

کلید هوشمند با قابلیت دریافت فرمان صوتی

گزارش فاز اول

گروه 9

اعضای گروه:

● سید علی مرعشیان سرائی - 97102441

● آراین اعتمادی حقیقی - 97110003

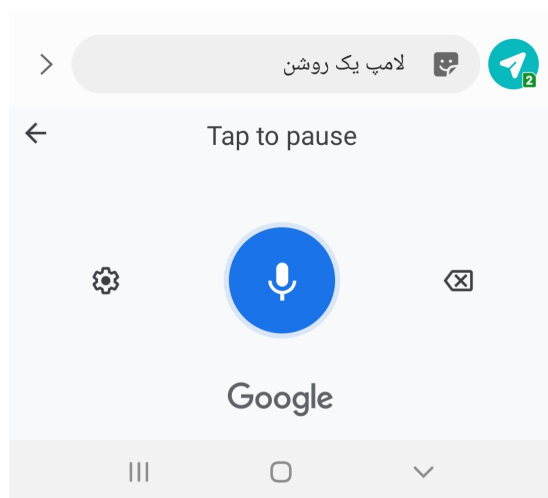
● محمدجواد حمزه - 97101553

بخش اول: فرمان های صوتی

یکی از مهم ترین بخش های پروژه، کنترل لامپ ها و در واقع کنترل قطعه NodeMCU از راه دور است. این کنترل از راه دور می تواند به وسیله اپلیکیشنی بر روی تلفن همراه کاربر، یا از طریق یک صفحه وب انجام شود. تصمیم فعلی گروه استفاده از صفحه وب است که هم بر روی دسکتاپ و هم بر روی تلفن های همراه قابل استفاده است. اما این تصمیم نهایی نیست و در ادامه بسته به ویژگی های هر بستر و در نظر گرفتن شرایط جدید پیش آمده، ممکن است تصمیم دیگری در این مورد گرفته شود.

برای کنترل لامپ ها با استفاده از فرمان های صوتی، نیاز به پیاده سازی یک روش تشخیص صوت (Speech Recognition) است. مروری بر روش های تشخیص صوت برتر که به صورت بی درنگ برای استفاده بر روی پردازنده های گوشی های همراه مناسب هستند، در [این صفحه](#) آمده است. این روش ها که همه از شبکه های عصبی مدرن استفاده می کنند، توسط گوگل در یک صفحه جمع آوری شده اند. در نتیجه، یک راه حل ممکن، استفاده از یکی از این شبکه ها و یادگیری (Train) آن بر روی داده های فارسی است. اما متأسفانه، این کار نیازمند تولید تعداد قابل توجهی داده صوت فارسی برای کلمه های به کار رفته در فرمان های صوتی ما است که کار بسیار دشواری است و در مقیاس یک فاز پروژه ما نمی گنجد. گرچه خوشبختانه، نیازی به اختراع دوباره چرخ توسط ما نیست و خود گوگل سرویس های تشخیص گفتارش را به رایگان در اختیار توسعه دهندگان می گذارد. این سرویس ها هم بهترین کیفیت موجود در بازار را دارند و هم از زبان فارسی پشتیبانی می کنند که برای پروژه ما کلیدی است.

این سرویس ها هم به هنگام استفاده از مرورگر گوگل یعنی گوگل کروم و هم بر روی سیستم عامل های اندروید فعال هستند. در شکل زیر می توانید یک نمونه از اجرای Google Voice Typing را بر روی یک تلفن همراه اندرویدی مشاهده کنید:



همان طور که می بینید، به خوبی یکی از فرمان های صوتی ما ("لامپ یک روشن") را تشخیص داده و به متن فارسی تبدیل کرده است. هنگام جستجو در گوگل کروم نیز امکان استفاده از این سرویس برای کاربران فراهم است.

اما توجه کنید که ما به دنبال استفاده از این ویژگی در برنامه وب (یا موبایل) خود هستیم. به همین هدف، یک پیاده سازی اولیه از نحوه به کارگیری [Web Speech API](https://developers.google.com/web/speech-api/) در برنامه های وب انجام دادیم. با استفاده از این API می توان بر روی مرورگرهای Google Chrome و Samsung Internet از سرویس تشخیص گفتار گوگل به رایگان استفاده کرد. تنها نیاز است که کاربر اجازه دسترسی صفحه به میکروفون خود را بدهد تا صوت او برای چند ثانیه ضبط شده، به سرور های گوگل ارسال شده و متن تشخیص داده شده آن به سرور وب سایت ما برگردد.

برای مشاهده این پیاده سازی اولیه، به آدرس arietemadi.github.io مراجعه کنید. ما یک صفحه وب ساده نوشته ایم که شامل یک فیلد متن و یک دکمه برای ورودی صوت است. برای تست می توانید بر روی دکمه کلیک کرده،

اجازه دسترسی صفحه به میکروفون را داده، و سپس عبارت "لامپ یک روشن" یا هر عبارت فارسی دیگری را به صورت شمرده، رسا و پیوسته در میکروفون خود بیان کنید و نتیجه را مشاهده کنید:



بخش دوم: روشن کردن LED با NodeMCU

در این بخش، مدار ساده ای را در بردبرد طراحی میکنیم و با تکه کدی، آن را کنترل میکنیم. برای این کار، از نرم افزار Arduino IDE استفاده میکنیم (به طور خاص، نسخه 1.8.19). برای اینکه با پورت USB به ماژول NodeMCU متصل شویم، به پورت سریال نیاز داریم. نسخه NodeMCU ما، از سریال چیپ CH340 استفاده میکند. درایور مخصوص به آن را از [اینجا](#) دریافت و نصب کردیم.

در ادامه، در محیط کاری Arduino IDE، به

File->Perferences->Settings

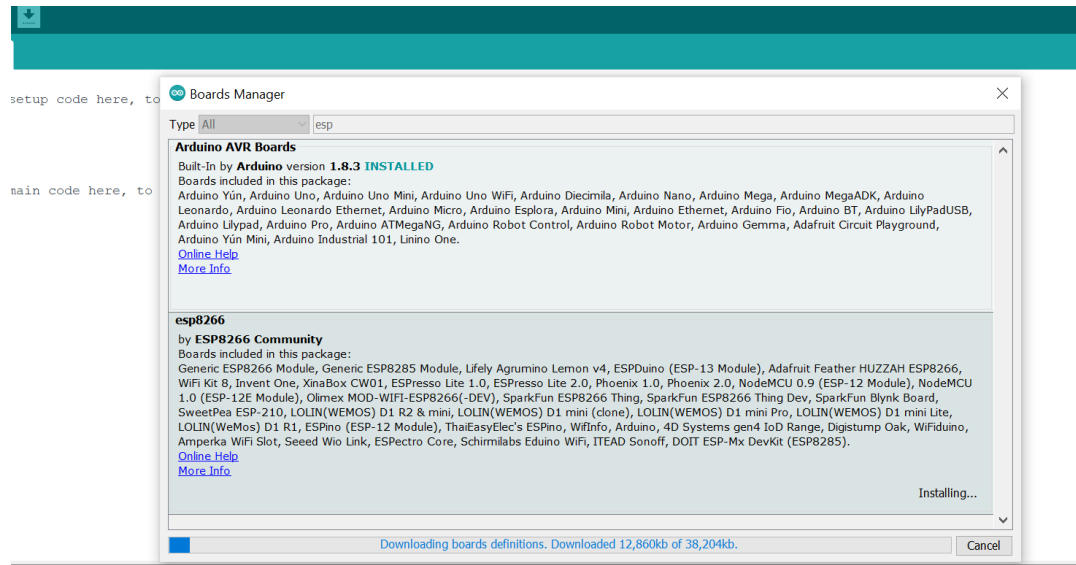
می رویم و در قسمت Additional Board Manager URLs، عبارت زیر را وارد میکنیم:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

حال OK را انتخاب میکنیم. بعد، به

Tools -> Board (Arduino ...) -> Board Manager

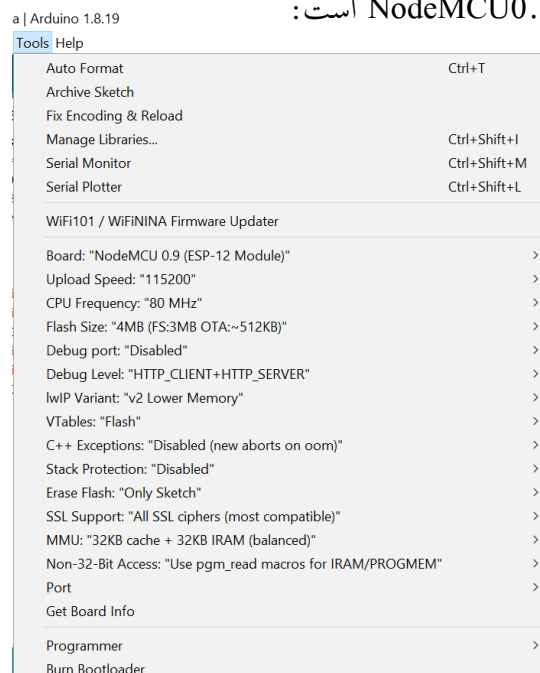
رفته و esp8266 by esp8266 Community را دانلود و نصب میکنیم:



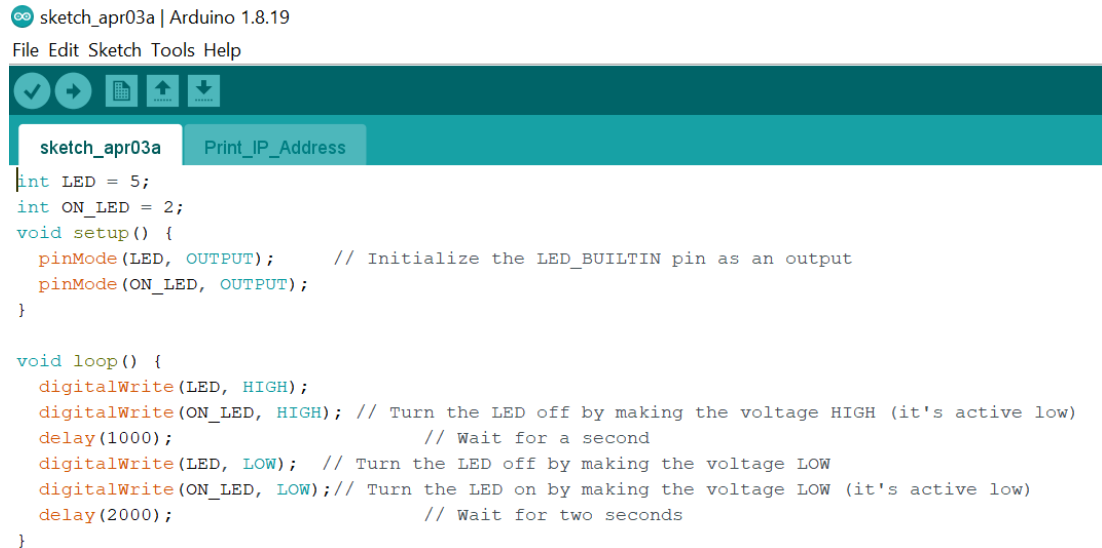
حال، از Device Manager میبینیم که نام پورت سریال چیست و در Tools آن را انتخاب میکنیم (و سیم USB را

که طرف دیگرش به NodeMCU متصل است، به آن پورت وصل میکنیم). همین از برد ها، نسخه NodeMCU

خود را انتخاب میکنیم. برای ما NodeMCU0.9 است:



حال، تکه کد زیر را وارد میکنیم:



```
sketch_apr03a | Arduino 1.8.19
File Edit Sketch Tools Help

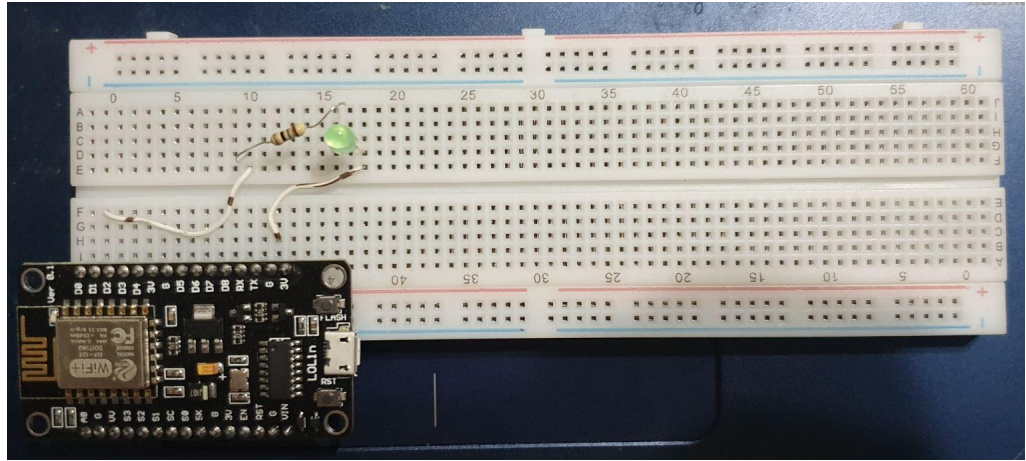
sketch_apr03a Print_IP_Address

int LED = 5;
int ON_LED = 2;
void setup() {
  pinMode(LED, OUTPUT);    // Initialize the LED_BUILTIN pin as an output
  pinMode(ON_LED, OUTPUT);
}

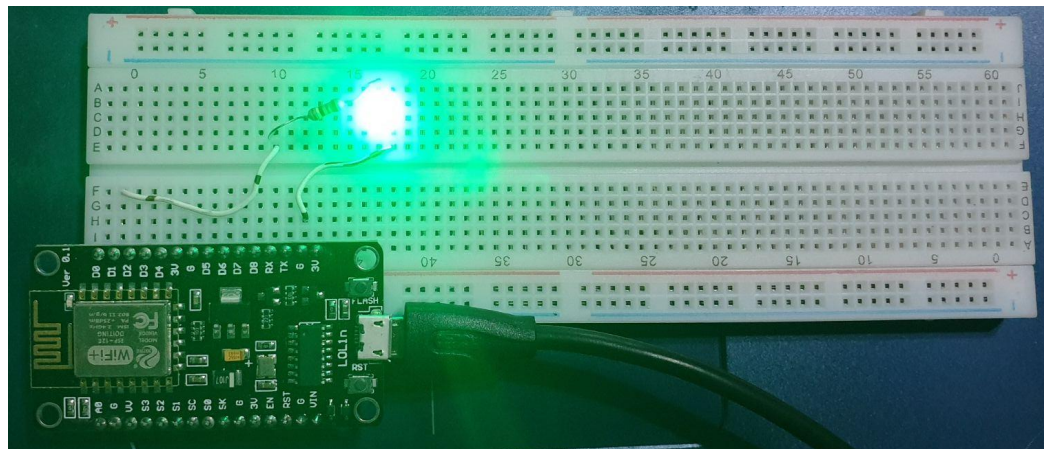
void loop() {
  digitalWrite(LED, HIGH);
  digitalWrite(ON_LED, HIGH); // Turn the LED off by making the voltage HIGH (it's active low)
  delay(1000);                // Wait for a second
  digitalWrite(LED, LOW);    // Turn the LED on by making the voltage LOW
  digitalWrite(ON_LED, LOW); // Turn the LED on by making the voltage LOW (it's active low)
  delay(2000);               // Wait for two seconds
}
```

در این تکه کد، به پین شماره 5 (که در NodeMCU پین D1 نامیده می شود و active high است) ولتاژ "بالا" میدهیم و LED روی بردبرد را روشن میکنیم. همزمان به پین شماره 2 (که به LED روی خود مازول متصل است و active low است) نیز ولتاژ بالا میدهیم و با این کار این LED را خاموش میکنیم. بعد از یک ثانیه توقف، حالا برعکس LED روی مازول را روشن میکنیم و دیگری را خاموش میکنیم.

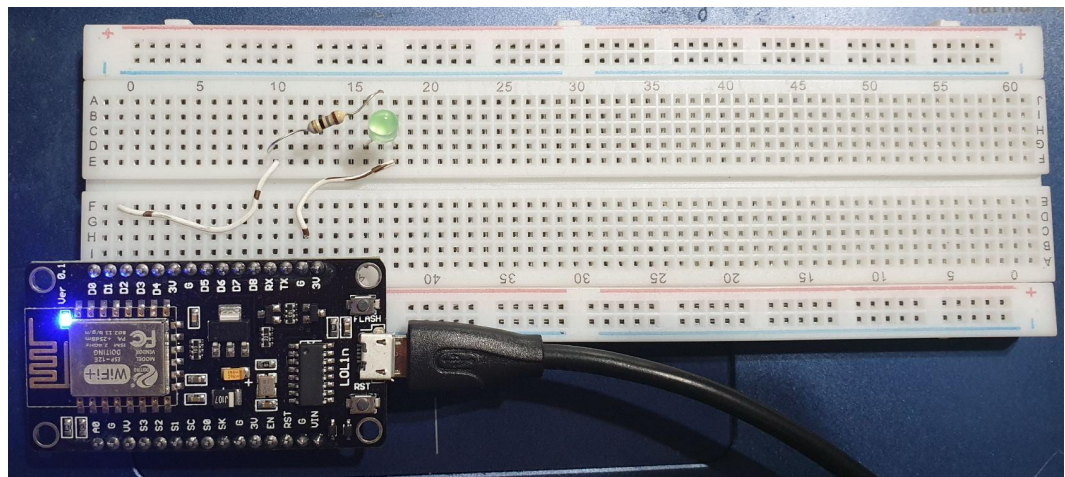
پیاده سازی مدار در شکل زیر آمده است:



وقتی که لامپ روی بردبرد روشن شود:



در نهایت، وقتی که لامپ روی خود NodeMCU روشن شود:



وظایف در نظر گرفته شده برای فاز بعد:

- تحقیق و پیاده سازی اولیه روش های مختلف اتصال قطعه NodeMCU به وای فای و کنترل آن از طریق وب سرور یا گوشی موبایل
- تحقیق در مورد استفاده از voice typing اندروید در اپلیکیشن های اندروید (توجه کنید که پیاده سازی متناظر بر روی وب را داریم. اما استفاده از بستر اندروید به جای صفحه های وب می تواند مزیت های دیگری داشته باشد و تحقیق ما به همین منظور است.)
- توسعه صفحه وب. پس از پیاده سازی اولیه تشخیص گفتار، نیاز است که متن های خروجی را تحلیل کنیم. همچنین کاربر باید بتواند برای خاموش روشن شدن چراغ ها در آینده برنامه دلخواه بچیند.