ورقة بحثية مجلة مفتوحة الوصول

سلسلة تعلم استخدام نظام ROS2 في نظام التشغيل Windows10 (الجزء الأول)

د. عيسى الغنام* ، د. إياد حاتم**

*(كلية الهندسة، جامعة المنارة ، البريد الإلكتروني: Essa.Alghannam@manara.edu.sy)

**(كلية الهندسة، جامعة المنارة، البريد الإلكتروني: iyadhatem@manara.edu.sy)

الملخص

نظام تشغيل الروبوت (ROS) هو إطار برمجي متاح مجاناً يتمثل بمجموعة من مكتبات البرامج والأدوات اللازمة لبناء تطبيقات الروبوت والتي تسهل بناء ومشاركة التعليمات البرمجية بين تطبيقات الروبوتات المختلفة. تقدم هذه السلسة إرشادات لكل المهتمين في مجال هندسة الميكاترونيكس والروبوتيك لفهم مكونات نظام ROS2 وأهميته وكيفية استخدامه في نظام التشغيل WINDOWS مباشرة من دون وجود VIRTUAL MACHINE.

كلمات مفتاحية: ROS2, WINDOWS10, ROBOTICS, MECHATRONICS.

I. مقدمة

نظام تشغيل الروبوت (ROS) هو إطار برمجي متاح مجاناً يمتلك الأدوات ومكتبات البرامج مفتوحة المصدر التي تحتاجها للتحكم ب الروبوتات، كما تعزز مجتمعاً عالمياً مفتوح المصدر يتكون من المهندسين والمطورين والمتحمسين الذين يتعاونون لتحسين الروبوتات وجعل هذا المجال في متناول الجميع. يعد نظام ROS بمثابة العمود الفقري لمجموعة واسعة من الأنظمة مثل السيارات ذاتية القيادة والأذرع الروبوتية الصناعية والطائرات بدون طيار وأنظمة البحث والتطوير على سبيل المثال لا الحصر. الجدير بالذكر أن ROS ليس نظام تشغيل بالمعنى التقليدي لإدارة العمليات وتنظيم الملفات وجدولة المهام، ولكنه بالأحرى هو طبقة اتصالات منظمة تقع أعلى نظام التشغيل المضيف. في هذه السلسلة، سوف نستخدم الإصدار الأخير ROS2 Iron Irwini. يعمل ROS2 Iron Irwini الذي أطلق في 2023. وهو بمثابة توزيع الدعم طويل الأمد (LTS) لـ ROS، مما يضمن الدعم لمدة لا تقل عن 5 سنوات.

منذ أن بدأ نظام ROS في عام 2007، تغير الكثير في مجتمع الروبوتات وأنظمة ROS. ففي عام 2007، نشأت ROS بهدف إنشاء إطار برمجي تعاوني مخصص للنهوض بتطوير الروبوتات. في ذلك الوقت، كان على علماء الروبوتات كتابة جميع أكواد التشغيل الخاصة بهم من الصفر. قد يستغرق تنفيذ حتى مهمة أساسية، بدءاً من كتابة التعليمات البرمجية، وتطوير وحدة تحكم إطار العمل، والتعامل مع الاتصال التسلسلي بين وحدة التحكم الدقيقة والكمبيوتر الرئيسي، والتعامل مع تصحيح الأخطاء، عدة أسابيع. عند التعامل مع مشاريع أكثر شمولاً تتضمن الكثير من التعليمات البرمجية والعديد من أجهزة الاستشعار والمحركات، يتضاعف التعقيد بشكل كبير. تم إنشاء ROS للتخفيف من التعقيد من خلال تقديم العديد من الأدوات والحزم المعدة مسبقاً والتي طورها علماء الروبوتات والمبرمجون السابقون. تعمل هذه الموارد والأدوات على تقليل مقدار الوقت اللازم لتطوير البرامج بشكل كبير، مما يسمح لك بتشغيل الروبوت الخاص بك بسرعة أكبر وبدون عناء. يمنحك ROS أيضاً القدرة على توسيع نطاق أنظمة الروبوتات الخاصة بك. يملأ ROS2 فجوات ROS1 وهو مناسب ليس فقط للأذرع الآلية ولكن أيضاً للروبوتات المتنقلة والروبوتات ذات الأرجل والطائرات بدون طيار وروبوتات السرب والسيارات المستقلة. يتم دعم C++ و Python رسمياً. هدفنا الأساسي في

هذه السلسة ليس أن نصبح مطورين لنظام ROS، بل فهمه والتعامل معه وتسخيره للاستفادة منه للتحكم في روبوتات مختلفة وبرمجتها.

II. تنصيب ROS2

هناك عدة خيارات لتنصيب ROS 2 Iron Irwini الخيار الأول هو Binary packages وأما الثاني Building from source. سيؤدي التثبيت من الحزم الثنائية Binary packages أو من المصدر From Source إلى تثبيت ROS2 كامل الوظائف وقابل للاستخدام. الحزم الثنائية Binary packages مخصصة للاستخدام العام وتوفر تثبيتاً مدمجاً بالفعل already-built install لـ ROS2. يعد هذا أمراً رائعاً للأشخاص الذين يرغبون في التعمق والبدء في استخدام ROS2 كما هو على الفور. البناء من المصدر مخصص للمطورين الذين يتطلعون إلى تغيير أو حذف أجزاء من قاعدة ROS2 بشكل صريح. يوصى به أيضاً للأنظمة الأساسية التي لا تدعم الثنائيات. يمنحك البناء من المصدر أيضاً خيار تثبيت الإصدار الأحدث المطلق من ROS2. عند التثبيت من المصدر ستواجه مشكلة تتعلق بضرورة استخدام كاسر بروكسي VPN. لذلك نوصي لتنصيب ROS2 على WINDOWS10 استخدام الحزم الثنائية. وفي ما يلي خطوات التنصيب:

أ. تنصيب CHOCOLATEY:

وهي مدير حزمة في ويندوز يمكن تنصيبها بسهولة من الموقع الرسمي وسيتم استخدامها لتنصيب أدوات تطوير أخرى.

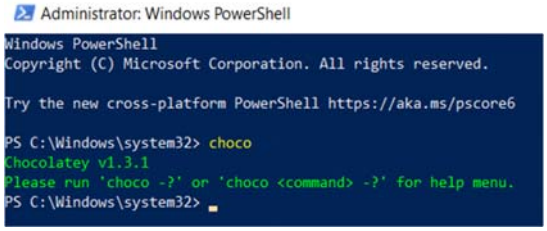
https://chocolatey.org/install

عموماً لتنصيبه تحتاج إلى الخطوات التالية من الرابط السابق:

افتح powershell كمدير، انسخ الأمر التالي وألصقه في نافذة shell التي فتحت واضغط enter:

```
Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
```

عندما تنتهي إذا لم تظهر أخطاء نفذ الأمر التالي choco لتتأكد من التثبيت:



الشكل 1 . تنصيب CHOCOLATEY

ب. تنصيب PYTHON:

افتح طرفية جديدة ونفذ الأمر التالي لتنصيب بايثون باستخدام Chocolatey:

```
choco install -y python --version 3.8.3
```

ت. تثبيت أدوات Visual C++ القابلة للتوزيع:

استخدم الأمر التالي:

```
choco install -y vcredist2013 vcredist140
```

ث. تثبيت OpenSSL:

قم بتنصيب Win64 OpenSSL v1.1 من الرابط التالي:

https://slproweb.com/products/Win32OpenSSL.html

(ملاحظة نسخ 32 بت أو نسخ لايت لا تعمل). خلال خطوات التنصيب استخدم الخيارات الافتراضية وعلى فرض قمت باستخدام مسار التنصيب الافتراضي على القرص الصلب الخاص بك، فاستخدم هذا الأمر لضبط متغيرات البيئة environment variable:

```
setx /m OPENSSL_CONF "C:\Program Files\OpenSSL-Win64\bin\openssl.cfg"
```

في النهاية لا بد من تعديل PATH الخاص على جهازك وإضافة OpenSSL-Win64 bin folder إليه. يمكنك القيام بذلك عن طريق النقر على أيقونة Windows، وكتابة:

"environment variable"، ثم النقر على:

"Edit the system environment variables". في مربع الحوار الناتج، انقر فوق "enviromment variables"، ثم انقر فوق "path" في الجزء السفلي، وأخيراً انقر فوق "edit" وأضف المسار أدناه.

```
C:\Program Files\OpenSSL-Win64\bin\
```

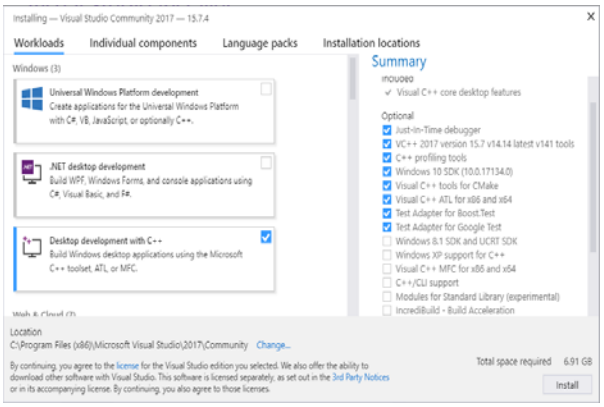
ج. تنصيب Visual Studio 2019:

توفر Microsoft إصداراً مجانياً من Visual Studio 2019، يسمى Community، والذي يمكن استخدامه لإنشاء تطبيقات تستخدم ROS2. الإصدارات الأخرى Professional أو Enterprise أيضاً يمكن استخدامها. يمكنك تنزيل برنامج التثبيت مباشرة من الموقع الرسمي لـ Microsoft. تأكد من تثبيت Visual C++ features.

إحدى الطرق السهلة للتأكد من تثبيتها هي اختيار Desktop development with C++ أثناء التنصيب.

تأكد من عدم تثبيت أي C++ CMake tools عن طريق إلغاء تحديدها في قائمة المكونات المراد تثبيتها.

الشكل التالي يبين الخيارات التي يجب تحديدها والتي يجب عدم تحديدها قي القائمة اليمينية.



الشكل 2 . Visual Studio 2019

ح. تنصيب OpenCV 3.4.6:

تتطلب بعض الأمثلة تثبيت OpenCV. يمكنك تنزيل نسخة مجمعة مسبقاً من OpenCV 3.4.6 من:

https://github.com/ros2/ros2/releases/download/opencv-archives/opencv-3.4.6-vc16.VS2019.zip

بافتراض أنك قمت بفك ضغطه إلى C:\opencv، اكتب ما يلي في موجه الأوامر (يتطلب امتيازات المسؤول):

```
setx /m OpenCV_DIR C:\opencv
```

نظراً لأنك تستخدم إصدار ROS مترجماً مسبقاً، فيجب أن نخبره بمكان العثور على مكتبات OpenCV. يجب عليك توسيع متغير PATH إلى:

```
C:\opencv\x64\vc16\bin
```

خ. تنصيب dependencies:

هناك بعض التبعيات غير المتوفرة في قاعدة بيانات حزمة Chocolatey. ولتسهيل عملية التثبيت اليدوي، نقوم بتوفير حزم Chocolatey اللازمة.

وبما أن بعض حزم Chocolatey تعتمد عليه، فإننا نبدأ بتثبيت CMake

```
choco install -y cmake
```

سوف تحتاج إلى إلحاق مجلد التالي إلى PATH.

```
C:\Program Files\CMake\bin
```

بداية من الرابط التالي في GitHub:

```
https://github.com/ros2/choco-packages/releases/latest
```

يجب تنزيل هذه الحزم:

- asio.1.12.1.nupkg
- bullet.3.17.nupkg
- cunit.2.1.3.nupkg
- eigen-3.3.4.nupkg
- tinyxml-usestl.2.6.2.nupkg
- tinyxml2.6.0.0.nupkg

بمجرد تنزيل هذه الحزم، افتح طرفية جديدة كمسؤول وقم بتنفيذ الأمر التالي:

```
choco install -y -s <PATH\TO\DOWNLOADS\> asio cunit eigen tinyxml-usestl tinyxml2 bullet
```

الرجاء استبدال <PATH\TO\DOWNLOADS> بالمجلد الذي قمت بتنزيل الحزم إليه.

بدايةً قم بالترقية الأولى لـ pip وsetuptools من خلال الأمر التالي:

```
python -m pip install -U pip setuptools==59.6.0
```

الآن قم بتثبيت بعض تبعيات بايثون الإضافية:

```
python -m pip install -U catkin_pkg cryptography empy importlib-metadata jsonschema lark==1.1.1 lxml matplotlib netifaces numpy opencv-python PyQt5 pillow psutil pycairo pydot pyparsing==2.4.7 pyyaml rosdistro
```

د. تنصيب miscellaneous prerequisites:

التثبيت التالي لـ xmllint: قم بتنزيل 64 bit binary archives الخاصة بـ libxml2 (وتوابعها Iconv وzlib) من:

```
https://www.zlatkovic.com/projects/libxml/
```

فك كافة المحفوظات على سبيل المثال في: `C:\xmllint`

أضف `C:\xmllint\bin` إلى `PATH`.

ذ. تنصيب QT5:

قم بتنزيل برنامج التثبيت 5.12.X offline installer من موقع Qt الرسمي.

```
https://www.qt.io/offline-installers
```

قم بتشغيل المثبت. تأكد من تحديد مكون `MSVC 2017 64-bit` ضمن الشجرة `Qt 5.12.12` -> `Qt`.

أخيراً، في window `cmd.exe` administrator، قم بتعيين متغيرات البيئة هذه. تفترض الأوامر أدناه أنك قمت بتثبيته على الموقع الافتراضي لـ `C:\Qt`.

```
setx /m Qt5_DIR C:\Qt\Qt5.12.12\5.12.12\msvc2017_64

setx /m QT_QPA_PLATFORM_PLUGIN_PATH C:\Qt\Qt5.12.12\5.12.12\msvc2017_64\plugins\platforms
```

ملحوظة: قد يتغير هذا المسار بناءً على إصدار MSVC المثبت والمسار الذي تم تثبيت Qt عليه وإصدار Qt المثبت.

ر. تنصيب توزيعات RQT:

لتشغيل rqt_graph، يلزمك تنزيل Graphviz وتثبيته.

```
https://graphviz.gitlab.io/_pages/Download/Download_windows.html
```

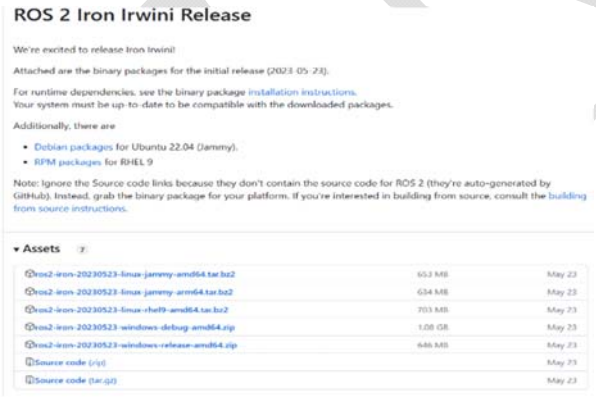
سيسألك المثبت عما إذا كنت تريد إضافة graphviz إلى PATH، اختر إما إضافته إلى المستخدم الحالي أو إلى كافة المستخدمين.

ز. تنصيب ROS2:

قم بتنزيل أحدث حزمة لنظام التشغيل Windows، على سبيل المثال: `ros2-package-windows-AMD64.zip`.

```
https://github.com/ros2/ros2/releases
```

ملحوظة: قد يكون هناك أكثر من خيار تنزيل ثنائي مما قد يتسبب في اختلاف اسم الملف.



الشكل 3 . تنصيب ROS2

قم بفك ضغط الملف المضغوط في مكان ما:

`C:\dev\ros2_iron`

III. ضبط إعدادات البيئة لتشغيل ROS2

تعتمد ROS2 على فكرة الجمع بين مساحات العمل workspaces باستخدام بيئة shell environment. أما بيئة العمل Workspace فهي مصطلح ROS لموقع ما على نظامك حيثما تقوم بتطوير شيء ما باستخدام ROS2.

تسمى مساحة عمل ROS2 الأساسية الطبقة السفلية underlay. تسمى مساحات العمل المحلية اللاحقة التراكبات overlays. عند العمل على تطوير شيء ما باستخدام ROS2، سيكون لديك عادةً العديد من مساحات العمل النشطة في نفس الوقت.

يؤدي الجمع Combining بين مساحات العمل workspaces إلى تسهيل عملية التطوير وفقاً لإصدارات مختلفة من ROS2، أو مقابل مجموعات مختلفة من الحزم different sets of packages. كما يسمح أيضاً بتثبيت العديد من توزيعات ROS2 على نفس الكمبيوتر والتبديل بينها.

يتم تحقيق ذلك عن طريق sourcing setup files في كل مرة تفتح فيها new shell، أو عن طريق إضافة the source command إلى shell startup script مرة واحدة.

بدون sourcing setup files، لن تتمكن من الوصول إلى أوامر ROS2 أو العثور على حزم ROS2 أو استخدامها. بمعنى آخر، لن تتمكن من استخدام ROS2 .

ستحتاج إلى تشغيل هذا الأمر على every new shell تفتحها لتتمكن من الوصول إلى أوامر ROS 2:

```
call C:\dev\ros2\local_setup.bat
```

يعتمد الأمر الدقيق على مكان تثبيت ROS2. إذا كنت تواجه مشكلات، فتأكد من أن مسار الملف يؤدي إلى مكان التثبيت على جهازك.

إذا كنت لا تريد أن تضطر إلى الحصول على ملف الإعداد source the setup file في كل مرة تفتح فيها new shell (تخطي الخطوة السابقة)، فيمكنك إضافة الأمر إلى البرنامج النصي لبدء تشغيل shell.

لمستخدمي PowerShell فقط، أنشئ مجلداً في "المستندات" يسمى "WindowsPowerShell". ضمن هذا المجلد، أنشئ ملف: "Microsoft.PowerShell_profile.ps1"، داخل الملف الصق:

```
C:\dev\ros2_humble\local_setup.ps1
```

سيطلب PowerShell إذناً لتشغيل هذا البرنامج النصي في كل مرة يتم فيها فتح new shell. لتجنب هذه المشكلة، يمكنك تشغيل:

```
Unblock-File C:\dev\ros2_humble\local_setup.ps1
```

للتراجع عن هذا، أزل ملف "Microsoft.PowerShell_profile.ps1" الجديد.

أما لمستخدمي cmd فقط، أنشئ ملفاً نصياً txt في أي مسار مثلاً: E:\dev وسمه "ros2-release.txt" مثلا. داخل الملف، الصق:

```
call "C:\Program Files\rti_connext_dds-5.3.1\resource\scripts\rtisetenv_x64Win64VS2017.bat"
call E:\dev\ros2_humble\local_setup.bat
cmd
```

احفظ الملف وغير اسمه الى ros2-release.cmd وعند تشغيله سواء double click أو run as administrator سيتم تنفيذ الأمرين المحتويين بداخله مباشرة أما الأمر الأخير cmd فليبقي النافذة في حالة عمل.

ملاحظة هامة: التشغيل run as administrator أفضل تجاوزاً لبعض المشاكل التي من الممكن أن تظهر عند double click لاحقاً.

للتراجع عن هذا، أزل ملف " ros2.txt " الجديد.

IV. فحص متغيرات البيئة:

نفذ الأمر التالي:

```
set | findstr -i ROS
```

تحقق من تعيين المتغيرات مثل ROS_DISTRO و ROS_VERSION

```
ROS_VERSION=2
```

```
ROS_PYTHON_VERSION=3
ROS_DISTRO=humble
```

تحقق من تعيين المتغيرات مثل ROS_DISTRO و ROS_VERSION

V. المتغير ROS_DOMAIN_ID

البرنامج الوسيط الافتراضي default middleware الذي تستخدمه ROS2 للاتصال communication هو DDS والذي هو اختصار لـ Data Distribution Service.

في DDS، تُعرف الآلية الأساسية لوجود شبكات منطقية مختلفة تشترك في شبكة فعلية different logical networks share a physical network باسم معرف المجال Domain ID. يمكن لـ nodes عقد ROS2 الموجودة في نفس المجال same domain اكتشاف الرسائل وإرسالها إلى بعضها البعض بحرية، بينما لا تستطيع عقد ROS2 الموجودة في المجالات المختلفة ذلك.

تستخدم جميع عقد ROS2 معرّف المجال domain ID افتراضياً 0. لتجنب التداخل بين مجموعات مختلفة من أجهزة الكمبيوتر التي تقوم بتشغيل ROS2 على نفس الشبكة، يجب تعيين معرف مجال مختلف لكل مجموعة.

لتخطي هذه background واختيار رقم آمن، ما عليك سوى اختيار معرف المجال بين 0 و 101، ضمناً.

```
set ROS_DOMAIN_ID=<your_domain_id>
set ROS_DOMAIN_ID=0
```

إذا كنت تريد أن تجعل هذا دائماً بين جلسات شل shell sessions ، فقم أيضاً بتشغيل:

```
setx ROS_DOMAIN_ID <your_domain_id>
```

VI. المتغير ROS_LOCALHOST_ONLY:

بشكل افتراضي، لا يقتصر ROS2 communication على المضيف المحلي localhost. يسمح لك متغير البيئة ROS_LOCALHOST_ONLY بتقييد اتصال ROS2 بالمضيف المحلي فقط. هذا يعني أن نظام ROS2 الخاص بك، موضوعاته topics، خدماته services وإجراءاته actions لن تكون مرئية لأجهزة الكمبيوتر الأخرى على الشبكة المحلية local network. يعد استخدام ROS_LOCALHOST_ONLY مفيداً في إعدادات معينة، مثل:

Classrooms, where multiple robots may publish to the same topic causing strange behaviours.

يمكنك ضبط متغير البيئة بالأمر التالي:

```
set ROS_LOCALHOST_ONLY=1
```

إذا كنت تريد أن تجعل هذا دائماً بين جلسات shell sessions، فقم أيضاً بتشغيل:

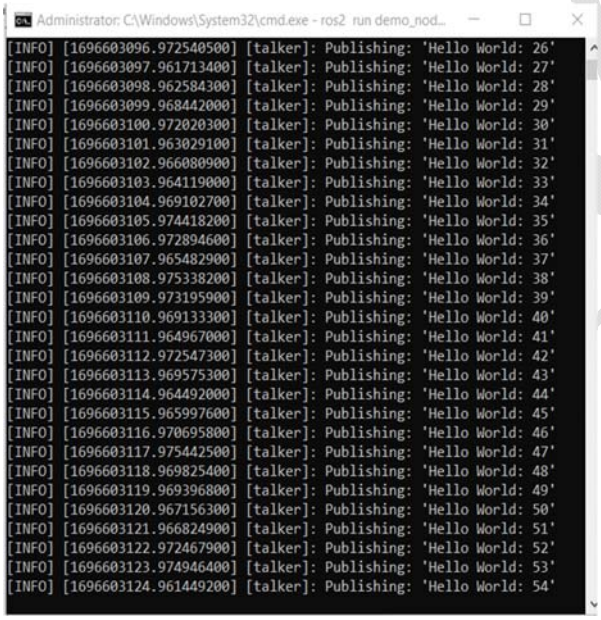
```
setx ROS_LOCALHOST_ONLY 1
```

VII. تطبيق مرسل مستقبل

في واجهة الأوامر، قم بإعداد بيئة ROS2 كما هو موضح أعلاه ثم قم بتشغيل C++ talker:

```
ros2 run demo_nodes_cpp talker
```

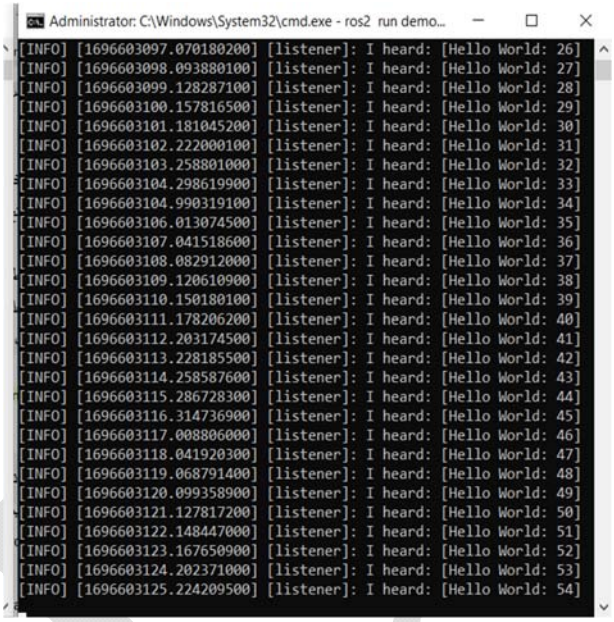
ستظهر النتيجة كما في الصورة التالية:



الشكل 4 . تشغيل C++ talker

ابدأ تشغيل واجهة أوامر أخرى وقم بتشغيل مستمع Python:

```
ros2 run demo_nodes_py listener
```



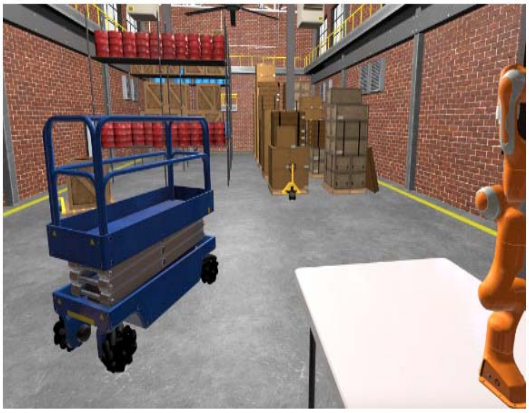
الشكل 5 . تشغيل مستمع Python

يجب أن ترى المتحدث يقول أنه ينشر الرسائل والمستمع يقول لقد سمعت تلك الرسائل. يتحقق هذا من أن كلاً من واجهات برمجة تطبيقات C++ و Python تعمل بشكل صحيح.

VIII. أدوات المحاكاة والإظهار في ROS2:

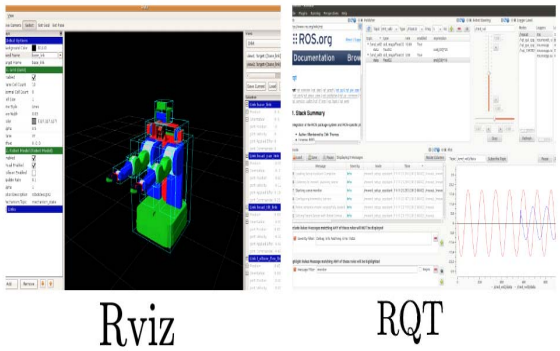
توفر ROS حزم برامج مجدولة لدعم محاكاة الروبوتات وتصور البيانات.

Gazebo عبارة عن محاكي روبوت مجاني ويمكنه توصيل البيانات عبر ROS2. لديه القدرة على تتبع مواقع الروبوت وتكرار حالة الروبوت الحقيقي. بالإضافة إلى ذلك، فهو يشتمل على أجهزة استشعار افتراضية، مما يتيح محاكاة بيانات الاستشعار الأصلية. يتيح لك ذلك تقييم واختبار التعليمات البرمجية الخاصة بك كما لو كنت تقوم بتشغيلها على نظام آلي فعلي.



الشكل 6 . Gazebo

Rviz وRQT مخصصان لأغراض التصور. Rviz عبارة عن مجموعة برامج شاملة لتصور البيانات ثلاثية الأبعاد التي تتيح التفاعل السلس مع البيانات باستخدام ROS2. وهو مجهز بقدرات تصور البيانات المختلفة، مما يتيح للمستخدمين التفاعل بشكل فعال مع الروبوت الخاص بهم. من ناحية أخرى، فإن RQT عبارة عن واجهة مستخدم رسومية قائمة على البرنامج المساعد ويمكن استخدامها مع ROS2. يأتي مزوداً بمكونات إضافية رسومية متغيرة مثل ناشر الموضوع، وعارض الصور، ومُحدِّث المعلمات، ومصور الرسم البياني للعقدة. يركز RQT بشكل أكبر على تفاعل المستخدم مع ROS2، بينما يركز Rviz على تصور البيانات ثلاثية الأبعاد.



الشكل 7 . rviz–rqt

يمكن تصميم الروبوتات عن طريق إنشاء نموذج URDF (تنسيق وصف الروبوت الموحد). URDF هو تنسيق ملف يستند إلى XML يُستخدم لوصف الخصائص الهندسية والحركية والبصرية للروبوت. يتضمن ملف URDF معلومات حول مفاصل الروبوت، والروابط، والهندسة المرئية وهندسة التصادم، وحدود المفاصل، والمزيد. جزء من URDF له هيكل مثل هذا:

```
<?xml version="1.0"?>
<robot name="two_link_arm">

  <!-- Link 1: Base -->
  <link name="base_link">
    <visual>
      <geometry>
        <cylinder length="0.1" radius="0.05"/>
      </geometry>
    </visual>
  </link>

  <!-- Link 2: First Arm Segment -->
  <link name="link1">
    <visual>
      <geometry>
        <box size="0.2 0.05 0.05"/>
```

الشكل 8 . تشغيل مستمع Python

IX. إلغاء التنصيب:

إذا قمت بتثبيت مساحة العمل الخاصة بك موضح أعلاه، فقد تكون "إزالة التثبيت" مجرد مسألة فتح محطة طرفية جديدة وعدم تحديد مصدر ملف إعداد مساحة العمل

The workspace's setup file not sourcing

بهذه الطريقة، سوف تتصرف بيئتك كما لو أنه لا يوجد تثبيت Iron على نظامك. إذا كنت تحاول أيضاً تحرير مساحة، فيمكنك حذف دليل مساحة العمل بالكامل باستخدام:

```
rmdir /s /q \ros2_iron
```

X. خلاصة

باختصار، يعد نظام تشغيل الروبوت (ROS) إطاراً برمجياً متعدد الاستخدامات للتحكم في الروبوت، وتعزيز التعاون والابتكار داخل مجتمع عالمي مفتوح المصدر. من خلال ROS، نستفيد من مجموعة أدوات قوية من المكتبات والأدوات المساعدة، مما يؤدي إلى تبسيط عملية إنشاء التعليمات البرمجية ومشاركتها عبر مجموعة واسعة من تطبيقات الروبوتات. لا يعمل نظام ROS بمثابة العمود الفقري للأنظمة المتنوعة فحسب، بل يوفر أيضاً منصة للتطوير الفعال والمعياري. بينما نبدأ رحلتنا مع ROS2 iron iwini وواجهة برمجة التطبيقات ROS–Python، فإنّ هدفنا ليس الخوض في تطوير ROS المعقد ولكن بدلاً من ذلك الاستفادة من هذه الأدوات للتحكم الفعال في ذراع الروبوت.

يجب أن يتم تهيئة configure بيئة تطوير ROS2 بشكل صحيح قبل الاستخدام. يمكن القيام بذلك بطريقتين: إما تحديد مصادر ملفات الإعداد sourcing the setup files في كل cmd جديد تفتحه، أو إضافة الأمر المصدر إلى البرنامج النصي لبدء التشغيل adding the source command to your startup .script

إذا واجهت أي مشاكل في تحديد موقع الحزم packages أو استخدامها مع ROS2، فإن أول شيء يجب عليك فعله هو التحقق من متغيرات البيئة environment variables الخاصة بك والتأكد من ضبطها على الإصدار version والتوزيع distro الذي تريده.

المراجع

[1]. S. M. Metev and V. P. Veiko, "*Laser Assisted Microtechnology"*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[2]. J. Breckling, Ed., "*The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser".* Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[3]. S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "*A novel ultrathin elevated channel low-temperature poly-Si TFT*," IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.

[4]. M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "*High resolution fiber distributed measurements with coherent OFDR*," in Proc. ECOC'00, 2000, paper 11.3.4, p. 109.

[5]. R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "*High-speed digital-to-RF converter*," U.S. Patent 5 668 842, Sept. 16, 1997.

[6]. (2002) The IEEE website. [Online]. Available: *http://www.ieee.org/*

[7]. M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/

[8]. https://www.ros.org/

[9]. http://wiki.ros.org/
[10]. http://wiki.ros.org/ROS/Tutorials

JMU