مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (٧) العدد(٣) ٢٠٢٣
Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (7) No. (3) 2023

إدارة الطاقة الكهربائية المؤتمتة لمنزل سكني بالاعتماد على تقنية إنترنت الأشياء "IoT"

د. م. ثائر أحمد ابراهيم *

م. رهام علي دباشي **

(تاريخ الإيداع ٢٠٢٣/٢/٥ . قُبِل للنشر في ٢٠٢٣/٤/٣)

□ ملخّص□

يتمحور العمل في هذا البحث حول تصميم وتنفيذ نظام إدارة للطاقة الكهربائية بالزمن الحقيقي لمنزل سكني بهدف الحد من الهدر في استهلاك الطاقة الكهربائية وخفض كمية الاستهلاك ضمن ساعات الذروة (Peak Hours)، وهي الفترة التي يزيد فيها الطلب على الطاقة الكهربائية وترتفع فيها الأحمال للحد الأعلى، وذلك من خلال الاعتماد على نظام الأولوية في تشغيل الأحمال (Priority system)، وإزاحة الأحمال الكهربائية ذات الاستطاعات العالية (جهاز تكييف الهواء – سخان المياه) لخارج أوقات الذروة.

بالاعتماد على إنترنت الأشياء (IoT) سيتم ربط النظام بالإنترنت باستخدام المُتحكّم "Node MCU" الذي يدعم إنترنت الأشياء، ومن خلال تصميم تطبيق خاص بالهواتف الذكية (Mobile Application) سنتمكن من المراقبة والتحكم في النظام المُصمم.

يستعرض البحث توصيف للتجهيزات المستخدمة في النظام المُصمم واختيار المُتحكّم المناسب، بالإضافة إلى توضيح آلية عمل النظام والبرمجيات التي تم استخدامها.

تمَ إجراء تجارب لاختبار أداء نظام إدارة الطاقة المُصمم والتحقق من إمكانياته، حيث أثبتت نتيجة الاختبارات العملية صحة عمل النظام وكفاءته في خفض استهلاك الطاقة الكهربائية ضمن ساعات الذروة، بالإضافة إلى إمكانية المراقبة والتحكم في حالة الأحمال الكهربائية عن بعد باستخدام التطبيق المُصمم.

الكلمات المفتاحية: إدارة الطاقة الكهربائية – نظام الأولوية – إنترنت الأشياء (IoT) – Node MCU

* أستاذ في قسم هندسة الأتمتة الصناعية – كلية الهندسة التقنية – جامعة طرطوس – سوريا.

**طالبة دراسات عليا في قسم هندسة الأتمتة الصناعية - كلية الهندسة التقنية – جامعة طرطوس – سوريا.

مجلة جامعة طرطوس للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (٧) العدد(٣) ٢٠٢٣
Tartous University Journal for Research and Scientific Studies - engineering Sciences Series Vol. (7) No. (3) 2023

Automated Electrical Energy Management for a Residential House Based on Internet of Things (IoT) Technology

Dr. Their Ahmad Ibrahim*
Eng. Reham Ali Debbashi **

(Received 5/2/2023 . Accepted 3/4/2023)

□ **ABSTRACT**

The work in this research focuses on the design and implementation of a real-time electrical energy management system for a Residential house with the aim of reducing waste in electrical energy consumption and reducing the amount of consumption during peak hours, which is the period in which the demand for electrical energy increases and the loads rise to the upper limit, by relying on the priority system in the operation of loads (Priority system), and the displacement of electrical loads of high capacities (air conditioner - water heater) outside peak times.

Based on the Internet of Things (IoT), the system will be connected to the Internet using the Node MCU controller that supports the Internet of Things, and by designing a mobile application, we will be able to monitor and control the designed system.

The research reviews a description of the equipment used in the designed system and the selection of the appropriate controller, in addition to clarifying the mechanism of work of the system and the software that was used.

Experiments were conducted to test the performance of the designed energy management system and verify its capabilities, where the results of practical tests proved the correctness of the system's work and efficiency in reducing electrical energy consumption during peak hours, in addition to the possibility of monitoring and controlling the state of electrical loads remotely using the designed application.

Key words: Energy management system - Priority system - Internet of thing (IoT) - Node MCU

*Professor in Department of Industrial Automation - Faculty of Technical Engineering, Tartous University - Syria.

** Graduate student in the Department of Industrial Automation Engineering - Faculty of Technical Engineering - University of Tartous – Syria.

١- المقدمة:

يُعتبر قطاع الطاقة الكهربائية في الجمهورية العربية السورية قطاعاً استراتيجياً بامتياز، ويمثل العامل الأهم في تحقيق

أي معدل للنمو، وذلك بمساهمته المباشرة وغير المباشرة في تكوين الناتج المحلي لكافة قطاعات الاقتصاد الوطني.

بصفة عامة يمكن تقسيم قطاعات استهلاك الطاقة الكهربائية في المجتمع السوري الى ما يلي: [١١]

- قطاع الأبنية: يضم الابنية السكنية والتجارية
- قطاع الزراعة: يضم الأراضي الزراعية
- قطاع النقل: يضم الشوارع والطرقات
- قطاع الصناعة: يضم المنشآت الصناعية.

ويعد قطاع الأبنية من أهم القطاعات المستهلكة للطاقة في معظم دول العالم ولاسيما بالنسبة للطاقة الكهربائية حيث يستهلك ما يفوق ٤٣% من إجمالي الطاقة المستهلكة [٦].

تعرَف إدارة جانب الطلب على الطاقة الكهربائية (Demand Side Energy Management) (DSEM): بأنها تخطيط وتنفيذ السياسات التي تساعد على تخفيض الطلب على الطاقة الكهربائية في وقت الذروة وعملياً هي استخدام المعدات التقنية لتوفير الطاقة الكهربائية بدل الحاجة لإنشاء محطات توليد جديدة [١٠].

إحدى الطرق لتخفيض ذروة الطلب في القطاع المنزلي هي التحكَم في استهلاك الطاقة الكهربائية أو تقليله خلال فترات الطلب الأقصى، وأظهرت النتائج إمكانية تخفيض استهلاك الطاقة وذروة الطلب باستخدام تقنيات إدارة الطاقة المؤتمتة، وتحقيق توفير في استهلاك الطاقة بنسبة تتراوح من ٤٠-٢٠% [٧].

أثبتت الدراسات أهمية التحكم في استخدام الأحمال الكهربائية ذات الاستطاعة العالية مثل (جهاز تكييف الهواء – سخان المياه الكهربائي – الفرن الكهربائي، ... الخ) ضمن أوقات الذروة وتشجيع المستهلك على استخدامها خارج أوقات الذروة [١].

وتوضح العديد من الدراسات الحديثة أهمية استخدام تقنية إنترنت الأشياء (IoT) للمراقبة والتحكَم في الأحمال الكهربائية في الوقت الحقيقي [2].

انطلاقاً مما سبق ذكره تبرز أهمية تصميم نظام مؤتمت لإدارة الطاقة الكهربائية في المنازل يهدف إلى استخدام الطاقة بالشكل الأمثل والحد من الهدر اعتماداً على التقنيات الالكترونية الحديثة.

٢- أهمية البحث وأهدافه:

تكمن أهمية البحث في مواكبة التطور التقني والاعتماد على التقنيات الحديثة لرفع كفاءة عملية استهلاك الطاقة الكهربائية في المنازل والعمل على إدارة جانب الطلب على الطاقة الكهربائية.

يمكن تلخيص أهداف البحث، كما يلي:

- تصميم وتنفيذ نظام مؤتمت لإدارة الطاقة الكهربائية في منزل سكني يعمل على خفض الطلب على الطاقة الكهربائية ضمن ساعات ذروة الطلب في الزمن الحقيقي.
- إزاحة استهلاك الطاقة الكهربائية إلى خارج أوقات الذروة.

- تصميم تطبيق Mobile Application خاص بالمستهلك يمكن من خلاله المراقبة والتحكم في النظام المُصمم عن بعد عبر الإنترنت.

٣- طرائق البحث، ومواده:

في البداية سيتم توصيف نظام إدارة الطاقة المصمم، وتوصيف التجهيزات الرئيسية المستخدمة، ومن ثم شرح آلية عمل النظام.

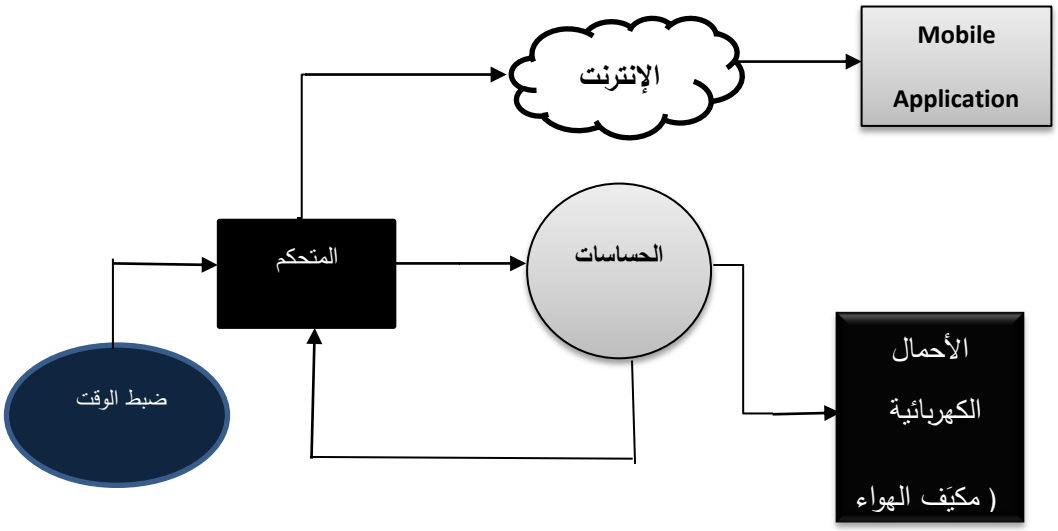
التجهيزات المستخدمة هي: المتحكم الرئيسي Node MCU – وحدة التحكم في الحرارة W١٢٠٩ – كونتاكتور عدد ٢– علبة قواطع– قاطع بتيار محدد عدد ٢ – شيلد ريليه – ريليه Finder عدد ٢ – وحدة تغذية كهربائية Power supply، سخان مياه كهربائي، مروحة.

من أجل التصميم البرمجي سيتم استعراض البرمجيات المستخدمة والتي هي البيئة البرمجية "Arduino IDE" و البرمجية "Remote X Y" لتصميم واجهة المستخدم.

٤- تصميم وتنفيذ النظام:

٤-١- المخطط الصندوقي للنظام:

يبين الشكل (١) المخطط الصندوقي لنظام إدارة الطاقة المُصمم.



الشكل (١): المخطط الصندوقي لنظام إدارة الطاقة المُصمم.

٤-١-١- المتحكَم Node MCU:

تم اختيار المتحكَم Node MCU للمراقبة والتحكم في العملية بحيث يتيح إمكانية ما يلي:

-ضبط وقت عمل النظام المراد تصميمه من خلال ربطه مع الإنترنت بشكل مباشر.

-المراقبة والتحكم في الأحمال الكهربائية في الوقت الحقيقي وعن بعد من خلال التطبيق المُصمم.

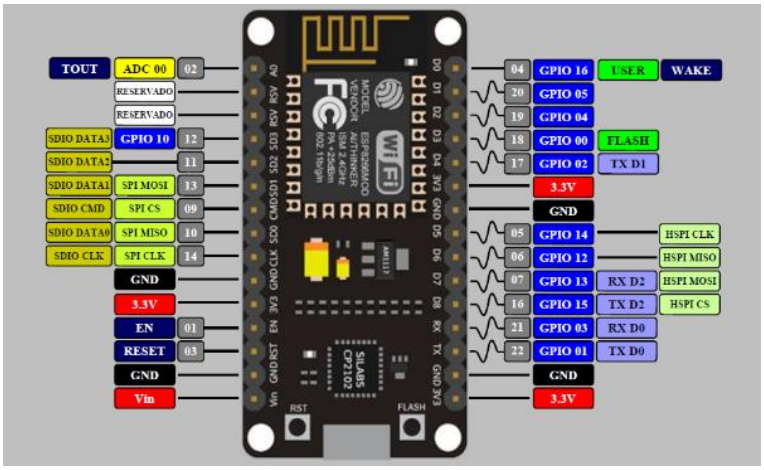
شريحة المتحكَم Node MCU:

لوحة Node MCU: هي عبارة عن لوحة الكترونية مفتوحة المصدر قابلة للبرمجة وتوفر خاصية إنترنت الأشياء (Internet of Things) والتي تسمح بربط الأشياء مع بعضها والتفاهم فيما بينها من خلال شبكة الإنترنت.

وتحتوي هذه اللوحة على شريحة ESP8266 ومنظم جهد ومحول شبكة لاسلكية WI-FI مدمج بداخله مع سرعة معالجة تبلغ ٨٠MHz بالإضافة لقدرة تخزين تصل إلى ٤MB، وهي شريحة قابلة للبرمجة عن طريق البيئة البرمجية Arduino IDE [٤].

تقدم هذه الشريحة إمكانية قراءة قيم الحساسات وإمكانية اصدار أوامر تحكمية بالإضافة إلى إمكانية إنشاء webserver وهي صفحة ويب يتم التحكم بالعملية من خلالها.

يبين الشكل (٢) أقطاب لوحة Node MCU.



الشكل (٢): أقطاب لوحة Node MCU

٤-١-٢- ضبط الوقت:

ستتم عملية التحكم والمراقبة في النظام المُصمم في الوقت الحقيقي والذي سيتم ضبطه عن طريق المتحكَم Node MCU وذلك للحفاظ على تزامن الوقت عن طريق الاستفادة من بروتوكول توقيت الإنترنت (Network Time Protocol) والذي يرمز له اختصاراً بـ NTP وهو بروتوكول إنترنت IP قياسي لمزامنة ساعة الحاسوب إلى الزمن المرجعي عبر الإنترنت، ويتم تحقيق ذلك برمجياً من خلال تضمين مكتبة <NTPClient.h> ضمن البيئة البرمجية Arduino IDE [٨].

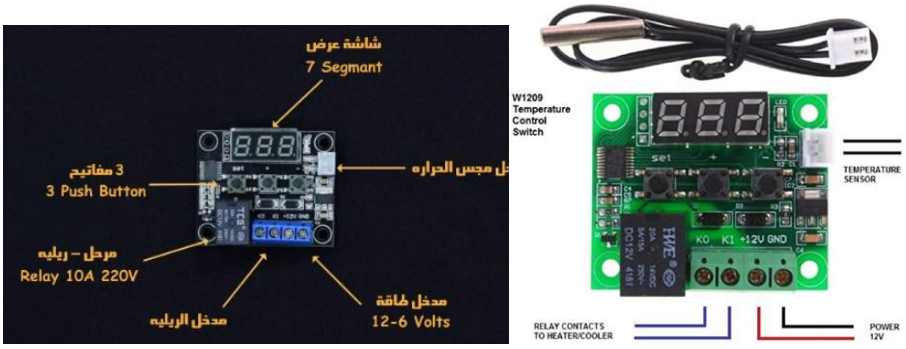
ومن خلال ذلك سيتم ضبط أوقات الذروة التي سيعمل النظام على تخفيض استهلاك الطاقة الكهربائية ضمنها من خلال تطبيق نظام الأولوية بين الأحمال الكهربائية ذات الاستطاعة العالية.

٤-١-٣- وحدة التحكم في الحرارة:

سيتم استخدام المتحكم الحراري لضبط ومراقبة درجة الحرارة التي ستعمل عندها الأحمال الكهربائية الممثلة بجهاز تكييف الهواء وسخان المياه الكهربائي.

المتحكم الحراري المراد استخدامه هو المتحكم w1209 هو عبارة عن وحدة تحكم في الحرارة عالية الدقة، ليست فقط لعرض درجة الحرارة بل تعمل على التحكم في الريليه (relay) إما بتشغيله أو إطفاءه عند التبريد أو عند التسخين وتكون اللوحة مزودة بحساس حرارة.

يوضح الشكل (٣): المتحكم الحراري w1209.



الشكل (٣): المتحكم الحراري w1209

٤-١-٤- جهاز قياس التوتر الكهربائي:

تمّ استخدام تجهيزة لقياس وإظهار جهد مصدر التغذية الكهربائية للمنزل خارج وضمن أوقات الذروة، بحيث أظهرت القياسات انخفاض التوتر الكهربائي إلى ما دون ١٨٠Volt ضمن أوقات الذروة التي يرتفع فيها معدل استهلاك الطاقة الكهربائية الأمر الذي يلحق الضرر في الكثير من التجهيزات الكهربائية، في حين أن جهد الشبكة الكهربائية التي تغذي الأحمال المنزلية ٢٢٠Volt.

يبين الشكل (٤) جهاز قياس التوتر الكهربائي: أ) القياسات ضمن أوقات الذروة – ب) القياسات خارج أوقات الذروة

أ) القياسات ضمن أوقات الذروة

ب) القياسات خارج أوقات الذروة

الشكل (٤) جهاز قياس التوتر الكهربائي

٤-١-٤- الكونتاكتور (Contactor):

تم تزويد النظام المُصمم بـ:

كونتاكتور عدد ٢ من فئة AC1 بحيث يتناسب هذ النوع مع الأحمال الأومية كالسخانات ووحدات الإضاءة، والأحمال الحثية الصغيرة كالمحركات ذات القدرات الصغيرة، وعند اختيار هذا النوع يجب أن يكون الكونتاكتور قادر على تحمل ١.٢٥ من تيار الحمل على الأقل.

قاطع أمبير عدد ٢ بحيث تتم تغذية المنزل بالطاقة الكهربائية عبر مسارين كل مسار يتم التحكم به عن طريق كونتاكتور وقاطع أمبير مضبوط على أمبير معين.

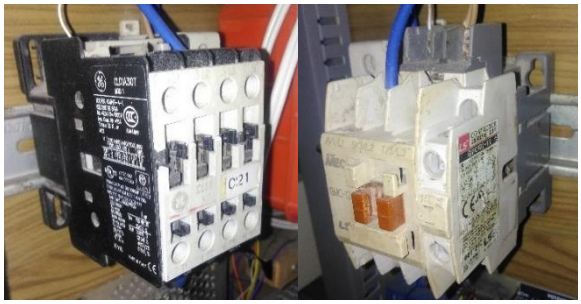
الفرق بين المسارين هو أن كل مسار له أمبير محدد يُضبط بناءً على فترات الذروة للشبكة الكهربائية وهي الفترات التي يزيد فيها الطلب على الطاقة الكهربائية وترتفع الأحمال للحد الأعلى وتكون بين الساعة (٤- ٩) مساءً في فصل الشتاء وبين الساعة (٥-١٠) مساءً في فصل الصيف [١٢]، وضمن هذه الفترات سيعمل المسار الأول ذو الأمبير المنخفض قيمته ١٦أمبير بهدف تخفيض الضغط عن الشبكة الكهربائية، أما خارج هاتين الفترتين سيعمل المسار ذو الأمبير الأعلى قيمته ٢٥أمبير.

ويتم حساب تيار القاطع الكهربائي المناسب للحمل وفق القانون التالي:

تيار القاطع الكهربائي = تيار الحمل الكهربائي * ١.٢٥

وبناءً عليه يتم اختيار أقرب قاطع مناسب للحمل ويتم اختيار الكونتاكتور المناسب.

يبين الشكل (٥): الكونتاكتورين المستخدمين في النظام.



الشكل (٥): الكونتاكتورين المستخدمين في النظام

٤-١-٥- الريليه (relay):

هو عنصر كهربائي يتكون من مفتاح ميكانيكي عبارة عن ملف مغناطيسي يستقبل إشارة تحكم معينة طبقاً لها يقوم بتوصيل أو فصل الدارة أو الحمل.

يصنف الريليه حسب نقاط التلامس وعدد حوامل هذه التلامسات.

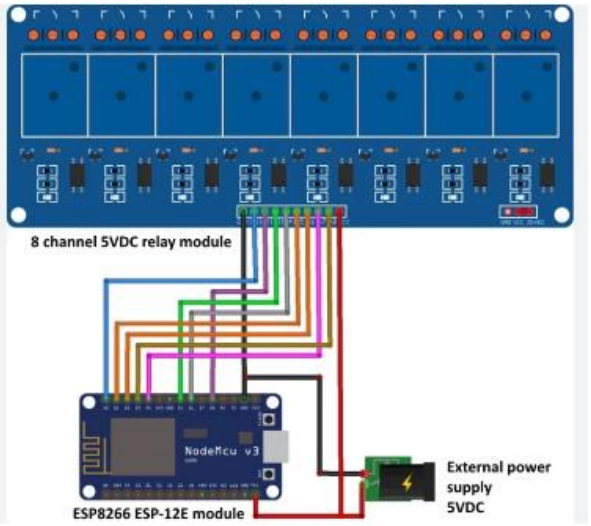
وسيتم في البحث استخدام نوعين من الريليهات هما:

- Relay Module 8-Channels: يتكون من ٨ ريليه له طرف موجب وأرضي وطرف تحكم لكل ريليه جهد التشغيل لكل ريليه ٥ فولت [٣]، و سيتم توصيله مع المتحكم Node Mcu لتأمين إشارات تحكم لملفي الكونتاكتورين المستخدمين، بالإضافة إلى تفعيل حالتي on/off ضمن تطبيق Remote xy بالنسبة للأحمال الكهربائية المستخدمة في النظام.
- ريليه Finder: سيتم استخدامه لتشغيل الأحمال الكهربائية المستخدمة في النظام المُصمم.

يبين الشكل (٦) الريليهات المستخدمة في النظام.

الشكل (٦): الريليهات المستخدمة في النظام

يوضح الشكل (٧) توصيل لوحة Node MCU مع Relay Module 8-Channels.



الشكل (٧): توصيل لوحة Node MCU مع Relay Module 8-Channels

٤-١-٦- الأحمال الكهربائية:

تقسم الأحمال الكهربائية إلى:

أحمال أومية: هي الأحمال التي لا تحتوي على ملفات نحاسية؛ مثل دوائر الإنارة والسخانات الكهربائية.

أحمال حثية: هي الأحمال التي تحتوي على ملفات نحاسية كالمحركات الكهربائية، مثل مضخات المياه، غسالة الملابس، الثلاجة، ومكيف الهواء، ....الخ.

يبين الجدول (١) استطاعة بعض الأحمال الكهربائية المنزلية مع ساعات العمل اللازمة.

الجهاز	قدرة الجهاز (واط)	العدد	معدل ساعات التشغيل اليومي	إجمالي استهلاك كل جهاز (واط / اليوم)
الثلاجة	٦٠٠	١	٨	٤٨٠٠
شاشة تلفاز	٦٥	١	٥	٣٢٥
الغسالة	١٠٠٠	١	٢	٢٠٠٠
سخان المياه الكهربائي	٢٥٠٠	١	٣	٧٥٠٠
مصابيح كهربائية	١٨	٦	٨	٨٦٤
المكواة	١٥٠٠	١	٠.٥	٧٥٠
مكيف الهواء ١طن	١٥٠٠	١	٥	٧٥٠٠
الخلاط الكهربائي	٧٠٠	١	٠.٥	٣٥٠
مجفف الشعر	١٦٠٠	١	٠.٥	٨٠٠

من الجدول السابق نحسب مايلي:

إجمالي استهلاك كل جهاز= قدرة الجهاز * العدد * عدد ساعات التشغيل اليومي

المجموع الكلي لاستهلاك الأجهزة الكهربائية = ٢٤٨٨٩ واط ساعي في اليوم = ٢٤.٨٨٩ كيلو واط ساعي في اليوم

حيث أنه يتم قياس استهلاك الطاقة الكهربائية بواحدة الكيلو واط ساعي وهو متوسط استهلاك الطاقة الكهربائية للساعة الواحدة.

نلاحظ مما سبق أن متوسط الاستهلاك لمنزل سكني حسب الأحمال الكهربائية المذكورة يساوي تقريباً ٢٥ كيلو واط ساعي في اليوم.

وفي هذا البحث تم إجراء الدراسة على حملين هما سخان المياه الكهربائي باستطاعة ٢٥٠٠Watt و مكيف الهواء باستطاعة ١٥٠٠Watt باعتبارها من بين الأحمال الأكثر استهلاكاً للطاقة الكهربائية في المنزل.

و بناءً على ذلك تمّ ضبط قيمة التيار الكهربائي لكل مسار بحيث أن المسار ذو التيار المنخفض وهو المسار الذي سيعمل ضمن أوقات الذروة، وضمن هذه الفترة سيتم تفعيل نظام الأولوية بالتالي لن يتم تشغيل الحملين في نفس الوقت أي أن أكبر قيمة للتيار الكهربائي المسحوب تبلغ ١١.٤ أمبير في حال تشغيل سخان المياه الكهربائي و ٧ أمبير في حال تشغيل مكيف الهواء، وبناءً على ذلك تم اختيار تيار القاطع الكهربائي وفق القانون:

تيار القاطع الكهربائي = تيار الحمل الكهربائي * ١.٢٥

تيار القاطع الكهربائي = ١١.٤*١.٢٥= ١٤.٢٥ أمبير ⟵ أقرب سعة قاطع مناسب ١٦ أمبير.

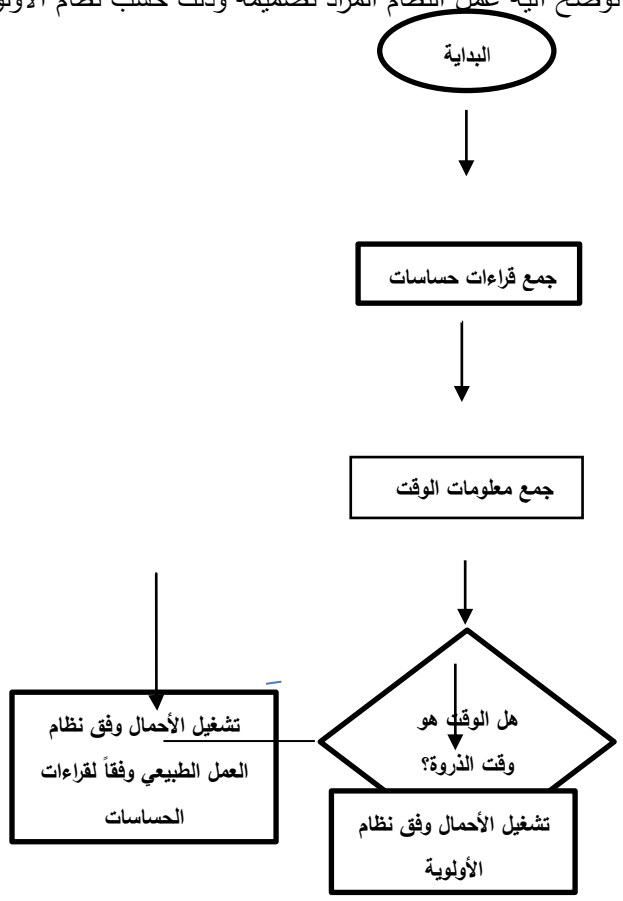
أما بالنسبة للمسار ذو التيار الأعلى وهو مسار العمل الطبيعي خارج أوقات الذروة يمكن تشغيل الحملين في نفس الوقت وستبلغ قيمة التيار المسحوب تقريباً ١٩ أمبير وبناءً على ذلك ثم اختيار تيار القاطع الكهربائي لهذا المسار وفق التالي:

تيار القاطع الكهربائي = ١٩*١.٢٥ = ٢٣.٧٥ أمبير ⟵ أقرب سعة قاطع مناسب ٢٥ أمبير.

في النموذج المُصمم تم استخدام أحمال كهربائية ذات استطاعات أصغر لإجراء التجارب واختبار النظام بحيث تم استخدام جهاز لتسخين المياه كبديل عن سخان المياه الكهربائي ومروحة كبديل عن مكيف الهواء.

٤-٢- خوارزمية عمل النظام:

يبين الشكل (٨) الخوارزمية التي توضّح آلية عمل النظام المراد تصميمه وذلك حسب نظام الأولوية.



الشكل (٨): آلية عمل النظام المُصمم

يمكن تلخيص آلية عمل نظام الأولوية كالتالي:

يعمل نظام الأولوية بشكل أتوماتيكي بين الأحمال ذات الاستطاعة العالية (جهاز تكييف الهواء – سخان المياه الكهربائي) ضمن أوقات الذروة بهدف تخفيض استهلاك الطاقة الكهربائية، ولقد ذكرنا في السابق أن النظام مزوّد بمسارين المسار الأول مضبوط عند قيمة أمبير منخفض ١٦أمبير والمسار الثاني مضبوط عند قيمة أمبير أعلى ٢٥أمبير، وعند بلوغ فترة الذروة سيتم تفعيل المسار الأول ونظام الأولوية.

وبفرض قمنا بضبط سخان المياه عند درجة حرارة ٦٠ درجة مئوية بالتالي عند عمل السخان لن يعمل جهاز التكييف حتى يصل السخان إلى درجة الحرارة المضبوطة وينطفئ، عندها يعمل جهاز التكييف، ويتم ذلك بشكل أتوماتيكي اعتماداً على معلومات الوقت وقراءات الحساسات.

بالتالي عند تطبيق نظام الأولوية لا يمكن تشغيل الحملين في نفس الوقت ولو أردنا التشغيل بشكل قسري.

يتم الحصول على معلومات الوقت وضبط أوقات الذروة وأوقات العمل الطبيعي باستخدام المتحكّم Node MCU المربوط مع شبكة الإنترنت عن طريق بروتوكول توقيت الإنترنت (NTP).

٤-٣- التصميم البرمجي:

٤-٣-١- برمجة المتحكّم Node MCU:

تتم برمجة المتحكم Node MCU ضمن البيئة البرمجية Arduino IDE المفتوحة المصدر وهي البيئة البرمجية المستخدمة في كتابة الأكواد البرمجية بلغة Arduino C وتحويلها بعد ذلك إلى صيغة تنفيذية يتم وضعها على المتحكم الصغري Esp8266 الموجود على اللوحة، وتتميز هذه البيئة البرمجية بالسهولة وبساطة لغة البرمجة [٥].

سيتم فيما يلي عرض جزء من الكود البرمجي المُستخدم، وفي المُلحق سيتم عرض الكود البرمجي بشكل كامل.

```
#define REMOTEXY_MODE__ESP8266WIFI_LIB
#include <ESP8266WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <RemoteXY.h>
```

يتضمن هذا الجزء تضمين المكتبات التي سيتم استخدامها ضمن الكود البرمجي.

```
// RemoteXY connection settings
#define REMOTEXY_WIFI_SSID "Reham"
#define REMOTEXY_WIFI_PASSWORD "123456789"
#define REMOTEXY_SERVER_PORT 6377
```

يتم في هذا الجزء ضبط إعدادات الاتصال الخاصة بالـ Remote xy مثل اسم الشبكة وكلمة المرور.

```
#define PIN_SWITCH_1 D0
#define PIN_SWITCH_2 D1
#define PIN_SWITCH_3 D2
const char *ssid     = "Reham";
const char *password = "123456789";
// Define NTP Client to get time
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org");
// Week Days
String weekDays[7]={"Sunday", "Monday", "Tuesday", "Wednesday", "Thursd, "Friday",
"Saturday"};
// Month names
String months[12]={"January", "February", "March", "April", "May", "June",
July", "August", "September", "October", "November", "December"};"
```

يتضمن الجزء السابق كيفية استخدام بروتوكول توقيت الشبكة (NTP) لضبط أوقات عمل النظام المُصمم.

٤-٣-٢- تصميم واجهة المستخدم:

واجهة المستخدم في النظام المُصمم هي عبارة عن تطبيق للهواتف الذكية (Mobile Application) تم تصميمه بواسطة البرمجية Remote XY وهو عبارة عن محرر عبر الانترنت (Online editor) يمكن بواسطته تصميم واجهات تفاعلية للمستخدم بحيث يتم من خلالها مراقبة النظام والتحكم به عن بعد، ويدعم Remote XY مجموعة من المتحكمات منها المتحكّم Node MCU [٩].

يتم استخدام خادم سحابي للاتصال ب Remote XY مما يتيح التحكم في الجهاز من أي مكان مع وجود اتصال بالإنترنت، بحيث سيتم إعداد وحدة ESP8266 كعميل من أجل الاتصال بشبكة لاسلكية (Wi-Fi).

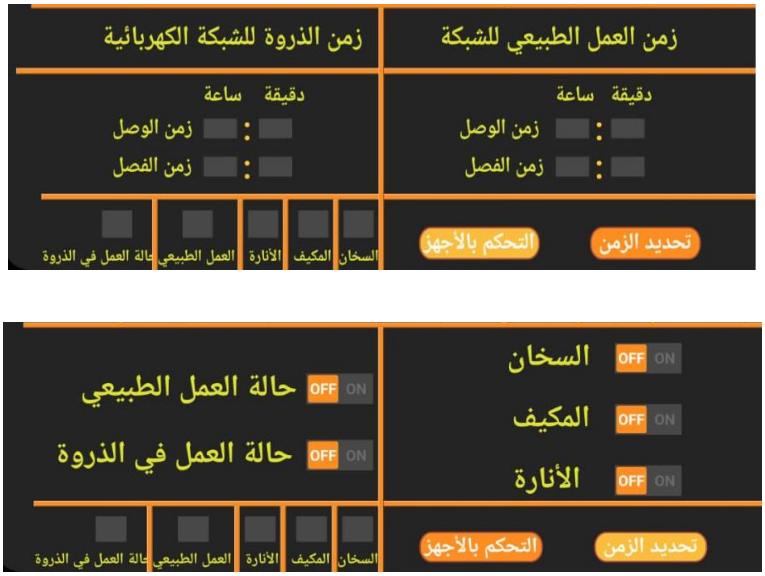
ويتم تحقيق الاتصال بين التطبيق المُصمم والمتحكّم Node MCU باستخدام بروتوكول الاتصال WI–FI.

تحوي واجهة التطبيق جزء للتحكم بحالتي النظام وهما حالة العمل الطبيعي وحالة العمل في الذروة، بالإضافة إلى التحكم في حالة عمل الأحمال الكهربائية وهما السخان وجهاز التكييف.

وتضمن الواجهة جزء لضبط الوقت بحيث يتيح للمستخدم إمكانية تحديد أوقات عمل النظام.

تم تجريب التطبيق من خلال التحكم في النظام والمراقبة عن بعد وتمّ التأكد من موثوقيته واستجابته السريعة.

ويوضح الشكل (٩) واجهة التطبيق الذي تم تصميمه.



الشكل (٩): واجهة التطبيق المصمم بواسطة Remote XY

٥– اختبار النظام المُصمم:

تمّ تصميم النموذج الذي يحاكي نظام إدارة الطاقة الكهربائية المدروس، وهو مكوّن من ثلاثة أقسام:

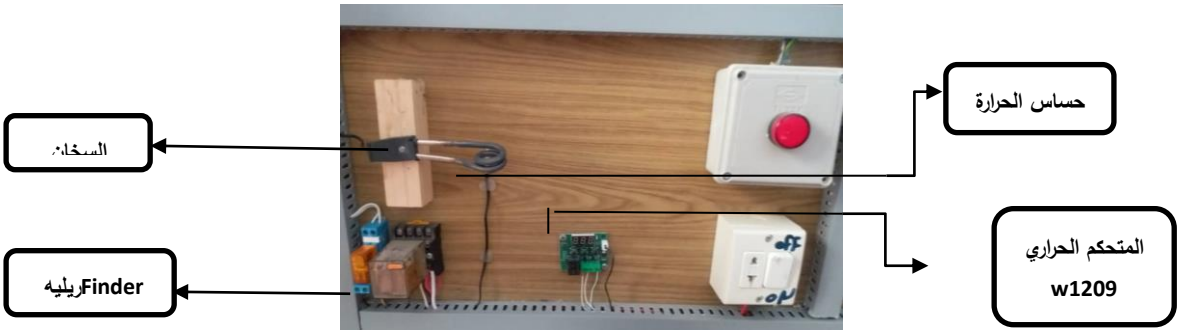
- قسم التحكّم في النظام ويتضمن المتحكم Node MCU والكونتاكتورات والقواطع.
- القسم الخاص بجهاز تكييف الهواء: يتضمن المتحكم الحراري W1209 ومروحة كبديل عن جهاز التكييف و ريليه Finder لتشغيل وإطفاء المروحة بالإضافة إلى مؤشر ضوئي للدلالة على حالة عمل الحمل.
- القسم الخاص بسخان المياه الكهربائي: يتضمن المتحكم الحراري W1209 وسخان مياه للتجريب كبديل عن سخان المياه الكهربائي وريليه Finder لتشغيل وإطفاء السخان بالإضافة إلى مؤشر ضوئي للدلالة على حالة عمل الحمل.

يبين الشكل (١٠) نظام إدارة الطاقة الكهربائية وأقسامه: أ– قسم التحكم في النظام، ب– القسم الخاص بجهاز تكييف الهواء، ج– القسم الخاص بسخان المياه.

نظام إدارة الطاقة المُصمم

أ) قسم التحكم في النظام

ب) القسم الخاص بجهاز تكييف الهواء



ج) القسم الخاص بسخان المياه

الشكل (١٠): نظام إدارة الطاقة الكهربائية وأقسامه

تمّ اختبار النظام المُصمم من خلال إجراء التجارب للتأكد من موثوقيته بحيث تمّ ضبط المتحكم الحراري لكل من السخان والمروحة عند درجات الحرارة المبينة وفق الجدول (٢).

الجدول (٢): ضبط الأولوية ودرجة الحرارة للأحمال الكهربائية المستخدمة في النموذج المُصمم.

الحمل	أولوية الحمل	إعدادات المستخدم
سخان المياه الكهربائي	١	درجة حرارة المياه ٦٠°C
المروحة	٢	درجة حرارة الغرفة ٢٤°C

عند اختبار النظام تمّ العمل وفق ما يلي:

- حالة العمل الطبيعي: وهي حالة العمل خارج أوقات الذروة بحيث يتم تشغيل الأحمال الكهربائية وفقاً لدرجات الحرارة المضبوطة.
- حالة العمل ضمن أوقات الذروة: يتم تشغيل نظام الأولوية بين الأحمال الكهربائية ضمن أوقات الذروة المحددة مسبقاً وهي بين الساعة(٤-٩) مساءً في فصل الشتاء وبين الساعة (٥-١٠) مساءً في فصل الصيف، وذلك وفق الجدول (٢) بحيث يعمل السخان الكهربائي حتى وصول درجة حرارة المياه للدرجة 60°C ومن ثمّ تعمل المروحة حتى الوصول للدرجة ٢٤°C، بالتالي عدم إمكانية تشغيل الحملين في نفس الوقت.

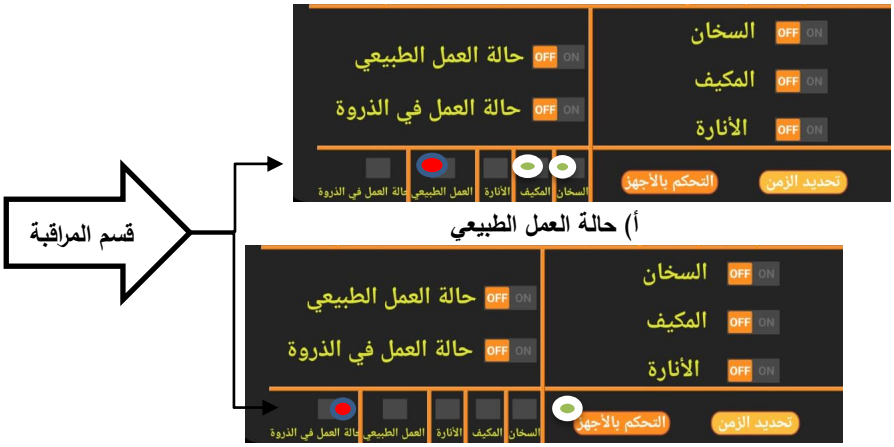
يبين الجدول (٣) كيفية تشغيل الحملين الكهربائيين ضمن الساعة (٠٤:٠٠ – ٠٤:٥٠) مساءً كمثال على تجربة النظام المُصمم ضمن ساعات الذروة.

الجدول (٣) كيفية تشغيل الأحمال الكهربائية ضمن الساعة (٠٤:٠٠ – ٠٤:٥٠) مساءً.

الأحمال الكهربائية	تشغيل الأحمال وفق نظام الأولوية	
السخان الكهربائي	Relay 1 on time: 04:00 pm	Relay 1 off time: 04:25 pm
مكيف الهواء	Relay 2 on time: 04:26 pm	Relay 2 off time: 04:50 pm

يبين الشكل (١١) مراقبة حالة عمل الأحمال ضمن واجهة المستخدم: أ) حالة العمل الطبيعي، ب) حالة العمل ضمن أوقات الذروة.

نلاحظ في الجزء السفلي من واجهة التطبيق قسم مراقبة حالة الأحمال الكهربائية بحيث أنه في وضع العمل الطبيعي هناك إمكانية لتشغيل الحملين في نفس الوقت، بينما أثناء العمل ضمن أوقات الذروة نلاحظ عدم إمكانية تشغيل الحملين في نفس الوقت.



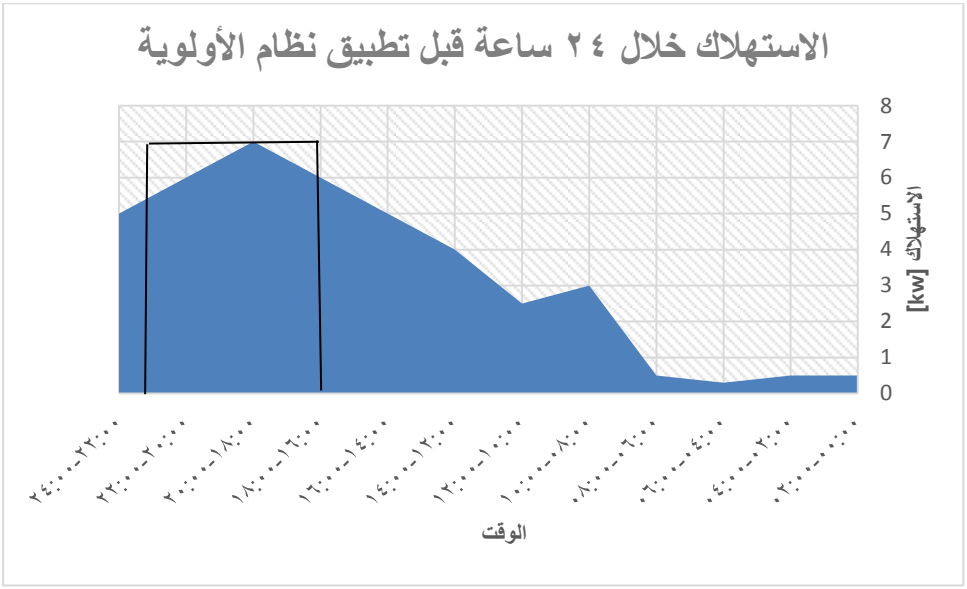
أ) حالة العمل الطبيعي

ب) حالة العمل ضمن أوقات الذروة

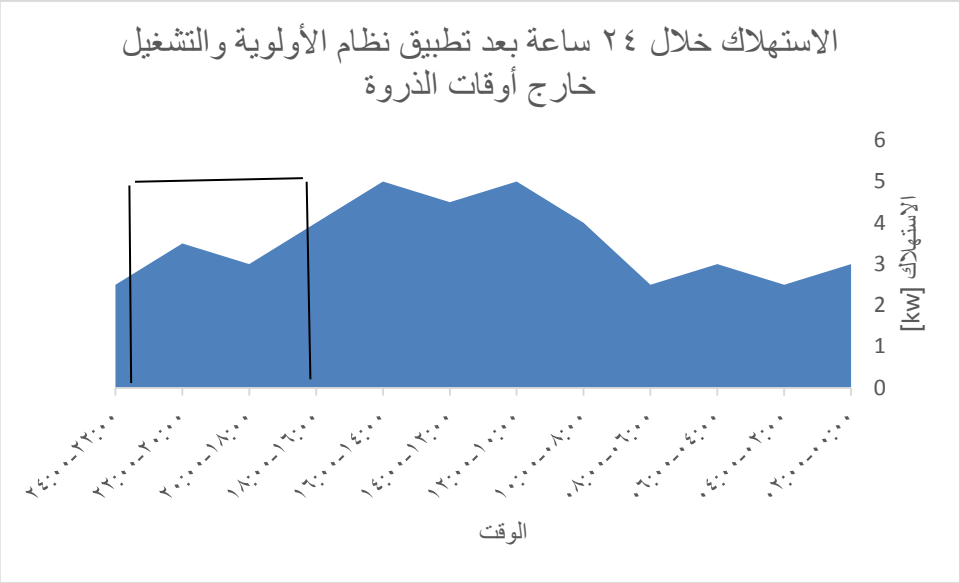
الشكل (١١): مراقبة حالة عمل الأحمال ضمن واجهة المستخدم.

يبين الشكل (١٢) الاستهلاك خلال ٢٤ ساعة قبل تطبيق النظام.

الشكل (١٣) الاستهلاك خلال ٢٤ ساعة بعد تطبيق النظام المُصمم.



الشكل (١٢): الاستهلاك خلال ٢٤ ساعة قبل تطبيق النظام المُصمم



الشكل (١٣): الاستهلاك اليومي بعد تطبيق النظام المُصمم

بالمقارنة بين الشكلين (١٢) و (١٣) نلاحظ انخفاض استهلاك الطاقة الكهربائية ضمن ساعات الذروة، بحيث انخفض الاستهلاك على سبيل المثال بين الساعة ١٦:٠٠ و ١٧:٠٠ بما يقارب ٢ كيلو واط.

دراسة النتائج ومناقشتها:

بعد إجراء التجارب اللازمة للنظام واختباره، يمكننا تلخيص النتائج والإمكانيات التي يمكن تحقيقها عن طريق النظام كما يلي:

- ✓ يحقق النظام إمكانية خفض في استهلاك الطاقة الكهربائية لمنزل سكني ضمن ساعات ذروة الطلب بنسبة تتراوح بين ٣٠% حتى ٤٠ % عن طريق تفعيل نظام الأولوية بين الأحمال الكهربائية ذات الاستطاعات العالية.
- ✓ يحقق النظام إمكانية إزاحة الأحمال الكهربائية ذات الاستطاعة العالية لخارج أوقات الذروة.
- ✓ يحقق النظام إمكانية مراقبة حالة الأحمال الكهربائية في الوقت الحقيقي.
- ✓ يحقق النظام إمكانية التحكم في حالة الأحمال الكهربائية (on/off) عن بعد بواسطة التطبيق المُصمم عبر الانترنت.

٦- الاستنتاجات والتوصيات:

- تبين من خلال البحث أهمية تطبيق أنظمة مؤتمتة لإدارة الطاقة الكهربائية ضمن المنازل السكنية مما يحسن من كفاءة استهلاك الطاقة الكهربائية.
- تبين من خلال البحث قدرة نظام إدارة الطاقة على خفض الاستهلاك ضمن ساعات ذروة الطلب مما يحسن من موثوقية التغذية الكهربائية.

- يوضح البحث أن المستفيدين من إدارة الطلب على الطاقة الكهربائية هما شركة الكهرباء والمستفيدين في آن واحد، حيث ستستفيد الشركة من خلال تخفيض ذروة الطلب بالتالي انخفاض عدد الخروج الإجباري لمجموعات التوليد وارتفاع الموثوقية فيها، وتأخير نمو الحمل وبالتالي تخفيض التكاليف اللازمة للتوليد والتوزيع، وتخفيض ضياعات النقل والتوزيع وزيادة المبيعات خارج أوقات الذروة، أما بالنسبة للمستهلك ستؤدي إلى تخفيض كلفة الكهرباء وزيادة الموثوقية.
- يمكن الاعتماد على لوحات "Node MCU" التي تدعم إنترنت الأشياء والتي تمتاز بسعرها المقبول واستهلاكها المنخفض للطاقة الكهربائية وسهولة برمجتها في تصميم أنظمة مؤتمتة لإدارة الطاقة الكهربائية.

في الأعمال المستقبلية، نقترح ما يلي:

- استخدام شبكات الحساسات اللاسلكية (WSN) للعمل على تحسين كفاءة نظام إدارة الطاقة.
- استخدام تقنيات الذكاء الاصطناعي التي تساعد على بناء نظام يتكيف مع سلوك المستهلك ومتطلباته.

المراجع:

المراجع الأجنبية:

[1] Butt, M. B., Dilshad, S., Abas, N., Rauf, S., & Saleem, M. S. (2021). *Design of home load management system for load rationing in Pakistan*. *Engineering Reports*, *3*(3), e12312.

[2] Madhu, G. M., & Vyjayanthi, C. (2018, June). *Implementation of cost effective smart home controller with Android application using node MCU and internet of things (IOT*). In *2018 2nd International Conference on Power, Energy and Environment: Towards Smart Technology (ICEPE)* (pp. 1-5). IEEE.

[3] Tadi, S., Polana, L. P., Potla, S., Vudata, V. T., & Vuyyala, N. L. (2018). *Home Control Using Node MCU*. *International Journal for Advance Research and Development*, *3*(2), 87-92.

[4] Saikrishna, M., & Vijaykiran, G. (2017). *IoT based home electrical appliances control using node MCU*. *International Journal of Scientific Engineering and Technology Research*, *6*(04), 0783-0788.

[5] Herger, L. M., & Bodarky, M. (2015). *Engaging students with open source technologies and Arduino*.In *2015 IEEE Integrated STEM Education Conference* (pp. 27-32). IEEE.

[6] Hainoun, A., Seif-Eldin, M. K., & Almoustafa, S. (2006). *Analysis of the Syrian long-term energy and electricity demand projection using the end-use methodology*. *Energy policy*, *34*(14), 1958-1970.

[7] Nasser Kutkut. (2006). *Energy Management through Peak Shaving and Demand Response: New Opportunities for Energy Savings at Manufacturing and Distribution Facilities", Advanced Charging Technology ACT*.

[8*] "Getting Current Data and Time with ESP8266 Node MCU using NTP Server – Client and Arduino IDE"*, 5July.2022.http://microcontrollerslab.com.

[9] *"RemoteXY open-source platform"*, 6 March.2022. https://remotexy.com/en/editor/.

المراجع العربية:

[١٠] ابعوه. الصديق، بادي. إبراهيم & شتوان. علي.(٢٠١٧). *إدارة جانب الطلب على الطاقة الكهربائية للقطاع السكني بليبيا، المجلة الدولية المحكمة للعلوم الهندسية وتقنية المعلومات.*

[١١] أبو خزام. وسيم. (٢٠١١). *العمارة الخضراء والمباني المستدامة*، الهيئة العامة لبحوث الطاقة، سوريا.

[١٢] طعمة. سنجار. *ترشيد الطاقة الكهربائية*، المركز الوطني لبحوث الطاقة NERC، سوريا.

المُلحق

```
#define REMOTEXY_MODE__ESP8266WIFI_LIB
#include <ESP8266WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <RemoteXY.h>

// RemoteXY connection settings
#define REMOTEXY_WIFI_SSID "Reham"
#define REMOTEXY_WIFI_PASSWORD "123456789"
#define REMOTEXY_SERVER_PORT 6377

// RemoteXY configurate
#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] =   // 945 bytes
{ 255,91,0,12,0,170,3,10,25,4,7,4,31,35,7,4,1,2,26,2,
11,7,4,31,42,7,4,1,2,26,2,11,7,4,42,35,7,4,1,2,
26,2,11,7,4,42,42,7,4,1,2,26,2,11,7,4,79,35,7,4,
1,2,26,2,11,7,4,79,42,7,4,1,2,26,2,11,7,4,90,35,
7,4,1,2,26,2,11,7,4,90,42,7,4,1,2,26,2,11,129,0,
0,18,47,5,1,3,216,178,217,133,217,134,32,216,167,217,132,216,176,216,
177,217,136,216,169,32,217,132,217,132,216,180,216,168,217,131,216,169,32,216,
167,217,132,217,131,217,135,216,177,216,168,216,167,216,166,217,138,216,169,0,
129,0,61,18,42,5,1,3,216,178,217,133,217,134,32,216,167,217,132,216,
185,217,133,217,132,32,216,167,217,132,216,183,216,168,217,138,216,185,217,138,
32,217,132,217,132,216,180,216,168,217,131,216,169,0,129,0,31,29,7,4,
1,3,216,179,216,167,216,185,216,169,0,129,0,43,29,7,4,1,3,216,
175,217,130,217,138,217,130,216,169,0,129,0,79,29,7,4,1,3,216,179,
216,167,216,185,216,169,0,129,0,90,29,7,4,1,3,216,175,217,130,217,
138,217,130,216,169,0,129,0,13,35,15,4,1,3,216,178,217,133,217,134,
32,216,167,217,132,217,136,216,181,217,132,0,129,0,13,42,15,4,1,3,
216,178,217,133,217,134,32,216,167,217,132,217,129,216,181,217,132,0,129,0,
60,35,15,4,1,3,216,178,217,133,217,134,32,216,167,217,132,217,136,216,
181,217,132,0,129,0,61,42,15,4,1,3,216,178,217,133,217,134,32,216,
167,217,132,217,129,216,181,217,132,0,130,1,234,26,179,1,1,2,129,0,
39,34,2,6,1,79,58,0,129,0,39,41,2,6,1,79,58,0,129,0,
87,34,2,6,1,79,58,0,129,0,87,41,2,6,1,79,58,0,131,1,
88,55,22,6,1,2,31,216,170,216,173,216,175,217,138,216,175,32,216,167,
217,132,216,178,217,133,217,134,0,131,0,60,55,27,6,2,2,31,216,167,
217,132,216,170,216,173,217,131,217,133,32,216,168,216,167,217,132,216,163,216,
172,217,135,216,178,216,169,0,2,0,54,18,13,6,2,2,26,31,31,79,
78,0,79,70,70,0,2,0,54,30,13,6,2,2,26,31,31,79,78,0,
79,70,70,0,2,0,54,42,13,6,2,2,26,31,31,79,78,0,79,70,
70,0,129,0,31,18,14,6,2,3,216,167,217,132,216,179,216,174,216,167,
217,134,0,129,0,32,30,13,6,2,3,216,167,217,132,217,133,217,131,217,
138,217,129,0,129,0,32,42,12,6,2,3,216,167,217,132,216,163,217,134,
216,167,216,177,216,169,0,130,1,239,49,179,1,0,2,130,1,237,14,179,
1,0,2,129,0,43,59,7,3,0,3,216,167,217,132,216,179,216,174,216,
167,217,134,0,129,0,24,59,7,3,0,3,216,167,217,132,217,133,217,131,
217,138,217,129,0,129,0,7,59,6,3,0,3,216,167,217,132,216,163,217,
134,216,167,216,177,216,169,0,129,0,241,0,112,8,0,2,217,131,217,132,
217,138,216,169,32,216,167,217,132,217,135,217,134,216,175,216,179,216,169,32,
216,167,217,132,216,170,217,130,217,134,217,138,216,169,45,217,130,216,179,217,
133,32,216,167,217,132,216,163,216,170,217,133,216,170,216,169,32,216,167,217,
132,216,181,217,134,216,167,216,185,217,138,216,169,0,129,0,244,8,107,6,
0,2,216,165,216,185,216,175,216,167,216,175,32,58,32,217,133,32,46,32,
216,177,217,135,216,167,217,133,32,216,175,216,168,216,167,216,180,217,138,45,
216,165,216,180,216,177,216,167,217,129,32,58,32,216,175,32,46,32,216,171,
216,167,216,166,216,177,32,216,163,216,173,217,133,216,175,32,216,167,216,168,
216,177,216,167,217,135,217,138,217,133,0,130,1,57,50,1,13,0,2,130,
1,18,50,1,14,0,2,130,1,37,50,1,14,0,2,67,5,44,52,6,
5,0,2,26,4,67,5,24,52,6,5,0,2,26,4,67,5,7,52,6,
5,0,2,26,4 };

// this structure defines all the variables and events of your control interface
struct {

// input variables
char T1[11];  // string UTF8 end zero
char T3[11];  // string UTF8 end zero
```

```
    char T2[11];  // string UTF8 end zero
    char T4[11];  // string UTF8 end zero
    char T5[11];  // string UTF8 end zero
    char T7[11];  // string UTF8 end zero
    char T6[11];  // string UTF8 end zero
    char T8[11];  // string UTF8 end zero
  uint8_t switch_1; // =1 if switch ON and =0 if OFF
  uint8_t switch_2; // =1 if switch ON and =0 if OFF
  uint8_t switch_3; // =1 if switch ON and =0 if OFF
                    // output variables
  char text_1[4];  // string UTF8 end zero
  char text_2[4];  // string UTF8 end zero
  char text_3[4];  // string UTF8 end zero

                    // other variable
  uint8_t connect_flag;  // =1 if wire connected, else =0

                    } RemoteXY;
                  #pragma pack(pop)

                  //////////////////////////////////////////
                  //         END RemoteXY include         //
                  //////////////////////////////////////////

                    #define PIN_SWITCH_1 D0
                    #define PIN_SWITCH_2 D1
                    #define PIN_SWITCH_3 D2

                  const char *ssid     = "Reham";
                  const char *password = "123456789";

                  // Define NTP Client to get time
                    WiFiUDP ntpUDP;
                  NTPClient timeClient(ntpUDP, "pool.ntp.org");

                    //Week Days
  String weekDays[7]={"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};

                    //Month names
  String months[12]={"January", "February", "March", "April", "May", "June",
  "July", "August", "September", "October", "November", "December"};

                    void setup()
                    {
                  digitalWrite(D0,HIGH);
                  digitalWrite(D1,HIGH);
                  digitalWrite(D2,HIGH);
                  digitalWrite(D3,HIGH);
                  digitalWrite(D4,HIGH);

                    RemoteXY_Init ();

                  pinMode (PIN_SWITCH_1, OUTPUT);
                  pinMode (PIN_SWITCH_2, OUTPUT);
                  pinMode (PIN_SWITCH_3, OUTPUT);
                    pinMode (D3, OUTPUT);
                    pinMode (D4, OUTPUT);

                    // TODO you setup code
                  // Initialize Serial Monitor
                      Serial.begin(115200);
                      // Connect to Wi-Fi
                  Serial.print("Connecting to ");
                      Serial.println(ssid);
                    WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
                      delay(500);
                    Serial.print(".");
                      }

                  // Initialize a NTPClient to get time
                      timeClient.begin();
  // Set offset time in seconds to adjust for your timezone, for example:
```

```
timeClient.setTimeOffset(10800); // GMT +1 = 3600 ,GMT +8 = 28800 ,GMT -1 = -3600 ,GMT 0 = 0
}
void loop()
{
RemoteXY_Handler ();

digitalWrite(PIN_SWITCH_1, (RemoteXY.switch_1==1)?LOW:HIGH);
digitalWrite(PIN_SWITCH_2, (RemoteXY.switch_2==1)?LOW:HIGH);
digitalWrite(PIN_SWITCH_3, (RemoteXY.switch_3==1)?LOW:HIGH);
int val = atoi (RemoteXY.T1);
int val2 = atoi (RemoteXY.T2);
int val3 = atoi (RemoteXY.T3);
int val4 = atoi (RemoteXY.T4);
int val5 = atoi (RemoteXY.T5);
int val6 = atoi (RemoteXY.T6);
int val7 = atoi (RemoteXY.T7);
int val8 = atoi (RemoteXY.T8);
if (digitalRead(D5)==LOW) strcpy (RemoteXY.text_1, "on");
else strcpy (RemoteXY.text_1, "off");
if (digitalRead(D6)==LOW) strcpy (RemoteXY.text_2, "on");
else strcpy (RemoteXY.text_2, "off");
if (digitalRead(D7)==LOW) strcpy (RemoteXY.text_3, "on");
else strcpy (RemoteXY.text_3, "off");
timeClient.update();
unsigned long epochTime = timeClient.getEpochTime();
Serial.print("Epoch Time: ");
Serial.println(epochTime);
String formattedTime = timeClient.getFormattedTime();
Serial.print("Formatted Time: ");
Serial.println(formattedTime);
int currentHour = timeClient.getHours();
Serial.print("Hour: ");
Serial.println(currentHour);
int currentMinute = timeClient.getMinutes();
Serial.print("Minutes: ");
Serial.println(currentMinute);
int currentSecond = timeClient.getSeconds();
Serial.print("Seconds: ");
Serial.println(currentSecond);
String weekDay = weekDays[timeClient.getDay()];
Serial.print("Week Day: ");
Serial.println(weekDay);
//Get a time structure
struct tm *ptm = gmtime ((time_t *)&epochTime);
int monthDay = ptm->tm_mday;
Serial.print("Month day: ");
Serial.println(monthDay);
int currentMonth = ptm->tm_mon+1;
Serial.print("Month: ");
Serial.println(currentMonth);
String currentMonthName = months[currentMonth-1];
Serial.print("Month name: ");
Serial.println(currentMonthName);
int currentYear = ptm->tm_year+1900;
Serial.print("Year: ");
Serial.println(currentYear);
//Print complete date:
String currentDate = String(currentYear) + "-" + String(currentMonth) + "-" + String(monthDay);
Serial.print("Current date: ");
Serial.println(currentDate);
Serial.println("");
delay(1000);
if (currentHour==val&&currentMinute==val2)
{digitalWrite(D3,LOW);}
else if (currentHour==val3&&currentMinute==val4)
{digitalWrite(D3,HIGH);}
if (currentHour==val5&&currentMinute==val6)
{digitalWrite(D4,LOW);}
else if (currentHour==val7&&currentMinute==val8)
{digitalWrite(D4,HIGH);}

}
```