

الحمد لله وبعد .. هنتكلم باذن الله النهارده عن الـ ROS :

خلينا نقول ان قبل كده شغنا روبوتات صغيره معموله بالـ Arduino والـ Ultrasonic علشان يتفادى الحيطان و بردو بالـ IR زي الـ Line follower .. كانت حاجات جميله وبسيطه كده .. خلينا بقى نعلى بالمستوى شويه و نقول اننا عايزين حاجه متطوره اكتر وبتعمل معالجة للصور و تتحكم فيها عن طريق الشبكات و الواي فاي وهكذا .. طيب هل هنبدأ نكتب كل حاجه من جديد خالص او حتى هنبدأ منين ؟ هنا بقى ييجي دور حبيبنا الجميل ROS .. لو حاولت تدخل تشوف الشروحات بتاعته على الموقع الرسمي هتلاقي الموضوع معقد شويه على ما تفهم هو ايه وبيعمل ايه وليه هو مفيد .. تعال بقى معايا نبسط الموضوع خالص ..

What is ROS, anyway?

1. بيقول لك يا سيدي انه عبارة عن نظام تشغيل للروبوت بتاعك - بيتسطب على Linux وانواع مختلفه من الكمبيوتر زي Raspberry Pi او laptop او الي سي عموما. طيب بيعمل ايه ؟
بيقدمك طريقة Standard Connect تبيها كل الحساسات اللي عندك زي :
(cameras, distance sensors, analog to digital converters, IMU)
actuators : (drive motors, servos, lights)
جنب بعض كده مع الكود اللي بياخذ القرارات وينفذ الـ Actions اللي عايزينها. هو زي اللزق كده اللي بيربط كل دول ببعض وبيوفر عليك ملل كتير جدا وتكتشف ازاى تمرر الداتا بتاعت كل حاجه من دول مع التانيه.
2. نظام ROS بيساعدك في توصيل أجهزة كمبيوتر متعددة أو معالجات دقيقة بسهولة على الروبوت أو عبر الشبكة. على سبيل المثال ، تقدر تتحكم في روبوت ROS ، من مكانك عبر شبكة الواي فاي ، أو أن يكون عندك كمبيوتر بيعمل عمليات ثقيلة زي معالجة صور مثلا و بعد م يخلصها بيعت الداتا دي للروبوت مثلا ياخذ بيها Actions.
3. ROS بيووفر مكتبات للعمليات المعقدة مثل بناء الخرائط و localization ، واستخدام LIDAR وأجهزة الاستشعار الأخرى ، بمجرد ما تسطبه عند على الروبوت بتاعك هو يقدر يدمج العمليات دي كلها مع بعضها من غير ما تكتب كل مكتبة لوحدها من البدايه.
4. نظام ROS برضو بيحتوي على بعض الأدوات المفيدة جدا لتصوير البيانات من أجهزة الاستشعار ويمكن تدفق هذه البيانات. يعني مثلا انت بتكتب كود عادي بالسي بياخذ تاريخ ميلاد المستخدم ويطبعه هو عنده كم سنه. هو بقى بيعمل لك Graph كده الداتا راичه فين وجايه منين .. يعني مثلا بص معي كده : دي عبارته عن Controller Node "هنشرح بعدين يعني اي نود" بتبعت السرعه للـ Motor Node اللي المفروض يتحرك بها



5. اخيرا بقى يا سيدي انت تقدر تختبر الكود بتاعك عن طريق Simulator من غير ما يكون معاك روبوت حقيقي وهاردوير يعني.

How it works?

بمجرد ما تفتح دماغك معي كده هتلاقي ان التصميم بتاع ROS رايق كده ونضيف (:

- الموضوع في الاول بيبدأ عندنا حاجة اسمها **roscore** ده زي ال Boss الكبير كده اي حاجة بتعوز تتواصل او تبعت و تستقبل داتا مع حاجة ثانيه لازم تسجل عنده هو الاول . يعني مثلاً انا من لوحه المفاتيح عايز احرك الروبوت - في البدايه المونتور هيكون مسجل على roscore هيقول له انا جاهز استقبل داتا فتقوم لوحه المفاتيح تقول له يا عمنا انا عايزه ابعت داتا فهو يروح يوصلهم ببعض وبعدها يقولهم بالإذن أنا بقا و هما يتواصلوا مع بعض مع أنفسهم.

طيب ازاي انا اتعامل مع الروبوت او ادي له Function معينه ؟ هنا بقي تيجي ال Node.. على سبيل المثال.. عندي نود مسؤولة عن أخذ التعليمات الحركية "السرعة الخطية و السيتا علشان لو هلف يمين او شمال" وتوجيه المونتورز للتحرك. ممكن يكون عندي نود ثانيه مسؤولة عن قراءة قيم المستشعر زي الانكودر مثلاً ، وابعثها للنود الثانيه. أخيراً ، النود ما هي الا ملف كود صغير بيعمل وظيفه محدده و اقدر اشغل Nodes كثيره جدا مع بعض نفس الوقت.

ال Node التي بتبعت المعلومات للنود الأخرى هي **publisher**. ال Node اللي تتلقى المعلومات دي هي **subscriber**. ممكن تكون النود بتبعت و بتستقبل داتا ثانيه خالص في نفس الوقت عادي

- الداتا اللي بتتبع دي ببعثها عن طريق حاجة اسمها **topic** - الداتا نفسها اسمها **messages** تعال اقول لك مثال هيفهمك الليله دي كلها :

- تخيل وركز معايا انك بتشرب عصير قصب.. بوقك هنا هو اللي **subscriber** والـ كوبايه هي **publisher**.

- الداتا نفسها هي عصير القصب ودي هي الـ **messages** اما الـ **topic** فهو الشاليموه اللي بنبعث عليه.

- بس كده يا عمنا بنمنى تكون فهمت عشان بصراحه مش لاقى مثال اكتر من كده اوضح بيه (:
- بص بردو ممكن تقول ان topics دي فيش .. فيشة الكهرباء العاديه اللي بيمشي جواها الكهرباء عادي يعني .. و اقرأ دا بتركيز هتفهم الليله كلها :

- The term 'publish' stands for the action of **transmitting** messages **to the topic**. The publisher node registers its own information and topic with the **master**, and sends a message to connected subscriber nodes that are interested **in the same topic**.

- The term 'subscribe' stands for the action of **receiving** relative messages corresponding to the topic. The subscriber node registers its own information and topic with the **master**, and receives publisher information that publishes relative topic from the master. Based on received publisher information, the subscriber node directly requests connection to the publisher node and receives messages from the connected publisher node. A subscriber is declared in the node and can be declared multiple times in one node.



- فاكر الصورة دي ؟ تعالى نشرحها بقي بكلام Technical شوية :

فالمثال دا ال كونترولر نود (**teleop_turtle**) بتاخد ال keyboard input from user و بت publish الداتا اللي هيا "velocity commands". نيجي لـ (motor controller node)

(**turtlesim**) دي نود بت (**subscribes** to motor velocity **topic** (cmd_vel

وبتحول الأوامر دي لـ motor movement.

دا الشكل بتاعها فـ ال simulation



طيب ايه الصعوبات اللي ممكن تواجهك وانت بتحرك السلحفاة دي؟ انك مثلاً تحاول تحركها وترسم مربع - هنا بقى هتعمل كود بالبايثون كده يعمل شويه حسابات ويرجع فيديباك للنود انها اتحركت مده قد كذا ف المفروض بعدها تتحرك ثاني بزاويه ٩٠ مده قد كذا.



دا شكل ال rqt graph بيوضح انهى نود بتتبع و بتستقبل والداتا وهكذا... هنرجله ثاني لا تقلق..

طيب عظيم يا عم.. هو احنا جايين نلعب بالسלحفة؟ فين الروبوت؟ اصبر حاضِر هقولك :)
لنفترض أنك عايز تشغل ROS على Raspberry Pi للتعامل مع الاتصالات والتحكم عالي المستوى ، وعايز
Arduino أيضًا للتحكم في الموتورز وقياس الفولت للبطارية. علاوة على ذلك ، عندك التراسونيك عايز
توصلها للحصول على قياسات لتجنب العوائق. هنشرح كل دا حدة حدة..

1.هننزل و نسطب ROS على الراسبيري باي.

2.بمجرد تشغيل ROS على Pi، نقدر نشغل roscore "أول أمر بتعمله كأنك بتقوله صحصح معايا".

3.هناك عدة طرق لجعل Arduino يتكلم ويبيع داتا إلى Pi ، غالبا الطريقة الأكثر وضوحًا هي
استخدام باكدج rosserial Arduino. إذا قمت بتوصيل Arduino بـ Pi عبر USB ، فيمكن لـ
Arduino التسجيل كـ **Node** مع roscore و Publish ال Topics و Subscribe فيها.

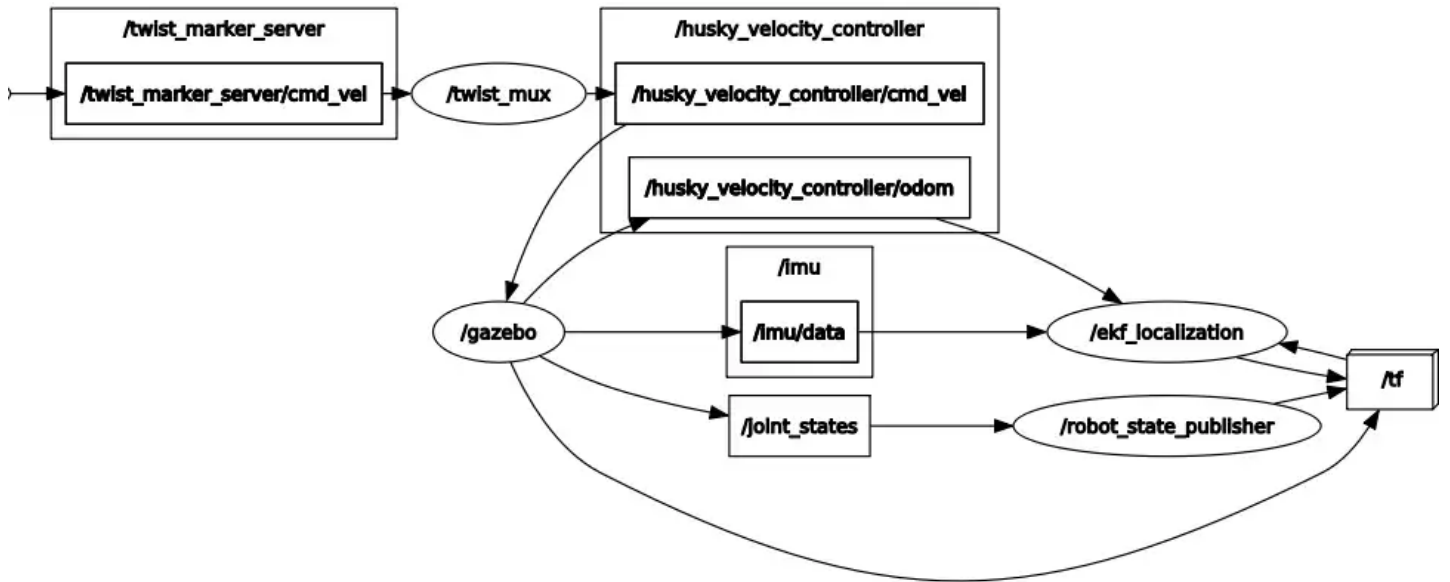
باقي الخطوات هكتبها لك انجلش علشان لما تقرأه بعد كدا تكون مستوعبه.. بص كدا:

4. The Arduino then **publishes** a topic for the measured battery voltages, and **subscribes** to a **motor velocity command topic**. Those might come from a tele-operation (teleop) node or from control software running on the Pi. The same Arduino can both publish topics and subscribe to topics.

5. A **Python controller program** on the Pi using the **ROS libraries connects to roscore** and **subscribes** to the **sensor topics being published by the Arduino**, and starts **publishing** motor velocity commands.

6. The Ultrasonic module could be handled in a variety of ways too. You could **connect** it to the **GPIO pins on the Pi** and have a simple Python node running on it that **published** Ultrasonic values, for example. It could also run on the Arduino, as long as you were careful not to **overload** it. "دا اردوينو غلبان يعني"

عايز تضيف RP-Lidar؟ لا مشكلة - وصله بـ Pi عبر USB واعمل نود تأخذ تلك البيانات وتـ **publish**.
الموضوع بسيط ان شاء الله.. بس نلاقي ليدار بس.. قولوا يارب :)



دي رسمة لـ Husky robot scanning the environment with a LIDAR - عايزك بس تقرأهم كده هتلاقي الداتا بتاعت الـ IMU داخله ع الكالمان فلترو controller/odom اللي هيا قرايات الانكودرز وهكذا.

متنساش الشكل دا اسمه rqt بيوضحك النودز اللي شغالة بالتوبيكس بالمسج وهكذا.

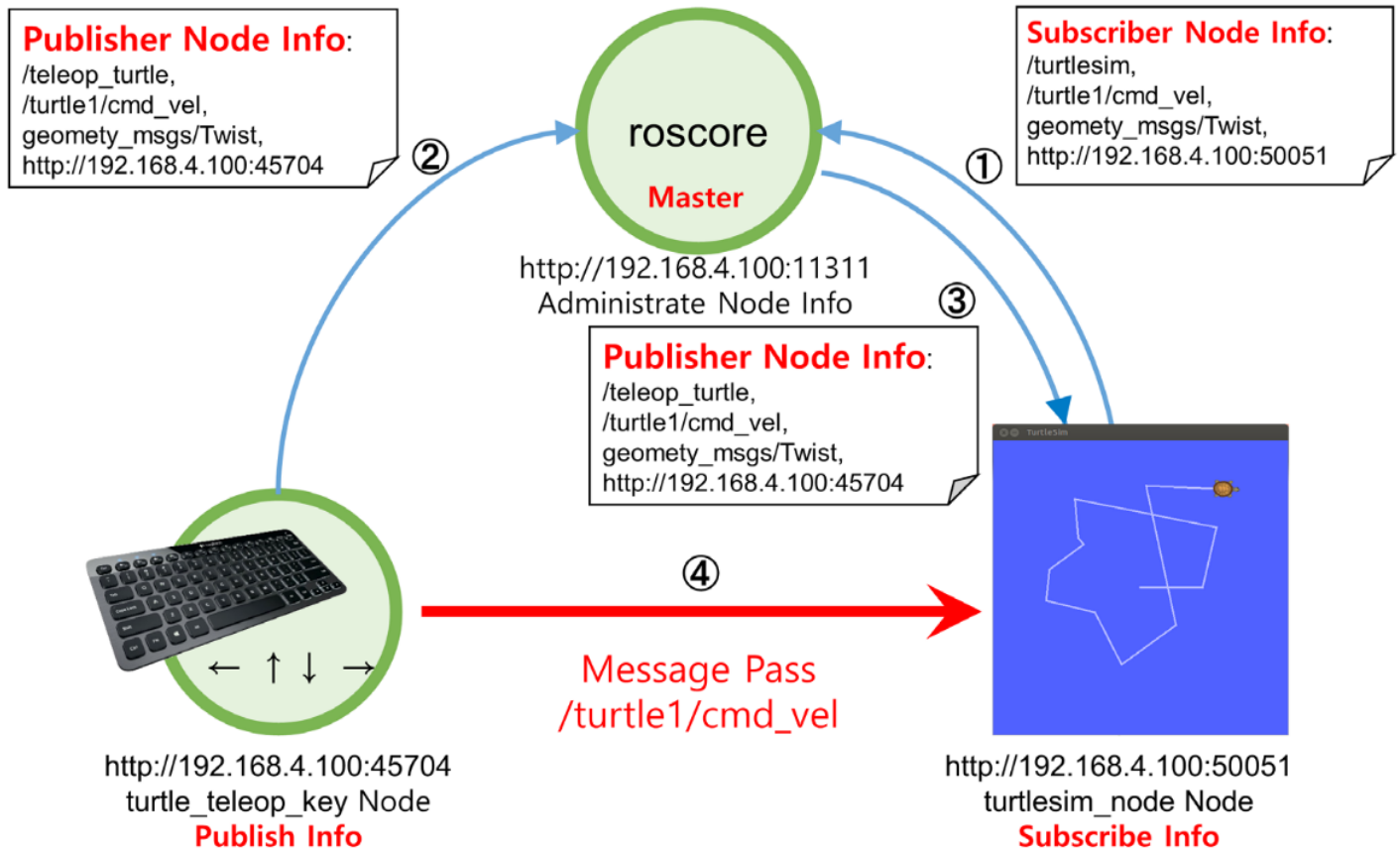
• ندخل بقي ع اخر حاجة و هيا **The Gazebo simulator** :

Gazebo is a 3D dynamic **simulator** with the ability to accurately and efficiently simulate **Robots** in complex **indoor** and **outdoor** environments. Gazebo offers physics simulation at a much higher degree of fidelity, a suite of **sensors**, and interfaces for both users and programs.

دا المحاكى بتاعنا - نقدر نرسم فيه شقة مثلا و الأوض اللي فيه و نعمل تبيست للأكواد بتاعتنا..

• واحدة من أروع ميزات ROS ، في رأيي ، هي أنه يمكن تشغيل النودز على أجهزة كمبيوتر **مختلفة** عن طريق **roscore**! أنت بس بتدي للكمبيوتر الذي يقوم بتشغيل النودز "الروبوت" (اسم الجهاز / IP الخاص بجهاز الكمبيوتر الذي يقوم بتشغيل roscore "اللاب") ، ويتصل ويبدا ينقلوا داتا مع بعض.

تعال بقي نلخص الموضوع دا كله ..



- **ROS Master** : acts as a name server for node-to-node connections and message communication. The command `roscore` is used to run the master, and if you run the master, **you can register the name of each node** and **get** information when needed. The connection between nodes and message communication such as topics and services are **impossible** without the master.
- **NODE** : A unit of processor running in ROS. one executable program
- **Message** : A node sends or receives data between nodes via a message. Messages are variables such as integer, floating point, and boolean.
- **Topic** : يعني مثلا عندي التوبيك بتاع السرعة - لوحه المفاتيح هتسجل على روس كور وتقول له انا عايز ابعت داتا على التوبيك دا والنود بتاعت السلحفة تقولة عايزة استقبل داتا على التوبيك دا كمان - كده لما يلاقهم على نفس التوبيك يوصلهم مع بعض دايركت.
- بس كدا لو وصلت لحد هنا ف احنا خلصنا يا عمنا .. متنساش الفيدباك الرايق وعرفني اذا كان فيه حاجة لسة مش مفهومة بالنسبالك ولا كله زي الفل.