الجمهورية الجزائرية الديمقراطية الشعبية

Ministry of Higher Education and Scientific Research

Yahiya Fares University of Medea

Faculty of Technology

Department of Electrical Engineering



Master's Thesis

In Electronics

Specialty: Electronics of Embedded System

Title

Smart Home Monitoring and Control System Based on CAN-BUS and IoT

Presented by

GHERBI Nadia

Supervised and proposed by

Dr .BOUHEDDA Mounir

Academic Year: 2019/2020

Work prepared in the Laboratory of Advanced Electronic Systems (LSEA) of Medea University, ALGERIA

Acknowledgments

I start in the name of the Almighty God, the most Gracious, the most Merciful, for bestowing upon me good health and mental state to carry on this work.
I would like to express my deepest and sincerest gratitude and appreciation to my supervisor, Doctor **Bouhedda Mounir** for giving me the opportunity to work under his supervision. I insist on thanking him for his understanding and support throughout the research process. He has always been by my side whenever I needed him, providing me with his full support, ideas and suggestions to improve my work.

My heartfelt thanks also to our candle teachers of the department, especially those they taught us during the five years, so, I would like to express our sincere appreciation to all of them for their efforts in the classroom.

Thanks for All …

Thanks for him …

Dedication

I dedicate this work to my beloved family

Words are not enough to show my gratitude and love to the “Queen of hearts”, dear mother, for her endless love, sacrifice, support

To my father

To my lovely sisters, Mira and Merieme
To my beloved brothers Abdurrahman, Mohamed Zineeddine, Amir
To my favorite “Houssam Eddine”

To my soul mate Mona who has been helpful, caring, and patient

To my closet friends Hadjer, Manel, Maissa, Loubna

Nadia

العنوان: نظام ذكي للمراقبة والتحكم المنزلي قائم على ناقل الكان وانترنت الاشياء

الملخص

الهدف من هذا المشروع هو ضمان اتصال CAN-bus بين عقدتين أو أكثر، وتحتوي هذه العقد على وحدات CAN-bus أين كل منها متصل بـ Arduino والمتصلة بأجهزة الاستشعار والمشغلات للحصول على تبادل بيانات ثنائي الاتجاه .يتم تحقيق واجهة انسان-الة تحت عقدة IBM الحمراء للإشراف على أجهزة الاستشعار والتحكم في المشغلات التي يتم تنفيذها في نموذج صغير.

Smart home monitoring and control system based on can bus (MCP 2515) and IOT

Abstract:

The objective of this project is to guarantee CAN-BUS communication between two or more nodes, these nodes contain CAN-BUS modules or each one is connected to Arduino which are in turn connected to sensors and actuators to obtain a bidirectional data exchange. A Human Machine Interface is realized under Node-RED from IBM for the supervision of sensors and actuator control implemented in a small prototype.

Keywords: smart home, CAN-BUS, node, communication, Internet of Things (IoT), Arduino, Node-RED.

Système de surveillance et de contrôle de maison intelligent base sur le bus CAN-BUS et IdO

Résume :

L’objectif de ce projet est de garantir la communication CAN-BUS entre deux ou plusieurs nœuds, ces nœuds contiennent des modules CAN-BUS ou chacun est relié à Arduino qui sont de leurs côtés reliés aux capteurs et actionneurs pour obtenir un échange de donnes bidirectionnelles. Une Interface Homme Machine est réalisée sous Node-RED d'IBM pour la supervision des capteurs et contrôle d'actionneurs implémenté dans un petit prototype.

Mots clés: maison intelligents, BUS-CAN, nœud, communication, internet des objets(IdO), Arduino, Node-RED.

LIST OF FIGURES

Chapter I. Presentation of smart home

Figure I.1.General architecture of a smart home.....5
Figure I.2.Smart Home characteristics 6
Figure I.3.Smoke detector.....7
Figure I.4.Carbon monoxide detector 7
Figure I.5.alarmes anti-intrusion 7
Figure I.6.monitoring system in smart home 8
Figure I.7.illustrates comfort in the Smart home 9
Figure I.8.saving energy in smart home 10
Figure I.9.connected home.....11
Figure I.10. Philips Hue bulbs 12
Figure I.11.Ecobee4 thermostat 12
Figure I.12.NetGear Arlo Q Security 13

Chapter II.CAN-BUS overview

Figure II.1.CAN BUS wires involved 16
Figure II.2.CAN output voltages when sending a message 16
Figure II.3.The layered ISO 11898 standard architecture 17
Figure II.4.Standard CAN: 11-Bit Identifier.....18
Figure II.5.Extended CAN: 29-Bit Identifier.....19
Figure II.6.Arbitration field, format standard. 19
Figure II.7.Control field.....20
Figure II.8. CRC field.....20
Figure II.9.bit-stuffing example 21
Figure II.10.CAN-BUS applications 25

Chapter III. Working Environment and Tools

Figure III.1.New dimension for IoT 27

Figure III.2.components of IOT 27
Figure III.3.Used Networks 28
Figure III.4.Gateways 29
Figure III.5.Architecture of IOT 32
Figure. III.6.Applications for the internet of things 33
Figure III.7.Node Red interface 36
Figure III.8.manage palette 36
Figure III.9.install dashboard 37
Figure III.10.bloc of dashboard 38
Figure III.11.Arduino UNO Board 39
Figure III.12.Arduino IDE 41
Figure III.13.MCP 2515 shield 42
Figure III.14.MCP 2515 Microchip 42
Figure III.15.Circuit Diagram for Interfacing MCP2515 with Arduino 43
Figure III.16.other diagram for module can-bus (MCP2515) 44
Figure III.17. Circuit Diagram for Interfacing MCP2515 with Arduino 44

Chapter IV.Practical implementation

Figure IV.1.monitoring and control system 48
Figure IV.2.Temperature and humidity sensor 49
Figure IV.3.mounting principle dht11 49
Figure IV.4.ultrasonic sensor (HC-SR04) 50
Figure IV.5.mounting principle HC-SR04 51
Figure VI.6.the way to get a distance 52
Figure VI.7.LDR sensor 53
Figure IV.8.servo motor 53
Figure IV.9.Mounting principle servo motor 54
Figure IV.10.how does the servo works 54
Figure IV.11. Example of an SPI bus configuration with one master and several 55
Figure VI.12.pin map of CAN BUS Shield V2.0 56
Figure VI.13.connection pins of two modules 57

Figure VI.14.Master/slave communication 58
Figure VI.15.Global architecture of the proposed system 59
Figure IV.16.diagram illustrates the system communication method 60
Figure IV.17.Node-red command on windows 61
Figure IV.18.intallation of dashboard 62
Figure IV.19.Serial palette 63
Figure IV.20.Serial Port Configuration 63
Figure VI.21.interface sensors and actuators with node red 64
Figure IV.22.data separation node 64
Figure IV.23.Configuration of the json function 65
Figure IV.24.Data separation from sensors 65
Figure IV.25.Internal configuration of functions 66
Figure IV.26. Temperature, humidity, distance and light display 66
Figure IV.27. Gauge Configuration 67
Figure IV.28.actuators control 68
Figure IV.29.firmata nodes 68
Figure IV.30.Swich configuration 69
Figure IV.31. Display different parameter of system 69

LIST OF TABLES

Table II.1.Comparison between CAN-BUS and PROFIBUS 23
Table IV.1 characteristics of DHT11 50
Table IV.2 Sensor Pin Configuration 51

Table of contents

General introduction 1

Chapter I. Presentation of smart home

I.1.introduction 3
I.2.History of smart home 3
I.3.Smart Home definition 4
I.4.Smart home technology 5
I.5.Smart Home characteristics 6
I.5.1.Security 6
I.5.2.monitoring 7
I.5.3.Control 8
I.5.4.Communication 8
I.5.5.Comfort 9
I.5.6.Health 10
I.5.7. Energy saving 10
I.6.connected home 11
I.7.Smart home features 11
I.8.Smart home devices 12
I.8.1.Lights (Philips Hue) 12
I.8.2.Thermostat Ecobee4 12
I.8.3.Security NetGear Arlo Q 13
I.9.conclusion 13

Chapter II.CAN-BUS overview

II.1.introduction 14
II.2.Communication protocols 14
II.2.1.Wireless network protocol 14
II.2.2.wired networks protocol 14
II.3.Serial and parallel links 14
II.4.Industrial Local Area Network 15
II.5.histery 15
II.6.definition 15

II.7.wires used in CAN-BUS 16
II.8.The CAN Standard.....16
II.9.Standard CAN or Extended CAN 17
II.9.1.The Bit Fields of Standard CAN and Extended CAN 18
II.9.1.1.Standard CAN 18
II.9.1.2.Extended CAN 19
II.10.Format of data frames 19
II.10.1.Start of frame.....19
II.10.2.Arbitration field 19
II.10.3.Control field 20
II.10.4.Data field.....20
II.10.5.CRC field 20
II.10.6.ACK Field 20
II.10.7.End of frame.....21
II.11. Bit-stuffing.....21
II.12.Error detection and management 21
II.13.Types of errors 21
II.14.Error frames 22
II.15.comparison between can bus and profibus 22
II.16.Why use CAN BUS protocol?.....23
II.17.CAN-BUS protocol application.....24
II.18.CAN-BUS uses in smart home 25
II.19.conclusion.....25

Chapter III. Working Environment and Tools

III.1.Introduction 26
III.2.Internet of Things 26
III.2.1.Definition..... 26
III.2.2.Components of internet of things..... 27
III.2.2.1.Smart devices and sensors – Device connectivity 28
III.2.2.2.Gateway.....29
III.2.2.3.Cloud.....29
III.2.2.4.Analytics.....30

III.2.2.5.User interface.....30
III.2.3.The architecture of internet of things: 31
III.2.3.1.Three layered architecture.....31
III.2.4.Applications of internet of things:.....32
III.2.5.adventages and disadvantages of internet of things 33
III.2.5.1.IOT Advantages.....33
III.2.5.2.IoT Disadvantages 34
III.2.6.IoT in Homes 35
III.3.Node-Red 35
III.4.Arduino 39
III.4.1.ARDUINO BOARD.....39
III.4.4.ARDUINO UNO.....39
III.4.2.ARDUINO SOFTWARE 40
III.5.MCP 2515 CAN-BUS shield 42
III.5.1.Definition.....42
III.5.2.CAN-BUS shield features.....46
III.5.Conclusion.....46

Chapter IV.Practical implementation

IV.1.Introduction.....47
IV.2.Logbook.....47
IV.3.General operation of monitoring and control system proposed.....47
VI.4.Human machine interface 48
IV.5.Presentation of sensors and actuators used.....48
IV.5.1.Sensors.....48
IV.5.1.1.DHT11 Temperature and Humidity Sensor 48
IV.5.1.2.Ultrasonic sensor 50
IV.5.1.3.LDR 53
IV.5.2.Actuators.....53
IV.5.2.1.Servo motor.....53
IV.6.Communication with the Module MCP2515 55
IV.6.1.SPI Link.....55
IV.6.2.pin map of CAN BUS Shield V2.0 56

TABLE OF CONTENTS

IV.7.Communication 57
IV.7.1.Transmission with MCP2515: 57
IV.8.Communication Protocol 58
IV.9.Operation of the global system.....58
VI.10.how communication occurs in control and monitoring system 59
VI.11.Human Machine Interface (HMI).....61
VI.11.1. Programming with Node Red 61
VI.11.1. 1.NODE RED INSTALLATION 61
VI.11.1.2.Installation of new NODE.....62
VI.11.1.3.Flow node red/actuators and flow sensor/node red 63
VI.11.1.4.data splitting 64
IV.11.2.data visualization.....66
IV.12.1.temperature, humidity, distance and light display 66
IV.12.2.Display of controls for actuators.....67
IV.12.3.Visualisation d'IHM.....69
IV.13.conclusion 70

General Conclusion……. 71

REFERENCES…… 72

General Introduction

General Introduction

Nowadays, the programmed electronics replace conventional electronics, it is known as embedded systems, and these systems have become more complex and efficient over time.

The development of technology and lifestyles concerns all sectors especially in buildings (homes, administrations, hotels, hospitals …) to realize the meaning of a smart city, homeowner take advantage of this improvement in quality of life particularly for the elderly and disabled persons.

The combination of different technologies (electronic, telecommunication, automatic, and informatics) allows the optimization of comfort, security, and management of a home, furthermore, the abundance of microcontrollers, sensors and their low prices, the ease of controlling them with smartphone or tablet, all of this has reduces the cost of smart home.

The large number of suppliers and standardization of equipment, allows any home to become a smart home, it includes several systems to make life enjoyable and to optimize electrical energy signals.

Standard communication protocols such as UART, I2C, SPI, are unreliable than other communication protocols such as CAN, LIN and FlexRay [1].

The CAN protocol describes precisely and in detail the connection of several devices to a bus, it is a very common bus in the industry. CAN-BUS use mainly developed for the automotive sector to ensure communication between components and their control unit, it is easy to interface with any microcontroller having SPI interface [2]. The protocol mainly sets the priority of access to the bus and solves the collision problem at the hardware level in case several devices start to communicate even time,

Our objective is to propose a CAN-BUS Communication Protocol for Control and monitoring home, based on the Internet of Things, this system is made up of a supervision and control interface intelligent.

The Internet of Things (IoT) consists of devices that generate, process, and exchange vast amounts of security and privacy-sensitive data and information, with using the internet of things you will be able to automates and assign domestic tasks with a simple tap on your smart home.

Communication between the different hardware parts is carried out through the USB serial port and MCP2515 wired communication, we use arduino type microcontrollers to collect data and send actions, as well as the human-machine interface used for the execution of the intelligent control and monitoring (HMI) algorithm created under the Node-Red platform

The remaining of this work is organized as follows:

In chapter I we briefly discuss general information on smart homes, its history, technology, characteristics and features.

In chapter II we outlines the CAN-BUS protocol.

In chapter III we present the work tools and the environment in which the proposed realization has been implemented.

In chapter IV we describe our system development by addressing both dimensions: hardware implementation and program development.

And we end our work with a general conclusion.

Chapter I: Presentation of Smart Home

I.1.introduction

The market of smart homes has witnessed a great development in these past few years due to the facilitations of equipment installation without harming or touching the infrastructures of the home. This equipment are all connected to internet based on complex development environment and specific interfaces.

Now, it's possible to create your own scenario about the home, starting from the light, heating and cooling system, meaning; the automation of your daily activities according to your life and people close to you.

A smartphone or tablet are used to control the entire system of the home, the orders are analyzed to get the necessary information about the residents of and his there to create the home's system, with putting in consideration the priorities like the fire.

An intelligent home is the best way to facilitate and upgrade a human's life, with a distance control it makes the far close, humans are motivated to use it especially for saving energy, comfort and security.

I.2.History of smart home

The first smart homes were ideas only and not structures the idea of home automation was just to imagine, like the short story of Ray Bradbury, "there is a light rain coming", imagine a future in which homes are interactive and self-managing and describe the home that continues to work even after humans die.

- Although the idea of smart home has existed for some time, actually it existed only for a short time. This timeline focuses on the actual inventions that led to the smart homes we know today and can expect from the near future.
- **1901 – 1920 - The invention of home appliances** – Although home appliances aren't what we'd consider "smart," they were an incredible achievement in the early twentieth century. These achievements began with the first engine-powered vacuum cleaner in 1901. A more practical electricity-powered vacuum was invented in 1907. Throughout two decades refrigerators would be invented, as well as clothes dryers, washing machines, irons, toasters, and so much more.
- **1966 – 1967 - ECHO IV and the Kitchen Computer** – Although it was never commercially sold, the ECHO IV was the first smart device. This clever device could compute shopping lists, control the home's temperature and turn appliances on and off. The Kitchen Computer, developed a year later, could store recipes, but had the unfortunate tagline, "If she can only cook as well as Honeywell can computer" and therefore sold no models.

- **1991- Gerontechnology** – Gerontechnology combines gerontology and technology and makes the lives of senior citizens easier. In the 1990s, there was a lot of new research and technology in this sector.
- **1998 – Early 2000s: Smart Homes** – Smart homes, or home automation, began to increase in popularity in the early 2000s. As such, different technology began to emerge. Smart homes suddenly became a more affordable option, and therefore a viable technology for consumers. Domestic technologies, home networking, and other gadgets began to appear on store shelves.
- **Today's Smart Homes** - Today's smart homes are more secure and living greener. Our smart homes are sustainable, and they help to ensure that our homes aren't expending unnecessary energy. They also help alert us to intruders (whether we're home or not).
- Current trends in home automation include remote mobile control, automated lights, automated thermostat adjustment, scheduling appliances, mobile/email/text notifications, and remote video monitoring[3].

I.3.Smart Home definition

A smart home, or smart house, is a home that incorporates advanced automation systems to provide the inhabitants with sophisticated monitoring and control over the building's functions. For example a smart home may control lighting, temperature, multi-media, security, window and door operations, as well as many other functions.

In 2003 the UK Department of Trade and Industry (DTI) came up with the following definition of a smart home:
"A dwelling incorporating a communications network that connects the key electrical appliances and services, and allows them to be remotely controlled, monitored or accessed".

Smart homes use 'home automation' technologies to provide homeowners with 'intelligent' feedback and information by monitoring many aspects of a home. For example, a smart home's refrigerator may be able to catalogue its contents, suggest menus, recommend healthy alternatives, and order replacements as food is used up. A smart home might even take care of feeding the cat and watering the plants.

Many new homes are being built with the additional wiring and controls which are required to run advanced home automation systems. Retrofitting (adding smart home technologies to an existing property) a house to make it a smart home is obviously significantly more costly than adding the required technologies to a new home due to the complications of routing wires and placing sensors in appropriate places [4].

I.4.Smart home technology

Smart home technology has covered up all the necessities of life as the (figure I.1) at the bottom shows. From the dedicated, smart home appliances to smart thermostats, smart lights, smart doorbells, smart locks, and even smart washers and dryers, you can leverage this technology for added benefits. Keeping in view their extensive usage, these smart devices are being designed in a way to consume less energy. As they would be more reliant on the internet, you would definitely notice a reduction in energy consumption at your home [5].



Figure I.1. General architecture of a smart home[6]

I.5.Smart Home characteristics

The Smart Home uses several key characteristics: security (alarms, cameras and remote monitoring), comforts of life (automation and programming of daily tasks), saving energy (heating, light), health (telehealth, telemedicine) and communication (with a network, Bluetooth, wifi etc. ...) (see figure I.2).



Figure I.2. Smart Home characteristics

I.5.1.Security

One of the fields of application of smart home is the safety of goods and people through alarm systems which, on the one hand, prevent technical risks (breakdowns or device malfunctions) and on the other hand possible intrusions into the home (burglary) In general we find:

- **Technical alarms**

  Technical alarms are based on sensors capable of detecting various incidents such as fire (figure I.3) and toxic releases (figure I.4), water leaks, gas leaks, etc. These different sensors are connected to an alarm center.



Figure I.3. Smoke detector



Figure I.4.carbon monoxide detector

- **Anti-intrusion alarms**

These are generally sensors on the doors (opening detection) or in rooms (presence detection) which are also connected to an alarm center. These sensors can be coupled with a network of digital surveillance cameras. During an intrusion, an alert message can be sent by e-mail or on a mobile phone (see figure I.5).



Figure I.5.alarmes anti-intrusion

I.5.2.monitoring

Smart Home Monitoring allows you to stay connected at multiple aspects of your home no matter where you are. You can keep an eye on your home from anywhere, monitoring systems allow you to control your home's security system, lighting, temperature and more remotely with an Internet-connected device such as a laptop, tablet, smart phone and even a smart watch such as the Apple Watch!

This sort of technology allows you to control your home from any location at any time. Smart home monitoring means safety, security, convenience.



Figure I.6.monitoring system in smart home

You can check your system's status over the Internet and on mobile devices. Lighting, thermostat and lock control technology allows your home to be at your fingertips. Minimal wiring is needed for additional modules when using the private WI-Fi network. By integrating your heating, lighting, air conditioning, alarm, video, internet and telephone systems together you allow everything to be controlled by one centralized software system. Smart homes are incredibly powerful ways to manage your property(figure I.6) [7].

I.5.3.Control

When you leave a home or business, the alarm will activate the following a series of controls and actions, (centralization of commands):

- Closing of all the lights, cutting off the gas supply,
- Checking that all windows are closed,
- Turning on the outside light for a few minutes if it is dark...

At the touch of a button, all the lights in the living room will be adjusted for the dinner, a television evening or the creation of a luminous atmosphere adapted to the occupant's activity.

I.5.4.Communication

Smart home communication is a fusion of information, communication technologies and electronics. In the field of home automation standards, there are different types of connections in a smart home:

- io-homecontrol is a wireless and secure radio technology, shared by specialists habitat with two-way communication;

- Bluetooth, radio protocol allowing transparent communication between all equipment located within a perimeter of a few meters;
- DSP (Digital Signal Processor) used in home theater amplifiers to manage the broadcasting of the sound signal to the speakers of the system (audio home automation);
- xPL: ultimate home automation management protocol (free, simple and documented) to make communicate all the equipment of the installation;
- Peer-to-peer (P2P), data exchange between two computers connected to the Internet. Establish a direct link between the two machines without requiring a central server;
- Ethernet, communication protocol allowing the transport of information on a network computer science;
- ZigBee, a high level protocol allowing the communication of small radios, reduced consumption for personal networks.

I.5.5.Comfort

Today, a Smart Home is able to know when you come home (using your Smartphone for example), and therefore is open the gate even before you were not happening. The shutters can open and close to the rhythm of the sun, and can even go until it adapts to the season and the temperature to let in the light and the heat of the sun, winter, or on the contrary keep cool in summer by closing the shutters on windows exposed to the sun.

Likewise, your home knows when you are and can therefore adapt your heating so that it is always at the ideal temperature for you. It is even possible automatically Play your favorite music list when you wake up or when you get home. All of this is not science fiction: It is completely achievable today (see figure I.7).



Figure I.7.illustrates comfort in the Smart home

I.5.6.Health

The Smart Home is now finding new applications in the health sector. In installing systems in the homes of people with disabilities, suffering from neurodegenerative diseases such as Alzheimer's disease or even the elderly, it is possible to help them in their daily life by automating as much as possible of the tasks considered as complexes.

It also allows the person to stay at home longer and to be followed distantly. For example, we can detect when a person is not drinking enough of water or when she forgets to feed. If the behavior is considered “worrying”, it is then possible to alert the family or the emergency services according to the scenarios programmed in the command interface.

I.5.7. Energy saving

By managing the shutters according to the season, as well as the heating, the smart home system allows you to save energy, and therefore money, even if initially we were only looking for comfort more. Energy consumption can be tracked very finely, whether it is your consumption of electricity, water, or even gas.

The simple act of activating the alarm when leaving will switch the heating to eco mode as (figure I.8) illustrate, and turn off all lamps and appliances left on standby, reducing your energy consumption in your absence. And this without any action on your part[8].



Figure I.8.saving energy in smart home [9]

I.6.connected home

A connected home is a house in which windows, doors, lighting, heating, alarm and many other elements can be controlled remotely using a smartphone. To say that a home is connected, there must be connected objects. It also involves the installation of a home automation system as well as the mobile applications of a smartphone or tablet (figure I.9) [10].



Figure I.9.connected home

I.7.Smart home features

There are some main features of smart homes:

- The smart home can realize the interaction between the user and the power grid enterprise, get the information of electricity consumption and electricity price, set the electricity consumption plan and so on, guide the scientific and rational electricity use and advocate the family's consciousness of energy saving and environmental protection.
- Smart home can enhance the comfort, safety, convenience and interactivity of home life, and optimize people's lifestyle.
- Smart home can support remote payment.
- Smart home can monitor and interact with the home through telephone, mobile phone and remote network, discover the abnormal and timely processing.
- The smart home realizes the real-time meter reading and security service of water meter, electric energy meter and gas meter, which provide more convenient conditions for the high-quality service.
- Support "triple networks" business and the perfect intelligent service [11].

I.8.Smart home devices

Today, smart home systems and devices are more impressive and all-encompassing than ever. Following are some examples.

I.8.1.Lights (Philips Hue)

Philips Hue bulbs enable you to control the intensity of the light - dimming or brightness on request - and the color of the lights. You can create special moods with consistent color, these lights work with most smart home systems (figure I.10).



Figure I.10. Philips Hue bulbs

I.8.2.Thermostat Ecobee4

The Ecobee4 allows you to control air temperature with voice commands, but unlike other smart thermostats, it also works as its own Amazon speaker .Ecobee4 integrates seamlessly with apps and other home ecosystems like Alexa or Apple Home Kit.

Plus, it’s able to control the room you are in, rather than the room it’s installed.



Figure I.11. Ecobee4 thermostat

I.8.3.Security NetGear Arlo Q

The NetGear Arlo Q is a truly useful tool for home security. The NetGear Arlo Q records high-quality video and audio, and produces exceptional quality images of people in pitch-blackness. It became a worthwhile investment compared to some of the other smart security systems [12].



Figure I.12. NetGear Arlo Q Security

I.9.conclusion

Smart home is so vast and riche domain, its concept is born and extended in the quotidian. In this chapter we will talk about the profound impact of a smart home in daily life and activities, the inclusion of IOT, the different ways of communication, the significant number of (sensors, actuators, microcontrollers), and the work done on the development environments, all this results the creation of an intelligent home, to assure the comfort of the habitant, economy of energy and safety.

Chapter II CAN-BUS Overview

II.1.introduction

Today there are more than fifty different specifications of Field networks, they found an exchange of information between different electronic control units, including the CAN-BUS.

The CAN (Controller Area Network) is a serial communication bus that is developed at the end of the development process of the 80's by the German company Robert Bosch. The objective was to provide the industry with an automobile bus little expensive for the embedded electronics of automobiles, such as alternative to the bulky and complex wires models of the time, the efficiency and robustness of this protocol has being used in many different countries. Many other industrial applications, especially those that do not require a large bandwidth up to 1 Mbits/s with a very low error rate.

The CAN-bus is leader in Automative industry and machinery, agriculture, as well as regarding medical equipment. However in application areas, especially in the field of automation of small buildings. The most important thing that distinguishes it is that the protocol fundamentally determines the priority of access to the bus and solves the collision problem at the hardware level in case of many devices start communicating at the same time.

II.2.Communication protocols

II.2.1.Wireless network protocol

This includes radio communications (cellular, Bluetooth, WiFi, SigFox, LoRaWAN...) but also luminous communications (the Li-Fi protocol uses frequencies of the light spectrum to transmit information). This type of communication is used in hospital environments because it avoids the nuisance caused by radio communications such as WiFi.

II.2.2.wired networks protocol

Wireless protocols are often supported by a fieldbus that extends the overall capabilities of the installation.

Decentralized protocols where sensors and actuators communicate directly with each other without a central point. Each approach has its advantages and disadvantages.

We find Modbus, PROFIBUS and the CAN-BUS [13].

II.3.Serial and parallel links

Digital signal transmissions can be performed in two ways: serial or parallel depending on whether they are done through one or more conductors. For a long time, the serial link was the most used because it only required a single conductor. Then in the 1980s, parallel transmission dethroned the serial link wherever it was possible to multiply the conductors to get wider communication channels.

It is effective over short distances and it is this principle which has been retained for the buses on the motherboard such as the FSB or the PCI bus.

This is also the case for IDE cables, for exchanging data with disks and for the parallel port intended for the printer. Serial communications were less efficient for this kind of fast and short distance links. Currently, parallel transmissions are in turn denigrated because of problems of electromagnetic interference between conductors arranged side by side. And the solution is none other than a return to the serial link, adapted it is true to be able to increase the transmission speeds without interfering or being subject to it. The serial link is made by two paired cables which transmit symmetrical signals. We will come back to that, speaking of SATA cables, USB ports or PCI express. Let's start by describing what serial and parallel transmissions are in principle [14].

II.4.Industrial Local Area Network

An industrial local area network is in first approximation a local area network used in a factory or any production system to connect various machines in order to ensure the control, monitoring, supervision, control, maintenance, product monitoring, management, in a word, the operation of the production facility; means in recent years the use of a system called SCADA (Supervisory Control and Data Acquisition)[15].

Below, we will give you an overview of a local network Industrial is the CAN-BUS network.

II.5.histery

The CAN data bus is a collaboration between the University of Karlsruhe and Bosch. It was first used in the automotive sector, but is currently used in most industries such as aerospace via standardized CAN-based protocols.

It was introduced with Intel in 1985, but was not standardized by ISO until early 1991. In 1992 several companies came together to create CAN in Automation, an association to promote CAN.

II.6.definition

A Controller Area Network is a serial communication bus designed for industrial and automotive applications. It is a message-based protocol used for communication between multiple devices. When multiple CAN devices are connected together like shown below, the connection forms a network acting like our central nervous system allowing any device to speak with any other device in the node[16].

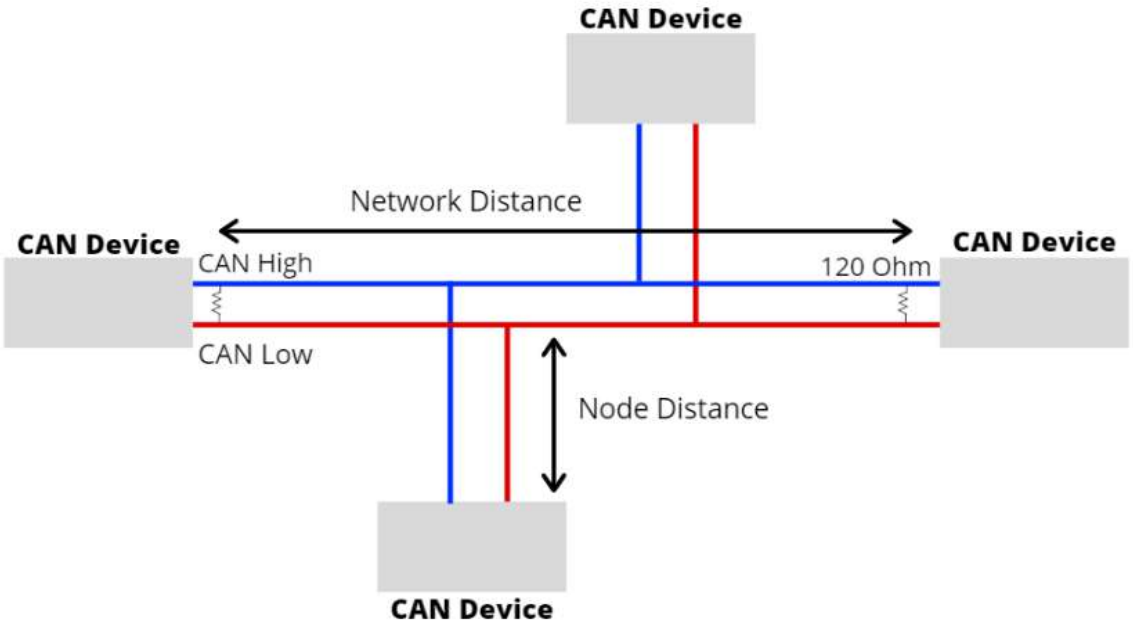


Figure II.1.CAN BUS wires involved

II.7.wires used in CAN-BUS

- The wires act as a differential line which means the CAN signal either 0 or 1, can be represented by the potential difference between the two wires For example, if the difference is positive and large than a certain minimum voltage = 1. If the difference is negative = 0
- For CAN termination, as you can see from the picture above, a single 120 Ohm is generally used at the 2 ends of the CAN network.

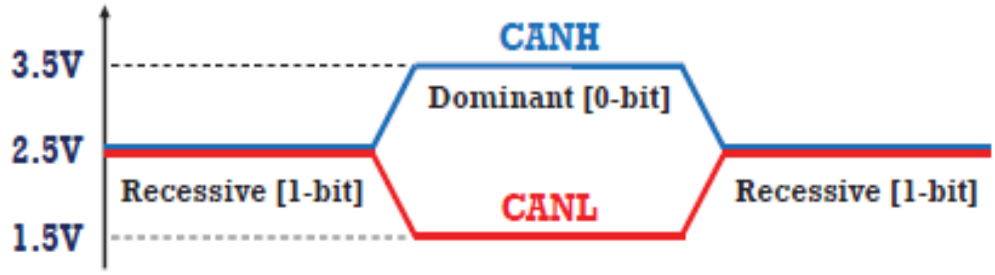


Figure II.2.CAN output voltages when sending a message[17]

II.8.The CAN Standard

CAN is an International Standardization Organization (ISO) defined serial communications bus originally developed for the automotive industry to replace the complex wiring harness with a two-wire bus. The specification calls for high immunity to electrical interference and the ability to self-diagnose and repair data errors.

These features have led to CAN's popularity in a variety of industries including building automation, medical, and manufacturing.

The CAN communications protocol, ISO-11898: 2003, describes how information is passed between devices on a network and conforms to the Open Systems Interconnection (OSI) model that is defined in terms of layers. Actual communication between devices connected by the physical medium is defined by the physical layer of the model. The ISO 11898 architecture defines the lowest two layers of the seven layer OSI/ISO model as the data-link layer and physical layer in (Figure II.3).

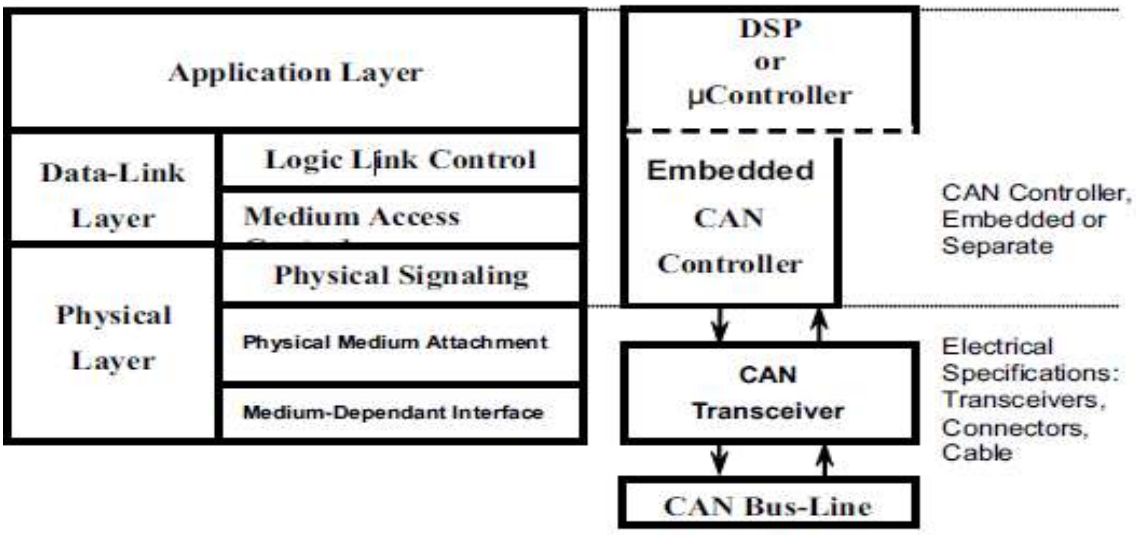


Figure II.3. The layered ISO 11898 standard architecture

In Figure II.3, the application layer establishes the communication link to an upper-level application specific protocol such as the vendor-independent CAN open protocol. This protocol is supported by the international users and manufacturers group, CAN in Automation (CIA). Additional CAN information is located at the CIA Web site, can-cia.de. Many protocols are dedicated to particular applications like industrial automation, diesel engines, or aviation. Other examples of industry-standard, CAN-based protocols are KVASER's CAN Kingdom and Rockwell Automation's Device Net.

II.9.Standard CAN or Extended CAN

The CAN communication protocol is a carrier-sense, multiple-access protocol with collision detection and arbitration on message priority (CSMA/CD+AMP). CSMA means that each node on a bus must wait for a prescribed period of inactivity before attempting to send a message. CD+AMP means that collisions are resolved through a bit-wise arbitration, based on a preprogrammed priority of each message in the identifier field of a message. The higher priority identifier always wins bus access. That is, the last logic high in the identifier keeps on transmitting because it is the highest priority. Since every node on a bus takes part in writing every bit "as it is being written," an arbitrating node knows if it placed the logic-high bit on the bus.

The ISO-11898:2003 Standard, with the standard 11-bit identifier, provides for signaling rates from 125 kbps to 1 Mbps. The standard was later amended with the "extended" 29-bit identifier. The standard 11-bit identifier field in (Figure II.4) provides for 211, or 2048 different message identifiers, whereas the extended 29-bit identifier in (Figure II.5) provides for 229, or 537 million identifiers.

II.9.1.The Bit Fields of Standard CAN and Extended CAN

II.9.1.1.Standard CAN



Figure II.4. Standard CAN: 11-Bit Identifier

The meaning of the bit fields of Figure I.10 are:

- SOF–The single dominant start of frame (SOF) bit marks the start of a message, and is used to synchronize the nodes on a bus after being idle.
- Identifier-The Standard CAN 11-bit identifier establishes the priority of the message. The lower the binary value, the higher its priority.
- RTR–The single remote transmission request (RTR) bit is dominant when information is required from another node. All nodes receive the request, but the identifier determines the specified node. The responding data is also received by all nodes and used by any node interested. In this way, all data being used in a system is uniform.
- IDE–A dominant single identifier extension (IDE) bit means that a standard CAN identifier with no extension is being transmitted.
- R0–Reserved bit (for possible use by future standard amendment).
- DLC–The 4-bit data length code (DLC) contains the number of bytes of data being transmitted.
- Data–Up to 64 bits of application data may be transmitted.
- CRC–The 16-bit (15 bits plus delimiter) cyclic redundancy check (CRC) contains the checksum (number of bits transmitted) of the preceding application data for error detection.
- ACK–Every node receiving an accurate message overwrites this recessive bit in the original message with a dominate bit, indicating an error-free message has been sent. Should a receiving node detect an error and leave this bit recessive, it discards the message and the sending node repeats the message after rearbitration. In this way, each node acknowledges (ACK) the integrity of its data. ACK is 2 bits, one is the acknowledgment bit and the second is a delimiter.

- EOF–This end-of-frame (EOF), 7-bit field marks the end of a CAN frame (message) and disables bit stuffing, indicating a stuffing error when dominant. When 5 bits of the same logic level occur in succession during normal operation, a bit of the opposite logic level is *stuffed* into the data.
- IFS–This 7-bit inter frame space (IFS) contains the time required by the controller to move a correctly received frame to its proper position in a message buffer area.

II.9.1.2.Extended CAN

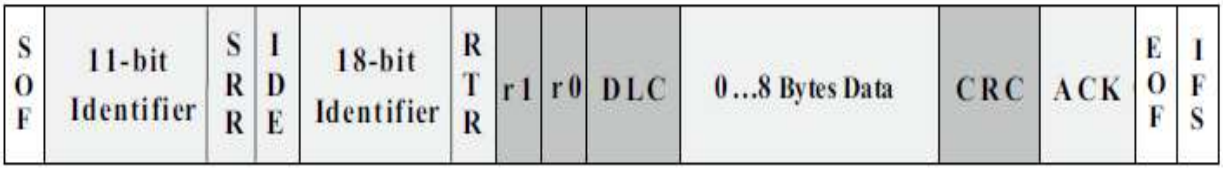


Figure II.5. Extended CAN: 29-Bit Identifier

As shown in (Figure II.5), the Extended CAN message is the same as the Standard message with the addition of:

• SRR–The substitute remote request (SRR) bit replaces the RTR bit in the standard message location as a placeholder in the extended format.

• IDE–A recessive bit in the identifier extension (IDE) indicates that more identifier bits follow. The 18-bit extension follows IDE.

• r1–Following the RTR and r0 bits, an additional reserve bit has been included ahead of the DLC bit[18] .

II.10.Format of data frames

The Data Frames are composed of 7 parts and are described below.
The format is specified e for frames that respect protocol 2.0A. It is stated a for 2.0B frame format.

II.10.1.Start of frame

The Start of frame (SOF) bit marks the start of a data frame or remote frame. It is a single dominant bit. A node can only transmit when the bus is free. Then, all the other nodes synchronize to the SOF of the node that started the transmission.

II.10.2.Arbitration field

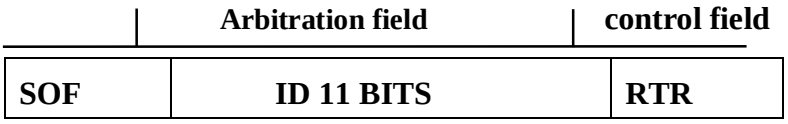


Figure II.6. Arbitration field, format standard.

The Arbitration field consists of the identifier and the bit RTR.
The identifier (ID) identifies the message. It is transmitted in the order ID10 at ID0, where ID0 is the least significant bit. The RTR (Remote Transmission Request) bit features Remote Frames. It is dominant in Data Frames and recessive in Remote Frames.

II.10.3.Control field

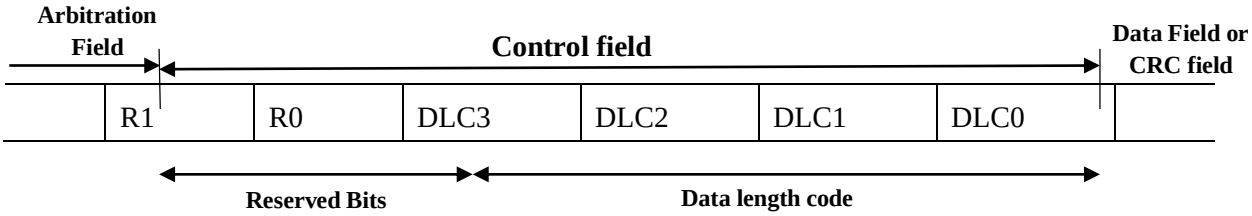


Figure II.7. Control field

The Control field is composed of 6 bits. The first 2 are reserves bits and the next 4 are the Data length code (DLC). The DLC indicates the number of bytes in the Data field. The DLC value is necessarily between 0 and 8, or 9 values. 4 dominant bits (0000) correspond to the value 0 for the DLC, while 1 bit recessive and 3 bits dominant (1000) correspond to the value 8.

II.10.4.Data field

This is the data transmitted by the Data frame. It can contain from 0 to 8 bytes, où each byte is passed with the weight bit first.

II.10.5.CRC field

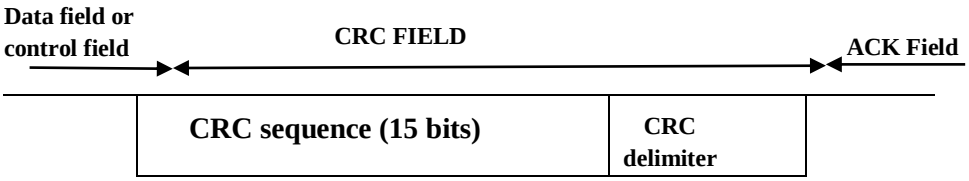


Figure II.8. CRC field

The CRC field is composed of the CRC 15-bit sequence followed by the CRC delimiter (1 bit recessive). CRC (Cyclic redundancy code) enables the integration of data The bits used in the CRC calculation are those of SOF, Arbitration field, Control field and data field.

II.10.6.ACK Field

The ACK field is composed of 2 bits, the ACK Slot and the ACK Delimiter (1 bit recessive). The node transmitting sends a bit recessive for the ACK Slot. A node having received correctly the message informs the transmitter by sending a dominant bit during the ACK Slot: It acknowledges the message.

II.10.7.End of frame

Each Data frame and Remote frame is terminated by a 7 bit scale.

II.11. Bit-stuffing

For Data Frames and Remote Frames, the bits from the Start of frame to the sequence are coded according to the bit stuffing method. When a transmitter detects 5 consecutive bits of the same value in the bits to be transmitted, it automatically adds a bit of opposite value.

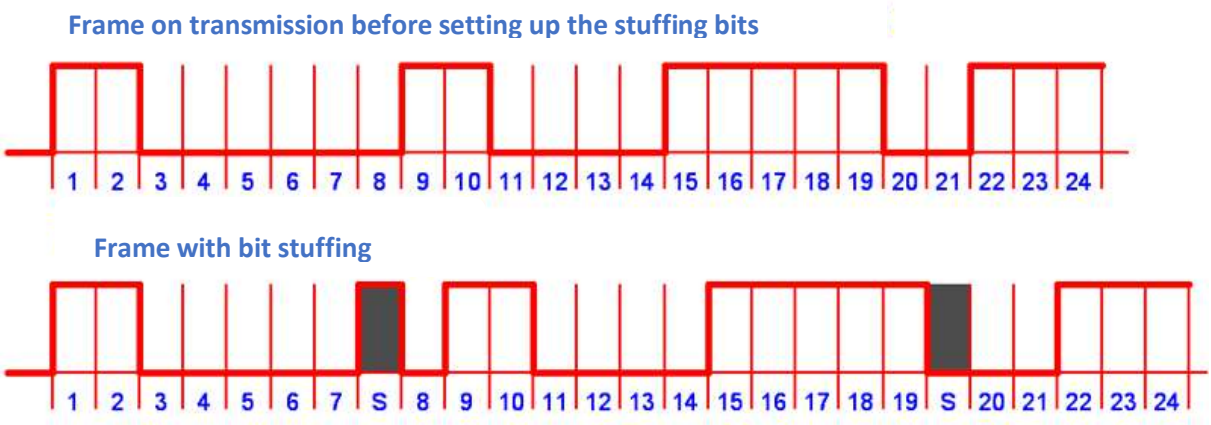


Figure II.9.bit-stuffing example [19]

II.12.Error detection and management

The CAN was created to operate in harsh environments and therefore it is designed to include many error detection mechanisms.

II.13.Types of errors

The CAN implements five error detection mechanisms, 2 at the bit level (the "bit") and 2 at the bit level (the "bit").monitoring and bit stuffing) and 3 at the message level (verification of CRC, the form of the frames and acquittal).

These five different types of errors that can be detected by a node are:

- Bit error: A node sending a bit on the bus also looks at the bits it recoils at the same time (Bit monitoring). It considers a bit error when the bit sent is different bit re cu, with the exception of sending a recessive bit during arbitration (case of loss arbitration) or during the ACK Slot (paid frame).
- Stuff error: The node detects a stuffing error when it recovers 6 consecutive bits of the same value in a part of a message that should be coded with the method of bit stuffing.
- CRC error: A CRC error is detected when the CRC computes by a receiver is different of the CRC value in the frame.
- Form error: A Form error is detected when a bit that should be `a 'has a certain value a different value (a delimiter for example).

- ACK error: The transmitter detects an acknowledgment error when it does not have a dominant bit during the ACK Slot.

II.14.Error frames

An error frame consists of two parts. The first one is formed by superimposing different "Error flags" put by the nodes of the bus. The second part is an eliminator.

A node that detects an error signals it by sending an Error flag. This flag violates the bit stuffing rule (6 consecutive dominant bits) and consequently all other nodes also detect an error and start sending an Error flag the dominant bit sequence which then exists on the bus is the result of the superposition of several Error flags, and its length varies between 6 and 12 bits.

There are two types of Error flags:

- Active error flag: 6 consecutive dominant bits.
- Passive error flag: 6 consecutive recessive bits, until they are overwritten by bits dominant.

The Error delimiter is composed of 8 recessive bits. In fact, after transmitting its Error flag, each node sends recessive bits and observes the bus until it detects a bit. Recessive, after which it sends another 7 additional recessive bits [20].

II.15.comparison between can bus and profibus

In the next table we have made a comparison between the communication protocol (CAN-BUS and PROFIBUS) we notice that the CAN-BUS is more efficient than the other.

Table II.1. Comparison between CAN-BUS and PROFIBUS [20]

	CAN-BUS	**PROFIBUS**
Protocol name	Control area network	Process field bus
Application field	Industrials Home automation	industrials
topology	BUS	BUS
max length (m)	40	1200
for a throughput of(Mb)	1	
Layer ISO/OSI	1and 2	1,2,7
Max flow (Kb)	1000	9.6
Min flow(Kb)		500
Max number of bytes of the message	8+(2)	246+(2)
priority	2048	2
manufacturers	MOTOROLA TOSHIBA	SIEMENS

II.16.Why use CAN BUS protocol?

Using the CAN BUS protocol has several advantages and benefits:

- **Low Cost**
  - ECUs communicates via a single CAN interface compared to, for example, direct analog signal lines which reduce errors, weight, and costs.
  - With its multiplex wiring that combines analog and digital signals and their transmission over a shared medium, it reduces the amount of wiring needed.
- **Centralized**
  - As CAN BUS supports centralized control over electronic devices that are connected to the network, it allows for central error diagnosis and configuration across all ECUs
  - Error handling is also built into the CAN protocol where nodes can check for errors in transmission while maintaining its own error counter. For example, the protocol supports different error detection capabilities such as bit error, ack error, form error, CRC error, etc.

- **Flexible**

- As each CAN connected ECU can receive all transmitted messages, it can also decide whether it is relevant and act accordingly.
- The CAN BUS protocol is also a message-based communication protocol where nodes on the bus have no identifying information.
- With the above features, nodes can easily be added or remove and modified.
- For beginners, it will be easy to integrate new electronic devices into the CAN BUS network without any significant programming overhead.

- **Robust**

- When choosing a communication protocol, durability and reliability are very important. You would want your communication protocol to be self-sustaining and durable for a long period of time without maintenance.
- With the CAN BUS, the system is robust towards electric disturbances and electromagnetic interference which makes it ideal for vehicles.

- **Efficient**

- CAN messages frames are prioritized by ID where the top priority will get bus access and yet frames would not be interrupted.
- Due to flash programming, it also saves time together with less and simple wiring.[20]

II.17.CAN-BUS protocol application

The applications of can bus are (figure II.10):

- Industrial automation
- Home/Building automation
- Automotive (VL, PL)
- Transportation (Train, aviation, ...)
- Matériel agricole, Travaux public
- Maritime
- Medical
- Instrumentation

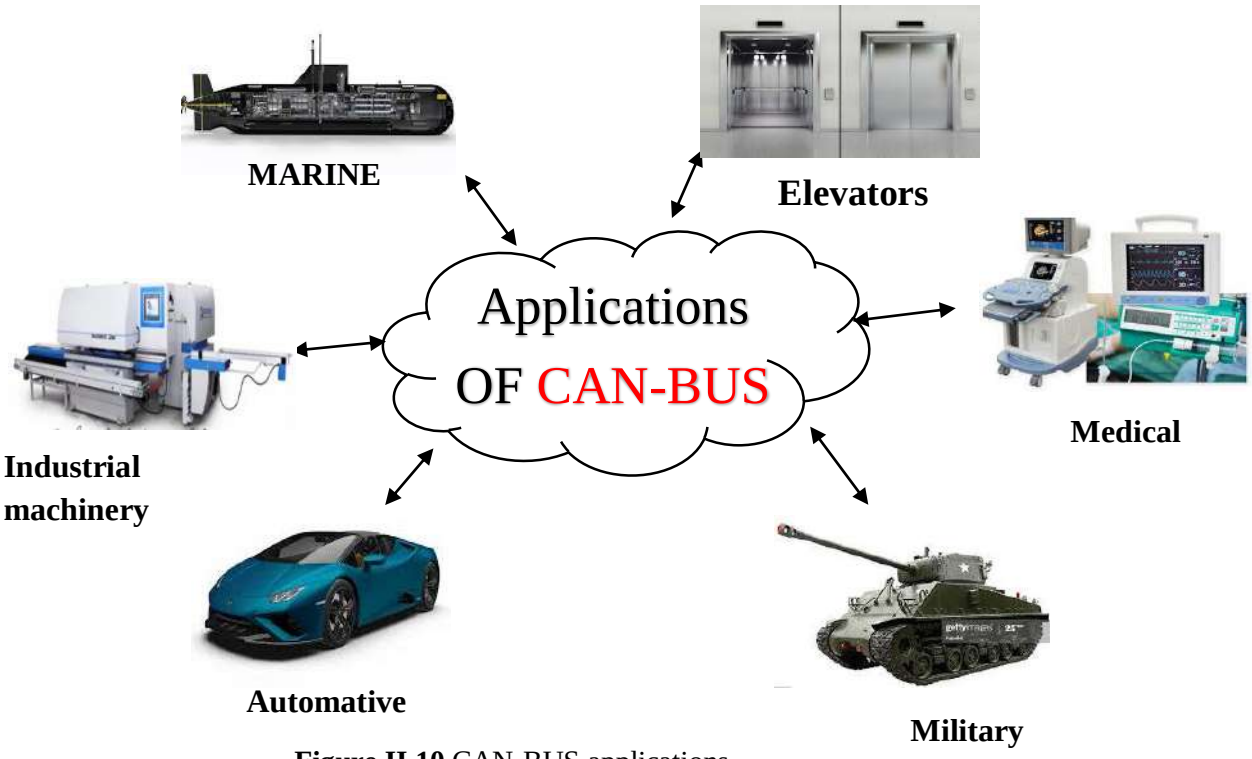


Figure II.10.CAN-BUS applications

II.18.CAN-BUS uses in smart home

CAN-BUS used is one of the best options network for smart home, it is used specially in the automation of system such as HVAC (heating, ventilation, air conditioning), elevators, etc. it has been proposed as a backbone network for security systems such as fire detection and to direct load control application in residential areas [21].

II.19.conclusion

Today, the efficiency and robustness of the CAN protocol has led it to be used in many industrial applications, in particular those requiring a high throughput of up to 1 Mbits/s with a very low error rate.

CAN has real time features and ease of implementation and programming of the nodes and therefore, a lower cost, it is a multi-master network of receiver-transmitter type, it is a good alternative as a data link layer over other networks in building management and control.

CAN-BUS have a high immunity against electrical interference and the ability to self-diagnose and repair data error ,it is suitable in application requiring a large number of short messages with high reliability in ragged environment.

Chapter III: Working Environment and Tools

III.1.Introduction

In this chapter we will introduce all the equipment (arduino, MCP2515), software used and sets of sensors and components.

Internets of objects allow devices to exchange data between them which permits us to control our smart home. For the development of our interface, we present the software Node Red.

III.2.Internet of Things

III.2.1.Definition

The Internet of Things is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment and the confluence of efficient wireless protocols, improved sensors, cheaper processors, and a bevy of start-ups and established companies developing the necessary management and application software has finally made the concept of the Internet of Things mainstream.

The Internet of Things is not a single technology, it’s a concept in which most new things are connected and enabled such as street lights being networked and things like embedded sensors, image recognition functionality, augmented reality, near field communication, are integrated into situational decision support, asset management, and new services. These bring many business opportunities and add to the complexity of IT.

A dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual “things” have identities, physical attributes, and virtual personalities use intelligent interfaces and are seamlessly integrated into the information network (see the figure III.1) [22].

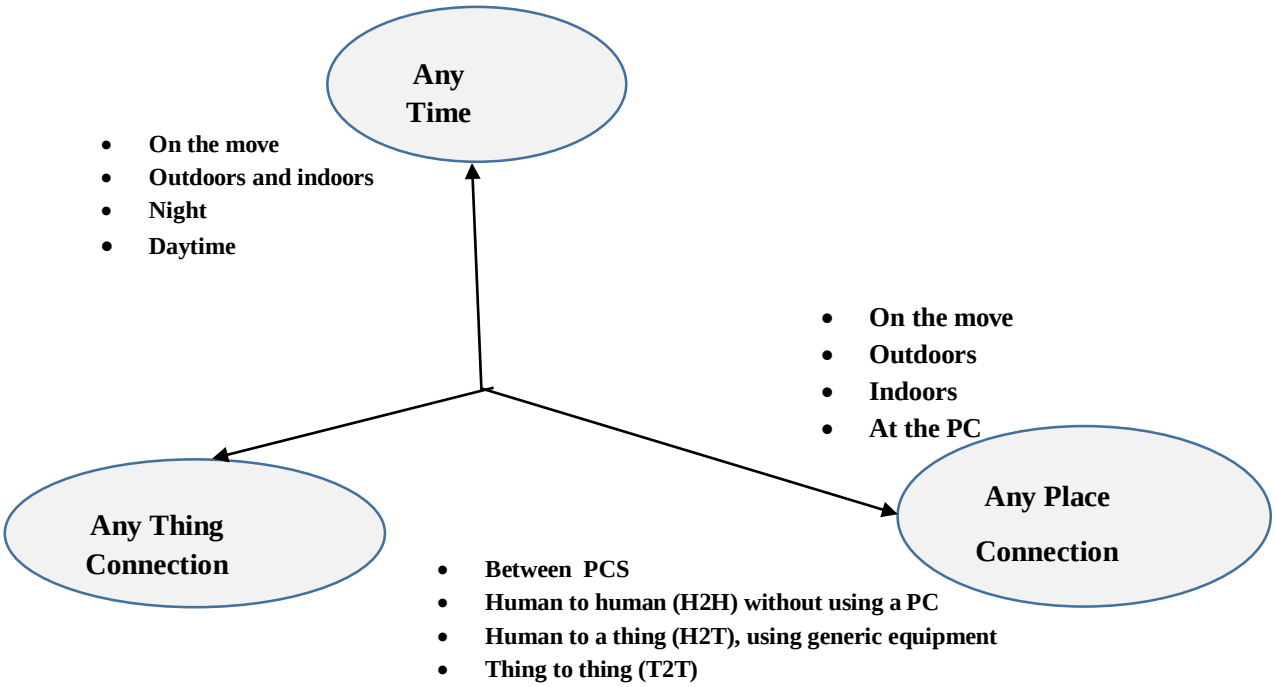


Figure III.1. New dimension for IoT

III.2.2.Components of internet of things

As we see above, internet of things relate devices and systems that interlink actual world sensors and actuators to the internet so we need to build our applications with the next components (figure III.2):

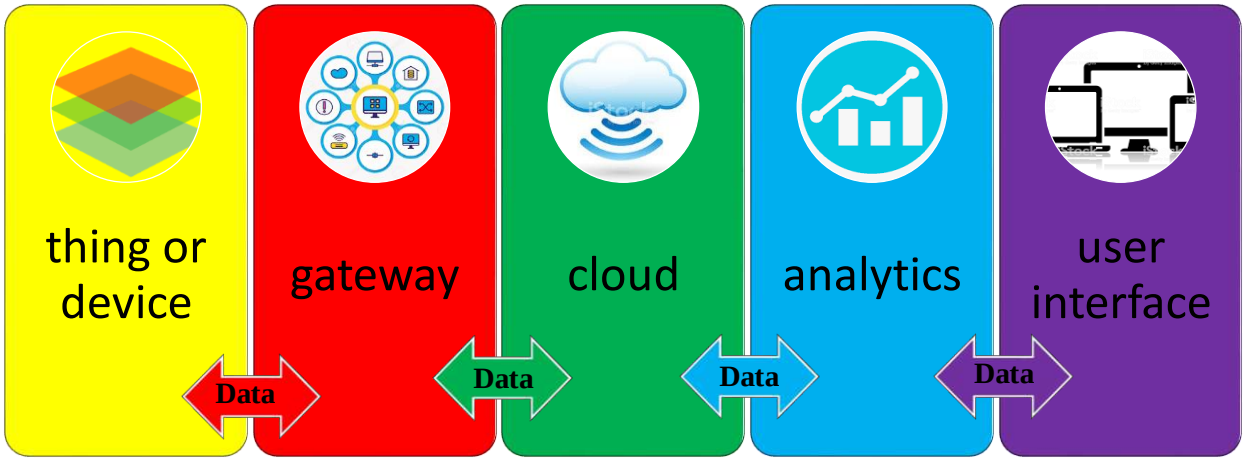


Figure III.2.components of IOT

III.2.2.1.Smart devices and sensors – Device connectivity

Devices and sensors are the components of the device connectivity layer. These smart sensors are continuously collecting data from the environment and transmit the information to the next layer. Latest techniques in the semiconductor technology is capable of producing micro smart sensors for various applications.

Common sensors are:

- Temperature sensors and thermostats
- Humidity / Moisture level
- Light intensity detectors
- Moisture sensors
- Proximity detection
- RFID tags

- **How the devices are connected?**

Most of the modern smart devices and sensors can be connected to low power wireless networks like Wi-Fi, ZigBee, Bluetooth, Z-wave, LoRAWAN etc… Each of these wireless technologies has its own pros and cons in terms of power, data transfer rate and overall efficiency (see the figure III.3).



Figure III.3. Networks used

Developments in the low power, low cost wireless transmitting devices are promising in the area of IoT due to its long battery life and efficiency.

Latest protocols like 6LoWPAN- IPv6 over Low Power Wireless Personal Area Networks have been adapted by many companies to implement energy efficient data transmission for IoT networks. Low PAN uses reduced transmission time (typically short time pulses) and thus saves energy.

III.2.2.2.Gateway

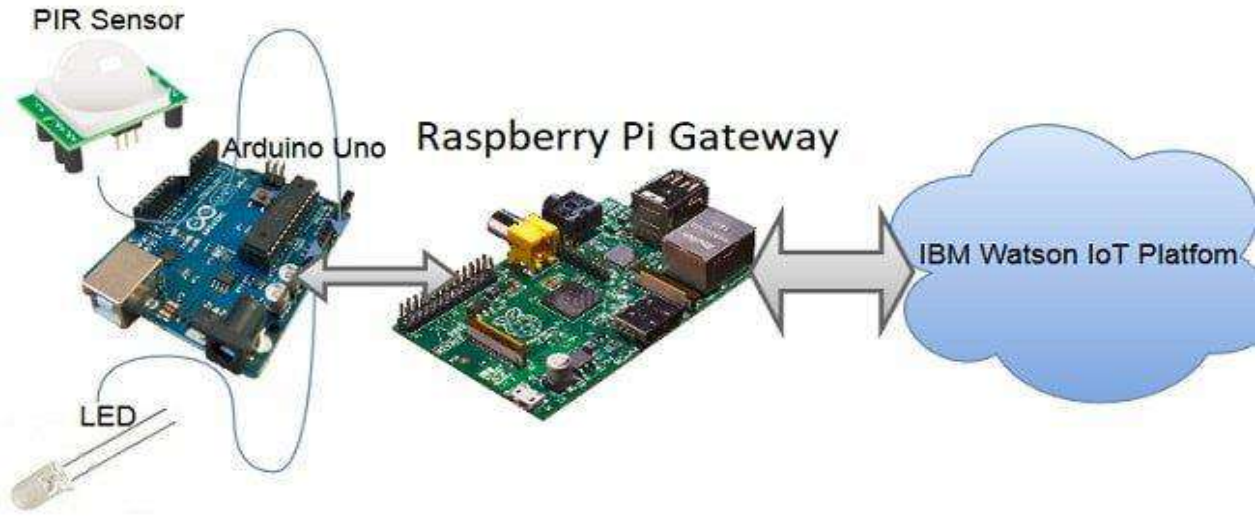


Figure III.4. Gatways

IoT Gateway manages the bidirectional data traffic between different networks and protocols. Another function of gateway is to translate different network protocols and make sure interoperability of the connected devices and sensors.

Gateways can be configured to perform pre-processing of the collected data from thousands of sensors locally before transmitting it to the next stage. In some scenarios, it would be necessary due to compatibility of TCP/IP protocol.

IoT gateway offers certain level of security for the network and transmitted data with higher order encryption techniques. It acts as a middle layer between devices and cloud to protect the system from malicious attacks and unauthorized access (figure III.4).

III.2.2.3.Cloud

Internet of things creates massive data from devices, applications and users which has to be managed in an efficient way. IoT cloud offers tools to collect, process, manage and store huge amount of data in real time. Industries and services can easily access these data remotely and make critical decisions when necessary.

Basically, IoT cloud is a sophisticated high performance network of servers optimized to perform high speed data processing of billions of devices, traffic management and deliver accurate analytics. Distributed database management systems are one of the most important components of IoT cloud.

Cloud system integrates billions of devices, sensors, gateways, protocols, data storage and provides predictive analytics. Companies use these analytics data for improvement of products and services, preventive measures for certain steps and build their new business model accurately.

III.2.2.4.Analytics

Analytics is the process of converting analog data from billions of smart devices and sensors into useful insights which can be interpreted and used for detailed analysis. Smart analytics solutions are inevitable for IoT system for management and improvement of the entire system.

One of the major advantages of an efficient IoT system is real time smart analytics which helps engineers to find out irregularities in the collected data and act fast to prevent an undesired scenario. Service providers can prepare for further steps if the information is collected accurately at the right time.

Big enterprises use the massive data collected from IoT devices and utilize the insights for their future business opportunities. Careful analysis will help organizations to predict trends in the market and plan ahead for a successful implementation.

Information is very significant in any business model and predictive analysis ensures success in concerned area of business line.

III.2.2.5.User interface

User interfaces are the visible, tangible part of the IoT system which can be accessible by users. Designers will have to make sure a well-designed user interface for minimum effort for users and encourage more interactions.

Modern technology offers much interactive design to ease complex tasks into simple touch panels controls. Multicolor touch panels have replaced hard switches in our household appliances and the trend is increasing for almost every smart home devices.

User interface design has higher significance in today’s competitive market, it often determines the user whether to choose a particular device or appliance. Users will be interested to buy new devices or smart gadgets if it is very user friendly and compatible with common wireless standards [23].

III.2.3.The architecture of internet of things:

Internet of things is known for many architecture but we discuss three layered architecture in this part.

III.2.3.1.Three layered architecture

We will explain the three-layer architecture, as shown in (Figure III.5)

- **The perception layer**

  Consists of two-dimension code tags and readers, RFID tags and readers, cameras, GPS, all kinds of sensors, sensor network, Machine to Machine (M2M terminals, and sensor gateways, etc. Perception layer includes perception control sub-layer and communication extension sub-layer. Perception control sub-layer realizes intelligent perception of the physical world, in- formation acquisition and processing and automatic control; while communication extension sub-layer connected the physical things to the network layer and the application layer with communication terminal module directly or through the extension network composed by the terminal modules.

- **The network layer**

  Consists of converged network formed by all kinds of communication network and the Inter- net. The concept of network layer has been widely accepted due to its mature technologies. Besides, the PIoT management Centre and information Centre are parts of the network layer.

  That is to say, the network layer can not only operate the network, but also can operate the information. Network layer mainly realizes information transmission, routing and control, including the access network and core network. The network layer can rely on industry-specific communication networks as well as the public telecommunications networks.

- **Application layer**

  Is a combination of IoT technologies and industry expertise to achieve a broad set of intelligent application solutions? Application layer includes application infrastructure/middleware and a variety of applications of IoT. Application infrastructure/middleware provides information processing, computing, and other common basic services, capacity and resources interface for IoT. The key issue of the application layer is information sharing and information security. Through the application layer, IoT can achieve deep integration of information technology with the industry. It will have great effect on economic and social development (figure III.5) [24].

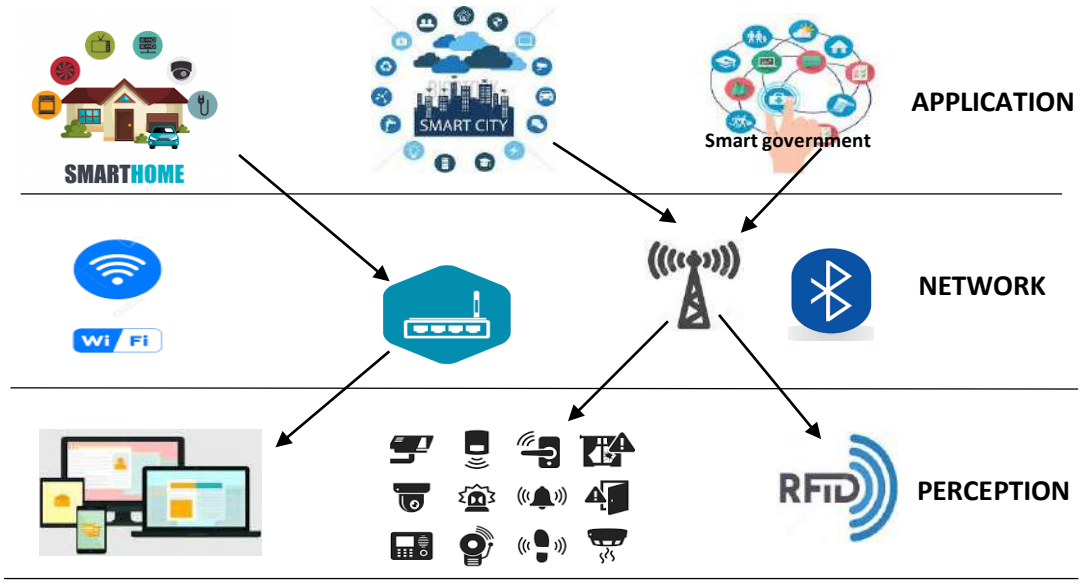


Figure III.5. Architecture of IOT

III.2.4.Applications of internet of things:

The IoT application covers “smart” environments/spaces in domains such as: Transportation, Building, City, Lifestyle, Retail, Agriculture, Factory, Supply chain, Emergency, Healthcare, military, User interaction, Culture and tourism, Environment and Energy (see the figure III.6).

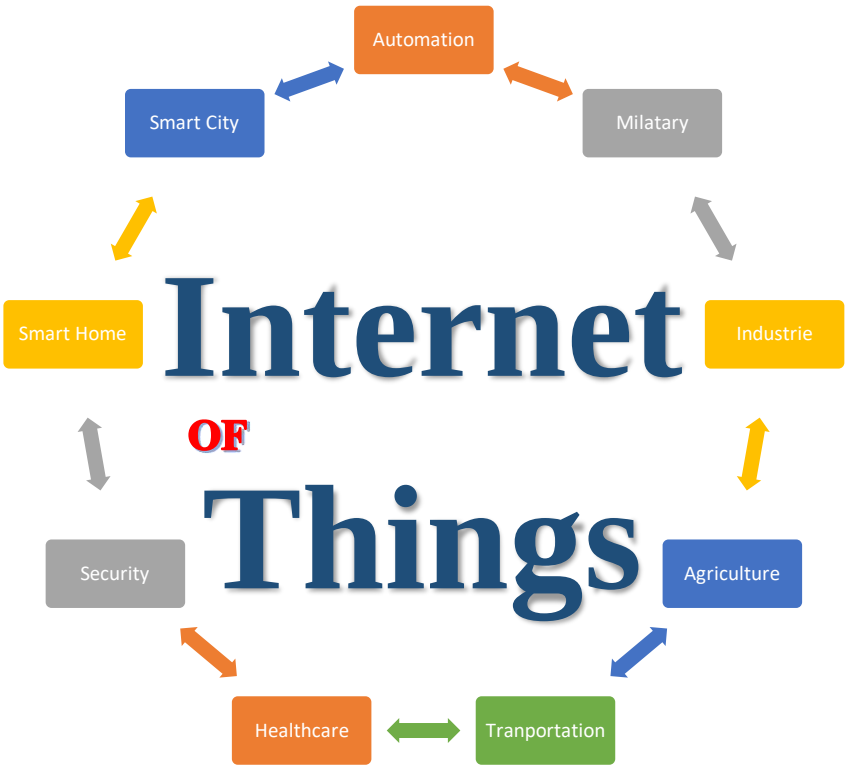


Figure. III.6. Applications for the internet of things

III.2.5.adventages and disadvantages of internet of things

III.2.5.1.IOT Advantages

The advantages of IoT span across every area of lifestyle and business. Here is a list of some of the advantages that IoT has to offer:

- **Improved Customer Engagement –** Current analytics suffer from blind-spots and significant flaws in accuracy; and as noted, engagement remains passive. IoT completely transforms this to achieve richer and more effective engagement with audiences.
- **Technology Optimization –** The same technologies and data which improve the customer experience also improve device use, and aid in more potent improvements to technology. IoT unlocks a world of critical functional and field data.
- **Reduced Waste –** IoT makes areas of improvement clear. Current analytics give us superficial insight, but IoT provides real-world information leading to more effective management of resources.

- **Enhanced Data Collection –** Modern data collection suffers from its limitations and its design for passive use. IoT breaks it out of those spaces, and places it exactly where humans really want to go to analyze our world. It allows an accurate picture of everything.

III.2.5.2.IoT Disadvantages

Though IoT delivers an impressive set of benefits, it also presents a significant set of challenges.

Here is a list of some its major issues:

- **Security –** IoT creates an ecosystem of constantly connected devices communicating over networks. The system offers little control despite any security measures. This leaves users exposed to various kinds of attackers.
- **Privacy –** The sophistication of IoT provides substantial personal data in extreme detail without the user's active participation.
- **Complexity –** Some find IoT systems complicated in terms of design, deployment, and maintenance given their use of multiple technologies and a large set of new enabling technologies.
- **Flexibility –** Many are concerned about the flexibility of an IoT system to integrate easily with another. They worry about finding themselves with several conflicting or locked systems.
- **Compliance –** IoT, like any other technology in the realm of business, must comply with regulations. Its complexity makes the issue of compliance seem incredibly challenging when many consider standard software compliance a battle.[25]

III.2.6.IoT in Homes

IoT enabled smart home or smart homes has no any generally accepted definition. A home that equipped with different smart appliances those are capable of exchanging data and information within with the help of internet connection facilitating automation and intelligence to control the various tasks and devices itself is known to be a smart home environment. Smart homes are the demand of this time that studies and surveys has found that IoT devices for smart homes will increase by 53.9% of US households.

III.3.Node-Red

In many projects, the HMI consists of a web application accessible from a browser. It is not always easy to program the link between the equipment (sensors and / or actuators) and the web page provided to the user. One solution is to use a Python script whose execution can be scheduled with Cron, to interact with the hardware, read the sensor data, and store it in a database like MySQL. A web server like Apache2, via a PHP web page, makes information available to users. This solution described in a previous article requires the mobilization of many technologies and several programming languages, forcing the project developers to delay the implementation of a prototype to train.

Combined with a hardware solution consisting of an Arduino board, Node-Red turns out to be a very interesting alternative: Node-RED is a powerful tool for building the Internet of Things (IoT) applications with an emphasis on simplifying programming, which is done using predefined blocks of code, called "nodes" to perform tasks. It uses a visual programming approach that allows developers to connect blocks of code. Connected nodes, typically a combination of input nodes, processing nodes, and output nodes, when wired together, constitute a "flow (see the figure III.7)".

Access to the programming interface is via a browser. It is preferable to use the browser of a computer connected to the same network as the arduino:

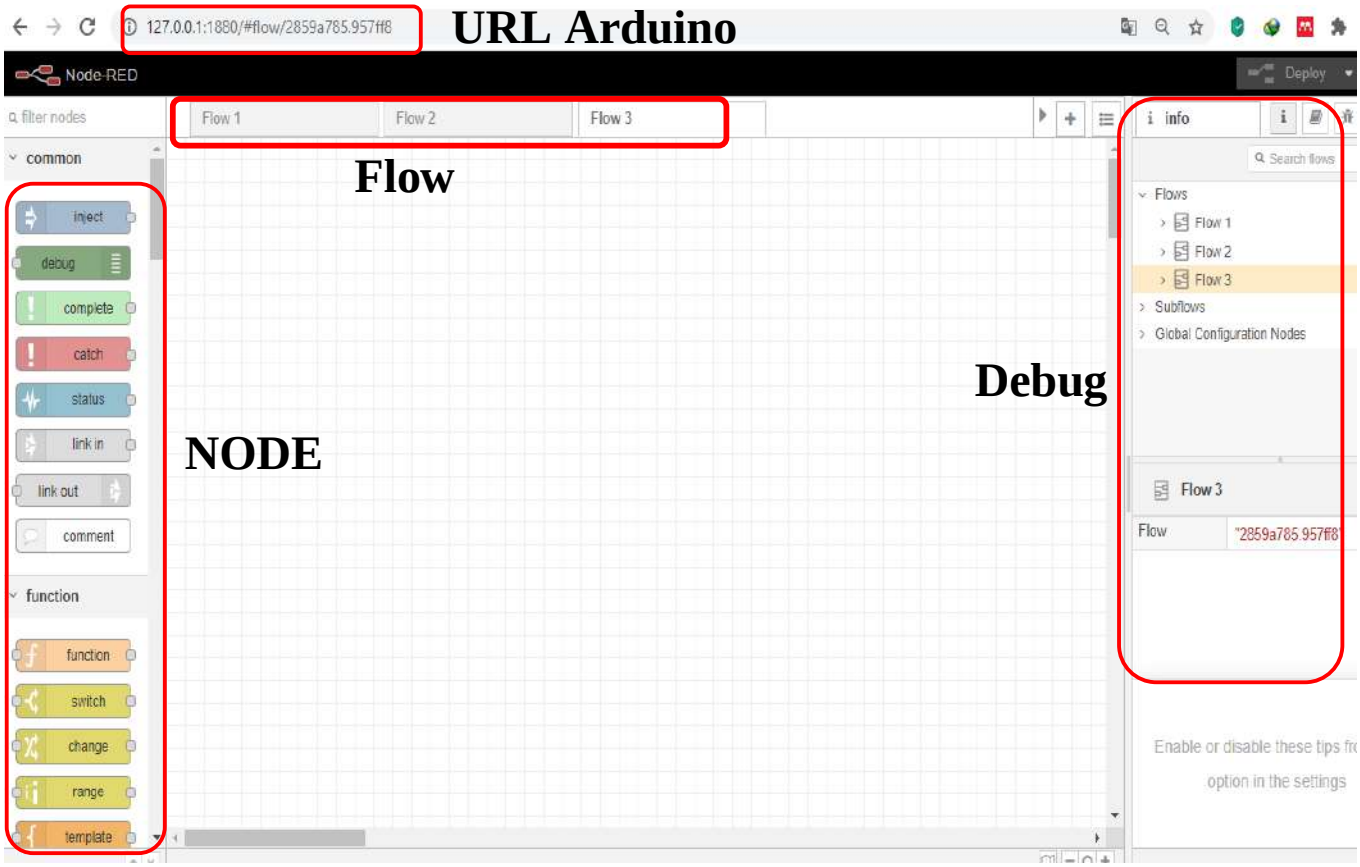


Figure III.7. Node red interface

We now have the node manager on the menu (see the figure III.8):

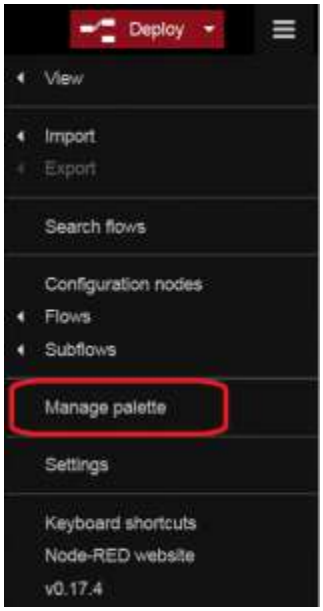


Figure III.8.manage palette

Click on Manage Palette then find the palette of nodes to install:

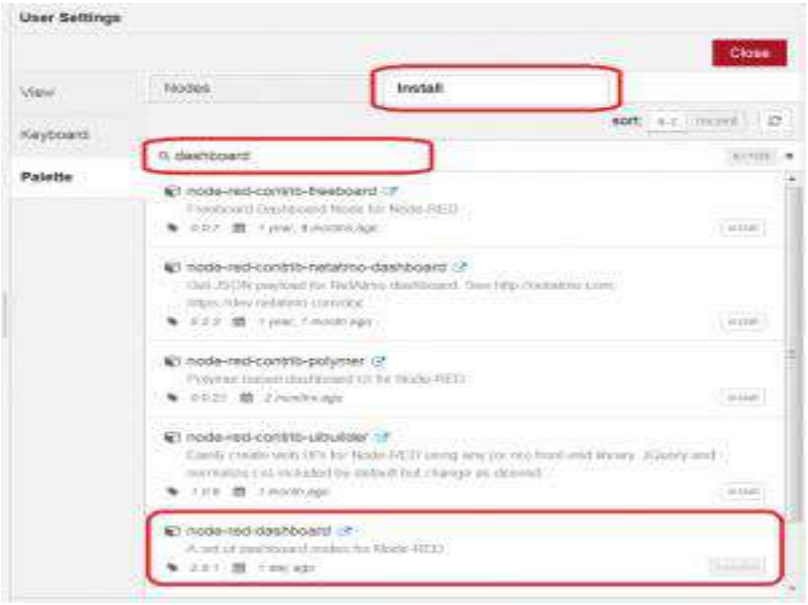


Figure III.9.install dashboard

Note that there are currently 1335 node palettes available at the moment. You should be able to find your happiness there!

A new palette is now available (see the figure III.10):

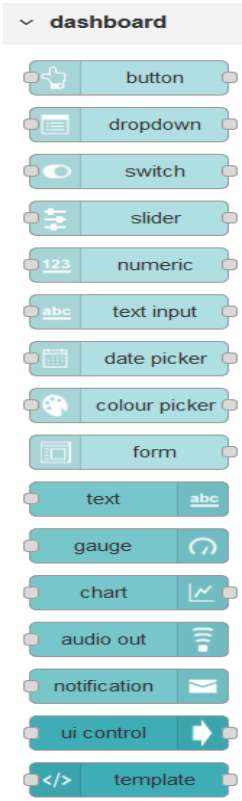


Figure III.10.bloc of dashboard

Let's start by making a very simple flow composed of a Slider node and a Gauge node. To change the default node configuration, simply double-click it.

For these components to be well arranged on the web page, you must put them in a Tab, then in a Group, in the group selection box, click Add new ui_group, then create a group (here: "The group") and do the same for the Tab (here: "Test").

Then click on the done button and deploy the Flow. You access the website generated by the URL: http://127.0.0.1:1880/#flow/80b5c321.15dcb

- **Warning**:

  Before starting a new Flow, remember to delete or disable (double-click on the Flow tab then Disabled) otherwise you will serve two flows, i.e. two programs at the same time, which may lead to strange effects [26].

III.4.Arduino

It is an open-source electronic board to develop a lot of ideas and projects related to automatic control easily and simply by using an open-source programming language. The controller on the board is programmed using a program called integrated development environment IDE which is available for free for download from the official Arduino site.

Arduino IDE software works on Mac, Windows, and Linux. It is due to this software that we will create, test, and send the programs to the Arduino.

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for anyone making interactive projects.

III.4.1.ARDUINO BOARD

Arduino senses the environment by receiving inputs from many sensors and affects its surroundings by controlling lights, motors, and other actuators.

III.4.4.ARDUINO UNO

The Arduino UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family(see the figure III.11).

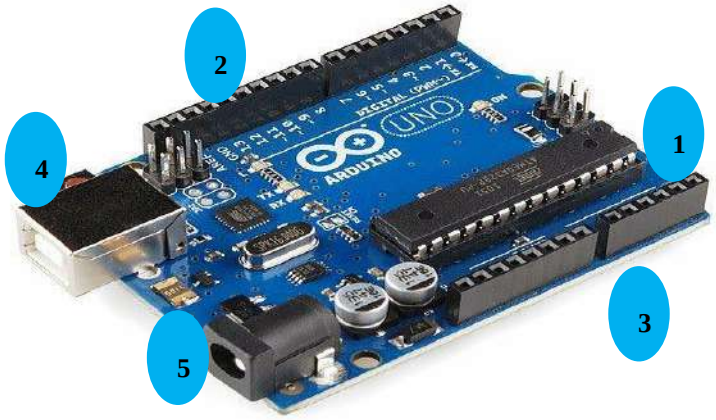


Figure III.11. Arduino UNO Board

Arduino UNO is a microcontroller (1) board based on the ATmega328P (datasheet). It has 14 digital input/output pins (2) (of which 6 can be used as PWM outputs), 6 analog inputs (3), a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection (4), a power jack (5), an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

"UNO" means one in Italian and was chosen to mark the release of the Arduino Software (IDE) 1.0. The UNO board and version 1.0 of the Arduino Software (IDE) were the reference versions of the Arduino, now evolved to newer releases. The UNO board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past, or outdated boards see the Arduino index of boards[27].

III.4.2.ARDUINO SOFTWARE

You can tell your Arduino what to do by writing code in the Arduino programming language and using the Arduino development environment (see the figure III.12).

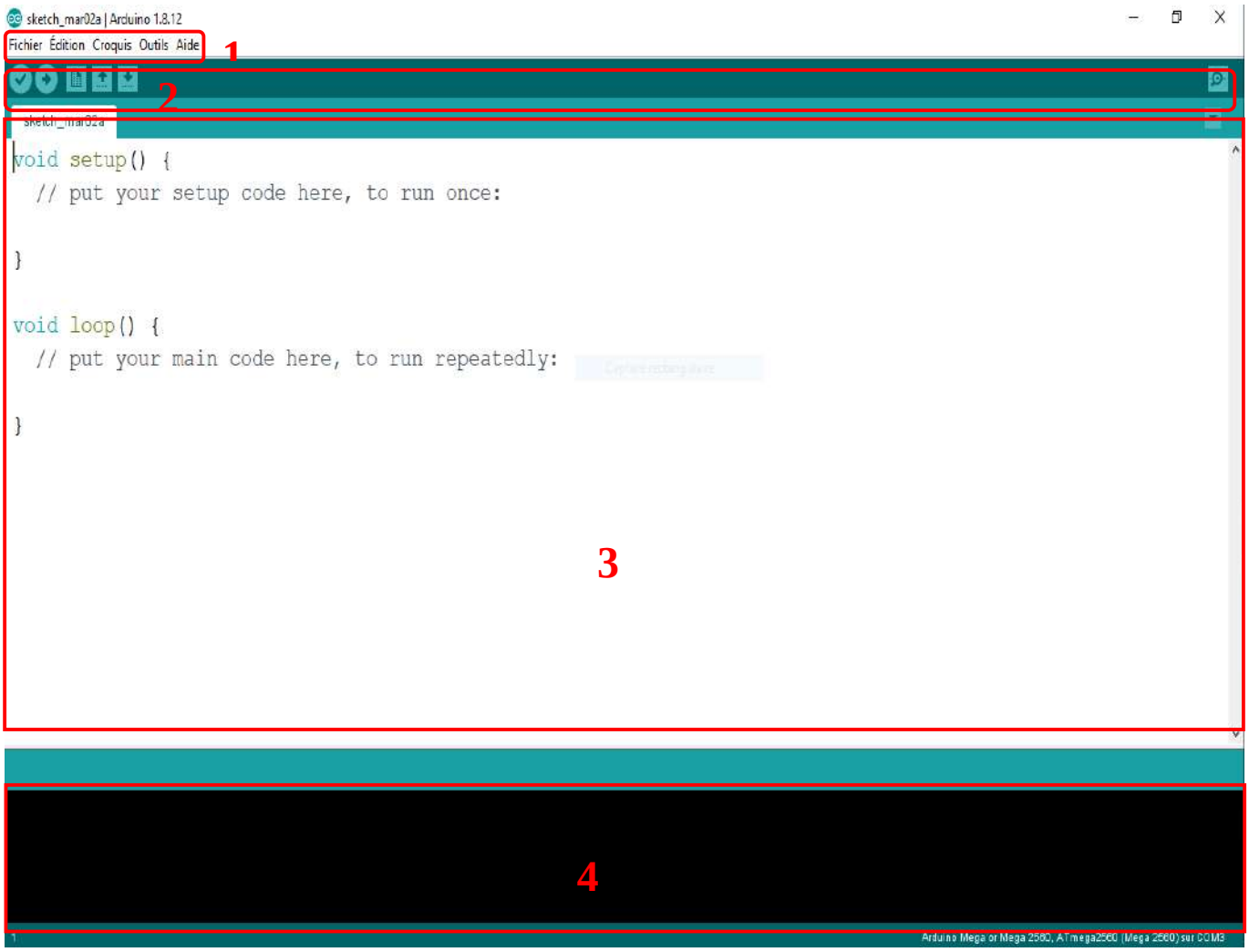


Figure III.12. Arduino IDE

- Number 1: these are the software configuration options
- Number 2: it contains the buttons that will be used when programming our cards
- Number 3: this block will contain the program that we are going to create
- Number 4: this one is important because it will help us to correct the mistakes in our program. This is the debugger.

III.5.MCP 2515 CAN-BUS shield

III.5.1.Definition

Microchip Technology's MCP2515 is a stand-alone Controller Area Network (CAN) controller that implements the CAN specification, version 2.0B. It is capable of transmitting and receiving both standard and extended data and remote frames.

The MCP2515 has two acceptance masks and six acceptance filters that are used to filter out unwanted messages, thereby reducing the host MCUs overhead.

The MCP2515 interfaces with microcontrollers (MCUs) via an industry standard Serial Peripheral Interface (SPI) (see the figure III.13) [28].



Figure III.13.MCP 2515 shield



Figure III.14.MCP 2515 Microchip

This particular module is based on MCP2515 CAN Controller IC and TJA1050 CAN Transceiver IC. The MCP2515 IC is a standalone CAN Controller and has integrated SPI Interface for communication with microcontrollers.

Coming to the TJA1050 IC, it acts as an interface between the MCP2515 CAN Controller IC and the Physical CAN Bus.

The following image shows the components and pins on a typical MCP2515 Module (see the figure III.15) [30].

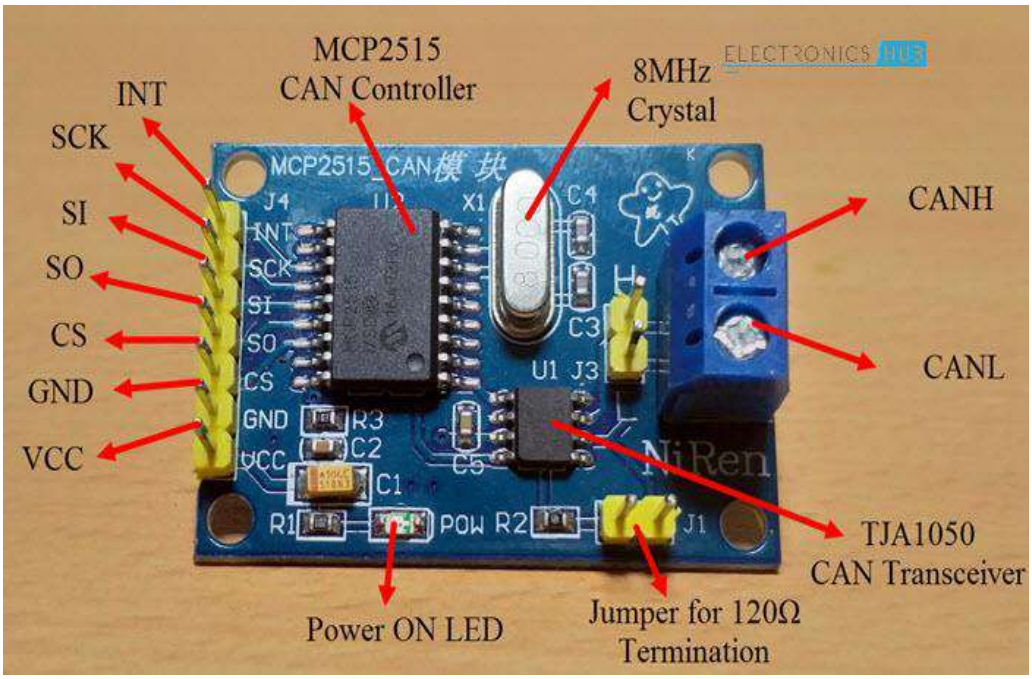


Figure III.15. Circuit Diagram for Interfacing MCP2515 with Arduino

MCP2515 IC is the main controller that internally consists of three main subcomponents: The CAN Module, the Control Logic and the SPI Block.

CAN Module is responsible for transmitting and receiving messages on the CAN Bus. Control Logic handles the setup and operation of the MCP2515 by interfacing all the blocks. The SPI Block is responsible for the SPI Communication interface.

Coming to the TJA1050 IC, since it acts as an interface between MCP2515 CAN Controller and the physical CAN Bus, this IC is responsible for taking the data from the controller and relaying it on to the bus(see the figure III.15).

- **Characteristics of can bus module**

1. DB9 Interface
2. V_OBD
3. Led Indicator
4. Terminal – CAN_H, CAN_L
5. Arduino Uno Pinout
6. Serial Groove Connector
7. I2C Groove Connector
8. ICSP Pins
9. Transceiver

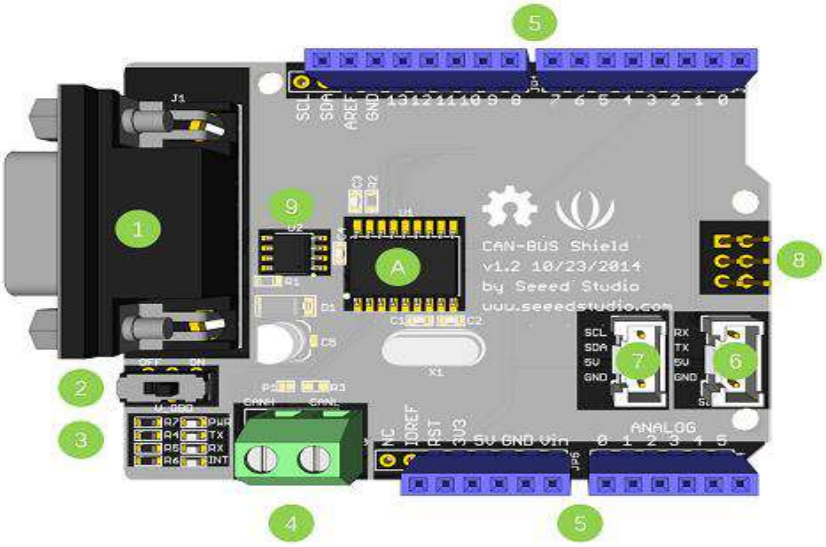


Figure III.16.other diagram for module can-bus (MCP2515) [29]

The following image shows the circuit diagram of interfacing MCP2515 CAN Module with Arduino and possible communication between two Arduino over CAN Protocol (see the figure III.16).

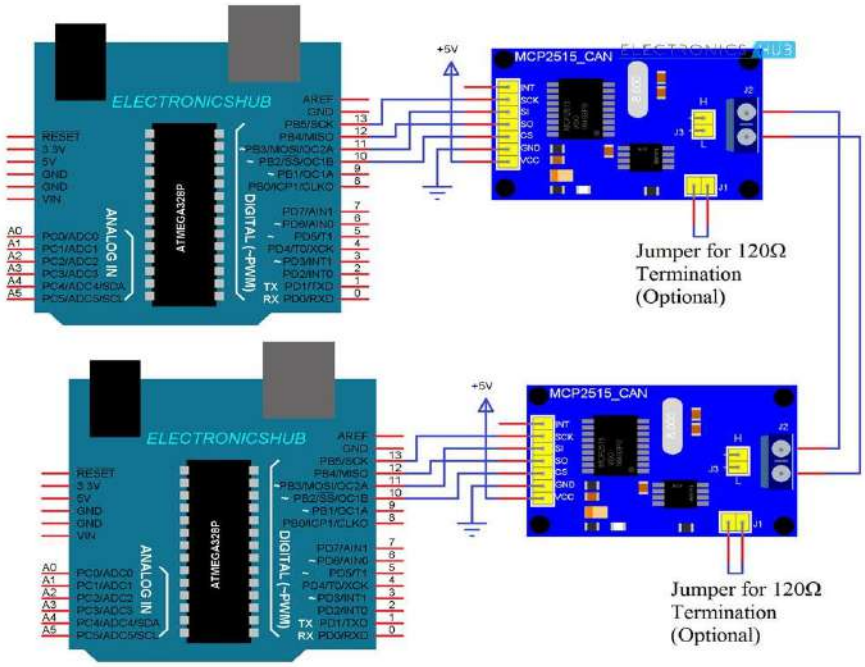


Figure III.17. Circuit Diagram for Interfacing MCP2515 with Arduino

- **Program MCP2515 in arduino**

The main function used in arduino code of MCP2515 send/receive

- Install the library to your Arduino IDE

One of the node acts as master, the other acts as slaver.

- We should initialize all the libraries used:

```
#include <mcp_can.h>
#include <SPI.h>
```

- Check Receive

The MCP2515 can operate in either a polled mode, where the software checks for a received frame, or using additional pins to signal that a frame has been received or transmit completed:

```
if(CAN_MSGAVAIL == CAN.checkReceive())
```

- Get CAN ID

When some data arrive, you can use the following function to get the CAN ID of the "send" node.

```
unsigned int canId = CAN.getCanId();
```

- Send Data

```
CAN.sendMsgBuf(0x00, 0, 8, stmp);
```

ID represents where the data comes from.

EXT represents the status of the frame. '0' means standard frame. '1' means extended frame.

LEN represents the length of this frame.

data_buf is the content of this message.

- Receive Data

```
CAN.readMsgBuf(&len,buf);
```

III.5.2.CAN-BUS shield features

Features of this CAN BUS shield includes:

- Implements CAN V2.0B at up to 1 Mb/s
- Industrial standard 9 pin sub-D connector
- OBD-II and CAN standard pin out selectable.
- Changeable chip select pin
- Changeable CS pin for TF card slot
- Changeable INT pin
- Screw terminal that easily to connect CAN_H and CAN_L
- Arduino Uno pin headers
- 2 Grove connectors (I2C and UART)
- SPI Interface up to 10 MHz
- Standard (11 bit) and extended (29 bit) data and remote frames
- Two receive buffers with prioritized message storage[30]

III.6.Conclusion

In this chapter we provided an overview of the internet of things (IOT) architecture, components, and applications, as well as how we work on the node-red programming tool, then we presented the hardware by giving an idea about arduino board and its IDE (integration development environment), finally we give a brief description to MCP2515 shield .

Chapter IV: Experimental

IV.1.Introduction

We have previously provided the tools used in this work, hardware and software, the purpose of this chapter is to present the concepts used for practical implementation, we give a detailed description of the communication between the sections of sending and receiving data, the collection of these data sent to the computer.

We can supervise and control our house remotely on computer or cell phone based on the internet of things, visualized by Node Red.

In the following we will explain the specifications and the hardware and programming solutions used.

Our work aims to design and build a connected smart home, in which the communication between the components manage by CAN-BUS, finally we will present the platform of HMI.

IV.2.Project specification

In order to realize our control and monitoring system based on the Internet of Things and the CAN-BUS communication protocol applied to a home, we have identified the important points required that guarantee the clarity and precision of our work.

- Manage data using protocol can-bus.
- Control temperature, humidity, distance, angle by using node-red.
- Monitoring of home parameters: temperature, humidity, distance, angle and light by a Human Machine Interface (IHM).
- We use internet of thing interface to control and monitor remotely.

IV.3.General operation of monitoring and control system proposed

- The supervision and control system developed (based on the Internet of Things) is applied to a transmission center. The system uses CAN-BUS protocol composed of several nodes. One of the nodes contains a temperature and humidity sensor (DHT11) for measuring the ambient temperature
- The operating mode of the system is selected from the HMI, either automatic or manual. If the mode is automatic, an algorithm is implemented in Arduino to modify the angle.
- An ultrasonic sensor is used to give information of the distance to the target from the transmission center. The measurements are sent to the HMI.
- The measured values: temperature, humidity, distance as well as the detected states: angle are sent to a node 2, through the CAN-BUS device as shown in the (figure IV.1). This node is connected to the pc through a serial

communication sends this information from sensors and detectors to the HMI (see figureVI.1).

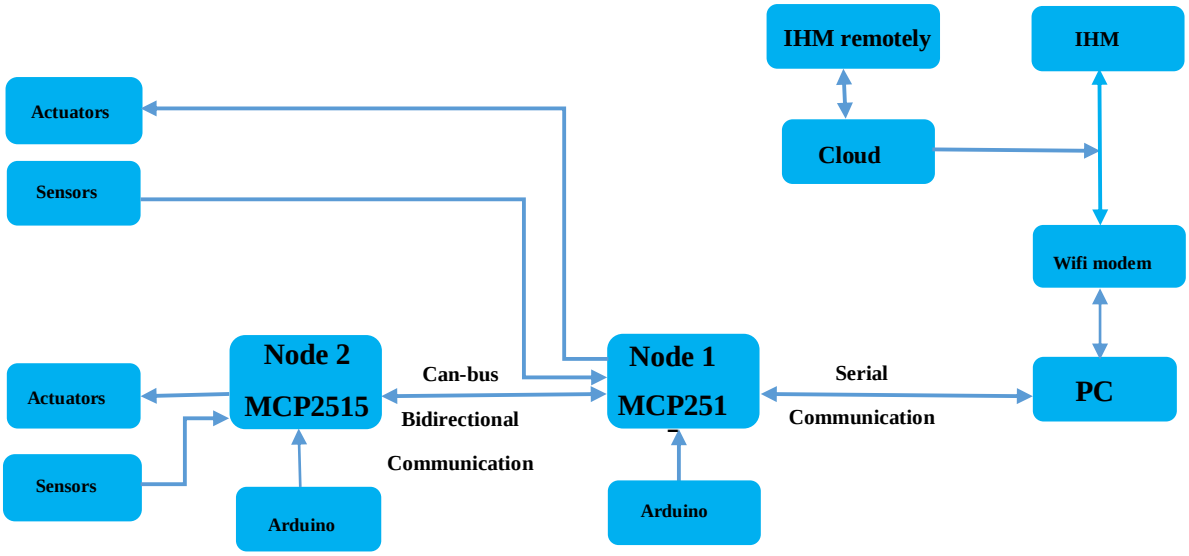


Figure IV.1.monitoring and control system

VI.4.Human machine interface

The Human Man Interface is a tool to connect a user to a system (devices, machines), we can monitor and control our systems with the HMI using: a screen integrates machines, computer screens, a tablet etc. Among the interfaces developed for this is the red node. The latter facilitates the creation of connected objects.

It is a graphic language (underlying in javascript) which allows you to create links between incoming and outgoing data flows.

It is free and requires (Open Source): NodeJS (javascript on the server side) software supported by IBM.

IV.5.Presentation of sensors and actuators used

IV.5.1.Sensors

IV.5.1.1.DHT11 Temperature and Humidity Sensor

The DHT11 sensor is capable of measuring temperatures from 0 to + 50 ° C with an accuracy of +/- 2 ° C and relative humidity levels of 20 to 80% with an accuracy of +/- 5%. A measurement can be taken every second.

DHT11 is compatible 3.3 volts and 5 volts (the manufacturer recommends however to always supply the sensor with 5 volts to have precise measurements). It also have the same wiring and the same communication protocol.

The DHT11 cannot measure (and withstand) negative temperatures or temperatures above 50 ° C. So be careful when using outdoors! (see figureVI.2).

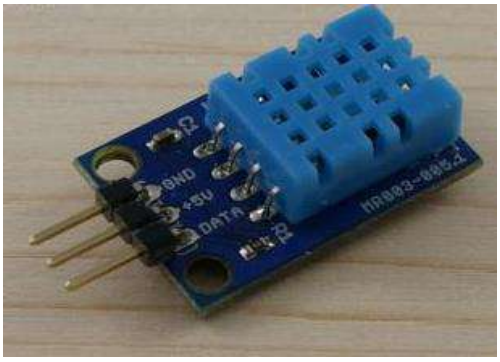


Figure IV.2.Temperature and humidity sensor

As specified previously, DHT11 sensor have the particularity of communicating with the microcontroller via a single input / output pin.

- **Mounting principle**

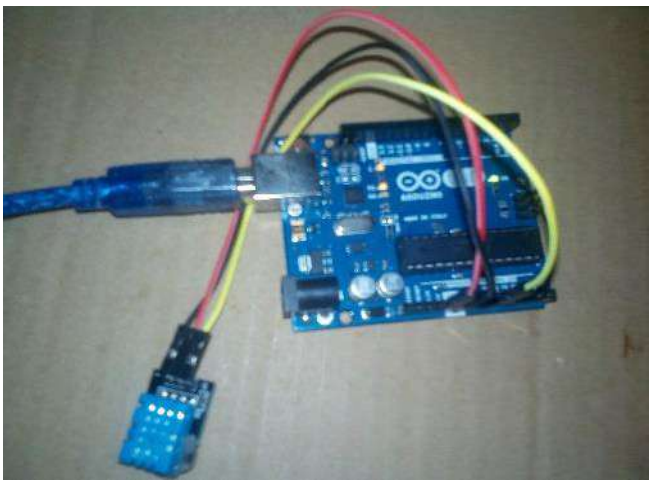


Figure IV.3.mounting principle dht11

The pin out of the sensor is as follows:

- Pin 1 is the communication pin. This must be connected to the power supply via a 4.7K ohm pull-up resistor (this is an open collector output).
- Pin 2 is the power pin (5 volts or 3.3 volts).
- Pin 3 is the sensor ground (GND) (see figureVI.3).

- **The characteristics**

Table IV.1 characteristics of DHT11

Relative humidity %	20 ~ 80%
Accuracy (humidity)	+/- 5%
Temperature	0 ~ + 50 ° C
Accuracy (temperature)	+/- 2 ° C
Supply voltage	3 ~ 5 volts
Long term stability	+/- 1% per year
Max measurement frequency	1Hz (1 measurement per second)

Communication with a DHT11 sensor is done in 3 steps:

- First, the master microcontroller (the Arduino board in our case) wakes up the sensor by placing the data line at LOW for at least 800µs (at least 18ms for the DHT11). During this time, the sensor will wake up and prepare a temperature and humidity measurement. Once the time has elapsed, the master will release the data line and start listening
- Once the data line is released, the sensor responds to the master (to show that it is fully awake) by holding the data line at LOW for 80µs and then at HIGH for 80µs.
- The sensor will then transmit a series of 40 bits (5 bytes). The first two bytes contain the humidity measurement. The next two bytes contain the temperature measurement and the fifth byte contains a checksum which verifies that the data read is correct [31].

IV.5.1.2.Ultrasonic sensor

Figure IV.4 shows an ultrasonic sensor module.

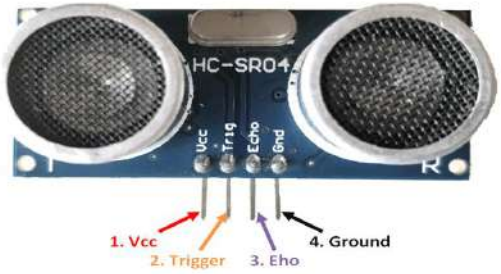


Figure IV.4.ultrasonic sensor (HC-SR04)

- **Ultrasonic Sensor Pin Configuration**

Table IV.1 characteristic of Ultrasonic Sensor

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize Measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be Equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

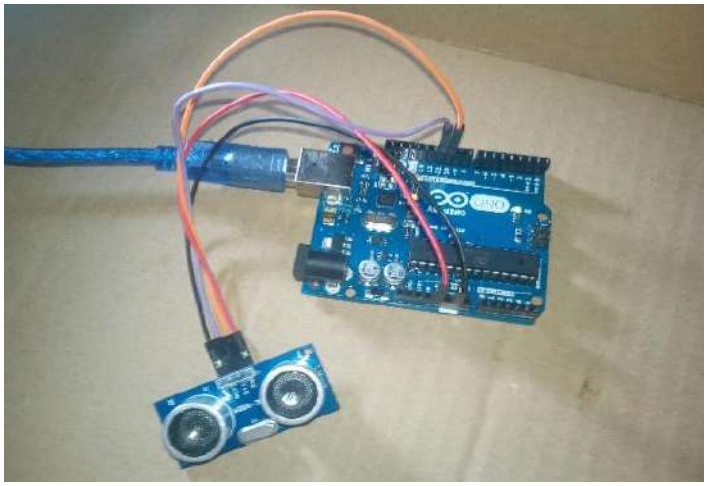


Figure IV.5.mounting principle HC-SR04

- **HC-SR04 Sensor Features**
  - Operating voltage: +5V
  - Theoretical Measuring Distance: 2cm to 450cm
  - Practical Measuring Distance: 2cm to 80cm
  - Accuracy: 3mm
  - Measuring angle covered: <15°

- Operating Current: <15mA
- Operating Frequency: 40Hz

➢ HC-SR04 Ultrasonic Sensor - Working

As shown above the HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

- **Distance = Speed × Time**

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the (figure VI.6) below.

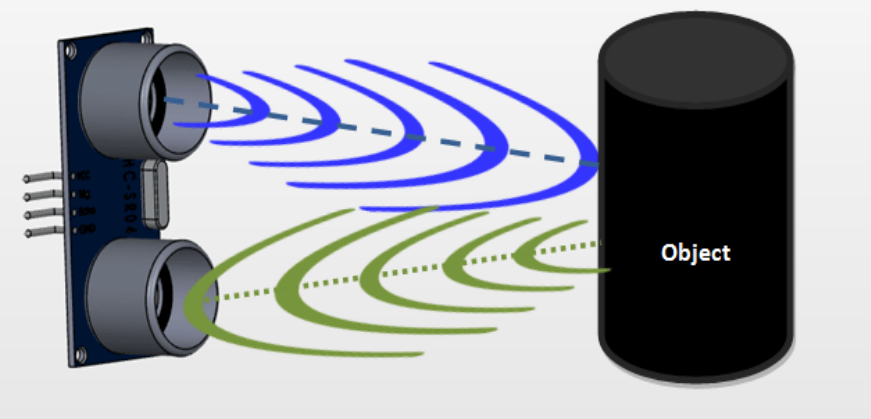


Figure VI.6.the way to get a distance

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

➢ Applications

- Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- Used to measure the distance within a wide range of 2cm to 400cm
- Can be used to map the objects surrounding the sensor by rotating it
- Depth of certain places like wells, pits etc. can be measured since the waves can penetrate through water [32].

IV.5.1.3.LDR

A photoresistor is an electronic component whose resistivity fluctuates (frequently a decrease) according to the increase in light reaching it. It can also be called light-dependent resistor (LDR) or photoconductor.

A photoresistor consists of a semiconductor with high resistivity. If the incident light is of sufficiently high frequency, the photons absorbed by the semiconductor will give the bound electrons enough energy to jump into the conduction band, the free electrons (with their electron holes) thus produced lowering the resistance of the whole [33](see figure IV.7).

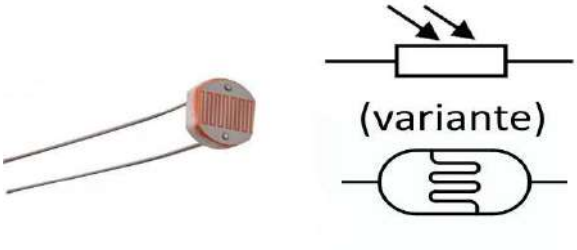


Figure VI.7.LDR sensor

IV.5.2.Actuators

In addition to ON/OFF actuator we use a servo motor which is a device that can turn to a specified position. Usually, it has a servo arm that can turn 180 degrees. Using the Arduino, we can tell a servo to go to a specified position and it will go there (see figureVI.8).



Figure IV.8.servo motor

The servo motor used in this example includes three pins:

- VCC PIN (typically red) needs to be connected to **VCC** (5V)
- GND PIN(typically black or brown) needs to be connected to **GND** (0V)
- SIGNAL PIN (typically yellow or orange) receives the PWM control signal from an Arduino's pin (see figureVI.9).

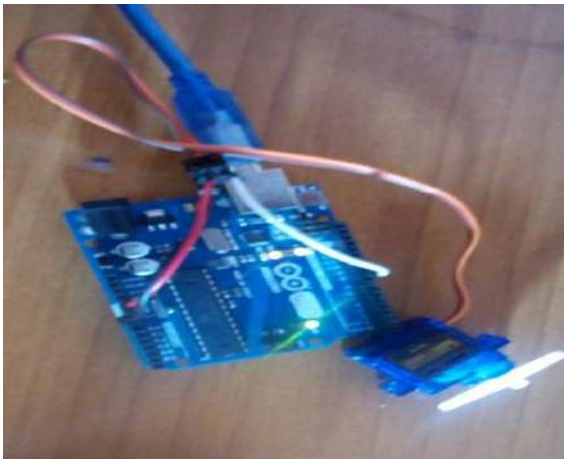


Figure IV.9. Mounting principle servo motor

Servo motors were first used in the Remote Control (RC) world, usually to control the steering of RC cars or the flaps on a RC plane. With time, they found their uses in robotics, automation, and of course, the Arduino world (see figureVI.10).

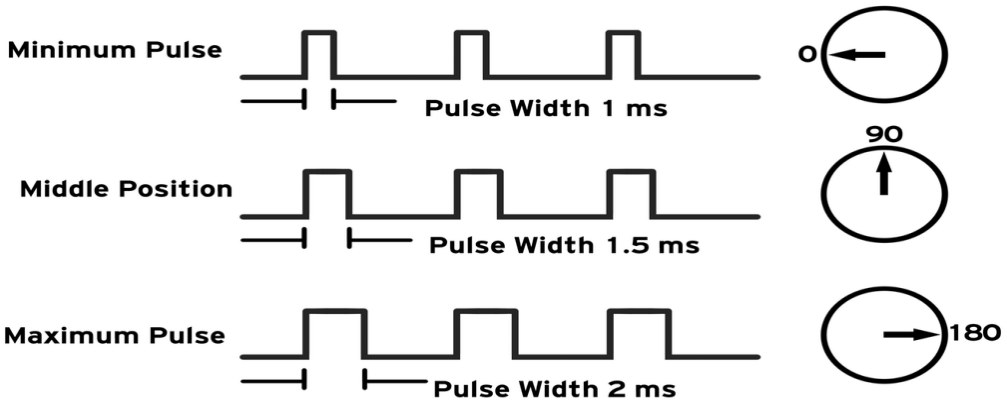


Figure IV.10.how does the servo works

Servos are using just one input pin, they receive the position from the Arduino and they go there. Internally, they have a motor driver and a feedback circuit that makes sure that the servo arm reaches the desired position. But what kind of signal do they receive on the input pin.

It is a square wave similar to PWM. Each cycle in the signal lasts for 20 milliseconds and for most of the time, the value is LOW. At the beginning of each cycle, the signal is HIGH for a time between 1 and 2 milliseconds. At 1 millisecond it represents 0 degrees and at 2 milliseconds it represents 180 degrees. In between, it represents the value from 0–180. This is a very good and reliable method. The graphic makes it a little easier to understand.

Remember that using the Servo library automatically disables PWM functionality on PWM pins 9 and 10 on the Arduino UNO and similar boards (see figureVI.10) [34].

IV.6.Communication with the Module CAN-BUS (MCP2515)

We explained in the previous chapter the module MCP2515 we give its definition, characteristics, features, library messages of transmitting and receiving.

In the following we explain how to interface a CAN module MCP2515 with Arduino and the possible communication between two Arduino via the CAN protocol.

IV.6.1.SPI Link

The SPI (Serial Peripheral Interface) link is a serial bus used for the synchronous transmission of data between a master and one or more slaves (multipoint). The transmission takes place in full duplex. The master (very often a µC) generates the clock and initiates the data transmission in selecting the slave (converter, shift register, memory ...) with whom it wants to communicate.

Each slave is selected by an SS (Slave Select) line and is active only when it is selected.

The SPI bus consists of two data lines and two signal lines, all unidirectional:

- MOSI (Master Out Slave In): On the MOSI line the master transmits data to the slave.
- MISO (Master in Slave Out): On the MISO line, the slave transmits data to the master.
- SCK (SPI Serial ClocK): Clock signal, generated by the master, which synchronizes the transmission.
- SS (Slave Select): This signal at NL0 is used to individually select (address) a slave. There are as many SS lines as there are slaves on the bus. The possible number of SS connections of the master will therefore limit the number of slaves (figure IV.10).

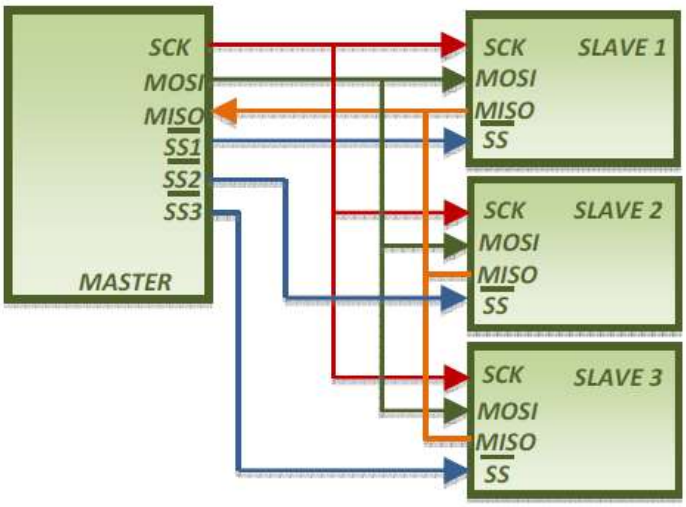


Figure IV.11. Example of an SPI bus configuration with one master and several slaves [35]

IV.6.2.pin map of CAN BUS Shield V2.0

CS pin: SPI_CS pin of V1.2 is connected to **D9** by default

SPI pins: The SPI pins (SCK, MISO, MOSI) are routed to the ICSP pins by default. But for some boards, the SPI pins are located at D11~D13.

As mentioned earlier, the CAN IC controller facilitates the SPI communication protocol for interfacing with any microcontroller. Therefore, connect the SPI pin, i.e. SCK, MOSI (SI), MISO (SO) and CS of the MCP2515 module to the corresponding SPI pins of Arduino (see figureVI.11).

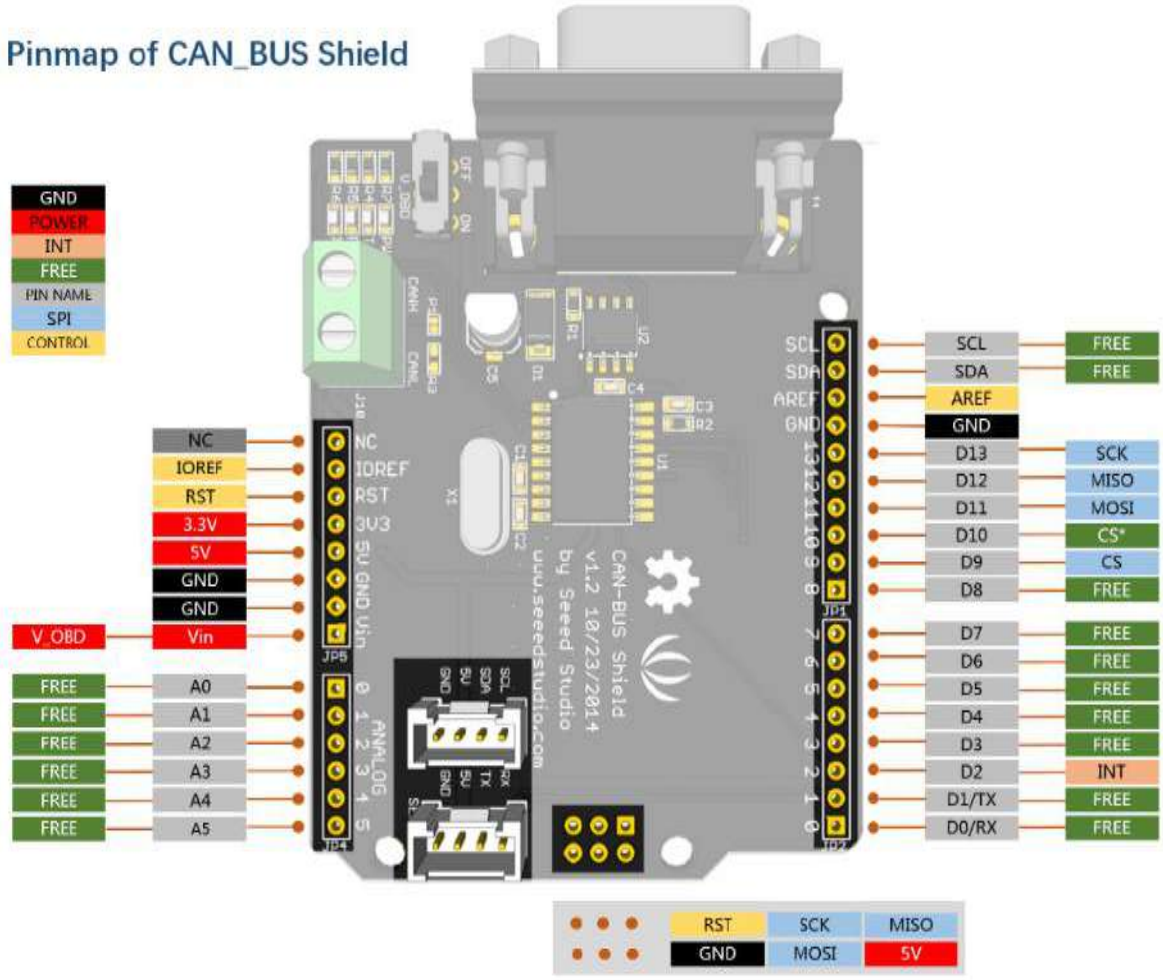


Figure VI.12.pin map of CAN BUS Shield V2.0

Make two of these connection one pair acts as transmitter and the other as receiver

Therefore for communication between transmitter and receiver connect the CAN_H and CAN_L pins of each MCP251 module (CAN_H to CAN_H, CAN_L to CAN_L) as show the (figure VI.12).

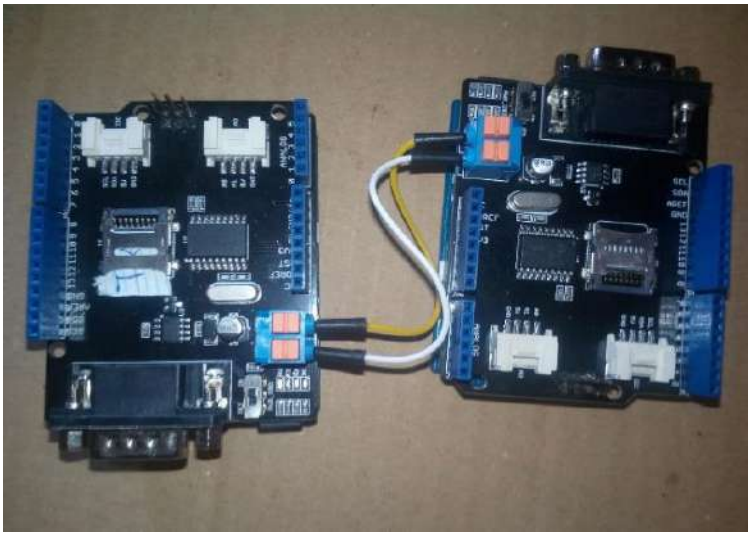


Figure VI.13.connection pins of two modules

IV.7.Communication

We work on our project with two methods of communication, wired communication using the MCP2515 module and communication with a serial port.

Transmission with MCP2515:

- **Initialization**

  In this step we will:

- Add the “SPI”, “mcp_can” libraries,
- Define “SPI_CS” pin and call the library to create the wired communication.
- Initialize the variables used in the program and create a database at transmit.

- **Configuration**

  The functions that we will use in this step are carried out by means of the library “CAN-BUS”:

- Start communication “Serial.begin(115200);”:
- Data transmission speed: (CAN_OK! = CAN.begin (CAN_500KBPS), you can work with: CAN_5KBPS until CAN_1000KBPS.
- The CAN ID is given as 0x036 (As per choice) and DLC as 8 because we have 8 data.

IV.8.Communication Protocol

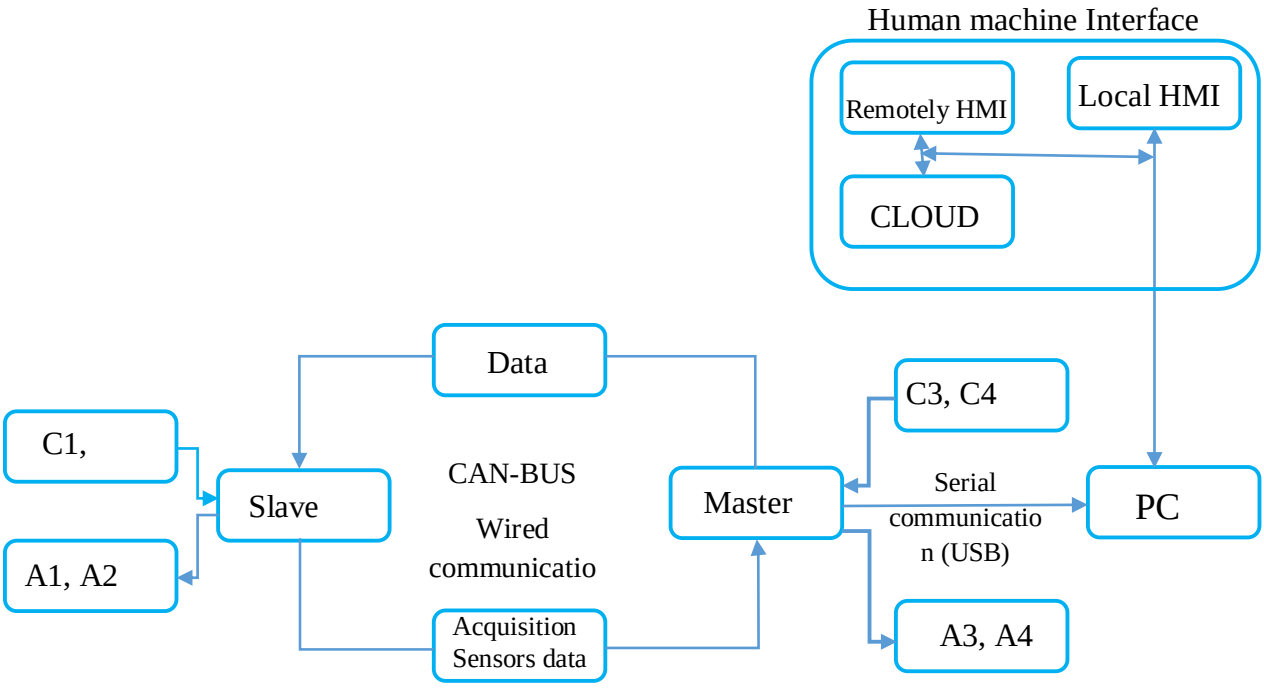


Figure VI.14. Master/slave communication

The operation of this project is simple because all the work is done by the libraries (SPI and CAN). Since CAN is a message-based communication, you have to send or receive a message between 0 and 8 bytes depending on the configuration of the inputs and outputs.

In our program arduino, we can extract the can id (identification CAN) of modules, Ide (state of frame: extended or standard frame), DLC (data area: contain the data).

We use a node red interface (we will explain it later) which is directly connected to node 1 (MASTER) with serial port and handles the actuators (A1, A2) and sensors (C1, C2), this node 1 sends the data to the CAN port. Stored for the interface, the communication between node 1 and 2 is bidirectional, so node 2 (Slave) sends the data (A3, A4), (C3, C4) to node 1 and node1 sends the data receive to the interface, so Node 1 acts as a bridge between node 2 and the interface(see figureVI.13).

IV.9.Operation of the global system

We present in figure (IV.13) the implementation of the materials to create our supervision system. Exchange data between the Arduino master and the slave is carried out by the MCP2515 module (wired communication), so that the PC can communicate with the master, it uses the USB port (serial communication).

Generally, arduino Uno has interfaced with the MCP2515 CAN module via SPI pins, you will need two Arduino boards of any type or other MCU and two of these MCP2515 CAN modules to build a basic configuration to begin with. The software is simplistic and the transmission program simply transmits certain data to the bus every 1/2 second and the receiving program uses the query to determine if there is a message available and displays it in the Serial Monitor window that will look like something like this. It displays the CAN ID, the number of bytes of data sent, and then displays the bytes of data that were transmitted. In our case, we simply increment the data bytes in each message sent (see figureVI.14).

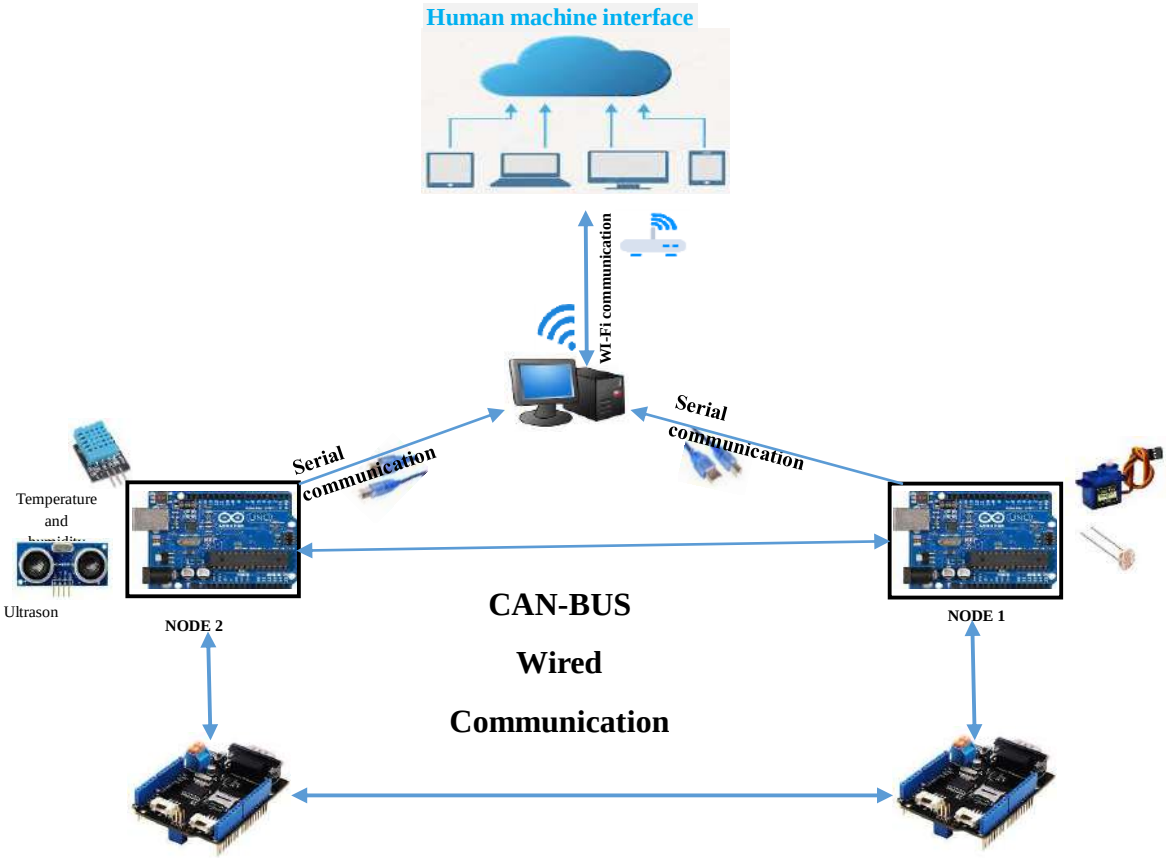


Figure VI.15. Global architecture of the proposed system

VI.10.how communication occurs in control and monitoring system

The can bus communication protocol of our project ensure the following tasks as the figure above shows (Figure IV.15):

- PC (Human Machine Interface) sends actuator commands (servo1, servo2, LED1, LED2) to node 1 (master) with serial communication (USB).
- Node 1 receives and checks the actuator commands, it only performs their commands (servo1, LED1), and send the data of (dht11) to display it on the PC (Human Machine Interface) so the others (servo2, LED2) will send to node 2.
- Communication between master and slave is done via a CAN-BUS connection, as we said previously, node 1 first works in interrogation mode to determine its frame to be executed (trigger command and display of DHT11 sensor data) if the frame is for node 2, it sends the frame to node 2.
- Node2 receives at the can bus level the frame which node1 send, it executes the commands (servo2, LED2), and sends (the data from the ultrasonic sensors hc-04, LDR) to the node1, then the node1 send the data to display in the PC (man-machine interface) with serial communication (USB).

L'émission et réception de donnes se produire dans les ½ second, une période très courte ou les nœuds après chaque émission et réception écoute de la trame pour éviter la collision.

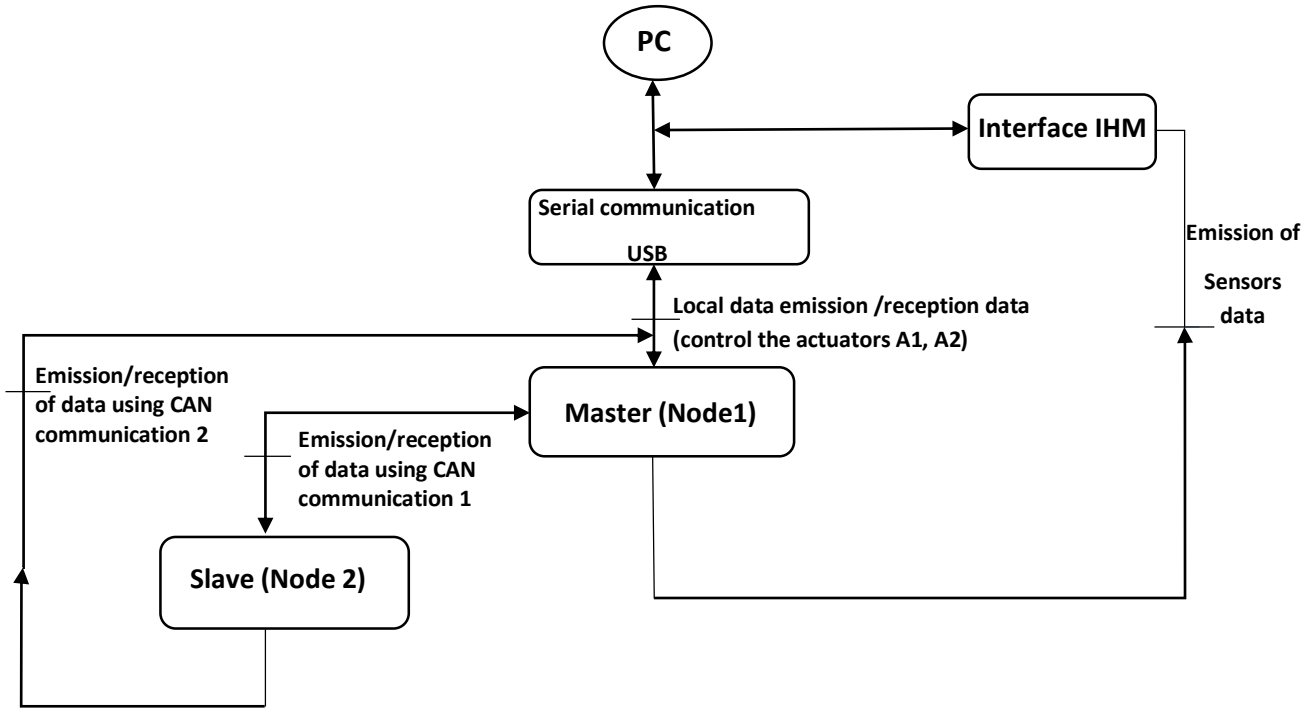


Figure IV.16.diagram illustrates the system communication method

VI.11.Human Machine Interface (HMI)

HMI is a web application accessible from a browser. It is not always easy to program the link between the hardware (sensors and/or actuators) and the web page provided to the user.

So it works like an ETL: a range of connectors, processing components, wiring possibilities between them.

Combined with a hardware solution consisting of a Raspberry and possibly an Arduino board, Node Red proves to be a very interesting alternative:

Being a very interesting alternative Node-RED is a powerful tool for building Internet of Things (IoT) applications with an emphasis on the simplification of programming that is done through predefined blocks of code, called «nodes» to perform tasks. It uses a visual programming approach that allows developers to connect blocks of code together. The connected nodes, usually a combination of input nodes, processing nodes and output nodes, when wired together, constitute a 'flow'.

VI.11.1. Programming with Node Red

VI.11.1. 1.NODE RED INSTALLATION

- **Installing on windows.**
- You must install: Node JS then Node-Red
- Install an LTS version of NodeJS.
- Install Node-red using the windows command line in a console in administrator mode:"npm install node-red" (to be checked)
- **Launch on Windows**
- Launch the windows console in administrator mode
- Go to folder: c: users/sb/appdata/roaming/npm
- Run the node-red command
- Allow access (see figureVI.16).

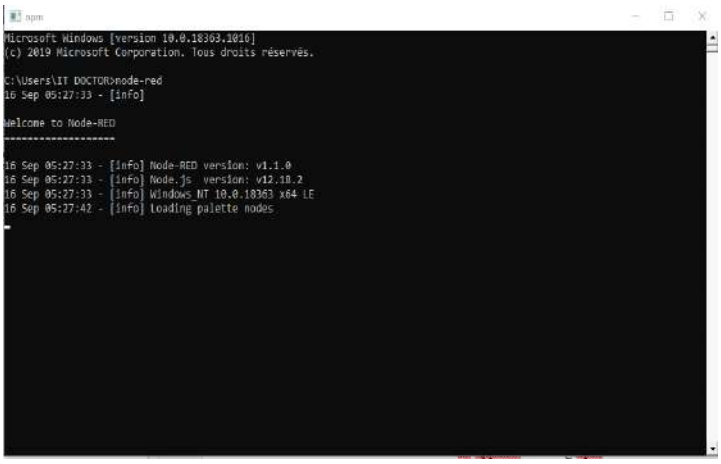


Figure IV.17. Node-red command on windows

Go with a browser to the ip:http://localhost:1880: the node-red interface works....

IP du Dashboard pour test : http://localhost:1880/ui/#

VI.11.1.2.Installation of new NODE

To install new "node" in the palette, use the right menu and choose: 'manage palette'. In 'manage palette', go to the 'install' tab and start a search: for example: dashboard or serial (for arduino):

Install the nodes you are interested in.

The nodes then appear in the palette (see figureVI.17).

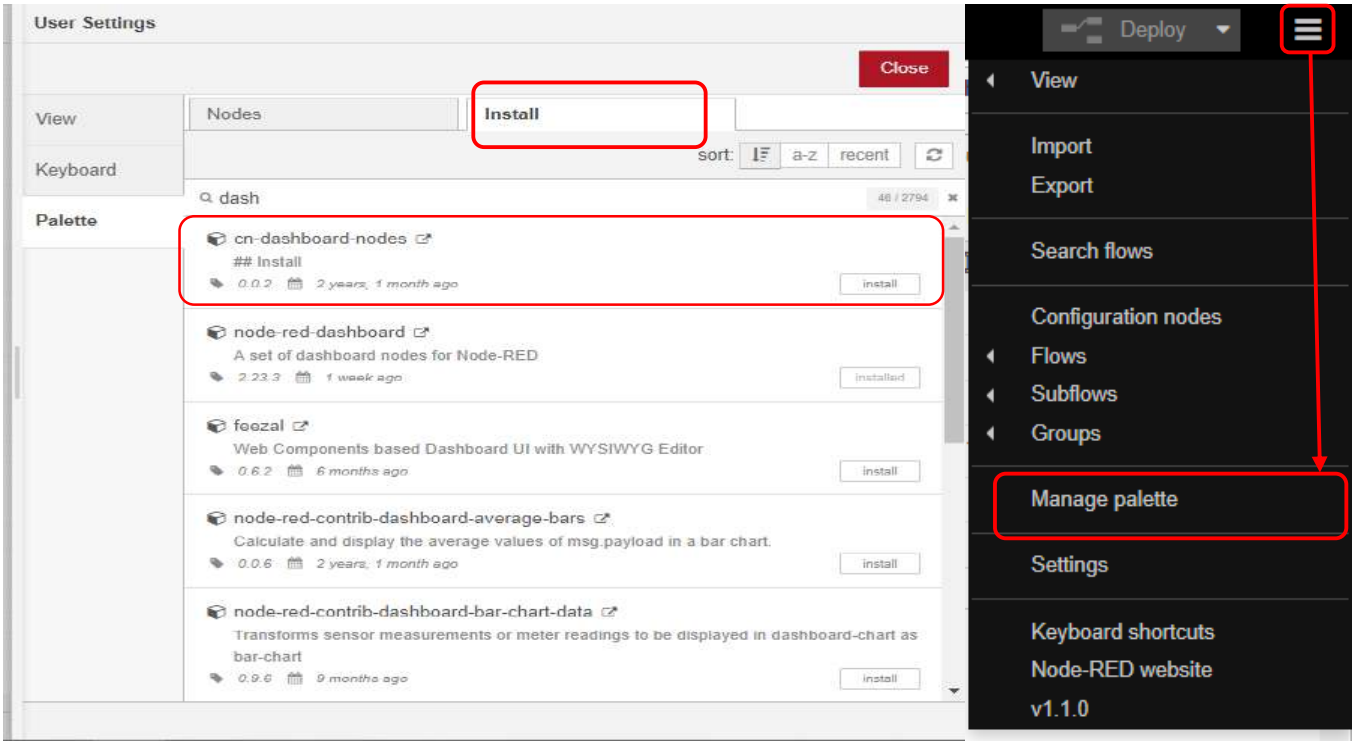


Figure IV.18.intallation of dashboard

You can install new nodes in other way by include run the following command in your Node-RED user directory, typically ~/.node-red

npm i node-red-node-serialport

The reception of the data sent by the Arduino master to PC, is done through the serial port, for the Node Red to use the USB port, one must add a "Serial" palette (see the figure IV.18).

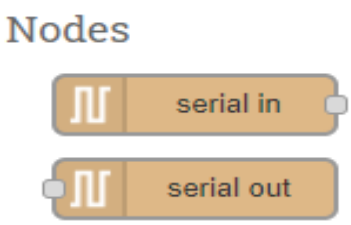


Figure IV.19. Serial palette

To receive or send via a local USB port, click on the node to configure the serial port parameters (specify COM, transmission rate...etc.) (See figureVI.19).

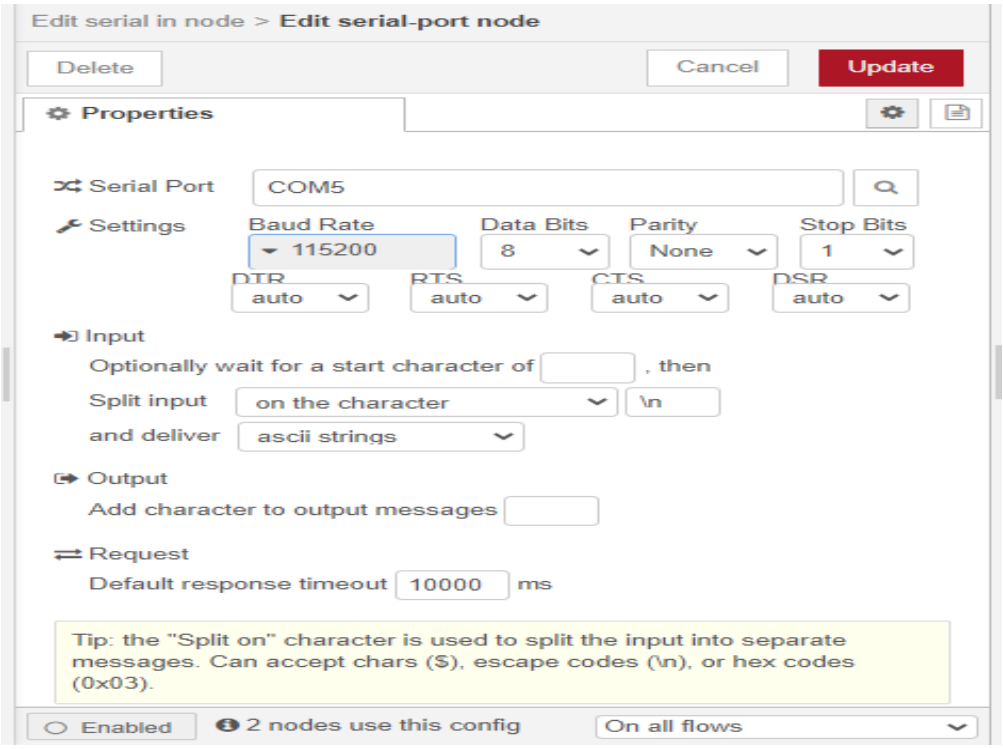


Figure IV.20. Serial Port Configuration

VI.11.1.3.Flow node red/actuators and flow sensor/node red

Interface our system of monitoring and control with node-red (see figure VI.19).

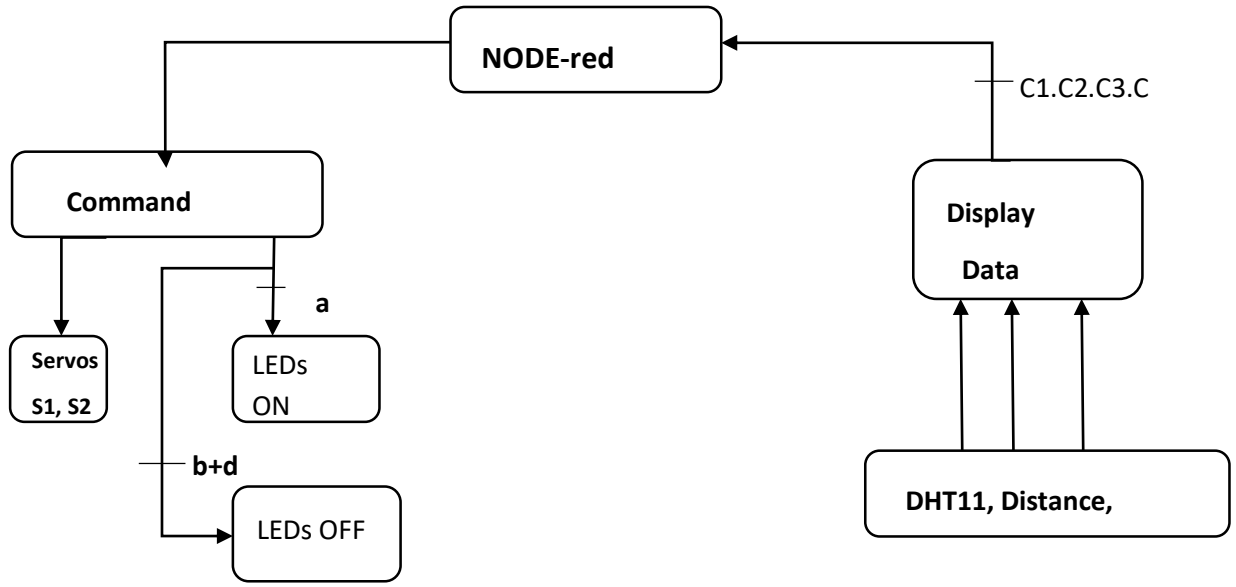


Figure VI.21.interface sensors and actuators with node red

VI.11.1.4.data splitting

We could send the measurements directly to the serial port of the Arduino with a separator, for example a special character (|, -, #…) but this strategy implies that we know precisely the position of each data. No problem with one or two data, it becomes more complicated when there are about ten. Another problem is the conversion of string which is an eternal problem in computers.

To avoid all this problem, we will format the data and indicate each time what it corresponds to. For this, we will use the JSON format. The advantage is that it is supported by all modern languages. This is even the default data structure of javascript, the language on which Node-RED is based.

The Serial port sends the sensor data (humidity, temperature and light) to the JSON function information from the Arduino (master) card reaches the PC via the serial port. Named Com3. So we are going to place on our flow a Serial node and configure it for a connection to this port. The internal configuration of the Serial input function is given in the (figure IV.19), (figureIV.21).



Figure IV.22. Data separation node

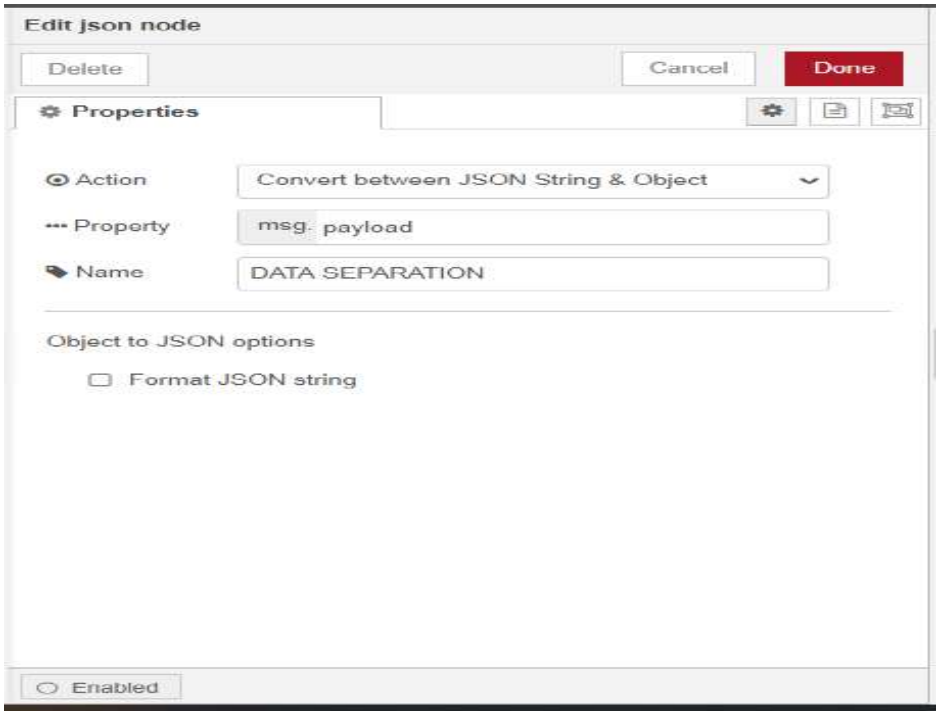


Figure IV.23. Configuration of the json function

The serial communication offers us a data frame, the latter consists of temperature, humidity, light and distance, so we need to separate these data to display them separately, for this we use a predefined processing function «Json», (see figure IV.23)

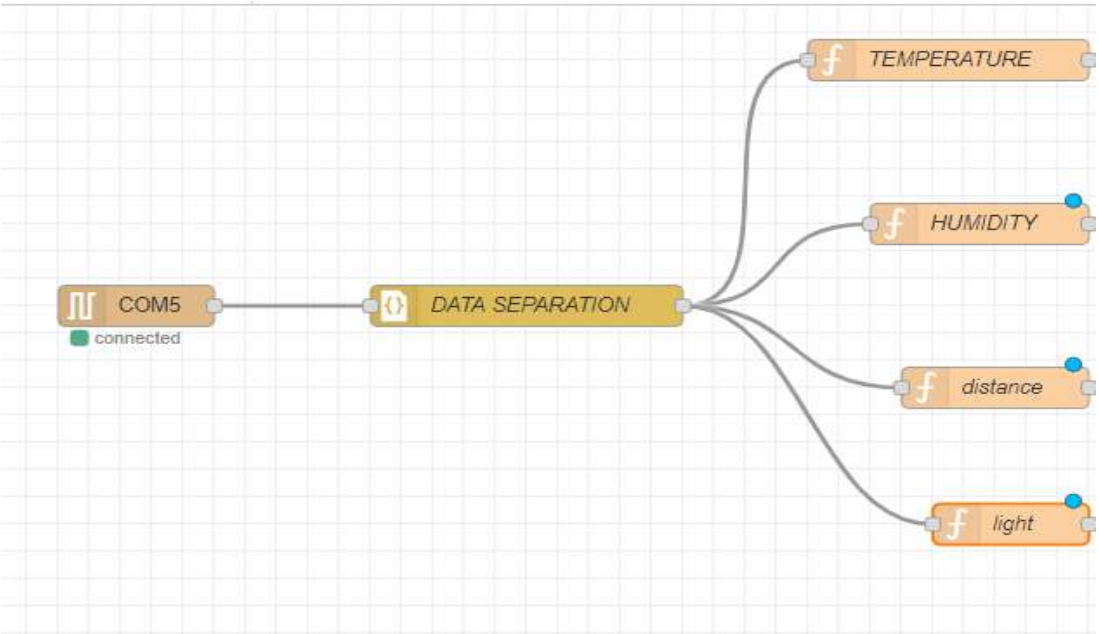


Figure IV.24. Data separation from sensors

We get a data on the serial port of the Arduino, we will convert it into a JSON object that can be used by Node-RED using the JSON node. Place it on the flow and connect it to the serial port.

Node-RED transfer of messages (msg) in JSON format between each Node (program node). The data is in the payload key. The first line extracts the DHT11 sensor measurement and overwrites the current payload (see figureVI.24).

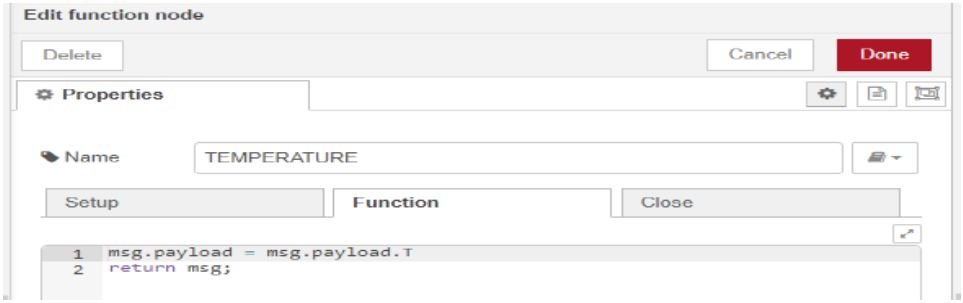


Figure IV.25. Internal configuration of functions

The function returns the updated message (msg). It returns only the measure of the first sensor. We do the same for the rest sensors by adding a flows

IV.11.2.data visualization

IV.12.1.temperature, humidity, distance and light display

We used dht11, HC-04 ultrasound and LDR components to display the (temperature, humidity, distance, light) variations, in order for these components to be properly placed on the web page (see figure IV.22**)**, we put them in a Tab (HOME), then in a group (sensors), in the group selection box, for example we configure a gauge to display the temperature, see the (figure IV.25).

Display the rest of sensors using the gauge it is the same configuration of temperature gauge.

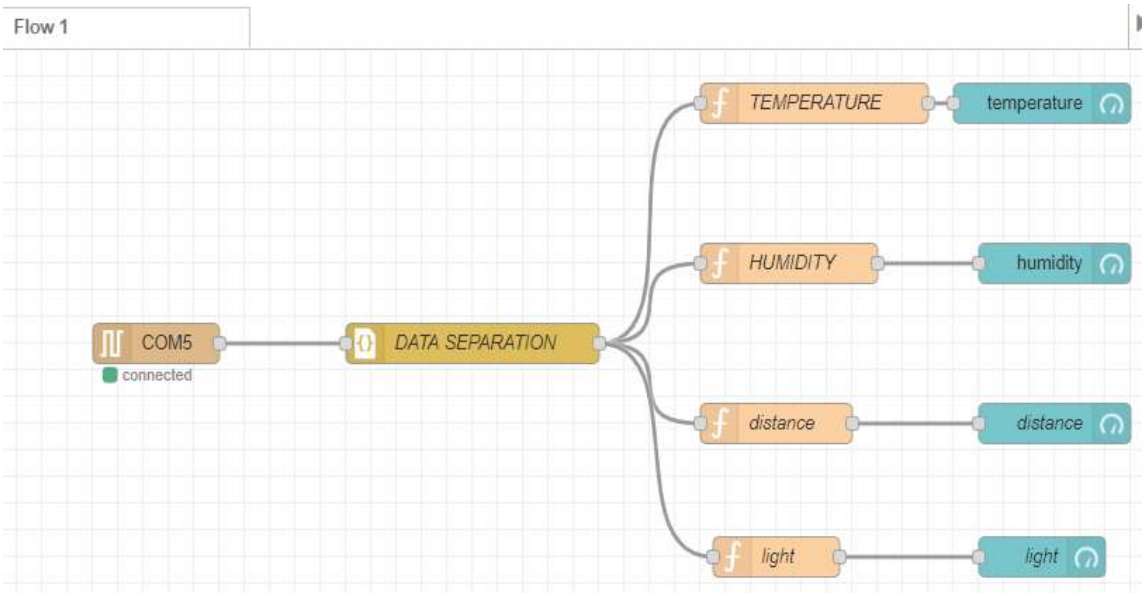


Figure IV.26. Temperature, humidity, distance and light display

The internal configuration of the gauge and the graph is shown in (Figure IV.26)

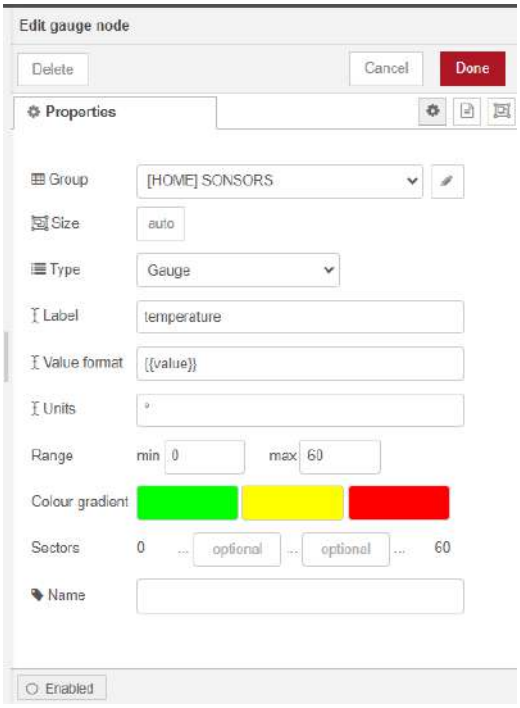


Figure IV.27. Gauge Configuration

IV.12.2.Display of controls for actuators

For controlling actuators we need specific devices in dashboard, for this we used switches.et slider Commands that the servo, and the light are sent from HMI via port COM5, see the (figure IV.27).

The control of the (servo1, servo2, LED1, LED2) operate to be sent by the supervisor via this port, (see figure IV.27).

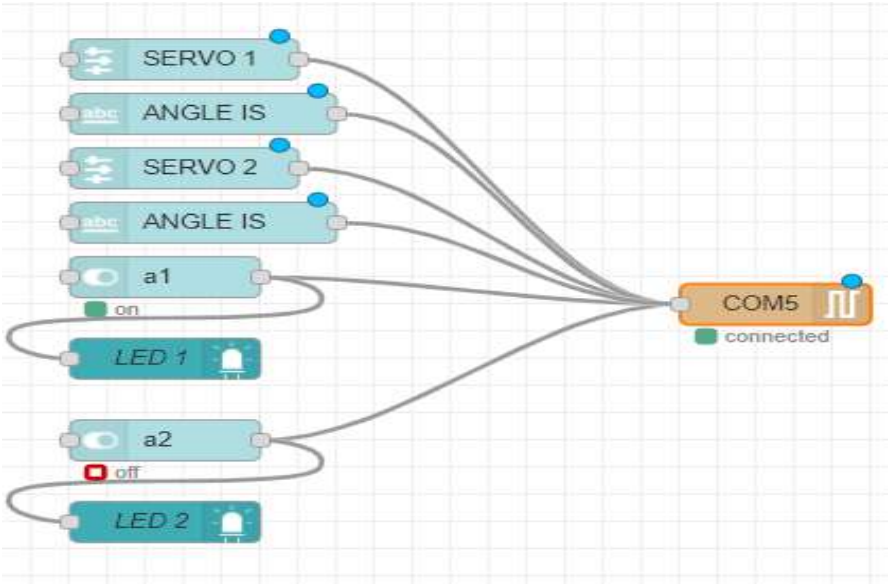


Figure IV.28.actuators control

You can control the servos in the arduino program from two methods:

- ❖ Create a simple program with instructions to control the servo angle by sending numbers to the desired angle or by icon slider in Dashboard.
- ❖ Directly use Library Firmata in the example of arduino software.
  When we use this library we have to download arduino from the dashboard because we can't use the serial.
  Firmata is a protocol for communicating between an Arduino (as well as other microcontrollers) and the host computer, providing direct access to the IO pins.

To install it in node red on command npm install node-red-node-arduino (See figure IV.28).

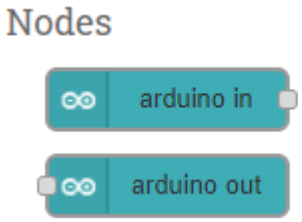


Figure IV.29.firmata nodes

To control the LEDs, you can use numbers or characters to change the state of the LED On (a, c) or OFF (b, d).

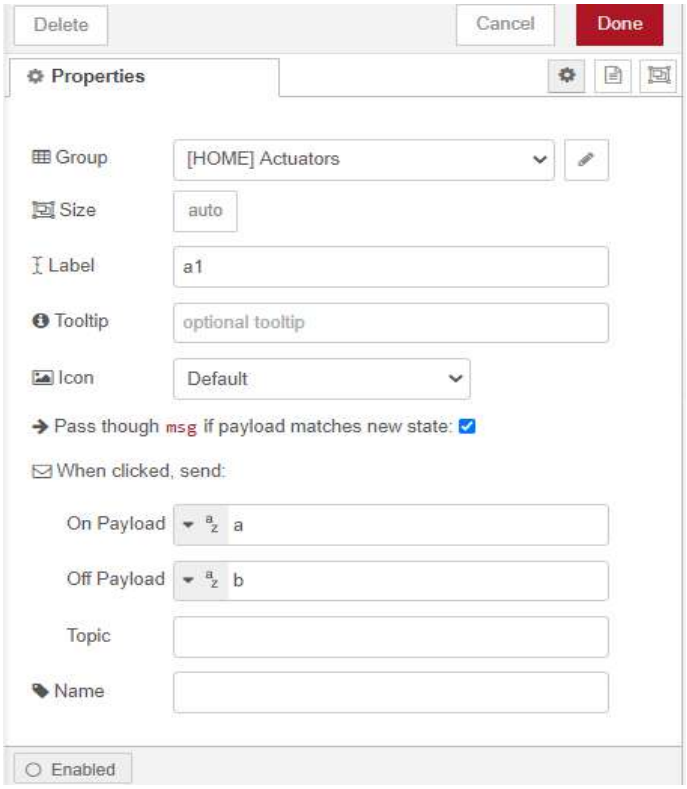


Figure IV.30. Swich configuration

IV.12.3.Visualisation d'IHM

We used the gauge to visualize the quantities (temperature, humidity, distance, light) and so we can visualize their variation in the HMI. The different sizes and states to display in the HMI are organized in the form of panels (see figureVI.30).

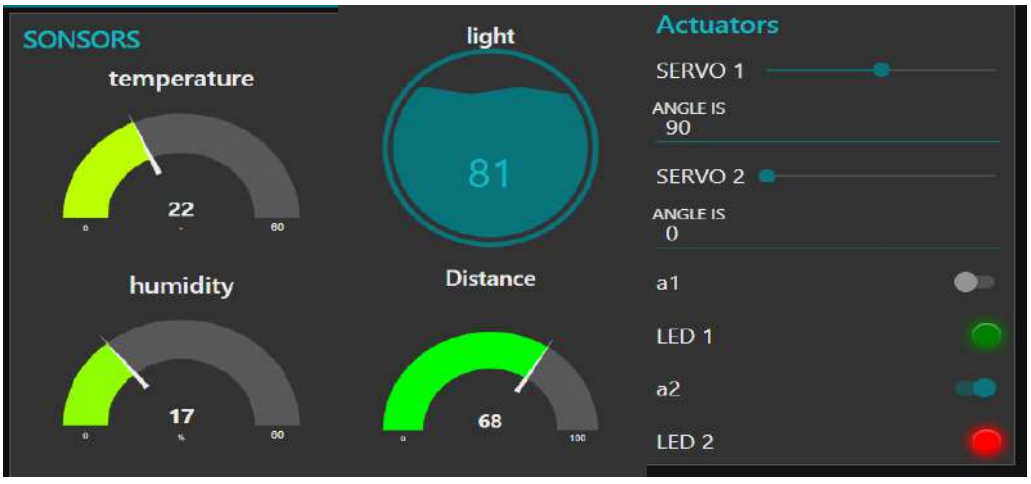


Figure IV.31. Display different parameter of system

IV.13.conclusion

In this chapter, we have given a presentation on our control and monitoring system, explaining its implementation clearly and precisely, and then we have presented the hardware side.

We have mentioned all the equipment used according to our needs, the communication between these devices being managed by the bus can protocol, this protocol manages the transmission and reception of data with the wired communication module. The development of the node-red web application is based on the notion of the internet of things.

General Conclusion

General Conclusion

In this work, a new architecture for a low cost and flexible home control and monitoring system (energy management, security, comfort ...) is used with CAN bus based on MCP2515.

The most important advantages of CAN-BUS are: It eliminates the need for multiple long distance cables by allowing electronic devices to communicate with each other with a single multiplex wire. This multiplex wire has for bus communications and all MCUs connect to these same 2 wires CAN bus hardware manages the movement of messages through the bus with collision prevention and error correction without requiring MCU overload. CAN bus has multi-master capability, which means any node on the bus can initiate communication with any other node in a network.

This project allowed us to discover the CAN bus, and to realize the importance of this network in the applications of practical life. The efficiency and robustness of the CAN protocol has brought it to use in many industrial applications, especially those requiring high throughput up to 1 Mbits / s with a very low error rate.

The implementation of project-related systems is carried out using electronic components for processing and communication (the Arduino and the wired communication module MCP2515) which are compatible with on-board applications and solutions internet of Things.

We tried our best to prototype this work, but we didn't had enough time to well present it due to the covid 19 consequences, as a perspective of this work, we propose to carry out our work in practice, integrate more CAN-BUS nodes with more sensors and actuators to supervise and control other parameters such as anomalies detection.

REFERENCES

REFERENCES

[1] WILFRIED Voss. Controller Area Network (CAN) prototyping with arduino: CAN Driver Implementation & Library Function, CAN Network Monitoring, Simulation & Diagnostics, CAN Data Display via Windows/Linux GIO. USA, 2014. ISBN-10: 1938581172,ISBN-13: 978-1-938581-17-5.

[2]. CAN-BUS SHIELD V2.0. [ONLINE]. [Consulted MARCH 20, 2019].Available on: HTTPS://WIKI.SEEEDSTUDIO.COM/.

[3] Drew Handricks. The history of smart home. [Online]. [Consulted April 22, 2014]. Available on https://www.iotevolutionworld.com/m2m/articles/376816-history-smart-homes.htm

[4] Kaitlyn Richardson. Smart Home Technology. [Online]. [Consulted 31 March 2020]. Available on https://universaldesign.org/smart-home-technology.

[5] Nadhrah Fuad. Smart Homes. . [Online]. [Consulted 6 February 2015] Available on: https://prezi.com/_bbgjgqvvlgs/smart-homes/.

[6] IJENKO - Internet of Things Platform. Smart Home for Energy Saving. [Consulted on 2020]. Available on: http://www.ijenko.com/smart-home-for-energy-saving/.

[7] TechTerms.Smart Home. [Online]. [Consulted on July 19, 2014]. Available on: https://techterms.com/definition/smart_home.

[8] MEKHALFIA,Toufik. GHADBANE Toufik. Etude et réalisation d'un système de commande à distance des installations électriques pour la domotique.63p.Master memory. Electronics of embedded systems: M'SILA, Mouhamed BOUDIAF university: 2018.

[9] SafeTech. Smart Home Monitoring Toronto - SafeTech Security. [Accessed: 4 September 2020] Available on : https://safetechalarms.com/residential/smart-home-monitoring-toronto/ .

[10] BH Magazine. Maison Connectée: Est-Ce intéressant Dans La Vie De Tous Les Jours? [Online]. [Consulted: 24 JUIN 2019]. Available on: https://www.bhmagazine.fr/1307/maison-connectee-est-ce-interessant-dans-la-vie-de-tous-les-jours/#:~:text=La%20domotique%20aide%20%C3%A9galement%20les,du%20travail%20%C3%A0%20la%20maison.

[11] M. Li et al, « Science Direct Smart Home : Architecture, Technologies and Systems Smart Home : Architecture, Technologies and Systems ». Procedia Comput. Sci., vol. 131, pp. 393–400, 2018, doi: 10.1016/j.procs.2018.04.219.

[12] Caroline Forsey.The 13 Best Smart Home Devices & Systems. [Online]. [Consulted: 28 JUNE 2019] Available on: https://blog.hubspot.com/marketing/smart-home-devices.

[13] Yassir Sennoun. IOT & Les Protocoles De Communication Pour Les Réseaux Sans Fils Et Les Filaire Comment Choisir? [Online]. [Consulted: 29 AUGUST 2018].

Available on: https://blog.engineering.publicissapient.fr/2018/08/29/iot-les-protocoles-de-communication-pour-les-reseaux-sans-fil-et-filaires-comment-choisir/.

[14] geeksForGEEKS.Diffrence Between Serial and Parallel Transmission. [Online]. [Consulted: 22 APRIL 2020]. Available on: https://www.geeksforgeeks.org/difference-between-serial-and-parallel-transmission/.

[15] Abbes BOUGUEZMIR.Bus CAN Et Application.77p.Master thesis: Industrial Computing: Oum El Bouaghi, University of Larbi Ben M'hidi.2015.

[16] Stephen St. Michael.Introduction to CAN (Controller Area Network). [Online]. [Consulted: 19 FEBRUARY 2019]. Available on: https://www.allaboutcircuits.com/technical-articles/introduction-to-can-controller-area-network/.

[17] Yida.Introduction to CAN-BUS and how to use it with arduino. [Online]. [Consulted: 27 NOVEMBER 2019]. Available on: https://www.seeedstudio.com/blog/2019/11/27/introduction-to-can-bus-and-how-to-use-it-with-arduino/#:~:text=The%20CAN%20message%20will%20contain,from%20one%20device%20to%20another.

[18] Steve, Corrigan. Introduction to the Controller Area Network (CAN). Dallas, Texas. 2003, 16p. Texas Instruments.

[19] Sylvain, Choisel. Matthieu, Liger. Vincent, Oberlé, Gestion de bus CAN. 3-February 2000.72p.

[20] Paolo Medagliani.Internet of things Applications-From Research and Innovation to Market Deployment. A. Duda. Rousseau. River Publisher Series in Communication. January 2014.143p.

[21] Manual Ortiz. Manuel Díaz Francisco Bellido. Edmundo Sáez. Smart Home Automation Using Controller Area Network. Conference: International Symposium on Distributed Computing and Artificial Intelligence, DCAI 2011, Salamanca, Spain, 6-8 April 2011. Law and rural space.10.1007/978-3-642-19934-9_21.

[22] Paolo Medagliani.Internet of things Applications-From Research and Innovation to Market Deployment. A. Duda. Rousseau. River Publisher Series in Communication. January 2014.143p.

[23] Rajiv. What the Major Components of Internet of Things. [Online]. [Consulted: 10 JANUARY 2018]. Available on: https://www.rfpage.com/what-are-the-major-components-of-internet-of-things/.

[24] Pallavi Sethi .Internet of Things: Architectures, Protocols, and Applications. Smruti R. Sarangi. Hindawi Journal of Electrical and Computer Engineering. Received 12 August 2016; Accepted 18 December 2016; Published 26 January 2017. Rajesh Khanna. Volume 2017. 25 p.

[25] TutorialPoint.Internet of Things-Overview. [Online]. [Consulted: 07 MARCH 2020]. Available on: https://www.tutorialspoint.com/internet_of_things/internet_of_things_overview.htm.

[26] Mark Silanus.Tutorial Node Red. [Online]. [Consulted: 18 FEBRUARY 2018]. Available on: http://silanus.fr/sin/?p=984.

[27]Arduino. Arduino Uno Rev 3. [Online]. Available on: https://store.arduino.cc/arduino-uno-rev3.

[28] WILFRIED Voss. Controller Area Network (CAN) prototyping with arduino: CAN Driver Implementation & Library Function, CAN Network Monitoring, Simulation & Diagnostics, CAN Data Display via Windows/Linux GIO. USA, 2014. ISBN-10: 1938581172, ISBN-13: 978-1-938581-17-5.

[29] ELECTRONICS HUB.Arduino MCP2515 Bus Interface Tutorial. [Online]. [Consulted: 23 AUGUST 2018]. Available on: https://www.electronicshub.org/arduino-mcp2515-can-bus-tutorial/.

[30] Semih Ozebir. Learn how to use Can Bus module. [Online]. [Consulted: 13 JULY 2018]. Available on: https://maker.pro/arduino/projects/arduino-can-bus-wired-communication.

[31] skywodd. Utiliser un capteur de température et d'humidité DHT11 / DHT22 avec une carte Arduino / Genuino. [Online]. [Consulted: 09 JULY 2016]. Available on: https://www.carnetdumaker.net/articles/utiliser-un-capteur-de-temperature-et-dhumidite-dht11-dht22-avec-une-carte-arduino-genuino/.

[32] skywodd.Mesurer Une Distance Avec Un Capteur à Ultrasons HC-SR04 Et Une Carte Arduino / Genuino. [Consulted: 21 FEBRUARY 2017]. Available on: https://www.carnetdumaker.net/articles/mesurer-une-distance-avec-un-capteur-ultrason-hc-sr04-et-une-carte-arduino-genuino/.

[33] Photoresistence. [Consulted: 07 APRIL 2010]. Available on: http://www.composelec.com/photoresistance.php.

[34] cornelam.Arduino Servo Motor. [Consulted: 27 July 2015]. Available on: https://www.instructables.com/id/Arduino-Servo-Motors/.

[35] Luc De Mye. Liaisons série et parallèle. . Available on: http://www.courstechinfo.be/Techno/LiaisonsSP.html.