TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechanical Engineering

Design of Communication Layer of Smart Factory

Master Thesis

Liberec 2020

Balaguru Baluchamy

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechanical Engineering

TECHNICAL UNIVERSITY OF LIBEREC

Faculty of Mechanical Engineering

Design of Communication Layer of Smart Factory

Master Thesis

Liberec 2020

Balaguru Baluchamy

Liberec 2020

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechanical Engineering

Design of Communication Layer of Smart Factory

Master Thesis

Study programme: N2301 Mechanical Engineering

Study branch: Manufacturing Systems and Processes

Author: **Balaguru Baluchamy**

Thesis Supervisors: Ing. Radek Votrubec, Ph.D.
Department of Manufacturing Systems and Automation

Liberec 2020

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechanical Engineering

Master Thesis Assignment Form

Design of Communication Layer of Smart Factory

Name and surname: **Balaguru Baluchamy**
Identification number: S18000459
Study programme: N2301 Mechanical Engineering
Study branch: Manufacturing Systems and Processes

Assigning department: Department of Manufacturing Systems and Automation
Academic year: **2019/2020**

Rules for Elaboration:

The aim of the diploma thesis is to create a communication layer protocol between individual components in a smart factory. 1. Meet with all components of the smart factory (milling machine, vehicle, check point and stack), Arduino controllers and wifi communication. 2. Create working diagrams of each Smart Thing component. Describe the communication relationships between the components. 3. Create set of functions to send and receive the data and design a suitable protocol. 4. Test the protocol on real model of factory.

Scope of Graphic Work: as required
Scope of Report: 50
Thesis Form: printed/electronic
Thesis Language: English

List of Specialised Literature:

[1] BEQUETTE, B. Process control: modeling, design, and simulation. Upper Saddle River, N.J.: Prentice Hall PTR, 2003. ISBN 0133536408.
[2] Arduino Learning: Getting Started with Arduino. In: Arduino [online]. 2014 [cit. 2015-01-09]. available from: http://arduino.cc/en/Guide/HomePage

Thesis Supervisors: Ing. Radek Votrubec, Ph.D.
Department of Manufacturing Systems and Automation

Date of Thesis Assignment: November 20, 2019
Date of Thesis Submission: May 20, 2021

L.S.

prof. Dr. Ing. Petr Lenfeld
Dean

Ing. Petr Zelený, Ph.D.
Head of Department

Liberec November 20, 2019

Declaration

I hereby certify, I, myself, have written my master thesis as an original and primary work using the literature listed below and consulting it with my thesis supervisor and my thesis counsellor.

I acknowledge that my bachelor master thesis is fully governed by Act No. 121/2000 Coll., the Copyright Act, in particular Article 60 – School Work.

I acknowledge that the Technical University of Liberec does not infringe my copyrights by using my master thesis for internal purposes of the Tech- nical University of Liberec.

I am aware of my obligation to inform the Technical University of Liberec on having used or granted license to use the results of my master thesis; in such a case the Technical University of Liberec may require reimbursement of the costs incurred for creating the result up to their actual amount.

At the same time, I honestly declare that the text of the printed version of my master thesis is identical with the text of the electronic version up- loaded into the IS/STAG.

I acknowledge that the Technical University of Liberec will make my master thesis public in accordance with paragraph 47b of Act No. 111/1998 Coll., on Higher Education Institutions and on Amendment to Other Acts (the Higher Education Act), as amended.

I am aware of the consequences which may under the Higher Education Act result from a breach of this declaration.

May 27, 2020 Balaguru Baluchamy

ACKNOLDEGEMENT

It is with immense gratitude, I acknowledge the support of my University for providing me this great opportunity to develop in deepest manner my engineering skills while accomplishing this diploma thesis and help them to further develop the importance in machine construction design.

I am indebted firstly to thank especially **Ing. Radek Votrubec, Ph. D** for his professional guidance, encouragement and good advices all along. This thesis becomes much better because of him. Thanks to his supervision and support.

I would like to thank our head of the department **Ing. Petr Zelený Ph.D.** who has been a great support in every way to pursue our academics. Without his help, I might otherwise have never encountered successfully.

Furthermore, would like to thanks, **Ing. Petr Keller, Ph.D.** for providing guidance for the thesis writing.

I would like to express my gratitude to The Department of Manufacturing System and process of Automation at the Technical university of Liberec for providing me the support I required, which gave me the opportunity to proceed freely with this work.

Finally, I am grateful to thank my parents and friends, who have been a moral support in all the situations and encouraging my works.

This work was partly supported by the Student Grant Competition of the Technical University of Liberec under the project Optimization of manufacturing systems, 3D technologies and automation No. SGS-2019-5011.

ABSTRACT

This diploma thesis deals with the design of a communication layer between the individual components of a smart factory. Each component of the Internet of Things has its own control system. In this case, the control units use Arduino development boards. The core of the Arduino is a microprocessor with easily programmable analog and digital inputs and outputs and other buses. All IoT components must communicate, exchange information and cooperate with each other. The control units are equipped with Wi-Fi modules and it is necessary to design program structures for communication. Control units must be able to solve combination and sequential problems based on automaton. Closed graphs of control unit automats are interconnected by means of Wi-Fi messages. The structure of these messages is designed, which must contain component addresses and transmitted information of various types, character, string, numeric variables and arrays. Within the work, a general library of functions containing typical elements of automata was designed. This general library has been implemented on two specific production systems. The first is a model of a smart factory with the production of a colored bracelet, where the individual components are a server, a vehicle and several stacks of beads. The second is production system of the little plate with name, which consists of a plate warehouse, a vehicle and a milling machine. Our universal control unit is designed to be used as a control system for all these components.

KEYWORDS

Control system, Arduino, Internet of Things, Wi-Fi communication, Industry 4.0.

ABSTRAKT

Tato diplomová práce se zabývá návrhem komunikační vrstvy mezi jednotlivými komponenty chytré továrny. Každá komponenta internetu věcí má svůj řídicí systém. V tomto případě řídicí jednotky používají vývojové desky Arduino. Jádro Arduina tvoří mikroprocesor s jednoduše programovatelnými analogovými a digitálními vstupy a výstupy a dalšími sběrnicemi. Všechny IoT komponenty spolu musí komunikovat, vyměňovat si informace a spolupracovat. Řídicí jednotky jsou vybaveny wifi moduly a je nutné navrhnout programové struktury pro komunikaci. Řídicí jednotky musí umět řešit kombinační i sekvenční úlohy na bázi automatů. Uzavřené grafy automatů řídicích jednotek jsou vzájemně propojeny pomocí wifi zpráv. Je navržena struktura těchto zpráv, které musí obsahovat adresy komponent a přenášené informace různých typů, znak, řetězec, číselné proměnné a pole. V rámci práce byla navržena obecná knihovna funkcí obsahující typické prvky automatů. Tato obecná knihovna byla implementována na dva konkrétní výrobní systémy. Prvním je model chytré továrny s výrobou barevného náramku, kde jednotlivé komponenty jsou server, vozík a několik zásobníků korálků. Druhý je výrobní systém na destičky se jménem, který tvoří sklad destiček, vozík a frézka. Naše univerzální řídicí jednotka je navržena tak, aby byla použitelná jako řídicí systém všech těchto component.

KLÍČOVÁ SLOVA

Řídicí systém, Arduino, Internet věcí, Wifi komunikace, Průmysl 4.0.

Table of Contents

LIST OF FIGURES

LIST OF SYMBOLS

AL – Artificial Intelligence

IoT – Internet of things

LAN – Local area network

ROM- Read only memory

DC – Direct current

IR – Infra red

1.INTRODUCTION

Electronics can be installed in devices that are used in factories and can be controlled by the control systems this allowing linked machines and computers to track output in real time. While the manufacturing industry used proprietary technology to link machines and make them smarter as well. The new Internet of Things really is giving this to a huge push. Integrated with intelligence electronics, the smart computer uses computer communication technology. This involves robots, and other cognitive computing devices that can make decisions and solve problems without human intervention. Smart machines make full use of technology such as cloud computing, big data, robotics and the Internet of Things. The main of aim of thesis is to communicate between the two control systems through WI-FI communication technologies. In the world of connected computers, products and people transmitting and receiving the information in real time and making decisions on their own or in a group a smart factory is built. Smart Factory involves structures such as cyber-physical technology, where the operation is managed by a computer that is connected to the Internet and its users. The Internet of Things allows users of cyber physical systems to communicate with each other and human beings in real time.

Figure 1: Communication layer of smart factory [1]

The smart factory control system comes with extremely advanced function helps to learn from the Industry 4.0. The control system which was the part of the smart factory which helps to control the production machines and other equipment's. And this control system of smart factory used to communicate with other smart factory through Wi-Fi technology. Which helps to automate the machines without help of human involvement. Which motivate to use internet of things and smart analytical technology in industries. This project is used to communicate between the guideways and the cutting tool which is used to transmit and receive the message between two systems by using this control system in the milling machine which helps to increase in production and utilization of milling systems in industries and used for communicating to the server vehicle etc. And to transmit and receive the message or command between the several units. Which help to simulate the axis interpolate actions. This project helps to develop the low cost of control systems. It therefore certifies that there will be no deflection from the goal due to human involvement. These control system has been programmed to control the DC motor these programming has been done with help of Arduino software. This control system knowledge with the smart factory. The goal is to communicate between the control systems. where the operation is managed by a computer that is connected to the Internet and its users. The Internet of Things allows users of cyber physical systems to communicate with each other and human beings in real time.

2. REVOLUTION OF INDUSTRY 4.0

It is used to refer the developmental process of manufacturing and chain production management. And this also refers to fourth industrial revolution. It was introduced on 2011 as industry 4.0 by different fields. It is based on current technology period enabled by advanced technology such as artificial intelligence, Machine learning, Internet of Things, Cyber physical systems, Robotics and automation into the strategic plans of manufacturing and factory operations and business procedures.

The application of information and communication are characterized by the industry and technologies. And the main benefits of industry 4.0 is to improve the productivity and efficiency and the knowledge has been increased by sharing and collaborative work. Up’s and down in the smart factory can be easily scaled up. It will automate the compliance including track and trace like quality inspections, data logging and serialization. Better customer experience by improving service that offer to customer. In addition, it has issues with product availability and the product quality will be increased. Need to invest upfront of cost to achieve it. Better use of resources fastest manufacturing less in production line down and resources material and product waste. And operating cost is low.

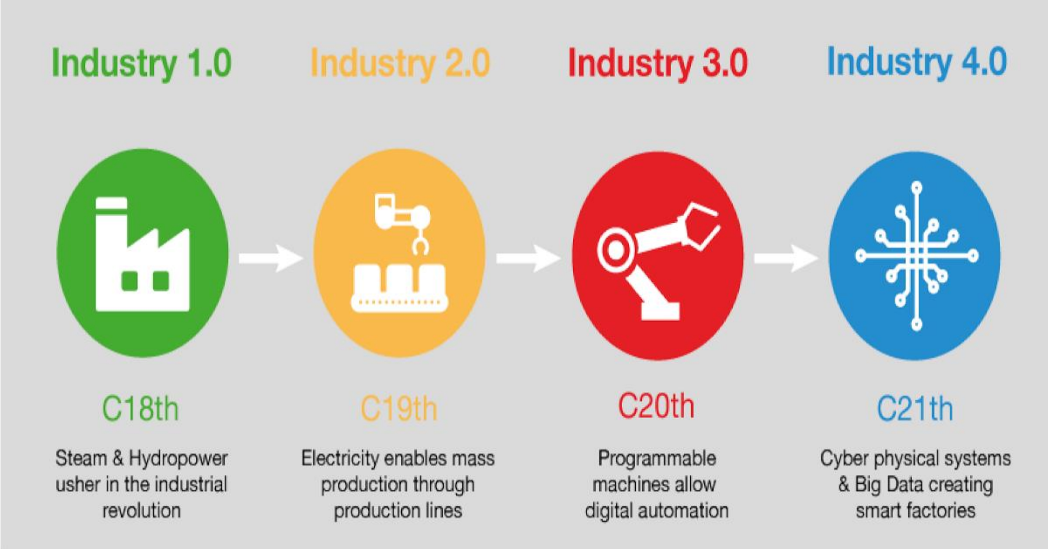

Figure 2: Evaluation of Industry 4.0 [2]

And it gives the greater knowledge about manufacturing process, supply chains, distribution chains, business chains, and business performance. And it creates opportunities to changing the business process. And to developing a new product, optimizing a supply chain. Enable to produce higher quality, higher margin and more innovative products. And it also can be called as smart factories modular structured and cyber physical systems will monitor physical process. And it will create a virtual copy of physical world.

2.1 INDUSTRY 1.0

In the 18th century the first industrial Revolution begins with the introduction of steam power and manufacturing mechanization. What threads on simple spinning wheels were developed before the mechanized version achieved eight times the volume at the same time. Steam control has already been established. The use of industrial purpose has been the greatest advance in rising humanity.

2.2 INDUSTRY 2.0

The second industrial revolution has been started in 19th century by discovering the electricity and the development of assembly lines. Machines were finally equipped with their own power sources, thereby making them more portable. This period has also seen the creation of number of management programs which have allowed manufacturing facilities to be more productive and more competitive.

2.3 INDUSTRY 3.0

The innovation and manufacture of electronic devices, such as transistor and later integrated circuit chips has made it possible to more completely automate individual machines to complement or replace operators within the previous few decades of 20th century. This time also spawned software device creation to capitalize on the electronic hardware. It reduces the pressure cost production of many manufactures to move part and assembly operation to low cost countries.

2.4 INDUSTRY 4.0

Industry 4.0 integrates the internet of things(IOT) with manufacturing process in the 21st century so that device can interpret and use information to direct intelligent behavior. It also combines state of the art innovations like manufacturing additives, robots, artificial intelligence and other cognitive technology, advanced material and enhanced reality. Creation of new technology was a key catalyst of industry 4.0 trend. [2]

2.5 PRINCIPLES OF INDUSTRY 4.0

The definition of industry 4.0 is based on four major principles to ensure that all production process is computerized.

- Interoperability

It mainly applies to machinery and associated component's ability to link and interact across the internet with the people. And it allows to share the information freely between the systems and allows companies to reduce the cost of gathering and maintaining the information and minimize the needless duplication and exploit applications.

- Transparency in information

This theory demands that information systems, by configuring digital data into sensor data, should be able to construct virtual copies of the physical world. To do this the raw sensor data must be aggregated with the background data that is compatible. This involves the aggregation of raw sensor data into higher value context information.

- Technical assistance

This concerns the system's ability to help human through extensive integration and knowledge visualization for better decision making and fast problem solving. Technical assistance also focuses on the capacity of cyber enabled systems to support the human resources physically by performing different activities that are considered time consuming, dangerous stressful to humans.

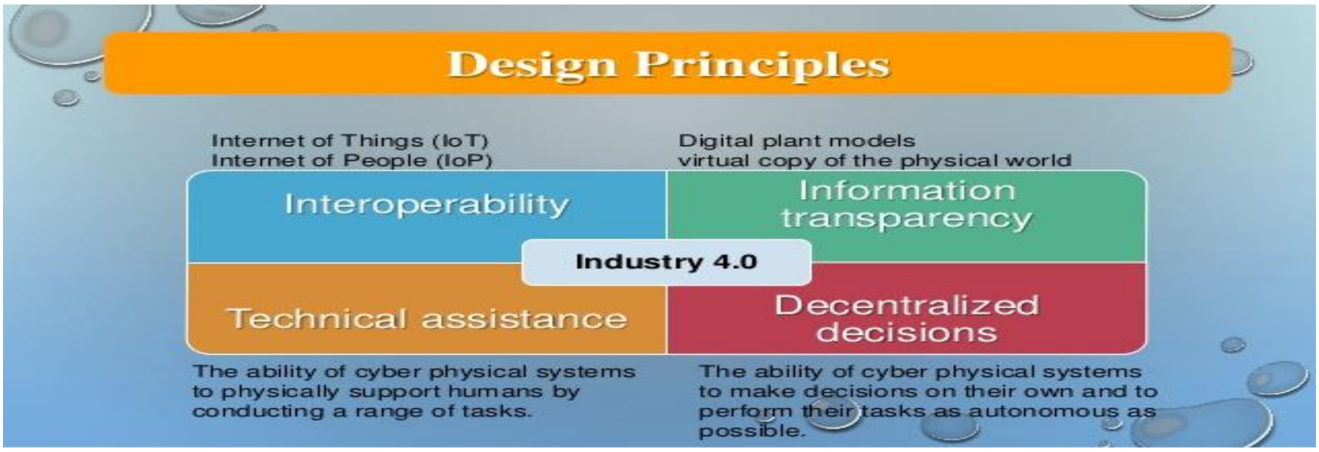

Figure 3: Principles of industry 4.0 [3]

- Decentralization of decisions

This theory refers to the ability of cyber enabled systems to make decisions and perform their dedicated function's independently. This can only be modified in the event of interferences or problems with the intended goals, which may entirely be handling the other activities at certain levels. The customization of products is characterized by the flexible manufacturing. [3]

3. SMART INDUSTRY

Smart industry stands for revolutionary digitalization, the relation between goods, machines and people and the use of modern technology in production. This offers of plenty of opportunities for current and new business in all industries as well as being good for competitiveness worldwide. Production optimization through the application of ICT (information and communication technology) and modern manufacturing technology such as 3D printing makes manufacturing more efficient, cheaper and improve quality. Smarter machines, robots and other components of the manufacturing process interact with each other and this optimizes their interaction not only within companies but also different companies and for this process.

Smart industry allows for the development of new business from board and diverse knowledge sources on the basis of new affective, technologies such as big data processing, internet of things (IOT), next generation autonomous robotics, 3D printing, nanotechnology and miniaturization as well as new sensor technology. It has strong industrial base of large corporations and SMEs. And it has an outstanding ICT network and therefore the development of regional field in which business, educational institutions and information institutions work together. [4]

4. SMART FACTORY

A smart factory is a manufacturing facility that is highly digitized and connected and that relies on smart manufacturing. The idea of the smart factory, thought to be the so-called factory of the future and still in its infancy, is considered a significant consequence of the fourth industrial revolution, or Industry 4.0. And it was used by manufacturing firms, a smart factory operates by using technologies such as artificial intelligence (AI), robotics, analytics, big data and the Internet of Things (IoT) and can operate completely autonomously with self-correcting capabilities. Through this explanation that smart factories go beyond pure automation.

The smart factory is a scalable device capable of self-optimizing output across a wider network, adapting to and learning from new conditions in real or near-real time, and running entire production processes autonomously. It is clear from this description that smart factories are going beyond simple automation.

It becomes evident through this explanation that smart factories go beyond pure automation. The smart factory is a scalable device capable of self-optimizing output across a wider network, adapting to and learning from new conditions in real or near-real time, and running entire production processes autonomously. [5]

Figure 4: Flow of smart industry [5]

4.1 Communication layer of smart factory

The smart factory vision success largely depends on underlying communication technologies hitting necessary performance levels. If the communication infrastructure is unable to meet the demanding requirements many applications will not operate as intended. A lot of ongoing efforts are currently being made to close the remaining gaps with new, creative advancements in data processing technologies. Communications has still many hurdles to address but from today perspective all the requisite solution can be given to make the smart factory dream a reality.

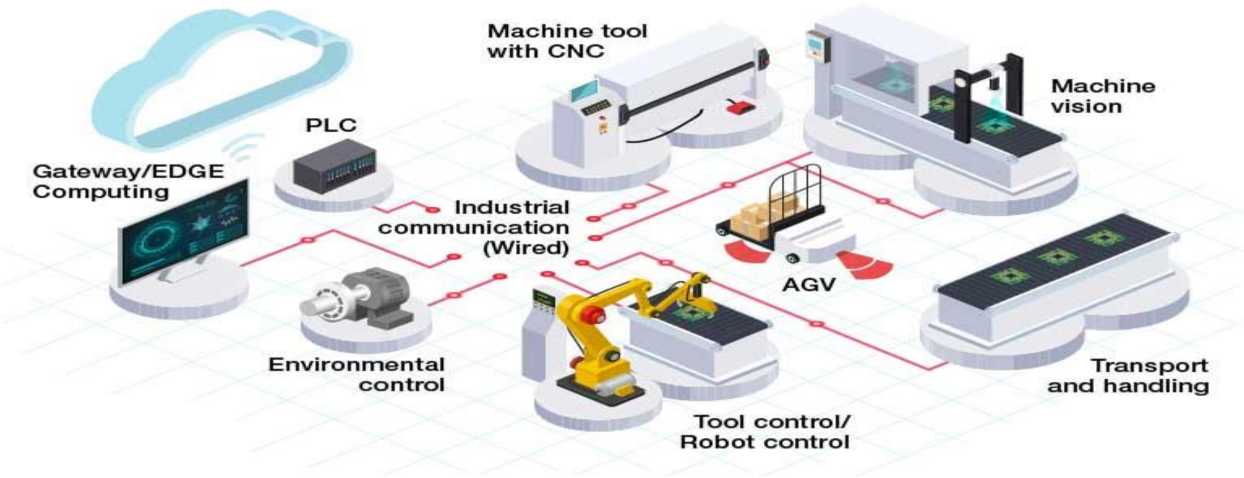

Figure 5: Communication layer of smart factory [6]

The following parts define the communication criteria and solution within a manufacturing site. [6]

4.1.1 Network Topology

The large number of connected device will obviously be higher in smart factory LAN than it is today. This will impact the topology of the network and the way devices are connected. Additional systems require more cabling, more installation time and more commissioning costs and additional attention to ongoing operations and maintenance. A large number of devices should be networked in hierarchy to simply *network* management and operation.

4.1.2 Data Rates

Data speeds on wired Ethernet continues to increases. New chip designs also implement gigabit Ethernet interfaces thereby reducing the cost for quicker connection. [7]

4.2 Cyber-physical manufacturing system

Cyber-physical systems have recently gained growing attention for their immense potential for smart systems for next generation that incorporate the cyber technology into physical process. Cyber-physical system does not implement either smart factories or smart manufacturing. The smart factory was first studied with the implementation of the internet of things(Iot). A systematic and through research on the linkages and relationships between these concepts is currently lacking. [8]

Integration platform for cyber-physical manufacturing system

- Autonomous system
- Medical monitoring system
- Industrial control system
- Robotic system

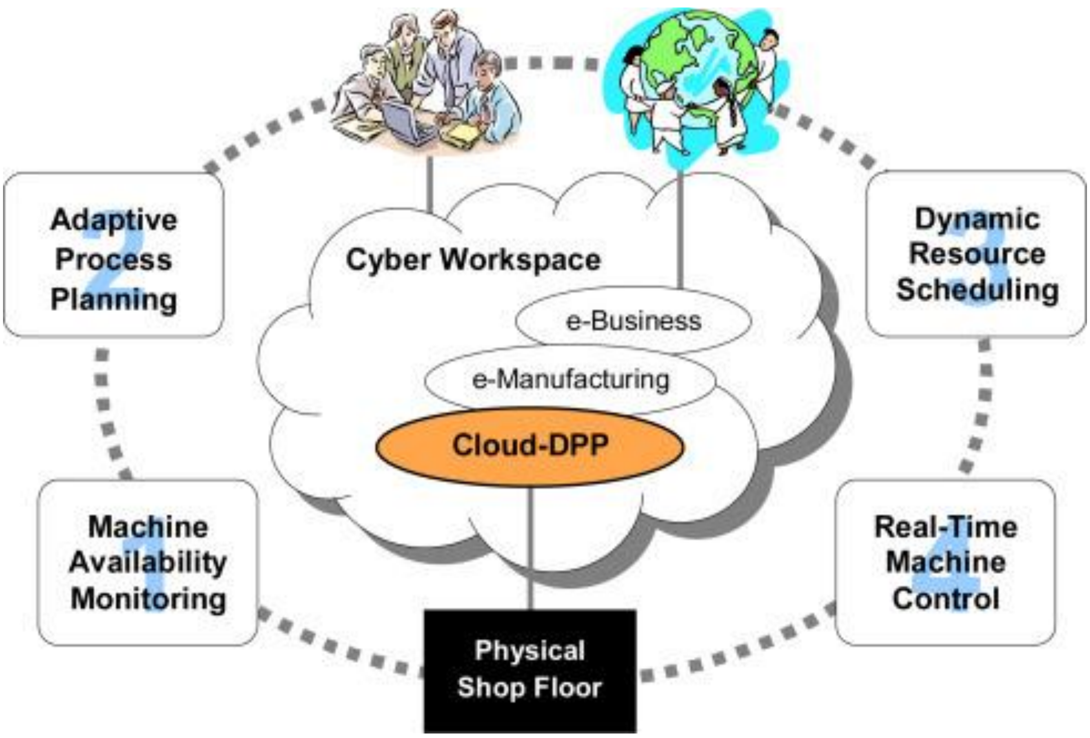

Figure 6: Flow work of cyber workspace [8]

4.3 Architecture of cyber-physical systems

Because of the complexity of the industries the design of the intelligent factory must follow a number of requirements and consider the applicability in different sectors and manufacturing process. In the field of cyber-physical production process, there are variety of works dealing with structure and standardization problems. It describes the method of realization through five-level of architecture referred to as five concepts. The model proposed is based on a standardized model which extends with new features. The core of work will describe the implementation in industry 4.0 of intelligent manufacturing systems, and will identify the modern elements required for the life of the units. This proposes a categorical and hierarchical structure in which the definition of INDUSTRY 4.0 is defined and achievable through the continuous and incremental development process. [9]

The main criteria of which the intelligent manufacturing system is highly automated at the manufacturing level are automation and intelligence and are self-respecting, self-optimizing, self-configuring. It is obvious that the autonomous intelligent manufacturing subsystems and their relationships are too complex for real-time control of human operators. The solution is to monitor the development cycle using neural network technology using an intelligent program with artificial intelligence. [10]

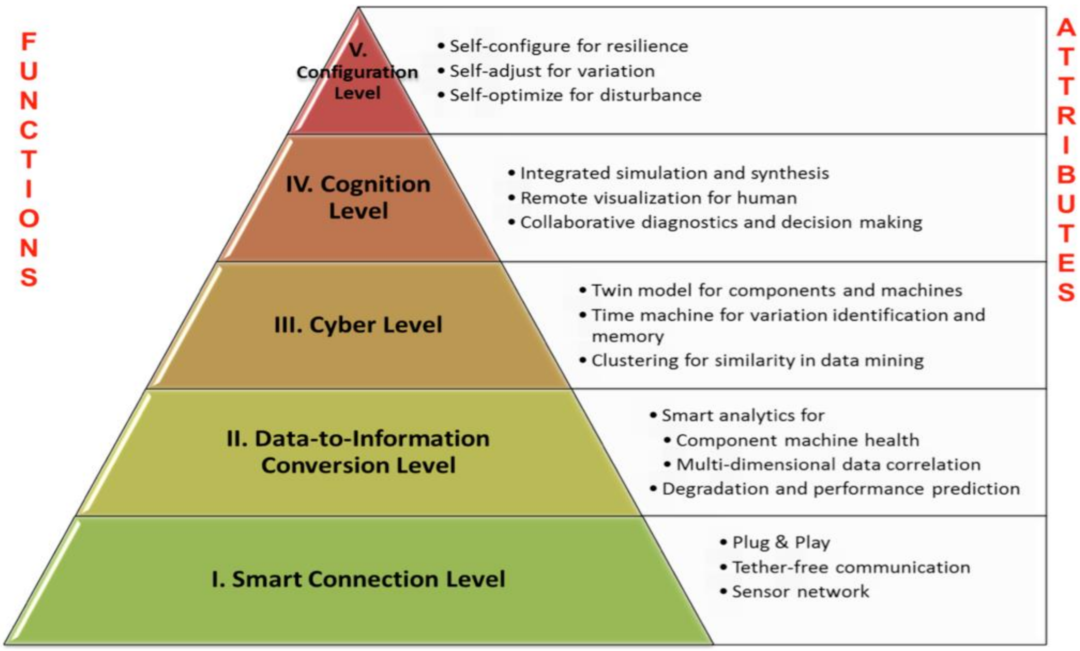

Figure 7: Architecture of cyber physical system [11]

In the pyramid model it assumed the architecture of the manufacturing enterprises realized in keeping with the definition of INDUSTRY 4.0 defined by layers where the lover level consists of smart sensors and regulators interconnected by IOT. The method of obtaining correct and reliable data from devices and their components represents the first step in cyber-physic implementation.

In the production data cycle data can be collected directly by sensors or obtained from management or corporate production system (EPR, MES). Since these are data representing a variety of variables of different types (time-dependent / independent), it is necessary to choose an appropriate method of measuring the measured data and to choose the correct recording form. Due to the ever-increasing use of sensors and machine in the network a large volume of data is generated continuously. [11]

5. Embedded system

An embedded device is a mixture of computer hardware and software, either fixed in capability or programmable, built within a larger system for a particular purpose or functionality. Industrial machinery, farm and production machinery, cars, medical equipment, cameras, household appliances, aircraft and toys, and mobile devices. Embedded systems are computer systems, but they can vary from having no user interface (UI) for instance, on devices where the system is programmed to perform a single function to complex graphical user interfaces (GUIs), such as in mobile devices. User interfaces can include buttons, LEDs, sensing touchscreens and more. Some systems are also utilizing remote user interfaces. [12]

5.1 Characteristics of Embedded systems

1. Single functioned - An embedded system normally performs a specialized operation and does the same rapidly
2. Tightly constrained - All computer systems have design metrics constraints but those on an embedded system can be especially tight.
3. Reactive and real time - Many embedded systems must respond continuously to changes in the environment of the device and must measure those results without delay in real time.
4. Microprocessor Based - It must be based on microprocessor or microcontroller.
5. Memory - It must have a memory, because it normally embeds its program in ROM. No secondary memories are required at the machine.
6. Connected - It has to have peripherals attached to link devices for input and output.

7. HW-SW systems - More functionality and versatility are used with the application. Hardware is used for protection and performance.
8. The advantage of embedded system is easily customizable and power consumption is very low, enhanced performance and lower in cost. The major disadvantage was development efforts are high and it takes larger time to build. [13]

6. ARDUINO – MICROCONTROLLER BOARD

Arduino is an open source hardware and software based electronic platform. Arduino boards will read the light inputs on sensor, a finger on a button and transform it into an output that activates the motor, and turns on LED. That we can instruct the microcontroller board that what to do. There are plenty of other microcontrollers and systems for physical computing and many others provide similar features. And both of these devices take messy specifies of programming microcontroller and package is an easy to use. Arduino is also used simplest way to operate the microcontroller board. Arduino is also known as a single board microcontroller to make the use of electronics more accessible in thousands of projects. A microcontroller is a small computer on a single integrated circuit that contains a processor core, memory and input/output computing.

Compared to other microcontroller platform Arduino boards relatively inexpensive. That we can assemble least costly version of the Arduino module by hand and even the preassembled Arduino models cost less than € 46. The Arduino concept is that we can directly load the programs into the computer without a hardware programmer having to burn the software.

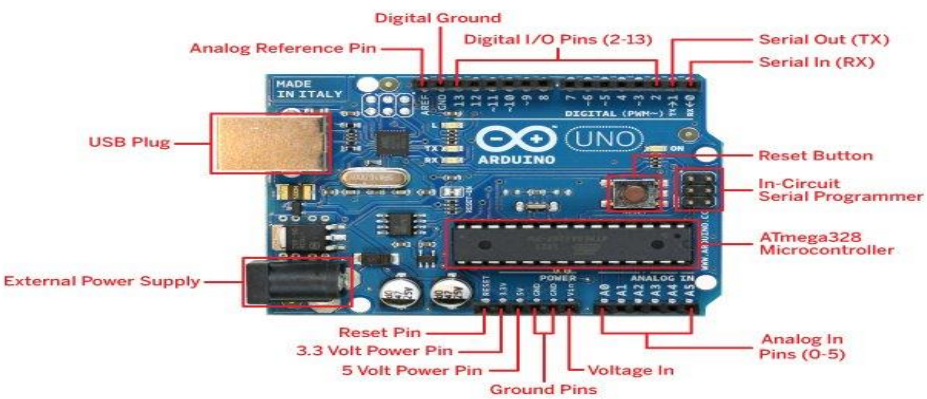

Figure 8: Arduino Microcontroller Board [14]

The Arduino software programming are open source tool which are available for extension. The language can be expended C++ libraries and those who want to understand the technical details can make the leap from Arduino to the AVR programming language on which it based. Arduino software runs on Windows, Macintosh OSX and Linux. Most of the microcontroller system runs on Windows only. [14]

6.1 Arduino Mega 2560

The Arduino Mega 2560 is a board based on the AT mega 2560 microcontroller. It has 54 digital input/output pins of which 14 can be used PWM outputs, 16 analog inputs, 4 UARTS (serial hardware ports) a 16 MHz crystal oscillators a USB interface, a power jack, an ICSP header, and a reset key. It contains everything necessary to support the microcontroller it has a simple connection. The Arduino mega can be powered with an external power supply or with a USB link. Power sources is automatically selected. External power can from either an AC to DC convertor or a battery that can attach an adapter by plugging a center positive 2.1 mm plug into power jack on the board. Leads from the battery can be inserted into the ground(Gnd) and Vin pin headers.

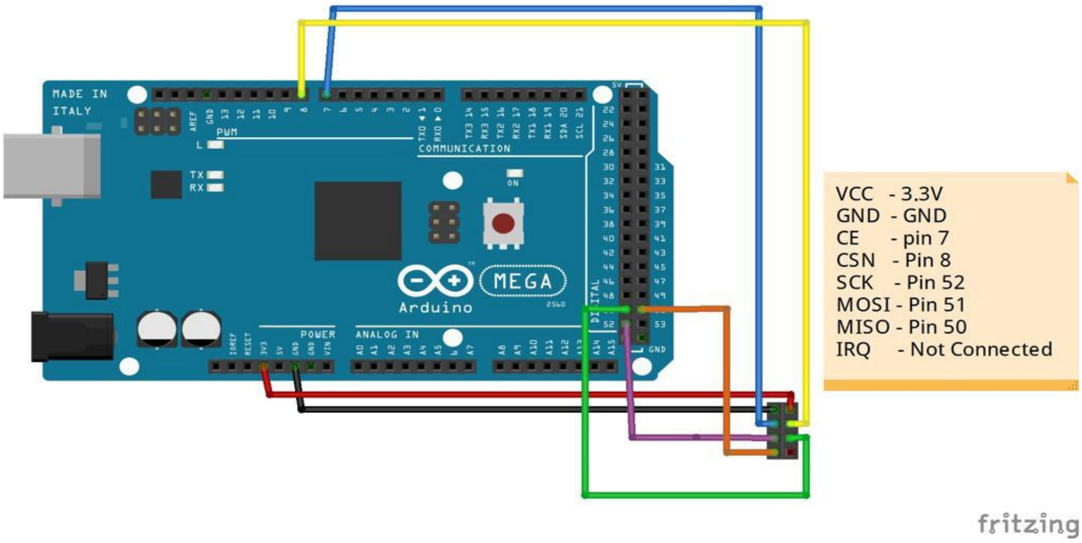

Figure 9: Arduino mega 2560 [15]

The board will work under on a 6 to 20-Volt external supply. However, if supplied with less than 7V, and the 5V pin will provide less than five volts and the board can be unstable. If the voltage regulator uses more than 12V it can be overheated and harm device. The suggested range between 7 and 12 Volts. The Arduino mega 2560 has 256 KB of flash memory to store code including 8 KB for boot loader 8KB of SRAM

(Static random access memory) and 4KB of EEPROM (Electrically erasable programmable Read-only memory). The Arduino Mega 2560 has many communication facilities with a computer and any other Arduinos and other microcontrollers. The Arduino programming contains a serial monitor allowing to spend simple textual data to and from the device. [15]

6.1.1 Specification of Arduino Mega 2560

- ATmega2560 : Microcontroller
- Voltage to operate : 5V
- Recommended Input Voltage : 7-12V
- Voltage of input (limit) : 6-20V
- Digital I / O pins : 54 (15 of which give output from PWM)
- Analog pins to input : 16
- Current DC per Pin I/O : 20 mA
- DC Current for Pin for 3.3v : 50 mA
- Flash Memory : 256 KB of which bootloader used 8 KB
- SRAM : 8KB
- EEPROM : 4KB
- Clock Speed : 16MHz
- LED_BULTIN : 13
- Length : 10.15cm
- Width : 5.33cm
- Weight : 37g

6.1.2 Arduino Mega pin configuration

Every pin of this Arduino super 2560 board is accompanied by a specific feature. Many of this Board's analog pins can be used as electric I / O pins. The Arduino super projected can be planned by using this device. These boards offer flexible work memory space which is the more & processing power which allows to work without delay with different types of sensors.

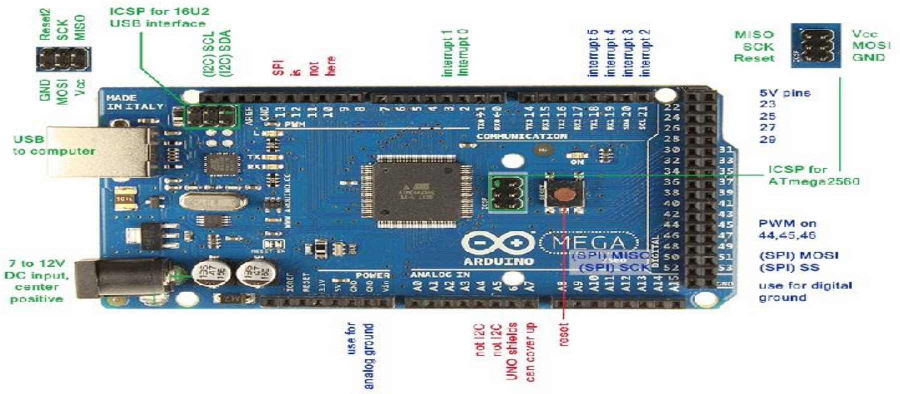

Figure 10: Pin configuration of Arduino mega 2560 [16]

These pins are used for supplying approximately 5V of regulated voltage O/P. This RPS (regulated power supply) provides the power supply over the Arduino mega board to the microcontroller furthermore on other components used. Otherwise USB cable is being obtained from the board's Vin-pin or another regulated voltage supply-5V, while 3.3V0-pin is to be offered for an additional voltage regulation. The Arduino mega board includes 5-GND pins where one in every of those pins where used depend upon the requirements.

The Reset pin of this board are often used for rearranging the board. The board are often rearranged by setting this pin to low. The rage of supplied input voltage to the board ranges from 7volts to 20volts. The voltage provided by the facility jack are often accessed through this pin to board are going to be automatically founded to 5V.

This board's serial pins such as TXD and RXD are used to transmit and receive the serial data. TX indicates the transfer of information while the RX indicates receiving data. That board's serial pins have four combinations. For serial 0 it includes TX (1) and Rx (0), for serial 1 it includes TX (18) & Rx (19), for serial 2 it includes the TX (16) & Rx (17) and finally for serial 3 it includes TX (18) and Rx (17).

6.1.3 Analog pins and external interrupts

The external interrupts will be formed by using 6-pin like interrupt 0(0), interrupt 1(3), interrupt 2(21), interrupt 3(20), interrupt 4(19) interrupt 5(18). These pins produce interrupts by multiple ways this is providing low value, rising or falling edge or changing the value to the interrupt pins. This Arduino board includes an LED linked to pin-13 called digital pin 13. This LED is operable depending on the pin's high and low values. This will give the real-time modification of the programming skills.

The Analog reference voltage (AREF) is a reference voltage for analog inputs. The board contains 16-analog pins which are labelled as A0-A15. Knowing that all the analog pins on this board can be used like digital I / O pins, is very important. Each analog pin can be accessed with the 10-bit resolution that can gauge from GND to 5 volts. However, the higher value can be altered using the AREF pin. The term SPI is a serial peripheral interface for the transmission of data between the controller and other components. The I2C communication can be supported by two pins where those pins stand for Serial Data Line (SDA) which is used to hold the data and other pin stands for Serial Clock Line (SCL) which is mostly used to offer data synchronization between devices.

6.1.4 Dimensions of Arduino Mega 2560

The Arduino Mega 2560 board dimensions include mostly the length as well as widths such as 101.6 mm or 4 inches X 53.34 mm or 2.1 inches. It is comparatively superior to other board types which are available on the marketplace. But, from the specified measurements the power jack and USB port are somewhat extended. Most of the guards used in other Arduino boards are well suited for Arduino boards. Confirm the guard's operating voltage is well suited to the board voltage planed before to use a guard. Most guards would have operating voltage of 3.3V otherwise 5V. Still, the board may be damaged by guards with high operating voltage. Additionally, the shield's distribution header will vibrate with the Arduino board's distribution button. That can simply connect the shield to the Arduino board and make it in a running state.

6.1.5 Programming for Arduino mega 2560

An Arduino Mega 2560 programming can be achieved with the aid of an IDE (Arduino Software), and it follows the language of C-programming. Here the code in the software that is burned inside the software and then transferred through a USB cable to the Arduino board. An Arduino mega board contains a boot loader that eliminates the use of an external burner to burn the program code into the Arduino board. A STK500 protocol can be used to contact the boot loader here. While compiling and burning the Arduino program, we can remove the power supply from the Arduino board by detaching the USB cable.

Whenever we plan to use the Arduino board for our project, the board's power supply can be supplied through a power jack otherwise Vin button. Another feature of this is multitasking wherever a mega board from Arduino comes handy. However, Arduino IDE Software does not allow multi-tasking, but for this purpose one can use additional operating systems to write C-program. With the help of an ISP connector this is flexible to use in your personal custom build program.

The Arduino mega 2560 datasheet is all about that. It is a replacement for older Arduino Mega board. It is usually not used for general projects because of the number of pins, but we can discover them in complex projects such as temperature sensing, 3D printers, IOT applications, radon detectors, real-time data applications monitoring, etc. This board's power supply can be connected to a PC through a USB cable or a battery or an AC-DC adapter [16]

6.2 Printed Circuit Board

A printed circuit board (PCB) mechanically supports and connects electrical or electronic components by means of conductive tracks, pads and other features etched from one or more copper layers laminated on and within sheet layers of a non-conductive substratum. Components are usually soldered onto the PCB to bind them electrically and fasten them mechanically to it. Printed circuit boards are used in all electronic goods, except the simplest. Some electrical devices, such as passive switch boxes. Wire wrap and point-to-point construction are alternatives to PCBs, both once common but now rarely used. PCBs require additional design effort to lay out the circuit but it can be automated to manufacture and assemble.

PCBs can be single-sided (one layer of copper), double-sided (two layers of copper on both sides of one layer of substratum), or multi-layered (outer and inner layers of copper alternating with substratum). Multi-layer PCBs allow a much higher component density, because circuit traces on the inner layers would otherwise take up surface space between components. The increase in popularity of multilayer PCBs with more than two, and particularly more than four copper planes was simultaneous with the adoption of surface mounting technology. Chemical etching divides the copper into separate conducting lines called tracks or circuit traces, contact pads, through for passing connections between copper layers and features such as strong conductive areas for electromagnetic shielding or other purposes. [17]

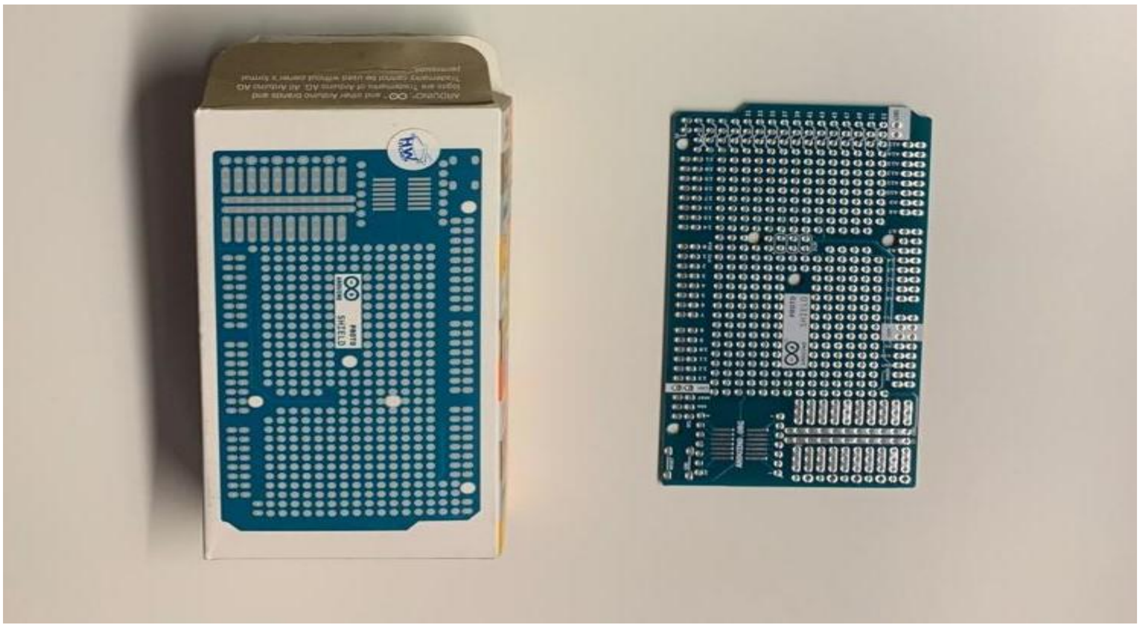

Figure 11: Blank PCB Board [own]

7. BLYNK APPLICATION

Blynk is a forum for controlling Arduino, Raspberry Pi and the likes of IOS and Android devices over the Internet. It's a digital dashboard where we can simply drag and drop widgets to create a visual interface for our project. Blynk deals with hundreds of hardware models and various types of connections. Choose sort of Hardware then select type of connection. From electronic components to development plans and data plans, the Blynk ecosystem can cover anything. [18]

7.1 Blynk Architecture

The following components are on the Blynk platform;

7.1.1 Blynk app builder

Allows to create apps with different widgets for projects. It is available for platforms running Android and iOS.

7.1.2 Blynk Server

Responsible for all communications between the Blynk app and the hardware running on mobile device. That can either use the Blynk Cloud, or run our Blynk server locally. It's open source, can handle thousands of devices quickly, and can even run on a Raspberry Pi.

7.1.3 Blynk libraries

Enables contact with the server and processes both incoming and out coming Blynk device and hardware commands. They exist for all the common hardware platforms.

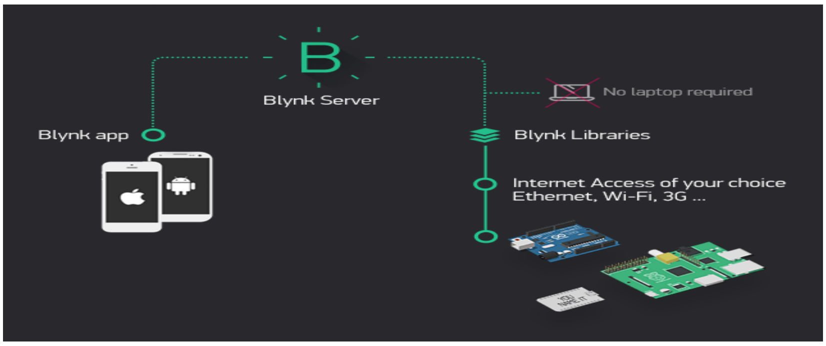

Figure 12: Blynk Application [18]

8. CONTROL SYSTEMS

In most industries there is this concern about reducing the cost of production and rising productivity along with ensuring the health of its workers. Control system based smart factory automation has been proposed and which will not only improve industry performance, but also ensure workers health. And as it could be regulated with anywhere in the world. A fundamental model of automation system in the industrial domain is a simple control system, which involves an input or sensor, a controller, and an output or an actuator. This model will help to develop any industrial application with the correct choice of hardware and software. A process of repeated sequential switching replicates a typical production line. Sequential practices are commonly used in the packaging, processing and related industries

With programming in C or C++ languages, microprocessors such as the Arduino mega 2560 can make a number of logical control decisions. Arduino UNO is compatible with various software, such as Matrix Laboratory. Parallax data acquisition platform apart from its main program of Arduino IDE. Using compatible shields and external shields, this board's applications can be expanded even further. Here the basic control system is used to control the control unit of smart factory like to communicate between the server and vehicle and gate with stack of beds. The bellow pictures show the basic control system.

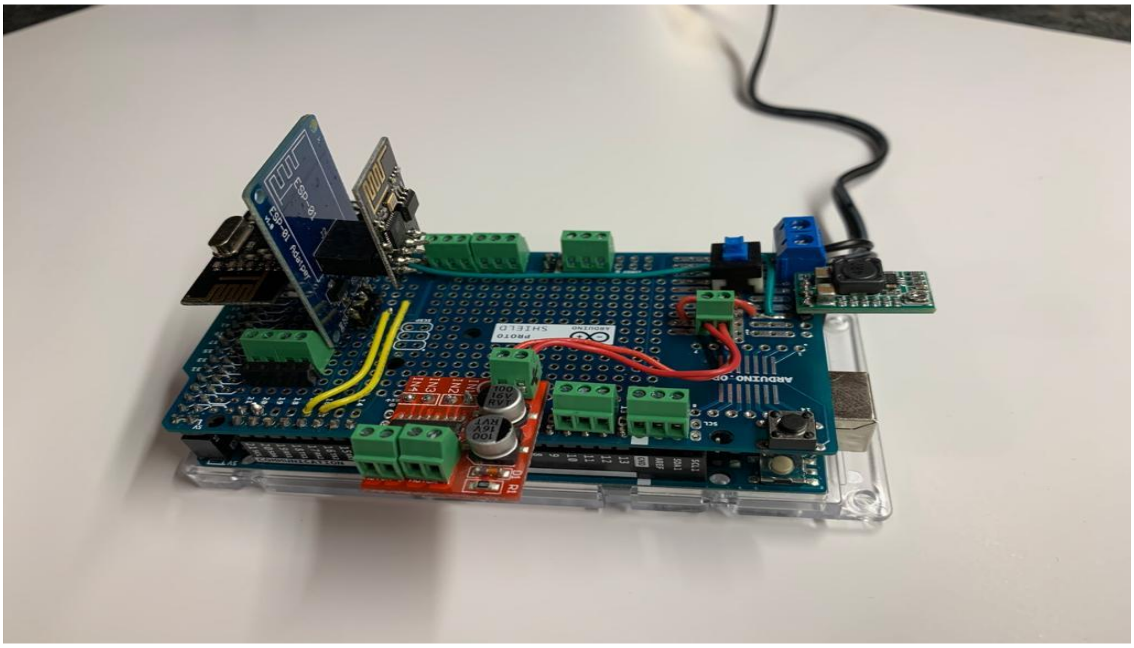

Figure 13: Control system [own]

8.1 Components of control system

These control systems consist of multiple components for communicating and for regulating the power supply and those components are mentioned bellow.

8.1.1 Wi-Fi Module (ESP8266)

The ESP8266 Wi-Fi Module is a self-contained SOC with an integrated TCP or IP protocol stack that allows access to your Wi-Fi network by any microcontroller. The ESP8266 can either host an application or offload all Wi-Fi networking functions from a separate application processor. The access for the ESP8266 is from the local area network or through the internet. Using an Arduino, the USB to TTL convertor to configure the module through the serial pins (RX and TX) receiver and transmitter. And it can be programmed for the communication to control the components. And for providing internet for communication is user friendly and lower in cost. The module can work both as an access point and as station, making the internet of things as easy as possible to collect and upload data on internet.

The conveying strength can be set as four rates from high to low it has to be recommended to use an additional one for external 3.3-volt power supply is enough for this module no longer maximum current is adequate for this. Its high degree of integration on chip enables minimal external circuitry and it designed to occupy minimal space area on pcb board. It includes a self-calibrated RF enabling it to operate under all operating conditions and there was no requirement of RF. These are used for smart home automation and for communicating between two boards and for control systems

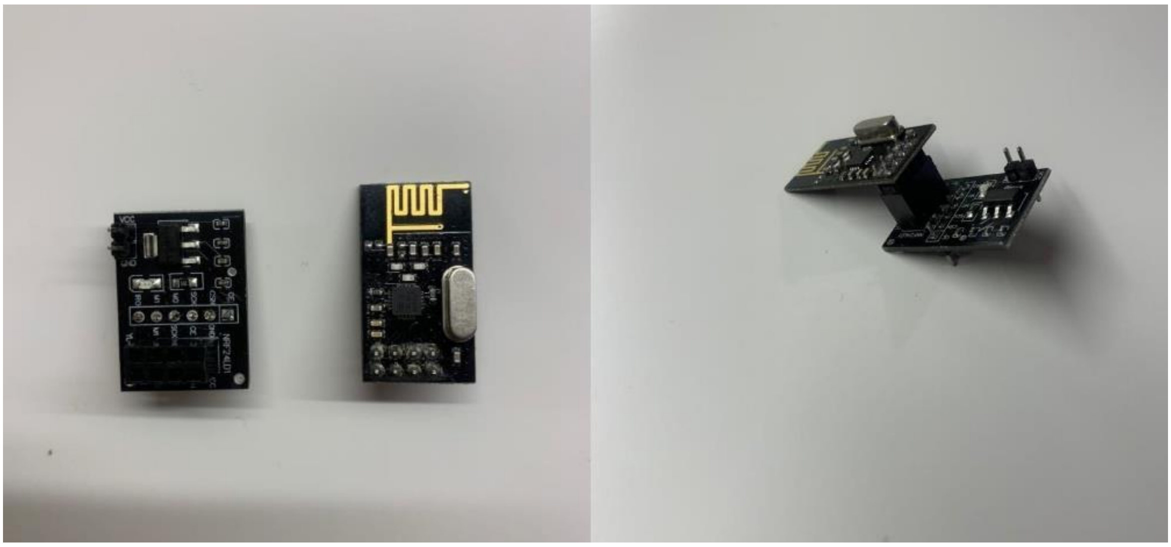

Figure 14: WI-FI Module(ESP8266) [own]

These wireless module absorbs less time for hundreds of mille amperes while import and export the data and it should be always connected to 10 micro farad capacitor the additional power connector for 3.3V voltage power supply has been soldered for the easy connect. It can transmit and receive the radio waves. The signal is received by a wireless router and decoded. The router will forward it to the other board. Up to 2.4 GHz to GHz frequency spectrum can be transmitted. The higher frequency allows more data to transmit the signal. It consists of 8 pin connected to Arduino. The operating voltage for analog and digital power supply are 1.3V to 3.3V voltage. It might be the board got damaged if there is a high power supply while transmitting analog signals. The pin configuration diagram for Wi-Fi module is mentioned bellow. [19]

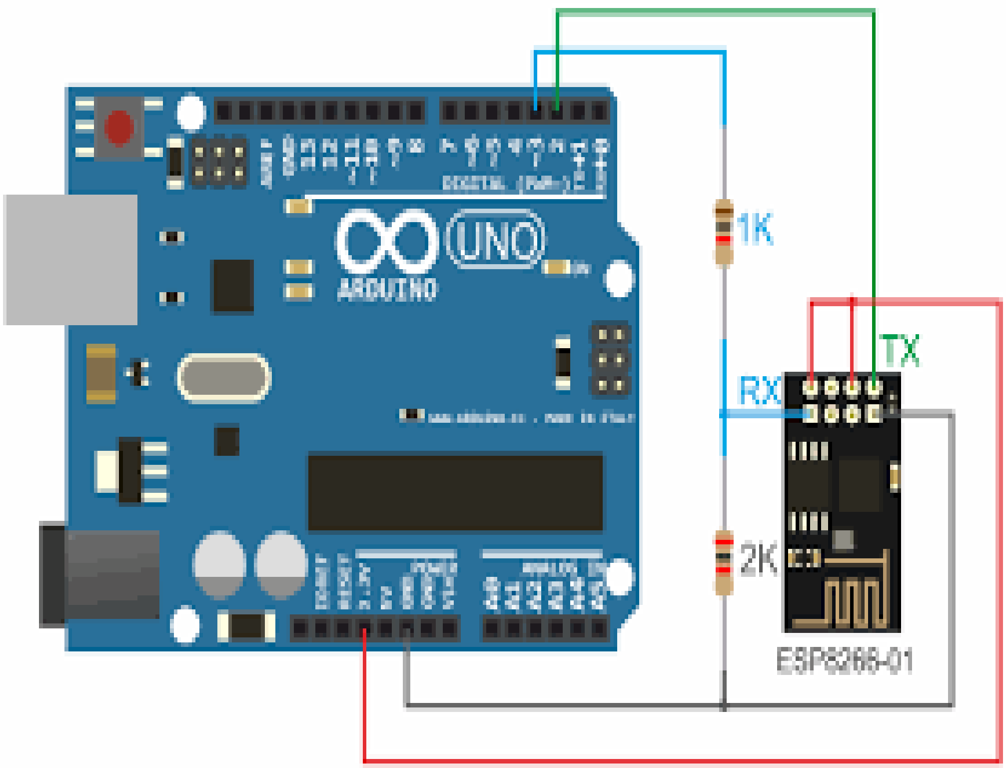

Figure 15: Connection of Wi-Fi module [19]

8.1.2 H- Bridge

An H bridge is an electronic circuit that changes the polarity of an applied voltage on a load. These circuits are frequently used in robotics and other applications to allow DC motors to run back or forth. It consists of four switches which control the current flow to the load. The load connected to the two switch sets. Using one current source, that by closing two switches can drive current in two directions. And two motors are connected to H-Bridge called Motor A and Motor B

It is used to control two DC motors connected with power supply of 12 volt or 5 volts according to the desired value for motor. One thing that very precious about working H-Bridge the short circuit will not be created if its created that it way to burn out the H-Bridge if the current drives close to the two switches in series it may be short out or burn out the H-Bridge. A solid-state H bridge is usually designed with opposite polarity devices.

If the motor drives and control in the same direction the circuit will be burned out. Most DC-to - AC converters use H-bridges such as power inverters, and most AC to DC converters, DC-to - DC push-pull converters, most motor controllers, and many other power electronics. Most stepper motor driven by a motor controller with two H-Bridges. Here the voltage pin (Vin) and ground pins are connected to the step down invertor for regulating voltage and it has 5 pins and two DC motors Motor A and Motor B are connected to the input terminals as shown below in the image. [20]

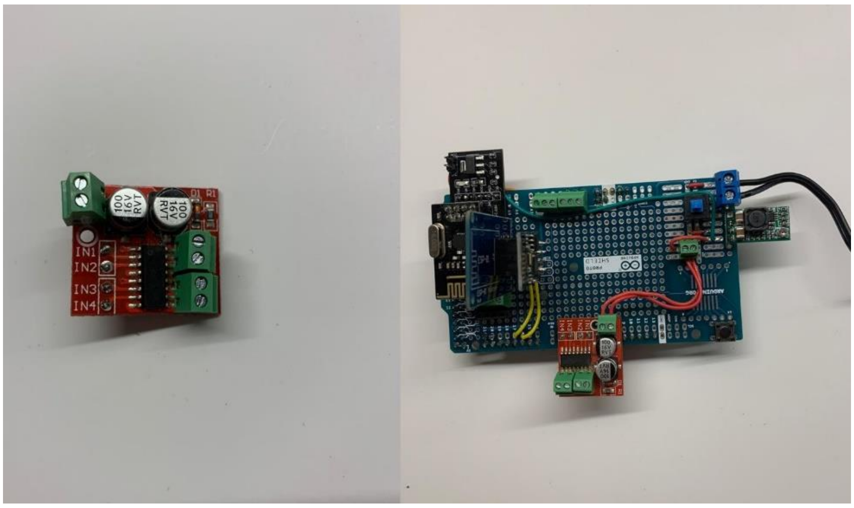

Figure 16: H-Bridge and its Connection

8.1.3 Stepper Motor

A stepper motor or step motor is a brushless DC electric motor splitting the complete rotation into many equal phases. The position of the motor that can be ordered to move and hold without any feedback sensor (open-loop controller) at one of these steps, as long as the motor is carefully sized to the application in terms of torque and speed. Reluctance motors are very large stepping motors with a reduced pole count, and are generally communicated to the closed loop.

Figure 17: Stepper Motor [20]

Next, one electromagnet is given power to make the motor shaft turn which attracts the teeth of the gear magnetically. When the teeth of the gear are aligned with the first electromagnet, they 're offset slightly from the next electromagnet. This means that the gear rotates slightly to align with the next one when the next electromagnet is turned on and the first is turned off. From there the process repeats itself of these rotations is called a step with an integer number of steps making the rotation complete.

8.1.4 Dc Motor

DC motor supply in dc is 5-12v, and specific range of RPM (rotation per minute) output can be obtained. Because of the right hand law of Fleming, any conductive material such as rod is put in an NS magnetic stator case and gives voltage to rotor winding, so that magnetic flux can be produced by a magnet pair and can rotate thereby.

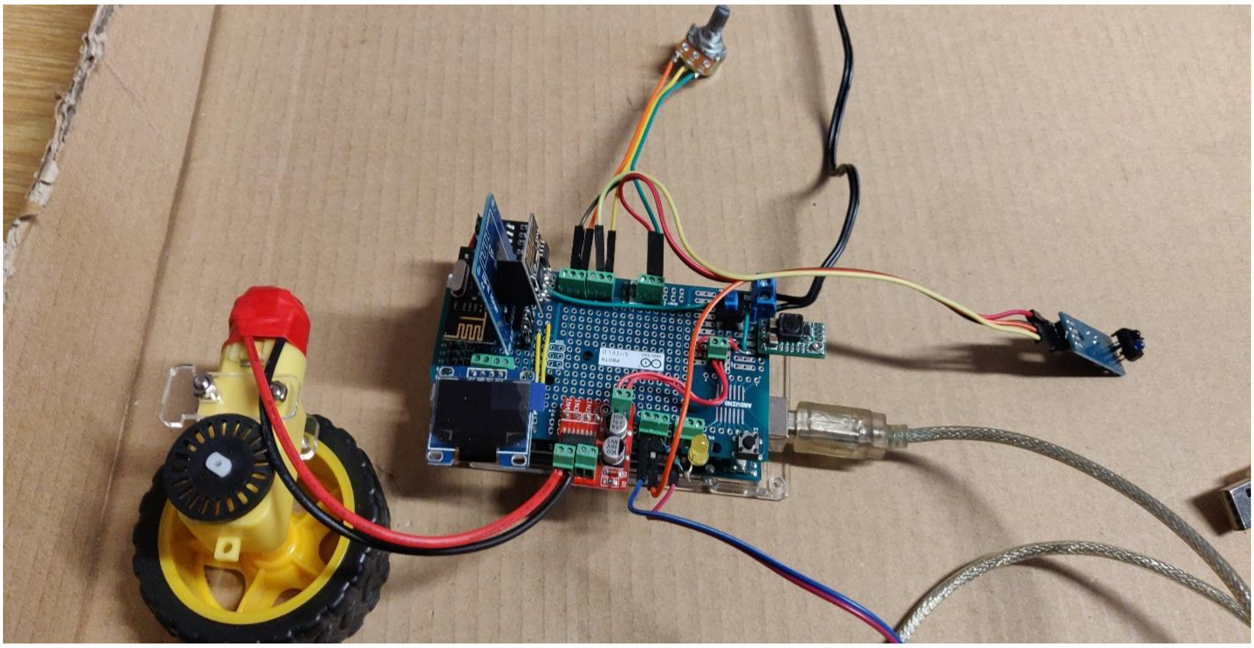

Figure 18: DC motor connected with control system [own]

A simple DC motor has in the stator a stationary collection of magnets and an armature with one or more isolated wire windings wrapped around a soft iron core that concentrates the magnetic field. The windings generally have multiple turns around the core, and several parallel current paths can be found in large motors.

The speed of a DC motor can be regulated over a wide range, either by using an H-Bridge voltage regulator or by adjusting the current intensity of its field windings and another way of controlling the speed by the average input voltage value is changed by transmitting a sequence of On or pulses, the output voltage is proportional to the pulse duration defined as the Duty Cycle.

A DC motor is one of a series of electric machines that transforms electrical direct current into mechanical power. Most types produce rotational motion in a straight line a linear motor directly generates force and motion. The bellow diagram shows that the connection of DC motor with H-Bridge and for to drive the wheel.

8.1.5 IR Sensor

An IR sensor is a device that detects radiation from IR that falls on it. There are various types of IR sensors, in my work I used IR sensor for tracing the path of the wheel. An IR sensor is essentially a device consisting of a pair of an IR LED and a photodiode collectively known as a photo-coupler. The IR LED emits IR radiation, reception and reception intensity which the sensor output is dictated by the photodiode.

It allows the vehicle to differentiate the object or surfaces based on dark or light surface on the path it illuminates the object with an infrared light and measures the light how much light is reflected back. The path tracking sensor is an analog sensor. The closer the object the greater the incident radiation intensity on the photodiode.

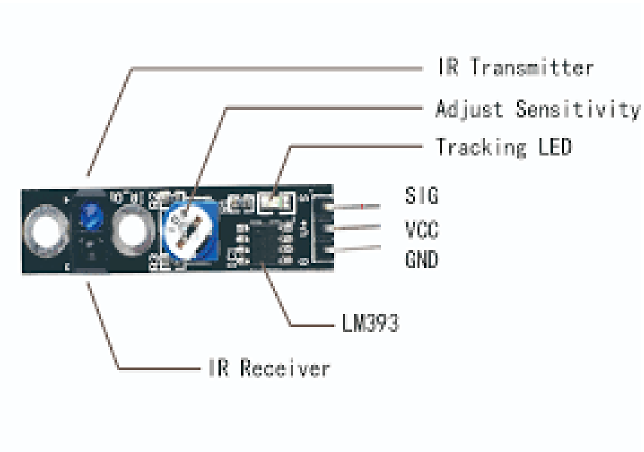

Figure 19: IR sensor [own]

This intensity is rendered by a circuit similar to a voltage, which is then used for the distance determination. It has five analog outputs, and the distance and color of the object being detected affects the output data. When the sensor approaches a black line, the output value gets smaller and smaller. Compare to other module the analog output gives more precious result

As the infrared transmitter emits rays on a piece of paper, as the ray's shine on a white surface, the receiver reflects them and absorbs them, and the pin S produces low level.

8.1.6 Potentiometer

A potentiometer is a simple knob providing a variable resistance which can be read as an analog value in the Arduino board. The value controls the rate an LED flashes at. The Arduino board is connected by three wires. The first one goes to ground from one of the potentiometer 's outer pins. Three wires connect to the Arduino frame. The first goes to ground from one of the potentiometer 's outer pins. The second goes from the potentiometer 5 volts to the other outer plate. The third goes from the potentiometer's analog input 2 to its middle board. By turning the potentiometer 's shaft that changes the amount of resistance on either side. That connected to the potentiometer 's center pin.

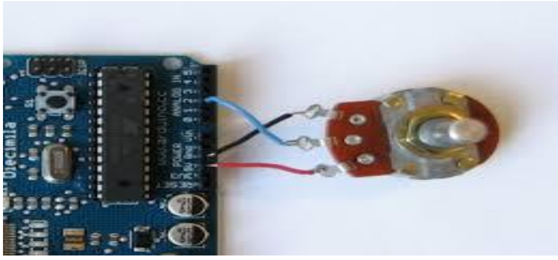

Figure 20: Connection of IR sensor [own]

8.1.7 Step Down Invertor

The Arduino mega requires 5v of DC power to run. The Arduino has a built-in the step down invertor but is better suited for voltages below 12v and controls the voltage to 5v by converting any excess to heat that is not sufficient. Hear the Voltage input (Vin) connected to the step down invertor to reduce the voltage from 6v-15v to 5v.

The input of the module connected to the constant source of power and output to the Arduino and the 5v pin connected to the positive terminal and ground (Gnd) connected to the negative terminal. The polarity and voltage should be in correct manner to secure the Arduino board and to prevent from the damage.

Hear the pins from the terminal are connected to the step down invertor by three analog pins from the terminal are grounded and Vin voltage input pin connected to the the step down invertor to reduce the excess voltage from 12v to 15v to 5v pin configuration of step down invertor are shown below through picture

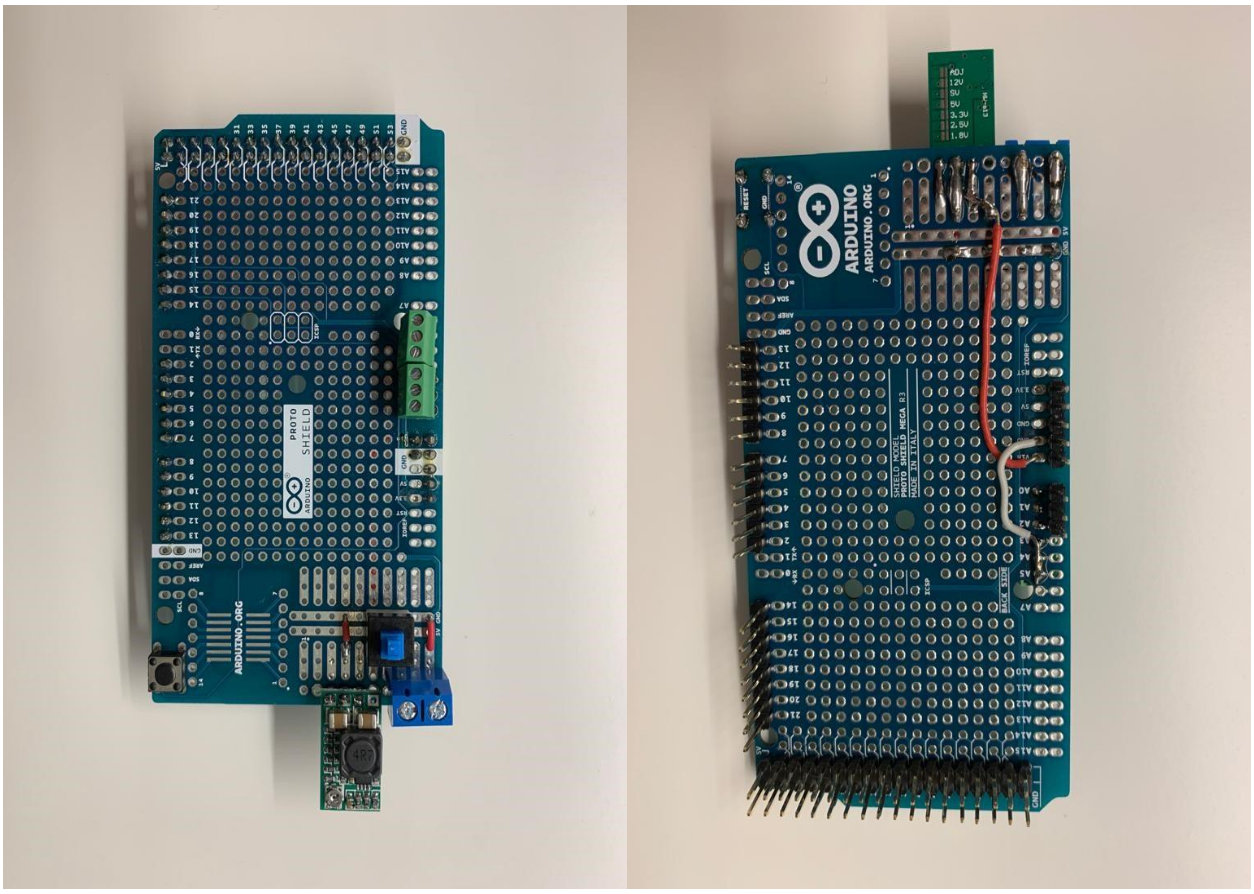

Figure 21: Step down invertor and its Connection [own]

8.1.8 LED screen

These LED screen have a parallel interface which means the microcontroller must interface and manipulate the several pins to control the display at once. A registry pins that controls where the data written into the LED memory. That can either pick a data register containing what is on the screen or an instruction register, which is where the LED controller is searching for instructions about what to do next.

The pin which for reading selects Read Pin R and the pin which for writing selects write pin W. The enable pin has been enabled while writing to the register. The pins from (A0-A7) these state of pins are high and low used to write in the register while writing and the register will read while reading.

And another four pins for display contrast pin (Vo) and one for ground pin (Gnd) and for supply voltage pin (+5v) the LED’s control in two modes of 4-bit and 8-bit. And another two pins connected to the Wi-Fi module for receiving input for display.

The four pins of display are connected to pins (18,19,20,21) in the Pcb board The display process for controlling and display in the data register and instruct to the register are shown below.

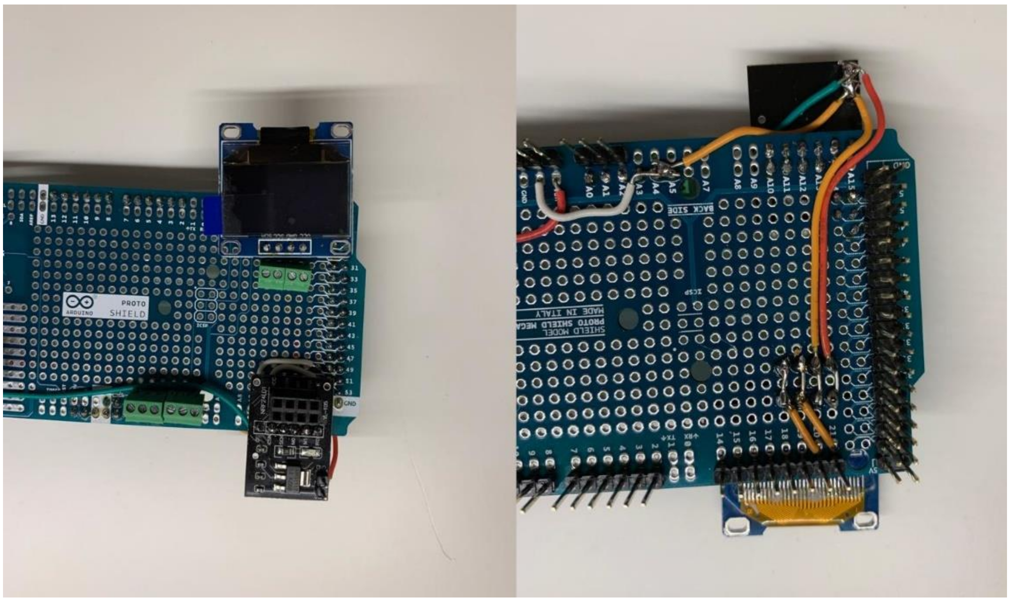

Figure 22: LED screen and its Connections of LED screen [own]

8.2 Application of control system

This basic control system can use to communicate between the machines like milling machine basic appliances of fans and basic mechanical subsystems some of the ideas and procedures are been explained bellow.

8.3 Control system of Milling machine

The milling machine is a device that can accept numerical control inputs to automate a part defined by accurate input positioning. The control system will connect directly to the main controller of subsystem through either a serial connection. The main controller subsystem is capable of interpreting and communicating the limitations of any job, and it can control the movement of the attachable tool heads directly. With direct connection to the main controller subsystem, the main subsystem of milling machine can accept the program which uploaded through the Arduino software the access incoming information through one of the active communication ports, and it will move the direction. The device framework is defined by a mechanical subsystem that allows movement in the x, y, axes and specifies limitations of any appropriate job

The main controller and mechanical subsystems are able to communicate with the motor driver subsystem by means of the power supplied. The motor driver subsystem must include a power supply capable of accepting input power through an electrical outlet plug-in, providing independent movement to the x, y, axes, and an extra board to control the acceleration axis. The most important aim of the mechanical design was to have three panels to ensure precision and to maintain the tool head position.

Each axis was designed to support its weight and resist the force incurred during milling while retaining this resistance to deflection in this ball screw type mechanism is used and rod and coupler connection as mechanical device. The mechanical drive uses ball screws to transform the torque in linear motion from the drive motors. Ball screws use rotating ball bearings inside the nut to minimize frictional losses and reduce wear compared to other alternatives in linear motion. Since the X-axis carries the greatest weight, the maximum forces on the ball screws were determined. That We consider 12V DC power supply in the electrical subsystem, Bidirectional stepper motor, DC motor, Stepper motor driver circuit using transducer, and Arduino mega 2560.

9. AUTOMATON

Automatons are abstract computer models that perform input computation by moving through a sequence of states of configurations. At each computing state, the transformation function decides the next configuration on the basis of the finite portion of the current configuration. The term automation itself closely linked to the term automation denotes automated processes for the processing of different processes. Simply stated automation theory deals with the logic of computation in the case of simple machines referred to as automation.

Through this automaton we can recognize how the system compute the functions and solves the problems with efficiently. Function to be specified as computable or to be represented as decidable. These are the abstract model for the systems. Computing the input by moving through a series of states or configurations. At each computing state, the transformation function decides the next configuration on the basis of the finite portion of the current configuration. As a consequence, once the computation has reached an agreed configuration, it accepts the data. And some representations are there for automatons which means circle represents for states like S1, S2, S3, S4 and ellipse represents for conditions like press and release server listening and box represent for actions like timer's system ON and system OFF the representations are shown below through shapes.

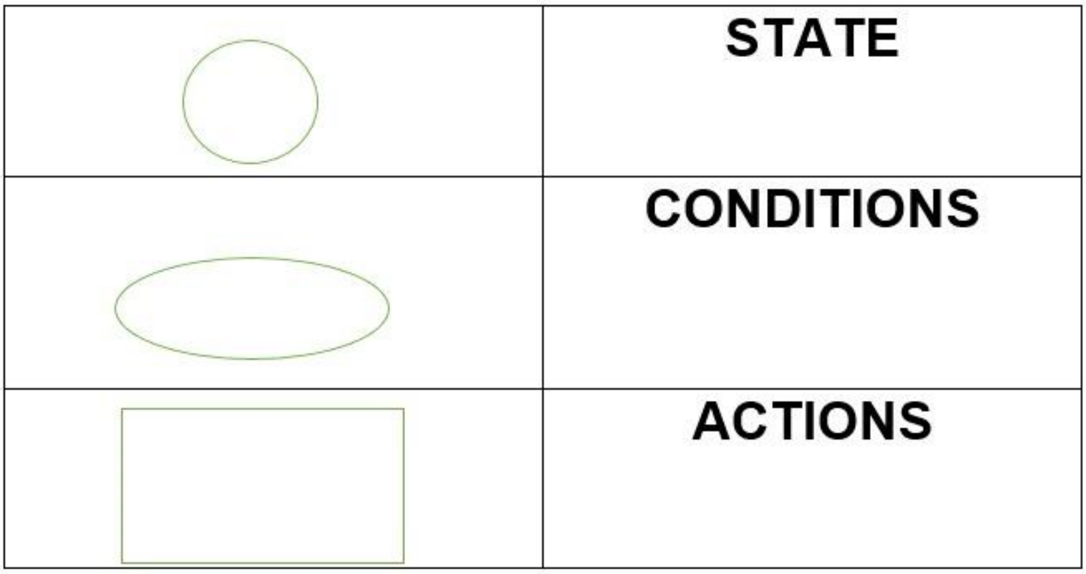

Figure 23: Representation of automatons [own]

9.1 Components of automatons

Common libraries that are ready to use for all control systems by using automatons like conditions actions commands. The Figure 24 shows the automaton graph described for general with conditions and by using blocks here the actions performed by using conditions here it runs automatically with conditions and S1 and S2 are two states after the reset state of button in first state and here the condition satisfies it will jump again to the first sub loop and same as for state of button two (S2).

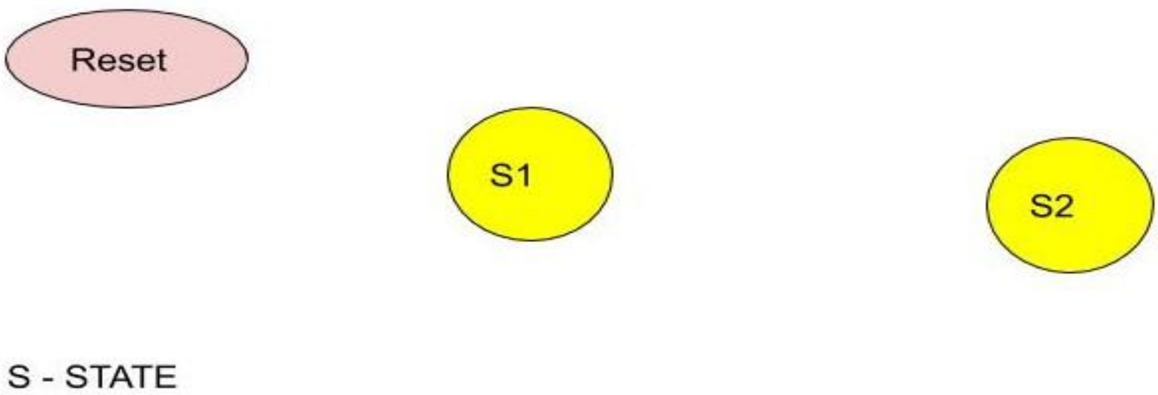

Figure 24: General automaton 1 [own]

The Figure 25 shows the program for the general control system if the conditions satisfy without jumping to the next loop the actions will be performed on the same loops the actions will be done on specified states.

```
int s1;
{
void setup()
s = 1;
}
void loop(){
  {
  if(s==1)
  //Actions to be performed by using conditions
  }
  if(s==2){
    //Actions to be performed by using conditions
  }
  }
```

Figure 25: General libraries for automaton.1 [own]

The Figure 26 shows the automaton graph describes the states with condition and from the reset the condition regarding state the condition satisfies and fulfill it jumps automatically to the next state S2 in the state two there are two conditions if one condition is fulfilled and another condition is failed to fulfilled it will jump to the next state and if both conditions fulfilled it will jump to the next state according to the actions

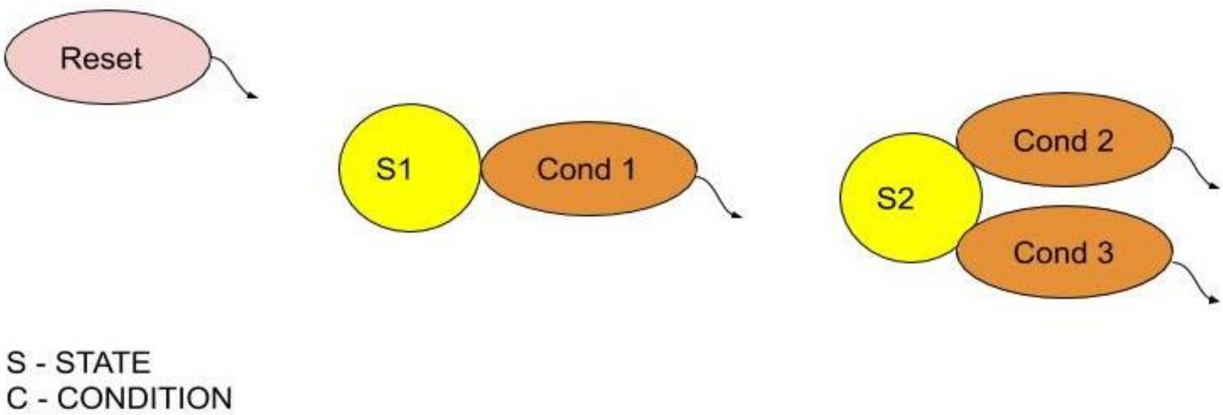

Figure 26: General automaton with conditions [own]

```
int s1;
{
void setup()
s = 1;
}
void loop(){
  {
  if(s==1)
  }
  if(cond 1..){ //if the condition fullfilled it jumps to next state automatically
  }
  if(s==2){
  }
  if(cond 2..){ //Both condition to be checked
  }
  if(cond 3..){
  }
  }
```

Figure 27: Program for jumping to the next loop automatically [own]

The Figure 27 describes. It has two states with conditions if the condition one is fulfilled it will jump to the next state in the loop and on the second state it has two conditions according to the actions the conditions has been fulfilled and it will jump to first state in the loop. After reset it runs automatically by keeping the actions without the conditions in a loop and it also called as an infinite loop with set of instructions here it has two states in the loop it will jump to the next state without any condition. And there are some examples for the condition that mentioned bellow.

```
{
if(a>=x)
}
```

Figure 28: Condition for variables [own]

The Figure 28 shows the condition represents of “if” statement that if some variable reaches some values in this if a greater or equal to the x value the condition said to be fulfilled and it jump to the next state

```
buttonstate = !digitalRead(buttonPin);
if(buttonstate){
}
```

Figure 29: Condition of Buttons for press [own]

This Figure 29 shows the condition is used to check whether the press of the button and the pushbutton state has shifted from its previous stored state. The button state is high and then it increases in button push counter and the output value will be displayed in serial monitor.

```
buttonstate = !digitalRead(buttonPin);
if(!buttonstate){
}
```

Figure 30: Condition of Buttons for release [own]

This Figure 30 shows the condition is used to test the release of the button. And the button state is low and then it decreases in button push counter output value will be appeared in serial monitor and to read the logic state of a pin digitalRead () function has been used. And these conditions are attached with states. These buttons are attached to the digital pin analog value which the sensor value will be low.

```
sensorValue = analogRead(sensorPin);
if(sensorValue >= a){
  }
```

Figure 31: Condition for testing analog sensor value [own]

The Figure 31 shows the condition is used to check the reaching of analog sensor value. For example, if proximity sensor can detect nearest and far object if there is no reflection on sensor it seems the output value of voltage in sensor was 0V if the object is closer than the output voltage will be 5v the analogRead () command converts the input voltage from 0v to 5volts.

```
sensorValue = digitalRead(sensorPin);
if(sensorValue){
  }
```

Figure 32: Condition for testing digital sensor value [own]

The Figure 32 shows the condition is used to check the reaching of digital sensor value. And it checks the position of the sensor that reaches some point. Most of the sensors uses a digital signal than analog signals. And each sensor has its own addresses.

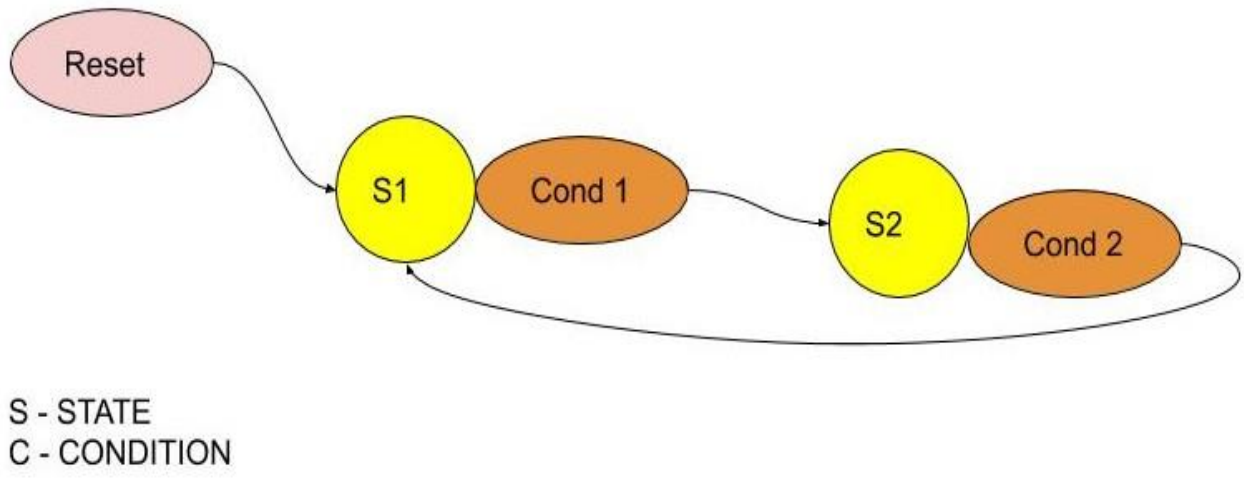

Figure 33: General automaton with jump to the next state [own]

The Figure 33 shows the automaton describes that it contains condition for two states for first state and second state after the reset if the condition satisfies and fulfilled it will jump to next state after the conformation of condition in first state and if the condition failed to fulfill it will not jump to the second state and after the condition fulfilled on the second state again it will jump to the state one the process will be done continually.

```
int s1;
{
  void setup()
  s = 1;
}
void loop(){
  {
    if(s==1)
  }
  if(cond 1..) { //if the condtion fuulfilled jump to the next state
    s = 2;
  }
}

if(s==2){
  if(cond 2..) {//if the condition fullfilled jump to the next state
    s = 1;
  }
}
```

Figure 34: Program for jumping to the next loop by condition [own]

The Figure 34 shows the program shows that it will check the condition and if the condition satisfies and fulfilled it will jump to the next state and on the second state again it will check the condition if the condition is fulfilled it will jump to first state this process has been done on same loop.

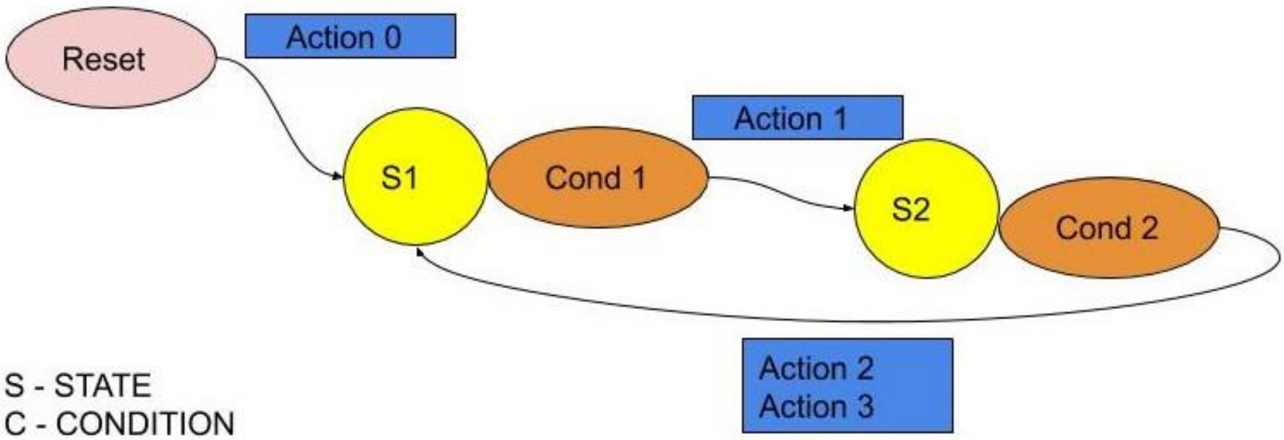

Figure 35: automaton for actions [own]

The Figure 35 shows the describes that each state consists of actions with certain conditions and actions are some command which is done during jump from first state to the second state and those conditions will obey those commands from actions.

```
int s1;
{
  void setup()
  s = 1;
}
void loop(){
  {
    if(s==1)
  }
  if(cond 1..) { //if the condtion fuulfilled jump to the next state
    action 1;
    s = 2;
  }
}

if(s==2){
  if(cond 2..) {//if the condition fullfilled jump to the next state
    action 2;
    action 3;
    s = 1;
  }
}
```

Figure 36: Program for the actions [own]

The Figure 36 shows the program shows that there are two states in the loop sate one and state two and each state consist of action with certain conditions and if the condition fulfilled the actions will be done during jump from one state to other state inside the loop.

```
digital write (actuatorPin,Low);
```

Figure 37:Program for the actions of digital write [own]

The Figure 37 shows the program shows that the command function digital write () actions which able to changes some value of digital actuators like (LED, Motor, Compressor, Valve).The configuration pin has been OUTPUT with pin mode () the corresponding voltage be 5V or according to board voltage as 3V, 3.3V and the pin configured as INPUT the digital write () pin will be enable as high or disable as low and it suggest to be set pinMode () function to input_pullup () and pull up resistor will be enabled.

The above mentioned program shows that the actions to changes the value for some analog actuators like (DC motor, LED with variable intensity of light) and many more like this. It can be used to light an LED at varying brightness or to power a motor at varying speeds. After an analog Write () function the pin generates a steady rectangular wave of the stated duty cycle until the next call.

```
analog write (actuatorPin,x);
```

Figure 38: Program for the actions of analog write [own]

```
void setup(){
  serial.beign(9600); //intialize the serial communication
}
void loop(){
  serial.print(...); //print out the ...
}
```

Figure 39: Program for the serial print [own]

The Figure 39 shows the program shows to print output on serial monitor to speed up the communication serial. begin function is used and 8 bits is equal to one byte bits per second. And serial. Print function used to print the output data on serial port and ASCII used to print the number floats and characters.

```
void setup(){
 lcd.begin(); //setup code
 lcd.backlight();
}
void loop(){
  lcd.clear(); //clear the datas on the screen
  lcd.print(); //Meassage has been printed on the screen
}
```

Figure 40: Program for the LCD print [own]

The Figure 40 shows the program shows that displays the output data on the LCD screen display controlling process involves that entering the data to the display into the data register. And register pin used to control the data on LCD memory enable pins allows to write the data on register.

```
void setup(){
 // Wifi setting and listening
 myRadio.begin();                //WIFI NRF
 myRadio.setChannel(0x60);
 myRadio.setPALevel(RF24_PA_MIN);

 void loop() {
 myRadio.stopListening(); //WIFI NRF TRANSMIT
 }
```

Figure 41: Program for the Wi-Fi communication to transmit [own]

The Figure 41 shows the program shows that to transmit the message through Wi-Fi by using myRadio.beign() function the WI-FI module has been started to transmit the message and Figure 42 shows that to receive the message from the WI-FI module.

```
void setup(){
 // Wifi setting and listening
 myRadio.begin();                //WIFI NRF
 myRadio.setChannel(0x60);
 myRadio.setPALevel(RF24_PA_MIN);

 void loop() {
 while (myRadio.available()) {
 myRadio.read(&dataRecieve, sizeof(dataRecieve));}
 }
```

Figure 42: Program for the Wi-Fi communication to receive [own]

10. IMPLEMENTATION OF CONTROL SYSTEM ON VEHICLE

The control system used to control the vehicle with stack of beads here the DC motor connected with the H-bridge to drive the motor and this motor connected with the wheel for rotation. And there are certain commands and actions to control the vehicle to reach the target point and those are explained by automaton.

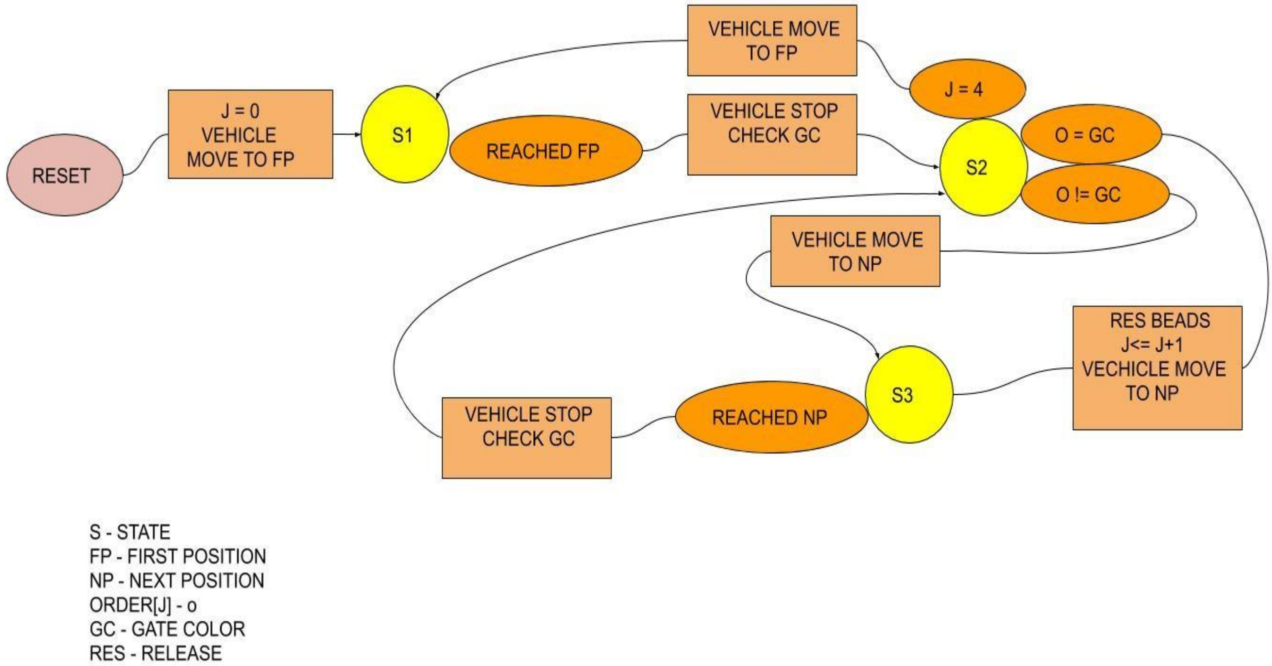

Figure 43: Automaton for vehicle and beads [own]

The Figure 43 shows the automaton describes the general strategy to use for the vehicle movement and it is easy to proceed here the vehicle used to move along with beads based on the actions. Hear after the reset the vehicle will check the position and move to the first position after reaching of first position the vehicle will stop and it will check the gate color of the beads which from order cell of array [J] and color of the gate should be same.

If the color of the beads and color of the gate is not equal the vehicle will move to the next position if the gate color and order color of the beads is equal it will receive the corresponding beads of the stack after that vehicle will move to the other state for collecting the beads and the process will continue up to receiving the order of all beads from the corresponding gates.

10.1 Tracking of vehicle

The vehicle contains five channel tracking sensor totally. In that one sensor is placed in front of the vehicle and another one is placed back side of the vehicle and these two sensors are used to track the line for movement by looking downwards side and other three sensors are place at right and left corner of vehicle by facing downwards and those three sensors are used to track the marks on the line and to stop the vehicle on the mark in account on factory these three tracking sensors used to stop the vehicle on the gate for collecting the beads according to order and control system will control the vehicle actions and the moments.

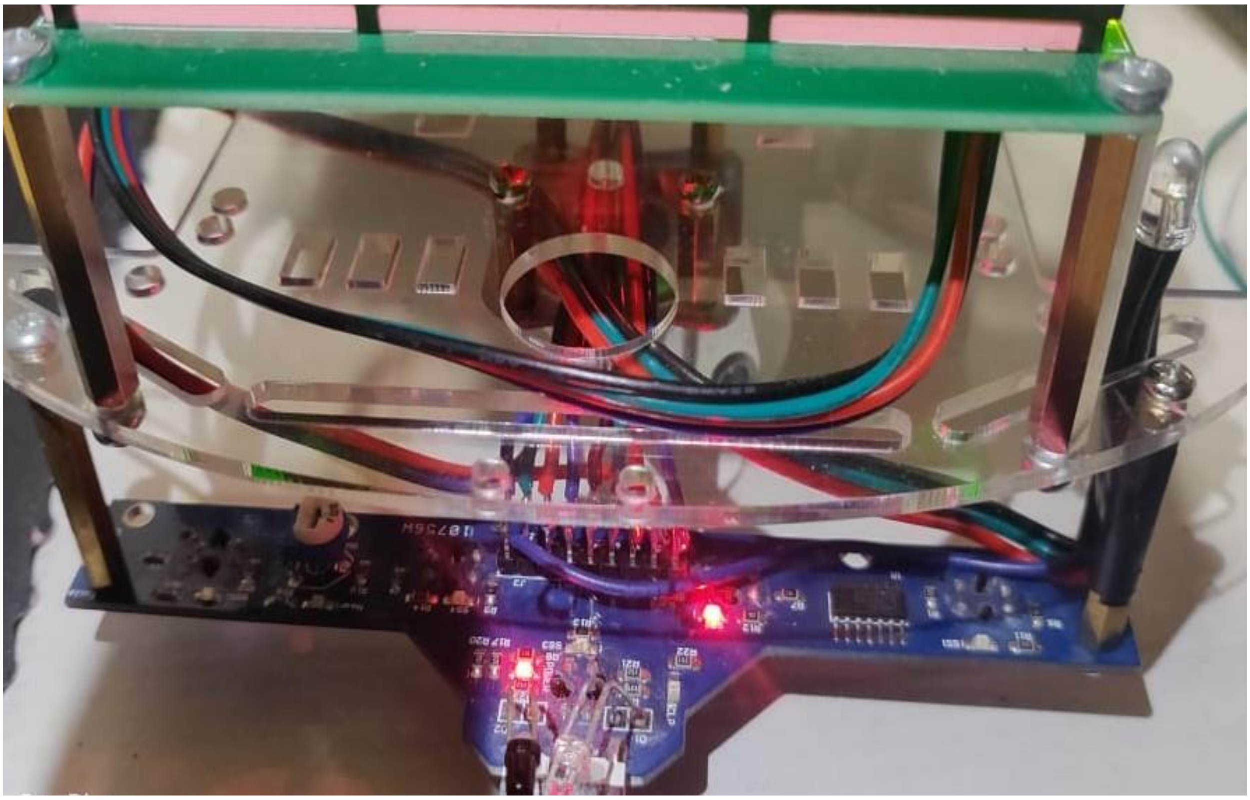

Figure 44: channel tracking sensor [own]

The channel tracking is highly sensitive and performance is stable and the operating voltage this sensor was 5V and on tracking black line the output will be low and while tracking on the white line the output will be high.

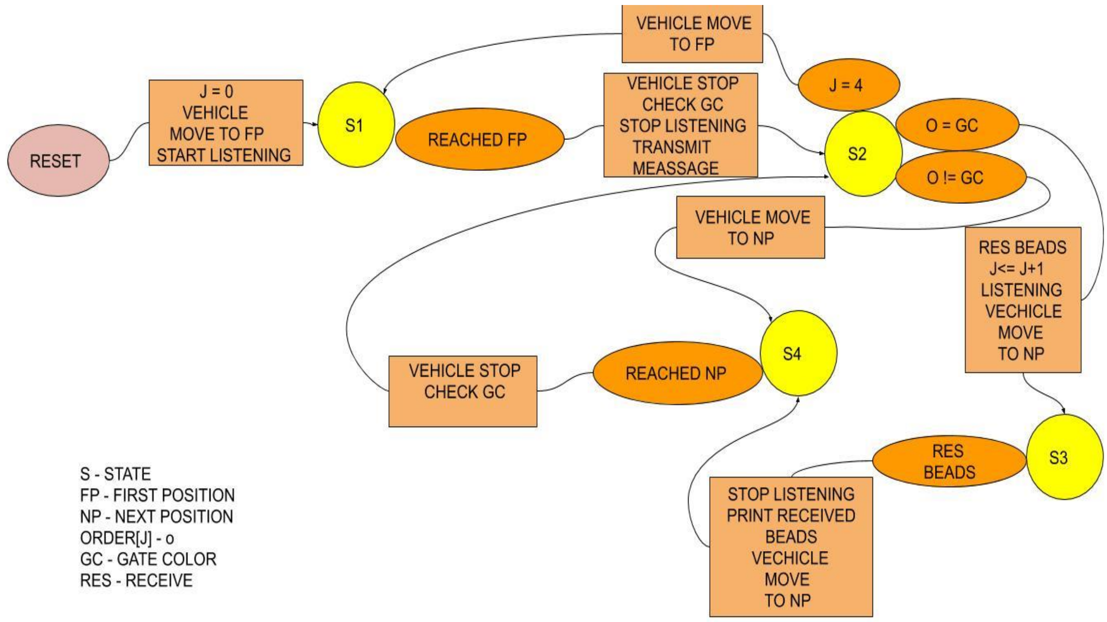

Figure 45: Automaton with communication for vehicle [own]

The Figure 45 describes that the vehicle moment with communication of WI-FI here after the rest the vehicle starts to listen and there will be no order [J] at initial position and the vehicle will move to the first position after it reached the first state it will stop the listening and it started to transmit the massage and it will check the gate color and it will move to the next after receiving of beads its start to listen and it will check the color of the gate and order of the beads [J] if its same it will jump to the next state and process will continue If the order of the beads and color of the gate or not equal the vehicle will move to the next position.

10.2 Control unit of LED

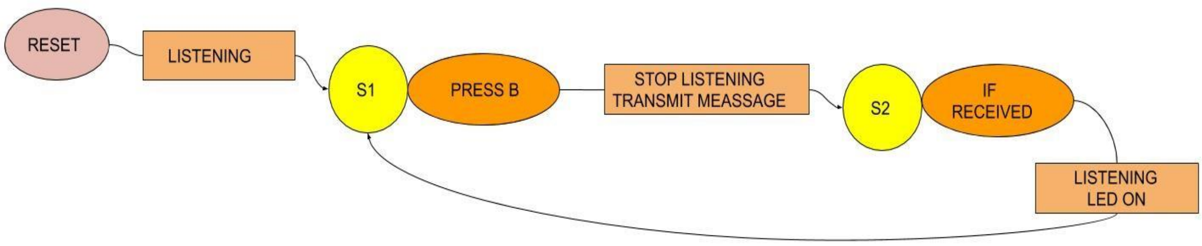

Figure 46: Automaton for glowing LED [Own]

The Figure 46 shows the automaton describes that after reset the server will start to listen. And at the first state the condition is to press the button after the press the server two on the second control unit will start to listen and after the transmission of message from first control unit from the server the second control unit server will stop to listen the message and if the message is received the LED will start to glow on the second control unit if the button press on the control unit one the LED will be glow on control unit two.

```
//WIFI Settings
const char WIFI_AUTH[] = "e25c2bec146c4ffb842df4216e8ea372";
const char WIFI_SSID[] = "Bala667";
const char WIFI_PASSWORD[] = "Hotspot76";
//int s=0;   //step variable
int Sb=1;  //state of buttons automaton
int Ss=1;  //state of main server automaton
int lastc2=0;
int led = 13;

int bR=0;                        //variables to control blynk buttons
int b=0;
int bClear=0;
int bFinish=0;
```

Figure 47: Program for control variables and for WI-FI settings [Own]

.

The Figure 47 shows the program of WI-FI communication setting and for variable controls for the button state and for state of main server automaton on control unit which used to send message for control unit two to glow LED.

```
void setup() {
  pinMode(12, INPUT_PULLUP);     // button for stepping
  Serial.begin(9600);
// LCD settings
  lcd.init();//  lcd.begin(); //begin or init depends on LCD library
  lcd.backlight();
  lcd.clear();
  printLcd(0,"Server-initializing");
  printLcd(1,"TX:");
  printLcd(2,"RX:");
  // Wifi settings
  EspSerial.begin(ESP8266_BAUD);
  delay(1000);
  Blynk.begin(WIFI_AUTH, wifi, WIFI_SSID, WIFI_PASSWORD);
  //timer.setInterval(1000L, sendSensor); //timer of blynk

  // Wifi settings and listening
  myRadio.begin();                //WIFI NRF
  myRadio.setChannel(0x60);
  myRadio.setPALevel(RF24_PA_MAX);
  //myRadio.setDataRate( RF24_250KBPS );
  printLcd(0,"Server-order led on");
  printLcd3();
  sendMessageToPhone3();
  pinMode(led, OUTPUT);
}
```

Figure 48: Setup function for control unit 1 [Own]

In the Figure 48 shows the setup function the button pin given input as 12 and for LED pin number given as input 13. And libraries and functions for LCD display are mentioned at server listening the lcd want to print as “server-initializing” and for transmitting and receiving message TX and RX should be printed on LCD after order from server the LCD want to print as “server-order led on” and the led will start glow after receiving the message from the server of control unit 1.

```
void SbAutomaton(){
 if (Sb==1) {if (bR==1) { Sb=2;
                          digitalWrite(led, HIGH);
                          printLcd3();
                          sendMessageToPhone3();
                         }
            if (bClear==1) {Sb=3;
                              digitalWrite(led, HIGH);
                              printLcd3();
                              sendMessageToPhone3();
                             }
```

Figure 49: Void loop for control unit 1 [Own]

The Figure 49 shows that the automaton loop for automaton which used to transmit message to control unit two it transmits to glow led and print on lcd screen.

```
void loop() {
 if (0==1) {
    bA = !digitalRead(pinA);
 }
 if(bA){
   digitalWrite(pinA ,1);

 }
```

Figure 50: Void loop for control unit 2 [Own]

The Figure 50 shows that the automaton loop function for control unit two after the button press on control unit one the message will transmit to the control unit two and if function here if button has been pressed the led should glow and if button statement is not equal to digital Read(pinA) it will jump to the previous state to check the condition.

10.3 Control unit for Milling Machine

This basic process used for controlling the vehicle with pallet for milling first the control unit on the vehicle will receive the order [J] from the phone here the vehicle want to choose the pallet according to the character size which the pallet have been placed on the warehouse and the vehicle will collect the corresponding pallet from the ware house and then vehicle will move to the milling machine and it will release the pallet from the vehicle to the milling machine for milling the pallet only the vehicle will move from the initial position after receiving the massage and the communication has been done through Wi-Fi between the phone and to the vehicle and one control unit placed on vehicle and another one control unit placed on milling machine and server will listen the massage on control unit.

10.3.1 Progress of work

Here the vehicle be placed on initial position and massage has been send through mobile phone to the vehicle and the server on the vehicle will listen the massage and it will check the size of the character. After that server will stop the listening and the vehicle will move to the ware house to collect the pallet according to the character size and after collecting the pallet from the ware house the server start to listen and it guide the vehicle to move towards on milling machine.

The vehicle will release the pallet from the milling machine and milling machine will receive the pallet from the vehicle and server will start to listen and vehicle will return back to the initial position and the process will continue until the work and vehicle will collect all necessary pallet from the ware house according to the size of characters.

10.3.2 List of commands

With the instruction of control unit and above mentioned progress of work are explained and by using this certain commands were created the vehicle will obey by using those commands which instructed by mobile phone and those commands provide actions and conditions to the vehicle. And those commands for vehicle are used to instruct step by step of progress. And by using this Automaton graph has been described and by using automaton graph Arduino program has been described for control unit. Work progress of command are mentioned bellow.

1. Reset
2. Vehicle move to initial position
3. Massage transmit to the vehicle
4. Vehicle listening
5. If vehicle Received message
6. Stop Listening
7. Vehicle move to ware house

8. If vehicle reached ware house
9. Select the pallet with character size
    9.1. If it's same get pallet j<=j+1 move to milling machine
10. Release the pallet to milling machine
11. Vehicle listening
12. Vehicle move to initial position.

10.3.3 Work progress diagram for control unit

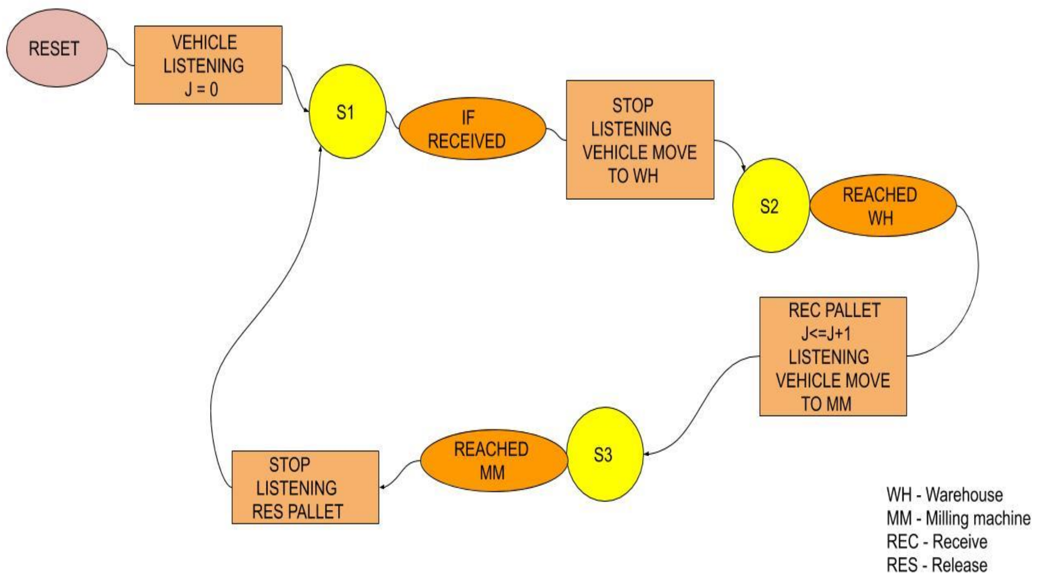

Figure 51: Automaton with communication for vehicle carrying pallet[own]

The Figure 51 shows the automaton describes that the work progress of control unit with vehicle for carrying pallet to milling and based on work progress command the automaton graph has been described. The commands are very important to guide the vehicle with splitting of states and conditions are checked by the states. And if the condition satisfy the actions are being performed by using the commands and based on this program has been described by using Arduino.

This above mentioned automaton shows that after the reset the vehicle start to listen the message or order which transmitted through the mobile phone to the server of the control unit on vehicle at the first state after the massage or order received to the vehicle the server will stop the listening and the vehicle will move the ware house.

After vehicle reached the ware house at the second state it will start to collect the pallet according to the character size send through message via mobile phone. After receiving of pallet the vehicle start moves towards the milling machine to milling the pallet of character and milling machine will collect the pallet from the vehicle at the third state. And after this vehicle start to listening and it will move to the initial position.

```
RF24 myRadio(49,53);                    //WIFI  pins
byte addresses[][6] = {"0"};            //WIFI
```

Figure 52: Program for the byte addresses[own]

The Figure 52 shows the program of the WI-FI pins and functions for the byte addresses and characters for the WI-FI functions

```
struct package {
  char text[20]="command";
  int pwd;
  int tx;
  int rx;
  int Int1;
  int Int2;
  float Float1;
  int Array1[5]={0,0,0,0,0};
};
typedef struct package Package;  // WIFI
Package dataRecieve;
Package dataTransmit;
```

Figure 53: Program for text variables[own]

The Figure 53 shows the program for the structure message which TX is the address for transmitting and RX is the target of the control unit two and int1, int2, and Array1 are the data those variables can configure as for the requirments.

```
void loop() {
  Blynk.run();    //timer.run();   //timer of blynk
  if (Ss==1){           //Ss1-creating order on the phone
          //SMofSb();    //SerialMonitor: debugging blynk buttons and Sb
          SbAutomaton();
          }
  }
  int readCodeFromWifi() {          // listens wifi and puts recieved code to dataRecieve
  while (myRadio.available()) {
    myRadio.read(&dataRecieve, sizeof(dataRecieve));
  }
}
void sendMessageToPhone(String message) {  // sends message to phone
  lcdPhone.clear();
  lcdPhone.print(0, 0, message);
}
void sendMessageToPhone3() {  // sends message line3 to phone
  lcdPhone.clear();
  lcdPhone.print(0, 0, "Order:"+String(text[20]));
}
void printLcd(int line, String message) {   //prints message at N line
  lcd.setCursor(0, line);
  lcd.print("                    ");
  lcd.setCursor(0, line);
  lcd.print(message);
}
void printLcd3() {
  printLcd(3,"Ss"+String(Ss)+" Order:"+String(text[20]));
```

Figure 54: void loop function[own]

The Figure 54 shows the program of the loop function for transmitting character to the control unit two for choosing the pallet according to the size of name characters which send through the mobile phone.

```
while (myRadio.available()) {myRadio.read(&dataRecieve, sizeof(dataRecieve));}   //WIFI NRF listening
if (/*lastcl<dataRecieve.c&&*/dataRecieve.rx==2&&dataRecieve.tx==1&&dataRecieve.val==2) {    //condition to S2 - if order recieved
  lastcl=dataRecieve.c;
  o=0;
  printLcd(2,"RX: c="+String(dataRecieve.c)+" "+String(dataRecieve.tx)+">"+String(dataRecieve.rx)+" "+String(dataRecieve.val)+" "+String(text[20]));
   printLcd(3,"S"+String(S)+" Order:"+String(text[20]));
   printLcd(0,"led-on obted");  delay(3000);
  printLcd(0,"led-conf sending");
   myRadio.stopListening();                         //WIFI NRF TRANSMIT
  dataTransmit.c=dataTransmit.c+1;
  dataTransmit.tx=2;
  dataTransmit.rx=1;
  dataTransmit.val=2;
  myRadio.openWritingPipe(addresses[0]);
  myRadio.write(&dataTransmit, sizeof(dataTransmit));
  printLcd(1,"TX: c="+String(dataTransmit.c)+" "+String(dataTransmit.tx)+">"+String(dataTransmit.rx)+" "+String(dataTransmit.val));
```

Figure 55: Automaton loop function for control unit 2[own]

The Figure 55 shows the program for control unit two after receiving the message from the phone the vehicle transmit the pallet size to the ware house and it received text character size as [20] which given as input on the scope.

10.4 WI-FI Communications

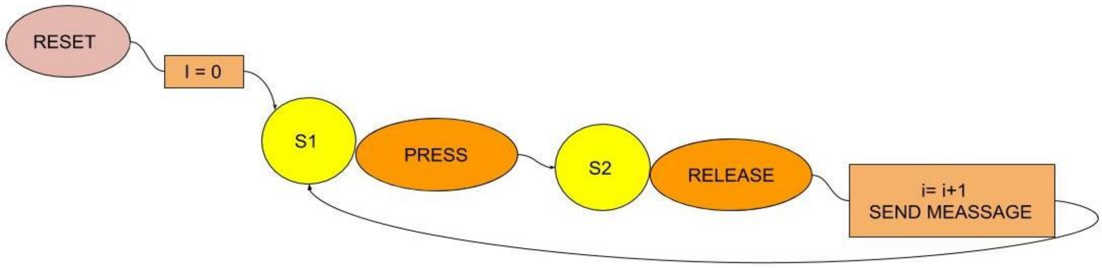

Figure 56: Automaton for transmitting the message[own]

The Figure 56 shows the automaton to transmit the message between the vehicle and to the ware house which the vehicle use some address like characteristics of name and size of the pallet to the warehouse it uses the vehicle to collect the pallet from the ware house for milling purpose here after the reset some order has been placed. And at first state after condition press and on the second state after the condition release the message has been transmitted and order has been placed as I=i+1 to the ware house with addresses.

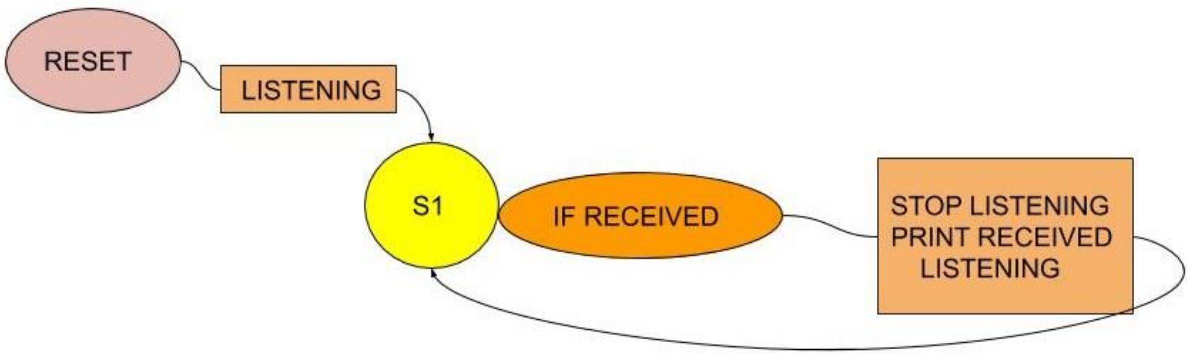

Figure 57: Automaton for Receiving the message[own]

The Figure 57 describes that receiving the message from the server in this ware house receives the message from the vehicle at the first state if the message is received that means the message received from the vehicle to the ware house the server will stop the listening and it will print output as received on the LED.

10.4.1 With conformation

In this while transmitting and receiving the message there will be some conformation after receiving the message and Tx is for transmitting and Rx is for receiving and communication between the two control units done by through Wi-Fi here the vehicle will transmit pallet size to the ware house and it receives the pallet size as message and the control unit from the ware house will send the conformation message to control unit of the vehicle as received the pallet size and again the control unit of vehicle will leave an conformation message as some variable to an ware house as received and this communication has been done with conformation as continually and with infinite loop to stop this communication the values of transmitting (Tx) and receiving (Rx) has to be changed with some variables. Here the Rx will give some conformation as received to the Tx this communication has done with infinite loop.

10.4.2 Without conformation

In this the message has been transmitted and received without any conformation here the control unit of vehicle will send a pallet size to ware house and the control unit of ware house will receive the message and there will be no conformation message gives to the control unit of the vehicle and this communication has done through Wi-Fi with finite loop. Tx will transmit the message

through the WI-FI on control unit and Rx will receive the message through an WI-FI and Rx will not send any conformation to Tx as received.

11. CONCLUSION

Complete study made with industry 4.0 and with communication layer of smart factory. Control units have designed to make communication between components of factory like vehicle to mobile phone and vehicle to the milling machine etc. The communications were done using WI-FI module. Created control unit is based on Arduino board. Control unit is able to solve combination and sequential problems based on automata. The unit has own LCD to monitoring its state. The unit offers using easily programmable analog and digital inputs and outputs and other buses like serial bus or i2c bus. It has two additional modules for Wi-Fi communication. One module serves to communication to mobile phone application and the second module communicates with other control units of the production system.

A library of general functions for automatons and structure of messages for Wi-Fi communication was designed. This control unit is suitable for small applications like to remotely glowing LED’s, remote control of valves etc. Several simple automatons were designed for the control units it helps to write the program and to describe the states of control unit. The functions and actions of control systems are designed very accurately the communication between control systems are done without any human interference. Designed universal control unit was tested in two real systems.

The first is a model of a smart factory with the production of a colored bracelet. This production system consists of a server, a vehicle and several stacks of beads. Created control unit is designed to be used as a control system for all these components. It was used like server and like vehicle control unit.

The second is production system of the little plate with name. It consists of a plate warehouse, a vehicle and a milling machine. The milling machine has its own control system based on the Arduino or Raspberry. Created control unit will be implemented like communication interface between vehicle and the control system. The communication layer and automaton for components were created.

SCOPE OF THESIS:

The control unit of smart factory can be further improved by functions and actions as the requirements required for smart factory.

- Control units should be placed on the milling machine to control the guide ways and the spindle of the tool.
- Control units can be carried out by using such as artificial intelligence (AI) internet of things (IoT) of the smart factory

REFERENCE

[1] Yin, Y., Stecke, K.E. and Li, D., 2018. The evolution of production systems from Industry 2.0 through Industry 4.0. *International Journal of Production Research*, *56*(1-2), pp.848-861.

[2] Matsui, T., Papadopoulos, G., Thubert, P., Watteyne, T., & Montavont, N. (2017). Poster: 4th Industrial Revolution: Toward Deterministic Wireless Industrial Networks.

[3] Basl, J., 2017. Pilot study of readiness of Czech companies to implement the principles of Industry 4.0. *Management and Production Engineering Review*, *8*.

[4] Radziwon, Agnieszka, et al. "The smart factory: exploring adaptive and flexible manufacturing solutions." *Procedia engineering* 69 (2014): 1184-1190.

[5] Wang, Shiyong, et al. "Implementing smart factory of industrie 4.0: an outlook." *International Journal of Distributed Sensor Networks* 12.1 (2016): 3159805.

[6] Jung, Kiwook, K. C. Morris, Kevin W. Lyons, Swee Leong, and Hyunbo Cho. "Mapping strategic goals and operational performance metrics for smart manufacturing systems." *Procedia Computer Science* 44, no. C (2015): 184-193.

[7] Shariat Zadeh, Navid, et al. "Integration of digital factory with smart factory based on Internet of Things." *26th CIRP Design Conference, 2016, KTH Royal Institute of Technology Stockholm, Sweden, 15 June 2016 through 17 June 2016*. Vol. 50. Elsevier, 2016.

[8] Lee, Jay, Behrad Bagheri, and Hung-An Kao. "A cyber-physical systems architecture for industry 4.0-based manufacturing systems." *Manufacturing letters* 3 (2015): 18-23.

[9] Lu, Yan, and Feng Ju. "Smart manufacturing systems based on cyber-physical manufacturing services (CPMS)." *IFAC-[Online]* 50.1 (2017): 15883-15889.

[10] Zhu, Quanyan, Craig Rieger, and Tamer Başar. "A hierarchical security architecture for cyber-physical systems." In *2011 4th international symposium on resilient control systems*, pp. 15-20.

[11] Bagheri, Behrad, Shanhu Yang, Hung-An Kao, and Jay Lee. "Cyber-physical systems architecture for self-aware machines in industry 4.0 environment." *IFAC-[Online]* 48, no. 3 (2015): 1622-1627.

[12] Lee, Edward Ashford, and Sanjit A. Seshia. *Introduction to embedded systems: A cyber-physical systems approach*. Mit Press, 2016.

[13] Kaelbling LP. Learning in embedded systems. MIT press; 1993.

[14] Barrett, Steven F. "Arduino microcontroller processing." *Synthesis Lectures on Digital Circuits and Systems* 8.4 (2013): 1-513.

[15] Kumar RH, Roopa AU, Sathiya DP. Arduino ATMEGA-328 microcontroller. International journal of innovative research in electrical, electronics, instrumentation and control engineering. 2015;3(4):27-9.

[16] Ramadhan, Ade Surya, and L. Budi Handoko. "Rancang Bangun Sistem Keamanan Rumah Berbasis Arduino Mega 2560." *Techno. com* 15, no. 2 (2016): 117-124.

[17] Shahparnia, Shahrooz, and Omar M. Ramahi. "Electromagnetic interference (EMI) reduction from printed circuit boards (PCB) using electromagnetic bandgap structures." *Transactions on Electromagnetic Compatibility* 46.4 (2004): 580-587.

[18] Todica, M. "Controlling Arduino board with smartphone and Blynk via internet." (2016).

[19] Singh, Pushkar, and Sanghamitra Saikia. "Arduino-based smart irrigation using water flow sensor, soil moisture sensor, temperature sensor and ESP8266 WiFi module." *2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*.

[20] Khoucha, Farid, et al. "Hybrid cascaded H-bridge multilevel-inverter induction-motor-drive direct torque control for automotive applications." *IEEE Transactions on Industrial Electronics* 57.3 (2009): 892-899.

APPENDICES INDEX

Appendix A - Study Program.1

```
#define BLYNK_PRINT Serial            // blynk
#include <ESP8266_Lib.h>              // blynk
#include <BlynkSimpleShieldEsp8266.h>   // blynk
#define EspSerial Serial2             // blynk
#define ESP8266_BAUD 115200           // blynk
ESP8266 wifi(&EspSerial);             // blynk
const char WIFI_AUTH[] = "e25c2bec146c4ffb842df4216e8ea372";
const char WIFI_SSID[] = "RadekV";
const char WIFI_PASSWORD[] = "Hotspot76";
BlynkTimer timer;
WidgetLCD lcdPhone(V15);
int bR=0;                   //variables to control blynk buttons
int bG=0;
int bB=0;
int bClear=0;
int bFinish=0;

#include <SPI.h>                      // WIFI
#include "RF24.h"                     // WIFI
#include "U8glib.h"                         // LCD library
U8GLIB_SSD1306_128X64 myOled(U8G_I2C_OPT_NONE); // LCD
unsigned long time;                         // variable for timer
const int b1Pin = 4;   //button pins
const int b2Pin = 5;
const int b3Pin = 6;
const int rPin = 13;    //light pins
const int oPin = 8;
const int gPin = 9;
bool b1;            //variables for reading button states
bool b2;
bool b3;
unsigned long stoptime;     // variable for timer
int S=1;     // variable for automaton and its initial state 1
int val=0;   // variable for testing wifi transfer
int cR=0;    //count of recieved messages
int cT=0;    //count of transmited messages
int pwd=156;

RF24 myRadio(49,53);                  //WIFI  pins
byte addresses[][6] = {"0"};          //WIFI
struct package {
  char text[20]="command";
  int pwd;
  int tx;
  int rx;
  int Int1;
  int Int2;
  float Float1;
  int Array1[5]={0,0,0,0,0};
};
```

```
typedef struct package Package;  // WIFI
Package dataRecieve;
Package dataTransmit;


void setup()
{ Serial.begin(9600);
  pinMode(b1Pin, INPUT_PULLUP);    //config of buttons
  pinMode(b2Pin, INPUT_PULLUP);
  pinMode(b3Pin, INPUT_PULLUP);
  pinMode(rPin, OUTPUT);           //config of lights
  pinMode(oPin, OUTPUT);
  pinMode(gPin, OUTPUT);


  myRadio.begin();                    //WIFI
  myRadio.setChannel(0x61);
  myRadio.setPALevel(RF24_PA_MIN);
  //myRadio.setDataRate( RF24_250KBPS );
  myRadio.openReadingPipe(1, addresses[0]); //automaton-action listen after RESET
  myRadio.startListening();
  myOled.firstPage();              //LCD
  myOled.setFont(u8g_font_unifont);
  myOledUpdate();

  EspSerial.begin(ESP8266_BAUD);      //blynk
  delay(1000);
  Blynk.begin(WIFI_AUTH, wifi, WIFI_SSID, WIFI_PASSWORD);
  //timer.setInterval(1000L, sendSensor); //timer of blynk


}

void loop()
{ Blynk.run();    //timer.run();   //timer of blynk
Serial.println(bR);


  if (S==1){ while (myRadio.available()) {myRadio.read(&dataRecieve, sizeof(dataRecieve));}
            if (dataRecieve.pwd==pwd) {val=dataRecieve.val;
                                       cR=cR+1;
                                       myOledUpdate();
                                      }
            b1=!digitalRead(b1Pin);
            b2=!digitalRead(b2Pin);
            b3=!digitalRead(b3Pin);
            if (b1){digitalWrite(rPin,1);
                    cT=cT+1;
                    val=val+2;
                    sendWifi();
                    S=2;
                    myOledUpdate();
                   }
            if (b2){digitalWrite(oPin,1);
```

```
                S=3;
                myOledUpdate();
               }
         if (b3){digitalWrite(gPin,1);
                S=4;
                myOledUpdate();
               }
        }
  if (S==2){ b1=!digitalRead(b1Pin);
         if (!b1){digitalWrite(rPin,0);
                 S=1;
                 myOledUpdate();
                }
        }
  if (S==3){ b2=!digitalRead(b2Pin);
         if (!b2){digitalWrite(oPin,0);
                 S=1;
                 myOledUpdate();
                }
        }
  if (S==4){ b3=!digitalRead(b3Pin);
         if (!b3){digitalWrite(gPin,0);
                 S=1;
                 myOledUpdate();
                }
        }
}

void myOledUpdate(void)
{ myOled.firstPage();                //LCD
  myOled.setFont(u8g_font_unifont);
 do {myOled.setPrintPos(0,12);
     myOled.print("Control Unit");
     myOled.setPrintPos(0,27);
     myOled.print("Hello Bala!");
     myOled.setPrintPos(0,42);
     myOled.print("val");
     myOled.print(val);
     myOled.print(" cR");
     myOled.print(cR);
     myOled.print(" cT");
     myOled.print(cT);
     myOled.setPrintPos(0,57);
     myOled.print("S");
     myOled.print(S);
     myOled.print(" ");
     myOled.print(bR);
     myOled.print(bG);
     myOled.print(bB);
     myOled.print(" ");

    } while(myOled.nextPage());
}
```

```
void readCodeFromWifi(void) {          // listens wifi and puts recieved code to dataRecieve
  while (myRadio.available()) {
    myRadio.read(&dataRecieve, sizeof(dataRecieve));
  }
}

void sendWifi(void) {
  myRadio.stopListening();             //WIFI NRF TRANSMIT
  dataTransmit.pwd=pwd;
  dataTransmit.tx=1;
  dataTransmit.rx=1;
  dataTransmit.val=val;
  myRadio.openWritingPipe(addresses[0]);
  myRadio.write(&dataTransmit, sizeof(dataTransmit));
  myRadio.openReadingPipe(1, addresses[0]);
  myRadio.startListening();
}
//    sync with Blynk
BLYNK_WRITE(V0) {
  bR = param.asInt();
}
BLYNK_WRITE(V1) {
  bG = param.asInt();
}
BLYNK_WRITE(V2) {
  bB = param.asInt();
}
BLYNK_WRITE(V3) {
  bClear = param.asInt();
}
BLYNK_WRITE(V4) {
  bFinish  = param.asInt();
}
```