نظرة على الهاردوير للNodeMcu

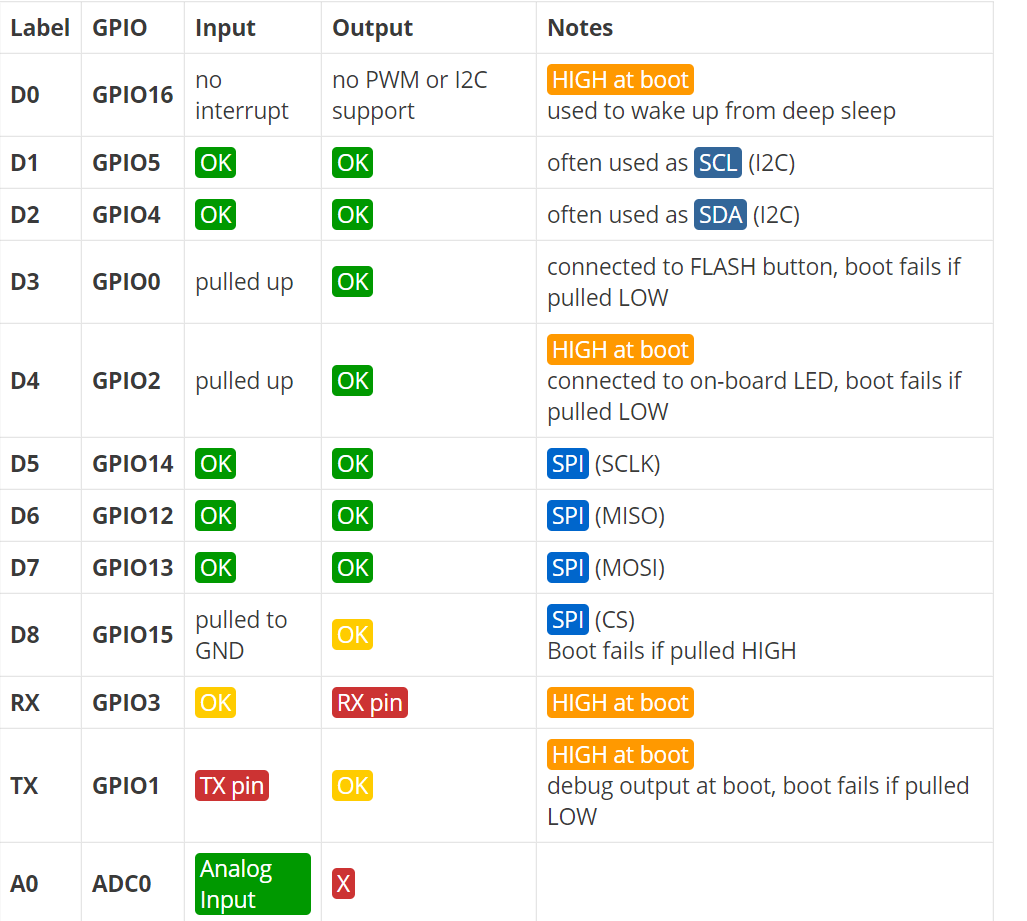
ان هذه البورد مبنية على المتحكم esp8266 من شركة espressif

في البداية كانت الطريقة الوحيدة لبرمجته هي استخدام البيئة espressif IDE لكن مع مرور الوقت اصبح esp8266 يدعم arduino IDE و اصبحت معظم المكتبات متوافقة معه.

المواصفات الاساسية لل esp8266

* جهد التشغيل(Operating Voltage) : 3.3 فولت (لكن جميع الارجل تستطيع تحمل 5 فولت)
* جهد الدخل (Input Voltage) : 12 – 7 فولت
* عدد المداخل و المخارج (GPIO) pin :16
* عدد الارجل التماثلية (Analog): 1
* UART : 1
* I2C: 1
* I2S : 1
* ذاكرة الفلاش : 4 MB
* SRAM :64 KB
* Clock Speed : 80 Mhz

جدول وظائف الارجل (GPIO) :



OK : يمكنك استخدامه

OK : يمكنك استخدامه لكن قد يسبب بعض المشاكل

OK : لا يمكنك استخدامه (غير موصى به)

مصدر الصورة من randomnerdtutorial.com

ترقيم (GPIO) في :arduino IDE

اذا اردت برمجة اي gpio على بورد NodeMcu ستجد اختلاف في طريقة الترقيم فمثلا ستجد على البورد نفسها الترقيم التالي D1,D2:

لكن هذا الترقيم لا يعمل في بيئة الاردوينو اذ ان D0 على سبيل المثال يقابله 16 و هكذا..

و لذلك فان في الجدول السابق اسم ال pin و يقابله الرقم الذي ستسخدمه للبرمجة.

الليد الموجود على البورد موصول ب D4 (يقابله 2)



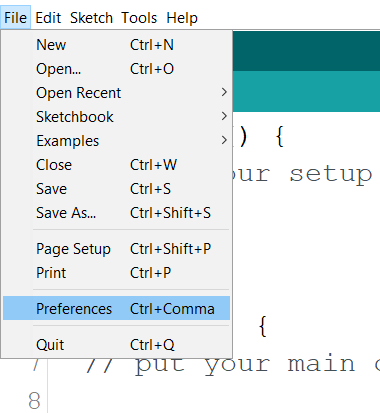
***نظرة على الهاردوير لل ESP32***

الESP32 هو متحكم (microcontroller) قوي جدا من تصنيع شركة espressif و يمكنك اعتباره الاخ الاكبر لل esp8266 (الذي بنيت على اساسه nodemcu) , يتميز esp32 بقوة معالجه و توافر العديد من الميزات مع دعم بيئة الاردوينو مما يجعله برأيي الشخصي افضل من esp8266 و خاصة ان فرق السعر بينهما ليس كبيرا .

المواصفات الاساسية لل ESP32

***تهيئة المتحكم على بيئة الاردينو***

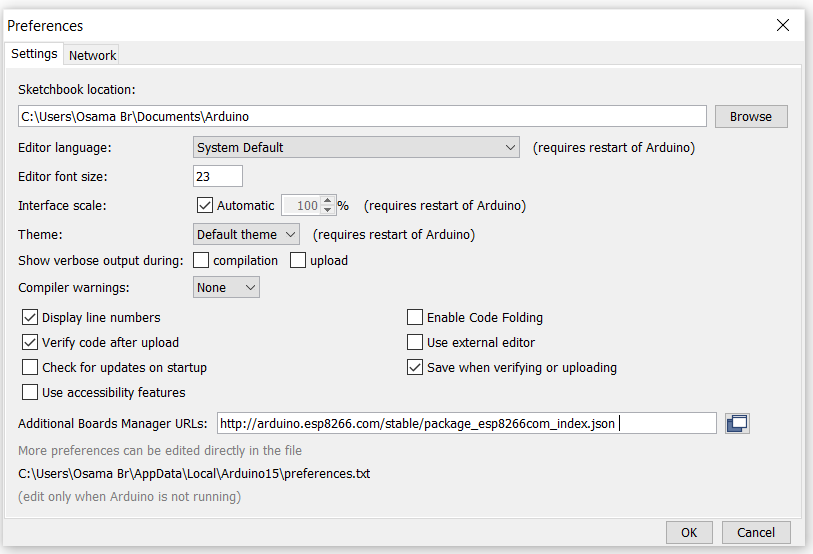
لكي تستطيع ان تبرمج esp8266 على بيئة الاردوينو تحتاج لان تقوم بتنزيل بعض المكتبات لان هذا المتحكم ليس مدعوما من البيئة افتراضيا (لسوء الحظ).

***خطوات تنصيب esp8266 على بيئة الاردوينو***

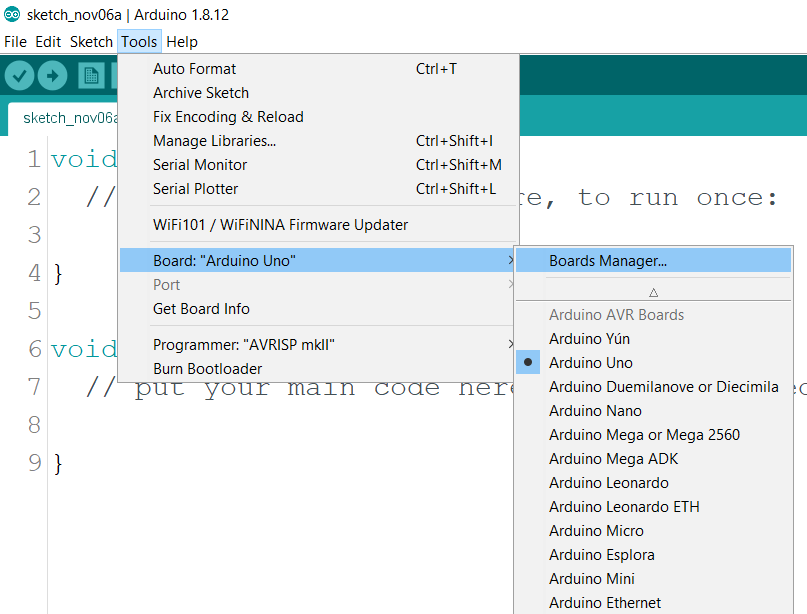
* اولا: من file ندخل الى preferences
* ثانيا : من preferences
* ننسخ هذا الامر و نضعه في additional boards manager URL

**http://arduino.esp8266.com/stable/package\_esp8266com\_index.json**

* ثم نضغط ok



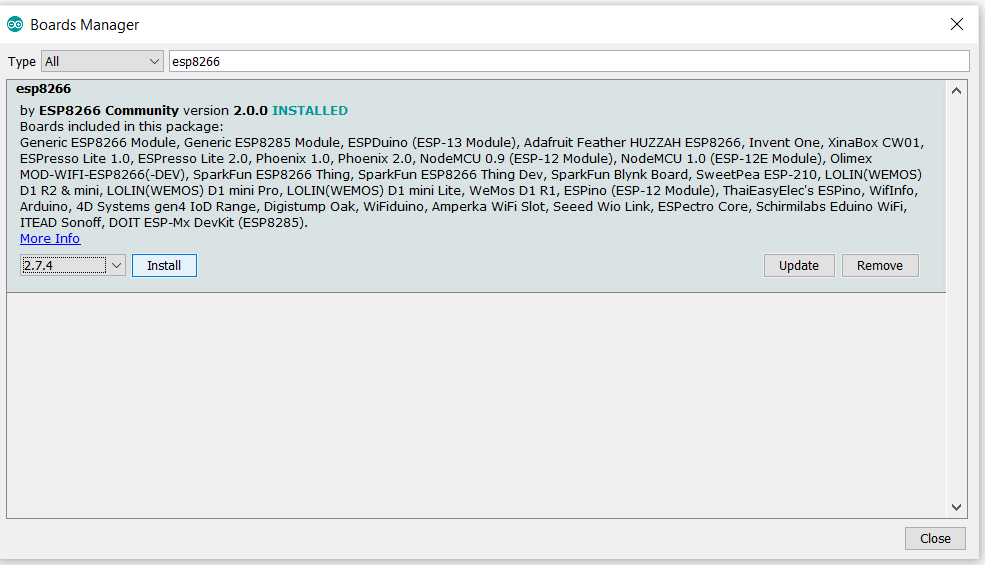
* ثالثا :

ادخل tools > boards > boards manager

و من boards manager

اكتب في مربع البحث” “esp8266 و ستظهر لك مكتبة خاصة بهذه المتحكمات

اختر اخر نسخة و اضغط install و سيقوم البرنامج بتحميل المكتبة من الانترنت (و هذا هو الجزء الاصعب حرفيا و خاصة في الدول العربية!!)



* رابعا : بعد ان انتهيت من التحميل (الف مبروك) قم باعادة تشغيل البرنامج

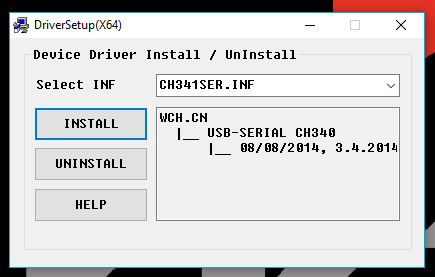
ادخل tools > boards و ستظهر لك مجموعة من بوردات esp8266

اذا كنت تملك nodemcu فاختر "NodeMcu 1.0 ESP 12E"

و الان قم بوصل NodeMcu خاصتك و اختر المنفذ المناسب و انتقل الى القسم الاخر من الكتاب لكي تبدا بالبرمجة و صنع اجهزة انترنت الاشياء.

ملاحظة : اذا لم يظهر اي منفذ في قائمة port فقم بتنصيب تعريف ch340 من الرابط : <https://sparks.gogo.co.nz/ch340.html>

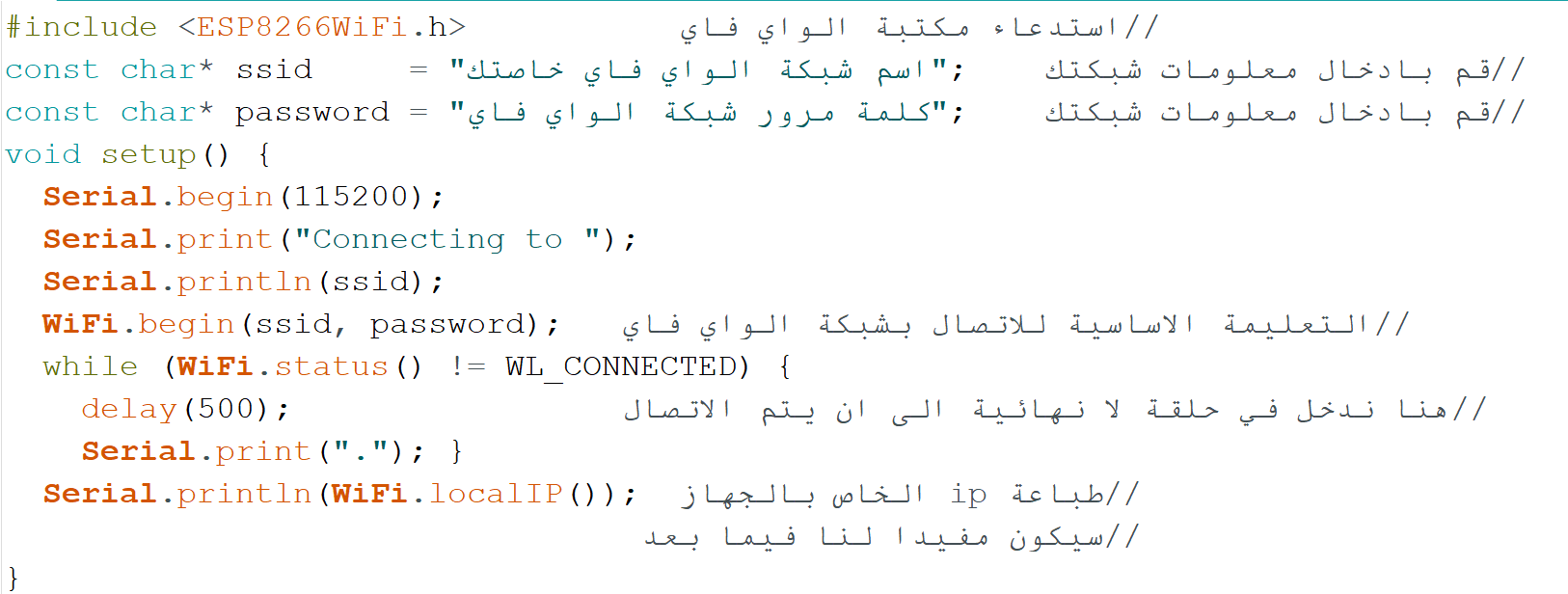
ثم قم بفتح البرنامج مع وصل المتحكم بالحاسوب و اضغط install



***توصيل المتحكم بالانترنت***

سنقوم الان بوصل Esp8266 بالواي فاي ليستطيع الوصول للانترنت و سنقوم الان بكتابة كود سنضعه في كل برنامج من الان فصاعدا و تستطيع نسخه و لصقه في اي برنامج تريد اي انه ما عليك الا ان تقوم بتغيير ssid و password و استمتع بالانترنت

الكود:



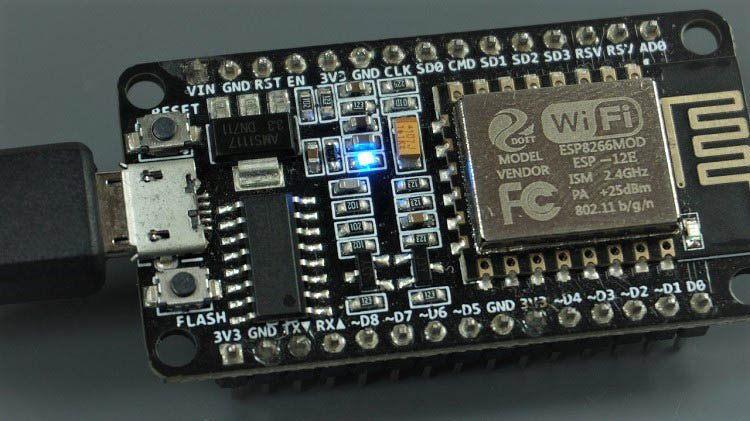
و هكذا نكون قد انتهينا من اول مرحلة في اي برنامج لجهاز انترنت الاشياء

و بعدها سنبدا بصناعة بعض الاجهزة الجميلة

***التعامل مع النصوص***

انا اعرف ان هذا قد يبدو غريبا بعض الشيء الا اننا سنضطر للتعامل مع النصوص من صفحات الويب المتعددة و هذا ما يستوجب كتابة هذا القسم.

في هذا القسم لن نتطرا لاي شيء عن انترنت الاشياء اي ان كل الاكواد الموجودة في هذا القسم تستطيع تجربتها **على اي بورد اردوينو و يمكنك تجربتها ايضا على nodemcu .**



***التعامل مع المحارف (char)***

هناك مجموعة من التعليمات الموجود في لغة البرمجة c-arduino و التي تساعدنا على معرفة جميع المعلومات عن اي محرف مثل (هل هو capital , رقم , اشارة ترقيم , فراغ , الخ....)

هذه التعليمات تعطينا قيمة اما True او False و لاحظ انها تفحص محرف واحد و ليس سلسلة نصية (اي انها لا تقوم بتحليل اكثر من محرف في ان واحد)

هذه الدالات تحتاج الى بارامتر واحد و هو المحرف بحد ذاته (اي انك ستضعه بين القوسين)

* isAlphaNumeric();

معرفة ما اذا كان النص يحوي ارقام و نصوص يعطي قيمة true اذا احتوى على رقم او حرف

* isAlpha();

معرفة ما اذا كان المحرف هو حرف (a-z , A-Z)

* isWhitespace();

معرفة ما اذا كان المحرف مكون من مسافة(ضغطة على زر المسافة)

* isDigit();

معرفة ما اذا كان المحرف رقما

* isLowerCase();

معرفة ما اذا كان الحرف صغيرا (a-z)

* isUpperCase();

معرفة ما اذا كان الحرف كبيرا (A-Z)

* isPunct();

معرفة ما اذا كان المحرف هو احد اشارات ترقيم(اشارة تعجب , استفهام , فاصلة منقوطة , نقطة , الخ...)

* isPrintable();

معرفة ما اذا كان المحرف قابل للطباعة

***جمع النصوص***

لنقل ان لديك متغيرين يحويان نص و تريد ان تطبعهما معا او تضعهما في متغير اخر

الطريقة الاولى:

باستخدام علامة الجمع + مثال :

String x = string1 + string2;

String x = string1 + “text”;

و يمكنك استعمال =+ ايضا لاضافة النص الجديد الى القديم

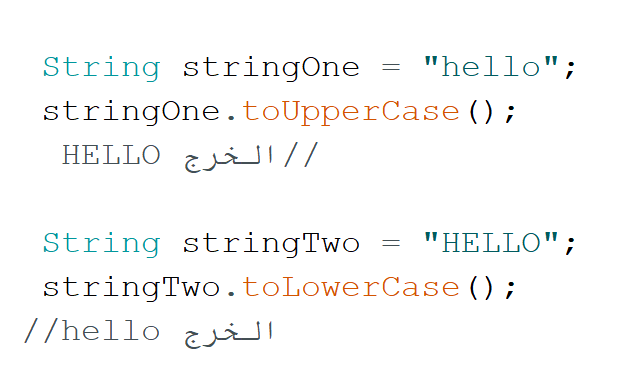
**ملاحظة :** تذكر اخي العزيز انه لا يمكنك جمع النصوص او طرحها او ايجاد الجذر التربيعي لها يا رعاك الله.

***تغيير حالة الاحرف في نص***

نستطيع تغيير حالة الاحرف في نص من كبير الى صغير و بالعكس باستخدام دالتين

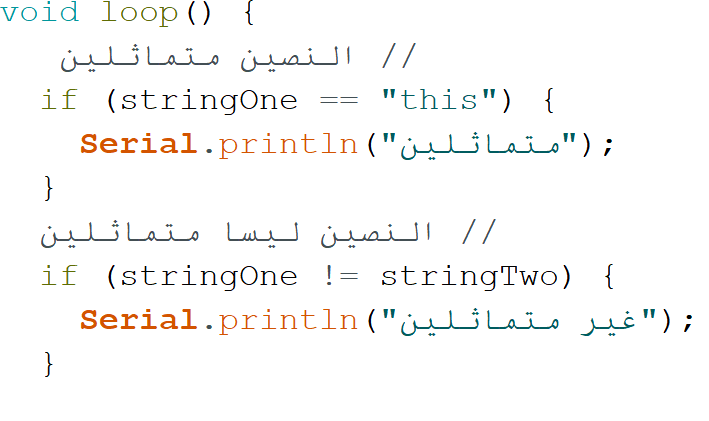
toUpperCase() و toLowerCase()

مثال:



***مقارنة السلاسل***

اذا كان لديك عدة سلاسل فانت تستطيع ان تقوم بالمقارنة بينها لتنفيذ شرط او حلقة تكرار.



و **لا تستطيع** استخدام نوع اخر من المقارنة غير == او =!

***معرفة مكان محرف او نص***

لنقل انه لديك سلسلة نصية و تريد معرفة مكان محرف او كلمة فيها لكي تقوم بحذفه او اضافة كلمة بعده او استخراج معلومات الخ...

للقيام بذلك فنحن نستخدم دالة indexOf();

في النصوص نقسم النص الى مربعات كل مربع فيه محرف نبدا العد من الصفر مثال: "hi Osama!"

المحرف h يوجد في المربع 0

المحرف i يوجد في المربع 1

المحرف " " (اي المسافة) يوجد في المربع 2

و نكمل الى نهاية النص

كيفية استخدام دالة indexOf()

**النص\_الذي تبحث\_فيه.indexOf(المحرف الذي تريد ايجاده) ;**

و تعطيك هذه الدالة رقم المربع الذي توجد فيه المحرف او الكلمة المطلوبة

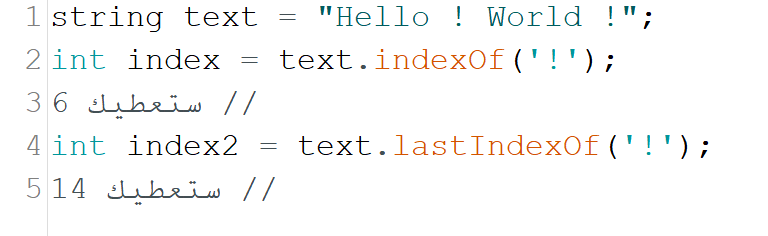
(طبعا اذا كنت تبحث عن كلمة فسوف تعطيك الدالة رقم المربع الذي فيه اول حرف من الكلمة)

ملاحظة : فلنقل ان لديك نصا "hello ello" و تريد البحث فيه عن الكلمة “ello” فسوف تدلك الدالة الى اول كلمة تجدها (الموجودة في كلمة hello)

و لذلك يوجد دالة شبيهة بهذه الدالة اسمها lastIndexOf()

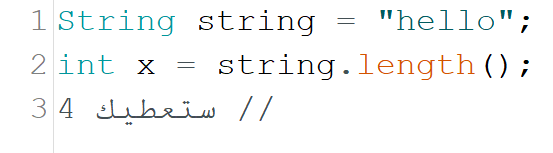
فاذا استخدمناها في المثال اعلاه فانها ستدلك الى "ello" الثانية و **ليس الاولى**

مثال شامل :



***معرفة حجم سلسلة نصية***

اذا اردت معرفة حجم سلسلة نصية فيمكنك استخدام length()

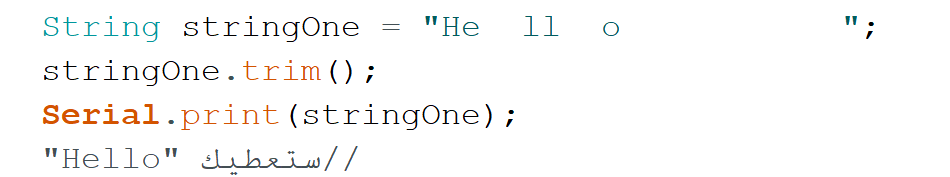


طبعا ستعطيك هذه الدالة عدد المربعات الموجودة في النص و ليس حجمه بالبايتات

***التخلص من الفراغات البيضاء***

لنقل ان لديك السلسة النصية "he ll o" و تريد ان تتخلص من الفراغات البيضاء فيمكنك استخدام الدالة trim() للقضاء على المسافات نهائيا (**حتى المسافات بين كل كلمة و اخرى ستختفي مما يجعلها غير مقروئة**)

استخدامها:

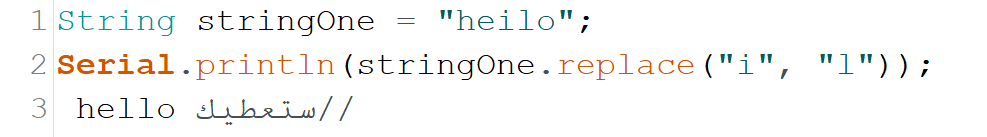


***تبديل محارف في سلسة نصية***

تخيل لو ان لديك سلسلة نصية و تحتاج الى تغيير محرف او اكثر فيها فانك تستخدم الدالة replace()

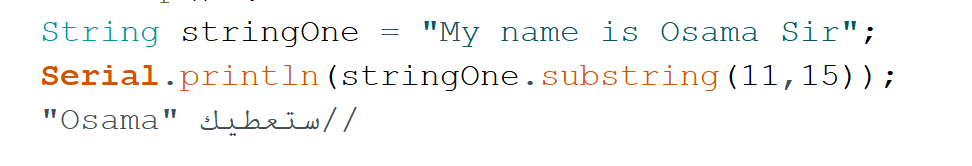
اذ تقوم باعطائها المحرف الاصلي و المحرف الذي تريد وضعه مكان المحرف القديم

مثال



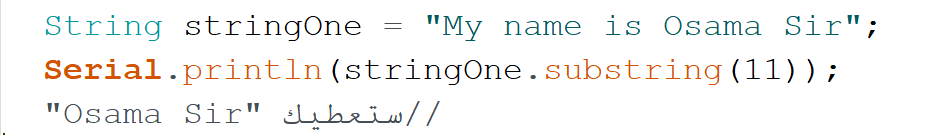
***اقتطاع جزء معين من سلسلة***

تخيل ان لديك السلسلة النصية "My name is Osama sir" و انت تريد فقط كلمة Osama و لا تريد السلسة كاملة ففي هذه نستخدم الدالة **;(نهاية النص المراد اقتطاعه,بداية النص المراد اقتطاعه).substringالنص\_الاصلي**



و يمكنك اذا اردت تحديد بداية الاقتطاع و **عدم تحديد النهاية** اذ سيقوم المترجم باعتبار نهاية النص هي نهاية النص المراد اقتطاعه

;(بداية النص المراد اقتطاعه).substringالنص\_الاصلي

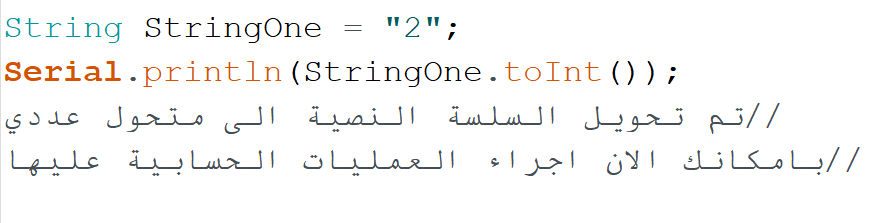


***تحويل سلسلة نصية الى عدد (int)***

لنفرض ان لديك السلسلة “abc1234abc” و تريد ان تقوم بعمليات حسابية على "1234" فستقوم بالتالي:

* ستقوم باقتطاع "1234" من خلال الدالة substring
* ستقوم باستخدام الدالة toInt لتحويل السلسلة المقتطعة الى متغير عددي يمكنك اجراء عمليات حسابية عليه

**.toInt(); النص\_المراد\_تحويله**



و هكذا نكون قد انتهينا من فصل معالجة النصوص لننتقل الى الفصل التالي.

***رفع البيانات الى الانترنت***

و الان سنبدأ بأول مشروع يستخدم الواي-فاي

سنقوم برفع بيانات الحساسات الى موقع thinkspeak لتسجيل تلك البيانات و حفظها من الضياع.

**ما هو موقع ThinkSpeak ؟**

هو موقع يسمح لك برفع البيانات عليه و من ثم عرضها في جداول (يوجد العديد من الانواع) و الوصول الى تلك المعلومات من اي مكان على وجه الارض في اي وقت

هناك العديد من الطرق لرفع البيانات الى موقع thingspeak و لكن اليوم سنستخدم اسهل طريقة لذلك باستخدام مكتبة ThingSpeak-Arduino

**تنصيب مكتبة ThingSpeak**

يمكنك تنصيبها من الرابط:

https://github.com/mathworks/thingspeak-arduino

او من خلال البحث "ThinkSpeak" في library manager المدمج في بيئة الاردوينو

**التوصيلات:**

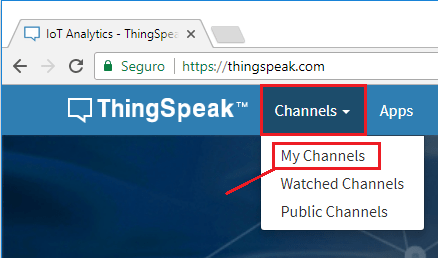
لا يوجد , التركيز في هذا الفصل على السوفتوير استمتع باضافة الحساسات التي تريدها مع اقل تعديلات على الكود.

**اعداد حساب على موقع ThingSpeak**

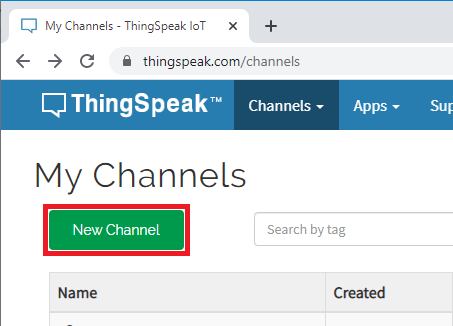
اذهب الى موقع thinkspeak <https://thingspeak.com>

و قم بانشاء حساب عليه بالضغط على زر "get started for free"

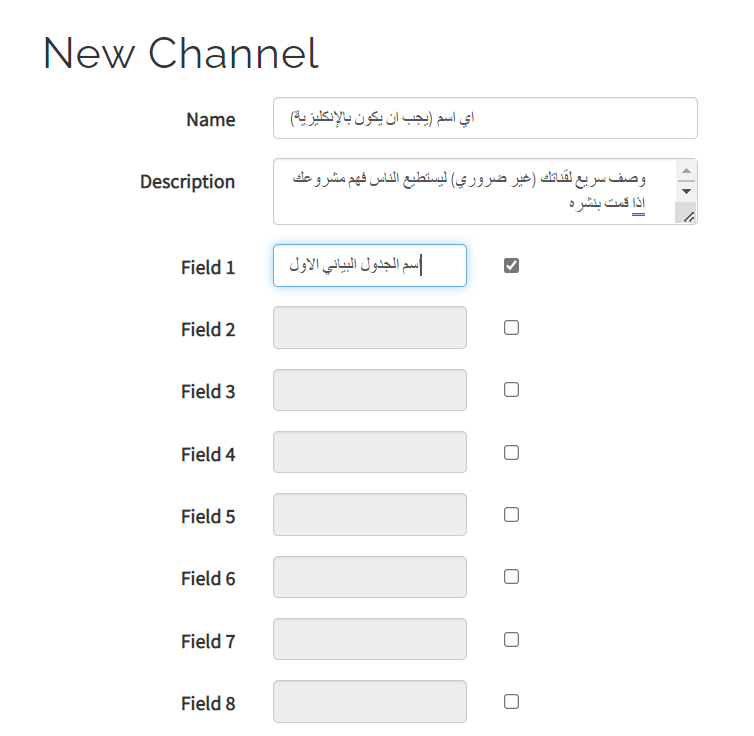
بعدما قمت بتسجيل الدخول قم بالضغط على ايقونة "Channels" و اختار "My Channel"



اضغط على زر "New Channel" لانشاء قناة جديدة لرفع البيانات عليها



ثم قم بملئ البيانات المهمة (name,field) اما باقي الخانات فلن يؤثر عليك اذا تركتها فارغة

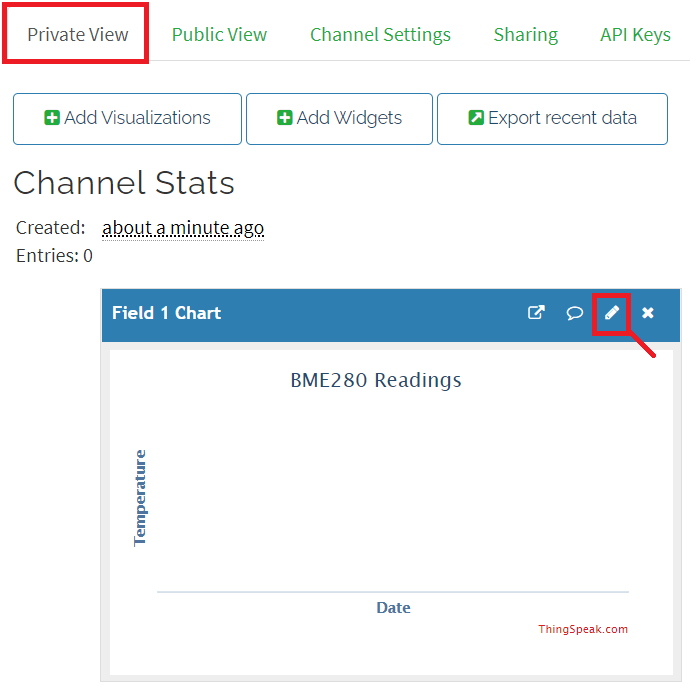


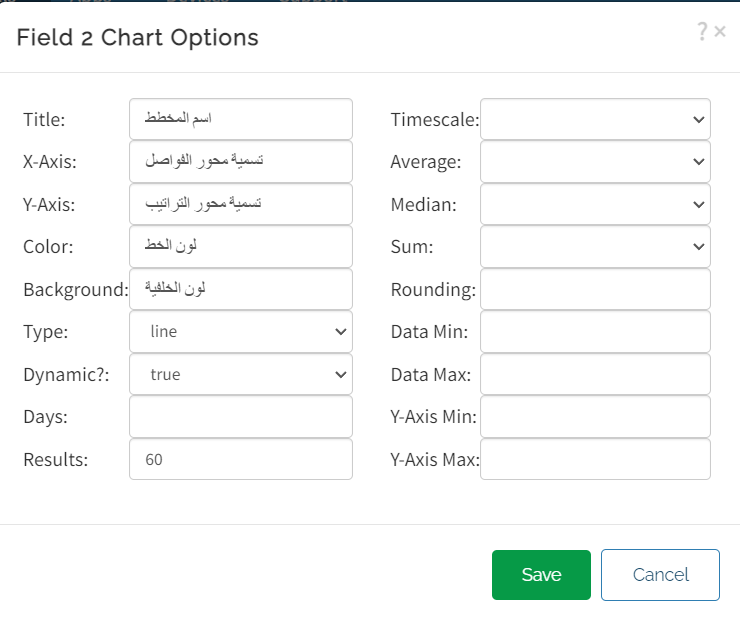


و بعدها اضغط على زر "save channel" لحفظ القناة الجديدة

***التعديل على المخطط البياني***

يمكنك التعديل على شكل المخطط البياني من خلال الذهاب الى private view و من ثم الضغط على ايقونة القلم الصغيرة المشار عليها في الصورة





بعد الضغط عليها ستظهر لديك قائمة بالاشياء التي يمكنك التعديل عليها و من اهمها:

* Title : عنوان المخطط البياني او بمعنى اخر اسم المخطط البياني
* X-Axis : يمكنك كتابة فيها ما يعبره محور x على سبيل المثال لنفترض ان لديك جدول يعبر عن تغير درجة الحرارة بالنسبة للزمن فيمكنك تسمية محور x ب "الزمن"
* Y-Axis : نفس الشيء تماما , يمكنك كتابة

ما يعبر عنه محور Y

* Color : تغيير لون الخط البياني
* Background : تغيير لون خلفية المخطط

اما باقي الحقول فهي لا تهمنا كثيرا في الوقت الحالي و تستطيع ايجاد معلومات اوضح على الانترنت و على كل الاحوال ليس من الضروري ان تقوم باي تعديلات على المخطط انما احببت ان اذكر ذلك في حال احببت ان تخرج الفنان الذي بداخلك 3>

***مفتاح API :***

لارسال المعلومات من جهاز esp32/8266 علينا اولا ان نحصل على مفتاح API

***ما هو API و لماذا هو موجود اصلا؟ّ!***

APIاو Application Programming Interface وباللغة العربية واجهة برمجة التطبيقات

API ببساطة هى نقطة الوصل بين بيئة تطويرية معينة والمطورين , حتى تتيح لهؤلاء المطورين الاستفادة من خدمات هذه البيئة بدون الحاجة إلى بناء كل شئ من الصفر او بمعنى اخر طريقة للتواصل مع البرامج او مواقع الويب سهلة و سريعة.

يوجد الكثير من الامثلة API’s مشهورة نصادفها كل يوم مثلا في الفيسبوك, يتيح API للمواقع لتسجيل الدخول عن طريقه بدون الحاجة إلى بناء نظام تسجيل دخول من الصفر

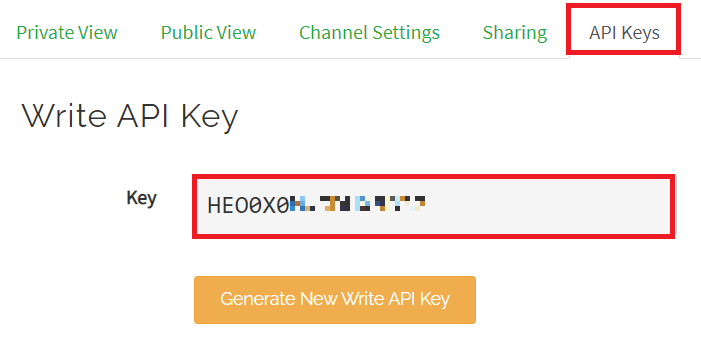
مثال اخر عندم تقوم بشراء شيء على الانترنت من موقع معين فهذا الموقع يستخدم API الخاصة بشركة VISA ليقوم بالتواصل مع موقع visa وخصم المال من بطاقتك

و بما ان عملية التواصل من خلال API مهمة فلا بد من وجود طريقة لمنع الاخرين من الوصول لمعلوماتك (تخيل لو ان اي شخص يستطيع الوصول ل API فيزا و طلب بعض الاموال من حسابك الشبه فارغ للاسف), ولذلك يوجد شيء يسمى ب API Key و هو عبارة عن مفتاح الكتروني (سلسلة نصية) يمكنك اعتبارها كباسوورد و لكل حساب مفتاح api خاص به ]اقرا المقالة [ماهو API وما فائدته ؟ (motwr.com)](https://motwr.com/%D9%85%D8%A7%D9%87%D9%88-api-%D9%88%D9%85%D8%A7-%D9%81%D8%A7%D8%A6%D8%AF%D8%AA%D9%87-%D8%9F/) [

و في موقع thingSpeak يوجد api key لكي لا يقوم اي شخص برفع بيانات او سرقة البيانات التي رفعتها

***كيفية الحصول على مفتاح API الخاص بك***

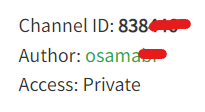
اضغط على زر "API Keys" ثم قم بنسخ **Write API Key**



و هكذا تكون قد انشأت حساب thingSpeak بنجاح و ما علينا الا ان نكتب كود لكي نرفع البيانات الى هذا الموقع

الحصول على معرف القناة:

بالاضافة الى API Key يوجد لكل قناة معرف خاص بها channel ID

و تستطيع نسخ المعرف من الزاوية اليسرى العليا من صفحة القناة الخاصة بك

و لمعرفة رقم المخطط البياني عليك الذهاب الى channel settings و رؤية رقم المخطط الذي تريد الرفع عليه (مثال field 1 هو المخطط رقم 1)

***الكود:***

#include <WiFi.h> // استدعاء المكتبات اللازمة

#include "ThingSpeak.h"

const char\* ssid = "REPLACE\_WITH\_YOUR\_SSID"; // ضع اسم شبكة الواي-فاي

const char\* password = "REPLACE\_WITH\_YOUR\_PASSWORD"; // كلمة مرور الشبكة

WiFiClient client;

unsigned long myChannelNumber = X; // برقم Xبدل

int myFieldNumber = X; // برقم Xبدل

const char \* myWriteAPIKey = "XXXXXXXXXXXXXXXX";

// متغيرات الوقت

unsigned long lastTime = 0;

unsigned long timerDelay = 30000;

String data = "hi";

void setup() {

Serial.begin(115200);

if(WiFi.status() != WL\_CONNECTED){ // الاتصال بشبكة الواي-فاي

Serial.print("Attempting to connect");

while(WiFi.status() != WL\_CONNECTED){

WiFi.begin(ssid, password);

}

Serial.println("Connected.");

}

ThingSpeak.begin(client); //thingSpeak تهيئة للاتصال بموقع

}

void loop() {

if ((millis() - lastTime) > timerDelay) { // رفع البيانات كل 30 ثانية

int x = ThingSpeak.writeField(myChannelNumber,myFieldNumber, data, myWriteAPIKey);

// تعليمة رقع البيانات

if(x == 200){

Serial.println("Channel update successful.");

}

else{

Serial.println("Problem updating channel. HTTP error code " + String(x));

}

lastTime = millis(); // الحصول على وقت اخر وقت محاولة للرفع

}

}

***شرح الكود:***

unsigned long myChannelNumber = X;

معرف القناة التي تريد الرفع عليها ضع معرف القناة مكان X

int myFieldNumber = X;

رقم المخطط الذي تريد القيام بعملية الرفع عليه

const char \* myWriteAPIKey = "XXXXXXXXXXXXXXXX";

ضع Write API Key الذي نسخته من الموقع

if ((millis() - lastTime) > timerDelay){

و بما اننا نستخدم النسخة المجانية للموقع فاننا نستطيع رفع قيمة واحدة كل 30 ثانية و الا لن يتم رفع البيانات بنجاح , و لذلك فان علينا جعل البرنامج لا يرفع قيم الا بعد مرور 30 ثانية , قد يفكر البعض باستخدام تعليمة delay() لكن لهذه التعليمة اثار سلبية على البرنامج لانها تقوم بايقاف كل العمليات و الانتظار , و لذلك فاننا سنستخدم طريقة للتخلص من delay() بمساعدة تعليمة millis()

ما هي تعليمة millis()

هي تعليمة تعطيك الوقت المنقضي منذ تشغيل البرنامج بالميلي ثانية ms

و لهذا فاننا نحتاج الى متغيرين timerDelay و lastTime

اما timerDelay فسنخزن فيه المدة الفاصلة بين كل عملية رفع و هنا عرفناها بانها 30000 ميلي ثانية او 30 ثانية

اما المتغير الاخر lastTime فسنخزن فيه زمن اخر عملية رفع (بداية الزمن هي لحظة تشغيل البرنامج)

تعليمة if هذه تختبر اذا كان (current time – last time) اكبر من 30000 و اذا تحقق هذا الشرط ستنتقل الى تعليمات رفع البيانات

int x = ThingSpeak.writeField(myChannelNumber, 1, data, myWriteAPIKey);

و هذه هي تعليمة الرفع التي سنستخدمها , ان تعليمة writeField تاخذ اربع مدخلات

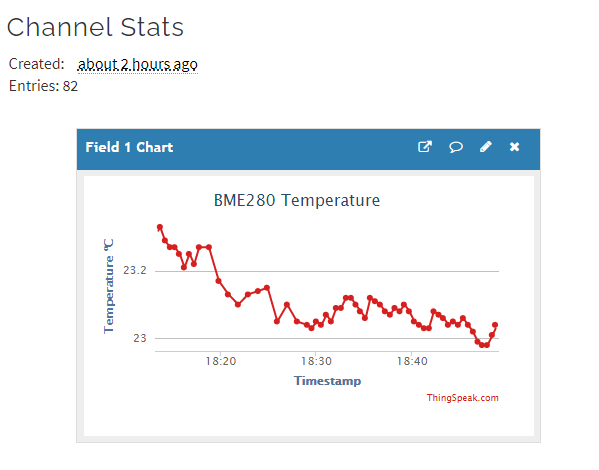
* myChannelNumber : رقم القناة التي تريد الرفع عليها
* FieldNumber : رقم المخطط الذي تريد الرفع عليه (يمكنك انشاء 8 مخططات لكل قناة) و قد وضعت 1 مكانه
* Data : البيانات التي تريد رفعها
* myWriteAPIKey : رقم مفتاح API الخاص بك

و قد تتسائل عزيزي القارئ لماذا و ضعت متغير x و السبب بكل بساطة هو ان سيرفر موقع thingSpeak يعطيك رقم استجابة لكي تعلم ما اذا تم الرفع بنجاح او لا

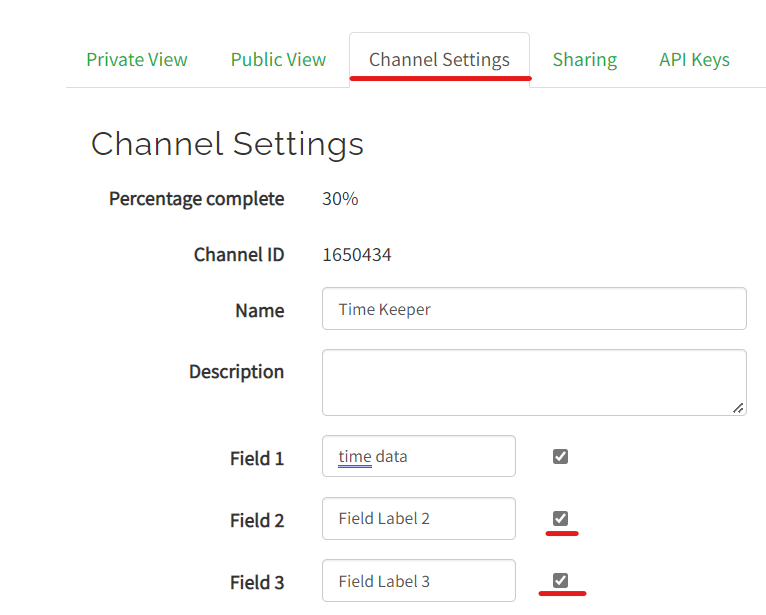
(html response) و اذا كانت قيمة x مساوية لـ 200 هذا يعني انه قد تم الرفع بنجاح والا فلا

lastTime = millis();

تعيين قيمة المتغير lastTime و هو وقت اخر محاولة عملية رفع (نعيد و نكرر بداية التوقيت لحظة تشغيل البرنامج و وحدة الزمن ms)



و اخيرا اذا فتحت موقع thingspeak و دخلت الى channels من private view سترى بياناتك موجودة على المخطط البياني كالشكل الاتي

الرفع الى اكثر من مخطط بياني:

اولا للرفع الى اكثر من مخطط بياني يجب عليك انشاء اكثر من مخطط بياني

و للقيام بذلك ادخل الى

Channel Settingsو بعدها قم بوضع علامة (صح) على كل مخطط تريد انشائه

و لا تنسى ان تحفظ التغييرات

الكود:

/\*

Adapted from Example from ThingSpeak Library (Mathworks)

Rui Santos

Complete project details at https://RandomNerdTutorials.com/esp32- thingspeak-publish-arduino/

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files.

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

\*/

#include <WiFi.h>

#include "ThingSpeak.h"

const char\* ssid = "REPLACE\_WITH\_YOUR\_SSID"; // اسم شبكة الواي فاي

const char\* password = "REPLACE\_WITH\_YOUR\_PASSWORD"; // كلمة مرور الواي فاي

WiFiClient client;

unsigned long myChannelNumber = X;

const char \* myWriteAPIKey = "XXXXXXXXXXXXXXXX";

unsigned long lastTime = 0;

unsigned long timerDelay = 30000;

float dataA;

float dataB;

float dataC;

void setup() {

Serial.begin(115200);

if(WiFi.status() != WL\_CONNECTED){ // الاتصال بشبكة الواي-فاي

Serial.print("Attempting to connect");

while(WiFi.status() != WL\_CONNECTED){

WiFi.begin(ssid, password);

}

Serial.println("Connected.");

}

ThingSpeak.begin(client); // Initialize ThingSpeak

}

void loop() {

if ((millis() - lastTime) > timerDelay) {

ThingSpeak.setField(1, dataA); // تعيين قيمة للمخطط

ThingSpeak.setField(2, dataB);

ThingSpeak.setField(3, dataC);

int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

if(x == 200){

Serial.println("Channel update successful."); }

else{

Serial.println("Problem updating channel. HTTP error code " + String(x));

}

lastTime = millis();

}

}

شرح الكود:

قد يراودك السؤال الاتي : لماذا لم استخدم نفس الكود السابق و قمت باستخدام تعليمة

ThingSpeak.writeFields(); في كل مرة و الجواب هو الوقت !

ففي كل مرة تقوم برفع قيمة جدول يجب عليك انتظار 30 ثانية حتى تستطيع الرفع مرة اخرى و لكن يوجد لدينا التعليمة التالية التي تسمح لنا بتعيين كافة البيانات قبل عملية الرفع

ThingSpeak.setField(FieldNumber, value);

يجب ان نضع مكان FieldNumber رقم المخطط الذي تريد تعيين القيمة له

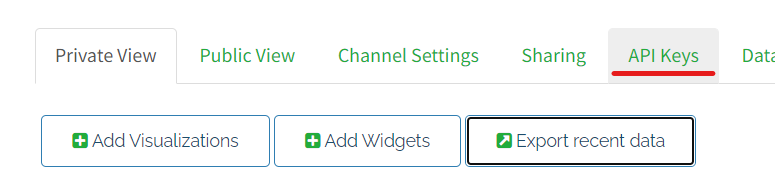
و تضع مكان value القيمة التي تريد

القراءة من مخطط بياني على موقع thingSpeak

و الان بما اننا تعلمنا كيفية رفع البيانات الى موقع thingSpeak اليوم يجب علينا ان نتعلم العكس و هو قراءة تلك البيانات اذ انه من الممكن ان تضطر الى قراءة البيانات التي رفعتها للتاكد منها او لاستخدمها على متحكم esp اخر

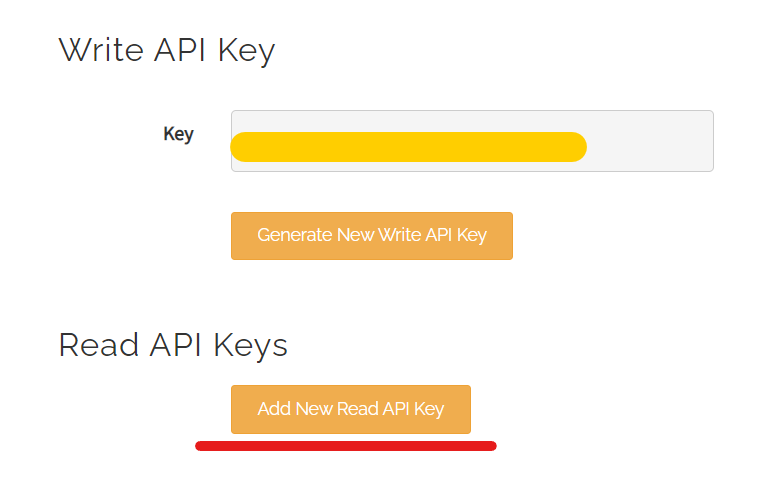
و لنقوم بذلك سنستخدم مكتبة thingspeak نفس المكتبة التي استخدمناها في الامثلة السابقة, و في الواقع الكود في المثالين لا يختلف كثيرا

و لكن قبل ان نبدا بكتابة الكود علينا ان نحصل على مفتاح API مخصص للقراءة Read API key و السبب في وجود مفتاح API للقراءة و اخر للكتابة هو من اجل تحديد صلاحيات المستخدمين فمثلا قد تريد ان تسمح لشخص بقراءة البيانات فقط دون اضافة بيانات جديدة و العكس صحيح

و للحصول على Read API key :

ادخل الى تبويب API KEYS

و بعدها اضغط على زر Add New Read API Key لانشاء مفتاح قراءة جديد و لا تنسى ان تحفظ التغييرات التي قمت بها



الكود:

/\*

Adapted from Example from ThingSpeak Library (Mathworks)

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files.

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

\*/

#include <WiFi.h>

#include "ThingSpeak.h"

const char\* ssid = "REPLACE\_WITH\_YOUR\_SSID";

const char\* password = "REPLACE\_WITH\_YOUR\_PASSWORD";

WiFiClient client;

unsigned long myChannelNumber = X; // برقم Xبدل

int myFieldNumber = X; // برقم Xبدل

const char \* myReadAPIKey = "XXXXXXXXXXXXXXXX";

// متغيرات الوقت

unsigned long lastTime = 0;

unsigned long timerDelay = 30000;

void setup() {

Serial.begin(115200);

if(WiFi.status() != WL\_CONNECTED){ // الاتصال بشبكة الواي-فاي

Serial.print("Attempting to connect");

while(WiFi.status() != WL\_CONNECTED){

WiFi.begin(ssid, password);

}

Serial.println("Connected.");

}

ThingSpeak.begin(client); // Initialize ThingSpeak

}

void loop() {

if ((millis() - lastTime) > timerDelay) {

int statuscode = 0;

String count = ThingSpeak.readStringField(myChannelNumber,myFieldNumber,myReadAPIKey);

statusCode = ThingSpeak.getLastReadStatus();

if(statuscode == 200){

Serial.println("Channel update successful."); }

else{

Serial.println("Problem updating channel. HTTP error code " + String(x));

}

lastTime = millis();

}

}

شرح الكود:

كما لاحظنا سابقا فان الكود لا يختلف كثيرا عن الكود السابق و لذلك فانني لن اعيد كتابة نفس الاشياء التي مرت سابقا و ساركز على الاشياء الجديدة

const char \* myReadAPIKey = "XXXXXXXXXXXXXXXX";

myReadAPIKey هو متغير سنخزن فيه مفتاح API المخصص للقراءة انتبه المخصص للقراءة و ليس للكتابة (الذي انشاناه سابقا)

String count = ThingSpeak.readStringField(myChannelNumber,myFieldNumber,myReadAPIKey);

سنقوم بقراءة البيانات الموجودة في المخطط البياني و تخزينها في متغير نصي

انتبه : البيانات التي حصلنا عليها هي string وليس int و لذلك اذا كنت تريد القيام بعمليات حسابية على البيانات التي حصلت عليها (بافتراض ان هذه البيانات هي ارقام بالاساس) فعليك استخدام دالة .toInt() التي شرحت استعمالها في فصل التعامل مع السلاسل

statusCode = ThingSpeak.getLastReadStatus();

لمعرفة ما اذا تمت القراءة بشكل صحيح او لا فاننا سنستخدم هذه الدالة و نعين متغير لحفظ القيمة التي ستعيدها , اذا كانت هذه القيمة تساوي 200 فامورك باللوز و الا مع الاسف لم تتم عملية الرفع بنجاح, و سنقوم باعادة تعيين قيمة المتغير في كل حلقة لتخزين بيانات جديدة

قراءة عدة مخططات بيانية:

لنفرض انك تريد قراءة اكثر من مخطط بياني في وقت واحد , طبعا انت تستطيع ان تكرر نفس الكود السابق لكن هذه الطريقة تستهلك كثيرا من الوقت المهم , لذلك فاليوم سنستخدم طريقة اكثر كفاءة لقراءة جميع المخططات في قناة واحدة فقط

الكود:

#include <WiFi.h> *// استدعاء المكتبات اللازمة*

#include "ThingSpeak.h"

**const** **char** \*ssid     = "phb";

**const** **char** \*password = "phb@1973";

WiFiClient  client;

**int** myFieldNumber = 2;

**unsigned** **long** myChannelNumber = 838446;

**const** **char** \* myReadAPIKey = "K5T1Z7MUFGIR2XMX";

*// متغيرات الوقت*

**unsigned** **long** lastTime = 0;

**unsigned** **long** timerDelay = 30000;

**void** setup() {

  Serial.begin(115200);

  if(WiFi.status() != WL\_CONNECTED){      *// الاتصال بشبكة الواي-فاي*

      Serial.print("Attempting to connect");

      while(WiFi.status() != WL\_CONNECTED){

        Serial.println(".");

        WiFi.begin(ssid, password);

        break;

      }

      Serial.println("Connected.");

    }

  ThingSpeak.begin(client);  *//thingSpeak تهيئة للاتصال بموقع*

}

**void** loop() {

**int** statusCode = 0;

   if ((millis() - lastTime) > timerDelay) {

   statusCode = ThingSpeak.readMultipleFields(myChannelNumber,myReadAPIKey);

  if(statusCode == 200){

**int** dataA = ThingSpeak.getFieldAsInt(1);

**float** dataB = ThingSpeak.getFieldAsFloat(2);

  String dataC = ThingSpeak.getFieldAsString(3);

  String createdAt = ThingSpeak.getCreatedAt();

  }

  else{

    Serial.println("Problem reading channel. HTTP error code " + String(statusCode));

  }

      lastTime = millis();

  }

}

شرح الكود:

statusCode = ThingSpeak.readMultipleFields(myChannelNumber,myReadAPIKey);

لقراءة بيانات كل المخططات البيانية في القناة , و المتغير statuscode يخزن قيمة html response و هي القيمة العددية التي تخبرنا ما اذا تمت عملية القراءة او لا , اذا كانت العملية ناجحة فان قيمة المتغير ستكون 200

**int** dataA = ThingSpeak.getFieldAsInt(1);

الحصول على البيانات الموجودة في المخطط رقم 1 و تخزينها كقيمة عددية (int)

ال

String dataC = ThingSpeak.getFieldAsString(3);

الحصول على البيانات الموجودة في المخطط البياني رقم 3 و تخزينها كقيمة نصية (string)

و بشكل عام تستطيع الحصول على البيانات بتغيير رقم المخطط في الدالة اعلاه

String createdAt = ThingSpeak.getCreatedAt();

الحصول على زمن انشاء القناة وانا حقا لا اعلم ماذا ستستفيد من معرفة ذلك لكنني رايت من انه لا مانع من الحصول على بعض المعلومات الغير مفيدة

خاتمة:

و هكذا اظن ان فصل موقع thingspeak قد انتهى و اتمنى انك قد استفدت من تلك المعلومات و استطعت تطبيقها بشكل عملي و مفيد.

المصادر التي استخدمتها :

randomnerdtutorials.com

فصل الحصول على الوقت

كلنا نعرف اهمية الوقت و اظن انني لا احتاج الى كتابة موضوع تعبير لبرهنة ذلك في كتاب يتحدث عن انترنت الاشياء , لكن بالاضافة الى اهمية الوقت في حياتنا اليومية , فان للوقت اهمية كبيرة جدا في مشاريع انترنت الاشياء , اذ ان معرفة التوقيت مهمة للاتصال بالسيرفرات و لتسجيل المعلومات , على سبيل المثال في الفصل السابق عندما كنا نتعامل مع موقع thingspeak كانت البيانات تسجل في قاعدة البيانات بالاضافة الى وقت رفعها , لكن ربما لم تلاحظ ذلك لان السيرفر كان يستخدم ساعته الخاصة ليسجل وقت الرفع اوتوماتيكا و لا يحتاج الى الحصول على تلك البيانات من esp , لكن قد تضطر احيانا الى ان تسجل وقت قراءة هذه المعلومات و ليس وقت رفعها الى السيرفر لذلك سنتحدث في هذا الفصل عن طرق الحصول على الوقت من الانترنت:

ما هو بروتوكول وقت الشبكة ntp :

بروتوكول وقت الشبكة او ntp هو بروتوكول يستخدم لمزامنة ساعات الاجهزة حول العالم , نستطيع الحصول على الوقت من هذا البروتوكول عن طريق ارسال طلب و ستقوم السيرفرات بحساب الوقت التاخير الزمني لوصول البيانات عبر الانترنت و ارسال الوقت اليك

تستخدم سيرفرات ntp ساعات نووية و ايضا ساعات GPS للحصول على وقت دقيق , كما انها لا تقوم بحساب منطقتك الزمنية (Time Zone) و انما تقوم السيرفرات باعطائك وقت غرينتش و عليك انت ان تقوم بالحسابات اللازمة لتعديل الوقت ليناسب منطقتك الزمنية

و لذلك في هذا الفصل سنستخدم سيرفرات NTP للحصول على وقت للesp

الحصول على الوقت باستخدام مكتبة ntpClient :

اولا قم بتحميل مكتبة ntpClient من الرابط <https://github.com/taranais/NTPClient>

لاحظ ان هذه المكتبة تختلف عن مكتبة ntpClient الموجودة في library manger و بالتالي عليك تحميلها من الرابط هنا حصرا و الا ستعاني من العديد من الاخطاء

(هذه المكتبة هي fork (يعني مثل نسخة معدلة) عن مكتبة ntpClient الاصلية)

قم بتحميل المكتبة على شكل ملف مضغوط (zip) ثم قم بتغيير اسم الملف من

**NTPClient-master الى NTPClient**

ثم قم بوضعه في ملف تحميل المكتبات الخاص بالاردوينو

و اخيرا قم باعادة تشغيل Arduino IDE

الكود:

#include <WiFi.h>

#include <NTPClient.h>

#include <WiFiUdp.h>

*//قم بادخال بيانات شبكتك*

**const** **char**\* ssid     = "REPLACE\_WITH\_YOUR\_SSID";

**const** **char**\* password = "REPLACE\_WITH\_YOUR\_PASSWORD";

*//ntp تهيئة الاتصال بسيرفر*

WiFiUDP ntpUDP;

NTPClient timeClient(ntpUDP);

*// متغيرات لحفظ الوقت و التاريخ*

String formattedDate;

String dayStamp;

String timeStamp;

**void** setup() {

  Serial.begin(115200);

*//الاتصال بالانترنت*

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL\_CONNECTED) {

    Serial.print(".");

  }

*// بدء الاتصال بالسيرفر*

  timeClient.begin();

*// قم بوضع فرق التوقيت عن غرينتش بالثواني و هذه امثلة*

*// كل ساعة تساوي 3600 ثانية*

*// GMT (X) = (X)\*3600*

*// GMT +1 = 3600*

*// GMT +8 = 28800*

*// GMT -1 = -3600*

*// GMT 2 = 7200*

*// GMT 0 = 0*

  timeClient.setTimeOffset(3600);

}

**void** loop() {

  while(!timeClient.update()) {

    timeClient.forceUpdate();

  }

*// البيانات التي تصلنا ستاتي على الشكل الاتي*

*// 2018-05-28T16:00:13Z*

*// و سنقوم باستخراج البيانات التي نريدها من خلال اجراء عمليات على النصوص كما تعلمنا سابقا*

  formattedDate = timeClient.getFormattedDate(); *// الحصول على الوقت و تخزينه في متغير*

  Serial.println(formattedDate);

*// استخراج البيانات*

**int** splitT = formattedDate.indexOf("T");

*// سنقوم بفصل النص الى قسمين واحد للوقت و الاخر للتاريخ*

  dayStamp = formattedDate.substring(0, splitT); *//التاريخ*

  Serial.print("DATE: ");

  Serial.println(dayStamp);

*// Extract time*

  timeStamp = formattedDate.substring(splitT+1, formattedDate.length()-1); *//الوقت*

  Serial.print("HOUR: ");

  Serial.println(timeStamp);

  delay(1000); *// سنقوم بالاستغناء عنها لاحقا لكن سنضعها الان في هذا المثال مؤقتا*

}

timeClient.setTimeOffset(3600);

هذه التعليمة نستخدمها لتحديد فرق توقيت غرينتش فعلى سبيل المثال انا اعيش في دمشق و دمشق فرق توقيتها عن غرينتش هو +2 ساعة و بالتالي ساضع الوقت 7200 ثانية

طبعا اذا وضعت الزمن بين قوسين 2 لن تحصل على وقت صحيح لان الدالة تاخذ بارامتر بالثواني و ليس بالساعات

الامر الاخر هو ان هذه المكتبة لا تدعم التوقيت الصيفي , بمعنى اخر اذا كانت دولتك تغير التوقيت عند الانتقال من فصل الشتاء الى الصيف باضافة او ازالة ساعة فالمكتبة لن تعدل نفسها تلقائيا , و الحل لهذه المشكلة ان تقوم بتحديد دالة تختبر ما اذا كان التاريخ يساوي التاريخ الذي يتم فيه الانتقال من او الى التوقيت الصيفي للقيام بوضع البارامتر المناسب لهذه الدالة

و وضعت بعض التعليقات تحت هذه الدالة لكي تفهم اكثر

\

*// البيانات التي تصلنا ستاتي على الشكل الاتي*

*// 2018-05-28T16:00:13Z*

*// و سنقوم باستخراج البيانات التي نريدها من خلال اجراء عمليات على النصوص كما تعلمنا سابقا*

  formattedDate = timeClient.getFormattedDate(); *// الحصول على الوقت و تخزينه في متغير*

  Serial.println(formattedDate);

و بعد القيام بتحديث بيانات الوقت من سيرفرات ntp سنقوم الان باستخراج هذه البيانات باستخدام دالة

formattedDate = timeClient.getFormattedDate();

لكن هذه الدالة تعطينا الوقت على شكل سلسلة نصية string : 2018–05–28T16:00:13Z

اذا لاحظت تستطيع ان تجد ان بيانات التاريخ تنتهي عند المحرف T و بعده تاتي بيانات الوقت و بالتالي فان المحرف الفاصل بين التاريخ و الوقت هو T و كل شيء قبله هو للتاريخ و كل شيء بعده هو للوقت , كما ان المحرف Z يدلنا على انتهاء بيانات الوقت (اي نهاية السلسلة النصية)

و بالتالي سنقوم الان بتطبيق ما اخذناه في الفصل الاول (العمل على السلاسل) لاستخراج بيانات الوقت و التاريخ

لقطع جزء من سلسلة نصية نستخدم الدالة subString(begin,end) و التي تاخذ بارامترين الاول هو المحرف الذي تريد ان تبدأ السلسلة الجديدة به و البارامتر الثاني هو المحرف الذي تريد السلسلة الجديدة ان تنتهي عنده

نحن الان نريد ان نحصل على التاريخ من السلسلة : 2018–05–28T16:00:13Z

لاحظ ان بيانات الوقت تبدأ من الصفر و تنتهي عند T , طبعا دالة subString تاحذ index و ليس محرف , بمعنى اخر يجب ان اخبرها موقع (index) الT و ساقوم يتخزينه في متغير لتسهيل البرمجة

-اذا كنت لا تعرف ما هو index راجع الفصل الاول قسم subsring-

و بالتالي ساسنخدم دالة indexOf(string) و التي تعطيني index لاي محرف او سلسلة نصية

**int** splitT = formattedDate.indexOf("T");

هنا قمت بتخزين index الخاص بالمحرف T في متغير int سميتهsplitT

  dayStamp = formattedDate.substring(0, splitT); *//التاريخ*

هنا سنقوم بتخزين السلسلة التي تحوي التاريخ في متغير نصي سنسميه dayStamp قمنا بالتصريح عنه مسبقا , و طبعا نحن نعلم ان سلسلة التاريخ هي محصورة بين اول محرف و المحرف قبل الحرف T