

المسابقة السنوية الرابعة للروبوت

المسابقة الجامعية المفتوحة

فئة المشاريع التطبيقية

الروبوت في خدمة البلديات

R.U.T Team



أعضاء الفريق:

راما الصوص

محمد جعفر مرتضى

محمد علي هرملاني

هادي الشعار

المدرّب:

المهندس عبدالعزيز النحاس

فهرس المحتويات:

1.	مقدمة	3
1.1.	مبررات المشروع:	3
1.2.	فكرة المشروع:	4
2.	المبادئ النظرية:	7
1.2.	ذكاء السرب Swarm intelligence:	7
2.2.	مقاربة مستعمرة النمل Ant Colony Optimization (ACO):	8
3.2.	نظرية البيان Graph Theory:	9
4.2.	انترنت الأشياء Internet Of Things:	11
1.4.2.	شريحة ESP8266:	12
2.4.2.	شريحة الراسبيري باي Raspberry Pi:	13
3.4.2.	واجهة برمجة التطبيقات Application Programming Interface:	15
5.2.	سجلات الإزاحة:	16
3.	التطبيق العملي:	19
1.3.	تمثيل الطريق ببيان:	19
2.3.	المخططات الصندوقي للروبوت المسؤول عن جمع القمامة:	20
3.3.	خوارزمية عمل روبوت جمع القمامة:	23
4.3.	المخطط الصندوقي للحاوية الذكية:	25
5.3.	آلية عمل الحاويات الذكية:	25
6.3.	المراقبة الدائمة باستخدام تطبيق الأندرويد:	26
7.3.	ملاحظة حول الارسل والاستقبال:	26
4.	التطلعات والرؤى المستقبلية:	29



1.4. إنشاء بيان للبلدية:.....29

2.4. روبوت السرب:30

3.4. الحاويات الذكيّة:.....32

المراجع: 33

الملحقات:34



الفصل الأول

المبادئ النظرية

Chapter 1



1. مقدمة:

1.1. مبررات المشروع:

تعاني الكثير من الدول من مشكلة النفايات المتراكمة في المكبات، وتظهر هذه المشكلة خصوصاً في الدول النامية التي مازالت تتبع طرق قديمة نسبياً في عمليات جمع النفايات والتخلص منها. ونعني بالنفايات المواد التي لم تعد ذات فائدة بعد أن تم استعمالها لمرة واحدة أو عدة مرات، أو هي مواد نتجت من عملية معينة سواء كانت بيولوجية أو صناعية. وفي حال عدم التعامل معها بشكل ملائم تشكل خطراً على الصحة وتهدداً للبيئة. وخصوصاً إن كنا نتحدث عن النفايات الطبية التي تنتج عن المشافي أو النفايات الصناعية .. الخ وأيضاً النفايات المنزلية لانتقل خطورة عن البقية.

تهدد النفايات أولاً الأشخاص الذين يقومون بجمعها، مثل عمال النظافة وعمال مرادم النفايات وعمال معامل ترميد (حرق النفايات)، وثانياً جمهور الناس خاصة عندما تتجمع وتتراكم في منطقة معينة، نتيجة مثلاً إضراب عمال النظافة كما حصل في غزة - فلسطين في كانون الثاني من عام 2017 أو انتهاء عقود التعامل مع شركات ترحيل النفايات كمشكلة لبنان التي برزت عام 2016 نتيجة انتهاء مدة العقد بين الحكومة اللبنانية وشركة سوكلين وعدم تجديدها وغالباً يلجأ السكان إلى حرق النفايات بطرق بدائية تجنباً لتراكمها وهو أمر لا يقل خطورة عن تراكمها، فبحسب تقرير صادر عن منظمة حقوق الإنسان Human Rights Watch بعنوان "كأنك تستنشق موتك": المخاطر الصحية لحرق النفايات في لبنان" [1] والمؤلف من 53 صفحة وثقت دراسات علمية المخاطر التي يشكلها دخان حرق النفايات المنزلية في الهواء الطلق على صحة الإنسان. الأطفال والمسنون مهددون بشكل خاص. وأوضحت المنظمة في التقرير على أنه يجب على لبنان إنهاء حرق النفايات في الهواء الطلق وتطبيق استراتيجية وطنية مستدامة لإدارة النفايات، تتماشى مع أفضل الممارسات في مجالي البيئة والصحة العامة ومع القانون الدولي. وقد بلغ أغلب السكان الذين تمت مقابلتهم عن آثار صحية رَدّوها إلى حرق النفايات المفتوح وتنشق الدخان المنبعث منه، منها مشاكل تنفسية مثل الانسداد الرئوي المزمن، السعال، تهيج الحلق، والربو. تتفق هذه الأعراض مع التعرض إلى حرق النفايات في الهواء الطلق، الموثقة آثاره في العديد من المؤلفات العلمية .

سبب آخر لتراكم النفايات هو وجود مكبات القمامة في حارات ضيقة، حيث تعاني العديد من حارات دمشق القديمة من هذه المشكلة بشكل خاص، حيث يتواجد المكب في منطقة ضيقة يصعب على شاحنة

القمامة التقليدية الدخول إليها ويجب عندها أن يتم انتظار سيارة صغيرة أخرى وهي غالباً من نوع شاحنة صغيرة وغير متخصصة بنقل القمامة مما يترتب عليه تشويه في المظهر الحضاري ونشر روائح كريهة وخطر على البيئة. ولهذا الأمر جوانب سلبية عديدة، فأولاً يلقي بأعباء مادية لا يستهان بها على عاتق الحكومة السورية، من حيث أجور عمال النظافة، تكلفة السيارات الشاحنة الصغيرة، وأيضاً الأعباء الإدارية وصعوبة مراقبة العمل في مثل هذه الحالات. وثانياً المخاطر الصحية ففي الحارات الضيقة تجمع صغير للقمامة سيؤدي إلى نشر العديد من الأمراض والأوبئة ونشر الروائح الكريهة، ناهيك عن تشويه المظهر الحضاري وهذا أمر جد خطير عند الأخذ بعين الاعتبار أن شوارع دمشق القديمة تعتبر جاذب سياحي مشهور.

1.2. فكرة المشروع:

من هنا أتت فكرة مشروعنا والتي يمكن أن نلخصها بسرب روبوتات يقوم بشكل آلي ومستمر بجمع القمامة من حاويات القمامة الموجودة في البلدية لتقوم بنقلها إلى المكب الخاص.

اعتمدنا في مشروعنا على خوارزمية تعرف بخوارزمية الـ Swarm أو خوارزمية السرب وبالأخص نوع محدد منها ما يعرف بخوارزمية مستعمرة النمل أو Ant Colony Optimization مع ادخال تعديلات لتلائم الغرض المطلوب وهذا لنشر مجموعة من الروبوتات المختصة بجمع القمامة بشكل آلي حيث قمنا بالاعتماد على فرع من الرياضيات يدعى بنظرية البيان Graph Theory باعتبار الطرق المنتشرة ضمن بلدية ما مجموعة من العقد Nodes و الحواف Edges وقمنا باستخدام شرائح الأردوينو Arduino لقيادة الروبوتات وباستخدام شرائح ESP قمنا بوصلها مع الأردوينو أتحنا لها إمكانية الوصول لشبكات الانترنت اللاسلكي WiFi، للاستفادة من مبادئ انترنت الأشياء (Internet Of Things (IOT فأصبح بإمكان الروبوت أن يتصل مع المخدم Server لطلب أوزان الطرق ليقوم بعدها الروبوت باختيار الطريق الأنسب له بحسب أوزان هذه الطرق.

بالنسبة للروبوتات قمنا باختيار شرائح Arduino من نوع UNO وقمنا بوصلها مع شرائح ESP-01 باستخدام الوصل التسلسلي، بالنسبة للمخدم Server قمنا باستخدام شريحة راسبيري باي Raspberry Pi 3 مزود ببرنامج (WEB SERVER) Apachi 2 وقمنا بكتابة عدد من واجهة برمجة التطبيقات Application Programming Interface (اختصاراً API) باستخدام لغة البرمجة Python واستخدمنا بروتوكول HTTP (Hypertext Transfer Protocol) لإجراء طلبات GET. ولإتاحة مراقبة دائمة لأماكن الروبوتات قمنا ببرمجة تطبيق أندرويد بواجهة بسيطة وسهلة الاستخدام يسمح بعرض أماكن وجود

الروبوتات. لأنّ هدف الأساسي لمشروعنا كان اثبات أنّ صحة عمل المبدأ وإجراء محاكاة فإنّنا قمنا باستخدام مجموعة مرسل/مستقبل أشعة تحت حمراء لإرسال أمر التفريغ للحاويات حيث قمنا بوصل حاويات القمامة إلى شريحة Arduino من نوع Mega حيث تتألف كل حاوية قمامة من مجموعة من الثنائيات المصدرة للضوء المتصلة إلى سجل إزاحة بطول 4bit ومجموعة مرسل/مستقبل أشعة تحت حمراء IR وطريقة تعبئة الحاويات بشكل عشوائي.

راعينا الأمور المتعلقة بالسلامة فيما يتعلق بمشروعنا بحيث كل روبوت مزود بحساس لقياس المسافة Ultrasonic لتجنب الاصطدام بأي عائق وبحيث يقوم الروبوت باتباع خوارزمية معينة عند مواجهة عائق ما سنقوم بشرحها لاحقاً. وكل روبوت يقوم بإعلام المخدم عن سلوكه لطريق ما لتجنب بقية الروبوتات سلوك هذا الطريق.

سنقوم بالفصول القادمة بشرح أولاً التقنيات المستخدمة في مشروعنا بالإضافة إلى الخوارزميات المستخدمة والمبادئ النظرية، ثم سنشرح التطبيق العملي الذي قمنا بإنجازه والنتائج التي توصلنا لها بالإضافة إلى الرؤى والتطلعات المستقبلية للمشروع ومناقشة إمكانية نقله من المحاكاة



الفصل الثاني

المبادئ النظرية

Chapter 2



2. المبادئ النظرية:

في هذا القسم سنستعرض الأسس النظرية القابعة وراء بعض الخوارزميات والتقنيات المستخدمة في المشروع:

1.2. ذكاء السرب Swarm intelligence:

عام 1989 قدم كل من العالمان Gerardo Beni and Jing Wang مصطلح ذكاء السرب في سياق النظم الروبوتية الخلوية أو ما يعرف اليوم بـ (CEBOT) cellular robotic system [2] هو مصطلح يصف السلوك الجماعي للنظم اللامركزية ذاتية الانتظام، سواء كانت طبيعية أم اصطناعية، أي أنه يصف مجموعة من الأغراض المتفاعلة فيما بينها بحيث تحقق انتظاماً ذاتياً، ويمكننا القول أن كل من هذه الأغراض يطبق خوارزمية قد تكون غبية بذاتها ولكن عندما تجتمع مع أغراض أخرى تطبق نفس الخوارزمية يصبح النظام ذاتي الانتظام ويعمل بفعالية عالية ليشكل نظاماً ذكياً. مع بداية تسعينات القرن الماضي كان الاتجاه إلى محاكاة النظم الكائنات الحية الأقل ذكاءً والتي لها إمكانيات محدودة كالنمل والطيور والأسماك والتي في نفس الوقت تبدي سلوكاً اجتماعياً شديد الذكاء ففي عام 1990 اقترح العالم الإيطالي ماركو دوريجو (Marco Dorigo) الخوارزمية "Ant Colony Optimization (ACO)" [4] التي تحاكي مستعمرات النمل. كما في عام 1995 اقترح كلا من العالمان Russell Eberhart و James Kennedy الخوارزمية Particle Swarm Optimization والتي تعتمد بشكل رئيسي على محاكاة أسراب الطيور. الخوارزمتان السابقتان كانتا بداية لفرع جديد من فروع الذكاء الصناعي "SI" Swarm Intelligence.

إذا يمكننا القول أن النظام المعتمد على SI يتألف من عدد من العملاء Agents وأحياناً يطلق عليها اسم Boids التي تتفاعل فيما بينها ومع البيئة المحيطة بها.

مثال بسيط على تطبيق خوارزميات ذكاء السرب هو محاكاة الحشود: حيث تستخدم تقنيات السرب في أفلام الحاسوب لتحريك حشود ضخمة من المخلوقات بشكل طبيعي عن طريق قواعد بسيطة للتفاعل المحلي بينها.

تطبيق خوارزميات ذكاء السرب في مجال الروبوتات خلق فرع جديد يدعى بسرب الروبوت أو Swarm Robotics، والذي يمكن أن نعرفه بـ: بأنه محاولة تنسيق سلوك نظام يتألف من عدد من الروبوتات بحيث تكون غالباً ذات تركيب فيزيائي بسيط بحيث ينبثق السلوك الجماعي كنتيجة للتفاعل بين الروبوتات

نفسها وبين كل روبوت والبيئة المحيطة به، أي أنه تطبيق لخوارزميات ذكاء السرب في مجال الروبوتات.



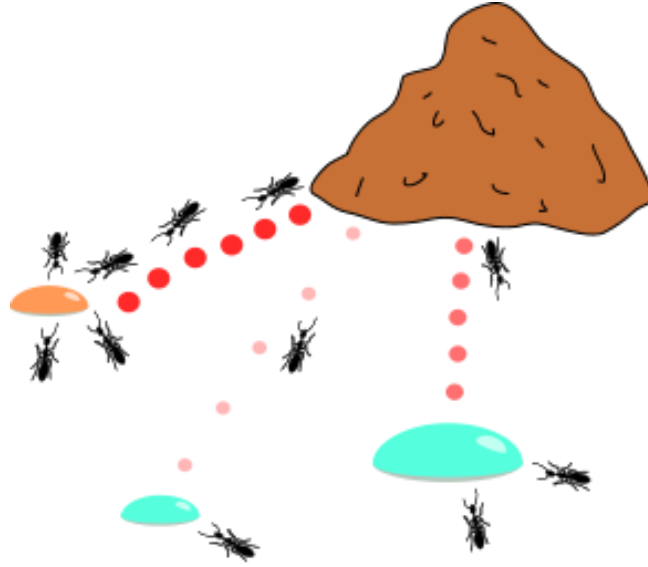
الشكل (1-2) - مجموعة روبوتات بسيطة تشكل سرب روبوتات

اليوم يوجد عدد لا بأس فيه من الخوارزميات المطبقة في مجال ذكاء السرب ونذكر منها على سبيل المثال لا الحصر مقارنة مستعمرة النمل، ومقاربة سرب الذرات وغيرها وقد أرتأينا في مشروعنا اختيار مقارنة مستعمرة النمل لهذا سنتوسع بشرحها بشكل خاص.

2.2. مقارنة مستعمرة النمل (ACO) Ant Colony Optimization:

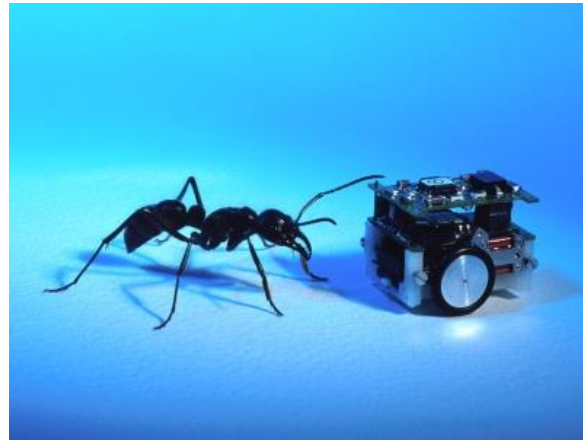
في علوم الحاسب وبحوث العمليات تعتبر خوارزمية مستعمرة النمل طريقة لحل المسائل التي تعتمد على الرسوم البيانية لإيجاد الطريق الأمثل ضمن جملة من الاحتمالات بطريقة تحاكي عمل النمل.

في الطبيعة وجد أنّ بعض أنواع النمل تقوم بالخروج بشكل عشوائي من المستعمرة بحثاً عن الطعام وتقوم بنشر الفرومونات أثناء الذهاب لتعتمد عليها في العودة. مع خروج نمل جديد من المستعمرة يقوم باتباع آثار فرومونات النمل العائد، ولكن ولأنّ الفرومونات تقوم بالاضمحلال مع الزمن فإنّ النمل الذي عثر على الطريق الأقصر ستظل فرومونات هي الأقوى وهذا ماسيجعل بقية النمل يسلك هذا الطريق.



الشكل (2-2) - شكل توضيحي للخوارزمية التي يطبقها النمل

تطبيق هذا المبدأ على مجال الروبوتات يتطلب أن نجعل لكل طريق وزن خاص به الأمر الذي يحاكي آثار الفرومونات التي يتركها النمل. ويوجد أكثر من طريقة اليوم لاختيار الطريق الأمثل احداها على سبيل المثال التي تعتمد على مقارنة أوزان الطرق واختيار الطريق ذو الوزن الأكبر.



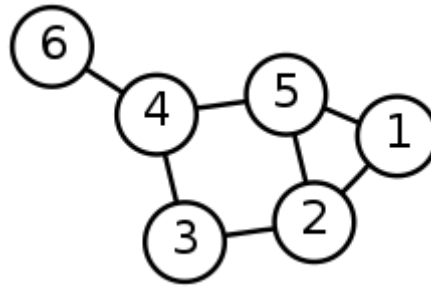
الشكل (2-3) - النمل والروبوتات

إذا خوارزمية النمل تعتمد وبشكل كبير على البيان وهو ما سنقوم بشرحه في الفقرة التالية.

3.2. نظرية البيان Graph Theory:

يعد البحث الذي كتبه ليونهارد أويلر ونشره في عام 1736 حول موضوع جسور كونيجسبرغ السبعة (مسألة على إيجاد مسار ضمن المدينة بحيث يتم العبور على كل جسر مرة واحدة فقط. لم يكن يمكن الوصول إلى الجسر بأي طريقة أخرى غير الجسور وكان يجب عبور الجسر كاملاً في كل مرة).

أول بحث في التاريخ في نظرية المخططات [3]. هذا البحث بالإضافة إلى المقالة التي كتبها فاندريوموند عن مسألة الفارس، بالإضافة إلى العمل الذي قام به غوتفريد لايبنتز في ذوضع علاقات لعدد الرؤوس بالأضلاع وأوجه متعدّدات السطوح المحدبة تعتبر بدايات لعلم الطوبولوجيا. وبالتعريف هي نظرية في الرياضيات وعلوم الحاسب، تدرس خواص المخططات حيث يتم تمثيل مجموعة كائنات تدعى رؤوساً، ترتبط ببعضها بأضلاع و تدعى أحياناً أقواساً، يمكن أن تكون موجهة أي مزودة باتجاه (تستخدم الاسم بدل الأضلاع) أو بدون اتجاه (أضلاع فقط). التمثيل لهذا المخطط يكون على الورق بمجموعة نقاط تمثل الرؤوس متصلة بخطوط هي حروف (ضلاع أو أسهم) المخطط. رياضياً يُمكن أن يُعطى المخطط عبر مصفوفة المجاورة (Adjacency Matrix).



الشكل (4-2) - بيان بسيط غير موجه

وفيما يلي نسرد تعاريف معظم المصطلحات التي يمكن أن تمر معنا في نظرية البيان:

• البيان: Graph:

هو زوج مرتب $G = (V, E)$ يشمل مجموعة من الرؤوس (أو العقد) $V \{a, b, c, \dots\}$ ومجموعة من الأضلاع (أو الحروف) $E \{\{a, c\}, \{b, d\}, \{a, d\}, \dots\}$ والتي هي بدورها مجموعة من الثنائيات الجزئية وفي حال كانت غير مرتبة سمي البيان بالبيان البسيط غير الموجه أما في حال كانت مزودة باتجاه فتصبح مرتبة ويسمى بالبيان الموجه.

• ترتيب البيان $|V|$:

هو عدد العقد أو الرؤوس فمثلاً بيان يحوي ثلاث عقد هو بيان من المرتبة الثالثة.

• حجم البيان $|E|$:

هو عدد أضلاع (حروف) البيان.

• درجة العقدة:

في البيان غير الموجه هي عبارة عن عدد الأضلاع المتصلة بالعقدة. أما بالنسبة للبيان الموجه فهي على نوعين: درجة الدخول للعقدة وهي عدد الأضلاع الداخلة للعقدة، ودرجة الخروج: وهي عدد الأضلاع الخارجة من العقدة.

• الارتباط والجوار:

إذا كانت عقدتين من مخطط مرتبطتان بضلع على الأقل نقول أنهما متجاورتان. بالطبع نظرية البيان نظرية واسعة ولها كثير من التطبيقات ونستطيع التوسع في شرحها كثيراً ولكننا اخترنا أن نذكر ما يهمنا منها في هذا المشروع.

4.2. انترنت الأشياء Internet Of Things:

نعني بانترنت الأشياء شبكة الأجهزة الفيزيائية كالمركبات المتحركة، الأجهزة المنزلية وغيرها من التجهيزات التي تكون مدمجة مع الدارات الإلكترونية، أنظمة التشغيل، حساسات ومحرركات والأهم طريقة اتصال تسمح لهذه الأجهزة بتبادل البيانات وكل جهاز يمكن تميزه بسهولة باستخدام نظامه الحاسوبي وقادر على التفاعل مع بنية شبكة الانترنت. تسمح IoT للأجهزة أن تُستشعر عن بعد وأن يتم التحكم به عن طريق الشبكة. عام ازداد عدد الأجهزة المعتمدة على انترنت الأشياء بنسبة 31% ويتوقع أن يصل عددها عام 2020 إلى حوالي 30 مليار جهاز.

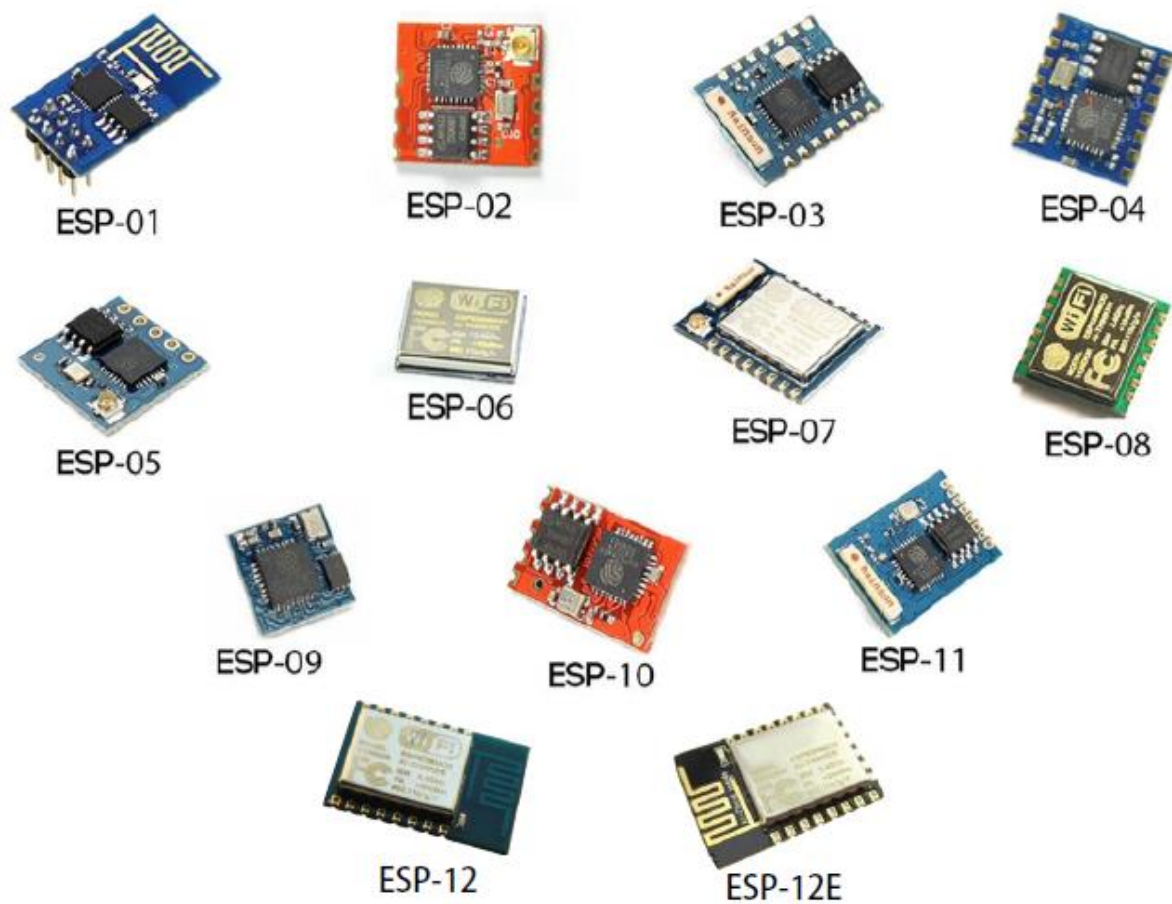


الشكل (5-2) - انترنت الأشياء

إذا الشرط الأول لانتترنت الأشياء أن تمتلك الأجهزة القدرة على الاتصال بشبكة الانترنت ولأنّ الأردوينو بمعظم إصداراتها (بعض الإصدارات تستطيع الاتصال بشبكات WiFi) لا تملك القدرة على الاتصال بشبكات الانترنت سواء السلكي أو اللاسلكي منها كان لابد من تأمين هذه الوسيلة والتي كانت عبارة عن شريحة تعرف بالـ ESP.

1.4.2. شريحة ESP8266:

هي عبارة عن شريحة WiFi مزودة بمتحكم صغيري Microcontroller وذات سعر منخفض بالمقارنة مع أداءها. تتوفر بعدة أنواع تختلف في مميزاتا وعدد أطرافها كما هو موضّح في الصورة التالية:

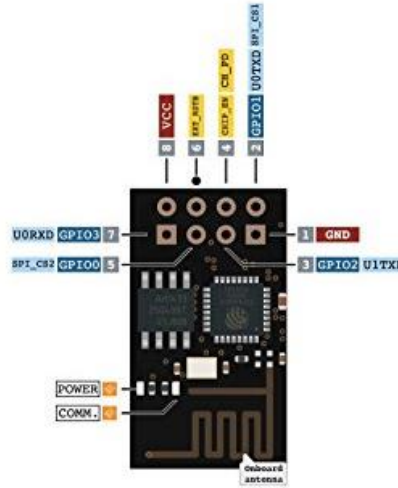


الشكل (2-6) - الأنواع المختلفة لشرائح ESP

في مشروعنا اخترنا شريحة ESP-01 والتي يتضح مميزاتا في الجدول التالي:

Name ⇅	Active pins ⇅	Pitch ⇅	Form factor ⇅	LEDs ⇅	Antenna ⇅	Shielded? ⇅	dimensions (mm) ⇅
ESP-01	6	0.1"	2x4 DIL	Yes	PCB trace	No	14.3 × 24.8

في بنية مشروعنا تلعب شريحة ESP8266 دور Client أو العميل ضمن الشبكة والمقصود بالعميل أنه الحاسب الذي يملك القدرة على الوصول إلى خدمة ما موجودة ضمن المخدم Server. والمقصود بالمخدم هو حاسب مادي مخصص لإنجاز خدمات لتؤد أغراض العملاء.



الشكل (2-7) - شريحة ESP-01

2.4.2. شريحة الراسبيري باي Raspberry Pi:

هي سلسلة من الشرائح الحاسوبية الصغيرة والتي لايتجاوز حجمها بطاقة الإئتمان حيث تمثل الشريحة حاسب متكامل تضم مكونات الحاسب التقليدية وهي وحدة المعالجة المركزية CPU معالج الرسوميات GPU مع ذاكرة الوصول العشوائي RAM بالإضافة إلى مداخل ومخارج من ذات أنواع متعددة، وتمتاز بعض إصدارات الشريحة (كالمستخدمة في مشروعنا) بميزة WiFi مدمج مما يجعل منها مناسبة لأغراض الشبكات.

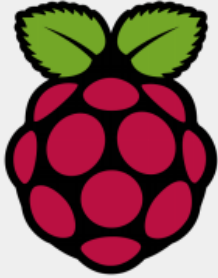


الشكل (2-8) - شرائح الراسبيري باي المختلفة

بالنسبة لوحدة المعالجة المركزية: تعتمد شرائح الراسبيري على معالجات ARM وهي معالجات قادرة على تشغيل نظم متكاملة ومعالجة فيديو وصوت ورسوم ثلاثية الأبعاد بسرعة عالية وتمتاز برخص ثمنها بالمقارنة مع أدائها، بالإضافة لكل ذلك فحجمها صغير ومتكاملة ضمن رقاقات صغيرة، فرقاقة Broadcom BCM2835 تحوي على CPU, GPU و RAM ومع ذلك مساحتها حوالي 1 سم² وهي أصغر بأربع مرات من متحكم دقيق من نوع PIC16f877.

تعتمد الشريحة في تشغيلها على أنظمة Linux مفتوحة المصدر حيث يتم تحميل التوزيع الخاصة بالشريحة على SD card. وكما نعلم فإن هذه الأنظمة مجانية ومفتوحة المصدر ما يعني أنه يمكنك استخدامها مجاناً ولك كامل الحرية في التعديل عليها كما تشاء، لأنه مفتوح المصدر ما يجعل الشيفرة المصدرية الخاصة به متاحة لك. توزيعه اللينكس الرسمية الخاصة بالراسبيري باي تسمى Raspbian ويمكن تحميلها مجاناً من الموقع الرسمي.

أخيراً شرائح الراسبيري باي تكون مزودة بمخرج HDMI وأيضاً مخرج للتلفاز RCA VIDEO OUT لأجهزة التلفاز القديمة، وتحتاج إلى 5 فولت لتعمل ويمكن تأمينها عن طريق مدخل mini USB وقد استخدمنا في مشروعنا شريحة Raspberry Pi 3 Model B. يبين الجدول التالي مواصفاتها:

	
Raspberry Pi 3 Model B	
Introduction Date	2/29/2016
SoC	BCM2837
CPU	Quad Cortex A53 @ 1.2GHz
Instruction set	ARMv8-A
GPU	400MHz VideoCore IV
RAM	1GB SDRAM
Storage	micro-SD
Ethernet	10/100
Wireless	802.11n / Bluetooth 4.0
Video Output	HDMI / Composite
Audio Output	HDMI / Headphone
GPIO	40
Price	\$35

3.4.2. واجهة برمجة التطبيقات Application Programming Interface:

هنا نتحدث عن نوع خاص وهو Web API وبالأخص من جهة المخدم وهو جزء من برنامج يلعب دور الوسيط بين مخدم الويب ومتصفح الويب. يمكن أن نستخدم المثال التالي لشرحه:

في المطعم يمكننا اعتبار المطعم هو المخدم والشخص الذي يريد أن يطلب الطعام الزبون، لا يمكن للزبون أن يتجه مباشرة للمطبخ ليطلب طعاماً، لهذا فهو بحاجة لوسيط بينهم ينقل "طلب" الزبون إلى المخدم ليحصل على الاستجابة والتي في مثالنا هي الطعام ويعيدها إلى الزبون، هذا الوسيط هو النادل في المطعم ويمكن اعتباره Web API. حيث يقوم الزبون بطلب API معين غالباً عن طريق بروتوكول HTTP موجود لدى المخدم ويمرر فيه بارامترت ليحصل بعدها على الاستجابة التي تكون على شكل XML أو JSON.

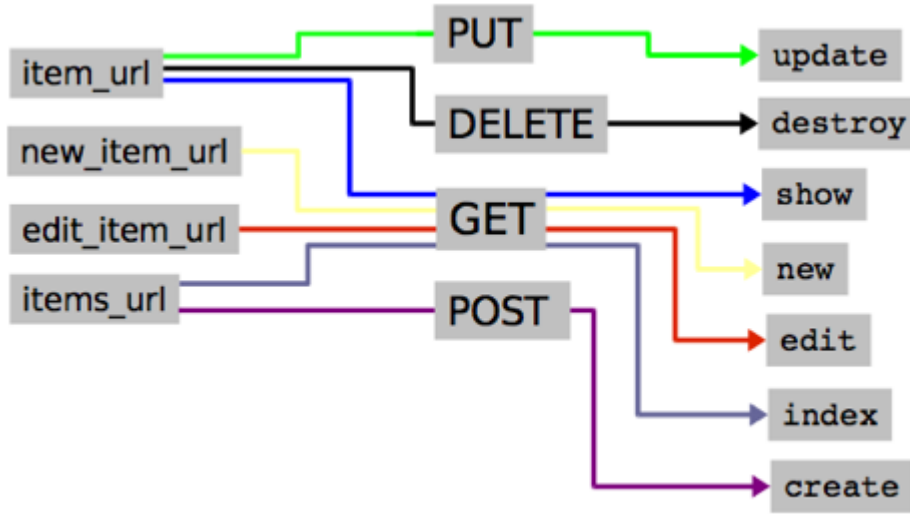
يوجد عدة عائلات اليوم للـ WEB API وقد اخترنا منها ما يعرف بـ REST \square state transfer. ويمكن أن نعرفه كما يلي:

هي طريقة للتخاطب بين الأنظمة في بيئة الويب تعتمد على بروتوكول HTTP وعلى الدوال المعروفة (GET | POST | PUT | DELETE) وعلى خاصية أن كل خدمة يتم وصفها والوصول إليها عن طريق رابط فريد . ظهرت في عام 2000 جزء بحث دكتورة للعالم Roy Fielding الفكرة عبارة عن تمثيل خدمة على موقع ويب برابط بحيث يقوم الطرف الثاني باستخدام هذه الخدمة عن طريق هذا الرابط سواء عن طريق (GET | POST | PUT | DELETE) وأغلب المواقع تستخدم GET وPOST. على حسب الخدمة. هذه الخدمة ترجع الناتج للطرف الثاني وذلك اما أن تكون على شكل (XML | JSON| ATOM| RSS) وهي ليست حكراً على هذه الصيغ إذ أنه يمكن إرجاع نص فقط.

ملاحظة:

- GET: لطلب بيانات من مصدر ما.
- POST: لإرسال بيانات لمصدر ما.
- PUT: لتحديث مصدر محدد.
- DELETE: لحذف مصدر ما وهذا الأمر ممكن كون هذا النوع من WEB API يعامل الأغراض على أنها Resources يمكن حذفها أو تدميرها.

يمكن كتابة الـ API بأي لغة برمجة معروفة وقد اخترنا لغة Python نظراً لسهولة التعامل معها وكونها تدعم العديد من الأنظمة.



الشكل (9-2) - مبادئ الـ RESTful

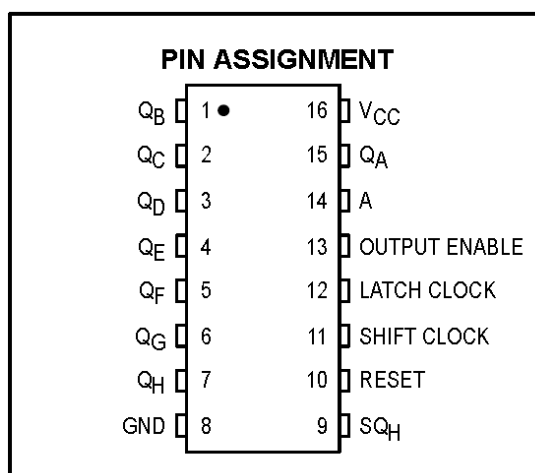
5.2. سجلات الإزاحة:

هو مجموعة متسلسلة من القلايات flip flop التي لها نفس دور الساعة CLK، وخرجها متصل بقلاب آخر، هذا يعمل على نقل البيانات بين المسجلات بتسلسل، وتنتقل البيانات عندما يتغير دور الساعة صعوداً أو هبوطاً.

تنقسم مسجلات الإزاحة إلى عدة أنواع و هي:

- دخل تسلسلي / خرج تسلسلي: SISO.
- دخل تسلسلي / خرج تفرعي: SIPO.
- دخل تفرعي / خرج تسلسلي: PISO.

في مشروعنا نريد أن نعطي قيم الحاويات على شكل دخل تسلسلي من الأردوينو لتظهر على شكل خرج تفرعي على الليدات. أي أننا استخدمنا سجل من نوع SIPO. وقد استخدمنا لهذا الغرض الدارة المتكاملة HC74595 والتي يبين الشكل التالي توزيع أقطابها:



حيث: **القطب 14**: هو المسؤول عن إدخال البيانات تسلسليا بحيث يتم إزاحة البيانات على مدخله. فيما **القطب 13**: وظيفته تمكين الخرج ففي حال أصبح الجهد عليه LOW سيظهر الخرج على الأقطاب. **القطب 12**: يقوم هذا القطب بإمساك البيانات وتخزينها. **القطب 11**: دخل الساعة عندما تكون في وضع مرتفع تسمح للبيانات على المدخل التسلسلي 14 بإزاحة البيانات.

طول هذا السجل 8 بت ولكن وباعتبار أننا لا نحتاج إلا إلى 4 بت فقط لكل حاوية (كل حاوية ستحتوي 4 ليدات) فإنّ سجل واحد سيكفي لحاويتين اثنتين.



الفصل الثالث

التطبيق العملي

Chapter 3



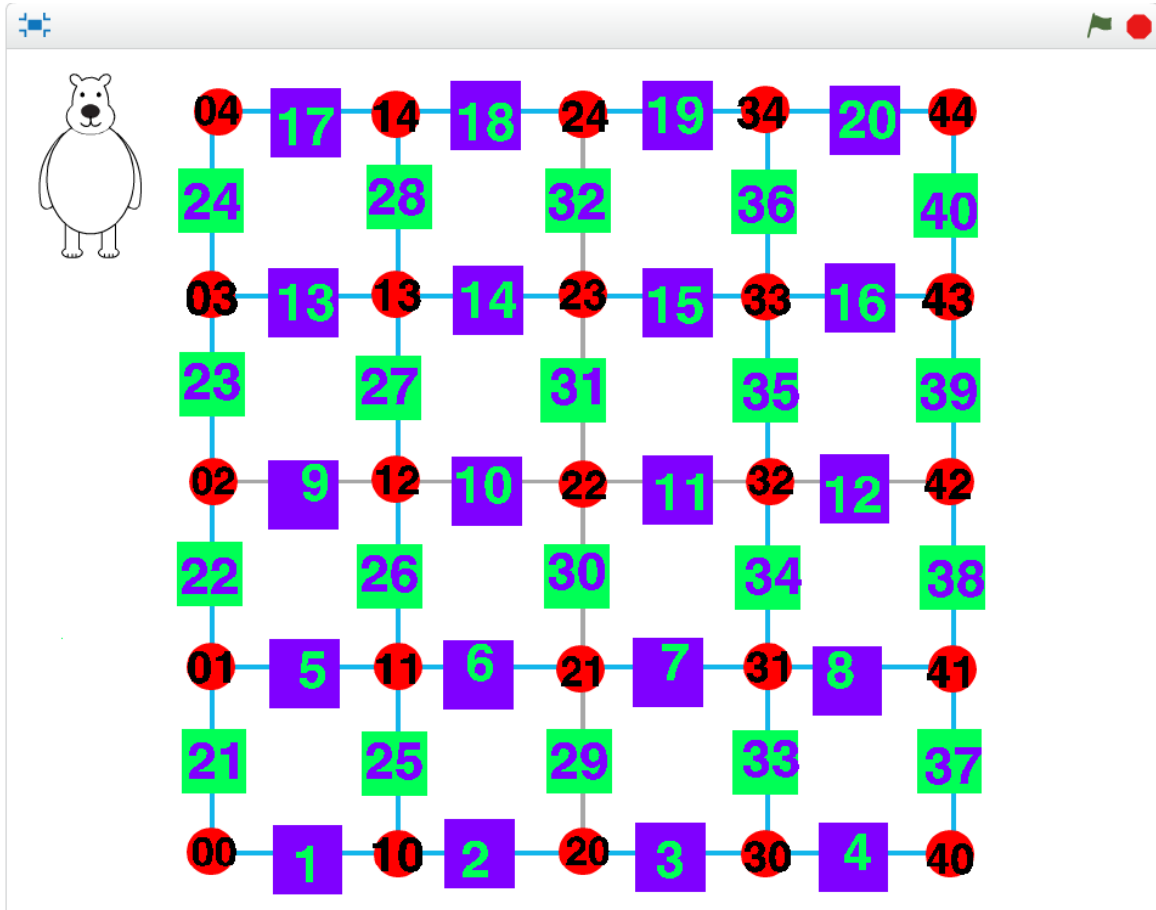
3. التطبيق العملي:

نستعرض في هذا القسم الإجراء العملي الذي قمنا به بناء على ما قمنا بشرحه ومناقشته في الفصل السابق.

1.3. تمثيل الطريق ببيان:

قمنا بتصميم حلبة من الخشب، وقمنا بتمثيلها ببيان من الدرجة: $|V| = 44$ وبحجم $E = 40$ استعنا بلغة البرمجة سكراتش لإجراء المحاكاة ورسم الطريق قبل تجربته على الأردوينو. (لغة سكراتش هي لغة برمجة بسيطة تعتمد على الغرافيكس من إنتاج جامعة MIT).

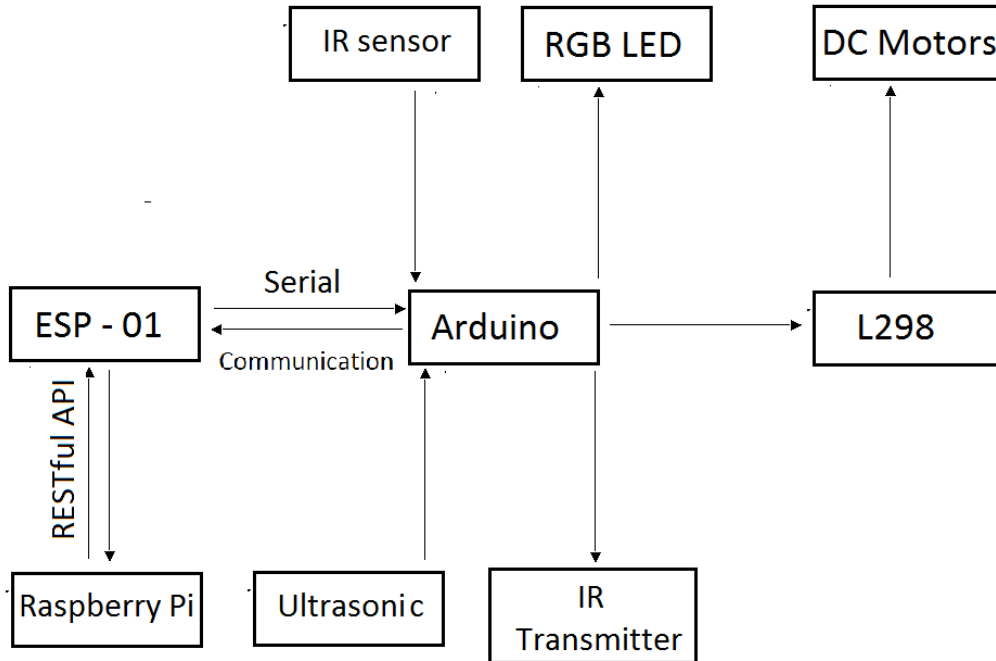
على الرغم من بساطة اللغة إلا أنها قد تكون قوية إذا استخدمت بشكل صحيح. يبين الشكل التالي المحاكاة التي قمنا بها:



الشكل (3-1) - المحاكاة باستخدام سكراتش

حيث اعتبرنا أنّ لكل طريق وزنان اثنان، الأول وزن حاويات القمامة التي فيه، أمّا الوزن الثاني فهو وزن الطريق وهو وزن تتم زيادته بشكل آلي عند مرور فترة زمنية ما ضمن المخدم، وعند مرور الروبوت من ضلع ما يقوم بتصفير وزن هذا الطريق، بهذه الطريقة نساهم في نشر الروبوتات بشكل أكبر، حيث أننا نعطي وزن للطرق التي لا تحوي حاويات قمامة.

2.3. المخططات الصندوقية للروبوت المسؤول عن جمع القمامة:



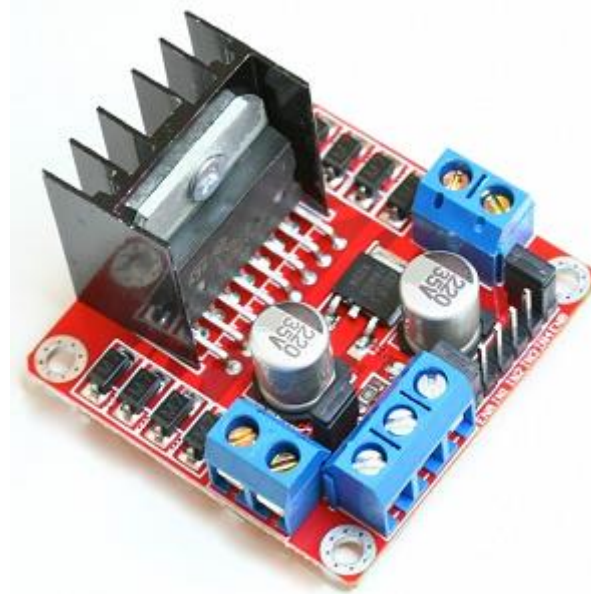
المخطط (1-3) - المخطط الصندوقي للروبوت السرب

يبيّن الشكل المخطط السابق المخطط الصندوقي لكل روبوت من السرب حيث:

كل روبوت يعتمد على شريحة أردوينو UNO كدالة لقيادة الروبوت بحيث تتصل إلى:

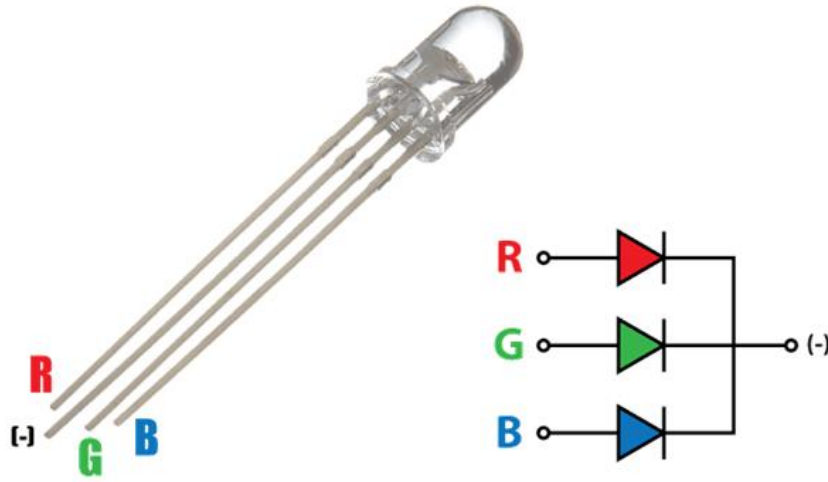
- دارة L298N وهي عبارة عن شريحة تعتمد على الدارة المتكاملة L298 (جسر H) وتستخدم

لقيادة محركين تيار مستمر اثنين. يبيّن الشكل التالي صورة الدارة:



الشكل (3-2) - دائرة L298N

- تتصل شريحة الأردوينو بثنائي مصدر للضوء من نوع RGB أي (RED GREEN BLUE) واستخدمناه للدلالة على درجة امتلاء الروبوت بالقمامة.



الشكل (3-3) - ليد الـ RGB

- ثنائي مصدر للضوء من نوع IR (بطول موجة 950 نانومتر) ونستخدمه لإرسال أمر التفريغ للحاويات.
- حساس تتبع خط من نوع IR.
- حساس قياس مسافة Ultrasonic، وهو يعمل بطريقة المقاطعة حيث في حال لاقى الروبوت أي عائق في طريقه فإنه سيتوقف مباشرة أيًا يكن العمل الذي يؤديه، يقف الروبوت وينتظر أن يبتعد العائق عن طريقه.

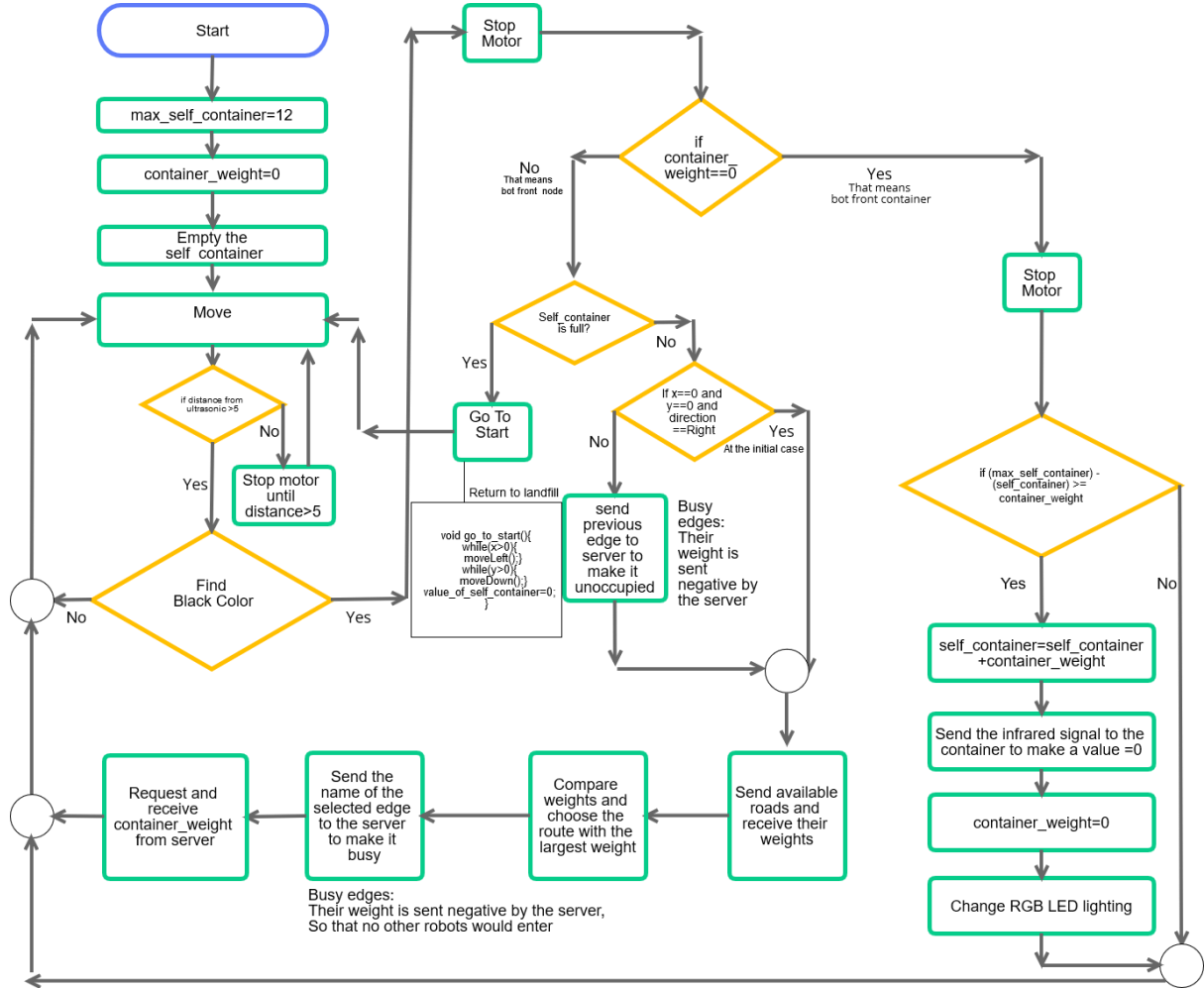
- باستخدام الاتصال التسلسلي تتصل شريحة الأردوينو مع شريحة الـ ESP-01. وتستطيع تمرير ما تريده لها. شريحة الـ ESP تستطيع التواصل مع المخدم عن طريق طلب واحد من اثنين من الـ API:

الأول: باسم: RorS يرسل الروبوت رقم الحافة التي وصل لها باستخدام طلب GET ويرسل معها إما: 0 ليخبر المخدم أنه انتهى من هذا الطريق فيقوم السيرفر بتصفير وزن الطريق الخاص به، أو 1 ليطلب من المخدم وزن الحاويات الخاص بالطريق ويقوم المخدم بإشغال هذا الطريق. لأنّ الـ ESP تملك متحكم صغير خاص بها فهي تتم برمجتها بشكل مستقل باستخدام مكاتب خاصة بها، ويتم الارسال والاستقبال بينه وبين الأردوينو عن طريق الاتصال التسلسلي فقط.

الثاني: باسم: getEdges وفيه يرسل الروبوت طلب GET ليرسل دلائل الحافات المتصلة بالعقدة التي وصل إليها ليحصل على وزن كل طريق، حيث يقوم بارسال وزني الطريق والحاويات مجموعين.

3.3. خوارزمية عمل روبوت جمع القمامة:

يبين المخطط (2-3) خوارزمية عمل روبوت السرب:

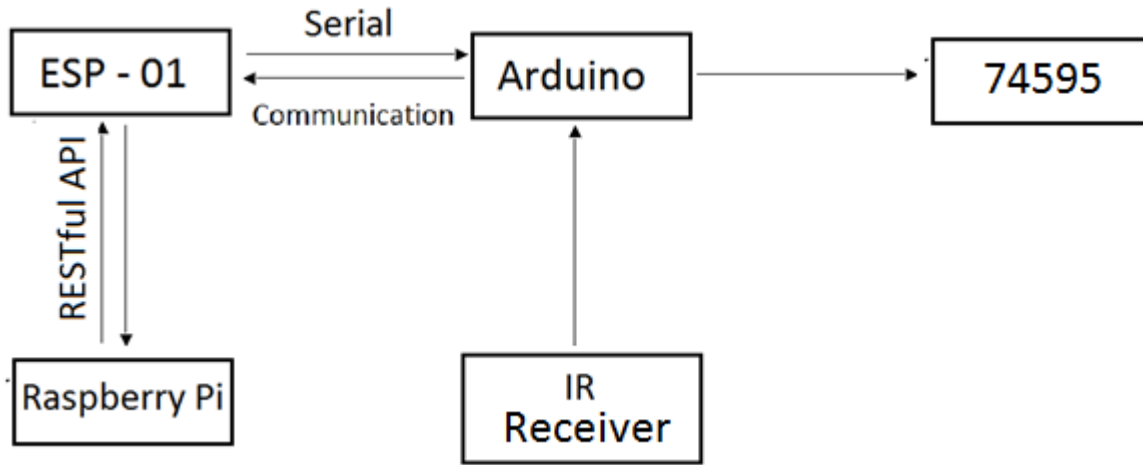


حيث:

- عند البدء يقوم الروبوت بتحديد القيمة العظمى للحاوية الذاتية له ويقوم بإعطاء قيمة 0 للمتحول container_weight الذي يعبر عن قيمة الحاوية في الطريق (الحافة) الذي يمشي فيه الروبوت، ثم يقوم بتفريغ الحاوية الذاتية في المكب المخصص له والموجود عند العقدة رقم 00. وفي حين أن الروبوت الحالي لا يملك ميكانيك تفريغ الحاوية فإنه يكتفي بتصفير قيمة المتغير المسؤول عن الامتلاء وتغيير إضاءة الليد RGB الذي يشير إلى الامتلاء.
- يبدأ الروبوت في المسير وفي حال واجه عائق (تحسس مسافة أقل من 5 سم) سيتوقف بانتظار أن تتغير هذه المسافة أي يبتعد العائق.

- سيستمر الروبوت في المسير إلى أن يتحسس لوناً أسود على حساس تتبع الخط. فعندها يوقف المحركات ويختبر قيمة المتحول `container_weight` الذي سيُرسل قيمته السيرفر لاحقاً (في غير الحالة الابتدائية) فإذا كانت القيمة 0 فيكون الطريق خالي من الحاويات ويكون الروبوت عند عقدة وإلا يكون الروبوت عند حاوية، وهذا الأمر يعتمد على أنه يوجد لون أسود عند الحاوية وعند العقدة، ويمكن للروبوت أن يفرق بينهم عن طريق الآلية الذي ذكرت.
- في حال كان الروبوت عند حاوية فإنه يقارن قيمتها `container_weight` بقيمة الحاوية الذاتية وإن أمكن تفريغها يرسل إشارة IR إلى الحاوية لتقوم بالتفريغ، ثم نعطي قيمة 0 لـ `container_weight` في الروبوت لكي يتعرف على العقدة لاحقاً
- في حال كان الروبوت عند عقدة سيقوم بإرسال اسم الحافة السابقة التي مر منها للسيرفر لتحريرها وجعلها غير مشغولة (جعل وزنها موجب بعد أن كان سالب عندما كانت مشغولة) لكي تدخلها الروبوتات الأخرى ويقوم بطلب الأوزان للطرق المتاحة له ومقارنتها واختيار الطريق ذو الوزن الأعلى والمسير فيه، كما أنه يخبر المخدم بسلوكه باختيار هذا الطريق ليجعل الوزن سالب (لكي لا تدخله الروبوتات الأخرى) ويعود للبحث عن اللون الأسود وهكذا
- ملاحظة : قبل طلب أوزان الطرق من السيرفر يختبر الروبوت قيمة الحاوية الذاتية فإذا كانت ممثلة تتوقف الخوارزمية الأساسية وتبدأ خوارزمية العودة إلى المكب في العقدة الأولى حيث $x=y=0$ عندها يتم تفريغ الحاوية الذاتية والبدأ بالخوارزمية الأساسية من جديد
- ونذكر التعديل الذي أدخلناه على مقارنة مستعمرة النمل فأولاً قمنا بإضافة وزن آخر للطريق غير وزن الحاويات وهو وزن يعبر عن مرور الروبوتات في هذا الطريق وكلما مر روبوت من طريق معين نقص وزن هذا الطريق؛ وهذا هو التعديل حيث أن في خوارزمية مقارنة مستعمرة النمل يزداد وزن الطريق كلما مرت النملة فيه، وثانياً وزن الطريق عبارة عن مجموع (وزن الطريق + وزن الحاوية) ويرسل السيرفر الوزن للروبوت عند طلبه على هذا الأساس.
- وهذا التعديل يفيدنا في بعثرة الروبوتات وتوزيعها على جميع الطرق وبهذا يتحقق ذكاء السرب

4.3. المخطط الصندوقي للحاوية الذكية:



المخطط (3-3) - المخطط الصندوقي للحاوية الذكية

يبين الشكل السابق المخطط الصندوقي للحاويات الذكية المستخدمة في مشروعنا حيث: استخدمنا شريحة Arduino MEGA واحدة لكل الحاويات كونها تحوي على عدد أكبر من المداخل والمخارج من بقية الإصدارات مما ساعدنا في زيادة عدد الحاويات المستخدمة. حيث تحوي 56 مدخل/مخرج رقمي و 16 مدخل تماثلي، مقارنة مع الـ UNO التي تحوي فقط على 14 مدخل/مخرج رقمي و 6 مداخل تماثلية. مع ملاحظة أنه حتى المداخل التماثلية يمكن استخدامها كمداخل/مخارج رقمية عند الحاجة. تتصل شريحة الأردوينو مع:

- سجل الإزاحة: كما أشرنا سابقاً من نوع 74595 وهو سجل إزاحة بطول 8 بت بحيث لكل حاويتين يوجد سجل إزاحة وحيد.
- شريحة الـ ESP-01 ويمكنها طلب API من اثنين وهما: UpdateEdges: حيث: باستخدام طلب GET تقوم بتمرير مصفوفة كافة أوزان الحاويات للمخدم ليقوم بتحديثها، reset: وهو يحتاج إلى وسيط وحيد وهو دليل الحافة ليقوم بتصفير وزن الطريق.
- مستقبلات IR بحيث لكل حاوية مستقبل يكون مسؤول عن تصفير وزن الحاوية.

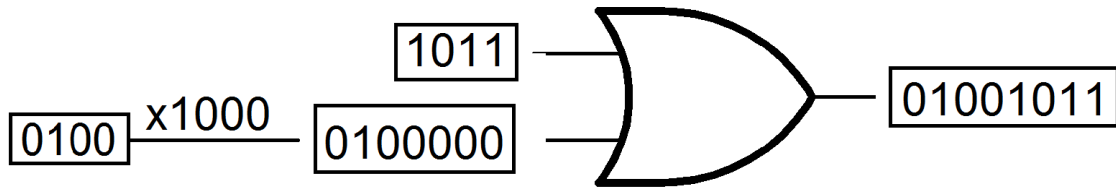
5.3. آلية عمل الحاويات الذكية:

في كل حاوية يوجد أربع ليدات بحيث تشير إلى نسبة امتلاءها، بحيث تقوم الأردوينو كل فترة زمنية معينة بتوليد رقم عشوائي بين الصفر والاثنين، وتضيفه إلى قيمة الحاوية القديمة شرط أن لا تتجاوز قيمة

الحاوية 4. لتقوم بتغيير قيمة الامتلاء لكل حاوية، ثم تقوم بطلب الـ API الخاص لترسل قيم الحاوية الجديدة إلى المخدم.

خلال أوقات الانتظار تقوم بفحص مستقبلات الـ IR لكل الحاويات وفي حال تلقت أمر التفريغ فإنها ستغير قيمة سجل الإزاحة وستطلب الـ API الآخر بغية تصفير قيمة وزن الحاوية للطريق.

بالنسبة لآلية التعامل مع سجل الإزاحة، لأن سجل الإزاحة بطول 8 بت وكل حاوية تحتاج فقط إلى 4 بت، فإننا نقوم بتوليد قيمة الحاوية الأولى ويكون من أربع خانات (ثنائية) فقط (xxxx) ومن ثم توليد قيمة الامتلاء للحاوية الثانية ونضربها بالقيمة 1000 لتصبح (xxxx0000) وأخيراً نجري العملية المنطقية OR قبل أن نمرر القيم إلى سجل الإزاحة كما يبين المخطط التالي:



المخطط (3-5) - مثال على طريقة عمل الأردوينو مع سجل الإزاحة

6.3. المراقبة الدائمة باستخدام تطبيق الأندرويد:

قد يحصل طارئ ما يؤدي إلى توقف روبوت ما، مما يؤدي إلى توقف العمل وقد يؤدي لحصول أزمة ممرورية في حال لم تأتي المساعدة. لهذا قمنا بتصميم تطبيق أندرويد باستخدام لغة JAVA وبمساعدة لغة XML (لغة رسوم) لبرمجة الواجهة

التطبيق بسيط نسبياً ما يقوم به هو طلب API خاص من المخدم للحصول على وزني الطريق والحاويات مجموعين وفي حال كان روبوت يشغل طريق ما سيظهر وزن هذا الطريق بلون أحمر. يظهر الشكل التالي واجهة التطبيق المستخدم:

7.3. ملاحظة حول الإرسال والاستقبال:

أي عملية إرسال واستقبال تجري ضمن المشروع سواء: - بين الأردوينو والـ ESP - بين الـ ESP والمخدم تكون على شكل مصفوفة حرفية String ويتم داخل المستقبل تحويل مصفوفة المحارف هذه إلى قيم

عددية صحيحة. لحسن الحظ هذا الأمر ممكن وسهل التطبيق بفضل المكاتب العديدة المتوفرة في لغتي البرمجة Python و C و Arduino.



الفصل الرابع

الرؤى والتطلعات المستقبلية

Chapter



4. التطلعات والرؤى المستقبلية:

بعد أن قمنا بتركيب الروبوت والمسارات، وقمنا باختبار المبدأ النظري وأثبتت بالفعل فعاليته في العمل سنناقش في هذا الفصل إمكانية نقل المشروع من مرحلة النموذج الأولي الذي صنعناه إلى مرحلة التطبيق الفعلي على مستوى البلدية وما هي التغيرات والتعديلات التي يجب أن تطرأ عليه لكي يلبي الحاجات الفعلية لمشروع بحجمه.

1.4. إنشاء بيان للبلدية:

إنّ أول خطوة مطلوبة هي من إدارة البلدية أن تقوم بتقديم خريطة تقسم فيها البلدية إلى عدة قطاعات بحيث توضّح في كل قطاع الطرق المتاحة للسير وأماكن حاويات القمامة بشكل واضح مع تبيان الإحداثيات الدقيقة لهذه الحاويات. ثم يجب تقريب هذه الخريطة إلى بيان باستخدام مبادئ نظرية البيان Graph Theory التي تمّ إيضاحها سابقاً بحيث يتم إيضاح أماكن العقد والحافات.

عندها يجب إعطاء أوزان (وهي الأوزان المأخوذة من مقارنة مستعمرة النمل) لهذه الطرقات (بغض النظر عن أوزان الحاويات) للمساهمة في نشر الروبوتات ما أمكن، كما يجب إيضاح أماكن وجود المشافي والمصانع إن وجدت بشكل خاص كون النفايات الطبية والنفايات الصناعية بحاجة لمعالجة خاصة، وبالتالي يجب تخصيص روبوتات خاصة لها، على سبيل المثال نفايات المشافي التي تعالج الأوبئة يجب تعقيمها قبل التخلص منها وبالتالي عملية دمجها مع بقية النفايات هي عملية خطيرة وقد تؤدي إلى نشر الأوبئة.

اليوم مع اتساع شبكة الانترنت واتساع الخدمات التي تقدّمها يمكن الحصول بسهولة على هذه البيانات، فمثلاً باستخدام موقع Google Map قمنا بالحصول على صورة فضائية لـ"حي الأمين" في دمشق وقمنا بالإشارة إلى أماكن تواجد مكبات القمامة باللون الأحمر.

بعد ذلك يمكن تقدير عدد الروبوتات اللازمة لكل بلدية، علماً أنّه من مميزات خوارزميات "نكاء السرب" أنها صالحة للعمل أيّاً كان عدد الروبوتات وبالتالي يمكن في أي لحظة إضافة روبوت إضافي للقطاع دون حدوث أي مشكلة أو تعديل جوهري.



الشكل (1-4) - حي الأمين النقاط الحمراء تشير لأماكن تجمعات القمامة

2.4. روبوت السرب:

في حين أننا قمنا في نموذجنا بالاعتماد على العقد لتحديد إحداثيات الروبوت (تغير الموقع x أو الموقع y) وهذا الأمر كان فعالاً، في حال قمنا بإطلاق الروبوت في الشوارع فإنّ مثل هذه الخوارزمية لن تكون فعّالة لهذا فإنّه يجب إضافة موديولات لتحديد المواقع بالاعتماد على نظام تحديد المواقع العالمي (GPS) Positioning System (GPS) ليستطيع كل روبوت تحديد موقعه بدقة. هذه الموديولات متوفرة للأردوينو ويبين الشكل التالي موديول يعرف بـ NEO6m لتحديد الموقع بسهولة وباستخدام توابع جاهزة.

ما منعنا من استخدام GPS في مشروعنا هو أنّ حساسية أي موديول متوافر لا تقل عن 1 متر وبالتالي لن نستطيع أن يتحسس أي تغير عند الانتقال لمسافات صغيرة ضمن الحلبة.



الشكل (4-2) - موديول GPS الخاص بالأردوينو

أيضاً تقنية الاتصال بالشبكة التي استخدمناها (راوتر مع شرائح ESP-01) لن تكون ذات فعالية عند الانتقال لمساحات كبيرة إلا في حال نشرنا مقويات إشارة في كل مكان، هذا الأمر قد يكون مكلف وذو سلبات عديدة. ولكن في حال تم إضافة موديول Global System for Mobile Communications (GSM) سيستطيع الروبوت بسهولة الولوج لشبكة الانترنت والاتصال مع المخدم لطلب API التي شرحناها سابقاً. ويبين الشكل التالي درع GSM الخاص بالأردوينو.



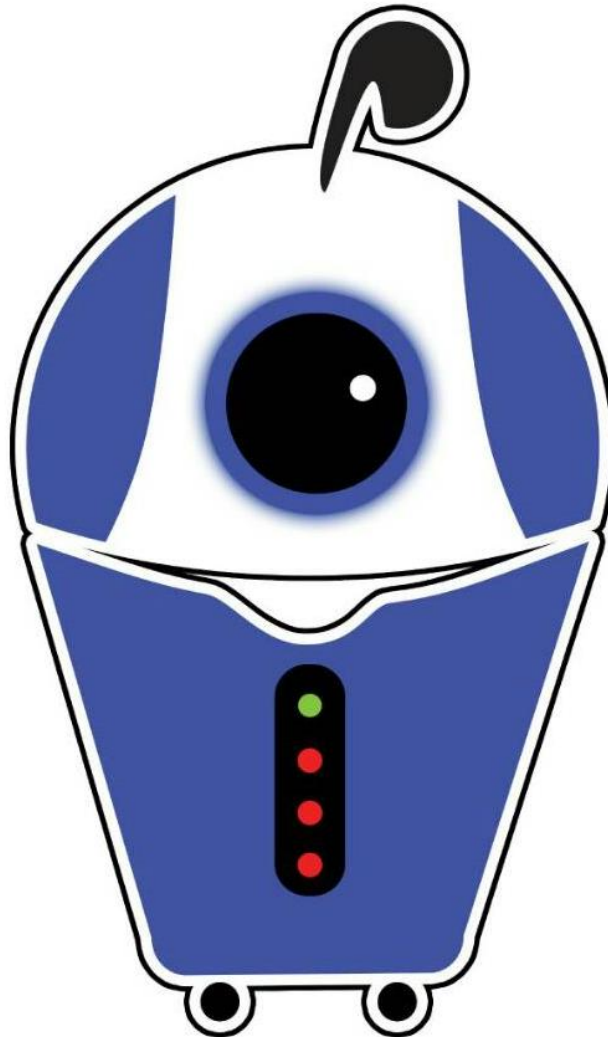
الشكل (4-3) - درع GSM

بالنسبة لمشكلة الطاقة التي بحاجة لها الروبوت يمكننا استخدام الطاقة الشمسية النظيفة بإضافة ألواح طاقة شمسية إلى الروبوت.

بالطبع يجب حل مشكلة سير الروبوت ضمن سكك خاصة وإطلاق الروبوت للمسير ضمن الطرقات بشكل طبيعي، ويمكن إنجاز هذا باستخدام الرؤية الحاسوبية لضمان سلامة المارة والممتلكات والروبوت وضمان وصوله إلى غايته، كما يجب إنجاز ميكانيك لتفريغ الحاويات ضمن الروبوت.

3.4. الحاويات الذكية:

كل حاوية يتعامل معها الروبوت يجب أن تكون مزودة بدارة تجعلها قادرة على تحسس مستوى الامتلاء وإعلام المخدّم بهذا المستوى لتعديل الأوزان. أي أنه لا يكفي دارة واحدة لجميع الحاويات كما أنجزنا في نموذجنا وإنما تخصيص دارة لكل حاوية والأخذ بعين الاعتبار حمايتها من العوامل الجوية ومن السرقة. هكذا نكون قد نقلنا بالفعل نموذج أثبت عمله إلى التطبيق العملي وقمنا بحل المشاكل التي ذكرناها في المقدمة وساهمنا في الحفاظ على شوارع بلدنا نظيفة.



المراجع:

- [1]. https://www.hrw.org/ar/report/2017/12/01/311575?_ga=2.44303076.1921310679.1511779285-132185475.1507982581
- [2]. Beni, G., Wang, J. Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30 (1989)
- [3]. Biggs, Norman, E. Keith Lloyd, and Robin J. Wilson. *Graph Theory, 1736-1936*. Oxford University Press, 1976.
- [4]. Coloni, A., Dorigo et M, Maniezzo V (1991) Distributed optimization by Ant Colonies, actes de la première conférence européenne sur la vie artificielle.



الملحقات:

- برمجة الـ API المستخدمة:

:AndroidAPI .1

```
#!/C:\Python27\python.exe"
print "Content-Type: text/html\n\n";
import cgi
import pickle
#load the Eadges
with open(r'C:\xampp\cgi-bin\Eadges.txt', "rb") as fp: # Unpickling
    Eadges = pickle.load(fp)
#load the EadgesRoads
with open(r'C:\xampp\cgi-bin\EadgesRoads.txt', "rb") as fp: # Unpickling
    EadgesRoads = pickle.load(fp)
tmp = 0;
toESP = [];
for i in range(0,40):
    if EadgesRoads[i]>=0:
        tmp = Eadges[i] + EadgesRoads[i]
        toESP.append(tmp)
    else:
        tmp = -Eadges[i] + EadgesRoads[i]
        toESP.append(tmp)
toESP = str(toESP);
toESP = toESP.replace("[", "")
toESP = toESP.replace("]", "")
toESP = toESP.replace(" ", "")
#Send toESP to ESP
print toESP;
```



:SorR .2

```
#!"C:\Python27\python.exe"
print "Content-Type: text/html\n\n";
import cgi
import pickle
form = cgi.FieldStorage()
x = form["index"].value
#Convert String to int
Eadge = []
tmp = ""
for i in range(0,len(x)):
    if ((x[i] != ',') & (i != len(x)-1)):
        tmp+=x[i]
    elif i == len(x)-1:
        tmp+=x[i]
        tmp = int(tmp)
        Eadge.append(tmp)
        tmp = ""
    else:
        tmp = int(tmp)
        Eadge.append(tmp)
        tmp = ""
Eadgeindex = Eadge[0] -1
status = Eadge[1]
#load the EadgesRoads
with open(r'C:\xampp\cgi-bin\EadgesRoads.txt', "rb") as fp: # Unpickling
    EadgesRoads = pickle.load(fp)
if status == 0: #0 is for reseting 1 for starting
    EadgesRoads[Eadgeindex] = 0;
if status == 1:
    EadgesRoads[Eadgeindex] *= -1;
#save The EadgesRoads List
with open(r'C:\xampp\cgi-bin\EadgesRoads.txt', "wb") as fp: #Pickling
    pickle.dump(EadgesRoads, fp)
print "OK"
```

3. :GetEadges

```
#!/bin/Python27python.27"
print "Content-Type: text/html\n\n";
import cgi
import pickle
form = cgi.FieldStorage()
x = form["index"].value
Eadge = []
tmp = ""
for i in range(0,len(x)):
    if ((x[i] != ',') & (i != len(x)-1)):
        tmp+=x[i]
    elif i == len(x)-1:
        tmp+=x[i]
        tmp = int(tmp)-1
        Eadge.append(tmp)
        tmp = ""
    else:
        tmp = int(tmp)-1
        Eadge.append(tmp)
        tmp = ""
#load the Eadges
with open(r'C:\xampp\cgi-bin\Eadges.txt', "rb") as fp: # Unpickling
    Eadges = pickle.load(fp)
#load the EadgesRoads
with open(r'C:\xampp\cgi-bin\EadgesRoads.txt', "rb") as fp: # Unpickling
    EadgesRoads = pickle.load(fp)
toESP = [];
for i in Eadge:
    if EadgesRoads[i]>=0:
        tmp = Eadges[i] + EadgesRoads[i]
        toESP.append(tmp)
    else:
        tmp = -Eadges[i] + EadgesRoads[i]
        toESP.append(tmp)
toESP = str(toESP);
toESP = toESP.replace("[", "")
toESP = toESP.replace("]", "")
toESP = toESP.replace(" ", "")
#Send toESP to ESP
print toESP;
```

:GetEdges.4

```
#!/bin/Python27\python27 "
print "Content-Type: text/html\n\n";
import cgi
import pickle
form = cgi.FieldStorage()
x = form["index"].value
#Convert String to int
x = int(x);
Eadindex = x -1
#load the Eadges
with open(r'C:\xampp\cgi-bin\Eadges.txt', "rb") as fp: # Unpickling
    Eadges = pickle.load(fp)
    Eadges[Eadindex] = 0;
#save The Eadges List
with open(r'C:\xampp\cgi-bin\Eadges.txt', "wb") as fp: #Pickling
    pickle.dump(Eadges, fp)
print "OK"
```

• برمجة شرائح الـ ESP:

1. شريحة الـ ESP المتصلة بروبوت السرب:

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
const char* ssid = "R.U.T";
const char* password = "Rabta123";
char R;
String URL;
String INDEX;
String path;
void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
    }
    //Connected
}
void loop() {
    // put your main code here, to run repeatedly:
    HTTPClient http;
    while(!Serial.available());
    R = Serial.read();
    if(R == 'G')
        URL = "http://192.168.1.101/cgi-bin/GetEadges.py?index=";
    else if (R == 'S')
        URL = "http://192.168.1.101/cgi-bin/RorS.py?index=";
    Serial.print("OK");
    while(!Serial.available());
    INDEX = Serial.readString();
    INDEX.trim();
    path = URL + INDEX;
    http.begin(path);
    int httpCode = http.GET();
    String payload = http.getString();
    if (httpCode > 0) { //Check the returning code
        String payload = http.getString(); //Get the request response payload
        payload.trim();
        Serial.print(payload);           //Print the response payload
    }
    http.end();
}
```


2. شريحة الـ ESP المتصلة بروبوت الحاويات:

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
const char* ssid = "R.U.T";
const char* password = "Rabta123";
char R;
String URL;
String INDEX;
String path;
void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
    }
    //Connected
}
void loop() {
    // put your main code here, to run repeatedly:
    while(!Serial.available());
    R = Serial.read();
    if (R == 'A') //Update All Eadges
        URL = "http://192.168.1.100/cgi-bin/UpdateAllEadges.py?index=";
    else
        URL = "http://192.168.1.100/cgi-bin/reset.py?index=";
    HTTPClient http;
    while(!Serial.available());
    INDEX = Serial.readString();
    INDEX.trim();
    path = URL + INDEX;
    http.begin(path);
    int httpCode = http.GET();
    String payload = http.getString();
    if (httpCode > 0) { //Check the returning code
        String payload = http.getString(); //Get the request response payload
        payload.trim();
        Serial.print(payload);           //Print the response payload
    }
    http.end();
}
```