International Conference on Advances in Materials, Computing and Communication Technologies
In Association withInternational Journal of Scientific Research in Science and Technology
Volume 9 | Issue 1 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)

A Compact Design Approach for Plant Temperature Steady State Analysis System Using Fuzzy Logic

Meenakumari S[1], Deekshitha S[2], Roja C[2], Sri Vishnu Priya N[2], Deepika J. M[3]

[1]Assistant Professor, Department of Electrical and Electronic Engineering, Er. Perumal Manimekelai college of Engineering, Hosur, Tamil Nadu, India

[2]Department of Electrical and Electronic Engineering, Er. Perumal Manimekelai college of Engineering, Hosur, Tamil Nadu, India

ABSTRACT

The purpose of the steady state analysis of the temperature parameter is to heat the system to a specified temperature, then maintain it at that temperature in controlled manner. Fuzzy Logic Controller (FLC) is best way in which this type of precision control can be accomplished by controller. In the last decades, a lot of effort has been made in order to improve the use of environmentally friendly and renewable energy sources. In a context of small energy usage, energy harvesting takes place and thermal energy sources are one of its main energy sources because there are several unused heat sources available in the environment that may be used as renewable energy sources. To rapidly evaluate the energy potential of such thermal sources is a hard task; therefore, a way to perform this is welcome. In this work, a thermal pattern emulation system to evaluate potential thermal source in a easy way is proposed. The main characteristics of the proposed system are that it is online and remote, that is, while the thermal-source-under-test is being measured, the system is emulating it and evaluating the generated energy remotely by using wireless application protocol. The main contribution of this work was to replace the conventional Proportional Integral Derivative (PID) controller to a Fuzzy-Proportional Integral (PI) controller. In order to compare both controllers, three tests were carried out, namely: (a) step response, (b) perturbation test, (c) thermal emulation of the thermal pattern obtained from a potential thermal source: tree trucks. Experimental results show that the Fuzzy-PI controller was faster than the PID, achieving a setting time 43.26% faster, and also was more efficient with a maximum error 55% smaller than the PID.

Keywords : Fuzzy Controller, Energy Harvesting, Thermal Emulation, Wireless Application Protocol.

I. INTRODUCTION

Energy harvesting is defined as the practice of capturing energy available in the environment as, for instance, sunlight, heat, movement, and others, and converting it into electric energy to be stored or consumed. A common usage for the harvested energy is to power low-power electronic devices as sensors and microcontroller-based boards and some applications are, for example, wireless sensors networks for environmental monitoring, aeronautics applications, and photovoltaic panels monitoring. We have developed a fuzzy-rule-based controller to perform on-line temperature control of a concentric-tube heat exchanger facility. The rules for the controller were derived from dynamical values of the mass flow rates and fluid temperatures in the heat exchanger. The fuzzy controller was embedded in a closed-loop, single-input single-output system to control the outlet temperature of the cold fluid. The development of the controller was done in two phases, the difference between the phases being the information about the system that was used to build the corresponding control scheme. To validate this fuzzy controller, a series of experiments, corresponding to regulation and tracking of a set point, was carried out for maintaining specific values of the outlet temperature under different perturbations. Results from this investigation demonstrate that the fuzzy-rule-based controller is able to effect control of the system, and that the amount of information about the plant provided to it is important in terms of accuracy and efficiency. The system parameters are monitored through SCADA.

1.1 EXISTING METHOD:

Bilateral tele-operation technology has been widely concerned by its unique advantages in the human machine interaction based cooperative operation systems. Communication delay, various nonlinearities and uncertainties in tele-operation system are the main challenging issues to achieve system's stability and good transparency performance. In this paper, a globally stable adaptive fuzzy back stepping control design is proposed for nonlinear bilateral tele-operation manipulators to handle the fore mentioned issues. For the communication channel, instead of direct transmission of environmental torque signals, the fuzzy-based non-power approximate environmental parameters are transmitted to the master side for the environmental torque prediction, which effectively avoids the transmission of power signals in the delayed communication channel and solves the passivity problem in the traditional tele-operation system. Subsequently, the nonlinear adaptive fuzzy back stepping controllers for master and slave are separately designed to handle the nonlinearities and uncertainties. Theoretically, the great transparency performance of both position tracking and force feedback can be achieved, and the global stability is still guaranteed under communication delay. The comparative experiments are conducted on the real platform, and verify the effectiveness and advantages of proposed control design in some typical working scenarios

1.2 PROPOSED METHOD:

The main characteristics of the proposed system are that it is online and remote, that is, while the thermal-source-under-test is being measured, the system is emulating it and evaluating the generated energy remotely by using wireless application protocol. The main contribution of this work was to replace the conventional Proportional Integral Derivative (PID) controller to a Fuzzy-Proportional Integral (PI) controller. In order to compare both controllers, three tests were carried out, namely: (a) step response, (b) perturbation test, (c) thermal emulation of the thermal pattern obtained from a potential thermal source: tree trucks. Experimental results show that the Fuzzy-PI controller was faster than the PID, achieving a setting time 43.26% faster, and also was more efficient with a maximum error 55% smaller than the PID. Results from this investigation demonstrate that the fuzzy-rule-based controller is able to effect control of the system, and that the amount of information about the plant provided to it is important in terms of accuracy and efficiency. The system parameters are monitored through SCADA.

II. LITERATURE SURVEY

1. **Shines T.S., S. Ramamoorthy Performance comparison of Quasi-Z-Source inverter with Current Source Switched Boost Quasi Impedance Source Inverter International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7, Issue-6, March 2019**

This system present the comparison of quasi impedance source inverter (QZSI) with boost switched based quasi impedance inverter (BSL-QZSI). Mainly this type of inverter is used for interfacing low voltage dc to high voltage ac conversion application for reducing converter stages and components count. The BSL-QZSI has additional switch for controlling the input power flow through inductor. The QZSI does not require control switch for input side. Both inverter circuits are simulated using sine PWM technique. Their performance is compared. The QZSI circuit has better performance compare than BSL-QZSI inverter. Sine PWM method is used to reduce the harmonics and improve the output voltage control.

2. **Venkateshmkumar Fuzzy Controller-Based MPPT of PV Power System oct 2018**

The power demand has been increasing day by day due to population growth, new industrial development, etc. Meeting power demand is one of the challenge factors for fossil fuel-based power generation alone as well as the environmental issue of carbon footprint. Consequently, there is a need to concentrate on alternate energy sources to meet the power demand. In this chapter, the photovoltaic (PV) cell operation under various weather conditions is analyzed, and based on the performance, the MPPT controller is developed by using fuzzy logic controller. The proposed system has been modeled in MATLAB environment, and the system performance has been analyzed. Finally, the simulation results are evaluated and compared with IEEE 1547 standard for proving the effectiveness of the proposed system

3. **[1]Mohit Sharma, [2]Sonam Gupta Adaptive Neuro Fuzzy Inference System technique for photovoltaic array system International Journal of Industrial Electronics and Electrical Engineering, ISSN: 2347-6982 Volume-4, Issue-8, Aug.-2016**

This system represents a more performance tracking of maximum power delivered from photovoltaic systems using adaptive (neural-fuzzy) inference systems (ANFIS). This procedure combines the learning abilities of artificial neural networks and the aptness of fuzzy logic to handle indistinct data. It is consequently able to manage non linear and time varying complication hence making it worthy for this work. It is expected that this method will be able to accurately track the maximum power point. This will make sure efficient use of PV systems and therefore leading to reduced cost of electricity.

4. **Po-Chen Cheng 1,†, Bo-Rei Peng 1,†, Yi-Hua Liu 1,*, Yu-Shan Cheng Optimization of a Fuzzy-Logic-Control-Based MPPT Algorithm Using the Particle Swarm Optimization Technique *Energies* 2015, *8*, 5338-5360; doi:10.3390/en8065338**

In this system, an asymmetrical fuzzy-logic-control (FLC)-based maximum power point tracking (MPPT) algorithm for photovoltaic (PV) systems is presented. Two membership function (MF) design methodologies that can improve the effectiveness of the proposed asymmetrical FLC-based MPPT methods are then proposed. The first method can quickly determine the input MF setting values via the power–voltage (P–V) curve of solar cells under standard test conditions (STC). The second method uses the particle swarm optimization (PSO) technique to optimize the input MF setting values. Because the PSO approach must target and optimize a cost function, a cost function design methodology that meets the performance requirements of practical photovoltaic generation systems (PGSs) is also proposed. According to the simulated and experimental results, the proposed asymmetrical FLC-based MPPT method has the highest fitness value, therefore, it can successfully address the tracking speed/tracking accuracy dilemma compared with the traditional perturb and observe (P&O) and symmetrical FLC-based MPPT algorithms. Compared to the conventional FLC-based MPPT method, the obtained optimal asymmetrical FLC-based MPPT can improve the transient time and the MPPT tracking accuracy by 25.8% and 0.98% under STC, respectively.

5. **R.Vidhya1, R.Aarthi2, M.Arun Kaarthick Design and Implementation of Quasi-Z-Source Inverter for Off-grid Photovoltaic Systems IJCSMC, Vol. 4, Issue. 3, March 2015, pg.626 – 633**

The rapidly increasing environmental degradation across the globe is posing a major challenge to develop commercially feasible alternative sources of electrical energy generation. Thus, a huge research effort is being conducted worldwide to come up with a solution in developing an environmentally benign and long-term sustainable solution in electric power generation. The major players in renewable energy generation are photovoltaic (PV), wind farms, fuel cell, and biomass. The quasi-Z-source inverter (qZSI) with the proposed battery operation can balance the turbulence of photovoltaic (PV) power injected to the load as in the existing topology, but overcomes the limitations in the rating, size and life of battery. This system proposes a new topology that is optimized for off grid applications. The characteristics of the proposed idea are analyzed in detail. In the proposed model, voltage boost and inversion are integrated in a single-stage inverter. A prototype is built to experiment the proposed circuit and to test the control methods. The results obtained are verified with the theoretical analysis and proves the effectiveness of the proposed control of the inverter's input and output power and battery power. The PV panel and energy-stored qZSI, setup used in the experiment demonstrates three operating modes that make it suitable for off grid applications.

6. **ABDULLAH NOMAN A fuzzy logic control method for MPPT of PV systems IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society**

Maximum power point trackers are so important in photovoltaic systems to increase their efficiency. Many methods have been proposed to achieve the maximum power that the PV modules are capable of producing under different weather conditions. This system proposed an intelligent method for maximum power point tracking based on fuzzy logic controller. The system consists of a photovoltaic solar module connected to a DC-DC Buck-boost converter. The system has been experienced under disturbance in the photovoltaic temperature and irradiation level. The simulation results show that the proposed maximum power tracker could track the maximum power accurately and successfully in all condition tested. Comparison of different performance parameters such as: tracking efficiency and response time of the system shows that the proposed method gives higher efficiency and better performance than the conventional perturbation and observation method. The other method is the constant voltage tracking (CVT). This method compares the measured voltage of the PV module with a reference voltage to continuously alter the duty cycle of DC-DC converter and hence operate the PV module at the predetermined point close to the MPP.

7. **G.Balasubramanian Assistant Professor Department of Electrical Engineering Annamalai University Fuzzy logic controller for the maximum power point tracking in photovoltaic system International Journal of Computer Applications (0975 – 8887) Volume 41– No.12, March 2012**

This system presents afuzzy logic controller for maximum power point tracking (MPPT) in photovoltaic system.An easy and accurate method of modeling photovoltaic arrays is proposed. The model and fuzzy based control strategies are combined to form intelligent controllers that are more accurate and robust.The model based controller is designed such that the reference signal for PWM generator of the converter can be adjusted to achieve maximum power generation from the photo voltaic system. The proposed fuzzy logic controller shows better performances compared to the P&O and PI MPPT based approach.A MATLAB based modeling and simulation scheme along with MPPT and fuzzy logic controller is proposed which are suitable for studying the I-V and P-V characteristics of a PV array under a non-uniform irradiation and different temperature. The model has been experimentally validated.

III. PROPOSED SYSTEM FUNCTION

3.1 FUNCTIONAL BLOCK DIAGRAM

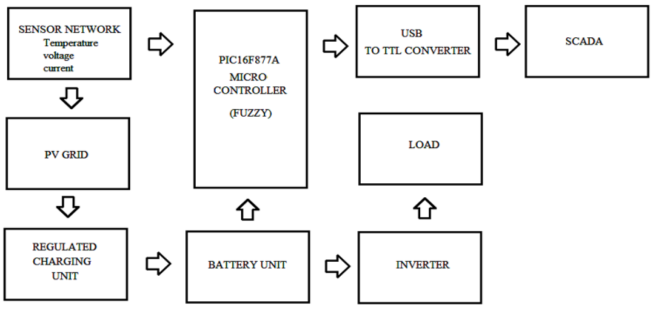

Fig.3.1 block diagram of the system

This system consists of LM35, 12V solar panel, lead acid battery, regulated charging unit, inverter, PIC16F877A microcontroller, CP2102 (USB to TTL Converter) and SCADA software. The LM35 is a transistor which is used to measure the temperature level of the PV panel. The LM35 generates the 10mV per 1^0C. The output of the sensor is applied to the input of the ADC (analog to digital converter) port of the controller. The PIC microcontroller has inbuilt ADC which converts the analog output of the sensor value into digital value. The sample voltage of the solar panel and battery unit is taken as an input of the ADC port through voltage divider circuit. The controller measure and store the voltage, temperature, current value of the PV grid and also the controller send all measured values to monitoring software such as SCADA (supervisory control and data acquisition). The solar output voltage is applied to the battery unit through charging regulator unit. The inverter is used to convert DC output of the battery unit into AC signal. The inverted output i.e., AC is used to drive the load.

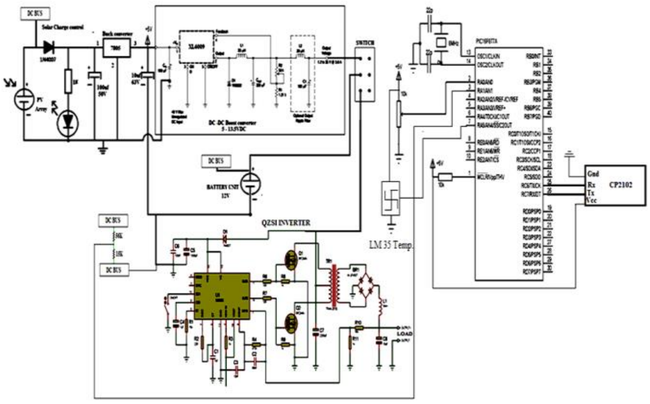

Fig: 3.2 Circuit diagram of the system

3.2 ANALYSIS OF FUZZY

FUZZY AS A CONTROL TOOL

Generally PI controller is widely used in BLDC motor control;however it does not give satisfactory results when controlparameters and loading condition changes rapidly. The fuzzylogic controller (FLC) will guarantee a stable operation, even ifthere is a change in motor parameters and load disturbances. The reason is obvious; any control system maps the input space to theoutput space. Generally, a desired set of outputs are calculated fora given set of inputs. This mathematical calculation is representedwith a formula, which demonstrates the system behavior.However, this mathematical formula may be too complex to usefor the real world issues. In these cases, fuzzy logic provides auseful methodology to create a practical solution for controllingcomplex systems. It is not necessary to know the exact model ofsuch complex systems in order to design a FLC. It is sufficient tounderstand the general behavior of the system.

Fuzzy logicenables the designer to express the general behavior of thesystems in an easier (linguistic) manner where it is allowed to usewords and sentences instead of numbers and equations. This isaccomplished by forming IF-THEN rules which describe thecharacteristics of the system. High degree of automation androbust nonlinear control is also possible by means of fuzzycontroller.

3.2.1. DESIGN OF FUZZY CONTROLLER

The purpose of the speed control of a brushless dc motor is toarrange the applied voltage in order to reach the reference speed.An error is determined by the difference between the actual speedand the reference speed. The applied voltage should be changedby increasing or decreasing the duty cycle of power transistors inorder to minimize the error.In order to accomplish this taskfuzzy controller is designed. Error and change in error are theinputs for the fuzzy controller whereas the output of the controlleris change in duty cycle. Two input single output Mamdani type offuzzy controller with 25 rules is designed for this work. Design offuzzy controller involves three steps namely fuzzification,inference mechanism and defuzzification.

3.2.2. Fuzzification

Fuzzy logic uses linguistic variables instead of numericalvariables. The process of converting a numerical variable in to alinguistic variable is called fuzzification. Five linguistic variablesNegative Big(NB), Negative Small(NS), Zero(Z), PositiveSmall(PS), Positive Big(PB) are used in this work. Triangularmembership function is assigned for input and output variablesdefined in different universe of discourses. They are shownbelow.

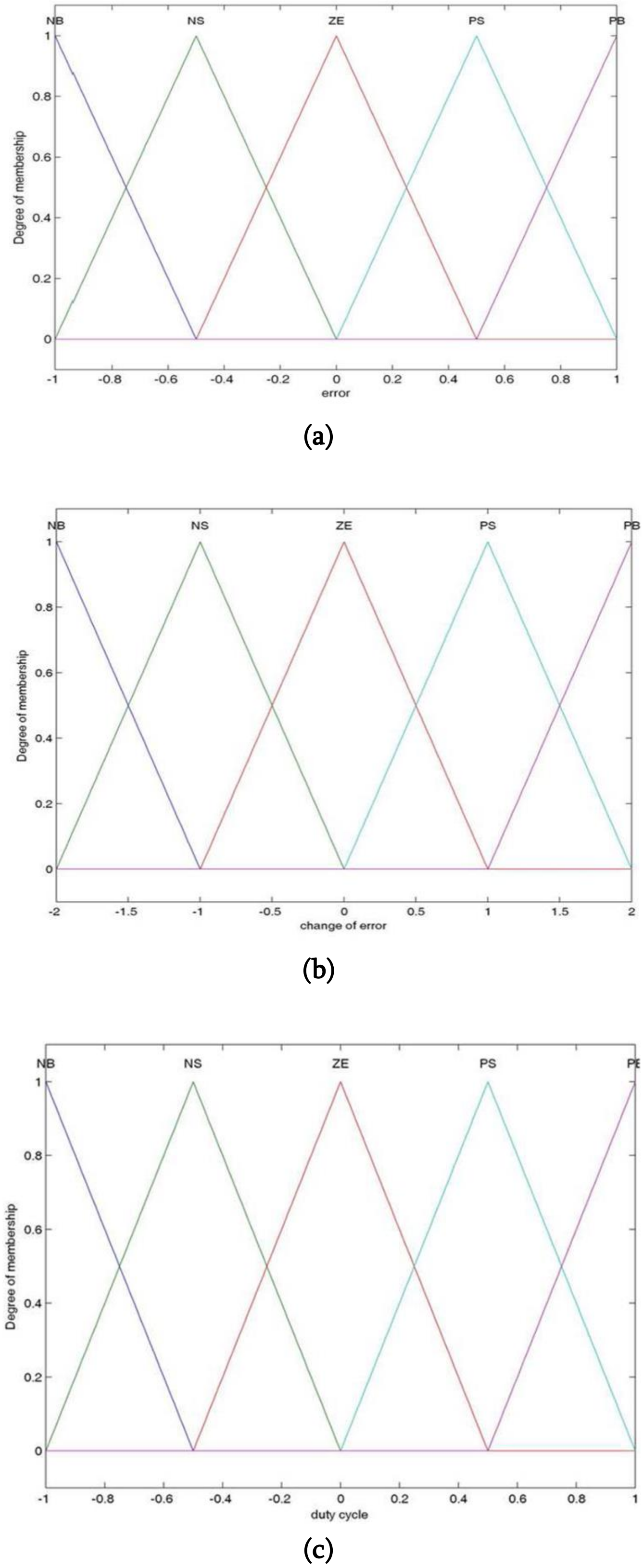

Fig.3.3(a,b,c) Membership functions used for fuzzy controller.

The inference mechanism

Table1: Rule base

CE E	NB	NS	Z	PS	PB
NB	NB	NB	NB	NS	Z
NS	NB	NB	NS	Z	PS
Z	NB	NS	Z	PS	PB
PS	NS	Z	PS	PB	PB
PB	Z	PS	PB	PB	PB

The rules are in the following format. If error is Ai, andchange in error is Bi then output is Ci. Here the if "part" of a ruleis called the rule-antecedent and is a description of a process statein terms of a logical combination of atomic fuzzy propositions.The "then" part of the rule is called the rule consequent and is adescription of the control output in terms of logical combinationsof fuzzy propositions. The rule table for the designed fuzzycontroller is given in the table 1.From the rule table the rules aremanipulated as follows.

- Rule1: If error is NB, and change in error is NB then output is NB
- Rule2: If error is NB, and change in error is NS then output is NB
- Rule3: If error is NB, and change in error is Z then output is NB
- Rule25: If error is PB, and change in error is PB then output is PB.

3.2.3. Defuzzification

The reverse of fuzzification is called defuzzification. The use ofFLC produces required output in a linguistic variable. Accordingto real world requirements, the linguistic variables have to betransformed to crisp Output. There are many methods ofdefuzzification. Centroid method of defuzzification is used in thiswork. The defuzzified output is obtained by the followingequation

$$\dot{Z} = \frac{\int \mu(z) z dz}{\int \mu(z) dz}$$

Adaptive Fuzzy Controller

Adaptive fuzzy controller is one which provides provision forchanging the parameters of fuzzy system based on performanceindex. Parameters of adaptation are

- The scaling factors for each variable.
- The fuzzy set representing the meaning of linguistic variables.
- The if-then rules.

In this work the slope of the membership function of error andchange in error are changed according to the values of error.When the error lies in the range -1 to -0.5 and 0.5 to 1, themembership function shown in figure 3.6 is used whereas when theerror lies in the range -0.5 to 0.5, the membership function shownin figure 3.7 is used.

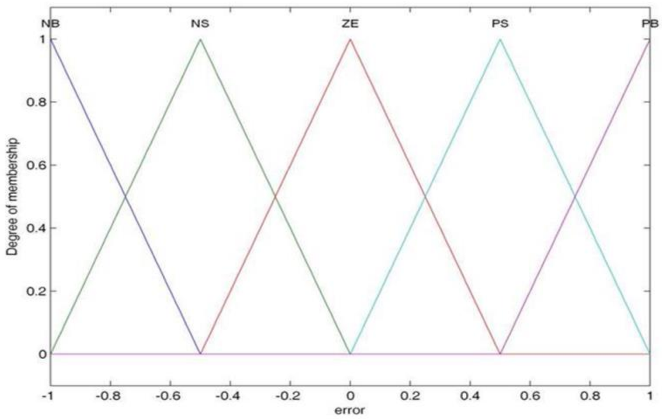

Fig.3.4 Membership function used for error in the range -1 to -0.5 and 0.5 to 1

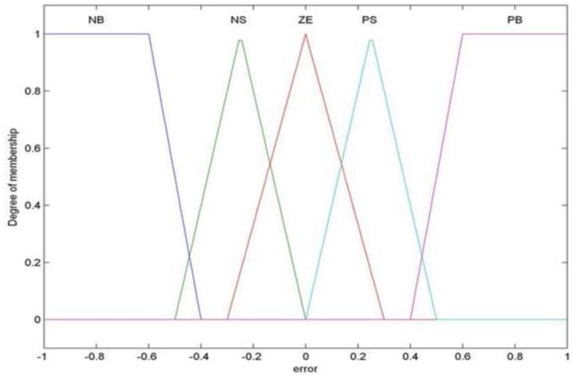

Fig.3.5 Membership function used for error in the range -0.5 to 0.5

IV. HARDWARE DETAILS

4.1 PIC16F877A MICROCONTROLLER

The PIC controller used in our project is PIC16F877A, the pin diagram of which is shown in figure. It is used to energize and de-energize the contactors during the weld and non-weld periods. The internal timer of the PIC microcontroller is used to set time delay between non-weld period and power cut off to the primary of the welding transformer.

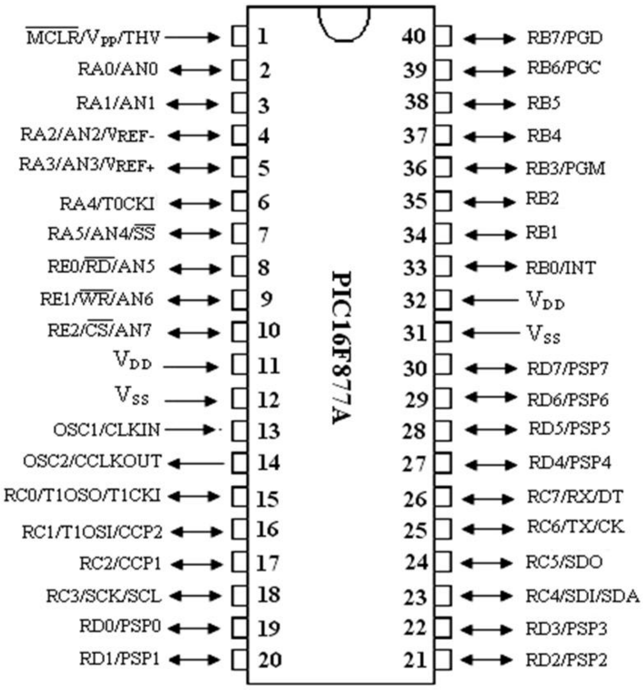

Fig: 4.1 Pin Diagram ofPIC16F877A

The advantages of PIC microcontroller are as follows

i. Increased reliability through a small part count.
ii. Reduced stock levels, as one microcontroller replaces several parts.
iii. Simplified product assembly
iv. Greater product flexibility and adaptability
v. Rapid product changes or development by changing the program and not hardware.
vi. Some practical controllers

This powerful (200 nanosecond instruction execution) yet easy-to-program (only 35 single word instructions) CMOS FLASH-based 8-bit microcontroller packs Microchip's powerful PIC® architecture into an 40- or 44-pin package and is upwards compatible with the PIC16C5X, PIC12CXXX and PIC16C7X devices. The PIC16F877A features 256 bytes of EEPROM data memory, self programming, an ICD, 2 Comparators, 8 channels of 10-bit Analog-to-Digital (A/D) converter, 2 capture/compare/PWM functions, the synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI™) or the 2-wire Inter-Integrated Circuit (I^2C™) bus and a Universal Asynchronous Receiver Transmitter (USART). All of these features make it ideal for more advanced level A/D applications in automotive, industrial, appliances and consumer applications.

High-Performance RISC CPU:

1. Only 35 single-word instructions to learn
2. All single-cycle instructions except for program branches, which are two-cycle
3. Operating speed: DC – 20 MHz clock input DC – 200 ns instruction cycle
4. Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory
5. Pinout compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules

- Capture is 16-bit, max. Resolution is 12.5 ns
- Compare is 16-bit, max. Resolution is 200 ns
- PWM max. Resolution is 10-bit

1. Synchronous Serial Port (SSP) with SPI
2. (Master mode) and I²C™ (Master/Slave)
3. Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
4. Parallel Slave Port (PSP) – 8 bits wide with external RD, WR and CS controls (40/44-pin only)
5. Brown-out detection circuitry for Brown-out Reset (BOR) Analog Features:
6. 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
7. Brown-out Reset (BOR)
8. Analog Comparator module with:

- Two analog comparators
- Programmable on-chip voltage reference (VREF) module
- Programmable input multiplexing from device inputs and internal voltage reference
- Comparator outputs are externally accessible

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

.CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

4.2 PHOTOVOLTAIC CELL

Fig. 4.2 schematic of solar cell (panel)

A **solar cell** (also called a **photovoltaic cell**) is an electrical device that converts the energy of light directly into electricity by the photovoltaic effect. It is a form of **photoelectric cell** (in that its electrical characteristics—e.g. current, voltage, or resistance—vary when light is incident upon it) which, when exposed to light, can generate and support an electric current without being attached to any external voltage source.

Introduction

The term "photovoltaic" comes from the Greek φῶς (*phōs*) meaning "light", and from "Volt", the unit of electro-motive force, the volt, which in turn comes from the last name of the Italian physicist Alessandro Volta, inventor of the battery (electrochemical cell). The term "photo-voltaic" has been in use in English since 1849.

Photovoltaics is the field of technology and research related to the practical application of photovoltaic cells in producing electricity from light, though it is often used specifically to refer to the generation of electricity from sunlight. Cells can be described as *photovoltaic* even when the light source is not necessarily sunlight (lamplight, artificial light, etc.). In such cases the cell is sometimes used as a photodetector (for example infrared detectors), detecting light or other electromagnetic radiation near the visible range, or measuring light intensity.

The operation of a photovoltaic (PV) cell requires 3 basic attributes:

1. The absorption of light, generating either electron-hole pairs or excites.
2. The separation of charge carriers of opposite types.
3. The separate extraction of those carriers to an external circuit.

In contrast, a solar thermal collector collects heat by absorbing sunlight, for the purpose of either direct heating or indirect electrical power generation. "Photoelectrolytic cell" (photo electrochemical cell), on the other hand, refers either a type of photovoltaic cell (like that developed by A.E. Becquerel and modern dye-sensitized solar cells) or a device that splits water directly into hydrogen and oxygen using only solar illumination.

i. Building block of a solar panel

Assemblies of photovoltaic cells are used to make solar modules which generate electrical power from sunlight. Multiple cells in an integrated group, all oriented in one plane, constitute a *solar photovoltaic panel* or "solar photovoltaic module," as distinguished from a "solar thermal module" or "solar hot water panel." The electrical energy generated from solar modules, referred to as *solar power*, is an example of *solar energy*. A group of connected solar modules (such as prior to installation on a pole-mounted tracker system) is called an "array."

Applications

Solar cells are often electrically connected and encapsulated as a **module**. Photovoltaic modules often have a sheet of glass on the front (sun up) side, allowing light to pass while protecting the semiconductor wafers from abrasion and impact due to wind-driven debris, rain, hail, etc. Solar cells are also usually connected in series in modules, creating an additive voltage. Connecting cells in parallel will yield a higher current; however, very significant problems exist with parallel connections. For example, shadow effects can shut down the weaker (less illuminated) parallel string (a number of series connected cells) causing substantial power loss and even damaging the weaker string because of the excessive reverse bias applied to the shadowed cells by their illuminated partners. Strings of series cells are usually handled independently and not connected in parallel, special paralleling circuits are the exceptions. Although modules can be interconnected to create an **array** with the desired peak DC voltage and loading current capacity, using independent MPPTs (maximum power point trackers) provides a better solution. In the absence of paralleling circuits, shunt diodes can be used to reduce the power loss due to shadowing in arrays with series/parallel connected cells.

The solar cell works in three steps:

1. Photons in sunlight hit the solar panel and are absorbed by semiconducting materials, such as silicon.
2. Electrons (negatively charged) are knocked loose from their atoms, causing an electric potential difference. Current starts flowing through the

material to cancel the potential and this electricity is captured. Due to the special composition of solar cells, the electrons are only allowed to move in a single direction.

3. An array of solar cells converts solar energy into a usable amount of direct current (DC) electricity.

PANEL SPEC:

1.	Module type	:	MP – 3Wp
2.	Peak power	:	Pmax (W)3
3.	Maximum current	:	Imp 0.34 (A)
4.	Maximum Voltage	:	Vmp 9(V)
5.	Short circuit current	:	Isc 0.41(A)
6.	Open circuit voltage	:	Voc 11.25(V)

4.3 LM35 TEMPERATURE SENSOR

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling.The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full –55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy.

Fig: 4.3 Pin Diagram ofLM35

It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a –55° to +150°C temperature range, while the LM35C is rated for a –40° to +110°C range (–10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Features:

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full –55° to +150°C range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain

4.4 BUCK CONVERTER / VOLTAGE REGULATOR

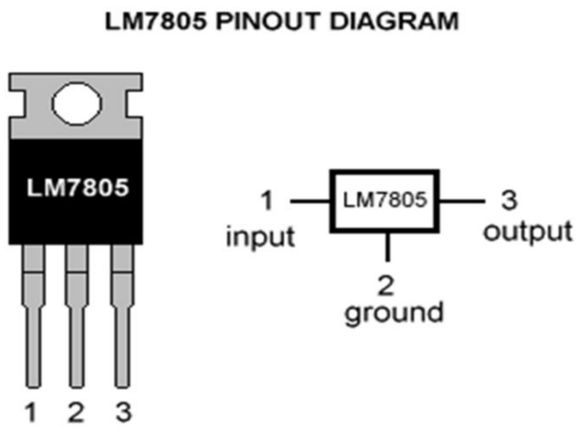

Fig: 4.4 Pin details of IC 7805

A **buck converter** (**step-down converter**) is a DC-to-DC power converter which steps down voltage (while stepping up current) from its input (supply) to its output (load). It is a class of switched-mode power supply (SMPS) typically containing at least two semiconductors (a diode and a transistor, although modern buck converters frequently replace the diode with a second transistor used for synchronous rectification) and at least one energy storage element, a capacitor, inductor, or the two in combination. To reduce voltage ripple, filters made of capacitors (sometimes in combination with inductors) are normally added to such a converter's output (load-side filter) and input (supply-side filter)

IC 7805 REGULATOR

7805 IC Rating:

- Input voltage range 7V- 35V
- Current rating $I_{c=}$ 1A
- Output voltage range $V_{Max=5.2V}$, $V_{Min=4.8V}$

4.5 BATTERY

Electric cell is a device that stores electric energy in the form of chemical energy and delivers electric energy when needed. The electric energy is released when a conductor is connected between the terminals of the cell. All cells consist of an electrolyte, i.e., solution containing ions, a positive electrode and a negative electrode. During energy release stage the negative electrode (cathode) reacts with the electrolyte to release electrons and positive electrode (anode) acquiring these electrons; thereby electricity is generated. When the cell has no power to release electricity, or in other words, electrolyte has reacted fully, the cell is said to be dead or discharged.

This is a secondary cell, i.e., it can be recharged when it is discharged fully. It generally consists of three or six cells connected in series. The electrolyte used is a dilute solution of sulfuric acid (H2SO4), the negative electrode consists of lead (Pb) and the positive electrode is made of lead dioxide (PbO2). This type of battery is used mostly in cars, trucks, aircraft, and other vehicles. Its major advantage is that it can deliver a strong current of electricity for starting an engine; however, it runs down very quickly. A lead-acid storage cell runs down when sulfuric acid is gradually converted into water and the electrodes are converted into lead sulfate. When the lead-acid battery is recharged, these chemical reactions are reversed until the chemicals have been restored t their original condition. A lead-acid battery has a useful life of about two to three years and they produce about 2 V per cell.

Fig: 4.5 Schematic diagram of Lead acid battery

When two or more atoms of different elements combine, they produce a molecule of a COMPOUND. For example, atoms of the elements carbon and oxygen combine to form molecules of carbon dioxide. Carbon dioxide is a COMPOUND, consisting of a combined form of the elements carbon and oxygen.

When a compound dissolves in certain substances - notably water - it breaks up into charged particles. These charged particles are called ions. Ions are NOT the same as atoms-ions are charged and atoms are not. You will remember that atoms contain an equal number of protons and electrons and therefore are neutral. But an ion of a dissolved compound either loses or gains one or more electrons. If you were to dissolve one molecule of sodium chloride - common table salt - in water, it would split into a sodium ion and a chloride ion. But the chloride ion holds on to one of the sodium ion's electrons. This gives the chloride ion a negative charge and the sodium ion a positive charge. It has been proved experimentally that a solution containing ions will conduct an electric current. The ions seem to "ferry" the current through the solution. This should explain to you why salt water is so likely to produce short circuits aboard ships. Because compounds that form ions in solution will conduct electric currents, they are called electrolytes.

BATTERY SPEC:

- Capacity : 6V / 4.5 AH
- Voltage Regulation : 7.20V – 7.50V
- Stand by use : 6.75 – 6.90
- Maximum charging current : 1.35A

4.6 USB TO TTL CONVERTER

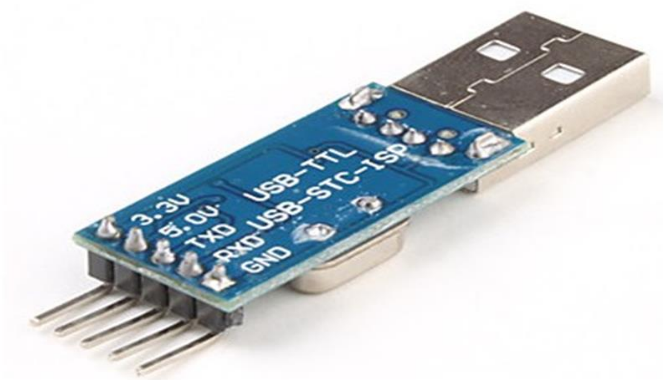

Fig.4.6 schematic of USB TO TTL

Key Features

- Model PL2303HX Color: Blue Material Copper + plastic + PCB Features: Genuine PL2303HX Chipset; Over-current protection; 500mA resettable fuse; 2 LED indicators; 3.3V and 5V connectors for different power supply conditions Application: Great for DIY project

Specifications

- Dimensions (cm): 5 x 1.5 x 0.7 Weight (kg): 0.01

4.7.1 CP2102

CP2102 chip from SiLabs is a single chip USB to UART bridge IC. It requires minimal external components. CP2102 can be used to migrate legacy serial port based devices to USB. Hobbyists can use it as a powerful tool to make all kinds of PC interfaced projects. This module helps all those who are comfortable with RS232 / Serial Communication protocol, to build USB devices very easily.

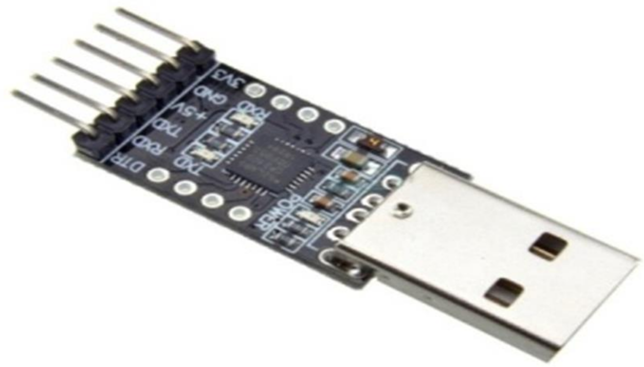

Fig.4.7 schematic diagram of CP2102

i. Working

This is an USB2.0 to TTL UART Converter module which is based on CP2102 Bridge by SiLabs. This module can be used with Laptop's which don't have standard serial port. This module creates a virtual COM port using USB on your computer which can support various standard Baud Rates for serial communication. You just need to install the driver using a setup file which automatically installs correct driver files for Windows XP/Vista/ 7. After driver installation, plug the module into any USB port of your PC. Finally a new COM port is made available to the PC. The feature which makes it more convenient is the TTL level data i/o. So you don't need to make a RS232 to TTL converter using chips like MAX232. The Rx and Tx pin can be connected directly to the MCUs pins (assuming 5v i/o).

ii. Pinouts

This module has 6 pin breakout which includes

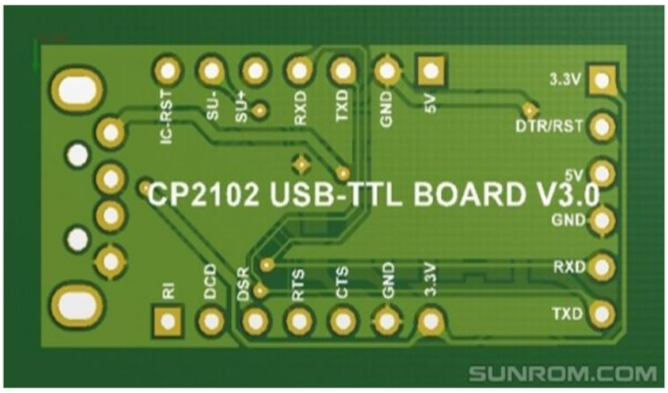

Fig.4.8 pinout deatails of CP2102

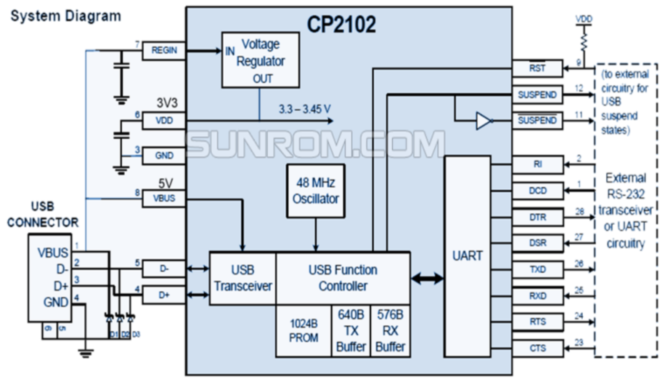

Fig.4.9 Blcoks of CP2102

- TXD = Transmit Output - Connect to Receive Pin(RXD) of Micro controller. This pin is TX pin of CP2102 on board.
- RXD = Receive Input - Connect to Transmit Pin(TXD) of Micro controller. This pin is RX pin of CP2102 on board.
- GND = Should be common to microcontroller ground.
- 3V3 = Optional output to power external circuit upto 50mA.
- 5V = Optional output to power external circuit upto 500mA
- DTR/RST = Optional output pin to reset external microcontrollers like Arduino.

This module can be used to upload Arduino sketches onto your Arduino loaded AVR devices over RXD, TXD & DTR/RST pins of module. High Quality USB to TTL converter comes with a 4-pin extension cable.

Features:

- Stable and reliable chipset CP2102.
- USB specification 2.0 compliant with full-speed 12Mbps.
- Standard USB type A male and TTL 6pin connector.
- 6pins for 3.3V, RST, TXD, RXD, GND & 5V.
- All handshaking and modem interface signals.
- Baud rates: 300 bps to 1.5 Mbps.
- Byte receive buffer; 640 byte transmit buffer.
- Hardware or X-On/X-Off handshaking supported.
- Event character support Line break transmission.
- USB suspend states supported via SUSPEND pins.
- Temperature Range: -40 to +85.
- Size: 42mm X 15mm.
- Weight: 4g

V. SCADA

5.1 SCADA

Supervisory control and data acquisition (SCADA) is a system of software and hardware elements that allows industrial organizations to:

- Control industrial processes locally or at remote locations
- Monitor, gather, and process real-time data
- Directly interact with devices such as sensors, valves, pumps, motors, and more through human-machine interface (HMI) software
- Record events into a log file

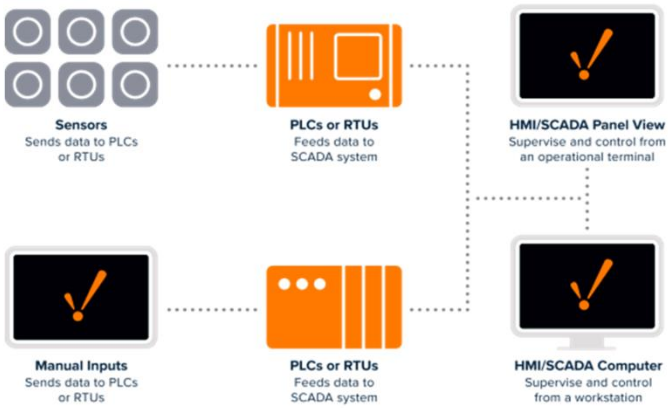

Fig: 5.1 basic blocks of SCADA

SCADA systems are crucial for industrial organizations since they help to maintain efficiency, process data for smarter decisions, and communicate system issues to help mitigate downtime.The basic SCADA architecture begins with programmable logic controllers (PLCs) or remote terminal units (RTUs). PLCs and RTUs are microcomputers that communicate with an array of objects such as factory machines, HMIs, sensors, and end devices, and then route the information from those objects to computers with SCADA software. The SCADA software processes, distributes, and displays the data, helping operators and other employees analyze the data and make important decisions.For example, the SCADA system quickly notifies an operator that a batch of product is showing a high incidence of errors. The operator pauses the operation and views the SCADA system data via an HMI to determine the cause of the issue. The operator reviews the data and discovers that Machine 4 was malfunctioning. The SCADA system's ability to notify the operator of an issue helps him to resolve it and prevent further loss of product.

SCADA systems are used by industrial organizations and companies in the public and private sectors to control and maintain efficiency, distribute data for smarter decisions, and communicate system issues to help mitigate downtime. SCADA systems work well in many different types of enterprises because they can range from simple configurations to large, complex installations. SCADA systems are the backbone of many modern industries, including:

- Energy
- Food and beverage
- Manufacturing
- Oil and gas
- Power
- Recycling
- Transportation
- Water and waste water
- And many more

Virtually anywhere you look in today's world, there is some type of SCADA system running behind the scenes: maintaining the refrigeration systems at the local supermarket, ensuring production and safety at a refinery, achieving quality standards at a waste water treatment plant, or even tracking your energy use at home, to give a few examples.

Effective SCADA systems can result in significant savings of time and money. Numerous case studies have been published highlighting the benefits and savings of using a modern SCADA software solution such as Ignition.

5.1.1 Modern SCADA Systems

Modern SCADA systems allow real-time data from the plant floor to be accessed from anywhere in the world. This access to real-time information allows governments, businesses, and individuals to make data-driven decisions about how to improve their processes. Without SCADA software, it would be extremely difficult if not impossible to gather sufficient data for consistently well-informed decisions.

Also, most modern SCADA designer applications have rapid application development (RAD) capabilities that allow users to design applications relatively easily, even if they don't have extensive knowledge of software development.The introduction of modern IT standards and practices such as SQL and web-based applications into SCADA software has greatly improved the efficiency, security, productivity, and reliability of SCADA systems.SCADA software that utilizes the power of SQL databases provides huge advantages over antiquated SCADA software. One big advantage of using SQL databases with a SCADA system is that it makes it easier to integrate into existing MES and ERP systems, allowing data to flow seamlessly through an entire organization.

VI. SOFTWARE DETAILS

6.1 EMBEDDED 'C'

Embedded C is a set of language extensions for the C Programming language by the C Standards committee to address commonality issues that exist between C extensions for different embedded systems. Historically, embedded C programming requires nonstandard extensions to the C language in order to support exotic features such as fixed-point arithmetic, multiple distinct memory banks, and basic I/O operations.

6.1.1 Embedded Systems Programming

- Embedded systems programming is different from developing applications on a desktop computers. Key characteristics of an embedded system, when compared to PCs, are as follows:

- Embedded devices have resource constraints (limited ROM, limited RAM, and limited stack space, less processing power).
- Components used in embedded system and PCs are different; embedded systems typically uses smaller, less power consuming components.
- Embedded systems are more tied to the hardware.

Two salient **features of Embedded Programming** are code speed and code size. Code speed is governed by the processing power, timing constraints, whereas code size is governed by available program memory and use of programming language. Goal of embedded system programming is to get maximum features in minimum space and minimum time.

Embedded systems are programmed using different type of languages:

- Machine Code
- Low level language, i.e., assembly
- High level language like C, C++, Java, Ada, etc.
- Application level language like Visual Basic, scripts, Access, etc.

Assembly language maps mnemonic words with the binary machine codes that the processor uses to code the instructions. Assembly language seems to be an obvious choice for programming embedded devices. However, use of assembly language is restricted to developing efficient codes in terms of size and speed. Also, assembly codes lead to higher software development costs and code portability is not there. Developing small codes are not much of a problem, but large programs/projects become increasingly difficult to manage in assembly language. Finding good assembly programmers has also become difficult nowadays. Hence high level languages are preferred for embedded systems programming.

Use of **C in embedded systems** is driven by following advantages

- It is small and reasonably simpler to learn, understand, program and debug.
- C Compilers are available for almost all embedded devices in use today, and there is a large pool of experienced C programmers.
- Unlike assembly, C has advantage of processor-independence and is not specific to any particular microprocessor/ microcontroller or any system. This makes it convenient for a user to develop programs that can run on most of the systems.
- As C combines functionality of assembly language and features of high level languages, C is treated as a 'middle-level computer language' or 'high level assembly language'
- It is fairly efficient
- It supports access to I/O and provides ease of management of large embedded projects.

Many of these advantages are offered by other languages also, but what sets C apart from others like Pascal, FORTRAN, etc. is the fact that it is a middle level language; it provides direct hardware control without sacrificing benefits of high level languages. Compared to other high level languages, C offers more flexibility because C is relatively small, structured language; it supports low-level bit-wise data manipulation.

Compared to assembly language, C Code written is more reliable and scalable, more portable between different platforms (with some changes). Moreover, programs developed in C are much easier to understand, maintain and debug. Also, as they can be developed more quickly, codes written in C offers better productivity. C is based on the philosophy 'programmers know what they are doing'; only the intentions are to be stated explicitly. It is easier to write good code in C & convert it to an efficient assembly code (using high quality compilers) rather than writing an efficient code in assembly itself. Benefits of assembly language programming over C are negligible when we compare the ease with which C programs are developed by programmers.

Dynamic C and B# are some proprietary languages which are also being used in embedded applications. Efficient embedded C programs must be kept small and efficient; they must be optimized for code speed and code size. Good understanding of processor architecture embedded C programming and debugging tools facilitate this.

Difference between C and Embedded C

Though **C and embedded C** appear different and are used in different contexts, they have more similarities than the differences. Most of the constructs are same; the difference lies in their applications. C is used for desktop computers, while **embedded C** is for microcontroller based applications. Accordingly, C has the luxury to use resources of a desktop PC like memory, OS, etc. While programming on desktop systems, we need not bother about memory. However, embedded C has to use with the limited resources (RAM, ROM, I/Os) on an embedded processor. Thus, program code must fit into the available program memory. If code exceeds the limit, the system is likely to crash.

Compilers for C (ANSI C) typically generate OS dependant executables. **Embedded C** requires compilers to create files to be downloaded to the microcontrollers/microprocessors where it needs to run. Embedded compilers give access to all resources which is not provided in compilers for desktop computer applications. Embedded systems often have the real-time constraints, which is usually not there with desktop computer applications. Embedded systems often do not have a console, which is available in case of desktop applications. So, what basically is different while programming with **embedded C** is the mindset; for embedded applications, we need to optimally use the resources, make the program code efficient, and satisfy real time constraints, if any. All this is done using the basic constructs, syntaxes, and function libraries of 'C'.

6.2 PIC C Compiler

This C compiler is fully optimized for use with PIC microcontrollers. Built in functions make coding the software very easy. Based on original K&R, the integrated C development environment gives developers a fast method to produce efficient code from an easily maintainable high level language.

CAPABILITIES

- Arrays up to 5 subscripts
- Structures and Unions may be nested.
- Custom bit fields (1-8 bits) within structures.
- ENUMurated types,
- CONSTant variables, arrays and strings.
- Full function parameter support (any number).
- Some support for C++ reference parameters.

This integrated C development environment gives developers the capability to quickly produce very efficient code from an easily maintainable high level language. The compiler includes built in functions to access the PIC hardware such as READ_ADC to read a value from the A/D converter. Discrete I/O is handled by describing the port characteristics in a PRAGMA. Functions such as INPUT and OUTPUT_HIGH will properly maintain the tri-state registers. Variables including structures may be directly mapped to memory such as I/O ports to best represent the hardware structure in C. The microcontroller clock speed may be specified in a PRAGMA to permit built in functions to delay for a given number of microseconds or milliseconds. Serial I/O functions allow standard functions such as GETC and PRINTF to be used for RS-232 like I/O. The hardware serial transceiver is used for applicable parts when possible. For all other cases a software serial transceiver is generated by the compiler. The standard C operators and the special built in functions are optimized to produce very efficient code for the bit and I/O functions. Functions may be implemented inline or separate.

Function parameters are passed in reusable registers. Inline functions with reference parameters are implemented efficiently with no memory overhead. During the linking process the program structure including the call tree is analyzed. Functions that call one another frequently are grouped together in the same page. Calls across pages are handled automatically by the tool transparent to the user. Functions may be implemented inline or separate. RAM is allocated efficiently by using the call tree to determine how locations can be re-used. Constant strings and tables are saved in the device ROM. The output hex and debug files are selectable and compatible with popular emulators & programmers including MPLAB for source level debugging. The Professional Package (PCW) provides both compilers in a powerful Windows environment.

6.3 TERMINAL V1.9B

Terminal is a simple serial port (COM) terminal emulation program. It can be used for communication with different devices such as modems, routers, embedded uC systems, GSM phones, GPS modules... It is very useful debugging tool for serial communication applications.

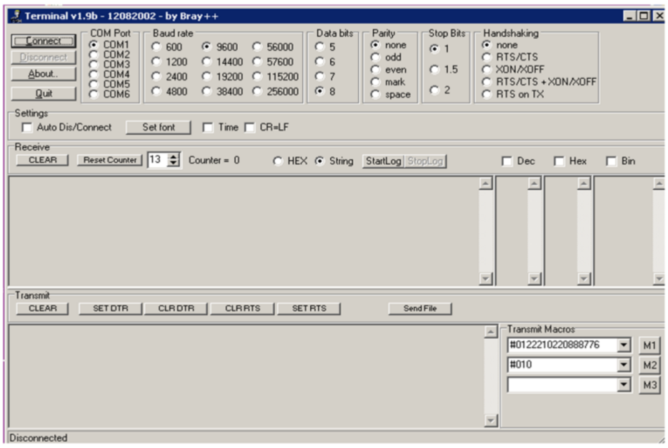

Fig.6.1 TERMINAL V1.9B WINDOW

Features

- without installation, only single and small .exe file ~300KB
- simple file send
- Rx and Tx characters counter
- baudrate up to 256kbps & custom baudrate
- up to 64 COM ports
- log to file (hex & string)
- 24 custom transmit macros with auto repeat function
- scripting (with graph/visualization support)
- remote control over TCP/IP - telnet
- run applications from macro commands
- open www sites from macro commands
- request/response macros
- CSV Graph - As a little 2012 New Year's Gift ;)
- 4th/black graph and scale factors for CSV values
- auto scroll checkbox - to prevent auto scrolling
- WriteToFile() procedure in scripting
- %SCRS"script.tsc" and %SCRE commands for macros - check help
- offset for %SUM and %XOR macro commands

6.3.1 Macros

In macros you can use all characters from keyboard and any ASCII char if you use $xx or #xxx. Where $xx is hex and #xxx dec format of ascii code. If you want to use # or $ char in macro you should type it twice ($$=$ and ##=#).To calculate XOR checksum byte use %XORxx command. To calculate SUM (1byte sum) checksum byte use %SUMxx command. Where xx is offset of first byte for calculation. To insert delay in macro string use %DLYxxxx, where xxxx is value 0000-9999 in ms. You can store macro in macro file. Active macro is "saved" even if you don't save it and will be available next time when you'll start Terminal. Macro string can be up to 256 characters long.

VII. CONCLUSION

In this work, Fuzzy controllers were proposed in order to control two distinct temperature levelsin a system capable of emulating thermal patterns in a remote way. The obtained results show thatthe Fuzzy controller achieves a steady state error 53% smaller and a settling time 41.26% smallerwhen compared with a conventional PID controller. In addition, the proposed Fuzzy controller wasrobust enough to correct variations caused by external factors as, for example, variations at roomtemperature and convection. Considering thermal emulation, it was observed that the changes in thesystem dynamics introduced by the controller made the emulation more reliable, with an RMSE equalto 0.04 _C and 0.15 _C for the upper face and bottom face, respectively. Therefore, the Fuzzy controllershows up to be more suitable to the proposed application. As future work, the Fuzzy controller ruleswill be extracted automatically using neuro-fuzzy techniques, because one of the main limitations ofthe proposed approach is that the rules extraction depends exclusively on the designer's knowledge.

VIII. REFERENCES

[1]. Shines T.S., S. Ramamoorthy Performance comparison of Quasi-Z-Source inverter with Current Source Switched Boost Quasi Impedance Source Inverter International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7, Issue-6, March 2019

[2]. Venkateshmkumar Fuzzy Controller-Based MPPT of PV Power System oct 2018

[3]. 1Mohit Sharma, 2Sonam Gupta Adaptive Neuro Fuzzy Inference System technique for photovoltaic array system International Journal of Industrial Electronics and Electrical Engineering, ISSN: 2347-6982 Volume-4, Issue-8, Aug.-2016

[4]. Po-Chen Cheng 1,†, Bo-Rei Peng 1,†, Yi-Hua Liu 1,*, Yu-Shan Cheng Optimization of a Fuzzy-Logic-Control-Based MPPT Algorithm Using the Particle Swarm Optimization Technique Energies 2015, 8, 5338-5360; doi:10.3390/en8065338

[5]. R.Vidhya1, R.Aarthi2, M.Arun Kaarthick Design and Implementation of Quasi-Z-Source Inverter for Off-grid Photovoltaic Systems IJCSMC, Vol. 4, Issue. 3, March 2015, pg.626 – 633

[6]. ABDULLAH NOMAN A fuzzy logic control method for MPPT of PV systems IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society

[7]. G.Balasubramanian Assistant Professor Department of Electrical Engineering Annamalai University Fuzzy logic controller for the maximum power point tracking in photovoltaic system International Journal of Computer Applications (0975 – 8887) Volume 41– No.12, March 2012

APPENDIX
SOURCE CODING

```
#include "ADCtoUART.h"
void main()
{
  float value_analog,value_analog2; // Program
variables
  float conv1,conv2;
  unsigned int8 i, j;

  setup_adc_ports(ALL_ANALOG);   //set pins
AN0-AN7 to analog
  setup_adc(ADC_CLOCK_INTERNAL);
  setup_ccp1(CCP_PWM);                // Configure
CCP1 as a PWM
 setup_ccp2(CCP_PWM);                // Configure
CCP2 as a PWM
 delay_ms(100);                     // Wait 100ms
 setup_timer_2(T2_DIV_BY_16, 255, 1);   // Set
PWM frequency to 488Hz
delay_ms(1000);
  while (true)
  {
    set_adc_channel(1);
    value_analog = read_adc();
    conv1=value_analog/16.5;
    printf("adc=%f",value_analog);
    printf("%f", conv1);
    //printf(",");
    delay_ms (2);   // Dá um tempinho
    i = read_adc();
    delay_ms(10);
    lcd_gotoxy(1,1);
    //printf(lcd_putc,"DC GRID1 :%2.2f  ",conv1);
    printf(lcd_putc,"GRID VALUE       ");

    set_adc_channel(2);
     value_analog2 = read_adc();
```

```
    conv2=value_analog2/3.35;
    printf("%f", conv2);
    /*printf(",");
    printf("000");*/
    //printf("\n");
    //printf("\r");
    delay_ms (2);
     j = read_adc();
    delay_ms(10);
     set_pwm1_duty(i);
    set_pwm2_duty(j);
    printf("   %2.2f  ",conv1);
            }
}
```