DESIGNING AN ALGORITHM AND CREATION OF A SOFTWARE APPLICATION FOR DATA COMMUNICATION IN ACCORDANCE WITH THE MQTT PROTOCOL

BOSOC Vlad-George

Facultatea: Inginerie Industrială și Robotică, Specializarea: Informatică Aplicată în Inginerie Industrială, Anul de studii: 4, e-mail: vlad_george.bosoc@stud.fiir.upb.ro

Conducător ştiinţific: Prof. dr. ing. **Tom SAVU**

ABSTRACT: Regarding the evolution of technology in different industries, be it energetics, computer science, manufacturing or banking, one concern that comes to an engineer or manager's mind is the ability to transmit information between different sections of an enterprise. No matter if it is one's home, a factory or an entire university, data communication is one of the most important aspects when it is desired to automate processes, monitor them for optimization or just keep "things" (as in Internet of Things) linked together to exchange information during operation. As the time went by there have been developed a lot of communication protocols in order to be adapted and used in accordance with one's application. The protocol on which this paper is focused on is called MQTT (Message Queuing Telemetry Transport).

KEY WORDS: industry, protocol, algorithm, communication, server

1. Introducere

Protocoalele de comunicație reprezintă o metodă de a avea o structură organizată și eficientă pentru transmiterea de date într-un mediu industrial între 2 sau mai multe dispozitive electronice. Aceste protocoale se întind pe mai multe domenii cum ar fi comunicațiile de date la nivel de infrastructură (IPv4/IPv6, RPL), comunicații prin protocoale de date (CoAP, AMQP, WebSocket) [1] sau comunicații prin protocoale inter-sistem (UART, USB) [2].

Această lucrare se axează pe realizarea unei aplicații pentru a permite comunicarea de date prin protocolul/standardul MQTT. Acesta este un protocol de date ce se află pe nivelul 7 al stivei OSI (Open Systems Interconnection Model [3]), se bazează pe transmiterea de mesaje prin intermediul unei tehnici de tipul „publish-subscribe" în care, după cum ii spune și numele, dispozitivele respectivului sistem IoT[1] publică mesaje la un broker/server MQTT sau se abonează și primesc mesajele publicate la broker. Acest protocol nu funcționează singur, întrucât acesta doar prelucrează datele primite, nu le și transportă. De aceea pentru transportul mesajelor între dispozitive și broker s-a apelat la protocolul TCP/IP (Transport Control Protocol/Internet Protocol) deoarece este eficient și ușor de implementat. Protocolul MQTT este util în mediile în care există o lățime de bandă limitată, iar viteza de comunicație este un element important.

O altă entitate ce stă la baza funcționării corecte a unui sistem cu protocol MQTT o reprezintă pachetele de control. Aceste pachete de control reprezintă pachetele de mesaje propriu-zise ce sunt transmise de la un dispozitiv la broker sau invers. Aceste pachete sunt în număr de 15 și anume:

- CONNECT = cererea de conectare la server-ul MQTT
- CONNACK = confirmarea cererii de conectare
- PUBLISH = publicare de informații
- PUBACK (QoS1[2]) = confirmarea publicării informațiilor
- PUBREC (QoS2) = prima parte a completării publicării
- PUBREL (QoS2) = a doua parte a completării publicării

1 – Internet of Things: concept al Ingineriei 4.0 ce presupune interconectivitatea dispozitivelor inteligente
2 – Quality of Service: mecanism de gestionare a traficului de mesaje și de eficientizare a acestuia [4]

- PUBCOMP (QoS2) = ultima parte a completării publicării
- SUBSCRIBE = cerere de abonare la un topic
- SUBACK = confirmarea abonării la un topic
- UNSUBSCRIBE = cerere de dezabonare de la un topic
- UNSUBACK = confirmarea dezabonării de la un topic
- PINGREQ = cerere de ping către server
- PINGRESP = răspuns de la server la ping-ul clientului
- DISCONNECT = deconectare de la broker
- AUTH = date de autentificare

Pachetele de control PUBLISH, PUBACK, PUBREL, PUBCOMP, DISCONNECT și AUTH pot fi transmise bidirecțional.

Fixed Header, present in all MQTT Control Packets
Variable Header, present in some MQTT Control Packets
Payload, present in some MQTT Control Packets

Fig. 1. Structura de bază a unui pachet de control [5]

După cum a fost deja menționat comunicația se realizează sub forma unor pachete de mesaje (sau pachete de control). Aceste pachete sunt construite pe mărimi mici, de ordinul octeților, având în componența lor anumite părți care sunt codificate sub formă de Bits, Binary Data, Two Byte Integer, Four Byte Integer, Variable Byte Integer, UTF-8 String sau UTF-8 String Pair. Această codificare ajută atât la procesarea mesajelor în cazul în care sistemul de operare sau dispozitivul care transmite mesaje este diferit față de un PC care rulează Windows, cât și la optimizarea memoriei pe care aceste pachete o ocupă. Datorită dimensiunilor mici ale pachetelor și ale acestor codificări se poate vorbi de o viteză de comunicație bună pe un număr mare de dispozitive interconectate.

În figurile următoare se regăsesc schema de principiu a funcționalității sistemului (realizată după modelul IDEF0[6]; acest model a fost adaptat pentru aplicația din această lucrare) și diagrama de dependență ale componentelor acestuia:

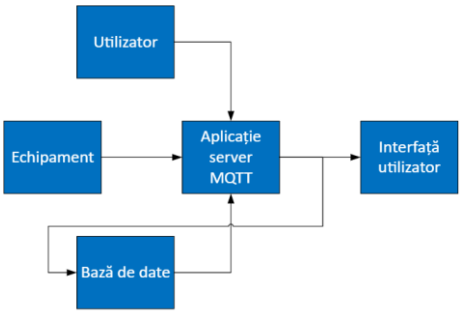

Fig. 2. Schemă de principiu după modelul IDEF0

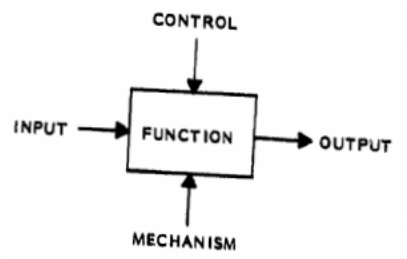

Fig. 3. Modul de reprezentare a unei funcții de tip model IDEF0

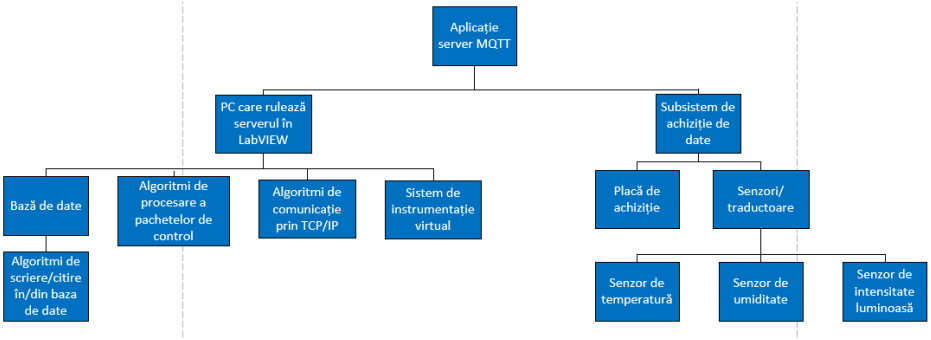

Fig. 4. Diagrama de dependență a componentelor sistemului

Pentru realizarea unei aplicații informatice care va rula pe un PC pentru a gestiona mesajele transmise prin protocolul MQTT s-a utilizat mediul de programare LabVIEW (Laboratory Visual Instrumentation Engineering Workbench). Acesta este un program renumit pentru instrumentația și codajul grafice (limbajul grafic este denumit „G”[7]) și este utilizat, în general, pentru achiziții de date, automatizări industriale sau pe post de instrumentație de control. În cazul de față a fost utilizat pentru a forma un broker MQTT, pentru a „asculta” cererile de conexiune prin rețea la broker prin implementarea unui „listener” de protocol TCP/IP și pentru realizarea unei interfețe de monitorizare a conexiunilor și sesiunilor cu clienții conectați.

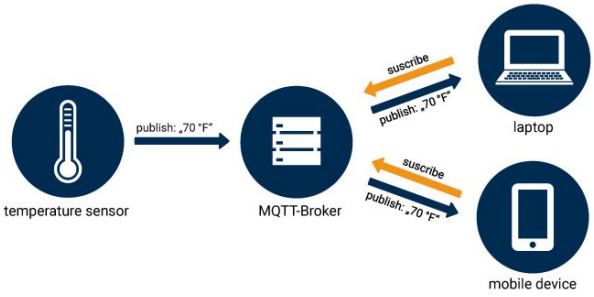

Fig. 5. Schemă de principiu a comunicației prin protocolul MQTT[8]

Obiectivul urmărit prin realizarea acestei aplicații este acela de a implementa un sistem IoT în cadrul Universității Politehnica din București. Cu acesta se poate ține evidența aparatelor și echipamentelor și se pot monitoriza sau chiar automatiza activitățile susținute de acestea, fie ele mașini-unelte sau roboți mobili.

2. Stadiul actual

În momentul de față aplicația are realizat complet pachetul CONNECT; atât structura generală a pachetului, cât și algoritmii de prelucrare a datelor acestuia și al restului de pachete de control. De asemenea, s-a stabilit și o metodă de conexiune la broker prin protocolul TCP/IP care, cel puțin la momentul actual, gestionează doar o conexiune activă.

Această conexiune este, totuși, acceptată numai dacă numele de utilizator și parola existente în pachetul CONNECT corespund numelui de utilizator și parolei aferente din baza de date. La momentul actual baza de date constă într-un fișier cu extensia “.txt” pentru simplitate în testări.

Codurile/instrumentele virtuale (“VI”), principale care au fost realizate pentru procesarea pachetului sunt următoarele (toate codurile/fișierele au extensia “.vi”):

- Check Packet (verifică pachetul pentru validare)
- Check Users Database (compară datele de autentificare cu cele din baza de date)
- Check_Fixed_Header (verifică header-ul fix al pachetului de control)
- Get Properties (obține proprietățile pachetului de control)
- MQTT Globals (VI pentru stocarea variabilelor globale)
- + vi-urile de codificare/decodificare ale anumitor segmente din pachet

3. Simulare model experimental

Pentru testarea funcționalității aplicației și a sistemului general s-a realizat un model experimental compus din următoarele componente:

- sistem pe un chip model ESP32
- fotocelulă/fotorezistor
- senzor DHT11
- 1 rezistență pull-up de 10kOhm
- 1 rezistență pull-down de 10kOhm
- fire de legătură
- cablu de alimentare prin micro-USB
- 2 breadboard-uri

Prin asamblarea acestor componente (figura 3) placa cu chip ESP32 este capabilă să recepționeze datele trimise de către senzorul de umiditate și temperatură (DHT11) și fotocelulă, urmând ca aceste date să fie trimise prin TCP/IP către broker-ul MQTT. Schema de conexiuni a fost realizată în programul EasyEDA.

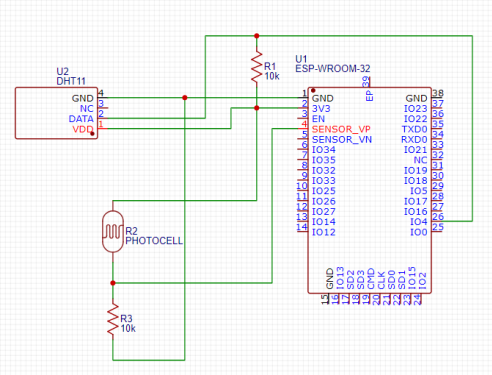

Fig. 6. Schema de conexiuni dintre componentele enumerate anterior

Deoarece nu s-a putut gasi un model exact al plăcii se menționează aici că placa este alimentată prin cablul micro-USB (5V) sau prin pinii de alimentare, anume VIN și GND.

Aceste procese au putut fi realizate prin codaj în C/C++ prin intermediul mediului de programare Arduino. Întregul proces se derulează astfel:

1. placa cu ESP32 inițializează portul ADC (Analog-to-Digital Converter) (acest pas este necesar pentru recepția de date de la fotocelulă);
2. placa se conectează la rețeaua locală utilizând SSID-ul și parola router-ului. Dacă eșuează, acest pas se repetă;

3. se inițializează senzorul DHT;
4. odată conectată la rețeaua locală placa se conectează la broker (PC-ul cu aplicația în LabVIEW). Dacă eșuează, acest pas se repetă;
5. odată ce conexiunea a reușit placa trimite seria de octeți care constituie pachetul CONNECT;
6. dacă numele de utilizator și parola existente în pachet corespund cu cele din baza de date, atunci aplicația permite broker-ului să accepte datele recepționate de la ESP32. Dacă datele de autentificare nu sunt acceptate, conexiunea se încheie și se reia procesul.

Pentru simplitate s-a ales opțiunea de a încheia direct conexiunea și de a opri VI-ul în schimbul trimiterii unui mesaj către ESP.

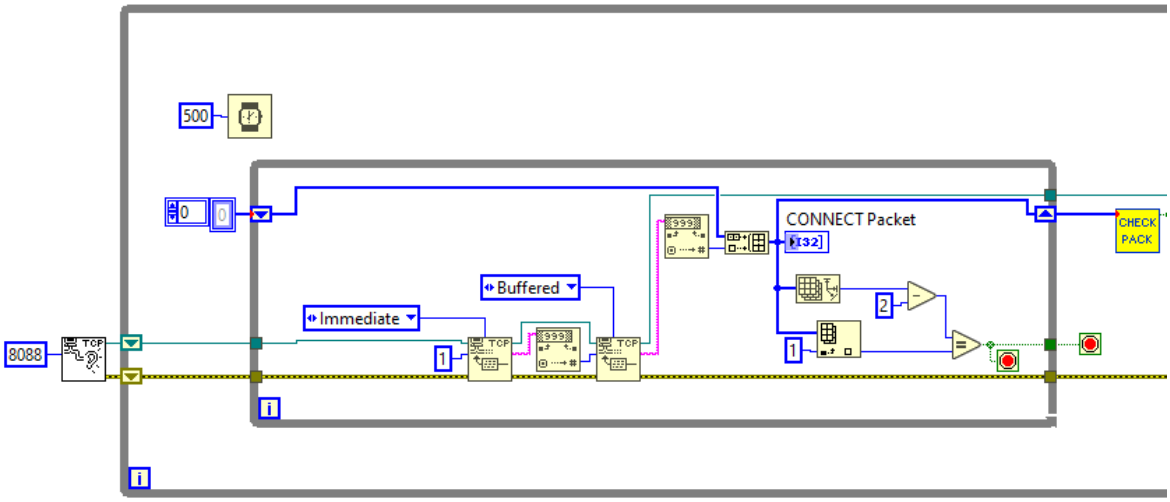

Fig. 7. Partea de ascultare a unei cereri de conexiune la PC, recepționare și procesare a pachetului CONNECT

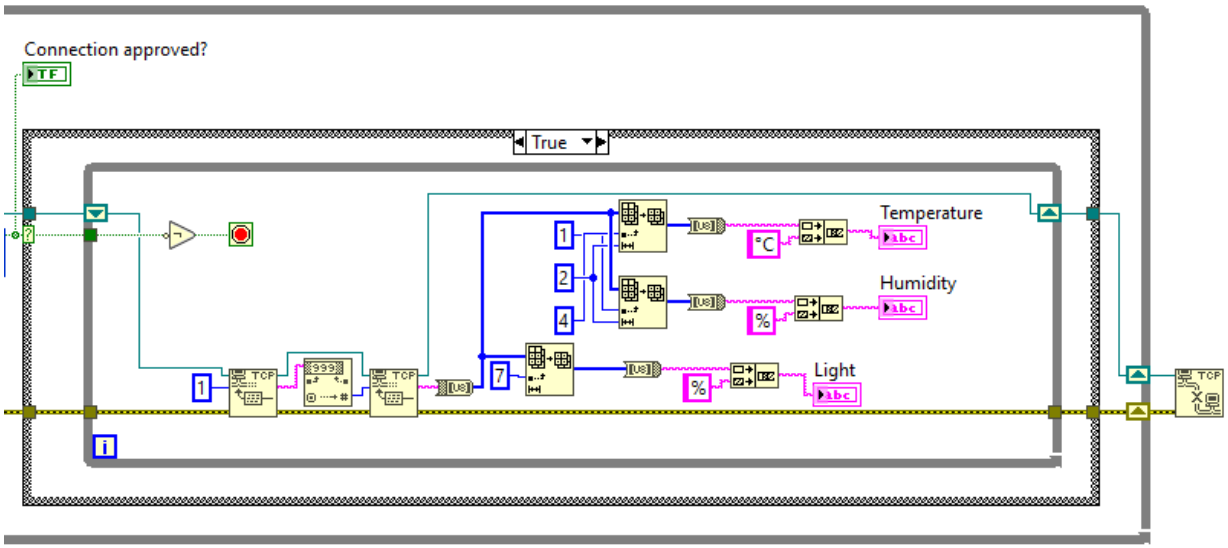

Fig. 8. Partea de procesare a datelor în cazul unui pachet CONNECT corect

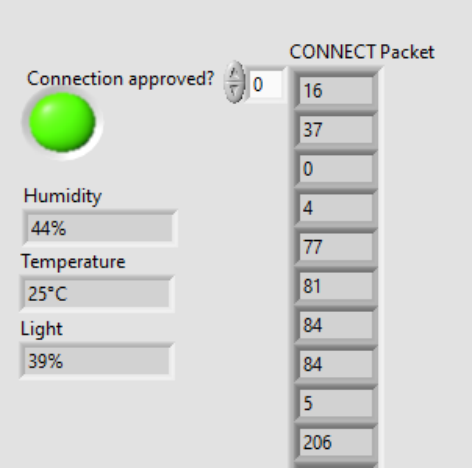

Fig. 9. Interfața cu datele primite de la ESP

4. Concluzii și îmbunătățiri ulterioare

Pentru realizarea acestui sistem s-a plecat de la ideea construirii acestuia utilizând un limbaj și mediu de programare diferite față de ce este obișnuit, întrucât LabVIEW permite vizualizarea datelor procesate și transmise și o interfață cu care se poate lucra direct. De asemenea, la un moment dat, s-a pus problema utilizării unor librării deja existente și, astfel, ar fi rămas de realizat doar conexiunea TCP/IP și managementul conexiunilor și sesiunilor. Cu toate acestea librăriile respective sunt realizate conform protocolului MQTT versiunea 3.1.1, în timp ce această aplicație este realizată conform versiunii 5.0, iar aceste librării puteau fi utilizate atât pentru client, cât și pentru server, doar cu LabVIEW instalat. Se dorește transmiterea datelor la și de la orice fel de dispozitiv, fie unul care rulează cu LabVIEW, fie altul care rulează cu un alt sistem de operare sau este un dispozitiv IoT (Arduino, Raspberry Pi, ESP etc.).

Deopotrivă se poate discuta opțiunea de realizare a unui dispozitiv de măsurare a parametrilor din ambient. Acest dispozitiv poate conține elementele enumerate anterior în cadrul standului experimental, cu excepția că acestea se vor afla pe un PCB (Printed Circuit Board) în loc de breadboard și toate să fie înglobate într-o carcasă realizată prin injecție de material plastic. Astfel acest echipament poate fi realizat în serie pentru a fi utilizat în mai multe incinte din cadrul Universității.

În timpul care urmează se propune realizarea pachetului PUBLISH pentru publicarea datelor către broker și a pachetului SUBSCRIBE pentru citirea datelor de la broker. Acesta este minimul necesar unui sistem IoT cu protocol MQTT, urmând, cu orice posibilitate, a se realiza toate pachetele de control tipice protocolului.

5. Bibliografie

[1]. https://www.postscapes.com/internet-of-things-protocols/
[2]. https://www.elprocus.com/communication-protocols/
[3]. https://en.wikipedia.org/wiki/MQTT
[4]. https://www.fortinet.com/resources/cyberglossary/qos-quality-of-service
[5]. https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html
[6]. https://en.wikipedia.org/wiki/IDEF0
[7]. https://ro.wikipedia.org/wiki/LabVIEW
[8]. https://www.opc-router.com/what-is-mqtt/#iLightbox[feature]/0