Politecnico di Torino

Dipartimento di Elettronica e Telecomunicazioni

Arduino-Based Wireless Sensor Networks Programming Using Matlab

Farnaz Mehdipour

Supervisor
Prof. Ladislau Matekovits

A Thesis for the Degree of Master of
Communications and Computer Networks Engineering

Politecnico di Torino

November, 2020

Abstract

Smart home automation systems are interactive systems providing interface between humans and household devices. Technological advancements have led to cost reduction and availability increase of sensors, wireless devices, and microcontrollers. Besides all of these, because of emerging of wireless sensor networks, realization of integrated home automation systems has become more cost-effective and easier than ever before. Regarding microcontroller, Arduino offers a high degree of freedom for implementing control systems of smart homes. On the other hand, Matlab is a widely used and well-documented multidisciplinary software that can communicate with many hardware. Exploiting advantages of the wireless network, Matlab, and Arduino hardware, this thesis presents a flexible and low-cost wireless sensor network based on Arduino hardware and Matlab software platforms.

The thesis is organized as follows. Chapter 1 presents background, motivation, and overview of wireless sensor networks for home automation applications. A literature review is carried on in Chapter 2. Matlab package for interfacing with Arduino is described in Chapter 3. Chapter 4 deals with hardware and Software details of the implemented wireless sensor network. Finally, in Chapter 5, operating results of the implemented network and details of the designed Graphical User Interface (GUI) application are presented.

Acknowledgment

I would like to express my deepest gratitude to my supervisors, professor. Ladislau Matekovits for his help and support with patience throughout my Master degree research.

This thesis is sincerely dedicated to my parents who showed me the true love in life and to my loving sister and brothers, And my love Adel, without his help, I wouldn't have been able to follow this path.

Contents

List of Figures

CHAPTER 1.

1. Background, Motivation, and Overview

1.1. Motivation

The present century is the age of smart technology. Modern life without electricity is unimaginable, but that has changed. Now it can be said that everyday life without the Internet is unimaginable. Due to development of the communication technology, automated and intelligent systems have reached higher level of advancement.

In the modern era, the fast progress of technology is inevitable and remarkably impacts our life. This progress has a long history of heading the modernization to its current level. Nowadays, innovation has become a harmonious part of people's lives. This has affected and continues to affect many parts of daily life, helping better social associations, ease of transportation, the ability to enjoy the excitement and the media, and advances in pharmacy. For example, smartphones play a substantial role in our daily life. Another important technology impacting our life is the Internet. If we look at the past few years, people have used washing machines, water heaters, hairdryers, etc. in housework. We can consider these machines as the beginning of home automation.

The fast-technological progress has remarkably altered our way of life. Now, electricity, television, internet, and telephone are inevitable parts of our living places. Many of works that used to be difficult operations nowadays are much more effortless thanks to available dedicated apparatus. As an illustration, the dishwasher machine makes cleaning of dishes an easy and automated work. Also, now it has become possible and quite normal to take shower at any time thanks to hot-water heaters. As another example, food storage is completely left to the refrigerators.

Heating, air conditioning, and air conditioning devices made our homes more comfortable and convenient environments. With a large number of available household appliances, an intelligent and integrated controlling system has turned into a more and more advantageous and desirable system. In the examples above, the automation system for homes provides the communication among different electrical/electronic devices and easily provide better energy efficiency and security. This integration of scattered devices is only possible through the wireless sensor network. The wireless sensor network is made up of independent spatial distributed appliances with wireless communication enabling observation and control of physical or environmental states by employing sensors and microcontrollers [1].

Usage of wireless sensor network technologies has been advancing rapidly in recent years. In many cases, the need for it is felt. These technologies, for example, have been used to monitor heart rate [2], climate [3], agriculture [4,5] and many other applications. The advantage of these systems/technologies is obvious because it significantly reduces the cost and extra work on wiring, as well as allows settings that allow you to communicate data over longer distances. In addition, the terminals are easily removable, and the devices use less energy. However, these sensors have been used using older analog sensors that use microcontroller analog pins [6], thus limiting the number of sensors that can be used for monitoring.

Recent developments in wireless sensor networks show a state-of-the-art direction in the development of OpenSource hardware/software. It uses the open-source Arduino hardware platform to provide a dynamic wireless sensor network system. On the other hand, it is a less expensive design than other designed networks and also very easy to work with.

Therefore, some of the major barriers to transforming sensor networks in engineering, commercial, and scientific applications in the past have been the lack of flexibility, scalability, reliability, performance, and sustainable maintenance. Some of these challenges have been addressed in this project.

1.2. Wireless transmission

Whenever we talk about wireless networks, there is always a lot of discussion about electronic devices. A general definition of wireless is a connection in which

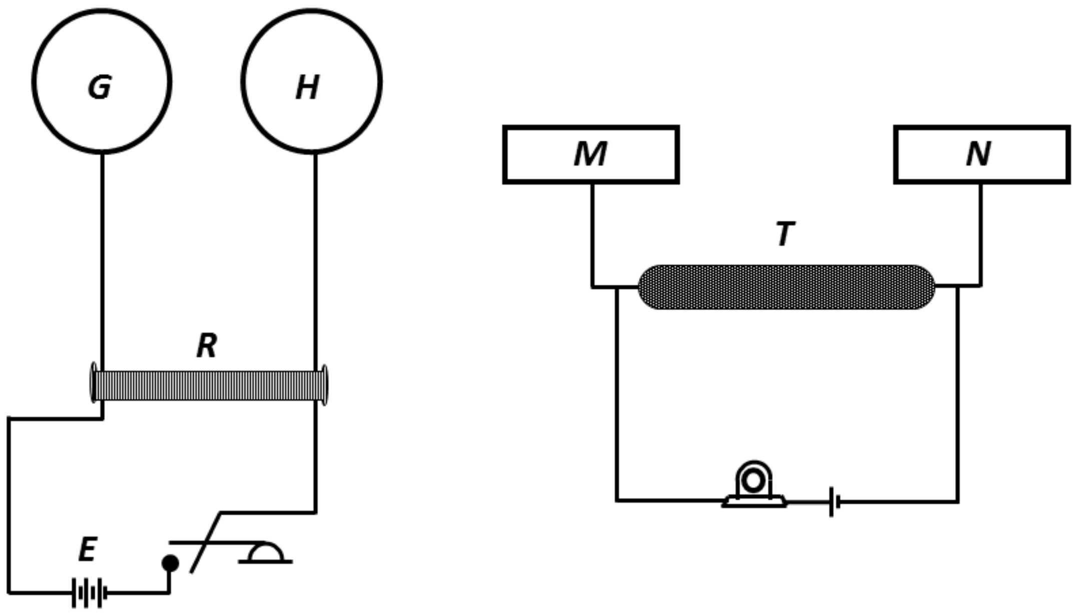


Figure 1.1. Marconi sender and receiver structure.

the wire is not used as an interface for transmitting the information. This definition is very simple, which is why wireless communication is so widespread.

Accordingly, electromagnetic waves are used in all wireless communication methods. These waves are caused by changes in the electromagnetic field in the transmitter. The main feature of this type of wave is the lack of a need for a material environment for propagation. The rate of change in this field indicates the frequency of the electromagnetic wave. In theory, this frequency can have any value. Depending on their frequency, electromagnetic waves are divided into several general categories: radio, microwave, infrared, visible, ultraviolet, X-ray, and gamma-ray (from low to high frequency, respectively).

1.2.1. Wireless transfer history

The oldest wireless communication dates back to pre-modern times, when smoke, fire, flags, etc were used to transmit messages over long distances [7]. Maxwell's mathematical theory of electromagnetic waves was proposed in 1873. Heinrich Hertz showed the existence of these waves in 1887 [8]. The radio wireless telecommunication was invented around 1897 by Guillermo Marconi [9]. Marconi managed to send a wireless telegram of the letter S about three kilometers away. Structure of Marconi sender and receiver is shown in Fig. 1.1.

Wireless telegraph was first used by the British Army in South Africa in 1900 during the Second Boer War [10]. In this war, the British navy used the Marconi device to talk between its ships in the Delagoa Bay. Until 1901, radio coverage was provided throughout the Atlantic Ocean. Since seafarers were the first customers of the wireless telegraph, wireless communication was common until 1912, when the Titanic used it to send messages of help [11]. In 1906, the radio was invented by Rajinald Abri Fasanden to send music with a range of modulation. In 1918, Edwin Howard Armstrong invented the Superheterodyne receiver, which was used to broadcast the first radio broadcast in 1920 in Pittsburgh. In 1921, the ground wireless mobile device was first used by Detroit police. In 1929, Vladimir Zurkin conducted the first televised experiment. In 1933, Edwin Howard Armstrong discovered frequency modulation. The first public mobile phone system was launched in 1946 in five US cities. The system was semi-double-sided and used 120 kHz of wavelength. In 1958, the launch of the SCORE satellite marked the beginning of the age of satellite communications. By about the mid-1960s, FM's bandwidth had dropped to 30 kHz. In the 1950s and 1960s, automatic trunked radio was introduced, which introduced a fully two-way system. In the 1970s, the concept of mobile cellular communication was proposed in Bell labs. First-generation systems were used in the 1980s, and second-generation systems were used in the 1990s [12]. During the twentieth century, various types of wireless systems emerged and later declined. For example, while the transmission of a television signal was initially made by wireless radio transmitters, these transmitters gradually gave way to cable lines. Point-to-point microwave circuits that have backed up the telecommunications network are being replaced by optical fibers.

On the other hand, some of the telephone systems that used to be all wired networks have given way to mobile phones. These changes are usually influenced by the emergence of new technologies.

Wireless telecommunications have been studied as a scientific discipline since the 1960s, but research has intensified since the mid-1990s. This has been the case for a number of reasons. Increasing public demand for wireless communications is the first reason [13]. Until the end of the first decade of the twenty-first century, demand was mainly for mobile phone systems. As a result, in 2002, the number of cellular phones around the world exceeded the number of landlines [14]. In 2005, there were about two billion mobile phone users in the world. In November 2007, the number of mobile phones reached 3.3 billion. But wireless data transfer applications (data) are expected to become more important in the future. The second reason for paying attention to wireless systems is the significant advances in VLSI technology, which has made it possible to implement complex signal processing algorithms in low-power, low-power signals. Finally, the third reason for the success of second-generation digital wireless telecommunication standards, and in particular the multiple access division code (CDMA) standard [15]. However, since the second generation of mobile network systems were designed primarily for voice transmission and their main features such as transmission rate and acceptable time delay were adjusted for audio applications, the third-generation mobile network has emerged and developed [16]. Researchers are currently researching and developing a fourth-generation mobile network that provides a transfer rate of up to 10 to 100 megabits per second (today's third-generation networks allow transmission at speeds of up to 2 Mb/s in some locations all over the world). The fourth generation of mobile phones is expected to have many applications in e-commerce and mobile commerce [17].

1.2.2. Structure

In the discussion of wireless network structure, radio waves or RFs cover frequencies from 30Hz to 300GHz, and it can be pointed out that almost all wireless communication methods fall within this range. Based on this, radio waves are usually talked about in wireless communications, and the rest of the electromagnetic waves are ignored. Contrary to what you may have thought at first, radio waves are not intended to be used on "radio devices", but other communications such as television, wireless internet, mobile phones, etc. also use these waves. The energy of the radio wave increases by increase of its frequency but with shorter wavelength and power. For this reason, the use of high frequencies is usually more suitable for near applications and low frequencies are more suitable for long distances (although not always). Wireless waves based on their frequency ranges are listed in Table 1.I.

Table 1.I. Wireless waves

Name of the frequency band	**Acronym**	**ITU band number**	**Frequency & Wavelength in the air**	**Application**
Very low frequency	VLF	4	3 – 30 KHz 10 - 100 Km	Submarine communication, avalanche beacons, wireless heart rate monitors, Geophysics
Low frequency	LF	5	30 -300 KHz 1 – 10 Km	Navigation, time signals, AM longwave broadcasting, Recognition system with radio waves
Medium freqUENCY	MF	6	300 – 3000 KHz 1 m – 100 Km	AM (medium-wave) broadcasts, amateur radio, Recognition system with radio waves
high frequency	HF	7	3 – 30 MHz 10 – 100 m	broadcasts, citizens' band radio, amateur radio and over-the-horizon aviation communications,
very high frequency	VHF	8	30 – 300 MHz 1 – 10 m	FM, broadcasts and line-of-sight ground-to-aircraft and aircraft-to-aircraft communications. Land Mobile and Maritime Mobile communications, amateur radio, TV
ultra high frequency	UHF	9	300 - 3000 MHz 100 mm – 1 m	broadcasts, ovens, mobile phones, wireless LAN, GPS and two-way radios such as Land Mobile, FRS and GMRS radios, amateur radio
Super high frequency	SHF	10	3 – 30 GHz 10 – 100 mm	devices, wireless LAN, most modern radars, communicatio ns satellites, amateur radio
Extremely high frequency	EHF	11	30 – 300 GHz 1 – 10 mm	Radio Astronomy, high-frequency microwave radio relay, microwave, amateur radio

With a wide range of waves, as mentioned above, as well as different methods of sending and receiving information, many different protocols have been introduced to create a wireless connection. WiFi, GPS, GSM, RFID, VHF, and. Each uses its own protocol to communicate. The technical discussion of how to create and receive radio waves, as well as how to send and receive information in this way, is quite a specialized discussion. Wireless includes a large category of communication types, but wireless usually refers to a specific part of this category with a frequency of 2.4GHz or WiFi, which operates according to the IEEE 802.11 b / g / n standard. This section introduces and explains some famous examples of wireless modules and describes how to run them with the help of Arduino.

1.2.3. Types of wireless network

Wireless networks have various types. The Table 1.II lists the general classification of wireless networks [13].

Table 1.II. Types of wireless network

	Wireless Personal Network (WPAN)	**Local Wireless Area Network (WLAN)**	**Wireless Metropolitan Area Network (WMAN)**	**Wireless Personal Area Network (WPAN)**
Technology	▪ Bluetooth ▪ ultra-wide band (UWB)	• IEEE 802.11 B • IEEE 802.11 E • IEEE 802.11 G or WiFi	• IEEE 802.16 • IEEE 802.16 E • IEEE 802.16 E or WiMax	• GSM • GPRS • • Multiple access code sharing
Data rate	Medium data rate 1 to 2 Mbps	High data rate 11to 54 Mbps	Very high data rates Up to 268 Mbps	Low to medium data rate 10 Kbps to 2.4 Mbps
Range	Very short range 3m	Range 100 m	Medium range 50km	Very wide range (global)
Connection	Laptop to computer, to accessories Device to system	Computer to computer and internet	Local area network or computer line High speed wired internet	Smartphones and personal digital assistant to extensive networks and the Internet

1.3. UART communication

One of the most important and useful protocols in electronic modules is Universal Asynchronous Receiver- Transmitter (UART). This protocol includes the use of only two wires, no need for a clock signal, and support for only one pair of transmitters/receivers at a time. In UART inputs and outputs are marked with Tx and Rx, which indicate Transmit and Receive meaning sending and receiving information. The important thing about UART is that the Tx of one device is connected to another Rx. The connection between two devices is shown in Fig. 1.2.

Hardware of UART controls computer interfaces to its connected serial devices [14]. Explicitly, UART hardware enables interfacing between a Data Terminal Equipment (DTE) and a Data Communication Equipment (DCE). For example, it allows data to be exchanged between modems and other serial gadgets. In the context of this interfacing, UART as well offers the following basic functions:

- To transfer input, it transforms serial bit flows to bytes of the system.
- It adds a parity bit to transferred date, then verifies the parity of received bytes (if chosen) and in the end dumps the parity bit.
- Adds start and stop bits to transferred data on outbound and removes them from inbound data.

In this connection, the following six 8bit registers are used.

1) Receive Shift Register (RSR)
2) Receive Data Register (RDR)
3) Transmit Data Register (TDR)
4) Transmitter Shift Register (TSR)
5) Serial Communications Control Register (SCCR)
6) serial communication status register (SCSR)

Suppose UART is connected to a microcontroller and the address bus so that reading and writing registers has become possible for CPU. The following registers are written from memory: RDR, TDR, SCCR, and SCSR. Using RDR, SCSR, SCCR, data bus can be driven through hard drive buffers. Also, through data bus, TDR and SCCR can be loaded [18].

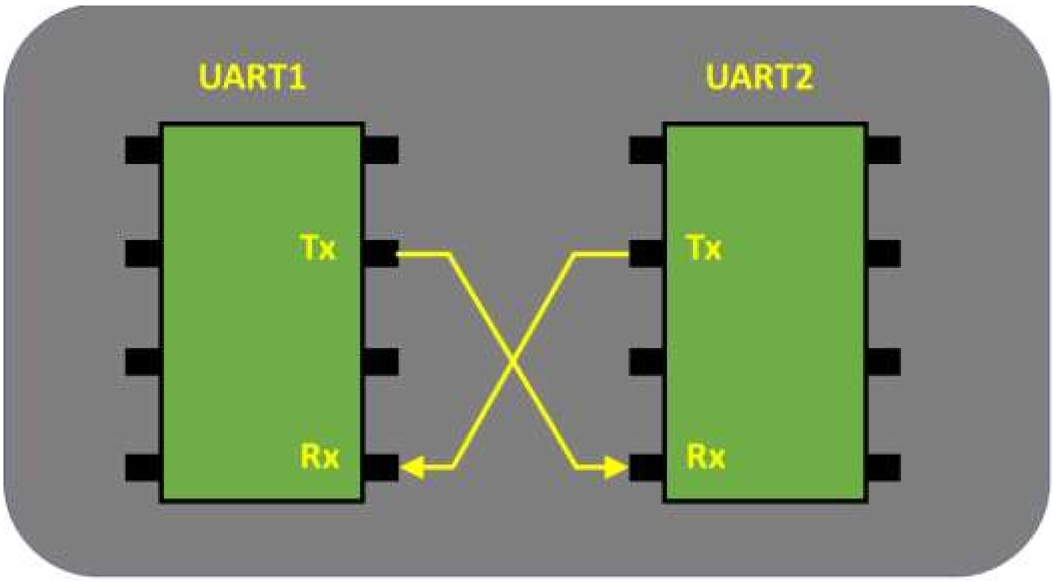


Figure 1.2. How to use UART.

Advantages:

- It uses only two wires.
- No need for clock signal.
- Errors can be checked by an equal value.
- It is possible to change the closed data structure if both sides are arranged for it.
- It is a widely known and documented technique.

Drawbacks:

- There is 9 bits size limit for the data frame.
- Multiple slave or multiple master systems are not supported.
- Baud rates at each UART should be 10% of each other.

1.4. WiFi Modules

1.4.1. ESP8266 module

This module is one of the most important and widely used modules for WiFi communication, this module is known as ESP8266 made by Espressif company (see Fig. 1.3). WiFi is a special wireless protocol that allows you to access the Internet and connect to WiFi routers.

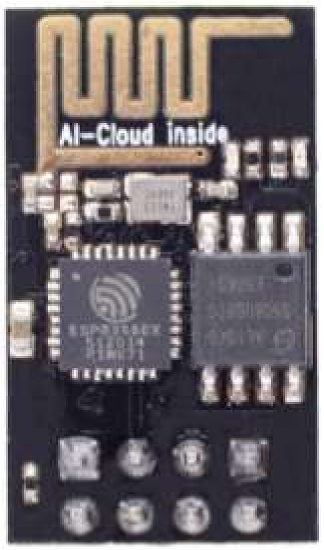


Figure 1.3. ESP8266 Module.

This protocol is great for doing Internet of Things (IoT) projects, whether you want to share information on the Internet, or you want to have a local network of sensors and modules. There is a wide variety of WiFi modules, most of which use the ESP8266 module inside themselves [19]. Another version is ESP8285 that is an ESP8266 with 1 MiB internal flash and enables single-chip devices to connect to WiFi [20].

1.4.2. NodeMCU board

Unlike the previous module, this board is a complete development board. In fact, it is like combining a WiFi module with an Arduino. NodeMCU is a powerful board that has the ESP8266 chip on it. In addition, it has 12 digital inputs / outputs, 1 analog inputs, 128KB of temporary memory, and 4MB of storage memory. NodeMCU also supports Serial Peripheral Interface (SPI), UART and Inter-Integrated Circuit (I^2C) protocols and connects to a computer using a microUSB interface. The board is shown in Fig. 1.4.

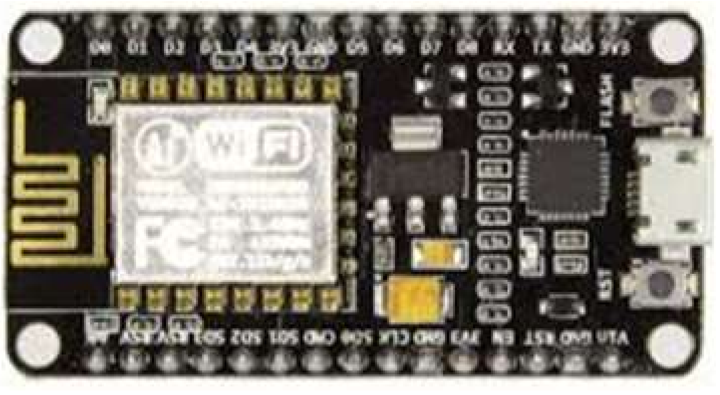


Figure 1.4. NodeMCU board.

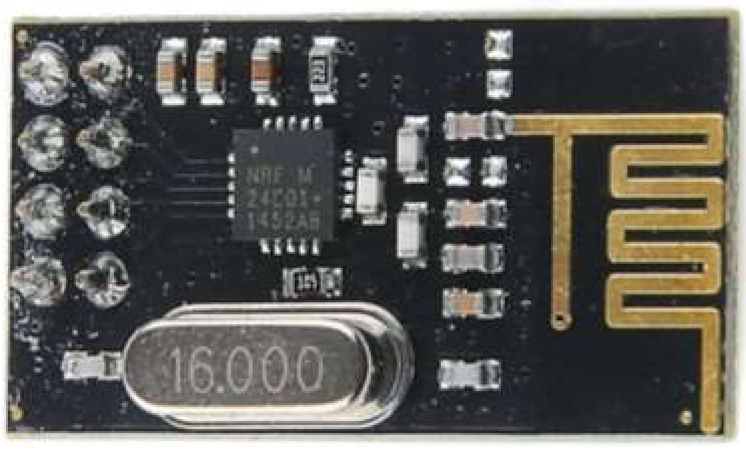


Figure 1.5. nRF module.

1.4.3. nRF module

Another practical and well-built wireless module is the nRF series of nordic semiconductor products. These modules are known for their high range and low power consumption. The nRF24L01 module is a simple and accessible wireless transmitter and receiver (see Fig. 1.5). This module allows to communicate up to 100 meters and with different bandwidths from 256Kbps to 2Mbps. Of course, this distance is for outdoor space, and naturally this amount decreases in indoor spaces.

1.5. Arduino

The term Arduino refers to a team that develops hardware, software, and design philosophy. The company was initially stablished in city of Ivrea in Italy. Arduino got the name from the Italian king, "Arduin of Ivrea". In Italian language, Arduino is a masculine noun that means "strong friend”.

Arduino has an open-source hardware and software framework. The framework includes a single-core microcontroller that forms the hardware part of the Arduino. In addition, the Arduino framework includes an Arduino IDE software designed for programming Arduino boards, and includes a software bootloader that is loaded on the microcontroller [21]. Arduino hardware is designed to generate interactive hardware projects and build devices that interact with the environment quickly and easily [22]. Of course, Arduino boards also pursue educational goals.

The board contains Atmel AVR ATmega8 microprocontroller and subsequent parts including serial ports, expansion slots, power-supply circuits, and different supporting elements [23].

As its inputs, Arduino accepts variables like surroundings light, keys or even an email. Following process of inputs, Arduino is able to produce outputs including turning ON an electrical device, adjusting color of LEDs or sending an email or so on [24].

1.5.1. History and background

The idea of the Arduino was created in 2003 at Interaction Design Institute Ivrea in Italy. The idea was to build simple, low-cost tools for students' digital projects, especially those unfamiliar with engineering and programming. Three key people were involved in making this idea a reality:

Hernando Barragán, Massimo Banzi, and Casey Reas.

Baragan was a student at the Ivrea Institute who decided to pursue his master’s degree thesis. Banzi, and Reas were his thesis supervisors. Until then, there was no name for Arduino. The result of Baragan's dissertation was very successful and led to the creation of hardware and software called Wiring. Wiring hardware had better features than other models on the market at the time, which was simple and inexpensive. Wiring software was also developed based on one of the existing programming languages called Processing.

Following the conclusion of the dissertation, Banzi sought to reduce the cost of hardware of Wiring, and in 2005, in collaboration with David Cuartielles and David Mellis (who were employees and students of the Ivrea Institute, respectively), developed the Wiring project and renamed it Arduino. The new name was derived from the name of the cafe called Arduino in Ivrea, where most of the group's meetings were held. The word Arduino is the name of one of the ancient princes of Italy, who was once the ruler of the city of Ivrea and came to the Kingdom of Italy in the 11th century [25].

1.5.2. Ardiono Uno

Arduino UNO board is used in this thesis which contains ATmega328 microprocessor and is shown in Fig. 1.6. The board contains six analog inputs, fourteen digital I/O pins, USB connection, 16 MHz ceramics oscillator, ICSP header, power jack, and reset button. As the software for programming, an Integrated Development Environment (IDE) is used that includes main libraries. Libraries are scripted by C and C ++ while Java was used for IDE.

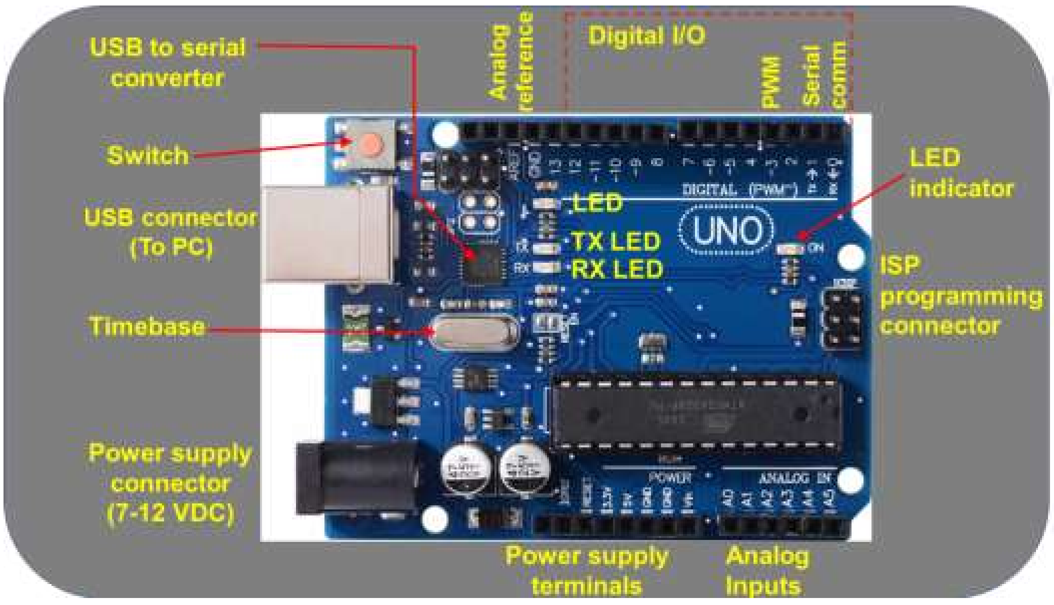


Figure 1.6. Arduino UNO board layout.

1.5.3. Hardware

Arduino has introduced various electronic boards to its users as hardware for the Arduino hardware. One of the most common is the Arduino board, the Uno version. The hardware of the Arduino Uno consists of the following four general sections (see Fig. 1.7):

1. Microcontroller
2. Power supply
3. Computer interface unit
4. Group ports

Atmel ATmega328 processor

Atmel ATmega328 is a microcontroller that acts as the Arduino Uno brain, and includes a Central Processing Unit (CPU), memory arrays, clock, and accessories. It is actually a computer on a chip. Figure 1.8 shows a simple chart of ATmega328.

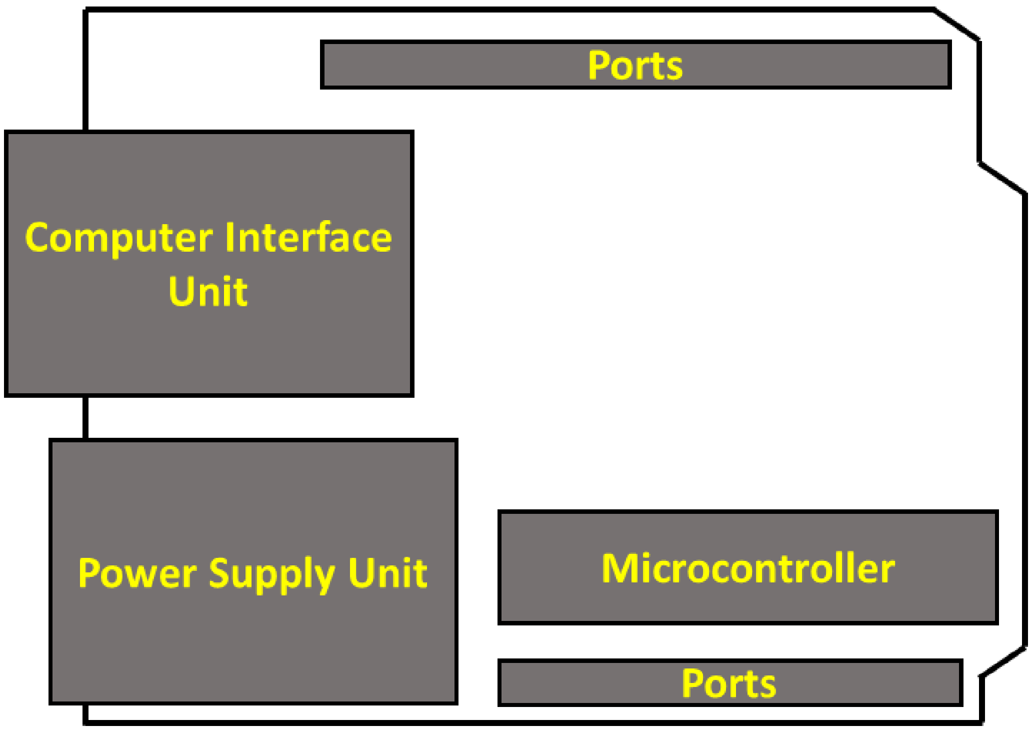


Figure 1.7. The main parts of the Arduino Uno board.

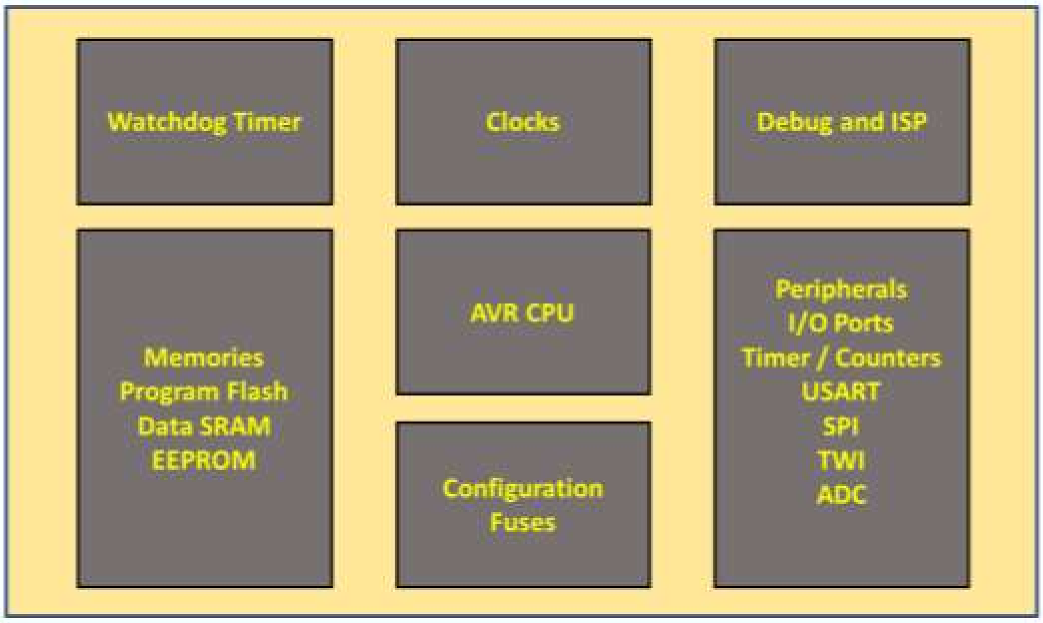


Figure 1.8. A basic diagram of ATmega328.

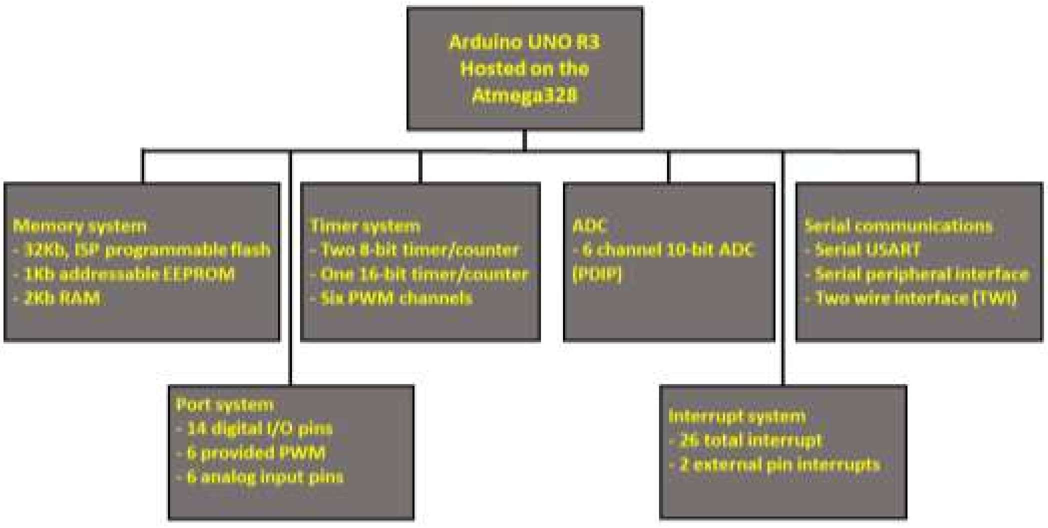


Figure 1.9. A basic diagram of the systems available in Arduino.

The ATmega328 can operate at 1.8 to 5.5 volts fitting it for various applications. Operating speed of ATmega328 depends on its supply voltage. Higher operating speeds can be realized by higher voltages. A minimum of 4.4 volts is required to run a maximum speed of 20 MHz. Because the Arduino board supplies ATmega328 with 5 V, the speed of the processor is not limited and can reach its maximum speed of 20 MHz. Various features of ATmega328 are as follows:

- Memory system

- Port system
- Timer system
- Analog to Digital Converter (ADC)
- Serial communications

A basic diagram of the Arduino board is shown in Fig. 1.9.

Serial port

The communication of Arduino with computer is provided by means of the serial port. Protocols of serial communications have different types. The serial port operates in a non-synchronous manner, meaning that it does not need a standalone clock-signal. In the asynchronous technique, there are two separate data transmission signal and data receiving signal.

Power supply

Arduino boards can be supplied in different ways. The most straightforward way is to supply Arduino using USB cable connected to a PC. Using USB, it is possible to provide a maximum of 100 mA for an infinite USB device and 500 mA

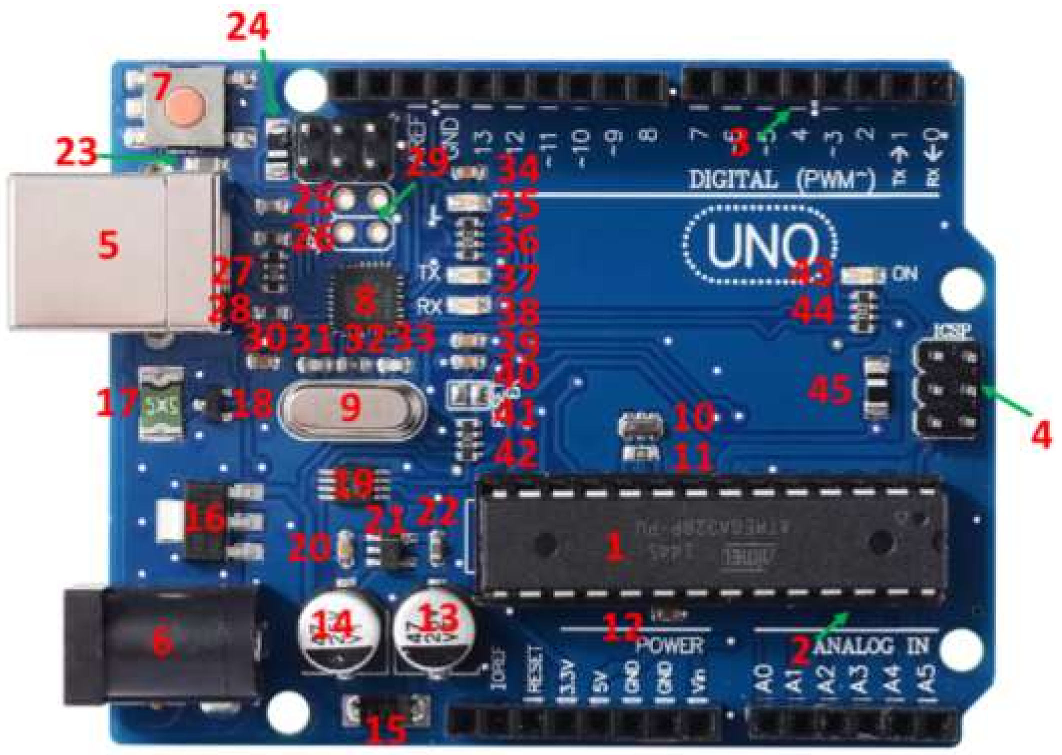


Figure 1.10. Arduino UNO V3 elements.

for normal USB, that is sufficient to power multiple LEDs and sensors. However, for higher power applications, a more powerful power supply is required.

Connections / expansion slots

Arduino Uno provides four goals for expansion connections:

- Add extra circuit
- Power supply connector
- 2 sets of I/O digital connections
- Analog connection

The power connection provides the connection to the mains voltage (3V3, 5V, Vin, and GND) along with the PIN-RESET. Also, pins A0 to A6 are analog pins, which can be adapted and employed as digital I/O pins. Each of I/O digital connectors of ATmega328, Ports B, C, and D, has an alternating side function that can be adjusted when programming an Arduino.

Different components of Arduino UNO V3 are shown in Fig. 1.10 and listed in Table 1.III.

Table 1.III. Different parts of Arduino UNO V3.

Part number	Part name	Part characteristic	Explanation
1	Microcontroller	ATmega328	Processing and memory
2	socket	28 pins	Ease of microcontroller replacement
3	Communication ports	Female header pins	Communication with microcontroller pins
4	Communication ports	male header pin	Serial connection (without USB) with microcontroller
5	USB port	USB port type B	Computer USB connection with microcontroller
6	Power port	5.5 mm	Connect the battery plug or adapter to power the board

7	Pressure switch	Single Switch	Reset the board and restart the program
8	Microprocessor	ATmega16U2	Serial connection to USB converter
9	Crystal oscillator	16 MHz	Generateing time signal for USB processor
10	Ceramic oscillator	16 MHz	Generating a time signal for the Arduino microcontroller (A 10 PF capacitor is also embedded in the part)
11	Resistor	1 MΩ	Maintaining the oscillation balance in the ceramic oscillator
12	Capacitor	1 µF	Removing noise from the microcontroller power supply
13	Capacitor	47 µF	Output power supply current filter
14	Capacitor	47 µF	Input power supply current filter
15	Diode	MRA4007T3G	Protection of circuit against reverse voltage of the power supply
16	Voltage regulator	LD1117S50TR	Regulating the input voltage to the Arduino
17	Fuse	MF-MSMF050-2	USB port fuse, 500 mA
18	Transistor	FDN340P	Switch between power supply and USB port (for powering the board)
19	IC	LMV358	Number 18 transistor controller
20	Capacitor	1 µF	Filter for the IC number 19
21	Voltage regulator	LP2985	3.3 V voltage regulator
22	Capacitor	2.2 0.1 µF	Output voltage filter for LP2985
23	Ferrite bead	BLM21	High frequency noise canceling filter
24	Diode	1N4148W-7-F	Control voltage fluctuations during reset
25	Capacitor	1 µF	USB processor Ucap and GND interfacing

26	Varistor	CG0603MLC	Preventing the entry of static electricity
27	Array resistor	22 Ω	Protecting USB ports
28	Varistor	CG0603MLC	Preventing the entry of static electricity
29	Direct port access	-	Providing access to ATmega16U2 IC ports (from PB4 to PB7)
30	Capacitor	0.1 µF	Removing noise from the microprocessor power supply
31	Capacitor	22 PF	Crystal oscillator capacitor
32	Resistor	1 MΩ	Maintaining the oscillation balance in the crystal oscillator
33	Capacitor	22 PF	Crystal oscillator capacitor
34	Capacitor	0.1 µF	Removing noise from base input voltage
35	LED	Yellow	Programmable flashing LED
36	Array resistor	1 KΩ	Protecting LEDs 35, 37, and 38
37	LED	Yello	Serial Data Exchange Indicator (Output)
38	LED	Yellow	Serial Data Exchange Indicator (Input)
39	Capacitor	0.1 µF	Noise removal for the microprocessor
40	Capacitor	0.1 µF	Sending reset pulse from ATmega16U2 to ATmega328
41	Connection	Solder	If disconnected, the Arduino will not reset automatically
42	Array resistor	10 KΩ	Microcontroller reset resistor
43	LED	Green	Arduino on/off indicator
44	Array resistor	1 KΩ	Exchange of serial data
45	Diode	cd1206	Controlling voltage fluctuations during reset

1.5.4. Modules and shields

Using Arduino and with the help of various modules and shields made for Arduino, various projects such as the Internet of Things, smart homes and interactive products can be implemented. Arduino boards can be programmed without the need for a programmer via a USB cable and open-source integrated software development platform.

Besides open-source hardware, Arduino possesses an integrated free and open-source software development environment. The Arduino software can program the microcontroller directly without using a programmer. The software of Arduino possesses beneficial properties like serial terminal allowing to troubleshoot written programs for the microcontroller by the serial and USB connections. Moreover, there is the possibility to send ADC values and pin status or further data to the serial terminal of Arduino and collect data from the microcontroller via PC.

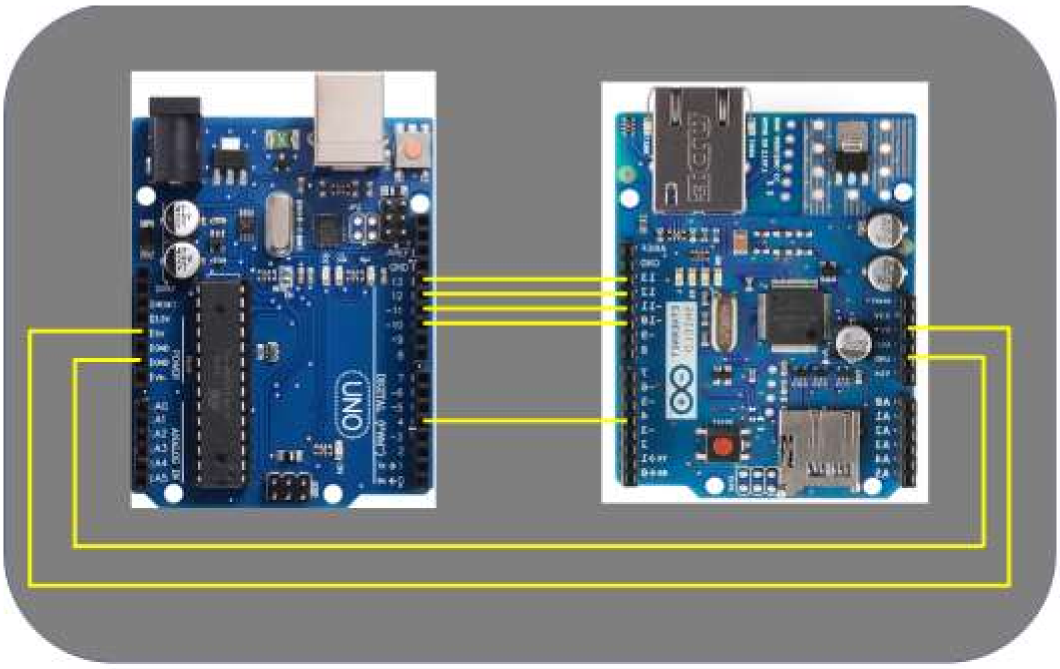


Figure 1.11. An illustration of the connection between the Ethernet Shield and the Arduino board whereas both are assembled [26].

Using the concept of the daughter card (be referred to "shields"), external hardware and supplementary features can be joined to Arduino. Expansion connections are used to add shields to the Arduino board. This addition allows I/O operating same as a small motherboard. This provides electrical and mechanical links for supplementary circuits (see Fig. 1.11). This thesis focuses on Arduino WiFi Shield, that is a fundamental element in the implemented wireless network.

1.5.5. Arduino programming

This section presents a brief overview on how to program the Arduino UNO. Arduino programing scheme is shown in Fig. 1.12. A separated compiler hosted on a PC uses source files of the program (.c and .h files) to build a device code for uploading to the Arduino board (file. Hex). This is done in three steps as follows:

1. Collection - .asm assembly code files (.asm) are generated from source files of the program.
2. Then, the assembler uses the assembly file and converts the file to a .x file which is the device code. This code can be loaded on the Arduino Uno.
3. The program is loaded on Arduino Uno.

Figure 1.12 illustrates the procedure.

Arduino Development Environment (ADE) offers a user-friendly interaction easing the procedure of the programming and development.

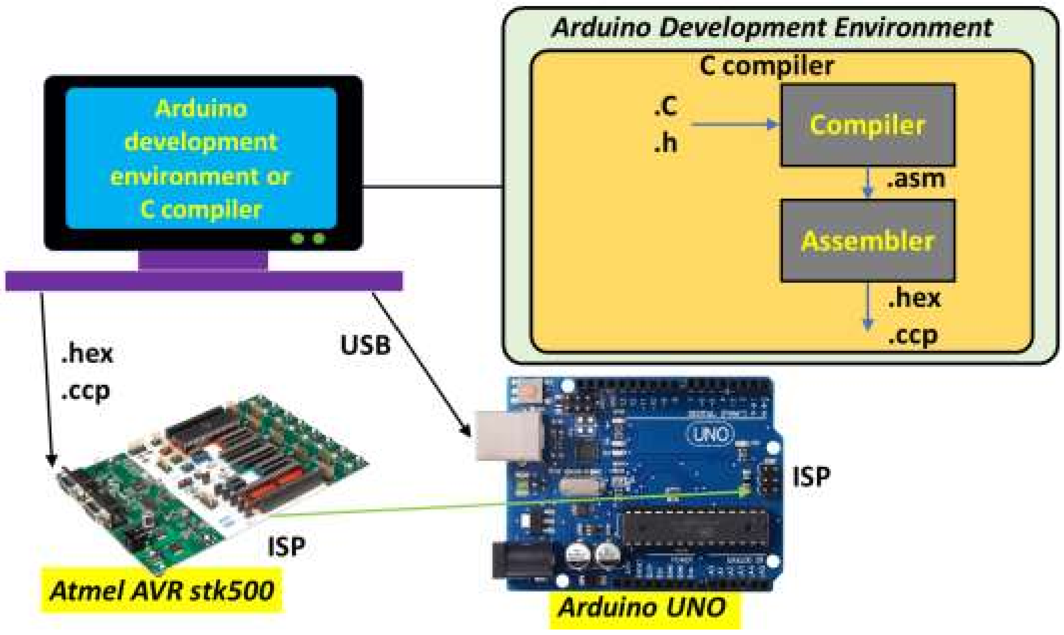


Figure 1.12 Arduino programming.

CHAPTER 2.

2. Literature Review

2.1. Wireless Sensor Networks

Due to recent remarkable advancements in sensor devices, microprocessors, computational methods, and telecommunication technologies, deployment of wireless sensor networks has attracted increasing research and industrial attentions. Wireless sensor networks are standalone networks containing a number of sensor nodes, microcontrollers, and wireless communication providing devices all networked together without cables. Sensor nodes are employed to measure, collect, and transmit specified parameters to the monitoring or controller system via a wireless network. In the case of interactive systems, in addition to sensor nodes, there are actuator nodes for implementing desired functions. Such networks are called Wireless Sensor Actuator Networks [27, 28].

Availability of low-cost hardware required for creating a wireless sensor network accelerates growth and demand of employing these networks [29]. In addition to the affordability of wireless sensor networks, unlike the conventional networks, because of lack of cables for communication between subsystems, topologies of wireless sensor networks are not restricted. This flexibility facilitates development of the wireless sensor network regardless of its scale and creates new areas of application for sensor networks.

Generally, wireless sensor networks are task oriented. In other words, operational principle and structure of the sensor network are determined by the considered application. Wireless sensor networks are utilized in a great variety of applications [30]. In many of application, the main motivation of deployment of wireless sensor networks is to eliminate space limits and other physical or environmental constraints. For example, in [31], a wireless sensor network was proposed for condition monitoring system of aircraft generator bearing. The proposed system adopted as it can eliminate weight and space of communication cables and increase the reliability of the system by removing cable rooting isolation issues and high operating temperature related issues of wirings. Also, in [32], wireless sensor network approaches for health monitoring system of bearings for train applications were reviewed.

In [33], wireless sensor network is employed in underwater pipeline monitoring system based on Autonomous-Underwater-Vehicles (AUVs). A sensor network with wireless communication for damage detection of infrastructures is proposed in [34].

2.2. Smart homes/offices

Home automation or smart home concept is an increasing trend in building design towards maturity of sustainable cities and communities. Along with ideas such as smart city and smart factory, the goal of the home automation is to increase integration among individual systems to optimize their performance on the basis of improving interaction between them and human. Based on motivations behind development of home automation systems, several research were conducted and reported in the literature. Most prominent motivations for home automation development are comfort, security, better management of energy consumption, and networking. Moreover, based on different parts of a home automation system, the state of art studies can be categories into different clusters indicating focus of the research.

Smart homes/offices are other applications of wireless sensor networks. Exploiting flexibility and cost efficiency of wireless sensor networks, home/office automation systems can be easily implemented. Using wireless sensor network, in [35], a home automation system is implemented for sensing environmental data and controlling household electronics devices. In the implemented wireless sensor network, sensor nodes are supplied by harvested energy from the indoor ambient light leading to an energy-efficient home automation system.

Regarding the microcontroller used in wireless sensor networks, Arduino is one of the most advantageous hardware platforms. OpenSource hardware/software of Arduino provides a high degree of flexibility for development of wireless sensor networks.

Due to advantages offered by Arduino, in this thesis, a wireless sensor network is implemented by using Arduino platform. As the network application, home automation has been chosen. To maintain the cost efficiency advantage of wireless sensor networks, in this thesis, it is tried to employ low-cost components. Also, since Matlab is one of the most universal and well-documented software environments, it is selected as the software platform for programming and interfacing Arduino hardware, sensor nodes and the user.

Sensors networking and interaction media between controller and human. As mentioned above, one motivation behind development of home automation systems is networking. It means that several home appliances are integrated to each other and create a home network that allows a centralized control possible. This leads to optimum control of home appliances for better responding to users' needs.

The most prominent trend for the networking in home automation systems is employment of wireless sensors [36, 37]. In this way, home appliances are monitored by using sensors that are connected to a central controller by means of wireless connections. For example, smart refrigerator systems were studied in [38, 39]. In these works, Radio Frequency Identification (RFID) sensors are used to monitor contents of the refrigerator. Also, temperature sensors are used for inside the fridge temperature measurement. Sensors are connected to the controller via RFID antenna. As the media for wireless connection between the controller and human, the Global System for Mobile Communication (GSM) is employed. Instead of GSM, the internet was adopted for human-system interaction media in [40]. Since various combinations of wireless connections between sensors and controller and between human and controller are possible, in the system proposed in [41], RFID sensors are connected to the controller by means of NodeMCU and WiFi while human-system interaction media is the internet.

In [42-44], home automation systems were proposed based on using Bluetooth for communication between sensors and the controller. Radio Frequency (RF) transmitter ICs were integrated into sensor nodes to transmit measured data by means of radio frequency [45]. As one of the most common modules for wireless sensor node applications, NodeMCU wireless module with an integrated WiFi module has broad application. Few examples of home automation system based on NodeMCU wireless module are studied in [46-51].

2.3. Control hardware for home automation systems

After establishing remote monitoring/actuator devices connected to each home appliances, measured data or/and control commands must be processed/generated in the control unit. As an inexpensive open-source programmable hardware, Arduino platform has a wide usage range for many applications where remote monitoring and control are needed [52]. For example, a control system to avoid children abandonment in unattended cars was developed by adopting Arduino platform [53]. Applications of Arduino in the mining industry were reviewed in [54]. The development of a brainwave acquisition device was proposed in [55]. The authors in [56] proposed a Dynamic Structural Identification system based on Arduino hardware for data acquisition. Several applications of Arduino in robotic were reported in [57]. As an identification of the broad range applications of Arduino, an alternator synchronizer system for electric power system applications was proposed based on Arduino platform [58].

2.3.1. Arduino programming using Matlab

As a way to program Arduino hardware, Matlab provided a support package for Arduino. Using Matlab, it is possible to communicate with Arduino through the Arduino support package. Matlab can be used for programming purpose and also to create user interface to interact with human by employing Matlab Graphical User Interface (GUI) features [59].

In [60], Matlab was used to read measured data from temperature sensors. An AC electric motor control was designed by using Matlab/Simulink [61]. The controller was designed in Matlab/Simulink environment and then Arduino codes were built buy Matlab for uploading on Arduino hardware. In [62], Matlab/Simulink environment was used for simulation of Maximum Power Point Tracking (MPPT) algorithm of photovoltaic panels. Then using Arduino support package, Arduino codes were built and uploaded on Arduino hardware for experimental implementation of the algorithm. As an example in the field of power electronics, a control system of inverter used in hybrid AC/DC home applications was developed in Matlab [63]. Generated codes by Matlab were used to program an Arduino board.

2.3.2. Home automation by using Arduino

As mentioned before, comfort, security, better management of energy consumption, and networking are main motivations for home automation systems. From the motivations it can be concluded that some of them indicate a necessary or quasi-necessary need for home automation systems. For example, the need of home automation systems designed for the elderly or people with disabilities has become an essential need, not just a system to increase comfort [64-66]. For similar reasons, the need for home automation systems to increase energy efficiency is a necessity, not a luxury. Therefore, to make these systems more affordable to a wider range of people, reducing their cost is of particular importance. Considering cost, features, and flexibility of Arduino platform as an open-source inexpensive hardware, Arduino can play a significant role in development of home automation systems [67, 68]. An Arduino-based environmental monitoring and home control system was proposed in [69]. In this work, low-cost, and flexibility of Arduino hardware were reported as the main reasons for employing Arduino hardware. A home automation system for people with disabilities was proposed in [70]. The proposed home automation system employs Arduino board as the control unit and a mobile phone application working with speech recognition voice control to make it convenient for users with disabilities. In [71-76], number of Arduino-based home automation systems with voice control feature were proposed. An Arduino-based auto following robot car for increasing safety of the elderly was proposed in [77]. Using Arduino UNO board, a brain-controlled home automation system was developed in [78]. The proposed system uses blinking and attention levels of users to generate control commands.

An Arduino-based home automation system aiming at presenting an affordable system to enhance comfort, safety, and energy management was proposed in [79]. Other examples of building management systems by employing Arduino hardware to achieve cost-effective home automation system with energy efficiency optimization were studied in [80-82].

CHAPTER 3.

3. Arduino Interface with Matlab

3.1. Introduction

Matlab has added a feature to provide direct access to the sensor data for processing by the software. In this way, we can make the necessary changes directly to it. For example, suppose the effect of changing the speed of a DC motor on a module or a sensor is required. Using Matlab, it is possible to make changes to the processing algorithm quickly and to rapidly change the motor speed.

3.2. Installation of the Matlab support package for Arduino

The Arduino package is not installed by default in Matlab environment. The package should be first downloaded and installed. To do this, first log-in to Matlab. Then open Add-ons tab in the Home menu. Click Get hardware support packages as shown in Fig. 3.1.

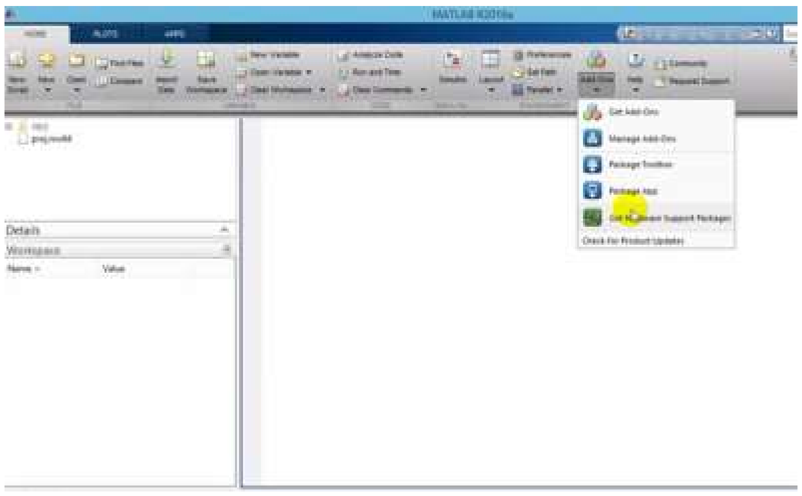
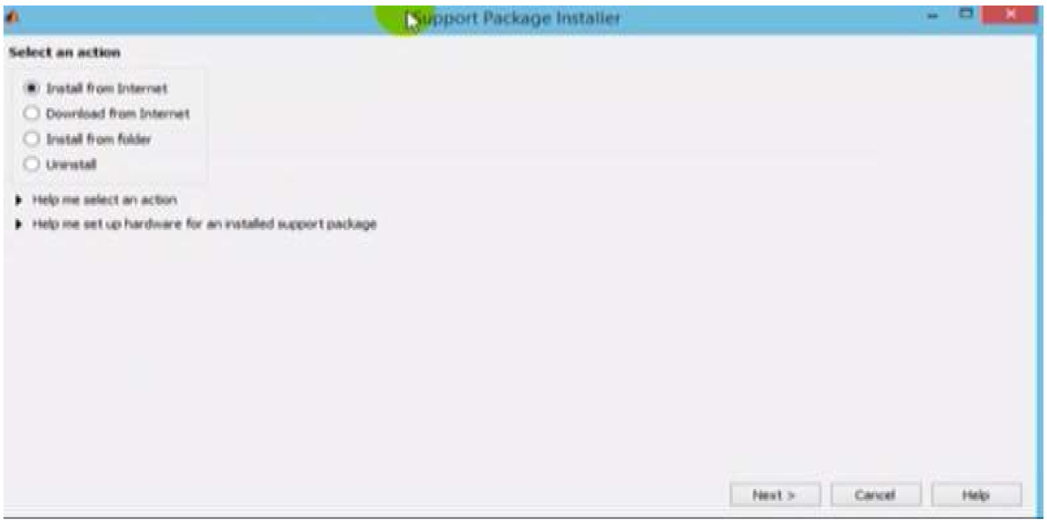


Figure 3. 1. Matlab Add-Ons menu.


Support Package Installer

Select an action

- [✓ Install from Internet]
- Download from Internet
- Install from folder
- Uninstall

▸ Help me select an action

▸ Help me set up hardware for an installed support package

Figure 3. 2. Support package installer options.

In the next step, the following window will open (see Fig. 3.2). Depending on the version of Matlab, this window may be different.

In Fig. 3.2, the "install from internet" option directly downloads and installs the files from the Mathworks website. However, if the "download from the Internet" is chosen, it will only download the files and the user must do the installing.

Click Next to go to the next page (see Fig. 3.3). Then, in the search field, writing the phrase "Arduino", Arduino packages are displayed (see Fig. 3.4).

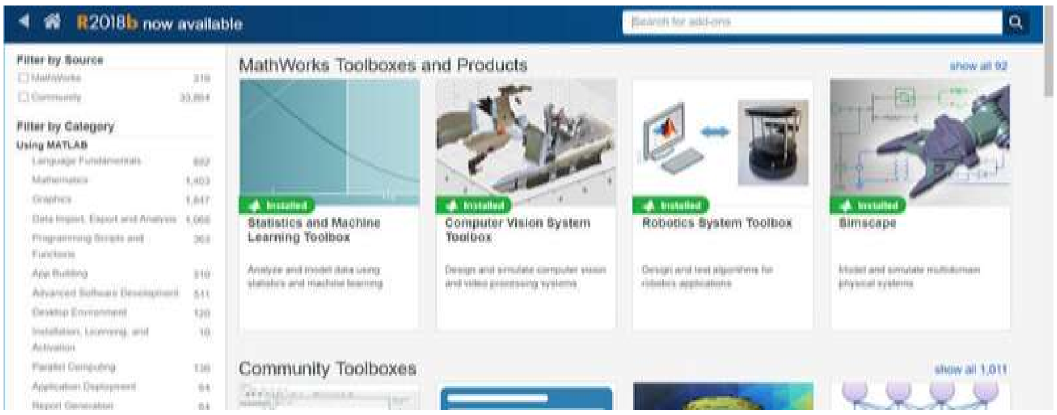
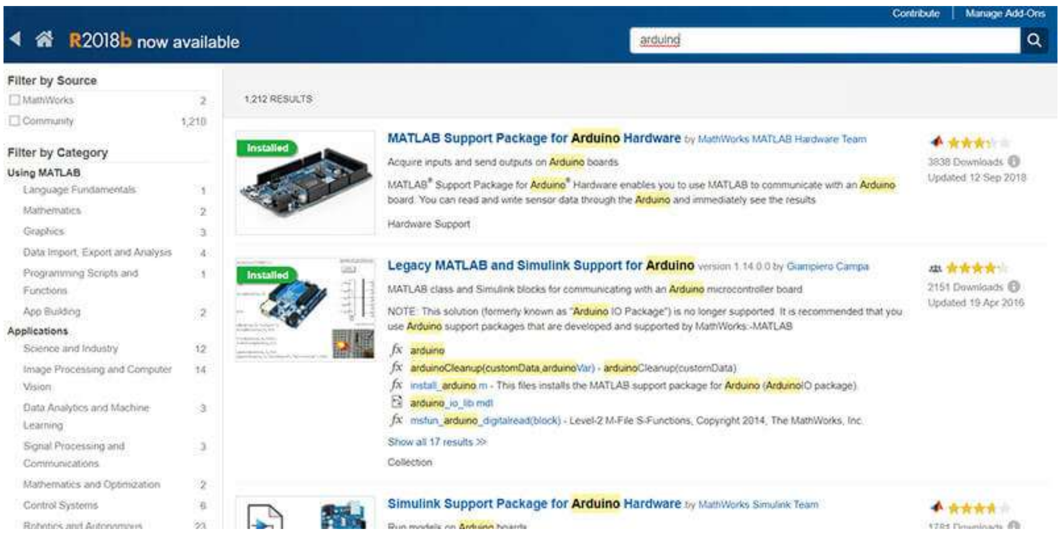


Figure 3. 3. Mathworks toolboxes and products.


R2018b now available
Contribute | Manage Add-Ons
arduind
Filter by Source
MathWorks 2
Community 1,210
Filter by Category
Using MATLAB
Language Fundamentals 1
Mathematics 2
Graphics 3
Data Import, Export and Analysis 4
Programming Scripts and Functions 1
App Building 2
Applications
Science and Industry 12
Image Processing and Computer Vision 14
Data Analytics and Machine Learning 3
Signal Processing and Communications 3
Mathematics and Optimization 2
Control Systems 6
Installed
Installed

1,212 RESULTS

MATLAB Support Package for Arduino Hardware by MathWorks MATLAB Hardware Team

Acquire inputs and send outputs on Arduino boards

MATLAB® Support Package for Arduino® Hardware enables you to use MATLAB to communicate with an Arduino board. You can read and write sensor data through the Arduino and immediately see the results

Hardware Support

3838 Downloads
Updated 12 Sep 2018

Legacy MATLAB and Simulink Support for Arduino version 1.14.0.0 by Giampiero Campa

MATLAB class and Simulink blocks for communicating with an Arduino microcontroller board

NOTE: This solution (formerly known as "Arduino IO Package") is no longer supported. It is recommended that you use Arduino support packages that are developed and supported by MathWorks:-MATLAB

- arduino
- arduinoCleanup(customData,arduinoVar) - arduinoCleanup(customData)
- install_arduino.m - This files installs the MATLAB support package for Arduino (ArduinoIO package)
- arduino_io_lib.mdl
- msfun_arduino_digitalread(block) - Level-2 M-File S-Functions, Copyright 2014, The MathWorks, Inc.

Show all 17 results >>

Collection

2151 Downloads
Updated 19 Apr 2016

Simulink Support Package for Arduino Hardware by MathWorks Simulink Team

Figure 3. 4. Arduino related packages in Matlab.

Selecting "Matlab Support Package for Arduino Hardware" download window opens. For the download to be started, login to Matlab user account is required. Then, click on Install. After downloading the files and installing them automatically by Matlab, it is possible to program Arduino in Matlab environment.

3.3. Description of the Matlab support package for Arduino

Matlab support package for Arduino hardware enables you to use Matlab to communicate with Arduino board via USB cable. It receives the commands that enter through the serial port, executes the commands, and returns the result if necessary. Main features of the package are as follows:

- It provides the possibility to start programming immediately and without any other toolbox.
- Interactive development and debugging in Matlab environment.
- Creating interactive programs to access analog and digital data, and control electric motors such as DC, servo, and escalator electric motors. Access devices and peripherals connected to the Inter-Integrated Circuit (I^2C) or the Serial Peripheral Interface (SPI).
- Run the control loops up to 25 Hz (not real time).
- This package supports Arduino UNO hardware, Arduino Mega 2560 and Arduino DUE.

- Using the Matlab support package, it is possible to use Matlab commands to control and recover data from Arduino hardware and its accessories.

3.3.1. Package update

To update the Arduino hardware support package, the following steps should be taken: On the Matlab home screen tab, in the Environment section, follow below steps:

click Add-Ons > *Check for Updates* > *Hardware Support Packages.*

3.3.2. Arduino hardware configuration

After installing the support package, as described in the support package installation, it is possible to configure the connection between the host computer and the Arduino board. It can be done by typing “arduinosetup” in the Matlab command window and select one of the following connection types:

1. Connection via USB
2. Connection via Bluetooth
3. Connection via WiFi

3.3.2.1. Connection to Arduino board via USB

As shown in Fig. 3.5, the Arduino hardware communicates with the host computer via a USB cable.

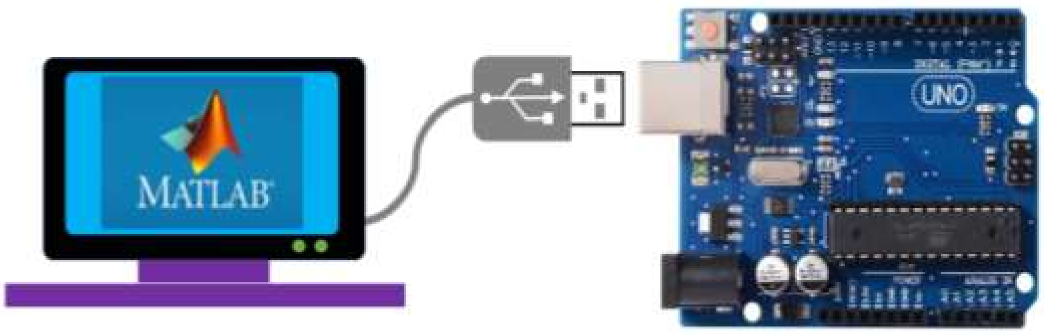


Figure 3. 5. Connection to Arduino board via USB.

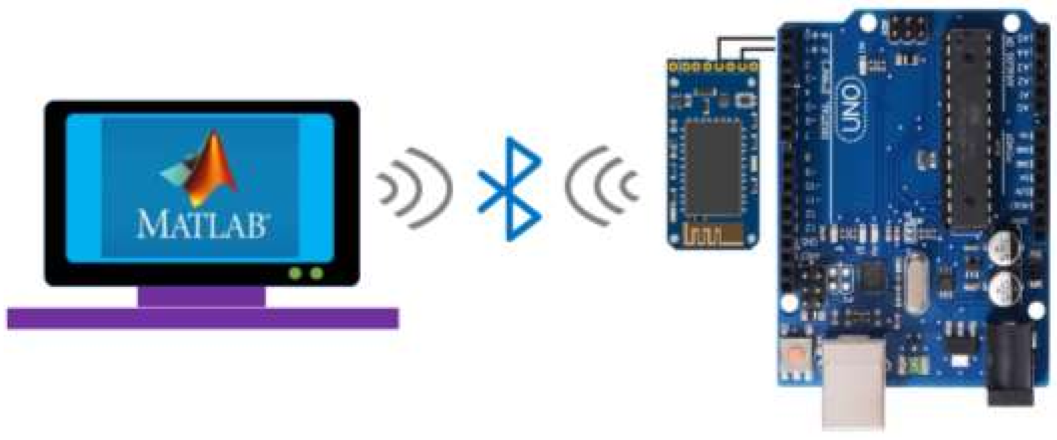


Figure 3. 6. Connection to Arduino board via Bluetooth.

Following steps must be taken to configure Arduino hardware for communicating via USB:

Connect the Arduino hardware via USB and select the type of connection as USB connection.

From "Choose board" and "Choose port", select the board type and port number. Also, select the libraries required to be included in the Arduino server. Click on "Program" to start loading the server on the Arduino board. After completion of loading, click on the "Test Connection" to test the connection between the Arduino board and your host computer. Click Finish to complete the hardware configuration.

3.3.2.2. Connection to Arduino board via Bluetooth

Note that a specific toolbox software (Instrument Control Toolbox) is required to connect to a Bluetooth-HC-05 or HC-06 Bluetooth modules. After setting up, Arduino hardware communicates with the host computer via Bluetooth, as shown in Fig. 3.6.

Following steps must be taken to configure Arduino hardware for communicating via Bluetooth:

1. Connect the Arduino hardware using USB and select the Bluetooth connection as the connection type.

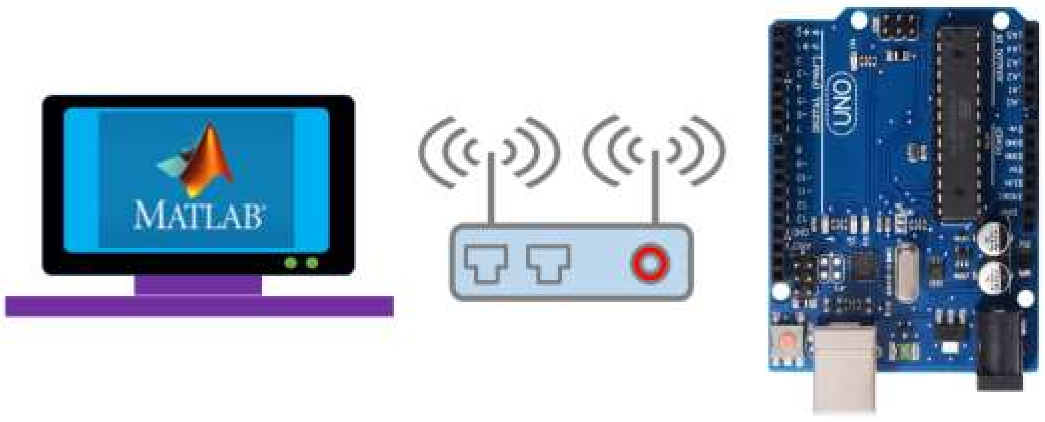


Figure 3. 7. Connection to Arduino board via WiFi.

2. From “Choose board” and “Choose port”, select the board type and port number. Also, select the libraries required to be included in the Arduino server. Click on “Program” to start loading the server on the Arduino.
3. To connect to the Arduino board, select the Bluetooth device. If HC-05 or HC-06 are used and the configuration was not done by using the arduinosetup interface, click on “No” in order to configure the Bluetooth device.
4. Once the Bluetooth device configuration is done, proceed to pair the Bluetooth device by following the on-screen links in the arduinosetup interface. Then get a serial port or address for the Bluetooth device.
5. After completion of the Bluetooth device pairing, click “Test connection” to test the connection between the Arduino board and the host computer.
6. To complete the hardware configuration, click on “Finish”.

3.3.2.3. Connection to Arduino board via WiFi

As shown in Fig. 3.7, after installation, Arduino hardware communicates to the host computer via WiFi.

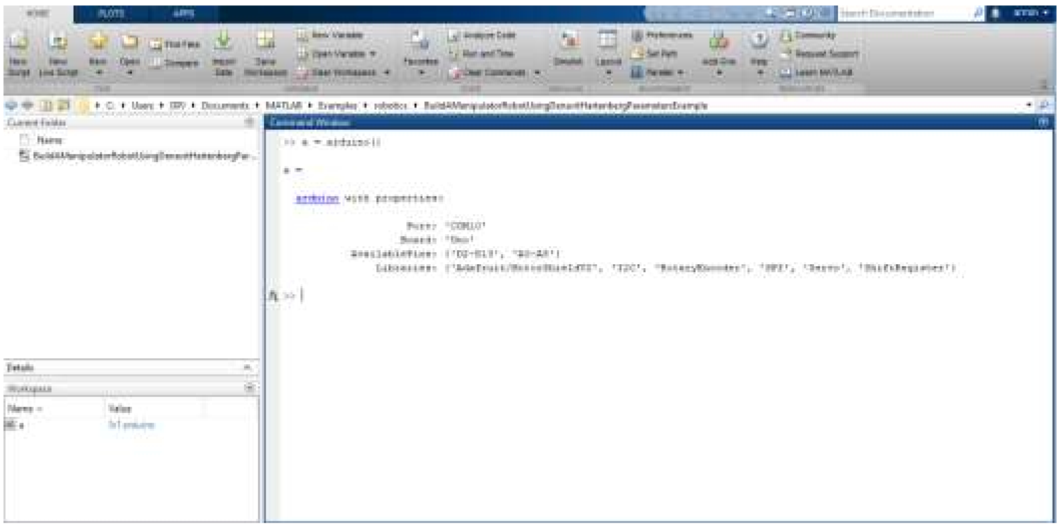
Following steps must be taken to configure Arduino hardware to communicate via WiFi:

1. Connect the Arduino hardware using USB and select the type of WiFi connection. Select your WiFi network encryption type and enter the required credentials.

2. If the Arduino board already was configured using WiFi using the arduinosetup interface, credentials from the previous configuration can be retrieved by selecting the “Retrieve last configuration from Arduino board”.
3. If the box “Use static IP address” is checked, it should be checked to make sure that an available static IP address is used, and the router allows using of the static IP.
4. The Subnet Mask must be 255.255.255.0 and the Default Gateway must be same as the network gateway IP address.
5. Select the board type as MKR1000 and its port number. Also, select the libraries required to be include on the Arduino board. Click “Program” to start loading the server on the Arduino board.
6. After completion of the loading, click the “Test Connection” to test the connection between the Arduino board and the host computer.
7. To complete the hardware configuration, click on “Finish”.

3.4. Connection to Arduino hardware using Matlab

In this section, using an example, it is shown how to connect to Arduino hardware in Matlab. First, it should be ensured that the Arduino hardware is connected to the computer. Then, using the following commands, the Arduino board can be interfaced to Matlab. Figure 3.8 shows a Matlab command to create an object named “a” and to show specifications of the Arduino board.



```
>> a = arduino()

a =

  arduino with properties:

                   Port: 'COM10'
                  Board: 'Uno'
          AvailablePins: {'D2-D13', 'A0-A5'}
              Libraries: {'Adafruit/MotorShieldV2', 'I2C', 'RotaryEncoder', 'SPI', 'Servo', 'ShiftRegister'}

>> 
```

Figure 3. 8. An example of a Matlab command for Arduino interfacing.

3.4.1. Data read and write

Before using the read and write functions, an Arduino object must be created using the “Arduino” function. Here are a few basic commands thar are required to read and write data.

The “**configurePin**” command is used to set the mode for each Arduino pin and can be used in the following two ways:

pinMode = configurePin(a,pin)

configurePin(a,pin,mode)

In the first command, the desired Pin mode is stored in the **pinMode** variable, and in the second command, a special mode is assigned to the Pin. for example:

```
a = arduino('COM4','Uno');
configurePin(a,'A2')
```

```
ans =
'Unset'
```

```
a = arduino('COM4','Uno');
configurePin(a,'A4','I2C');
```

```
pinMode = configurePin(a,'A4')
```

```
pinMode =
'I2C'
```

Pin mode is specified as a character vector. Valid pin modes are as follows:

AnalogInput: Get analog signals from the pin.

DigitalInput: Get digital signals from the pin.

DigitalOutput: Generate digital signals from PIN.

I2C: Specify a pin to use the I2C protocol.

Pullup: Specify the pin to use the pull-up switch.

PWM: Set the pin to use a pulse width moderator.

Servo: Specify the pin to use a servo.

SPI: Specify a pin to use with the SPI protocol.

Unset: It clears the pin mode. The pin is no longer secure and can be adjusted automatically in subsequent operations.

The pins are configured in the first use. To change the mode, pin mode must be reset. For example, if you want to use pullup, you need to set the "pullup" mode.

The next command is **readDigitalPin**, which reads data from a specific pin from the Arduino.

```
a = arduino;
readDigitalPin(a,'D13')
```

The **writeDigitalPin** command is used to write information to one of the Arduino pins. It is used in the following example. In this example, first, the Arduino is identified as an object called “a” in Matlab. Then, with the help of **writeDigitalPin** command, the pin D12 is set to logic 0 and D11 to logic 1. Using the pause command, 1-second delay is imposed into signals and then the logic of each pin is reversed by using the same command.

```
a = arduino()
 writeDigitalPin(a, 'D12', 0);
 writeDigitalPin(a, 'D11', 1);
 pause(1);
 writeDigitalPin(a, 'D11', 0);
 writeDigitalPin(a, 'D12', 1);
```

Another practical command is **“writePWMVoltage(a, pin, voltage)”** command, through which a certain voltage can be applied to a desired pin. **writePWMDutyCycle(a, pin, duty cycle)** sets the PWM duty cycle on a digital pin of Arduino hardware.

playTone(a, pin, ferq, duration) creates sound in Piezo speakers connected to a digital pin on the Arduino hardware with a specified frequency for a defined period of time.

voltage = readVoltage(a, pin) reads the voltage on the analog input pins of the Arduino hardware.

CHAPTER 4.

4. Implementing a Wireless Sensor Network, Hardware and Software Aspects

4.1. Introduction

This chapter deals with hardware and software aspects of this dissertation. Explanations will be given about the parts used in this project and how they work. Also, the libraries used are described. Moreover, descriptions of codes are presented.

4.2. ESP8266 NodeMcu development board with Wifi and CP2102 converter module

The development board (shown in Fig. 4.1), outfitted with ESP-12E module that contains ESP8266 chip, which has a 32-bit Tensilica Xtensa® LX106 RISC microprocessor operating at an adjustable clock frequency of 80 to 160 MHz. It also supports Real-Time Operating System (RTOS).

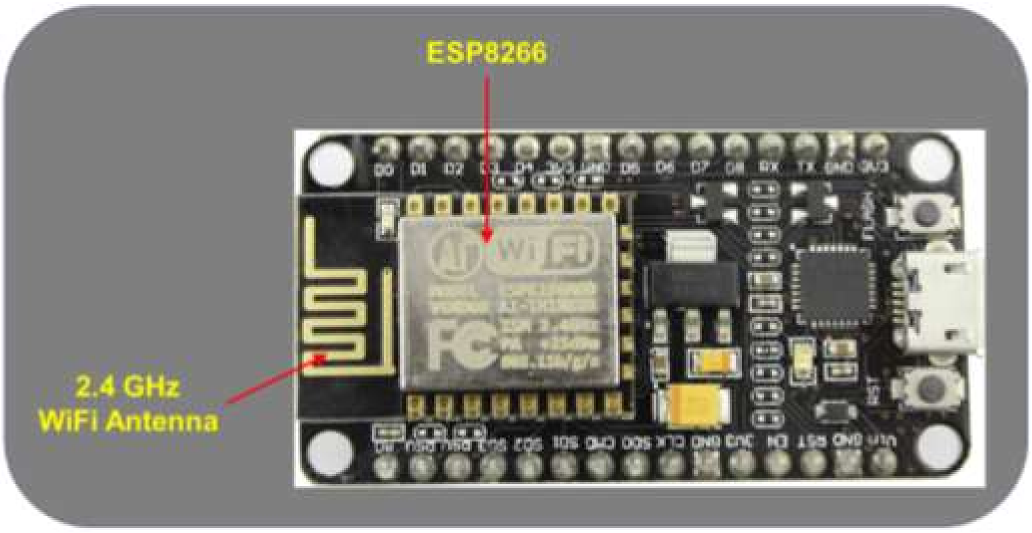


Figure 4. 1. ESP8266 NodeMcu development board.

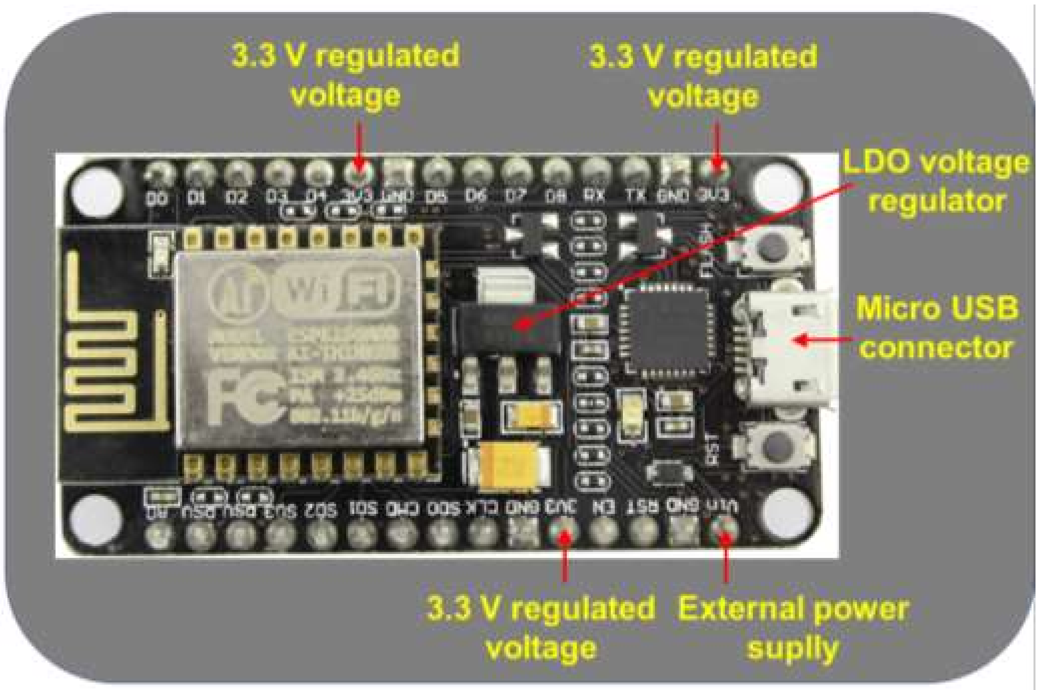


Figure 4. 2. ESP8266 NodeMcu power pins.

128 KB of RAM and 4 MB of flash memory (for storing data and apps) are sufficient to work with big strings of web pages, JSON / XML data and IoT applications.

ESP8266 WiFi receiver is integrated with the 802.11b / g / n HT40 HT40, so not only it can be connected to a WiFi network and communicate with the Internet, but also can create its own network. Therefore, it can allow direct connection from separate devices. So, ESP8266 NodeMCU can be very useful.

4.2.1. Required power

Because the operating voltage range of ESP8266 is from 2.5 V to 3.6 V, this board has a Low-DropOut (LDO) voltage regulator to maintain the supply voltage constant at 3.3 volts (see Fig. 4.2). This will provide a current of 600 mA, which should be more than enough when transferring Radio Frequency (RF). The output of the regulator is also taken to some pins in each side of the board and marked as "3V3". These pins may be utilized to provide power for external components.

The NodeMCU ESP8266 power supply is powered by a USB MicroB USB connector. On the other hand, if an adjustable 5V voltage source is available, it can be used by connecting to the VIN pin for direct supplying of the ESP8266 and connected accessories.

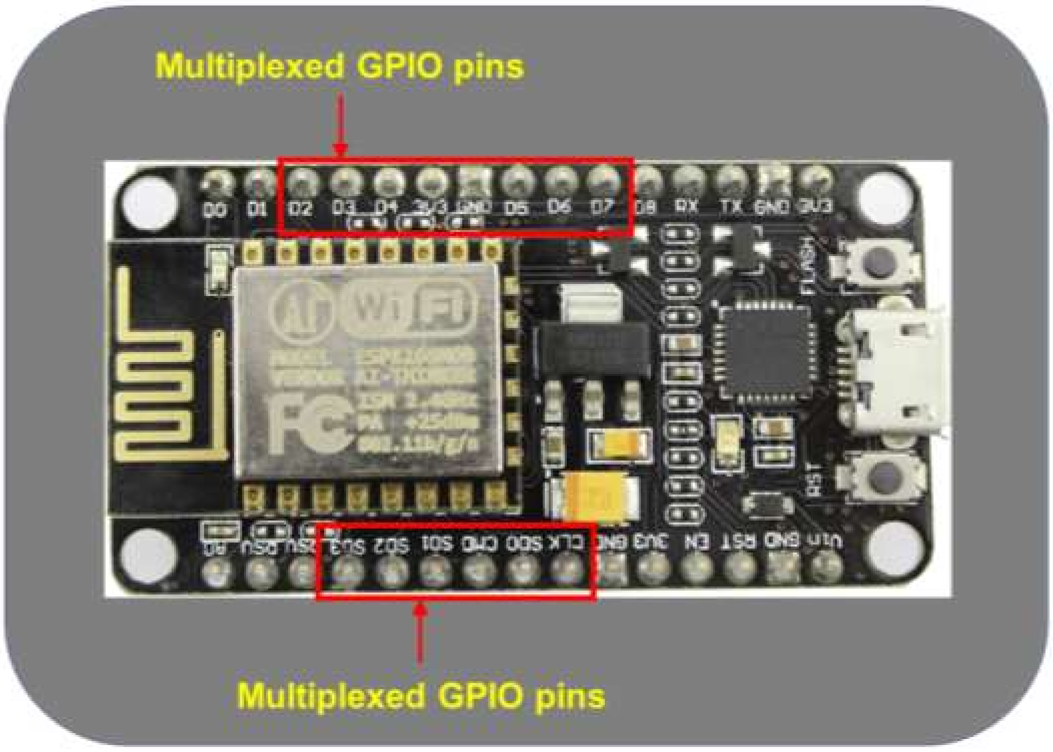


Figure 4. 3. ESP8266 NodeMcu GPIO pins.

4.2.2. GPIO Pins

The ESP8266 NodeMCU possesses 17 GPIO pins located on the header pin on either side of the expansion board as (see Fig. 4.3):

- ADC channel - 10-bit ADC channel.
- UART interface - UART interface is utilized for loading serial codes.
- PWM output - PWM pins may be used for motor control or LED dimming.
- SPI, I^2C & Inter-IC Sound (I2S) interface - SPI and I^2C interface for connecting sensors and accessories.
- I2S interface - Used for adding sound to the project.

4.2.3. On-board switches and LED indicator

Buttons and LED indicator of ESP8266 NodeMCU are shown in Fig. 4.4. ESP8266 NodeMCU has two buttons. The reset button utilized for resetting ESP8266 chip, is one of the RST features located in the upper left corner. There is another FLASH button in the lower left corner of the board that is the download button, which is employed when updating the system. The board also has an LED indicator that can be programmed by the user and connected to the board's D0 pin.

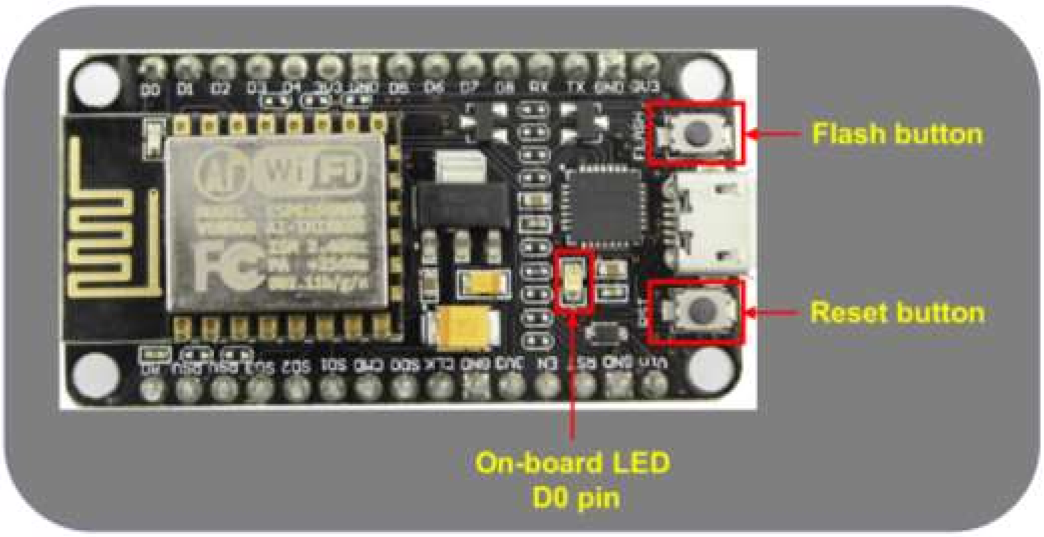


Figure 4. 4. ESP8266 NodeMcu buttons and LED indicator.

4.2.4. Serial connection

The board contains CP2102 USB-UART Bridge Controller developed by Silicon Labs (see Fig. 4.5), which converts USB signal into serial Transistor-Transistor Logic (TTL) numbers and allows the communication with PC for programming and interacting with ESP8266 chip.

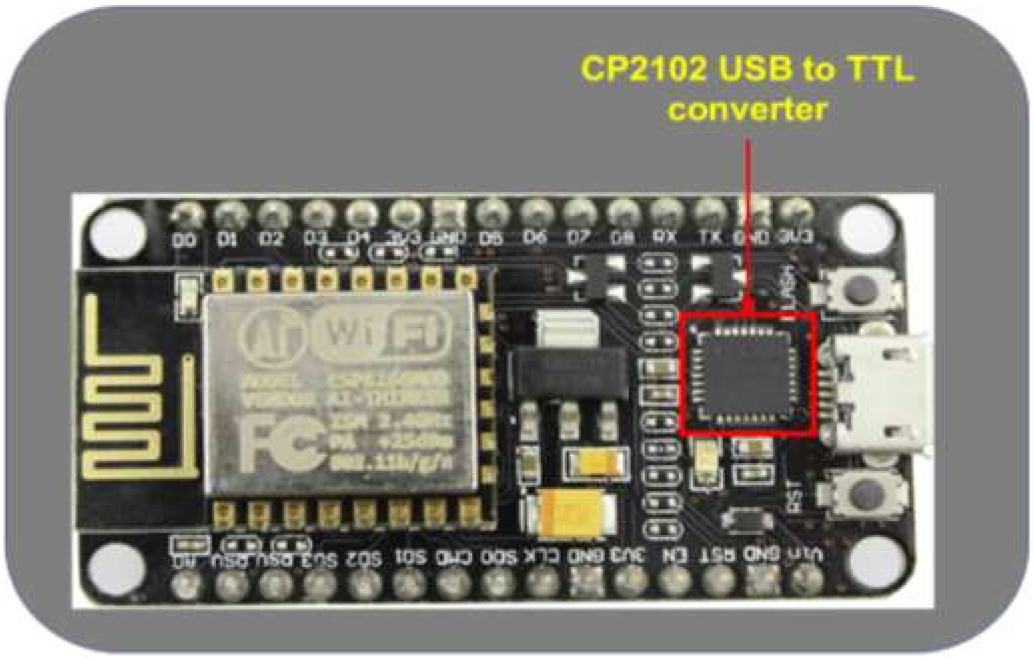


Figure 4. 5. ESP8266 NodeMcu USB to TTL converter.

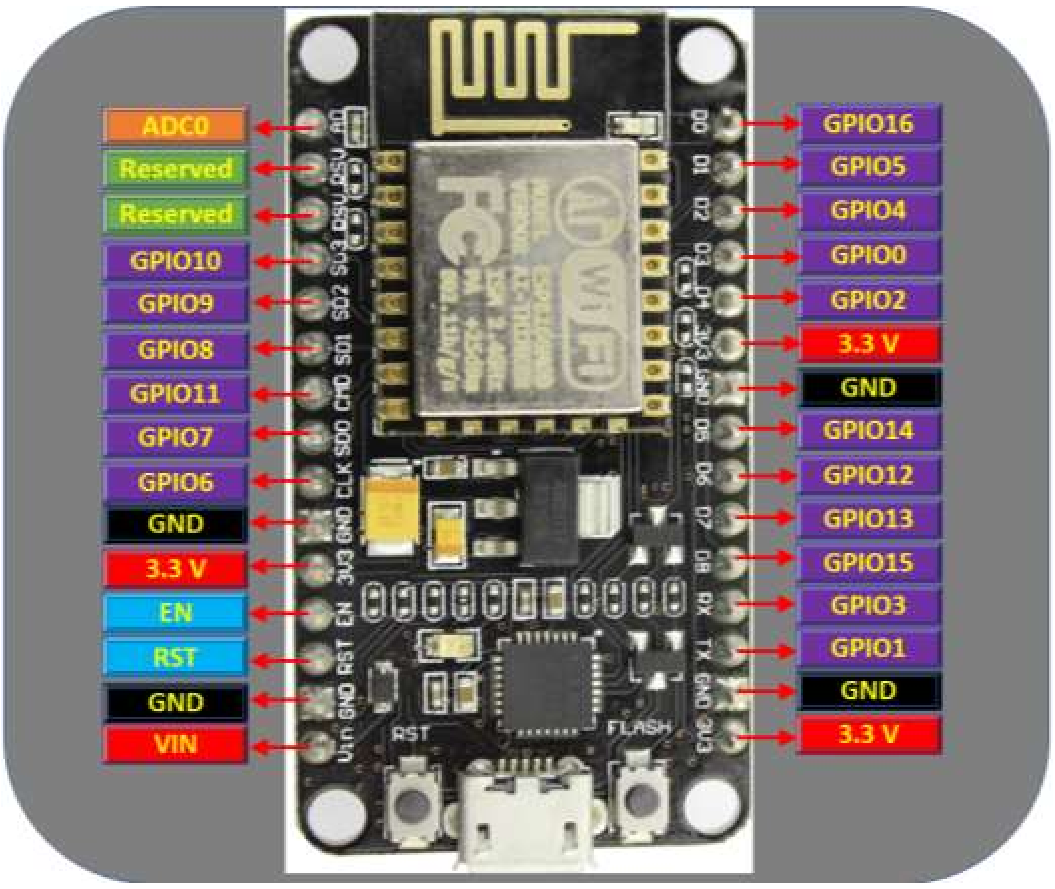


Figure 4. 6. ESP8266 NodeMCU Pinout.

4.2.5. ESP8266 NodeMCU Pinout

The ESP8266 NodeMCU contains overall 30 pins for connection to the external world (see Fig. 4.6). Groups of pins with similar capabilities are as follows:

Power Pins: ESP8266 NodeMCU contains 4 power pins. Three 3.3V pins and one VIN pin. A regulated 5V power supply can be connected to the VIN pin to provide direct power to the ESP8266 and its accessories. The 3.3V output pins are output of the internal voltage regulator. The 3.3V output pins are utilized to feed external components.

GND: It is the ground pin of the NodeMCU ESP8266 development board.

I^{2}C pins: These pins are utilized for connecting a variety of I^{2}C sensors and accessories. Both I^{2}C Master and I^{2}C slave are supported. The performance of the I^{2}C interface is achievable by the program and its maximum clock frequency is 100 kHz. It is worth noting that the clock frequency of I^{2}C must be greater than the lowest clock frequency of the slave device.

GPIO Pins: ESP8266 NodeMCU contains 17 GPIO pins that may be allocated to a variety of tasks like I2S, I^2C, PWM, UART, InfraRed remote Control, LED light and Button. Every active digital GPIO may be adapted to drag-up, pull-down, or adjusted to a high impedance. Once it is adapted as an input, it may be adjusted to the edge of the trigger or the trigger surface to create a CPU.

ADC: The ADC channel of NodeMCU is integrated with a Successive-Approximation-Register (SAR) ADC with resolution of 10-bits. Using ADC, VDD3P3 Voltage of power pin of VDD3P3 can be tested. In addition, ADC may be utilized to test voltage of TOUT pin. Although, it is not possible to run both tests simultaneously.

UART Pins: UART links of ESP8266 NodeMCU are UART0 and UART1. These interfaces make wireless communications (RS232 and RS485) possible and they are able to interact with maximum speed of 4.5 Mbps. TXD0, RXD0, RST0 and CTS0 pins of UART0 are utilized for communication. This controls and assists data flow. Although, PIN TXD1 of UART1 just contains a transfer signal for data.

SPI Pins: SPI pins of ESP8266 are SPI and HSPI with capability of operating in slave mode as well as master mode. In addition, SPI and HSPI uphold following characteristics of the general-purpose SPI.

- 4 SPI format transfer timing modes.
- Up to 80 MHz.
- First-In, First-Out (FIFO) buffering to maximum 64 bytes.

SDIO Pins: ESP8266 contains a Secure Digital Input and Output (SDIO) interface that can be utilized for direct interfacing to Secure Digital (SD) cards. Also, it supports 4-bit SDIO v1.1 @ 25 MHz and 4-bit SDIO v2.0 @ 50 MHz.

PWM pins: This board contains four Pulse Width Modulation (PWM) channels. PWM output is run by program and it is utilized for LED or motor control. The PWM frequency range can be adjusted from 100 Hz to 1 kHz.

Control pins: These pins are utilized for controlling the ESP8266. Control pins are Chip activation pin (EN), the reset pin (RST), and the WAKE pin.

- EN PIN: By activating EN PIN, it is possible to activate the ESP8266 chip. Once it is adjusted on low, operating power of the chip is minimum.
- RST pin: This pin is utilized for resetting the ESP8266 chip.

- WAKE PIN: When the chip is in the deep sleep mode, this pin is utilized to wake it up.

4.3. ESP8266 Development Programs

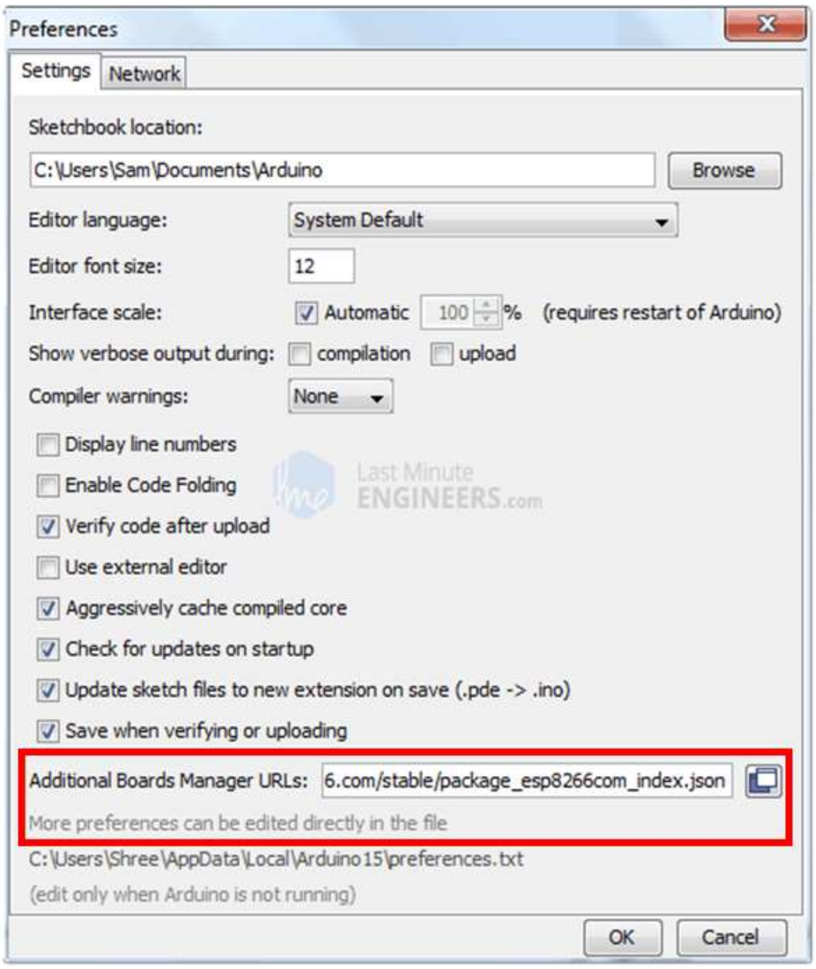
There are several types of developer frameworks that can be used to plan ESP8266. The Espruino - JavaScript SDK can be used for the operating system close to Node.js or used in the Mongoose framework that is an operation system for IoT devices (the program developed by Google Cloud IoT and Espressif Systems). Also, it can be used for Software Development Kit (SDK) developed by Espressif.

Fortunately, ESP8266 has taken an Arduino plugin to step up the IDE selection. This is the environment recommended for beginners in programming ESP8266.

4.3.1. Installation of ESP8266 Core on Windows operating system

To get started, the board manager should be updated with a dedicated URL. As it is shown in Fig. 4.7, Open the Arduino IDE then in the File tab select Preferences. next, in the Settings tab and in Additional Broad Manager URLs field, add the URL board manager (indicated below).

http://arduino.esp8266.com/stable/package_esp8266com_index.json



Preferences

Settings | Network

Sketchbook location:

C:\Users\Sam\Documents\Arduino Browse

Editor language: System Default

Editor font size: 12

Interface scale: ✓ Automatic 100 % (requires restart of Arduino)

Show verbose output during: ☐ compilation ☐ upload

Compiler warnings: None

☐ Display line numbers

☐ Enable Code Folding

✓ Verify code after upload

☐ Use external editor

✓ Aggressively cache compiled core

✓ Check for updates on startup

✓ Update sketch files to new extension on save (.pde -> .ino)

✓ Save when verifying or uploading

Additional Boards Manager URLs: 6.com/stable/package_esp8266com_index.json

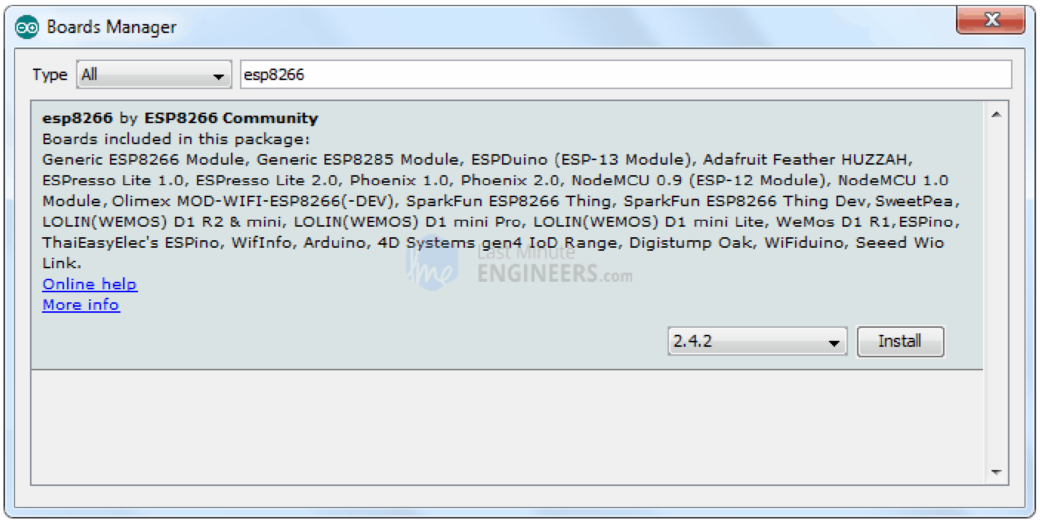
More preferences can be edited directly in the file

C:\Users\Shree\AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)

OK Cancel

Figure 4. 7. Arduino IDE Preferences page.



Boards Manager

Type All esp8266

esp8266 by **ESP8266 Community**
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 Module, Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino, ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Seeed Wio Link.
Online help
More info

2.4.2 Install

Figure 4. 8. Board manager of Arduino IDE.

Click ok. Then in the tab Tools go to Boards, then from Boards Manager go to Manager Board. Apart from the common Arduino boards, you must have several new entries. Searching for esp8266, the search of boards can be limited. Click on that input and select Install (see Fig. 4.8).

After installation, a small text message (“INSTALLED”) will appear next to the input. Then, select “esp8266 by ESP8266 Community” and click on Install to install it.

4.4. ESP8266 practical modes

As well as the ability of ESP8266 to connect to existent wireless network, it may establish its own network and connect to other devices. This allows a direct connection to the internet for accessing web pages. ESP8266 can work in the following different modes:

1 Working in station mode (STA).

2 Working in Soft Access Point mode (AP).

3 Simultaneously working in STA and AP.

The third mode can be used to build mesh networks.

4.4.1. Station mode (STA)

The case in that ESP8266 links with a WiFi network that is already established by using a wireless router is referred to the Station mode (STA). In this case, IP address of ESP8266 IP can be get from the wireless router. Using the obtained IP address, ESP8266 can create a web server. This web server can convey web pages to other devices in the WiFi network (see Fig. 4.9 for an example).

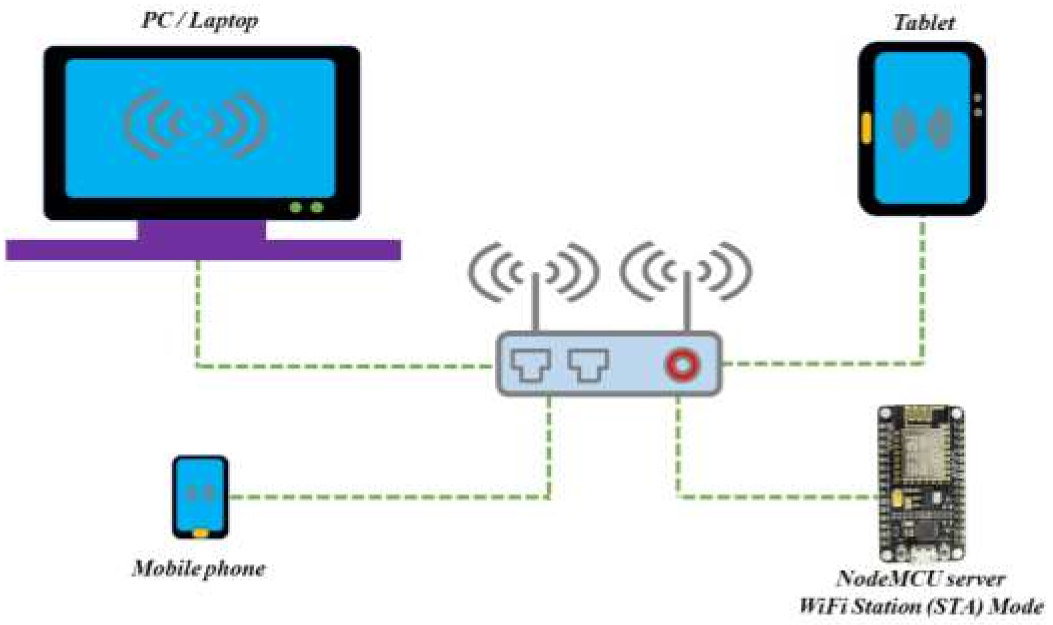


Figure 4. 9. An example of a network with NodeMCU in WiFi Station (STA) mode.

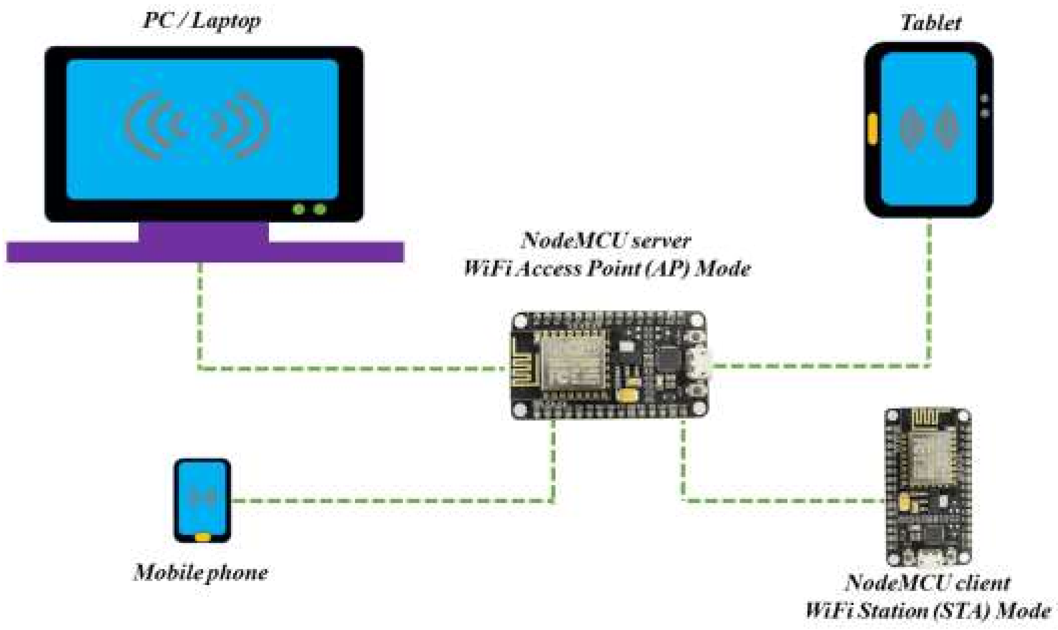


Figure 4. 10. An example of a network with NodeMCU in WiFi Access Point (AP) mode.

4.4.2. Soft access mode (AP)

The case in that ESP8266 sets up a WiFi network autonomously and operates as a hub exactly resemble to a router refers to the Access Point (AP) mode. The established WiFi network can include one or several stations (up to 5). It should be mentioned that dissimilar to a proper WiFi router, ESP8266 contains no interfaces for connecting to the wired network. Therefore, this method is known as the Soft Access Point (soft-AP).

This way, ESP8266 sets up its fresh WiFi network and determines Service Set Identifier (SSID or network name) with an IP address. Web pages can be delivered toward other devices in the network by using the given IP address.

NodeMCU is a development board that is specifically used on the Internet of Things. This module has a FIRMWARE that runs on the ESP8266 WiFi chip, which is open and programmable for device control purposes. The ESP8266 module consists of a 32-bit microcontroller with multiple inputs and outputs, called a GPIO.

The most important feature of this microcontroller is having a WiFi protocol that supports all TCP / IP capabilities and can be used in both Access Point and Client Mode. Using CH340 chip, this microcontroller can employ a USB to serial converter.

The new NodeMCU V3 is a low-cost Wi-Fi that provides high-speed and supports a high-level programming based on LUA. This integrated unit contains many on-board resources, internal USB-TTL serial, industrial power CH340 with extraordinary reliability is supported in all operating systems. It is very useful for implementing projects based on Arduino or other development boards with available I/O pins. NodeMCU supports several programming languages. Therefore, it is possible to download programs from any computer through the micro USB port.

Since NodeMCU takes advantageous of the widely used ESP8266 technology, to implement powerful solutions for the microcontroller program many resources that are accessible in the web can be used. NodeMCU can be employed as a part of the Internet of Things (IoT) projects. It contains an ESP-12-based WiFi series with an integrated processor and offers ADC, PWM, GPIO, I^2C and 1-WIRE resources.

Arduino Due employs the MCU board with non AVR-based processors like SAM/ARM MCUs. Therefore, Arduino IDE has been changed for supporting replacement devices. This makes it possible to compile Arduino C / C ++ for non

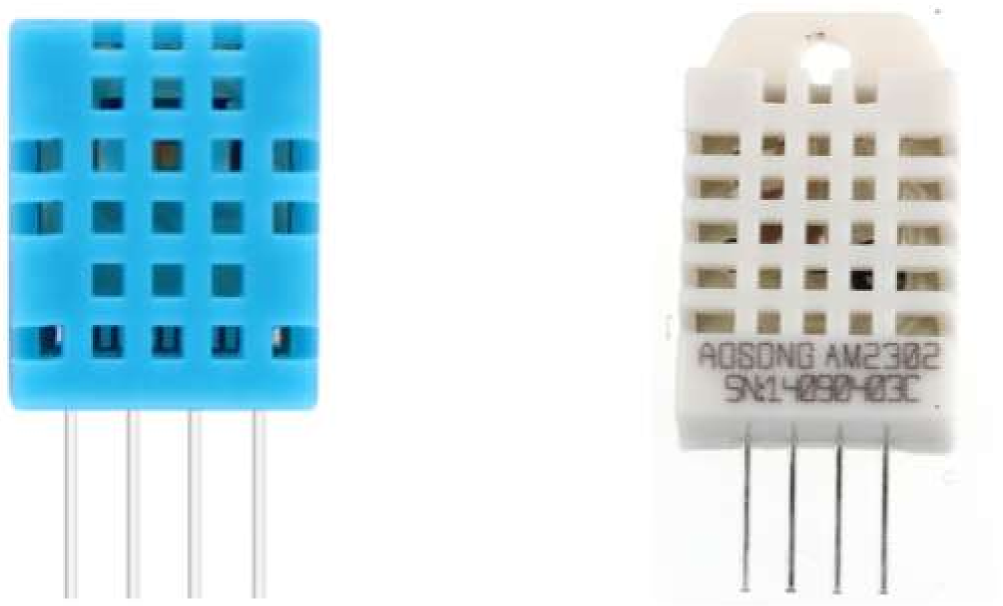
AVR-based processors. For this reason, SAM Core was introduced. This Core contains a group of software constituents that are needed to create Arduino C / C ++ source files [83]. For NodeMCUs and other development modules based on ESP8266, SAM core has turned to a chief platform.

Features:

- Communication voltage: 3.3 volts
- Antenna type: An integrated PCB antenna is available
- Modes of Wireless Network: SoftAP, STA, STA + SoftAP
- WIFI @ 2.4 GHz 802.11 b / g / n
- Support for WPA / WPA2 security mode

4.5. DHT11 humidity and temperature sensor

Currently, two editions of the DHTxx sensor series are available. Both versions are shown in Fig. 4.11. As can be seen, physically they look a bit alike with identical pinout. However, they have different features that make them distinguishable from each other. Specification of DHT11 and DHT22 are listed in Table 4.I.



(a) (b)

Figure 4. 11. (a) DHT11, (b) DHT22.

DHT22 is a more expensive than DHT11 with superior features. Range of temperature measurement of DHT22 is between -40°C and + 125°C and temperature range of DHT11 is between 0°C and 50°C. Measurement precision of DHT22 is ±0.5°C, and for DHT11 is ±2°C. In addition, DHT22 offers a wider humidity measurement range between 0% and 100% while humidity measurement range of DHT11 is from 20% to 80%. DHT22 is capable of measuring the humidity with a higher accuracy than DHT11. The humidity measurement accuracy of DHT22 is 2-5%, while the accuracy of DHT11 in humidity measurement is 5%.

Table 4.I. Specifications of DHT11 and DHT22 sensors.

	DHT11	**DHT22**
OPERATING VOLTAGE	**3 V to 5 V**	**3 V to 5 V**
MAX OPERATING CURRENT	**2.5 mA max**	**2.5 mA max**
HUMIDITY RANGE	**20% to 80% with 5% accuracy**	**0 to 100% with 2.5% accuracy**
TEMPERATURE RANGE	**0 to50°C with ±2°C accuracy**	**-40 to 80°C with ±0.5°C accuracy**
SAMPLING RATE	**1 Hz**	**0.5 Hz**
BODY SIZE	**15.5 mm × 12mm × 5.5mm**	**15.1 mm × 25mm × 7.7mm**
MAIN ADVANTAGE	**Low cost**	**More accurate**

Although DHT22 measures a wider range of temperatures and humidity with higher accuracy. There are three things that make DHT11 better than DHT22 for some projects. These features are more affordable, smaller size and higher sampling rate. DHT22 sampling rate is 0.5Hz, which means it measures once every two seconds. The sampling rate of DHT11 is 1Hz, which means it reads once every second. So DHT11 is faster in terms of sampling.

Both sensors have the operating voltage from 3 V to 5V. Their maximum consumption current during conversion is 2.5mA. Above all, DHT11 and DHT22 sensors are "interchangeable". This means that if a project is based on one sensor, it can be replaced by the other (since pinouts and operating voltage and current are same). Codes may change a bit but at least the wiring is the same.

DHT11 is a digital humidity and temperature sensor offering basic features in a very low cost. It employs a capacitive humidity sensing component for humidity measurement and contains a thermistor used for temperature measurement of environment. After measurement, the sensor and sends a signal in digital form to its data pin therefore the analog input pin is not required). DHT11 is a user-friendly sensor but a precise time scheduling is required for acquiring the information. The Main disadvantage of DHT11 is that it imposes a delay of 2 seconds in reading of measured values. This is because DHT11 updates the measured values every 2 seconds.

Currently, two editions of DHT11 are available. These two versions differ from each other in terms of their pin numbers. One version contains 4 pins and the other contains 3 pins and it is installed on a PCB. The PCB-installed edition works very well as it contains a 10KΩ installed resistor for signal linearization. Figure 4.12 shows output pins for both versions.

4.5.1. DHT11 and DHT22 hardware overview

Inside the housing of DHT11 and DHT22 there are two parts., one humidity sensor and an NTC temperature sensor (see Fig. 4.13). The humidity measurement component, of course, is used to measure humidity, which has two electrodes with a humidity-retaining material between them (see Fig. 4.14). The humidity-retaining material is typically conductive plastic polymer or salt. Once the humidity-retaining material absorbs humidity, it releases Ions. These ions increase conductivity of the connection within the electrodes. The humidity is proportional to resistance change of electrodes connections. The resistance between the electrodes reduces by increase of humidity and it increases by decrease of humidity.

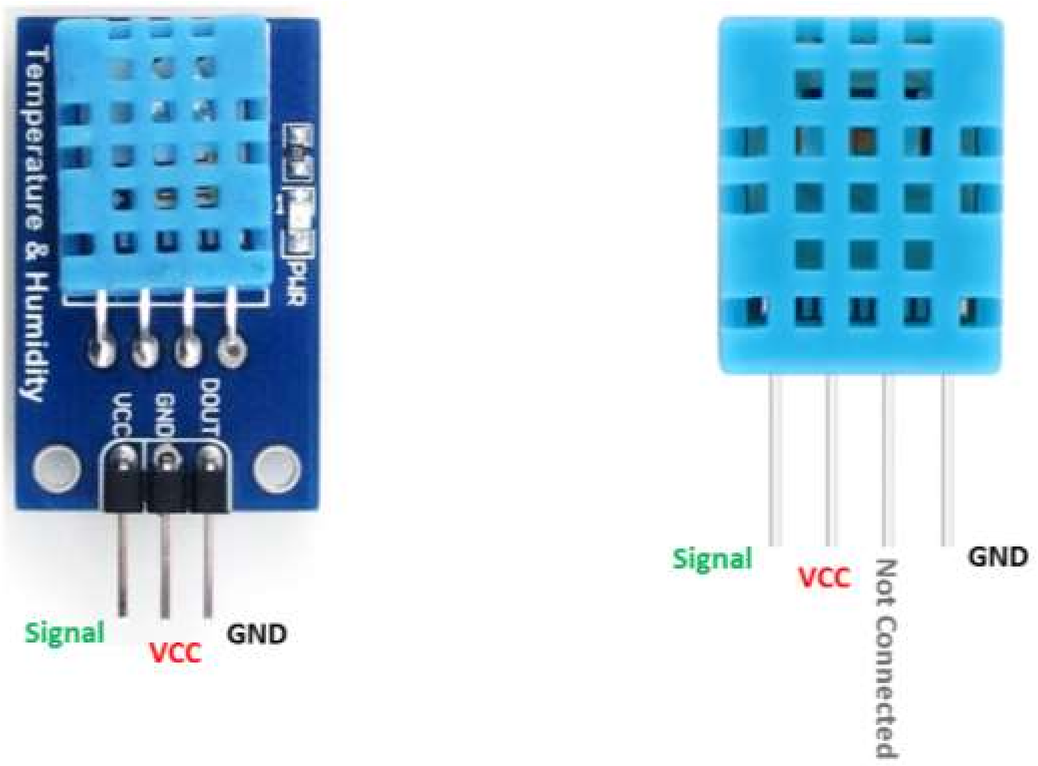


Figure 4. 12. Output pins of DHT11 different versions.

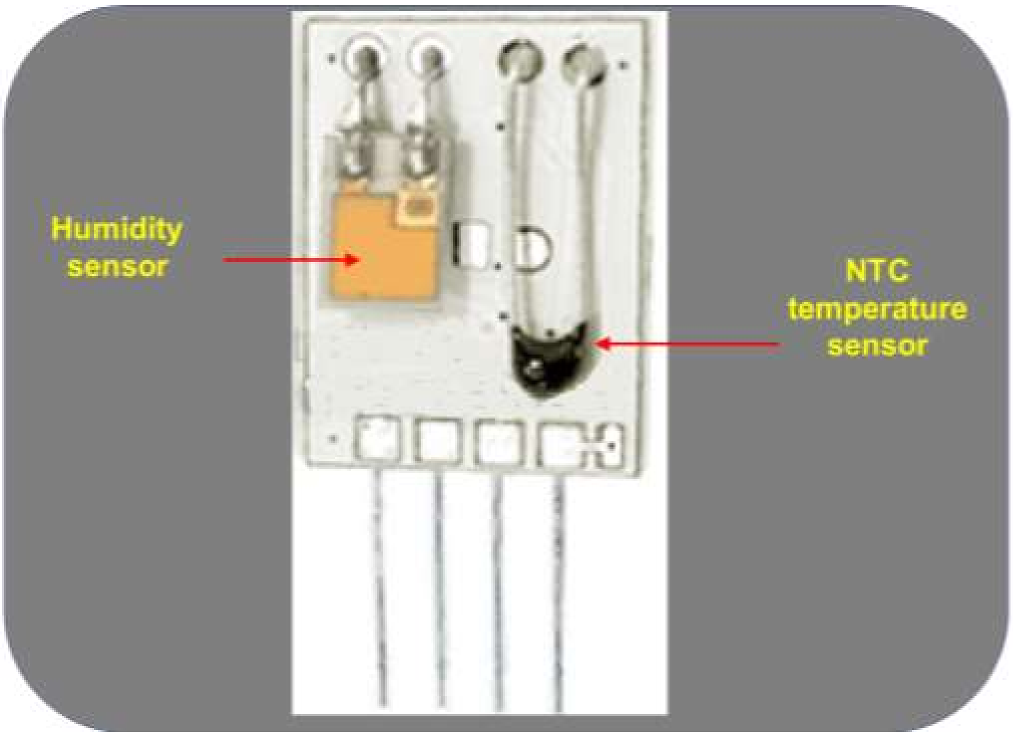


Figure 4. 13. DHT11 internal components.

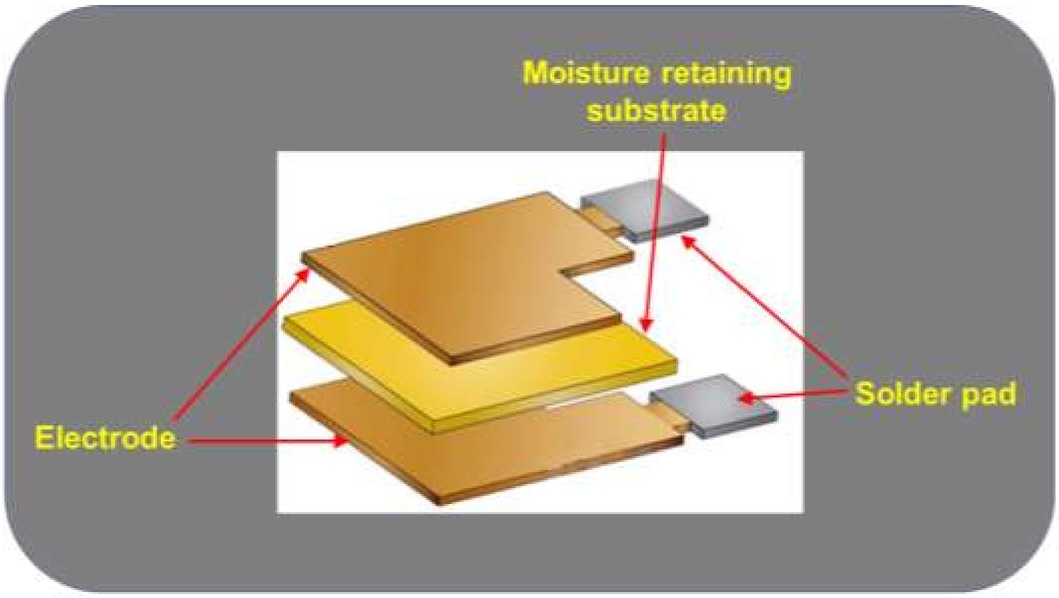


Figure 4. 14. DHT11 humidity measurement component.

In addition, an NTC thermometer is employed for the temperature measurement. In fact, Thermistors are thermal resistors and their resistance is sensitive to the temperature. The resistance of a thermal resistor changes by variation of temperature. These changes are used as an indication of increase or decrease of temperature. Figure 4.15 shows resistance changes of an NTC versus temperature. As can be seen, increasing the temperature, resistance of NTC decreases.

4.5.2. DHT11 and DHT22 Pinout

The VCC pin provides the power for the sensor. Voltage range of the sensor is from 3.3 V to 5.5 V, however 5 V power supply is more recommended. Supplying the sensor by 5 V, it is possible to use the sensor with a wire up to 20 meters. But, supplying the sensor by 3.3 V, length of wire must not be more than 1 m. Otherwise, voltage drop of the wire will result in a measurement error.

The data pin is used for communication between the microcontroller and the sensor.

NC labeled pins are not connected.

GND pin must be connected to the ground of the microcontroller.

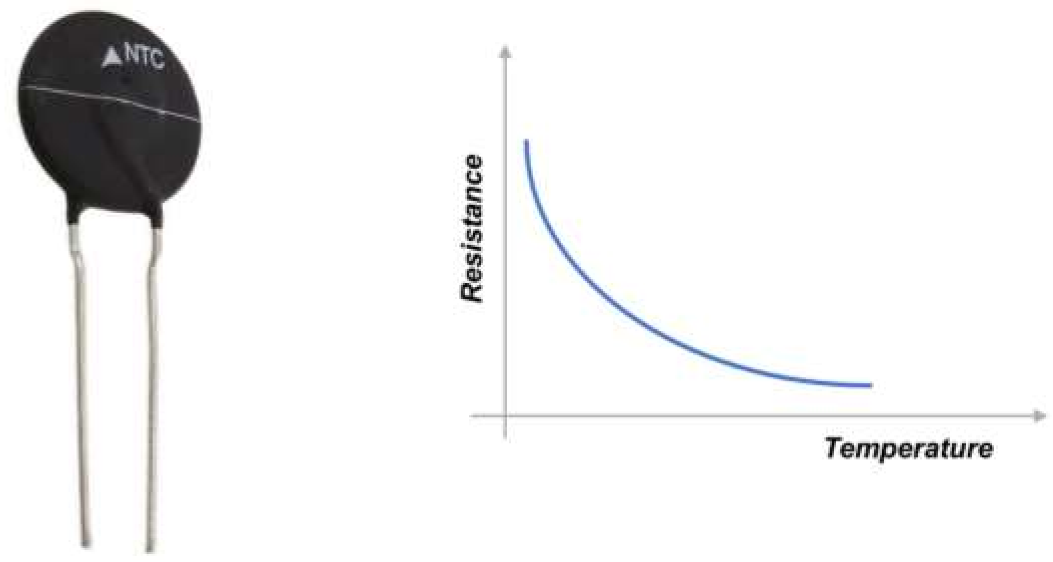


Figure 4. 15. Characteristic curve of NPC.

4.6. Connecting DHT11 to NODEMCU

In this section, the connection of the DHT11 to the NodeMCU is described. Connections may be different depending on number of pins of the sensor.

The wiring connections are as follows:

1. Pin + goes from DHT11 to NodeMCU 3V3 pin.
2. Pin out of DHT11 connects to one of the NodeMCU digital pins (Pin D1).
3. PIN - Goes from DHT11 to NodeMCU Ground Pin (GND).

Figure 4.16 shows the implemented wireless sensor network including nodes, Arduino and computer communicating with each other. Node 2 and Node 3 act as static points and are connected to node 1. This node operates as the access point and it is capable to communicate with other nodes. On the other hand, it is connected to two LEDs and controls their switching ON and OFF through the html screen. Finally, Arduino connected to the sensor is connected to the computer, and the connection between the temperature sensor and the computer is made possible in this way. Figure 4.17 shows the access point node and one of the station nodes connected to a sensor is shown in Fig. 4.18.

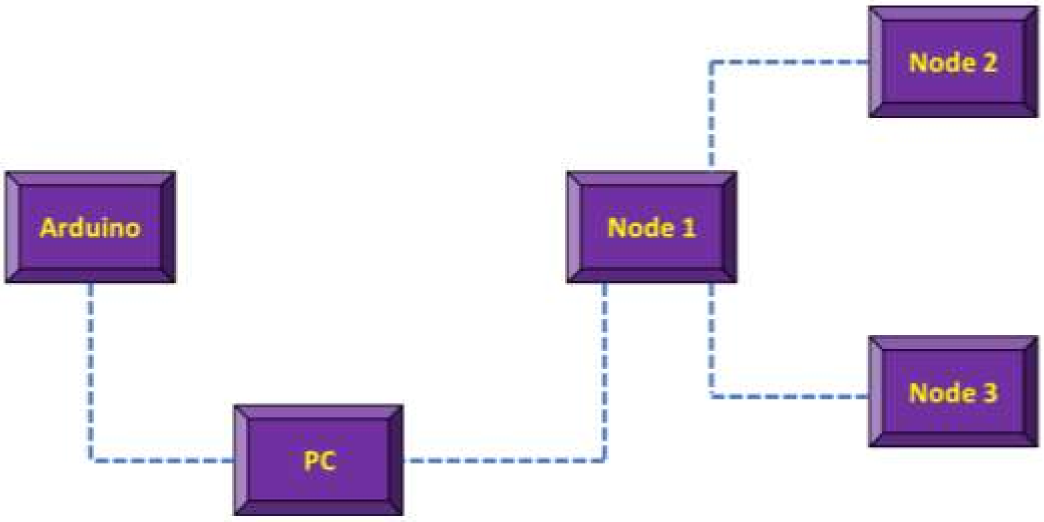


Figure 4. 16. Schematic of the implemented wireless sensors network.

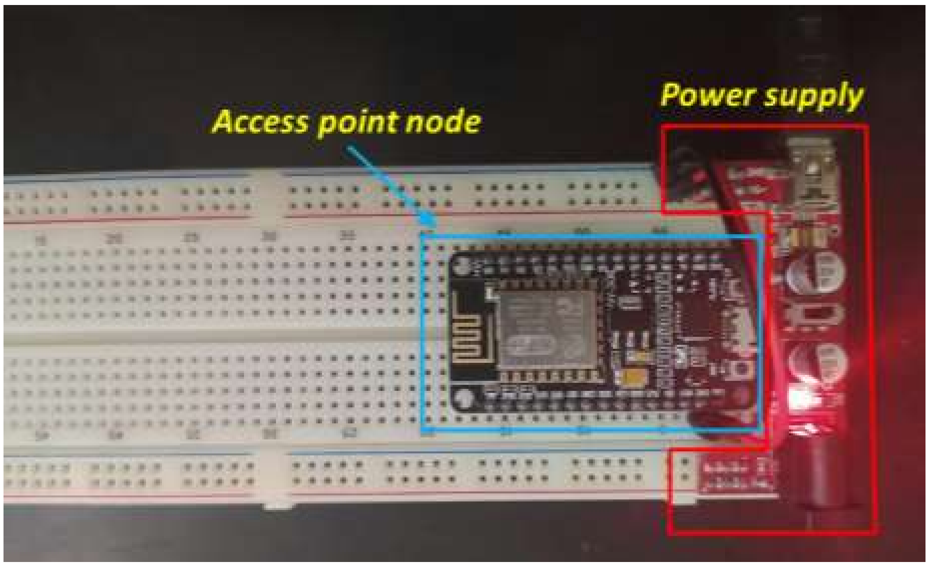


Figure 4. 17. Access point node.

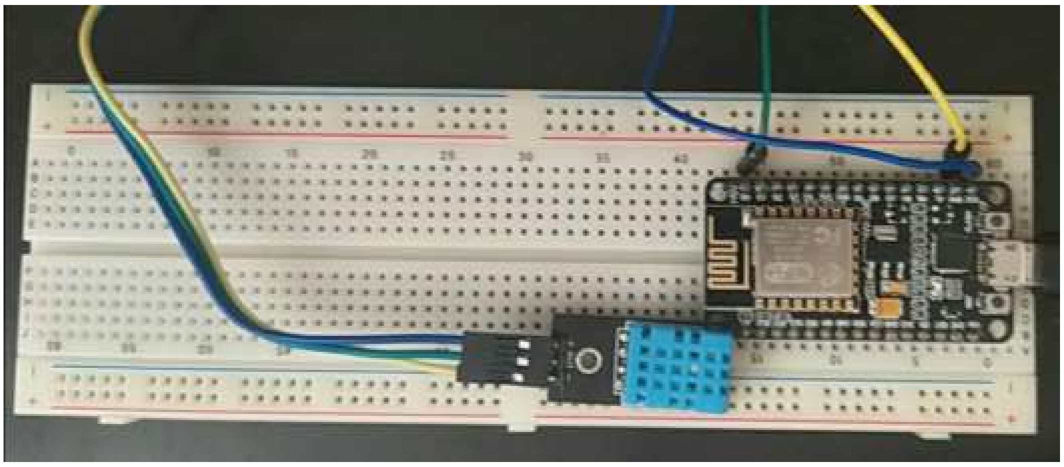


Figure 4. 18. One station node connected to a sensor.

4.7. Codes

4.7.1. For nodes connected to a sensor

Before using DHT11 with NodeMCU, DHTLib library must be installed. Installing the library, it receives all the tasks required to obtain humidity and temperature from the sensor. Open the Arduino IDE. Next, from “Sketch” select “Library”, then select “Library Management”, and finally “DHTLib”. Codes that must be loaded on NodeMCU are as follows.

The humidity and temperature readings are indicated with one-second intervals.

```
//define lib

#include <ESP8266WiFi.h>

#include <ESP8266WebServer.h>

#include "DHT.h"

#define DHTTYPE DHT11   // For DHT 11 sensor

//#define DHTTYPE DHT21   // For DHT 21 (AM2301)

/*Enter SSID & Password*/
```

```
const char* ssid = "MCU";   // Here enter SSID

const char* password = "12345678";   //Here enter your Password

ESP8266WebServer server(80);

// DHT Sensor

uint8_t DHTPin = D2; // assigning pin D2 for the sensor input

// Initialize DHT sensor.

DHT dht(DHTPin, DHTTYPE);

float Temperature; //declaration of variable

float Humidity; //declaration of variable (not used in this project)

void setup() {

  Serial.begin(115200);// Serial connection beginning

  delay(100);//delay

  pinMode(DHTPin, INPUT);// Mode determining (input) for the connected sensor
pin.

  dht.begin(); // Start of sensor operation

  Serial.println("Connecting to ");// Writes “connecting to”

  Serial.println(ssid);

  //connection to the local available WiFi network

  WiFi.begin(ssid, password);//Start of WiFi

IPAddress ip(192,168,1,200);   // Assigning a specific IP for this node

IPAddress gateway(192,168,1,254);
IPAddress subnet(255,255,255,0);

WiFi.config(ip, gateway, subnet); //Canfiguration wifi

 //WiFi connection checking

  while (WiFi.status() != WL_CONNECTED) {

  delay(1000);

  Serial.print(".");// Printing points until WiFi is connected.

  }

  Serial.println("");

  Serial.println("WiFi connected..!");// WiFi connection message

  Serial.print("Got IP: ");  Serial.println(WiFi.localIP());

  server.on("/", handle_OnConnect);

  server.onNotFound(handle_NotFound);

  server.begin();// Start of the server

  Serial.println("HTTP server started");// start of reading of the server http

}

void loop() {

  server.handleClient();

}

void handle_OnConnect() {

 Temperature = dht.readTemperature(); // Measured temperature value is given

  Humidity = dht.readHumidity(); // Measured humidity value is given
  server.send(200,  "text/html",  SendHTML(Temperature,Humidity));  //  Send  to
HTML

}

void handle_NotFound(){

  server.send(404, "text/plain", "Not found");// Error of the IP is wrong

}

String SendHTML(float Temperaturestat,float Humiditystat){

  String ptr = "<!DOCTYPE html> <html>\n";// Beggining of Html codes

  ptr +="<head><meta name=\"viewport\" content=\"width=device-width, initial-
scale=1.0, user-scalable=no\">\n";

  ptr +="<title>ESP8266 Weather Report</title>\n";

  ptr +="<style>html { font-family: Helvetica; display: inline-block; margin: 0px
auto; text-align: center;}\n";

  ptr +="body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto
30px;}\n";

  ptr +="p {font-size: 24px;color: #444444;margin-bottom: 10px;}\n";

  ptr +="</style>\n";

  ptr +="</head>\n";

  ptr +="<body>\n";

  ptr +="<div id=\"webpage\">\n";

  ptr +="<h1>ESP8266 NodeMCU Weather Report</h1>\n";

  ptr +="<p>Temperature: ";
  ptr +=(int)Temperaturestat;

  ptr +="°C</p>";

  ptr +="<p>Humidity: ";

  ptr +=(int)Humiditystat;

  ptr +="%</p>";

  ptr +="</div>\n";

  ptr +="</body>\n";

  ptr +="</html>\n";

  return ptr;

}
```

Note that the highlighted section is a separate address for the node that collects information from the rest of the nodes and acts as an access point.

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
/* Enter Password and SSID */
const char* ssid = "MCU";  // Here enter SSID
const char* password = "12345678";  //Here enter Password
/* Enter details of IP Address */
IPAddress local_ip(192,168,1,1);//IP of the access point
IPAddress gateway(192,168,1,1);
IPAddress subnet(255,255,255,0);
ESP8266WebServer server(80);
void setup() {
  Serial.begin(115200);
  WiFi.softAP(ssid, password);
  WiFi.softAPConfig(local_ip, gateway, subnet);
  delay(100);
  server.on("/", handle_OnConnect);
server.onNotFound(handle_NotFound);
  server.begin();
  Serial.println("HTTP server started");
}
void loop() {
  server.handleClient();
}
void handle_NotFound(){
  server.send(404, "text/plain", "Not found");
}
```

4.7.2. Connection between the node and Arduino with serial communication

Figure 4.19 indicates how Arduino board and NodeMCU serial connection is established. Arduino RX and TX pins are connected to TX and RX pins of NodeMCU, respectively. Also, the ground on both boards must be connected to the same point.

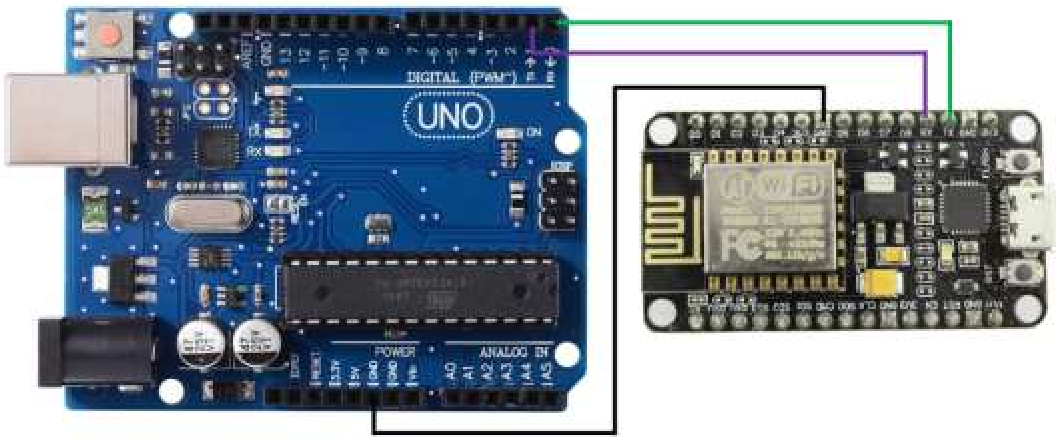


Figure 4. 19. Connection between a node and Arduino.

4.7.2.1. Codes of the node and Arduino are in separate form.

.1

```
#include <SoftwareSerial.h>
SoftwareSerial s(D6,D5);
#include <ArduinoJson.h>

void setup() {
  // Serial port initialization
  Serial.begin(9600);
  s.begin(9600);
  while (!Serial) continue;

}

void loop() {
 StaticJsonBuffer<1000> jsonBuffer;
  JsonObject& root = jsonBuffer.parseObject(s);
  if (root == JsonObject::invalid())
    return;

  Serial.println("JSON received and parsed");
  root.prettyPrintTo(Serial);
  Serial.print("Data 1 ");
  Serial.println("");
  int data1=root["data1"];
  Serial.print(data1);
  Serial.print("   Data 2 ");
  int data2=root["data2"];
  Serial.print(data2);
  Serial.println("");
  Serial.println("---------xxxxx---------");
}
```

4.7.2.2. Arduino codes for the serial connection

```
#include <SoftwareSerial.h>
#include <ArduinoJson.h>
SoftwareSerial s(5,6);

void setup() {
s.begin(9600);
}

void loop() {
 StaticJsonBuffer<1000> jsonBuffer;
 JsonObject& root = jsonBuffer.createObject();
  root["data1"] = 100;
  root["data2"] = 200;
if(s.available()>0)
{
 root.printTo(s);
}
}
```

CHAPTER 5.

5. Results

5.1. Introduction

This chapter presents experimental results obtained from the project. Also, in this chapter, codes for Matlab Graphical User Interface (GUI) are explained.

To test the performance of the network, the average daily temperature of the sensors was measured with 6-hour measurement intervals. Results of the measurement in one week are listed in Table 5.I. All temperatures are in degrees Celsius.

Table 5. I. Measured temperatures.

	Monday	Tuesday	Wednesdat	thursay	Friday	Saturday	Sunday
Room1	**25**	**23**	**22**	**21**	**26**	**22**	**24**
room2	**24**	**23**	**23**	**21**	**24**	**23**	**25**
outside	**23**	**22**	**19**	**18**	**26**	**20**	**22**

kitchen	**27**	**26**	**25**	**25**	**28**	**28**	**26**

The consistency of the results shows that the sensors and the network are working well.

5.2. GUI Design

In the GUI section, using the app designer section, an app was designed to provide graphical interface between the user and the network. The following program includes a pushbutton and a dropdown. The button is named temperature and the dropdown named region. In the region dropdown, four areas are defined as four different strings called ‘room1’, ‘room2’, ‘outside’ and ‘kitchen’. Two function are activated from the menu in the appdesigner (in codeview). Pressing the button leads to a defined action and if the drop value is changed, the value returned by it will change. Mentioned functions are added to the program code. The overview of the GUI is shown in Fig. 5.1. Figure 5.2 shows the html page. The html page for LED control of the access point is shown in Fig. 5.3.

In the following code, if the temperature button is pressed, first the dropdown value is checked to determine its specified string. Then, via the IP assigned to that area, enters the html page for that area. If the dropdown value is changed, these changes will be saved in the room variable. Matlab appdesigner generated the rest of the code automatically.

5.2.1. Function codes

```
function temperatureButtonPushed(app, event)
        room = app.regionDropDown.Value;
        if room=='room1'
           url = '192,168,1,200';
           web(url)
        elseif room=='room2'
           url = '192,168,1,201';
           web(url)
        elseif room=='outside'
           url = '192,168,1,202';
        web(url)
    elseif room=='kitchen'
        url = '192,168,1,203';
        web(url)
    end
end
% Value changed function: regionDropDown
function regionDropDownValueChanged(app, event)
    room = app.regionDropDown.Value;
 end
```

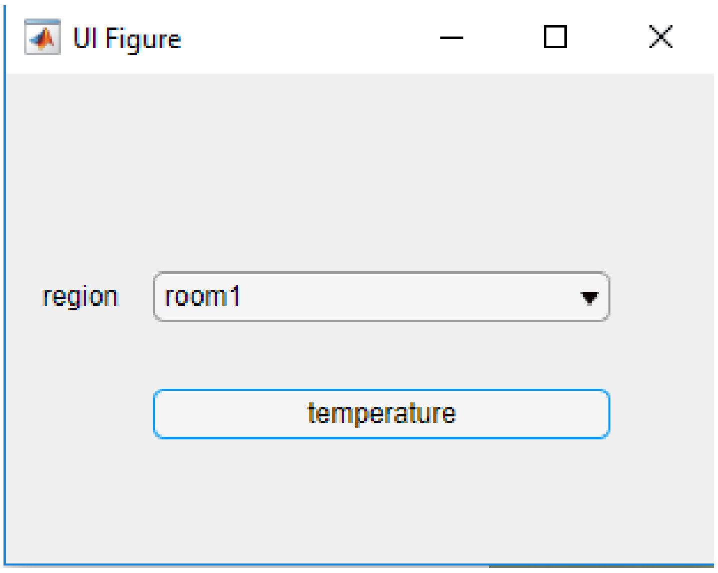
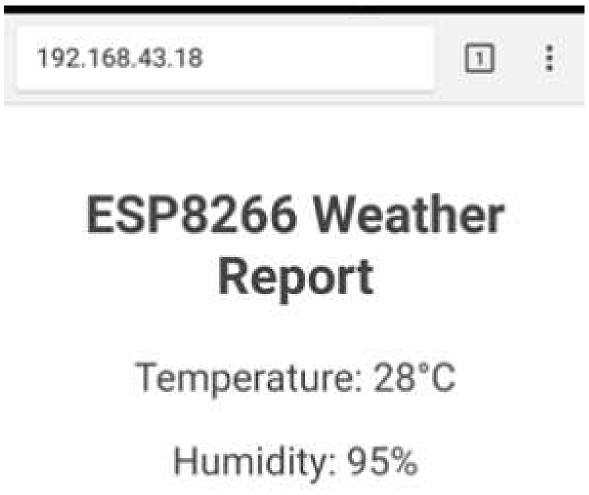


Figure 5. 1. GUI.



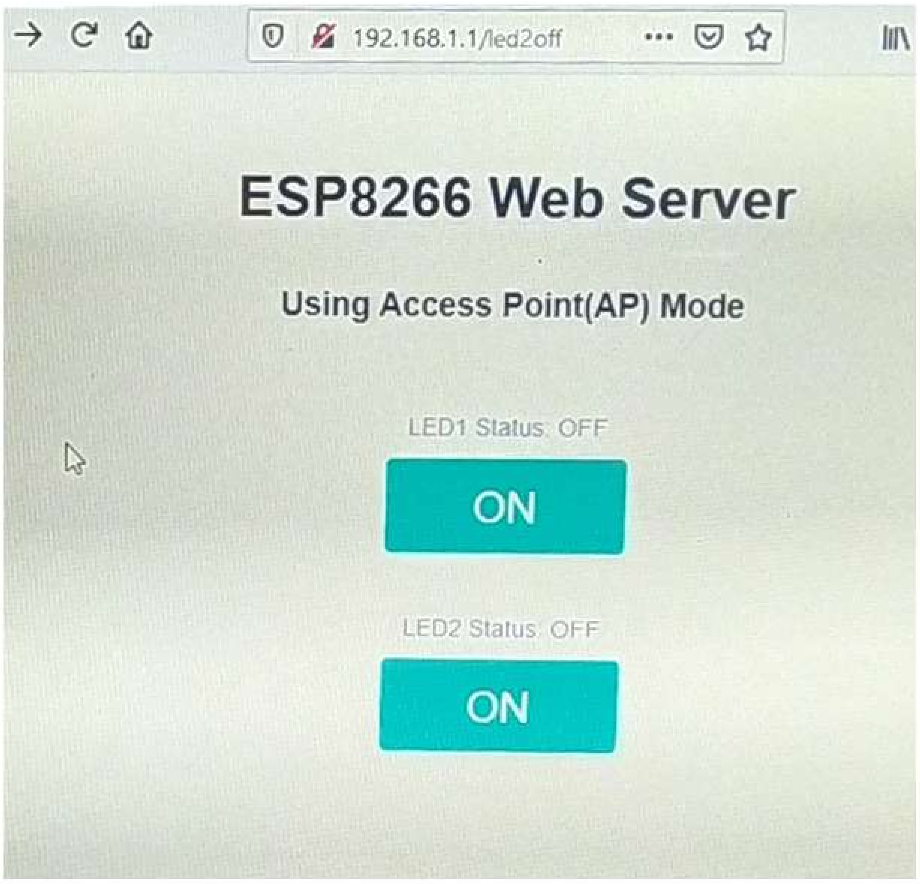
192.168.43.18

ESP8266 Weather Report

Temperature: 28°C

Humidity: 95%

Figure 5. 2. Html page.



192.168.1.1/led2off

ESP8266 Web Server

Using Access Point(AP) Mode

LED1 Status: OFF

LED2 Status: OFF

Figure 5. 3. Html page for LED control of the Access point.

5.2.2. GUI codes

```
classdef app1 < matlab.apps.AppBase
    % App components' properties
    properties (Access = public)
        UIFigure            matlab.ui.Figure
        regionDropDownLabel  matlab.ui.control.Label
        regionDropDown       matlab.ui.control.DropDown
        temperatureButton    matlab.ui.control.Button
    end
```

```
methods (Access = private)
    % Pushed button function: temperatureButton
    function temperatureButtonPushed(app, event)
        room = app.regionDropDown.Value;
        if room=='room1'
            url = '192,168,1,200';
            web(url)
        elseif room=='room2'
            url = '192,168,1,201';
            web(url)
        elseif room=='outside'
            url = '192,168,1,202';
            web(url)
        elseif room=='kitchen'
            url = '192,168,1,203';
            web(url)
        end
    end
    % Value changed function: regionDropDown
    function regionDropDownValueChanged(app, event)
        room = app.regionDropDown.Value;

    end
end
% Construction and initialization of app
methods (Access = private)
    % Components and UIFigure creation
    function createComponents(app)
        % UIFigure creation
        app.UIFigure = uifigure;
        app.UIFigure.Position = [100 100 209 213];
        app.UIFigure.Name = 'UI Figure';
        % regionDropDownLabel creation
        app.regionDropDownLabel = uilabel(app.UIFigure);
```

```
            app.regionDropDownLabel.HorizontalAlignment = 'right';
            app.regionDropDownLabel.Position = [11 106 39 22];
            app.regionDropDownLabel.Text = 'region';
            % Create regionDropDown
            app.regionDropDown = uidropdown(app.UIFigure);
            app.regionDropDown.Items = {'room1', 'room2', 'outside', 'kitchen'};
            app.regionDropDown.ValueChangedFcn    =    createCallbackFcn(app,
@regionDropDownValueChanged, true);
            app.regionDropDown.Position = [65 106 100 22];
            app.regionDropDown.Value = 'room1';
            % Create temperatureButton
            app.temperatureButton = uibutton(app.UIFigure, 'push');
            app.temperatureButton.ButtonPushedFcn    =    createCallbackFcn(app,
@temperatureButtonPushed, true);
            app.temperatureButton.Position = [65 55 100 22];
            app.temperatureButton.Text = 'temperature';
        end
    end
    methods (Access = public)
        % App construction
        function app = app1
            % Components creation and configuration
            createComponents(app)
            % Registration of the app with App Designer
            registerApp(app, app.UIFigure)
            if nargout == 0
                clear app
            end
        end
        % Executive code before deletion of app
        function delete(app)
            % Once app is deleted, delete UIFigure
            delete(app.UIFigure)
        end
        end
    end
```

6. References

[1] Lalatendu Muduli, Devi Prasad Mishra, Prasanta K. Jana, “Application of wireless sensor network for environmental monitoring in underground coal mines: A systematic review,” Journal of Network and Computer Applications,Volume 106,pp 48-67, 2018.

[2] M. Foster et al., "Preliminary Evaluation of a Wearable Sensor System for Heart Rate Assessment in Guide Dog Puppies," IEEE Sensors Journal, vol. 20, no. 16, pp. 9449-9459, 2020.

[3] T. W. Foster, D. V. Bhatt, G. P. Hancke and B. Silva, "A Web-Based Office Climate Control System Using Wireless Sensors," IEEE Sensors Journal, vol. 16, no. 15, pp. 6104-6113, 2016.

[4] Z. Hu et al., "Application of Non-Orthogonal Multiple Access in Wireless Sensor Networks for Smart Agriculture," IEEE Access, vol. 7, pp. 87582-87592, 2019.

[5] D. Wohwe Sambo, A. Forster, B. O. Yenke, I. Sarr, B. Gueye and P. Dayang, "Wireless Underground Sensor Networks Path Loss Model for Precision Agriculture (WUSN-PLM)," IEEE Sensors Journal, vol. 20, no. 10, pp. 5298-5313, 2020.

[6] A. A. N. Azlin, H. Mansor, A. Z. Hashim and T. S. Gunawan, "Development of modular smart farm system," 2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), Putrajaya, pp. 1-6, 2017.

[7] Andrew Donnelly on March 25, 2013, https://www.mikogo.com/2013/03/25/evolution-of-communication-infographic/

[8] G. S. Smith, "Analysis of Hertz's Experimentum Crucis on Electromagnetic Waves [Historical Corner]," IEEE Antennas and Propagation Magazine, vol. 58, no. 5, pp. 96-108, Oct. 2016.

[9] M. Fabbri and G. Pelosi, "Guglielmo Marconi: Some New Documents Covering the Years 1894–1896 [Historical Corner]," IEEE Antennas and Propagation Magazine, vol. 55, no. 2, pp. 291-302, April 2013.

[10] B. A. Austin, "Wireless in the boer war," IEEE Electromagnetic Compatibility Magazine, vol. 6, no. 1, pp. 30-35, 2017.

[11] Rajeev Bansal, "Communication Systems," From ER to E.T.: How Electromagnetic Technologies Are Changing Our Lives, IEEE, pp.157-173, 2017.

[12] R. D. Gitlin and G. I. Zysman, "The expanding world of wireless technology," Bell Labs Technical Journal, vol. 1, no. 2, pp. 3-6, Autumn 1996.

[13] R. Jordan and C. T. Abdallah, "Wireless communications and networking: an overview," IEEE Antennas and Propagation Magazine, vol. 44, no. 1, pp. 185-193, Feb. 2002.

[14] Jonathan Donner (2008) Research Approaches to Mobile Use in the Developing World: A Review of the Literature, The Information Society, 24:3, 140-159, 2008.

[15] A. C. Chen, "Overview of code division multiple access technology for wireless communications," IECON '98. Proceedings of the 24th Annual

Conference of the IEEE Industrial Electronics Society (Cat. No.98CH36200), Aachen, Germany, pp. T15-T24 vol.1, 1998.

[16] B. Sarikaya, "Packet mode in wireless networks: overview of transition to third generation," in IEEE Communications Magazine, vol. 38, no. 9, pp. 164-172, Sept. 2000.

[17] C. Prehofer, W. Kellerer, R. Hirschfeld, H. Berndt and K. Kawamura, "An architecture supporting adaptation and evolution in fourth generation mobile communication systems," Journal of Communications and Networks, vol. 4, no. 4, pp. 336-343, Dec. 2002.

[18] U. Nanda and S. K. Pattnaik, "Universal Asynchronous Receiver and Transmitter (UART)," 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, pp. 1-5, 2016.

[19] J. Mesquita, D. Guimarães, C. Pereira, F. Santos and L. Almeida, "Assessing the ESP8266 WiFi module for the Internet of Things," 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, pp. 784-791, 2018.

[20] https://www.generationrobots.com/media/ESP8285_datasheet.pdf

[21] D. K. Halim, T. C. Ming, N. M. Song and D. Hartono, "Arduino-based IDE for Embedded Multi-processor System-on-Chip," 2019 5th International Conference on New Media Studies (CONMEDIA), Bali, Indonesia, pp. 135-138, 2019.

[22] https://www.arduino.cc/en/guide/introduction

[23] https://create.arduino.cc/projecthub/hami/programming-atmega8-using-arduino-ide-90c2ad

[24] https://arduinogetstarted.com/tutorials/arduino-send-email

[25] https://nancyfriedman.typepad.com/away_with_words/2014/04/how-arduino-got-its-name.html

[26] https://www.arduino.cc/en/reference/ethernet

[27] P. Park, "Traffic Generation Rate Control of Wireless Sensor and Actuator Networks," IEEE Communications Letters, vol. 19, no. 5, pp. 827-830, May 2015.

[28] K. Bur, P. Omiyi and Y. Yang, "Wireless sensor and actuator networks: Enabling the nervous system of the active aircraft," IEEE Communications Magazine, vol. 48, no. 7, pp. 118-125, July 2010.

[29] A. Z. Alkar, J. Roach and D. Baysal, "IP based home automation system," IEEE Transactions on Consumer Electronics, vol. 56, no. 4, pp. 2201-2207, November 2010.

[30] Divya Pandey, Vandana Kushwaha, “An exploratory study of congestion control techniques in Wireless Sensor Networks,” Computer Communications, Volume 157, pp. 257-283, 2020.

[31] B. Zaghari et al., "High-Temperature Self-Powered Sensing System for a Smart Bearing in an Aircraft Jet Engine," IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 9, pp. 6165-6174, Sept. 2020.

[32] V. J. Hodge, S. O'Keefe, M. Weeks and A. Moulds, "Wireless Sensor Networks for Condition Monitoring in the Railway Industry: A Survey," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 3, pp. 1088-1106, June 2015.

[33] I. Jawhar, N. Mohamed, J. Al-Jaroodi and S. Zhang, "An Architecture for Using Autonomous Underwater Vehicles in Wireless Sensor Networks for Underwater Pipeline Monitoring," IEEE Transactions on Industrial Informatics, vol. 15, no. 3, pp. 1329-1340, March 2019.

[34] W. Contreras and S. Ziavras, "Low-Cost, Efficient Output-Only Infrastructure Damage Detection With Wireless Sensor Networks," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 3, pp. 1003-1012, March 2020.

[35] C. S. Abella et al., "Autonomous Energy-Efficient Wireless Sensor Network Platform for Home/Office Automation," IEEE Sensors Journal, vol. 19, no. 9, pp. 3501-3512, 1 May1, 2019.

[36] J. A. Stankovic, "Wireless Sensor Networks," in Computer, vol. 41, no. 10, pp. 92-95, Oct. 2008.

[37] M. Tubaishat and S. Madria, "Sensor networks: an overview," IEEE Potentials, vol. 22, no. 2, pp. 20-23, April-May 2003.

[38] H. Gürüler, "The design and implementation of a GSM based user-machine interacted refrigerator," 2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA), Madrid, pp. 1-5, 2015.

[39] A. Soylemezoglu, M. J. Zawodniok and S. Jagannathan, "RFID-Based Smart Freezer," IEEE Transactions on Industrial Electronics, vol. 56, no. 7, pp. 2347-2356, July 2009.

[40] A. Hachani, I. Barouni, Z. Ben Said and L. Amamou, "RFID Based Smart Fridge," 2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Larnaca, pp. 1-4, 2016.

[41] M. Edward, K. Karyono and H. Meidia, "Smart fridge design using NodeMCU and home server based on Raspberry Pi 3," 2017 4th International Conference on New Media Studies (CONMEDIA), Yogyakarta, pp. 148-151, 2017.

[42] G. Song, Z. Wei, W. Zhang and A. Song, "Design of a Networked Monitoring System for Home Automation," IEEE Transactions on Consumer Electronics, vol. 53, no. 3, pp. 933-937, Aug. 2007.

[43] S. Misra, S. K. Roy, A. Roy, M. S. Obaidat and A. Jha, "MEGAN: Multipurpose Energy-Efficient, Adaptable, and Low-Cost Wireless Sensor Node for the Internet of Things," IEEE Systems Journal, vol. 14, no. 1, pp. 144-151, March 2020.

[44] M. Dunbar, "Plug-and-play sensors in wireless networks," IEEE Instrumentation & Measurement Magazine, vol. 4, no. 1, pp. 19-23, March 2001.

[45] J. Lu, H. Okada, T. Itoh, T. Harada and R. Maeda, "Toward the World Smallest Wireless Sensor Nodes With Ultralow Power Consumption," IEEE Sensors Journal, vol. 14, no. 6, pp. 2035-2041, June 2014.

[46] W. A. Jabbar et al., "Design and Fabrication of Smart Home With Internet of Things Enabled Automation System," IEEE Access, vol. 7, pp. 144059-144074, 2019.

[47] R. Kishore Kodali, S. C. Rajanarayanan, L. Boppana, S. Sharma and A. Kumar, "Low Cost Smart Home Automation System using Smart Phone," 2019 IEEE R10 Humanitarian Technology Conference (R10-HTC)(47129), Depok, West Java, Indonesia, pp. 120-125, 2019.

[48] L. Shkurti, X. Bajrami, E. Canhasi, B. Limani, S. Krrabaj and A. Hulaj, "Development of ambient environmental monitoring system through wireless sensor network (WSN) using NodeMCU and “WSN

monitoring”," 2017 6th Mediterranean Conference on Embedded Computing (MECO), Bar, pp. 1-5, 2017.

[49] H. K. Singh, S. Verma, S. Pal and K. Pandey, "A step towards Home Automation using IOT," 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, pp. 1-5, 2019.

[50] A. Škraba, A. Koložvari, D. Kofjač, R. Stojanović, V. Stanovov and E. Semenkin, "Streaming pulse data to the cloud with bluetooth LE or NODEMCU ESP8266," 2016 5th Mediterranean Conference on Embedded Computing (MECO), Bar, pp. 428-431, 20116.

[51] G. Suprianto and Wirawan, "Implementation of Distributed Consensus Algorithms for Wireless Sensor Network Using NodeMCU ESP8266," 2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS), Batu, East Java, Indonesia, pp. 192-196, 2018.

[52] C. Severance, "Massimo Banzi: Building Arduino," in Computer, vol. 47, no. 1, pp. 11-12, Jan. 2014.

[53] P. Visconti, R. de Fazio, P. Costantini, S. Miccoli and D. Cafagna, "Innovative complete solution for health safety of children unintentionally forgotten in a car: a smart Arduino-based system with user app for remote control," in IET Science, Measurement & Technology, vol. 14, no. 6, pp. 665-675, 2020.

[54] SM Kim, Y Choi, J Suh, “Applications of the Open-Source Hardware Arduino Platform in the Mining Industry: A Review,” Applied Sciences, 2020.

[55] S H Pratama, A Rahmadhani, A Bramana, P Oktivasari, N Handayani, F Haryanto, Suprijadi and S N Khotimah, “The development of Arduino-based low-cost wireless modular device for brainwave acquisition,” Journal of Physics: Conference Series, Volume 1248, 18th Asia-Oceania

Congress of Medical Physics (AOCMP) & 16th South-East Asia Congress of Medical Physics (SEACOMP) Connexion Conference & Events Centre, Bangsar South, Kuala Lumpur, 11–14 November 2018.

[56] D. de Santana Nunes, J. L. V. de Brito and G. N. Doz, "A low-cost data acquisition system for dynamic structural identification," IEEE Instrumentation & Measurement Magazine, vol. 22, no. 5, pp. 64-72, Oct. 2019.

[57] C. Rajan, B. Megala, A. Nandhini, C. Rasi Priya, "A Review: Comparative Analysis of Arduino Micro Controllers in Robotic Car," International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering Vol:9, No:2, 2015.

[58] J. Riedemann et al., "Design and Building of an Automatic Alternator Synchronizer Based on Open-Hardware Arduino Platform," IEEE Access, vol. 7, pp. 105116-105122, 2019.

[59] C. Galindo and J. -A. Fernández-Madrigal, "Grounding Concepts and Methods of Real-Time Scheduling in Reality Using Arduino," IEEE Transactions on Education, vol. 63, no. 3, pp. 224-231, Aug. 2020.

[60] S. U. Jan, Y. Lee, J. Shin and I. Koo, "Sensor Fault Classification Based on Support Vector Machine and Statistical Time-Domain Features," IEEE Access, vol. 5, pp. 8682-8690, 2017.

[61] Zulkifli, Shamsul Aizam. *Matlab-simulink Controller Design for Arduino Target On AC Motor Control Application*. UM Power Energy Dedicated Advanced Centre (UMPEDAC), 2014.

[62] M. Fannakh, M. L. Elhafyani and S. Zouggar, "Hardware implementation of the fuzzy logic MPPT in an Arduino card using a Simulink support package for PV application," IET Renewable Power Generation, vol. 13, no. 3, pp. 510-518, 25 2 2019.

[63] N. Sasidharan and J. G. Singh, "A Novel Single-Stage Single-Phase Reconfigurable Inverter Topology for a Solar Powered Hybrid AC/DC Home," IEEE Transactions on Industrial Electronics, vol. 64, no. 4, pp. 2820-2828, April 2017.

[64] K. Maswadi, N. B. A. Ghani and S. B. Hamid, "Systematic Literature Review of Smart Home Monitoring Technologies Based on IoT for the Elderly," IEEE Access, vol. 8, pp. 92244-92261, 2020.

[65] M. R. Alam, M. B. I. Reaz and M. A. M. Ali, "A Review of Smart Homes—Past, Present, and Future," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 6, pp. 1190-1203, Nov. 2012.

[66] G. J. Lacey and D. Rodriguez-Losada, "The Evolution of Guido," IEEE Robotics & Automation Magazine, vol. 15, no. 4, pp. 75-83, Dec. 2008.

[67] D. Gota, A. Puscasiu, A. Fanca, L. Miclea and H. Valean, "Smart home automation system using Arduino microcontrollers," 2020 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, pp. 1-7, 2020.

[68] P. U. Okorie, A. Abdu Ibraim and D. Auwal, "Design and Implementation of an Arduino Based Smart Home," 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, pp. 1-6, 2020.

[69] L. Liao et al., "Design and Validation of a Multifunctional Android-Based Smart Home Control and Monitoring System," IEEE Access, vol. 7, pp. 163313-163322, 2019.

[70] M. H. Abd Wahab, "IoT-based home automation system for people with disabilities," 2016 5th International Conference on Reliability, Infocom

Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, pp. 51-51, 2016.

[71] M. A. Khalid et al., "Design and development of low-cost voice control smart home device in the South Pacific," Asia-Pacific World Congress on Computer Science and Engineering, Nadi, pp. 1-6, 2014.

[72] D. Boucha, A. Amiri and D. Chogueur, "Controlling electronic devices remotely by voice and brain waves," 2017 International Conference on Mathematics and Information Technology (ICMIT), Adrar, pp. 38-42, 2017.

[73] D. Sunehra and V. Tejaswi, "Implementation of speech based home automation system using Bluetooth and GSM," 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, pp. 807-813, 2016.

[74] N. bt Aripin and M. B. Othman, "Voice control of home appliances using Android," 2014 Electrical Power, Electronics, Communicatons, Control and Informatics Seminar (EECCIS), Malang, pp. 142-146, 2014.

[75] M. Ebrahim Abidi et al., "Development of Voice Control and Home Security for Smart Home Automation," 2018 7th International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, pp. 1-6, 2018.

[76] M. Mahith, D. S. B. Kumar, K. C. Prajwal and M. Dakshayini, "Bluetooth Home Automation," 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT), Bangalore, India, pp. 603-607, 2018.

[77] H. Chu, M. Chien, T. Lin and Z. Zhang, "Design and implementation of an auto-following robot-car system for the elderly," 2016 International

Conference on System Science and Engineering (ICSSE), Puli, pp. 1-4, 2016.

[78] M. Nafea, A. B. Hisham, N. A. Abdul-Kadir and F. K. Che Harun, "Brainwave-Controlled System for Smart Home Applications," 2018 2nd International Conference on BioSignal Analysis, Processing and Systems (ICBAPS), Kuching, pp. 75-80, 2018.

[79] S. Gunputh, A. P. Murdan and V. Oree, "Design and implementation of a low-cost Arduino-based smart home system," 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), Guangzhou, pp. 1491-1495, 2017.

[80] V. Hassanpour, S. Rajabi, Z. Shayan, Z. Hafezi and M. M. Arefi, "Low-cost home automation using Arduino and Modbus protocol," 2017 5th International Conference on Control, Instrumentation, and Automation (ICCIA), Shiraz, pp. 284-289, 2017.

[81] C. Nayyar, B. Valarmathi and K. Santhi, "Home security and energy efficient home automation system using arduino," 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, pp. 1217-1221, 2017.

[82] K. Mukendi and M. Adonis, "The development of a remotely controlled home automation system for energy saving," 2017 International Conference on the Domestic Use of Energy (DUE), Cape Town, pp. 265-270, 2017.

[83] J. Fernández, W. Gemin, R. Rivera, M. Revuelta, M. Kuzman and R. Hidalgo, "Digital filter design with Arduino DUE and Matlab," 2015 XVI Workshop on Information Processing and Control (RPIC), Cordoba, pp. 1-6, 2015.