

## استخدام الواقع الافتراضي لمحاكاة أداء الحساسات

م. وسيم أحمد

كلية الهندسة ، جامعة المنارة

( البريد الإلكتروني: wassim\_eng@yahoo.ca )

## الملخص

تعتبر نظم التحكم بنماذجها المختلفة وطرق ربطها مع أنواع مختلفة من المتحكمات أحد المجالات الهامة في الوقت الحاضر والتي تهدف إلى أتمتة العديد من النظم التي نستخدمها في حياتنا اليومية، كما يعتبر الربط مع النظم الحاسوبية للقيام بالمراقبة واستخلاص المعرفة واحدة من العمليات التي يمكن أن تساعد المشغل على مراقبة وتحديد رد الفعل المقدم من النظام ككل حتى يوافي متطلبات المستخدمين .

يسمح استخدام برمجيات المحاكاة بالحصول على تعريف بأداء نظم الحساسات ضمن بيئة قريبة من الواقع الافتراضي، حيث تتضمن بعض النظم محاكاة لسير عمل الدارة ومقارنة الخرج الحقيقي مع الخرج المتوقع، في حين تسمح برمجيات أخرى بمحاكاة تنفيذ النظام على أرض الواقع وكيفية عمله بصورة تفاعلية .

تهدف الدراسة إلى إنشاء نظام تقاعلي لمحاكاة عمل منظومة حساس فوق صوتي متصل مع متحكم أردوينو ومن ثم إعداد محاكاة افتراضية لهذه الدارة باستخدام لغة بايثون ومكتبة الواقع الافتراضي الخاصة بها.

**كلمات مفتاحية - متحكم أردوينو ، واقع افتراضي .**

نماذجها والتي تعطى اسم Modules مثل نماذج الحساسات الفوق صوتية ونماذج قيادة والتحكم بالمحركات ونماذج الاتصال مثل نماذج البلوتوث ونماذج Wi-fi .

## . II . الرؤية الحاسوبية:

تعرف الرؤية الحاسوبية على أنها إحدى مجالات علم الحاسوب، تهدف إلى بناء تطبيقات ذكية قادرة على فهم محتوى الصور كما يفهمها الإنسان ويمكن عن طريقها بناء تطبيقات واقع افتراضي .

حيث من الممكن أن تأخذ بيانات الصور التي نقوم بدراستها عدة أشكال كالصور المتعاقبة (ملفات الفيديو) والمشاهد من عدة كاميرات، تمثل كذلك بالبيانات ذات العدة أبعاد المأخوذة من جهاز تصوير طبي بهدف رسم الشكل الفراغي على سبيل المثال.

## . I . مقدمة

إن إحدى المشاكل الرئيسية التي نعاني منها التعامل مع لوحة الأردوينو التقيد دائماً بعدد من الحدود التشغيلية، حيث أنه لا يمكن - على سبيل المثال - قراءة أي بيانات منها إلا عن طريق شاشة ال serial monitor أو عرضها من خلال شاشة عرض كريستالية، كما تكون البيانات المقروءة دائماً في صورة نصية و بدائية للغاية، لذلك يكون من الأفضل لنا أن نبنى بيئة رسومية افتراضية للدارات ومحاكاة عمل هذه الدارات فيزيائياً بحيث نرفع من فعالية تقييم الدارات التي يتم بناءها باستخدام لوحة الأردوينو.

تعتبر لوحات الأردوينو من لوحات التحكم ذات الكلفة المنخفضة والأداء العالي بالإضافة إلى دمجها مع العديد من النظم الطرفية والتي تم تصميمها بصورة رئيسية للتوافق مع اللوحة بأغلب

### III. لغة البايثون PYTHON

لتكون مشابهة لبيئة OpenGL من ناحية بناء العناصر ضمن نوافذ برمجية ومن ثم تطبيق مجموعة من العمليات الأساسية عليها كالانزياح والدوران والقص وغيرها.

تعمل هذه المكتبة بشكل فعال مع جميع إصدارات لغة بايثون إلا أنها تعاني من بساطة النماذج التي يمكن لنا أن نقوم ببنائها فيها وإن كانت مناسبة لعمليات المحاكاة البسيطة.

#### C. لوحة الأردوينو UNO:

وهو اللوح الأكثر استخداماً و شيوعاً في بناء المشاريع الالكترونية البسيطة، ويعود الامر الى سهولة استخدامه خاصة للمبتدئين بالإضافة إلى فعاليته العالية في قيادة دارات التحكم. يستخدم في بناء هذا اللوح متحكم من شركه اتميل من نموذج ATmega328 حيث توفر هذه اللوحة المتكاملة منافذ لتوصيل المكونات الالكترونية كالمجسات الى المتحكم مباشرة عن طريق 14 (مدخل | مخرج) من النوع الرقمي ويمكن استخدام ستة مداخل منها للحصول على نظام التحكم بعرض النبضة PWM (Pulse-Width modulation)، كما تحتوي الدارة بالإضافة لذلك مولد تردد ساعة كريستالي Oscillator بتردد 16MHz، بالإضافة الى مدخل USB من أجل التواصل مع الحاسب، و هناك مدخل منفصل للطاقة. يوضح الشكل (1) التالي توزيع منافذ اللوحة.

تتنوع الذاكرة المستخدمة في المتحكم طبقاً لنوع استخدامها والمهمة المسندة لها وتقسم الى:

Boot loader وهو البرنامج المسؤول عن كيفية فهم الدارة للغة البرمجية المعتمدة من لوحة الأردوينو ArduinoC وهي بحجم 0.5Kbyte.

SRAM تعتبر هذه الذاكرة بأنها الذاكرة المستخدمة في تسجيل المتغيرات بصورة مؤقتة بحجم 2KB.

Flash Disk مساحة تخزينية تستخدم في حفظ البرنامج الذي يستعمل لتشغيل المتحكم وهي بحجم 32KB.

كانت اللغة البرمجية النصية scripting language الأفضل في العالم دون منازع هي لغة الجمل بيرل Perl ، حتى جاءت لغة بايثون Python والتي تعرف بأنها اللغة النصية الأكثر فعالية واستخداماً في الوقت الحالي. تتخطى لغة بايثون كل عيوب لغة بيرل Perl التي توصف بأنها لغة قذرة Dirty Language ، ففي لغة بيرل من السهل كتابة برنامج لكن قد يكون شبه مستحيل قراءته أو تعديله أو إصلاحه. بل إن هناك تعارف معتمد بإنشاء برامج تحل مشاكل معقدة بسطر واحد بلغة بيرل. أما لغة بايثون فإنها تحمي المبرمج من نفسه، حيث تلزمه بالوضوح. لكن هذا ينفر المبرمجين الجدد منها في البداية.

جاءت لغة بايثون من مراكز الأبحاث مرفقة بعدد ضخم من الحزم الإضافية المتخصصة بالحوسبة الفائقة Super computing والحوسبة المتوازية Parallel computing ويظهر هذا واضحاً من خلال النظر في مالكي حقوق النسخ عليها وهم Stichting Mathematics' Centrum ثم Corporation for National Research Initiatives ثم Python Software Foundation وقد غدت هذه اللغة المفضلة عند علماء الرياضيات والمهندسين والباحثين وقد حلت محل لغة FORTRAN

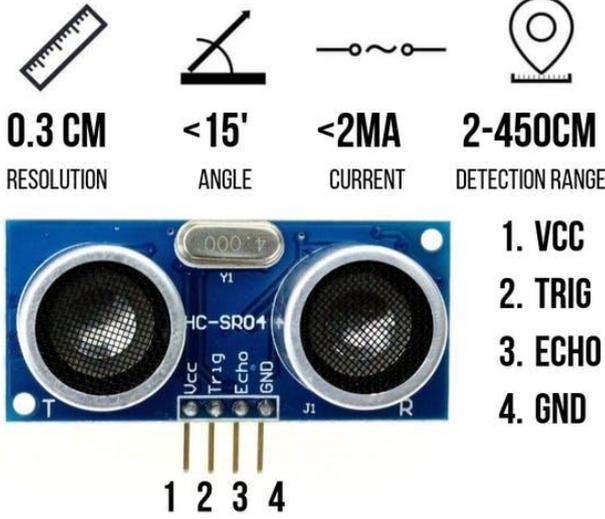
#### A. بايثون سيريال pyserial:

وهي مجموعة من المكتبات والحزم البرمجية والتي تسمح لبيئة بايثون أن تتعامل مع الوسط الخارجي من خلال منفذ الاتصال التسلسلي، تعتبر هذه المكتبات مشابهة لمكتبات الأردوينو والتي تعمل مع بايثون تتيح الفرصة للتعامل مع شاشة الاتصال التسلسلي الخاصة بالأردوينو.

#### B. Vpython:

وهي مكتبة برمجية تتيح تصميم واقع افتراضي موازي ثلاثي الأبعاد لمحاكاة مستشعرات الأردوينو، تم إنشاء هذه المكتبة

إشارة الصدى. من أجل أن يعمل هذا الحساس بصورة صحيحة يجب أن يتم توليد نبضة على منفذ Trigger في لوحة الأردوينو والذي يتم تعريفه كمرجع ضمنها وانتظار إشارة الصدى الواردة إلى اللوح والتي تعتبر كإشارة دخل.



الشكل 2 . مودل الحساس الفوق صوتي

يتم تلقي إشارة أمر من الأردوينو بعرض 10micro second ليقيم بإطلاق ثمانية نبضات فوق صوتية. يقوم الحساس بإرسال موجات فوق صوتية و عند اصطدامها بحائل ترتد مرة أخرى فيمكن للأردوينو حساب الوقت الذي استغرقته الموجات للذهاب و العودة لذلك لابد من القسمة على 2 لأننا نريد حساب المسافة و هذا يعتمد على وقت الذهاب فقط.

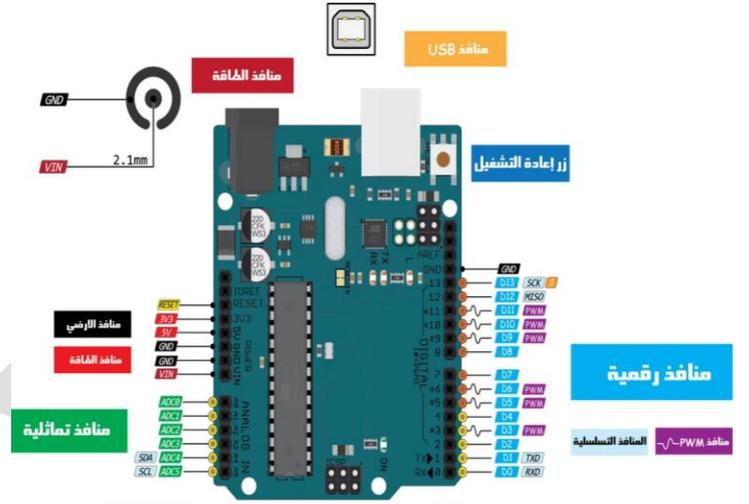
سرعه الصوت في الهواء في الحالة العادية تقريبا 340 متر لكل ثانية فيكون الوقت المستهلك لعبور 1 سم هو:

$$\frac{1}{340} \times 10^6$$

تم التقسيم على 100 للتحويل من متر الى سم، و  $6^{10}$  للتحويل من ثانية الى ميكروثانية.

يوضح الشكل (3) التالي توليد نموذج الإشارة التي يتم استخدامها في الحساس الفوق صوتي.

EEPROM وهي الذاكرة المسؤولة عن تسجيل بعض المتغيرات بصورة دائمة داخل المتحكم و تظل محتفظة بقيمتها حتى بعد فصل الكهرباء بحجم 1Kbyte.



الشكل 1 . لوحة الأردوينو ومنافذها الرئيسية

يمكن تزويد لوحة UNO بالطاقة إما من خلال منفذ الـ USB فقط أو عن طريق استخدام مصدر خارجي كمحول كهربائي أو بطارية 9 فولت أو 4 بطاريات 1.5 فولت. عند استخدام UNO ينصح باستعمال جهد يتراوح بين 6-12 فولت، رغم قدرتها على تحمل جهد يتراوح بين الـ 6-20 فولت، ولكن يجب الحذر حيث ان امداد القطعة بجهد يقل عن 6 فولت يمنع المخارج والتي تقدم بشكل طبيعي جهد بمقدار 5 فولت من تأمين الجهد المطلوب مما يؤدي الى عدم استقرار الدارة. من ناحية اخرى فإن تزويد الدارة بجهد أعلى من 12 فولت يؤثر على عنصر تنظيم الجهد voltage regulator ويؤدي الى ارتفاع درجة حرارته مما يؤدي الى تلف الدارة لذلك فإن مجال الجهد الافضل يتراوح من 6 الى 12 فولت.

#### D. الدارة التطبيقية:

يوضح الشكل (2) التالي توصيل الحساس الفوق صوتي مع لوحة الأردوينو، حيث يتضمن الحساس الفوق صوتي على أربعة منافذ رئيسية وهي منفذي التغذية والتأريض بالإضافة إلى منافذ Trigger لتوليد النبضات المرسله و Echo المخصص للتبنيه عن

المقابل في Arduino	المنفذ Ultrasonic
13	Trigger
11	Echo

```

تعريف متحول لتوصيل المنفذ 13 //
int trigPin=13;

تعريف متحول لتوصيل منفذ الصدى //
int echoPin=11;

تعريف متحول لتخزين زمن الإشارة //
float pingTime;

المسافة للهدف بالانث //
float targetDistance;

سرعة الصوت بالميل بالساعة //
float speedOfSound=776.5;

عند درجة حرارة الغرفة
void setup() {
// تم في هذا القسم تعريف الاتصال بالمنفذ التسلسلي وبمعدل
بود بمقدار 9600 بت في الثانية في حين تقوم التعليمات
pinMode بتحديد البوابة ونمط العمل الخاص بها كدخول أو
خرج.

```

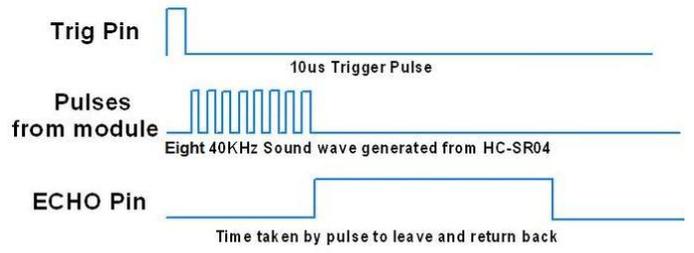
```

Serial.begin(9600);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
}

void loop() {
// put your main code here, to run repeatedly:
سيتم في المرحلة التالية تنفيذ البرنامج التالي بهدف إرسال
نبضة التردد الفوق صوتي كما في التسلسل الموضح.
digitalWrite(trigPin, LOW); //Set trigger pin low
delayMicroseconds(2000); //Let signal settle
digitalWrite(trigPin, HIGH); //Set trigPin high
delayMicroseconds(15); //Delay in high state
digitalWrite(trigPin, LOW); //ping has now been sent
delayMicroseconds(10); //Delay in low state
سنقوم بعد ذلك لوحة الاردوينو بالانتظار حتى ورود إشارة
عالي من لوح التردد الفوق صوتي (والتي تعني ارتداد ووصول
إشارة الصدى) حيث سيتم تحويلها إلى ثانية ومن ثم ضبطها
للتوافق مع السرعة المحددة بتعريف مسبق
pingTime = pulseIn(echoPin, HIGH); //pingTime is
presented in microceconds

```

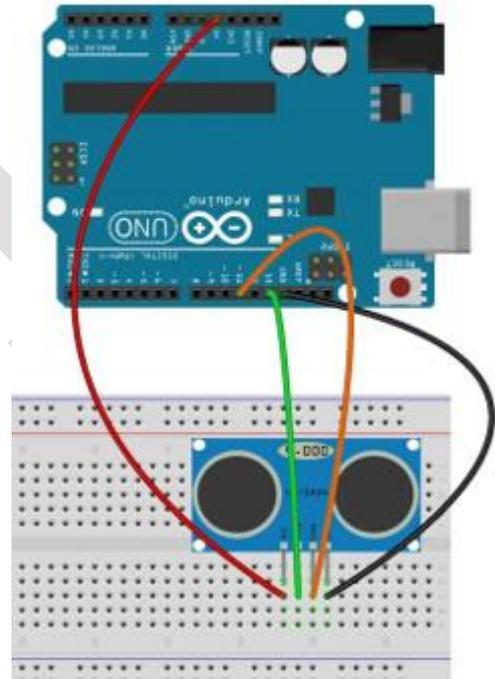
### Ultrasonic HC-SR04 module Timing Diagram



الشكل 3 . نموذج توليد النبضات المطلوب لعمل الحساس الفوق صوتي HC-SR04

### E. الدارة التطبيقية:

يوضح الشكل (4) التالي تركيب الدارة العملية بين لوحة الأردوينو والحساس الفوق صوتي:



الشكل 4 . تركيب الدارة التطبيقية للحساس الفوق صوتي

يوضح الإجراء البرمجي التالي عملية التحكم بالحساس الفوق صوتي حيث تم اعتماد التوصيلات التالية:

من أجل إعداد العناصر ضمن المشهد كما هو موضح في الشكل (3) حيث نحتاج لمستطيل واسطوانتين من أجل نمذجة شكل الحساس الفوق صوتي واسطوانة لتحديد المسار بالإضافة للوح يمثل الهدف الذي تتعكس الغشارة عنه.

على سبيل المثال ومن أجل رسم أسطوانة يتم استخدام الأمر cylinder ونضع ضمنها خصائص هذه الأسطوانة من ناحية الطول ونصف القطر والموضع ويتم رسم اللائحة النصية باستخدام label في حين يتم رسم المربعات باستخدام الأمر Box. L=0.1

```
rod = cylinder(length=6 ,color=color.yellow, radius=.1,
pos=(-3,-2,0)) # Draw cylinder
```

```
lengthLabel = label(text = 'Target Distance is:',
pos=(0,2,0), height=20, box= false) # Add label
```

```
target = box (color=color.red,length=L, width=3,
height=3,pos =(3,-.5,0)) # Draw object
```

# Ultrasonic Sensor draw

```
sensorBox = box (color=color.green,length=L, width=2,
height=1,pos = (-3.5,-1.5,0)) # Draw Board
```

```
sensorc1 = cylinder(length=0.5 ,color=color.white,
radius=.3, pos=(-3.5,-1.5,.5)) # Draw cylinder
```

```
sensorc1 = cylinder(length=0.5 ,color=color.white,
radius=.3, pos=(-3.5,-1.5,-.5)) # Draw cylinder
```

يتم بعد ذلك تلقي بيانات النافذة التسلسلية القادمة من لوحة الأردوينو والتي تقوم بحساب بعد الحاجز عن الحساس الفوق صوتي ومن ثم تغيير موقع الحساس في الواقع الافتراضي اعتماداً على التعليمات vector.

```
while (1==1): #loop while the sensor is reading
```

```
rate(25)
```

```
while (ArduinoSerialData.inWaiting()==0):
```

```
pass
```

```
if(ArduinoSerialData.inWaiting(>0):
```

```
myData= ArduinoSerialData.readline()
```

```
distance= float(myData) #convert to floating
```

```
number
```

```
print distance
```

```
Mylabel = 'Target Distance is:' + distance
```

```
lengthLabel.text = Mylabel
```

```
pingTime=pingTime/1000000; //convert pingTime to
seconds by dividing by 1000000 (microseconds in a
second)
```

```
pingTime=pingTime/3600; //convert pingtime to hourse
by dividing by 3600 (seconds in an hour)
```

```
targetDistance= speedOfSound * pingTime; //This will
be in miles, since speed of sound was miles per hour
```

```
targetDistance=targetDistance/2; //Remember ping
travels to target and back from target, so you must divide
by 2 for actual target distance.
```

```
targetDistance= targetDistance*63360; //Convert miles
to inches by multipling by 63360 (inches per mile)
```

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

```
delay(100); //delay tenth of a second to slow things
down a little.
```

```
}
```

#### F. تجهيز الواقع الافتراضي باستخدام بايثون:

من أجل إنشاء الواقع الافتراضي باستخدام بايثون فقد ذكرنا مسبقاً، يجب أن نقوم بتحميل مكتبة VPython وهي المكتبة التي تشابه مكتبات OpenGL من ناحية بناء العناصر.

يتم في البداية استدعاء المكتبات في البايثون باستخدام التعليمات import حيث تقوم المكتبة serial بتأمين التوصيل مع الوسط الخارجي في حين تقوم Visual بإضافة المكتبات الافتراضية.

```
import serial
```

```
from visual import * # Add All visual libraries
```

يتم بعد ذلك وكما في مكتبات OpenGL بإعداد مشهد على شكل نافذة باستخدام التعليمات display وتحديد خيارات حجم هذه النافذة.

```
MyScene = display (title= 'Arduino VR')
```

```
MyScene.width = 800 # Screen width
```

```
MyScene.height = 800 # Screen height
```

عند توصيل لوح الأردوينو مع الحاسب فإنه سيتم عنونته كأحد المنافذ المتاحة في جهاز الحاسب، في حالتنا فإن المنفذ هو com4 ومعدل سرعة البود المقدم والمتاح.

```
ArduinoSerialData = serial.Serial('com4', 9600)
```

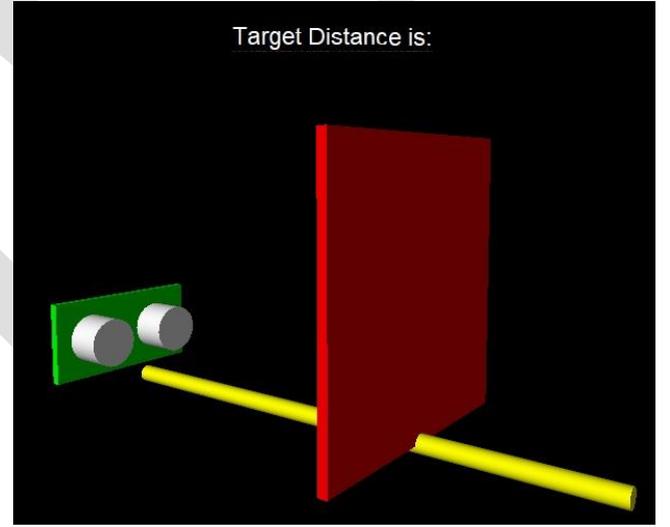
تمكنت الدارة من قياس المسافات بصورة صحيحة وتحريك العنصر ضمن واجهة الواقع الافتراضي بصورة قريبة لما يحدث في العالم الواقعي.

### المراجع:

- [1]. (2021) Arduino Projecthub website. [online]. Available: <https://create.arduino.cc/projecthub/abdularbi17/ultrasonic-sensor-hc-sr04-with-arduino-tutorial-327ff6>
- [2]. (2021) Vpython website. [online]. Available: <https://vpython.org/>
- [3]. (2021) SIMPLE VIRTUAL WORLD USING ULTRASONIC SENSOR. [online]. Available: <https://www.instructables.com/SIMPLE-VIRTUAL-WORLD-USING-ULTRASONIC-SENSOR/>
- [4]. S. Linner, M. Lischewski, M. Richerzhagen, "PYTHON Introduction to the Basics" Forschungszentrum Jülich, 2021.
- [5]. Arduino UNO, Product Reference Manual, SKU: A000066, Modified: 25/02/2022

```
target.pos = vector(-3+distance,-.5,0) # move the
target depend on real object
rod.lenght=distance # dynamic change lenght with
ultrasonic
```

قد تكون الرسومات بالألوان الكاملة. سيتم الاحتفاظ بجميع الألوان على القرص المضغوط. يجب ألا تستخدم الرسومات أنماط تعبئة نقطية لأنه قد لا يتم إعادة إنتاجها بشكل صحيح. يُرجى استخدام ألوان SOLID FILL فقط والتي تتميز بكونها تتباين جيدًا على الشاشة وعلى نسخة مطبوعة بالأبيض والأسود ، كما هو موضح في الشكل(5).



الشكل 5. نموذج الواقع الافتراضي الناتج

### IV. الاستنتاجات

يتميز الواقع الافتراضي بالقدرة العالية على محاكاة عمل نظم الحساسات مما يسمح للطلاب ببناء صورة واقعية عن عمل الدارات, كما تتميز البايثون بقدرتها العالية كبيئة تفاعلية تتضمن العديد من المكتبات الفعالة والقادرة على التواصل مع النظم البرمجية والفيزيائية المختلفة.