AN UNIQUE METHODOLOGY FOR TRANSMISSION LINE BREAKAGE DETECTION AND ALERTING SYSTEM

A PROJECT REPORT

Submitted by

R. GOKUL (211419105037)

S. GOUTHAM (211419105039)

N.P HAREESH (211419105042)

A. KAVIARASAN (211419105065)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL2023

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"AN UNIQUE METHODOLOGY FOR TRANSMISSION LINE BREAKAGE DETECTION AND ALERTING SYSTEM"** is the bonafide work of "**R. GOKUL [211419105037] GOUTHAM. S [211419105039] N.P HAREESH [211419105042] A.KAVIARASAN [211419105065]"** who carried out the project work under my supervision.

SIGNATURE

Dr. S. SELVI, M.E., Ph.D.,
HEAD OF THE DEPARTMENT
DEPARTMENT OF EEE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLE,
CHENNAI - 600123

SIGNATURE

Mr. S. BALAJI, M.E.
ASSISTANT PROFESSOR
DEPARTMENT OF EEE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLE,
CHENNAI - 600123

Submitted for End Semester Project Viva Voice held on …….. at

Panimalar Engineering College, Chennai.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Our sincere thanks to our Honourable Founder and Chairman, **Dr. JEPPIAAR, M.A., B.L., Ph.D.,** for his sincere endeavour in educating us in his premier institution.

We would like to express our deep gratitude to our beloved **Secretary and Correspondent, Dr.P. CHINNADURAI, M.A., M.Phil. Ph.D.** for his enthusiastic motivation which inspired us a lot in completing this project and our sincere thanks to our directors **Mrs.C. VIJAYA RAJESWARI, Dr.C. SAKTHI KUMAR, M.E., Ph. D** and **Dr. SARANYASREE SAKTHIKUMAR, B.E, M.B.A, Ph. D** for providing us with the necessary facilities for the completion of this project.

We would like to express thanks to our Principal, **Dr. K. Mani M.E., Ph.D.,** for having extended his guidance and cooperation.

We would also like to thank our **Head of the Department**, **Dr.S. Selvi, M.E., Ph.D., Professor and Head, Department of Electrical and Electronics Engineering** for her encouragement.

Personally, we thank our Guide **Mr. S. BALAJI, M.E. in Department of Electrical and Electronics engineering** for the persistent motivation and support for this project, who at all times was the mentor of germination of the project from a small idea.

We express our sincere thanks to the project coordinators **Dr.S. Deepa & Dr.N. MANOJ KUMAR, M.E., Ph.D., in Department of Electrical and Electronics Engineering** for the Valuable suggestions from time to time at every stage of our project.

Finally, we would like to take this opportunity to thank our family members, faculty and non-teaching staff members of our department, friends, well-wishers who have helped us for the successful completion of our project.

ABSTRACT

Many times, we read in newspapers that Humans & Animals die due to electrical shock in remote areas or in agricultural areas as contact with broken & hanging live supply wires. Safety circuitry of Distribution Company is inadequate and due to this line remains live with broken wires. Here we describe a modification to existing power distribution system for wire break detection and a power supply breaking mechanism. Circuit breaker with shunt trip mechanism breaks the supply and avoids damages from electrical accidents due to overhead transmission lines conductor breakage problems. We utilize various communication devices to provide indication about the number of voltages transmitted from one section to another. We employ a deep learning technique by this system which enables a monitoring of electrical voltage transmitted comparing with threshold value and takes decision accordingly.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

LCD	LIQUID CRYSTAL DISPLAY
LED	LIGHT EMITTING DIODE
PWM	PULSE WIDTH MODULATION
IOT	INTERNET OF THINGS
PCB	PRINTED CIRCUIT BOARD
USB	UNIVERSAL SERIAL BUS
SOC	SYSTEM ON CHIP
SMPS	SWITCHED MODE POWER SUPPLY
UART	UNIVERSAL ASYNCHRONOUS RECEIVER-TRANSMITTER
HVDC	HIGH VOLTAGE DIRECT CURRENT
ICSP	INCIRCUIT SERIAL PROGRAMMING
LR-WPAN	LOW-RATE WIRELESS PERSONAL AREA NETWORK
ASCII	AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE
DWT	DISCRETE WAVELET TRANSFORM
WTC	WAVELET TRANFORM COEFFICIENTS

CHAPTER 1

1. INTRODUCTION:

In this project we will monitor transmission line multiple fault detection and indication send to electricity board via through alarm and message. The steady state operating mode of a power system is balanced 3-phase ac. However, due to sudden external or internal changes in the system, this condition is disrupted. When the insulation of the system fails at one or more points or a conducting object comes in contact with a live point, a short circuit or fault occurs. The causes of faults are numerous, e.g., lighting, heavy winds, trees falling across lines, vehicles colliding with towers or poles, birds, line breaks etc. A fault involving all the three phases is known as symmetrical fault while one involving only or one or two phases is known as unsymmetrical fault. We are working on monitoring these faults during occurring on transmission line. In this project we indicate the fault on LCD and transmit the fault message via GSM. These faults are monitoring by using microcontroller. In this the information is passed using the zigbee module by wireless communication. The step-down transformer is used for reduce the voltage in microcontroller to control the entire simulation which consist of two pole x& y which is communicate using the zigbee module. We use the relay for the transmission of power from one node to other by using the embedded system simulation using Isis pro in real time. This methodology has several advantages over traditional transmission line breakage detection systems. First, the use of advanced sensors and machine learning algorithms allows for more accurate and reliable fault detection. Second, the system can detect faults in real-time, allowing for faster repairs and minimizing the potential for damage or power outages. Finally, the system can be easily integrated into existing power transmission networks, making it a cost-effective solution for improving system reliability and reducing downtime. In conclusion, the proposed unique methodology for transmission line breakage detection and alerting system offers a reliable and efficient solution for power transmission networks to detect and respond to faults in real-time, improving system reliability, and cost-effectiveness.

CHAPTER 2

2. LITERATURE SURVEY:

A NOVEL FAULT-LOCATION METHOD FOR HVDC TRANSMISSION LINES BASED ON SIMILARITY MEASURE OF VOLTAGE SIGNALS

ABSTRACT: In this paper, a method for fault locating in HVDC transmission lines is proposed which only uses the voltage signal measured at one of the line terminals. The postfault voltage signal, in a relatively short-time window, is considered and the corresponding fault location is estimated based on the similarity of the captured voltage signal to existing patterns. In this approach, the Pearson correlation coefficient is used to measure the similarity. Despite simplicity and low complexity of the proposed fault-location method, it does not suffer from the technical problems which are associated with the traveling-wave-based methods, such as the difficulty of identifying traveling wave fronts or the strong dependency of accuracy on the sampling frequency. Numerous training and test patterns are obtained by simulating various fault types in a long overhead HVDC transmission line under different fault location, fault resistance, and pre fault current values. The accuracy of the proposed fault-location method is verified using these patterns.
AUTHORS: MOHAMMAD FARSHAD, JAVAD SADEH.

AN ADAPTIVE PMU BASED FAULT DETECTION/LOCATION TECHNIQUE FOR TRANSMISSION LINES. I. THEORY AND ALGORITHMS

ABSTRACT: An adaptive fault detection/location technique based on a phasor measurement unit (PMU) for an EHV/UHV transmission line is presented. A fault detection/location index in terms of Clarke components of the synchronized voltage and

current phase is derived. The line parameter estimation algorithm is also developed to solve the uncertainty of parameters caused by aging of transmission lines. This paper also proposes a new discrete Fourier transform (DFT) based algorithm (termed the smart discrete Fourier transform, SDFT) to eliminate system noise and measurement errors such that extremely accurate fundamental frequency components can be extracted for calculation of fault detection/location index. The EMTP was used to simulate a high voltage transmission line with faults at various locations. To simulate errors involved in measurements, Gaussian-type noise has been added to the raw output data generated by EMTP. Results have shown that the new DFT based method can extract exact phases in the presence of frequency deviation and harmonics. The parameter estimation algorithm can also trace exact parameters very well. The accuracy of both new DFT based method and parameter estimation algorithm can achieve even up to 99.999% and 99.99% respectively, and is presented in Part II. The accuracy of fault location estimation by the proposed technique can achieve even up to 99.9% in the performance evaluation, which is also presented in Part II.

AUTHORS: JOE-AIR JIANG, JUN-ZHE YANG, YING-HONG LIN.

DISCUSSION OF "A NEW PROTECTION SCHEME FOR FAULT DETECTION, DIRECTION DISCRIMINATION, CLASSIFICATION, AND LOCATION IN TRANSMISSION LINES"

ABSTRACT: In this paper, the author discusses the original paper "A New Protection Scheme for Fault Detection, Direction Discrimination, Classification, and Location in Transmission Lines" In that paper, the original authors proposed a new transmission line fault protective scheme using synchronized phasor measurements through global positioning system (GPS) satellites. The author believes the original authors should be commended for their paper investigating the proposed scheme extensively by numerous simulation studies.

AUTHORS: M. SANAYE-PASAND.

NEW ALGORITHM FOR DETECTION AND FAULT CLASSIFICATION ON PARALLEL TRANSMISSION LINE USING DWT AND BPNN BASED ON CLARKE'S TRANSFORMATION.

ABSTRACT: This paper presents a new algorithm for fault detection and classification using discrete wavelet transform (DWT) and back-propagation neural network (BPNN) based on Clarke's transformation on parallel transmission. Alpha and beta (mode) currents generated by Clarke's transformation were used to convert the signal of discrete wavelet transform (DWT) to get the wavelet transform coefficients (WTC) and the wavelet energy coefficient (WEC). Daubechies4 (Db4) was used as a mother wavelet to decompose the high frequency components of the signal error. The simulation was performed using PSCAD/EMTDC for transmission system modeling. Simulation was performed at different locations along the transmission line with different types of faults and fault resistance, fault location and fault initial angle on a given power system model. Four statistic methods utilized are in the present study to determine the accuracy of detection and classification faults. The results show that the best Clarke transformation occurred on the configuration of 12-24-48-4, respectively. For instance, the errors using mean square error method, the errors of BPNN, Pattern Recognition Network and Fit Network are 0.03721, 0.13115 and 0.03728, respectively. This indicates that the BPNN results are the lowest error.

AUTHORS: ABDULLAHASUHAIMI, MOHD ZIMAKMURSAIN, MOHD WAZIR, MUSTAFA AHMAD, RIZALSULTAN, RAHIMUDDIN.

DESIGN AND PERFORMANCE EVALUATION OF SUBSYNCHRONOUS DAMPING CONTROLLER WITH STATCOM

ABSTRACT: A long transmission line needs controllable series as well as shunt compensation for power flow control and voltage regulation. This can be achieved by suitable combination of passive elements and active FACTS controllers. In this paper, series passive compensation and shunt active compensation provided by a static synchronous compensator (STATCOM) connected at the electrical center of the transmission line are considered. It is possible to damp sub synchronous resonance (SSR) caused by series capacitors with the help of an auxiliary sub synchronous damping controller (SSDC) on STATCOM. The objective of this paper is to investigate the SSR characteristics of the system and propose a new design procedure for SSDC based on nonlinear optimization to meet the specifications on the damping torque in the range of critical torsion frequencies. The SSDC uses the Thevenin voltage signal to modulate the reactive current reference of STATCOM. The controller regulates either reactive current (supplied by the STATCOM) or the bus voltage. The 3-phase model of the STATCOM is based on switching functions. By neglecting harmonics in the switching function, D-Q model is derived which is combined with similar models of the other system components for linear analysis. The results of the linear analysis are validated by carrying out transient simulation based on the detailed nonlinear models. The study is performed on the system adapted from the IEEE First Benchmark Model.

AUTHORS: K.R. PADIYAR, N. PRABHU.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In the existing process there is no automatic system for finding the power line break occurrence. Until now human needs are the only solution for finding those problems. To overcome this, we are going to deal with the new techniques.

3.2 PROPOSED SYSTEM:

A transmission line breakage detection and alerting system is a critical component of any power transmission network. It helps to quickly detect and isolate faults in the system, allowing for timely repairs and preventing potential damage to equipment and power outages. One unique methodology for a transmission line breakage detection and alerting system involves the use of advanced sensors and machine learning algorithms. The system would consist of a network of sensors placed along the transmission lines, which would constantly monitor various parameters such as voltage, current, and temperature. The sensor data would be transmitted to a central monitoring system, which would use machine learning algorithms to analyze the data in real-time. The algorithms would be trained to detect patterns and anomalies in the data that are indicative of a line breakage or other fault in the system. When a fault is detected, the monitoring system would immediately alert the appropriate personnel, such as power grid operators or maintenance crews. The alert could be in the form of an automated message or an alarm, depending on the severity of the fault. This methodology has several advantages over traditional transmission line breakage detection systems. First, the use of advanced sensors and machine learning algorithms allows for more accurate and reliable fault detection. Second, the system can detect faults in real-time, allowing for faster repairs and minimizing the potential for damage or power outages. Finally, the system can be easily integrated into

existing power transmission networks, making it a cost-effective solution for improving system reliability and reducing downtime.

3.3 ADVANTAGES:

1. Early Fault Detection: The system uses advanced sensors and machine learning algorithms to detect faults in real-time. This allows for early detection of faults before they can escalate into more severe issues, reducing the risk of damage to equipment and power outages.

2. Improved System Reliability: By detecting faults early, the system can help prevent power outages and reduce downtime. This improves the overall reliability of the power transmission network and enhances customer satisfaction.

3. Cost-Effective: The system is cost-effective as it can be easily integrated into existing power transmission networks. It reduces the need for expensive manual inspections and repairs, which can be time-consuming and labor-intensive.

4. Increased Safety: The system improves safety by identifying faults before they can cause potential hazards such as electrical fires, explosions, or other accidents.

5. Scalability: The system can be scaled up or down depending on the size and complexity of the transmission network. It can also be customized to suit specific requirements and needs.

6. Real-time Alerting: The system provides real-time alerts to the appropriate personnel, allowing them to respond quickly and take corrective action.

CHAPTER 4

4. COMPONENTS

4.1 MICROCONTROLLER:

Arduino is a great platform for beginners into the World of Microcontrollers and Embedded Systems. With a lot of cheap sensors and modules, you can make several projects either as a hobby or even commercial. As technology advanced, new project ideas and implementations came into play and one particular concept is the Internet of Things or IoT. It is a connected platform, where several “things” or devices are connected over internet for exchange of information. In DIY community, the IOT projects are mainly focused on Home Automation and Smart Home applications but commercial and industrial IoT projects have far complex implementations like Machine Learning, Artificial Intelligence, Wireless Sensor Networks etc. The important thing in this brief intro is whether it is a small DIY project by a hobbyist or a complex industrial project, any IoT project must have connectivity to Internet. This is where the likes of ESP8266 and ESP32 come into picture. If you want to add Wi-Fi connectivity to your projects, then ESP8266 is a great option. But if you want build a complete system with Wi-Fi connectivity, Bluetooth connectivity, high resolution ADCs, DAC, Serial Connectivity and many other features, then ESP32 is the ultimate choice.

4.1.1 What is ESP32?

ESP32 is a low-cost System on Chip (SoC) Microcontroller from Espressif Systems, the developers of the famous ESP8266 SoC. It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Tensilica’s 32-bit Xtensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth. The good thing about ESP32, like ESP8266 is its integrated RF components like Power Amplifier, Low-Noise Receive

Amplifier, Antenna Switch, Filters and RF Balun. This makes designing hardware around ESP32 very easy as you require very few external components.

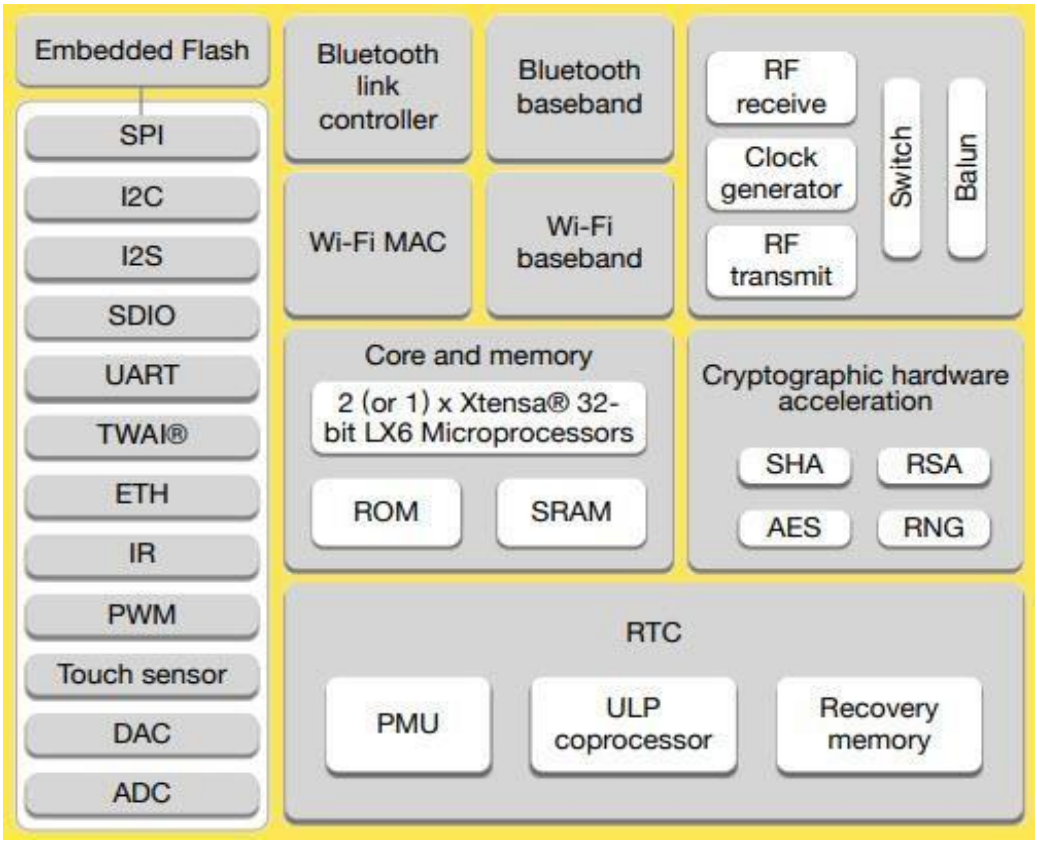


Fig 4.1 ESP32 MICROCONTROLLER

Another important thing to know about ESP32 is that it is manufactured using TSMC's ultra-low-power 40 nm technology. So, designing battery operated applications like wearables, audio equipment, baby monitors, smart watches, etc., using ESP32 should be very easy.

4.1.2 GENERAL DESCRIPTION:

ESP32 is a series of low-cost, low-power system on a chip microcontroller with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series employs either a Tensilica

Xtensa LX6 microprocessor in both dual-core and single-core variations, Xtensa LX7 dual-core microprocessor or a single-core RISC-V microprocessor and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power-management modules. ESP32 is created and developed by Espressif Systems, a Shanghai-based Chinese company, and is manufactured by TSMC using their 40 nm process.

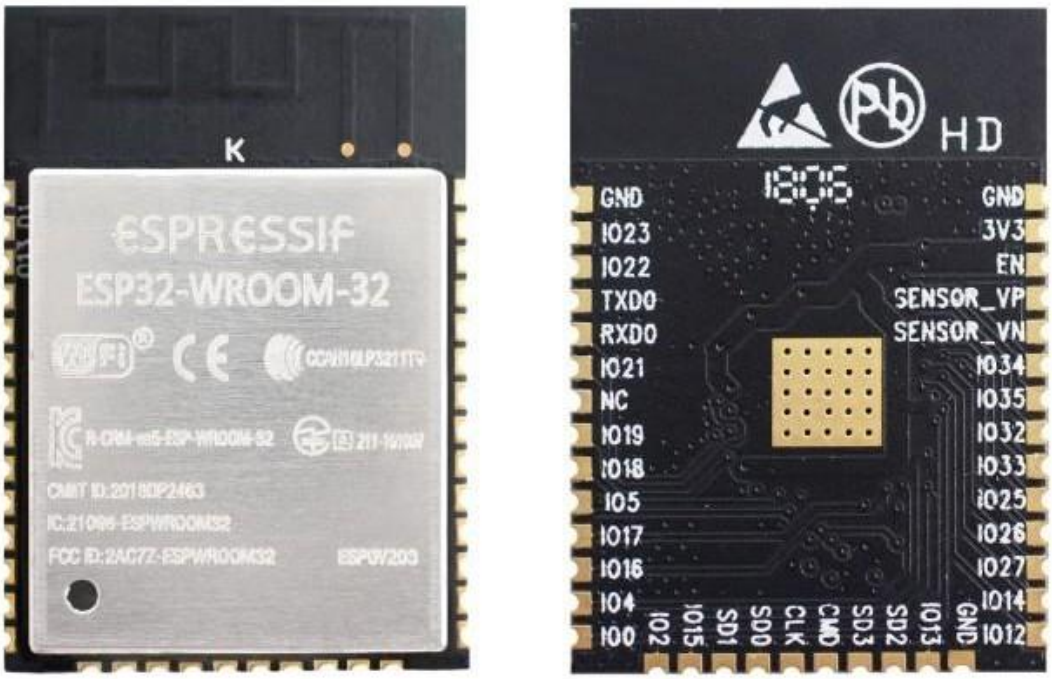


FIG 4.2 ESP32-WROOM-32

4.1.3 SPECIFICATIONS OF ESP32:

ESP32 has a lot more features than ESP8266 and it is difficult to include all the specifications in this Getting Started with ESP32 guide. So, I made a list of some of the important specifications of ESP32 here. But for complete set of specifications, I strongly suggest you to refer to the Datasheet.

- Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz
- 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM.
- Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.
- Support for both Classic Bluetooth v4.2 and BLE specifications.
- 34 Programmable GPIOs.
- Up to 18 channels of 12-bit SAR ADC and 2 channels of 8-bit DAC
- Serial Connectivity include 4 x SPI, 2 x I^2C, 2 x I^2S, 3 x UART.
- Ethernet MAC for physical LAN Communication (requires external PHY).
- 1 Host controller for SD/SDIO/MMC and 1 Slave controller for SDIO/SPI.
- Motor PWM and up to 16-channels of LED PWM.
- Secure Boot and Flash Encryption.
- Cryptographic Hardware Acceleration for AES, Hash (SHA-2), RSA, ECC and RNG.

4.2 POWER SUPPLY:

4.2.1 ADAPTER:

An adapter is a device that converts attributes of one electrical device or system to those of an otherwise incompatible device or system. Some modify power or signal attributes, while others merely adapt the physical form of one electrical connector to another. In a computer, an adapter is often built into a card that can be inserted into a slot on the computer's motherboard. The card adapts information that is exchanged between the computer's microprocessor and the devices that the card supports.

4.2.2 PRODUCT DESCRIPTION:

An electric power adapter may enable connection of a power plug, sometimes called, used in one region to an AC power socket used in another, by offering connections for the disparate contact arrangements, while not changing the voltage. An AC adapter, also called a "recharger", is a small power supply that changes household electric current from distribution voltage) to low voltage DC suitable for consumer electronics. Some modify power or signal attributes, while others merely adapt the physical form of one electrical connector to another. For computers and related items, one kind of serial port adapter enables connections between 25-contact and nine-contact connectors, but does not affect electrical power- and signaling-related attributes.

FIG 4.3 ADAPTER

4.2.3 FEATURES:

- Output current:1A
- Supply voltage: 220-230VAC
- Output voltage: 12VDC
- Reduced costs

- Increased value across front-office and back-office functions
- Access to current, accurate, and consistent data
- It generates adapter metadata as WSDL files with J2CA extension.

4.2.4 APPLICATIONS:

- Back-end systems which need to send purchase order data to oracle applications send it to the integration service via integration server client.
- SMPS applications.

4.3 ZIGBEE:

4.3.1 GENERAL DESCRIPTION:

ZigBee is an IEEE 802.15.4-based specification for a suite of high-level communication protocols used for wireless networking. It is a wireless technology developed as an open global standard to address the unique needs of low-cost, low-power wireless M2M networks. ZigBee (CC2500) is a low-cost true single chip 2.4 GHz transceiver designed for very low power wireless applications. The RF transceiver is integrated with a highly configurable baseband modem.

4.3.2 PRODUCT DESCRIPTION:

ZigBee devices are required to conform to the IEEE 802.15.4-2003 Low-Rate Wireless Personal Area Network (LR-WPAN) standard. The standard specifies the lower protocol layers are the physical layer (PHY), and the Media Access Control portion of the data link layer (DLL). The technology defined by the ZigBee specification is intended to be simpler and less expensive than other wireless personal area networks (WPANs), such

as Bluetooth or Wi-Fi. Its low power consumption limits transmission distances to 10–100 meters line-of sight, depending on power output and environmental characteristics. ZigBee devices can transmit data over long distances by passing data through a mesh network of intermediate devices to reach more distant ones. ZigBee is typically used in low data rate applications that require long battery life and secure networking. ZigBee has a defined rate of 250 kbit/s, best suited for intermittent data transmissions from a sensor or input device.

ZIGBEE 2.4GHZ TI CC2500

FIGURE 4.4 ZIGBEE 2.4GHZ TI CC2500

4.3.3 FEATURES:

- Supply voltage: 5v DC
- Detection range: (10-30) m
- RS232 Output
- TTL uart also provided
- Frequency: 2.4GHz
- Tx and Rx Status LEDs
- Low power

4.3.4 APPLICATIONS:

- Lighting controls
- Switching
- Wireless keyboard and mouse
- Consumer electronics

4.4 DC VOLTAGE SENSOR:

4.4.1 GENERAL DESCRIPTION:

A DC voltage sensor is going to be able to determine and even monitor and measure the voltage supply. It is then able to take those measurements and turn them into a signal that one will then be able to read. The signal will often go into a specialized electronic device for recording, but sometimes, an observer will be present to manually read the sensor output. For reasons of accuracy, if more than one Voltage Sensor is being used in a circuit, ensure that they share a common earth. DC voltage sensor works according to Magnetic Modulation and is designed for DC voltage measurement. The output signal of this sensor is proportional to the input DC voltage. It can be used for continuous dc voltage monitoring of the system.

4.4.2 PRODUCT DESCRIPTION:

DC Voltage Sensors are used to measure the potential difference between the ends of an electrical component. This can be used to measure the DC voltage in the circuits. The sensor is mechanically fixed by soldering the secondary circuit pins to the PCB. The primary connection can also be integrated in the sensor. Pulsating voltage with a galvanic

insulation between primary and secondary circuits. The voltage detector indicates the presence of a voltage higher than a limit. The Voltage Sensors are equipped with a micro controller that greatly improves the sensor accuracy, precision and consistency of the readings. They are supplied calibrated and the stored calibration (in Volts) is automatically loaded when the Voltage Sensor is connected.

4.4.3 FEATURES:

- Under-voltage, over-voltage, or voltage band models.
- Powered from sensing input lines or from separate DC supply.
- Available with time delays on pull-in and/or drop-out or with customized.
- voltage-time trip curves.

4.4.4 APPLICATIONS:

- Power generation unit
- Power transmission unit
- Industry and other applications

4.5 16×2 LCD:

4.5.1 GENERAL DESCRIPTION:

LCD stands for liquid crystal display. They come in many sizes 8x1, 8x2, 10x2, 16x1, 16x2, 16x4, 20x2, 20x4, 24x2, 30x2, 32x2, 40x2 etc. Many multinational companies like Philips Hitachi Panasonic make their own special kind of LCD'S to be used in their products. All the LCD'S performs the same functions (display characters numbers special characters ASCII characters etc.) Their programming is also same and

they all have same 14 pins (0-13) or 16 pins (0 to 15). Alphanumeric displays are used in a wide range of applications, including palmtop computers, word processors, photocopiers, point of sale terminals, medical instruments, cellular phones, etc.

4.5.2 PRODUCT DESCRIPTION:

This is an LCD Display designed for E-blocks. It is a 16 character, 2-line alphanumeric LCD display connected to a single 9-way D-type connector. This allows the device to be connected to most E-Block I/O ports. The LCD display requires data in a serial format, which is detailed in the user guide below. The display also requires a 5V power supply. Please take care not to exceed 5V, as this will cause damage to the device. The 5V is best generated from the E-blocks Multi programmer or a 5V fixed regulated power supply. The 16 x 2 intelligent alphanumeric dot matrix displays are capable of displaying 224 different characters and symbols. A full list of the characters and symbols is printed on pages 7/8 (note these symbols can vary between brand of LCD used). This booklet provides all the technical specifications for connecting the unit, which requires a single power supply (+5V).

FIG 4.5 16 x 2 LCD

4.5.3 FEATURES:

- Input voltage: 5v
- E-blocks compatible.
- Low cost
- Compatible with most I/O ports in the E-Block range.
- Ease to develop programming code using Flow code icons.

4.5.4 APPLICATIONS:

- Monitoring

4.6 L293D DC MOTOR DRIVER MODULE:

The project designed around L293D IC. The L293D device is quadruple high-current half-H driver. The 293D is designed to provide bidirectional drive current up to 600mA a voltage from 5V to 36V. L293D Adapter Board can be used as dual DC motor driver or bipolar stepper motor driver. Useful in robotics application, bidirectional DC motor controller and stepper motor driver. Separate logic supply to reduce dissipation. L293D includes the output clamping diodes for protections.

4.6.1 SPECIFICATIONS:

- Motor/Logic supply 5 to 36 V
- Logic controls input 7 VDC max
- Inhibit facility/enable
- High Noise immunity

- Capable of delivering output current up to 600 mA per channel
- The control/interface lines are accessible with Berg connector
- Header connector for motor and supply connection
- PCB dimensions 36 mm x 24 mm

4.6.2 L293D IC PIN OUT:

The L293D is a 16 pin IC, with eight pins, on each side, to controlling of two DC motor simultaneously. There are 4 INPUT pins, 4 OUTPUT pins and 2 ENABLE pin for each motor.

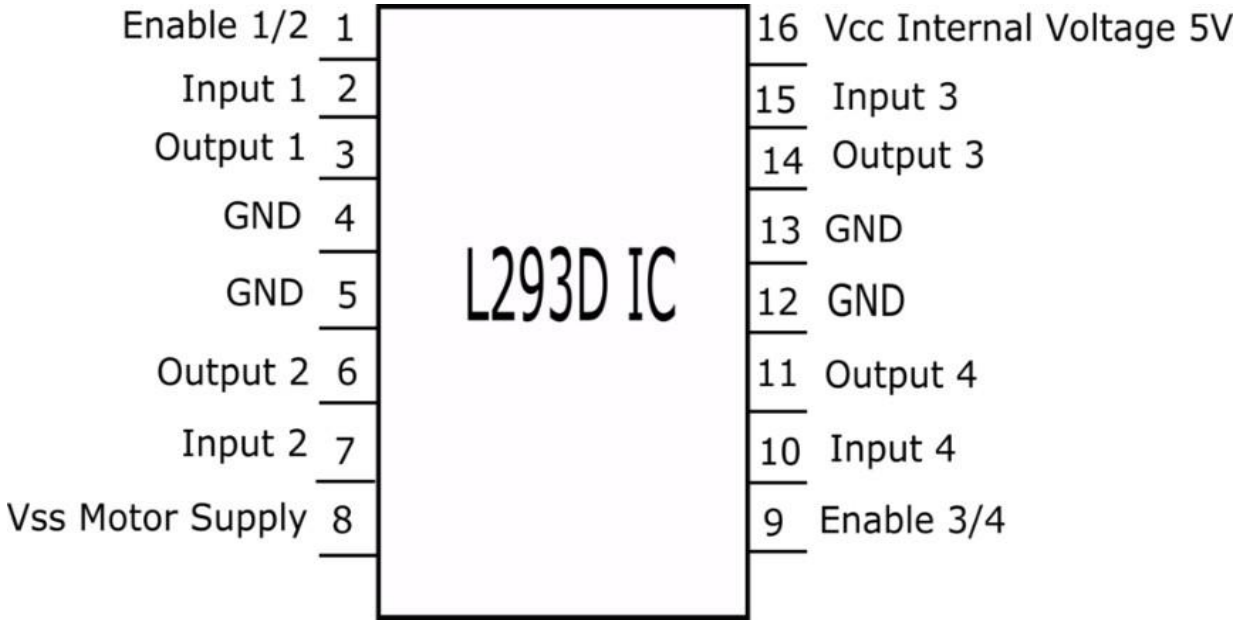


FIG 4.6 L293D IC PIN OUT

4.6.3 WORKING MECHANISM:

Rotation of motor depends on Enable Pins. When Enable 1/2 is HIGH, motor connected to left part of IC will rotate according to following manner:

Input 1	Input 2	Result
0	0	Stop
0	1	Anti Clockwise
1	0	Clockwise
1	1	Stop

4.7 SOFTWARE DESCRIPTION:

4.7.1 EMBEDDED C:

High-level language programming has long been in use for embedded-systems development. However, assembly programming still prevails, particularly for digital-signal processor (DSP) based systems. DSPs are often programmed in assembly language by programmers who know the processor architecture inside out. The key motivation for this practice is performance, despite the disadvantages of assembly programming when compared to high-level language programming. If the video decoding takes 80 percent of the CPU-cycle budget instead of 90 percent, for instance, there are twice as many cycles available for audio processing. This coupling of performance to end-user features is characteristic of many of the real-time applications in which DSP processors are applied. DSPs have a highly specialized architecture to achieve the performance requirements for

signal processing applications within the limits of cost and power consumption set for consumer applications. Unlike a conventional Load-Store (RISC) architecture, DSPs have a data path with memory-access units that directly feed into the arithmetic units. Address registers are taken out of the general-purpose register file and placed next to the memory units in a separate register file. A further specialization of the data path is the coupling of multiplication and addition to form a single cycle Multiply-accumulate unit (MAC). It is combined with special-purpose accumulator registers, which are separate from the general-purpose registers. Data memory is segmented and placed close to the MAC to achieve the high bandwidths required to keep up with the streamlined data path. Limits are often placed on the extent of memory-addressing operations. The localization of resources in the data path saves many data movements that typically take place in a Load-Store architecture. The most important, common arithmetic extension to DSP architectures is the handling of saturated fixed-point operations by the arithmetic unit. Fixed-point arithmetic can be implemented with little additional cost over integer arithmetic. Automatic saturation (or clipping) significantly reduces the number of control-flow instructions needed for checking overflow explicitly in the program. Changes in technological and economic requirements make it more expensive to continue programming DSPs in assembly. Staying with the mobile phone as an example, the signal-processing algorithms required become increasingly complex. Features such as stronger error correction and encryption must be added. Communication protocols become more sophisticated and require much more code to implement. In certain markets, multiple protocol stacks are implemented to be compatible with multiple service providers. In addition, backward compatibility with older protocols is needed to stay synchronized with provider networks that are in a slow process of upgrading. Today, most embedded processors are offered with C compilers. Despite this, programming DSPs is still done in assembly for the signal processing parts or, at best, by using assembly-written libraries supplied by manufacturers. The key reason for this is that although the architecture is well matched to the requirements of the signal-processing application, there is no way to express the algorithms efficiently and in a natural way in

Standard C. Saturated arithmetic. For example, is required in many algorithms and is supplied as a primitive in many DSPs. However, there is no such primitive in Standard C. To express saturated arithmetic in C requires comparisons, conditional statements, and correcting assignments. Instead of using a primitive, the operation is spread over a number of statements that are difficult to recognize as a single primitive by a compiler.

4.7.2 DESCRIPTION:

Embedded C is designed to bridge the performance mismatch between Standard C and the embedded hardware and application architecture. It extends the C language with the primitives that are needed by signal-processing applications and that are commonly provided by DSP processors. The design of the support for fixed-point data types and named address spaces in Embedded C is based on DSP-C. DSP-C [1] is an industry-designed extension of C with which experience was gained since 1998 by various DSP manufacturers in their compilers. For the development of DSP-C by ACE (the company three of us work for), cooperation was sought with embedded-application designers and DSP manufacturers. The Embedded C specification extends the C language to support freestanding embedded processors in exploiting the multiple address space functionality, user-defined named address spaces, and direct access to processor and I/O registers. These features are common for the small, embedded processors used in most consumer products. The features introduced by Embedded C are fixed-point and saturated arithmetic, segmented memory spaces, and hardware I/O addressing. The description we present here addresses the extensions from a language-design perspective, as opposed to the programmer or processor architecture perspective.

4.7.3 MULTIPLE ADDRESS SPACES:

Embedded C supports the multiple address spaces found in most embedded systems. It provides a formal mechanism for C applications to directly access (or map onto) those

individual processor instructions that are designed for optimal memory access. Named address spaces use a single, simple approach to grouping memory locations into functional groups to support MAC buffers in DSP applications, physical separate memory spaces, direct access to processor registers, and user-defined address spaces. The Embedded C extension supports defining both the natural multiple address space built into a processor's architecture and the application-specific address space that can help define the solution to a problem. Embedded C uses address space qualifiers to identify specific memory spaces in variable declarations. There are no predefined keywords for this, as the actual memory segmentation is left to the implementation. As an example, assume that **X** and **Y** are memory qualifiers. The definition:

```
X int a[25] ;
```

Means that **a** is an array of 25 integers, which is located in the **X** memory. Similarly (but less common):

```
X int * Y p;
```

Means that the pointer **p** is stored in the **Y** memory. This pointer points to integer data that is located in the **X** memory. If no memory qualifiers are used, the data is stored into unqualified memory. For proper integration with the C language, a memory structure is specified, where the unqualified memory encompasses all other memories. All unqualified pointers are pointers into this unqualified memory. The unqualified memory abstraction is needed to keep the compatibility of the **void *** type, the **NULL** pointer, and to avoid duplication of all library code that accesses memory through pointers that are passed as parameters.

4.7.4 NAMED REGISTERS:

Embedded C allows direct access to processor registers that are not addressable in any of the machine's address spaces. The processor registers are defined by the compiler-specific, named-register, storage class for each supported processor. The processor registers are declared and used like conventional C variables (in many cases volatile variables). Developers using Embedded C can now develop their applications, including direct access to the condition code register and other processor-specific status flags, in a high-level language, instead of inline assembly code. Named address spaces and full processor access reduces application dependency on assembly code and shifts the responsibility for computing data types, array and structure offsets, and all those things that C compilers routinely and easily do from developers to compilers.

4.7.5 I/O HARDWARE ADDRESSING:

The motivation to include primitives for I/O hardware addressing in Embedded C is to improve the portability of device-driver code. In principle, a hardware device driver should only be concerned with the device itself. The driver operates on the device through device registers, which are device specific. However, the method to access these registers can be very different on different systems, even though it is the same device that is connected. The I/O hardware access primitives aim to create a layer that abstracts the system-specific access method from the device that is accessed. The ultimate goal is to allow source-code portability of device drivers between different systems. In the design of the I/O hardware-addressing interface, three requirements needed to be fulfilled:

1. The device-drive source code must be portable.
2. The interface must not prevent implementations from producing machine code that is as efficient as other methods.

3. The design should permit encapsulation of the system-dependent access method.

The design is based on a small collection of functions that are specified in the <iohw.h> include file. These interfaces are divided into two groups; one group provides access to the device, and the second group maintains the access method abstraction itself.To access the device, the following functions are defined by Embedded C:

```
unsigned int iord(  ioreg_designator );

void iowr( ioreg_designator, unsigned int value );

void ioor( ioreg_designator, unsigned int value );

void ioand( ioreg_designator, unsigned int value );

void ioxor( ioreg_designator, unsigned int value );
```

These interfaces provide read/write access to device registers, as well as typical methods for setting/resetting individual bits. Variants of these functions are defined (with **buf** appended to the names) to access arrays of registers. Variants are also defined (with l appended) to operate with **long** values. All of these interfaces take an I/O register designator **ioreg_designator** as one of the arguments. These register designators are an abstraction of the real registers provided by the system implementation and hide the access method from the driver source code. Three functions are defined for managing the I/O register designators. Although these are abstract entities for the device driver, the driver does have the obligation to initialize and release the access methods. These functions do not access or initialize the device itself because that is the task of the driver. They allow, for example, the operating system to provide a memory mapping of the device in the user address space.

```
void iogroup_acquire( iogrp_designator );

void iogroup_release( iogrp_designator );

void iogroup_map( iogrp_designator, iogrp_designator );
```

The **iogrp_designator** specifies a logical group of I/O register designators; typically, this will be all the registers of one device. Like the I/O register designator, the I/O group designator is an identifier or macro that is provided by the system implementation. The map variant allows cloning of an access method when one device driver is to be used to access multiple identical devices.

4.7.6 EMBEDDED C PORTABILITY:

By design, a number of properties in Embedded C are left implementation defined. This implies that the portability of Embedded C programs is not always guaranteed. Embedded C provides access to the performance features of DSPs. As not all processors are equal, not all Embedded C implementations can be equal for example, suppose an application requires 24-bit fixed-point arithmetic and an Embedded C implementation provides only 16 bits because that is the native size of the processor. When the algorithm is expressed in Embedded C, it will not produce outputs of the right precision. In such a case, there is a mismatch between the requirements of the application and the capabilities of the processor. Under no circumstances, including the use of assembly, will the algorithm run efficiently on such a processor. Embedded C cannot overcome such discrepancies. Yet, Embedded C provides a great improvement in the portability and software engineering of embedded applications. Despite many differences between performance-specific

processors, there is a remarkable similarity in the special-purpose features that they provide to speed up applications. Writing C code with the low-level processor-specific support may at first appear to have many of the portability problems usually associated with assembly code. In the limited experience with porting applications that use Embedded C extensions, an automotive engine controller application (about 8000 lines of source) was ported from the eTPU, a 24-bit special-purpose processor, to a general-purpose 8-bit Freescale 68S08 with about a screen full of definitions put into a single header file. The porting process was much easier than expected. For example, variables that had been implemented on the processor registers were ported to unqualified memory in the general-purpose microprocessor by changing the definitions in the header definition and without any actual code modifications. The exercise was to identify the porting issues and it is clear that the performance of the special-purpose processor is significantly higher than the general-purpose target.

4.8 ARDUINO SOFTWARE (IDE):

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a portable installation. When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

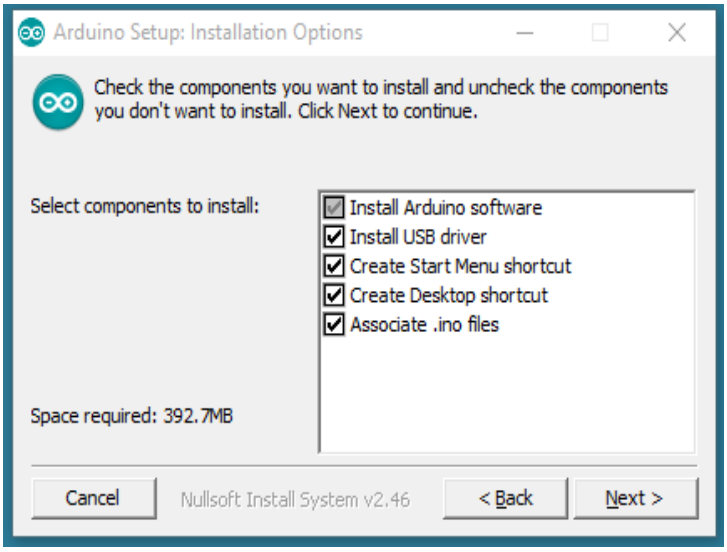


FIG 4.7 ARDUINO SOFTWARE INSTALLATION

Choose the components to install.

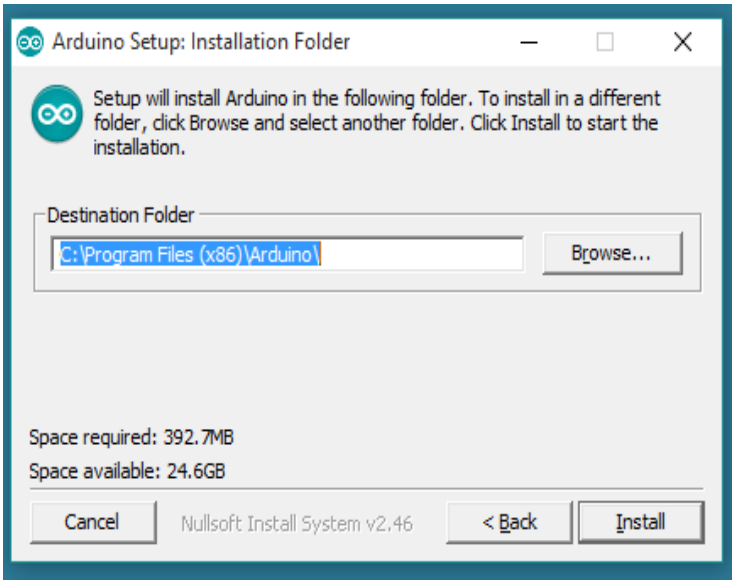


FIGURE 4.8 ARDUINO INSTALLATION DIRECTORY

Choose the installation directory (we suggest to keep the default one)

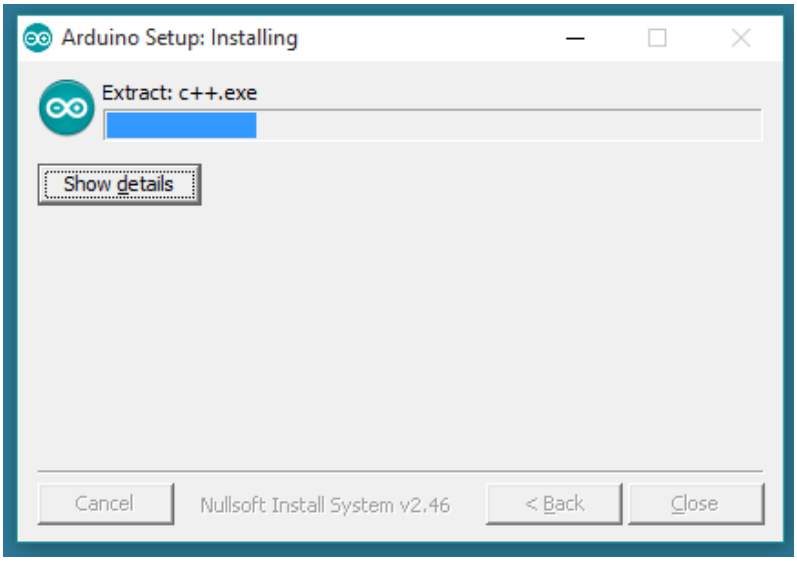


FIG 4.9 ARDUINO SOFTWARE EXTRACTION AND INSTALL

The process will extract and install all the required files to execute properly the Arduino Software (IDE).

4.8.1 ARDUINO BOOTLOADER ISSUE:

The current bootloader burned onto the Arduino UNO is not compatible with ROBOTC. In its current form, you will be able to download the ROBOTC Firmware to the Arduino UNO, but you will not able to download any user programs. The reason for this is because there is a bug in the Arduino UNO firmware that does not allow flash write commands to start at anywhere but the beginning of flash memory (0x000000). See the bottom of this page for more technical details. Because ROBOTC is not able to burn a new bootloader as of today, you will need to use the Arduino's Open-Source language with a modified bootloader file to re-burn your bootloader on your Arduino UNO boards. The enhanced bootloader is backwards compatible with the original one. That means you'll still be able to program it through the Arduino programming environment as before, in addition to ROBOTC for Arduino.

4.8.2 HARDWARE NEEDED:

To burn a new version of the Arduino bootloader to your UNO, you'll need an AVR ISP Compatible downloader.

USING AN AVR ISP (In System Programmer)

- Your Arduino UNO (to program)
- An AVR Programmer such as the AVR Pocket Programmer
- An AVR Programming Cable (the pocket programmer comes with one)

If you have extra Arduino boards, but no ISP programmer, SparkFun.com has a cool tutorial on how to flash a bootloader using an Arduino as an ISP.

USING ANOTHER ARDUINO AS AN ISP

- Your Arduino UNO (to program)
- A Working Arduino (doesn't matter what kind)
- Some Male-to-Male Jumper Cables

For instructions on this method, take a look at the SparkFun.com website: http://www.sparkfun.com/tutorials/247

4.8.3 SOFTWARE NEEDED:

ROBOTC is not currently able to burn a bootloader onto an Arduino board, so you'll need to download a copy of the latest version of the Arduino Open-Source programming language.

- Arduino Official Programming Language - Download Page

In addition, you'll need the ROBOTC modified bootloader. You can download that here:

- ROBOTC Modified UNO Bootloader - Modified Bootloader

4.8.4 BOOTLOAD DOWNLOAD INSTRUCTIONS:

- Download the Arduino Open Source Software and a copy of the Modified Bootloader File
- Copy the Modified Bootloader File into the /Arduino-1.0/hardware/Arduino/bootloaders/stk500v2/ and overwrite the existing bootloader.

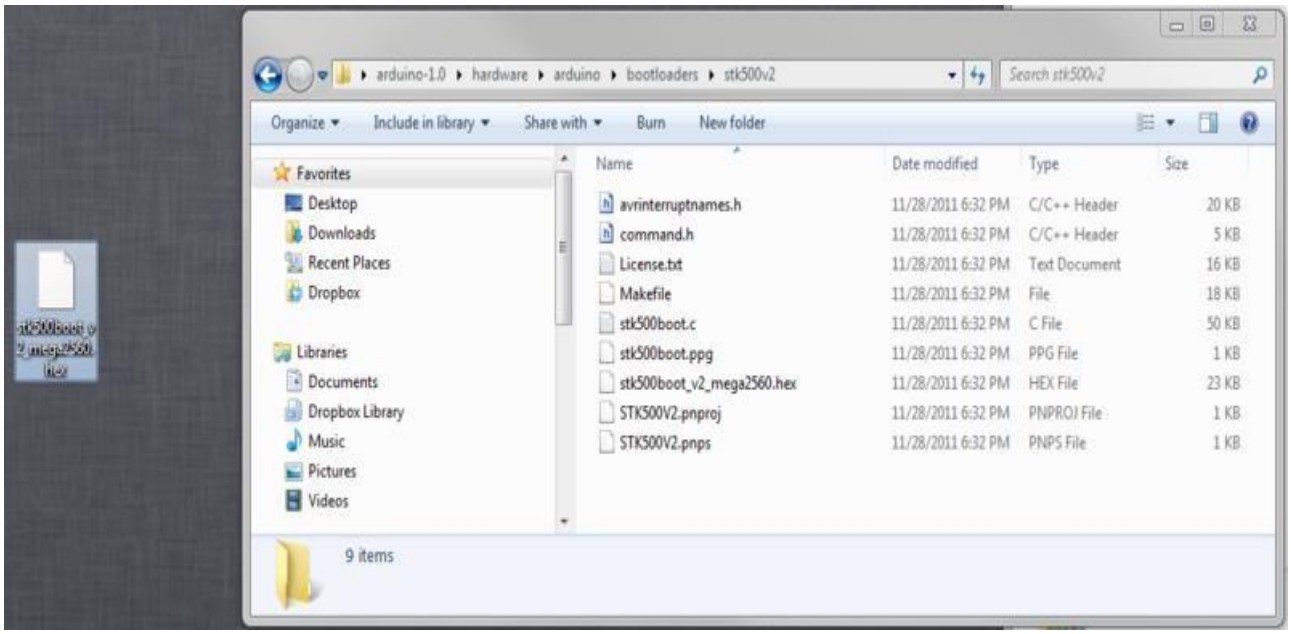


FIG 4.10 BOOTLOADER DOWNLOAD INSTRUCTIONS

- Power up your Arduino UNO (either via USB or external power)
- Plug in your AVR ISP Programmer to your computer (make sure you have any required drivers installed)
- Connect your AVR ISP Programmer into your Arduino UNO Board via the ISP Header (the 2x3 header pins right above the Arduino Logo)
- Launch the Arduino Open-Source Software

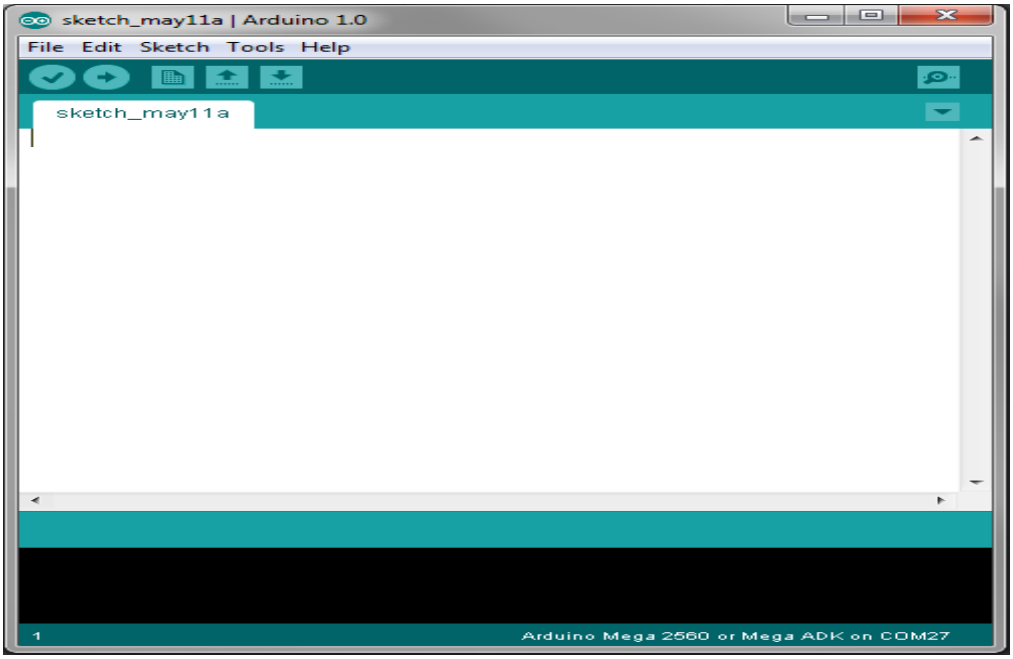


FIG 4.11 ARDUINO SOFTWARE

- Change your settings in the Arduino Software to look for an Arduino UNO.
- Change your settings in the Arduino Software to select your ISP Programmer Type (Check your programmer's documentation for the exact model)

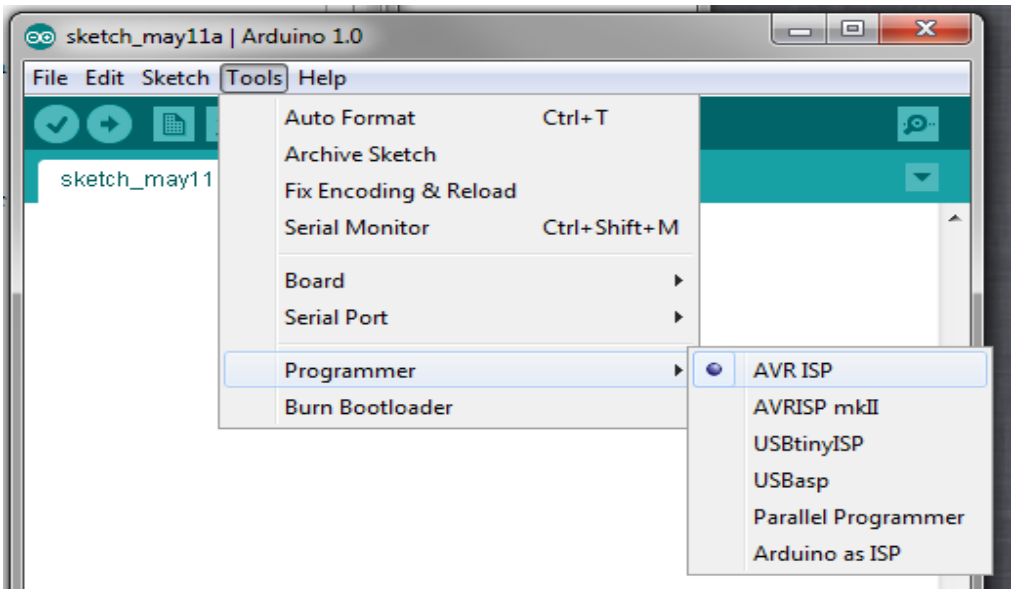


FIG 4.12 MODIFIED BOOTLOADER

- Select the "Burn Bootloader" option under the "Tools" menu. The modified bootloader will now be sent to your Arduino. This typically take a minute or so.

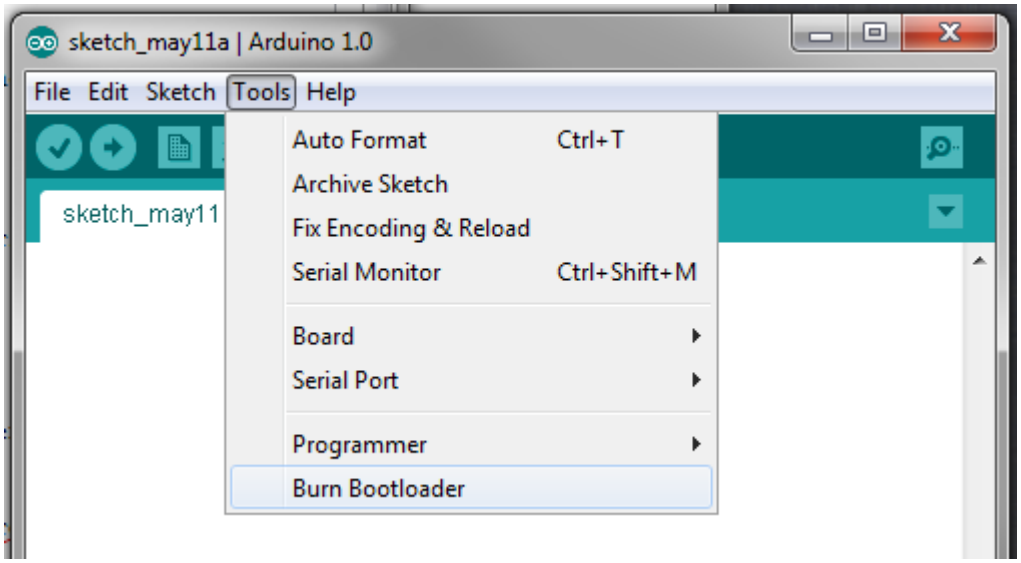


FIG 4.13 ARDUINO UNO WITH ROBOTC

- You should be all set to download ROBOTC firmware and start using your Arduino UNO with ROBOTC.

4.8.5 TECHNICAL DETAILS:

The Arduino Boot loader sets the "erase Address" to zero every time the boot loader is called. ROBOTC called the "Load Address" command to set the address in which we want to write/verify when downloading program. When writing a page of memory to the Arduino, the Arduino boot loader will erase the existing page and write a whole new page. In the scenario of downloading firmware, everything is great because the Erase Address and the Loaded Address both start at zero. In the scenario of writing a user program, we start writing at memory location 0x7000, but the Boot loader erases information starting at location zero because the "Load Address" command doesn't update where to erase. Our

modification is to set both the Load Address and the Erase Address so the activity of writing a user program doesn't cause the firmware to be accidentally erased!

4.8.6 PRODUCT OVERVIEW:

The Arduino UNO is a microcontroller board based on the Arduino UNO (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

4.8.7 TECHNICAL SPECIFICATION:

EAGLE files: arduino-reference-design.zip Schematic: arduino-schematic.pdf

4.8.8 SUMMARY:

Microcontroller	**Arduino UNO**
Operating Voltage	5V
Input Voltage(recommended)	7-
12V input Voltage(limits)	6-
20V	
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40mA

DC Current for3.3VPin	50mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8KB
EEPROM	4KB
Lock Speed	16MHz

The Arduino UNO can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.**A3.3voltsupplygeneratedbytheon-boardregulator. Maximum current draw is 50mA.

- **GND.** Ground pins.

The ATmega has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library). Each of the 54 digital pins on the Mega can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a changing value. See the attach Interrupt() function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the analog Write() function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I^2C: 20 (SDA) and 21 (SCL).** Support I^2C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I^2C pins on the Duemilanove.

The Arduino UNO has 16 analog inputs, each of which provide 10 bits of resolution (i.e., 1024 different values). By default, they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and analog Reference () function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with analog Reference().
- **ESET.** Bring this line LOW to reset the microcontroller. Typically used to add reset button to shields which block the one on the board.

4.8.9 COMMUNICATION:

The Arduino UNO has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The Arduino UNO provides four hardware UARTs for TTL(5V) serial communication. An ATmega on the board channels one of these over USB and provides a virtual comport to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and1). A Software Serial library allows for serial communication on any of the digital pins. The Arduino UNO also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation on the Wiring website for details. To use the SPI communication, please see the Arduino UNO datasheet.

4.8.10 PROGRAMMING:

The Arduino UNO can be programmed with the Arduino software (download).

For details, see the reference and tutorials. The Arduino UNO on the Arduino UNO comes pre burned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

4.8.11 AUTOMATIC (SOFTWARE) RESET:

Rather than requiring a physical press of the reset button before an upload, the Arduino UNO is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the Arduino UNO via a 100 nano farad capacitor. When this line is asserted, the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Arduino UNO is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the UNO. While it is programmed to ignore malformed data (i.e., anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110-ohm resistor from 5V to the reset line; see this forum thread for details.

CHAPTER 5

WORKING MODEL

5.1 HARDWARE KIT:

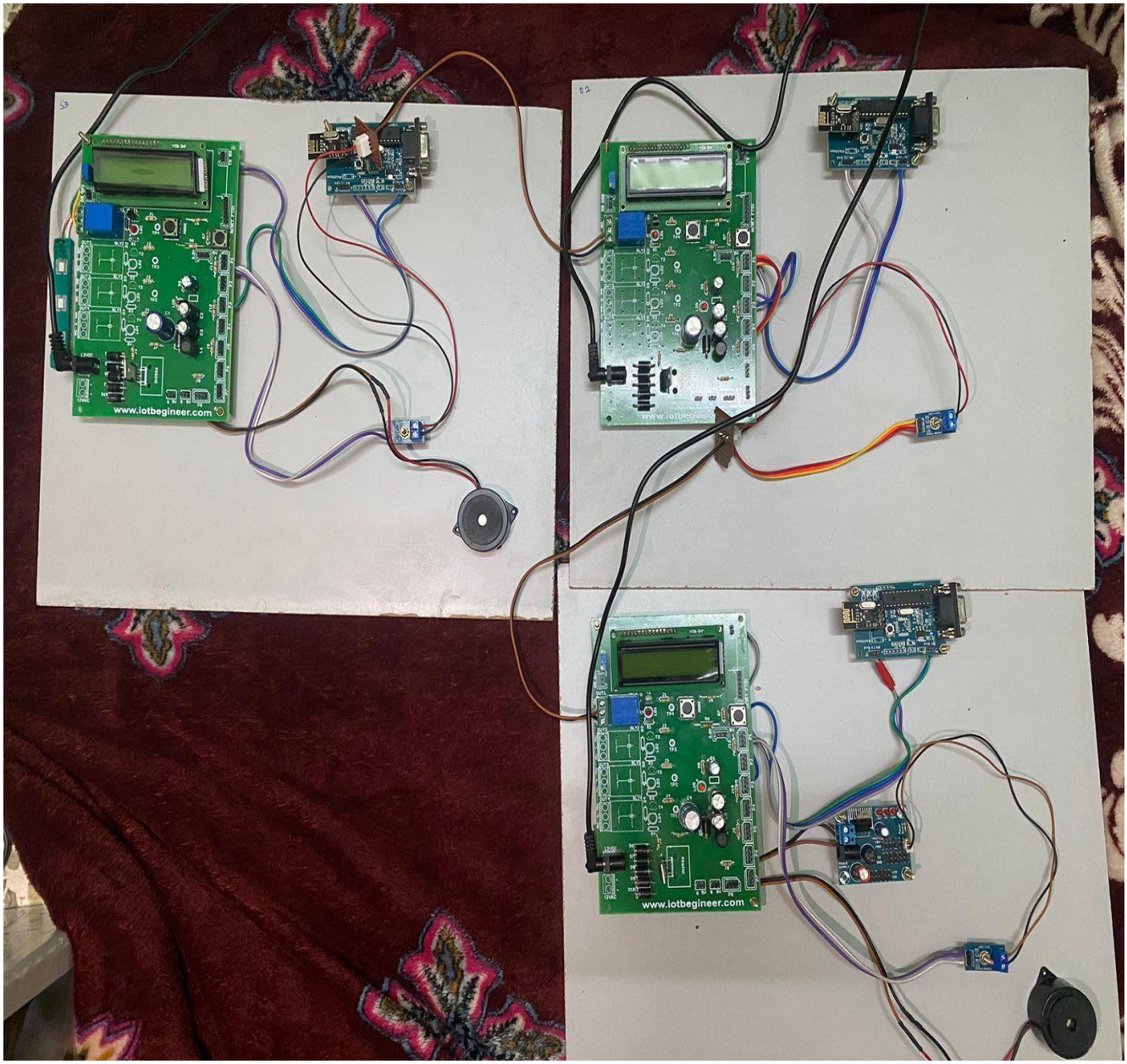


FIG 5 HARDWARE KIT

5.2 HARDWARE OUTPUT:

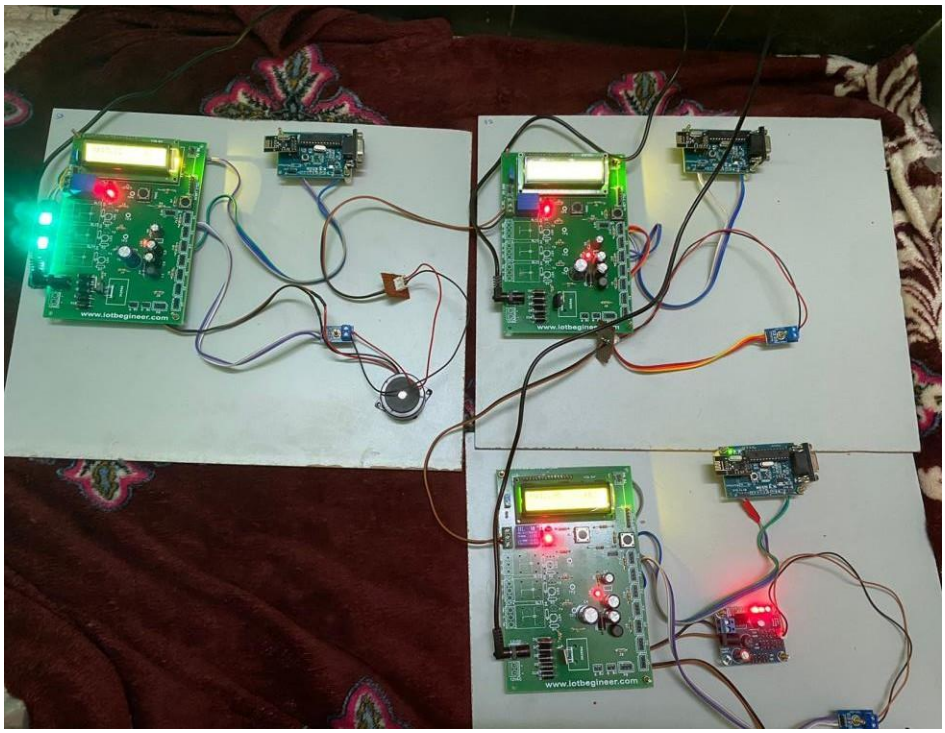


FIG 5.1 HARDWARE KIT IN WORKING ORDER

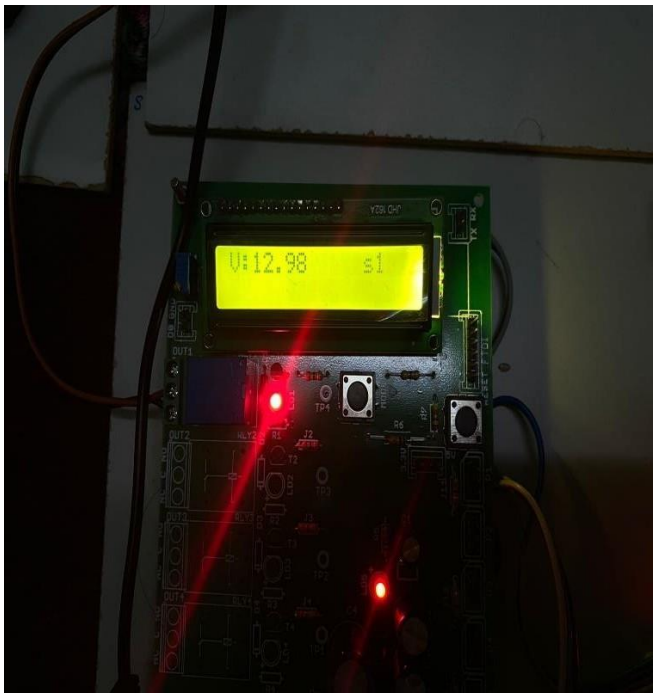
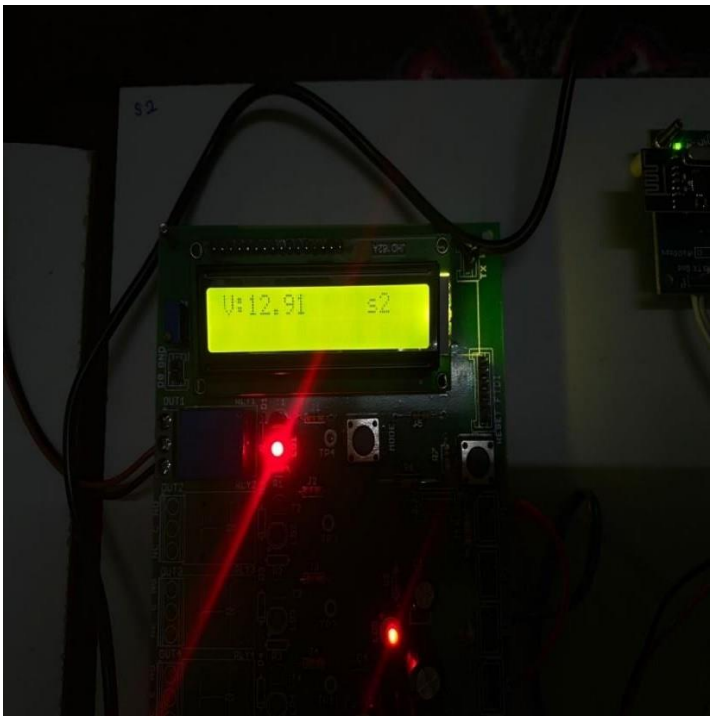
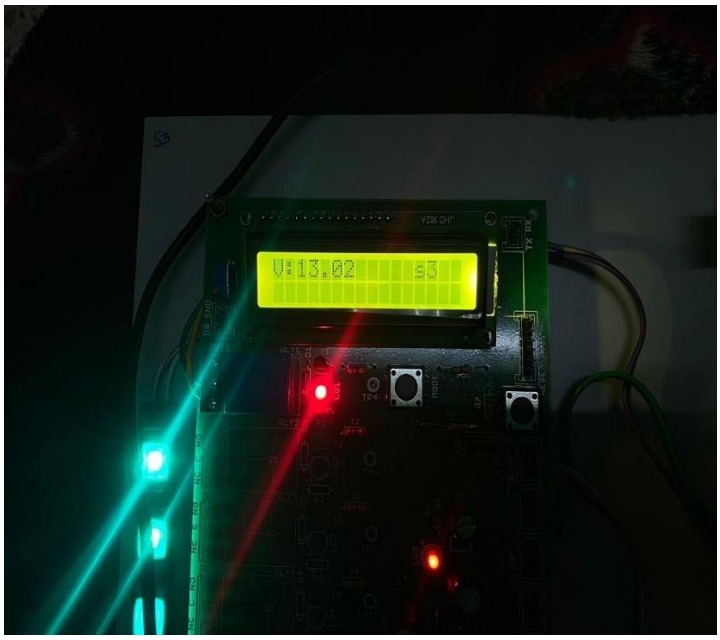


FIG 5.2 (a) TRANSMISSION LINE ONE



(b) TRANSMISSION LINE TWO



(c) TRANSMISSION LINE THREE

5.3 HARDWARE OUTPUT DURING FAULT CONDITIONS:

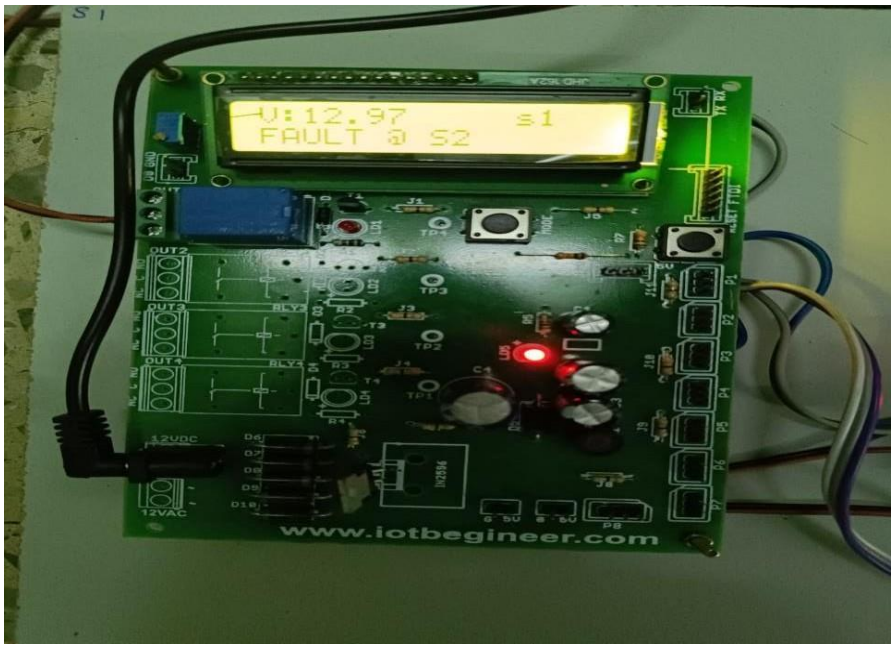
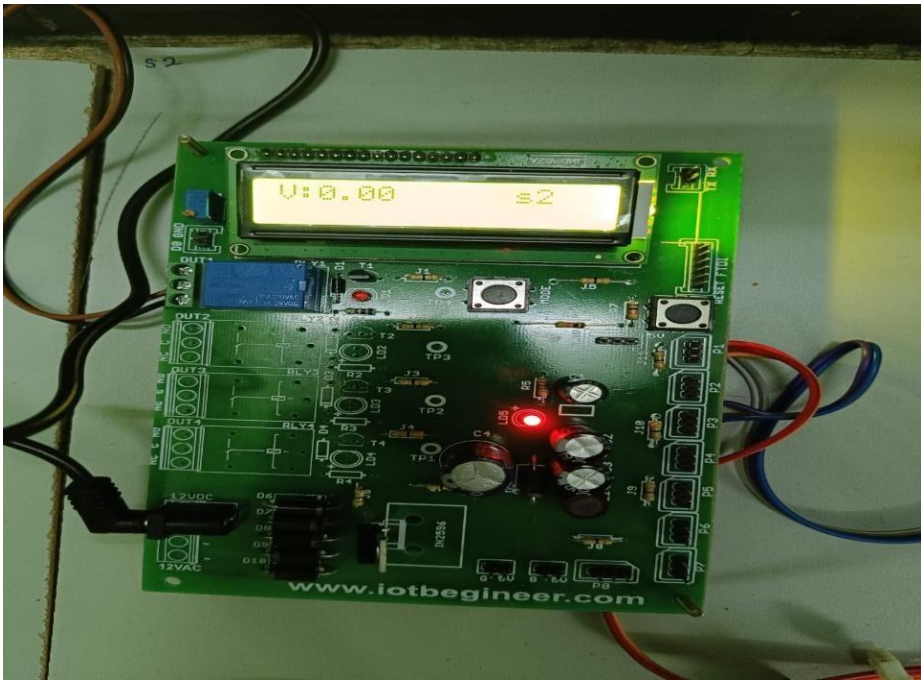


FIG 5.3 (a) TRANSMISSION LINE ONE



(b) TRANSMISSION LINE TWO WITH FAULT

(c) TRANSMISSION LINE THREE WITH FAULT

5.4 SIMULATION RESULTS:

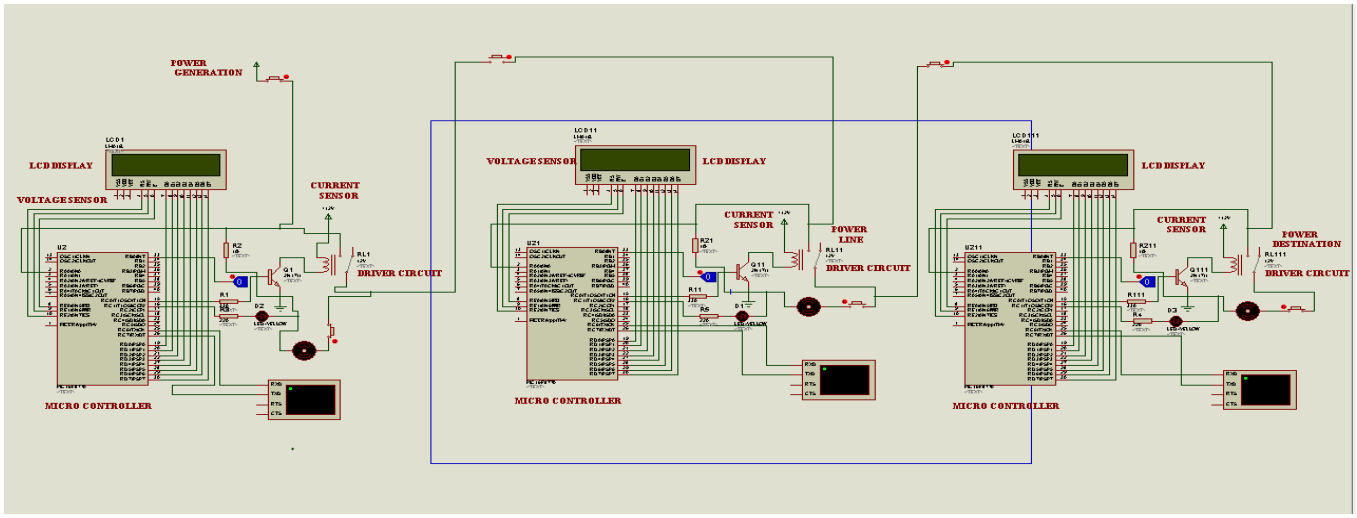


FIG 5.4 SIMULATION CIRCUIT OF THE PROPOSED SYSTEM

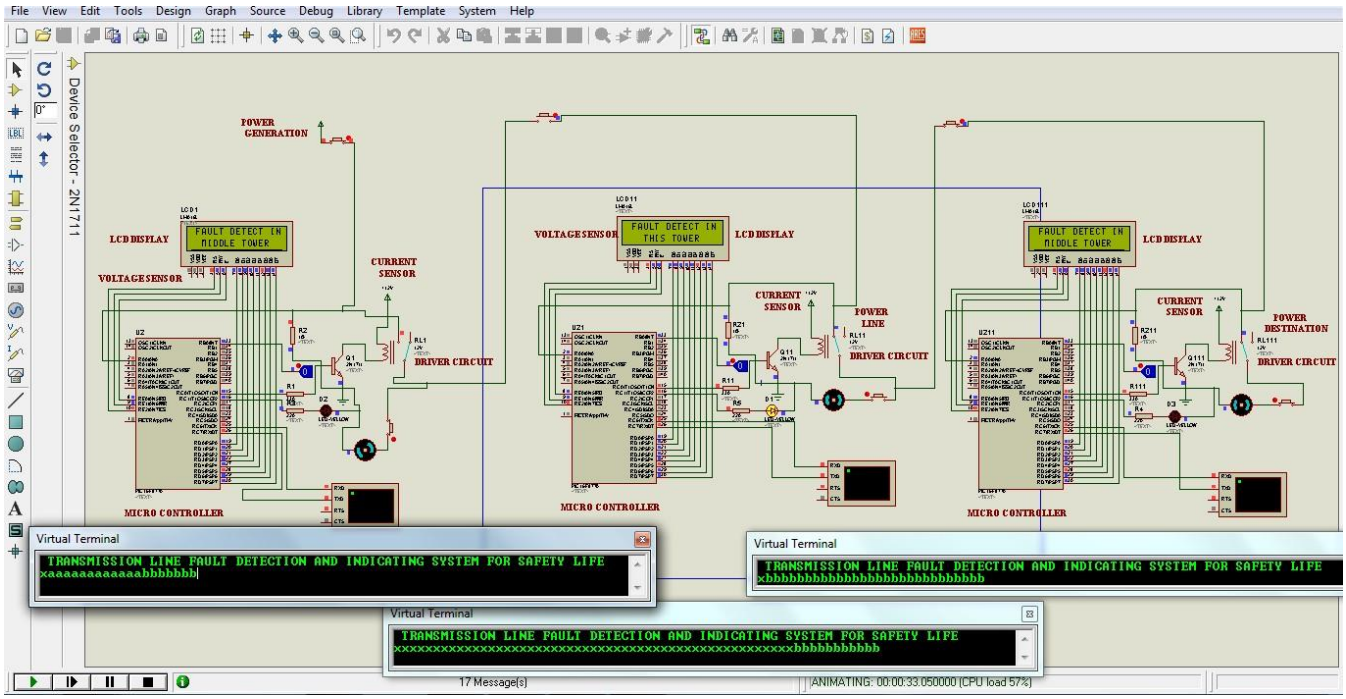


FIG 5.5 SIMULATION OUTPUT IN LCD DISPLAY

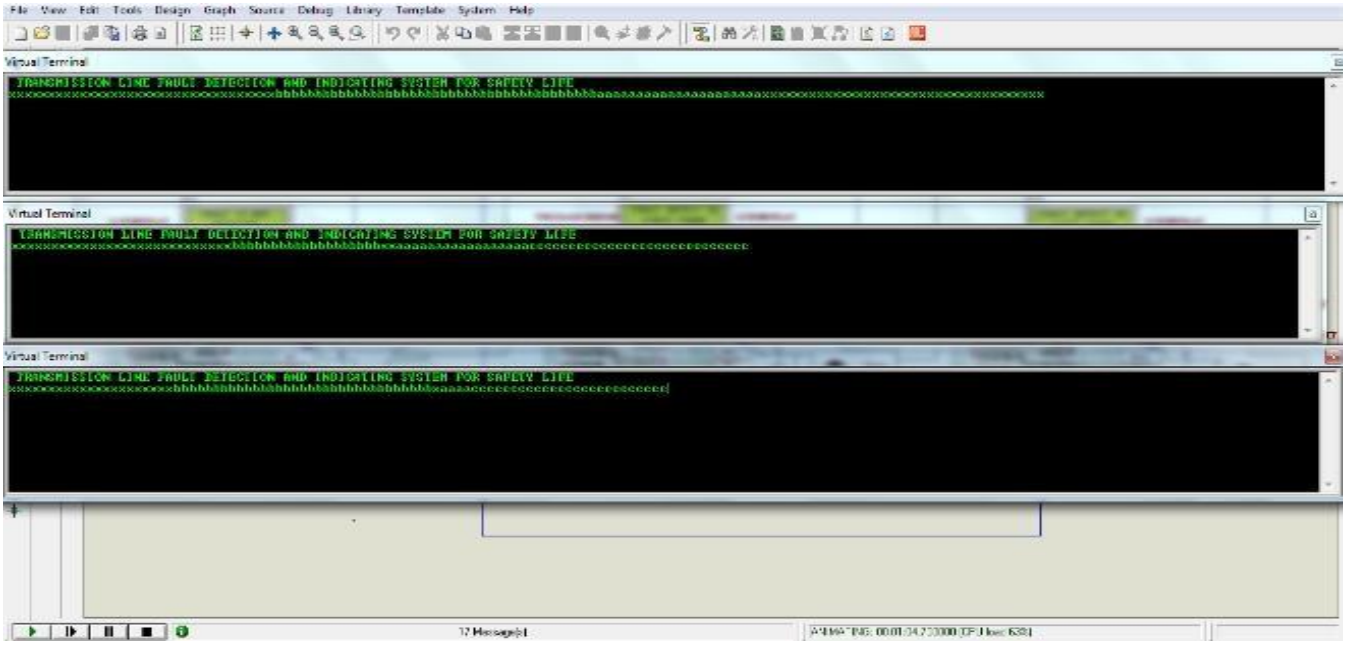


FIG 5.6 SIMULATION OUTPUT IN VIRTUAL TERMINAL

5. CONCLUSION:

In conclusion, the unique methodology for transmission line breakage detection and alerting system is an important innovation in the field of electrical engineering. By utilizing a combination of advanced sensors, machine learning algorithms, and communication technologies, this system can quickly and accurately detect breaks in transmission lines and alert operators to the issue. One of the key benefits of this methodology is its ability to reduce downtime and prevent power outages by detecting and addressing transmission line issues before they escalate into larger problems. Additionally, this system can also help utilities optimize their maintenance and repair schedules, reducing costs and improving overall system efficiency. Overall, the transmission line breakage detection and alerting system is an important development that has the potential to revolutionize the way we monitor and maintain transmission lines. As the technology continues to evolve and improve, we can expect to see even greater benefits in terms of improved reliability, increased efficiency, and reduced costs for both utilities and consumers alike.

5.1 FUTURE ENHANCEMENT:

1. Data collection: Collect data from various sources, such as sensors, weather data, historical outage data, etc. The data should be collected in real-time to enable quick analysis and detection of any abnormalities.
2. Data preprocessing: Preprocess the collected data to remove any noise and inconsistencies. This step is essential to ensure that the machine learning algorithms can accurately detect any deviations from the normal pattern.
3. Feature selection: Select the most relevant features that could help in detecting transmission line breakages. These features could include voltage levels, current levels, temperature, wind speed, humidity, and other environmental factors.

REFERENCES:

1. F. M. Albatsh, S. Mekhilef, S. Ahmad, H. Mokhlis, and M. Hassan, “Enhancing power transfer capability through flexible AC transmission system devices: a review," Frontiers of Information Technology & Electronic Engineering, vol. 16, pp. 658-678, 2015.
2. C. Chakraborty, H. H. C. and D. D.-C. Lu, "Power converters, control, and energy management for distributed generation," IEEE Transactions on Industrial Electronics, vol. 62, pp. 4466-4470, 2015.
3. J. Chivite-Zabalza, M. A. R. Vidal, P. Izurza-Moreno, G. Calvo and D. Madariaga, " A Large-Power Voltage Source Converter for FACTS Applications Combining Three-Level Neutral-Point-Clamped Power Electronic Building Blocks, “
4. S. Yang, Y. Liu, X. Wang, D. Gunasekaran, U. Karki, and F. Z. Peng, "Modulation and Control of Transformer less UPFC," IEEE Transactions on Power Electronics, vol. 31, pp. 1050-1063, 2016.
5. B. Bhattacharyya and S. Kumar, "Approach for the solution of transmission congestion with multi-type FACTS devices," in IET Generation, Transmission & Distribution, vol. 10, no. 11, pp. 2802-2809, 2016.
6. A. K. R. K. and S. P. Singh, "Congestion mitigation using UPFC," in IET Generation, Transmission & Distribution, vol. 10, no. 10, pp.2433-2442, 2016.
7. A. Rajabi-Ghahnavieh, M. Fotuhi-Firuzabad and M. Othman, "Optimal unified power flow controller application to enhance total transfer capability," in IET Generation, Transmission & Distribution, vol. 9, no.4, pp. 358-368, 2015.
8. J. Sun, H. Zheng, C. L. DeMarco and Y. Chai, "Energy Function- Based Model Predictive Control with UPFCs for Relieving Power System Dynamic Current Violation," in IEEE Transactions on Smart Grid, vol. 7, no. 6, pp. 2933-2942, 2016.
9. W. M. Lin, K. H. Lu & T. C. Ou, "Design of a novel intelligent damping controller for unified power flow controller in power system connected offshore power applications,"

in IET Generation, Transmission & Distribution, vol. 9, no. 13, pp. 1708-1717, 2015.

10. O. Rodríguez, A. Medina and G. Andersson, "Closed-form analytical characterization of non-linear oscillations in power systems incorporating unified power flow controller," in IET Generation, Transmission & Distribution, vol. 9, no. 11, pp. 1019-1032, 2015.