Gujarat Technological University

2180905: Project I

A UDP Project Report

On

SMARTMETER WITH GSM MODULE

Prepared by:

Amey Nashikkar-130410109002
Shrey Bhatt-130410109007
Mrinal Hinduja-130410109052
Hasnain Thakrawala-130410109118

GROUP ID – 86472

Internal Guide:

Mr. Rakesh C. Gajjar
ANNUAL ACADEMIC YEAR: 2016-17

Electrical Engineering Department
Sardar Vallabhbhai Patel Institute of Technology
Gujarat Technological University, Vasad.

CERTIFICATE

This is to certify that Amey Nashikkar (130410109002), Shrey Bhatt (130410109007), Mrinal Hinduja (130410109052) and Hasnain Thakrawala (13041010118) of final year Electrical Engineering have satisfactorily completed their complete project work entitled "SMARTMETER WITH GSM MODULE" for the subject 2170001 -Project-I in the academic year 2016-17 for the complete fulfillment award of the Bachelor of Engineering in Electrical Engineering at Gujarat Technological University.

Date:

Project Guide (Internal)
Mr. Rakesh C. Gajjar

Head of Electrical Department
Dr. C.D.Kotwal

Principal of Institute
Dr. Jayesh Deshkar

UNDERTAKING ABOUT ORGINIALITY OF WORK

AMEY NASHIKKAR(130410109002), SHREY BHATT(130410109007), MRINAL HINDUJA(130410109052), HASNAIN THAKRAWALA(130410109118) give undertaking that the major project entitled “Smart Energy meter Using Gsm module” submitted for partial fulfillment of the requirement for the degree of bachelor of engineering in Electrical Engineering of SVIT,VASAD.

We have checked the write up of our udp project report using anti-plagiarism data vase and it is in the allowable limit. In case of any complaints pertaining to plagiarism we certify that we shall be solely responsible for the same and understand that as per norms of the university can even revoke BE degree conferred upon the students submitting this udp project report in case it is found to be plagiarized.

NAME	ENROLLMENT	SIGNATURE
AMEY NASHIKKAR	130410109002	
SHREY BHATT	130410109007	
MRINAL HINDUJA	130410109052	
HASNAIN THAKRAWALA	130410109118	

PLACE :- _________ DATE :-___________

Mr. Rakesh C. Gajjar

(Internal Guide, SVIT – VASAD)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would have been incomplete without the mentioning of the people whose constant guidance and encouragement made it possible. We take pleasure in presenting before you, our project, which is result of studied blend of both research and knowledge.

We express our deepest and earnest gratitude to our internal guide, Assistant Professor Mr Rakesh C. Gajjar, Electrical Engineering department, our project guide, for his constant and unconditional support, encouragement and guidance. We are grateful for his cooperation and his valuable suggestions.

Finally, we express our gratitude and gladly congratulate to all other members who are involved either directly or indirectly for the completion of this project.

ABSTRACT

The main objective of the project is to develop an energy meter to know the exact billing that each load is going to consumed with wireless technology.

For this research work we had taken a energy meter and is connected to an Opto-coupler, which counts the led that glows (depending upon the requirement we can increase the led glowing on/off time pulses), so its gives an interrupt after every time set by us is given to a programmable micro controller of 8051 family, The microcontroller takes the reading from the energy meter via an Opto-isolator and displays the reading on the LCD duly interface to a microcontroller. The reading of the energy meter is also sent to the cell phone of the user by a message through GSM modem being fed from the microcontroller via level shifter IC and RS232 link. The desired number is auto saved on the microcontroller over a missed call.

The power supply consists of a step down transformer 230/12V, which steps down the voltage to 12V AC. This is converted to DC using a Bridge rectifier and it is then regulated to +5V using a voltage regulator 7805 which is required for the operation of the microcontroller and other components.

INDEX PAGE NO.

REPORT

1. GENERAL QUESTIONS

1. What is smart energy meter?

A smart meter is a new kind of gas and electricity meter that can digitally send meter readings to your energy supplier. This can ensure more accurate energy bills. Smart meters also come with monitors, so you can better understand your energy usage.

Every home in Britain will have a smart meter installed by 2020.

2. What is a smart meter and how does it work?

Smart meters are a next generation meter for both gas and electricity. They are a replacement for your existing meters, which still use technology created decades ago.

Smart meters use a secure national communication network (called the DCC) to automatically and wirelessly send your actual energy usage to your supplier. This means households will no longer rely on estimated energy bills, have to provide their own regular readings, or have meter readers come into their homes to read the meter.

Smart meters will also come with an in-home display.

This display gives the household real-time usage info, including kWh use and cost.

3. What are the benefits of having a smart meter?

There are several benefits to smart meters:

- **More accurate bills** Smart meters mean the end of estimated bills, the end of having to remember to provide meter readings and/or have a stranger come into your home to read your meter
- **Better understanding of your usage** With the smart meter display, you can see the direct impact your habits and lifestyle have on your bill. This is particularly useful to prepayment meter customers, who can better track how their usage impacts their available credit. By making your energy usage easier to understand, you can make smarter decisions to save energy and money, including feeling more confident switching energy supplier.
- **Faster and easier energy switching** Because your usage data is so easily accessible, the aim is to make energy switching as quick as just a half hour.

- **Bringing Britain's energy system into the 21st century** The future is smart, and smart meters are part of the effort to create a smart grid, which is part of providing low-carbon, efficient and reliable energy to Britain's households.
- **Innovative energy tariffs** Using the data collected on when and how households are using energy, suppliers can create more competitive time-of-use tariffs with cheaper prices for off-peak use.

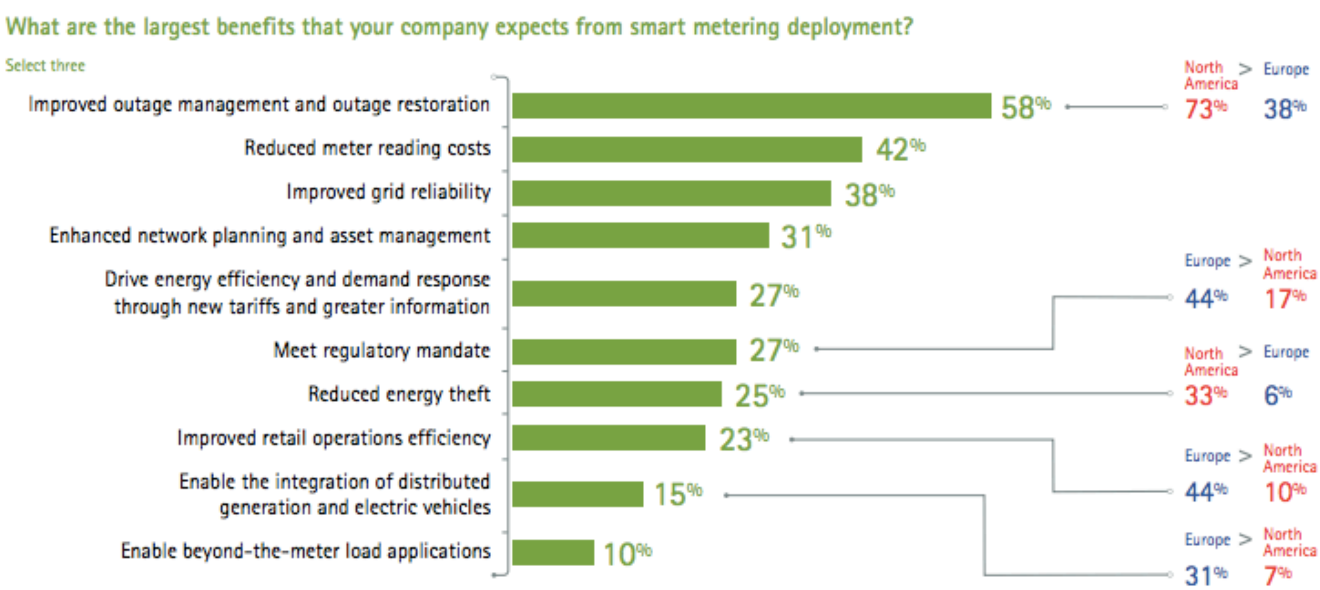
4. What are the current issues with smart meters?

- **Smart meters may currently lose smart functionality** When switching suppliers, meters may have to turn to "dumb" mode. Early adopters of smart meters may have a first generation meter (SMETS1) that is not compatible with all suppliers. In this case, you would have to revert to giving meter readings. This issue is set to be resolved by 2018.
- **Some smart meters aren't currently compatible with solar or microgeneration** You may find that your supplier cannot offer you smart meters just yet as they are not able to work with solar or microgeneration.
- **The location of your meter could be inaccessible** If your meter is located in a place where signal may be an issue (e.g. in the basement) your supplier's current generation of meter may be unable to achieve an appropriate signal to send information remotely to your supplier — in this case you won't presently be offered one.

2. SMART METER SCENARIO, WORLDWIDE

Smart metering deployment represents a common first step into smart grid solutions at scale for many utilities. The 10 largest national deployments worldwide are expected to add 500 million new smart meters by 2020,1 approximately tripling the 2012 global installed base, and the locus of growth shifting from North America to Europe, then Latin America and Asia. Despite the ongoing rollouts, many utilities are still unclear about the optimal route to extracting value from these large investments. Whether utilities are at the stage of planning, preparation or actual deployment, the blanket term "smart" masks a more complex reality. Smart metering means different things to different utilities, given the variety of prevailing industry structures, legal frameworks, regulatory mandates, availability of technology, network infrastructure stability and the operational environments. There is a wide array of possible approaches to deploying smart technologies and benefit areas on which to focus most aggressively. Against this backdrop, Accenture recommends focusing on five areas to help support strong, ongoing benefits realization from smart metering across the full breadth of the business: 1. Putting the consumer and the community at the heart of the design 2. Managing the complexities of deployment 3. Focusing on the people and process change 4. Future proofing the technology 5. Releasing further value from analytics

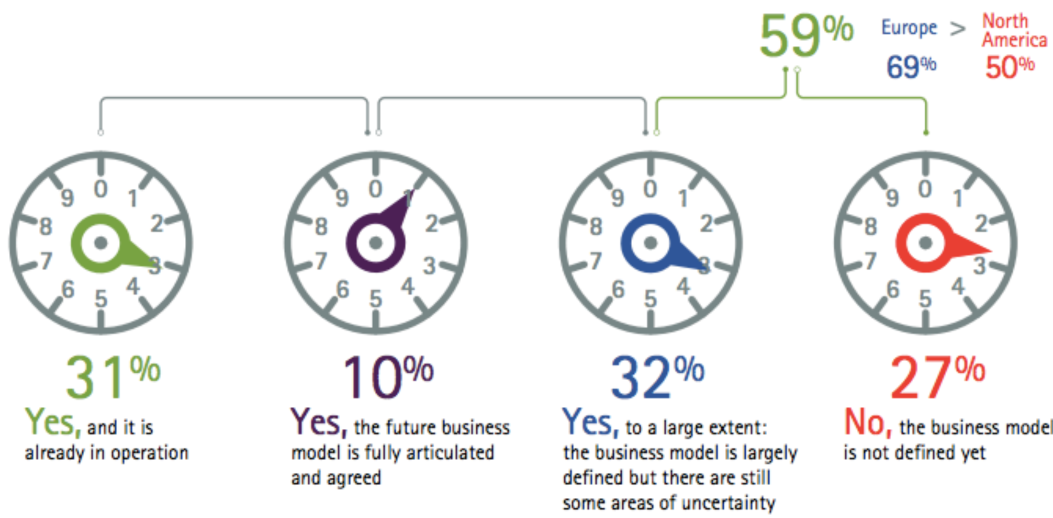
Figure 1. Benefits expected from smart metering deployment.



Base: All respondents, smart metering section.
Source: Accenture's Digitally Enabled Grid program, 2013 executive survey.

Figure 2. Degree of business model clarity for smart metering deployments.

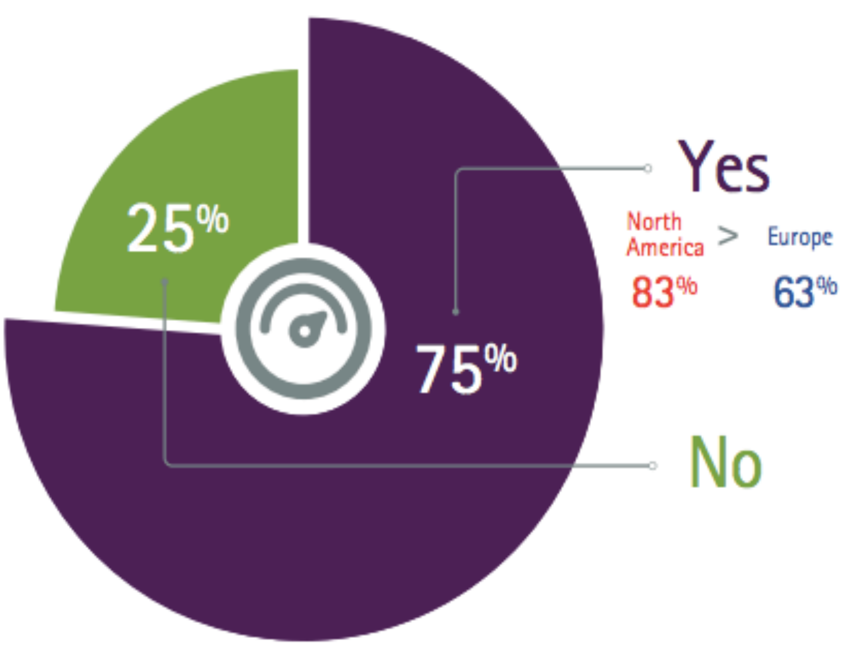
Is the business model in your company well defined for smart metering (e.g., understanding of the impact on the day-to-day operations, clear process ownership, visability on costs and revenue flows, clarity on data governance, etc.)?



Base: All respondents, smart metering section.
Source: Accenture's Digitally Enabled Grid program, 2013 executive survey.

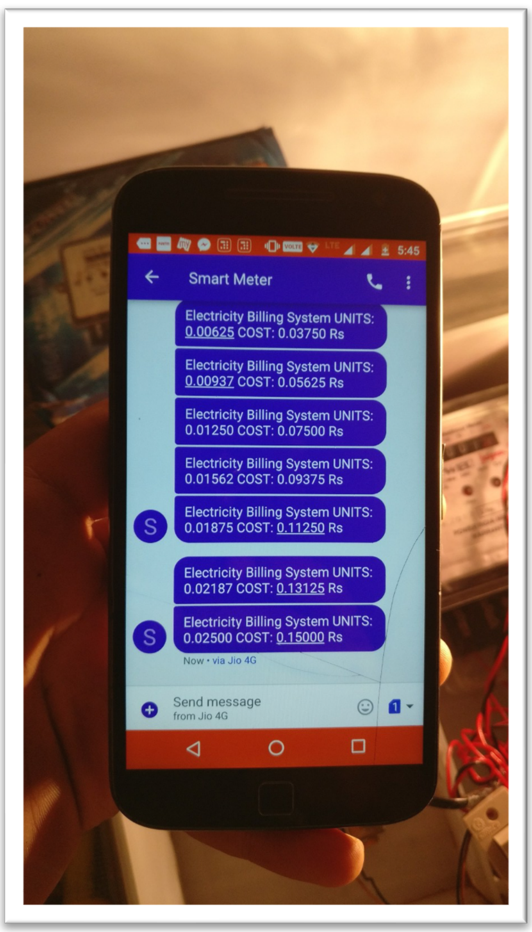
Figure 3. Degree of business case clarity for smart metering deployments.

Is the business case for smart metering articulated and agreed between key players (e.g., regulator, distribution company, retailer, municipality, consumer advocates, etc.)?



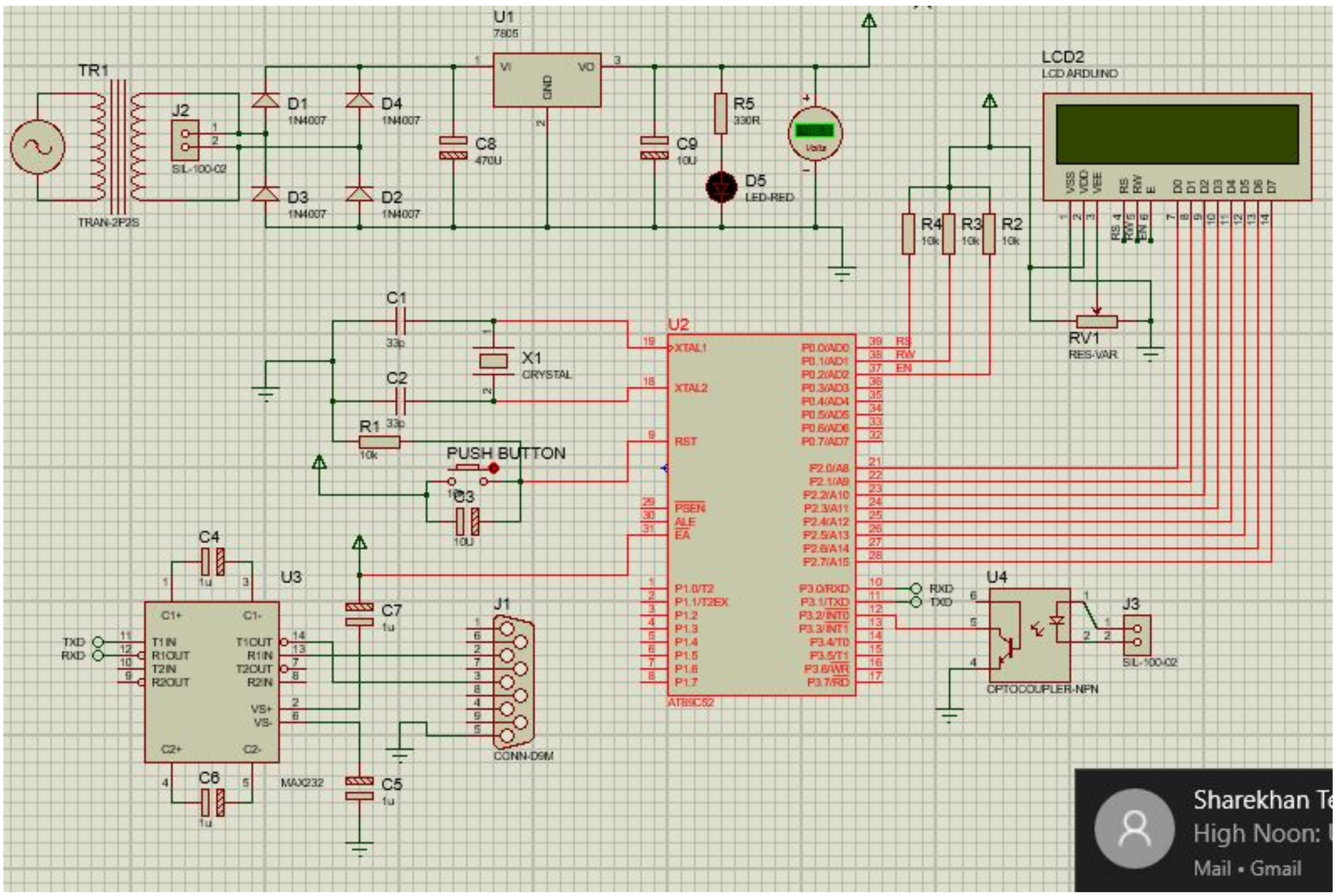
Base: All respondents, smart metering section.
Source: Accenture's Digitally Enabled Grid program, 2013 executive survey.

3.PROJECT

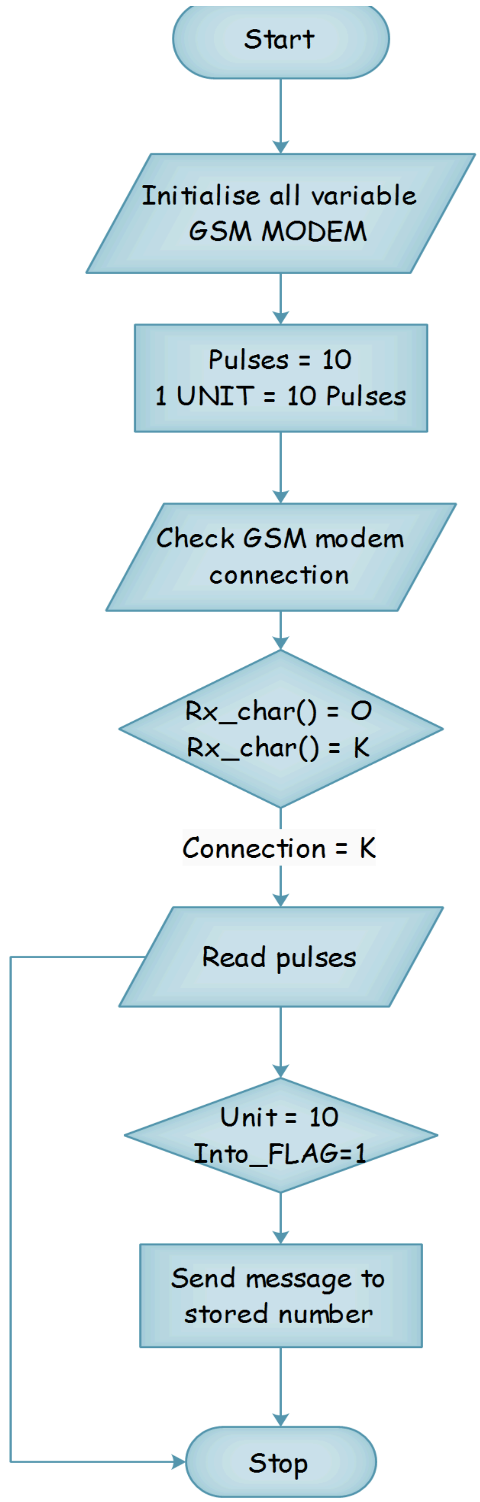

Smart Meter
Electricity Billing System UNITS: 0.00625 COST: 0.03750 Rs
Electricity Billing System UNITS: 0.00937 COST: 0.05625 Rs
Electricity Billing System UNITS: 0.01250 COST: 0.07500 Rs
Electricity Billing System UNITS: 0.01562 COST: 0.09375 Rs
Electricity Billing System UNITS: 0.01875 COST: 0.11250 Rs
Electricity Billing System UNITS: 0.02187 COST: 0.13125 Rs
Electricity Billing System UNITS: 0.02500 COST: 0.15000 Rs
Now • via Jio 4G
Send message
from Jio 4G


Units:0.01562
Rs:0.09375

SCHEMATIC DIAGRAM



3.1 Flow Chart



4. 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 $\overline{EA}$ 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.

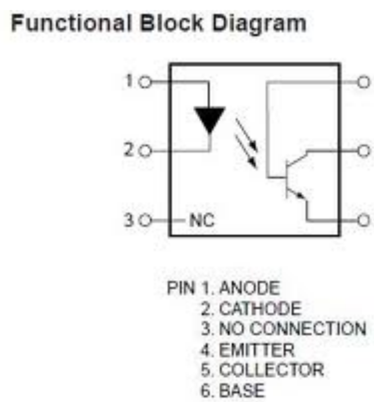
OPTOCOUPLER

Opto coupler is a 6 pin IC. It is a combination of 1 LED and a transistor. Pin 6 of transistor is not generally used and when light falls on the Base-Emitter junction then it switches and pin5 goes to zero.

If input of the diode is zero and other end of diode is GND then the output is one.

- When logic zero is given as input then the light doesn't fall on transistor so it doesn't conduct which gives logic zero as output.

- When logic 1 is given as input then light falls on transistor so that it conducts, that makes transistor switched ON and it forms short circuit this makes the output is logic zero as collector of transistor is connected to ground.



MAX232

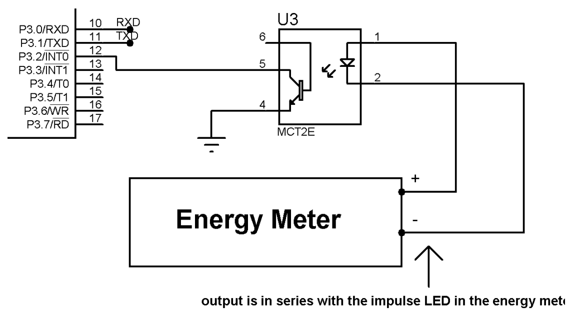
The MAX232 used in the project is an integrated circuit that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits like microcontroller. The MAX232 is a dual driver/receiver and typically converts the RX, TX, CTS and RTS signals

OPERATION

Connections:

The output of power supply which is 5v is connected to 40^{th} pin of microcontroller and Gnd is connected to 20^{th} pin of microcontroller. Pin 2.0 to pin 2.7 of port 2 of MC is connected to data pins i.e., D_0 to D_7 of LCD display. Pins 4, 5, 6 i.e., RS, Rw, EN. Of LCD are given to p 0.0 to p 0.2 of port of 0 of MC. Pin 3.0 and pin 3.1 of port 3 of MC are connected to pin 11 and pin 12 of Max232. Pin 3.3 of port 3 of MC is connected to pin 5 of Opto coupler. Pin 1 & 2 of Opto coupler are connected to energy meter. Pin 14 & 13 of Max232 are given to pins 2 & 3 of DB9 male connector. Pin 2 & pin 5 of DB9 Female Connector are given to GSM modem.

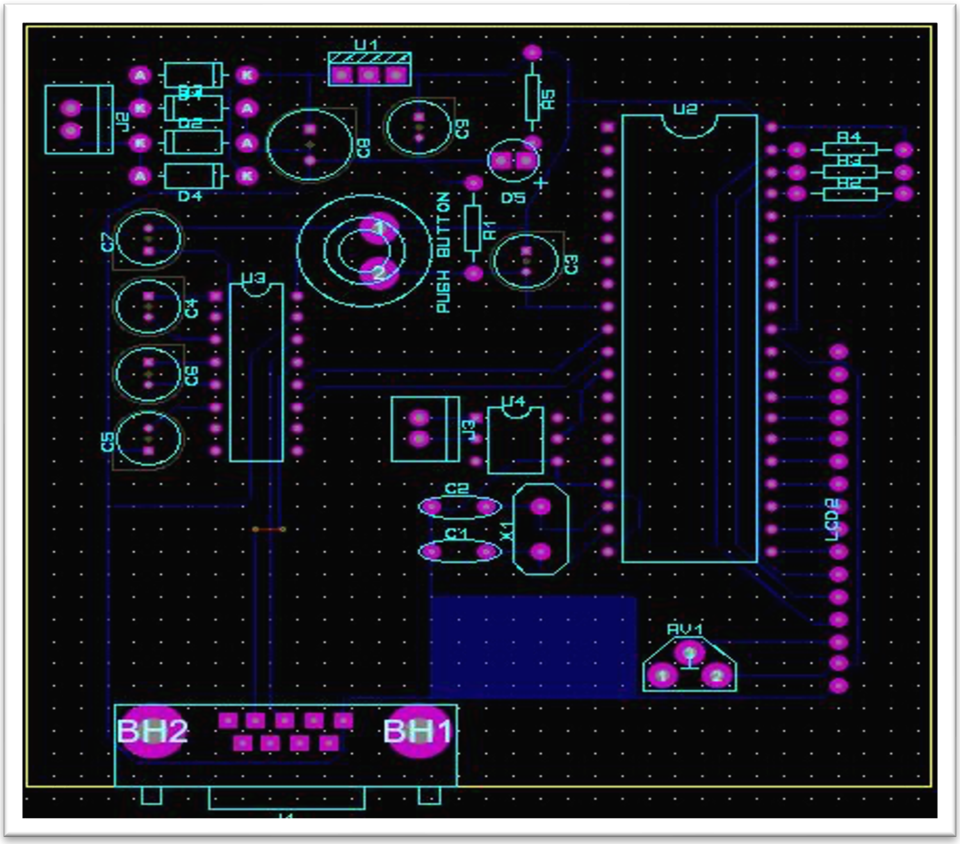
Working:

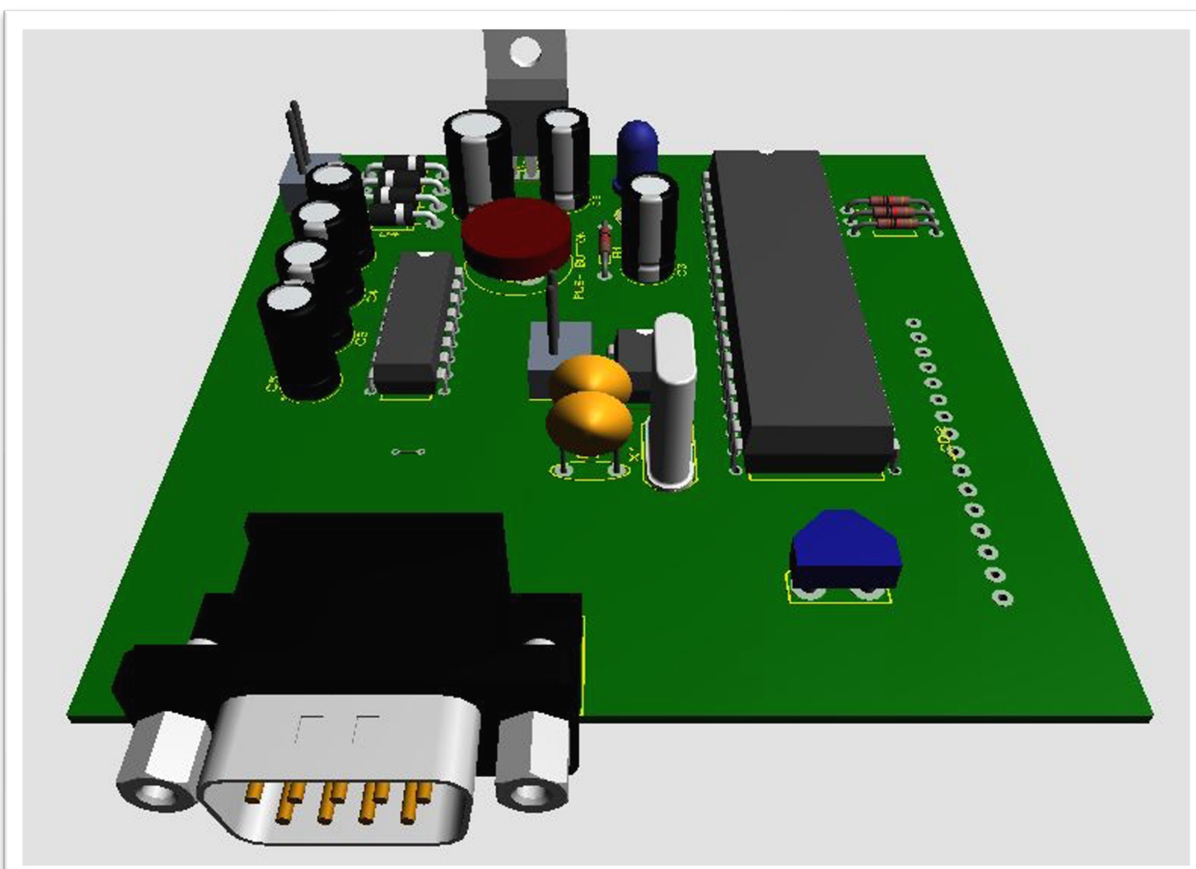


The project uses a commercial digital energy meter and derives positive pulses from the same by

Using an opto isolator led in series with the energy pulsing LED of the energy meter. The pulsing LED pulses 3200 times for 1 unit of electrical energy i.e., 1 kilo watt hour. Thus the opto led which is in series with the energy pulsating led blinks at the same rate developing logic zero and high pulses across the opto transistor which is interfaced to microcontroller. As it is not feasible to wait for consuming 1000 Wt Hr. the program assumes 10 pulses for unit, which is fed to the microcontroller pin no 12 to send 1 unit consumption through the GSM modem duly interfaced to the microcontroller through Max232, Therefore the 1 unit so sent shown have to be read as 1/3200 unit. Upon a missed a call to the project board GSM modem ,the caller's number gets stored in the microcontroller for further communication to that number only. This gives the unique flexibility for changing number by the user at will without going through the cumbersome process of writing the number while burning the program on to the microcontroller .Thus in that case only that number is used for communication and the user has no option to change that .

PCB LAYOUT DIAGRAM





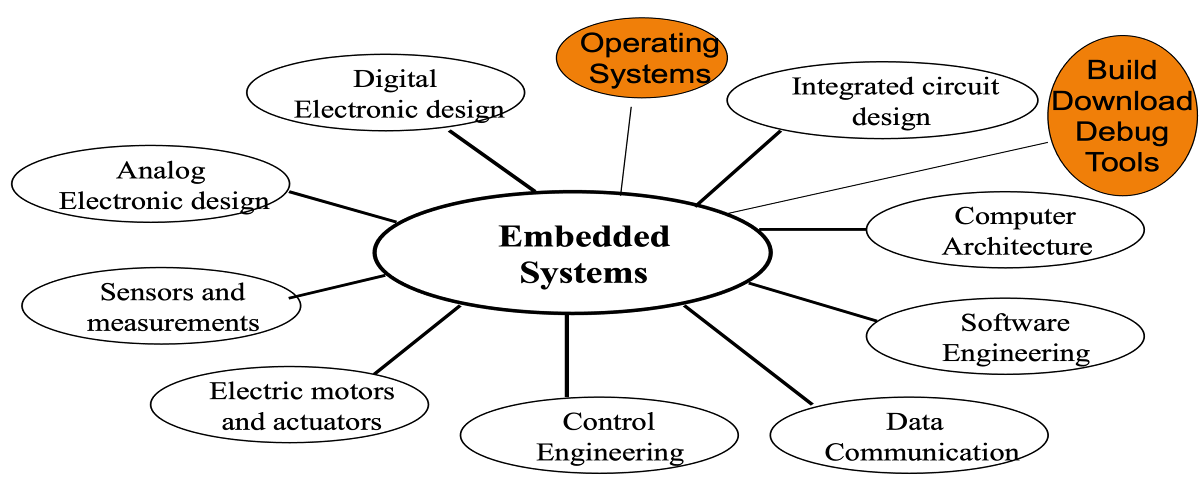
5.INTRODUCTION TO EMBEDDED SYSTEMS

What is Embedded system?

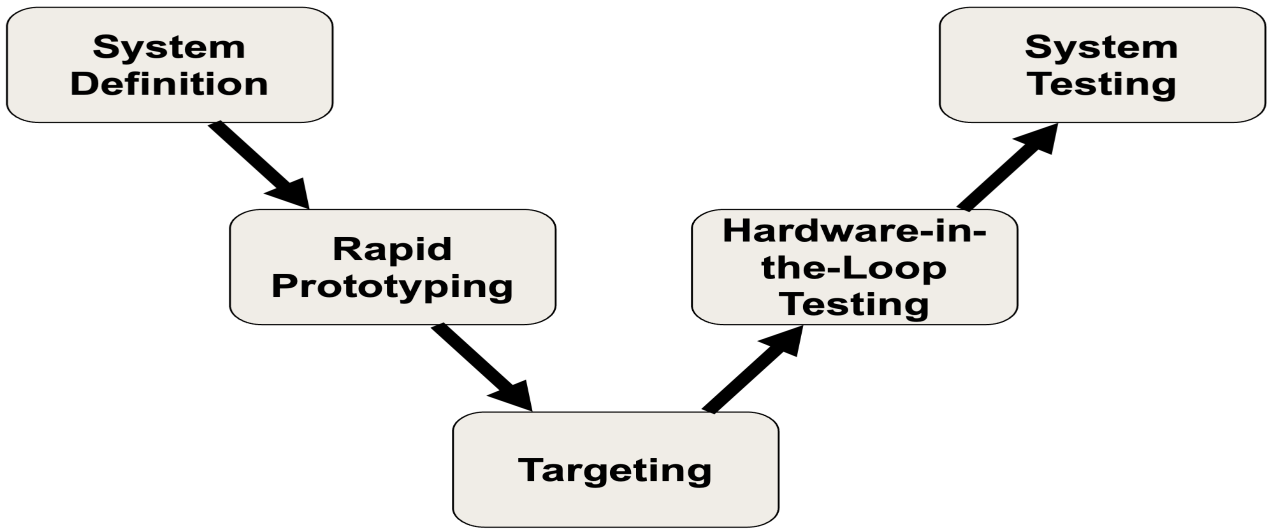
An Embedded System is a combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a specific function. An embedded system is a microcontroller-based, software driven, reliable, real-time control system, autonomous, or human or network interactive, 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. Examples Small controllers and devices in our everyday life like Washing Machine, Microwave Ovens, where they are embedded in.

EMBEDDED SYSTEM DESIGN CYCLE



Embedded system design calls on many disciplines



Figuren 3.1(b) “V Diagram”

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.

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.

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

RTS CLASSIFICATION

- Hard Real Time Systems
- Soft Real Time System

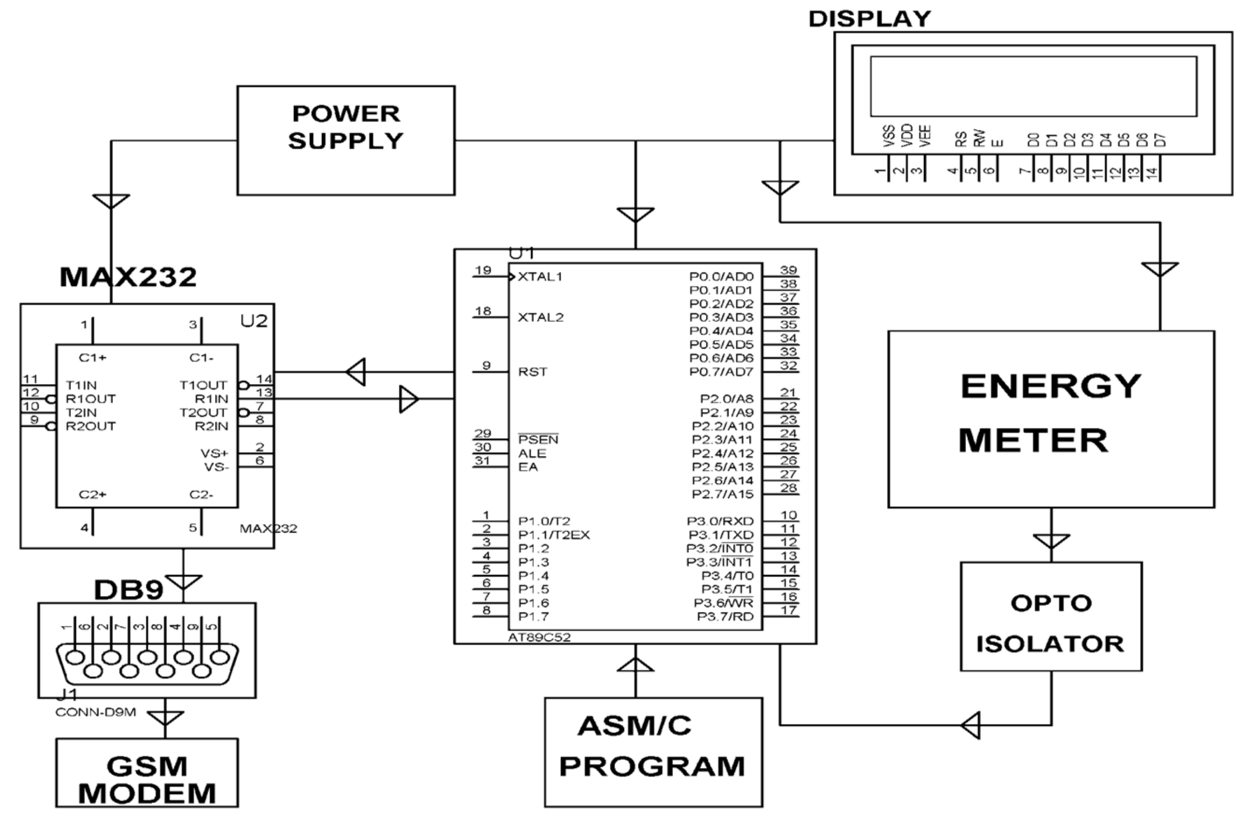
HARD REAL TIME SYSTEM

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

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.

6. BLOCK DIAGRAM



7. HARDWARE REQUIREMENTS

HARDWARE COMPONENTS:

1. TRANSFORMER (230 – 12 V AC)
2. VOLTAGE REGULATOR (LM 7805)
3. RECTIFIER
4. FILTER
5. MICROCONTROLLER (AT89S52/AT89C51)
6. GSM COMMUNICATION
7. GSM MODEM
8. LCD DISPLAY
9. MAX 232
10. DB9 CONNECTOR
11. ENERGY METER
12. OPTO COUPLER
13. 1N4007
14. RESISTOR
15. CAPACITOR

7.1 VOLTAGE REGULATOR 7805

Features

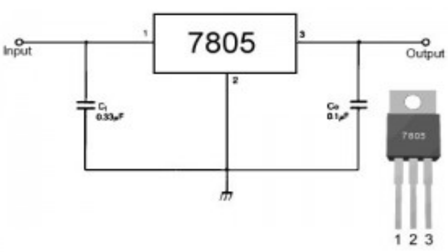
• Output Current up to 1A.

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

• Thermal Overload Protection.

• Short Circuit Protection.

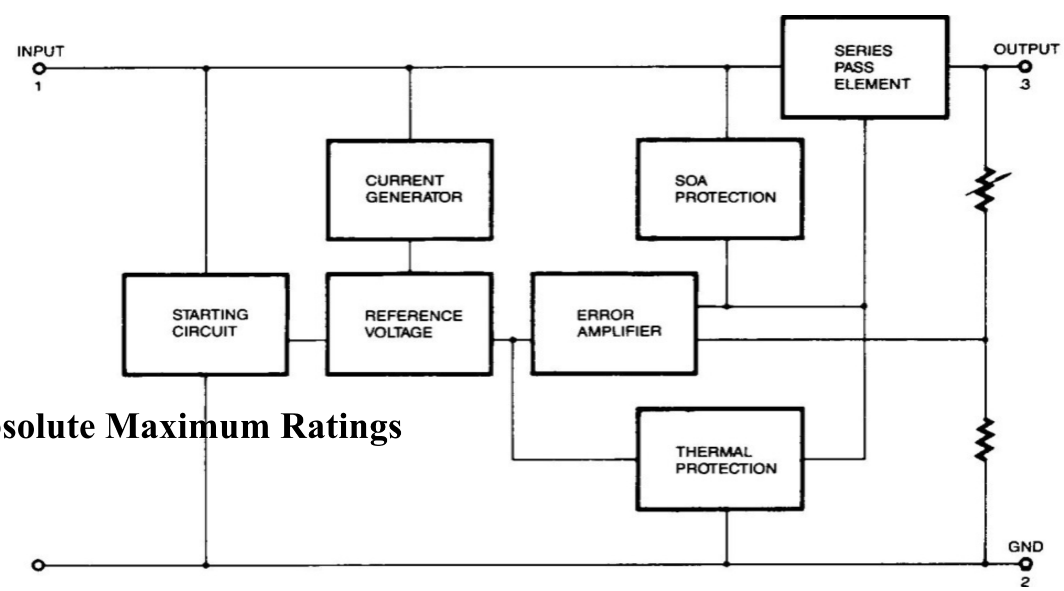
• Output Transistor Safe Operating Area Protection.



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.

Internal Block Diagram



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

TABLE 4.2(b): RATINGS OF THE VOLTAGE REGULATOR

7.2 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

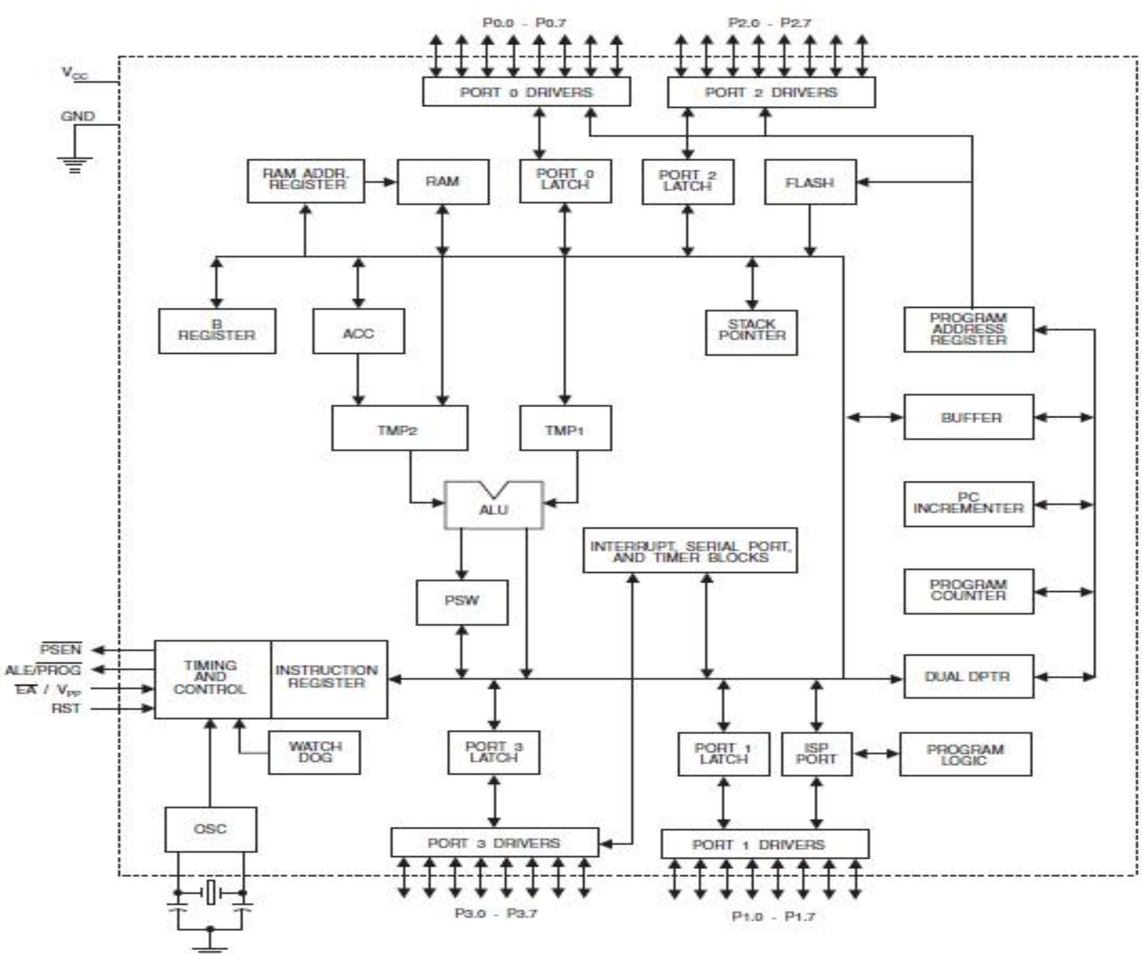
• Watchdog Timer

• Dual Data Pointer

• Power-off Flag

• Fast Programming Time

• Flexible ISP Programming (Byte and Page M.



Pin Configurations of AT89S52

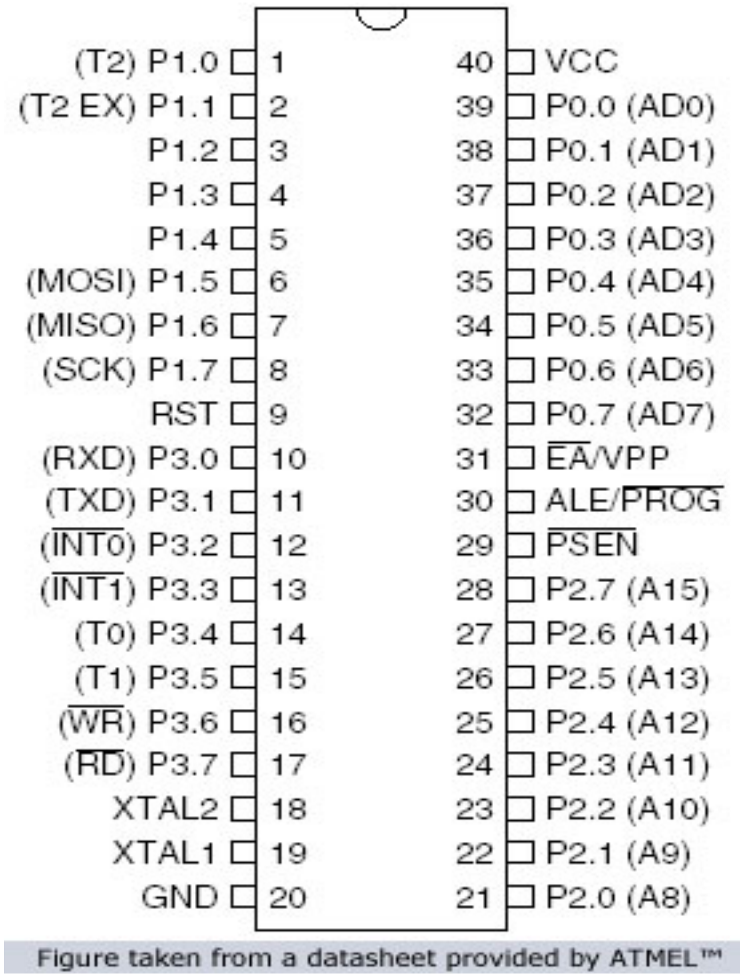


FIG 4.5(b): PIN DIAGRAM OF AT89S52

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 uses 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 uses 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.

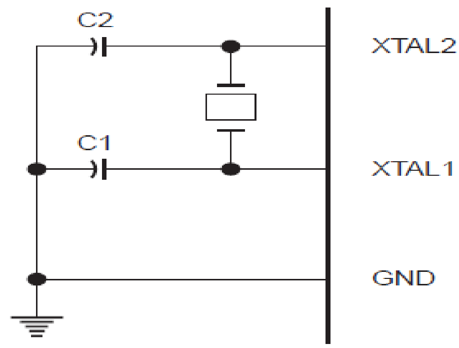


FIG 4.5(c): Oscillator Connections

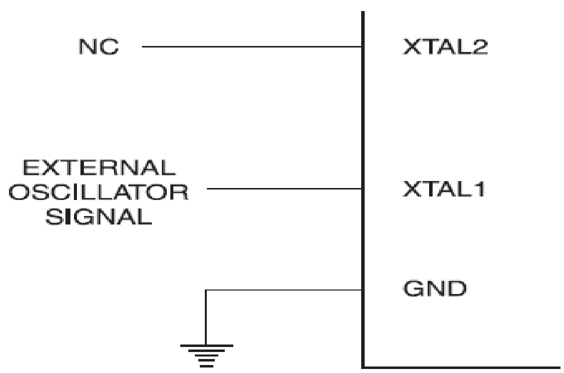


FIG 4.5(d): External Clock Drive Configuration

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 stable.

7.3 GSM COMMUNICATION

GSM for mobile system is increasingly popular and established throughout the world. The term GSM usually means the GSM standard and protocols in the frequency spectrum around 900MHz. There is also DCS1800 - GSM protocols but at different air frequencies around 1800 MHz - and in the United States, where spectrum for Personal Communication Services (PCS) was auctioned at around 1900MHz. As a result of this, the original and most widely-used GSM frequency implementation is known as GSM900, and DCS1800 is also known as GSM1800. Though the physical frequencies used are differed, the protocols and architecture remain the same. The following sections describe about the functional entities, the radio interface signaling protocol, the logical and physical channel structure and the TDMA structure based on GSM.

System architecture

The figure below shows the GSM system architecture, which consists of the switching system, the base station system and the user equipment. Functional entities are briefly explained as follows.

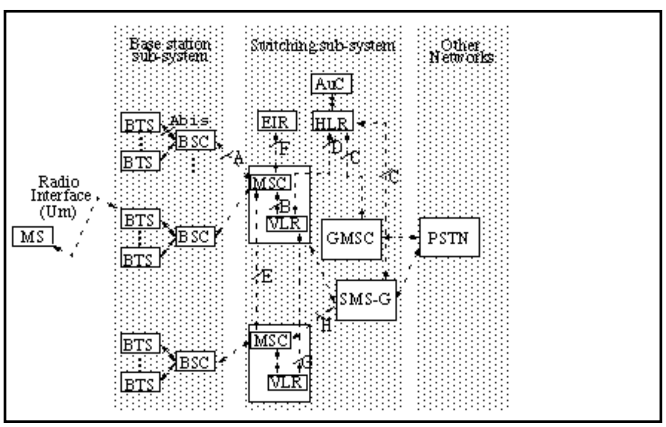


FIG:4.6(a) DETAILED ARCHITECTURE OF GSM

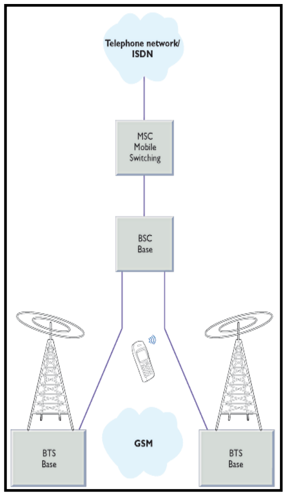


FIG:4.6(b) BASIC GSM NETWORK

DESCRIPTION OF MODULES IN GSM ARCHITECTURE

MS - Mobile Station:

The MS is the physical equipment used by a subscriber, most often a normal hand-held cellular telephone.

BTS -Base Transceiver Station:

The BTS comprises the radio transmission and reception devices, and also manages the signal processing related to the air interface.

TRAU -The Transcoder Rate Adaptor Unit:

The TRAU functionally belongs to the BTS. The TRAU enables the use of lower rates (32, 16 or 8 kbps) over the A-bits interface instead of the 64 kbps ISDN rate for which the MSC is designed. The TRAU can be located at the BTS, the BSC, or (immediately in front of) the MSC.

BSC- Base Station Controller:

The BSC manages the radio interface, mainly through the allocation, release and handover of radio channels.

BSS -Base Station System:

The BSS consists of a BSC and one or more BTS's.

MSC -Mobile Switching Centre:

The MSC is basically an ISDN-switch, coordinating and setting up calls to and from MS's. An Inter-Working Function (IWF) may be required to adapt.

VLR -Visitor Location Register:

The VLR contains all the subscriber data, both permanent and temporary, which are necessary to control a MS in the MSCs coverage area. The VLR is commonly realized as an integral part of the MSC, rather than a separate entity.

AuC -Authentication Centre:

The AuC database contains the subscriber authentication keys and the algorithm required to calculate the authentication parameters to be transferred to the HLR.

HLR- Home Location Register:

The HLR database is used to store permanent and semi-permanent subscriber data; as such, the HLR will always know in which location area the MS is (assuming the MS is in a coverage area), and this data is used to locate an MS in the event of a MS terminating call set-up.

EIR -Equipment Identity Register:

The EIR database contains information on the MS and its capabilities. The IMEI (International Mobile Subscriber Identity) is used to interrogate the EIR.

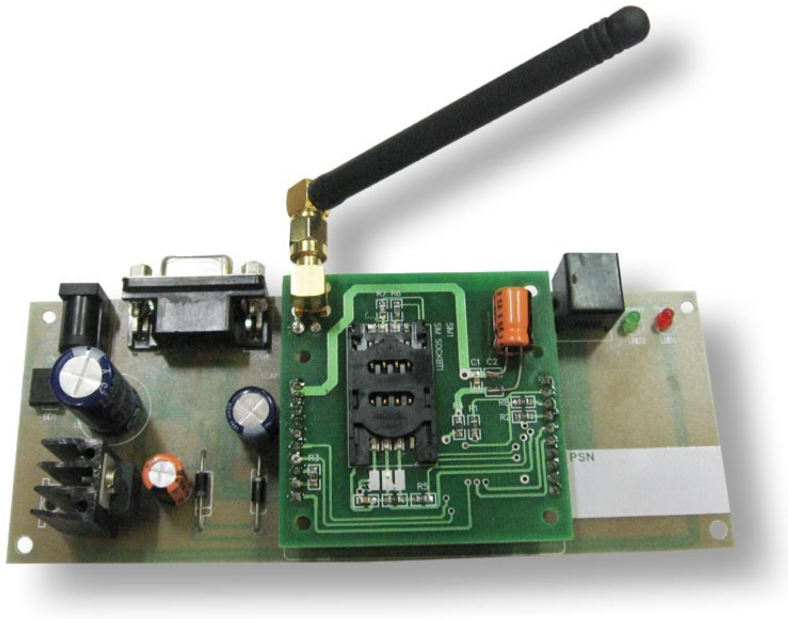
GMSC -Gateway Mobile Switching Centre:

The GMSC is the point to which a MS terminating call is initially routed, without any knowledge of the MS's location. The GMSC is thus in charge of obtaining the MSRN (Mobile Station Roaming Number) from the HLR based on the MSISDN (Mobile Station ISDN number, the "directory number" of a MS) and routing the call to the correct visited MSC. The "MSC" part of the term GMSC is misleading, since the gateway operation does not require any linking to a MSC.

SMSG: This is the term used to collectively describe the two Short Message Services Gateways described in the GSM recommendations. The SMS-GMSC (Short Message Service Gateway Mobile Switching Centre) is for mobile terminating short messages and SMS-IWMSC (Short Message Service Inter-Working Mobile Switching Centre) for mobile originating short messages. The SMS-GMSC role is similar to that of the GMSC, whereas the SMS-IWMSC provides a fixed access point to the Short Message Service Centre.

7.4 GSM MODEM

A **GSM modem** is a specialized type of modem which accepts a SIM card, and operates over a subscription to a mobile operator, just like a mobile phone. From the mobile operator perspective, a GSM modem looks just like a mobile phone.



When a GSM modem is connected to a computer, this allows the computer to use the GSM modem to communicate over the mobile network. While these GSM modems are most frequently used to provide mobile internet connectivity, many of them can also be used for sending and receiving SMS and MMS messages.

A GSM modem can be a dedicated modem device with a serial, USB or Bluetooth connection, or it can be a mobile phone that provides GSM modem capabilities.

For the purpose of this document, the term GSM modem is used as a generic term to refer to any modem that supports one or more of the protocols in the GSM evolutionary family, including the 2.5G technologies GPRS and EDGE, as well as the 3G technologies WCDMA, UMTS, HSDPA and HSUPA.

A GSM modem exposes an interface that allows applications such as SMS to send and receive messages over the modem interface. The mobile operator charges for this message sending and receiving as if it was performed directly on a mobile phone. To perform these tasks, a GSM modem must support an "extended AT command set" for sending/receiving SMS messages, as defined in the ETSI GSM 07.05 and 3GPP TS 27.005 specifications.

GSM modems can be a quick and efficient way to get started with SMS, because a special subscription to an SMS service provider is not required. In most parts of the world, GSM modems are a cost effective solution for receiving SMS messages, because the sender is paying for the message delivery. A GSM modem can be a dedicated modem device with a serial, USB or Bluetooth connection, such as the Falcom Samba 75. (Other manufacturers of dedicated GSM modem devices include Wavecom, Multitech and iTegno. We've also reviewed a number of modems on our technical support blog.) To begin, insert a GSM SIM card into the modem and connect it to an available USB port on your computer.

A GSM modem could also be a standard GSM mobile phone with the appropriate cable and software driver to connect to a serial port or USB port on your computer. Any phone that supports the "extended AT command set" for sending/receiving SMS messages, as defined in ETSI GSM 07.05 and/or 3GPP TS 27.005, can be supported by the Now SMS & MMS Gateway. Note that not all mobile phones support this modem interface.

Due to some compatibility issues that can exist with mobile phones, using a dedicated GSM modem is usually preferable to a GSM mobile phone. This is more of an issue with MMS messaging, where if you wish to be able to receive inbound MMS messages with the gateway, the modem interface on most GSM phones will only allow you to send MMS messages. This is because the mobile phone automatically processes received MMS message notifications without forwarding them via the modem interface.

It should also be noted that not all phones support the modem interface for sending and receiving SMS messages. In particular, most smart phones, including Blackberries, iPhone, and Windows Mobile devices, do not support this GSM modem interface for sending and receiving SMS messages at all at all. Additionally, Nokia phones that use the S60 (Series 60) interface, which is Symbian based, only support sending SMS messages via the modem interface, and do not support receiving SMS via the modem interface.

7.5 LCD DISPLAY

Description:

This is the example for the Parallel Port. 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.

Pros:

- Very compact and light
- Low power consumption

- No geometric distortion
- Little or no flicker depending on backlight technology
- Not affected by screen burn-in
- No high voltage or other hazards present during repair/service
- Can be made in almost any size or shape
- No theoretical resolution limit

LCD Background:

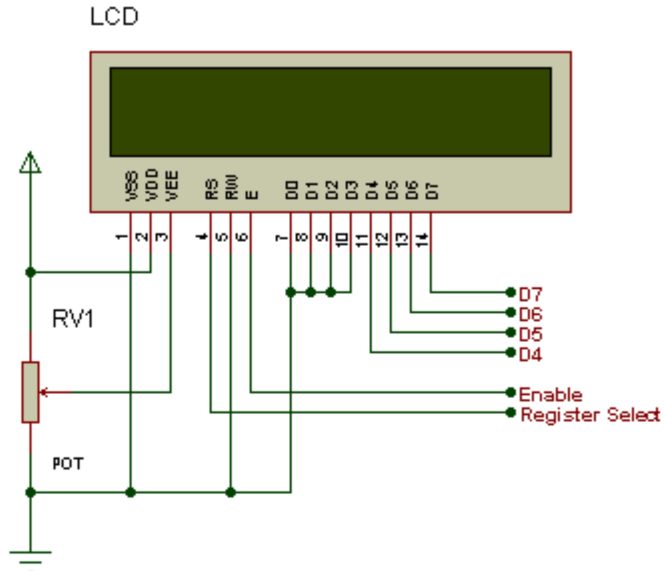
Frequently, an 8051 program must interact with the outside world using input and output devices that communicate directly with a human being. One of the most common devices attached to an 8051 is an LCD display. Some of the most common LCDs connected to the 8051 are 16x2 and 20x2 displays. This means 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 4.10: 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.

7.6 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.

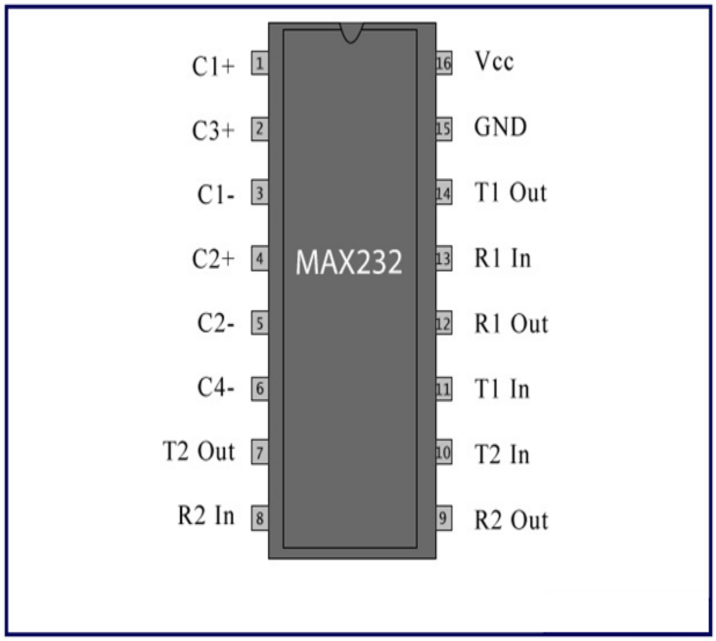
The later MAX232A is backwards compatible with the original MAX232 but may operate at higher baud rates and can use smaller external capacitors (0.1 µF) in place of the 1.0 µF capacitors used with the original device. The newer MAX3232 is also backwards compatible, but operates at a broader voltage range,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

7.7 <u>DB9 CONNECTOR</u>

The DB9 (originally DE-9) connector is an analog 9-pin plug of the D-Sub miniature connector family (D-Sub or Sub-D). The DB9 connector is mainly used for serial connections, allowing for the asynchronous transmission of data as provided for by standard RS-232 (RS-232C).

Fig 4.10: DB9 CONNECTOR

Pins:

Pin number	Name	
1	CD - Carrier Detect	
2	RXD - Receive Data	
3	TXD - Transmit Data	
4	DTR - Data Terminal Ready	
5	GND - Signal Ground	
6	DSR - Data Set Ready	
7	RTS - Request To Send	
8	CTS - Clear To Send	
9	RI - Ring Indicator	
	Shield	

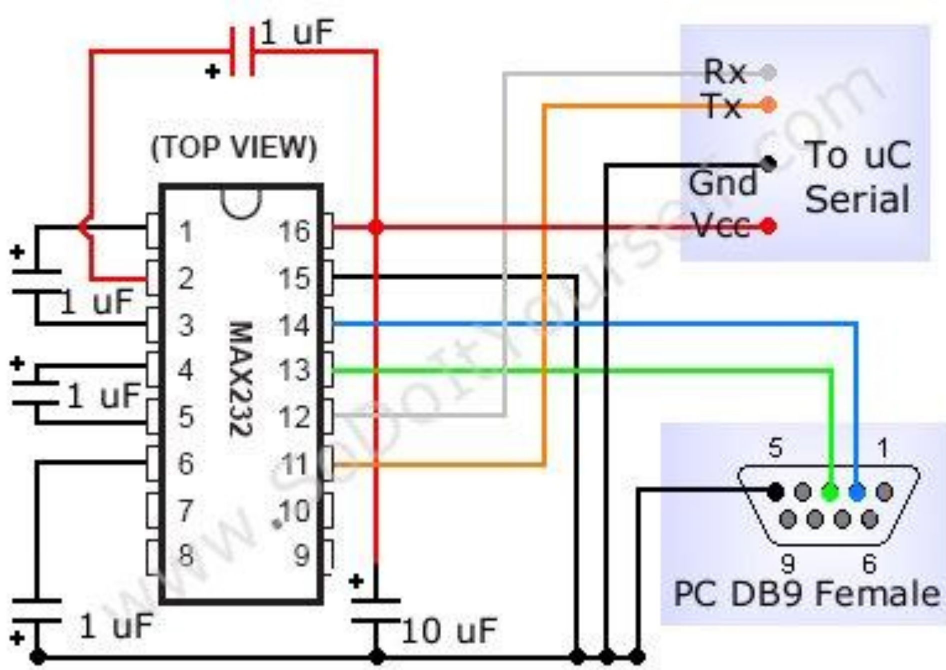
This is a common connector used in many computer, audio/video, and data applications. The official name is D-sub miniature, but many people call it "D-sub" or just "DB". The connector gets its name from its trapezoidal shape that resembles the letter "D". Most DB connectors have two rows of pins. Common types of D-sub connectors are DB9 and DB25, used on PCs for serial and parallel ports.

One special type of D-sub connectors is the High-Density DB style, which looks just like a regular DB connector, only with pins that are slightly smaller and placed closer together. This is typically referred to as an "HD" connector. HD connectors often have three rows of pins instead of two. The most common HD connector is the HD15, which is found on PC video cards and monitors. DB- and HD-connectors use thumbscrews to secure the connector in place.

Another type of D-sub is the MD, or Micro DB connector. This connector is slimmer than a standard D-sub, with pins even smaller than the ones used on HD connectors. The MD is also commonly called a "half-pitch" DB connector. These are often used in SCSI applications, and the most popular types are the MD50 and MD68 connections. MD-connectors can use latch clips or thumbscrews as anchoring mechanisms.

D-sub connectors are usually described by the total number of pins that they can hold. In some cases, a DB25 connector may only have 4 or 5 pins loaded into it; however, it is still called a "DB25" connector and not a

“DB4” or “DB5”. Another example is the HD15 connector used by monitors—most monitor cables only are loaded with 14 pins, but it is still called an HD15 connector.



7.8 ENERGY METER

An energy or electric meter is a device that measures the amount of electrical energy consumed by a residence, business, or an electrically-powered device.

Fig 4.2.10: Energy Meter

Electric meters are typically calibrated in billing units, the most common one being the kilowatt hour. Periodic readings of electric meters establish billing cycles and energy used during a cycle.

In settings when energy savings during certain periods are desired, meters may measure demand, the maximum use of power in some interval. In some areas, the electric rates are higher during certain times of day, to encourage reduction in use. Also, in some areas meters have relays to turn off nonessential equipment.

Unit of measurement

The most common unit of measurement on the electricity meter is the *kilowatt hour*, which is equal to the amount of energy used by a load of one kilowatt over a period of one hour, or 3,600,000 joules.

Demand is normally measured in watts, but averaged over a period, most often a quarter or half hour.

Reactive power is measured in "Volt-amperes reactive", (varh) in kilovar-hours. By convention, a "lagging" or inductive load, such as a motor, will have positive reactive power. A "leading", or capacitive load, will have negative reactive power.

Volt-amperes measures all power passed through a distribution network, including reactive and actual. This is equal to the product of root-mean-square volts and amperes.

Meters which measured the amount of charge (coulombs) used, known as ampere-hour meters, were used in the early days of electrification. These were dependent upon the supply voltage remaining constant for accurate measurement of energy usage, which was not a likely circumstance with most supplies.

Some meters measured only the length of time for which charge flowed, with no measurement of the magnitude of voltage or current is being made. These were only suited for constant-load applications.

7.9 OPTO COUPLER MCT2E

Opto-isolators, or Opto-couplers, are made up of a light emitting device, and a light sensitive device, all wrapped up in one package, but with no electrical connection between the two, just a beam of light. The light emitter is nearly always an LED. The light sensitive device may be a photodiode, phototransistor, or more esoteric devices such as thyristors, triacs e.t.c.

A lot of electronic equipment nowadays is using optocoupler in the circuit. An optocoupler or sometimes refer to as optoisolator allows two circuits to exchange signals yet remain electrically isolated. This is usually accomplished by using light to relay the signal. The standard optocoupler circuits design uses a LED shining on a phototransistor-usually it is a npn transistor and not PNP. The signal is applied to the LED, which then shines on the transistor in the IC.

The light is proportional to the signal, so the signal is thus transferred to the phototransistor. Optocouplers may also comes in few module such as the SCR, photodiodes, TRIAC of other semiconductor switch as an output, and incandescent lamps, neon bulbs or other light source.

The optocoupler usually found in switch mode power supply circuit in many electronic equipment. It is connected in between the primary and secondary section of power supplies. The optocoupler application or function in the circuit is to:

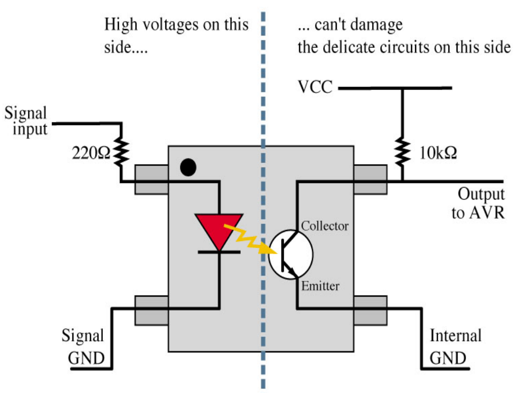
1. Monitor high voltage
2. Output voltage sampling for regulation

3. System control micro for power ON/OFF

4. Ground isolation

If the optocoupler IC breakdown, it will cause the equipment to have low power, blink, no power, erratic power and even power shut down once switch on the equipment. Many technicians and engineers do not know that they can actually test the optocoupler with their analog multimeter. Most of them thought that there is no way of **testing** an IC with an analog meter.

This is the principle used in Opto−Triacs and opto−SCRs, which are readily available in Integrated circuit (I.C.) form, and do not need very complex circuitry to make them work. Simply provide a small pulse at the right time to the Light Emitting Diode in the package. The light produced by the LED activates the light sensitive properties of the Triac or Thyristor gate and the power is switched on. The isolation between the low power and high power circuits in these optically connected devices is typically several thousand volts.



OPTO-ISOLATOR PARAMETERS

Collector-emitter voltage

This is the maximum voltage that can be present from the collector to the emitter of the receiving phototransistor (when it is turned off – no light) before it may break-down.

Cree page distance

This is physically how far a spark would have to travel around the outside of the package to get from one side to the other. If the package has contaminants on it, solder flux, or dampness, then a lower-resistance path can be created for noise signals to travel along.

Forward current

This is the current passing through the sending LED. Typically, an Opto-isolator will require about 5mA to turn the output transistor on.

Forward voltage

This is the voltage that is dropped across the LED when it is turned on. Most normal diodes drop about 0.7v, but with LEDs it is typically 1 – 2 volts.

Collector dark current

This is the current that can flow through the output phototransistor when it is turned off.

Collector-emitter saturation voltage

When the output transistor is fully turned on (saturated), this is the voltage there will be between the collector and emitter.

Isolation resistance

This is the resistance from a pin in the input side to a pin on the output side. It should be very high.

Response time

The rise and fall times are the times that the output voltage takes to get from zero to maximum. The rise time is very much dependant on the load resistor, since it is this that is pulling the output up. Therefore this value is always quoted with a fixed load resistance. Note however that the value, 100 Ohms, is much less than you are likely to use in practice. This is another of the manufacturer's attempts to make the product look better than it is!

Cutoff frequency

This is effectively the highest frequency of square wave that can be sent through the Opto-isolator. It is actually the frequency at which the output voltage is only swinging half the amplitude than at DC levels (-3Db = half). It is therefore linked with the rise and fall times.

Current Transfer Ratio (CTR)

This is the ratio of how much collector current in the output transistor that you get given a certain amount of forward current in the input side LED. It is affected by how close the LED and phototransistor are inside the device, how efficient they both are, and many other factors. In fact it is not a constant but varies wildly with LED forward current.

8. SOFTWARE REQUIREMENTS

8.1 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.

8.2 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.

8.3 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.

8.4 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.

8.5 BUILDING AN APPLICATION IN µVISION2

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

1. Select Project -(forexample,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.

8.6 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.

8.7 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.

8.8 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.

8.9 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.

8.10 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.

8.11 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 accesses. 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.

8.12 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.

8.13 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.

8.14 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.

8.15 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.

9. CODING

```
#include<at89x52.h>

#define LCDDATA P2

#define PULSE 10

sbit BUSY = P2^7;

sbit RS = P0^0;

sbit RW = P0^1;

sbit EN = P0^2;

unsigned char Buff[80];

unsigned char Rx_Flag = 0;

unsigned char INTO_Flag = 0;

unsigned int UNITS = 0;

unsigned int Rounds = 0;

void Tx_Char(unsigned char data);

void Tx_string(unsigned char*str);

unsigned char Rx_Char();

void ISR_INT0()interrupt0

{
```

```
	static int i = 0;
if(i == PULSES)
{
	INT0_Flag = 1;
	i = 0;
	UNITS++;
}
i++;
}
void LCD_Byte(unsigned char Byte,unsigned char mode)
{
	LCDDATA = Byte;
	RS = Mode;
	RW = 0;
	EN = 0;
	EN = 1;


}
void LCD_String(unsigned char*str)
```

```
{
    while(*str)
    {
        LCD_Byte(*str,1);
    }
}
void Init_LCD()
{
    LCD_Byte(0x38,0);
    LCD_Byte(0x06,0);
    LCD_Byte(0x0C,0);
    LCD_Byte(0x02,0);
}
void Init_SCOM()
{
    TMOD = 0x20;
    TH1 = 0xE8;
    TR1 = 1;
    SCON= 0x30;
```

```
}
void Tx_Char(unsigned char Data)
{
        SBUF = Data;
        while = (!TI);
        TI = 0;
}
unsigned char Rx_Char()
{
        while(!RI);
        RI = 0;
        return SBUF;
}
void Tx_String(unsigned char*str)
{
        while(*str)
        {
                Tx_Char(*str);
        }
```

```
}
void LCD_NUM(unsigned int num)
{
	unsigned char Buff[5],i=0;
	while(num)
	{
		Buff[i++] = (unsigned char)((num %10)+0x30);
		num = num/10;
}
if(!i)
	LCD_Byte(0x30,1);
while(i--)
{
	LCD_Byte(Buff[i],1);
	}
}
void Tx_NUM(unsigned int num)
{
	unsigned char Buff[5],i=0;
```

```
        while(num)
{
Buff[i++] = (unsigned char)((num%10)+0x30);
num = num/10;
         }
if(!i)
Tx_Char(0x30);
while(i--)
{
Tx_Char(Buff[i]);
}
}
void Delay(unsigned int Time)
{
unsigned int i=0;
for(i=0;i<Time;i++);
}
void Init_GSM()
```

GSM Modem

```
{
unsigned char exit = 0,i = 0;
while(!exit)
{
LCD_Byte(0x80,0);
LCD_String("ELEC BILLING SYSTEM");
Tx_String("AT");
Tx_Char(13);
while(Rx_char()! = 0x1C);
character
if(Rx_Char() == 'O')
if(Rx_Char()=='K')
{
exit = 1;
}
}
else
{
LCD_Byte(0xC0,0);
```

```
starting position

LCD_String("Check Modem Conc");

}

}

Tx_String("AT+CMGF=1");

Tx_Char(0x0d);

for(i=19;i>0;i++);

{

LCD_Byte(0xC8,0);

position

LCD_NUM(i);

Lcd

LCD_String(" ");

Delay(10000);


}

LCD_Byte(0xC0,0);

positon

LCD_String("Module Activated");
```

```
}
void Init_INT0()
{
        IT0 = 1;
        EX0 = 1;
}
void Send_Message(unsigned char*M_NO,unsigned int UNITS)
{
        Tx_String("AT+CMGS=");
for message send
        Tx_Char("");
        Tx_String(M_NO);
        Tx_Char("");
        Tx_String("ELECTRICITY BILLING SYSTEM\r");
        Tx_String("UNITS:");
        Tx_NUM(UNITS);
UNITS
        Tx_String("\rCOST:");
        Tx_NUM(UNITS*6);
```

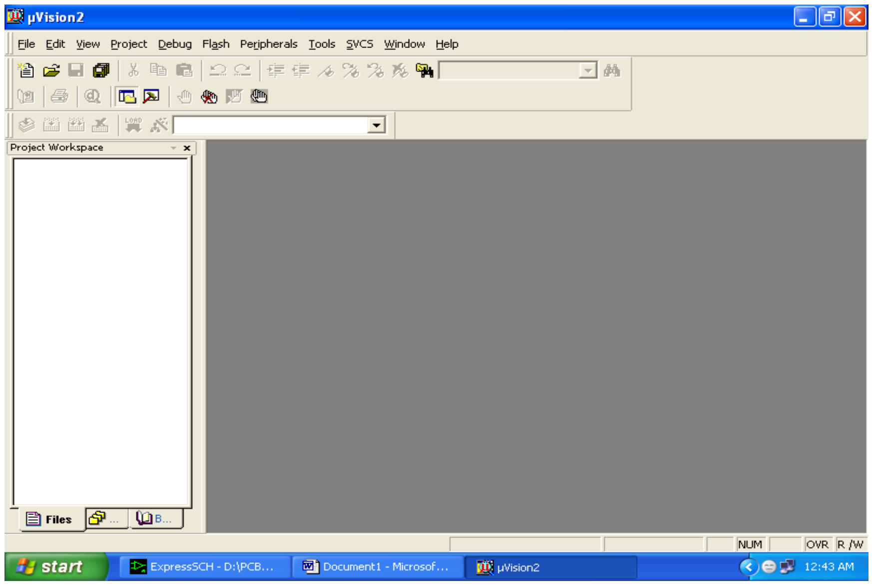
```
        Tx_String("Rs");
}
void main()
{
        Init_LCD();
        Init_SCOM();
        Init_INT0();
        Init_GSM();
        LCD_Byte(0x80,0);

LCD_String("ELECTRICITY BILLING SYSTEM");
lcd
while(1)
        {
                if(INT0_Flag)
                {
                        INT0_Flag = 0;
                        Send_Message("9429412582",Units);
                        LCD_Byte(0xC0,0);
```

position

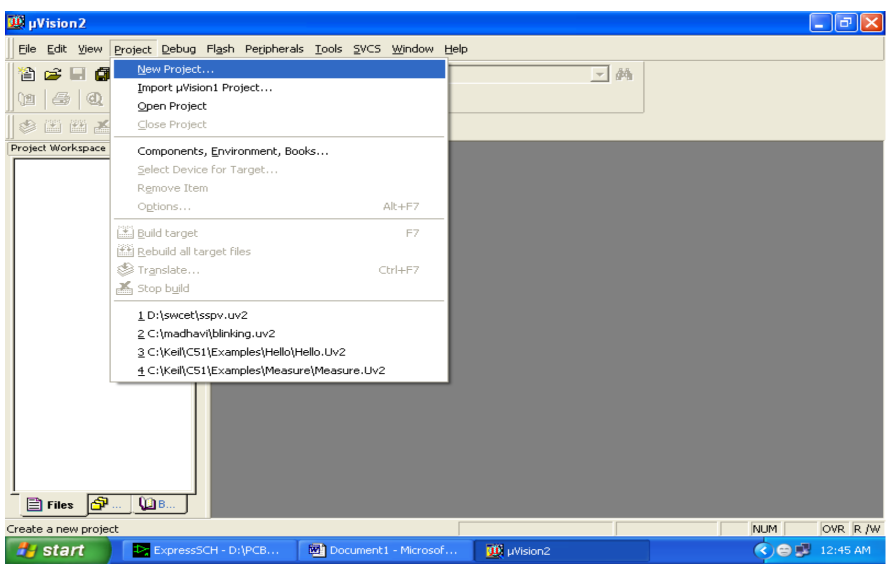
```
                    LCD_String("UNITS=");

                    LCD_NUM(UNITS);

                    LCD_Strings("COST=");

                    LCD_NUM(UNITS*6);

                    }

        }

}
```

9.1 COMPILER

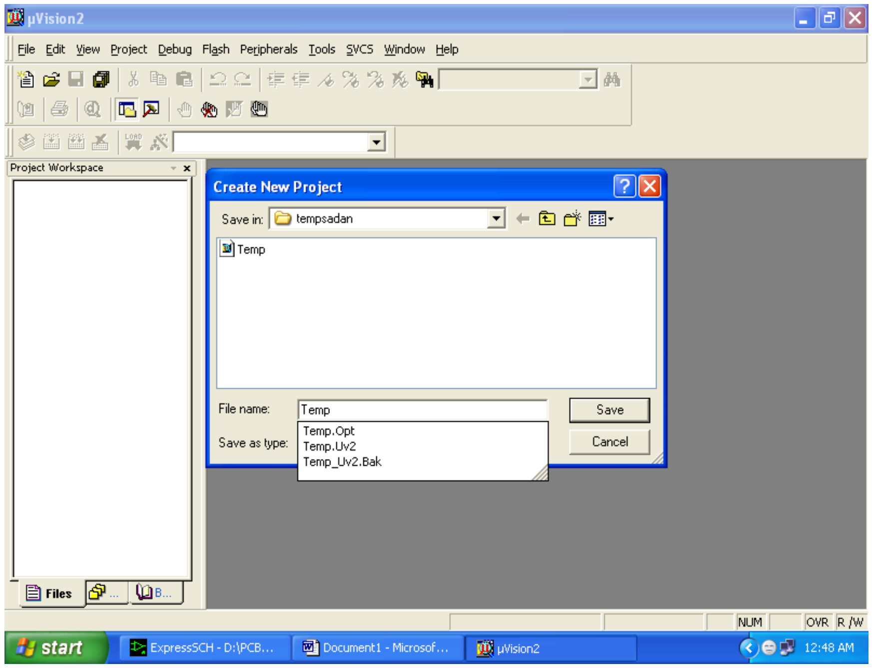
1. Click on the Keil Vision Icon on Desktop
2. The following fig will appear



3. Click on the Project menu from the title bar
4. Then Click on New Project



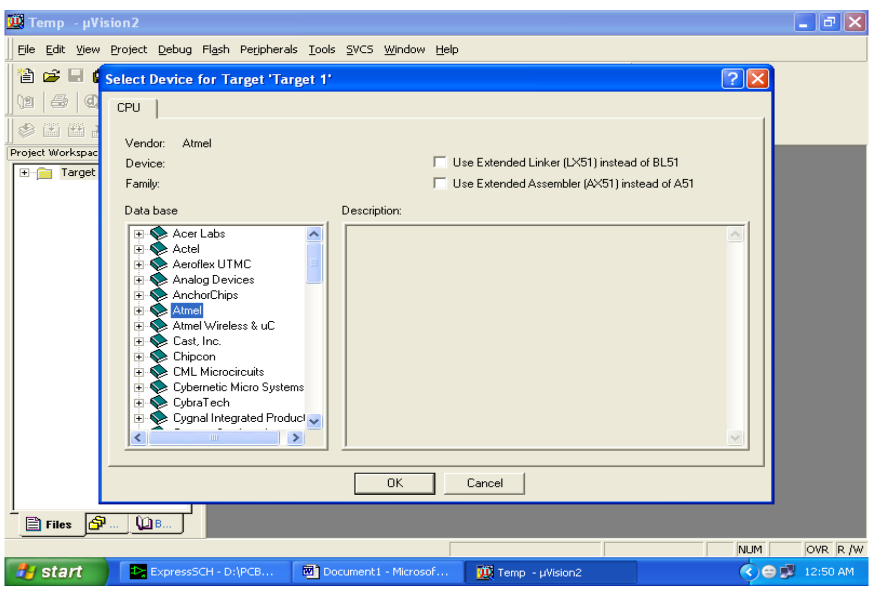
5. Save the Project by typing suitable project name with no extension in u r own folder sited in either C:\ or D:\



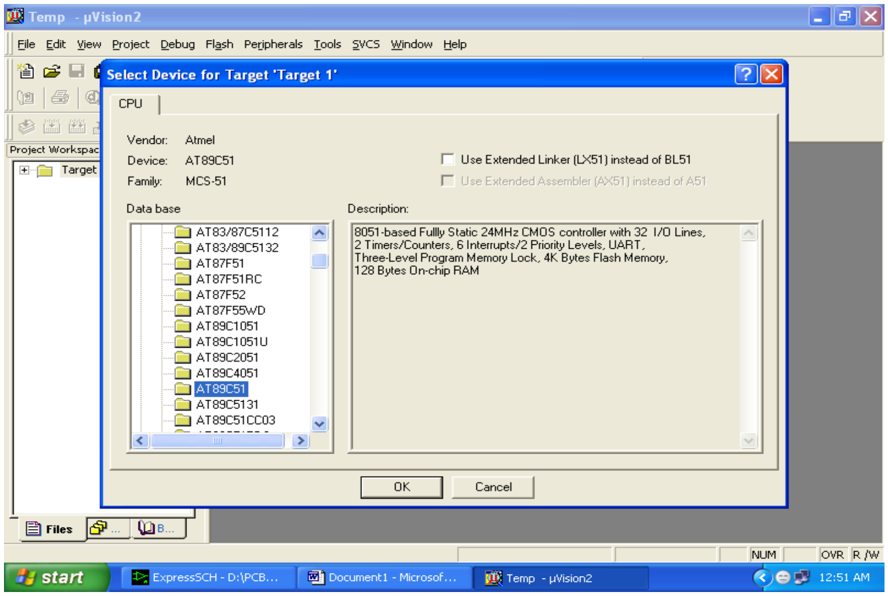
6. Then Click on Save button above.

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

8. Click on the + Symbol beside of Atmel

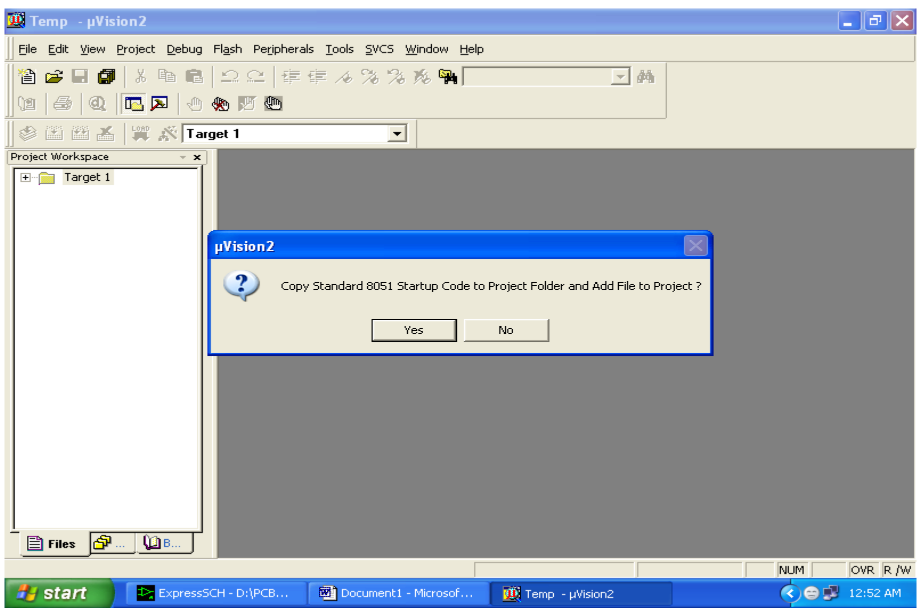


9. Select AT89C51 as shown below



10. Then Click on "OK"

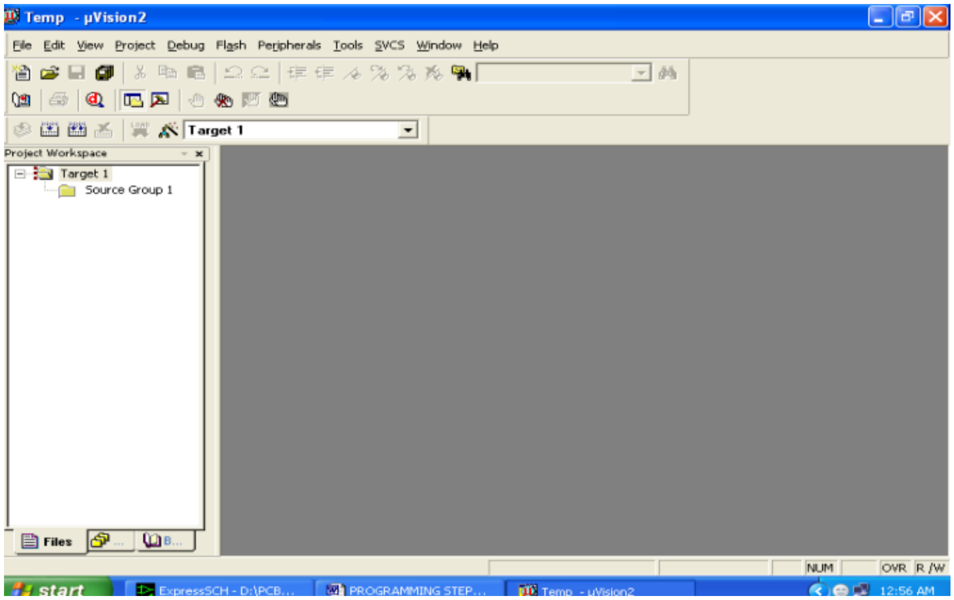
11. The Following fig will appear



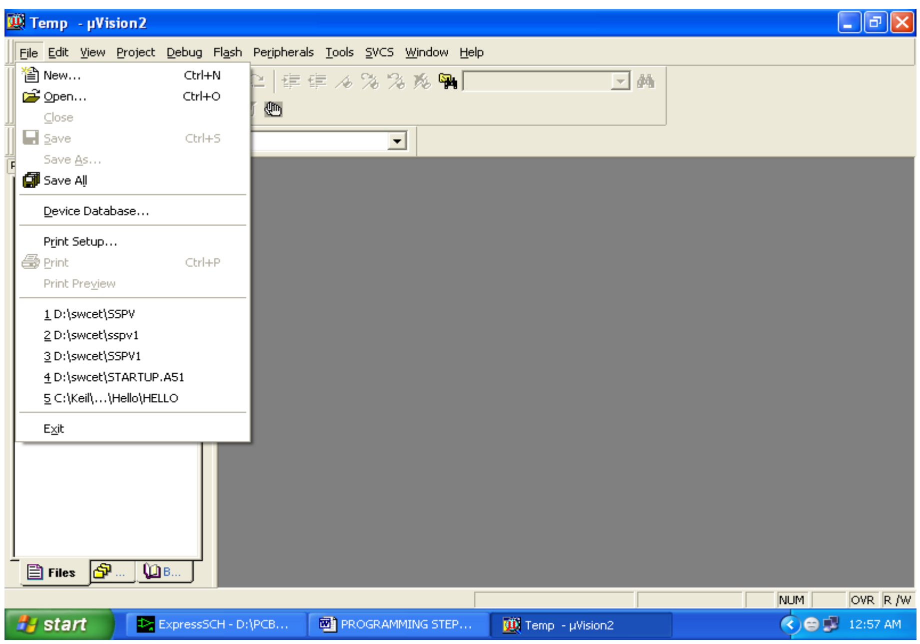
12. Then Click either YES or NO………mostly "NO".

13. Now your project is ready to USE.

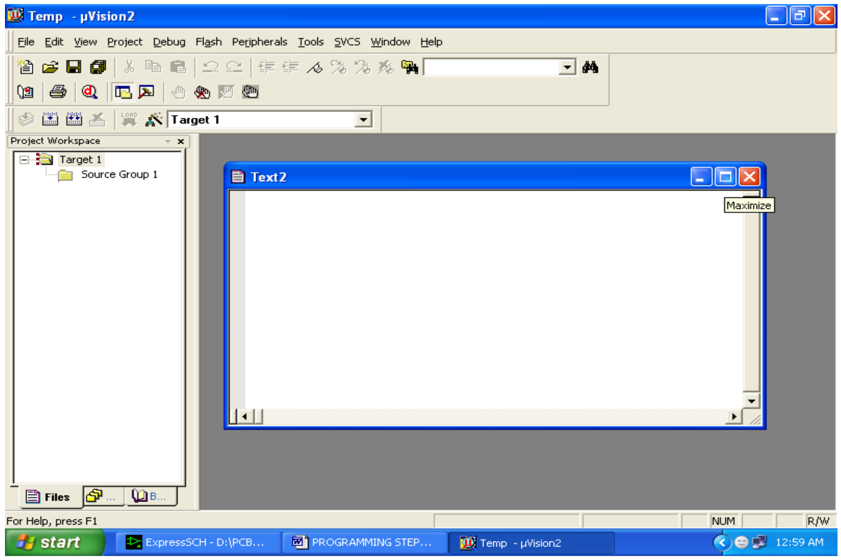
14. Now double click on the Target1, you would get another option "Source group 1" as shown in next page.



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

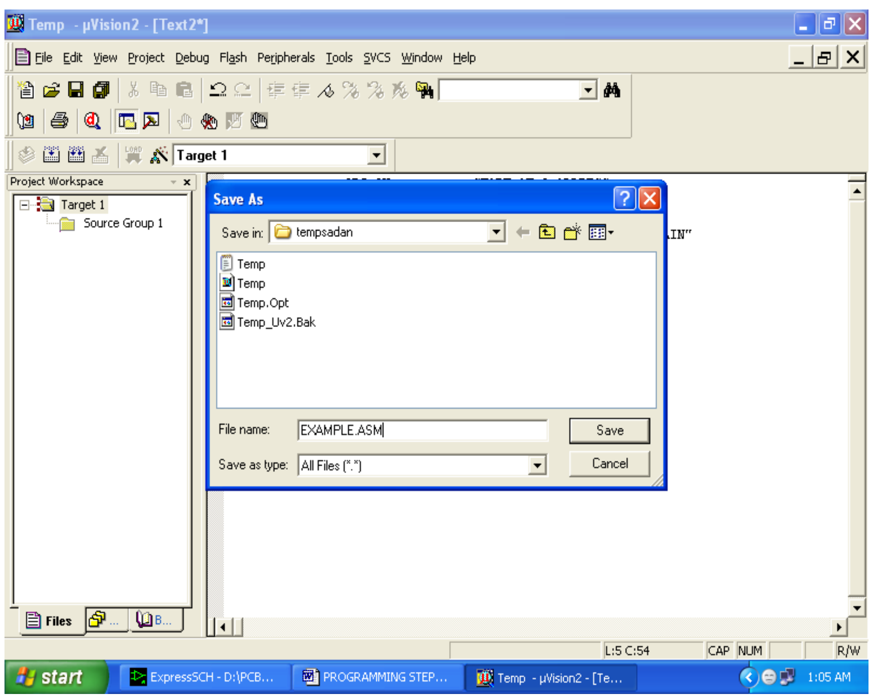


16. The next screen will be as shown in next page, and just maximize it by double clicking on its blue boarder.

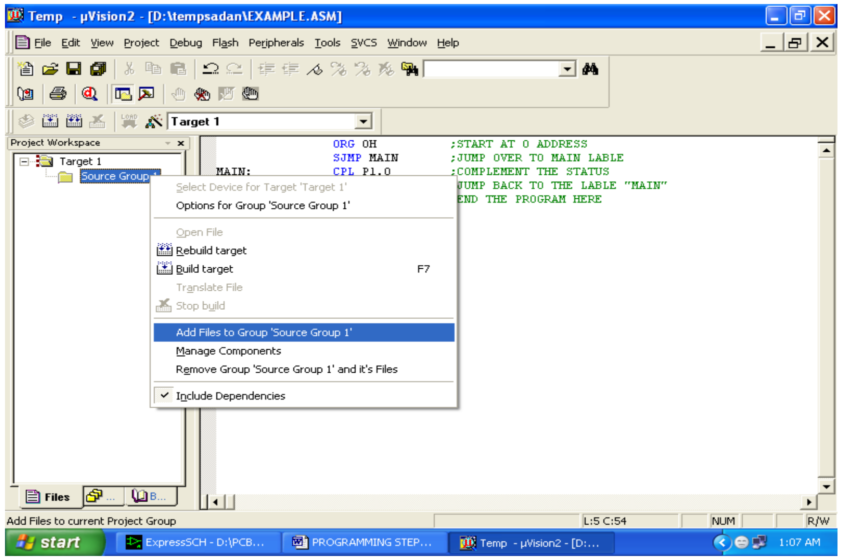


17. Now start writing program in either in “EMBEDDED C” or “ASM”.

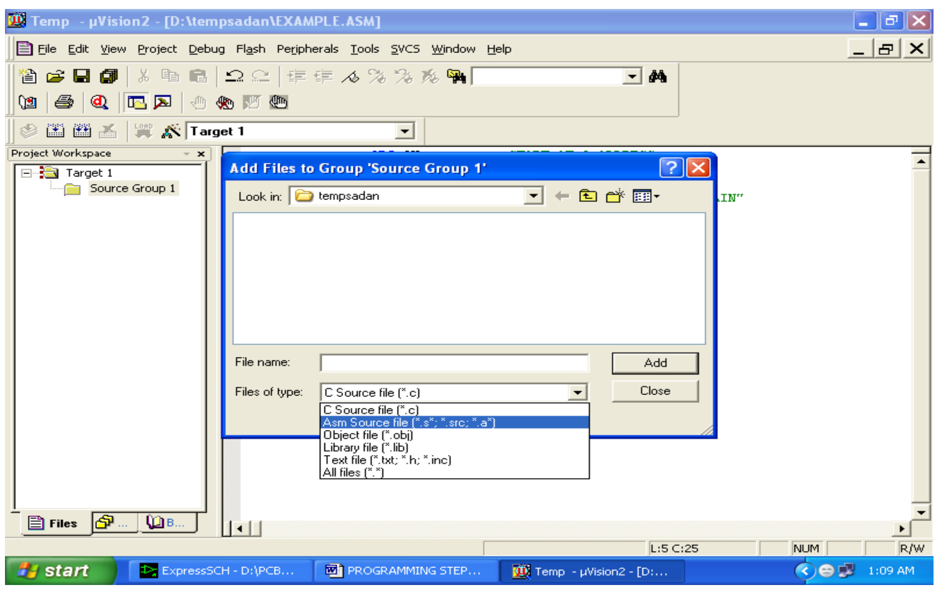
18. For a program written in Assembly, then save it with extension “. asm” and for “EMBEDDED C” based program save it with extension “ .C”



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



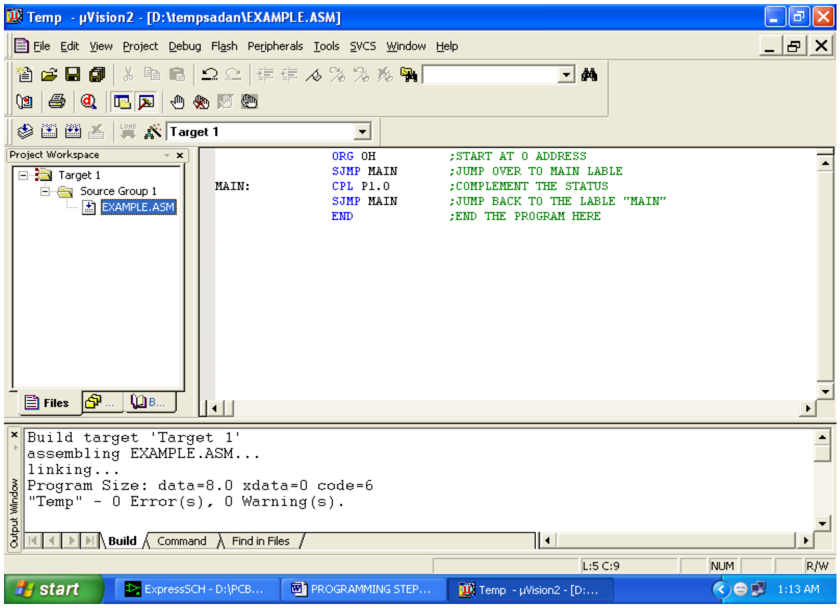
20. Now you will get another window, on which by default "EMBEDDED C" files will appear.



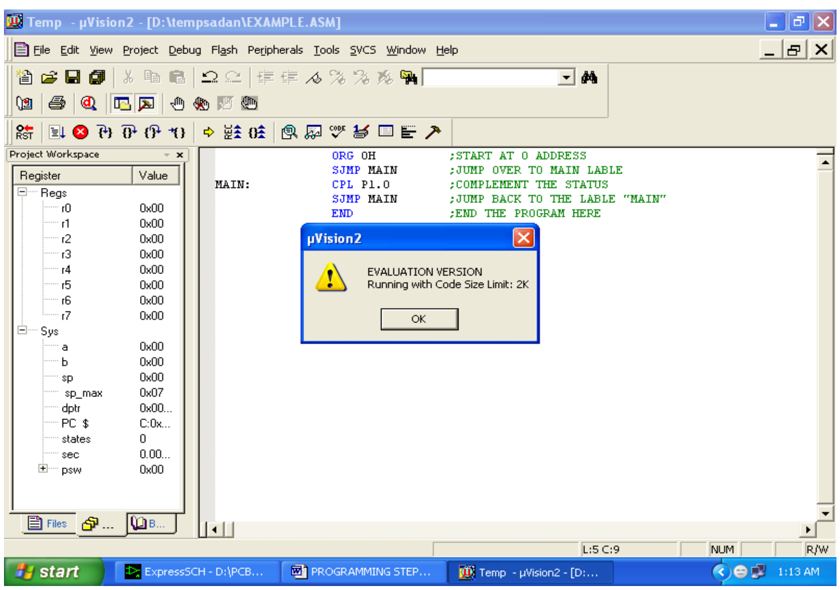
21. Now select as per your file extension given while saving the file

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

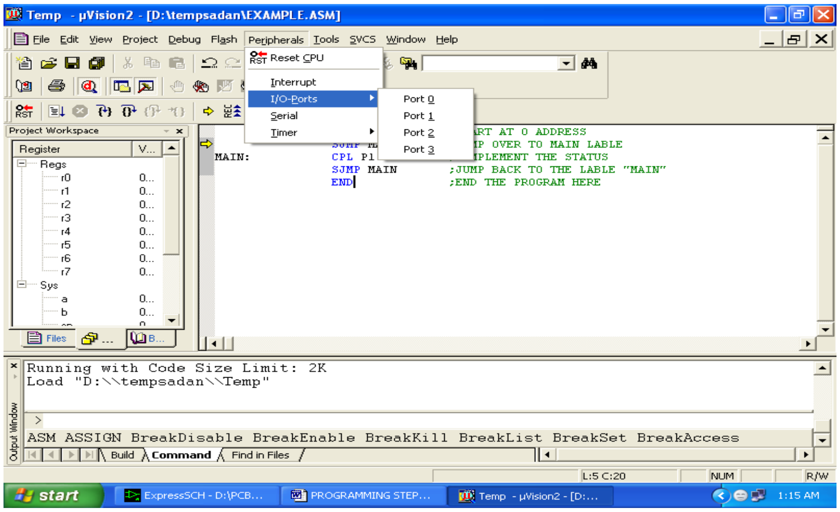
23. Now Press function key F7 to compile. Any error will appear if so happen.



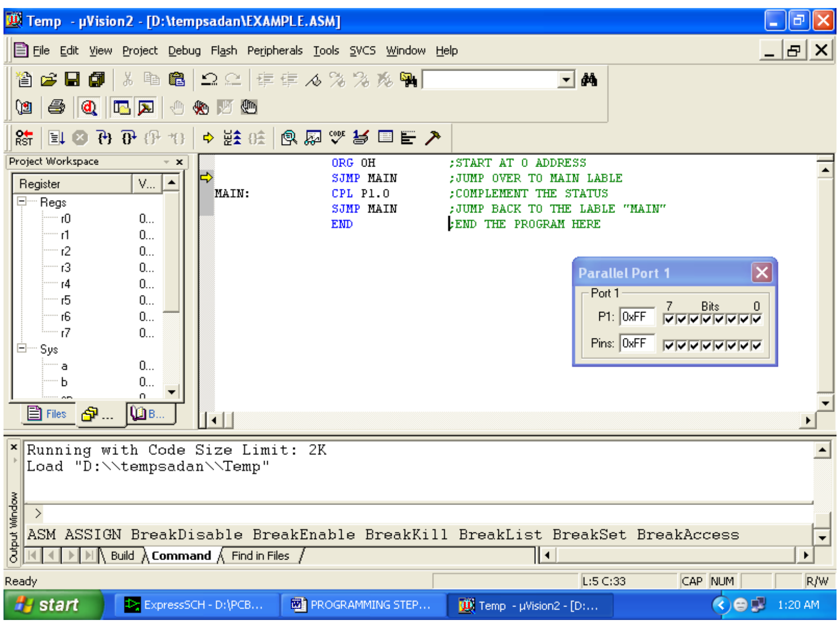
24. If the file contains no error, then press Control+F5 simultaneously.
25. The new window is as follows



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.

10.BILL OF MATERIALS

Material	Quantity	Cost	Total
Transformer Step down (230v to 12v)	1	45	45
Voltage Regulator (LM7805)	1	12	12
8051 microcontroller (AT89S52)	1	45	45
16 x 2 LCD display	1	100	100
Opto-isolator	1	100	100
GSM modem 900A	1	1000	1000
Energy Meter	1	550	550
Airtel Sim Card	1	100	100
Max – 232	1	45	45
Diode 1N007	4	2	8
Reset Switch	1	10	10
PCB Printing	2	500	1000
DB9 Connector cord	1	220	220
Db9 Connector Male	1	25	25
Red LED	1	6	6
Capacitor 10uF	2	3	6
Capacitor 470uF	1	8	8
Capacitor 33PF	2	1	2
Capacitor 1uF	4	1	4
Resistor 330 ohm	1	0.25	0.25
Resistor 10K	3	0.25	0.75
Potentiometer	1	10	10
Crystal Oscillator 11.0592MHZ	1	130	130
Nut Bolt	15	2	30
Wires	1	40	40
Grand Total			**3497**

11. HARDWARE TESTING

11.1 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.

11.2 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.

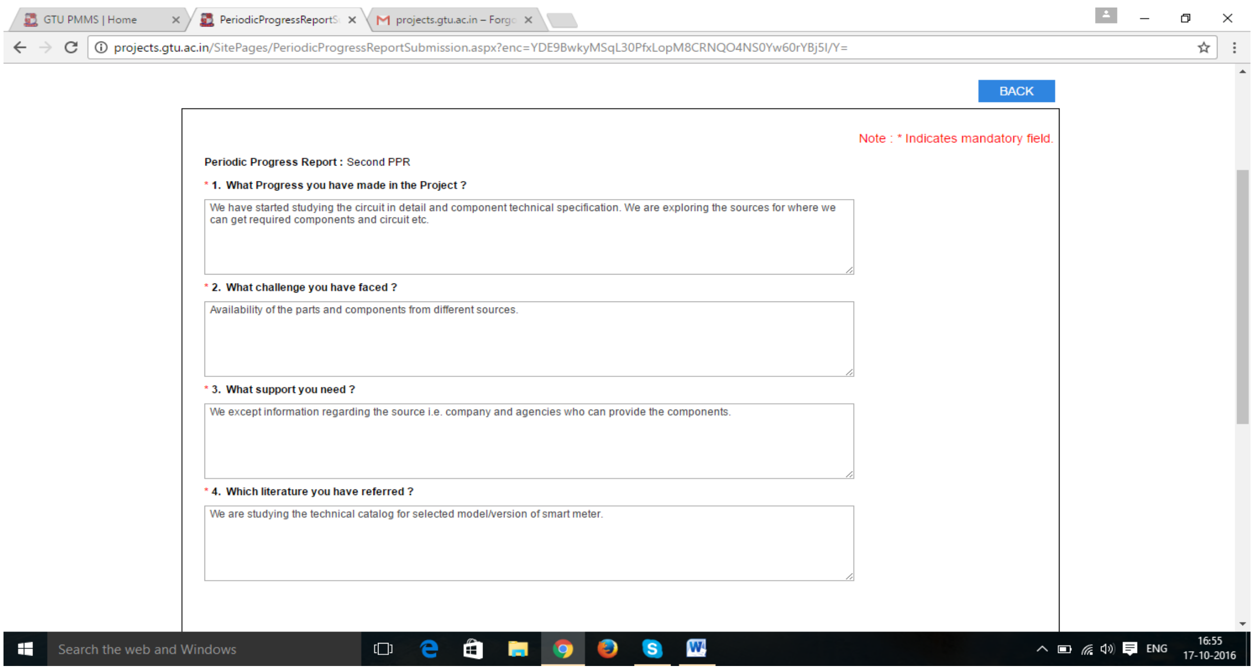
Then we apply this voltage to the power supply circuit. Note that we do this test without microcontroller because if there is any excessive voltage, this may lead to damaging the controller. We check for the input to the voltage regulator i.e., are we getting an input of 12v and an output of 5v. This 5v output is given to the microcontrollers' 40^{th} pin. Hence we check for the voltage level at 40^{th} 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.

12. CONCLUSION

Hence after studying theoretical and practical concepts of smart meter with GSM module we are able to perform and calculate the total energy consumed and from which we can modulate our costing as per the per unit costing of the consumed energy by the MGVCL on the registered mobile number.

13.PPR

PPR-1



Note : * Indicates mandatory field.

Periodic Progress Report : Second PPR

* 1. What Progress you have made in the Project ?

We have started studying the circuit in detail and component technical specification. We are exploring the sources for where we can get required components and circuit etc.

* 2. What challenge you have faced ?

Availability of the parts and components from different sources.

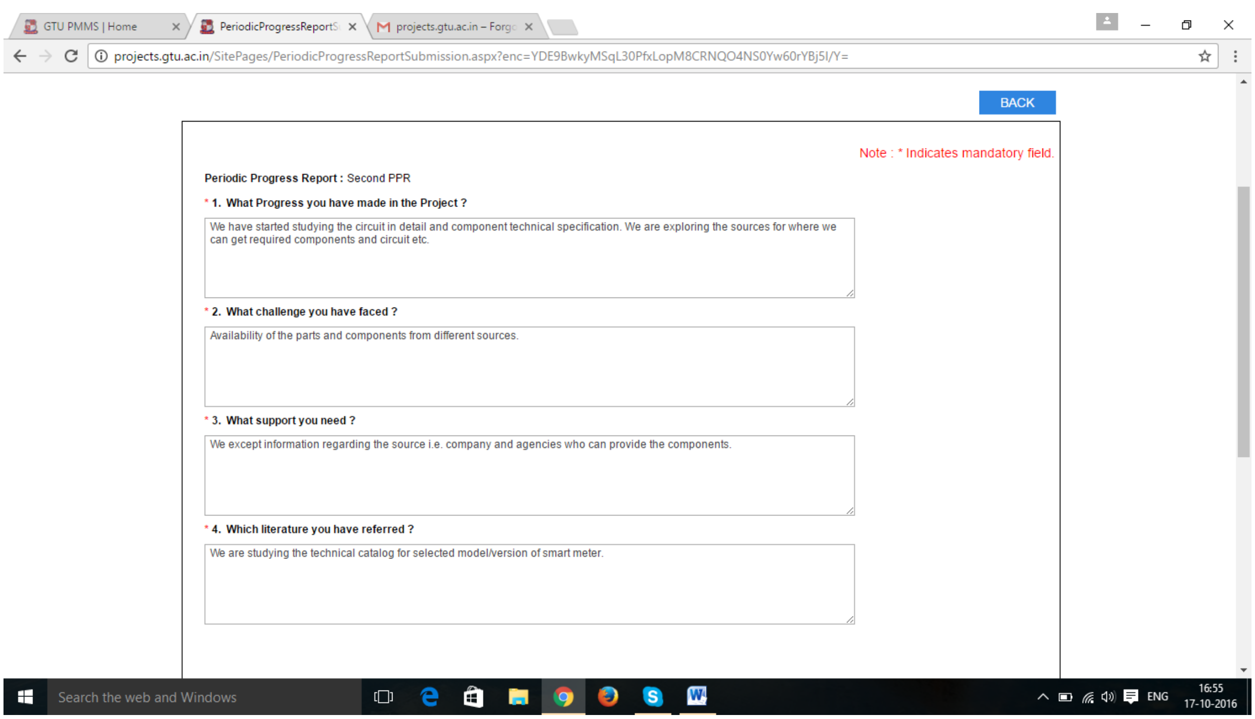
* 3. What support you need ?

We except information regarding the source i.e. company and agencies who can provide the components.

* 4. Which literature you have referred ?

We are studying the technical catalog for selected model/version of smart meter.

PPR-2



Note : * Indicates mandatory field.

Periodic Progress Report : Second PPR

* 1. What Progress you have made in the Project ?

We have started studying the circuit in detail and component technical specification. We are exploring the sources for where we can get required components and circuit etc.

* 2. What challenge you have faced ?

Availability of the parts and components from different sources.

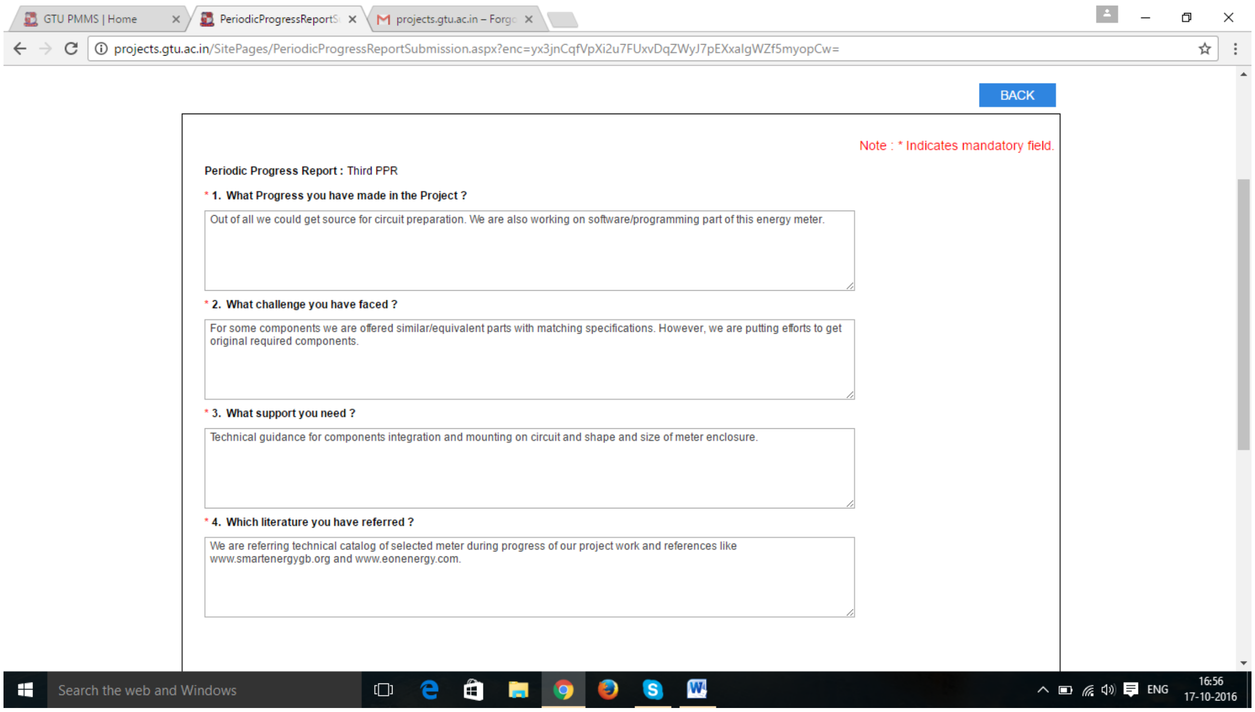
* 3. What support you need ?

We except information regarding the source i.e. company and agencies who can provide the components.

* 4. Which literature you have referred ?

We are studying the technical catalog for selected model/version of smart meter.

PPR-3



projects.gtu.ac.in/SitePages/PeriodicProgressReportSubmission.aspx?enc=yx3jnCqfVpXi2u7FUxvDqZWyJ7pEXxalgWZf5myopCw=

BACK

Note : * Indicates mandatory field.

Periodic Progress Report : Third PPR

* **1. What Progress you have made in the Project ?**

Out of all we could get source for circuit preparation. We are also working on software/programming part of this energy meter.

* **2. What challenge you have faced ?**

For some components we are offered similar/equivalent parts with matching specifications. However, we are putting efforts to get original required components.

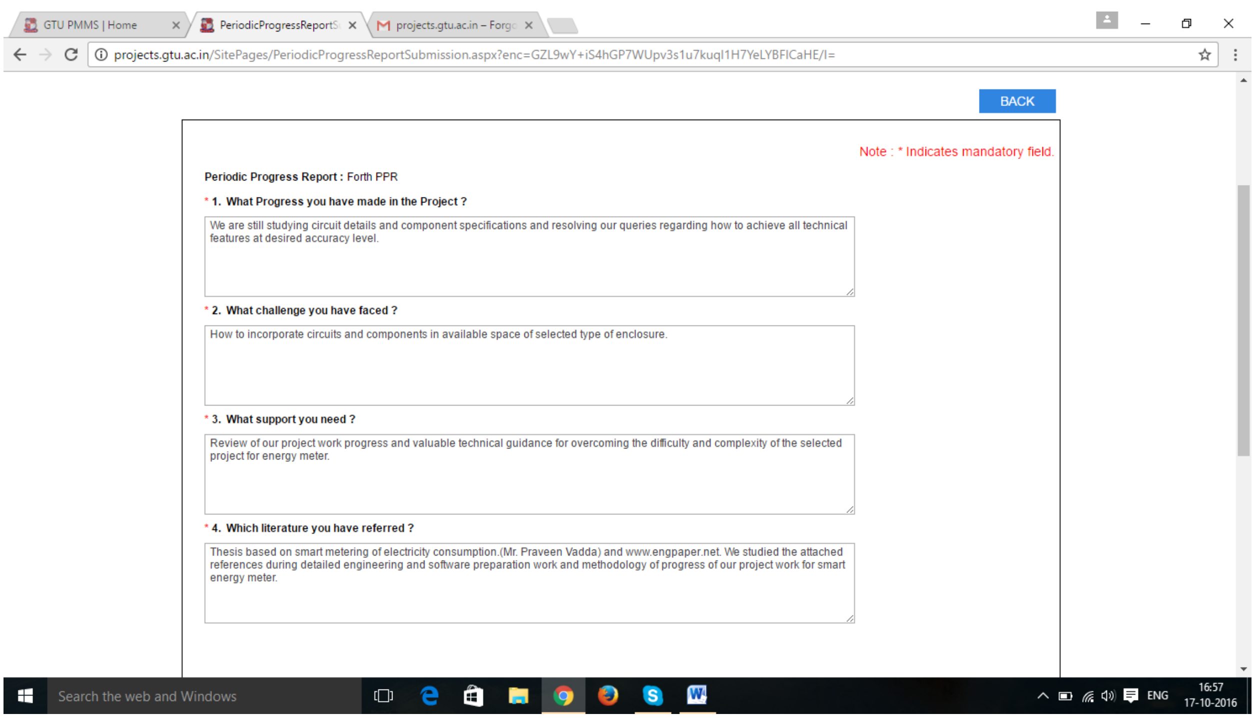
* **3. What support you need ?**

Technical guidance for components integration and mounting on circuit and shape and size of meter enclosure.

* **4. Which literature you have referred ?**

We are referring technical catalog of selected meter during progress of our project work and references like www.smartenergygb.org and www.eonenergy.com.

PPR-4



projects.gtu.ac.in/SitePages/PeriodicProgressReportSubmission.aspx?enc=GZL9wY+iS4hGP7WUpv3s1u7kuql1H7YeLYBFICaHE/I=

BACK

Note : * Indicates mandatory field.

Periodic Progress Report : Forth PPR

* **1. What Progress you have made in the Project ?**

We are still studying circuit details and component specifications and resolving our queries regarding how to achieve all technical features at desired accuracy level.

* **2. What challenge you have faced ?**

How to incorporate circuits and components in available space of selected type of enclosure.

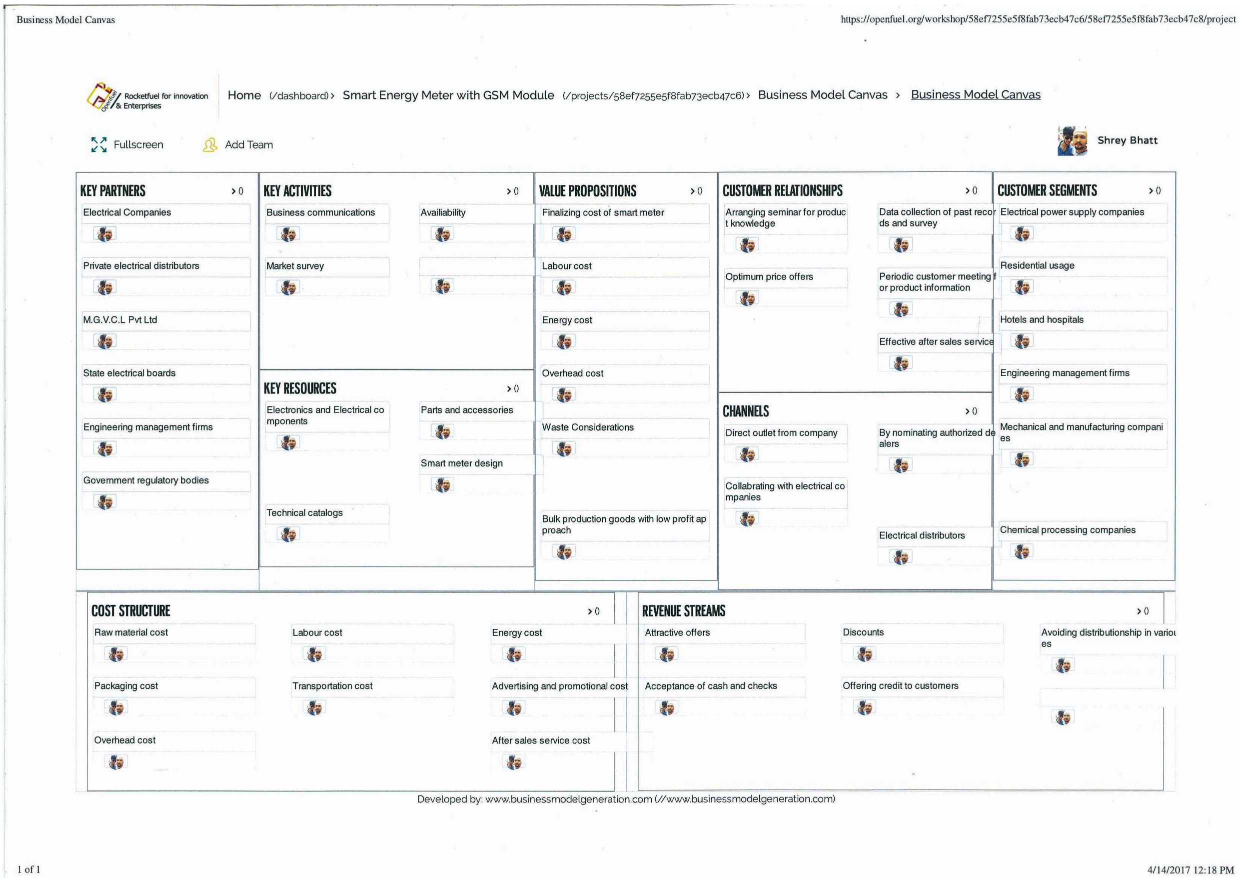
* **3. What support you need ?**

Review of our project work progress and valuable technical guidance for overcoming the difficulty and complexity of the selected project for energy meter.

* **4. Which literature you have referred ?**

Thesis based on smart metering of electricity consumption.(Mr. Praveen Vadda) and www.engpaper.net. We studied the attached references during detailed engineering and software preparation work and methodology of progress of our project work for smart energy meter.

14.BMC IMAGE



Business Model Canvas
https://openfuel.org/workshop/58ef7255e5f8fab73ecb47c6/58ef7255e5f8fab73ecb47c8/project
Rocketfuel for innovation & Enterprises
Home (/dashboard) › Smart Energy Meter with GSM Module (/projects/58ef7255e5f8fab73ecb47c6) › Business Model Canvas › Business Model Canvas
Fullscreen
Add Team
Shrey Bhatt
KEY PARTNERS
Electrical Companies
Private electrical distributors
M.G.V.C.L Pvt Ltd
State electrical boards
Engineering management firms
Government regulatory bodies
KEY ACTIVITIES
Business communications
Availiability
Market survey
KEY RESOURCES
Electronics and Electrical co mponents
Parts and accessories
Smart meter design
Technical catalogs
VALUE PROPOSITIONS
Finalizing cost of smart meter
Labour cost
Energy cost
Overhead cost
Waste Considerations
Bulk production goods with low profit ap proach
CUSTOMER RELATIONSHIPS
Arranging seminar for produc t knowledge
Data collection of past recor ds and survey
Optimum price offers
Periodic customer meeting f or product information
Effective after sales service
CHANNELS
Direct outlet from company
By nominating authorized de alers
Collabrating with electrical co mpanies
Electrical distributors
CUSTOMER SEGMENTS
Electrical power supply companies
Residential usage
Hotels and hospitals
Engineering management firms
Mechanical and manufacturing compani es
Chemical processing companies
COST STRUCTURE
Raw material cost
Labour cost
Energy cost
Packaging cost
Transportation cost
Advertising and promotional cost
Overhead cost
After sales service cost
REVENUE STREAMS
Attractive offers
Discounts
Avoiding distributionship in variou es
Acceptance of cash and checks
Offering credit to customers
Developed by: www.businessmodelgeneration.com (//www.businessmodelgeneration.com)
1 of 1
4/14/2017 12:18 PM

<u>15.BMC REPORT</u>

"BUSINESS MODEL CANVAS"

Submitted By

1. Nashikkar Amey R. 131070107018
2. Bhatt Shrey S. 131070107023
3. Hinduja Mrinal R. 130410109052
4. Thakrawala Hasnain I. 130410109118

In fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

In

ELECTRICAL ENGINEERING

SARDAR VALLABHBHAI PATEL INSTITUTE OF TECHNOLOGY, VASAD.Gujarat Technological University, Ahmadabad

CERTIFICATE

May, 2017

SARDAR VALLABHBHAI PATEL INSTITUTE OF TECHNOLOGY, VASAD.

ELECTRICAL ENGINEERING.

2017

Date: 18/05/2017

This is to certify that the dissertation entitled **"Business Canvas"** has been carried out by **Nashikkar Amey R., Bhatt Shrey S., Hinduja Mrinal R.,** and **Thakrawala Hasnain I.** under my guidance in fulfillment of the degree of Bachelor of Engineering in ELECTRICAL (8th Sem) of Gujarat Technological University, Ahmedabad during the academic year 2016-17.

Guide: **Head of Department**

[Mr. Rakesh Gajjar] **[Mr.Chetan Kotwal]**

TABLE OF CONTENT

➢ Key partners

- Electrical Companies
- Private Electrical Distributors
- State Electrical Boards
- MGVCL PVT LTD
- Engineering Management Firms

➢ Key activity

- Availiability
- Market Survey
- Business Communications.

➢ Key resources

- Technical Catalogs
- Smart meter design
- Electrical and Electronics components
- Parts and accessories.

➢ Value proposition

- Finalizing the smart meter costing.
- Labour Cost.
- Energy Cost.
- Overhead Cost.
- Waste Consideration.
- Bulk production with the small profit approach.

➢ Customer relationship

- Arranging seminar for product knowledge
- Data collections of past records and survey.
- Optimum price offers.
- Periodic customer meeting
- Effective after sales service.

➢ Channels

- Direct outlet from company
- By nominating authorized dealers
- Collaborating with Electrical Companies
- Electrical Distributors
-

➢ Customer segment

- Electrical Power Supply Companies
- Hotels and Hospitals
- Residential Usage
- Engineering management firms
- Chemical Processing Companies
- Mechanical and Manufacturing Companies.

➢ Cost structure

- Raw Material Cost
- Labour Cost
- Energy Cost
- Overhead Cost
- Wastage Cost
- Packing Cost
- Advertising and Promotional Cost
- Transportation Cost.
-

➢ Revenue stream

- Discounts
- Attractive Offers
- Acceptance of cash and cheques
- Offering credits to customers
- Avoiding distributionship in various state

16.BIBLIOGRAPHY

TEXT BOOKS REFERED:

1. “The 8051 Microcontroller and Embedded systems” by Muhammad Ali Mazidi and Janice Gillispie Mazidi , Pearson Education.

2. ATMEL 89S52 Data Sheets.

WEBSITES

- www.atmel.com
- www.beyondlogic.org
- www.wikipedia.org
- www.howstuffworks.com
- www.alldatasheets.com
- https://pdfs.semanticscholar.org/0ce2/ab6e4a1f95128d53cee39a3c8bb7f14e2fb6.pdf
- https://www.accenture.com/fi-en/~/media/Accenture/Conversion-Assets/DotCom/Documents/Global/PDF/Industries_9/Accenture-Smart-Metering-Report-Digitally-Enabled-Grid.pdf
- https://www.youtube.com
- https://www.electrical4u.com
- http://electronicsforu.com/