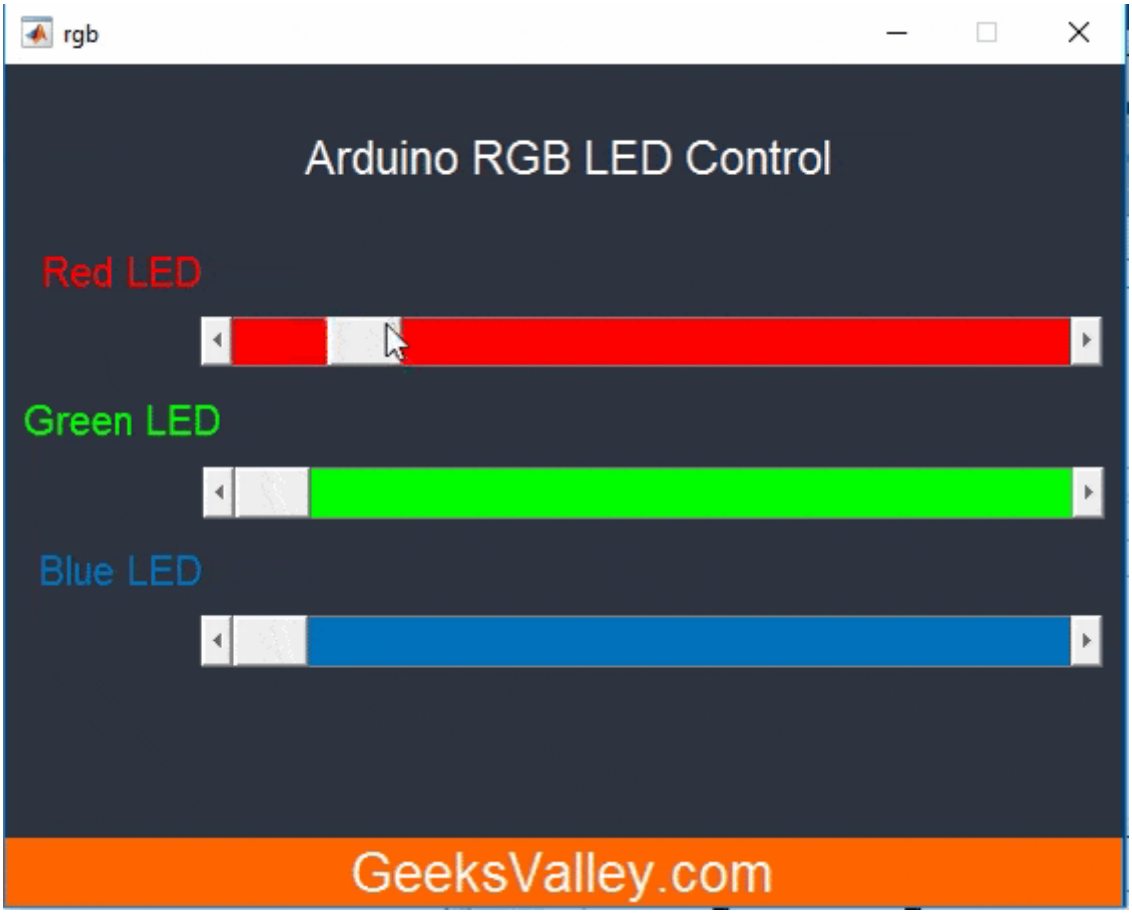
التحكم بإضاءة RGB LED من خلال واجهة رسومية عبر Matlab

الواجهات الرسومية هي عبارة عن عرض بياني ، رسومي في نافذة او أكثر ، يتضمن وسائل ومكونات تتيح للمستخدم إنجاز مهام فعالة وجذابة ضمن بيئة معينة. كما يمكنك من خلال الواجهات كتابة وقراءة البيانات ،وايضاً الربط بين أكثر من واجهة ومشاركة البيانات بينها ،بالإضافة إلى إمكانية عرض هذه البيانات على شكل رسوم بيانية وجداول .

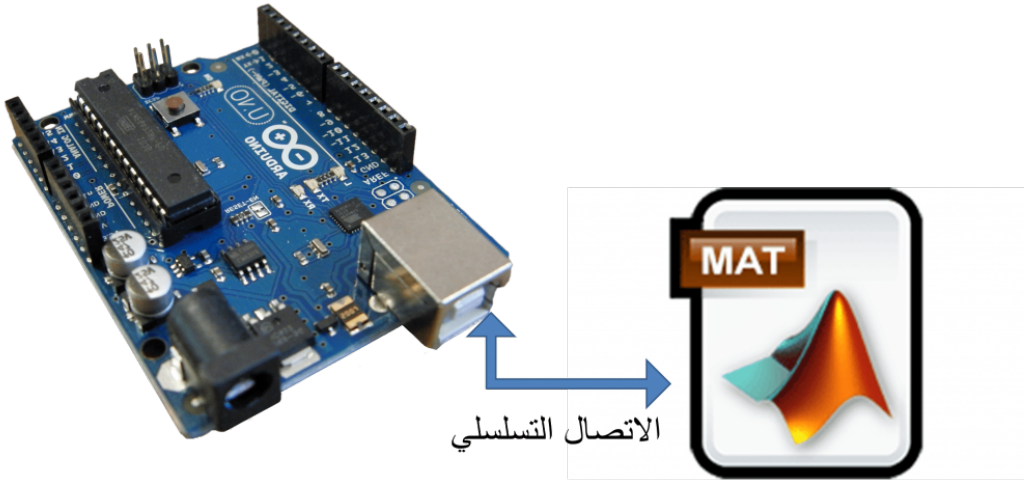
في هذا المشروع ستتعلم كيفية انشاء واجهة المستخدم الرسومية بإستخدام الماتلاب ، للتحكم في RGB LED المتصل بالاردوينو.



الماتلاب Matlab

الماتلاب أنسب للمشاريع المعقدة حيث يسمح بالتلاعب حسابيا بالمصفوفات، وبالرسم البياني للتوابع الرياضية، وبتنفيذ الخوارزميات المختلفة، وإنشاء واجهات المستخدم الرسومية، ومعالجة الصور فهي تعالج تخصص تحليل الصور وكتابة خوارزميات لترتيب البكسلات. فهو أكثر فاعلية , و يمكن تنفيذ أوامر الإدخال و الإخراج لحظياً دون الحاجة للبرمجة، الترجمة، التحميل و التنفيذ كل مرة.

سنقوم بإنشاء واجهة للمستخدم باستخدام برنامج الماتلاب (Matlab) للتحكم بالاجهزة المرتبطة بالاردوينو, ليتم ارسال الأوامر من جهاز الحاسوب (Matlab) للأردوينو عبر المنفذ التسلسلي (Serial port).



تجهيز الأردوينو و إعداد الماتلاب:

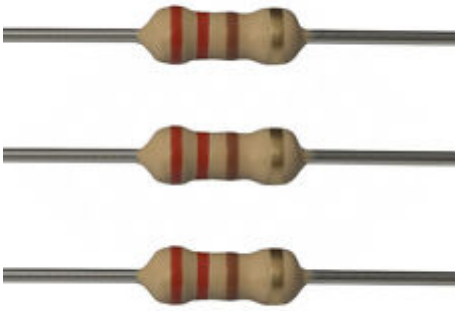
1- قم بتحميل برنامج الماتلاب من الرابط هنا.

2- قم بتحميل حزمة الدعم للأردوينو من الرابط هنا : لقراءة, وكتابة, وتحليل بيانات الأجهزة المربوطة مع الاردوينو .

القطع المطلوبة:

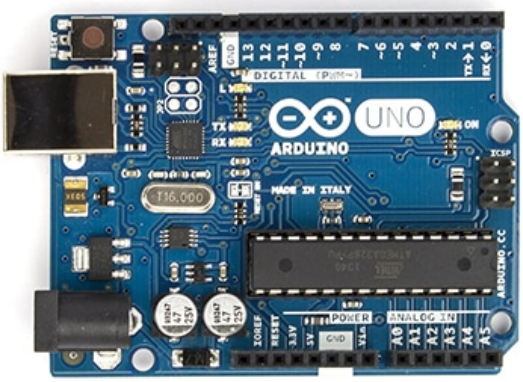
الأدوات التي تحتاجها لهذا المشروع :

RGB 5mm LED

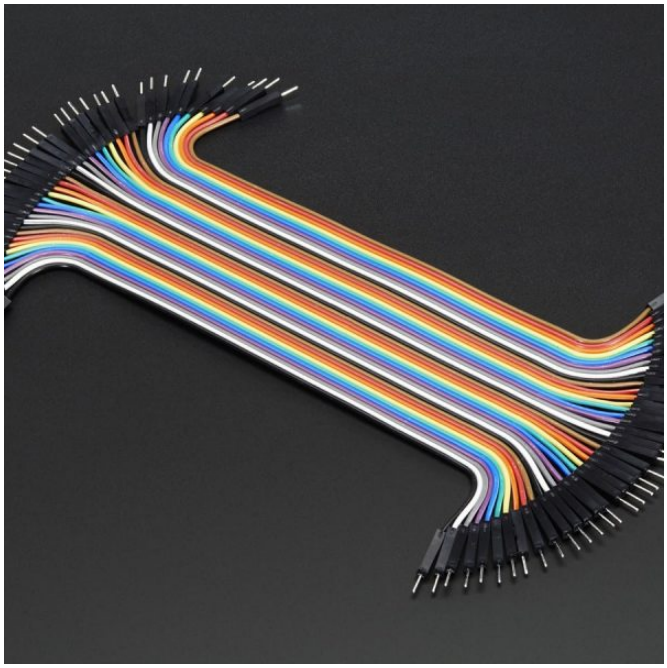


مقاومة 220 اوم

لوح تجارب صغير (small size breadboard)



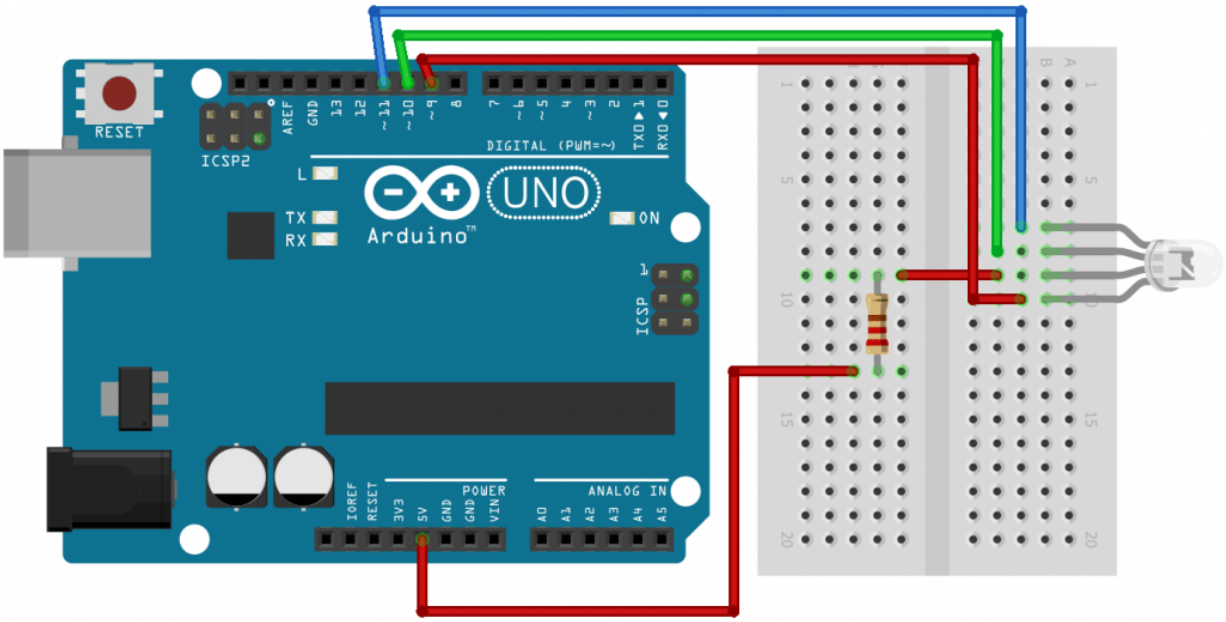
Arduino Uno R3



اسلاك توصيل ذكر/ذكر (Jumper Wires Male Male)

تصميم لوحة التجارب:

قم بتوصيل الدائرة الكهربائية على لوحة التجارب كما هو موضح بالصورة:



يمكنك العودة للدرس الرابع للاطلاع على شرح و تفاصيل اكثر حول RGB LED .

إنشاء الواجهة الرسومية

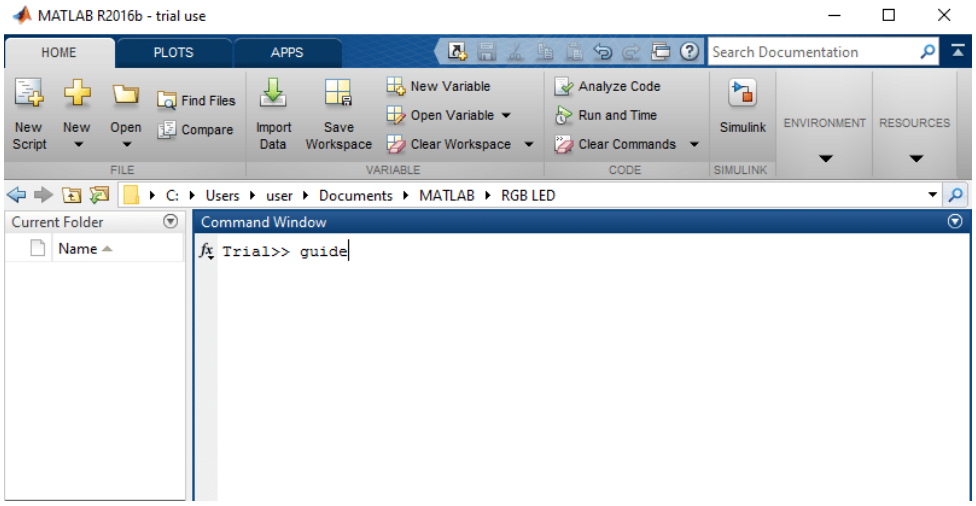
واجهات المستخدم الرسومية GUI's في الماتلاب يتم إنشاؤها بواسطة أداة تدعى :

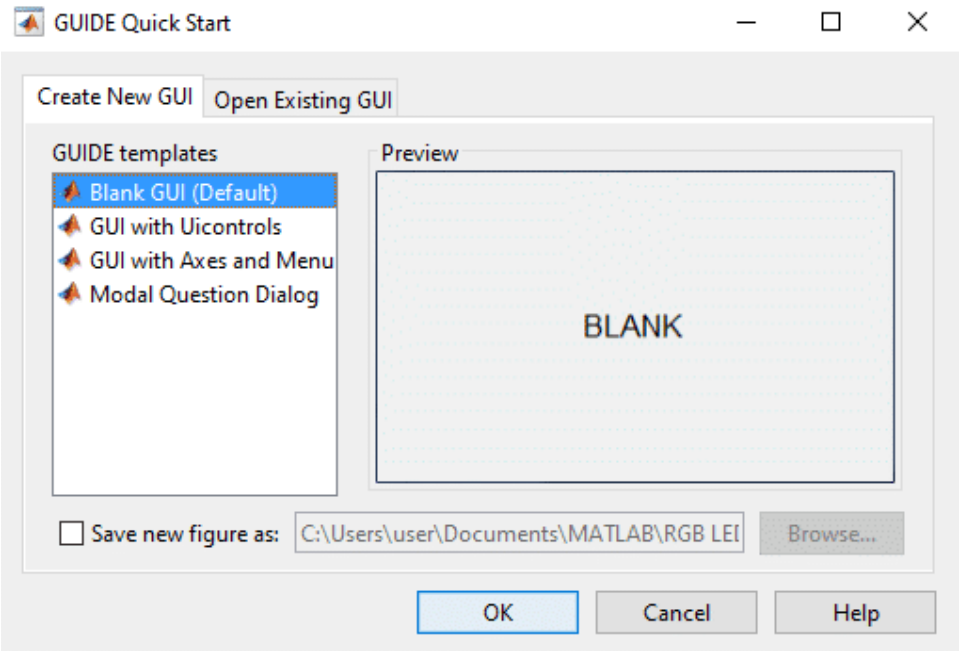
GUIDE : Graphical User Interface Development Environment

هذه الأداة تسمح للمبرمج أن يقوم بتحرير الواجهة الرسومية GUI ،وتنسيقها وإضافة العناصر التي يريدها إلى الواجهة،كما يمكنه تغيير

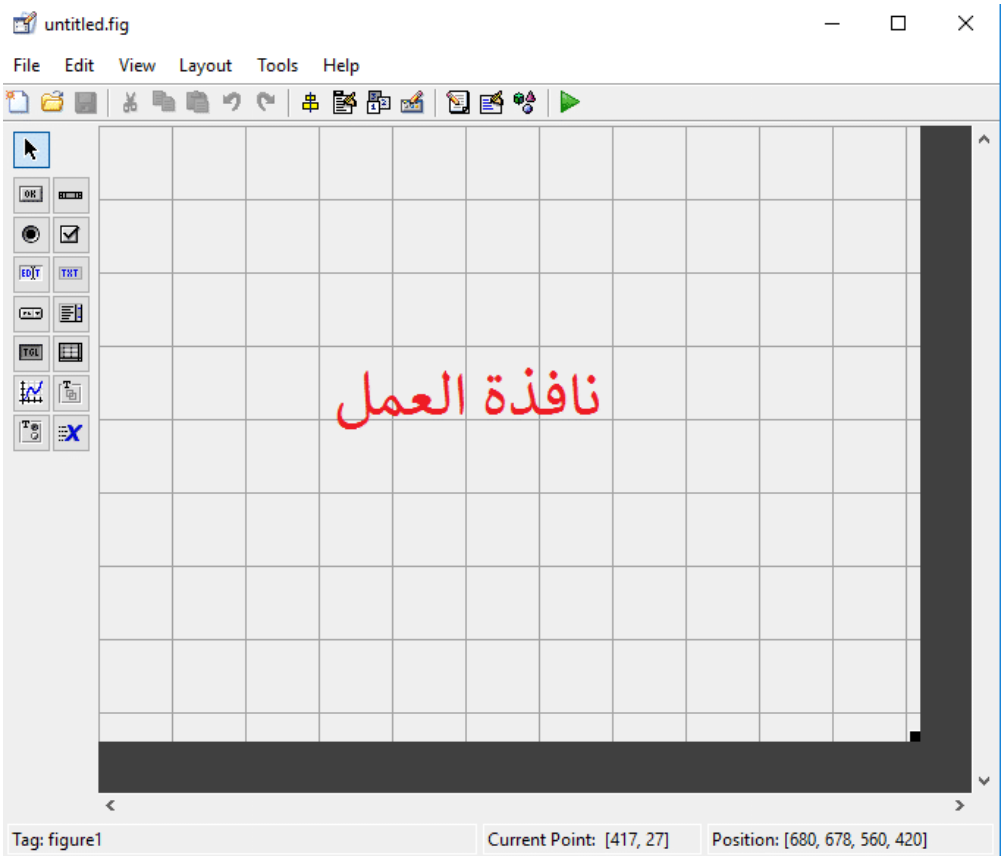
الخصائص لكل كائن في الواجهة: الإسم،اللون،الحجم،نوع الخط,وغيرها من الخصائص .

على شاشة الاوامر للماتلاب قم بكتابة الأمر : guide



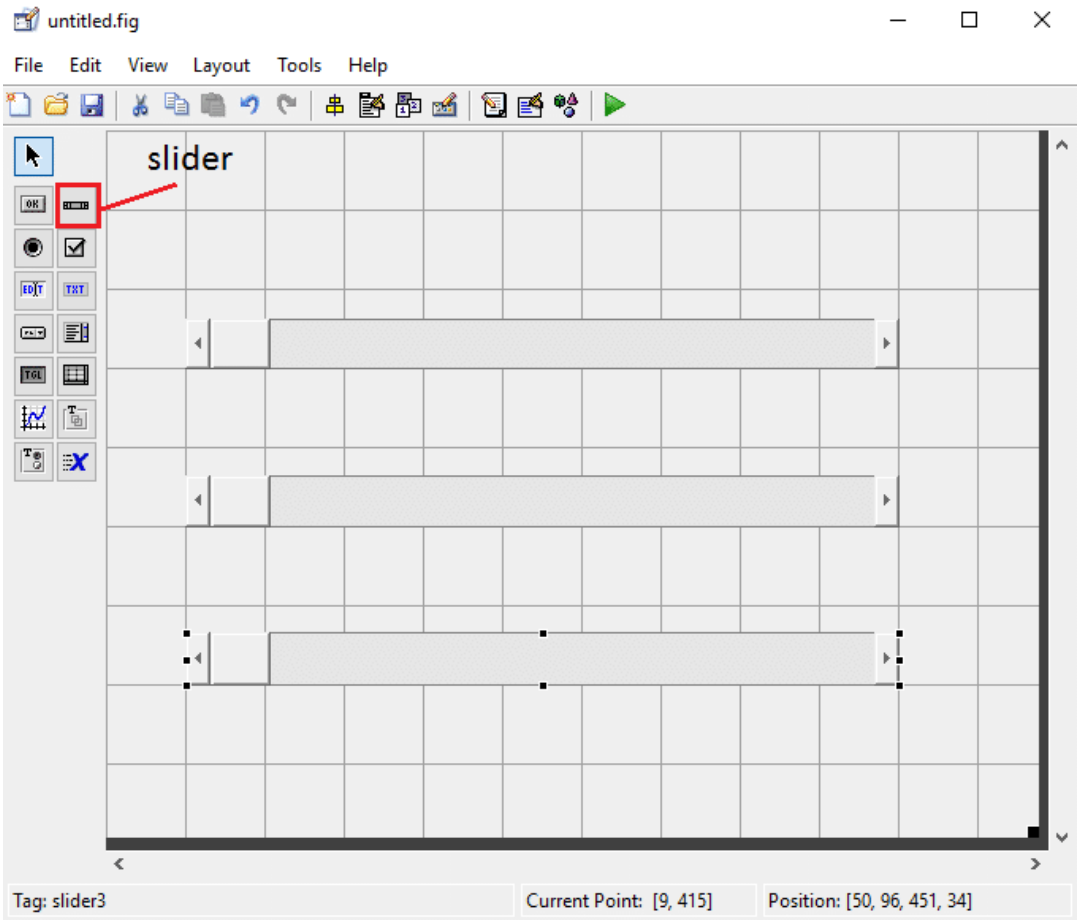


قم بالضغط على Blank GUI(Default) الموجودة في علامة التويب (Create New GUI) كما هو مبين بالصورة اعلاه.
ثم OK ليظهر محرر التخطيط الرسومي للواجهات.

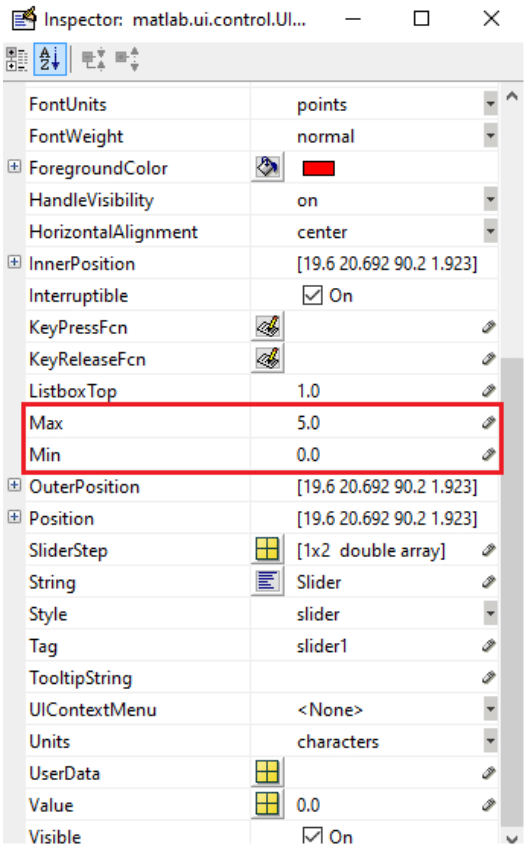


سنقوم بإنشاء ثلاثة Sliders لتتحكم بـ RGB LED عن طريق الواجهة الرسومية.

قم بالضغط على Slider, ثم قم برسم الشكل على نافذة العمل , كما هو مبين بالشكل التالي :

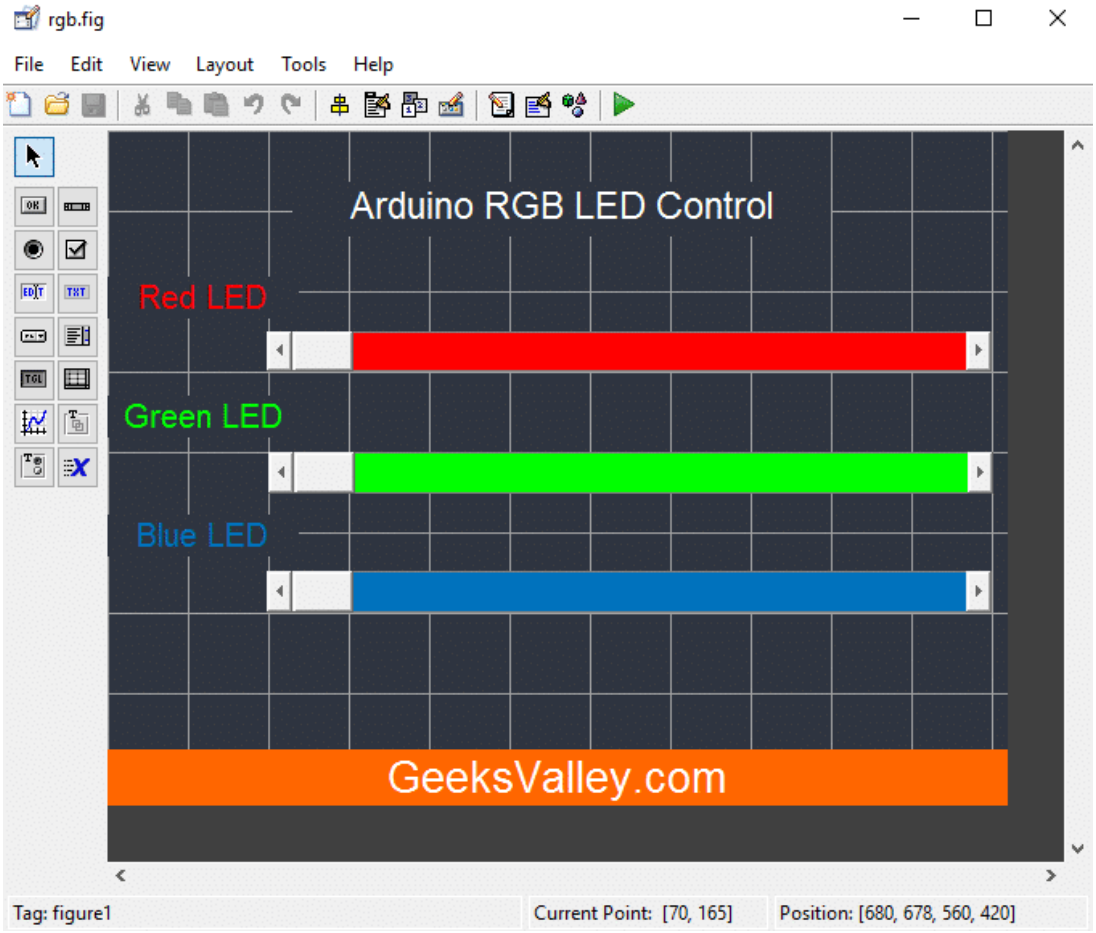


انقر نقرا مزدوجا على كلSlider ليفتح اطار جديد يمكن من خلاله معرفة الخصائص للعناصر المضافة و تغيرها مثل تغيير الخلفية.
قم بتحديد القيمة الأكبر و الاقل لكل slider كما هو موضح بالشكل التالي



في الاردوينو, الجهد على المداخل التناظرية يكون محدود بين 0v-5v .

يمكنك اضافة نص من خلال Static text، و تغير الخصائص المتعلقة بالخط ,و تغير لون خلفية الواجهة.



قم بحفظ save الشكل figure كملف،عندما تقوم بحفظه سيتم تقائيا توليد ملفين لهما نفس الإسم لكن ذو لاحقة مختلفة.

Fig file: يحتوي على الواجهة الرسومية GUI التي قمت بإنشائها.

M-File : يحتوي code التعليمات البرمجية التي تتحكم بتنفيذ وسلوك الواجهة.

الكود البرمجي للواجهة

بعد بحفظ الواجهة الرسومية التي بنيتها بإستخدام الأداة GUIDE، GUIDE سيقوم بتوليد الكود البرمجي لتعمل الواجهة، يتضمن التوابع الهيكلية التي يمكن للمبرمج أن يُعدّل عليها ويتحكم بتنفيذ وسلوك الواجهة الرسومية.

نقوم بفتح والتعديل على الكود البرمجي للواجهة (rgb.m) كما يلي :

```
function varargout = rgb(varargin)

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @rgb_OpeningFcn, ...
                   'gui_OutputFcn',  @rgb_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before rgb is made visible.
function rgb_OpeningFcn(hObject, eventdata, handles, varargin)
% Choose default command line output for rgb
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = rgb_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
clear all;
global a;
a=arduino;


% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata, handles)
b=get(hObject,'Value');
global a;
writePWMVoltage( a , 'D9' , 5-b );

% --- Executes during object creation, after setting all properties.
function slider1_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on slider movement.
function slider2_Callback(hObject, eventdata, handles)
c=get(hObject,'Value');
global a;
writePWMVoltage( a , 'D10' , 5-c );

% --- Executes during object creation, after setting all properties.
function slider2_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on slider movement.
function slider3_Callback(hObject, eventdata, handles)
d=get(hObject,'Value');
```

```
global a;
writePWMVoltage( a , 'D11' , 5-d );

function slider3_CreateFcn(hObject, eventdata, handles)

if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
```

لمحة عن الكود :

إنشاء الاتصال

لإنشاء الاتصال بين الماتلاب ولوحة الاردوينو :

```
a = arduino;
```

او إذا كان لديك أكثر من لوحة اردوينو متصلة قم بتحديد المنفذ و نوع اللوحة كما يلي :

```
a = arduino('COM5', 'uno');
```

الكتابة على المنافذ :

لتعيين القيمة المتغيرة لكل منفذ(PWM pin) نستخدم الامر:

```
writePWMDutyCycle(a, pin, dutyCycle);
```

في هذا المشروع يتم تغير القيمة لكل منفذ اعتمادا على حركة ال Slider ، فتم اضافة الكود التالي لكل Slider كما يلي :

```
b=get(hObject,'Value');
global a;
writePWMVoltage( a , 'Pin' , 5-b );
```

يتم تغير الـ Pin لكل Slider اعتمادا على المدخل المرتبط به على لوحة الاردوينو

قم بحفظ و تشغيل الملف للتحكم بالـ RGB LED .

***يمكنك تنزيل واجهة المستخدم و الكود لهذا المشروع من هنا.**