Visvesvaraya Technological University

Belgaum, Karnataka-590 018

A Project Report on

"RF-Based Multiple Device Control"

Project Report submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering
In
Electrical & Electronics Engineering

Submitted by

Kushagra Sahay (1CR17EE031)
Mayank Kumar (1CR17EE033)
Meharbaan Singh (1CR17EE034)
Prabhat Sachan (1CR17EE042)

Under the Guidance of

Dr. S Zahid Nabi Dar
Asst. Professor, Department of Electrical & Electronics Engineering
CMR Institute of Technology

CMR Institute of Technology, Bengaluru-560 037

Department of Electrical & Electronics Engineering

2020-2021

CMR INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

AECS Layout, Bengaluru-560 037

Certificate

Certified that the project work entitled **"RF-Based Multiple Device Control"** carried out by Mr. Kushagra Sahay (USN 1CR17EE031); Mr. Mayank Kumar (USN 1CR17EE033); Mr. Meharbaan Singh (USN 1CR17EE034); Mr. Prabhat Sachan (USN 1CR17EE042) are bonafied students of CMR Institute of Technology, Bengaluru, in partial fulfillment for the award of Bachelor of Engineering in Electrical & Electronics Engineering of the Visvesvaraya Technological University, Belgaum, during the year 2020-2021. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Signature of the Guide	*Signature of the HOD*	*Signature of the Principal*
-----------------------------	--------------------------------	-----------------------------
Dr. S Zahid Nabi Dar, Asst. Professor EEE Department CMRIT, Bengaluru	Dr. K. Chitra Professor & HOD EEE Department CMRIT, Bengaluru	Dr. Sanjay Jain Principal, CMRIT, Bengaluru

External Viva

Name of the Examiners	Signature & Date
1.	
2.	

CMR INSTITUTE OF TECHNOLOY

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

AECS Layout, Bengaluru-560 037

DECLARATION

We, Mr. Kushagra Sahay (USN 1CR17EE031); Mr. Mayank Kumar (USN 1CR17EE033); Mr. Meharbaan Singh (USN 1CR17EE034); Mr. Prabhat Sachan (USN 1CR17EE042)**,** hereby declare that the report entitled **"RF-Based Multiple Device Control"** has been carried out by us under the guidance of **Dr S Zahid Nabi Dar, Asst.** Professor, Department of Electrical & Electronics Engineering, CMR Institute of Technology, Bengaluru, in partial fulfillment of the requirement for the degree of **BACHELOR OF ENGINEERING in ELECTRICAL & ELECTRONICS ENGINEERING**, of Visveswaraya Technological University, Belgaum during the academic year 2020-21. The work done in this report is original and it has not been submitted for any other degree in any university.

Place: Bengaluru

Date:04/01/2021

Mr. Kushagra Sahay (1CR17EE031)

Mr. Mayank Kumar (1CR17EE033)

Mr. Meharbaan Singh (1CR17EE034)

Mr. Prabhat Sachan (1CR17EE042)

Abstract

The project RF based home automation system is developed to automate the use of conventional lighting mechanism (wall switches) in house by using RF controlled remote. The project requires a RF remote that is interfaced to microcontroller (of 8051 family) on transmitter side which sends ON/OFF signals to the receiver. Receivers are connected with loads that can be turned ON/OFF by operating remote switches on transmitter wirelessly.

Here the loads are interfaced to microcontroller by utilizing opto-isolators and triacs. Thus, the system serves a convenient way of lighting up the house without any physical movements.
As technology is advancing so houses are also getting smarter. Modern houses are gradually shifting from conventional switches to centralized control system, involving RF controlled switches. Presently, conventional wall switches located in different parts of the house makes it difficult for the user to go near them to operate. Even more it becomes more difficult for the elderly or physically handicapped people to do so. Remote controlled home automation system provides a simpler solution with RF technology.

Acknowledgement

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people, who are responsible for the completion of the project and who made it possible, because success is outcome of hard work and perseverance, but steadfast of all is encouraging guidance. So, with gratitude we acknowledge all those whose guidance and encouragement served us to motivate towards the success of the project work.

We take great pleasure in expressing our sincere thanks to ***Dr. Sanjay Jain****,* ***Principal, CMR Institute of Technology, Bengaluru*** *for providing an excellent academic environment in the college and for his continuous motivation towards a dynamic career. We would like to profoundly thank* ***Dr. B Narasimha Murthy****, Vice-principal of CMR Institute of Technology and the whole* ***Management*** *for providing such a healthy environment for the successful completion of the project work.*

We would like to convey our sincere gratitude to ***Dr. K Chitra****,* ***Head of Electrical and Electronics Engineering Department, CMR Institute of Technology, Bengaluru*** *for her invaluable guidance and encouragement and for providing good facilities to carry out this project work.*

We would like to express our deep sense of gratitude to ***Dr. S Zahid Nabi Dar,***

Asst. Professor, Electrical and Electronics Engineering, CMR Institute of Technology, Bengaluru *for his/her exemplary guidance, valuable suggestions, expert advice and encouragement to pursue this project work.*

We are thankful to all the faculties and laboratory staffs of ***Electrical and Electronics Engineering Department, CMR Institute of Technology, Bengaluru*** *for helping us in all possible manners during the entire period.*

Finally, we acknowledge the people who mean a lot to us, our parents, for their inspiration, unconditional love, support, and faith for carrying out this work to the finishing line. We want to give special thanks to all our friends who went through hard times together, cheered us on, helped us a lot, and celebrated each accomplishment.

Lastly, to the ***Almighty****, for showering His Blessings and to many more, whom we didn't mention here.*

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Brief Background of the Research

1. Radio Frequency Identification (RFID)is a wireless identification device. Technology that can be used to build access management technologies
2. The literature has revealed the use of this technology to automate various processes ranging from industrial sector to home control [1].
3. Bo Yan [2] has reported the use of RFID technology to automate sight spot ticket management system.
4. The system hardware consists of RFID electronic tickets, RFID readers, computer terminals, optical networks, computer servers and site controllers
5. G. Ostojic [3] has developed an automatic vehicle parking control system based on RFID technology in the city of Novi Sad, Republic of Serbia.
6. Nova Ahmed [4] has described RFID based indoor guidance and monitoring system known as Guardian Angel in pervasive environment.
7. RFID based indoor guidance and monitoring system known as Guardian Angel in pervasive environment. The beauty of the system is that it can generate dynamic queries in real time through user interface
8. Kuo-shien Huang [5] has described a business model-based approach for utilizing RFID technology in automating the process according to enterprise strategic vision and goals.
9. In this work we are interested to develop an RFID enterprise for multitasking interface

1.2 Objectives of the Thesis

2. The objective of project to design of RF based transmitter and receiver and its
3. implementation to control multiple devices.
4. Further we can use controller in home automation where we can directly control all home appliances
5. with a universal remote irrespective of place.

1.3 What is embedded system?

An Embedded System is a combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a specific function. An embedded system is a microcontroller-based, software driven, reliable, real-time control system, autonomous, or human or network interactive, operating on diverse physical variables and in diverse environments and sold into a competitive and cost-conscious market.

An embedded system is not a computer system that is used primarily for processing, not a software system on PC or UNIX, not a traditional business or scientific application. High-end embedded & lower end embedded systems. High-end embedded system - Generally 32, 64 Bit Controllers used with OS. Examples Personal Digital Assistant and Mobile phones etc .Lower end embedded systems - Generally 8,16 Bit Controllers used with an minimal operating systems and hardware layout designed for the specific purpose.

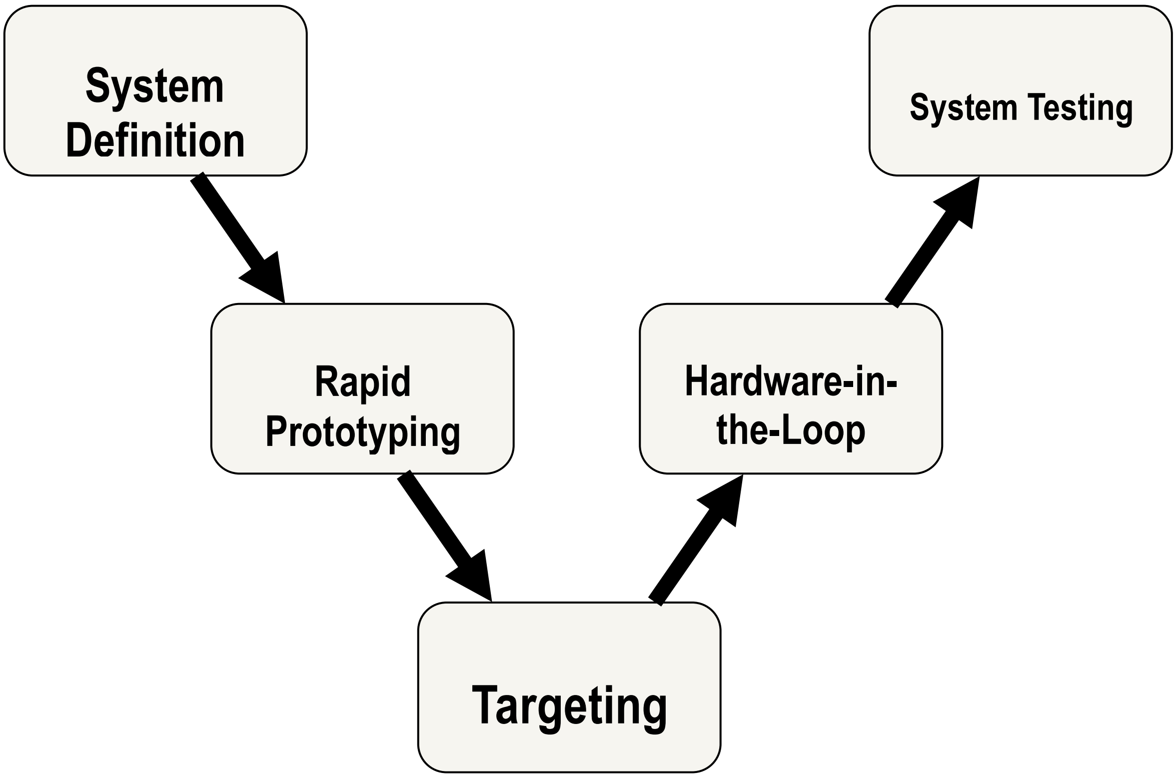


Fig: V Diagram of Embedded System

1.3.1 Characteristics of Embedded System

- An embedded system is any computer system hidden inside a product other than a computer.
- They will encounter a number of difficulties when writing embedded system software in addition to those we encounter when we write applications
  - Throughput – Our system may need to handle a lot of data in a short period of time.
  - Response–Our system may need to react to events quickly
  - Testability–Setting up equipment to test embedded software can be difficult
  - Debugability–Without a screen or a keyboard, finding out what the software is doing wrong (other than not working) is a troublesome problem

- Reliability – embedded systems must be able to handle any situation without human intervention
- Memory space – Memory is limited on embedded systems, and you must make the software and the data fit into whatever memory exists
- Program installation – you will need special tools to get your software into embedded systems
- Power consumption – Portable systems must run on battery power, and the software in these systems must conserve power
- Processor hogs – computing that requires large amounts of CPU time can complicate the response problem
- Cost – Reducing the cost of the hardware is a concern in many embedded system projects; software often operates on hardware that is barely adequate for the job.

- Embedded systems have a microprocessor/ microcontroller and a memory. Some have a serial port or a network connection. They usually do not have keyboards, screens or disk drives.

1.3.2 APPLICATIONS

1) Military and aerospace embedded software applications
2) Communication Applications
3) Industrial automation and process control software
4) Mastering the complexity of applications.
5) Reduction of product design time.
6) Real time processing of ever increasing amounts of data.
7) Intelligent, autonomous sensors.

1.3.3 CLASSIFICATION

- Real Time Systems.
- RTS is one which has to respond to events within a specified deadline.
- A right answer after the dead line is a wrong answer.

1.3.4 RTS CLASSIFICATION

- Hard Real Time Systems
- Soft Real Time System

1.3.5 HARD REAL TIME SYSTEM

- "Hard" real-time systems have very narrow response time.
- Example: Nuclear power system, Cardiac pacemaker.

1.3.6 SOFT REAL TIME SYSTEM

- "Soft" real-time systems have reduced constrains on "lateness" but still must operate very quickly and repeatable.
- Example: Railway reservation system – takes a few extra seconds the data remains valid.

CHAPTER 2

LITERATURE REVIEW

1. Radio Frequency Identification (RFID)is a wireless identification device. Technology that can be used to build access management technologies
2. The literature has revealed the use of this technology to automate various processes ranging from industrial sector to home control [1].
3. Bo Yan [2] has reported the use of RFID technology to automate sight spot ticket management system.
4. The system hardware consists of RFID electronic tickets, RFID readers, computer terminals, optical networks, computer servers and site controllers
5. G. Ostojic [3] has developed an automatic vehicle parking control system based on RFID technology in the city of Novi Sad, Republic of Serbia.
6. Nova Ahmed [4] has described RFID based indoor guidance and monitoring system known as Guardian Angel in pervasive environment.
7. RFID based indoor guidance and monitoring system known as Guardian Angel in pervasive environment. The beauty of the system is that it can generate dynamic queries in real time through user interface
8. Kuo-shien Huang [5] has described a business model-based approach for utilizing RFID technology in automating the process according to enterprise strategic vision and goals.
9. In this work we are interested to develop an RFID enterprise for multitasking interface

CHAPTER 3

PROPOSED MODEL WITH

THEORETICAL BACKGROUND

3.1 Monitoring System

TRANSMITTER SECTION

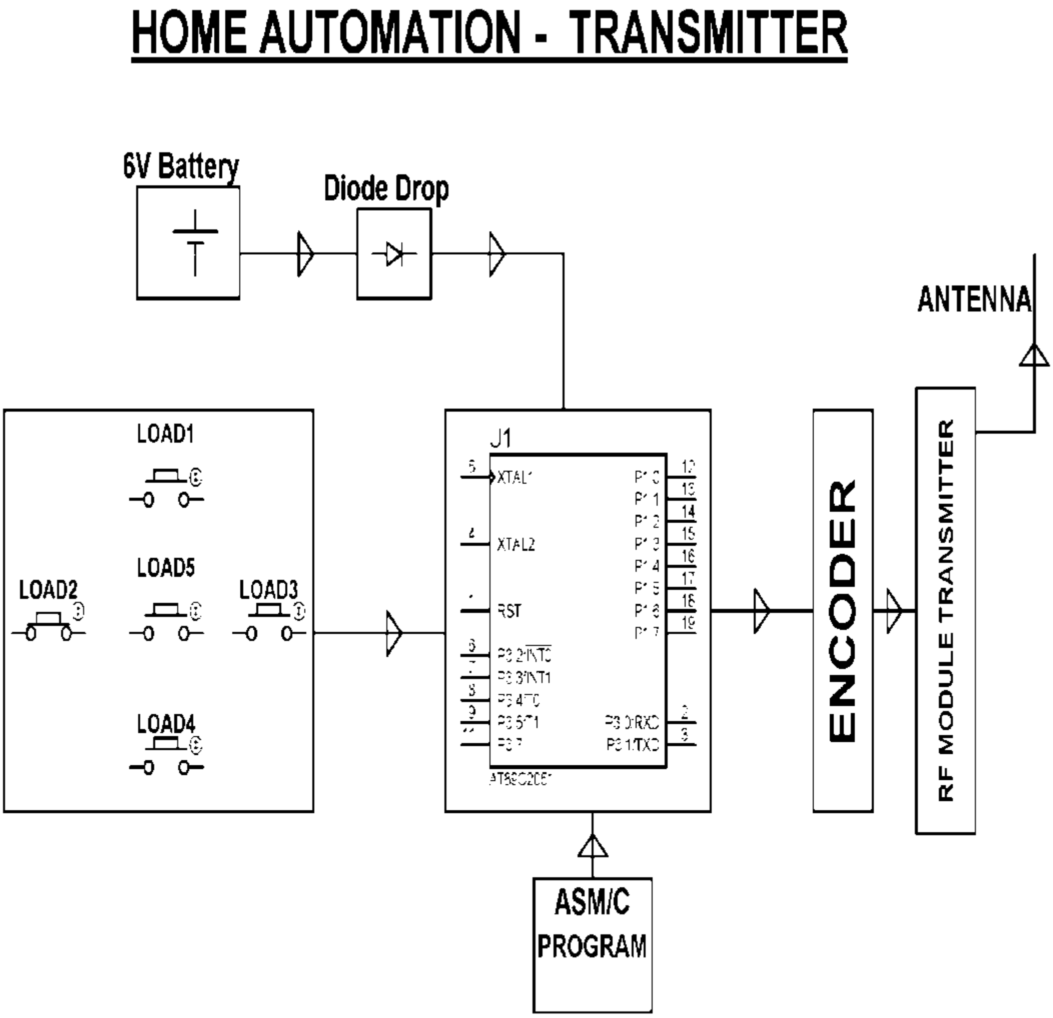


Fig2: Transmitter Section

RECIEVER SECTION

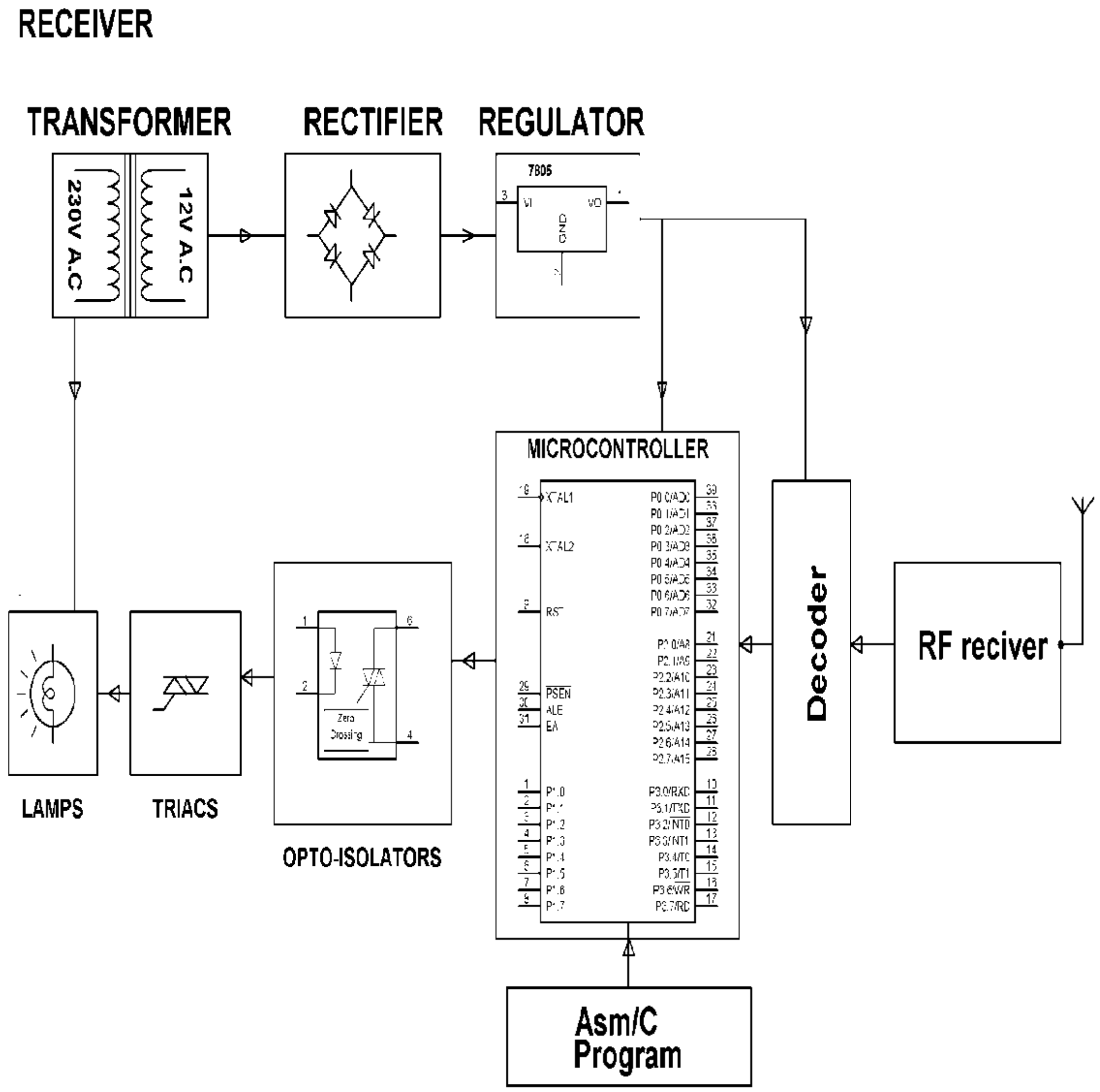


Fig3: Receiver Section

3.2 Hardware Specifications

- 8051 series Microcontroller
- Transformer
- Opto-isolator
- RF modules
- TRIAC
- Crystal
- Diodes
- Lamps
- Voltage Regulator
- Push Buttons

Software Specifications

- Keil µVision IDE
- MC Programming Language: Embedded C

3.3 Methodology

Fig. shows the block diagram for RF-based multiple device control using microcontroller. Signals from the keypad are fed to microcontroller **AT89C2051**, which, in turn, is interfaced to the RF transmitter through encoder **HT12E**. The microcontroller continuously reads the status of the keys on the keypad.

When any key is pressed, data is passed to the encoder and then to the RF transmitter from where it is transmitted. The RF receiver receives this data and gives it to the RF decoder. The decoder serially converts the serial bit data into four-bit data at a port of microcontroller **AT89C51**. The microcontroller energizes the corresponding relay through a relay driver.

Devices are connected to normally-open (N/O) contacts of the relays.

3.4 TRANSFORMER

Transformers convert AC electricity from one voltage to another with a little loss of power. Step-up transformers increase voltage, step-down transformers reduce voltage. Most power supplies use a step-down transformer to reduce the dangerously high voltage to a safer low voltage.

Fig4: Transformer

The input coil is called the primary and the output coil is called the secondary. There is no electrical connection between the two coils; instead they are linked by an alternating magnetic field created in the soft-iron core of the transformer. The two lines in the middle of the circuit symbol represent the core. Transformers waste very little power so the power out is (almost) equal to the power in. Note that as voltage is stepped down and current is stepped up.

The ratio of the number of turns on each coil, called the turn's ratio, determines the ratio of the voltages. A step-down transformer has a large number of turns on its primary (input) coil which is connected to the high voltage mains supply, and a small number of turns on its secondary (output) coil to give a low output voltage.

$$\text{TURNS RATIO} = (V_p / V_s) = (N_p / N_s)$$

Where,

V_p = primary (input) voltage.

V_s = secondary (output) voltage

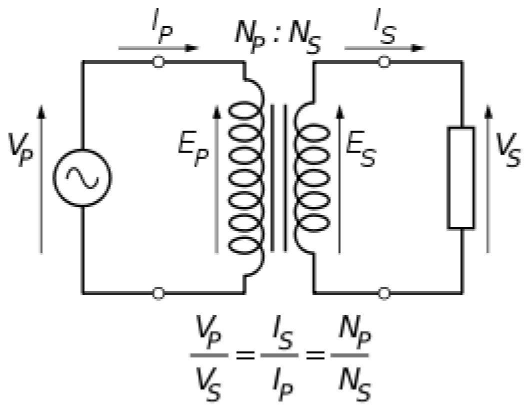
N_p = number of turns on primary coil

N_s = number of turns on secondary coil

I_p = primary (input) current

I_s = secondary (output) current.

Ideal power equation



The ideal transformer as a circuit element

If the secondary coil is attached to a load that allows current to flow, electrical power is transmitted from the primary circuit to the secondary circuit. Ideally, the transformer is perfectly efficient; all the incoming energy is transformed from the primary circuit to the magnetic field and into the secondary circuit. If this condition is met, the incoming electric power must equal the outgoing power:

$$P_{\text{incoming}} = I_{\text{p}}V_{\text{p}} = P_{\text{outgoing}} = I_{\text{s}}V_{\text{s}},$$

Giving the ideal transformer equation

$$\frac{V_{\text{s}}}{V_{\text{p}}} = \frac{N_{\text{s}}}{N_{\text{p}}} = \frac{I_{\text{p}}}{I_{\text{s}}}.$$

Transformers normally have high efficiency, so this formula is a reasonable approximation.

If the voltage is increased, then the current is decreased by the same factor. The impedance in one circuit is transformed by the *square* of the turns ratio. For example, if an impedance Z_s is attached across the terminals of the secondary coil, it appears to the primary circuit to have an impedance of $(N_p/N_s)^2Z_s$. This relationship is reciprocal, so that the impedance Z_p of the primary circuit appears to the secondary to be $(N_s/N_p)^2Z_p$.

3.5 VOLTAGE REGULATOR 7805

Features

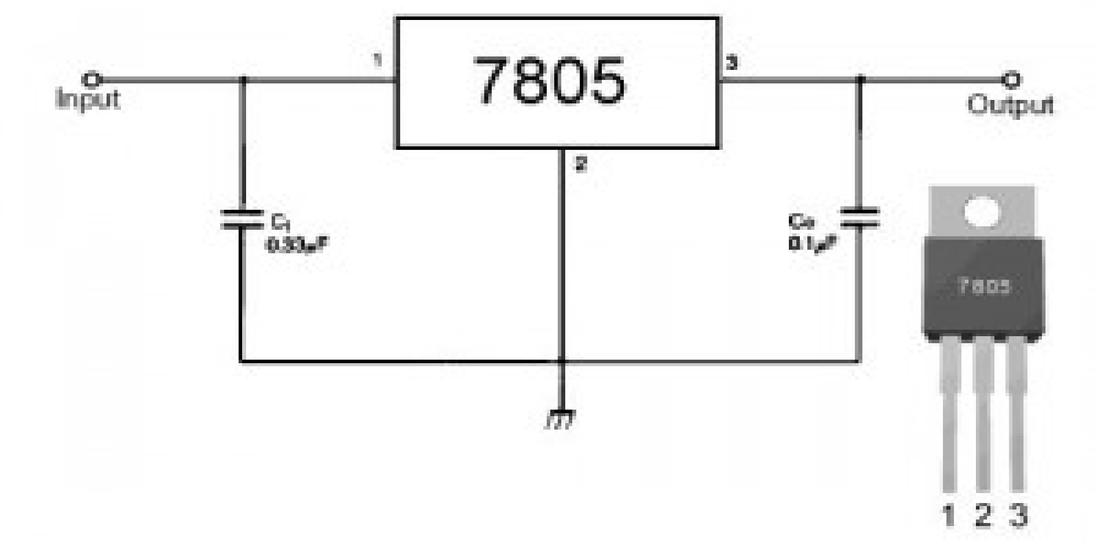
• Output Current up to 1A.

• Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V.

• Thermal Overload Protection.

• Short Circuit Protection.

• Output Transistor Safe Operating Area Protection.



Description

The LM78XX/LM78XXA series of three-terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a Wide range of applications. Each type employs internal current limiting, thermal shutdown and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output Current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

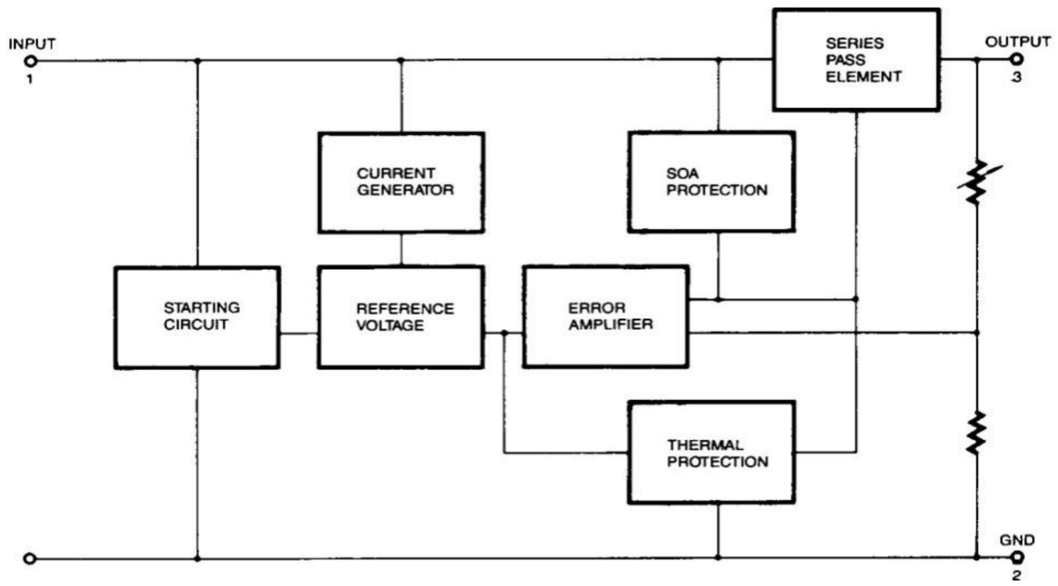


Fig5: BLOCK DIAGRAM OF VOLTAGE REGULATOR

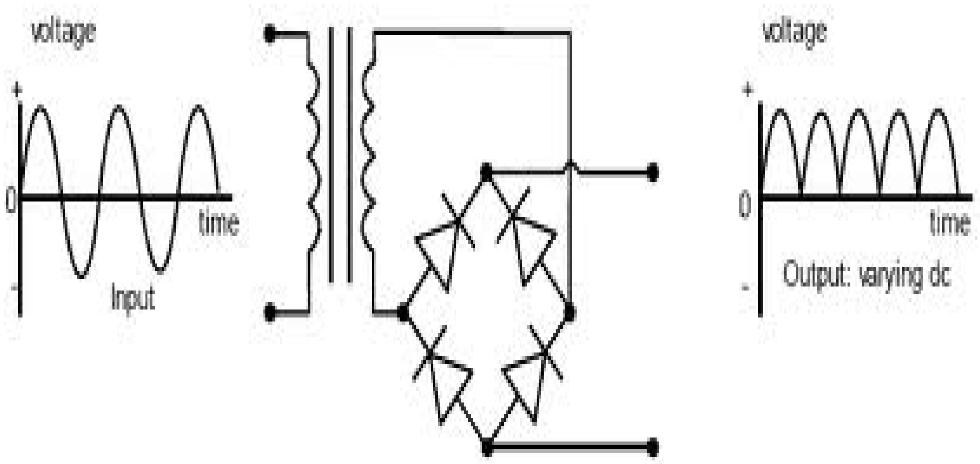
Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Input Voltage (for V_O = 5V to 18V) (for V_O = 24V)	V_I V_I	35 40	V V
Thermal Resistance Junction-Cases (TO-220)	$R_{\theta JC}$	5	°C/W
Thermal Resistance Junction-Air (TO-220)	$R_{\theta JA}$	65	°C/W
Operating Temperature Range (KA78XX/A/R)	T_{OPR}	0 ~ +125	°C
Storage Temperature Range	T_{STG}	-65 ~ +150	°C

TABLE1: RATINGS OF THE VOLTAGE REGULATOR

3.6 RECTIFIER

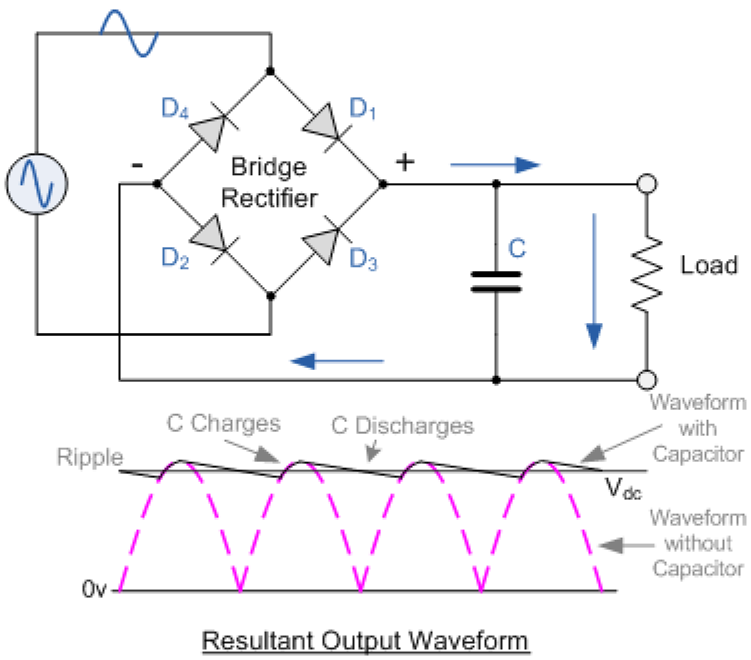
A rectifier is an electrical device that converts alternating current (AC), which periodically reverses direction, to direct current (DC), current that flows in only one direction, a process known as rectification. Rectifiers have many uses including as components of power supplies and as detectors of radio signals. Rectifiers may be made of solid state diodes, vacuum tube diodes, mercury arc valves, and other components. The output from the transformer is fed to the rectifier. It converts A.C. into pulsating D.C. The rectifier may be a half wave or a full wave rectifier. In this project, a bridge rectifier is used because of its merits like good stability and full wave rectification. In positive half cycle only two diodes (1 set of parallel diodes) will conduct, in negative half cycle remaining two diodes will conduct and they will conduct only in forward bias only.



3.7 FILTER

Capacitive filter is used in this project. It removes the ripples from the output of rectifier and smoothens the D.C. Output received from this filter is constant until the mains voltage and load is maintained constant. However, if either of the two is varied, D.C. voltage received at this point changes. Therefore a regulator is applied at the output stage.

The simple capacitor filter is the most basic type of power supply filter. The use of this filter is very limited. It is sometimes used on extremely high-voltage, low-current power supplies for cathode-ray and similar electron tubes that require very little load current from the supply. This filter is also used in circuits where the power-supply ripple frequency is not critical and can be relatively high. Below figure can show how the capacitor changes and discharges.



-

3.8 MICROCONTROLLER AT89S52

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density non-volatile memory technology and is compatible with the industry standard 80C51 instruction set and pin out. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional non-volatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications. The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

Features:

• Compatible with MCS®-51 Products

• 8K Bytes of In-System Programmable (ISP) Flash Memory

– Endurance: 10,000 Write/Erase Cycles

• 4.0V to 5.5V Operating Range

• Fully Static Operation: 0 Hz to 33 MHz

• Three-level Program Memory Lock

• 256 x 8-bit Internal RAM

• 32 Programmable I/O Lines

• Three 16-bit Timer/Counters

• Eight Interrupt Sources

• Full Duplex UART Serial Channel

• Low-power Idle and Power-down Modes

• Interrupt Recovery from Power-down Mode

• Watchdog Timer

• Dual Data Pointer

• Power-off Flag

• Fast Programming Time

• Flexible ISP Programming (Byte and Page Mode)

• Green (Pb/Halide-free) Packaging Option

Block Diagram of AT89S52:

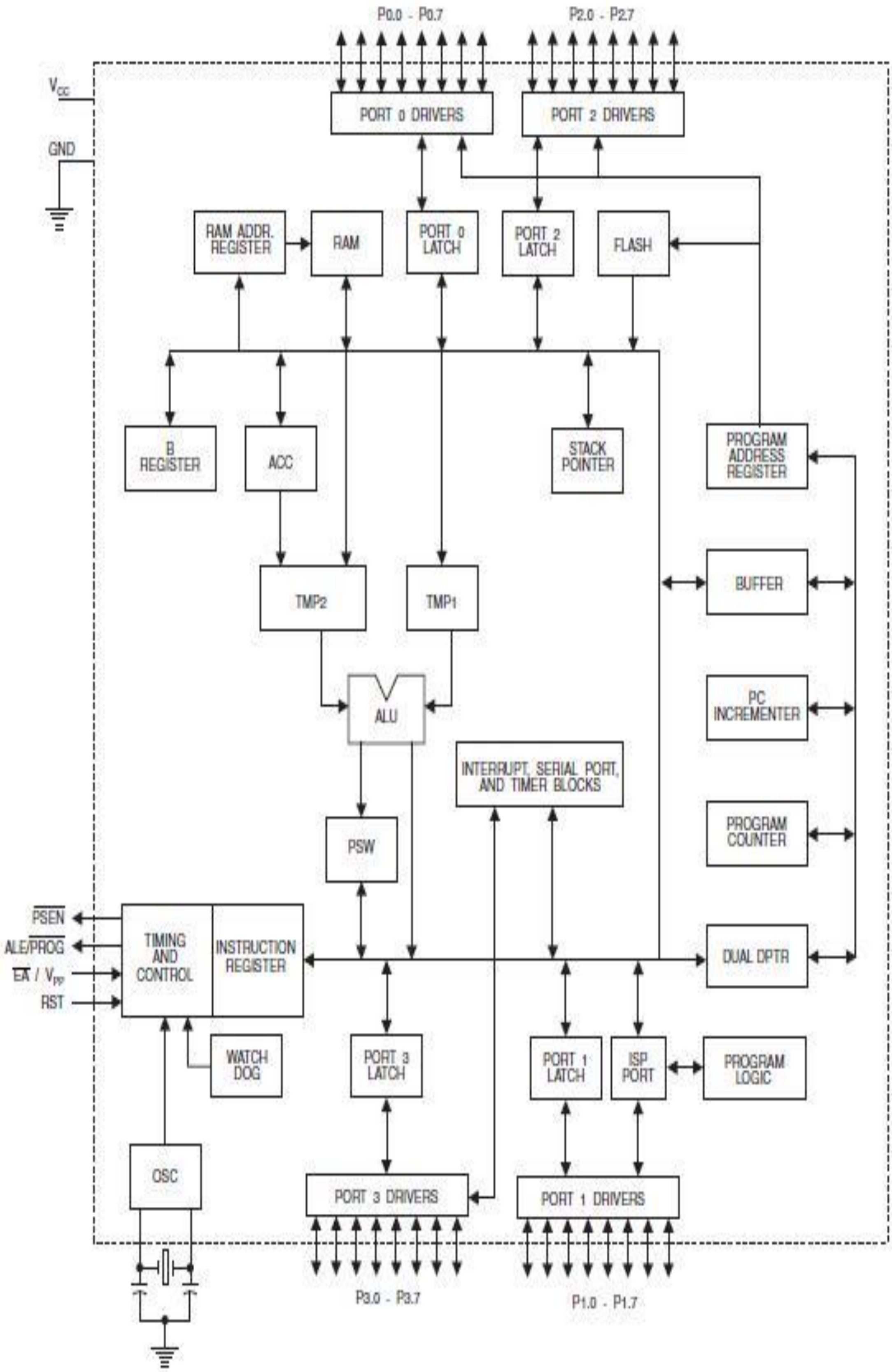


FIG6: BLOCK DIAGRAM OF AT89S52

Pin Configurations of AT89S52

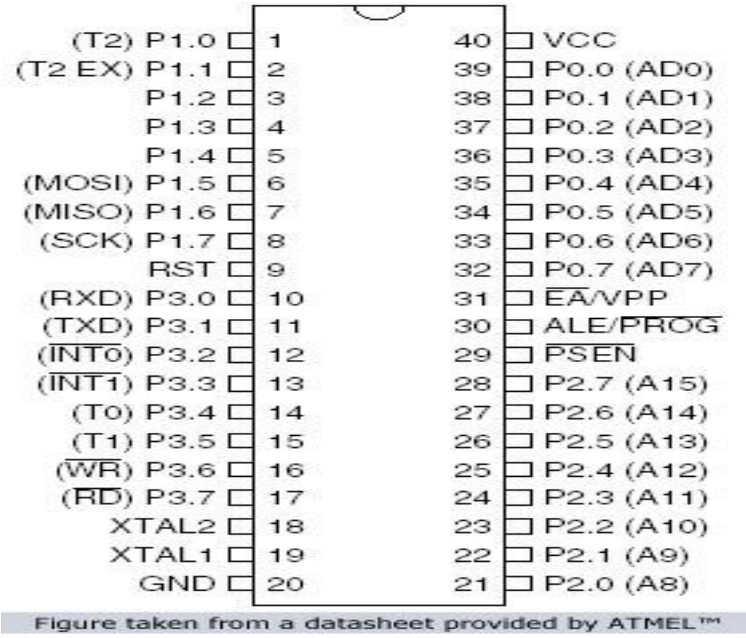


FIG7: PIN DIAGRAM OF AT89S52

Pin Description:

VCC:

Supply voltage.

GND:

Ground

Port 0:

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs. Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups. Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pull-ups are required during program verification.

Port 1:

Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the `internal pull-ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX).

Port 2:

Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL) because of the internal pull-ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 3:

Port 3 is an 8-bit bidirectional I/O port with internal pull-ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull-ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pull-ups.

RST:

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives high for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

ALE/PROG:

Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

PSEN:

Program Store Enable (PSEN) is the read strobe to external program memory. When the AT89S52 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

EA/VPP:

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset. EA should be strapped to VCC for internal program executions. This pin also receives the 12-volt programming enable voltage (VPP) during Flash programming.

XTAL1:

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2:

Output from the inverting oscillator amplifier

Oscillator Characteristics:

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 6.2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

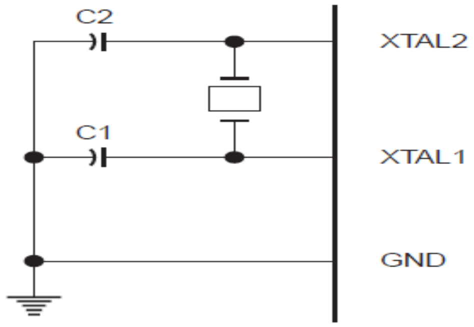


FIG: Oscillator Connections

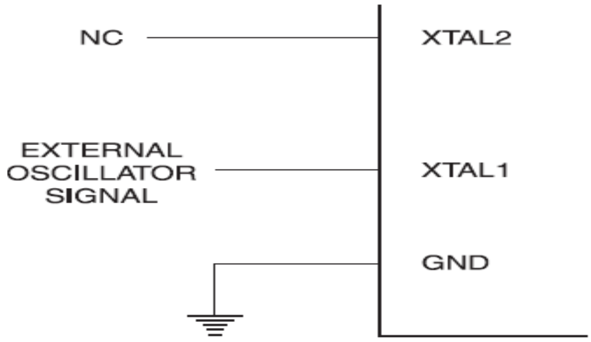


FIG8: External Clock Drive Configuration

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Power down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before VCC is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

3.9 RF MODULES

What is RF?

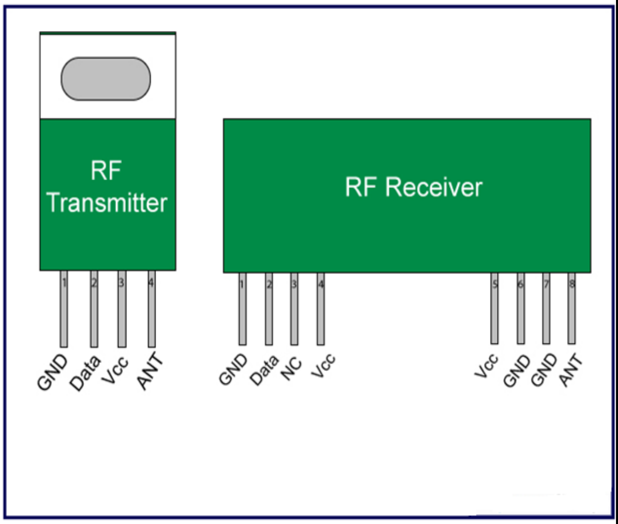
RF itself has become synonymous with wireless and high-frequency signals, describing anything from AM radio between 535 kHz and 1605 kHz to computer local area networks (LANs) at 2.4 GHz. However, RF has traditionally defined frequencies from a few kHz to roughly 1 GHz. If one considers microwave frequencies as RF, this range extends to 300 GHz. The following two tables outline the various nomenclatures for the frequency bands. The third table outlines some of the applications at each of the various frequency bands.

Table 2: Frequency Band Designations

f	λ	Band	Description
30–300 Hz	10^4–10^3 km	ELF	Extremely low frequency
300–3000 Hz	10^3–10^2 km	VF	Voice frequency
3–30 kHz	100–10 km	VLF	Very low frequency
30–300 kHz	10–1 km	LF	Low frequency
0.3–3 MHz	1–0.1 km	MF	Medium frequency
3–30 MHz	100–10 m	HF	High frequency
30–300 MHz	10–1 m	VHF	Very high frequency
300–3000 MHz	100–10 cm	UHF	Ultra-high frequency
3–30 GHz	10–1 cm	SHF	Superhigh frequency
30–300 GHz	10–1 mm	EHF	Extremely high frequency (millimeter waves)

Features

- Range in open space (Standard Conditions) : 100 Meters
- RX Receiver Frequency: 433 MHz
- RX Typical Sensitivity: 105 Dbm
- RX Supply Current: 3.5 mA
- RX IF Frequency: 1MHz
- Low Power Consumption
- Easy For Application
- RX Operating Voltage: 5V
- TX Frequency Range: 433.92 MHz
- TX Supply Voltage: 3V ~ 6V
- TX Out Put Power: 4 ~ 12 Dbm



RF TRANSMITTER:

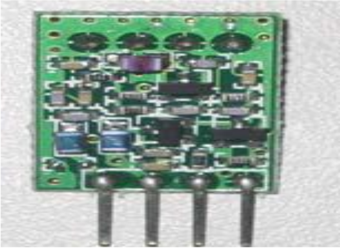


Fig 9: 315/433 MHz TRANSMITTER

General Description:

The ST-TX01-ASK is an ASK Hybrid transmitter module. ST-TX01-ASK is designed by the Saw Resonator, with an effective low cost, small size, and simple-to-use for designing.

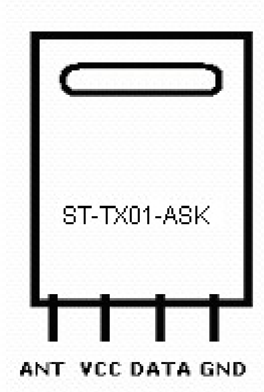
Frequency Range: 315 / 433.92 MHZ.

Supply Voltage: 3~12V.

Output Power: 4~16dBm

Circuit Shape: Saw

PIN Description



Absolute Maximum Ratings

Parameter	Symbol	Condition		Specification				Unit
				Min.	Typical			
Operation Voltage					3V	5V	12V	V
Output power	Psens	DATA 5V 1Kbps Data Rate	315MHz		4	10	16	dBm
			Supply current		11	20	57	mA
			434MHz		4	10	16	dBm
			Supply current		11	22	59	mA
Tune on Time	Ton	Data start out by Vcc turn on		10	20			ms
Data Rate				200	1k			bps
Input duty		Vcc=5V; 1kbps data rate		40				%
Temperature				-20				℃

Applications

*Wireless security systems

*Car Alarm systems

*Remote controls.

*Sensor reporting

RF RECEIVER:

Fig 10: 315/434 MHz ASK RECEIVER

General Description:

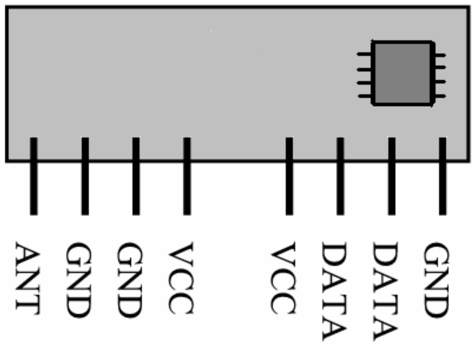
The ST-RX02-ASK is an ASK Hybrid receiver module. A effective low cost solution for using at 315/433.92 MHZ. The circuit shape of ST-RX02-ASK is L/C. Receiver Frequency: 315 / 433.92 MHZ Typical sensitivity: -105dBm

Supply Current: 3.5mA IF Frequency: 1MHz

Features:

- Low power consumption
- Easy for application
- Operation temperature range: －20℃～＋70℃
- Operation voltage: 5 Volts.
- Available frequency at: 315/434 MHz

Pin Description



3.10 RF ENCODER AND DECODER

RF ENCODER HT 12E:

Features

- Operating voltage is 2.4V~12V for the HT12E.
- Low power and high noise immunity CMOS technology
- Low standby current: 0.1_A (typ.) at VDD=5V
- HT12A with a 38kHz carrier for infrared transmission medium
- Minimum transmission words are Four words for the HT12E
- Built-in oscillator needs only 5% resistor
- Data code has positive polarity
- Minimal external components
- HT12A/E: 18-pin DIP/20-pin SOP package

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

GENERAL DESCRIPTION:

The RF encoders are a series of CMOS LSIs for remote control system m applications. They are capable of encoding information which consists of N address bits and 12_N data bits. Each address/ data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header bits via an RF or an infrared transmission medium. Upon receipt of a trigger signal. The capability to select a TE trigger on the HT12E or a DATA trigger on the HT12A further enhances the application for flexibility of the 2^12 series of encoders. The HT12A additionally provides a 38 kHz carrier for infrared systems.

TE trigger

HT12E

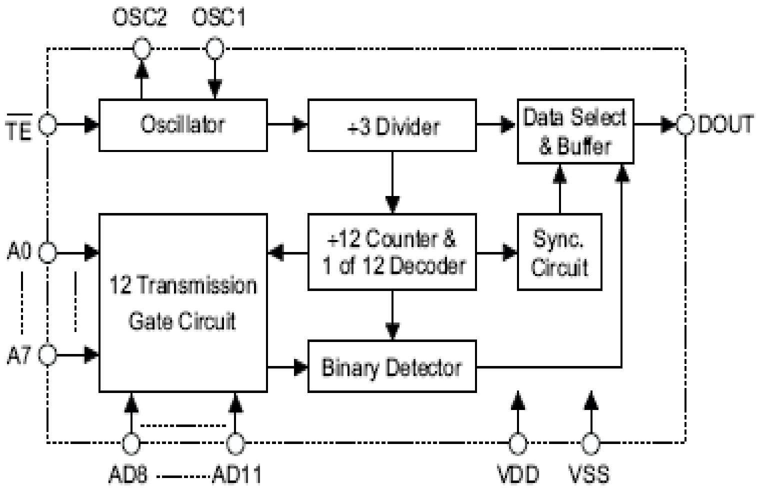


FIG 11: BLOCK DIAGRAM OF HT 12E

PIN DIAGRAMS

FIG 12: PIN DIAGRAM OF HT12E

PIN DESCRIPTION:

Pin Name	I/O	Internal Connection	Description
A0~A7	I	CMOS IN Pull-high (HT12A) NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address A0~A7 setting These pins can be externally set to VSS or left open
AD8~AD11	I	NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address/data AD8~AD11 setting These pins can be externally set to VSS or left open
D8~D11	I	CMOS IN Pull-high	Input pins for data D8~D11 setting and transmission enable, active low These pins should be externally set to VSS or left open (see Note)
DOUT	O	CMOS OUT	Encoder data serial transmission output
L/MB	I	CMOS IN Pull-high	Latch/Momentary transmission format selection pin: Latch: Floating or VDD Momentary: VSS

FIG 13: PIN DESCRIPTION OF HT12E

RF DECODER(HT 12D):

Features

- Operating voltage: 2.4V~12V.
- Low power and high noise immunity CMOS technology.
- Low standby current.
- Capable of decoding 12 bits of information.
- Binary address setting.
- Received codes are checked 3 times.
- Address/Data number combination for HT12D: 8 address bits and 4 data bits.
- Built-in oscillator needs only 5% resistor
- Valid transmission indicator
- Easy interface with an RF or an infrared transmission medium
- Minimal external components
- Pair with Holtek's 212 series of encoders
- 18-pin DIP, 20-pin SOP package

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 212 decoders are a series of CMOS LSIs for remote control system applications. They are paired with Holtek's 212 series of encoders (refer to the encoder/decoder cross reference table). For proper operation, a pair of encoder/decoder with the same number of addresses and data format should be chosen. The decoders receive serial addresses and data from a programmed 212 series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. They compare the serial input data three times continuously with their local addresses. If no error or unmatched codes are found; the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission. The 212 series of decoders are capable of decoding information that consists of N bits of address and 12_N bits of data. Of this series, the HT12D is arranged to provide 8 address bits and 4 data bits, and HT12F is used to decode 12 bits of address information

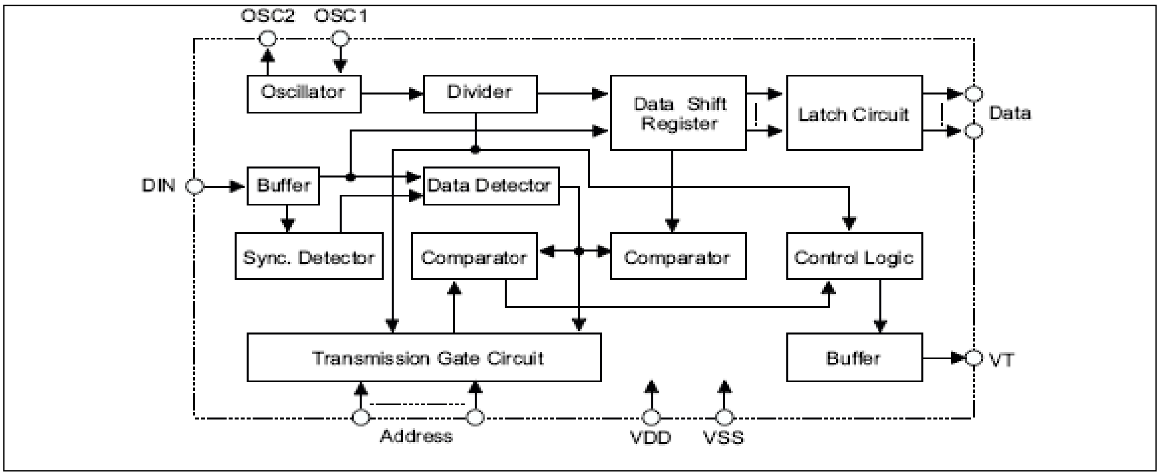


FIG 14: BLOCK DIAGRAM OF HT 12D

PIN DIAGRAMS:

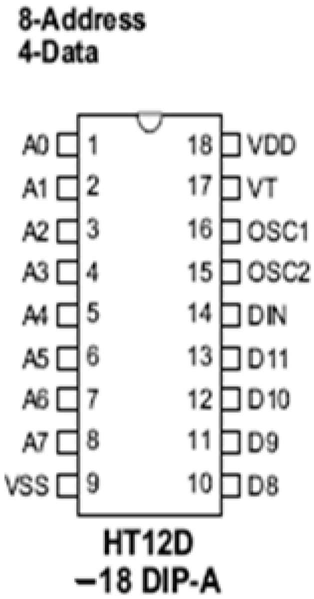


FIG 15: PIN DIAGRAM

PIN DESCRIPTION:

Pin Description

Pin Name	I/O	Internal Connection	Description
A0~A11 (HT12F)	I	NMOS Transmission Gate	Input pins for address A0~A11 setting These pins can be externally set to VSS or left open.
A0~A7 (HT12D)			Input pins for address A0~A7 setting These pins can be externally set to VSS or left open.
D8~D11 (HT12D)	O	CMOS OUT	Output data pins, power-on state is low.
DIN	I	CMOS IN	Serial data input pin
VT	O	CMOS OUT	Valid transmission, active high
OSC1	I	Oscillator	Oscillator input pin
OSC2	O	Oscillator	Oscillator output pin
VSS	—	—	Negative power supply, ground
VDD	—	—	Positive power supply

FIG 16: PIN DESCRIPTION

3.11 LIQUID CRYSTAL DISPLAY(LCD)

Description:

This is the first interfacing example for the Parallel Port. We will start with something simple. This example doesn't use the Bi-directional feature found on newer ports, thus it should work with most, if not all Parallel Ports. It however doesn't show the use of the Status Port as an input for a 16 Character x 2 Line LCD Module to the Parallel Port. These LCD Modules are very common these days, and are quite simple to work with, as all the logic required running them is on board.

LCD Background:

Frequently, an 8051 program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an 8051 is an LCD display. Some of the most common LCDs connected to the 8051 are 16x2 and 20x2 displays. This means 16 characters per line by 2 lines and 20 characters per line by 2 lines, respectively.

Fortunately, a very popular standard exists which allows us to communicate with the vast majority of LCDs regardless of their manufacturer. The standard is referred to as HD44780U, which refers to the controller chip which receives data from an external source (in this case, the 8051) and communicates directly with the LCD.

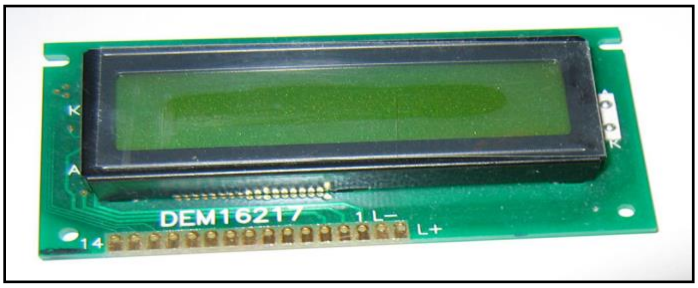


FIG 17: LCD

44780 LCD BACKGROUND

The 44780 standard requires 3 control lines as well as either 4 or 8 I/O lines for the data bus. The user may select whether the LCD is to operate with a 4-bit data bus or an 8-bit data bus. If a 4-bit data bus is used the LCD will require a total of 7 data lines (3 control lines plus the 4 lines for the data bus). If an 8-bit data bus is used the LCD will require a total of 11 data lines (3 control lines plus the 8 lines for the data bus).

The three control lines are referred to as **EN**, **RS**, and **RW**.

The **EN** line is called "Enable." This control line is used to tell the LCD that you are sending it data. To send data to the LCD, your program should make sure this line is low (0) and then set the other two control lines and/or put data on the data bus. When the other lines are completely ready, bring **EN** high (1) and wait for the minimum amount of time required by the LCD datasheet (this varies from LCD to LCD), and end by bringing it low (0) again.

The **RS** line is the "Register Select" line. When RS is low (0), the data is to be treated as a command or special instruction (such as clear screen, position cursor, etc.). When RS is high (1), the data being sent is text data which should be displayed on the screen. For example, to display the letter "T" on the screen you would set RS high.

The **RW** line is the "Read/Write" control line. When RW is low (0), the information on the data bus is being written to the LCD. When RW is high (1), the program is effectively querying (or reading) the LCD. Only one instruction ("Get LCD status") is a read command. All others are write commands--so RW will almost always be low .Finally, the data bus consists of 4 or 8 lines (depending on the mode of operation selected by the user). In the case of an 8-bit data bus, the lines are referred to as DB0, DB1, DB2, DB3, DB4, DB5, DB6, and DB7.

3.12 1N4007

Diodes are used to convert AC into DC these are used as half wave rectifier or full wave rectifier. Three points must he kept in mind while using any type of diode.

1.Maximum forward current capacity

2.Maximum reverse voltage capacity

3.Maximum forward voltage capacity

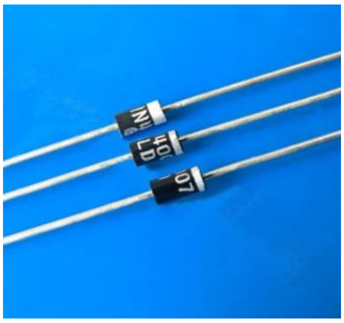


Fig18: 1N4007 diodes

The number and voltage capacity of some of the important diodes available in the market are as follows:

- Diodes of number IN4001, IN4002, IN4003, IN4004, IN4005, IN4006 and IN4007 have maximum reverse bias voltage capacity of 50V and maximum forward current capacity of 1 Amp.
- Diode of same capacities can be used in place of one another. Besides this diode of more capacity can be used in place of diode of low capacity but diode of low capacity cannot be used in place of diode of high capacity. For example, in place of IN4002; IN4001 or IN4007 can be used but IN4001 or IN4002 cannot be used in place of IN4007.The diode BY125made by company BEL is equivalent of diode from IN4001 to IN4003. BY 126 is equivalent to diodes IN4004 to 4006 and BY 127 is equivalent to diode IN4007.

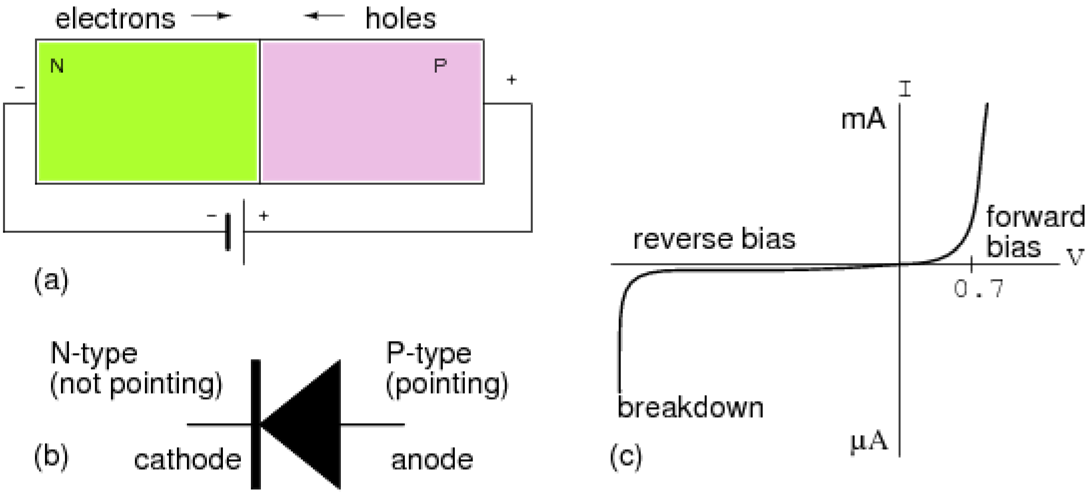


Fig19: PN Junction diode

PN JUNCTION OPERATION

Now that you are familiar with P- and N-type materials, how these materials are joined together to form a diode, and the function of the diode, let us continue our discussion with the operation of the PN junction. But before we can understand how the PN junction works, we must first consider current flow in the materials that make up the junction and what happens initially within the junction when these two materials are joined together.

Current Flow in the N-Type Material

Conduction in the N-type semiconductor, or crystal, is similar to conduction in a copper wire. That is, with voltage applied across the material, electrons will move through the crystal just as current would flow in a copper wire. This is shown in figure 1-15. The positive potential of the battery will attract the free electrons in the crystal. These electrons will leave the crystal and flow into the positive terminal of the battery. As an electron leaves the crystal, an electron from the negative terminal of the battery will enter the crystal, thus completing the current path. Therefore, the majority current carriers in the N-type material (electrons) are repelled by the negative side of the battery and move through the crystal toward the positive side of the battery.

Current Flow in the P-Type Material

Current flow through the P-type material is illustrated. Conduction in the P material is by positive holes, instead of negative electrons. A hole moves from the positive terminal of the P material to the negative terminal. Electrons from the external circuit enter the negative terminal of the material and fill holes in the vicinity of this terminal. At the positive terminal, electrons are removed from the covalent bonds, thus creating new holes. This process continues as the steady stream of holes (hole current) moves toward the negative terminal

3.13 MAX 232

The MAX232 is an integrated circuit that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. The MAX232 is a dual driver/receiver and typically converts the RX, TX, CTS and RTS signals.

The drivers provide RS-232 voltage level outputs (approx. ± 7.5 V) from a single + 5 V supply via on-chip charge pumps and external capacitors. This makes it useful for implementing RS-232 in devices that otherwise do not need any voltages outside the 0 V to + 5 V range, as power supply design does not need to be made more complicated just for driving the RS-232 in this case. The receivers reduce RS-232 inputs (which may be as high as ± 25 V), to standard 5 V TTL levels. These receivers have a typical threshold of 1.3 V, and a typical hysteresis of 0.5 V.

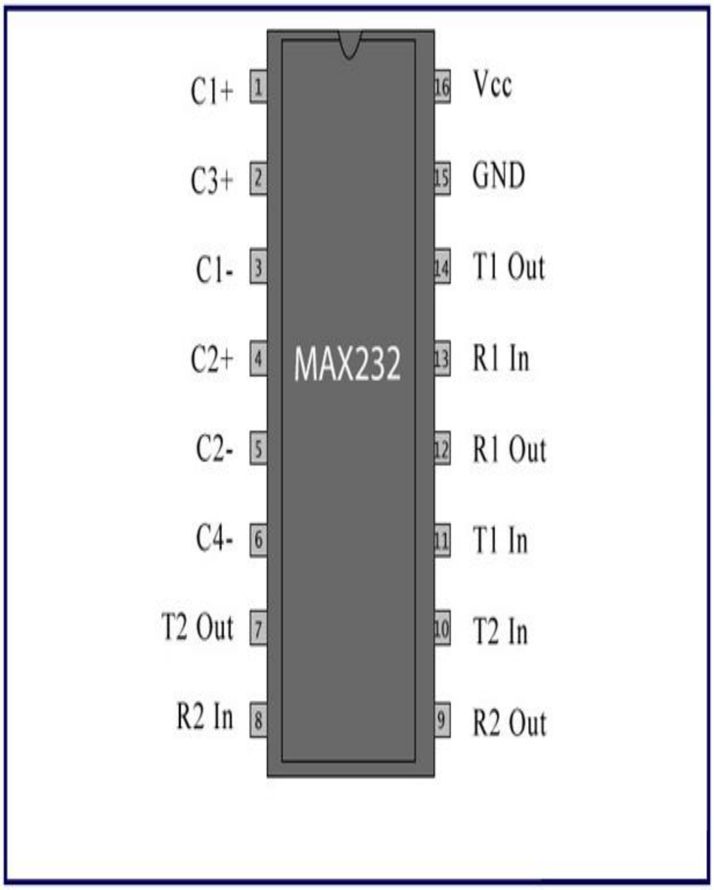
The later MAX232A is backwards compatible with the original MAX232 but may operate at higher baud rates and can use smaller external capacitors (0.1 μF) in place of the 1.0 μF capacitors used with the original device. The newer MAX3232 is also backwards compatible, but operates at a broader voltage range, from 3 to 5.5V.

Voltage levels:

It is helpful to understand what occurs to the voltage levels. When a MAX232 IC receives a TTL level to convert, it changes a TTL Logic 0 to between +3 and +15V, and changes TTL Logic 1 to between -3 to -15V, and vice versa for converting from RS232 to TTL.

This can be confusing when you realize that the RS232 Data Transmission voltages at a certain logic state are opposite from the RS232 Control Line voltages at the same logic state. To clarify the matter, see the table below. For more information see RS-232 Voltage Levels.

RS232 Line Type & Logic Level	RS232 Voltage	TTL Voltage to/from MAX232
Data Transmission (Rx/Tx) Logic 0	+3V to +15V	0V
Data Transmission (Rx/Tx) Logic 1	-3V to -15V	5V
Control Signals (RTS/CTS/DTR/DSR) Logic 0	-3V to -15V	5V
Control Signals (RTS/CTS/DTR/DSR) Logic 1	+3V to +15V	0V



Pin Description:

Pin No	Function	Name
1	Capacitor connection pins	Capacitor 1 +
2		Capacitor 3 +
3		Capacitor 1 -
4		Capacitor 2 +
5		Capacitor 2 -
6		Capacitor 4 -
7	Output pin; outputs the serially transmitted data at RS232 logic level; connected to receiver pin of PC serial port	T_2 Out
8	Input pin; receives serially transmitted data at RS 232 logic level; connected to transmitter pin of PC serial port	R_2 In
9	Output pin; outputs the serially transmitted data at TTL logic level; connected to receiver pin of controller.	R_2 Out
10	Input pins; receive the serial data at TTL logic level; connected to serial transmitter pin of controller.	T_2 In
11		T_1 In
12	Output pin; outputs the serially transmitted data at TTL logic level; connected to receiver pin of controller.	R_1 Out
13	Input pin; receives serially transmitted data at RS 232 logic level; connected to transmitter pin of PC serial port	R_1 In
14	Output pin; outputs the serially transmitted data at RS232 logic level; connected to receiver pin of PC serial port	T_1 Out
15	Ground (0V)	Ground
16	Supply voltage; 5V (4.5V – 5.5V)	Vcc

Application:

The MAX232 has two receivers (converts from RS-232 to TTL voltage levels) and two drivers (converts from TTL logic to RS-232 voltage levels). This means only two of the RS-232 signals can be converted in each direction.

Typically, a pair of a driver/receiver of the MAX232 is used for

- TX and RX

And the second one for

- CTS and RTS.

There are not enough drivers/receivers in the MAX232 to also connect the DTR, DSR, and DCD signals. Usually, these signals can be omitted when e.g., communicating with a PC's serial interface. If the DTE really requires these signals either a second MAX232 is needed, or some other IC from the MAX232 family can be used.

3.14 SOFTWARE REQUIREMENTS

3.14.1 INTRODUCTION TO KEIL MICRO VISION (IDE)

Keil an ARM Company makes C compilers, macro assemblers, real-time kernels, debuggers, simulators, integrated environments, evaluation boards, and emulators for ARM7/ARM9/Cortex-M3, XC16x/C16x/ST10, 251, and 8051 MCU families.

Keil development tools for the 8051 Microcontroller Architecture support every level of software developer from the professional applications engineer to the student just learning about embedded software development. When starting a new project, simply select the microcontroller you use from the Device Database and the µVision IDE sets all compiler, assembler, linker, and memory options for you.

Keil is a cross compiler. So first we have to understand the concept of compilers and cross compilers. After then we shall learn how to work with keil.

3.14.2 CONCEPT OF COMPILER

Compilers are programs used to convert a High Level Language to object code. Desktop compilers produce an output object code for the underlying microprocessor, but not for other microprocessors. I.E the programs written in one of the HLL like 'C' will compile the code to run on the system for a particular processor like x86 (underlying microprocessor in the computer). For example compilers for Dos platform is different from the Compilers for Unix platform So if one wants to define a compiler then compiler is a program that translates source code into object code.

The compiler derives its name from the way it works, looking at the entire piece of source code and collecting and reorganizing the instruction. See there is a bit little difference between compiler and an interpreter. Interpreter just interprets whole program at a time while compiler analyses and execute each line of source code in succession, without looking at the entire program.

The advantage of interpreters is that they can execute a program immediately. Secondly programs produced by compilers run much faster than the same programs executed by an interpreter. However compilers require some time before an executable program emerges. Now as compilers translate source code into object code, which is unique for each type of computer, many compilers are available for the same language.

3.14.3 CONCEPT OF CROSS COMPILER

A cross compiler is similar to the compilers but we write a program for the target processor (like 8051 and its derivatives) on the host processors (like computer of x86). It means being in one environment you are writing a code for another environment is called cross development. And the compiler used for cross development is called cross compiler. So the definition of cross compiler is a compiler that runs on one computer but produces object code for a different type of computer.

3.14.4 KEIL C CROSS COMPILER

Keil is a German based Software development company. It provides several development tools like

- IDE (Integrated Development environment)
- Project Manager
- Simulator
- Debugger
- C Cross Compiler, Cross Assembler, Locator/Linker

The Keil ARM tool kit includes three main tools, assembler, compiler and linker. An assembler is used to assemble the ARM assembly program. A compiler is used to compile the C source code into an object file. A linker is used to create an absolute object module suitable for our in-circuit emulator.

3.14.5 Building an application in µVision2

To build (compile, assemble, and link) an application in µVision2, you must:

1. Select Project -(for example, 166\EXAMPLES\HELLO\HELLO.UV2).
2. Select Project - Rebuild all target files or Build target.µVision2 compiles, assembles, and links the files in your project.

3.14.6 Creating Your Own Application in µVision2

To create a new project in µVision2, you must:

1. Select Project - New Project.
2. Select a directory and enter the name of the project file.
3. Select Project - Select Device and select an 8051, 251, or C16x/ST10 device from the Device Database™.

4. Create source files to add to the project.
5. Select Project - Targets, Groups, Files. Add/Files, select Source Group1, and add the source files to the project.
6. Select Project - Options and set the tool options. Note when you select the target device from the Device Database™ all special options are set automatically. You typically only need to configure the memory map of your target hardware. Default memory model settings are optimal for most applications.
7. Select Project - Rebuild all target files or Build target.

3.14.7 Debugging an Application in µVision2

To debug an application created using µVision2, you must:

1. Select Debug - Start/Stop Debug Session.
2. Use the Step toolbar buttons to single-step through your program. You may enter G, main in the Output Window to execute to the main C function.
3. Open the Serial Window using the Serial #1 button on the toolbar.

Debug your program using standard options like Step, Go, Break, and so on.

3.14.8 Starting µVision2 and Creating a Project

µVision2 is a standard Windows application and started by clicking on the program icon. To create a new project file select from the µVision2 menu Project – New Project…. This opens a standard Windows dialog that asks you for the new project file name. We suggest that you use a separate folder for each project. You can simply use the icon Create New Folder in this dialog to get a new empty folder. Then select this folder and enter the file name for the new project, i.e. Project1. µVision2 creates a new project file with the name PROJECT1.UV2 which contains a default target and file group name. You can see these names in the Project.

3.14.9 Window – Files.

Now use from the menu Project – Select Device for Target and select a CPU for your project. The Select Device dialog box shows the µVision2 device data base. Just select the microcontroller you use. We are using for our examples the Philips 80C51RD+ CPU. This selection sets necessary tool Options for the 80C51RD+ device and simplifies in this way the tool Configuration.

3.14.10 Building Projects and Creating a HEX Files

Typical, the tool settings under Options – Target are all you need to start a new application. You may translate all source files and line the application with a click on the Build Target toolbar icon. When you build an application with syntax errors, µVision2 will display errors and warning messages in the Output Window – Build page. A double click on a message line opens the source file on the correct location in a µVision2 editor window. Once you have successfully generated your application you can start debugging.

After you have tested your application, it is required to create an Intel HEX file to download the software into an EPROM programmer or simulator. µVision2 creates HEX files with each build process when Create HEX files under Options for Target – Output is enabled. You may start your PROM programming utility after the make process when you specify the program under the option Run User Program #1.

3.14.11 CPU Simulation

µVision2 simulates up to 16 Mbytes of memory from which areas can be mapped for read, write, or code execution access. The µVision2 simulator traps and reports illegal memory access. In addition to memory mapping, the simulator also provides support for the integrated peripherals of the various 8051 derivatives. The on-chip peripherals of the CPU you have selected are configured from the Device.

3.14.12 Database selection

You have made when you create your project target. Refer to page 58 for more Information about selecting a device. You may select and display the on-chip peripheral components using the Debug menu. You can also change the aspects of each peripheral using the controls in the dialog boxes.

3.14.13 Start Debugging

You start the debug mode of µVision2 with the Debug – Start/Stop Debug Session Command. Depending on the Options for Target – Debug Configuration, µVision2 will load the application program and run the startup code µVision2 saves the editor screen layout and restores the screen layout of the last debug session. If the program execution stops, µVision2 opens an editor window with the source text or shows CPU instructions in the disassembly window. The next executable statement is marked with a yellow arrow. During debugging, most editor features are still available.

For example, you can use the find command or correct program errors. Program source text of your application is shown in the same windows. The µVision2 debug mode differs from the edit mode in the following aspects:

_ The “Debug Menu and Debug Commands” described on page 28 are available. The additional debug windows are discussed in the following.

_ The project structure or tool parameters cannot be modified. All build commands are disabled.

3.14.14 Disassembly Window

The Disassembly window shows your target program as mixed source and assembly program or just assembly code. A trace history of previously executed instructions may be displayed with Debug – View Trace Records. To enable the trace history, set Debug – Enable/Disable Trace Recording.

If you select the Disassembly Window as the active window all program step commands work on CPU instruction level rather than program source lines. You can select a text line and set or modify code breakpoints using toolbar buttons or the context menu commands.

You may use the dialog Debug – Inline Assembly… to modify the CPU instructions. That allows you to correct mistakes or to make temporary changes to the target program you are debugging. Numerous example programs are included to help you get started with the most popular embedded 8051 devices.

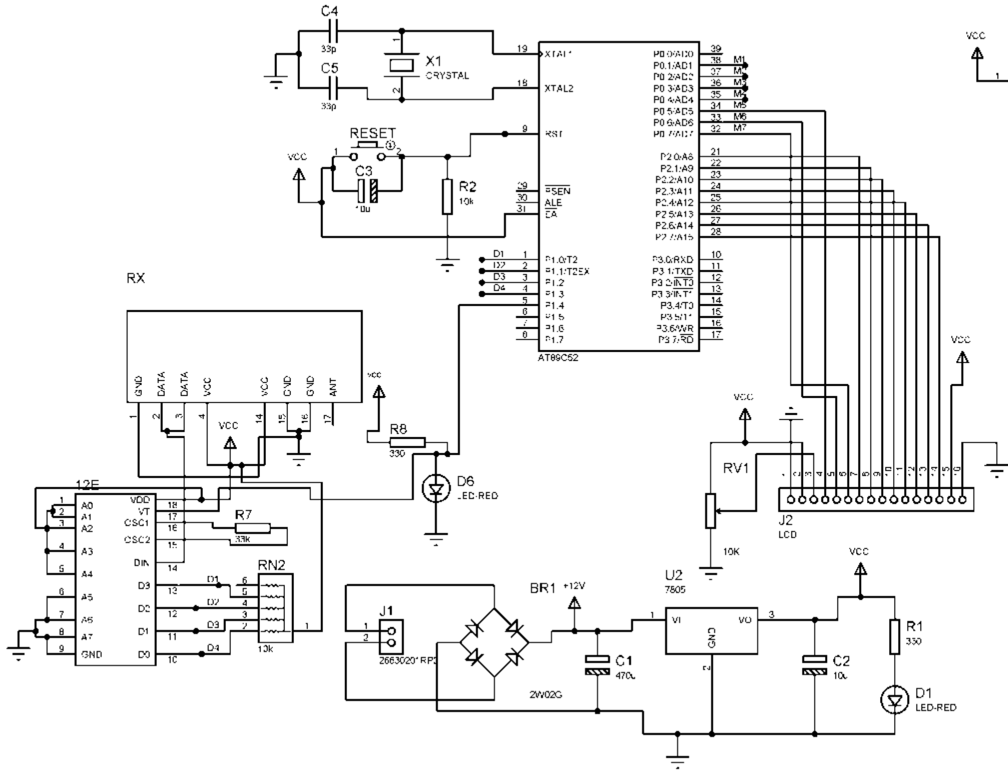
The Keil µVision Debugger accurately simulates on-chip peripherals (I^2C, CAN, UART, SPI, Interrupts, I/O Ports, A/D Converter, D/A Converter, and PWM Modules) of your 8051 device. Simulation helps you understand hardware configurations and avoids time wasted on setup problems. Additionally, with simulation, you can write and test applications before target hardware is available.

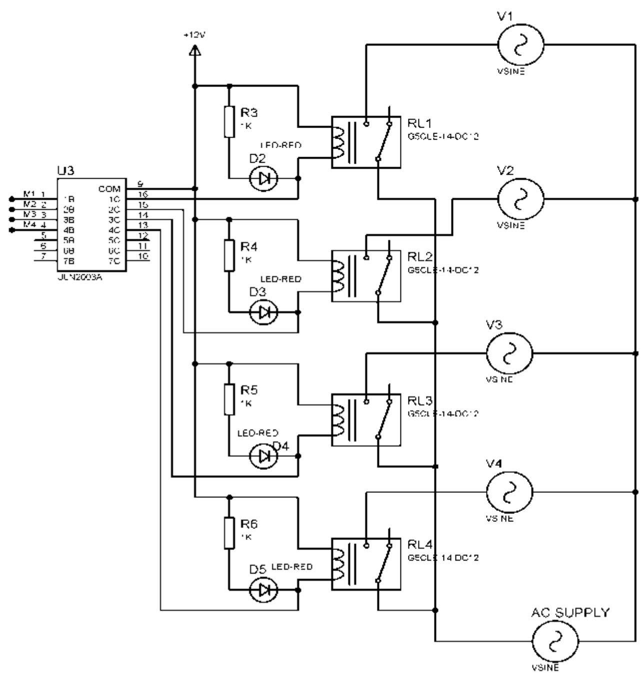
3.14.15 EMBEDDED C

Use of embedded processors in passenger cars, mobile phones, medical equipment, aerospace systems and defense systems is widespread, and even everyday domestic appliances such as dish washers, televisions, washing machines and video recorders now include at least one such device.

Because most embedded projects have severe cost constraints, they tend to use low-cost processors like the 8051 family of devices considered in this book. These popular chips have very limited resources available most such devices have around 256 bytes (not megabytes!) of RAM, and the available processor power is around 1000 times less than that of a desktop processor. As a result, developing embedded software presents significant new challenges, even for experienced desktop programmers. If you have some programming experience - in C, C++ or Java - then this book and its accompanying CD will help make your move to the embedded world as quick and painless as possible.

3.15. SCHEMATIC DIAGRAM





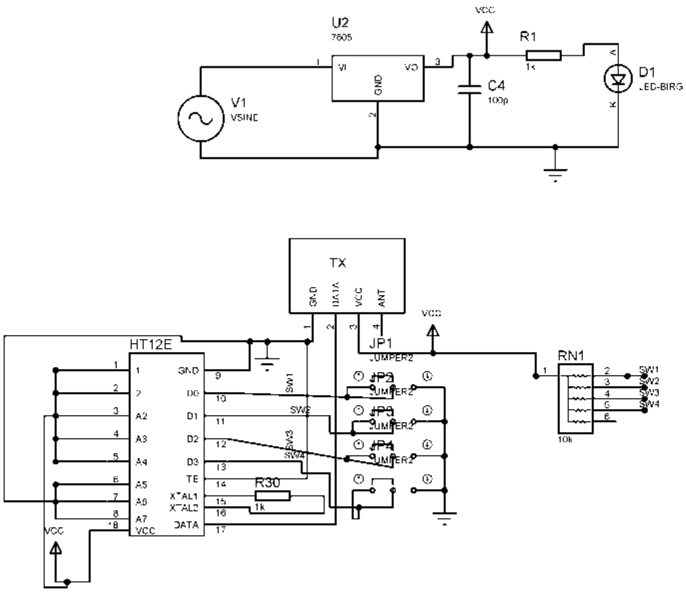


FIG 20: Schematic Diagram

3.15.1 DESCRIPTION

POWER SUPPLY

The circuit uses standard power supply comprising of a step-down transformer from 230Vto 12V and 4 diodes forming a bridge rectifier that delivers pulsating dc which is then filtered by an electrolytic capacitor of about 470µF to 1000µF. The filtered dc being unregulated, IC LM7805 is used to get 5V DC constant at its pin no 3 irrespective of input DC varying from 7V to 15V. The input dc shall be varying in the event of input ac at 230volts section varies from 160V to 270V in the ratio of the transformer primary voltage V1 to secondary voltage V2 governed by the formula V1/V2=N1/N2. As N1/N2 i.e., no. of turns in the primary to the no. of turns in the secondary remains unchanged V2 is directly proportional to V1. Thus, if the transformer delivers 12V at 220V input it will give 8.72V at 160V.Similarly at 270V it will give 14. 72V.Thus the dc voltage at the input of the regulator changes from about 8V to 15V because of A.C voltage variation from 160V to 270V the regulator output will remain constant at 5V.

The regulated 5V DC is further filtered by a small electrolytic capacitor of 10µF for any noise so generated by the circuit. One LED is connected of this 5V point in series with a current limiting resistor of 330Ω to the ground i.e., negative voltage to indicate 5V power supply availability. The unregulated 12V point is used for other applications as and when required.

STANDARD CONNECTIONS TO 8051 SERIES MICRO CONTROLLER

ATMEL series of 8051 family of micro controllers need certain standard connections. The actual number of the Microcontroller could be “89C51” , “89C52”, “89S51”, “89S52”, and as regards to 20 pin configuration a number of “89C2051”. The 4 set of I/O ports are used based on the project requirement. Every microcontroller requires a timing reference for its internal program execution therefore an oscillator needs to be functional with a desired frequency to obtain the timing reference as t =1/f.

A crystal ranging from 2 to 20 MHz is required to be used at its pin number 18 and 19 for the internal oscillator. It may be noted here the crystal is not to be understood as crystal oscillator It is just a crystal, while connected to the appropriate pin of the microcontroller it results in oscillator function inside the microcontroller. Typically 11.0592 MHz crystal is used in general for most of the circuits using 8051 series microcontroller. Two small value ceramic capacitors of 33pF each is used as a standard connection for the crystal as shown in the circuit diagram.

RESET

Pin no 9 is provided with an re-set arrangement by a combination of an electrolytic capacitor and a register forming RC time constant. At the time of switch on, the capacitor gets charged, and it behaves as a full short circuit from the positive to the pin number 9. After the capacitor gets fully charged the current stops flowing and pin number 9 goes low which is pulled down by a 10k resistor to the ground. This arrangement of reset at pin 9 going high initially and then to logic 0 i.e., low helps the program execution to start from the beginning. In absence of this the program execution could have taken place arbitrarily anywhere from the program cycle. A pushbutton switch is connected across the capacitor so that at any given time as desired it can be pressed such that it discharges the capacitor and while released the capacitor starts charging again and then pin number 9 goes to high and then back to low, to enable the program execution from the beginning. This operation of high to low of the reset pin takes place in fraction of a second as decided by the time constant R and C.

For example: A 10µF capacitor and a 10kΩ resistor would render a 100ms time to pin number 9 from logic high to low, there after the pin number 9 remains low.

External Access(EA):

Pin no 31 of 40 pin 8051 microcontroller termed as EA¯ is required to be connected to 5V for accessing the program form the on-chip program memory. If it is connected to ground then the controller accesses the program from external memory. However as we are using the internal memory it is always connected to +5V.

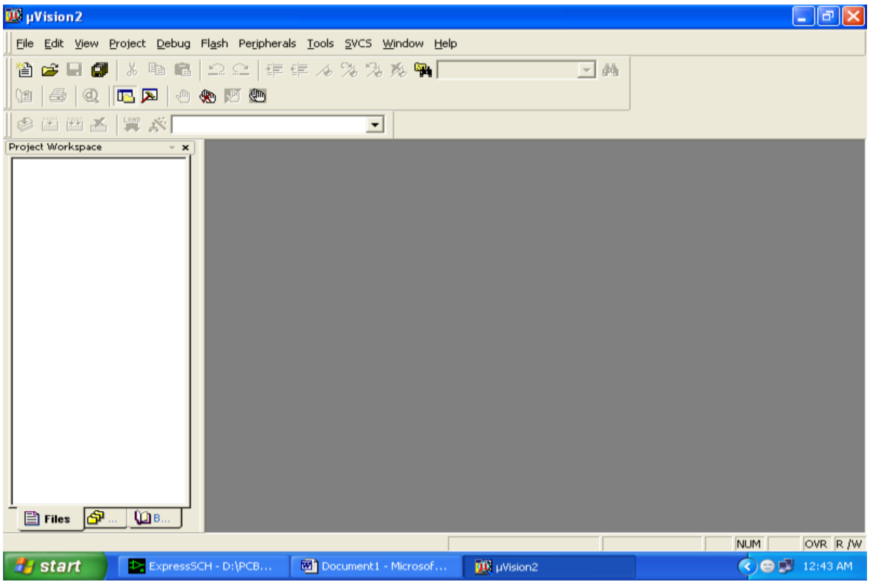
CHAPTER 4

DESIGN PROCESS

1.1 Steps

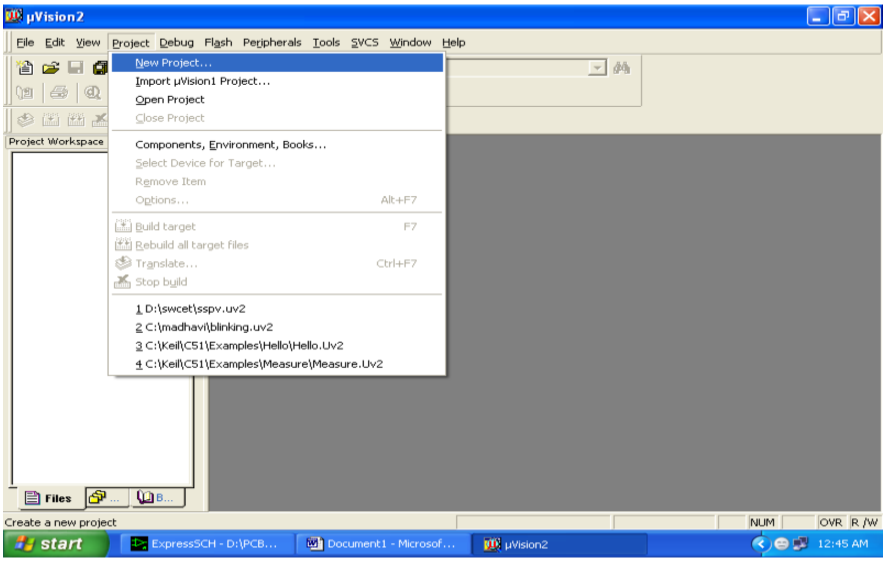
1. Click on the Keil Vision Icon on Desktop

2. The following fig will appear

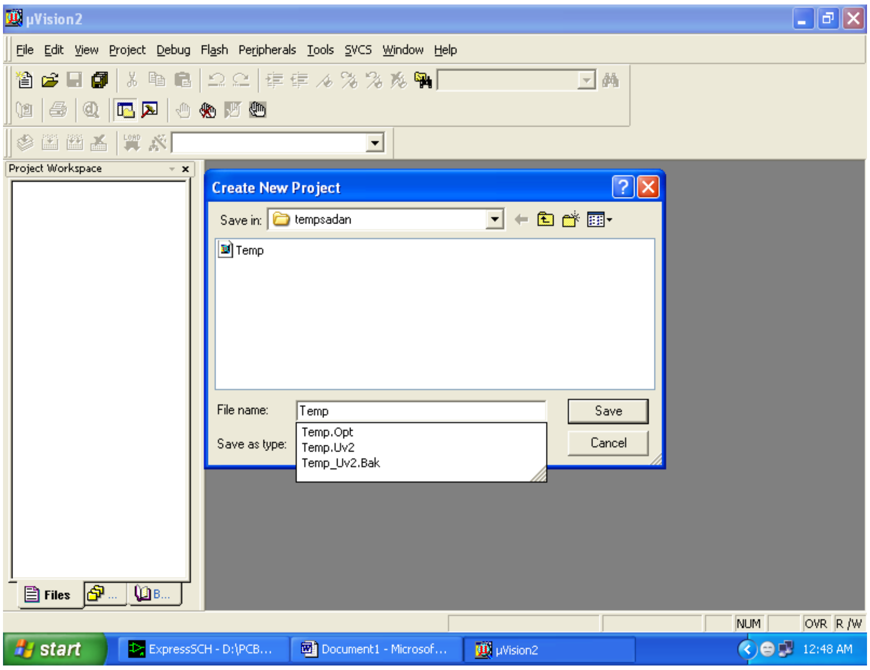


3. Click on the Project menu from the title bar

4. Then Click on New Project



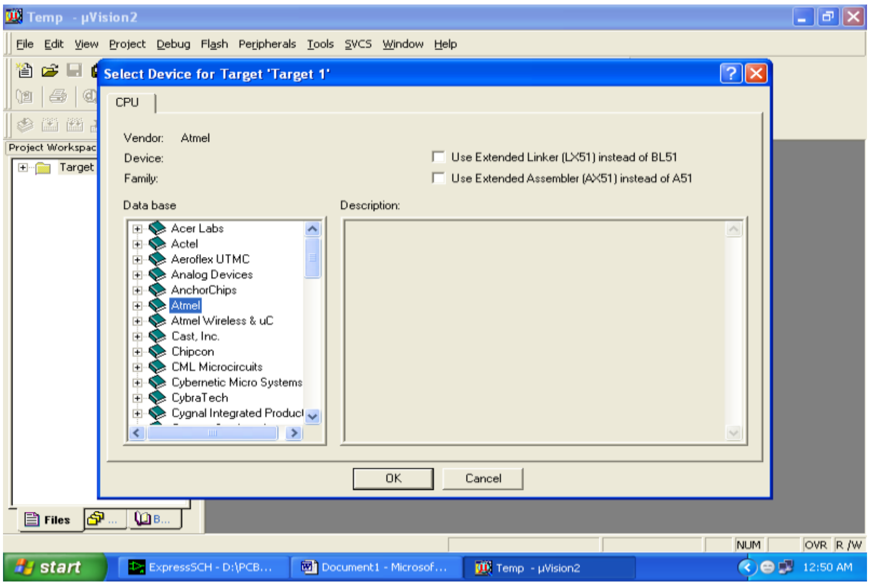
5. Save the Project by typing suitable project name with no extension in u r own folder sited in either C:\ or D:\



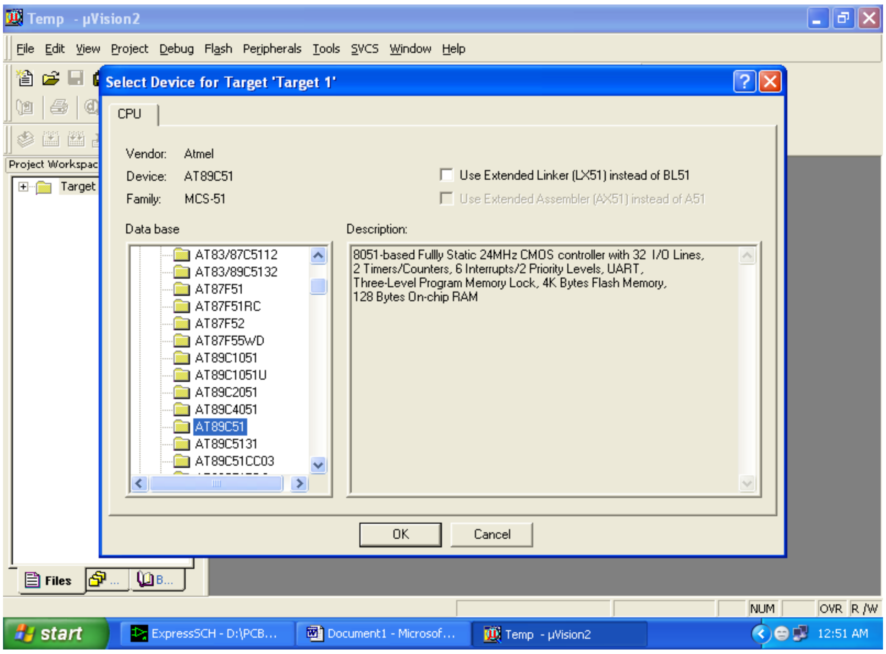
6. Then Click on Save button above.

7. Select the component for u r project. i.e. Atmel……

8. Click on the + Symbol beside of Atmel.

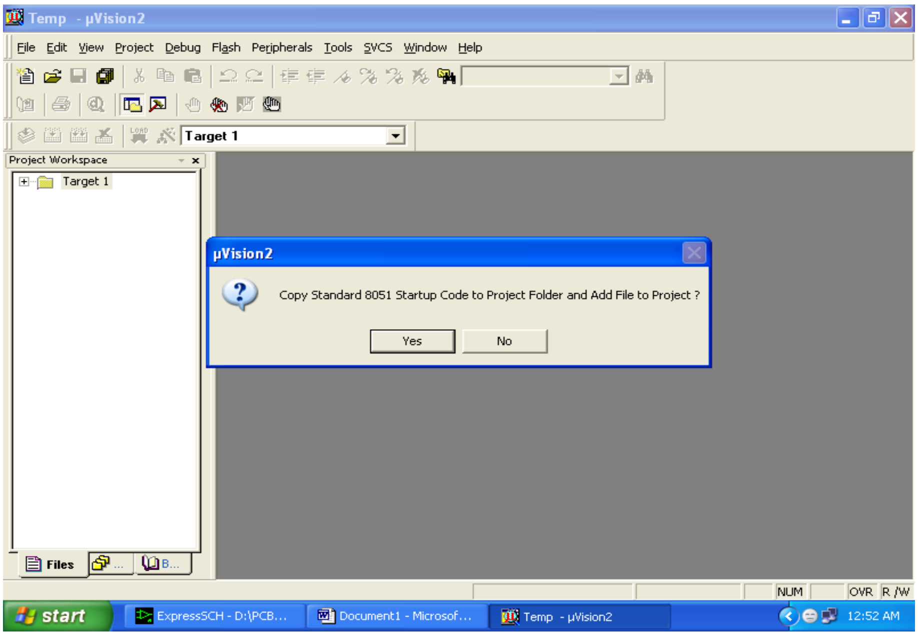


9. Select AT89C51 as shown below.



10. Then Click on "OK".

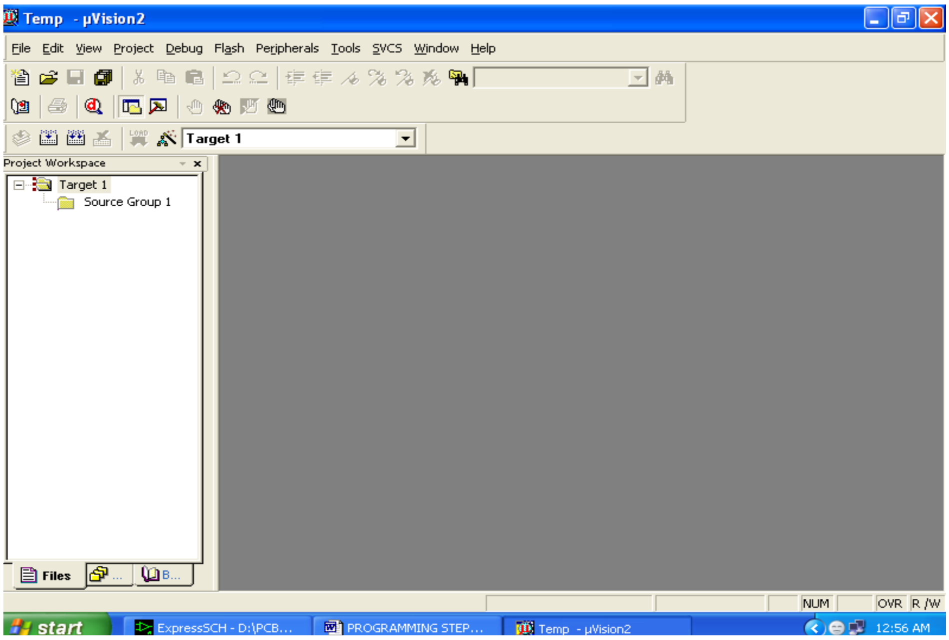
11. The Following fig will appear.



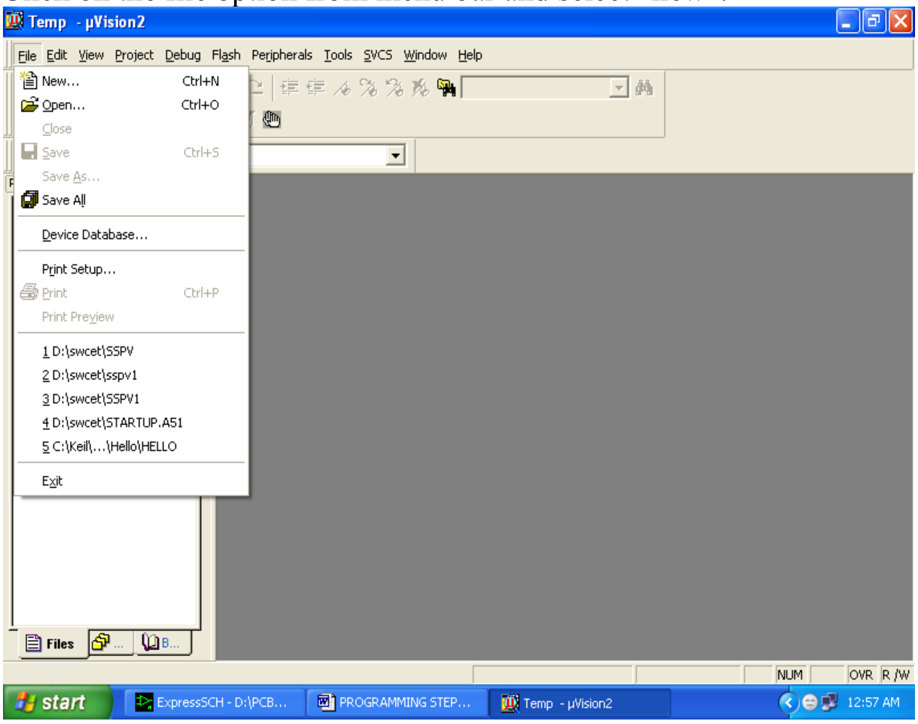
12. Then Click either YES or NO………mostly "NO".

13. Now your project is ready to USE.

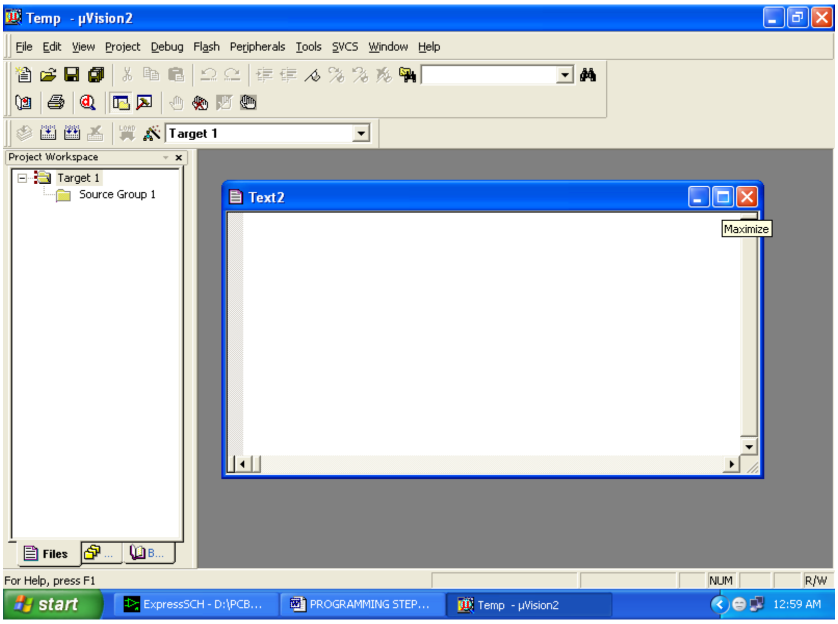
14. Now double click on the Target1, you would get another option "Source group 1" as shown in next page.



15. Click on the file option from menu bar and select "new".

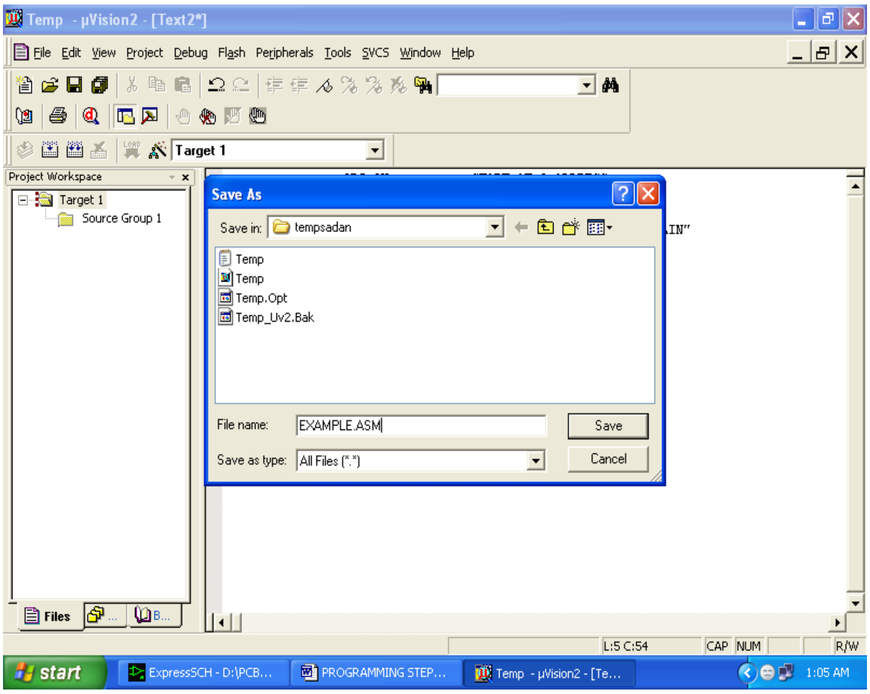


16. The next screen will be as shown in next page, and just maximize it by double clicking on its blue boarder.

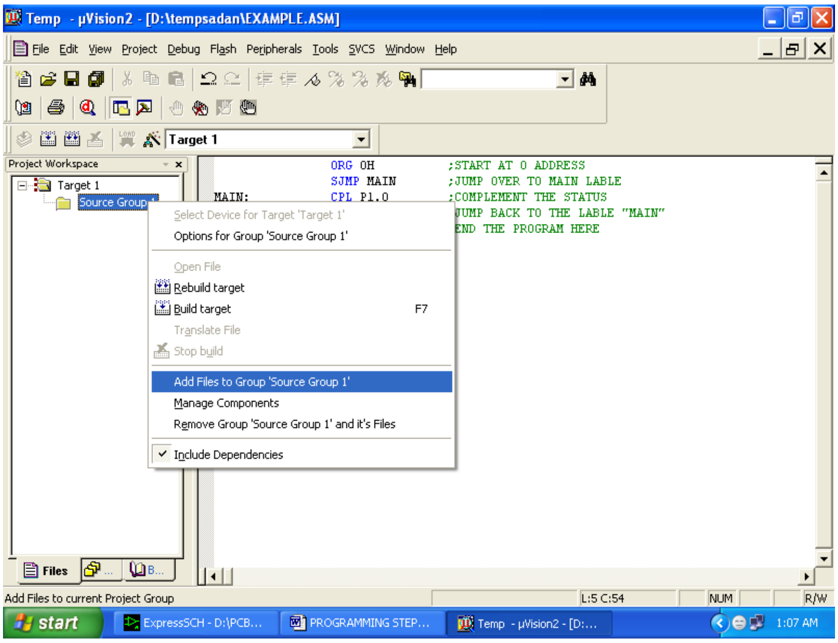


17. Now start writing program in either in "EMBEDDED C" or "ASM".

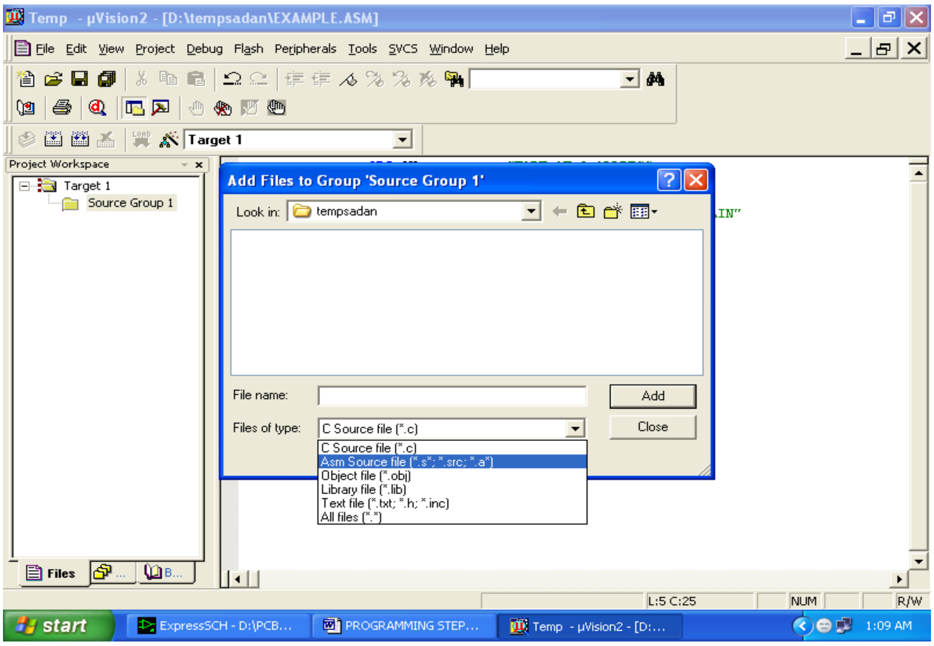
18. For a program written in Assembly, then save it with extension ". asm" and for "EMBEDDED C" based program save it with extension " .C"



19. Now right click on Source group 1 and click on "Add files to Group Source".



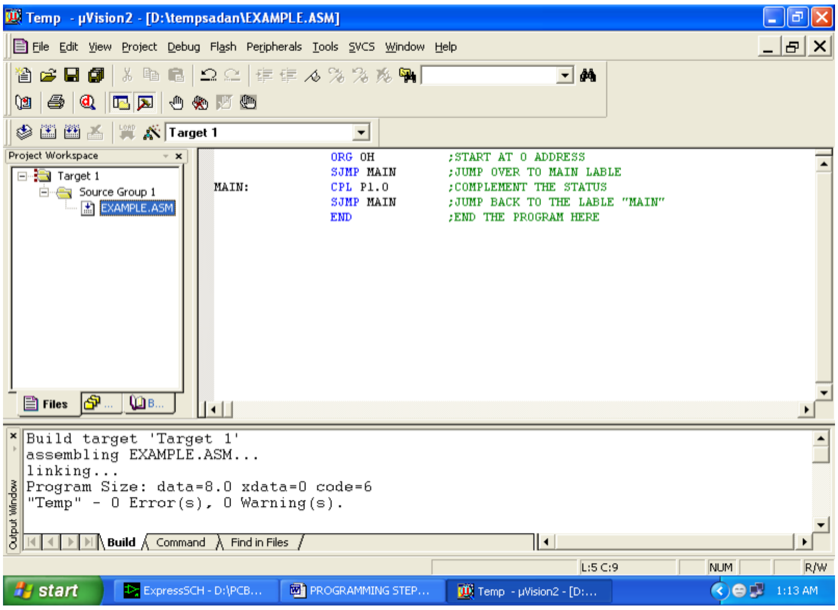
20. Now you will get another window, on which by default "EMBEDDED C" files will appear.



21. Now select as per your file extension given while saving the file.

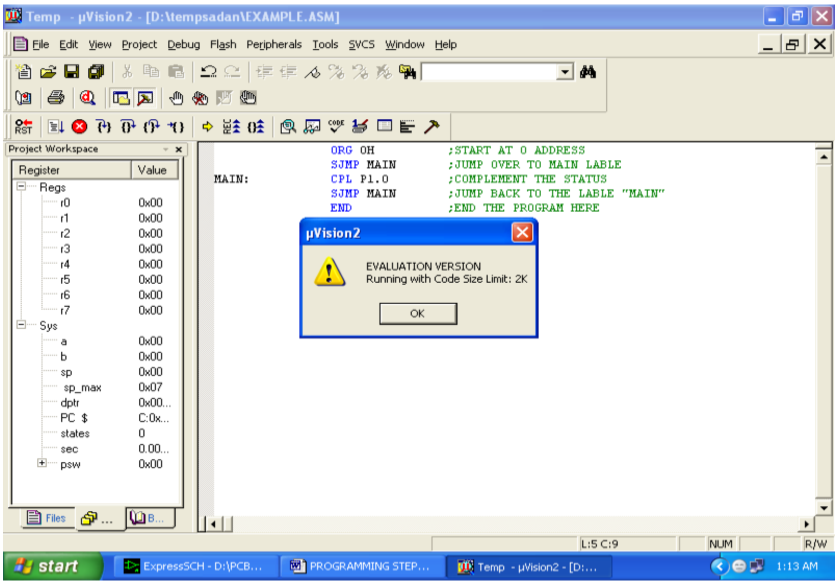
22. Click only one time on option "**ADD**".

23. Now Press function key F7 to compile. Any error will appear if so happen.



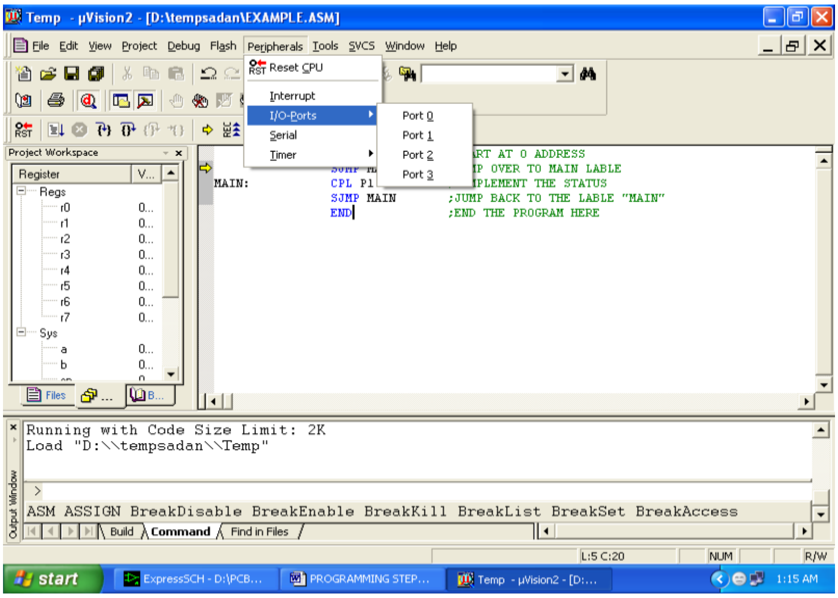
24. If the file contains no error, then press Control+F5 simultaneously.

25. The new window is as follows.

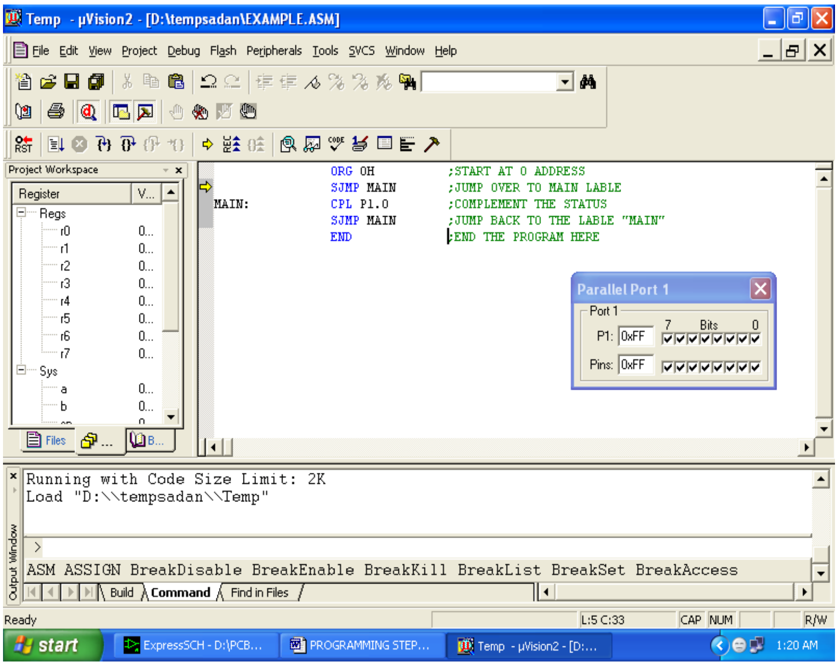


26. Then Click "OK".

27. Now click on the Peripherals from menu bar, and check your required port as shown in fig below.



28. Drag the port a side and click in the program file.



29. Now keep Pressing function key "F11" slowly and observe.

30. You are running your program successfully.

4.2 Source Code

```
# include <at89x52.h>
# define LCDPort P2
```

```
sbit RS=P0^5;
sbit RW=P0^6;
sbit EN=P0^7;

sbit device1=P0^1;
sbit device2=P0^2;
sbit device3=P0^3;
sbit device4=P0^4;


unsigned char;
unsigned int;

bit      ;

void main();
void device_process();
void device_display();
void init();
void init_display();
void init_timer0();
void delay(unsigned int delay_ms);
void lcd_cmd(unsigned char cmd);
void lcd_init();
void lcd_data(unsigned char data1);


void isr_t0() interrupt 1 using 1
        {
                init_timer0();


        }

void main()
{
EA=1;
P0=0x00;
P3=0xff;
device1=0;
device2=0;
device3=0;
device4=0;
```

```
D1=1;
D2=1;
D3=1;
D4=1;
VT=1;

init();
        while(1)
        {
         device_process();
                device_display();
        }

}

void init()
{
        lcd_init();
        init_display();
        init_timer0();
}
void init_display()
        {
                lcd_cmd(0x80);
                lcd_string("    RF BASED ");
                delay(6);
                lcd_cmd(0xC0);
                lcd_string("HOME AUTOMATION");
                delay(500);
        }

void init_timer0()
{
        TMOD|=0x01;
        TL0=0xfe;
        TH0=0x4b;
        TR0=1;
        ET0=1;
}

void delay(unsigned int delay_ms)
{
```

```
		unsigned int count,i;

		for(count = 0;count<delay_ms;count++)
		{
			for(i =0;i < 120;i++);
		}
		delay_ms--;
	}

	void lcd_cmd(unsigned char cmd)
		{
			RS=0;			//rs = 0 for command
			RW=0;				//rw = 0 for writing
			LCDPort=cmd;
			EN=1;			//en=1 enable high
			delay(2);
			EN=0;			//en=0 enable low
		}

	void lcd_data(unsigned char data1)
		{
			RS=1;
			RW=0;
			LCDPort=data1;
			EN=1;
			delay(2);
			EN=0;
		}

	void lcd_string(char *str)
	{
		int i;
		for(i=0;str[i]!='\0';i++) lcd_data(str[i]);
	}

	void lcd_init()
		{



	void device_process()
	{
	 if(D1==0 && VT==1)
```

```
            {
             device1=0;
                        delay(100);
            }
  if(D2==0 && VT==1)
            {
                        device2=0;
                        delay(100);
            }
  if(D3==0 && VT==1)
            {
                        device3=0;
                        delay(100);
            }
  if(D4==0 && VT==1)
            {
                        device4=0;
                        delay(100);

            }
  if(D3==1 && VT==1)
            {
            device3=1;
            delay(100);
            }
  if(D4==1 && VT==1)
            {
                        device4=1;
                        delay(100);
            }
}


void device_display()
{
            lcd_cmd(0x80);
                        lcd_string("D_1= ");
                        lcd_cmd(0x88);
                        lcd_string("D_2= ");
                        lcd_cmd(0xC0);
                        lcd_string("D_3= ");
                        lcd_cmd(0xC8);
                        lcd_string("D_4= ");
```

```
		if(device1==1)
		{
			lcd_cmd(0x84);
			lcd_string("ON  ");
		}
		if(device1==0)
		{
			lcd_cmd(0x84);
			lcd_string("OFF ");
		}
		if(device2==1)
		{
			lcd_cmd(0x8d);
			lcd_string("ON ");
		}
		if(device2==0)
		{
			lcd_cmd(0x8d);
			lcd_string("OFF");
		}
		if(device3==1)
		{
			lcd_cmd(0xC4);
			lcd_string("ON  ");
		}
		if(device3==0)
		{
			lcd_cmd(0xC4);
			lcd_string("OFF ");
		}
		if(device4==1)
		{
			lcd_cmd(0xCd);
			lcd_string("ON ");
		}
		if(device4==0)
		{
			lcd_cmd(0xCd);
			lcd_string("OFF");
		}
}
```

4.3HARDWARE TESTING

4.3.1 CONTINUITY TEST:

In electronics, a continuity test is the checking of an electric circuit to see if current flows (that it is in fact a complete circuit). A continuity test is performed by placing a small voltage (wired in series with an LED or noise-producing component such as a piezoelectric speaker) across the chosen path. If electron flow is inhibited by broken conductors, damaged components, or excessive resistance, the circuit is "open".

Devices that can be used to perform continuity tests include multi meters which measure current and specialized continuity testers which are cheaper, more basic devices, generally with a simple light bulb that lights up when current flows.

An important application is the continuity test of a bundle of wires so as to find the two ends belonging to a particular one of these wires; there will be a negligible resistance between the "right" ends, and only between the "right" ends.

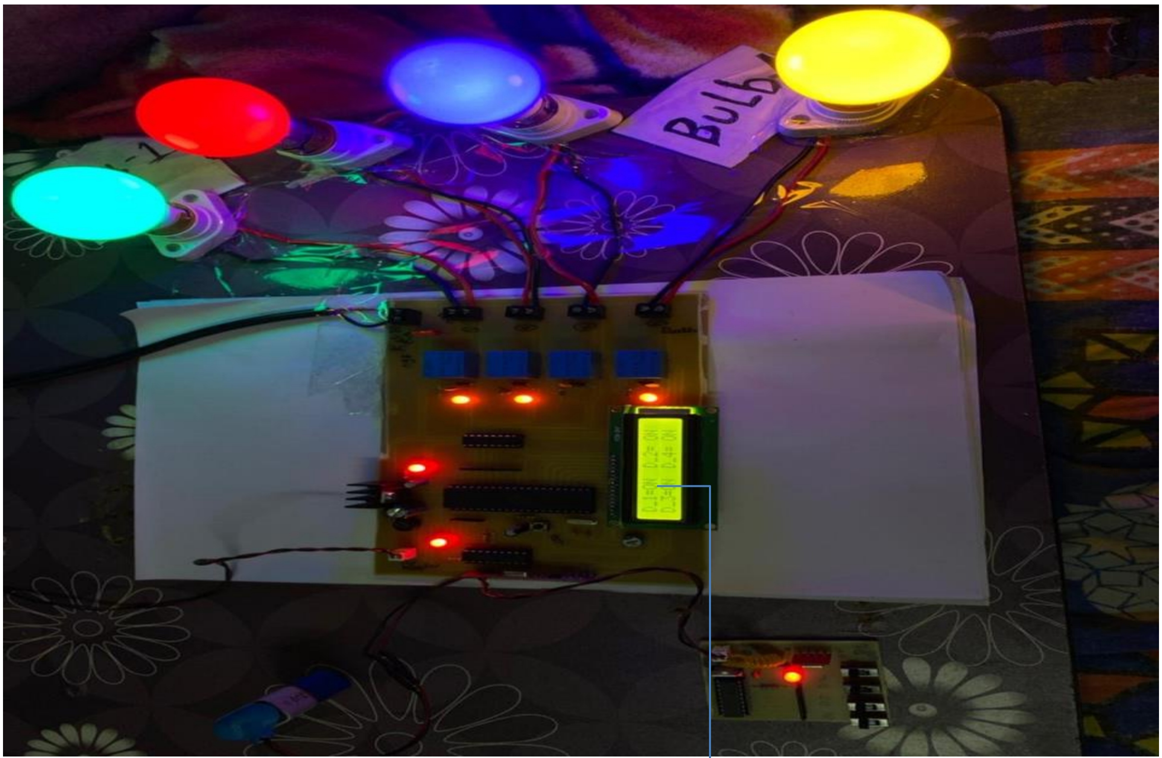
This test is the performed just after the hardware soldering and configuration has been completed. This test aims at finding any electrical open paths in the circuit after the soldering. Many a times, the electrical continuity in the circuit is lost due to improper soldering, wrong and rough handling of the PCB, improper usage of the soldering iron, component failures and presence of bugs in the circuit diagram. We use a multi meter to perform this test. We keep the multi meter in buzzer mode and connect the ground terminal of the multi meter to the ground. We connect both the terminals across the path that needs to be checked. If there is continuation then you will hear the beep sound.

4.3.2 POWER ON TEST:

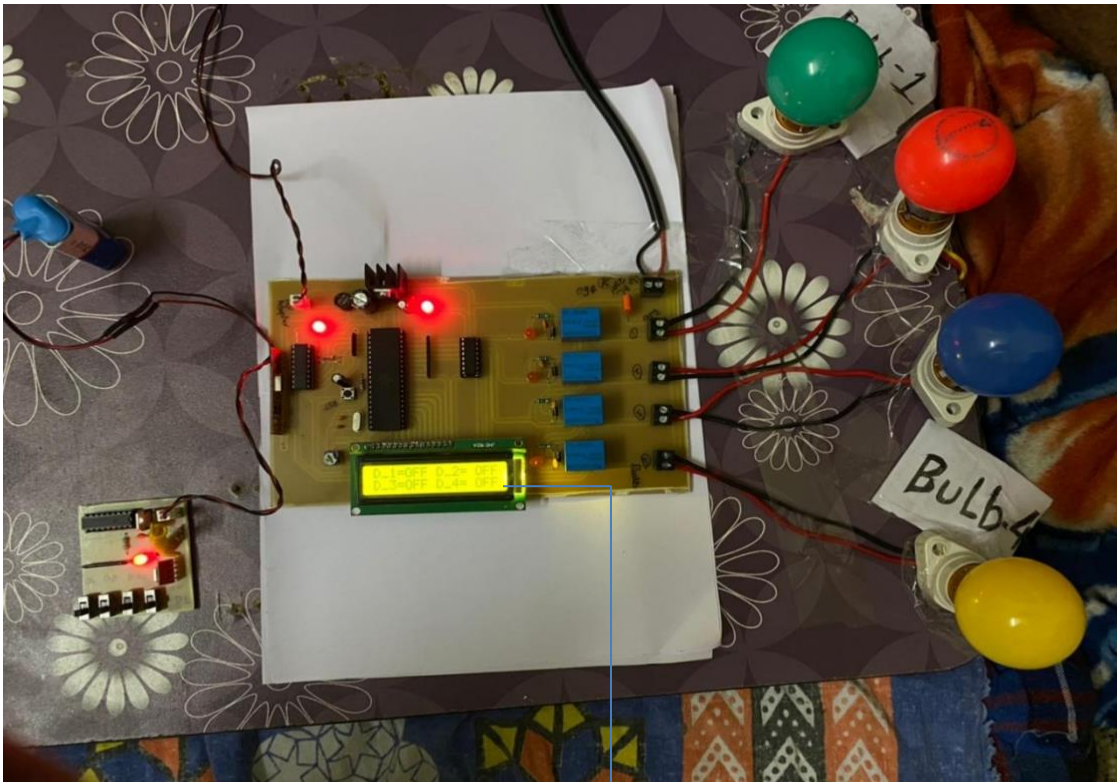
This test is performed to check whether the voltage at different terminals is according to the requirement or not. We take a multi meter and put it in voltage mode. Remember that this test is performed without microcontroller. Firstly, we check the output of the transformer, whether we get the required 12 v AC voltage.

Then we apply this voltage to the power supply circuit. Note that we do this test without microcontroller because if there is any excessive voltage, this may lead to damaging the controller. We check for the input to the voltage regulator i.e., are we getting an input of 12v and an output of 5v. This 5v output is given to the microcontrollers' 40th pin. Hence we check for the voltage level at 40th pin. Similarly, we check for the other terminals for the required voltage. In this way we can assure that the voltage at all the terminals is as per the requirement.

4.4 OUTCOMES



All Loads in ON Position



All Loads in OFF Position

CHAPTER 5

APPLICATION AND DEMANDS

5.1 APPLICATIONS

- It can be use in Home automation system
- We can use transmitter as universal controller
- Access control systems
- Remote control for various types of household appliances
- Many other applications fields related to RF wireless controlling

5.2 ADVANTAGES

- We can turn on/off the receiver with transmitter from any place.
- The RF wireless signal can pass through walls, floors and doors.
- One/several transmitters can control one/several receivers
- If we use two or more receivers in same place, you can set them with different codes.
- Convenience and simplicity

REFERNCES:

- **https://www.electronicsforu.com/radio-frequency**
- **D. L. Wu, Wing W. Y. NG, D. S. Yeung, and H. L. Ding, "A brief survey on current RFID applications," in *Proc. International Conference on Machine Learning and Cybernetics*, Baoding, July (12-15, 2009, pp. 2330-2334).**
- **B. Yan and D. Y. Lee, "Design of spot ticket management system based on RFID," in *Proc. International Conference on Networks Security, Wireless Communications and Trusted Computing*, 2009, pp.496-499**
- **G. Ostojic, S. Stankowski, and M. Lazarevic, "Implementation of RFID technology in parking lot access control system," in *Proc. Annual RFID Eurasia Conference*, 2007, pp. 1-5**
- **N. Ahmad, S. Butler, and U. Ramachandran, "Guardian Angel: An RFID based indoor guidance and monitoring system," 2010, pp. 546-551 RFID based indoor guidance and monitoring system," 2010, pp.546-551**
- **K. S. Huang and S. M. Tang, "RFID applications strategy and deployment in bike renting system," in *Proc. ICACT* 2008, pp.660-663**
- **www.alldatasheets.com**