Syrian Arab Republic

Ministry of Higher Education

Syrian Virtual University

الجمهورية العربية السورية

وزارة التعليـم العالـــــي

الجامعة الافتراضية السورية

إدارة ومراقبة عمل مصنع تعبئة الغاز المنزلي بواسطة تقنية انترنت الأشياء (IoT)

Managing and Monitoring the Work of Domestic Gas Filling Factory by Using IoT Technology

بحث أعد لنيل درجة ماجستير تأهيل وتخصيص في تقانات الويب

إعداد الطالب

عمار حسن محمود

بإشراف الدكتور المهندس

سامر سليمان

2022-2021

الإهداء

بداية أوجه هذا الجهد وأعتبره خالصاً لوجه الله تعالى،فأسألك اللهم القبول...

إلى من جرع الكأس فارغاً ليسقيني قطرة حب

إلى من كلّت أنامله ليقدم لنا لحظة سعادة

إلى من حصد الأشواك عن دربي ليمهد لي طريق العلم

*إلى القلب الكبير .....**(والدي العزيز)***

إلى من أرضعتني الحب والحنان

إلى رمز الحب وبلسم الشفاء

*إلى القلب الناصع بالبياض .....**(والدتي الحبيبة)***

إلى القلوب الطاهرة الرقيقة والنفوس البريئة إلى رياحين حياتي

(إخوتي)

إلى كل شخص ترك أثراً جميلا في حياتي وذكريات رائعة مليئة بالحب

(أقاربي وأصدقائي)

الباحث:عمار محمود

شكر وتقدير:

أتقدم بجزيل الشكر والتقدير إلى كل من وقف معي وساندني خلال فترة دراستي في الجامعة الافتراضية السورية،وأخص بالذكر أستاذي العزيز ،المشرف على الرسالة الدكتور المهندس **سامر سليمان** لكرم عطائه ومساعدته المستمرة من خلال توجيهاته وملاحظاته التي ساهمت بشكل كبير في إنجاز هذا العمل.

كما أتقدم بالشكر لجميع أعضاء الكادر التدريسي في الجامعة الافتراضية السورية لما قدموه لي خلال فترة دراستي من تسهيلات ودعم.

وأتقدم بوافر الشكر والتقدير وعظيم الامتنان للجنة المناقشة الأفاضل الذين أثروا هذا البحث من خلال ملاحظاتهم وإرشاداتهم القيمة.

فهرس المحتويات

فهرس المحتويات I
فهرس الأشكال III
فهرس الجداول V
جدول المصطلحات والاختصارات: VI
الملخص VIII
1 الفصل الأول مقدمة 1
1-1 منهجية البحث 3
2-1 هدف البحث: 3
3-1 المشكلة العلمية للبحث: 4
4-1 أهمية البحث: 4
5-1 هيكلية الأطروحة: 4
2 الفصل الثاني البنية المعمارية لانترنت الأشياء 6
2-1-1 العلاقة بين طبقات انترنت الأشياء : 7
2-1-2 مكونات انترنت الأشياء: 7
2-2 خصائص انترنت الأشياء: 9
2-3 بروتوكولات الاتصالات في انترنت الأشياء: 10
2-4 منصات انترنت الأشياء: 12
3 الفصل الثالث تحليل وتصميم نظام إدارة ومراقبة معمل تعبئة الغاز المنزلي 17
3-1 تجهيزات معمل تعبئة الغاز المنزلي : 17
3-2 متطلبات تصميم نظام التحكم و المراقبة: 20
3-3 تصميم نظام التحكم والمراقبة: 21
3-3-1 المحاكي (التحكم المحلي): 21
3-3-1-1 مبدأ العمل : 22
3-3-1-2 نظام التحكم عن بعد: 26
3-3-2 واجهة التحكم والإدارة عن بعد : 27
3-3-2-1 مبدأ العمل : 28
3-3-3 منصة انترنت الأشياءIBM Bluemix & Watson IoT Platform: 29
3-3-4 خطوات استخدام منصتي IBM Bluemix و IBM Watson IoT : 32
3-3-5 إضافة أجهزة إلى منصة IBM Watson IoT : 32

3-3-6 إضافة تطبيقات إلى منصة IBM Watson IoT : 33
4 الفصل الرابع الاختبارات و النتائج 35
4-1 التحليل الرياضي لزمن استجابة واجهة التحكم عن بعد : 35
4-2 سيناريو الاختبار : 36
4-3 مناقشة النتائج : 37
5 الفصل الخامس الخاتمة والأعمال المستقبلية 39
5-1 الخاتمة: 39
5-2 الاستنتاجات : 39
5-3 الأعمال المستقبلية : 40
المراجع 41
الملاحق : 43

فهرس الأشكال

الشكل(1–1) آلة إلى آلة.....1
الشكل(2–1) البنية المعمارية لانترنت الأشياء.....6
الشكل(2–2) المكونات المشكلّة لتقنية انترنت الأشياء.....8
الشكل(2–3) مبدأ عمل بروتوكول AMQP.....11
الشكل(2–4) مبدأ عمل بروتوكول MQTT.....11
الشكل(2–5) منصة Google Cloud 13
الشكل(2–6) منصة IBM BlueMix.....14
الشكل(2–7) منصة ThingWorx 15
الشكل(2–8) منصة Microsoft Azure.....15
الشكل(3–1) الأجزاء الرئيسة لنظام إدارة و مراقبة معمل تعبئة الغاز المنزلي 17
الشكل(3–2) وحدة الغسل 18
الشكل(3–3) السير الناقل للأسطوانات.....18
الشكل(3–4) وحدة التعبئة الآلية الدائرية 19
الشكل(3–5) وحدة كشف التسرب 19
الشكل(3–6) نظام الإطفاء بالماء 20
الشكل(3–7) نموذج معمل تعبئة الغاز المقترح كمحاكي على صفحة ويب، والأجزاء الرئيسة له...22
الشكل(3–8) واجهة التحكم المحلية المضمنة في نموذج المعمل.....23
الشكل(3–9) تشغيل وإيقاف نظام تعبئة الغاز المنزلي 24
الشكل(3–10) مخطط الحالة لـ تابع Update State.....25
الشكل(3–11) المخطط التدفقي لتشغيل و إيقاف نظام إطفاء الحريق 25
الشكل(3–12) المخطط التدفقي لاستلام أمر من واجهة التحكم والإدارة عن بعد 26
الشكل(3–13) مخطط الحالة لعمل تابع Update State.....27
الشكل(3–14) واجهة التحكم والإدارة عن بعد المعتمدة على تقنيات الويب 27
الشكل(3–15) المخطط التدفقي لأمر تشغيل أو إيقاف نظام تعبئة الغاز المنزلي من واجهة التحكم 28
الشكل(3–16) المخطط التدفقي لأمر التشغيل أو الإيقاف لنظام الإطفاء من واجهة التحكم عن بعد 29
الشكل(3–17) منصة IBM Watson IoT.....31
الشكل(3–18) الأجهزة المعرفة على منصة IBM Watson IoT 33

الشكل(3–19) التطبيقات المعرفة على منصة IBM Watson IoT 34
الشكل(4–1) المراحل الزمنية لاستجابة واجهة التحكم عن بعد لبيانات محاكي نظام التعبئة 35
الشكل(4–2) اختبار الـ Ookla لمعرفة السرعة الحالية لشبكة الانترنت المستخدمة من قبل محاكي نظام التعبئة وواجهة التحكم عن بعد.....36
الشكل(4–3) المخطط البياني للاستجابة الوسطية لواجهة التحكم عن بعد لبيانات محاكي نظام التعبئة عند سرعات انترنت مختلفة 38

فهرس الجداول

الجدول(2–1) مقارنة بين البروتوكولات المستخدمة في انترنت الأشياء.....12
الجدول(2–2) مقارنة بين بعض منصات انترنت الأشياء.....16
الجدول(4–1) سيناريوهات العمل بناءً على سرعات الانترنت المحددة بواسطة برنامج SelfishNet 37
الجدول(4–2) الاستجابات الوسطية المقاسة لكل سيناريو عمل.....38

جدول المصطلحات والاختصارات:

الاختصارات	المصطلحات	
IoT	Internet of Things	انترنت الأشياء
IBA	Internet Board Architecture	مجلس هندسة الانترنت
ITU	The UN's International Telecommunications Union	اتحاد الاتصالات الدولي
CAD	computer-aided design	تصميم بمساعدة الحاسب
RFID	Radio-frequency identification	تحديد الهوية بموجات الراديو
HTTP/HTTPS	Hypertext Transfer Protocol / Hypertext Transfer Protocol Secure	بروتوكول نقل النص التشعبي / بروتوكول نقل النص التشعبي الآمن
TCP	Transmission Control Protocol	بروتوكول التحكم بالنقل
UDP	User Datagram Protocol	بروتوكول مخطط بيانات المستخدم
SSL	Secure Socket Layer	بروتوكول طبقة التوصيل الآمن
CoAP	Constrained Application Protocol	بروتوكول التطبيق المقيد
AMQP	Advanced Message Queuing Protocol	بروتوكول وضع الرسائل في قائمة الانتظار المتقدم
MQTT	Message Queuing Telemetry Transport	بروتوكول نقل القياس عن بعد لخدمة الرسائل في قائمة الانتظار
IETF	Internet Engineering Task Force	فريق مهام هندسة الإنترنت
OASIS	Organization for the Advancement of Structured Information Standards	منظمة النهوض بمعايير المعلومات المنظمة
ISO	International Organization for Standardization	المنظمة الدولية للمقاييس

TLS	Transport Layer Security	بروتوكول أمن طبقة النقل
PaaS	Platform as a Service	المنصة كخدمة
M2M	Machine To Machine	آلة إلى آلة
XHTML	Extensible Hyper Text Markup Language	لغة ترميز النص التشعبي القابلة للتوسيع
XML	Extensible Markup Language	لغة التوصيف الموسعة
CSS	Cascading Style Sheet	ورقة الأنماط المتتالية
JSF	Java Server Faces	وجوه مخدم جافا
VPN	Virtual Private Network	الشبكة الافتراضية الخاصة
CDN	Content Delivery Network	شبكة إيصال المحتوى
IaC	Infrastructure as Code	البنية التحتية كرمز
REST	Representational State Transfer	بروتوكول نقل الحالة التمثيلي

الملخص

تعدّ تقنية انترنت الأشياء ثورة في مجال تكنولوجيا المعلومات والشبكات، حيث تسهم هذه التقنية في بناء بنية تحتية متكاملة، تربط مختلف الأجهزة مع بعضها البعض عبر شبكة الانترنت. إنّ لهذه التقنية القدرة على تغيير الحياة البشرية، وجعلها أسهل من خلال اختصار الجهد، والزمن، والمسافات، ويظهر ذلك جلياً في العديد من تطبيقاتها، مثل: المنزل الذكي، والعناية بالصحة، ونظام مراقبة الازدحام، وغيرها من التطبيقات التي بدأت تنتشر بشكل واسع، بعد أن أثبتت فعاليتها ونجاحها. وكتطبيق آخر لتقنية انترنت الأشياء، قمنا في هذا البحث بطرح المشكلات التي تعاني منها منشآت تعبئة الغاز المنزلي من حيث انخفاض مستوى الأداء والإنتاجية، وتضمين هذه التقنية في عمل هذه المنشآت لعلاج هذه المشكلات والارتقاء بهذه المنشآت إلى مستوى أعلى من التحكم الآلي والسلامة المهنية ،من خلال تصميم محاكي، يحاكي عملية تعبئة الغاز في معمل تعبئة الغاز المنزلي الآلي باستخدام تقنيات الويب المختلفة، والتحكم بأهم التجهيزات المستخدمة في عملية التعبئة، ومراقبتها عن بعد باستخدام تقنية انترنت الأشياء. كما قمنا بقياس زمن استجابة واجهة التحكم عن بعد، لبيانات عملية التعبئة ضمن ظروف تشغيل معينة. استخدمت منصة Watson IoT platform في عملية تصميم النظام، التي قدمت البروتوكولات الضرورية، والمناسبة لتقنية انترنت الأشياء، وسمحت لنا بإضافة الأجهزة المراد التحكم فيها، ومراقبتها عن بعد.

الكلمات المفتاحية: انترنت الأشياء، آلة إلى آلة، المحاكاة، المراقبة والتحكم عن بعد.

Abstract

Internet of Things is a revolution in information technology and networks, as this technology contributes to building an integrated infrastructure that connects various devices with each other over the Internet. This technology has the ability to change human life and make it easier by shortening effort, time and distance, and this is obvious in many of its applications, such as: smart home, health care, congestion control system and other applications that began to spread widely after they proved their effectiveness and success. As another application of the Internet of Things, in this research, we presented the problems that domestic gas filling facilities suffer from in terms of low level of performance and productivity and include this technology in the work of these facilities and raise these facilities to a higher level of automatic control and occupational safety ,through a simulated design that simulates the process of filling gas in an automated home gas filling plant using various web technologies and controlling the most important equipment used in the process Filling and monitoring it remotely using IoT technology. We also measured the response time of the remote-control interface to the filling process data under certain operating conditions. The Watson IoT platform was used in the system design process that provided the necessary and appropriate protocols for IoT technology and allowed us to add devices to be controlled and monitored remotely.

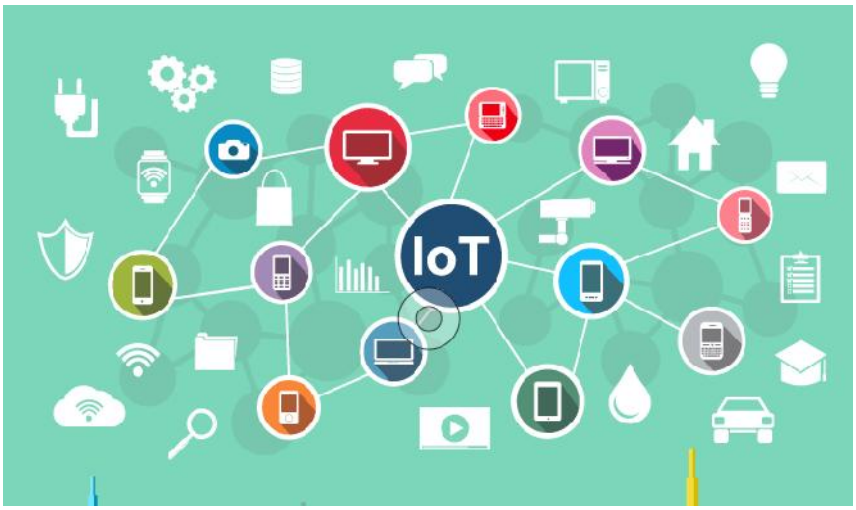
Key words: Internet of Things, Machine-to-Machine, Simulation, Remote Monitoring and Controlling.

الفصل الأول

مقدمة

إن التطور في علوم الشبكات، والالكترونيات، وانتشار الشبكات اللاسلكية، وخدمة الانترنت، أدى إلى ظهور مفهوم جديد في علم الشبكات والمسمى بإنترنت الأشياء؛ حيث يمكن اختصار هذا المفهوم بأنه شبكة من الأجهزة المتصلة مع بعضها البعض عبر شبكة الانترنت، وهذا يعني إمكانية تواجد هذه الأجهزة في أي مكان حول العالم،وعلى بعد ملايين الكيلومترات عن بعضها البعض، التي يمكن التحكم فيها، ومراقبتها عن بعد.

يعدّ انترنت الأشياء موجة اقتصادية في صناعة المعلومات بعد الانترنت، وهو شبكة ذكية تربط مختلف الأجهزة بشبكة الانترنت، بغرض تبادل المعلومات، والاتصال بواسطة بروتوكولات متفق عليها، وبالنتيجة إمكانية إدارة ومراقبة هذه الأجهزة [1].ووفقا لمجلس هندسة الانترنت (IBA)Architecture Internet Board،يعدّ انترنت الأشياء تكنولوجيا، تقوم من خلالها عدد من الأجهزة المدمجة الغير متحكم فيها بشكل مباشر من قبل الإنسان، بتوظيف خدمات الاتصالات، وبروتوكولات الانترنت، وفي تعريف آخر يعدّ IEEE انترنت الأشياء بأنه نظام يكون فيه كل شيء ممثل، ومعطى في الانترنت، ويجعل التفاعل بين الأشياء، والتطبيق ممكنا في سحابة [2] .



الشكل(1-1) آلة إلى آلة

على الرغم من أن انترنت الأشياء ولد بين عامي 2008 – 2009 إلا أن هذا المصطلح ليس جديدا حيث يقول Kevin Ashton– هو مبتكر وخبير حساسات وقد استخدم لأول مرة عبارة انترنت الأشياء في عرض تقديمي له عام 1999– :

" انترنت الأشياء لديه القدرة على تغيير العالم كما فعل الانترنت و ربما أكثر " [2] .

إن أول تطبيق لإنترنت الأشياء، كان آلة مشروبات في جامعة Carnegie Melon في أوائل الثمانينات؛ حيث استطاع المبرمجون أن يربطوا الآلة بشبكة الانترنت، ويختبروا حالة الآلة، وإمكانية طلب مشروب بارد في حال توفره. في عام 2000، أعلنت شركة LG عن أول براد متصل بالإنترنت، وفي عام 2005 نشر اتحاد الاتصالات الدولي(ITU) أول تقرير له عن انترنت الأشياء[3] .

منذ ولادة الـ IoT بدأ التفكير بإمكانيات هذه التقنية، والمجالات التي يمكن أن تكون فيها فعالة، ومنتجة، نذكر بعض هذه المجالات:

- نظام معلومات الازدحام: يستطيع هذا النظام تحصيل معلومات الازدحام، كحالة الطرق إذا كانت مزدحمة أو لا، والمناطق المزدحمة عن طريق تتبع معلومات المواقع لعدد ضخم من المركبات، ومن ثم يقوم النظام بمساعدة السائق على تحديد أفضل طريق غير مزدحم [1].
- نظام تحسس بيئي: يستطيع هذا النظام تجميع،ومعالجة جميع أنواع البارامترات الفيزيائية والكيميائية البيئية عن طريق أجهزة تحسس، منتشرة بشكل واسع أو محلي، وعادة ما تكون هذه البارامترات هي معلومات عن الحرارة، والرطوبة، والضجيج، وشدة الضوء، والإشعاع،والتلوث [1].
- نظام إدارة النفايات: هو أحد الأنظمة المستعملة في المدن الذكية، حيث تعد مشكلة تزايد النفايات مع تزايد الكثافة السكانية، من الأمور الهامة التي ينبغي معالجتها في المدن الذكية. ويقوم هذا النظام بمهمتين أساسيتين ،هما: تحديد الأماكن التي يجب جمع النفايات منها، بناء على عوامل اجتماعية،واقتصادية، وبيئية، والمهمة الثانية، تحديد الطريق الأقصر لسيارات جمع النفايات بما يوفر الوقت، واستهلاك الوقود [4].
- نظام المنزل الذكي: يشير مصطلح المنزل الذكي إلى بيت مجهز بأجهزة عالية التقنية، وحساسات متصلة مع بعضها بشبكة اتصالات، بما يسمح للمستخدم بالوصول، والمراقبة والتحكم بالتجهيزات المنزلية ، وفقا إلى حاجة المستخدم. ولكن على الرغم من التاريخ الطويل، والنمو المذهل لهذه التقنية، إلا أنها لم تلق رواجا كبيرا لأسباب عدة منها: غلاء تجهيزاتها، وانخفاض الطلب عليها [5].
- نظام مراقبة الصحة: يستطيع هذا النظام فحص حالة المستخدم الصحية، والكشف عن أي ضرر جسدي موجود، والتنبؤ بحدوث مشاكل صحية،و ذلك بالاعتماد على حساسات عديدة كحساسات الضغط، والحرارة، والتسارع، بالإضافة إلى تقنية معالجة الصورة.وبالإمكان مراقبة هذه المعطيات وتخزينها على أي هاتف خلوي متصل بشبكة الانترنت [6].

سنقوم في هذا البحث بتصميم نظام مراقبة عملية تعبئة الغاز المنزلي كتطبيق آخر من تطبيقات انترنت الأشياء،الذي سيسمح بالتحكم،ومراقبة تجهيزات معمل التعبئة من أي مكان في العالم،وبزمن قصير مقارنة مع التسلسل الإداري لنقل المعلومة، وبالنتيجة تحقيق التفاعل بين المنشآت العاملة بهذه التقنية.

1-1 منهجية البحث

اعتمدنا في هذا البحث على الطريقة التحليلية في عملية تصميم النظام، والطريقة التجريبية عن طريق نمذجة هذا النظام، وجعله يحاكي ظروف التشغيل قدر الإمكان، ومن ثم حل المشكلات التي تواجه موضع بحثنا من خلال هذا النموذج .

استخدمنا في عملية النمذجة وتصميم النظام،عدداً من الأدوات نذكر أهمها:

- برنامج NetBeansIDE8.1 : هو بيئة تطوير مدمجة مجانية، ومفتوحة المصدر بشكل كامل، تساعد على إبداع مختلف التطبيقات البرمجية وتؤمن الدعم لجميع أنواع تطبيقات الجافا كما تساعد على تطوير المشاريع التي تستخدم تقنيات الجافا.[7]
- برنامج SolidWorks : هو برنامج تصميم حديث بمساعدة الحاسب (CAD) يسمح للمصممين بإنشاء أنموذج صلب صحيح رياضيا لجسم معين، والذي يمكن تخزينه في قاعدة بيانات .[8]
- برنامج Photoshop : هو برنامج تحرير رسومات نقطية مطور من قبل شركة Adobe.يستخدم الفوتوشوب بشكل رئيسي في تحرير الصور، ولكن يمكن استخدامه في إنشاء أنواع مختلفة من ملفات الرسومات .[9]
- برنامج Movavi Video Editor 15 Plus : هو أداة فعالة تسمح بتحرير ملفات الفيديو، وإنشاء فيديو بتأثيرات متنوعة من الصور بكل سهولة .[10]
- متصفح انترنت : هو تطبيق برمجي لاسترداد، وعرض، ونقل مصادر المعلومات على شبكة الانترنت.يعمل متصفح الانترنت كواجهة بين المستخدم، و مخدمات الويب .[11]
- برنامج SelfishNet : هو برنامج يتيح للمستخدم التحكم الكامل بشبكة الانترنت الخاص به، من خلال معرفة جميع المستخدمين المتصلين بهذه الشبكة، والتحكم بعرض الحزمة لشبكاتهم، أو حتى حجب المستخدمين من الوصول إلى هذه الشبكة .[32]

1-2 هدف البحث:

يتوقع من النظام المقترح في نهاية البحث أن يكون قادراً على التحكم بتجهيزات المعمل المقترحة، ضمن البحث كخط السير المسؤول عن نقل أسطوانات الغاز من وإلى وحدات التعبئة داخل المعمل،وأنظمة إطفاء الحريق،التي تعمل تلقائياً في حالة وجود حريق، أو يدوياً بهدف الاختبار.بالإضافة إلى ذلك سيهتم النظام بتبادل المعلومات المتعلقة بعملية التعبئة،وحالة المعمل إلى الشخص المسؤول بالزمن الحقيقي.

1-3 المشكلة العلمية للبحث:

إن معظم المشكلات التي تعاني منها المنشآت الصناعية؛ هو انخفاض مستوى الأداء، والإنتاجية وذلك للأسباب الآتية:

- صعوبة إيصال حالة المنشأة إلى الخبير، وصاحب القرار في الوقت المناسب، الذي ينتج عن انخفاض عدد الخبراء، أو عدم قدرة الخبير، والمراقب على التواجد في المنشآت كلها في اللحظة ذاتها، بالإضافة إلى عدم جدولة إيصال الحالة بالطريقة المثلى، اعتماداً على أهميتها وأولويتها.
- صعوبة إدارة وتفاعل المنشآت المنتشرة على بقع جغرافية مختلفة مع بعضها البعض، المعتمدة على مبدأ الإدارة المستقلة والتخطيط المركزي،الذي سيؤدي إلى تخطيط غير مثالي لهذه المنشآت؛ بسبب عدم المعرفة الواضحة، والآنية للمشاكل التي تواجهها كل منشأة على حدا.

1-4 أهمية البحث:

في وقتنا الحالي، وبسبب التطور التقني المتسارع، أصبحت المؤسسات في مختلف مجالات الصناعة، تستخدم إنترنت الأشياء لتعمل بكفاءة أكبر، ولإفهام العملاء بشكل أفضل،كي تقدم خدمة مطورة لعملائها، وتحسن من اتخاذ القرار، وتزيد من قيمة العمل.

يوفر إنترنت الأشياء للشركات نظرة آنية في الزمن الحقيقي حول كيفية عمل أنظمتها حقًا، حيث تقدم معلومات دقيقة، وآنية حول كل شيء يعمل ضمن الشركة،بدءاً من أداء الآلات، إلى سلسلة التوريد، وعمليات النقل والإمداد. يمكّن إنترنت الأشياء الشركات من جعل العمليات تلقائية، وتقليل تكاليف العمالة. كما أنه يحسن من تقديم الخدمات، مما يجعل تصنيع البضائع، وتسليمها أقل تكلفة. بالإضافة إلى توفير الشفافية في معاملات العملاء.

يعدّ استخدام تقنيات انترنت الأشياء للتحكم، ومراقبة معامل تعبئة الغاز المنزلي من الأمور الضرورية في وقتنا الحالي، لما يقدمه من مراقبة آنية، ومستمرة على مدار الساعة. بالإضافة إلى الاستجابة السريعة للحالات الحرجة عن طريق التحكم بآلاته وأنظمته عن بعد، مما يوفر الزمن اللازم للوصول إلى منطقة العطل، أو الحادث، وأيضاً حماية العمال من التعرض للأذى جراء تواجدهم في المنطقة المنكوبة.

1-5 هيكلية الأطروحة:

تنقسم هذه الأطروحة إلى خمسة فصول ، يتناول الفصل الأول مقدمة عن تقنية انترنت الأشياء وبعض تطبيقاتها بالإضافة إلى الأدوات المستخدمة في عملية النمذجة وتصميم النظام المقترح وأخيراً المشكلة العلمية للبحث وأهمية هذا البحث من الناحية الاقتصادية والسلامة المهنية. يسلط الفصل الثاني

الضوء على تقنية انترنت الأشياء من حيث البنية المعمارية والطبقات المكونة لها وأهم البروتوكولات المستخدمة في هذه التقنية، كما يذكر بعض المنصات الأكثر استخداما في تطوير، وتنفيذ تطبيقات انترنت الأشياء. يختص الفصل الثالث بتحليل نظام إدارة ومراقبة معمل تعبئة الغاز المنزلي وذكر أهم التجهيزات المستخدمة في معامل تعبئة الغاز المنزلي الآلية ومتطلبات تصميم نظام يحاكي عمل هذه التجهيزات بما يحقق عملية تعبئة اسطوانات الغاز المنزلي بالإضافة إلى شرح مبدأ عمل النظام المقترح بشكل مفصل والتعريف بمنصتي IBM Bluemix و IBM Watson IoT كأحد المكونات الأساسية في عملية التصميم. يعالج الفصل الرابع تأثير سرعة الانترنت على زمن استجابة واجهة التحكم عن بعد لبيانات محاكي نظام تعبئة الغاز المنزلي باعتبار عامل الزمن من العوامل الهامة في تطبيقات انترنت الأشياء من خلال خمسة سيناريوهات عمل، إذ يمثل كل سيناريو عمل سرعة محددة للانترنت والأزمنة الوسطية لعمليات قياس الزمن الذي يستغرقه إرسال بيانات حالة نظام التعبئة إلى واجهة التحكم عن بعد وبالنتيجة الحصول على زمن وسطي كلي لاستجابة واجهة التحكم عن بعد لبيانات محاكي نظام التعبئة لكل سيناريو عمل على حدا. يلخص الفصل الخامس الأعمال المنجزة والنتائج والاستنتاجات التي توصلنا لها في هذا البحث وأهم الأعمال المستقبلية التي نطمح لانجازها في الدراسات اللاحقة.

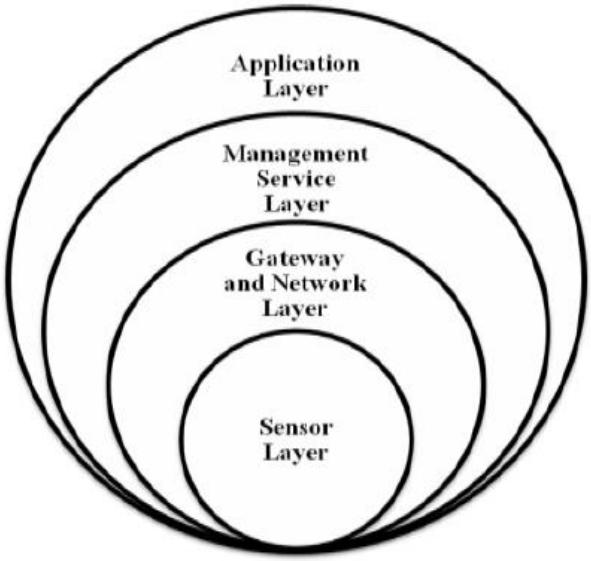
الفصل الثاني

البنية المعمارية لانترنت الأشياء

لا يملك انترنت الأشياء بنية معمارية عامة متفق عليها. يعود ذلك إلى اتساع هذا المفهوم ،ما يجعله أحد المشاكل الأساسية لانترنت الأشياء.

تتكون البنية المعمارية لانترنت الأشياء بشكل عام من أربع طبقات كما هو موضح في الشكل(2–1)، وسيتم شرحها بشكل موجز [23] [12] :

1– طبقة الحساسات (Sensors Layer) : وهي الطبقة الدنيا في البنية المعمارية، وتتكون من شبكات الحساسات، والأنظمة المضمنة، وحساسات (RFID)Radio-frequency identification وغيرها من أنواع الحساسات. تسمح هذه الطبقة بالتفاعل مع الوسط الفيزيائي، والرقمي عن طريق جمع،ومعالجة المعلومات بالزمن الحقيقي. تستطيع الحساسات أخذ قياسات مختلفة كالحرارة، والرطوبة، والسرعة، والتدفق، والضغط، وغيرها من الظواهر الفيزيائية، ومن ثم تحويلها إلى إشارات مفهومة من قبل الآلة.



الشكل(2–1) البنية المعمارية لانترنت الأشياء

2– طبقة البوابة والشبكة (Gateway and Network Layer) : هذه الطبقة مسؤولة عن نقل المعلومات التي تم جمعها من قبل الحساسات إلى الطبقة اللاحقة. ينبغي أن تدعم هذه الطبقة بروتوكولات قياسية، ومرنة، وعالمية لنقل البيانات بين مختلف أنواع الحساسات كما ينبغي أن تمتلك شبكة ذات أداء عالي .

3– طبقة خدمة الإدارة (Management Service Layer) :تعمل هذه الطبقة كواجهة (منفذ اتصال) بين طبقة البوابة والشبكة، وطبقة التطبيقات؛ وهي مسؤولة عن إدارة الأجهزة، والمعلومات، والتقاط

كميات كبيرة من البيانات، واستخراج معلومات ذات صلة من البيانات المخزنة، ومن بيانات حالة الأجهزة المتبادلة في الزمن الحقيقي .

4- طبقة التطبيقات (Application Layer) :وهي الطبقة العليا في نظام IoT التي تؤمن الواجهة العملية للمستخدم للوصول إلى مختلف التطبيقات ، من مثل : تطبيقات النقل ، العناية بالصحة ، الزراعة ، التطبيقات الحكومية ،....

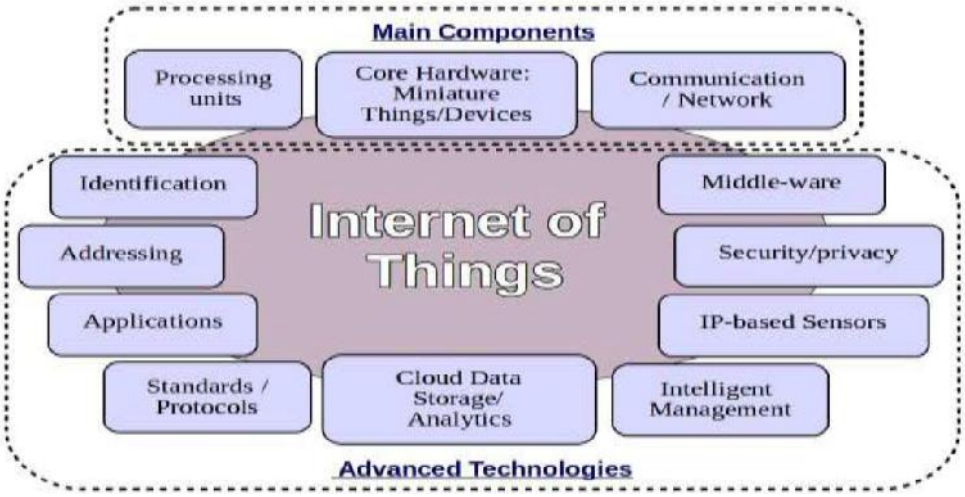
2-1-1 العلاقة بين طبقات انترنت الأشياء :

لفهم العلاقة بين هذه الطبقات، نأخذ مثالاً من الحياة العملية، وليكن الإضاءة الذكية على وفق شدة إضاءة الغرفة الذي يعمل كالآتي :

- بدايةً ، تقوم الحساسات بمراقبة شدة إضاءة الغرفة بشكل مستمر، وعندما يحدث انخفاض في شدة الإضاءة الغرفة إلى مستوى محدد، فإنّ الحساسات تتحسس لهذا الانخفاض وترسل هذه المعلومات، وهذا يمثل وظيفة طبقة Sensors Layer .
- تحول البيانات التشابهية التي أرسلت من قبل الحساسات إلى الشكل الرقمي، وترسل إلى النظام. ويمكن أن تكون قناة الاتصال Bluetoothأو Wi-Fi أو تردد راديوي ،حيث تقوم طبقة Gateway and Network Layer بوظيفة نقل البيانات .
- حالما تصل البيانات إلى النظام، يتم تخزينها وبناءً على هذه المعلومات يستنتج النظام أن شدة الإضاءة انخفضت إلى مستوى معين وبحاجة تغيير إضاءة الغرفة؛ وبالنتيجة اتخاذ الإجراء المناسب لتغيير الإضاءة على وفق البيانات المستلمة، وهذه وظيفة طبقة Management Service Layer .
- يتم تنفيذ إجراءات التحكم بالإضاءة التي يمكن للمستخدم ملاحظة التغير في السطوع على وفق تغير شدة الإضاءة،وهذه وظيفة Application Layer .

2-1-2 مكونات انترنت الأشياء:

تحتاج تقنية انترنت الأشياء إلى مجموعة من المكونات التي تعمل معاً لتشكل هذه التقنية، كما هو موضح في الشكل(2-2) ، والتي سيتم شرحها بشكل موجز [13]



الشكل(2–2) المكونات المشكّلة لتقنية انترنت الأشياء

- التعريف والعنونة (Identification and Addressing) : قد يحتوي انترنت الأشياء على عدد كبير من الأجهزة المنتشرة في أي مكان في العالم، وبالنتيجة فإنّ تحديد الأجهزة، ومخططات العنونة من التقنيات الأساسية لانترنت الأشياء.

يجب لكل غرض مضّمن في شبكات انترنت الأشياء أن يكون معرّفا من حيث الموقع، والوظيفة. بالإضافة إلى إمكانية التحكم فيها عن بعد عن طريق الانترنت،ولتحقيق ذلك استخدمت تقنيات مختلفة مثل : RFID ، IPV6 [28] [29].

- الحساسات المضمّنة (Embedded Sensors) : تقدمت شبكات الحساسات اللاسلكية مع تقدم تقنيات الحساسات، وأصبح بإمكانها العمل قي بيئات خطرة وقاسية. تستخدم شبكات الحساسات اللاسلكية عدداً كبيراً من الحساسات المنتشرة مكانيا أو الحساسات المضمّنة، التي تتعاون بشكل كبير مع تقنية RFID .

تستخدم هذه العناصر لأداء وظائف مختلفة مثل مراقبة الظروف الفيزيائية، وتعقب حالة الأشياء مثل الموقع، والحرارة،والحركة .

إن التحسينات في حجم ووزن الأجهزة، واستهلاك الطاقة، والكلفة، والتحسينات في الاتصالات اللاسلكية، مكّن تقنية IoT من استخدام الحساسات الذكية كتقنية أساسية في معظم شبكاتها.

تستخدم شبكات الحساسات الذكية تقنية الاستشعار عن بعد بالزمن الحقيقي، مما يعطيها القدرة على جمع، وتحليل، ومعالجة، ومشاركة، وتوزيع مختلف المعلومات الفيزيائية ضمن الأنظمة المركزية.

- البروتوكولات والبرمجيات الوسيطة (Protocols and Middleware):يحتوي انترنت الأشياء على ملايين الأجهزة التي تملك إمكانات مختلفة، وتحتاج إلى وسائل لتبادل، ونقل المعلومات المجمعة، والمولدة في مستوى الأجهزة .

ينبغي أن تكون الأجهزة المستخدمة ضمن انترنت الأشياء، قادرة على الاتصال مع بعضها البعض؛ أي قادرة على تحديد الوجهة إلى الهدف، وفهم الرسائل المستقبلة، والاستجابة بشكل مناسب. لتحقيق ذلك أصبح استخدام بروتوكولات اتصال قياسية من الاحتياجات الأساسية في مجال IoT .

تعطي البرمجيات الوسيطة إمكانية تفاعل الأشياء مع بعضها البعض عبر الانترنت، فهي تعمل كواجهة تمكّن مختلف التطبيقات المتواجدة على أنظمة غير متجانسة من التواصل مع بعضها البعض بسهولة، وسلاسة.

إن لطبقة البرمجيات الوسيطة دور أساسي في تنفيذ الإجراءات المعيارية، التي تستخدم كمكتبات مضمّنة ضمن التطبيقات، مما يسهل تطويرها ضمن البيئة الموزعة بشكل مستقل عن التقنيات المعيارية.

- التخزين والتحليل السحابي (Cloud-based Storage and Analytics):تحتاج بعض تطبيقات الـ IoT إلى مساحة تخزين، ومعدل معالجة عاليين لتحقيق التحكم في الزمن الحقيقي. إضافة إلى حزم بيانات عالية،مما يتطلب استخدام خوارزميات ذكية لتلبية هذه المتطلبات. يؤمن التخزين، والتحليل، والمعالجة السحابية حلاً مثالياً للتخزين والمعالجة بالزمن الحقيقي لكم كبير من البيانات من عدد كبير من الأجهزة .
- الأجهزة الأساسية (Core Hardware) : تتضمن الذواكر ، وحدات المعالجة ، وحدات الطاقة ، تقنيات اتصال وغيرها .

تتفاعل هذه الأجهزة مع التقنيات الأخرى لالتقاط التغيرات، والحوادث الفيزيائية، ومن ثم تمريرها إلى التطبيقات التي تعالجها، وتفسرها إلى معلومات ذات معنى .

2-2 خصائص انترنت الأشياء:

تتمتع تقنية انترنت الأشياء بالخصائص الآتية :[17]

- الاتصال Communication: يؤمن انترنت الأشياء تبادل المعلومات بين عناصر الأنظمة (أشخاص وأجهزة)، كمثال على ذلك، حالة جهاز ضمن نظام معين (متصل أو غير متصل ، مغلق أو مفتوح ، فارغ أو ممتلئ)؛ إذ إنّ هذه المعلومات لم يكن بالإمكان الوصول إليها تلقائياً،أو كانت تجمع بشكل يدوي ومتقطع .

- التحكم والأتمتة Control and Automation: يسمح انترنت الأشياء بالتحكم عن بعد في الأنظمة المتصلة بالانترنت، على سبيل المثال :
يمكن لنظام معين تشغيل أو إطفاء جهاز محدد أو وظيفة منه كتغيير درجة حرارة بيئة العمل في منظومة متحكم بها مناخياً.
- توفير الكلفة Cost Saving: يمكن لانترنت الأشياء أن يستخدم معلومات حساسات النظام لمراقبة الأجهزة، وإجراء صيانة مخطط لها بناء على هذه المعلومات، مما يساعد مسؤول النظام على خفض تعطل الأجهزة، وبالنتيجة خفض كلفة التشغيل.

3-2 بروتوكولات الاتصالات في انترنت الأشياء:

كما ذكرنا سابقا تسمح تقنية انترنت الأشياء للأجهزة من الاتصال مع بعضها البعض، و لتحقيق ذلك فإنها تستخدم بروتوكولات اتصال، تستخدم عند نقل البيانات بين أجهزة انترنت الأشياء ومنصات السحابة أو بين منصة السحابة، وتطبيقات انترنت الأشياء، التي سيتم شرحها بشكل موجز [26] :

- HTTP/HTTPS : هو برتوكول متواجد في طبقة التطبيقات يستعمل برتوكول TCP ويستعمل لنقل البيانات عبر الشبكة. إنّ بروتوكول HTTP/HTTPS هو بروتوكول عديم الحالة؛ أي إنه لا يعدل حالة الاتصال، وينهي الاتصال بعد كل استجابة لطلب استعلام .
يتم إنشاء اتصال جديد عند استعلام جديد، وهذا يجعل من الصعب استعمال HTTP بمفرده بالنسبة إلى انترنت الأشياء كونها – أي أجهزة انترنت الأشياء– ، تولد حجم كبير من البيانات في الزمن الحقيقي، وعملية فصل ووصل الاتصال، سيضيف عبأً على المخدم . لذلك تقنية انترنت الأشياء بحاجة إلى بروتوكول خفيف وذو موثوقية .
- WebSocket : هو برتوكول تتبع الحالة، حيث يرسل البيانات بالاتجاهين، ويبقي الاتصال بشكل مستمر. مصمم للعمل في طبقة النقل؛ وهو مكمّل لبرتوكول HTTP.يتم الاتصال بواسطة Socket واحدة بين الزبون والمخدّم، ويمكن لبرتوكولات أخرى أن تعمل على WebSocket مثل برتوكول MQTT و AMQP لإعطاء المزيد من المرونة والقوة .
يساعد HTML5 WebSocket في بناء تطبيقات الانترنت التي تعمل بالزمن الحقيقي .
- CoAP (Constrained Application Protocol) : هو برتوكول لنقل البيانات بين العقد التي تتمتع بخصائص محددة، ومقيدة، تعمل ضمن الشبكات ذات الخصائص المقيدة، من مثل: الشبكات منخفضة الطاقة ذات الضياعات. يعتمد هذا البرتوكول على نموذج request/response ؛ وهو مصمم لتطبيقات آلة إلى آلة (M2M) Machine to Machine، ويستعمل بروتوكول UDP. نظراً لكون بروتوكول UDP من البرتوكولات غير الموثوقة، اعتمد برتوكول CoAP في

تحقيقه للموثوقية من خلال رسالة تأكيد (CON (confirmable message ورسالة تأكيد استلام ACK (acknowledgement).

- AMQP(Advanced Message Queuing Protocol) : هو برتوكول تبادل رسائل مفتوح المصدر، يسمح ببناء تطبيقات عبر المنصات باستخدام مكتبات متعددة ووسطاء تجاريين .

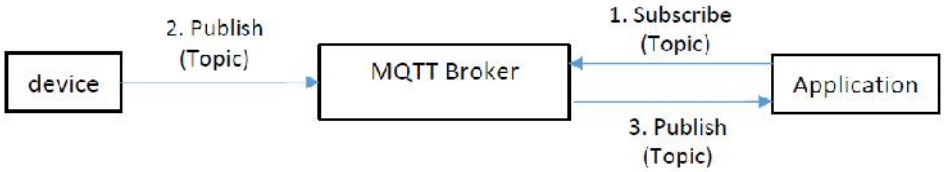
يبين الشكل(2-3) مبدأ عمل بروتوكول AMQP :

الشكل(2-3) مبدأ عمل بروتوكول AMQP

ينشر المرسل البيانات على شكل رسالة إلى المخدم Exchange(جزء من broker) ، الذي يقوم بمعالجتها، و توجيهها إلى قائمة الانتظار، ومن قائمة الانتظار، ترسل الرسالة إلى المستقبل الذي طلب هذا البيانات.يؤمن AMQP اتصال باتجاهين، ويمتاز بأنه موثوق من حيث ترتيب الإطارات. [27]

- MQTT(Message Queuing Telemetry Transport) : صمم عام 1999 للتحكم بخطوط أنابيب النفط في الصحراء والهدف منه تصميم برتوكول فعال من حيث عرض الحزمة، واستهلاك الطاقة .

يبين الشكل(2-4) مبدأ عمل بروتوكول MQTT :



الشكل(2-4) مبدأ عمل بروتوكول MQTT

إنّ برتوكول MQTT هو برتوكول مقاد بالحدث يسمح للرسائل المتضمنة في البيانات أن ترسل إلى المشتركين .

في البداية يقوم المشترك بتقديم طلب للاشتراك في خدمة ما، والتي تتيح له استقبال القيم، التي تم نشرها من قبل أجهزة انترنت الأشياء بما يتوافق مع تلك الخدمة .

يظهر الجدول الآتي مقارنة بين البرتوكولات السابقة كما ورد في المرجع [26] :

الجدول(2–1) مقارنة بين البروتوكولات المستخدمة في انترنت الأشياء

	HTTP/HTTPS	WebSocket	CoAP	AMQP	MQTT
Standard	IETF	IETF	IETF	OASIS & ISO	OASIS
Messaging	Request/ Response	Request/ Response	Request/ Response	Publish/ Subscribe	Publish/ Subscribe
Transport	TCP	TCP	UDP	TCP	TCP
Security	HTTP is not secure. HTTPS, runs over SSL/TLS is secure.	WebSocket (ws://) is not secure. WebSocket over SSL/TLS (wss://) is secure.	CoAP uses DTLS (Datagram Transport Layer Security).	AMQP uses TLS or SASL (Simple Authentication and Security Layer).	By default, MQTT does not use any encryption. But, "secure-MQTT" with SSL/TLS security can be used.
Quality of service (QoS)	No QoS	No QoS	Yes, available	Yes, available	Yes, available

يمكن الاستنتاج من الدراسات المرجعية، أنّ استعمال البرتوكولات الخفيفة من مثل :CoAP، MQTT،AMQP مناسب للتطبيقات التي تعتمد الاتصال من نوع جهاز إلى جهاز آخر أو جهاز إلى سحابة، في حين أنّ برتوكولات HTTP/HTTPS و WebSocket مناسبة للتطبيقات التي تعتمد الاتصال من نوع ويب إلى ويب (زبون إلى زبون) مثل الاتصال بين سحابة وتطبيق انترنت أشياء .

2–4 منصات انترنت الأشياء:

عندما نقوم بتطوير التطبيقات فإن المنصة تسمح لهذه التطبيقات بالعمل، والتنفيذ، حيث يمكن أن تكون المنصة تركيبة من الأجهزة، والبرامج، أو قد تتكون من الأجهزة التي يعمل عليها نظام التشغيل الذي سيسمح للتطبيقات بالعمل عليه، عبر تأمين بيئة التنفيذ الضرورية. تقدم منصة انترنت الأشياء مجموعة شاملة من الوظائف المستقلة، التي يمكن استعمالها لبناء تطبيقات IOT.

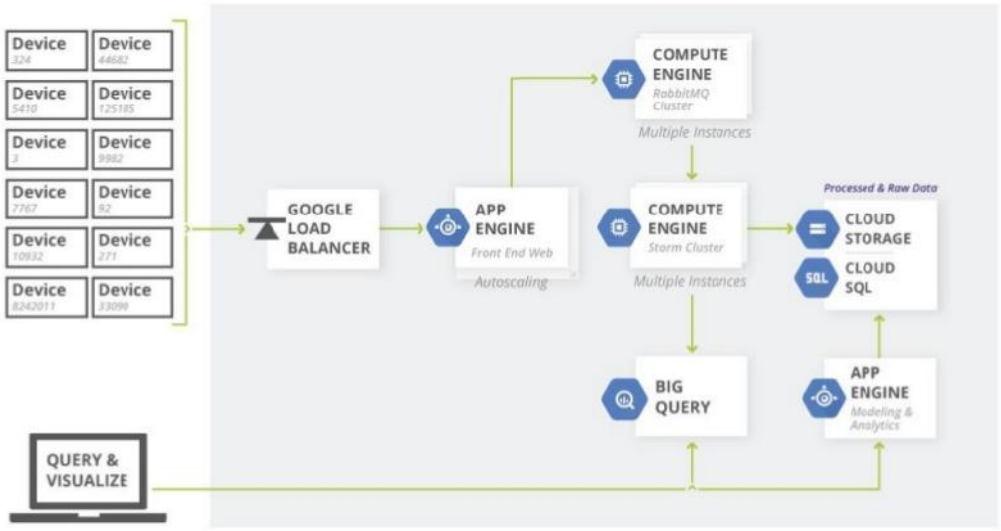
عندما يكون هناك خط اتصال واحد بين أجهزة من نوع واحد، وأجهزة أخرى من النوع نفسه، يمكن استعمال نظام بخدمة محددة لتأمين المطلوب. أما في حالة الاتصال بين أجهزة مختلفة الأنواع فإننا بحاجة إلى منصة تطبيقات قياسية شائعة، تخفي عدم التجانس بين أنواع الأجهزة عبر تأمين بيئة عمل مناسبة.

تعدّ منصة انترنت الأشياء حلاً افتراضياً؛ أي إنها تتواجد على سحابة ، حيث تعدّ البيانات الكيان الذي يحقق ذكاء التطبيقات، وكل جهاز يتواصل مع الأجهزة الأخرى عبر البيانات.[14]

تستطيع منصة انترنت الأشياء أن تترجم بيانات الأجهزة إلى معلومات بواسطة الاتصال السحابي، وبذلك تؤمن للمستخدم الوسائل الضرورية لتنفيذ الأعمال، والصيانة الوقائية،والتحليل وإدارة البيانات بالزمن الحقيقي؛ وبذلك تؤمن المنصة مجموعة كاملة من الخدمات لتطوير التطبيقات، وصيانتها.

سنتطرق في الفقرات الآتية إلى بعض المنصات الأكثر استخداما في تطوير، وتنفيذ تطبيقات انترنت الأشياء[14]:

1. منصة **Google Cloud:** يستطيع المطورون برمجة، واختبار، ونشر تطبيقاتهم ضمن بنية تحتية قياسية ذات موثوقية عالية، تمّ تطوير هذه البنية من قبل غوغل، الذي بدوره يستعملها. يتوجب على المطورين أن يهتموا بأمور البرمجة فقط، بينما تهتم منصة غوغل بباقي الأمور التي تتعلق بالبنية، وقوة الحوسبة، ومكان تخزين البيانات. تعدّ منصة غوغل أحد أشهر المنصات؛ بسبب سرعة الشبكة العالمية، و أدوات غوغل للبيانات الضخمة، التي تدعم مختلف خدمات السحابة .
يظهر الشكل(2–5)معالجة التدفق بالزمن الحقيقي من قبل غوغل، حيث تقوم الأجهزة بإرسال معلومات الحالة إلى APP ENGINE.



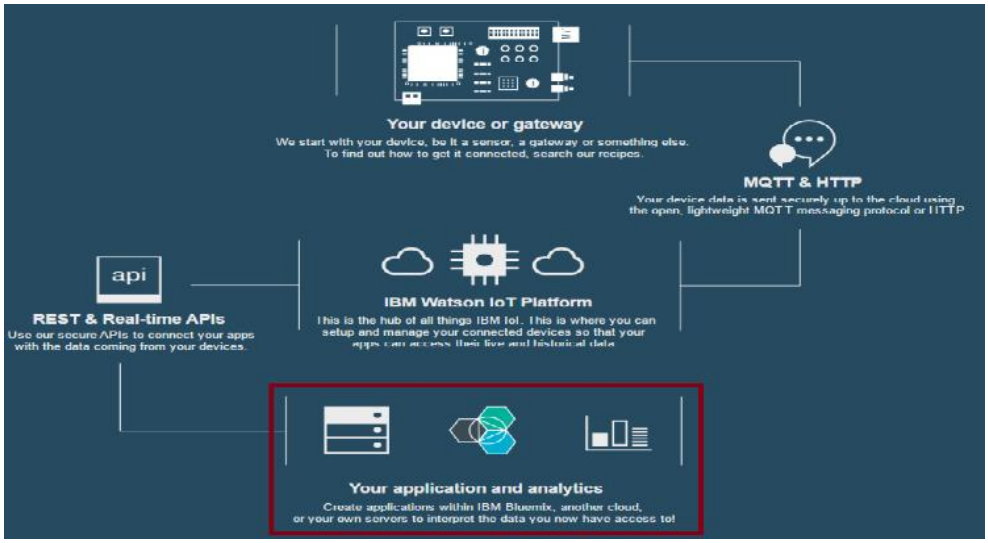
الشكل(2–5) منصة Google Cloud

بداية يتأكد LOAD BALANCER بأن الحمل متوازن بين مختلف التطبيقات APP ENGINE ثم ينجز COMPUTE ENGINE حساب البيانات وينشرها .
يمكن استعمال نماذج مختلفة من COMPUTE ENGINE لضمان التوافقية مع المعايير القياسية والموثوقية .تستخدم CLOUD STORAGE لتخزين واستعادة البيانات. يسمح BIG QUERY بالإدخال السريع للبيانات في جداول قاعدة بيانات السحابة.وأخيرا تظهر النتائج للمستخدم عبر وسائل تحليل، وتقنيات إظهار مختلفة.

2. منصة **IBM BlueMix :**هي سحابة من نوع (PaaS)Platform as a Service مطورة من قبل IBM ، تدعم لغات برمجة متعددة من مثل: Java، Php ، Python ، Node.js، Go. تؤمن منصةIBM BlueMix وصولاً قوياً للتطبيقات إلى بيانات انترنت الأشياء وأجهزتها.يدعم IBM BlueMix التطور السريع للتطبيقات التحليلية، وتطبيقات انترنت الأشياء الخاصة بالهواتف الذكية. تعدّ منصة IBM BlueMix لانترنت الأشياء المجمع الأساسي، حيث تستطيع بناء وإدارة الأجهزة المتصلة، التي تستعمل بروتوكول MQTT لإيصال البيانات إلى السحابة بشكل آمن .

تؤمن هذه المنصة خدمات مجانية محدودة منها خدمة **Watson IoT platform** التي كانت كافية لتحقيق هدف المشروع المتمثل بالتحكم ومراقبة تجهيزات معمل تعبئة الغاز المنزلي .

يبين الشكل(2-6) تفاعل مكونات منصة **IBM BlueMix** مع بعضها البعض .



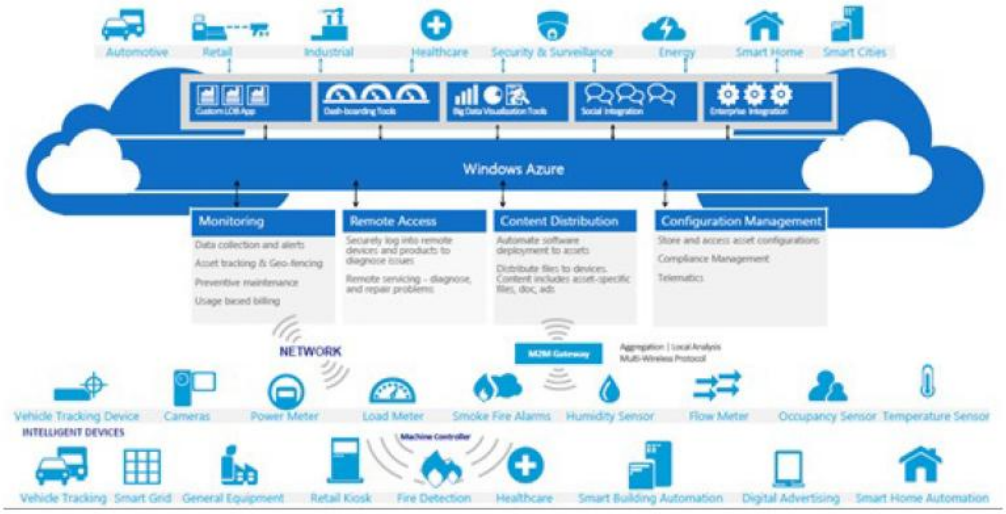
الشكل(2-6) منصة IBM BlueMix

3. منصة **ThingWorx:**هي أول منصة برمجية مصممة لبناء، وتشغيل التطبيقات المتصلة حول العالم. تركز منصة ThingWorx على تقليل الزمن، والكلفة، والخطورة التي يتطلبها بناء تطبيقات من نوع آلة إلى آلة (M2M) ، وتطبيقات انترنت الأشياء. تتميز هذه المنصة بتسريع تطوير تطبيقات انترنت الأشياء عبر ضغط دورة (التصميم – التطوير –النشر)،وتقليل زمن جاهزية التطبيق للاستخدام.تدعم منصةThingWorx التصميم، والبناء السريع لتطبيقات ذكية من مثل: تطبيق المنزل الذكي، والشبكة الذكية، وشبكة المياه الذكية، وغيرها من التطبيقات، من خلال توفر خدمات لهذه التطبيقات.

يوضح الشكل (2-7) تفاعل مكونات منصة **ThingWorx** مع بعضها البعض.

الشكل(2–7) منصة ThingWorx

4. منصة **Microsoft Azure** : تؤمن شركة مايكروسوفت خدمة النظام الذكي **Microsoft Azure**الذي يشكل منصة مدمجة، تحتوي على خدمات، تستخدم في بناء أنظمة، وتطبيقات انترنت الأشياء عبر جمع، وتخزين، ومعالجة البيانات. يساعد نظام خدمات ذكي مبني من قبل Microsoft Azure على الاتصال الآمن، وإدارة،والتقاط ، وتحويل بيانات الآلة إلى بيانات مفيدة. يتيح نظام الحوسبة السحابية لميكروسوفت القيام بعمليات جمع البيانات، وتحليلها، ومعالجتها.تؤمن السحابة حلول تخزين، ومعالجة، واستخدام، وتحليل البيانات بالزمن الحقيقي.
يبين الشكل(2–8)تفاعل مكونات منصة **Microsoft Azure**مع بعضها البعض .



الشكل(2–8) منصة Microsoft Azure

5. منصة ThingSpeak : هي منصة تطبيقات لتطوير أنظمة انترنت الأشياء ،التي تساعد على بناء التطبيقات المعتمدة على البيانات المجمعة من قبل الحساسات. تعدّ منصة ThingSpeak من المنصات مفتوحة المصدر، التي تستخدم لتطوير تطبيقات انترنت الأشياء.تستخدم هذه المنصة

في تطوير وأمثلة الأنظمة الموجودة للاستفادة من تقنيات انترنت الأشياء. تؤمن منصة ThingSpeak القدرة على مشاركة البيانات مع مجموعة واسعة من منصات انترنت الأشياء الأخرى.

تؤمن منصة ThingSpeak الإمكانات الآتية :

- الجمع : إرسال البيانات المجمعة من الحساسات إلى السحابة، وبذلك يمكن تحليلها لاحقاً.
- التحليل : تستطيع المنصة تحليل البيانات المستقبلة من الحساسات، واستخلاص التمثيل الافتراضي للبيانات .
- التمثيل : بناءً على التحليل تقوم المنصة بتفعيل الاستجابة المناسبة لتحقيق هدف تطبيقات نظام انترنت الأشياء .

يظهر الجدول الآتي مقارنة بين منصات انترنت الأشياء السابقة من حيث قابلية التوسع ، التوفر ، الأمن، والخصوصية ، والتوصيل والتشغيل ، والدعم لملايين الأجهزة، وغيرها من البارامترات .

تشير الاختصارات في الجدول إلى :H (high) مرتفع ، M(Moderate) معتدل ، F.T (Free Trial) تجريب مجاني ، O.S(Open Source) مفتوح المصدر.

الجدول(2–2) مقارنة بين بعض منصات انترنت الأشياء

	Google Cloud Compute	**IBM Blue Mix**	**ThingWorx**	**Microsoft Azure**	**ThingSpeak**
Scalability	√	√	√	√	√
24*7 Availability	√	√	√	√	√
Security and Privacy provisioning	√	√	√	√	X
Plug and Play	√	√	√	√	X
Support for millions devices	√	√	√	√	X
Real time data	√	√	√	√	√
Storage of data	√	√	√	√	X
Provision of support	√	√	√	√	√
Developer Friendly	H	H	M	H	M
Solution Type	PaaS	PaaS	Complete IoT	PaaS	Data Analytic
Solution Type	F.T	F.T	F.T	F.T	O.S

الفصل الثالث

تحليل وتصميم نظام إدارة ومراقبة معمل تعبئة الغاز المنزلي

يتكون هذا النظام من ثلاثة أقسام رئيسة كما هو موضح في الشكل(3-1):

1. تجهيزات معمل تعبئة الغاز المنزلي: دراستنا هذه ستكون عبارة عن تجهيزات افتراضية ممثلة بواسطة صفحة ويب عن طريق تصميم محاكي لمعمل التعبئة، يحاكي الخدمات الأساسية من مثل: عملية التحكم بخط السير، ونظام الإنذار، وإطفاء الحريق.
2. واجهة التحكم وإدارة تجهيزات معمل التعبئة: هي عبارة عن صفحة ويب –أيضاً– تحتوي على مجموعة من أزرار التحكم، إضافة إلى معلومات تخص عملية التعبئة، مهمتها التحكم ومراقبة المعمل المدروس عن بعد باستخدام تقنية انترنت الأشياء .
3. منصة انترنت الأشياء: استخدمنا في هذه الدراسة منصة الـ **IBM Watson IoT**، التي سبق ذكرها في قسم منصات انترنت الأشياء، لتبادل البيانات بين تجهيزات المعمل وواجهة التحكم .

الشكل(3-1) الأجزاء الرئيسة لنظام إدارة و مراقبة معمل تعبئة الغاز المنزلي

سنقوم في هذا الفصل بدراسة كل قسم من أقسام النظام على حدا ، وشرح مكونات المشروع الخاصة بكل قسم.

3-1 تجهيزات معمل تعبئة الغاز المنزلي :

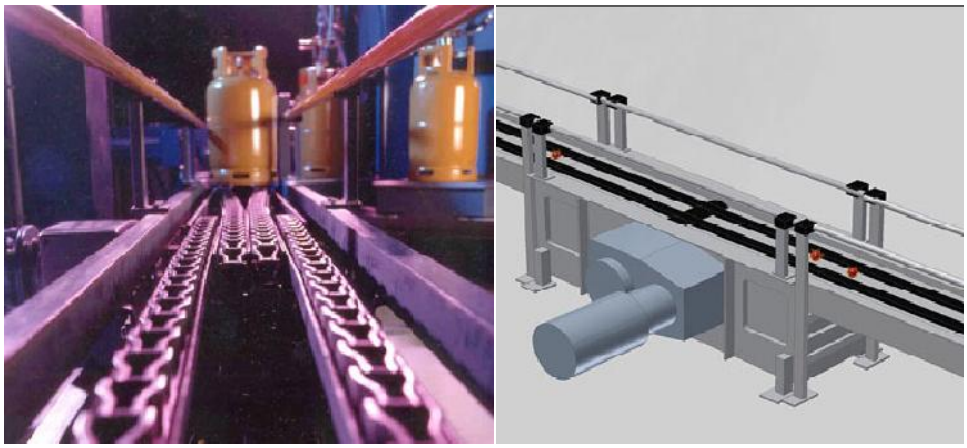
يتكون الغاز المنزلي بشكل أساسي من غاز البوتان أوالبروبان أو مزيج من كليهما. إضافة إلى غازات أخرى بنسب صغيرة، وهذا المزيج عديم اللون، والرائحة، وسريع الاشتعال. يستعمل الغاز المنزلي في الطبخ، والتدفئة، ونتيجة تزايد الطلب عليه ظهرت معامل التعبئة الآلية.[15]

تحتوي معظم معامل تعبئة الغاز المنزلي بشكل عام على التجهيزات الرئيسة الآتية: [15] [16]

1) وحدة الغسل: تستعمل لغسيل الاسطوانات من الأوساخ العالقة بها، وهي تستخدم مضخة ماء لتنظيف الأسطوانات، ومن ثم تجفيفها عن طريق وحدة تجفيف مرتبطة معها، كما هو موضح في الشكل(3-2).

الشكل(3-2) وحدة الغسل

2) السير الناقل للأسطوانات: هو عبارة عن سلاسل متصلة ميكانيكيا بمحرك كهربائي يعمل على تحريكها، ناقلا الاسطوانات على طول خط الإنتاج، كما هو موضح في الشكل (3-3).



الشكل(3-3) السير الناقل للأسطوانات

3) وحدة وزن الاسطوانة : مهمة هذه الوحدة قياس وزن الاسطوانات الفارغة قبل إدخالها إلى وحدات التعبئة، وذلك بسبب وجود أسطوانات غاز مختلفة الأوزان، مما يستدعي معرفة وزن الاسطوانة فارغة لتعبئتها بكمية الغاز المحدد بشكل دقيق.

4) وحدة التعبئة الآلية الدائرية:هي آلة حديثة صممت لتعبئة اسطوانات الغاز بمجموعات كبيرة، وتتكون من إطار دائري يدور بواسطة عجلات حول محور،يؤمن الغاز والهواء كما هو موضح في الشكل(3-4).

مبدأ عمل هذه الوحدة هو إدخال الاسطوانة إلى أحد رؤوس التعبئة، وتعبئتها خلال عملية دوران الإطار، بحيث يتم تعبئتها بكمية الغاز المحددة عند وصولها إلى منفذ الخروج.يمكن أن يتراوح عدد رؤوس التعبئة في الإطار الدوار من ثمانية إلى اثنين وسبعين رأساً .

الشكل(3–4) وحدة التعبئة الآلية الدائرية

5) وحدة كشف التسرب: هي عبارة عن جهاز مصمم لكشف تسرب الغاز من صمام الأسطوانة الشكل(3–5)، وبالنتيجة تجنب تعبئة الاسطوانات المعطوبة.إنّ التسرب الذي ينتج من جسم الاسطوانة، يتم كشفه يدويا بالماء والصابون.

الشكل(3–5) وحدة كشف التسرب

6) نظام الإطفاء بالماء: يتكون من مضخة ديزل أو مضخة كهربائية مهمتها ضخ الماء إلى شبكة أنابيب خاصة بنظام الإطفاء موزعة في أرجاء المعمل الشكل(3–6).

الشكل(3-6) نظام الإطفاء بالماء

3-2 متطلبات تصميم نظام التحكم والمراقبة:

إن تشغيل محاكي معمل تعبئة الغاز المنزلي وواجهة التحكم والإدارة عن بعد يتطلب توافر المكتبات والبرامج الآتية على الحاسب المشغل:

- برنامج NetBeansIDE أو أي بيئة تطوير أخرى تسمح ببناء تطبيقات ويب جافا .
- عند استخدام برنامج NetBeansIDE ينبغي التأكد من تنصيب Java Web and EE وهي plugin تؤمّن الدعم لتطوير تطبيقات JSF .
- يحتوي ملف pom.xml على<dependencies> التي تسمح بتنزيل مكتبات الجافا الخاصة بمنصة Watson IoT ومكتبات أخرى ضرورية للتشغيل بشكل آلي من الانترنت عند التشغيل لأول مرة؛ لذلك يجب الانتظار ريثما ينتهي التنصيب لهذه المكتبات .
- إنّ البرنامج المستعمل في تنفيذ صفحات الويب هيGlassFish Server 4.1.

بعد التعرف على أهم التجهيزات الموجودة في معامل تعبئة الغاز، قمنا بنمذجة نظام يحاكي عملية التعبئة بما يخدم غرض دراستنا هذه، حيث تم تمثيل عملية التعبئة بمحاكاة التجهيزات 2 و 4 و 6 وذلك باعتبارها العناصر الضرورية، والكافية لإتمام عملية التعبئة، التي تؤدي غرض الدراسة المتمثل بالتحكم

بمعمل تعبئة الغاز المنزلي عن بعد عن طريق منصة انترنت الأشياء. اعتمدنا في نمذجة النظام على لغات البرمجة الآتية :

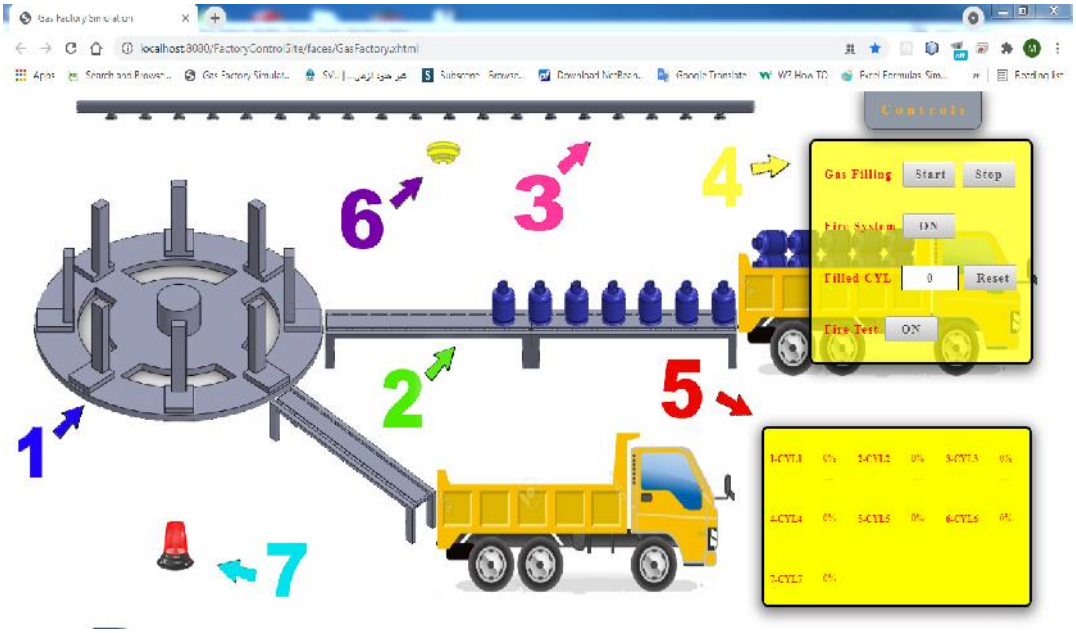
- لغة XHTML : تعتبر لغة XHTML هي اختصار لـ Extensible Hyper Text Markup Language تقاطع بين لغتي HTML و XMLهي شبيهة بلغة HTML ولكنها أكثر صرامة من حيث بناء الجملة، ومراعاة الأحرف الصغيرة والكبيرة[18] . تعدّ ملفات الـXHTML ذات تصميم جيد، وتستخدم مترجمات XMLالقياسية[18].
- لغة JavaScript :هي لغة برمجة ديناميكية خفيفة، تستخدم بشكل واسع في صفحات الويب،وتسمح للنصوص البرمجية من جهة الزبون بالتفاعل مع المستخدم، وجعل صفحات الويب ديناميكية[18] .
- لغة CSS (Cascading Style Sheet): هي لغة تصميم بسيطة هدفها تبسيط عملية تصميم صفحات ويب جيدة المظهر[18]. باستخدام CSS نستطيع التحكم بلون الخط ، نوع الأحرف ، لون خلفية الصفحة، وتوضع العناصر على الصفحة [18].
- لغة Java: هي لغة غرضية التوجه معتمدة على الصفوف (Class based) ،هي منصة حوسبة تستخدم لتطوير التطبيقات، وتتميز لغة الجافا بأنها لغة آمنة وموثوقة [21] .
- تقنيةJSF (java server faces):تستخدم في بناء واجهات المستخدم من جهة المخدم حيث تعدّ نموذجاً معتمداً على مكونات لتطوير تطبيقات الانترنت، مشابهة لتطبيقات GUI[20]. تؤمن تقنية الـ JSF مجموعة من المكونات (input text ، command button،..)، وتساعد على إدارة حالتها، كما تؤمن المصادقة من جهة السيرفر، وتبادل البيانات، والعرض في الصفحة.[19]
- jQuery: هي مكتبة ضمن لغة الـ javaScript تتميز بأنها بسيطة، ومدمجة مهمتها تبسيط استخدام javaScript في تصميم صفحات الويب .[22]

إن صفحات XHTML هي صفحات ساكنة (statics) وبالنتيجة لجعل الصفحة ديناميكية، استخدمنا لغة JavaScript التي تعطي ديناميكية للصفحة من جهة الزبون (المستخدم)، وتقنية الـ JSF(java server faces) التي تتيح استخدام عناصر يمكن التحكم فيها من جهة المخدم.

3-3 تصميم نظام التحكم والمراقبة:

3-3-1 المحاكي (التحكم المحلي):

يظهر الشكل(3-7) نموذج معمل تعبئة الغاز المقترح كمحاكي على صفحة ويب والأجزاء الرئيسة له :



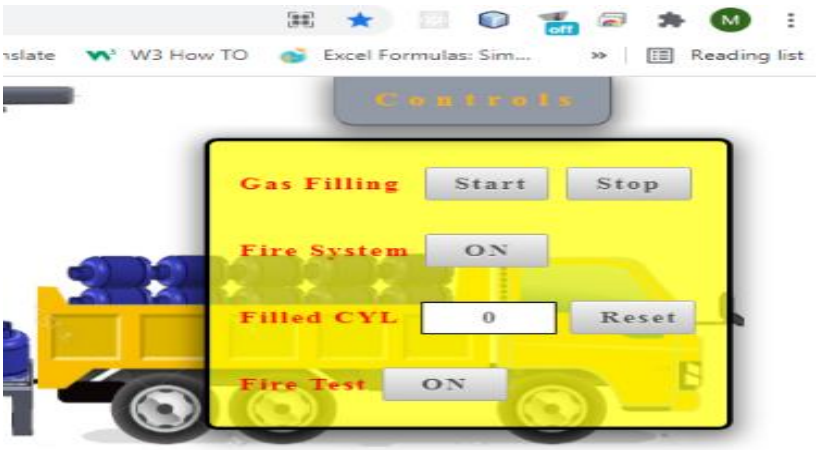
الشكل(3–7) نموذج معمل تعبئة الغاز المقترح كمحاكي على صفحة ويب، والأجزاء الرئيسة له

1) وحدة التعبئة الآلية الدائرية .
2) السير الناقل للأسطوانات .
3) نظام الإطفاء بالماء .
4) واجهة التحكم في المعمل .
5) واجهة إظهار النسبة المئوية لعملية تعبئة الأسطوانات لكل وحدة تعبئة على حدا .
6) حساس الحريق .
7) جهاز إنذار .

3–3–1–1 مبدأ العمل :

يمكن تشغيل وإيقاف نظام التعبئة ونظام الإطفاء بطريقتين :

- الطريقة الأولى :باستخدام واجهة التحكم محلية Local Control Panel المضمنة في نموذج المعمل كما هو موضح في الشكل(3–8) و تمثل التشغيل المباشر للمعمل (التشغيل ضمن المعمل حصرا) مهمتها تشغيل نظام التعبئة، ونظام الإطفاء، ومراقبة عدد الأسطوانات المعبئة للمعمل،واختبار حالة حدوث حريق.
- الطريقة الثانية :باستخدام واجهة تحكم عن بعد Remote Control Panel وتمثل التشغيل غير المباشر للمعمل (التشغيل من أي مكان في العالم عن طريق متصفح ويب) و سيتم التحدث عنها فيما بعد.



الشكل(3-8) واجهة التحكم المحلية المضمنة في نموذج المعمل

بداية عند تشغيل المحاكي يتم استدعاء تابعين :

- التابع الأول مكتوب بلغة الJava: StateOutputTimer.order(ProductLine,device) حيث البارامتر device يمثل نوع الجهاز المتحكم فيه ضمن المعمل بينما المعامل ProductLine يمثل حالة الجهاز (حالة تشغيل أم إيقاف)، مهمته الاتصال وإرسال حالة خط التعبئة، ونظام الإطفاء جهاز الإنذار،إضافة إلى عدد الاسطوانات المعبأة إلى منصة انترنت الأشياء بشكل مستمر عن طريق مؤقت زمني.
- التابع الثاني من نوعJavaScript: ()preprocessing يستخدم كمؤقت زمني محلي مهمته استدعاء تابع java هو()commandReceived.command يقوم باستقبال الأوامر من واجهة التحكم عن بعد عبر منصة انترنت الأشياء، ليقوم بتغيير حالة نظام التعبئة، و نظام الإطفاء على وفق الأوامر المستقبلة.

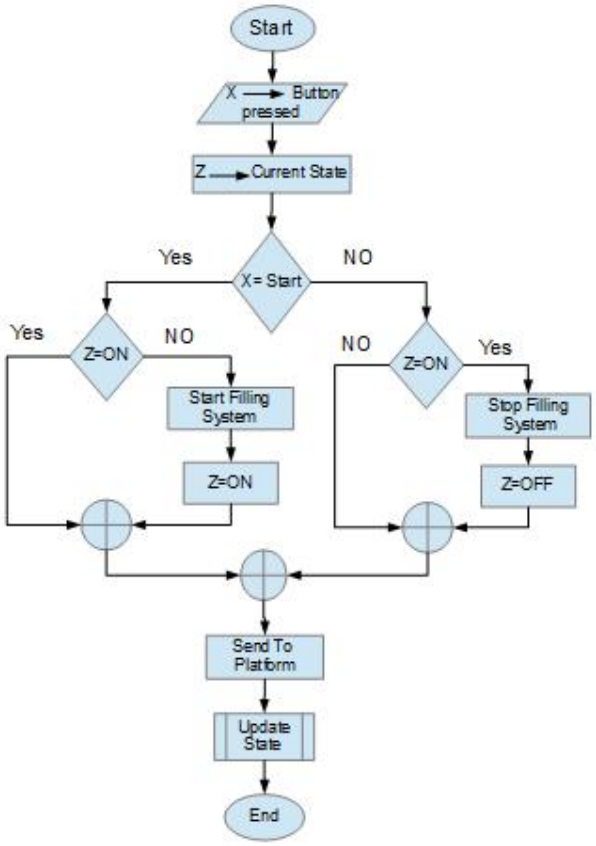
عند الضغط على زرStart الخاص بنظام التعبئة من واجهة التحكم المحلية، يتم استدعاء تابعي JavaScript و تابعي Java لها المهام الآتية :

- تابع JavaScript start(pos1, pos2): مهمته تشغيل نظام التعبئة من الإحداثيات الممررة له.
- تابع JavaScript btnState(btn): مهمته تفعيل أو إلغاء تفعيل الزر بحسب القيمة الممررة له ، و يستخدم لجعل عمل زر التشغيل أقرب إلى الواقع.
- تابع Java StateOutputTimer.order(state, device) : مهمته إرسال حالة الجهاز المتحكم فيه بحسب القيم الممررة له إلى منصة انترنت الأشياء (ON,OFF) .

- تابع java commandReceived.ProductLineState(state) : مهمته الاحتفاظ بحالة نظام التعبئة الحالية (تشغيل أو إيقاف) وبالنتيجة ضمان عدم تكرار تنفيذ الأمر المستلم من واجهة التحكم أكثر من مرة.

عند الضغط على زر الـ Stop، سيتم استدعاء تابع JavaScript stop() لإيقاف نظام التعبئة بدلاً من تابع start()، واستدعاء التوابع السابقة مع تمرير قيم تناسب إيقاف تشغيل نظام التعبئة .

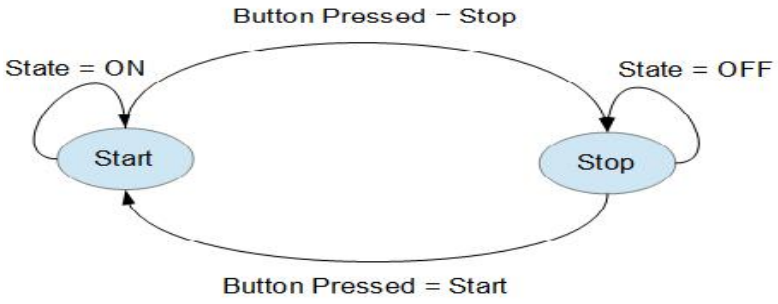
يمكن تمثيل حالة ضغط زر التشغيل أو الإيقاف لنظام التعبئة بمخطط تدفقي موضح في الشكل(3-9) حيث X تمثل الزر المضغوط (Start , Stop) و Z حالة الجهاز المتحكم فيه (ON,OFF).



الشكل(3-9) تشغيل وإيقاف نظام تعبئة الغاز المنزلي

يظهر المخطط التدفقي السابق التابع Update State ومهمة هذا التابع إعادة تنفيذ أخر شرط محقق لضغط الزر بشكل مستمر حتى يتغير هذا الشرط، وهذا يسمح بتحديث حالة نظام التعبئة بشكل مستمر.

يمكن تمثيل عمل تابع Update State بمخطط الحالة الآتي :

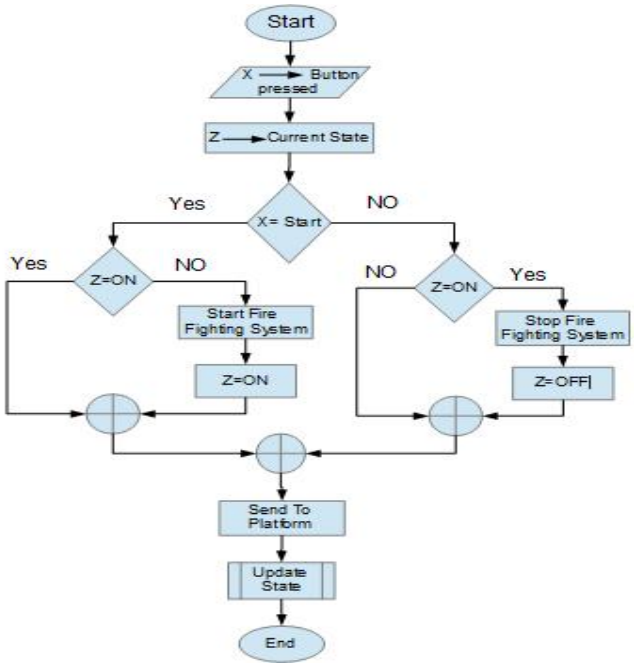


الشكل(3–10) مخطط الحالة لـ تابع Update State

عند ضغط زر Start الخاص بنظام الإطفاء[1] يتم استدعاء التوابع الآتية :

- تابع JavaScript firesystem() : مهمته تشغيل، وإيقاف نظام إطفاء الحريق .
- تابع Java StateOutputTimer.order(state, device): مهمته إرسال حالة نظام الإطفاء إلى منصة انترنت الأشياء بحسب القيمة الممررة له.
- تابع Java commandReceived.FireSystemState(): مهمته الاحتفاظ بحالة نظام الإطفاء الحالية؛ وبالنتيجة ضمان عدم تكرار تنفيذ الأمر المستلم من واجهة التحكم أكثر من مرة.

يمكن تمثيل حالة ضغط زر التشغيل أو الإيقاف لنظام الإطفاء بمخطط تدفقي موضح في الشكل(3–11)



الشكل(3–11) المخطط التدفقي لتشغيل و ايقاف نظام إطفاء الحريق

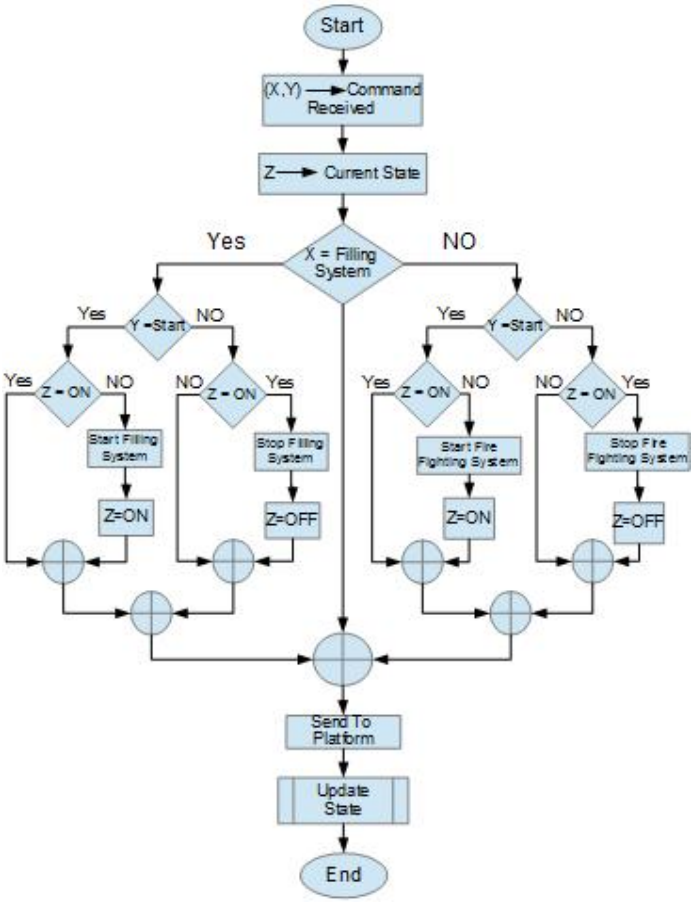
1هو من نوع Toggle أي أن قيمته تتغير عند الضغط عليه وفي حالتنا هذه من Start **إلى** Stop **وبالعكس**

إن عمل التابع Update Stateمماثل في كل من النظامين، نظام الإطفاء، ونظام تعبئة الغاز .

3-3-1-2 نظام التحكم عن بعد:

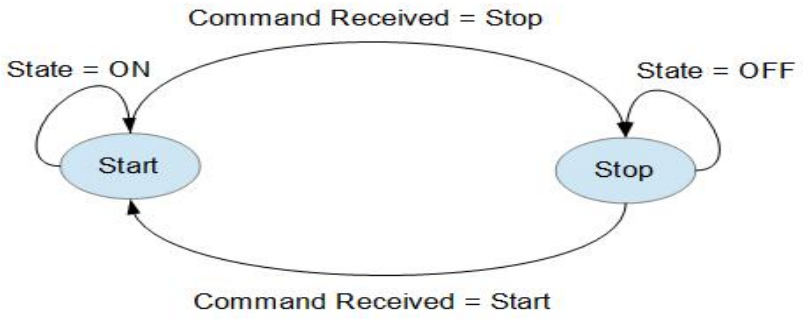
كما ذكرنا سابقاً ، يمكن تشغيل نظام تعبئة الغاز، ونظام الإطفاء عن طريق واجهة تحكم متواجدة على حاسب متصل بالإنترنت، وهذا يتم بمساعدة التابع commandReceived.command() الذي يعمل عند بداية تشغيل المحاكي، و يبقى في حالة انتظار لاستلام الأوامر من واجهة التحكم، **والإدارة عن بعد.**

يظهر الشكل(3-12) مبدأ عمل هذا التابع حيث X تمثل نوع العملية (Filling system, Fire Fighting system) و Y الأمر المرسل و Z حالة الجهاز المتحكم فيه (ON,OFF) .



الشكل(3-12) المخطط التدفقي لاستلام أمر من واجهة التحكم والإدارة عن بعد

يمكن تمثيل عمل تابع Update Stateضمن المخطط التدفقي السابق بمخطط الحالة الآتي :



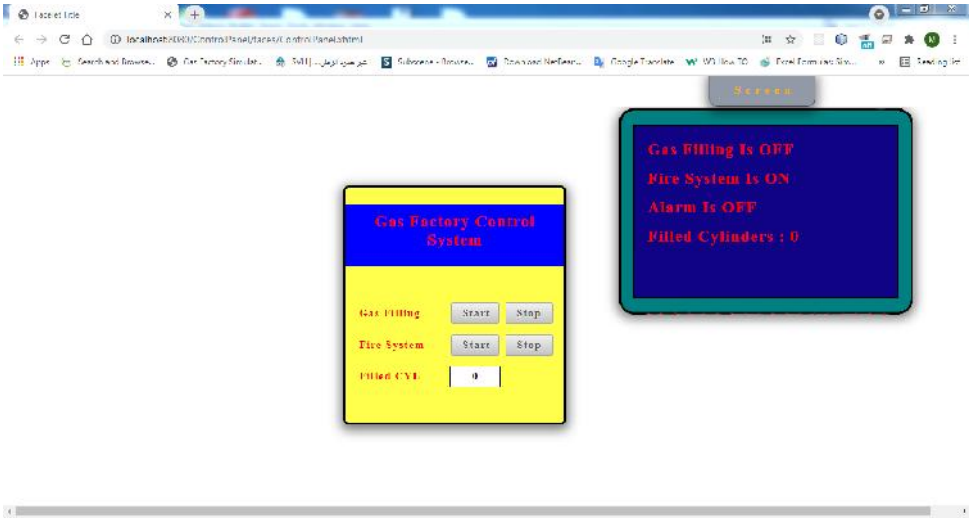
الشكل(3–13) مخطط الحالة لعمل تابع Update State

يسمح زر الـ fire test بمحاكاة نشوب حريق في المعمل، وبالنتيجة اختبار عمل نظام الإطفاء، الذي يقوم بتشغيله بشكل آلي، وإيقاف نظام التعبئة مع إرسال إنذار إلى واجهة التحكم، والإدارة عن بعد بحدوث حريق .

3-3-2 واجهة التحكم والإدارة عن بعد :

استخدمنا في تصميم هذه الواجهة التقنيات نفسها المستخدمة في محاكي نظام تعبئة الغاز.الغرض الأساسي لهذه الواجهة، التحكم بمعمل تعبئة الغاز المنزلي عن بعد من أي حاسب متصل بالانترنت، وذلك من خلال اتصال هذه الواجهة بمنصة انترنت الأشياء، تؤمن لها استقبال حالة أجهزة المعمل ، إضافة إلى إرسال الأوامر عبرها إلى نظام تعبئة الغاز.

يظهر الشكل(3–14)واجهة التحكم والإدارة عن بعد المعتمدة على تقنيات الويب.



الشكل(3–14) واجهة التحكم والإدارة عن بعد المعتمدة على تقنيات الويب

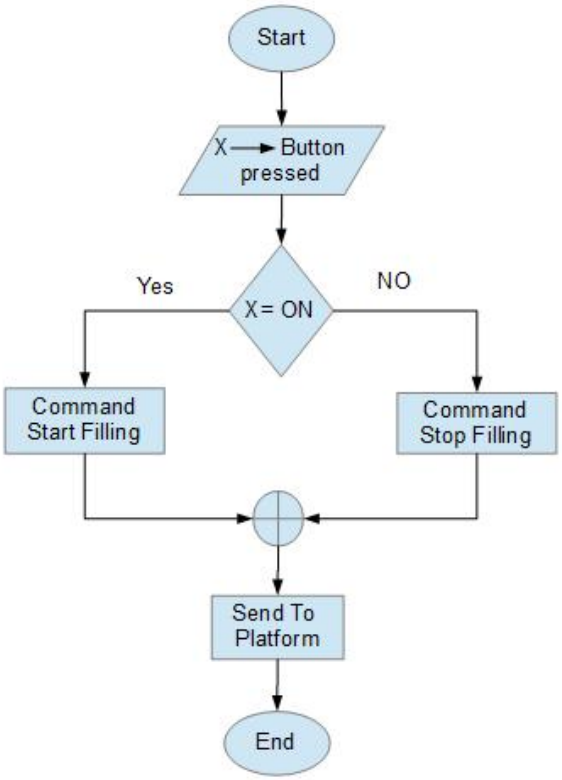
3-3-2-1 مبدأ العمل :

عند تشغيل واجهة التحكم والإدارة عن بعد، يتم استدعاء تابعRecievedStatesbean.recievedStates() مكتوب بلغة الـJavaعن طريق تابع JavaScript، الذي يعمل كمؤقت زمني. تكمن مهمة تابع الjava في استقبال حالة نظام التعبئة، ونظام الإطفاء، وتغيير حالة أزرار التحكم، وقيم شاشة الإظهار عل وفق الحالة المستقبلة بشكل مستمر.

عند الضغط على زر الـ Start الخاص بنظام التعبئة، يتم استدعاء تابع commandBean.order(state,device) ممرر له قيم تمثل تشغيل نظام التعبئة.يقوم التابع السابق بالاتصال بمنصة انترنت الأشياء، ويرسل أمر التشغيل إلى المنصة ،التي ترسل بدورها الأمر إلى المحاكي (نظام التعبئة).

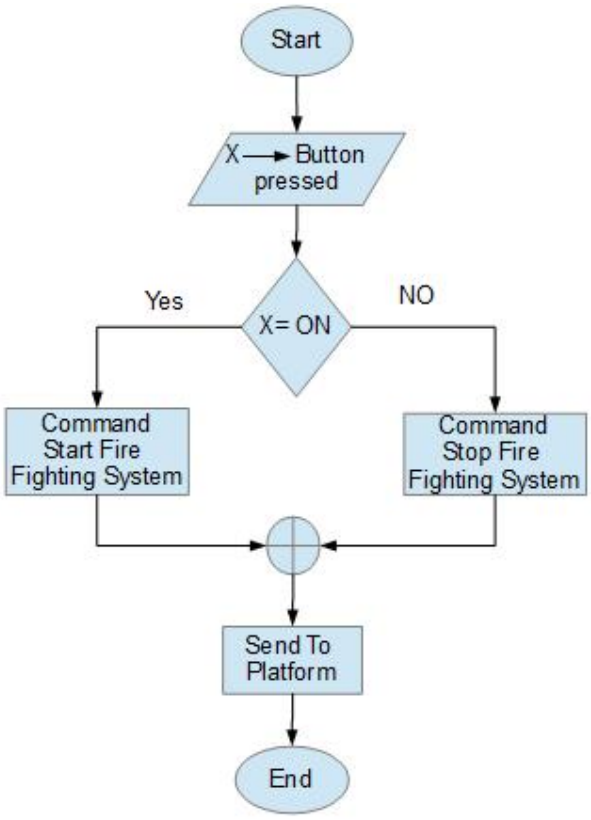
عند الضغط على الزر الـ Stop يتم استدعاء التابع commandBean.order(state,device) ممرراً له قيم تمثل إيقاف نظام التعبئة.

يمكن تمثيل حالة ضغط زر أمر التشغيل أو الإيقاف لنظام التعبئة بالمخطط التدفقي الموضح في الشكل(3-15)



الشكل(3-15) المخطط التدفقي لأمر تشغيل أو إيقاف نظام تعبئة الغاز المنزلي من واجهة التحكم

كذلك الأمر عند ضغط زر Start الخاص بنظام الاطفاء، سيتم استدعاء التابع commandBean.order(state,device) –أيضاً– مع تغيير القيم الممررة بما يتناسب مع تشغيل نظام الإطفاء.وبشكل مماثل يظهر الشكل(3–16) المخطط التدفقي لحالة ضغط زر التشغيل أو الإيقاف لنظام الإطفاء من واجهة التحكم عن بعد:



الشكل(3–16) المخطط التدفقي لأمر التشغيل أو الإيقاف لنظام الإطفاء من واجهة التحكم عن بعد

3–3–3 منصة انترنت الأشياءIBM Bluemix & Watson IoT Platform:

يوجد العديد من منصات انترنت الأشياء المتاحة، ولكن القليل منها يماثل حجم، ومدى تطور منصة الـ IBM.يحتاج الزبون إلى أن تكون التقنيات المطلوبة لانترنت الأشياء منخفضة السعر، وقابلة للتطور، ومرنة لتلبي مختلف احتياجاتهم. من وجهة نظر الزبون، فإنّ المهارة الأساسية المطلوبة، هي تطوير التطبيقات التي تحتاج إلى بنية أساسية، تقدم التطور السريع، وهذه المتطلبات غالباً لا تكون موجودة في بيئات الـ IT التقليدية البطيئة التطور، وذات البنية الأساسية باهظة الثمن. يعتمد انترنت الأشياء على التحليل المتطور لإدارة التطبيقات، وهذه التقنية الجديدة صعبة التحقيق، وبحاجة مهارات جديدة.تقدم IBM Bluemix & Watson IoT Platform الحلول لهذه المواضيع . [24]

منصة IBM Bluemix :هي بيئة تطوير متكاملة وشاملة مع مجموعة واسعة من الوظائف المتوفرة في سحابة.تجعل منصة IBM نشر التطبيقات بسيطاً مع قدرات توسع كبيرة بحسب الطلب. كما تقوم هذه المنصة بتزويد تطبيقات انترنت الأشياء متضمنة خدمات الأمن على مستوى التطبيقات في APIs قياسية .إن دمج منصة IBM لمئات المكتبات الخارجية، والمفتوحة المصدر يسمح بالنمذجة، والاختبار، والنشر بشكل سريع للتطبيقات الجديدة .

تدعم منصة IBM Bluemix لغات برمجة متنوعة من مثل :java، Node.js ، PHP، Python ولغات برمجية أخرى. تسمح منصة IBM Bluemix بالوصول إلى أدوت و خدمات IBM،وهذا يتضمن IBM Watson ووظائف IBM Cloud فيما يخص الحوسبة بلا خادم ، إضافة إلى وظائف خارجية من قبل البائعين . [24]

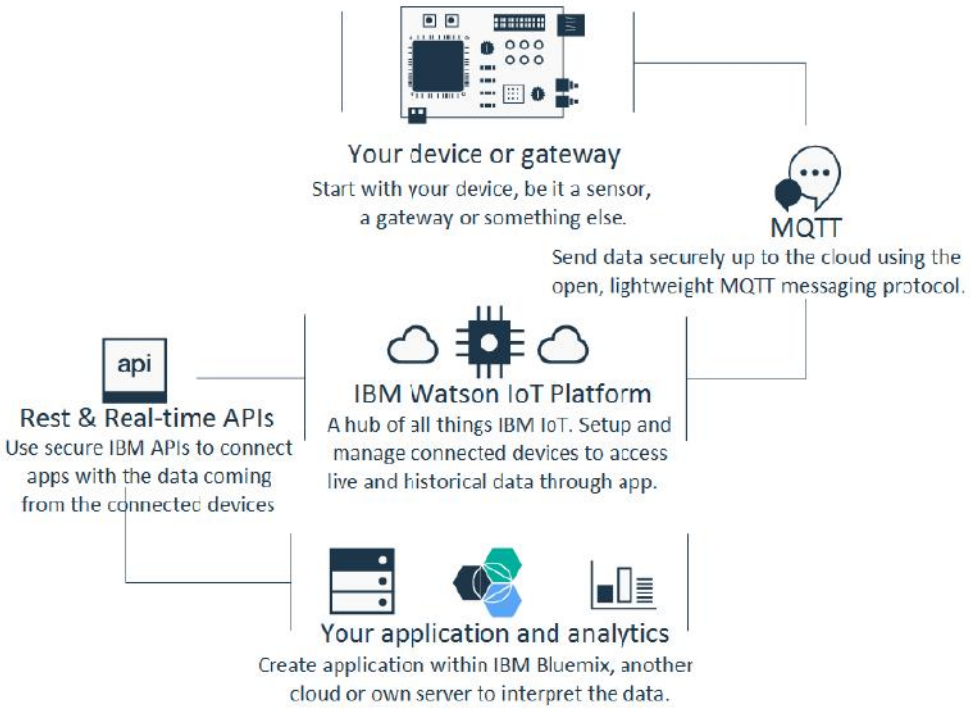
منصة IBM Watson: تكمّل منصة IBM Bluemix التي تعدّ الرائد في التكنولوجيا الإدراكية. تقدم منصة Watson التحليلية سياق إدراكي متنبأ به بالزمن الحقيقي لبيانات انترنت الأشياء، وذلك من خلال استخدام آخر التطورات في مجال التعلم الآلي، والذكاء الاصطناعي. تؤمن هذه المنصة محرك تحليلات، وقدرات تحليلية، تسمح بإدارة بيانات أجهزة انترنت الأشياء، وتسريع فهم الشروط، وتطوير اتخاذ القرار[24].

تقدم منصة IBM Bluemix أكثر من 170 خدمة نذكر منها [25]:

- الحوسبة Compute : تقدم مختلف مصادر الحوسبة من مثل: المخدمات الافتراضية، والحوسبة دون مخدمات بحيث يمكن للشركات استضافة أعمالهم .
- الشبكة Network : تقدم خدمات شبكات سحابية من مثل: موازن الأحمال load balancer، وشبكة إيصال المحتوى Content Delivery Network(CDN)، وقنوات الشبكة الخاصة الافتراضية (VPN)، وجدران الحماية.
- التخزين Storage : تقدم تخزين لبيانات السحابة .
- الإدارة Management : توفر الأدوات لإدارة ومراقبة عمليات النشر في السحابة من مثل: تحليلات السجل، والأتمتة، والبنية التحتية كرمز Infrastructure as Code(IaC).
- الحماية Security: تتضمن خدمات تتبع النشاط، وإدارة الهوية، والوصول،والمصادقة .
- إدارة البيانات Data management : تؤمن قواعد بيانات SQL و NoSQL إضافة إلى أدوات الاستعلام عن البيانات و الهجرة .

- التحليل Analytics :تقدم أدوات علوم البيانات من مثل:Apache Spark و Apache Hadoop و IBM Watson Machine Learningإضافة إلى خدمات التحليل لتدفق البيانات .
- الذكاء الاصطناعي (Artificial Intelligence (AI: **يستخدم** IBM Watson لإيصال خدمات من مثل: التعلم الآلي، ومعالجة اللغات، والتمييز البصري .
- انترنت الأشياء (Internet of things (IoT : **تتضمن منصة** IBM لانترنت الأشياء ، تؤمن خدمات الاتصال، والإدارة لأجهزة انترنت الأشياء، وتحلل البيانات التي تنتجها هذه الأجهزة .
- الهاتف المحمول Mobile : تسمح لفريق مطورين من بناء ومراقبة تطبيقات Mobileالخاصة بهم .

تسمح منصة Watson IoT للتطبيقات بالاتصال بالأجهزة، والبوابات بواسطة برتوكول خفيف هو برتوكول (MQTT)Message Queue Telemetry Protocol الذي ذكرناه سابقا في فصل البنية المعمارية ومنصات انترنت الأشياء، و يعتمد على البنية Publish/Subscribe في تحقيق هذا الاتصال .



الشكل(3–17) منصة IBM Watson IoT

إنّ(REST) Representational State Transfer النقل التمثيلي للحالة هو أسلوب معماري لبناء التطبيقات الشبكية، وتستخدم هذه التطبيقات HTTP requests لإنشاء / قراءة / تحديث / حذف البيانات. باستعمال REST وتطبيقات الزمن الحقيقي يمكن لبيانات الحساسات أن ترسل إلى التطبيق .

يوضح الشكل(3-17) منصة Watson IoT وطريقة اتصال المنصة بالأجهزة، والتطبيقات باستعمال برتوكول MQTT و REST .

ينبغي أن تكون الأجهزة مسجلة على منصة IBM Bluemix قبل إنشاء اتصال MQTT مع MQTT Broker الخاص بمنصة Watson IoT وبعد تحقيق اتصال ناجح تقوم منصة Watson باستقبال بيانات الأجهزة، ومن ثم إرسالها إلى التطبيق المنشأ على منصة IBM .

3-3-4 خطوات استخدام منصتي IBM Bluemix و IBM Watson IoT :

بعد التعرف على منصتي IBM Bluemix و IBM Watson IoT والخدمات التي تقدمها ينبغي علينا إنشاء حساب على منصة IBM Bluemix وإضافة الأجهزة المراد مراقبتها وأخذ البيانات منها إلى منصة IBM Watson IoT. تم شرح خطوات استخدام منصتي IBM Bluemix و IBM Watson IoT بالتفصيل ضمن الملحقِ A .

3-3-5 إضافة أجهزة إلى منصة IBM Watson IoT :

قبل إضافة الأجهزة إلى المنصة ينبغي علينا إضافة أنواع إلى هذه الأجهزة . والغرض من تعريف نوع الجهاز عند إضافته، هو إمكانية إرسال أوامر إلى الأجهزة من نوع واحد أو استقبال حالة الأجهزة ذات النوع الواحد، وهذا يعطي سلاسة في التحكم والمراقبة.

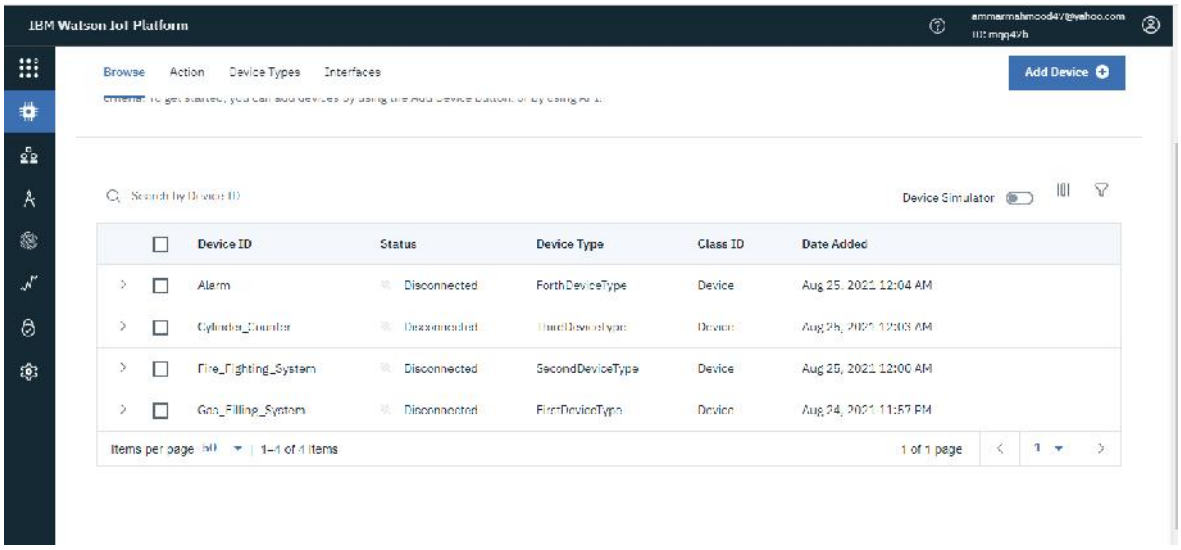
عند إضافة جهاز، نحصل على أربع بارامترات ضرورية للوصول إلى هذا الجهاز من خلال المنصة وهذه البارامترات هي :

- Organization ID : تمثل معرف حساب IBM الخاص بالمستخدم .
- Device Type : يمثل هذا المتغير نوع الجهاز، وكما ذكرنا سابقاً يساعد في تحديد الأجهزة أو التطبيقات المراد التعامل معها .
- Device ID : المعرف الخاص بالجهاز وبالنتيجة إمكانية الوصول إلى هذا الجهاز من خلال معرفه .
- Authentication Method : طريقة المصادقة من خلال token وهي الطريقة الوحيدة المتاحة .
- Authentication Token : لكل جهاز معرف على المنصة رمز Token خاص به ينشأ عند إضافته.

يظهر الشكل(3-18) الأجهزة المعرفة على المنصة الخاصة ببحثنا وهي أربعة أجهزة :

- السير الناقل ووحدة التعبئة الآلية (نظام التعبئة)Gas_Filling_System .
- نظام مكافحة الحريقFire_Fighting_System .

- جهاز عدّ الأسطوانات Cylinder_Counter.
- جهاز الإنذار Alarm .



الشكل(3–18) الأجهزة المعرفة على منصة IBM Watson IoT

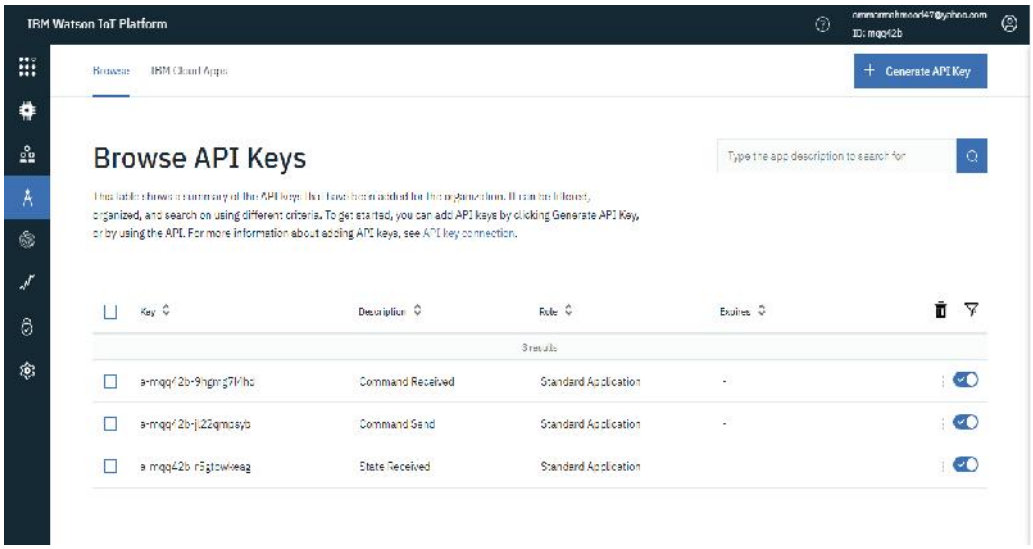
3-3-6 إضافة تطبيقات إلى منصة IBM Watson IoT :

في حالة العمل بنظام M2M يتم استقبال، وإرسال البيانات، والأوامر بين الأجهزة، والمنصة بشكل مباشر دون الحاجة إلى وجود تطبيقات وسيطة بينها. ولكن في حالة العمل بين تطبيق زبون وأجهزة، فإننا بحاجة إلى وجود تطبيقات معرفة على المنصة، تمكّن تطبيق الزبون من الحصول على حالة الأجهزة المتصلة بالمنصة،وبياناته،ا وإرسال الأوامر من تطبيق الزبون إلى هذه الأجهزة .

كل تطبيق معرف على المنصة يملك Api Key خاص به لذلك ينبغي علينا توليد Api Key قبل إضافة أي تطبيق، والغرض من هذا البارامتر الخصوصية، والمصادقة .

يظهر الشكل(3–19) التطبيقات المعرفة على المنصة التي استخدمناها في دراستنا لإرسال واستقبال الأوامر والبيانات إلى تجهيزات معمل تعبئة الغاز المنزلي ،هي :

- State Received:مهمته استقبال حالة تجهيزات المعمل (الأجهزة المعرفة على المنصة).
- Command Send : مهمته إرسال أوامر التحكم في تجهيزات المعمل إلى المنصة .
- Command Received : كون تجهيزات المعمل افتراضية تم تمثيلها على صفحة ويب فإننا بحاجة إلى تطبيق وسيط بين المنصة، والمحاكي لاستلام أوامر التحكم بنمذجة المعمل من واجهة التحكم الخارجية، وهذه وظيفة تطبيق Command Received .



الشكل(3–19) التطبيقات المعرفة على منصة IBM Watson IoT

الفصل الرابع

الاختبارات و النتائج

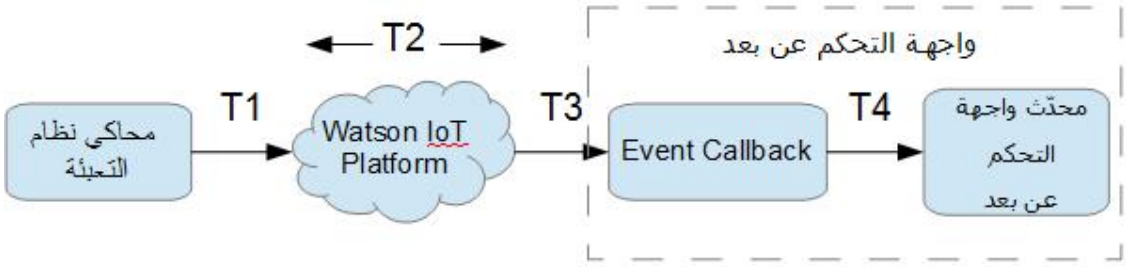
إنّ تطبيق تقنية انترنت الأشياء في بيئات العمل التي تتواجد فيها الأجهزة المستخدمة لهذه التقنية، تعتمد بشكل أساسي على توافر خدمة الانترنت، وجودة هذه الخدمة . تحدد جودة خدمة الانترنت بعوامل عدة أهمها عرض الحزمة (Bandwidth) و Ping Time .

- عرض الحزمة (Bandwidth) : تعبر عن أقصى كمية للبيانات يمكن أن ترسل عبر اتصال انترنت في واحدة الزمن وتقاس بـ Mbps .
- Ping Time : يعبر عن الزمن الذي يستغرقه إرسال رزمة بيانات إلى السيرفر، و عودتها إلى المستخدم، و يقاس بالميللي ثانية .

وبالنتيجة كلما زاد عرض الحزمة، ونقص زمن Ping،كلما زادت جودة خدمة الانترنت[30]. باعتبار أن جودة خدمة الانترنت عامل أساسي في تقنية انترنت الأشياء، سنعمد إلى إجراء عدد من الاختبارات تظهر استجابة واجهة التحكم عن بعد[2] لبيانات محاكي نظام تعبئة الغاز المنزلي عند سرعات انترنت مختلفة.

4-1 التحليل الرياضي لزمن استجابة واجهة التحكم عن بعد :

يمكن تقسيم زمن استجابة واجهة التحكم عن بعد لبيانات محاكي نظام التعبئة إلى أربع متغيرات زمنية كما هو موضح في الشكل(4-1) وهي :



الشكل(4-1) المراحل الزمنية لاستجابة واجهة التحكم عن بعد لبيانات محاكي نظام التعبئة

- T1 : هو الزمن اللازم لانتقال بيانات محاكي نظام التعبئة إلى منصة انترنت الأشياء، تعتمد قيمة هذا المتغير على سرعة خط الانترنت، وحالة وصلة الانترنت .

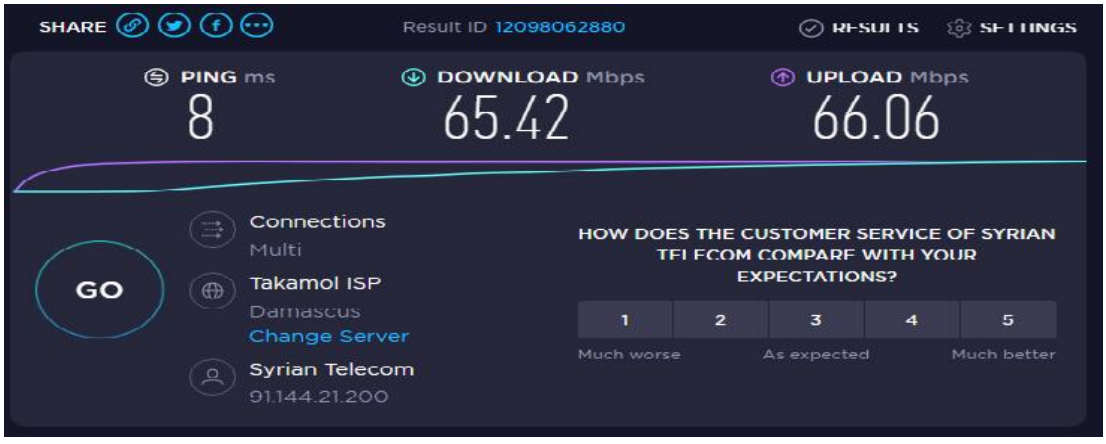
[2] الفرق بين لحظة إرسال البيانات من حساسات معمل تعبئة الغاز المنزلي و لحظة استقبال واجهة التحكم عن بعد لهذه البيانات دون الأخذ بعين الاعتبار الزمن اللازم لتحديث بيانات واجهة التحكم عن بعد .

- T2 : هو زمن المعالجة الذي تحتاجه منصة الـ Watson IoT لإعداد بيانات المحاكي المستقبلة للنشر باتجاه واجهة التحكم عن بعد، إن قيمة هذا المتغير صغيرة مقارنة مع الأزمنة الأخرى، ويمكن إهمالها.
- T3 : هو الزمن اللازم لانتقال بيانات المحاكي الجاهزة للنشر من منصة الـ Watson IoT إلى مستقبل الأحداث Event Callback المضمن في واجهة التحكم عن بعد .
- T4 : هو الزمن اللازم لتحديث بيانات واجهة التحكم عن بعد. إنّ قيمة هذا المتغير في دراستنا هذه تعتمد على زمن المؤقت المستخدم في تحديث بيانات واجهة التحكم عن بعد و تتراوح بين 1 – 100ميللي ثانية .

4-2 سيناريو الاختبار :

كما ذكرنا سابقاً، إنّ زمن استجابة واجهة التحكم عن بعد لبيانات محاكي نظام التعبئة، عبارة عن مجموع الأزمنة الأربعة التي تستغرقها بيانات المحاكي لتصل إلى واجهة التحكم عن بعد، وعرضها للمستخدم. إنّ ما يهم في هذه الدراسة معرفة الزمن الكلي لهذه الاستجابة عند سرعات مختلفة لشبكة الانترنت للوصول إلى نتائج، تظهر سرعة الانترنت الواجب استخدامها للحصول على زمن الاستجابة المطلوب، لذلك سنكتفي بقياس هذا الزمن دون قياس الأزمنة الجزئية الأربعة المكونة له .

بداية ينبغي علينا معرفة سرعة، وحالة شبكة الانترنت بين منصة IoT وكل من واجهة التحكم عن بعد ومحاكي نظام التعبئة. تتوفر أدوات عديدة لمعرفة سرعة الانترنت، منها: خدمة اختبار سرعة الانترنت من قبل Ookla؛ وهي شركة رائدة عالمياً في اختبار جودة خدمة الانترنت والبيانات، والتحليل من خلال الموقع https://www.speedtest.net [31].



الشكل(4-2) اختبار الـ Ookla لمعرفة السرعة الحالية لشبكة الانترنت المستخدمة من قبل محاكي نظام التعبئة وواجهة التحكم عن بعد

عند إجراء اختبار Ookla لمعرفة السرعة الحالية لشبكة الانترنت المستخدمة من قبل محاكي نظام التعبئة، وواجهة التحكم عن بعد ،التي ستعدّ السرعة العظمى (دون استخدام محدد سرعة) خلال عملية الاختبار حصلنا على النتائج الموضحة في الشكل(4–2).

باستخدام برنامج الـ SelfishNet – الذي سبق وتحدثنا عنه عند ذكر الأدوات المستخدمة في عملية النمذجة وتصميم نظام التعبئة – استطعنا التحكم بسرعة شبكة الانترنت المستخدمة من قبل نظام المحاكي وواجهة التحكم عن بعد (سرعة التحميل Download والرفع إلى المخدّم Upload) وبالنتيجة معرفة تأثير جودة خدمة الانترنت على استجابة نظام التعبئة وواجهة التحكم عن بعد .

تم قياس زمن الاستجابة الوسطية عن طريق قياس الزمن الذي يستغرقه إرسال بيانات حالة نظام التعبئة إلى واجهة التحكم عن بعد عشر مرات متتالية، وحساب المتوسط الحسابي للقيم المقاسة، تمّ تكرار هذه العملية عدد من المرات للأخذ بعين الاعتبار حالة وصلة الانترنت .

4–3 مناقشة النتائج :

تم استخدام سرعات مختلفة لوصلة الانترنت المستخدمة من قبل محاكي نظام التعبئة وواجهة التحكم عن بعد بواسطة برنامج SelfishNet كما هو موضح في الجدول(4–1):

الجدول(4–1) سيناريوهات العمل بناءً على سرعات الانترنت المحددة بواسطة برنامج SelfishNet

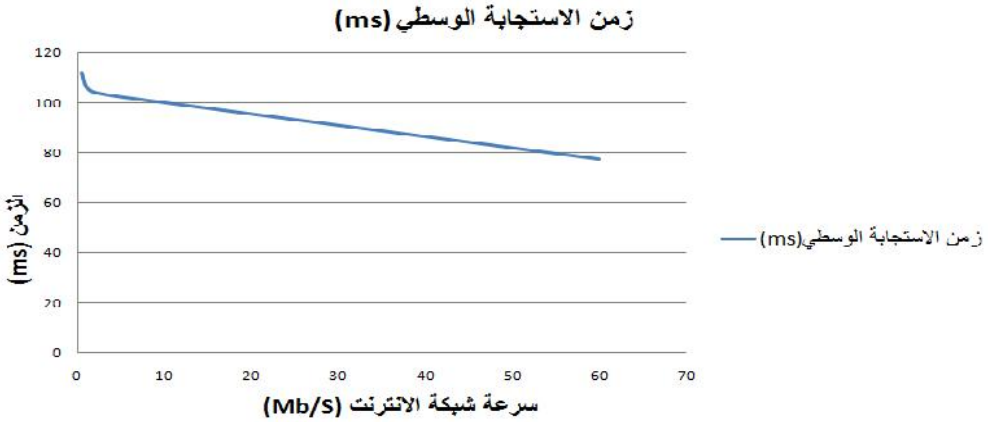
رقم السيناريو	سرعة شبكة الانترنت	قيم برنامج SelfishNet	
		Download	Upload
1	512 Kb/s	64 KB/s	64 KB/s
2	1 Mb/s	128 KB/s	128 KB/s
3	2 Mb/s	256 KB/s	256 KB/s
4	5 Mb/s	640 KB/s	640 KB/s
5	Max Speed(65Mb/s)	-	-

و كانت الاستجابات الوسطية المقاسة بناءً على السيناريوهات السابقة كما هي موضحة في الجدول(4–2):

الجدول(4-2) الاستجابات الوسطية المقاسة لكل سيناريو عمل

	الاستجابة الوسطية لكل عملية (ms)										الاستجابة الوسطية الكلية لكل سيناريو (ms)
رقم العملية	1	2	3	4	5	6	7	8	9	10	
السيناريو الأول	111	106	106	105	113	137	120	113	114	97	112.2
السيناريو الثاني	124	105	90	111	108	95	111	99	108	116	106.7
السيناريو الثالث	87	118	89	84	100	128	108	87	100	142	104.3
السيناريو الرابع	110	85	101	107	109	113	99	111	101	89	102.5
السيناريو الخامس	77	87	76	65	74	73	91	74	73	86	77.6

يمكن تمثيل الاستجابة الوسطية الكلية الموافقة لكل سيناريو بالمخطط البياني الموضح في الشكل(4-3) :



الشكل(4-3) المخطط البياني للاستجابة الوسطية لواجهة التحكم عن بعد لبيانات محاكي نظام التعبئة عند سرعات انترنت مختلفة

نلاحظ من النتائج السابقة أنّ أزمنة الاستجابة عند سرعات انترنت تتراوح بين 5Mb/s-512Kb/s مقبولة في تطبيقات انترنت أشياء لا تتطلب التحكم الحرج والآني، ولكن في تطبيقات حساسة كمراقبة معمل تعبئة الغاز من الواجب استخدام سرعات انترنت عالية لا تقل عن 50Mb/s للتقليل قدر الإمكان من المخاطر التي يمكن أن تحدث في مثل هذه المنشآت.

نستنتج مما سبق أنّ تقنية انترنت الأشياء قادرة على إيصال بيانات الأنظمة المستخدمة لهذه التقنية، وتحليلها بأزمنة قصيرة، وكفاءة عالية، ولكن هذا يتطلب تقديم إمكانيات محددة لها تتناسب مع أهمية المنشآت المتحكم فيها، وسرعة الاستجابة المطلوبة لهذه الأنظمة، وكمية البيانات المتبادلة .

الفصل الخامس
الخاتمة والأعمال المستقبلية

5-1 الخاتمة:

إنّ استخدام تقنية انترنت الأشياء في منشآت حيوية كمنشآت تعبئة الغاز، ومنشآت تخزين المشتقات النفطية – التي يمكن أن تتواجد في أماكن جغرافية متباعدة عن بعضها – يعدّ حلاً لمشكلات السلامة البشرية، والبيئية المحتمل حدوثها في أي لحظة في مثل هذه المنشآت. كما يمكن لهذه التقنية أن تسهم بشكل فعال في إيصال البيانات الضرورية للشخص المسؤول بالزمن الحقيقي، وبالنتيجة اتخاذ الإجراءات المناسبة بناء على هذه البيانات .

تمكنّا في هذا البحث من تصميم محاكي، يحاكي عملية تعبئة الغاز في معمل تعبئة الغاز المنزلي الآلي باستخدام تقنيات الويب المختلفة، والتحكم بأهم التجهيزات المستخدمة في عملية التعبئة، ومراقبتها عن بعد باستخدام تقنية انترنت الأشياء، كما توصلنا لنتائج جيدة فيما يتعلق بزمن استجابة واجهة التحكم عن بعد لبيانات عملية التعبئة ضمن ظروف تشغيل معينة، حيث تراوحت أزمنة الاستجابة بين77ms – 112 ms وهو زمن قصير مقارنة مع التحكم التقليدي المعتمد على نقل المعلومة بحسب التسلسل الإداري. أسهمت منصة **Watson IoT platform** التي تعدّ أحد خدمات منصة **IBM BlueMix** في تقديم تجربة مجانية محدودة سمحت لنا بإضافة الأجهزة المراد التحكم فيها، ومراقبتها عن بعد، وبالنتيجة توفير الوقت، والكلفة اللازمين لتصميم منصة انترنت أشياء .

5-2 الاستنتاجات :

لا يمكن إنكار الميزات الرائعة التي تقدمها تقنية انترنت الأشياء في مختلف مجالات الحياة، من حيث زيادة الإنتاجية، وسهولة التحكم، وآنية البيانات، واختصار المسافات، والزمن، وغيرها من الخصائص لكن شأنها شأن جميع التقانات الأخرى، تمتلك سلبيات عدة ولاسيما فيما يخص مشكلات الأمن، وسلامة البيانات، والمساهمة في رفع مستويات البطالة، مما يجعل استخدم هذه التقنية مرتكزاً على إمكانية معالجة هذه المشكلات، أو التخفيف منها قدر الإمكان .

في الختام نستطيع القول، إنه كما كان ظهور تقنية انترنت الأشياء نتيجة لانتشار تقنية الانترنت، وسيطرتها على مختلف مجالات حياتنا سلباً أو إيجاباً، فقد تكون تقنية انترنت الأشياء بداية لظهور تقنيات أخرى ستؤثر على حياتنا سلباً أو إيجاباً، ونأمل الخير دائماً .

3-5 الأعمال المستقبلية :

نأمل في الدراسات اللاحقة أن نطبّق تقنية انترنت الأشياء على أرض الواقع في معمل تعبئة الغاز المنزلي، وليس فقط بواسطة نمذجة محاكي؛ لأنه مهما كانت المحاكاة دقيقةً، وقريبةً من الواقع، تبقى هناك عوامل ثانوية غير مأخوذة بعين الاعتبار قد يكون لها دور مؤثر في عمل النظام ككل.كما نطمح إلى جعل التحكم عن بعد ممكناً في الهواتف الذكية عبر تطوير تطبيق أندرويد، ليعمل عمل واجهة التحكم عن بعد، بما يحقق سهولة الوصول، والاستخدام. وهذا يتطلب المزيد من الاعتبارات الأمنية المرتبطة بأهمية المنشأة المتحكم فيها، لذلك لابد من الاهتمام بالمشكلات الأمنية، وسلامة البيانات كأحد المواضيع الهامة في الدراسات القادمة.

المراجع

[1]Shanzhi Chen, Hui Xu ,eta… ," A Vision of IoT: Applications, Challenges, andOpportunities With China Perspective" , IEEE INTERNET OF THINGS JOURNAL, VOL. 1, NO. 4, AUGUST 2014 .

[2]Emrah Irmak , Mehmet Bozdal , "Internet of Things (IoT): The Most Up-To-Date Challenges, Architectures, Emerging Trends and Potential Opportunities" , International Journal of Computer Applications (0975 – 8887) Volume 179 – No.40, May 2018.

[3]SomayyaMadakam, R. Ramaswamy, Siddharth Tripathi ," Internet of Things (IoT): A LiteratureReview", Journal of Computer and Communications,May 2015.

[4]Tran Anh Khoa , Cao Hoang Phuc,eta…, "Waste Management System Using IoT-Based MachineLearning in University",HindawiWireless Communications and Mobile ComputingVolume 2020.

[5]Heetae Yang, Wonji Lee, and HwansooLee ,"IoT Smart Home Adoption: The Importance of ProperLevel Automation",HindawiJournal of Sensors2018.

[6] Ahmed Abdelgawad, Kumar Yelamarthi ,"Internet of Things (IoT) Platform for Structure Health Monitoring",HindawiWireless Communications and Mobile Computing2017.

[7] GeertjanWielenga,"Beginning NetBeans IDEfor Java Developers",2015.

[8] Matt Lombard," SolidWorks® 2009 Bible", 2009.

[9] Shalini Verma , Deepak Sharma,"A REVIEW PAPER ON BASIC WORKSPACE OF ADOBE PHOTOSHOP",International Journal of Current Research Vol. 11,August, 2019.

[10] Movavi," Movavi Video EditorUser's Guide, manualzz, available :https://manualzz.com/doc/6932175/movavi-video-suite-user-s-manual ,last access:2021.

[11] M.Loganathan, K.E.Dharani," ADVANCED WEB BROWSER WITH EFFECTIVE TOOLS", International Journal of Computer Science and Mobile ComputingVol.3 Issue.2, February- 2014.

[12] Soumyalatha, Shruti G Hegde," Study of IoT: Understanding IoT Architecture, Applications, Issues and Challenges", International Journal of Advanced Networking & Applications, May 2016.

[13] AntarShaddad Abdul-Qawy, Pramod P. J,eta,"The Internet of Things (IoT): An Overview",Journal of Engineering Research and Applications , Vol. 5, December 2015.

[14] Bhumi Nakhuva , Prof. Tushar Champaneria," STUDY OF VARIOUS INTERNET OF THINGS PLATFORMS",International Journal of Computer Science & Engineering Survey (IJCSES) Vol.6, December 2015.

[15] Mohammad Shimul Hossain*, Mohammad Golam MuntasirShehab,eta…"Liquefied Petroleum Gas (LPG) Bottling Process and Required Safety During Bottling: A Case Study",Petroleum Science and Engineering, 2019.

[16] Kosan Krisplant Group," Systems, equipment & services for the LPG industry", Kosan KrisplantGroup ,available :http://www.kosancrisplant.com/en/home/products/general-product-catalogue , last access 2021.

[17] Diana - Iuliana BOBOC, Ştefania - Corina CEBUC," Internet of Things (IoT)",Database Systems Journal, 2019.

[18] Guy W. Lecky-Thompson," Just EnoughWeb Programmingwith XHTML, PHP, and MySQL",2008.

[19] Francisco Javier Díaz, Claudia A. Queiruga, Pablo José Iuliano"a solution to build privacy JSF mobile applications",January 2010.

[20] DimanSalih,"Comparison of Java web application frameworks", February 2019.

[21]Pat Niemeyer,Jonathan Knudsen,” Learning Java”,May 2000.
[22] Richard York,” Beginning JavaScript® and CSSDevelopment withjQuery”,2009.
[23] Rishi Kashyap, Palak Bansal,eta…,” Architecture, Features and Security Concern of IoT”,Advances in Wireless and Mobile Communications,Volume 10, Number 5 (2017).
[24] Andrew Bickley,” IoT Solutions from Arrowand IBM”, Arrow Central Europe GmbH, May 10, 2017.
[25] Jit Chatterjee, Tushar Debnath,” Environmental Monitoring Using Sense HAT based on IBM Watson IoTPlatform”, International Research Journal of Engineering and Technology (IRJET) Volume: 05 ,July 2018.
[26] Pawan Kumar,”Internet of Things (IoT) Development Platforms – A Case of IBM Bluemix”,University of Koblenz-Landau, September 25, 2016.
[27] Sanjay Aiyagari ,Cisco Systems ,Alexis Richardson,eta…,” Advanced Message Queuing Protocol”, Version 0-9-1, 13 November 2008.
Protocol Specification
[28] Mirona Popescu, Dan Dumitriu, Markus Helfert,” Adoption, Architecture and Technology of Enterprise IoT Systems – Towards a framework of concerns in IoT environments”, AICS 2017.
[29] Z. Z. AbouSalem,M. A. Ashabrawy,” Compared Between Ipv6 And With Ipv4,Differences And Similarities”, International Journal of Computers & Technology, Volume: 17, 29-10-2018.
[30]Yu-Hsin Liu, Jeffrey Prince, Scott Wallsten,” Distinguishing Bandwidth and Latency in Households’ Willingness-to-Pay for Broadband Internet Speed”, August 2017.
[31] Ookla,”speed test”,Ookla ,available :https://www.speedtest.net/about ,last access :2021.
[32] selfihnet,” download-selfishnet-for-windows”,wazaps,available : https://en.wazaps.com/windows/download-selfishnet-for-windows, last access :2021.

الملاحق :

الملحق A :

إنشاء حساب على منصةIBM Watson IoT :

لإنشاء حساب على منصة انترنت الأشياء نتبع ما يلي :

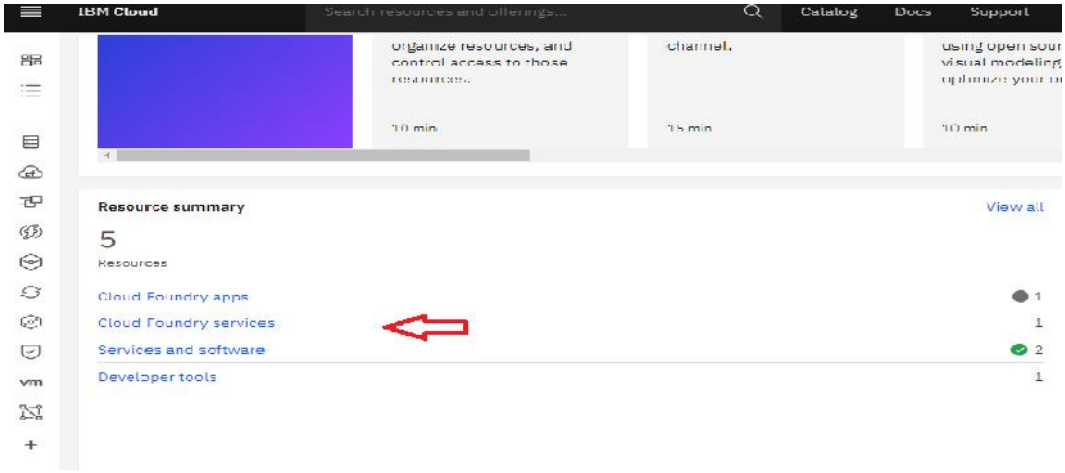
- بداية نقوم بالدخول إلى موقع https://cloud.ibm.com حيث تعرض هذه الصفحة الخدمات التي تقدمها المنصة و تتيح لنا إنشاء حساب على منصةIBM Bluemix **أو الدخول إلى حساب** IBM **إن وجد** .
- إن إنشاء حساب **على منصة**IBM Bluemix يتطلب إدخال Email خاص بك و كلمة مرور يجب حفظها.
- بعد تسجيل الدخول إلى حساب IBM تظهر لنا صفحة تتيح لنا الوصول إلى الخدمات التي تقدمها منصة IBM Bluemix من خلال الخيار Create Resource.
- نحتاج من هذه الخدمات خدمة Internet of Things Platform والتي تتيح لنا إنشاء خدمة انترنت الأشياء على منصة IBM Watson IoT .

إنّ العمل على منصة IBM Watson IoT يتطلب تحديد خطة (Plan) للمنصة، كل خطة تعطي المستخدم إمكانيات معينة كالتالي :

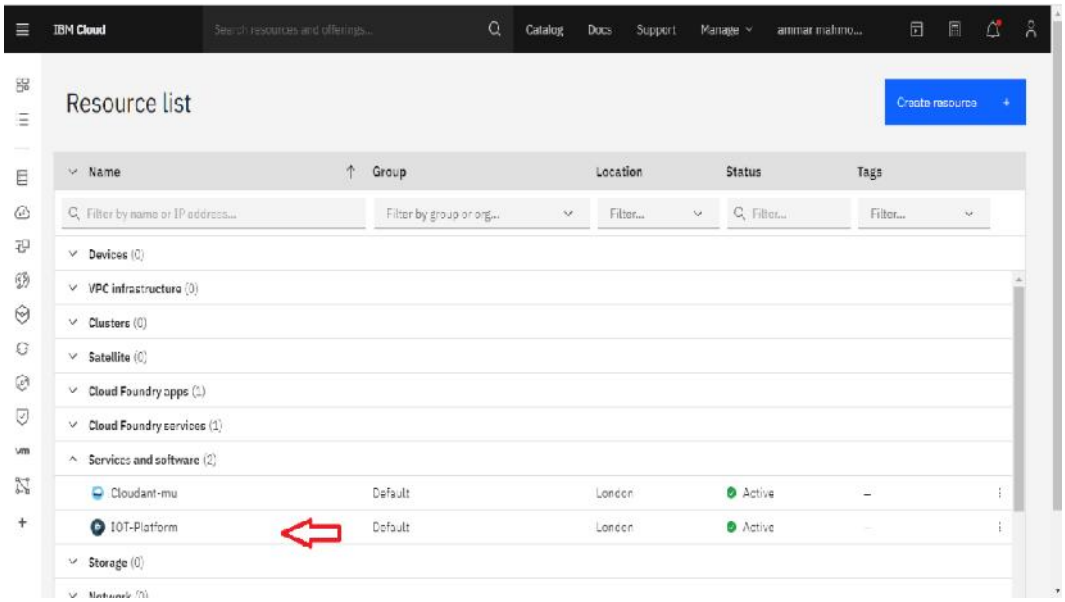
- الخطة Lite : وهي خطة مجانية تتيح استخدام 500 جهاز مسجل و500 تطبيق مسجل و200 MB كحد أقصى للبيانات المتبادلة والبيانات التي يتم تحليلها و وبيانات الحافة التي يتم تحليلها شهرياً و تحذف الخطة بعد 30 يوم في حال لم يتم استخدام الخطة خلال هذه الفترة . سنعتمد هذه الخطة في بحثنا هذا باعتبارها كافية لإنجاز المطلوب .
- الخطة Watson IoT Platform Connection and Analytics Service Capacity Level 1 : تقدم منصة IoT جاهزة للتشغيل ومتكاملة ذات خدمة من نوع(SaaS) Software as a service مع قدرات اتصال وإدارة بيانات وتحليلات متقدمة وهي خطة مدفوعة تسمح باستخدام 5000 جهاز مسجل و مليون حدث تحليلي .
- الخطة Watson IoT Platform Connection and Analytics Service Capacity Level2 : تقدم منصة IoT جاهزة للتشغيل ومتكاملة ذات خدمة من نوع SaaS مع قدرات اتصال وإدارة بيانات وتحليلات متقدمة وهي خطة مدفوعة تسمح باستخدام 25000 جهاز مسجل و5 مليون حدث تحليلي .

بعد تحديد الخطة المناسبة نقوم بإنشاء خدمة انترنت الأشياء وهذا يسمح لنا بالوصول إلى الخدمات التي تقدمها منصة IBM Watson .

يظهر الشكل –1– الصفحة الرئيسية لحساب المستخدم على منصة IBMBluemix بعد إنشاء خدمة انترنت الأشياء، عند الضغط على الخيار Services and Software ستظهر لدينا صفحة بالخدمات النشطة كما في الشكل–2– حيث تظهر خدمتان فعالتان هما cloudant-mu و Iot Platform .



الشكل 1 الصفحة الرئيسية لحساب المستخدم على منصة IBMBluemix بعد إنشاء خدمة انترنت الأشياء



الشكل 2 الخدمات النشطة على منصة IBM Bluemix

تسمح خدمة Cloudant-mu بإنشاء قاعدة بيانات على المنصة و في بحثنا لم نستخدم هذه الخدمة كوننا اعتمدنا على البيانات المرسلة والمستقبلة بالزمن الحقيقي ولم نعتمد على البيانات المخزنة في قاعدة بيانات ، في حين تسمح خدمة IOT-Platform بالوصول إلى منصة IBM Watson وهناك ستتم إضافة الأجهزة والتطبيقات التي سنأخذ البيانات منها و نقوم بمراقبتها و التحكم بها .

الملحق B :

الكود البرمجي لتهيئة أجهزة محاكي معمل التعبئة (StateOutputTimer.java) :

يتم في هذا الكود الاتصال بمنصة IBM Watson IoT وربط محاكي معمل التعبئة (الأجهزة المراد التحكم بها) مع الأجهزة المعرفة على المنصة ومن ثمّ إرسال بيانات محاكي معمل التعبئة إلى منصة انترنت الأشياء بشكل مستمر باستخدام مؤقت زمني.

```
package com.mycompany.factorycontrolsite;
import com.google.gson.JsonObject;
import com.ibm.iotf.client.device.DeviceClient;
import java.io.Serializable;
import java.time.Instant;
import java.util.Properties;
import java.util.Timer;
import java.util.TimerTask;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.enterprise.context.SessionScoped;
import javax.inject.Named;
@Named(value = "StateOutputTimer")
@SessionScoped
public class StateOutputTimer implements Serializable{
   int i=0;
   int period=2000;
   String productstate=null;
   String firestate=null;
   String alarm=null;
   int filledCylCount=-1;
   Properties options = new Properties();
DeviceClient ProductLineDevice = null;
DeviceClient FireSystemDevice = null;
DeviceClient CylCounterDevice = null;
DeviceClient fireSensorDevice = null;
JsonObject event1= new JsonObject();
JsonObject event2 = new JsonObject();
JsonObject event3= new JsonObject();
JsonObject event4= new JsonObject();
   public StateOutputTimer(){
      //product-line
options.setProperty("org", "mqq42b");
```

```
options.setProperty("type", "FirstDeviceType");
options.setProperty("id", "Gas_Filling_System");
options.setProperty("auth-method", "token");
options.setProperty("auth-token", "uFJ_0F2mWxecr&rj7+");
      try {
ProductLineDevice = new DeviceClient(options);
ProductLineDevice.connect();
      }
      catch (Exception ex) {
         Logger.getLogger(StateOutputTimer.class.getName()).log(Level.SEVERE, null, ex);
      }
      //fire-system
options.setProperty("type", "SecondDeviceType");
options.setProperty("id", "Fire_Fighting_System");
options.setProperty("auth-method", "token");
options.setProperty("auth-token", "TAtNZK&ilLU0IV-H&h");
      try {
FireSystemDevice = new DeviceClient(options);
FireSystemDevice.connect();
      }
      catch (Exception ex) {
         Logger.getLogger(StateOutputTimer.class.getName()).log(Level.SEVERE, null, ex);
      }
      //cylinder counter
options.setProperty("org", "mqq42b");
options.setProperty("type", "ThirdDeviceType");
options.setProperty("id", "Cylinder_Counter");
options.setProperty("auth-method", "token");
options.setProperty("auth-token", "Yi5z_1f8qJkRbQ8CK*");
      try {
CylCounterDevice = new DeviceClient(options);
CylCounterDevice.connect();
      }
      catch (Exception ex) {
         Logger.getLogger(StateOutputTimer.class.getName()).log(Level.SEVERE, null, ex);
      }
      //fire sensor and alarm
options.setProperty("org", "mqq42b");
options.setProperty("type", "ForthDeviceType");
options.setProperty("id", "Alarm");
options.setProperty("auth-method", "token");
```

```
options.setProperty("auth-token", ")r5YYe0u5UZq*Z0wOH");
        try {
fireSensorDevice = new DeviceClient(options);
fireSensorDevice.connect();
        }
        catch (Exception ex) {
            Logger.getLogger(StateOutputTimer.class.getName()).log(Level.SEVERE, null, ex);
        }
        }
    class timerfunc extends TimerTask {
private  String state;
private  String device;
timerfunc( String state,String device )
        {
this.state = state;
this.device = device;
        }
        @Override
        public void run() {
            Instant start=null;
            if(device.equals("START")){
productstate=state;
firestate=state;
                alarm=state;
            }
            else if(device.equals("product-line")){
productstate=state;
            }
            else if(device.equals("fire-system")){
                device="";
                if(firestate.equals("ON"))
firestate="OFF";
                else
firestate="ON";
            }
            else if(device.equals("fireSensor_And_Alarm")){
                device="";
                if(alarm.equals("ON")){
                  alarm="OFF";
                }
                else
```

```
            alarm="ON";
        }
        event1.addProperty("device","product-line");
        event1.addProperty("product_line",productstate);
        event2.addProperty("device","fire-system");
        event2.addProperty("fire_system",firestate);
ProductLineDevice.publishEvent("status", event1);
        try {
Thread.sleep(500);
        } catch (InterruptedException ex) {
            Logger.getLogger(StateOutputTimer.class.getName()).log(Level.SEVERE, null, ex);
        }
FireSystemDevice.publishEvent("status", event2);
        try {
Thread.sleep(500);
        } catch (InterruptedException ex) {
            Logger.getLogger(StateOutputTimer.class.getName()).log(Level.SEVERE, null, ex);
        }
        event4.addProperty("device","fireSensor_And_Alarm");
        event4.addProperty("Alarm",alarm);
fireSensorDevice.publishEvent("status", event4);
        }
    }


    Timer mytimer1=new Timer();
    public void order(String state,String device) throws InterruptedException{
        if(i==1){
        mytimer1.cancel();
        //mytimer.purge();
        mytimer1=new Timer();
        }
        mytimer1.scheduleAtFixedRate(new timerfunc (state,device), 0, period);
        if(i==0){i=1;}
    }
    public void cylcounter() throws InterruptedException{
filledCylCount=filledCylCount+1;
        event3.addProperty("device","Cylinder_Counter");
        event3.addProperty("Cylinder_Counts",filledCylCount);
CylCounterDevice.publishEvent("status", event3);
    }
    public void fireSensorAndAlarm(){
```

```
        if(alarm.equals("OFF")){
           alarm="ON";
        }
        else
           alarm="OFF";
        event4.addProperty("device","fireSensor_And_Alarm");
        event4.addProperty("Alarm",alarm);
fireSensorDevice.publishEvent("status", event4);
    }
    public void resetCyl(){
filledCylCount=0;
    }
}
```

الكود البرمجي لاستقبال الأوامر (commandReceived.java):

يتم في هذا الكود الاتصال بمنصة IBM Watson IoT وربط محاكي معمل التعبئة مع التطبيق الخاص باستقبال أوامر التحكم من واجهة التحكم عن بعد المعرف على المنصة ومن ثمّ تشغيل أو إطفاء الأجهزة المتحكم بها وفقاً للأوامر المستقبلة.

```
package com.mycompany.factorycontrolsite;
import com.google.gson.JsonObject;
import com.ibm.iotf.client.app.ApplicationClient;
import java.io.Serializable;
import java.util.Properties;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.enterprise.context.SessionScoped;
import javax.faces.context.FacesContext;
import javax.faces.el.ValueBinding;
import javax.faces.view.facelets.FaceletContext;
import javax.inject.Named;
import org.primefaces.context.RequestContext;
@Named(value = "commandReceived")
@SessionScoped
public class commandReceived implements Serializable{
ApplicationClient myClient = null;
JsonObject event = new JsonObject();
    String deviceType;
```

```
        String oldCommandProductLine="NONE";
        String oldCommandFireSystem="NONE";
        int j=0;
    FaceletContext faceletContext = (FaceletContext)
FacesContext.getCurrentInstance().getAttributes().get(FaceletContext.FACELET_CONTEXT
_KEY);
        public commandReceived() {
            Properties options = new Properties();
    try{
    options.put("org", "mqq42b");
    options.put("id", "CoomandReceived");
    options.put("Authentication-Method","apikey");
    options.put("API-Key","a-mqq42b-9hgmg7l4hd");
    options.put("Authentication-Token","n9rZNfoYkia*xCrKbx");
    myClient = new ApplicationClient(options);
    myClient.connect();
            }
            catch (Exception ex) {
                Logger.getLogger(StateOutputTimer.class.getName()).log(Level.SEVERE, null,
ex);
            }
        }
        public void command() throws InterruptedException{
    FacesContext Context=FacesContext.getCurrentInstance();
    RequestContext Request=RequestContext.getCurrentInstance();
    myClient.setEventCallback( new MyEventCallback(faceletContext));
    myClient.subscribeToDeviceCommands();
    deviceType=faceletContext.getAttribute("device").toString();
    if(deviceType.equals("\"product-line\"")){
                String commandRecieved=faceletContext.getAttribute("ProductLine").toString();
    faceletContext.setAttribute("ProductLine", "NULL");
    if(commandRecieved.equals("\"ON\"") &&
!commandRecieved.equals(oldCommandProductLine)){
    oldCommandProductLine=commandRecieved;
    System.out.println("Command Product Line Is "+commandRecieved);
    ValueBinding binding =
Context.getApplication().createValueBinding("#{StateOutputTimer}");
    StateOutputTimer obj = (StateOutputTimer)binding.getValue(Context);
```

```
    obj.order("ON","product-line");
    Request.execute("start(925,650);btnState(1);");
            }
            else if(commandRecieved.equals("\"OFF\"") &&
!commandRecieved.equals(oldCommandProductLine)){
    oldCommandProductLine=commandRecieved;
    System.out.println("Command Product Line Is "+commandRecieved);
    ValueBinding binding =
Context.getApplication().createValueBinding("#{StateOutputTimer}");
    StateOutputTimer obj = (StateOutputTimer)binding.getValue(Context);
    obj.order("OFF","product-line");
    Request.execute("stop();btnState(2);");
            }
        }
        else if(deviceType.equals("\"fire-system\"")){
            String commandRecieved=faceletContext.getAttribute("FireSystem").toString();
    faceletContext.setAttribute("FireSystem", "NULL");
    if(commandRecieved.equals("\"ON\"") &&
!commandRecieved.equals(oldCommandFireSystem)){
    oldCommandFireSystem=commandRecieved;
    System.out.println("Command Fire System Is "+commandRecieved);
    ValueBinding binding =
Context.getApplication().createValueBinding("#{StateOutputTimer}");
    StateOutputTimer obj = (StateOutputTimer)binding.getValue(Context);
    obj.order("ON","fire-system");
    Request.execute("firesystem();");
            }
            else if(commandRecieved.equals("\"OFF\"") &&
!commandRecieved.equals(oldCommandFireSystem)){
    oldCommandFireSystem=commandRecieved;
    System.out.println("Command Fire System Is "+commandRecieved);
    ValueBinding binding =
Context.getApplication().createValueBinding("#{StateOutputTimer}");
    StateOutputTimer obj = (StateOutputTimer)binding.getValue(Context);
    obj.order("OFF","fire-system");
    Request.execute("firesystem();");
            }
        }
```

```
    }
    public void ProductLineState(String state){
oldCommandProductLine=state;
    }
    public void FireSystemState(){
if(oldCommandFireSystem.equals("\"ON\"")){
oldCommandFireSystem="OFF";
      }
      else if(oldCommandFireSystem.equals("\"OFF\"")){
oldCommandFireSystem="OFF";
      }
    }
}
```

الكود البرمجي لاستقبال الحالة (StateInput.java) :

يتم في هذا الكود الاتصال بمنصة IBM Watson IoT وربط واجهة التحكم عن بعد مع التطبيق الخاص باستقبال بيانات محاكي معمل التعبئة المعرف على المنصة ومن ثمّ تغيير حالة واجهة التحكم عن بعد لتتناسب مع بيانات المحاكي.

```
package com.mycompany.controlpanel;
import com.google.gson.JsonObject;
import com.ibm.iotf.client.app.ApplicationClient;
import java.io.Serializable;
import java.time.Instant;
import java.util.Properties;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.inject.Named;
import javax.enterprise.context.SessionScoped;
import org.primefaces.context.RequestContext;
import javax.faces.context.FacesContext;
import javax.faces.view.facelets.FaceletContext;
@Named(value = "RecievedStatesbean")
@SessionScoped
public class StateInput implements Serializable{
    int i=0;
    int period=1000;
```

```
        String oldStateProductLine="NONE";
        String oldStateFireSystem="NONE";
        String oldCylCounts="NONE";
    ApplicationClient myClient = null;
    JsonObject data = new JsonObject();
        int timetest=0;
    FaceletContext faceletContext = (FaceletContext)
FacesContext.getCurrentInstance().getAttributes().get(FaceletContext.FACELET_CONTEXT
_KEY);
        public StateInput() {
            Properties options = new Properties();
    try{
    options.put("org", "mqq42b");
    options.put("id", "StateReceived1");
    options.put("Authentication-Method","apikey");
    options.put("API-Key","a-mqq42b-r5gtowkeag");
    options.put("Authentication-Token","4W6wKVjEs&hU-@67G7");
    myClient = new ApplicationClient(options);
    myClient.connect();
            }
            catch (Exception ex) {
    Logger.getLogger(StateInput.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
        public void recievedStates() throws InterruptedException{
                //data.addProperty("product-line", state);
    RequestContext Request=RequestContext.getCurrentInstance();
    FacesContext Context=FacesContext.getCurrentInstance();
    if(myClient.isConnected()){
                timetest=Integer.parseInt(faceletContext.getAttribute("timeincreaser").toString());
    myClient.setEventCallback(new MyEventCallback(faceletContext,timetest));
    myClient.subscribeToDeviceEvents();}
    if(faceletContext.getAttribute("Device").equals("\"product-line\"")){
                    String productline=faceletContext.getAttribute("ProductLine").toString();
                    if(productline.equals("\"ON\"") && !productline.equals(oldStateProductLine)){
    oldStateProductLine=productline;
    Request.execute("btnState(1);outputScreen(1,\"ON\");");
```

```
                }
                else if(productline.equals("\"OFF\"") &&
!productline.equals(oldStateProductLine)){
    oldStateProductLine=productline;
    Request.execute("btnState(2);outputScreen(1,\"OFF\");");
                }
            }
            else if(faceletContext.getAttribute("Device").equals("\"fire-system\"")){
                String firesystem=faceletContext.getAttribute("FireSystem").toString();
                if(firesystem.equals("\"ON\"") && !firesystem.equals(oldStateFireSystem)){
    oldStateFireSystem=firesystem;
    Request.execute("btnState(3);outputScreen(2,\"ON\");");
                }
                else if(firesystem.equals("\"OFF\"") &&
!firesystem.equals(oldStateFireSystem)){
    oldStateFireSystem=firesystem;
    Request.execute("btnState(4);outputScreen(2,\"OFF\");");
                }
            }
            else if(faceletContext.getAttribute("Device").equals("\"Cylinder_Counter\"")){
                String CylCounts=faceletContext.getAttribute("CylCounts").toString();
                if(!CylCounts.equals(oldCylCounts)){
    oldCylCounts=CylCounts;
      Request.execute("cylinderCounter("+CylCounts+");outputScreen(3,"+CylCounts+");");
                }
            }
            else
if(faceletContext.getAttribute("Device").equals("\"fireSensor_And_Alarm\"")){
                String alarm=faceletContext.getAttribute("Alarm").toString();
                if(alarm.equals("\"ON\"")){
    Request.execute("alarmfun(\"ON\");outputScreen(4,\"ON\");");
                }
                else if(alarm.equals("\"OFF\"")){
    Request.execute("alarmfun(\"OFF\");outputScreen(4,\"OFF\");");
                }
            }
        }
```

}

الكود البرمجي لإرسال الأوامر (commandBean.java) :

يتم في هذا الكود الاتصال بمنصة IBM Watson IoT وربط واجهة التحكم عن بعد مع التطبيق الخاص بإرسال أوامر واجهة التحكم عن بعد المعرف على المنصة ومن ثمّ أوامر التحكم إلى منصة انترنت الأشياء بشكل مستمر باستخدام مؤقت زمني.

```
package com.mycompany.controlpanel;
import com.google.gson.JsonObject;
import com.ibm.iotf.client.app.ApplicationClient;
import java.io.Serializable;
import java.util.Properties;
import java.util.Timer;
import java.util.TimerTask;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.enterprise.context.SessionScoped;
import javax.inject.Named;
@Named(value = "commandBean")
@SessionScoped
public class commandBean implements Serializable{
   int i=0;
   int period=500;
ApplicationClient myClient = null;
JsonObject data = new JsonObject();
   public commandBean(){
      Properties options = new Properties();
try{
options.put("org", "mqq42b");
options.put("id", "commandsend1");
options.put("Authentication-Method","apikey");
options.put("API-Key","a-mqq42b-jl22qmpsyb");
options.put("Authentication-Token","_)2vTsOnf6SL6@s&(U");
myClient = new ApplicationClient(options);
myClient.connect();
      }
      catch (Exception ex) {
```

```
    Logger.getLogger(commandBean.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    class timerfunc extends TimerTask {
        private final String ord;
        private final String device;
    timerfunc(String ord,String device){
    this.ord=ord;
    this.device=device;
        }
        @Override
        public void run() {
            if(ord.equals("start") &&device.equals("product-line")){
    data.addProperty("product_line","ON");
    data.addProperty("device",device);
    myClient.publishCommand("FirstDeviceType","Gas_Filling_System", "state",data);
    cancelTimer();
            }
            else if(ord.equals("stop") &&device.equals("product-line")){
    data.addProperty("product_line","OFF");
    data.addProperty("device",device);
    myClient.publishCommand("FirstDeviceType","Gas_Filling_System", "state",data);
    cancelTimer();
            }
            else if(ord.equals("start") &&device.equals("fire-system")){
    data.addProperty("fire_system","ON");
    data.addProperty("device",device);
                myClient.publishCommand("SecondDeviceType","Fire_Fighting_System",
"state",data);
    cancelTimer();
            }
            else if(ord.equals("stop") &&device.equals("fire-system")){
    data.addProperty("fire_system","OFF");
    data.addProperty("device",device);
                myClient.publishCommand("SecondDeviceType","Fire_Fighting_System",
"state",data);
    cancelTimer();
```

```
            }
        }
    }
    Timer mytimer2=new Timer();
    public void cancelTimer(){
        mytimer2.cancel();
    }
    public void order(String ord,String device) throws InterruptedException{
        if(i==1){
        mytimer2.cancel();
        mytimer2=new Timer();
        }
        mytimer2.scheduleAtFixedRate(new timerfunc(ord,device), 0, period);
        if(i==0){i=1;}
    }
}
```

علما أن باقي الأكواد البرمجية المستخدمة في هذا البحث تمّ وضعها في CD مرفق مع ملف المشروع.