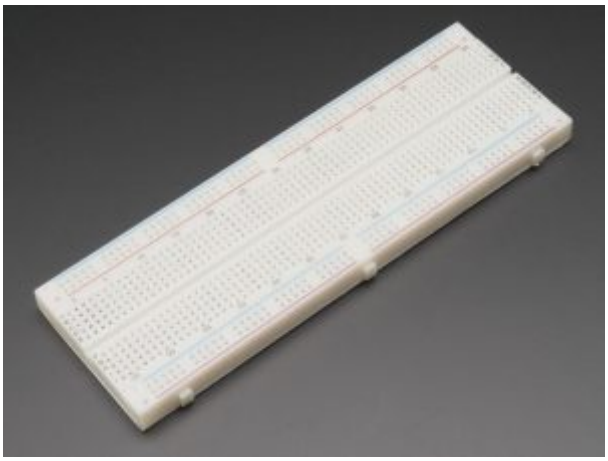
التحكم بأصوات تطبيقات الحاسوب باستخدام بالأردوينو

مقدمة

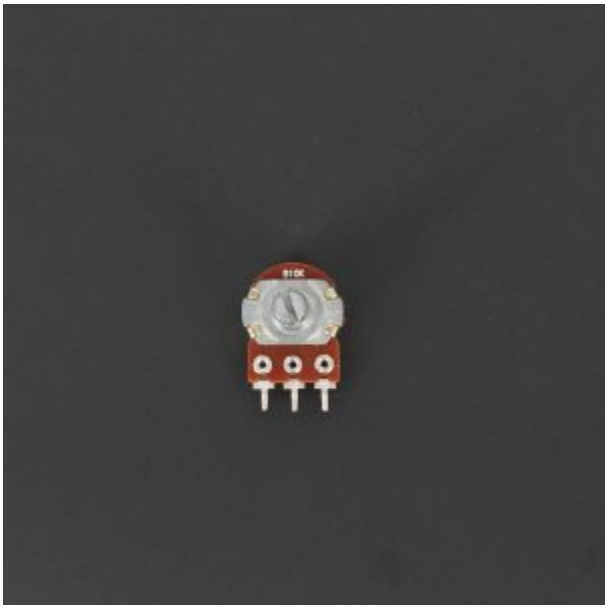
سنتعلم في هذا الدرس ضبط مستويات الصوت لمتصفح الإنترنت ومشغل الفيديو والمحادثات الصوتية وغيرها من البرامج على جهازك ، وذلك باستخدام اردوينو والمقاومة المتغيرة، دون الحاجة إلى ايقاف ما تقوم بالعمل عليه.

المواد و الأدوات

X1 اردوينو

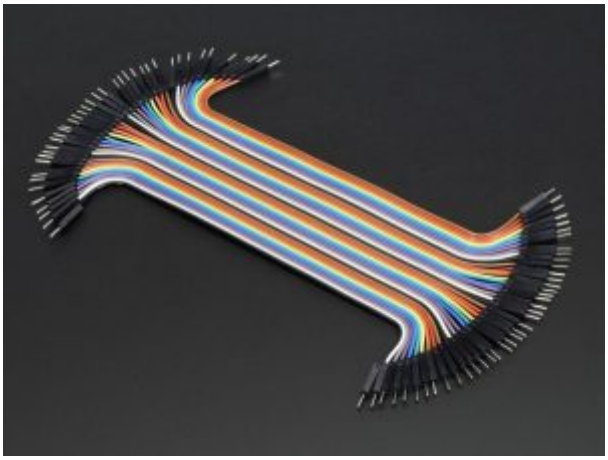


X1 لوحة تجارب



X5 مقاومة متغيرة

X1 سلك أردوينو

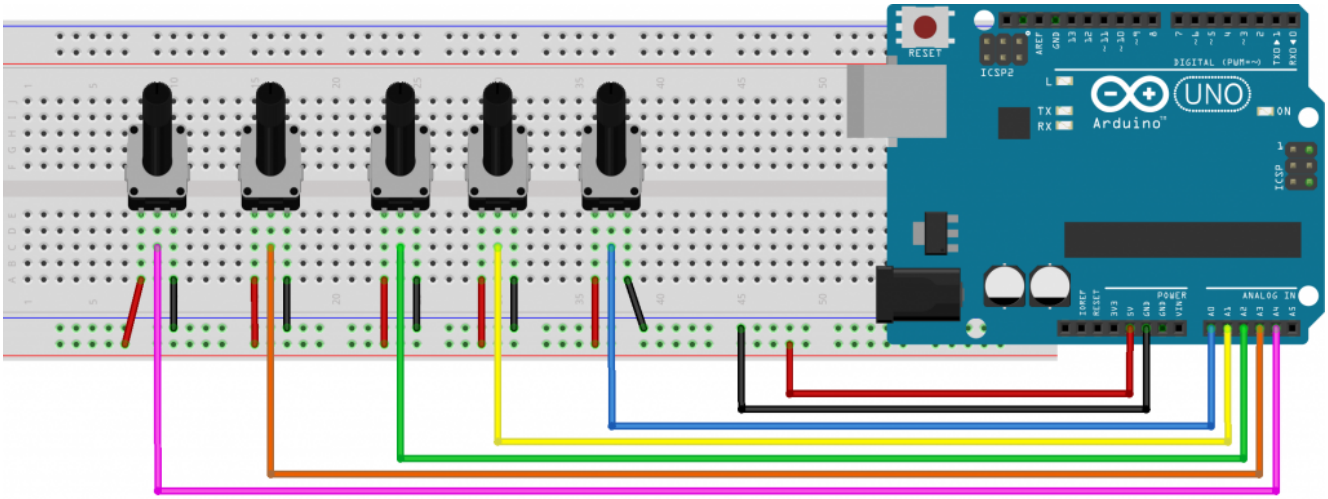


مجموعة أسلاك توصيل (ذكر/ذكر)

X1 خيوط الطابعة ثلاثية الأبعاد نوع (PLA) (اختياري)

X1 طابعة ثلاثية الأبعاد (اختياري)

توصيل الدائرة

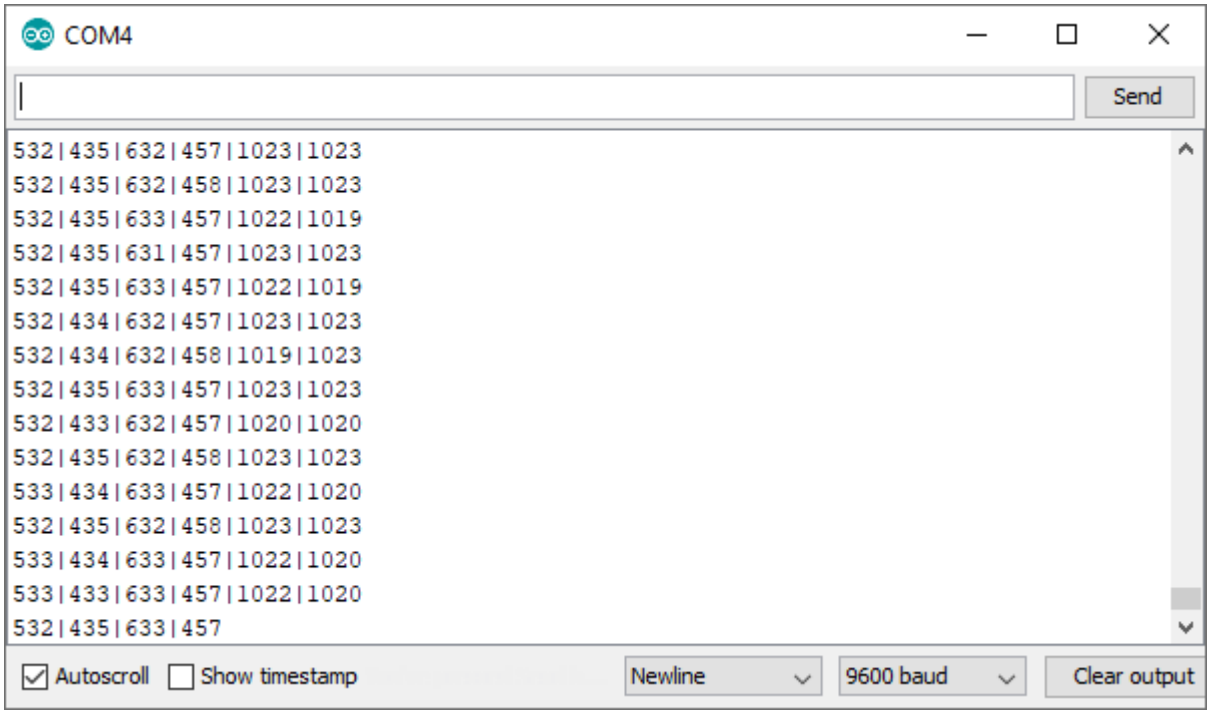


الكود البرمجي

```
const int NUM_SLIDERS = 5;
const int analogInputs[NUM_SLIDERS] = {A0, A1, A2, A3, A4};

int analogSliderValues[NUM_SLIDERS];

void setup() {
for (int i = 0; i < NUM_SLIDERS; i++) {
pinMode(analogInputs[i], INPUT);
}

Serial.begin(9600);
}
```

```
void loop() {
updateSliderValues();
sendSliderValues(); // Actually send data (all the time)
// printSliderValues(); // For debug
delay(10);
}

void updateSliderValues() {
for (int i = 0; i < NUM_SLIDERS; i++) {
analogSliderValues[i] = analogRead(analogInputs[i]);
}
}

void sendSliderValues() {
String builtString = String("");

for (int i = 0; i < NUM_SLIDERS; i++) {
builtString += String((int)analogSliderValues[i]);

if (i < NUM_SLIDERS - 1) {
builtString += String("|");
}
}

Serial.println(builtString);
}

void printSliderValues() {
for (int i = 0; i < NUM_SLIDERS; i++) {
String printedString = String("Slider #") + String(i + 1) + String(": ") +
String(analogSliderValues[i]) + String(" mV");
Serial.write(printedString.c_str());

if (i < NUM_SLIDERS - 1) {
Serial.write(" | ");
} else {
Serial.write("\n");
}
}
}
```

ارفع الأمر البرمجي السابق على لوحة اردوينو ثم قم بقراءة القيم على شاشة يجب أن تحصل على قراءات بين 0–1024 عند تدوير المقاومة المتغيرة



بعد التحقق من أن الاردوينو يعطي قراءات صحيحة اغلق واجهة الاتصال التسلسلي

شرح الكود البرمجي

نعرف متغير يشر إلى عدد المنافذ التي يتم توصيلها مع المقاومات
و نعرف المنافذ التناظرية على لوحة الاردوينو التي تم توصيلها مع المقاومات

```
const int NUM_SLIDERS = 5;
const int analogInputs[NUM_SLIDERS] = {A0, A1, A2, A3, A4};

int analogSliderValues[NUM_SLIDERS];
```

في دالة void setup() نعرف أن المنافذ التناظرية هي منافذ ادخال و نهيئة الاتصال التسلسلي

```
void setup() {
for (int i = 0; i < NUM_SLIDERS; i++) {
pinMode(analogInputs[i], INPUT);
}

Serial.begin(9600);
}
```

في دالة void loop نستدعي الدوال التالية
updateSliderValues() و sendSliderValues()و printSliderValues()

و التي سيتم انشاؤها لاحقا

```
void loop() {
updateSliderValues();
sendSliderValues(); // Actually send data (all the time)
// printSliderValues(); // For debug
```

```
delay(10);
}
```

دالة لتحديث القراءة من المنافذ التناظرية

```
void updateSliderValues() {
for (int i = 0; i < NUM_SLIDERS; i++) {
analogSliderValues[i] = analogRead(analogInputs[i]);
}
}
```

دالة لارسال القيم إلى جهاز الحاسوب

```
void sendSliderValues() {
String builtString = String("");

for (int i = 0; i < NUM_SLIDERS; i++) {
builtString += String((int)analogSliderValues[i]);

if (i < NUM_SLIDERS - 1) {
builtString += String("|");
}
}

Serial.println(builtString);
}
```

دالة طباعة قيم المنافذ التناظرية

```
void printSliderValues() {
for (int i = 0; i < NUM_SLIDERS; i++) {
String printedString = String("Slider #") + String(i + 1) + String(": ") +
String(analogSliderValues[i]) + String(" mV");
Serial.write(printedString.c_str());

if (i < NUM_SLIDERS - 1) {
Serial.write(" | ");
} else {
Serial.write("\n");
}
}
}
```

اعداد ملف التكوين

قم بتحميل الملف بالرابط يحتوي الملف المضغوط على ملف deej.exe و config.yaml قم بفك الضغط و احفظ كل من deej.exe و config.yaml في ملف واحد.

يحدد ملف التكوين التحكم بصوت التطبيقات بالاردوينو التي يتم تعيينها، المنفذ المستخدم للاتصال بلوحة اردوينو . قم بفتح ملف التكوين config.yaml عن طريق برنامج notepade الموجود على جهازك

```
slider_mapping:
  0: master
  1: chrome.exe
  2: spotify.exe
  3:
    - pathofexile_x64.exe
    - rocketleague.exe
  4: discord.exe

# set this to true if you want the controls inverted (i.e. top is 0%, bottom is
100%)
invert_sliders: false

# settings for connecting to the arduino board
com_port: COM4
baud_rate: 9600

# adjust the amount of signal noise reduction depending on your hardware quality
# supported values are "low" (excellent hardware), "default" (regular hardware) or
"high" (bad, noisy hardware)
noise_reduction: default
```

يمكن تعديل الملف حسب عدد المقاومات المتغيرة في مشروعك و البرامج التي ستتحكم بها، كذلك يمكنك التحكم بأكثر من برنامج عن طريق مقاومة واحدة

```
slider_mapping:
  0: master
  1: chrome.exe
  2: spotify.exe
  3:
    - pathofexile_x64.exe
    - rocketleague.exe
  4: discord.exe
```

`master` هو خيار خاص للتحكم في مستوى الصوت الرئيسي للنظام

`mic` هو خيار خاص للتحكم في مستوى إدخال الميكروفون

`deej.unmapped` هو خيار خاص للتحكم في جميع التطبيقات التي لا ترتبط بأي بمؤشر خاص لمستوى الصوت

`deej.current` هو خيار خاص للتحكم في أي تطبيق يتم التركيز عليه

`system` هو خيار خاص للتحكم في مستوى أصوات النظام

بعد رفع الكود سيقوم الاردوينو بإرسال قيم المقاومة المتغيرة الحالية باستمرار إلى جهاز الحاسوب، ثم يقوم الحاسوب بتشغيل العميل deej في الخلفية. يقرأ هذا العميل القيم من الاردوينو ويضبط مستوى الصوت وفقًا لملف التكوين المحدد

الهيكل

يمكنك تحميل ملف الهيكل بالرابط اذا توفرت لديك طابعة ثلاثية الأبعاد و يمكن ايضا أن تحاول صنع هيكل جهاز التحكم بصوت