

الإهداء

إلى من لا يضاهيهم أحد في الكون، إلى من أمرنا الله ببرّهما، إلى من بذلا الكثير، وقدّما ما لا يمكن أن يردّ وكانوا خير داعم لي طوال مسيرتي الدراسية..

والدي ووالدتي.

ليس تقاسما للسقف والطعام الواحد فحسب، بل هو تقاسم للضحكات والدموع، للتعثر والنجاح، لأمر الأيام وأحلالها، هو تقاسم للروح والملامح الواحدة..

إلى إخوتي وأخواتي.

إلى خالدي الذكر الذين وافتهم المنية وكانوا خير مثال للصدق والمحبة والاحترام..
جدي صبحي وعمي رضوان.

إلى من صاقت السطور في وصفهم فوسعهم قلبي، من وقفوا بجانبي وكانوا عوناً وسنداً لي في سنواتي الدراسية..
إلى جميع أفراد عائلتي.

إلى أصدقاء الطفولة بذكرياتها الجميلة، من شاركني أصعب اللحظات وأجملها..
مالك وعبدالله.

إلى أصدقاء الطريق الطويل، من شاركني حياتي الجامعية بحلوها ومرّها..
أحمد سفريني، مصطفى سيد أحمد، محمد علي، إبراهيم ناشد، يامن قصاص، محمد بوسطجي،
أحمد قيموز، رولا ملقي، لمياء الحريري، صهيب ديناوي، محمد سكر، نزار جبل، بيتس العبد،
مصطفى أبوسعدى، محمد أسعد.

شكر وتقدير

الحمد لله عزّ وجلّ الذي وفقني في إتمام هذا المشروع، والذي ألهمني الصحة والعافية والصبر والعزمية..
فالحمد لله حمداً كثيراً.

عن رسول الله صلى الله عليه وسلم أنه قال: "لَيْسَ مَنْ لَمْ يُجِلَّ كَبِيرَنَا، وَيَرْحَمْ صَغِيرَنَا، وَيَعْرِفْ لِعَالَمِنَا حَقَّهُ".

أتقدم بجزيل الشكر والتقدير إلى الدكتور المشرف عبد القادر جوخدار على كل ما قدمه لي من توجيهات ومعلومات قيمة ساهمت في إثراء موضوع البحث في جوانبه المختلفة.

كما نتقدم بجزيل الشكر إلى أعضاء لجنة المناقشة الموقرة المكونة من رئيس قسم الميكاترونิกس الدكتور أيمن نعال ونائب العميد للشؤون الإدارية الدكتور عماد الروح والدكتور يمان جانات.

والشكر إلى جميع أعضاء الهيئة التدريسية في كلية الهندسة الكهربائية والإلكترونية وأخص منهم: الدكتورة رانيا العجمي والمهندسين والمهندفات رعد العيسى، عبدالله العلو، داليا قس حنا، نانسي مقدس أنطوان، هشام دباغ.

الفهرس

1	المقدمة Introduction
2	نبذة مختصرة Abstract
3	الفصل الأول: الروبوتات Robots
4	(1-1) تاريخ الروبوتات وتطورها عبر الزمن The History and Evolution of Robots
6	(2-1) تصنیف الروبوتات Classification of Robots
9	(3-1) التحكم في الروبوتات Robots Control
11	(1-3-1) التحكم القائم على الرؤية Vision-Based Control
12	(4-1) نظم الروبوتات Robotics Systems
13	(1-4-1) أنظمة الروبوتات النقالة Mobile Robots Systems
14	الفصل الثاني: كاميرا Microsoft Kinect
15	(1-2) نبذة عن الكاميرا Microsoft Kinect
16	(2-2) بنية الكاميرا Microsoft Kinect
18	(3-2) برمجة الكاميرا و معالجة الصورة Programming the Kinect and Image Processing
19	(4-2) استخدامات وتطبيقات الكاميرا Microsoft Kinect Application
21	(1-4-2) التحكم في حركة الروبوتات باستخدام Microsoft Kinect
23	الفصل الثالث: الاردوينو Arduino
24	(1-3) لمحه عن المتحكمات المصغرة Microcontrollers
25	(2-3) مقدمة عن الاردوينو Introduction to Arduino
26	(1-2-3) أنواع الاردوينو Arduino Boards
29	(2-2-3) برمجة الاردوينو Arduino Programming

31	Arduino Wireless Communication (3-2-3)
34	الفصل الرابع: الدراسة النظرية Theoretical Study
35	(1-4) الدراسة الحركية Kinematic Study
37	(2-4) الدراسة الديناميكية Dynamic Study
40	الفصل الخامس: المكونات المادية للنظام System Hardware
41	(1-5) محرك التيار المستمر DC-Motor
43	(2-5) دارة قيادة محرك التيار المستمر DC-Motor Driver L298N
46	(3-5) تصميم الروبوت النقال Mobile Robot Design
47	(4-5) وحدة البلوتوث Bluetooth Module HC-05
49	(5-5) الاردوينو Arduino UNO
49	(6-5) كاميرا Microsoft Kinect
50	(7-5) المخطط الصندوقى ومخطط الدارة Block Diagram and Circuit Diagram
52	الفصل السادس: برمجة ومعالجة النظام System Software
53	(1-6) بيئة Processing IDE
54	(2-6) مكتبيات SimpleOpenNI and Kinect Libraries
55	(1-2-6) خطوات عمل البرنامج في بيئة Processing IDE
56	(3-6) النص البرمجي للمشروع Program Code
66	الصعوبات والآفاق المستقبلية
67	المراجع

المقدمة

مع التطور التكنولوجي ودخول الروبوتات بمختلف أنواعها تدريجياً إلى حياتنا أصبح من المهم دراستها ودراسة وتطوير استراتيجيات التحكم بها، وما زال التحكم بالروبوتات عن طريق الإيماءات أو إشارات اليد تحد صعب للمبرمجين والمطوريين.

التحكم في الروبوتات عن بعد يتم عادةً عن طريق جهاز تحكم يكون متصلًا بشكل لاسلكي مع الروبوت، وهناك روبوتات ذاتية التحكم حيث يكون لديها معرفة ببعض المعلومات مثل احداثيات موقعه الحالي وموقع الهدف الذي يريد الوصول إليه وكيفية الوصول إلى هذا الهدف وبعضها تتمكن من اكتشاف العقبات الموجودة في طريقها للهدف وتتفاداها باستخدام حساسات معينة، ولكن هذا النوع من الروبوتات تتم برمجته للقيام بمهمة محددة ولا يمكن استخدامه لأي مهمة أخرى إلا بعد أن تتم إعادة برمجته.

في هذا البحث سيتم التحكم بروبوت متelligent بطريقة مختلفة، لن يتم استخدام جهاز تحكم ولن يكون الروبوت محدد بمهمة واحدة، بل سيقدم هذا المشروع روبوت متelligent يتحرك وفقاً لإشارات يد الإنسان، سيتم تحديد حركة ذراع الإنسان في الفراغ ثلاثي الأبعاد بواسطة الكاميرا Microsoft Kinect ومعالجتها وإرسالها لاسلكياً إلى المتحكم المصغر والذي بدوره سوف يقوم بمعالجة البيانات المرسلة إليه وتحويلها إلى تعليمات تتحكم بمشغلات الروبوت (المحركات).

إن إطلاق هذا النموذج وتعليمه كيفية التفاعل مع حركات جسد الإنسان، يؤدي إلى أن الرؤية الاصطناعية يمكن أن تضع الروبوتات في مستوى جديد ومختلف تماماً.

نبذة مختصرة

هذا المشروع يشرح منهجية للتحكم بروبوت متقل بواسطة الإيماءات، باستخدام حركات يد الإنسان والكاميرا Microsoft Kinect.

قدرة الكاميرا Microsoft Kinect على تتبع كافة أعضاء جسم الإنسان قد تم استخدامها في هذا المشروع بهدف تطوير برمجة تطبيقات التحديد والتعرف على إيماءات الجسم وتحويلها إلى أوامر تحكم.

المنهجية المقترحة في هذا المشروع تم تجربتها عملياً، نتائج التقييم التجاري عرضت في هذا البحث، وقد أظهرت النتائج أن المنهجية المقترحة دقيقة وموثوقة ويمكن اعتمادها واستخدامها للتحكم في الروبوتات المتنقلة في التطبيقات المختلفة.

ويهدف هذا البحث أيضاً إلى تقديم فكرة معالجة الصورة باستخدام الكاميرا Microsoft Kinect وبرنامج المعالجة Processing IDE، بالإضافة إلى ربط البيئة البرمجية للكاميرا مع البيئة البرمجية للأردوينو حيث سيقوم الأخير بتشغيل الروبوت المتقل.



الفصل الأول

الروبوتات Robots

1-1- تاريخ الروبوتات وتطورها عبر الزمن The History and Evolution of Robots

الروبوت يمكن أن يسمى بالعربية (الإنسان الآلي) وهي آلة قادرة على القيام بأعمال مبرمجة مسبقا، إما بتحكم وسيطرة مباشرة من الإنسان أو بتوجيهه من برامج حاسوبية.

غالباً ما تكون الأفعال التي يبرمج الروبوت على أدائها أعمالاً شاقة أو خطيرة أو دقيقة، مثل البحث عن الألغام والتخلص من النفايات المشعة، أو أعمالاً صناعية دقيقة أو شاقة.

ظهرت الكلمة "روبوت" لأول مرة عام 1920، في مسرحية الكاتب المسرحي التشيكي كارل تسابيك، التي حملت عنوان "رجال روسو الآلية العالمية"، ترمز الكلمة "روبوت" في اللغة التشيكية إلى العمل الشاق، إذ أنها مشتقة من الكلمة (Robota) التي تعني العمل الإجباري، ومبتكر هذه الكلمة هو جوزيف تسابيك، أخ الكاتب المسرحي المذكور سابقا، والذي ابتدعها في محاولة منه لمساعدة أخيه على ابتكار اسم ما للآلات الحية في العمل المسرحي.

وببدأً من هذا التاريخ، بدأت هذه الكلمة تنتشر في كتب وأفلام الخيال العلمي التي قدمت عبر السنوات عدد من الأفكار والتصورات لتلك الآلات وعلاقتها بالإنسان، الأمر الذي كان من شأنه أن يفتح أفق كبيرة للمخترعين ليبتكرו ويطورو ما أمكن منها.

على الرغم من أن علم الروبوتات لم يظهر حتى القرن العشرين، إلا أن الحركة الآلية التي اخترعها الإنسان تعود إلى قبل ذلك بكثير، وتشهد العديد من المصادر على أن الروبوتات تعود إلى العصور القديمة، حيث طور الإغريق والرومان بعض الروبوتات البسيطة كأدوات وألعاب أطفال. يمكن أن ترجع جذور الروبوتات إلى أجهزة آلية اخترعت في الماضي البعيد وأطلق عليها "الآلات ذاتية الحركة"، ففي عهد قدماء المصريين حوالي عام 15000 قبل الميلاد كان يوجد تمثال للملك ممنون يصدر أصواتاً جميلة في الصباح.

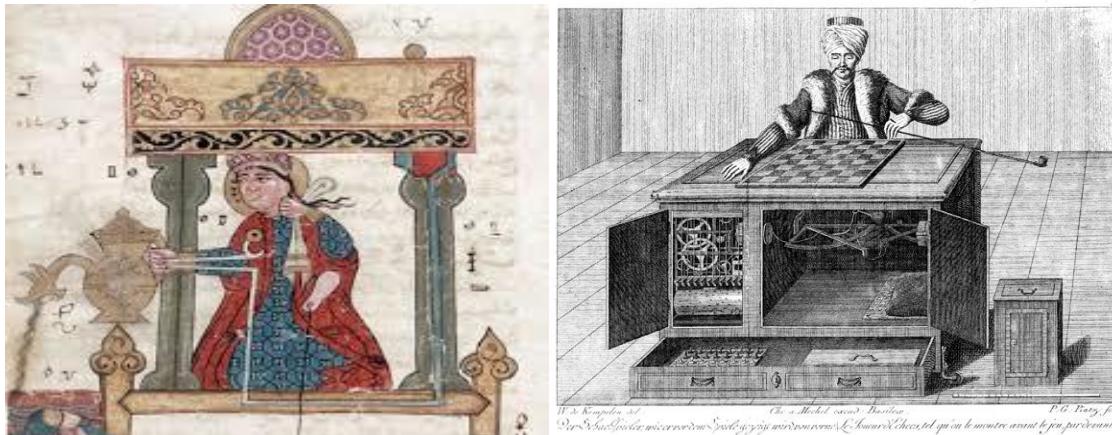
وفي اليونان - في القرن الرابع ق.م. - اخترع أركيتاس عالم الرياضيات، حمامه آلية يمكنها الطيران.

وفي القرن الثالث قبل الميلاد، اخترع ستيسبيوس العديد من الأجهزة الآلية ومنها ساعة مائية مزودة بجهاز يجعل مستوى المياه ثابتًا.

وعبر القرون التالية، ظهرت مخترعات رائعة في الشرق الأقصى والأوسط، في الصين، وفي الهند وفي اليابان وفي الجزيرة العربية، وفي كتاب رسالة الجزمي الذي يتضمن سرداً للأجهزة الآلية التي اخترعها العرب وصفاً لأحد هذه الأجهزة والتي أطلق عليها نافورة الطاووس التي كانت

تستخدم لغسل الأيدي، فتقدم المياه والصابون والمنشفة آلياً، وبسبب هذا الاختراع يطلق على الجزي "أبي الإنسان الآلي".

في أوروبا في القرون الوسطى، اهتم الفيلسوفان ألبرت فاجنوس وروجر باكون اهتماماً كبيراً بالآلات ذاتية الحركة، بل وصنعاً البعض منها.



الشكل (1-1-1) الروبوتات خلال القرون الوسطى.

في القرن الثامن عشر، أنتج صناع اللعب عدداً كبيراً من الآلات ذاتية الحركة والتي كانت في شكل الإنسان ويمكنها الكلام وعزف الموسيقى والكتابة وحتى لعب الشطرنج، ومن أشهر المخترعين لهذه اللعبة رجل فرنسي اسمه جاك دي فوكاسون الذي صمم نولا نسيجياً آلياً.

في القرن الثامن عشر تم اختراع جهازان يعملان بمبدأ التغذية العكسية (Feedback) التي تعتبر شرطاً أساسياً لنظم المراقبة الآلية ذاتية التغذية، وهذان الجهازان الآليان هما مروحة الطاحونة الهوائية التي تبقي الريش متوجهة نحو الريح ومن ثم تستمر الطاحونة الهوائية في الدوران، أما الجهاز الثاني فكان المنظم والمتحكم الآلي للمحرك البخاري وهو الذي يجعله مستمراً في الدوران بسرعة ثابتة.

تقدمت الروبوتات في القرنين العشرين والحادي والعشرين بشكلٍ كبير لتشمل الآلات القادرة على تجميع آلات أخرى، وامتدت هذه الروبوتات إلى اليابان وكوريا الجنوبية وأجزاء كثيرة من أوروبا على مدار نصف القرن الماضي لتجد طريقها إلى مجالات الحياة المختلفة.



الشكل (1-1-2) الروبوتات خلال القرون (21-18).

2-2- تصنیف الروبوتات Classification of Robots

تختلف الروبوتات الموجودة حول العالم بحسب وظائفها والمهام الموكلة إليها فمنها ما يسير على الأرض أو يطير في السماء وأخرى تأخذ شكل أذرع مثبتة لا تتحرك وغيرها الكثير من الأنواع والأشكال، الشكل (1-2-1)، وبشكل عام تصنف الروبوتات نسبة إلى ثلاثة محاور أساسية وهي:

- تكوينها المادي Physical Configuration
- قاعدة الروبوت Robot Base
- أنظمة التحكم Control System



الشكل (1-2-1) أشكال مختلفة للروبوتات.

تصنيف الروبوتات حسب تكوينها المادي :Physical Configuration

- 1- تكوين ديكاري Cartesian configuration
- 2- تكوين أسطواني Cylindrical configuration
- 3- تكوين قطبي Polar configuration
- 4- تكوين الذراع والمفاصل Joint-arm configuration

تصنيف الروبوتات حسب قاعدة الروبوت Robot Base

1- الروبوتات الثابتة Fixed Robots

ومن الأمثلة عليها الروبوتات المستخدمة في المجال الصناعي والتي لا يمكن نقل قاعدتها بعيداً عن موقع العمل الجاري، الشكل (2-2-1).



الشكل (1-2-2) ذراع روبوت صناعية.

2- الروبوتات النقالة Mobile Robots

في هذا النوع من الروبوتات عادة ما تكون قاعدة الروبوت عبارة عن منصة بها عجلات أو مسارات متقللة لتساعدها على الحركة، ومن الممكن أيضاً وصل ذراع ثابتة على نفس المنصة، الشكل (3-2-1).



الشكل (1-2-3) روبوت نقال مع ذراع ثابتة على سطحه.

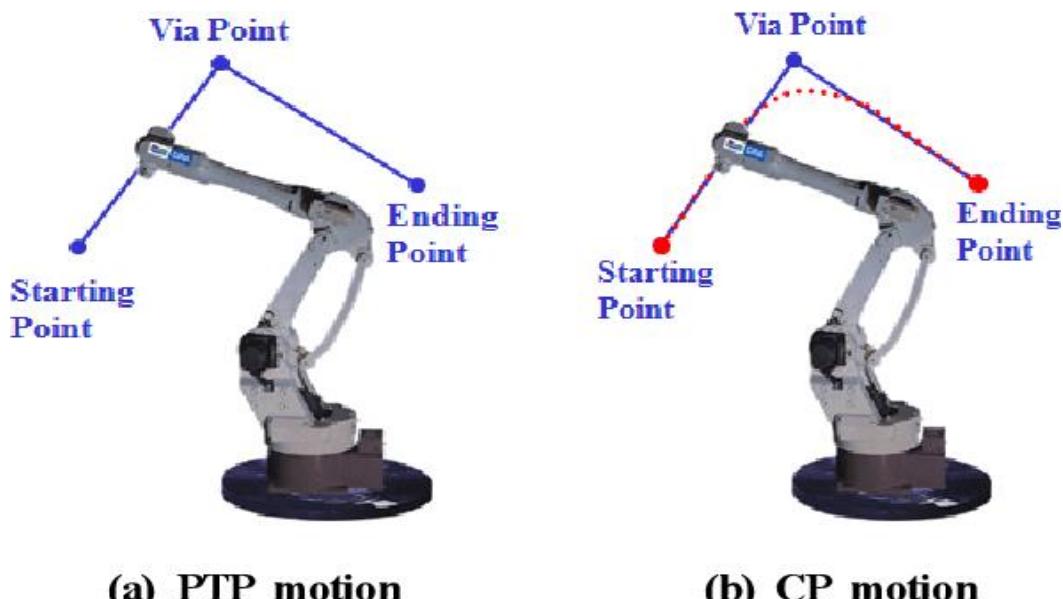
تصنيف الروبوتات حسب نظام التحكم :Control System

1- الروبوتات التي تعمل بنظام تحكم Point to Point System (PTP)

الروبوتات التي تعمل وفق هذا النظام تنتقل من نقطة الى نقطة أخرى حيث تسجل هذه النقاط في ذاكرة المتحكم، وتستخدم هذه الروبوتات في تطبيقات كالتعبئة والتغليف والتجميع.

2- الروبوتات التي تعمل بنظام المسار المستمر Continuous-path (CP)

تقوم هذه الروبوتات بتخزين عدة نقاط في الذاكرة وتنقل مروراً بها في مسار مستمر، ويمكنها التوقف عند أي نقطة من هذه النقاط ومن ثم استكمال حركتها، وتستخدم هذه الروبوتات في رش الطلاء واللصق الحراري وعمليات اللحام.



الشكل (4-2-1) حركة روبوت (PTP) وروبوت (CP).

3- روبوتات المسار المتحكم فيه Controlled-path robot

في هذا النوع يتم تخزين نقطة البداية ونقطة النهاية وتعريف المسار فقط في الذاكرة، ويولد المتحكم مسارات مختلفة لحركة الروبوت مثل الخطوط المستقيمة والدوائر والمنحنيات، وتكون الحركة بدقة عالية في جميع نقاط المسار، بالإضافة إلى أن هذا النوع له القدرة على تصحيح المسار في حال وجود الأخطاء.

3-1 التحكم في الروبوتات Robot Control

تم استخدام الروبوتات بشكل فعال للعمل في القطاعات الصناعية، حيث أنها شكلت أكثر تكنولوجيات الأتمتة الموجودة في وقتنا الحالي ذكاءً ومورونة، وكما حدث في السنوات الأخيرة، فقد أصبحت أنظمة التصنيع أكثر استقلالية، فهي بذلك بحاجة إلى تدخل أقل من البشر المشغلين لها ودرجة أعلى من التخصيص والتكون لتطبيقات واستخدامات أكثر تشابه، حيث تعد برمجة الروبوت عاملًا رئيسيًا في بناء التطبيقات المختلفة للمصانع المستقبلية، وبالتالي ستعتمد إعدادات التصنيع هذه بشكل كبير على التكنولوجيا المتقدمة، وتكنولوجيات الذكاء الاصطناعي، والروبوتات الذكية وطرق التحكم بها، يتم التحكم في الروبوت بعدة طرق منها:

1- التحكم في الروبوت بواسطة الكمبيوتر Controlling Robot with Computer

التحكم بالروبوت من خلال أجهزة الحاسوب، حيث سيتمكن الإنسان من التحكم المباشر في هذا الروبوت، وذلك من خلال إعطائه أوامر ومهام مباشرة له من خلال أجهزة الحاسوب.

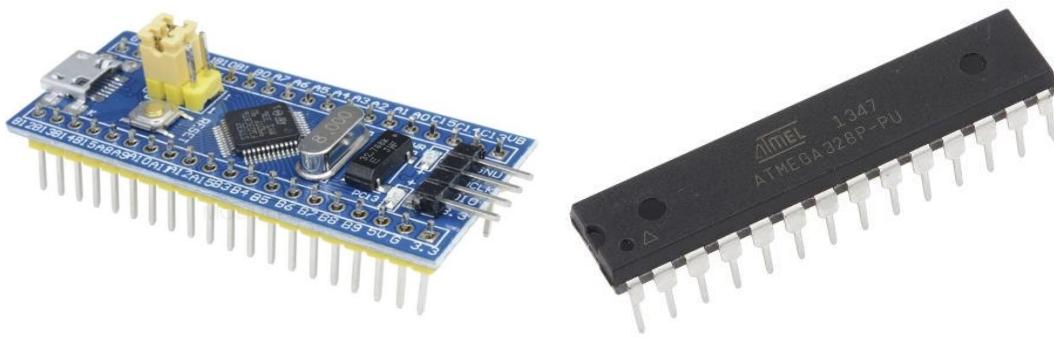


الشكل (1-3-1) التحكم في الروبوت بواسطة الكمبيوتر

2- التحكم في الروبوت بواسطة المتحكمات المصغرة Controlling Robot with Microcontrollers

في هذا النوع يتم التحكم في الروبوت فيها عن طريق متحكمات صغيرة، وهي عبارة عن دارة إلكترونية مدمجة جاهزة للاستخدام.

يتم تركيبها على الروبوت وتوصيلها بالحساسات والمشغلات في دارات كهربائية، ويتم تحميل الكود البرمجي الذي يريد المستخدم، لتعمل بعد ذلك بشكل ذاتي دون أي تدخل بشري آخر، ومن أشهر هذه المتحكمات `STM32F103C8T6` ، `ATmega328`

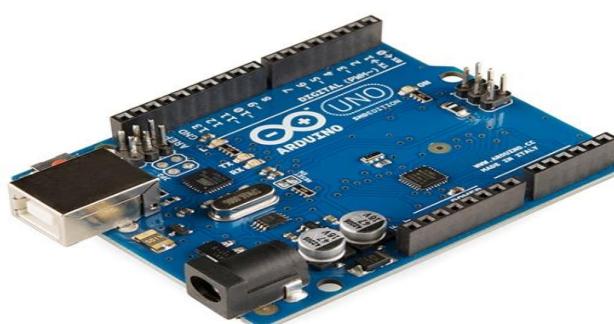


الشكل (3-1) أشهر المتحكمات المصغرة

3- التحكم في الروبوت بواسطة الاردوينو **Controlling Robot with Arduino**

حيث يعمل بنفس الطريقة السابقة وله مميزات وخصائص كثيرة، ومن أهم مميزاته المفيدة في علم الروبوتات سهولة وبساطة التعامل معه بصورة أكبر من المتحكمات المصغرة، وكما يتميز كذلك بكونه مفتوح المصدر فيمكن التعديل والتطوير عليه، حيث يمكن استخدام نصوص برمجية جاهزة قد تجدها بسهولة على الإنترنت، لأنه مفتوح المصدر لذلك يستخدمه المبرمجين كثيراً في المشاريع المختلفة لهم.

في هذا المشروع سيتم استخدام الاردوينو بجانب كاميرا Kinect للتحكم بروبوت متelligent.



الشكل (3-1) لوحة الاردوينو

4- التحكم في الروبوت بواسطة جهاز التحكم المنطقي القابل للبرمجة **Controlling Robot with Programmable Logic Controller (PLC)**

يمكن التحكم في الروبوت من خلال جهاز التحكم المنطقي القابل للبرمجة، المعروف اختصاراً بـ(PLC)، وهو عبارة عن وحدة مدمجة كذلك بدارات داخلية جاهزة، يتم توصيل الحساسات والمشغلات بها، كما ويتم برمجته بإحدى لغات البرمجة الخاصة بكل شركة منتجة له، مثل شركة (SAMSUNG) و(LG) وكل شركة منها لها جهازها المختلف وطريقة البرمجة الخاصة بها.



الشكل (4-3-1) PLC من شركة LG

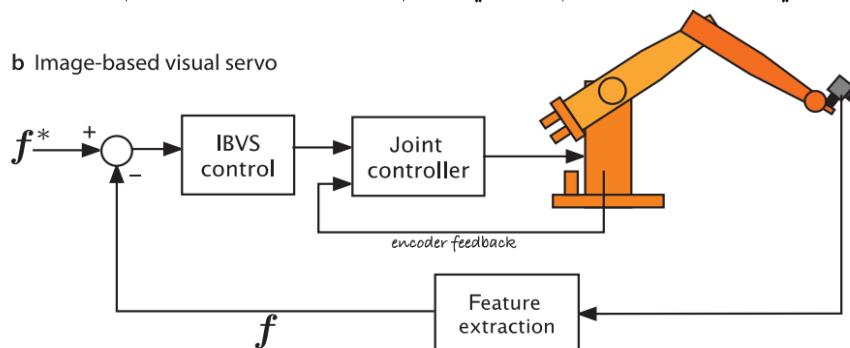
لا يكون الاختلاف بينها بشكل كلي ولكن في طريقة كتابة الكود او في المدخل والمخرج، وهذه الطريقة للتحكم تجدها دائمًا مع الروبوتات الصناعية، أو يتم استخدامها لمجموعة من الروبوتات التي تشغّل مصنع بأكمله.

1-3-1 التحكم القائم على الرؤية

يعد التحكم القائم على الرؤية البصرية في الوقت الحالي طريقة فعالة للتحكم في الروبوتات باستخدام الميزات البصرية المأخوذة من الصورة في العالم الحقيقي، يمكن اعتبار هذه الطريقة اندماجاً لرؤية الكمبيوتر والروبوتات والتحكم، ويمكن تصنيف هذه التقنية إما على أساس الموقع أو على أساس الصورة، اعتماداً على ما إذا كانت وضعية الكاميرا أو ميزات الصورة تستخدم في قانون التحكم العام.

الغاية من هذه التقنية هو التحكم في وضع النهاية المنفذة End-Effector للروبوت بالنسبة للهدف، حيث يمكن تثبيت الكاميرا على الروبوت أو وتنبيتها في أحد النقاط.

سوف نستعرض في هذا البحث التحكم المرئي القائم على الصورة باستخدام كاميرا Kinect.



الشكل (1-3-1) يوضح مبدأ عمل التحكم المرئي القائم على الصورة.

4-4- نظم الروبوتات Robots System

أنظمة الروبوت هي طريقة لأتمتة التطبيقات الصناعية وتقليل كمية العمالة وتوفير المال والوقت المرتبطان بالعملية الصناعية، هذه الطريقة أحدثت ثورة في المجال الصناعي ليصبح المصنعون الآن قادرين على إنتاج منتج عالي الجودة في فترة زمنية قصيرة وتكلفة منخفضة، تقسم أنظمة الروبوت بشكل عام إلى ثلاثة أقسام رئيسية وهي أنظمة الروبوت للمعالجة وأنظمة الروبوت للحصول على البيانات وأنظمة الروبوتات المتنقلة.

1-أنظمة الروبوت للمعالجة Manipulation Robotic System

هو الأكثر استخداماً في مجال التصنيع، يتكون من العديد من أذرع الروبوتات الصناعية ذات الأربع إلى ست محاور (axes 6-4) ودرجات متقاومة من الحرية (DoF) وتتمكن من أداء الكثير من الوظائف المختلفة كاللحام وتطبيقات مناولة وإزالة المواد.

2-أنظمة الروبوت للحصول على البيانات والتحكم بها Data Acquisition and Control Robotic System

تستخدم هذه التقنية لجمع المعلومات والبيانات ومعالجتها وتحويلها إلى مجموعة متنوعة من الإشارات، يمكن للعديد من أنظمة الروبوت المتنقل استخدام الإشارات المعالجة من هذه التقنية، كما أنها تستخدم في البرمجيات في مجالات الهندسة والأعمال.



الشكل (1-4-1) أنظمة الروبوت للحصول على البيانات.

1-4-1 أنظمة الروبوتات المتنقلة Mobile Robotic System

تمثل القسم الثالث من أنواع أنظمة الروبوت وهي مختلفة قليلاً عن الأنظمة السابقة، يتكون هذا النظام بشكل عام من منصة تقوم بنقل العناصر من مكان إلى آخر، الشكل (1-4-2)، ويستخدم بكثرة في المجالات الصناعية لحمل الأدوات وقطع الغيار وغيرها، يستخدم أيضاً في المجال الزراعي لنقل المنتجات ويمكن استخدامه في الكثير من المجالات لقدرته على السير على الأرض أو الطيران أو حتى السباحة.



الشكل (1-4-1) أنظمة الروبوت المتنقلة.

الفصل الثاني

كاميرا Microsoft Kinect

1-2- نبذة عن الكاميرا Microsoft Kinect

في الأول من يونيو عام 2009 أعلنت شركة مايكروسوف特 عن مشروع كاميرا لألعاب الفيديو خاصة بأجهزة (Xbox, Windows PC) تحت اسم (نثال)، وفي الثالث عشر من نفس الشهر عام 2010 تم إطلاق الجهاز بشكل رسمي وتم تغيير الاسم إلى (Kinect)، الشكل (1-1-2).

حقق الجهاز مبيعات كبيرة وصلت إلى ثمانية ملايين قطعة بعد إطلاقها بستين يوم فقط وبهذا دخل الجهاز موسوعة غينيس للأرقام القياسية كأكثر جهاز كتروني في التاريخ تحقيقاً للمبيعات في وقت قياسي.

تعتبر كاميرا Kinect أول جهاز استشعار يسمح للمستخدم بالتفاعل مع العاب الفيديو من خلال واجهة المستخدم الطبيعية (باستخدام الإيماءات وأوامر الصوت بدلاً من عصا التحكم بالألعاب).

كاميرا Kinect هي جهاز طرفي مصمم لأجهزة (Xbox, Windows PC) وظيفته مشابهة لوظيفة كاميرا الويب ولكن ما يميز مستشعر Kinect هو انه يقدم خريطة عمق الصورة بالإضافة إلى الصورة الملونة، هذه الميزة بالإضافة إلى سهولة ربط الكاميرا مع الحاسوب الشخصي مكنت الباحثين والمطورين من استخدام المستشعر في التعليم والبحث العلمي والأمنية المنزلية والتكنولوجيا الصحية ومعالجة الصورة ومجال الروبوتิก.



الشكل (1-1-2) Microsoft Kinect

2-2- بنية الكاميرا Microsoft Kinect

تتكون كاميرا Kinect من كاميرا RGB ومستشعر العمق الذي يحتوي على جهاز عرض ليزر بالأشعة تحت الحمراء بالإضافة إلى مستشعر الأشعة تحت الحمراء CMOS وميكروفون متعدد المصفوفات له القدرة على تثبيت المصدر الصوتي وقمع الضوضاء المحيطة، وت تكون أيضاً من ثنائي صوتي LED ومقاييس تسارع ثلاثي ومحرك سيرفو Servo Motor للتحكم في ميل الجهاز، الشكل (1-2-2).



الشكل (1-2-2) المكونات الداخلية للكاميرا.

أطلقت شركة مايكروسوف特 لاحقاً عدة إصدارات للكاميرا تحتوي على اختلافات بسيطة في الشكل الخارجي أو دقة الصورة أو في نوع مأخذ USB ، وفي الإصدار الذي تم دراسته في هذا البحث لدينا مأخذ من نوع USB-B وهو مأخذ لا يمكن وصله بشكل مباشر مع جهاز الحاسب لعدم قدرة منفذ USB-A الموجود في الحاسب على توليد الطاقة الكافية للكاميرا، وبالتالي تحتاج إلى تغذية الكاميرا من مصدر طاقة آخر وتحويل USB-B إلى USB-A، الشكل (2-2-2).



الشكل (2-2-2).

مميزات ومواصفات كاميرا Microsoft Kinect

يوضح الجدول التالي، الشكل (3-2-2)، اهم المميزات التي تتمتع بها كاميرا Kinect :

Microsoft Kinect كاميرا	الميزة
640x480 30fps / 1280x960 12fps	دقة الصورة الملونة
640x480 30fps	دقة الصورة ذات الأشعة تحت الحمراء
640x480 30fps / 320x240 30fps 80x60 30fps	دقة مستشعر العمق
43° V - 57° H	مجال الرؤية
(الوضع القريب) 0.4-3m (الوضع الافتراضي) 0.8-8m	مجال مستشعر العمق
يمكن تتبع هيكلين وتتبع 20 مفصل في كل هيكل	تقنية تتبع الهيكل العظمي
USB Port 2.0	مأخذ USB

الشكل (3-2-2) اهم المواصفات والمميزات.

3- برمجة الكاميرا و معالجة الصورة Image Processing

بعيداً عن استخدام كاميرا Kinect في مجال العاب الفيديو وبعد أن تمكن المطورين والباحثين من إيجاد استخدامات أخرى أكثر أهمية لهذا الجهاز، استطاع الباحثين تطوير بيئة برمجية بسيطة تقوم بمعالجة الصور والتفاهم مع المعلومات التي تقدمها الكاميرا وترجمتها وتحويلها إلى تعليمات وأوامر برمجية، هذه الأوامر يتم ارسالها إلى متحكم صغير ليقوم بتنفيذها.

تكمن أهمية هذه البرمجيات في أنها تعمل عمل المترجم الذي يترجم حركة المستخدم وصوته إلى أوامر يتم تنفيذها على ارض الواقع، وهي أيضاً تفتح الباب أمام العديد من التطبيقات الأخرى كـ تتبع الهيكل العظمي لجسم الإنسان وبصمة اليد والتعرف على قزحية العين وتحليل موجات الصوت.

هناك عدة برامج تمكنا من التعامل مع الصور والمعلومات المستقبلة من الكاميرا ومعالجتها، الشكل (4-2-2)، ومن أهمها:

Processing IDE –

هي لغة برمجة مفتوحة المصدر وبيئة مناسبة للأشخاص الذين يرغبون بإنشاء الصور والرسوم المتحركة. في البداية تم تطور هذه اللغة لتكون بمثابة كتيب رسم للبرمجيات Sketchbook وللتدريس أساس برمجة الكمبيوتر ضمن سياق بصري، وقد تطورت هذه اللغة لتصبح أداة مهمة في العمل المهني، وفي هذه الأيام أصبح عشرات الآلاف من الطلاب والباحثين والمصممين والهواة يستخدمون هذه اللغة للتعلم والتطوير وإنشاء النماذج الأولية والابتكار. في هذا المشروع سيتم العمل على هذه اللغة لكونها لغة بسيطة غير معقدة وتطبيقاتها مفيدة لهذا البحث وأيضاً لأنها لغة مفتوحة المصدر.

Microsoft Kinect SDK –

بعد نجاح المخترقين في إيجاد وسيلة لاستخدام الكاميرا Kinect في تطبيقات بعيدة عن مجال العاب الفيديو وتمكنوا من تحقيق ثورة في عالم الالكترونيات البصرية، اثار هذا النجاح شركة مايكروسوفت المصنعة للكاميرا وبدلاً من محاربة المخترقين ومقاضاتهم قامت الشركة بفتح الباب لهم لمشاركة ابحاثهم وتطويرها، وعندها قامت الشركة بإصدار البرنامج الرسمي الخاص ببرمجة الكاميرا على نظام التشغيل Windows. لم يتم استخدام هذا البرنامج في مشروعنا بسبب انه لا يعطي الحرية الكاملة للباحث في استخدام التطبيقات المختلفة للكاميرا ولكونه أكثر تعقيداً من البرنامج السابق.

Visual Studio –

بيئة تطويرية مشابهة لـ Processing IDE تدعم عدة لغات منها: Java، Python، C++، C#، MATLAB و هناك برامج أخرى يمكن استخدامها ك برنامج Html.



الشكل (2-2-4) برامج معالجة كاميرا Kinect.

4-2- تطبيقات و استخدامات الكاميرا Microsoft Kinect Application

إلى جانب التطبيق الأساسي الذي تقدمه كاميرا Kinect وهو التفاعل مع العاب الفيديو، وبعد عدّة أبحاث استطاع أفضل المطورين استخراج تطبيقات متقدمة بفضل خصائص هذه الكاميرا، ومن هذه التطبيقات، الشكل (2-2-5):

-التجسيد ثلاثي الأبعاد 3D-Reconstruction

تتيح كاميرا Kinect إمكانية إعادة تجسيد ثلاثة الأبعاد لجسم ما، وذلك باستخدام بيانات مستشعر العمق Depth Map الذي يقوم بتتبع الاحداثيات ثلاثة الأبعاد للجسم وإعادة بناء نموذج جيوميتري دقيق للجسم الفيزيائي. يمكن استخدام هذا التطبيق في المجال الطبي وهندسة الطرق والأرصفة وفي التطبيقات ذات البيئة الافتراضية.

-التعرف على لغة الإشارة Sign Language Recognition

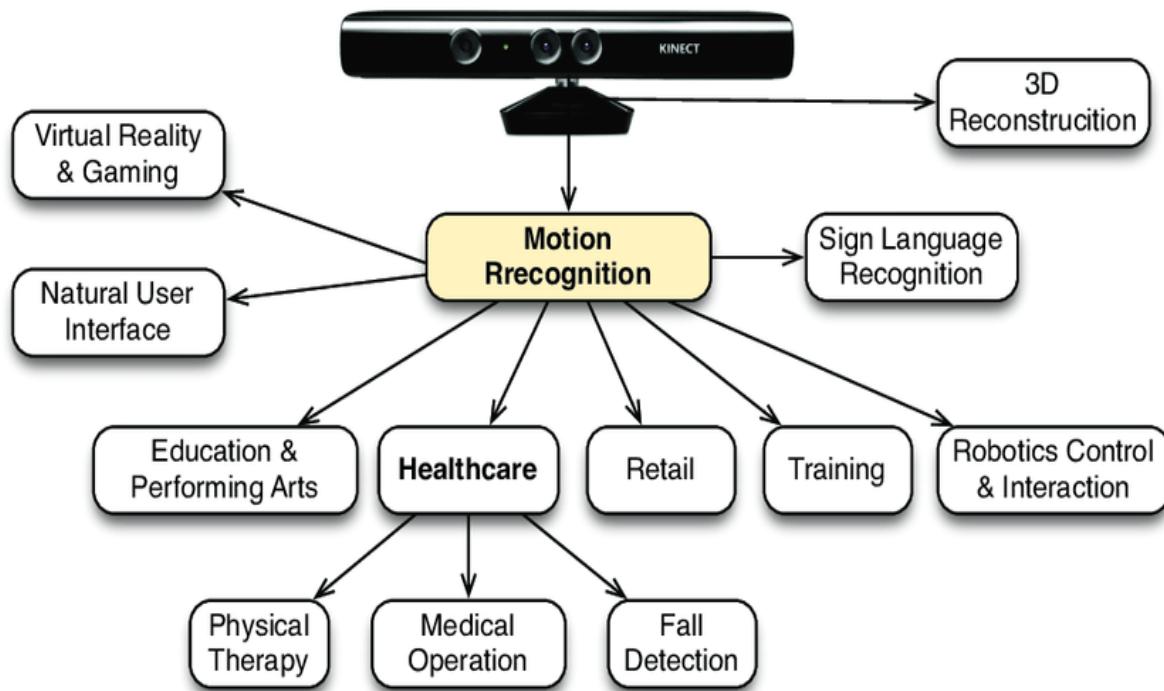
يمكن استخدام الكاميرا Kinect كوسيلة ترجمة تعمل على استشعار الحركات في لغة الإشارة وتحويلها إلى نصوص مفهومة وفق خوارزمية معينة.

الرعاية الطبية HealthCare

من أهم تطبيقات الكاميرا Kinect هي تطبيقات المجال الطبي، وبفضل خاصية مستشعر العمق الموجودة في الكاميرا يمكن استخدام الكاميرا في الكثير من المهام الطبية مثل العلاج الطبيعي والعمليات الجراحية.

التعليم والفنون المسرحية Education and Performing Arts

واجهة المستخدم الطبيعية التي تم تطويرها بواسطة الكاميرا Kinect تتيح الفرصة للطلاب للوصول إلى مستوى تعليمي عالي وفي نفس الوقت يوفر طرق تدريسية تفاعلية وجديدة، وفي مجال الفنون اقترح نظام موسيقي تفاعلي يقوم بتنظيم الموسيقى وفقاً لحركة يد المستخدم، وفي نفس المجال تم تطوير تطبيق لتحريك دمى الظل عن طريق تتبع حركة الجسم الحقيقي ومحاكاتها بدلاً من سحب الخيوط أو القصبان.



الشكل (2-2) تطبيقات واستخدامات كاميرا Kinect .

1-2-4- التحكم في حركة الروبوتات باستخدام الكاميرا Microsoft Kinect

من التطبيقات الهامة التي تقدمها كاميرا Kinect هو استخدامها في التحكم بالروبوتات، حيث يتم ذلك بواسطة المستشعرات الموجودة في بنية الكاميرا كمستشعر العمق والصورة او يتم التحكم من خلال موجات الصوت.

بعد هذا التطبيق واسع وشامل لعدة استخدامات ويقبل طرق عديدة يمكن من خلالها التحكم بحركة الروبوتات المختلفة، من الأمثلة على استخدام كاميرا Kinect في مجال الروبوتات:

-استخدام الكاميرا للتحكم بالروبوتات عن بعد

يتم ذلك عن طريق استشعار إشارات المستخدم او استشعار الأمواج الصوتية وفهم هذه الإشارات وتحويلها الى أوامر للروبوت لكي يقوم بمحاكاتها على ارض الواقع، الشكل (7-2-7).



الشكل (7-2-7) استخدام الكاميرا للتحكم في الروبوتات عن بعد.

-روبوتات تجنب العائق Obstacle Avoiding Robot

يتم ربط الكاميرا مع الروبوت وتقوم الكاميرا باستشعار العقبات وارسال البيانات الى المتحكم ثم الى الروبوت ليقوم بتجنب العقبة.

–استخدام الكاميرا في ملاحة الروبوت Navigation Robot Using Kinect

في هذا التطبيق يتم استخدام كاميرا Kinect كمستشعر رئيسي للروبوت حيث يسمح للروبوت بالتنقل بأمان وتجنب العقبات وتحطيم الطريق الأفضل باستخدام بيانات مستشعر واخذ النقاط ثلاثية الابعاد، الشكل (2-8) يوضح مثال على هذا التطبيق، وهنا يمكن استبدال الحاسب المحمول بمحكم صغير او الكمبيوتر المصغر Raspberry Pi.



الشكل (2-8) استخدام الكاميرا في نظم الملاحة.

الفصل الثالث

الاردوينو Arduino

3-1- لمحة عن المتحكمات المصغرة Microcontrollers

المتحكم المصغر عبارة عن شريحة إلكترونية متكاملة يمكن النظر إليها على أنها حاسوب صغير، وهذا يعني أن المتحكم المصغر يحتوي على كافة العناصر الأساسية للنظام الحاسوبي أي يحتوي على وحدة معالجة مركبة CPU، ذواكر Memory Units، وحدات إدخال وإخراج البيانات I/O، بالإضافة لطيفيات أخرى تقدم وظائف متعددة للمتحكمات المصغرة، تعتبر المتحكمات المصغرة شبيهة جداً بالأنظمة المدمجة على شريحة (System on Chip)، إلا أن الأنظمة المدمجة على شريحة تعتبر أكثر شمولية، حيث تحتوي الكثير من هذه الأنظمة على متحكم مصغر ضمنها بالإضافة لوحدات وظيفيات أخرى تساعد على القيام بمهام أكثر تعقيداً، كمثال على الأنظمة المدمجة ضمن شريحة هو شرائح المعالجة الخاصة بالهواتف الذكية، الاستخدام الأساسي للمتحكمات المصغرة هو في الأجهزة والمنتجات المتحكم بها تلقائياً، أي التي لا يتطلب عملها وجود تحكم مباشر من المستخدم، بل تعمل من تلقاء نفسها اعتماداً على برنامجها أو نظام التشغيل المزود به، الأمثلة على مثل هذه الأجهزة والمنتجات هي ما يعرف اختصاراً بالأنظمة المدمجة Embedded Systems، مثل أنظمة التحكم الإلكترونية في محركات السيارات والطائرات، الأنظمة الإلكترونية المشغلة للأجهزة الطبية المزروعة، الأنظمة الإلكترونية في الأجهزة الطبية المحمولة ذات استهلاك الطاقة المنخفض، الأنظمة الإلكترونية المستخدمة في تطبيقات قياس الحرارة، الرطوبة، قراءة المعلومات البيومترية (بصمة اليد، بصمة العين...).

تقوم اليوم الكثير من الشركات بصناعة شرائح المتحكمات الصغيرة، وتعتمد بعض الشركات على تصميم بنى خاصة بها بينما تقوم العديد منها باعتماد بنى مصممة من قبل شركات أخرى، ولا يوجد اليوم شركة واحدة تسيطر على سوق المتحكمات الصغيرة، بل تأخذ المنافسة شكلًا كبيرًا بوجود العديد من المصنعين في السوق، وأبرز هذه الأسماء في الوقت الحالي هي شركة Microchip وشركة STMicroelectronics.



الشكل (3-1-3) أشهر المتحكمات المصغرة.

2-3- مقدمة عن الاردوينو Introduction to Arduino

هي عبارة عن لوحة تطوير إلكترونية تتكون من دارة إلكترونية مفتوحة المصدر مع متحكم دقيق على لوحة واحدة يتم برمجتها عن طريق الكمبيوتر وهي مصممة لجعل عملية استخدام الإلكترونيات التفاعلية في مشاريع متعددة التخصصات أكثر سهولة، ويستخدم الاردوينو بصورة أساسية في تصميم المشاريع الإلكترونية التفاعلية أو المشاريع التي تستهدف بناء حساسات بيئية مختلفة (مثل درجات الحرارة، الرياح، الضغط...) ويمكن توصيل الاردوينو ببرامج مختلفة على الحاسوب الشخصي، وتعتمد الاردوينو في برمجتها على لغة البرمجة مفتوحة المصدر Processing، وتتميز النصوص البرمجية الخاصة بلغة الاردوينو أنها تشبه لغة (C++) وتعتبر من أسهل لغات البرمجة المستخدمة في كتابة برامج المتحكمات الدقيقة.

يتكون الاردوينو من متحكم صغير Microcontroller وهو يمثل القلب النابض للوحة الاردوينو بالإضافة إلى بعض الدوائر التي تساعد المتحكم المصغر في العمل مثل دوائر الطاقة Power، ودائرة الـ Programmer وبعض الدوائر التي تضمن عمله بشكل صحيح Supply.



.Arduino Board (1-2-3)

يستخدم الاردوينو من قبل المصنعين والمصممين وطلاب وخريجي الهندسة لتطبيق وتجربة أفكارهم ومشاريعهم بشكل سهل وسريع، ويعود الاردوينو من الطرق السهلة والبسيطة حقاً لتعلم كيفية برمجة المتحكم المصغر للتفاعل مع العالم الخارجي والتجاوب معه حتى عبر الانترنت.

Arduin Boards - 1-2-3

يوجد أكثر من 40 نوع من ألواح الاردوينو Arduino Boards، تختلف في القدرات والشكل والحجم والثمن حتى تتناسب مع جميع الأفكار والتصميمات، لكن أهمها وأكثرها انتشارا هي:

Arduino UNO-

يعتبر من أشهر أنواع الاردوينو وأكثرها انتشارا واستخداما، تحتوي لوحة Arduino Uno على منفذ لوصلة USB والتي تستطيع من خلالها تزويد الاردوينو بالطاقة وتحميل الكود البرمجي على المتحكم المصغر ومنفذ آخر لتزويد اللوحة بطاقة خارجية منفصلة مثل البطاريات ذات الـ 9 فولت، كما تحتوي هذه اللوحة على زر إعادة التشغيل، وسوف نقوم باستخدام هذا النوع من الأردوينو في مشروعنا، يوضح الجدول التالي، الشكل (1-1-2-3)، أهم الميزات والخصائص التي تتمتع بها هذه اللوحة.

المتحكم	ATmega328P
توتر الاستعمال	5V
التوتر الموصى به	7-12V
مخارج الديجيتال (0 او 5 فولت)	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
تيار الاستعمال لقطب 5 فولت	20 mA
تيار الاستعمال لقطب 3.3 فولت	50 mA
ذاكرة الفلاش	32 KB (ATmega328P) of which 0.5 KB used by bootloader
ذاكرة SRAM	2 KB (ATmega328P)
ذاكرة EEPROM	1 KB (ATmega328P)
سرعة المعالج	16 MHz
الطول	68.6 mm
العرض	53.4 mm
الكتلة	25 g

الشكل (1-1-2-3) مميزات وخصائص Arduino UNO

Arduino Nano-

له نفس المتحكم المستخدم في لوحة Arduino UNO اي لها قدرات متشابهة ولكن ما يميز التوfer هو حجمه الصغير.

فيما يلي جدول يوضح مميزات لوح Arduino Nano، الشكل (2-1-2-3).

المتحكم	Atmel ATmega168 or ATmega328
توfer الاستعمال	5V
التوfer الموصى به	7-12V
الفولتية الحدية	6-20V
مخارج الديجيتال (0 او 5 فولت)	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	8
تيار الاستعمال لقطب 5 فولت	20 mA
ذاكرة الفلاش	16 KB (ATmega168) or 32 KB(ATmega328)
ذاكرة SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
ذاكرة EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
سرعة المعالج	16 MHz
الطول	45 mm
العرض	18mm
الكتلة	5 g

الشكل (2-1-2-3) مميزات وخصائص Arduino Nano

Arduino Mega-

يعتمد على المتحكم ATmega2560، ما يميزه أنه يملك أكبر ذاكرة بين أنواع الاردوينو الأخرى وعدد أكبر من المداخل والمخارج، يعتبر أفضل وأكبر نوع من أنواع الاردوينو، وهو مخصص للمشاريع الكبيرة.

ويوضح الجدول التالي، الشكل (3-1-2)، اهم مميزات Arduino Mega .

المتحكم	ATmega2560
توتر الاشتغال	5V
التوتر الموصى به	7-12V
الفولتية الحدية	6-20V
مخارج الديجيتال (0 او 5 فولت)	54 (of which 15 provide PWM output)
PWM Digital I/O Pins	15
Analog Input Pins	16
تيار الاشتغال لقطب 5 فولت	20 mA
تيار الاشتغال لقطب 3.3 فولت	50 mA
ذاكرة الفلاش	256 KB of which 8 KB used by bootloader
SRAM ذاكرة	8 KB
EEPROM ذاكرة	4 KB
سرعة المعالج	16 MHz
الطول	101.52 mm
العرض	53.3 mm
الكتلة	37 g

الشكل (2-1-2-3) مميزات وخصائص Arduino Mega .

ويوجد انواع أخرى مختلفة من الاردوينو وكل منها لديها خصائص وقدرات تميزها عن غيرها ومنها Arduino Leonardo ، Arduino Pro Mini ، LilyPad Arduino . الشكل (3-1-2-3).

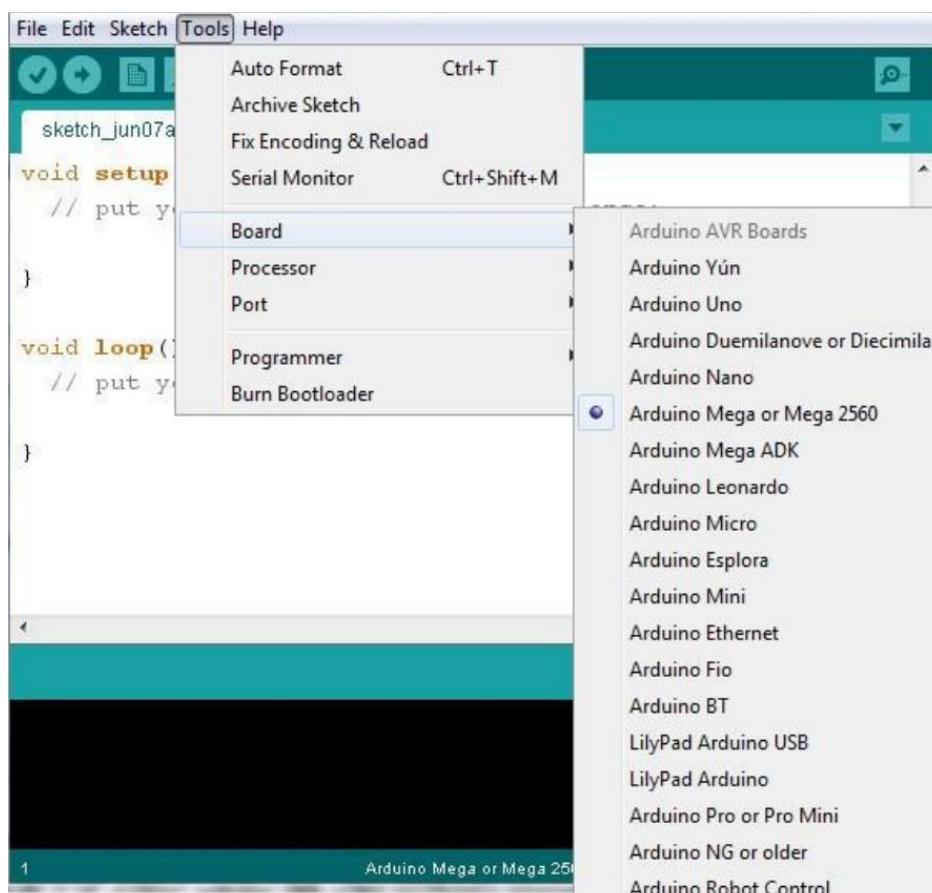


الشكل (3-1-2-3) الأنواع المختلفة من الاردوينو .

2-2-3- برمجة الاردوينو Arduino Programming

للبدء في برمجة وتشغيل الاردوينو يجب في البداية تنصيب بيئة التطوير Arduino IDE، وهو تطبيق عبر النظام الأساسي مكتوب بلغة JAVA وهو مشتق من لغة Processing IDE التي تحدثنا عنها سابقا.

اللغة المستخدمة في كتابة الكود البرمجي للاردوينو هي لغة C/C++، ومن ثم تتولى بيئة IDE Arduino مسؤولية ترجمة الكود وحقنه في المتحكم لذلك فهي تومن مرونة في التعامل مع المتحكمات حيث أنها مدعومة بالمكتبات الجاهزة التي تخدم معظم أنواع الحساسات والمشغلات، وتدعم هذه البيئة معظم لوحة الاردوينو حيث يمكننا ان نختار نوع اللوحة التي نعمل بها من خلال التطبيق، الشكل (1-2-2-3).



الشكل (3-2-2) طريقة اختيار لوحة الاردوينو التي نريد العمل عليها.

عند بناء أي مشروع في بيئة Arduino IDE فإن هناك تابعين أساسيين يتم استدعائهما بشكل افتراضي للسماح بعمل برنامج قابلة للتشغيل والاستخدام، التابع الأول (`setup()`) يتم استدعاء هذا التابع لمرة واحدة فقط وهي عند تشغيل البرنامج، يستخدم هذا التابع لتعريف المتغيرات ولتحديد عمل الأقطاب كمداخل أو مخارج ولتشغيل المكتبات.

التابع الثاني (`loop()`) يتم استدعاء هذا التابع باستمرار كحلقة لانهائية، ويتم فيها الكود البرمجي المراد تفيذه، ويستمر التابع بتنفيذ الكود الموجود داخل الحلقة حتى يتم فصل التعذية عن الاردوينو او إعادة تشغيله او باستخدام تعليمية شرطية تسمح بالخروج من الحلقة عند حصول حدث معين.

```

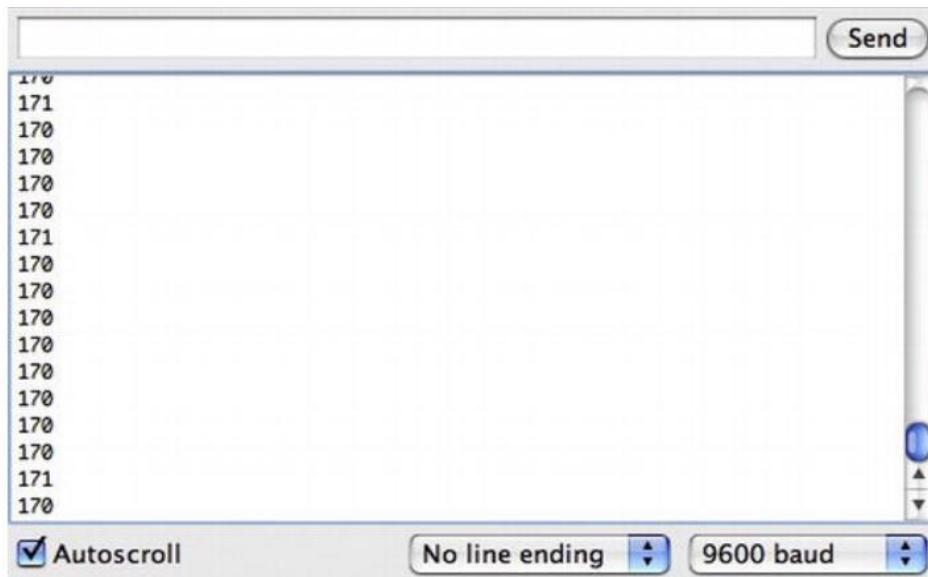
File Edit Sketch Tools Help
sketch_jun07a §
void setup() {
    // put your setup code here, to run once:
}

void loop() {
    // put your main code here, to run repeatedly:
}

```

تحتوي هذه البيئة على مرصد تسلسلي Serial Monitor وهو أداة لعرض البيانات التسلسلية التي يتم ارسالها من الاردوينو الى الحاسوب، الشكل (3-2-2-3)، يتم ذلك بمطابقة معدل الباود في الشاشة التسلسلية مع معدل الباود الذي تم تحديده في الكود البرمجي، ويعرف معدل الباود Baud على انه عدد التغيرات في الرموز المرسلة خلال ثانية واحدة.

الشاشة التسلسلية مفيدة جدا في العثور على العيوب والمشاكل في الكود البرمجي واصلاحها، وخاصة عندما يصعب تحديد مكان الخطأ او سببه، حيث يقوم الاردوينو بطباعة الرسائل على الشاشة التسلسلية حتى تتمكن من معرفة حالة البرنامج في نفس اللحظة التي يعمل بها.



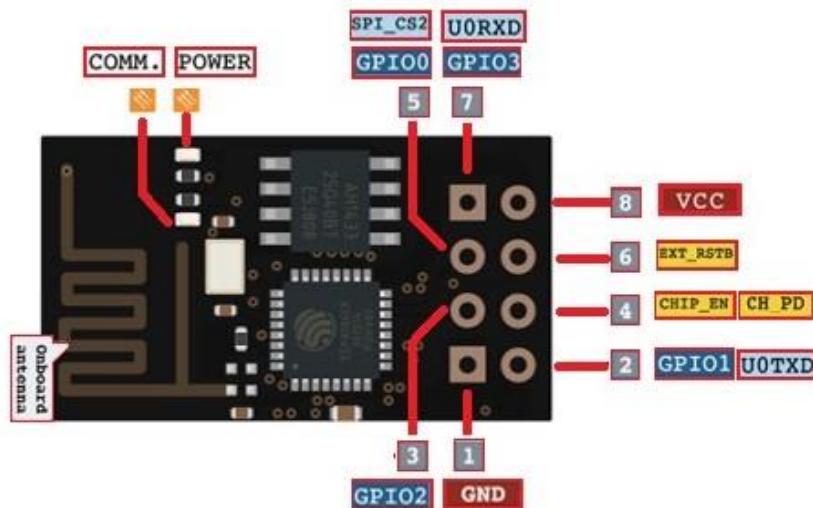
الشكل (3-2-3) الشاشة التسلسلية في الاردوينو .

3-2-3- وحدة الاتصال اللاسلكي في الاردوينو **Arduino Wireless Communication**

يشكل الوصل السلكي للاردوينو مع الحاسوب عائقا امام الكثير من تطبيقات الاردوينو، لذا يعد الوصل اللاسلكي للاردوينو مع الأجهزة الأخرى او حتى مع لوحة اردوينو آخر مهم للغاية لتطبيقات التحكم عن بعد بالروبوتات او غيرها، يوجد طرق متعددة للوصل اللاسلكي تختلف فيما بينها بالسرعة والمدى ومعدل استهلاك الطاقة ونذكر منها:

-وحدة الاتصال اللاسلكية Wi-Fi Modules

هي شرائح إلكترونية تستخدم في العديد من المنتجات لتحقيق اتصال لاسلكي بالإنترنت، تعد شريحة ESP-8266 أشهر هذه الشرائح وهي هي وحدة مرسل ومستقبل متسلسل Wi-Fi، وتعمل على بروتوكول TCP / IP وهي متوافقة تماما مع الاردوينو، الشكل (3-2-3-1).

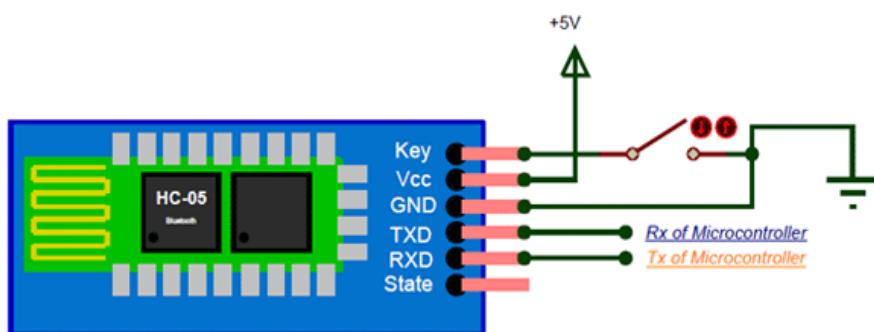


الشكل (3-2-1)

وحدة البلوتوث - Bluetooth Modules

يعد البلوتوث أحد البروتوكولات اللاسلكية الأكثر استخداماً في تطبيقات الوصل اللاسلكي لسرعتها وسهولة التعامل معها، يمكن وصل هذه الشريحة بأي متحكم عن طريق الاتصال التسلسلي UART، ويمكن استخدامها مع Arduino أيضاً، وتمتاز وضعية عمل Master / Slave يمكن ضبط حالة العمل عن طريق أوامر At commands .

وفي الحالة الافتراضية تعمل كـ Master و Slave ويمكن ضبطها لتعمل إما Master أو Slave عن طريق استخدام أوامر AT. ومن الأمثلة على شرائح البلوتوث شريحة HC-05 والتي سيتم استخدامها في هذا المشروع لربطarduino لاسلكياً مع الحاسوب، الشكل (3-2-2).



الشكل (3-2-2)

Xbee - شرائح

تم انتاج هذه الشرائح من قبل شركة Digit International وهي عبارة عن معالج مصغر يستخدم بروتوكول IEEE 802.15.4 (ZigBee) وهو بروتوكول اتصال عالي المستوى يستخدم لإنشاء شبكة شخصية باستخدام طاقة منخفضة، الشكل (3-3-2-3).



.Xbee (3-3-2-3)

Radio Frequency (RF-Modules) -

تعتبر وحدة الترددات اللاسلكية الراديوية RF Module من أشهر العناصر المتعددة الاستعمال في الاردوينو، حيث تستعمل في تشكيلة واسعة من التطبيقات التي تتطلب التحكم اللاسلكي، وهي أجهزة إرسال واستقبال، مما يعني أن كل وحدة تستطيع إرسال واستقبال البيانات، وتكون هذه الوحدات منخفضة التكلفة ويمكن استعمالها مع أي متحكم مصغر، الشكل (4-3-2-3).



.Nrf24L01 (4-3-2-3)

الفصل الرابع

الدراسة النظرية للنظام Theoretical Study

1-4- الدراسة الحركية Kinematic Study

تعد الدراسة الحركية للروبوت مهمة لأنها تصف حركته، ولأنه نستطيع من خلال دراستها حساب جوانب مختلفة من الحركة مثل السرعة والتسارع والازاحة والزمن.

في هذا النظام لدينا جملتين:

-جملة ثابتة مرجعية (Global)-

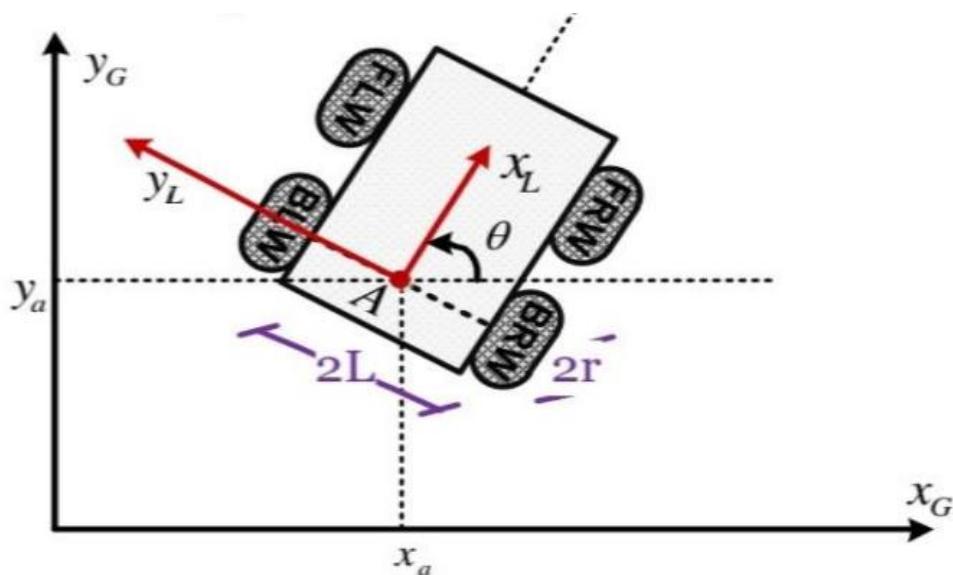
$$q^G = [X_a^G \quad Y_a^G \quad \theta]^T$$

-جملة متحركة (Local)-

$$q^L = [X_a^L \quad Y_a^L \quad \theta]^T$$

التدوير يكون على المحور Z، ننسب الجملة المتحركة (Local) للثابتة (Global) فنجد أن:

$$\dot{q}_G = R(\theta)\dot{q}_L$$



الشكل التوضيحي للدراسة الحركية.

وتكون مصفوفة التدوير على محور Z بالشكل التالي:

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

-السرعة:

تمثل السرعة الانسحابية للروبوت بالعلاقة:

$$U = \frac{U_R + U_L}{2} = r \frac{\dot{\phi}_R + \dot{\phi}_L}{2}$$

والسرعة الدورانية للروبوت (السرعة الزاوية):

$$\omega = \frac{U_R - U_L}{2L} = r \frac{\dot{\phi}_R - \dot{\phi}_L}{2L}$$

وبما ان السرعة تفاضلية يكون:

-الحركة الى الامام: المحركات في الطرف الأيمن والأيسر بنفس السرعة.

-الحركة الى الخلف: المحركات في الطرف الأيمن والأيسر بنفس السرعة.

-الحركة الى اليسار: يتم كبح محركات الطرف الأيمن وتعطى كل السرعة لمحركات الطرف الأيسر.

-الحركة الى اليمين: يتم كبح محركات الطرف الأيسر وتعطى كل السرعة لمحركات الطرف الأيمن.

بالتالي تكون السرعة على المحور X:

$$\dot{X}_a^L = r \frac{(\dot{\phi}_R + \dot{\phi}_L)}{2}$$

وعلى المحور Y تكون مساوية للصفر:

$$\dot{Y}_a^L = 0$$

وتكون السرعة الدورانية:

$$\dot{\theta} = \omega = r \frac{(\dot{\phi}_R - \dot{\phi}_L)}{2L}$$

ربط العجلات مع الجملة المتحركة:

$$\dot{q}^L = \begin{bmatrix} r/2 & r/2 \\ 0 & 0 \\ r/2L & r/2L \end{bmatrix} \begin{bmatrix} \dot{X}_a^L \\ \dot{Y}_a^L \\ \dot{\theta} \end{bmatrix}$$

الربط مع الجملة الثابتة:

نضرب بمصفوفة الدوران حول المحور Z فنجد:

$$\dot{q}^G = \begin{bmatrix} \dot{X}_a^G \\ \dot{Y}_a^G \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\theta & \frac{r}{2}\cos\theta \\ \frac{r}{2}\sin\theta & \frac{r}{2}\sin\theta \\ \frac{r}{2L} & \frac{r}{2L} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix}$$

بالتعميض وبحسب المعادلات السابقة نستطيع الانتقال ل(ϑ, ω):

$$\dot{q}^G = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \vartheta \\ \omega \end{bmatrix}$$

2-4- الدراسة الديناميكية Dynamic Study

هي الدراسة لحركة النظام الميكانيكية مع النظر إلى اختلاف القوى التي تؤثر على الروبوت وهي على عكس الدراسة الحركية التي لا تأخذ القوى بعين الاعتبار، وهي أساسية لنماذج حركة النظام وشرح القوى والعزوم اللازمة لتوليد الحركة.

- إيجاد المعادلات الديناميكية باستخدام طريقة نيوتن

لدينا الموضع:

$$\hat{r} = r e^{j\theta}$$

نقوم باشتقاء الموضع للحصول على السرعة:

$$\dot{\hat{r}} = [r]e^{j\theta} + [r\dot{\theta}]e^{j(\theta+\frac{\pi}{2})}$$

نقوم باستفادة السرعة للحصول على التسارع:

$$\ddot{r} = [\ddot{r} - r\dot{\theta}^2]e^{j\theta} + [2\dot{r}\dot{\theta} + r\ddot{\theta}]e^{j(\theta+\frac{\pi}{2})}$$

وباستنتاج الجزء الحقيقي عن التسارع المماسى:

$$a_u = \dot{v}_u - v_\omega \dot{\theta} \quad [1]$$

استنتاج الجزء التخيلى عن التسارع الناظمى:

$$a_\omega = \dot{v}_\omega - \dot{v}_u \dot{\theta} \quad [2]$$

لدينا قوانين نيوتن:

$$Ma_u = F_{uL} + F_{uR} \quad \text{للحركة المماسية:}$$

$$Ma_w = F_{wL} - F_{wR} \quad \text{للحركة الناظمية:}$$

$$J\ddot{\theta} = L(F_{uR} - F_{uL}) + d(F_{wR} - F_{wL}) \quad \text{قانون نيوتن للعزم:}$$

وبتعويض [1] و [2] في قوانين نيوتن السابقة نحصل على:

$$\dot{v}_u = v_w \dot{\theta} + \frac{F_{uR} + F_{uL}}{M}$$

$$\dot{v}_w = -v_u \dot{\theta} + \frac{F_{wR} - F_{wL}}{M}$$

$$\ddot{\theta} = \frac{L}{J}(F_{uL} - F_{uR}) + \frac{d}{J}(F_{wR} - F_{wL})$$

الانتقال من النقطة A إلى النقطة C:

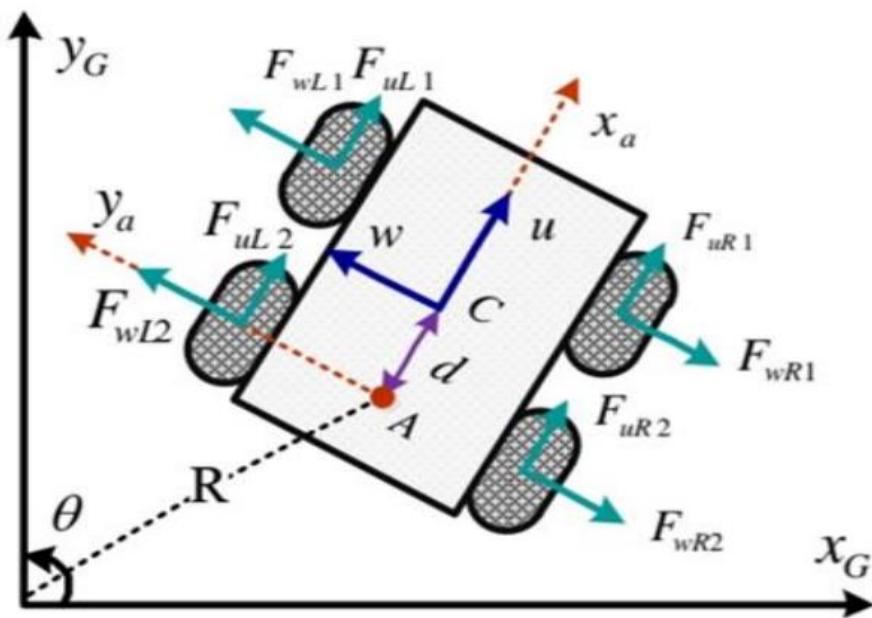
$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} v_u \\ v_w \end{bmatrix}$$

وبالإسقاط على المحاور:

$$x_c = x_a + d \cos \theta$$

$$y_c = y_a + d \sin \theta$$

نحصل على السرعة المماسية:



الشكل التوضيحي للدراسة الديناميكية.

وبهذا تكون قد قمنا بالدراسة الحركية والدراسة الديناميكية (Kinematic and Dynamic) للروبوت المتنقل، حيث قمنا بتحديد موضع وسرعة وتسارع الروبوت بدون تأثير القوى وبعد ذلك تم الربط بين العزم والسرعة والقوى المؤثرة على الروبوت، يعرض الجدول (1-2-4) المتحولات المستخدمة في الدراسة.

الوصف:	البارامتر:
المسافة بين العجلات	$2L$
مركز الكتلة	C
نقطة المنتصف بين العجلات الخلفية	A
المسافة بين C و A	d
نصف قطر العجلات	r
السرعة الزاوية للعجلات	ω_R, ω_L
السرعة الناظمية	v
السرعة الزاوية	ω
نسبة الترس	N
كتلة الروبوت	M
عزم العطالة الكلي للروبوت	J
القوى المماسية المبذولة من العجلات	F_{uR}, F_{uL}
القوى الشعاعية المبذولة من العجلات	F_{wR}, F_{wL}

الجدول (1-2-4).

الفصل الخامس

المكونات المادية للنظام System Hardware

سنتحدث في هذا الفصل عن الأجهزة والأدوات المستخدمة في هذا المشروع ومبدأ عملها ودورها وسبل اختيارها.

1-5-محرك التيار المستمر DC-Motor

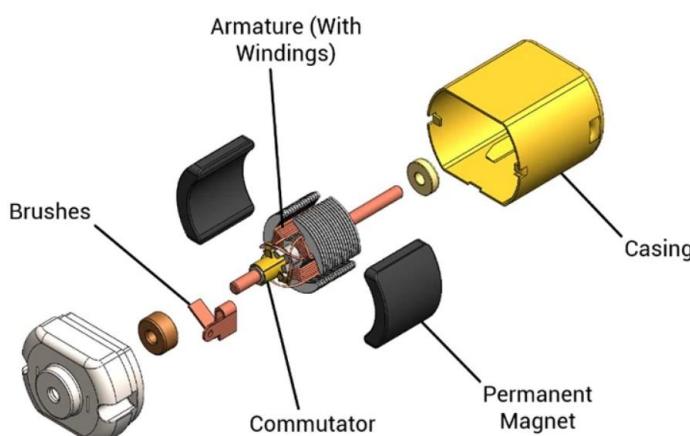
هو محرك يعمل بالتيار الكهربائي المستمر ويتمتع بسلوك بدء تشغيل (إلاع) جيد وإمكانية ضبط جيدة لعمليات التشغيل.

يتكون مmotor التيار المستمر من الثابت (مغناطيس كهربائي أو دائم) Stator والدوار Rotor وهو مصنوع من الحديد والمجمع Commutator وفرش الكربون او الغرافيت، الشكل (1-1-5).

-مبدأ عمله:

يتم خلق مجال مغناطيسي بواسطة العضو الثابت ويتم تثبيت العضو الدوار داخل هذا المجال المغناطيسي، يتم لف العضو الدوار بملف نحاسي وتوصيلها بمصدر كهرباء باستخدام فرش الكربون كلامسات منزلقة، وعندما يتدفق التيار المستمر عبر الملف النحاسي، يخلق الملف مجالاً مغناطيسياً ويصبح قلب الحديد الدوار مغناطيسياً كهربائياً، وهذا بدوره يؤدي إلى نشوء قوى بين العضو الثابت (مغناطيس دائم أو كهربائي) والعضو الدوار (مغناطيس كهربائي).

الأقطاب المغناطيسية المتشابهة تتفافر، بينما تجاذب الأقطاب المغناطيسية المختلفة، تسبب قوى الجذب والتفافر بين الأقطاب المغناطيسية في دوران العضو الدوار للmotor، وحتى يستمر المغناطيس الكهربائي في الدوران، يجب عكس قطبته المغناطيسية في كل دورة، فبمجرد أن تتلاقي الأقطاب المغناطيسية للعضو الثابت والعضو الدوار باتجاه بعضها البعض بشكل مباشر، يقوم المجمع الكهربائي بعكس قطبية المجال المغناطيسي للعضو الدوار عن طريق عكس اتجاه التيار، بحيث تستمر حركته الدورانية الناتجة عن عملية التجاذب والتفافر.



الشكل (1-1-5) مكونات محرك التيار المستمر.

في هذا المشروع سنقوم باستخدام محرك تيار مستمر مزود بعلبة سرعة المعروفة أيضاً باسم TT Motor، يوفر هذا المحرك طريقة سهلة وغير مكلفة للتحكم في الروبوتات المتنقلة. يمكن تغذية هذا المحرك بجهد يتراوح من 3v إلى 6v، وتختلف سرعته والتيار المار فيه باختلاف جهد التغذية، الجدول (5-1-5)، التالي يوضح تغيرات السرعة والتيار حسب تغيير الجهد:

السرعة	التيار	الجهد
120RPM بدون حمل	150mA	3VDC
185RPM بدون حمل	155mA	4.5VDC
250RPM بدون حمل	160mA	6VDC

الجدول (5-1-2) تغير السرعة والتيار حسب الجهد.

يتمتع TT Motor بخصائص ومميزات مختلفة عن المحركات الأخرى بالإضافة إلى احتوائه على علبة السرعة مما يجعله مثالياً لتطبيقات الروبوتات والمشاريع الصغيرة وهذا هو سبب اختيارنا لهذا المحرك في هذا المشروع، الشكل (3-1-5).



الشكل (3-1-5) محرك TT Motor مع علبة السرعة.

يوضح الجدول (3-1-5)، التالي أهم خصائص هذا المحرك:

90RPM +/- 10%	أقل سرعة عند جهد 3v
200RPM +/- 10%	أقل سرعة عند جهد 6v
0.15Nm-0.60Nm	العزم
1:48	نسبة الترس
70x22x18mm	ابعاد جسم المحرك
30.6g	الوزن

الجدول (3-1-5) خصائص TT Motor

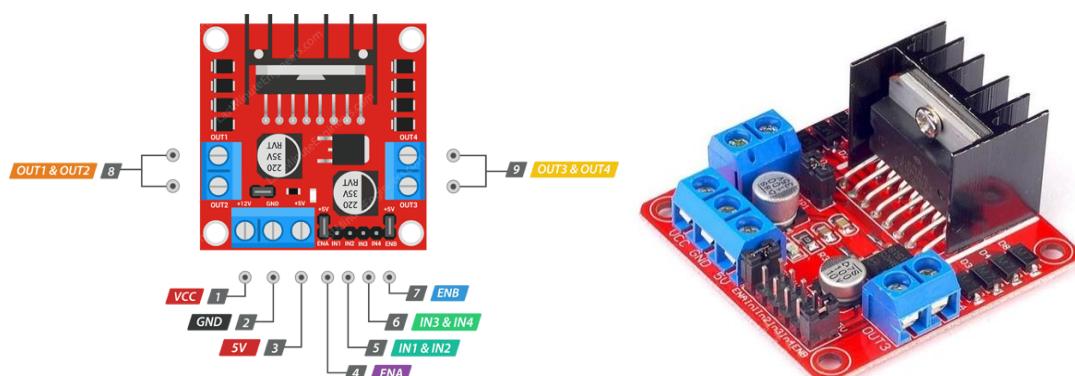
سنقوم بتركيب عجلات على كل محرك من المحركات الأربعة المستخدمة في الروبوت المتنقل لتسهيل حركته، الشكل (4-1)، ولكن من غير الممكن قيادة هذه المحركات بشكل مباشر بواسطة المتحكم المصغر او الاردوينو لذلك سوف نوصل المحركات مع دارة القيادة L298N والتي سوف تتحدث عنها بشكل مفصل في الفصل القادم.



الشكل (4-1) العجلات المستخدمة في الروبوت.

5-2- دارة قيادة محرك التيار المستمر DC-Motor Driver L298N

هي دارة الكترونية تستخدم للتحكم بسرعة واتجاه محرك التيار المستمر DC-Motor، الشكل (5-1)، ويتم التحكم بالسرعة عن طريق خاصية تعديل عرض النبضة PWM التي يمكن تطبيقها من لوحة Arduino أو أي متحكم آخر يدعم خاصية عرض النبضة، يمكن لدارة L298N التحكم بمحركين DC معاً، تستخدم هذه الدارة بكثرة في مجال الروبوتات وسوف نقوم باستخدامها في هذا المشروع للتحكم في محركات الروبوت المتنقل.



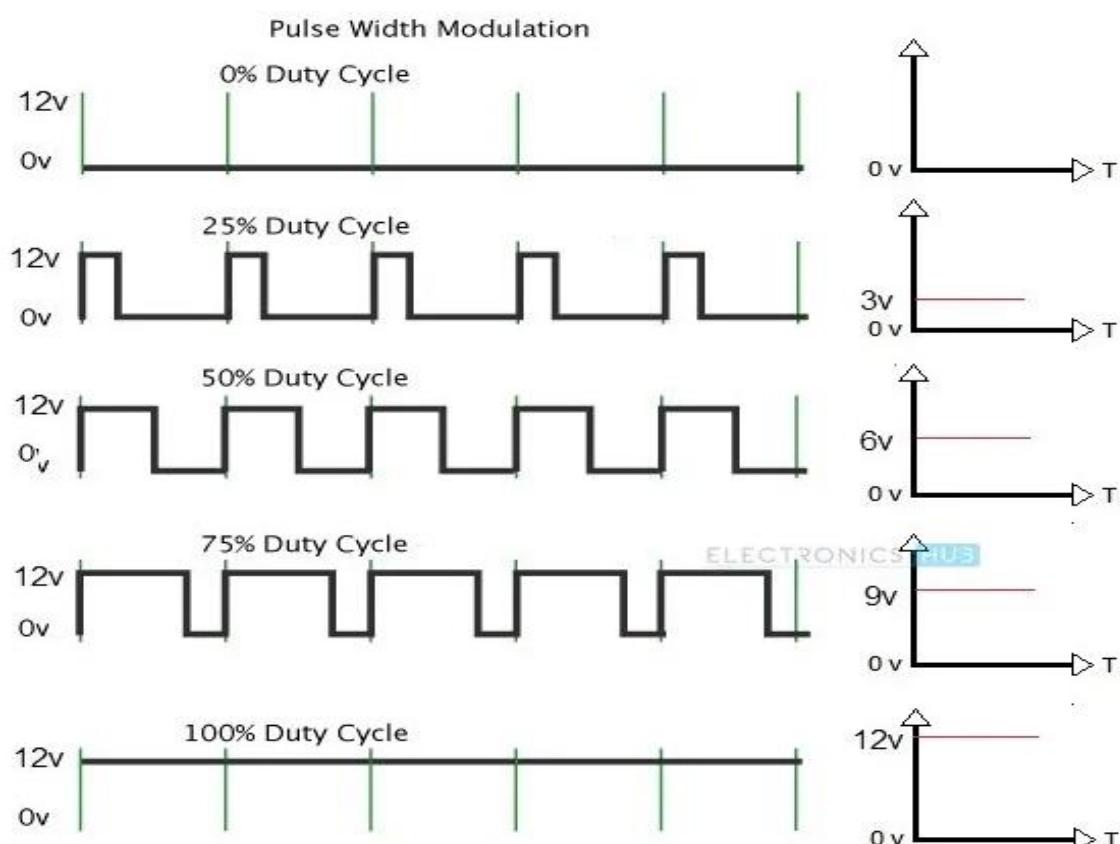
الشكل (5-1) دارة L298N.

تتكون دارة القيادة L298N من عدد من العناصر الالكترونية المتكاملة في الأداء، ومنها الدارة المتكاملة LM298 Motor Driver IC والتي هي عبارة دارتين H-Bridge، منظم الجهد 87M05، المكثفات، المقاومات، مبرد الحرارة، مؤشر الطاقة (LED)، وصلة الـ 5V.

-مبدأ عمل دارة القيادة-

باستخدام تقنية PWM سيتم التحكم بقيمة الجهد الوسطي المطبق على المحرك عن طريق فصل وتوصيل الطاقة الكهربائية عن المحرك بمعدل سرعة عال جداً، وسيكون تردد هذا التبديل بين الحالتين من فئة عدة عشرات من الكيلوهرتز، هنا سيعتمد الجهد الوسطي المطبق على المحرك على ما يسمى دور التشغيل (Duty Cycle)، إن دور التشغيل لتقنية تعديل عرض النبضة تمثل بالنسبة بين الوقت الذي تكون فيه الإشارة بوضع تشغيل (ON or HIGH) إلى الدور الإجمالي للإشارة أي مجموع زمن حالي (ON+OFF)، عادة ما يعبر عن دور التشغيل كنسبة مئوية.

يوضح الشكل التالي إشارات PWM مختلفة لمصدر طاقة 12V مع دورات تشغيل مختلفة: 0%， 25%， 50%， 75%， 100% بالترتيب، الشكل (2-2-5).



.الشكل (2-2-5).

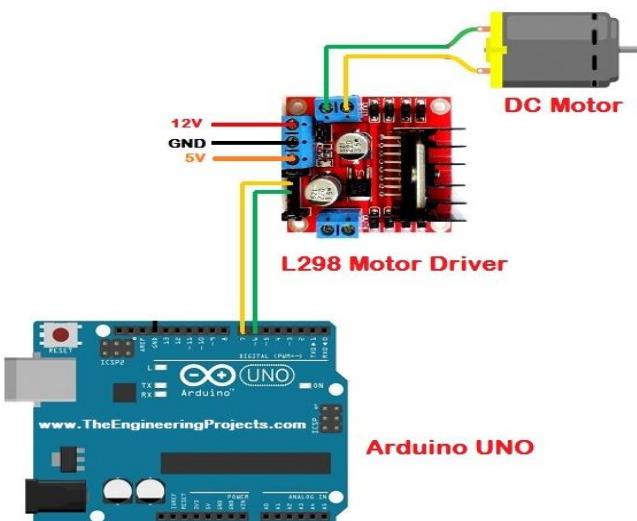
ـ منافذ الدارة وكيفية الوصل والتحكم مع المحركات

يوضح الجدول (3-2-5)، التالي منافذ الدارة ووظيفة كل منها:

المنفذ	وظيفة وأداء المنفذ
IN1,IN2	التحكم في حركة واتجاه دوران المحرك الأول
IN3,IN4	التحكم في حركة واتجاه دوران المحرك الثاني
ENA	تشغيل وإيقاف المحرك الأول
ENB	تشغيل وإيقاف المحرك الثاني
OUT1,OUT2	منفذ تشغيل المحرك الأول
OUT3,OUT4	منفذ تشغيل المحرك الثاني
5V	منفذ تغذية الدارة، يعمل على اخراج 5V وتيار 0.5A
12V	منفذ التغذية الإضافي الذي سيقوم بتشغيل المحركات
GND	الطرف السالب للدارة

الشكل (3-2-5) منافذ دارة القيادة L298N.

كل ما يلزم للتحكم في دوران المحرك هو تطبيق جهد منطقي مرتفع (HIGH) او جهد منطقي منخفض (LOW) على هذه المنافذ، يوضح الجدول (5-2-4) آلية التحكم في اتجاه دوران محرك موصول مع الشريحة.



اتجاه الدوران للمotor	IN2	IN1
المحرك لا يدور	0 فولت	0 فولت
يتحرك الى الامام	0 فولت	5 فولت
يتحرك الى الخلف	5 فولت	0 فولت
يتوقف	5 فولت	5 فولت

الشكل (5-2-5).

3-5- تصميم الروبوت النقال Mobile Robot Design

سيتم في هذا المشروع تصميم روبوت نقال للتحكم به بواسطة إيماءات اليد، وتحدثنا سابقاً عن أنظمة الروبوتات المتنقلة ومبدأ عملها، وفيما يلي نستعرض الأجزاء المكونة للروبوت النقال المستخدم في مشروعنا، الشكل (1-3-5).

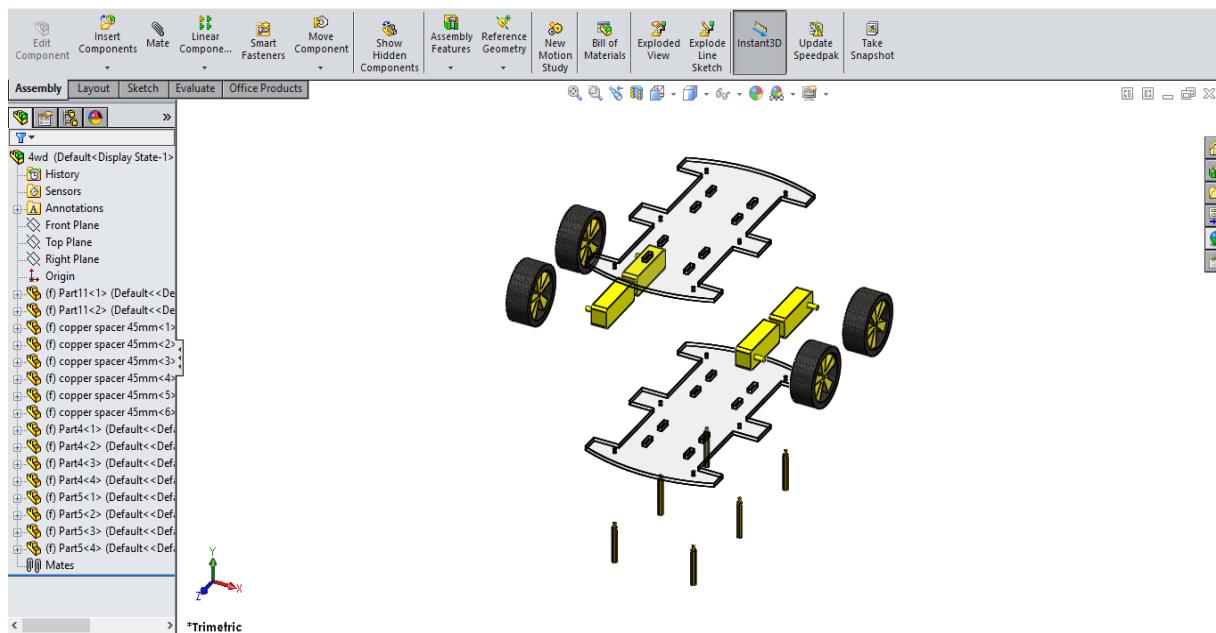
مكونات الروبوت:

- محركات التيار المستمر DC-Motors، ومهمتها تحريك الروبوت من مكان آخر، وقد تم الحديث عنها سابقاً في هذا الفصل.
- العجلات، مصنوعة من المطاط وتكون موصولة مع المحركات لجعل الحركة أكثر مرنة.
- الجسم العلوي والسفلي وهو عبارة عن سطح مستوي مصنوع من مادة الأكريليك الشفاف بسمك 2mm.
- براغي ووصلات تستخدم لثبيت المحركات وثبيت الجسم العلوي والسفلي.



الشكل (1-3-5) الأجزاء المكونة للروبوت النقال.

تم تصميم نموذج للروبوت النقال على برنامج SolidWorks، الشكل (2-3-5).



.الشكل (2-3-5) تصميم الروبوت النقال على برنامج SolidWorks

4-4- وحدة البلوتوث Bluetooth Module HC-05

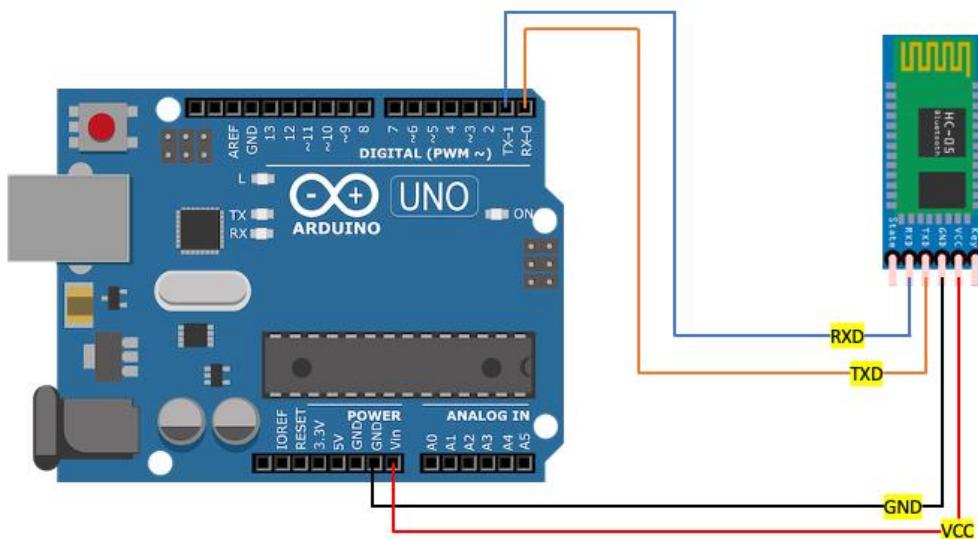
تحدثنا في الفصل الثالث عن طرق الاتصال اللاسلكي في الاردوينو، وتحدثنا عن الشريحة Bluetooth Module HC-05 التي سوف نستخدمها في هذه المشروع ل تقوم بالربط اللاسلكي بين الاردوينو والحاسوب.

يعتمد مبدأ عمل الشريحة على تقنية البلوتوث للتحكم عن بعد، وهي أحد تقنيات التحكم بواسطة الموجات الراديوية RF، تسمح هذه التقنية للشريحة بالاتصال مع الأجهزة الأخرى كالحاسوب او الهاتف النقال وتسمح أيضا باتصال لوحين اردوينو او أي متحكمات اخرى مع بعضهم البعض، وتستخدم بكثرة لسهولة التعامل معها ونطاق استجابتها العالي وتكلفتها المنخفضة.

-وحدة البلوتوث Bluetooth Module HC-05 مع الأردوينو

تعمل وحدة البلوتوث بطريقة التواصل (Serial Communication) حيث يقوم الحاسب بواسطة أحد البرامج بإرسال بيانات للبلوتوث في حالة وقوع حدث معين.

وفي الناحية الأخرى عندما تسلمه وحدة البلوتوث بيانات من الجهاز المقترن به، يقوم بإرسال هذه البيانات إلى لوحة الأردوينو عن طريق قطب RX من القطب TX في لوحة الأردوينو، يوضح الشكل (1-4-5) طريقة وصل الأردوينو مع شريحة البلوتوث.



الشكل (1-4-5) وصل الأردوينو مع شريحة البلوتوث.

ويوضح الجدول التالي أطراف التوصيل الخاصة بالشريحة ووظيفة كل طرف، الشكل (5-4-2).

وظيفته	المنفذ
يستخدم للتبديل بين وضع البيانات و وضع AT Commands	Enable/Key
طرف تغذية الشريحة +5V	Vcc
الطرف الأرضي للشريحة	Ground
طرف ارسال البيانات	TX
طرف استقبال البيانات	RX
قطب الحالة ويستخدم للتحقق مما اذا كانت الوحدة تعمل بشكل صحيح	STATE

الشكل (5-4-2) شرح أطراف التوصيل لوحدة البلوتوث.

5-5- الأردوينو Arduino UNO

تحدثنا في الفصل الثالث عن المتحكمات المصغرة بشكل عام وعن لوح الأردوينو تحديداً، وتم ذكر مبدأ عمله ومكوناته وأهم أنواعه وميزاته، وفي مشروعنا سوف نستخدم لوح الأردوينو من نوع Arduino UNO، للأردوينو أهمية كبيرة في هذا المشروع لكونه يمثل أساس المشروع الذي يتم من خلاله الربط اللاسلكي والتحكم بمشغلات الروبوت والتواصل مع البيئة البرمجية المستخدمة في معالجة الصور من كاميرا Kinect.

تم اختيار Arduino UNO لسهولة التعامل معه ولكونه مفتوح المصدر ومناسب لتطبيقات الروبوتات بالإضافة إلى موثوقيته وتكلفته المنخفضة واستهلاكه القليل للطاقة.

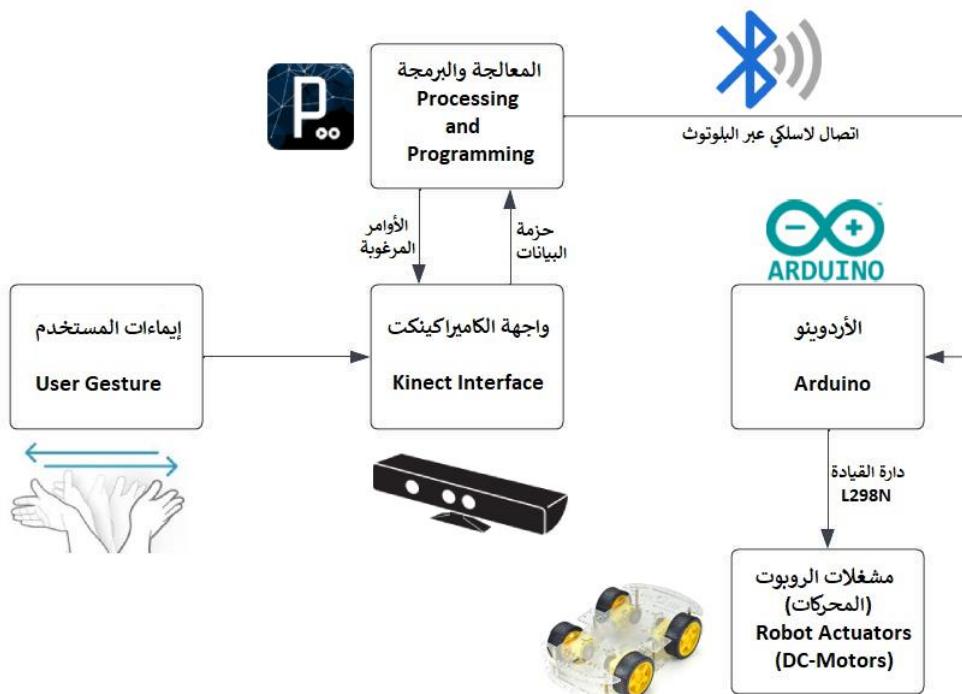
5-6- كاميرا Microsoft Kinect

سيتم استخدام كاميرا Kinect في هذا المشروع لقدرتها على تسجيل الإشارات والaimاءات التي سيقوم بها المستخدم وفهمها وتحويلها إلى أوامر نصية، وقد تم التوسع في شرح مبدأ عمل الكاميرا وذكر مكوناتها وخصائصها بالتفصيل في الفصل الثاني، وكما ذكرنا سابقاً فإن الواجهة المستخدمة للتعامل برمجياً مع صور الكاميرا هي Processing IDE والتي سيتم الحديث عنها بالتفصيل في الفصل القادم.

وقد تم اختيار هذه الكاميرا لوجود مستشعر العمق Depth Sensor المفيد في تطبيقات التحكم القائم على الرؤية ويؤمن هذا المستشعر أيضاً الدقة في تتبع إشارات الإنسان وحركاته، وإمكانية تضمين مكتبيات خاصة بها مصممة للتعرف على جسم الإنسان وإشاراته، هذا بالإضافة إلى تكلفتها المنخفضة وسهولة وصلها مع الحاسب.

7-5- المخطط الصندوقى ومخطط الدارة Block Diagram and Circuit Diagram

يوضح الشكل (7-1)، المخطط الصندوقي للمشروع:



الشكل (7-1) المخطط الصندوقي للمشروع.

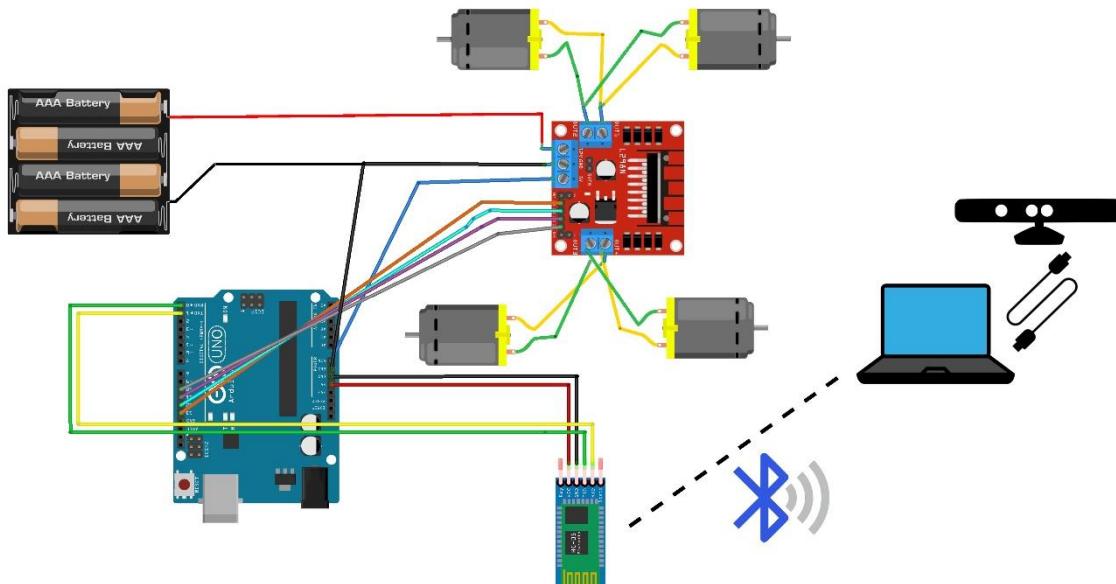
بدايةً يقوم المستخدم بال الوقوف أمام الكاميرا Kinect وتظهر صورته على واجهة الكاميرا التي تكون متصلة بالحاسب، وعند قيام المستخدم بإيماءة محددة يتم التعرف عليها وارسالها إلى وحدة المعالجة، حيث تكون الإيماءة المرغوبة محددة من وحدة البرمجة مسبقاً.

في وحدة المعالجة بعد استقبال البيانات من واجهة الكاميرا وفهمها وتحويلها إلى أوامر نصية، يتم ارسالها بعد ذلك إلى لوحةarduino لاسلكياً عبر وحدة البلوتوث والتي قد تم الاتصال بها سابقاً من قبل الحاسب.

تأتي مهمة لوحةarduino بعد ذلك وهي القيام بتشغيل محركات الروبوت المتเคล ببناءً على المعلومات التي تلقاها من وحدة المعالجة، ويتم تشغيل المحركات والتحكم بها بواسطة دارة القيادة المستخدمة.

مخطط الدارة :Circuit Diagram

تم رسم المخطط الموضح بالشكل (7-5-2)، بواسطة برنامج تصميم الدارات Fritzing.



الشكل (7-5-2) مخطط توصيلات الدارة.

يتم تغذية دارة القيادة L298N ببطاريات ذات جهد ثابت 12V، يتم وصل الأقطاب OUT1,OUT2 مع محركات الجهة اليمنى والأقطاب OUT3,OUT4 مع محركات الجهة اليسرى، دارة القيادة الأساسية تستخدم لقيادة محركات فقط ولكن في مشروعنا سنقوم بقيادة أربع محركات بواسطة دارة قيادة واحدة لكون حركة الروبوت لدينا بسيطة ولا تحتاج إلى التحكم بكل محرك لوحده، ويتم ذلك عن طريق وصل محركات الجهة اليمنى مع بعضها ومحركات الجهة اليسرى مع بعضها.

ويتم تغذية لوحةarduino من منفذ الـ 5V الموجود في دارة القيادة ووصل GND مشترك بين البطاريات ودارة القيادة والاردوينو ، ولربط الاردوينو بالمحركات يتم وصل الأقطاب D10,D11,D12,D13 من دارة القيادة مع اقطاب الاردوينو الرقمية IN1,IN2,IN3,IN4 .

وبالنسبة لوحدة البلوتوث سنقوم بتغذيتها من قطب الـ 5V الموجود في الاردوينو ووصل GND الوحدة مع الاردوينو واقطاب TX,RX الخاصة بالوحدة يتم وصلها مع اقطاب الاردوينو TX حيث يتم وصل (RX Arduino-TX Bluetooth) و (TX Arduino-RX Bluetooth).

وتكون الكاميرا Kinect موصولة مع الحاسوب عن طريق منفذ USB2.0 ويكون الحاسوب متصل لاسلكياً مع وحدة البلوتوث.

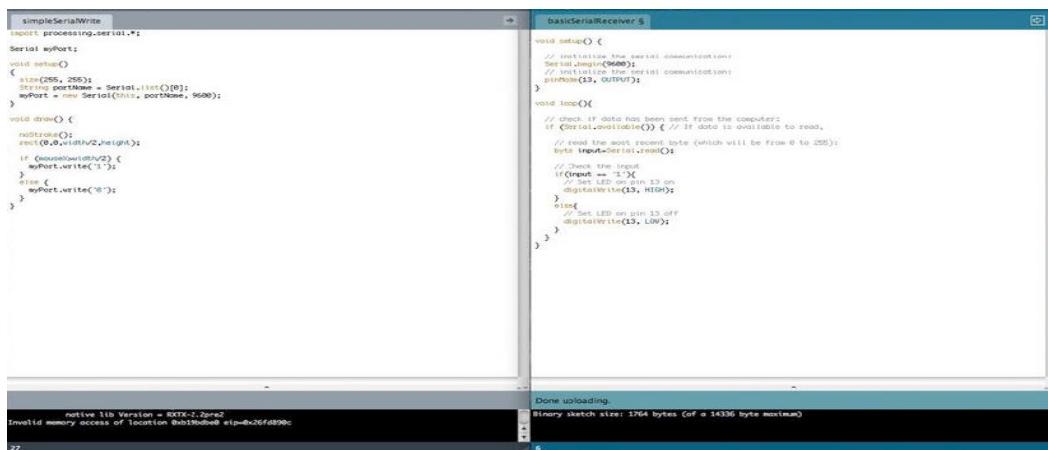
الفصل السادس

برمجة ومعالجة النظام System Software

Processing IDE - 1-6 بيئة

تحدثنا في الفصل الثاني عن البرامج التي تمكنا من التعامل برمجياً مع كاميرا Kinect وذكرنا منها برنامج Processing IDE وهو البرنامج الذي سنقوم باستخدامه في هذا المشروع والعمل عليه لعرض خرج الكاميرا وتحويل البيانات المستقبلة منها إلى أوامر ترسل لاسلكياً إلى لوحة الأردوينو.

من أهم الخصائص تتميز بها لغة ال Processing أنها مبنية على أساس لغة Java وهذا يسهل عملية عرض الصور والأشكال بالإضافة إلى واجهة البرنامج البسيطة والمشابهة لواجهة بيئة Arduino IDE، الشكل (1-6)، ويتمثل دور هذه البيئة في عرض صورة مستشعر العمق وصورة ال RGB الناتجة من الكاميرا، الشكل (2-1)، ثم التعرف على إيماءات أو إشارات محددة وفهمها ومعالجتها ويتم ذلك بواسطة مكتبيات تضاف إلى هذه البيئة والتي سنتحدث عنها لاحقاً.



الشكل (1-6) واجهة Processing IDE مشابهة لواجهة Arduino IDE



الشكل (2-1) عرض صورة العمق والصورة الملونة من خلال Processing باستخدام المكتبيات.

2-6- مكتبيات SimpleOpenNI and Kinect Libraries

-مكتبة Kinect

تكمّن أهمية هذه المكتبة بأنها تمكّنا من عرض صورة العمق Depth Image والصورة الملونة RGB للكاميرا والوصول إلى بيانات هذه الصور وفهمها من خلال بيئة Processing IDE.

-مكتبة SimpleOpenNI

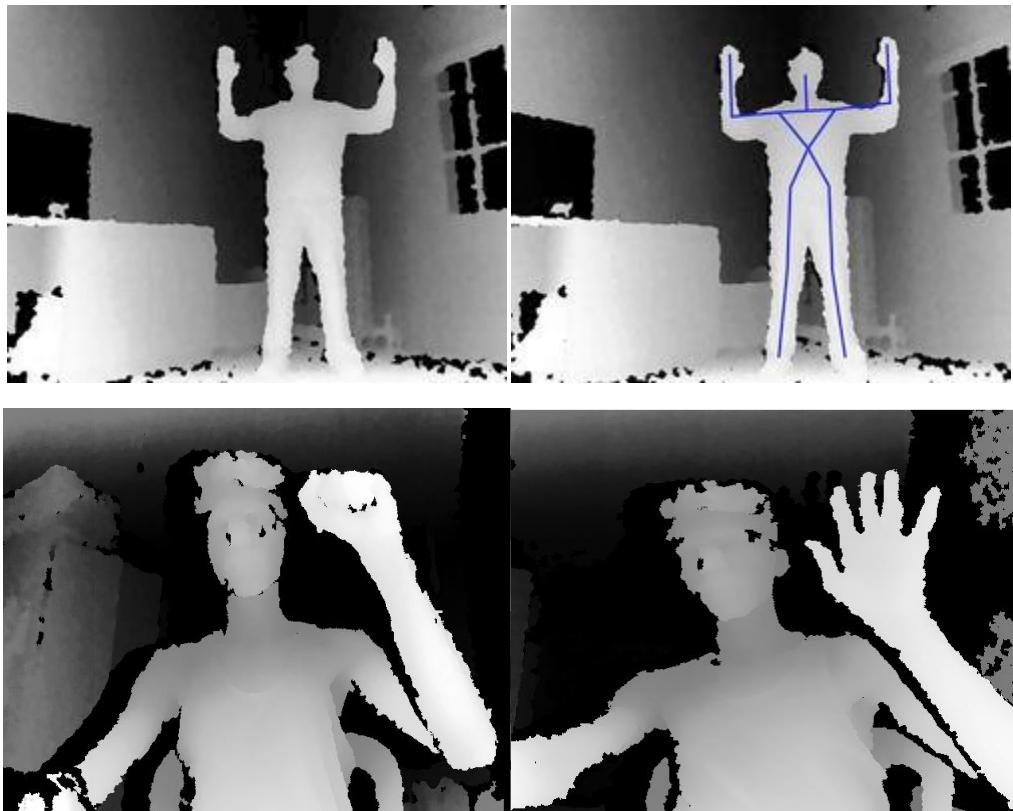
هي مكتبة مبنية ومنفذة بلغة البرمجة C++, تم تصميم جميع التوابع والمنهجيات الموجودة في هذه المكتبة بطريقة يسهل الوصول إليها في بيئة JAVA، وبعدها تم إنشاء مكتبة خاصة تحتوي على كل هذه الميزات على برنامج Processing IDE.

تضُم هذه المكتبة العديد من التوابع التي تسهل عملية التعرف على الأشكال والوجوه والأيدي وهيكل الإنسان، ويوضح الجدول (1-2-6)، أهم التوابع التي سوف نستخدمها في مشروعنا ووظيفتها تابع:

التابع	وظيفته
Kinect.AddGesture()	للتعريف عن الإيماءات التي سوف نستخدمها في البرنامج والتي تكون مصممة مسبقاً في المكتبة.
depthMapRealWorld()	يستخدم لتطبيق خصائص صورة العمق على الصورة في العالم الحقيقي
kinect.enableDepth()	يقوم هذا التابع بتفعيل مستشعر العمق للصورة للكاميرا Kinect
kinect.enableGesture()	تفعيل وضع التعرف على الإيماءات
kinect.enableHands()	تفعيل وضع التعرف على اليد
Kinect.Update()	لتحديث البيانات (الصور) المستقبلة من الكاميرا
SendSerialData()	تابع الاتصال التسلسلي ويهتمي بالبيانات التي نريد ارسالها إلى الأردوينو
StartTrackinghands()	لبدء في تتبع حركة اليد وتسجيل بيانات الحركة
kinect.createSessionManager()	إعداد توابع NITE وتحديد الإيماءات التي نريد استشعارها

.الشكل (1-2-6) أهم التوابع في مكتبات SimpleOpenNI and NITE

يمكنا من خلال التوابع السابقة بتحديد حركات مختلفة في البرنامج لكي تقوم الكاميرا باستشعارها وتتبعها من خلال توابع أخرى مخصصة لهذه المهمة، ويوضح الشكل (6-2-2)، بعض الأمثلة على الحركات التي تكون معرفة بشكل أساسى في المكتبة.



الشكل (6-2-6) بعض الإيماءات المعرفة في المكتبات.

6-2-1- خطوات عمل البرنامج في بيئة Processing IDE

في البداية سنقوم بتعريف المكتبات التي ذكرناها سابقاً ومن خلال التوابع والتعليمات الخاصة بهذه المكتبات سنقوم بفتح نافذة لعرض الصورة من الكاميرا، أبعاد هذه النافذة سوف تكون مصممة لعرض صورة الكاميرا، وسيتم تعريف إيماءات محددة ونسبها إلى توابع تتضمن شروط منطقية وأوامر مختلفة، حيث تفعل هذه التوابع تلقائياً عند القيام بأحد هذه الإيماءات.

فعلى سبيل المثال عند التلویح باليد سيتم تفعيل التابع الخاص بهذه الحركة والذي يتضمن تعليمات تتبع حركة اليد واتجاهها وموقعها الحالي بالنسبة إلى نقطة مرجعية.

وبعد الحصول على المعلومات الناتجة من حركة اليد نقوم بكتابه تعليمات الاتصال التسلسلي لإرسال هذه البيانات الى لوحة الأردوينو.

تكون البيئة البرمجية للأردوينو Arduino IDE بدورها مزودة بتوابع خاصة بتشغيل محركات الروبوت تفعل عند حصولها على بيانات الكاميرا من خلال وحدة الاتصال التسلسلي.

وبهذه الطريقة وبنفس خطوات المثال السابق سنقوم بتعريف عدة إيماءات وجمع البيانات المختلفة عن حركة اليد وتبادل البيانات تسلسليا مع لوحة الأردوينو، سوف تكون قادرین على كتابة النص البرمجي الذي يمكننا من التحكم بحركة الروبوت باستخدام إيماءات اليد.

3-3- النص البرمجي للمشروع Program Code

النص البرمجي في بيئة Processing IDE

```
import SimpleOpenNI.*;
import processing.opengl.*;
import processing.serial.*;

SimpleOpenNI kinect;
Serial myPort;

XnVSessionManager sessionManager;
XnVPointControl pointControl;
XnVCircleDetector circleDetector;

PFont font;

String Ctext;
String dir;

boolean handsTrackFlag = false;
boolean circleTrackFlag = false;

PVector screenHandVec = new PVector();
```

```
PVector handVec = new PVector();  
int t = 0;  
float rad;  
  
PVector centerVec = new PVector();  
PVector screenCenterVec = new PVector();  
PVector v = new PVector();  
  
boolean serial = true;  
  
int Val_1, Val_2, Val_3;  
  
float temp01, temp02, temp03, temp04;  
  
void setup() {  
    kinect = new SimpleOpenNI(this);  
    kinect.setMirror(true);  
    kinect.enableDepth();  
    kinect.enableGesture();  
    kinect.enableHands();  
    sessionManager = kinect.createSessionManager("Wave", "RaiseHand");  
    pointControl = new XnVPointControl();  
    pointControl.RegisterPointCreate(this);  
    pointControl.RegisterPointDestroy(this);  
    pointControl.RegisterPointUpdate(this);  
    circleDetector = new XnVCircleDetector();  
    circleDetector.RegisterCircle(this);  
    circleDetector.RegisterNoCircle(this);  
    sessionManager.AddListener(pointControl);  
    sessionManager.AddListener(circleDetector);  
    size(kinect.depthWidth(), kinect.depthHeight());
```

```
smooth();  
font = loadFont("SansSerif-12.vlw");  
dir = "-";  
  
if (serial) {  
  
String portName = Serial.list()[0]; // This gets the first port  
myPort = new Serial(this, portName, 9600);  
}  
}  
  
void draw()  
{  
background(0);  
  
PVector centerL = new PVector(width/2, height/2);  
kinect.update();  
  
kinect.update(sessionManager);  
image(kinect.depthImage(), 0, 0);  
  
v.x = screenHandVec.x-centerL.x;  
v.y = screenHandVec.y-centerL.y;  
  
if (handsTrackFlag){  
  
drawHand();  
drawArrow(v, centerL);  
controlCar();  
}  
  
if (circleTrackFlag){  
  
drawCircle();  
}  
  
textDisplay();
```

```
if (serial){  
    sendSerialData();  
}  
}  
  
void controlCar() {  
    temp01 = screenHandVec.x-width/2;  
    temp02 = height/2- screenHandVec.y;  
    temp03 = int(map(temp01, -width/2, width/2, 0, 255));  
    if (temp03 < 75) {  
        Val_1 = 1;  
    }  
    if (temp03 > 175) {  
        Val_1 = 2;  
    }  
    if ((temp03 > 75) && (temp03 < 175)) {  
        Val_1 = 0;  
    }  
    temp04= int(map(temp02, -height/2, height/2, -255, 250));  
    if (temp04 > 65) {  
        Val_2 = 1;  
    }  
    else if (temp04 < -65) {  
        Val_2 = 2;  
    }  
    else {  
        Val_2 = 0;  
    }  
}
```

```
}

}

void drawHand() {
stroke(255, 0, 0);
pushStyle();
strokeWeight(6);
kinect.convertRealWorldToProjective(handVec, screenHandVec);
point(screenHandVec.x, screenHandVec.y);
popStyle();
}

void drawCircle() {
noFill();
strokeWeight(6);
stroke(0, 0, 255);
ellipse(screenCenterVec.x, screenCenterVec.y, 2*rad, 2*rad);
textAlign(LEFT);
}

void drawArrow(PVector v, PVector loc){
pushMatrix();
float arrowsize = 4;
translate(loc.x, loc.y);
stroke(255, 0, 0);
strokeWeight(2);
rotate(v.heading2D());
float len = v.mag();
line(0, 0, len, 0);
}
```

```

line(len, 0, len-arrowsize, +arrowsize/2);
line(len, 0, len-arrowsize, -arrowsize/2);
popMatrix();
}

void textDisplay(){
int value;
if(Val_2 == 0) {
value=0;
}
else {
value = Val_3;
}
text("Direction: "+dir, 10, kinect.depthHeight()-50);
if ((Val_2 == 1) && (Val_1 == 0)) {
dir ="N";
}
if ((Val_2 == 2) && (Val_1 == 0)) {
dir="S";
}
if ((Val_2 == 0) && (Val_1 == 1)) {
dir="W";
}
if ((Val_2 == 0) && (Val_1 == 2)) {
dir="E";
}
if ((Val_2 == 1) && (Val_1 == 2)) {

```

```
dir="NE";
}

if ((Val_2 == 2) && (Val_1 == 2)) {
    dir="SE";
}

if ((Val_2 == 2) && (Val_1 == 1)) {
    dir="SW";
}

if ((Val_2 == 1) && (Val_1 == 1)) {
    dir="NW";
}

void sendSerialData() {
    myPort.write('S');
    myPort.write(Val_1);
    myPort.write(Val_2);
    myPort.write(Val_3);
}

void onPointCreate(XnVHandPointContext pContext) {
    println("onPointCreate:");
    handsTrackFlag = true;
    handVec.set(pContext.getPtPosition().getX(),
    pContext.getPtPosition().getY(),
    pContext.getPtPosition().getZ());
}

void onPointDestroy(int nID) {
```

```
println("PointDestroy: " + nID);

handsTrackFlag = false;

}

void onPointUpdate(XnVHandPointContext pContext) {

handVec.set(pContext.getPtPosition().getX(),
pContext.getPtPosition().getY(),
pContext.getPtPosition().getZ());

}

void onCircle(float fTimes, boolean bConfident, XnVCircle circle) {

println("onCircle: " + fTimes + " , bConfident=" + bConfident);
circleTrackFlag = true;
t++;
centerVec.set(circle.getPtCenter().getX(),
circle.getPtCenter().getY(), handVec.z);
kinect.convertRealWorldToProjective(centerVec, screenCenterVec);
rad = circle.getFRadius();

}

void onNoCircle(float fTimes, int reason) {

println("onNoCircle: " + fTimes + " , reason= " + reason);
circleTrackFlag = false;
t = 0;
}
```

-النص البرمجي في بيئة Arduino IDE-

```
float val_1, val_2, val_3;  
int left_f = 13;  
int left_r = 12;  
int right_r = 11;  
int right_f = 10;  
void setup() {  
    Serial.begin(9600);  
    pinMode(left_f, OUTPUT);  
    pinMode(left_r, OUTPUT);  
    pinMode(right_f, OUTPUT);  
    pinMode(right_r, OUTPUT);  
}  
void loop() {  
    if(Serial.available()){  
        char val = Serial.read();  
        if(val == 'S') {  
            val_1 = Serial.read();  
            val_2 = Serial.read();  
            val_3 = Serial.read();  
        }  
        move forward//  
        if (val_2==1){  
            digitalWrite(right_r,HIGH);  
            digitalWrite(left_r,HIGH);  
        }
```

```
}

move backward//  
else if (val_2==2){  
digitalWrite(right_f,HIGH);  
digitalWrite(left_f,HIGH);  
}  
  
TurnRight//  
else if (val_1==2){  
digitalWrite(left_r,HIGH);  
digitalWrite(left_f,LOW);  
}  
  
TurnLeft//  
else if (val_1==1){  
digitalWrite(right_r,HIGH);  
digitalWrite(right_f,LOW);  
}  
  
stop//  
else if (val_1==0 or val_2==0){  
digitalWrite(right_f,LOW);  
digitalWrite(left_f,LOW);  
digitalWrite(right_r,LOW);  
digitalWrite(left_r,LOW);  
}  
  
delay(100);  
}
```

الصعوبات:

من أهم الصعوبات والتحديات التي واجهتني هي إيقاف شركة Microsoft لمشروع Kinect Camera وسحبها من الأسواق وبالتالي إيقاف تطوير خدماتها البرمجية مما شكل صعوبة في العمل على البيئة البرمجية المستخدمة وصعوبة في البحث عن المكتبات المستقادة منها في هذا المشروع.

تم العمل على الإصدار الأول من Microsoft Kinect وذلك بسبب عدم توفر الإصدار الثاني في السوق.

قلة توافر مواد العمل والشرايح التي تحتاجها لعملية التصميم.

الآفاق المستقبلية:

تزويد الروبوت بحساسات ومشغلات إضافية تتناسب مع المهام التي يمكن تكليفه بها، حيث تم التركيز في هذا المشروع على منهجية التحكم بحركة الروبوت ولم يتم تكليف الروبوت للقيام بأي مهمة محددة.

استبدال الحاسب الشخصي بالراسبيري Pi.

تطوير تقنية الاتصال اللاسلكي بين الحاسب والأردوينو، استخدام تقنية Wi-Fi بدلاً من البلوتوث.

تطوير البنية البرمجية لكل لجعل التحكم بالروبوت ممكناً من خلال حركة الجسد ككل وليس مقتضراً على حركة اليد.

المراجع:

-مقرر الدكتور عبدالقادر جوخدار في هندسة الروبوت

-مقرر الدكتور عبدالقادر جوخدار في أنظمة الميكاترونكس

-مقرر الدكتورة فاطمة قداد في معالجة الصورة

-مقرر الدكتور يمان جانات في التصميم والتصنيع باستخدام الحاسب

Robotics Vision & Control book by Peter Corke-

Arduino and Kinect Projects book by Enrique Ramos and Ciriaci Diaz-

Programming Arduino: Getting Started with Sketches book by Pedri Munoz-

Arduino Robotics book by Josh Adams-

Making Things See:3D Vision with Kinect book by Greg Borenstein-

L298N Dual H-Bridge Datasheet-

Bluetooth Module HC-05 Datasheet-

Kinect Gesture Control Paper by Sayali Patil-

A Survey of Applications and Human Motion Recognition with Microsoft – Kinect by Roanna Lun and Wenbing Zhao

Kinect hands tracking Paper by Leonardo Merza-

Processing and OpenNI book by Sourosh Falahati-

OpenNI and NITE Libraries by Max Rehiner-

Introduction to Mobile Robot Control book by S.G. Tzafestas-

Learning Processing by Daniel shiffman-

