



جامعة حلب - المناطق المحررة

كلية الهندسة المعلوماتية

السنة الرابعة

النظم الرقمية المبرمجة

تقرير بعنوان

WIFI CAR



إعداد الطلاب: محمد ضبيط & مهند حمود & أحمد الحاج حسين

الرقم الجامعي: 53 & 65 & 11

إشراف: د.م. عبد القادر غزال

العام الدراسي

2018/2019

مقدمة:

يهدف هذا التقرير إلى شرح كيفية صنع سيارة صغيرة وبرمجتها والتحكم بها عن طريق الـ WiFi باستخدام الحاسوب.

المحتويات:

- ❖ الأدوات المستخدمة، وتتضمن شرح عن كل أداة.
- ❖ طريقة التوصيل.
- ❖ برمجة شريحة ESP8266.
- ❖ برمجة تطبيق للتحكم بالسيارة.

الأدوات المستخدمة:

- (1) روبوت سيارة دفع رباعي 4WD Robot Car.
- (2) دائرة قيادة المحركات L298N Dual H-Bridge Motor Drive.
- (3) لوحة Node MCU تحوي شريحة ESP-12F WIFI ESP8266.
- (4) بطاريات ليثيوم Li Ion Battery.
- (5) باور سبلاي YwRobot 545043 Breadboard Power Supply (MB102).
- (6) كاميرا (هاتف Android).
- (7) أسلاك توصيل Jumpers.

(1) روبوت سيارة دفع رباعي 4WD Robot Car:

- تحتوي هذه السيارة على عدة مكونات جعلت منها روبوت قابل للحركة وهذه المكونات هي:
- ✓ أربع محركات (DC Gear Motor) صغيرة تعمل المحركات على جهد 6V فولت مع علبة سرعة لزيادة العزم.
 - ✓ أربع عجلات (Trolley Wheels).
 - ✓ هيكل سيارة مع قطع تثبيت (براغي، صمن، ...).



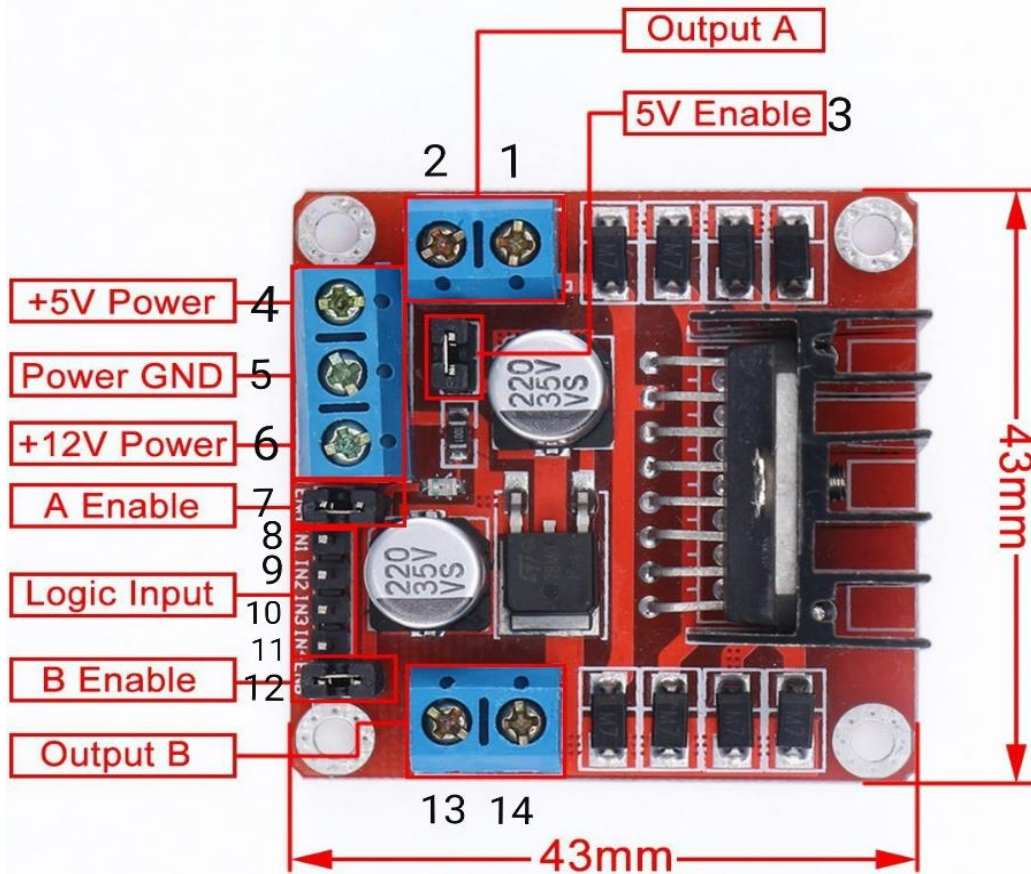
الشكل (1): مكونات روبوت سيارة دفع رباعي

(2) دائرة قيادة المحركات L298N Dual H-Bridge Motor Drive:

إن استخدام الدارة المتكاملة بحد ذاته يحتاج مركبات إلكترونية جانبية، مثل الديودات والمقاومات ومبددات الحرارة، لذلك قمنا باستخدام الشريحة المسماة L298N وهي لوحة جاهزة، تسهل التعامل مع هذه الدارة المتكاملة وتختصر الوقت والجهد، و L298N هي دائرة تستخدم لقيادة محركين كهربائيين بتيار أعظمي 2A لكل محرك، وتتميز بسهولة التحكم وتحملها لجهد يصل حتى 36V.

أطراف دائرة القيادة L298N:

- (1) القطب الموجب للمحرك الأول (A+).
- (2) القطب السالب للمحرك الأول (A-).
- (3) وصلة 12V، (يتم إزالتها في حالة المحركات التي تحتاج أكثر من 12V).
- (4) مقبس توصيل الجهد اللازم لتشغيل المحركات من 5V وحتى 35V.
- (5) مقبس توصيل الأرضي GND.
- (6) مخرج 5V مناسب لتوصيل الإلكترونيات الخاصة بالتحكم في حالة استخدام جهد أعلى من 12V ويمكن استخدامه لتشغيل لوحة Arduino، أو ESP 8266 ...
- (7) تمكين المتحكم من التحكم بالمحرك A (من أجل التحكم بالسرعة يتم توصيله بمنفذ PWM).
- (8) مدخل 1.
- (9) مدخل 2.
- (10) مدخل 3.
- (11) مدخل 4.
- (12) تمكين المتحكم من التحكم بالمحرك B (من أجل التحكم بالسرعة يتم توصيله بمنفذ PWM).
- (13) القطب الموجب للمحرك الثاني (B+).
- (14) القطب السالب للمحرك الثاني (B-).



الشكل (2): أطراف دائرة القيادة

(3) لوحة Node MCU تحوي شريحة ESP8266 ESP-12F WIFI

هي عبارة عن لوحة صغيرة الحجم مفتوحة المصدر تحوي على معالج صغير 32bit، وهي قابلة للبرمجة وتحتوي أيضاً على دائرة WiFi لربط المتحكم بالانترنت مما يوفر خاصية انترنت الأشياء والتي تسمح بربط الأشياء مع بعضها والتفاهم فيما بينها من خلال شبكة الأنترنت (يقصد بعبارة الأشياء جميع الأجهزة الذكية).
الأطراف:



- عشرة أطراف D1 ~ D10 نستطيع استخدامها كدخل أو خرج وتدعم خاصية PWM.
- طرف D0 لا يدعم خاصية PWM.
- طرف A0 يستخدم كدخل تماثلي.
- طرف Vin (3.3V - 10V).
- ثلاث أطراف 3.3V.
- أربعة أطراف أرضي GND.

الأبعاد:

تأتي اللوحة بطول 47 ملم وعرض 26 ملم.

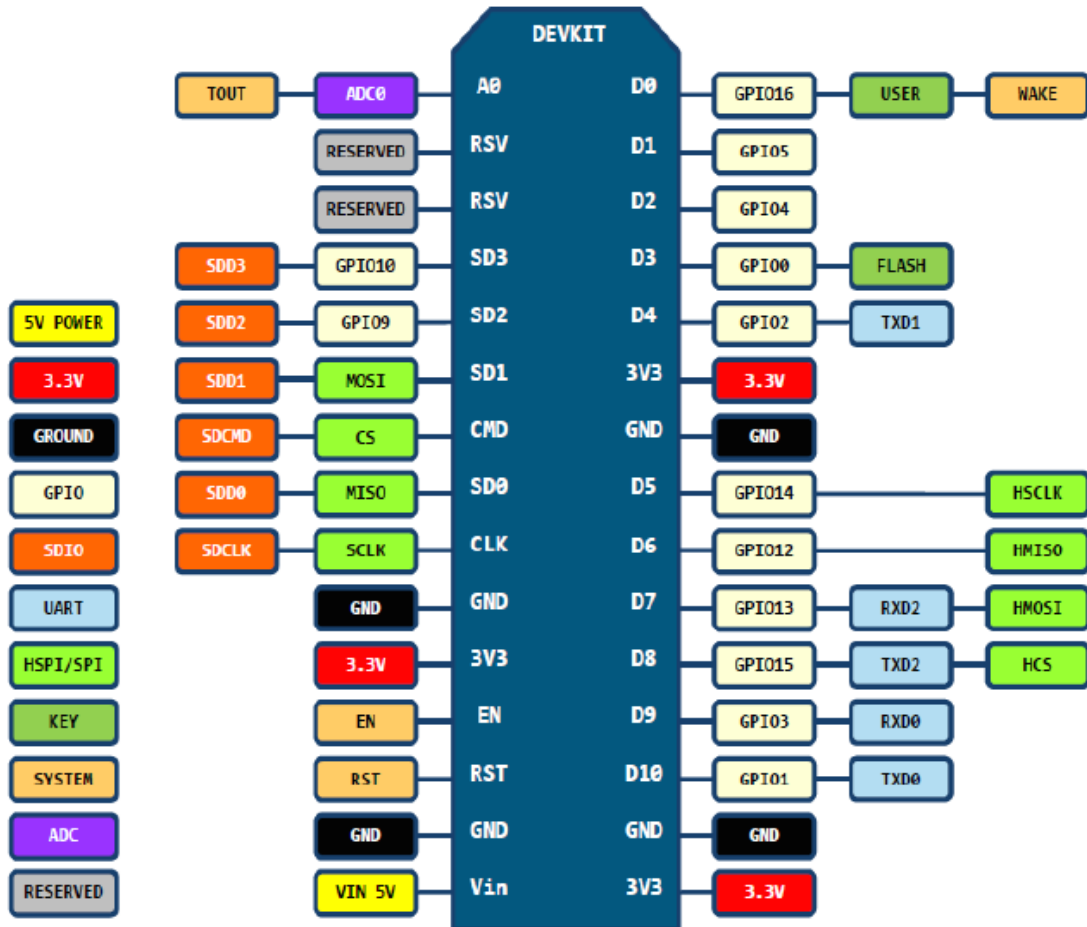
المفاتيح:

تحتوي اللوحة على مفتاحين من النوع (Push Button):

1. المفتاح الأول Flash Button: يستخدم عند تثبيت نظام NodeMCU.
2. المفتاح الثاني Reset Button: يستخدم عند رفع برنامج (شيفر برمجية) جديدة على NodeMCU.

الطاقة:

تعمل شرائح ESP8266 على جهد 3.3V، ولكن الإصدار الذي نستخدمه (الإصدار الثاني) يمكن تشغيله على جهد يتراوح بين 3.3V - 10V والسبب أنها مزودة بمنظم جهد لتخفيض الجهد إلى 3.3V. يعتبر 10V أعلى جهد يمكن استخدامه ويفضل استخدام 5V.



الشكل (4): تفاصيل الأطراف في مخطط اللوحة

➤ نلاحظ وجود طرفين باللون الرمادي (NC) غير متصلة أي أنها غير مستخدمة.

(4) بطاريات ليثيوم Li Ion Battery:

تم استخدام ثلاث أزواج من بطاريات ليثيوم (البطارية الواحدة 3.7V) من أجل التغذية.

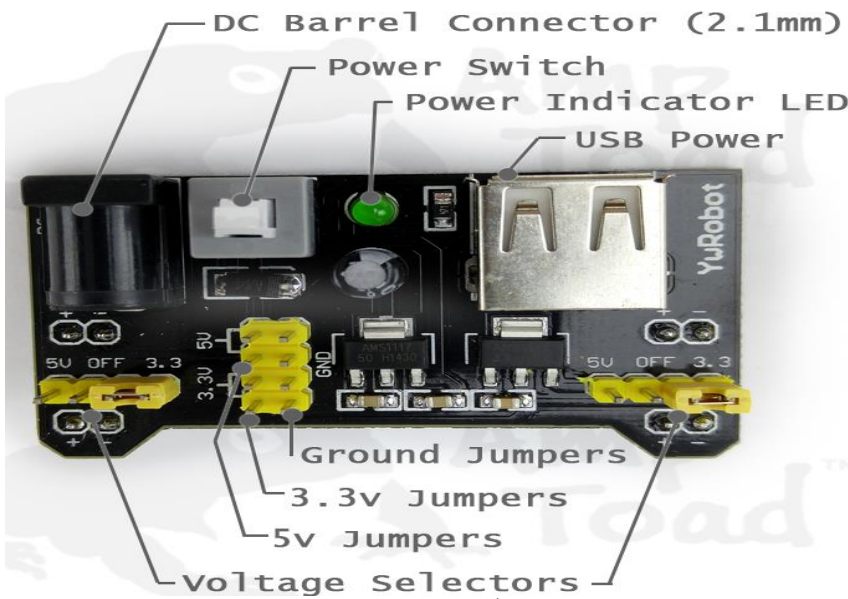


الشكل (5): بطاريات ليثيوم 3.7V

(5) باور سبلاي YwRobot 545043 Breadboard Power Supply (MB102):

تقوم هذه اللوحة بتحويل الجهد المدخل من (6.5V ~ 12V DC) إلى (3.3V / 5V)، لن نتطرق لشرح مفصل عن الشريحة، وسنكتفي بشرح مبسط عن الأطراف المستخدمة:

- مفتاح لفتح أو إغلاق الطاقة (On / Off Switch).
- مدخل 6.5V ~ 12V DC.
- مخرج (5V / 3.3V) Usb.



الشكل (6): لوحة YwRobot 545043

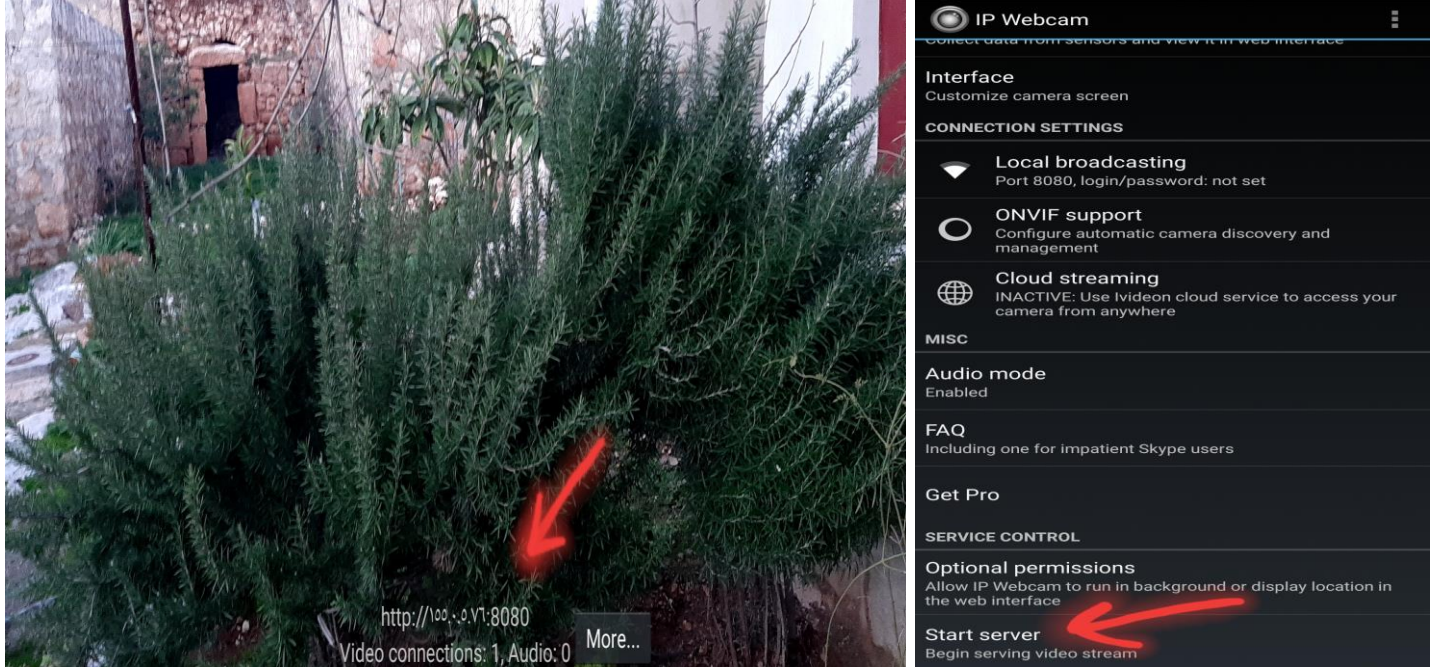
(6) كاميرا (هاتف Android):

يمكن استخدام الكاميرا الخاصة بالهاتف المحمول ونقل الصورة عن طريق أحد برامج نقل الصورة من الهاتف مثلاً IP Webcam، وبفضل هذا التطبيق يمكن تحويل هاتف الأندرويد إلى كاميرا مراقبة مما يمكن من مراقبة أي شيء عن بعد، هذا التطبيق متوفر على جوجل بلاي وهو متوافق مع جميع إصدارات نظام الأندرويد.

طريقة عمل تطبيق IP Webcam:

- نقوم بتنصيب التطبيق على الهاتف المحمول ثم نقوم بفتح التطبيق - نضبط الإعدادات أو نتركها كما هي - وستظهر واجهة التطبيق، نحرك إلى الأسفل حتى نجد كلمة start server كما في الشكل (7) نضغط عليها.

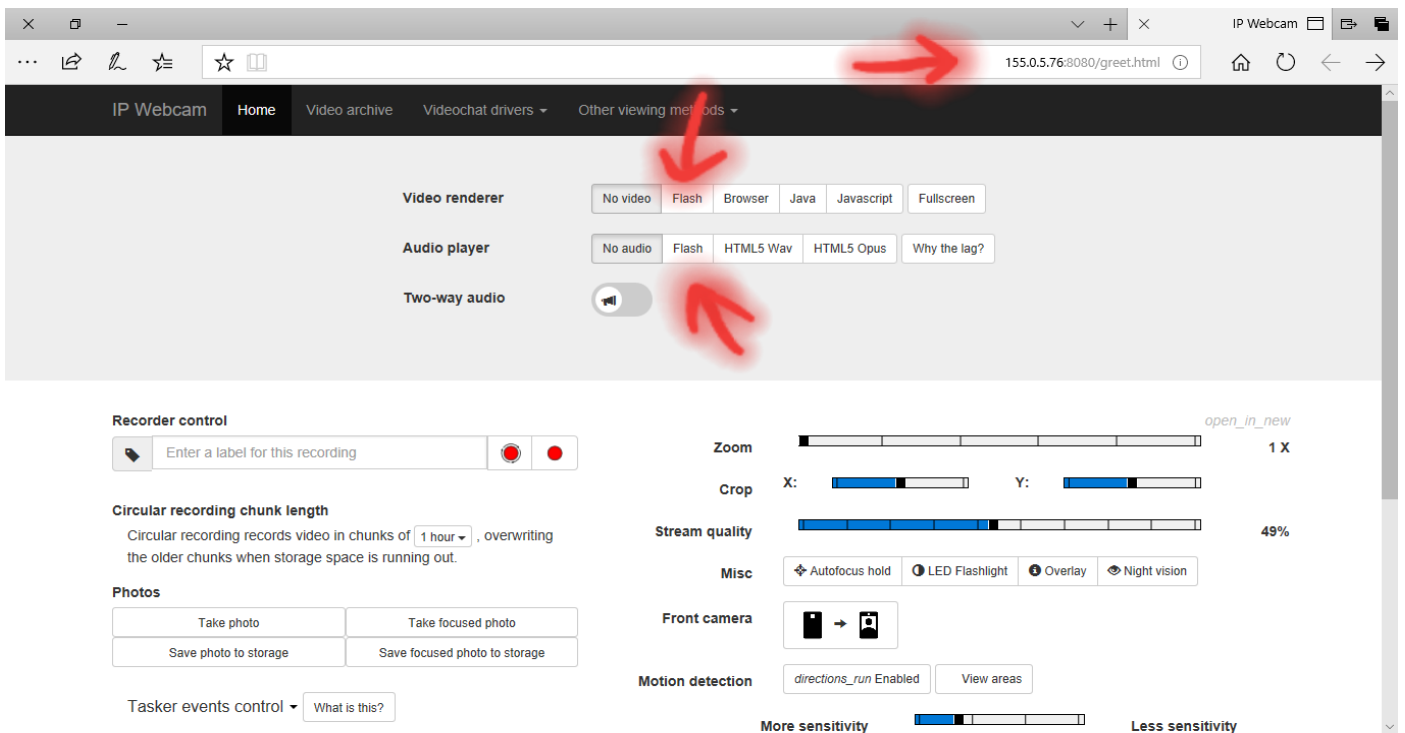
- قبل الضغط على زر Start server يجب أن نتأكد من أننا متصلين مع الحاسوب في نفس الشبكة (إن لم توجد شبكة متوفرة يمكننا عمل شبكة لاسلكية Hotspot باستخدام الحاسوب وتوصيل الهاتف بها أو بالعكس).
- بعد الضغط على الزر Start server سيظهر بث مباشر لما تلتقطه كاميرا هاتف الأندرويد، نقوم بأخذ عنوان IP الذي يظهر منتصف أسفل الشاشة كما في الشكل (8) ونقوم بإدخاله في المتصفح الخاص بنا على الحاسب.



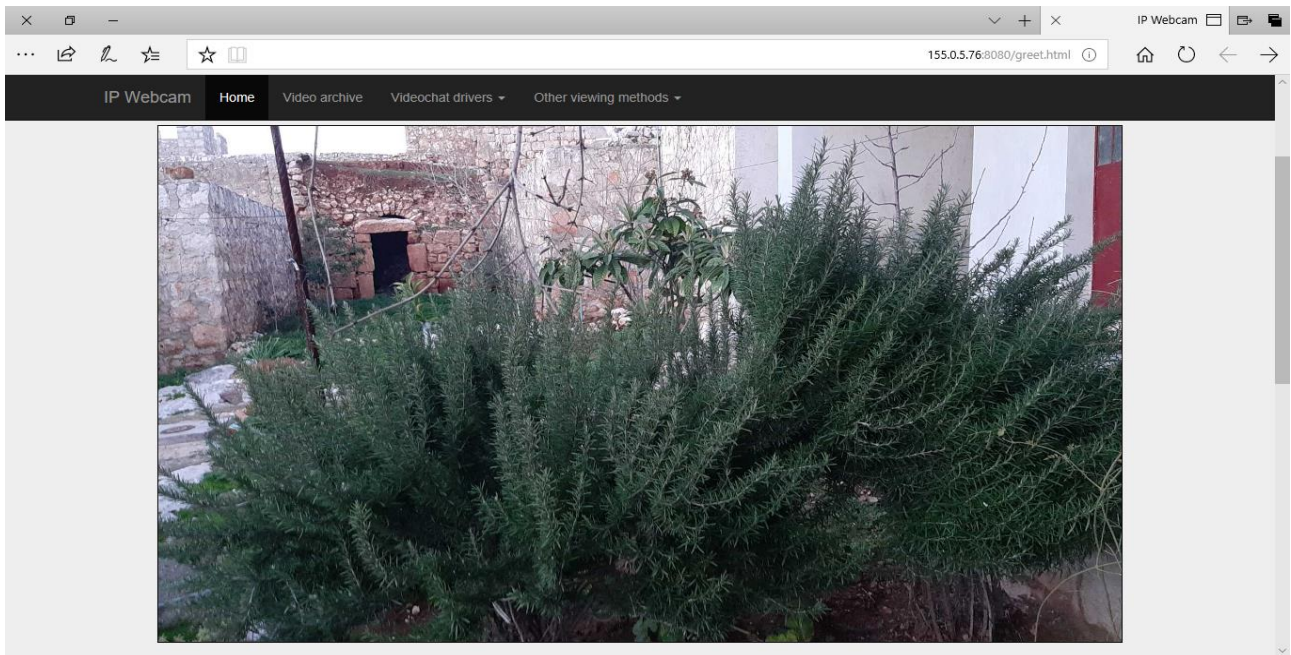
الشكل (8): البث المباشر على هاتف الأندرويد

الشكل (7): واجهة تطبيق IP Webcam

- بعد إدخال عنوان IP في المتصفح، ستظهر صفحة كما في الشكل (9)، ولمشاهدة البث الحي الذي يلتقطه الهاتف نقوم بالضغط على كلمة Flash الموجودة أمام كلمة Video Renderer، وللاستماع إلى صوت البث نقوم بالضغط على Flash التي أمام Audio Player.



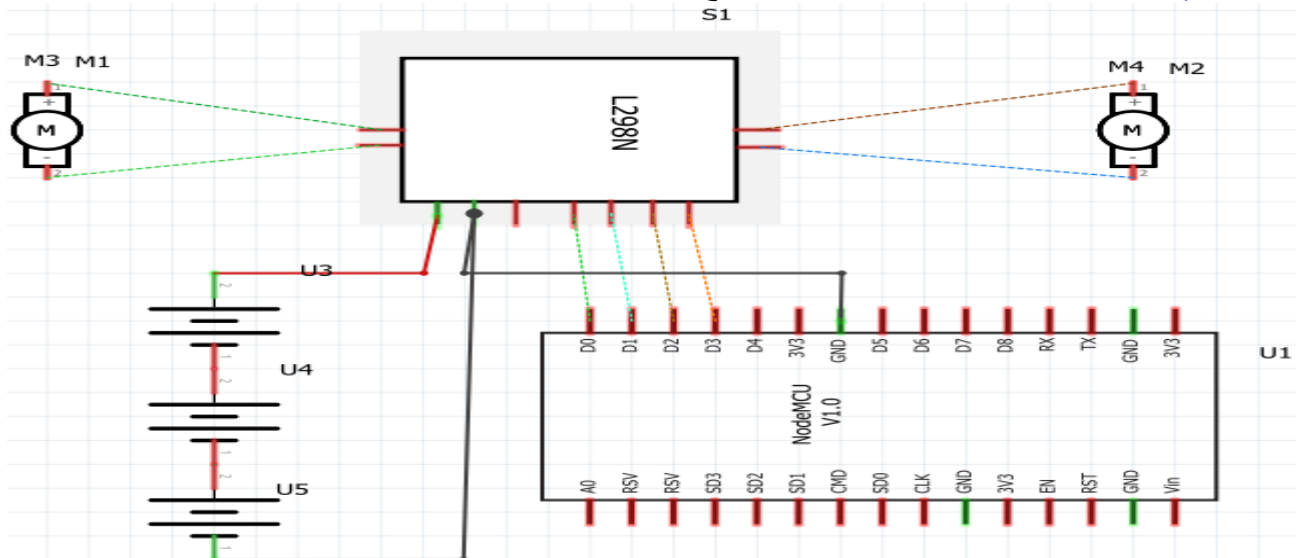
الشكل (9): الصفحة التي تظهر بعد كتابة الـ IP على متصفح الحاسب



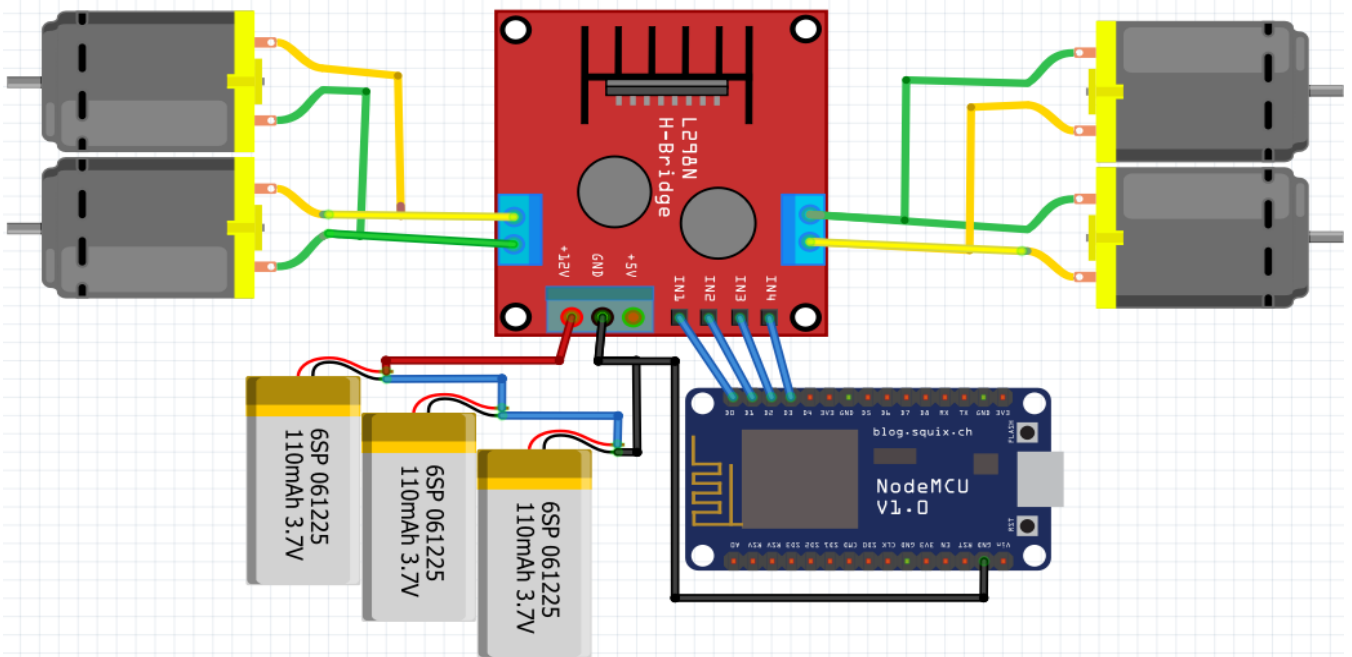
الشكل (10): البث المباشر للهاتف على متصفح الحاسب

طريقة التوصيل:

- نقوم بوصل كل بطاريتين مع بعض على التفرع ثم نقوم بوصل الأزواج الناتجة على التسلسل فنحصل على تغذية ذات جهد خرج 11.1V وسعة 1500uA، ثم نوصل البطارية الناتجة مع دائرة القيادة كالتالي:
 - (1) نقوم بتوصل الجهد الموجب للبطارية بمدخل 12V لدائرة القيادة.
 - (2) نقوم بتوصل الجهد السالب للبطارية بمدخل GND لدائرة القيادة.
- نقوم بوصل المحركات مع دائرة القيادة كالتالي (لكي نستطيع التحكم باتجاه دوران السيارة بسهولة):
 - (1) نقوم بربط المحركين الأول والثاني مع بعضهما لنتمكن من التحكم بهما كمحرك واحد ثم نقوم بربط المحرك بالمدخلين Out1 , Out2.
 - (2) نقوم بربط المحركين الثالث والرابع مع بعضهما لنتمكن من التحكم بهما كمحرك واحد ثم نقوم بربط المحرك بالمدخلين Out3 , Out4.
- نقوم بوصل لوحة NodeMCU مع دائرة القيادة كالتالي:
 - (1) القطب D0 من لوحة NodeMCU مع القطب IN1 من دائرة القيادة.
 - (2) القطب D1 من لوحة NodeMCU مع القطب IN2 من دائرة القيادة.
 - (3) القطب D2 من لوحة NodeMCU مع القطب IN3 من دائرة القيادة.
 - (4) القطب D3 من لوحة NodeMCU مع القطب IN4 من دائرة القيادة.



الشكل (11): مخطط التوصيل Schematics



الشكل (12): رسمة الدارة على برنامج محاكاة

برمجة شريحة ESP8266:

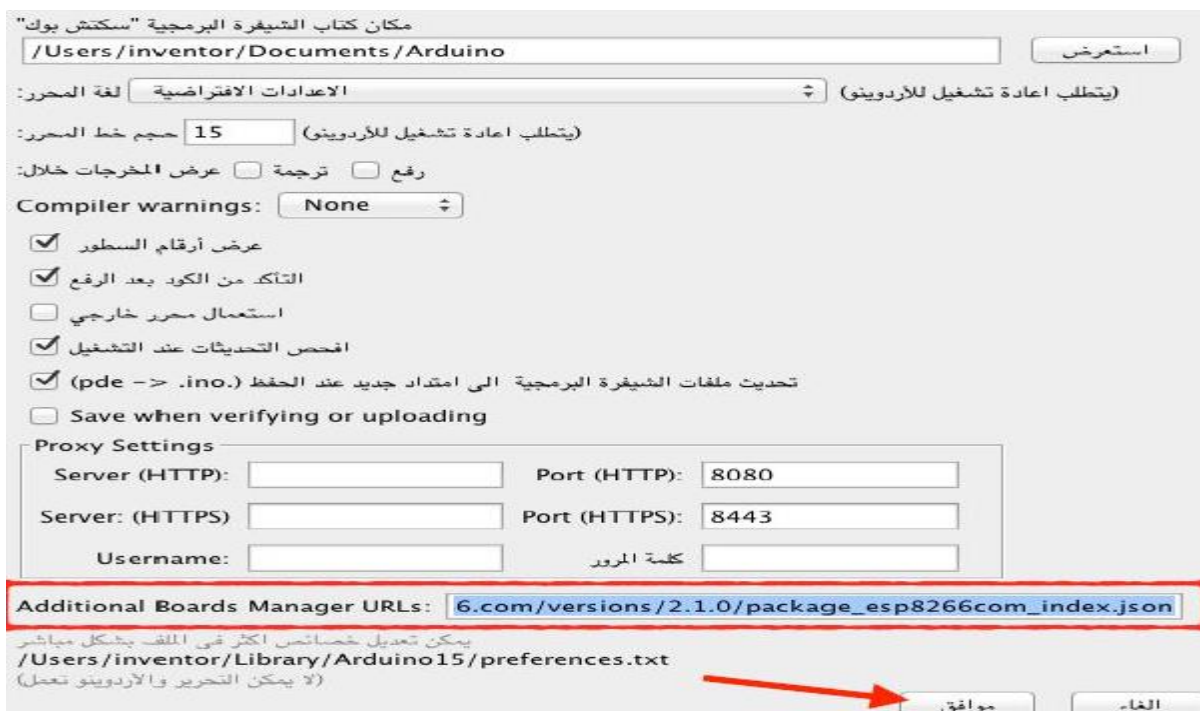
من أجل برمجة شريحة ESP8266 سنستخدم برنامج Arduino IDE وهو بيئة تطويرية يتم كتابة الأسطر والأوامر البرمجية عليه من أجل برمجة أنواع مختلفة من الشرائح والمتحكمات.

نقوم بتحميل برنامج Arduino IDE من الموقع التالي: <https://www.arduino.cc/en/Main/Software>

وبعد تثبيت البرنامج يجب أن نضيف حزمة التشغيل الخاصة بشريحة ESP8266 من خلال الخطوات التالية:

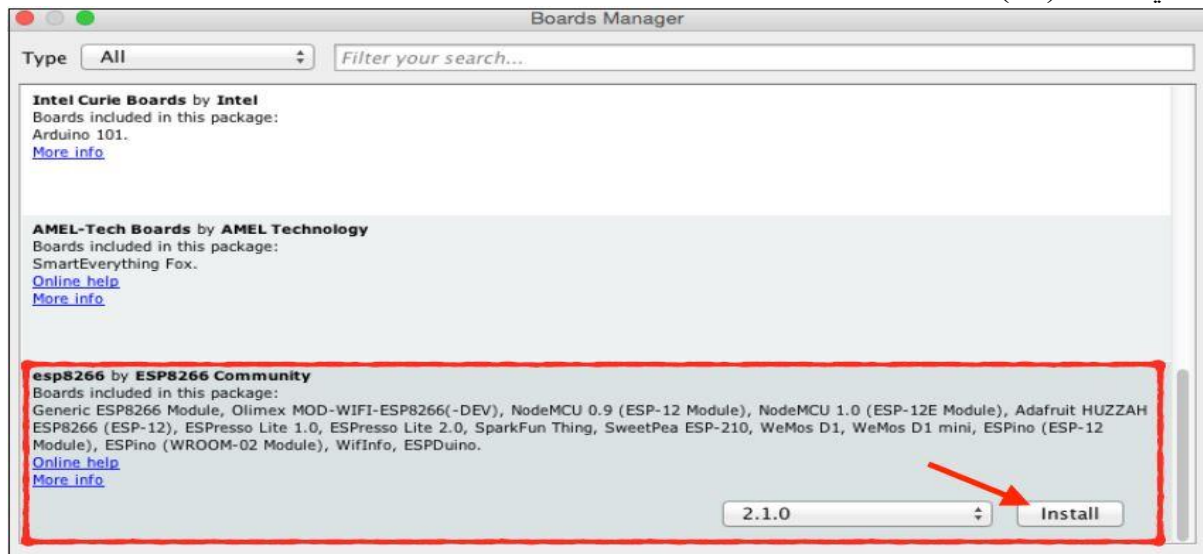
➤ نفتح برنامج الأروينو وننتقل إلى المسار التالي: File ثم Preferences، ستظهر صفحة التفضيلات، نتجه لخاصة (additional boards manager urls) كما في الشكل (13)، ونضع في هذه الخانة الرابط

التالي: http://arduino.esp8266.com/versions/2.1.0/package_esp8266com_index.json ونضغط موافق.



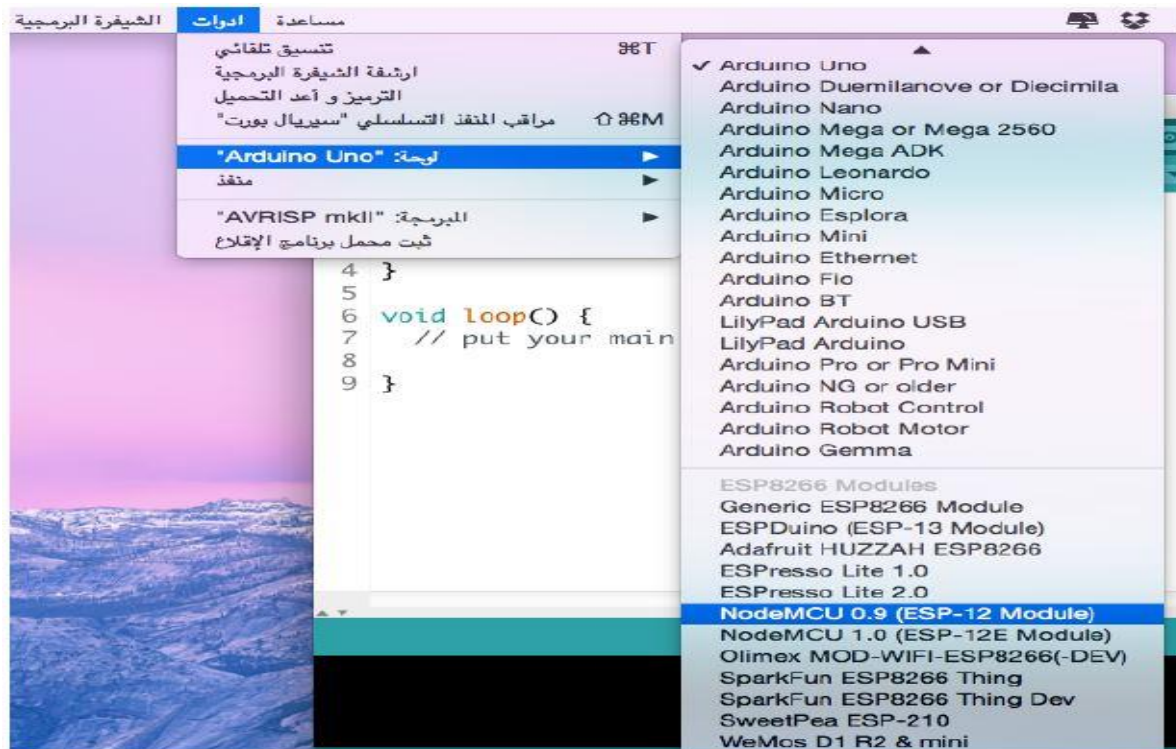
الشكل (13): صفحة التفضيلات في برنامج Arduino IDE

- بعدها سننتقل للمسار التالي: Tools ثم Board ثم Boards Manager ، ستظهر صفحة إدارة اللوحات، نلاحظ وجود العديد من الحزم، نبحث عن حزمة ESP8266 ونحدد عليها ثم نضغط على زر Install كما في الشكل (14)، وسيبدأ تثبيت الحزمة.



الشكل (14): صفحة إدارة اللوحات في برنامج Arduino IDE

- بعد انتهاء تثبيت الحزمة سننتقل إلى المسار التالي: Tools ثم Board وسنلاحظ وجود لوحات إضافية من ضمنها لوحة NodeMCU كما في الشكل (15).



الشكل (15): واجهة برنامج Arduino IDE

- الآن نكتب الكود التالي :

```
#include <ESP8266WiFi.h> // تضمين مكتبة الواي فاي
WiFiClient client; // تعريف عميل
WiFiServer server(80); // سنستخدم المنفذ 80 في المتصفح للوصول للشريحة
const char* ssid = "Abo kamel"; // اسم الشبكة وكلمة السر
```

```
const char* password = "123456aa";
```

```
String data = ""; // متحول نصي لتخزين الأمر المرسل إلى الشريحة
```

```
void setup () {
```

```
  Serial. Begin (9600);
```

```
  delay (10);
```

```
  pinMode (D5, OUTPUT);
```

```
  pinMode (D6, OUTPUT);
```

```
  pinMode (D7, OUTPUT);
```

```
  pinMode (D8, OUTPUT);
```

```
  /***** Connect to WiFi network *****/
```

```
  Serial.println();
```

```
  Serial.print("Connecting to ");
```

```
  Serial.println(ssid);
```

```
  WIFI. Begin (ssid, password);
```

```
  while (WIFI. Status () != WL_CONNECTED) { // عدم بدء البرنامج حتى الاتصال بالشبكة
```

```
    delay (500);
```

```
    Serial.print(".");
```

```
  }
```

```
  Serial.println("");
```

```
  Serial.println("WiFi connected");
```

```
  /***** start server communication *****/
```

```
  server. Begin (); // بدء السيرفر
```

```
  Serial.println("Server started");
```

```
  /***** print the IP address *****/
```

```
  Serial.print("Use this URL to connect: ");
```

```
  Serial.print("http://");
```

```
  Serial.print(WiFi.localIP());
```

```

Serial.println("/");
}

void loop () {

  /***** If the server available, run the "check Client" function *****/

  client = server. Available ();

  if (!client) return;

  data = checkClient () ; // الأمر المرسل يخزن عن طريق التابع

  /***** Run function according to incoming data from application *****/

  if(data=="f")    // إذا كان الأمر المرسل حرف الألف شغل المحركات باتجاه الأمام لمدة نصف ثانية ثم توقف
  {analogWrite(D8,700);
  analogWrite(D7,0);
  analogWrite(D6,700);
  analogWrite(D5,0);
  delay (500);
  stop ();
  Serial.println(" forward is Ok"); }

  else if(data=="b")
  { analogWrite(D8,0);
  analogWrite(D7,700);
  analogWrite(D6,0);
  analogWrite(D5,700);
  delay(500);
  stop();
  Serial.println(" back is Ok"); }

  else if(data=="r")
  { analogWrite(D8,900);
  analogWrite(D7,0);
  analogWrite(D6,0);
  analogWrite(D5,0);

```



```

delay(500);

stop();

Serial.println(" right is Ok"); }

else if(data=="I")
{ analogWrite(D8,0);
analogWrite(D7,0);
analogWrite(D6,900);
analogWrite(D5,0);
delay(500);
stop();
Serial.println(" left is Ok"); }

else stop();
}

```

/****** RECEIVE DATA FROM The WEB *****/

```

String checkClient (void) // تابع استلام الأمر من العميل
{
    while(!client.available()) delay(1); // ما دام العميل متاح استلم الأوامر منه
    String request = client.readStringUntil('\r'); // اقرأ أول سطر من بيانات العميل
    request.remove(0, 5); // يحذف أول خمس محارف متعلقة بالبيانات المرسلة من العميل
    request.remove(request.length()-9,9); // يحذف آخر تسع محارف من رسالة العميل
    return request; // يصبح الأمر المرسل مستخلص من الزوائد ويتم إرجاعه
}

```

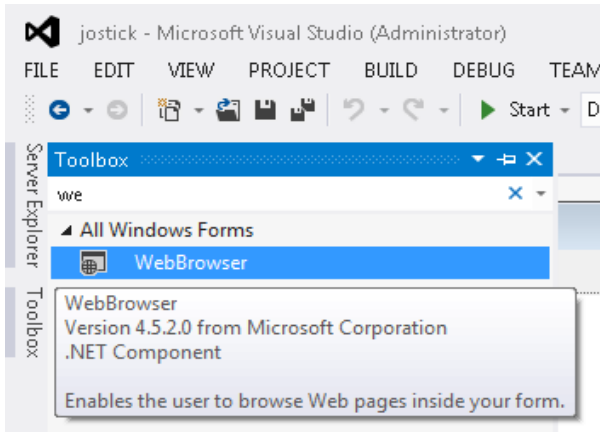
```

void stop(void) {
    digitalWrite(D5,0);
    digitalWrite(D6,0);
    digitalWrite(D7,0);
    digitalWrite(D8,0);
    Serial.print(" stop is Ok"); }

```

برمجة تطبيق للتحكم بالسيارة:

قمنا ببرمجة تطبيق على بيئة الـ Visual Studio باستخدام لغة C#، وباستعمال الـ Windows Forms Application كالتالي:



➤ نقوم بإضافة الأداة: WebBrowser كما في الشكل (16).

➤ نضيف textbox مع أربعة أزرار للتحكم بالاتجاهات الأربعة وزر تشغيل.

➤ سوف نقوم بوضع الـ ip داخل الـ textbox عند التشغيل.

➤ نعرف متحول عام من نوع String نسميه ip.

الشكل (16): إضافة أداة WebBrowser

➤ داخل زر التشغيل نكتب:

```
ip = "http://" + txtip.Text + "/"; groupBox1.Enabled = true;
```

➤ داخل زر الأمام نكتب:

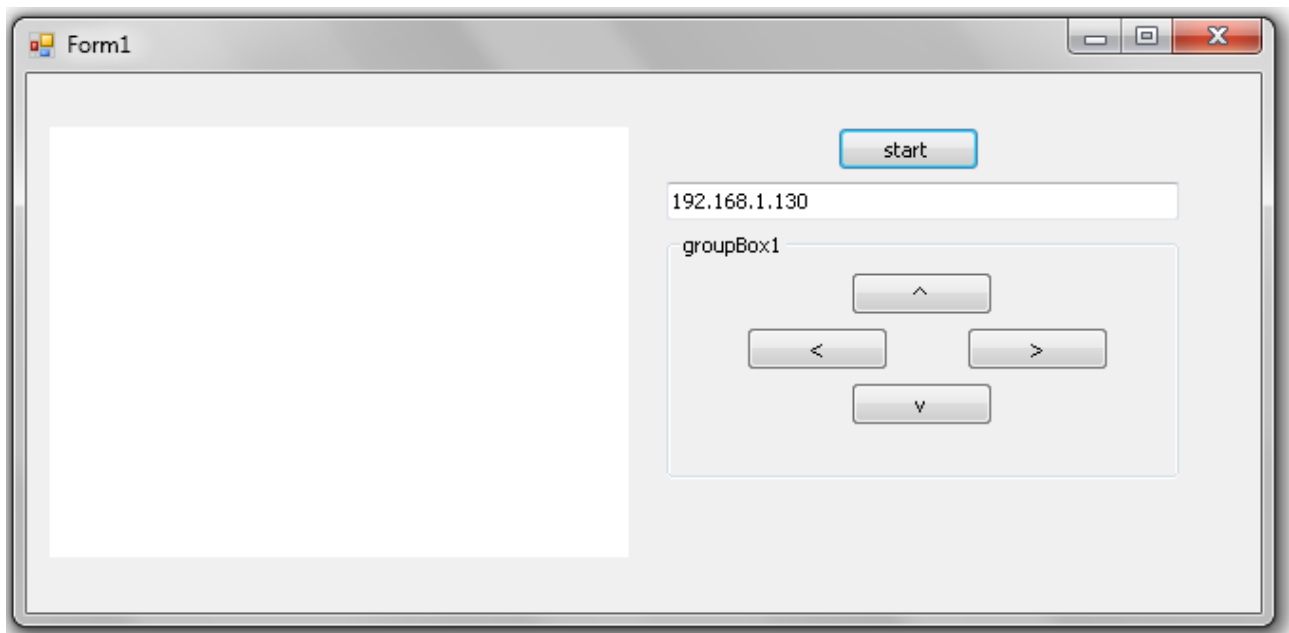
```
webBrowser1.Navigate(ip + "f");
```

➤ بنفس الطريقة نكتب داخل أزرار الخلف واليمين واليسار تبعاً:

```
webBrowser1.Navigate(ip + "b");
```

```
webBrowser1.Navigate(ip + "r");
```

```
webBrowser1.Navigate(ip + "l");
```



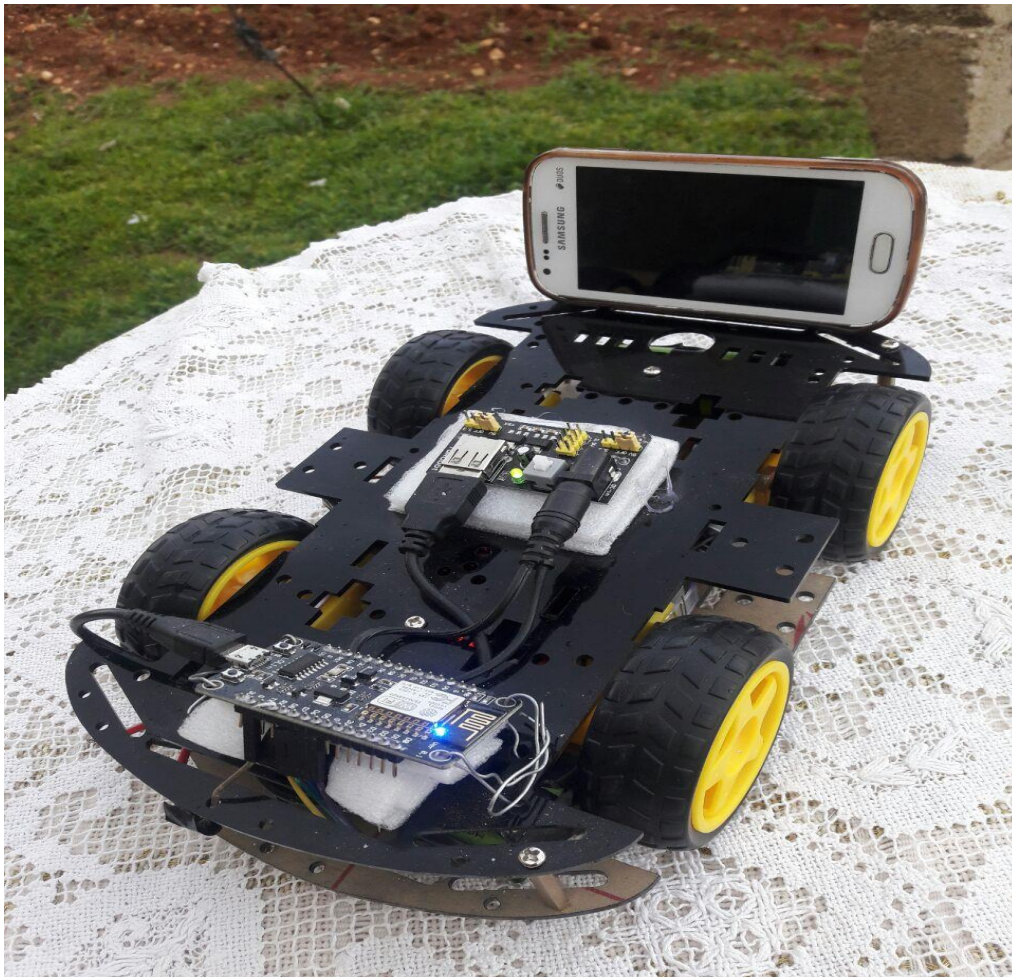
الشكل (17): واجهة التطبيق

النهاية:

الآن أصبح لدينا سيارة يتم التحكم بها عن طريق الـ WiFi.



الشكل (18): صورة للسيارة أثناء تجربتها للتأكد من نجاح العمل



الشكل (19): صورة للسيارة أثناء تجربتها للتأكد من نجاح العمل

المراجع:

- جهاد طلعت بسيوني, شريحة ESP8266 التحكم اللاسلكي من خلال الواجهات الإلكترونية
- <https://www.instructables.com/id/Control-DC-and-stepper-motors-with-L298N-Dual-Moto/>
- https://www.petervis.com/Raspberry_PI/Breadboard_Power_Supply/YwRobot_Breadboard_Power_Supply.html
- <https://youtu.be/uW8YVcBjPGU>