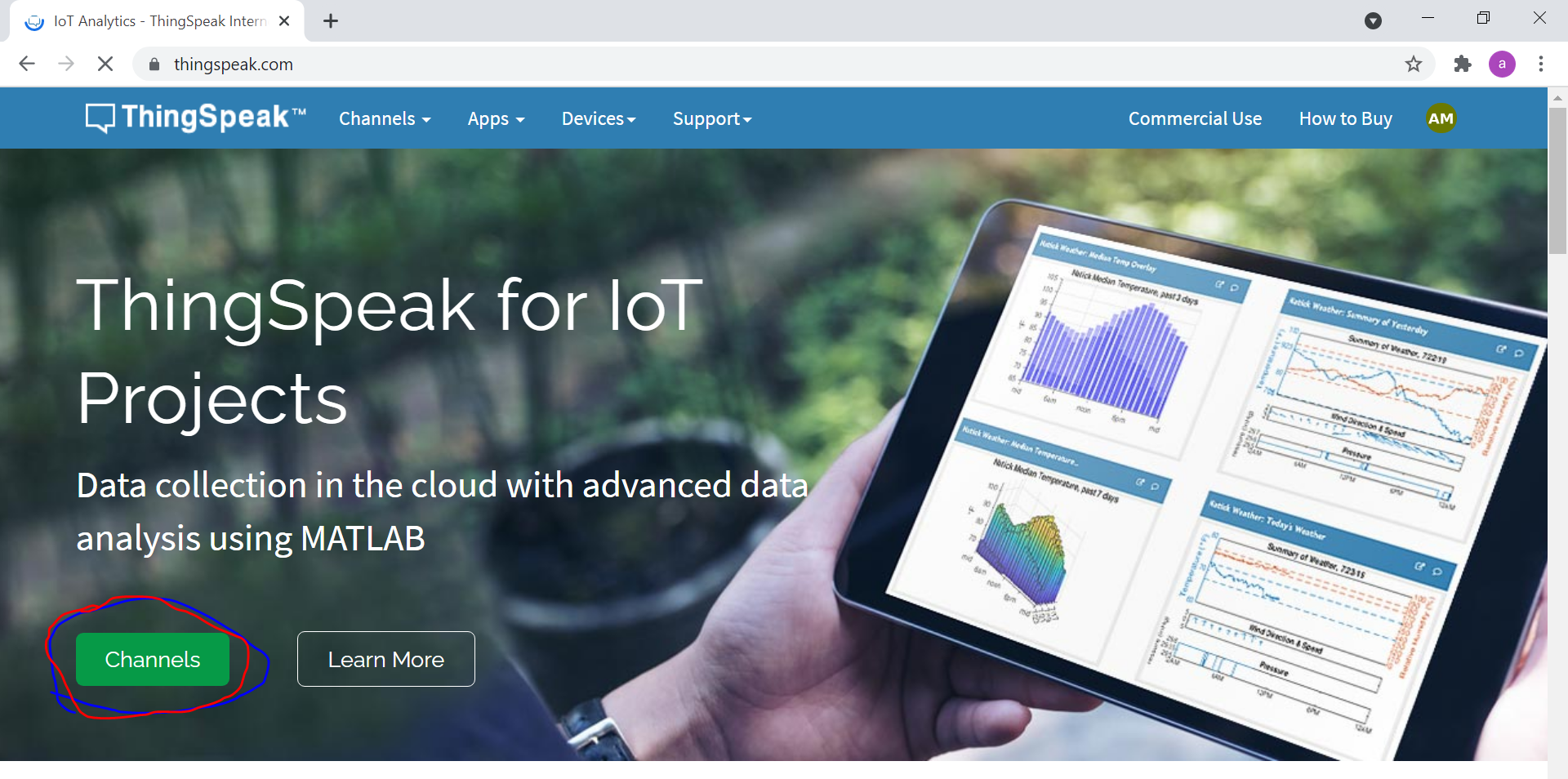
. این سایت دو API KEY متفاوت به هر کاربر میدهد یکی برای دستور نوشتن و دیگری برای دستور خواندن . برای استفاده از سایت باید اول در آن ثبت نام کرد به ما API KEY میدهد و ما به تعداد لازم فیلد در این سایت میسازیم



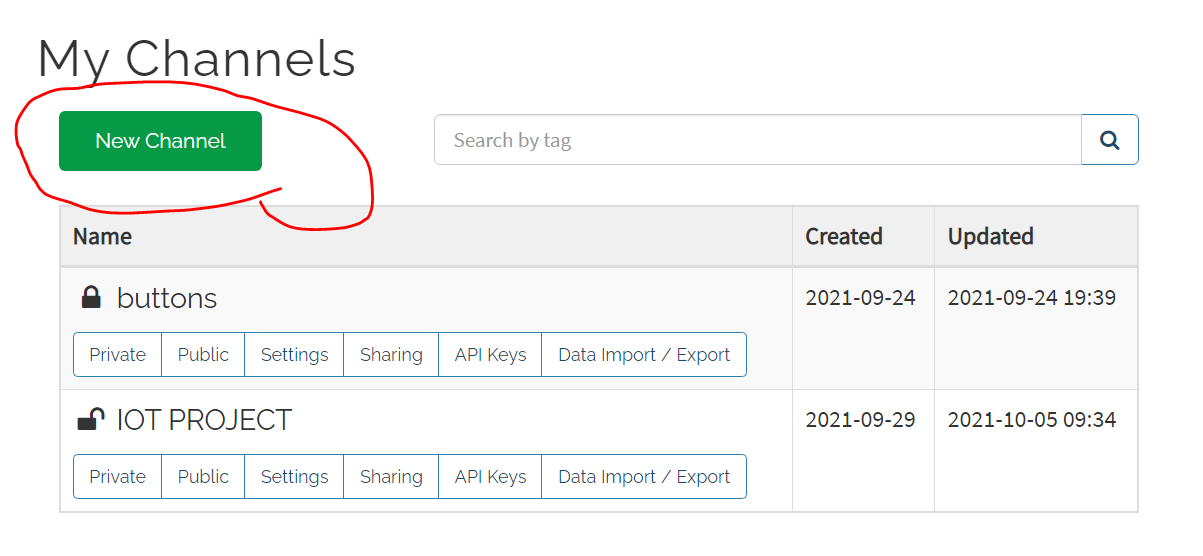
در این شکل سه فیلد درست شده ما میتوانیم با استفاده از اپلیکیشن مقادیر جدید برای هر فیلد از thingpeaks ارسال کنیم و سپس با تغییر مقدار فیلد ESP ، آن را میخواند و با متوجه شدن تغيير مقدار کاری را که برایش برنامه نوشته شده اجرا میکند

هر پانزده ثانیه یک بار میتوان به فیلد ها مقدار جدید داد

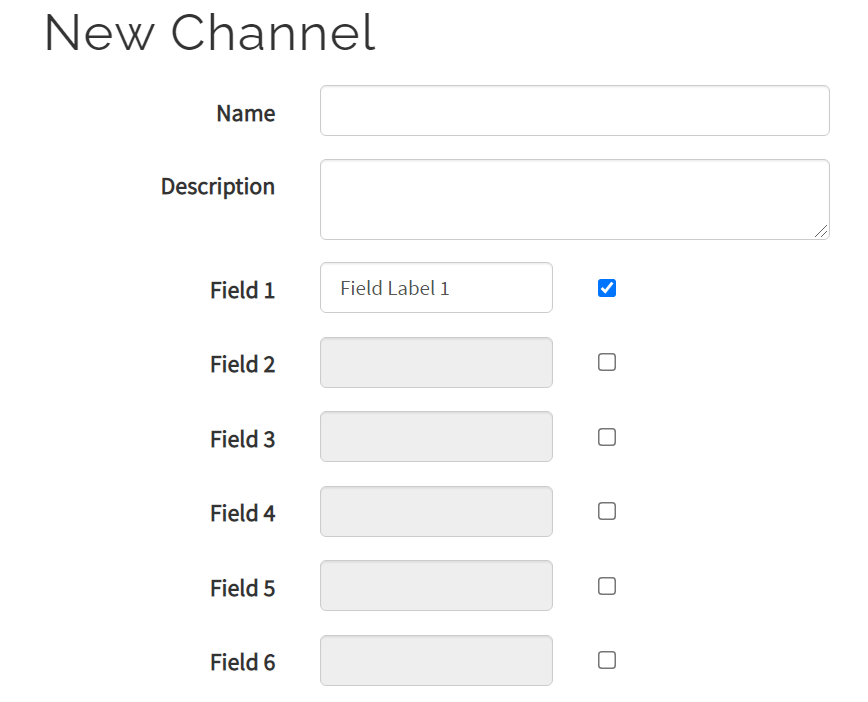
بعد از ساختن اکانت دکمه ی channels را میزنیم



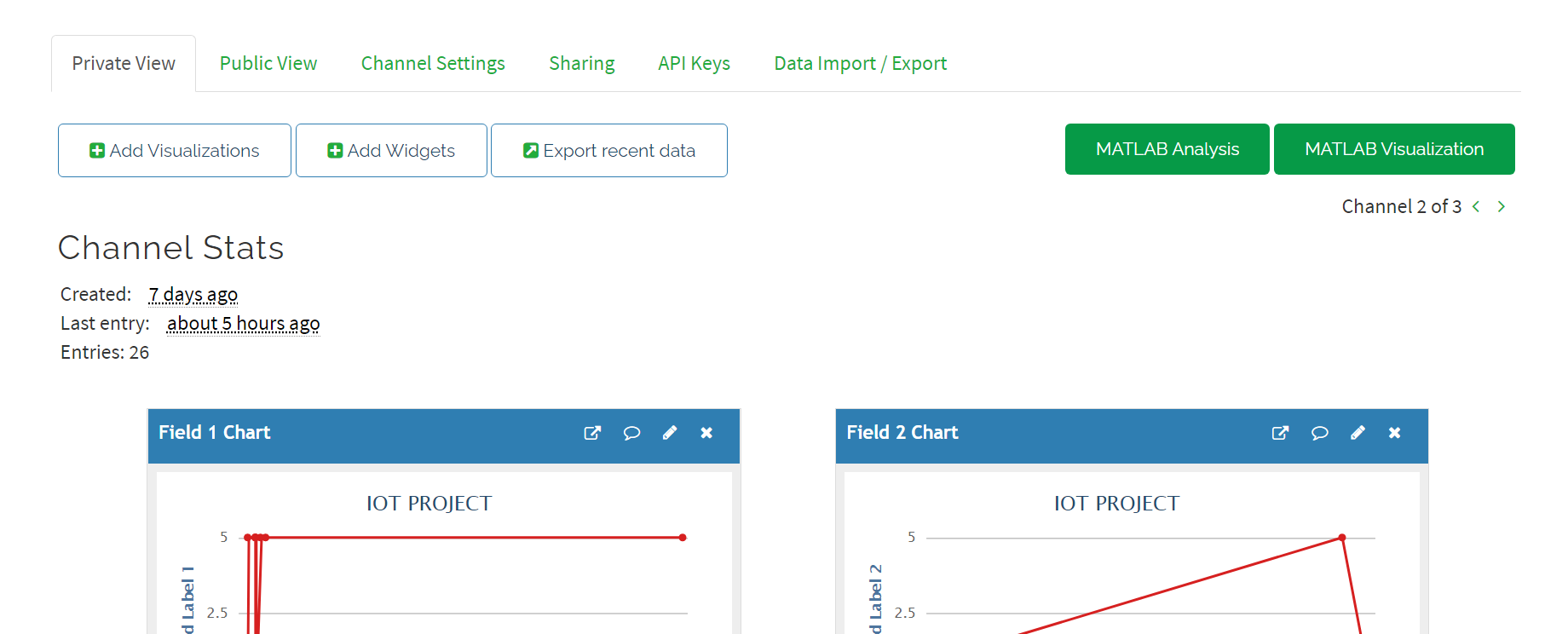
سپس new channel را کلیک میکنیم



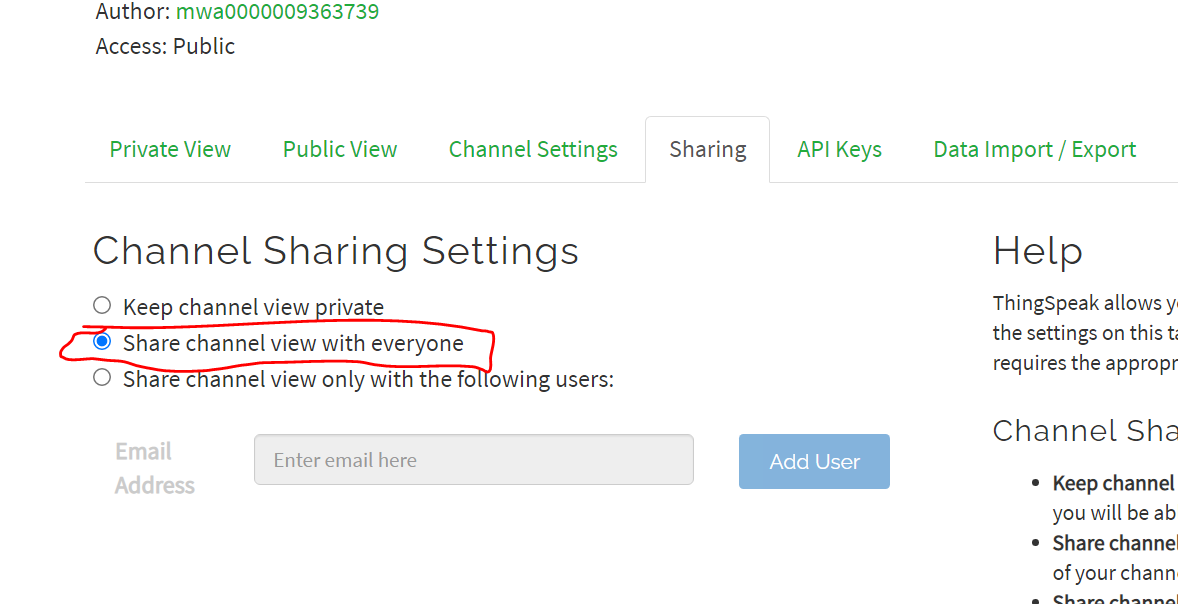
نام و توضیحات و نام و تعداد فیلد ها را اینجا مشخص میکنیم



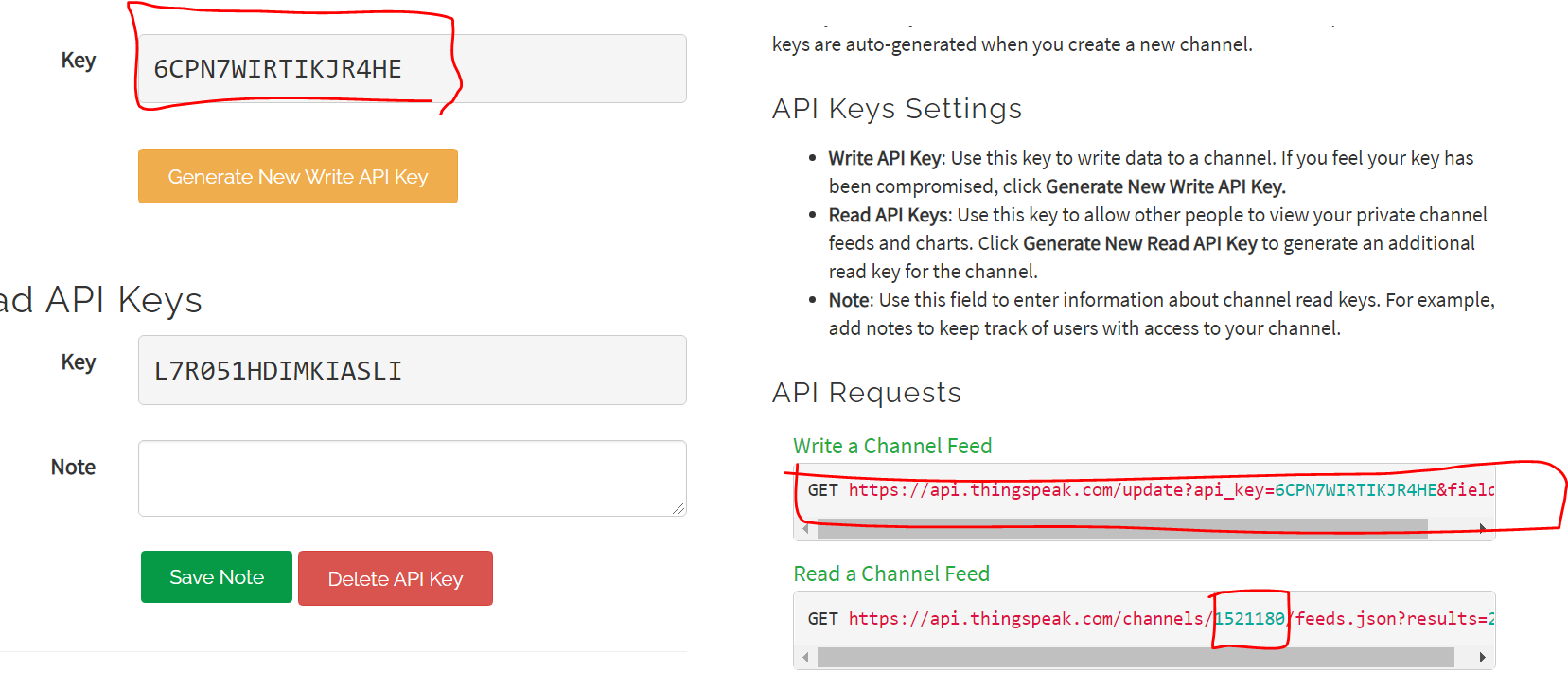
سپس بر روی save channel کلیک میکنیم و بعد از آن باید به این صفحه هدایت شویم



بر روی sharingکلیک کرده و به این صفحه هدایت میشویم و این کلیک را میزنیم تا دسترسی به فیلد ها عمومی شود



سپس به قسمت API Keys رفته و این مقادیر و همچنین این لینک را یادداشت میکنیم



لینک اول

https://api.thingspeak.com/update?api\_key=6CPN7WIRTIKJR4HE&field1=0

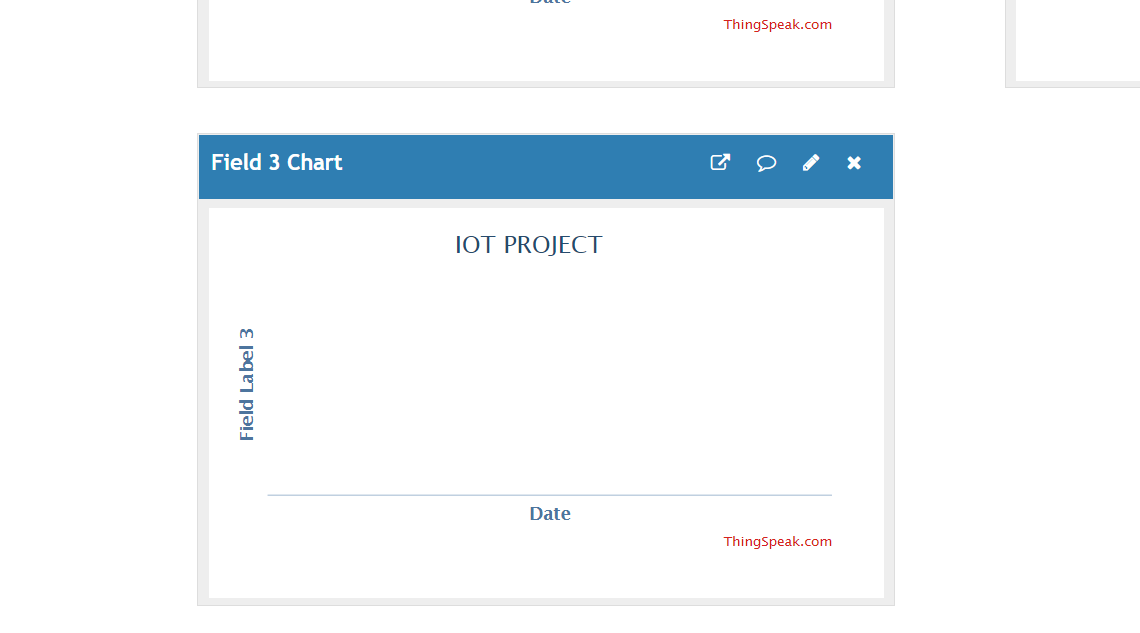
این http ریکوئست برای فرستادن دستور 0 به فیلد 1 است و برای فرستادن دستور 5 به فیلد 2 میتوان دستور زیر را فرستاد

https://api.thingspeak.com/update?api\_key=6CPN7WIRTIKJR4HE&field2=5

با فرستادن این کد ها به thingspeak میتوان تغییر آن ها را در فیلد مشاهده کرد

به عنوان مثال :

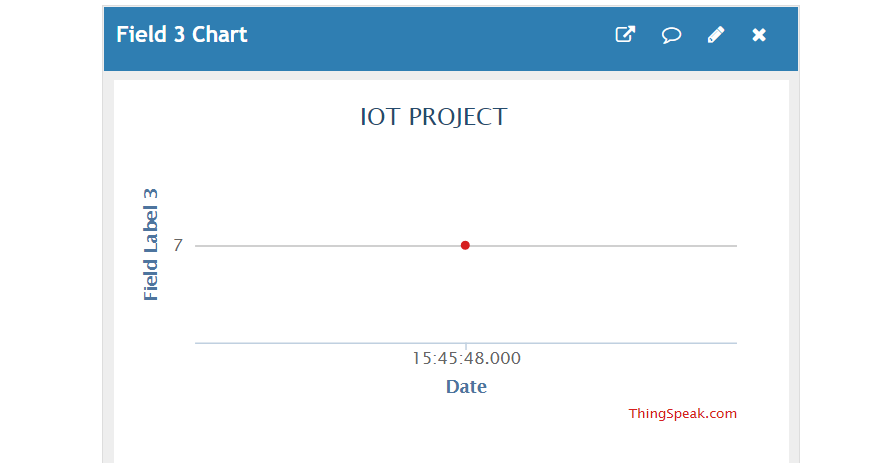
این لینک خالی است



با فرستادن این دستور http

https://api.thingspeak.com/update?api\_key=6CPN7WIRTIKJR4HE&field3=7

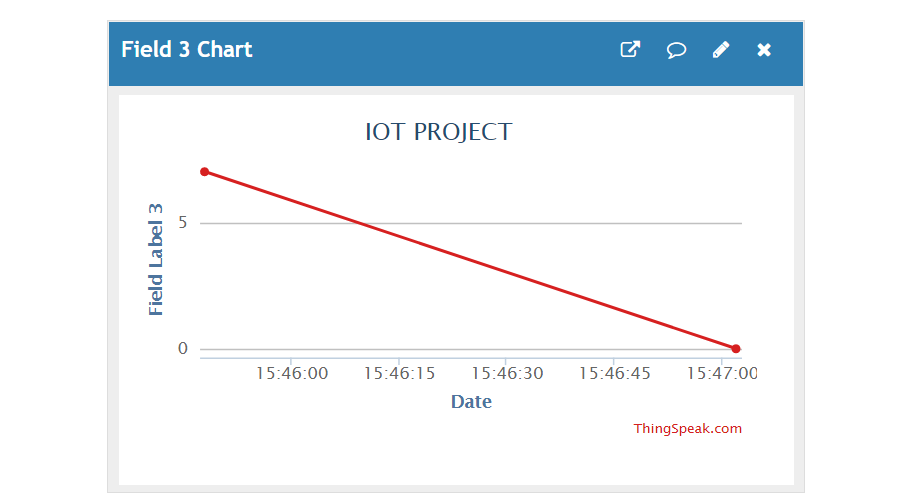
میتوان تغییر مقدار فیلد را مشاهده کرد



و بعد از پانزده ثانیه میتوان دستور جدیدی فرستاد برای مثال این دستور که مقدار صفر را برای فیلد سه میفرستد

https://api.thingspeak.com/update?api\_key=6CPN7WIRTIKJR4HE&field3=0

و این هم نتیجه

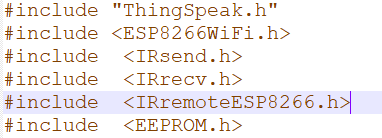


**فصل دوم (برنامه نویسی ماژول)**

## کتابخانه چیست؟

کتابخانه ها یک مجموعه کد می باشند که اتصال سنسور، صفحه نمایش، ماژول و غیره را تسهیل می بخشند. به عنوان مثال، کتابخانه آماده LiquidCrystal برقراری ارتباط با صفحه نمایش LCD کاراکتری را آسان می کند. صدها کتابخانه جدید جهت دانلود در اینترنت موجود می باشد. کتابخانه های داخلی و برخی از این کتابخانه های جدید در مرجع لیست شده اند. جهت استفاده از کتابخانه های جدید، باید آنها را نصب کنید.

اول کتاب خانه ها را وارد میکنیم



**کتاب خانه ی thingspeak**

این کتابخانه برای اتصال ماژول esp به سرور thingpeak است و دارای دستوراتی برای خواندن فیلد هاست

**کتاب خانه ی ESP8266WIFI.h**

این کتاب خانه برای اتصال ESP به وایفای گذاشته شده و دارای دستورات مربوط به وایفای است

**کتاب خانه ی IRsend.h**

این کتاب خانه برای فرستادن دیتای ir به کار میرود

**کتاب خانه ی IRrecv.h**

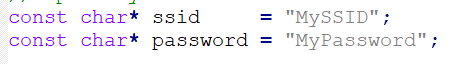
این کتاب خانه برای دریافت دیتای ir به کار میرود

**کتاب خانه ی IRremoteESP8266.h**

این کتاب خانه برای توابع کلی و استراکچر های دیتای IR استفاده میشود

**کتاب خانه EEPROM.h**

این کتاب خانه برای توابع EEPROM دستورات خواندن نوشتن و آپدیت EEPROM استفاده میشود هر خانه ی EEPROM هشت بیت است و میتواند تا مقدار 255 را در خود زخیره کند eeprom حافظه ماندگار است و nodemcu 4 کیلو بایت حافظه eeprom دارد.

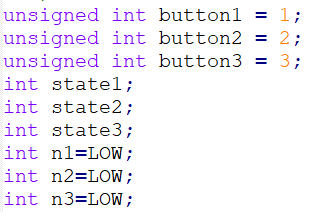


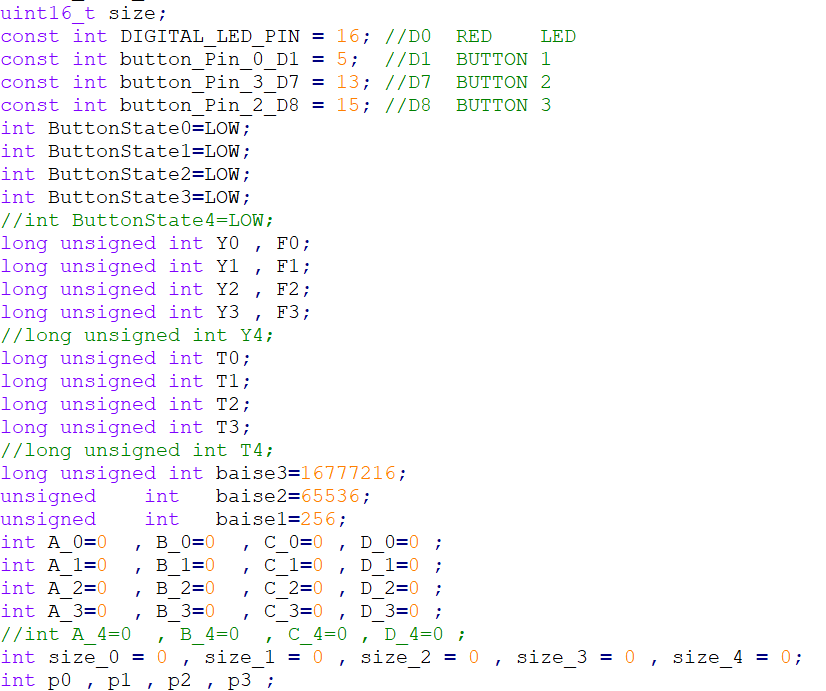
با این دستور اول مقدار username و password را در متغیر های اشاره گر ذخیره میکنیم

با این دستور مقدار ای دی channel را ذخیره میکنیم

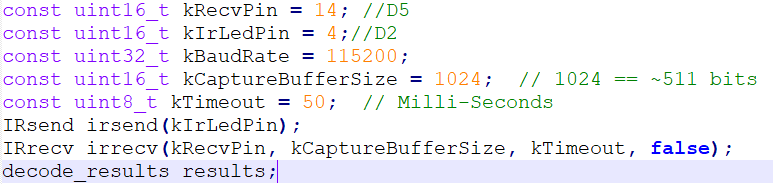


حال متغیر ها را وارد میکنیم





در اینجا پایه فرستنده و گیرنده را انتخاب میکنیم و تنظیمات فرستنده و گیرنده را اعمال میکنیمboadrate را اعمال میکنیم

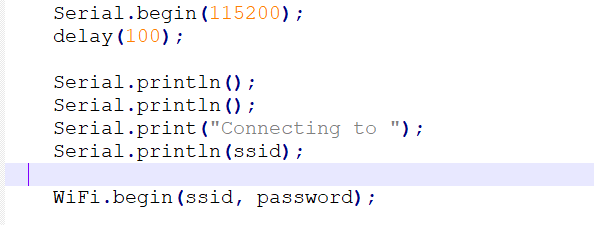


و در آخر اینجا وضعیت وای فای را بر روی کلاینت میگذاریم

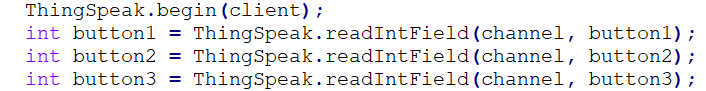


حال در بخش setup

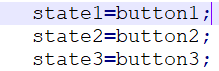
ابتدا boudrate سریال رو مشخص میکنیم سپس در مقدار اشاره گر user , password را وارد به wifi.begin وارد میکنیم



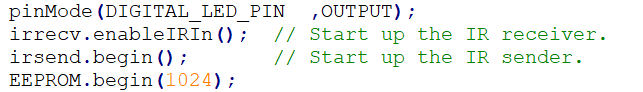
سپس با این دستور ارتباط با سرور thingspeak انجام میشود و با سه دستور بعدی مقدار اولیه ی سه فیلد به صورت اینتیجر در متغیر ها ریخته میشود



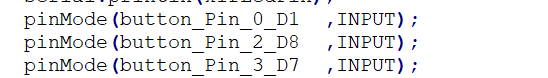
مقدار button ها رو درون state ها میریزیم برای مقایسه ی رخداد تغییرات در صورت تحریک فیلد ها



در اینجا پایه ی D0 خروجی کرده و IR reciver و IR sender را فعال میکنیم و بعد از ان حافظه ی EEPROM را اماده میکنیم



در اینجا پایه های D1 , D8 , D7 را ورودی میکنیم



در اینجا DATA 32 bit IR را که درEEPROM ذخیره کردیم باز گردانی میکنیم از آنجایی که هر خانه از حافظه ی EEPROM فقط هشت بیت است باید عدد 32bit را مجزا به چهار بخش هشت بیت کرد و در آخر آن ها را به هم چسباند

که در این قسمت کد آن ها را به هم می‌چسبانیم و جلوتر عملیات جدا سازی را انجام میدهیم

در واقع ما در اینجا عدد رو به مبنای 256 تبدیل و ان را ذخیره کردیم و اینجا عدد را دوباره به مبنای ده بر میگردانیم

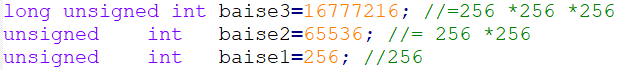
برای اینکار باید رقم اول از مبنای 256 را در یک ضرب کنیم

به علاوه ی رقم دوم در مبنای 256 ضرب در توان یک مبنا بکنیم که خود 256 است

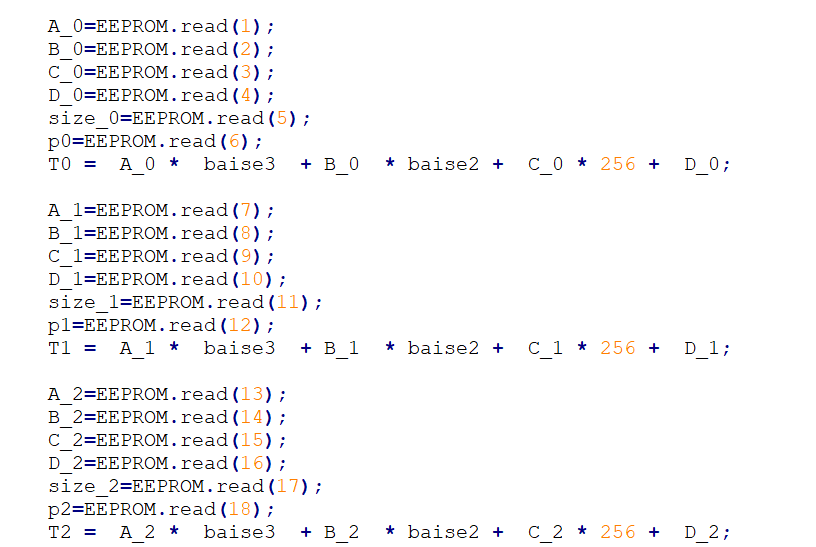
آن را با رقم سوم مبنای 256 ضرب در( 256 به توان دو =65536 ) جمع کنیم

بعد آن را با رقم چهارم مبنای 256 ضرب در( 256 به توان سه= 16777216) جمع کنیم

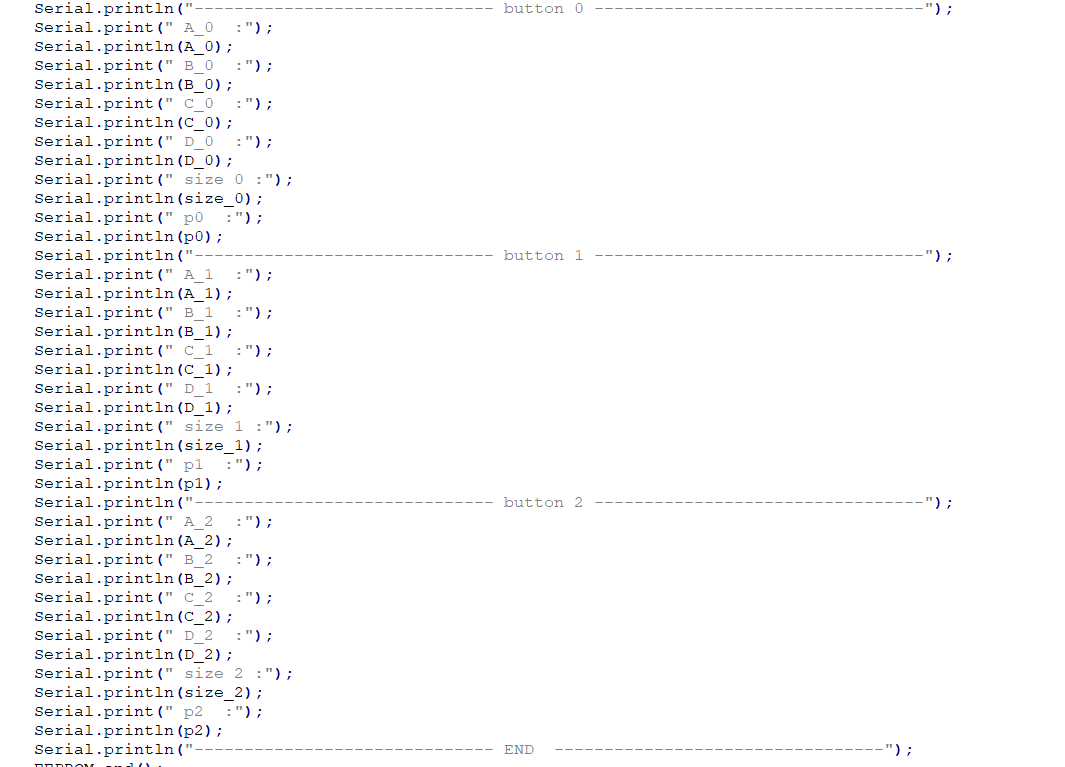
عدد حاصل همان مقدار 32 bit در مبنای ده است



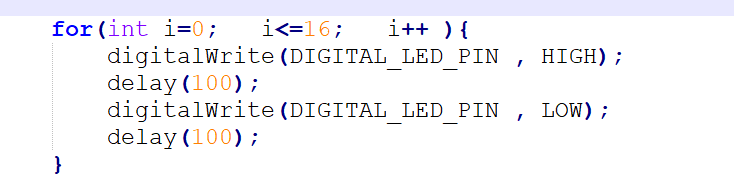
مقدار سایز و مقدار p مربوط به مشخص شدن پروتکل ارسالی هر دیتا میباشد



حال مقدار EEPROM در سریال نمایش داده میشود

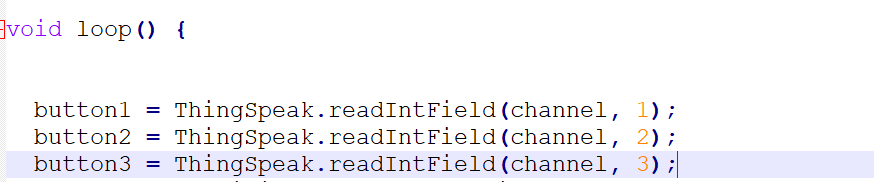


سپس بعد از تمام این ها LED 16 بار چشمک میزند به نشان اینکه تمام کار ها به درستی انجام شده

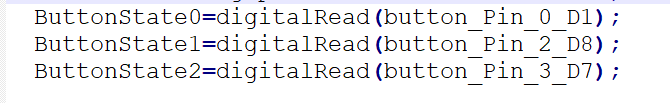


**بخش لوپ**

در اینجا دوباره دیتای مقدار هر فیلد را می‌خوانیم و درون buttons میریزیم



حال وضعیت push buttons های مدار که به پین های D1,D8,D7 وصل هستند را میخوانیم و درون متغیر های ButtonStates میریزیم



**بعد وارد if میشویم**

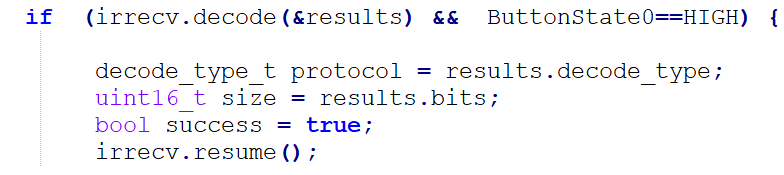
اگر سیگنال دریافت شد و 0pushbutton فشار داده شده باشد

نوع متغیر را در استراکچر پیدا کن

اندازه ی متغیر رو بریز توی size

مقدار success را true کن

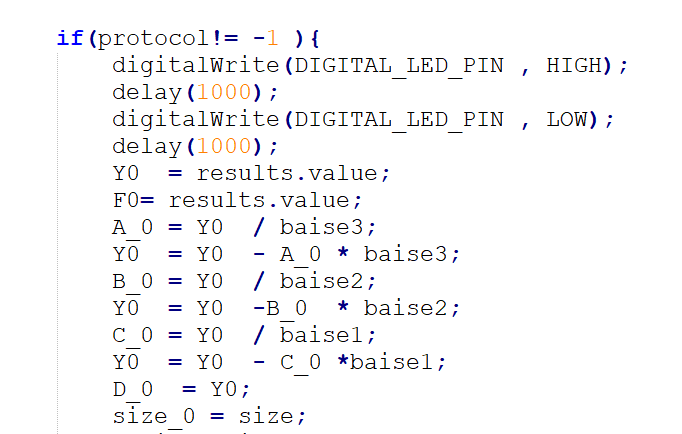
دریافت را متوقف کن



اگر DATA نا معلوم (-1) نبود (هر نوع DATA مقدار مثبتی را در استراکچر دارد و DATA UNNOWN مقدار -1 دارد )

LED را یک ثانیه روشن و خاموش کن تا بدانیم دیتا تعریف شده است

سپس data را شکسته و به مبنای 256 تبدیل میکنیم

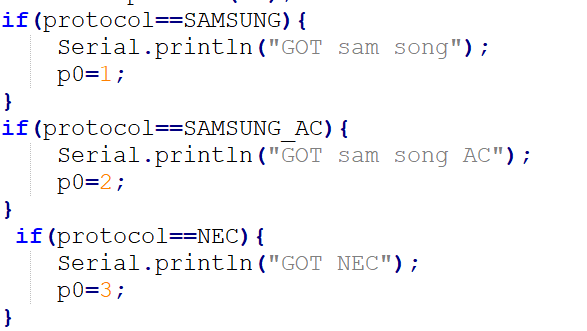


حال مینویسیم اگر پروتکل sumsong بود p=1

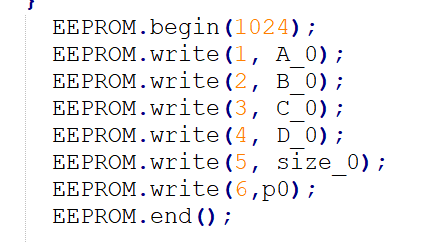
اگر پروتکل sumsong\_ac بود p=2

اگر پروتکل NEC بود p=3

**در این پروژه این سه پروتکل را میتوان ارسال کرد**



و بعد EEPROM را آغاز کرده مقادیر را در حافظه ریخته و بعد ان را میبندیم



**If بعدی pushbutton1**

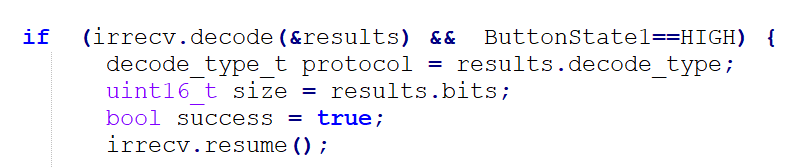
اگر سیگنال دریافت شد و 1pushbutton فشار داده شده باشد

نوع متغیر را در استراکچر پیدا کن

اندازه ی متغیر رو بریز توی size

مقدار success را true کن

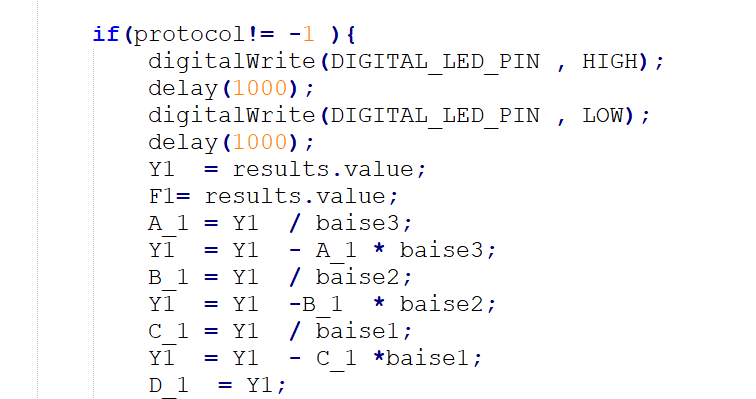
دریافت را متوقف کن



اگر DATA نا معلوم (-1) نبود

LED را یک ثانیه روشن و خاموش کن تا بدانیم دیتا تعریف شده است

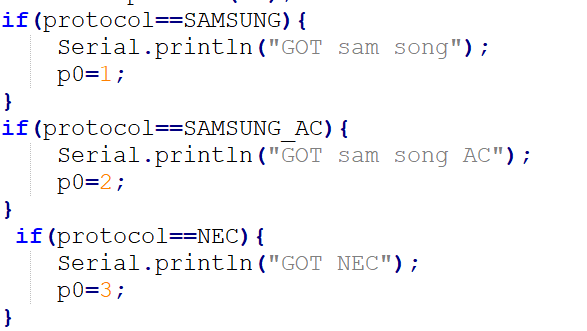
سپس data را شکسته و به مبنای 256 تبدیل میکنیم



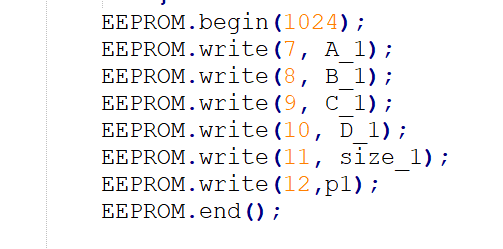
حال مینویسیم اگر پروتکل sumsong بود p=1

اگر پروتکل sumsong\_ac بود p=2

اگر پروتکل NEC بود p=3



و بعد EEPROM را آغاز کرده مقادیر را در حافظه ریخته و بعد ان را میبندیم



**ifبعدی pushbutton2**

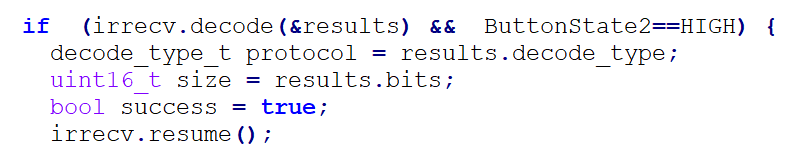
اگر سیگنال دریافت شد و 2pushbutton فشار داده شده باشد

نوع متغیر را در استراکچر پیدا کن

اندازه ی متغیر رو بریز توی size

مقدار success را true کن

دریافت را متوفق کن



اگر DATA نا معلوم (-1) نبود

LED را یک ثانیه روشن و خاموش کن تا بدانیم دیتا تعریف شده است

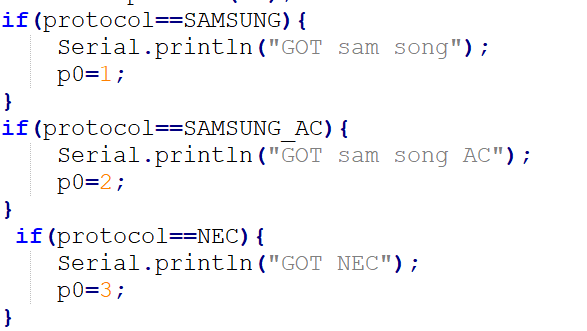
سپس data را شکسته و به مبنای 256 تبدیل میکنیم



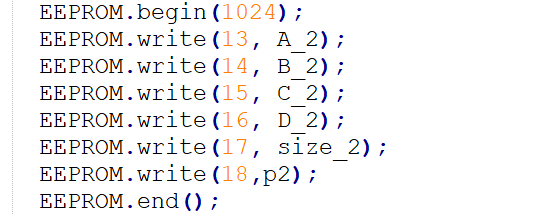
حال مینویسیم اگر پروتکل sumsong بود p=1

اگر پروتکل sumsong\_ac بود p=2

اگر پروتکل NEC بود p=3



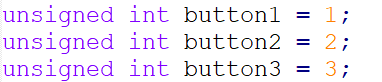
و بعد EEPROM را آغاز کرده مقادیر را در حافظه ریخته و بعد ان را میبندیم

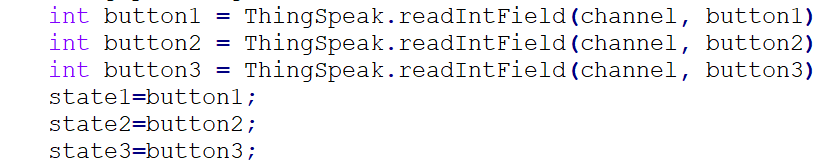


**تحریک فیلد ها**

**حال اگر مقدار فیلد ها تغییر کند :**

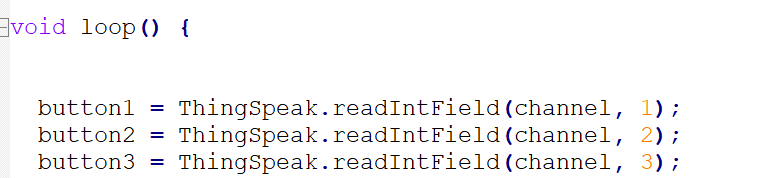
از آنجایی که مقدار فیلد ها در بخش SETUP اول با دستور





در button و از button به state ریخته میشوند

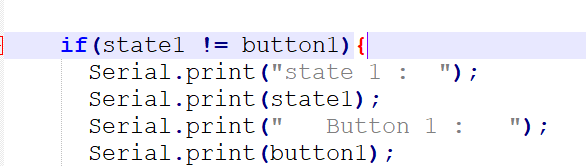
و در قسمت loop هم دوباره مقدار فیلد خوانده میشود و در داخل button ها ریخته میشوند :



اگر مقدار T+1 button ها نا مساوی با مقدار T button که(button(T)=state)

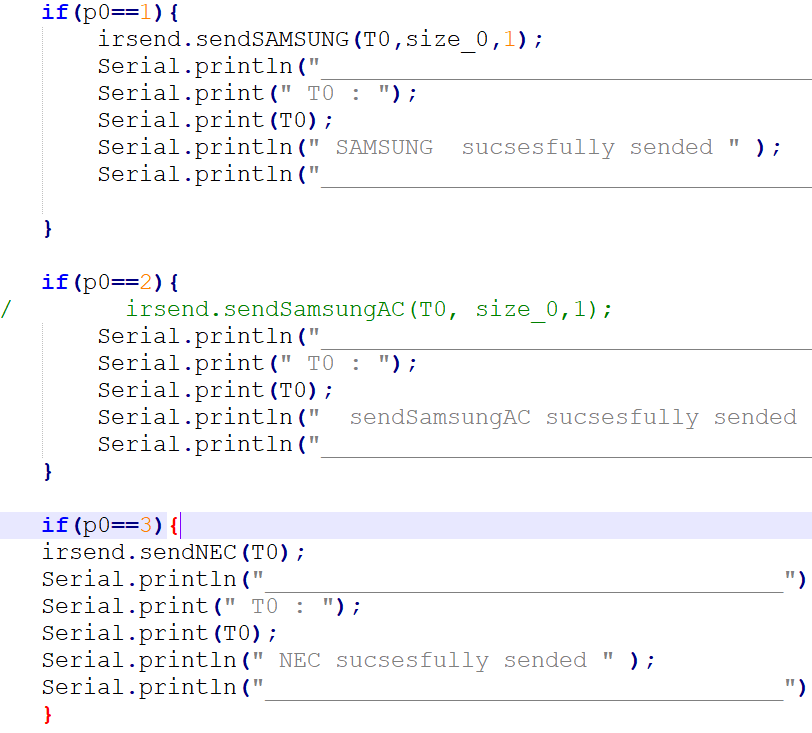
میتوان با یک شرط if شرایطی اعمال کرد که با تحریک شدن فیلد یک عملیات انجام شود

در if به فیلد یک مربوط میباشد و در صورتی که فیلد یک تحریک شود این If اجرا میشود

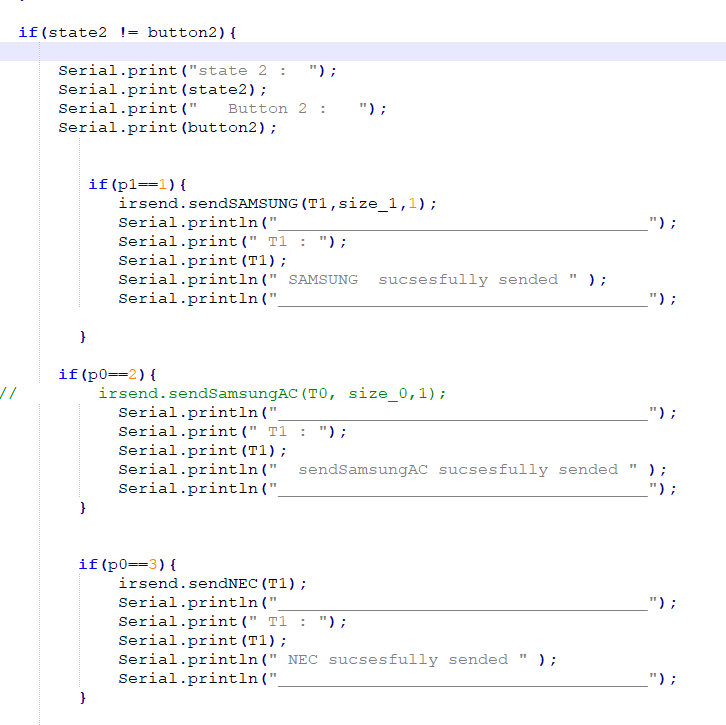


و درون آن if یک if دگر میگذاریم که اگر p=1 بود با پروتکل Samsung ارسال کن

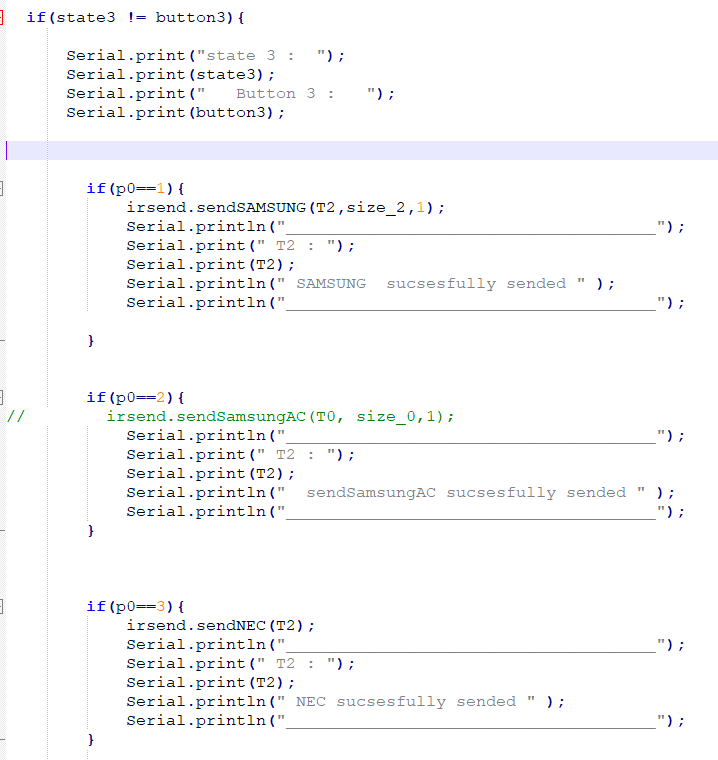
و اگر p=3 بود با پروتکل NEC ارسال کن



**اگر فیلد دو تحریک شد :**



**و اگر فیلد سه تحریک شد :**



**و در آخر :**



یک ثانیه مکث...

بعد مقدار state را برابر با مقدار button میکنیم تا این دو مجددا برابر شده و شرط if بدون تحریک فیلد فعال نشود

هر یک ثانیه یک بار فیلد های thingspeak توسط ماژول Esp خوانده میشود

**شیوه ی تحریک فیلد ها:**

برای تحریک فیلد ها باید دستور http را ارسال کرد که در ان اشاره شده باشد که مقدار جدید فیلد چقدر باشد

دستوراتی به مانند این ها

https://api.thingspeak.com/update?api\_key=6CPN7WIRTIKJR4HE&field1=0

https://api.thingspeak.com/update?api\_key=6CPN7WIRTIKJR4HE&field1=5

https://api.thingspeak.com/update?api\_key=6CPN7WIRTIKJR4HE&field2=0

https://api.thingspeak.com/update?api\_key=6CPN7WIRTIKJR4HE&field2=5

https://api.thingspeak.com/update?api\_key=6CPN7WIRTIKJR4HE&field3=0

https://api.thingspeak.com/update?api\_key=6CPN7WIRTIKJR4HE&field3=5

فصل سوم (برنامه نویسی وب)

.

.

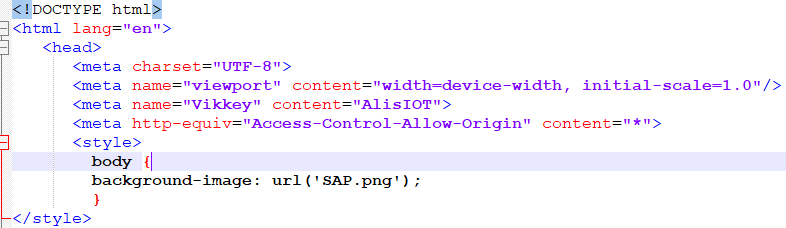
هدف این است که در سایت برنامه ایی با کمک html و css و javascript بنویسیم که وقتی مقدار

Field =0 با فشردن دکمه Field را پنج کند و وقتی Field=5 با فشردن دکمه Field را صفر کند

همچنین از آنجایی که thingspeak فقط در هر پانزده ثانیه میتواند یک مقدار بگیرد با فشردن دکمه به مدت 16 ثانیه تمام دکمه ها غیر فعال شوند

به طور ساده تر

وقتی دفعه ی قبل 0 را ارسال کردیم این بار 5 را ارسال میکنیم و برعکس

در اینجا تنظیمات اولیه ی HTML و عکس صفحه ی زمینه لحاظ شده 

در اینجا دکمه ها و نام آنها چینش صفحه و هدر و ...

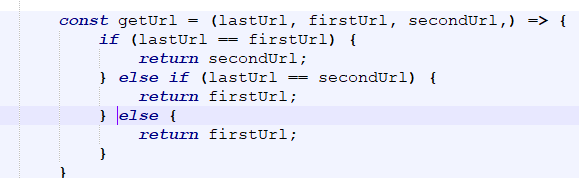


بعد از ان کد JAVASCRIPTS را شروع میکنیم کد JS درون دو <script> نوشته میشود

اول یک تابع مینویسیم :

در صورتی که lastUrl == firstUrl بود تابع getUrl مقدار secendUrl را بر میگرداند

در صورتی که lastUrl == secendUrl بود تابع getUrl مقدار firstUrl را بر میگرداند



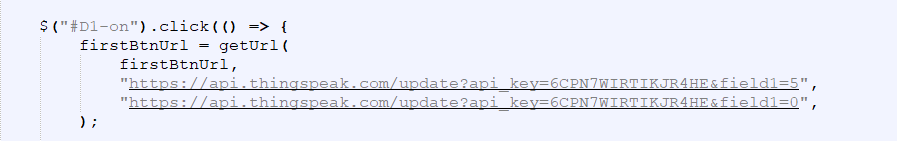
متغیر firstBtnUrl1 را به صورت کاراکتر معرفی میکنیم



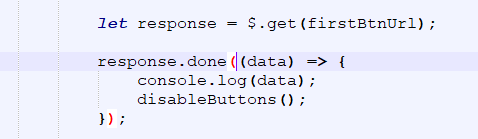
تابع کلیک بر روی button1 را معرفی میکنیم

اگر کلیک زده شد مقدار firstBtnUrl = return get url شود

سپس بعد از ان جای lastUrl قرار بگیرد که با هر بار کلیک مقدار بازگشت تغییر کند

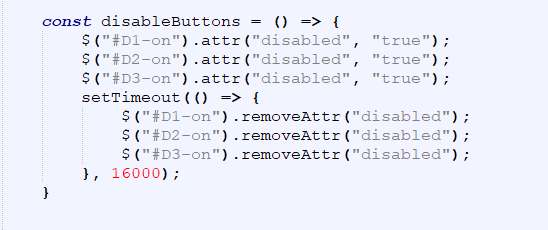


حال در اینجا دستور http را ارسال میکنیم و میگویم اگر ارسال شد با تابع disableButtons دکمه ها را به مدت 16s غیر فعال کن

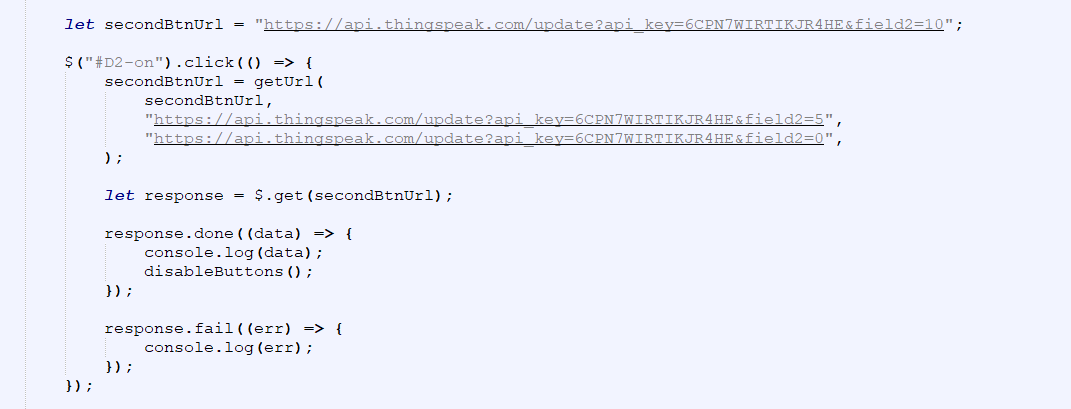


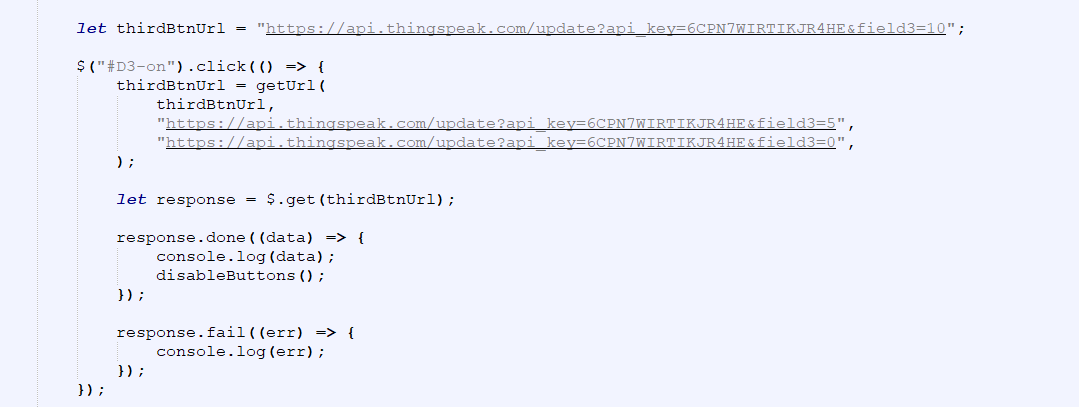
**با تابع disableButtons:**

این تابع میگوید در صورتی که هر کدام از http request ها با موفقیت انجام شد به مدت 16 ثانیه اتریبیوت های دکمه ها را غیر فعال کن



همین فرایند را برای دو button و فیلد بعدی هم انجام میدهیم





**بخش css:**

حال در این برای باتن ها طراحی ایی درست میکنیم که دلنواز تر باشند

Css بین دو تگ <style> نوشته میشود

**دکمه ها در حالت عادی :**

در اینجا رنگ زمینه ی دکمه ها در حالت عادی سبز شده

برای مرز دکمه ها متودی به کار برده نشده

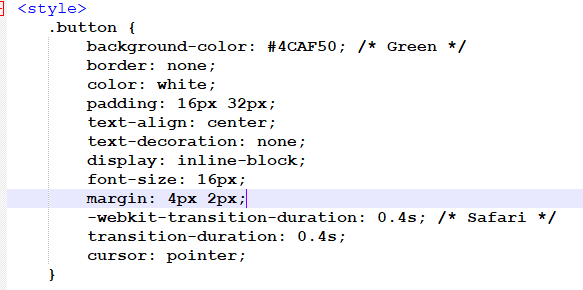
رنگ نوشته درون دکمه ها سفید شده

متن مرکز دکمه نوشته شده

طول عرض دکمه ها در حالت expand و collaps (اول باز شدن سایت ) مشخص شده

نوع دیسپلی دکمه ها مشخص شده

زمان دیسپلی (expand) 0.4 ثانیه تنظیم شده



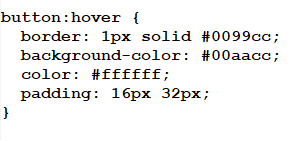
**دکمه ها در حالت نیمه تحریک:**

یک پیکسل مرز

تغيير رنگ 00aacc

زنگ تکست سفید ffffff

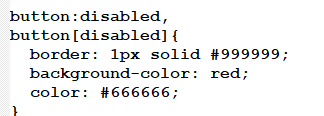
اندازه در حالت نیمه تحریک



رنگ مرز 999999

رنگ زمینه قرمز

رنگ تکست 666666



**تهیه ی هاست :**

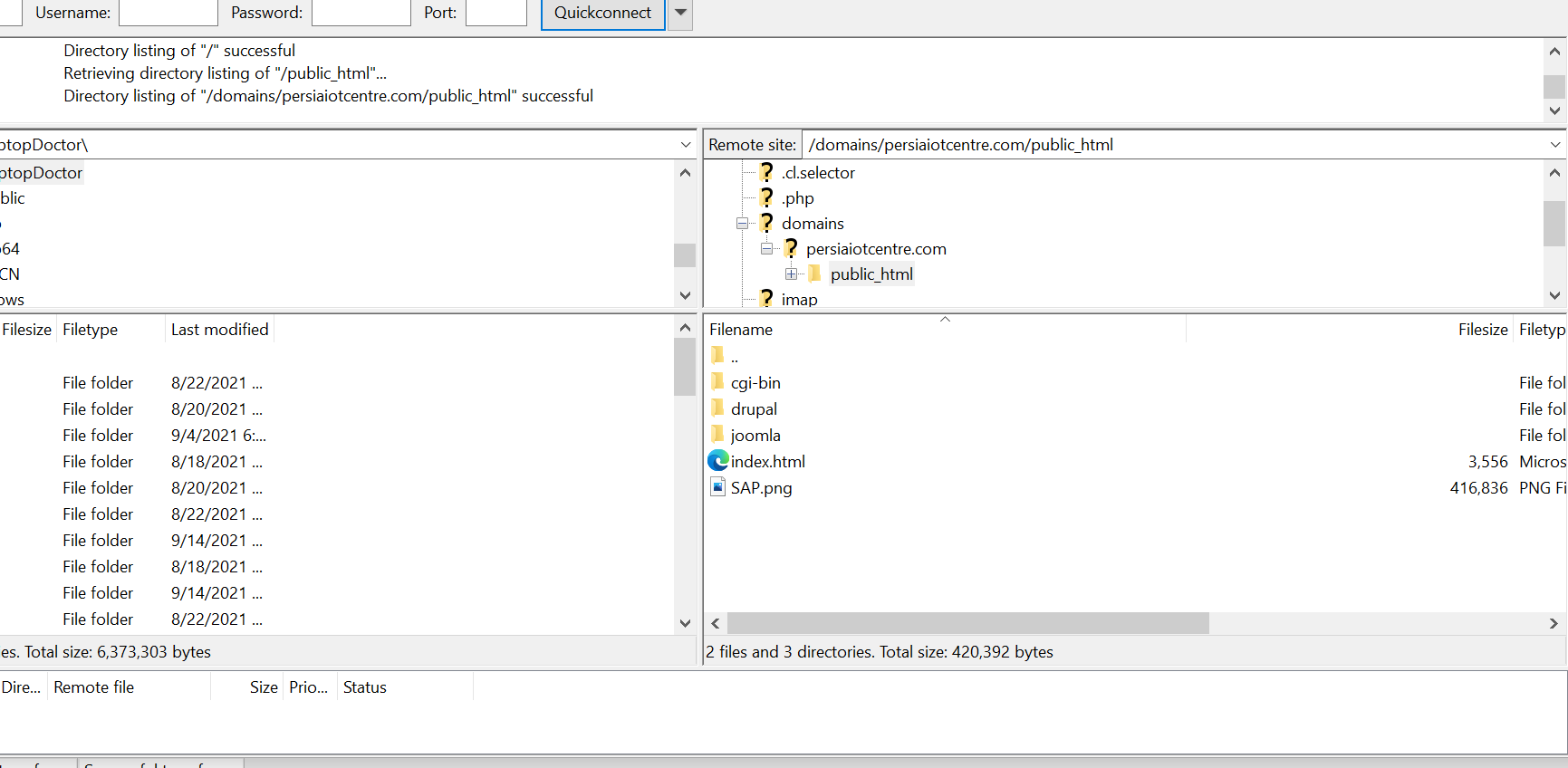
هاست و دامنه از سایت parspack.com تهیه شده

برای آپلود فایل html js برنامه filezilla رو نصب میکنیم

با وارد کردن یوزر و پسورد وارد میشویم

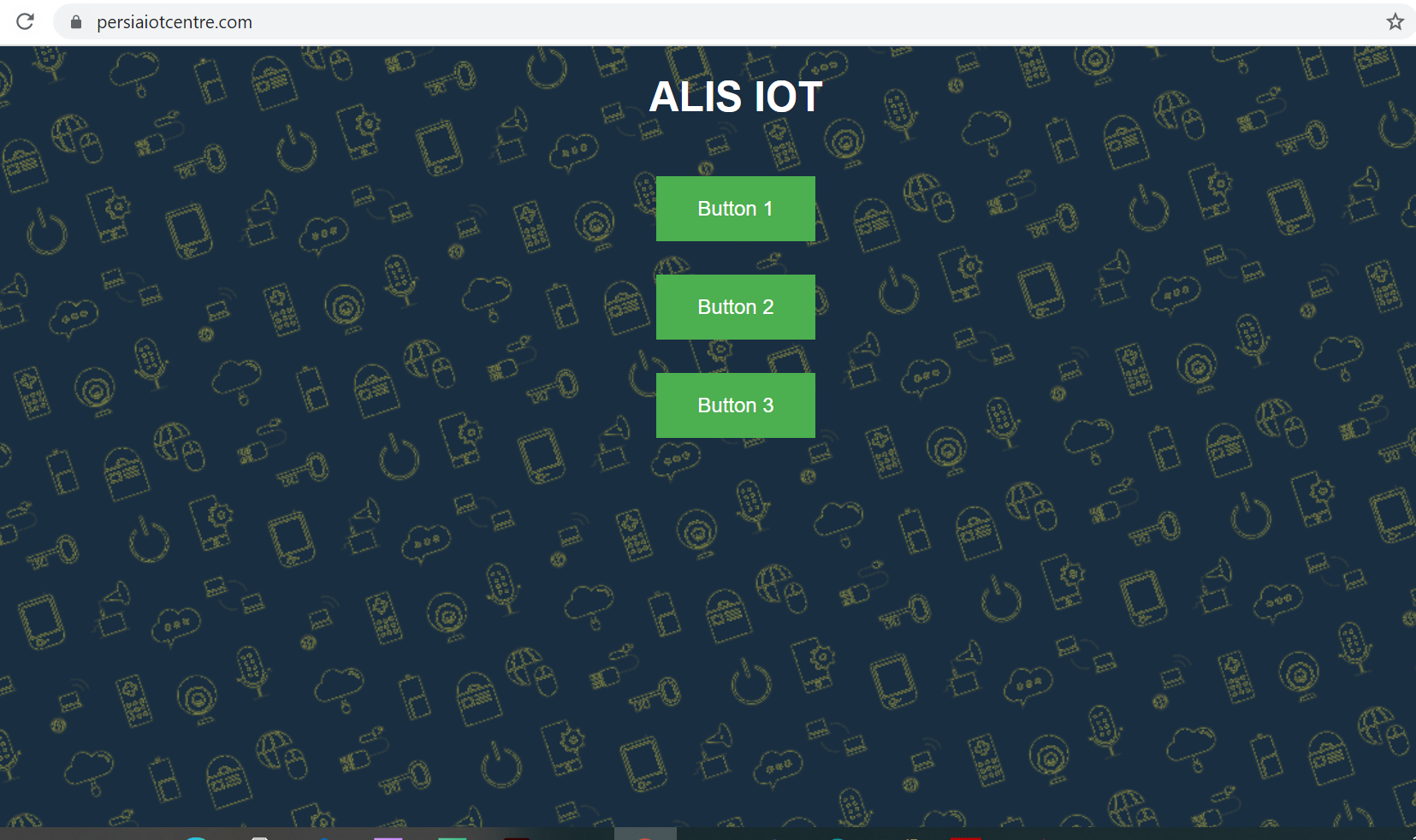
سپس پوشه ی public.html را باز کرده

فایل ها را همراه با عکس پس زمینه درون ان قرار میدهیم



وارد addres دامنه میشویم

<http://www.persiaiotcentre.com>

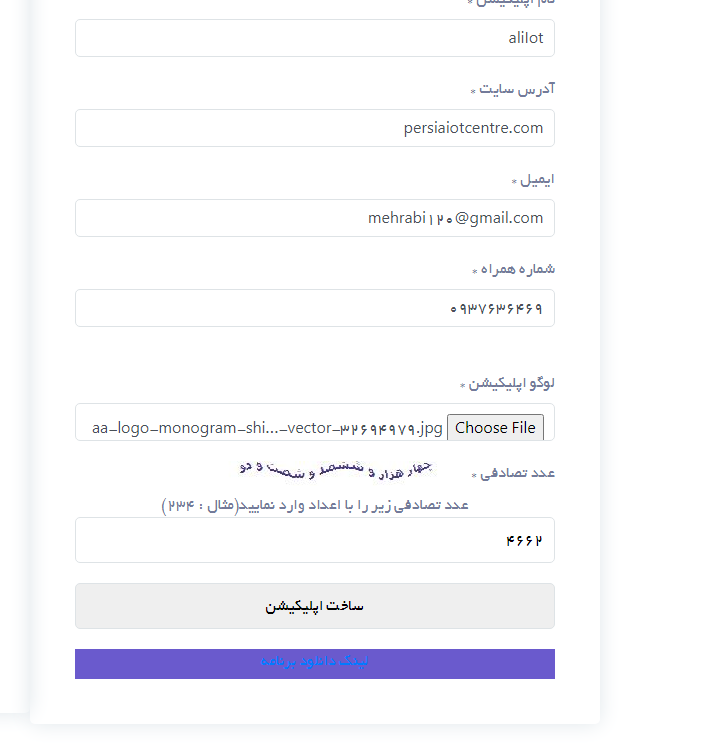


سپس سایت را با استفاده از سایت های اپلیکیشن ساز به اپلیکیشن تبدیل میکنیم

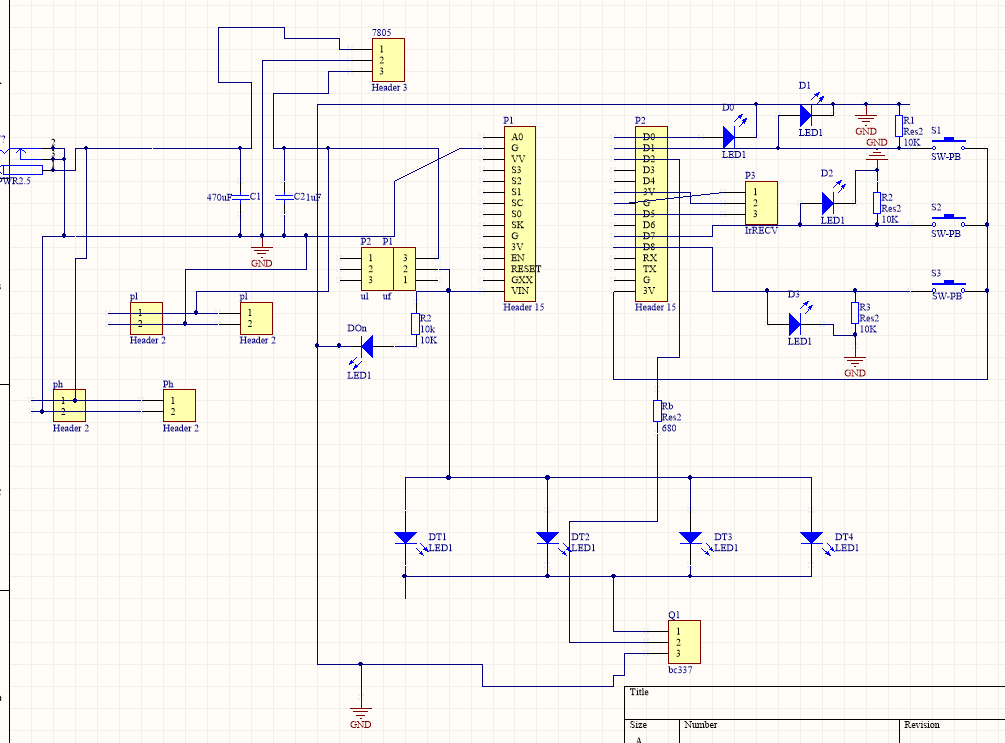


بر روی ساخت اپلیکیشن کلیک کرده

فرم ها را پر میکنیم



**شماتیک :**



پایه های D1 , D8, D7 pull down شده

گیرنده به D5 وصل شده

بیس ترانزیستور های فرستنده ها با یک مقاومت بیس 10k موازی شده و در حالت اشباه قرار گرفته

رگلاتور هم ولتاژ 12 را به پنچ برای تغذیه مدار محیا میکند

منابع:

سایت مخصوص ساخت اپلیکیشن https// website2app.ir

سایت api هست: thingspeak.com

سایت خرید هاست: parspack.com

مزایای esp8266

<https://febiot.com/everything-about-nodemcu/>

کتاب خانه چیست

http://arduino.ir/Guide/Libraries

**این پروژه کوچک شاید گامی بزرگ برای دو دانشجوی رشته برق باشد**

**اما حتما میتواند سرآغازی برای رشد و شکوفایی عظیم در راه پیشرفت و تغییر مسیر بشریت در راه تغییر و آسان نمودن زندگی شود.**

**همبنگونه که پدران و پدر بزرگ هایمان توانستند بدون تلفن و تلویزیون و موبایل به راحتی زندگی کنند اما نسل حالا بدون این ها نمیتواند زندگی کند**

**این پروژه نیز سر آغازی خواهد بود برای نسل های آینده**

**تا زندگی آنها هر چه بیشتر و بیشتر آسان شود**

علی مهرابی

و

علی نری میسا

پایان