ورقة بحثية مجلة مفتوحة الوصول

التحكم بمشاريع المتحكمات الصغرية عن طريق هواتف أندرويد باستخدام تطبيق MIT App Inventor

الدكتور فادي متوج

(كلية الهندسة، جامعة المنارة، البريد الإلكتروني: fadi.motawej@manara.edu.sy)

الملخص

يعد تطبيق MIT App Inventor بمثابة مقدمة مبتكرة للمبتدئين في مجال البرمجة وإنشـــاء التطبيقات للهواتف الذكية التي تعمل بنظام أندرويد، حيث يعمل على تحويل لغة البرمجة المعقدة المعتمدة على كتابة أكواد نصية إلى طريقة برمجة رسومية سهلة تعتمد على بلوكات برمجية قابلة للسحب والإفلات(drag and drop). تمنح الواجهة الرسومية البسيطة حتى المبتدئ عديم الخبرة بالبرمجة النصية القدرة على إنشاء تطبيق أندرويد كامل الوظائف خلال زمن قصير جداً. نشرح في هذه المقالة كيفية التحكم بالمشاريع المعتمدة على المتحكمات الصغرية من خلال تطبيق على الهاتف الذكي تم تصميمه و برمجته باستخدام MIT App Inventor .

كلمات مفتاحية _ Arduino ، MIT App Inventor ، بلوتوث.

ABSTRACT

The MIT App Inventor application is an innovative introduction for beginners to the field of programming and creating applications for smartphones running the Android system, as it works to transform a complex programming language that relies on writing **text-based programs** into an easy graphical programming method that relies on drag-and-drop programming blocks. . The simple graphical user interface (GUI) gives even a beginner with no scripting experience the ability to create a fully functional Android application in a very short time. In this article, we explain how to control microcontroller-based projects through a smartphone application that was designed and programmed using MIT App Inventor.

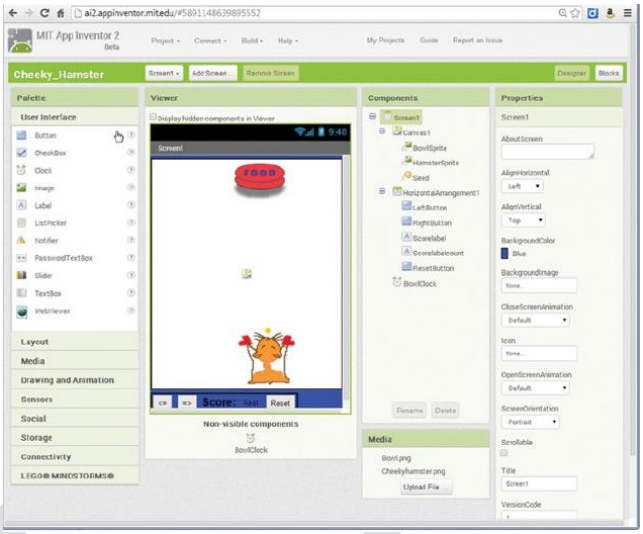
Keywords _ Arduino, MIT App Inventor, Bluetooth.

I. مقدمة

من المعروف أنَّ تطبيقات الهواتف الذكية يتم إنشاؤها عادةً بلغات برمجة عالية المستوى مثل Java ، و هذا يتطلب معرفة الكثير عن البرمجة بهذه اللغات. بالمقابل، إن تطبيق MIT App Inventor هو عبارة عن لغة برمجة يمكن استخدامها عبر متصفح الإنترنت لتصميم وإنشاء تطبيقات لهواتف Android دون الحاجة إلى تنصيب أي برنامج على جهاز الكمبيوتر. يتميز هذا التطبيق ببيئة برمجة رسومية، مما يعني أننا لسنا مضطرين إلى كتابة تعليمات برمجية معقدة. وبدلاً من ذلك، يمكننا سحب وإفلات الكائنات على الشاشة وتوصيل كتل التعليمات البرمجية معاً تماماً مثل أحجية الصور المقطعة [1] ، [2] .

II. تصميم شاشة التطبيق

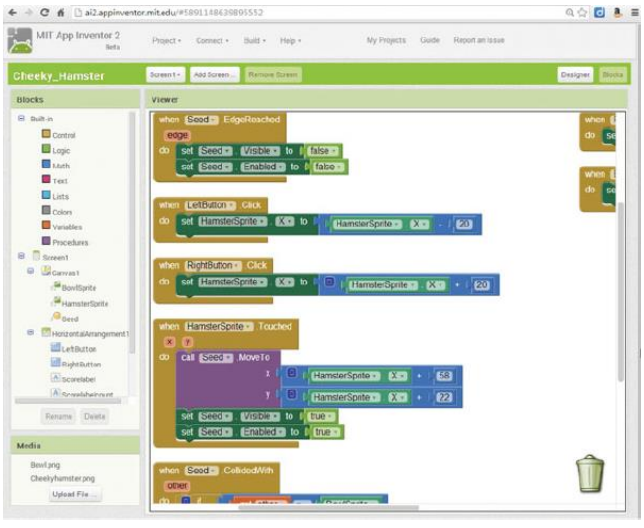
نافذة التصميم، و التي يطلق عليها "Designer" (الشكل1) هي المكان الذي يتم فيه تحديد شكل ومظهر التطبيق المطلوب إنشاؤه، وتحديد الوظائف التي يجب أن يتمتع بها. يمكن اختيار عناصر لواجهة المستخدم مثل الأزرار(Buttons) والصور(Images) ومربعات النص (Text boxes) والوظائف مثل تحويل النص إلى كلام (Text-to Speech) واستخدام العديد من الحساسات الموجودة في الهاتف.



الشكل 1 . نافذة Designer

III. برمجة وظائف التطبيق

بعد تصميم واجهة التطبيق المراد، يمكن استخدام محرر خاص يُعرف باسم Blocks لبرمجة الوظائف التي ينفذها التطبيق و إخباره بما يجب فعله. باستخدام أداة Blocks، لايتعين على المستخدم تعلُّم أو الإلمام بلغة مثل Java. وبدلاً من ذلك، يمكن تجميع البلوكات البرمجية بشكل رسومي لإنشاء وظائف التطبيق. يوضح الشكل 2 نافذة Blocks.



الشكل 2 . نافذة Blocks

IV. اختبار البرنامج

إذا لم يتوفر لدينا هاتف Android لاختبار التطبيق، فإن App Inventor يوفر محاكي Android يعمل على جهاز الكمبيوتر الخاص بنا. المحاكي، الموضح في الشكل 3، هو عبارة عن هاتف Android افتراضي محاكي لهاتف حقيقي. عندما نستخدم App Inventor لتطوير تطبيق ما، يظهر التطبيق ويعمل على شاشة المحاكي. يمكن التفاعل مع المحاكي بالعديد من الطرق نفسها التي نتفاعل بها مع الهاتف الذكي الفعلي. على الرغم من أنَّ المحاكي محدود (على سبيل المثال، لايحتوي على حساس GPS لتحديد موقعه، ولايمكنه إجراء مكالمات هاتفية)، فإنه يوفر العديد من الميزات الأساسية للهاتف الذكي الحقيقي.

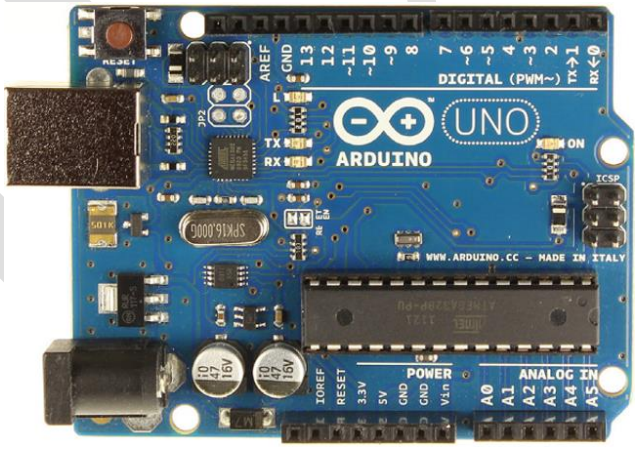
الشكل 3 . محاكي هاتف الأندرويد

نحتاج إلى ثلاثة أشياء لاستخدام App Inventor:

- جهاز كمبيوتر يعمل بنظام Windows، أو Macintosh، أو GNU/Linux.
- اتصال بالإنترنت ومتصفح Chrome أو Firefox أو Safari.
- حساب Google مجاني.

V. منصة ARDUINO

Arduino عبارة عن منصة متحكمات صغرية مفتوحة المصدر تعتمد على عناصر وبرامج سهلة الاستخدام. لوحات Arduino قادرة على قراءة إشارات الدخل (ضوء، أو ضغط على سويتش، أو حرارة) وتحويلها إلى مخرجات (تشغيل محرك، أو تشغيل LED، أو نشر شيء ما عبر الإنترنت). يمكن إخبار بورد Arduino بما يجب فعله عن طريق إرسال مجموعة من التعليمات إلى المتحكم الصغري Microcontroller الموجود على اللوحة. للقيام بذلك، يمكن استخدام لغة برمجة Arduino، وبيئة التطوير Arduino (IDE) (الشكل 4).



الشكل 4 . منصة Arduino

منذ ظهوره، كان Arduino بمثابة الدماغ لآلاف المشاريع، بدءاً من مشاريع الطلاب البسيطة وحتى الروبوتات والمركبات الفضائية. لقد اجتمع مجتمع عالمي من المبدعين (الطلاب والهواة والفنانين والمبرمجين والمهنيين) حول هذه المنصة مفتوحة المصدر، وقد أضافت مساهماتهم قدراً كبيراً من المعرفة التي يمكن الوصول إليها والتي يمكن أن تكون ذات فائدة كبيرة للمبتدئين الهواة والخبراء على حد سواء.

ظهر بورد Arduino كأداة سهلة لبناء النماذج الأولية السريعة، تستهدف الطلاب الذين ليس لديهم خلفية قوية في الإلكترونيات والبرمجة. بمجرد انتشارها ووصولها إلى عدد مستخدمين كبير ومجتمع أوسع، بدأت لوحة Arduino في التغيير للتكيف مع الاحتياجات والتحديات الجديدة، وتميزت خياراتها من لوحات بسيطة 8 بت إلى لوحات خاصة بتطبيقات إنترنت الأشياء، والطباعة ثلاثية الأبعاد، والنظم المدمجة.

VI. إنشاء تطبيق ANDROID للتحكم في تشغيل LED

سنوضح فيما يلي خطوة بخطوة كيفية إنشاء تطبيق Android يتصل بـ Arduino. تحتوي واجهة التطبيق زرَّين عند الضغط على أحدهما يضيئ الـ LED بينما يؤدي الضغط على الزر الآخر إلى إطفائه.

1- نذهب إلى الموقع الإلكتروني الخاص بالتطبيق [3] ، حيث عند الدخول لأول مرة سيُطلب منا تسجيل الدخول باستخدام معرِّف Gmail وكلمة المرور.

2- عند تسجيل الدخول إلى بيئة تطوير MIT App Inventor لأول مرة، لن تكون هناك أي مشاريع. ننقر فوق "start new project" لإنشاء تطبيق Android الأوّل الخاص بنا.

3- نقوم بتسمية مشروعنا الجديد بأي اسم نريده وننقر فوق "ok" للدخول إلى الشاشة الرئيسية (الشكل 5).

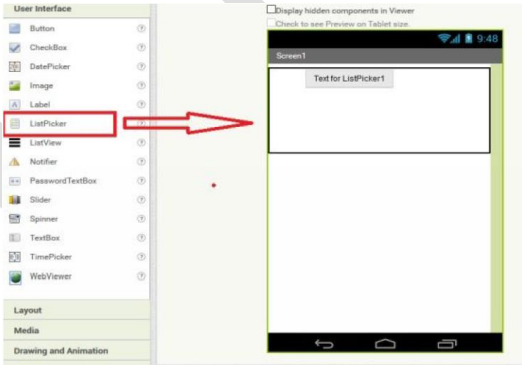
الشكل 5 . نافذة تسمية التطبيق

بعد تسمية التطبيق، سوف ننتقل إلى النافذة الرئيسية لإنشاء تطبيقنا، وهي نافذة "Designer"، حيث نحتاج إلى تصميم واجهة المستخدم من خلال إدراج الأزرار، والعناصر المختلفة المرغوبة. على الجانب الأيسر لنافذة Designer ، توجد خزانة

الأدوات "palette" التي تحتوي على عدة صـــناديق من الأدوات كالوسائط والرسم و الحساسات و الاتصال وما إلى ذلك.

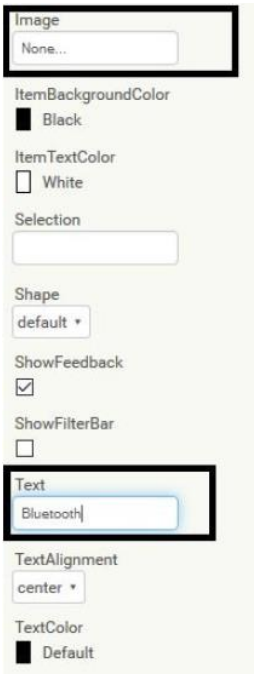
لإنشـــاء واجهة المســـتخدم الخاصـــة بتطبيقنا، ســـنقوم بإدراج العناصر التالية:

1- الأداة Listpicker من صندوق الأدوات User Interface ونضعها في الشاشة الرئيسية. تستخدم هذه الأداة للاتصال بموديول الـــ Bluetooth المتصل مع بورد أردوينو. عند النقر فوق هذا المكون، ســـيتم عرض أجهزة Bluetooth الموجودة بالجوار (الشكل 6).



الشكل 6 . إدراج الأداة ListPicker

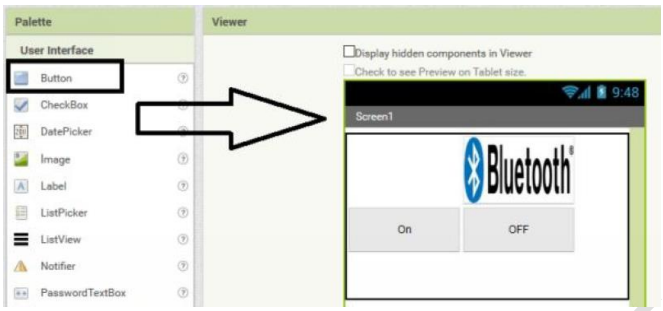
ثم نحدد ListPicker من قائمة المكونات و ننتقل إلى



الشكل 7. خصائص الأداة ListPicker

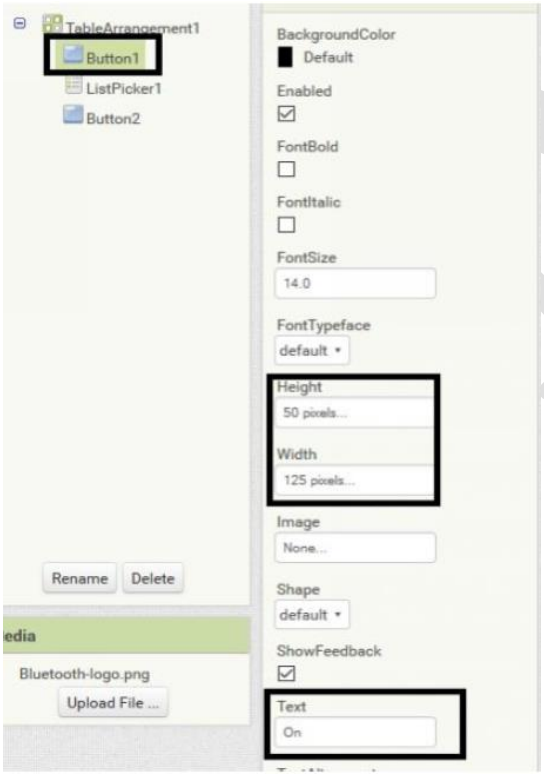
خصائصها حيث نقوم بتغيير النص إلى Bluetooth ويمكننا أيضًا تحميل ملف صورة إلى ListPicke (الشكل 7).

2- الأداة Button حيث نضعه في الصف الثاني من الشاشة الرئيسية ونضيف Button آخر إلى جانبه في نفس الصف (الشكل 8).



الشكل 8 . إدراج الأداة Button

ننقر على الـ Button الأول وننتقل إلى خصائصه ونقوم بتغيير اسمه إلى ON، ونقوم بنفس الشيء بالنسبة للـ Button الثاني عن طريق تغيير اسمه إلى OFF (الشكل 9).



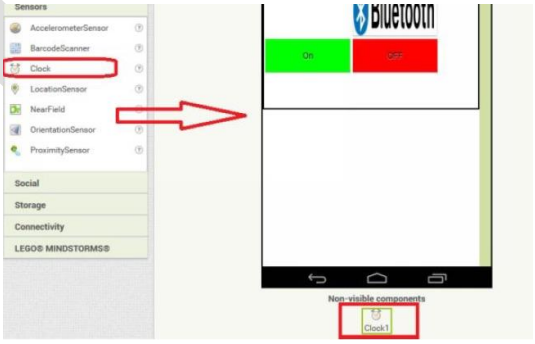
الشكل 9 . تغيير خصائص الأداة Button

كما يمكن تغيير لون الـ Button عن طريق تغيير لون الخلفية من قائمة خصائص الـ Button (الشكل 10).

3- نحتاج أيضاً إلى بعض المكونات غير المرئية لإنشاء هذا التطبيق، المكون الأول هو الساعة clock التي يمكن اختيارها ووضعها من صندوق الأدوات sensors (الشكل 11).

الشكل 10 . تغيير لون الأداة Button

يستخدم هذا المكون الساعة الداخلية للهاتف. يمكنه تفعيل مؤقت زمني على فترات زمنية محددة بانتظام وإجراء حسابات الوقت والتحويلات المختلفة. نحن نستخدم clock لجعل الأنشطة المختلفة تحدث في فترة زمنية محددة مسبقاً.



الشكل 11 . إدراج المكون clock

4- الأداة الأخيرة التي سنقوم بإدراجها هي BluetoothClient من صندوق الأدوات Connectivity ووضعها داخل الشاشة الرئيسية (الشكل 12).

الشكل 12 . إدراج المكون BluetoothClient

يسمح المكون BluetoothClient لنا بالاتصال بجهاز آخر ويجعلنا نرسل ونستقبل البيانات بين أجهزة Bluetooth .

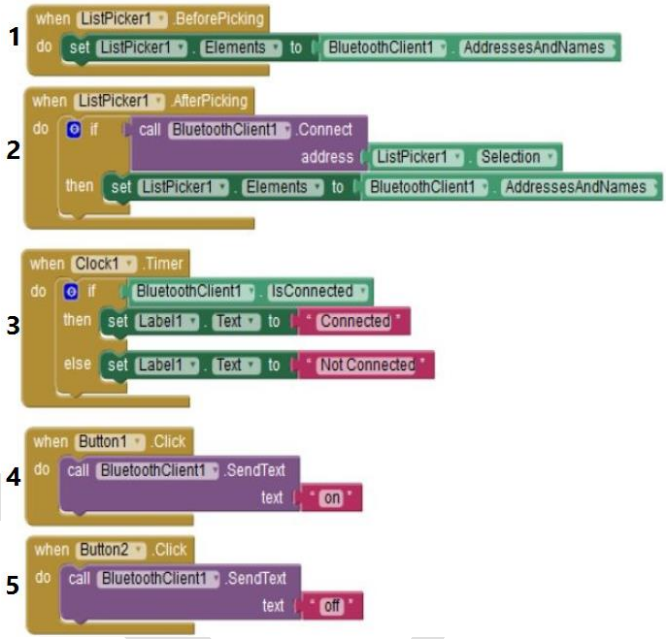
لقد أكملنا حتى الآن تصميم واجهة المستخدم للتطبيق المطلوب، والآن يتطلب هذا التصميم برنامجاً يخبر التطبيق بمايجب القيام به من وظائف معينة.

يتم الانتقال إلى مرحلة البرمجة من خلال النقر على الزر Blocks الموضح بالشكل 13. لإنشاء برنامج نقوم بربط كتل (بلوكات) برمجية مختلفة وهو أمر سهل كما ذكرنا سابقاً.

الشكل 13 . الانتقال إلى مرحلة البرمجة

يبين الشكل 14 الكتلة الكاملة للبرنامج الخاص بالتحكم في إضاءة و إطفاء LED متصل ببورد Arduino حيث تم ترقيم الكتل البرمجية و فيما يلي سنقوم بشرح وظيفة كل كتلة على

الترتيب:



الشكل 14 . الكتلة الكاملة للبرنامج

الكتلة 1 : توضح أنه عند النقر فوق ListPicker، يجب أن يتم عرض أجهزة Bluetooth المتوفرة والمتصلة بالفعل بجهازنا مسبقًا.

الكتلة 2 : توضح أنه عند اختيار جهاز Bluetooth من القائمة المتاحة، يجب استخدام الجهاز الذي تم اختياره كجهاز حالي وسيتم تخزين عنوان Bluetooth وسيتواصل التطبيق مع الجهاز المحدد.

الكتلة 3 : ستقوم هذه الكتلة بمراقبة حالة اتصال البلوتوث بين الطرفين حيث تظهر على الشاشة كلمة متصل في حال كان الاتصال قائماً ، بينما تظهر عبارة غير متصل في حال كان الاتصال مقطوعاً.

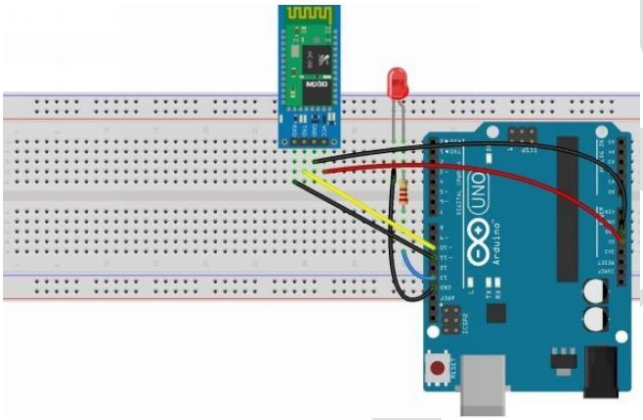
الكتلتين 4 و 5 : وظيفة هاتين الكتلتين هي إنشاء مهمة تحكم للزرين المدرجين في الشاشة الرئيسية. فعندما نضغط على الـ Button الأول، سيتم إرسال السلسلة المحرفية "on" إلى جهاز Bluetooth متصل بـ Arduino ، وبالنسبة للـ .Button الثاني، فعند الضغط عليه يتم إرسال السلسلة المحرفية "off".

عندما يستقبل Arduino إحدى هاتين السلسلتين، يتعين علينا حسب البرنامج الخاص ببورد Arduino بناءً على هذه السلسلة اتخاذ قرار إما بتشغيل أو إيقاف تشغيل الـ LED.

VII. توصيل الدارة

نحتاج إلى توصيل موديول البلوتوث HC-05 أو HC-06 أو أي طراز آخر إلى بورد Arduino كما هو موضح بالشكل 15 . ثم نقوم بتشـــغيل Bluetooth على الهاتف ونبحث عن الأجهزة الموجودة، حيث نتصـــل بجهاز Bluetooth الخاص بنا (HC-05 أو HC-06) باسـتخدام كلمة المرور الافتراضـية، "1234" أو "0000".

كما نقوم بتوصـــيل LED بالرجل 13 من Arduino. نقوم بتوصـــيل TX الخاص بالبلوتوث بالرجل 10وRX الخاص بالبلوتوث بالرجل 11 (الشكل15). يمكن أيضًا توصيله بالرجلين TX وRX المتاحين وهما 0 و1، مع التنويه أنه عند الاتصـــال بهذين المنفذين، لا يمكن توصيل Arduino بجهاز الكمبيوتر أو الكمبيوتر المحمول.



الشكل 13 . الدارة المستخدمة في المثال

VIII. برمجة الأردوينو

نبين فيما يلي الكود الخاص ببورد Arduino:

```
#include <SoftwareSerial.h>
SoftwareSerial BT(10, 11); //TX, RX respectively
String state; // string to store incoming message from bluetooth
void setup() {
BT.begin(9600);// bluetooth serial communication will take place on pin 10 and 11
Serial.begin(9600); // serial communication to check the data on serial monitor
pinMode(13, OUTPUT); // LED connected to 13th pin
}
//————————————————————//
void loop() {
while (BT.available()){ //Check if there is an available byte
delay(10); //Delay added to make thing stable
char c = BT.read(); //Conduct a serial read
state += c; //build the string- either “On” or “off”
}
if (state.length() > 0) {
Serial.println(state);
if(state == “on”) // if the received string is on, turn on led
{
digitalWrite(13, HIGH);
}
else if(state == “off”) // if the received string is off, turn off
{
digitalWrite(13, LOW);
}
state =””;}} //Reset the variable
// code ends here
```

IX. الاستنتاجات

تم في هذه المقالة شـــرح خطوات تصـــميم أي تطبيق لهواتف Android باســتخدام MIT App Inventor . كما تضــمنت الدراسـة تصـميم تطبيق للتحكم بإضـاءة و إطفاء LED متصـل مع بورد Arduino ، حيث اتضـــحت ســـهولة تطوير تطبيقات للهواتف الذكية باســتخدام MIT App Inventor بالمقارنة مع لغات البرمجة العالية المستوى.

المراجع:

[1]. P .Beer and C. Simmons *" Hello App Inventor! Android programming for kids and the rest of us"*, Manning Publications, 2015.

[2]. D. Wolber, H. Abelson, E. Spertus and L. Looney, *" App Inventor 2 Create Your Own Android Apps "*, O’Reilly, 2015.

[3]. (2024) The website. [online]. Availabla: *http://ai2.appinventor.mit.edu*