IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

FABRICATION OF A MOBILE USING GSM BOARD AT COMMAND ARDUINO PROGRAMMING

K. Tharun Teja*, Chinthareddy Sri Sai Ravali, Jammula Hima Bindu
*B.Tech Student, Department of ECE, CVR College of Engineering, India.
B.Tech Student, Department of ECE, CVR College of Engineering, India.
B.Tech Student, Department of ECE, CVR College of Engineering, India.

ABSTRACT

A Mobile is a small computing device typically small enough to be handheld having a display screen and a miniature keyboard with weight less than 0.2 kg. A hand held computing device has an operating system and can run various types of application soft wares known as applications. The most common size of the mobile computing device is the pocket sized that can be handled. The inherent flexibility in the hardware and software makes mobile unique compared to other technologies. In the proposed work a mobile unit is fabricated using GSM board at command Arduino programming. The Arduino is the open source hardware. The Arduino board can act as a standalone system with capabilities to take input and then process the input and thereby generate a corresponding output. Arduino consists of both physical programmable circuit and a piece of software on IDE that runs on your computer. The program here is written in C or C++.

KEYWORDS: Arduino, GSM, Mobile, AT mega.

INTRODUCTION

Embedded systems are present everywhere now a days and make possible the creation of systems with the functionality that cannot be provided by human beings. Mobile phone is one of the most important applications of embedded systems. A mobile phone is a very small portable radio telephone. In addition to being a telephone, modern mobile phones also support many additional services such as SMS, E - mail, internet access, music, videos…etc. The aim of this paper is to fabricate a mobile that is capable of making and receiving calls and sending the messages. First we program the Arduino board using the Arduino IDE software accordingly to perform the required operations. By using the keypad the required input is given which will be displayed on the LCD screen. The GSM module interfaced with an Arduino will enable the user to control the system by sending messages or making calls. The communication is done through serial communication which is done by sending or receiving commands.

GSM MODULE AT COMMAND ARDUINO PROGRAMMING

The Arduino is a family of microcontroller boards to simplify electronic design, prototyping and experimenting for artists, hackers, hobbyists, but also many professionals. People use it as brains for their robots, to build new digital music instruments, or to build a system that lets your house plants tweet you when they're dry. Arduinos (we use the standard Arduino Uno) are built around an AT mega microcontroller — essentially a complete computer with CPU, RAM, Flash memory, and input/output pins, all on a single chip. Unlike, say, a Raspberry Pi, it's designed to attach all kinds of sensors, LEDs, small motors and speakers, servos, etc. directly to these pins, which can read in or output digital or analog voltages between 0 and 5 volts. The Arduino connects to your computer via USB, where you program it in a simple language (C/C++, similar to Java) from inside the free Arduino IDE by uploading your compiled code to the board. Once programmed, the Arduino can run with the USB link back to your computer, or stand-alone without it — no keyboard or screen needed, just power.

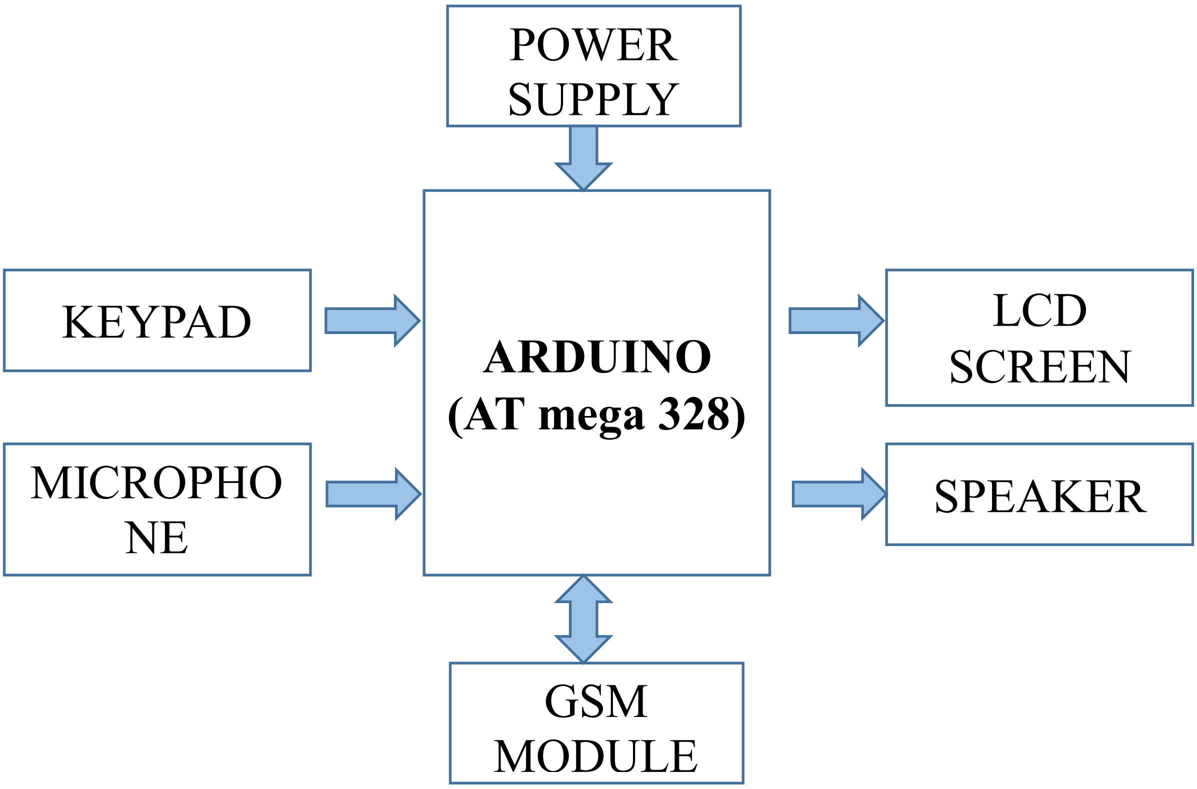


Fig 2.1: Block Diagram of Mobile using GSM Module AT command Arduino Programming

Looking at the board from the top to down, this is an outline of what we will see of the parts of the board you might interact with in the course of normal use are highlighted.

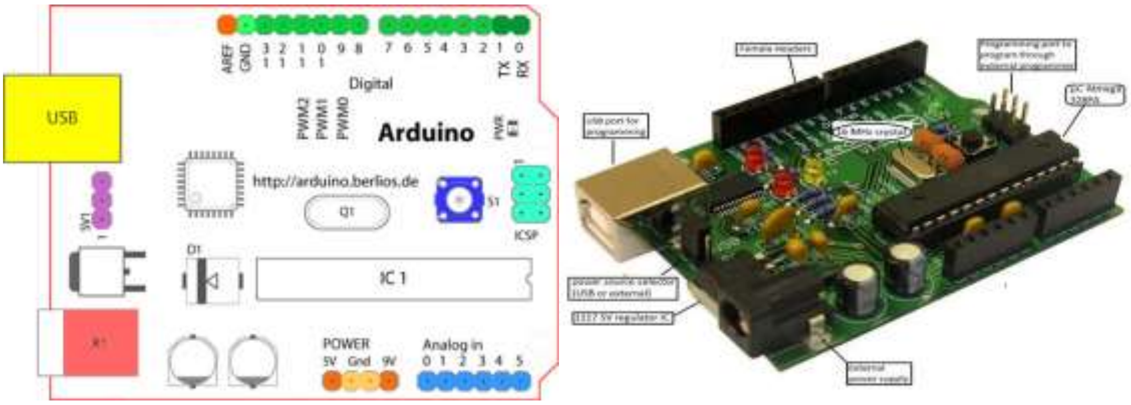


Fig 2.2: Arduino Board

Starting clockwise from the top center:

- Analog Reference pin represented in orange.
- Digital Ground represented in light green.
- Digital Pins 2-13 represented in green.
- Digital Pins 0-1/Serial In/Out - TX/RX represented in dark green. These pins cannot be used for digital I/O (Digital Read and Digital Write) if you are using serial communication like Serial Begin.
- Reset Button - S1 represented in dark blue.
- In-circuit Serial Programmer represented in blue-green.
- Analog In Pins 0-5 represented in light blue.
- Power and Ground Pins represented as power: orange, grounds: light orange.
- External Power Supply In (9-12VDC) - X1 represented in pink.

- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 represented in purple.
- USB which is used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board is represented in yellow.

AT COMMANDS

The AT commands include a prefix AT which indicates the beginning of the command to MODEM; and a carriage return which indicates the end of the command.

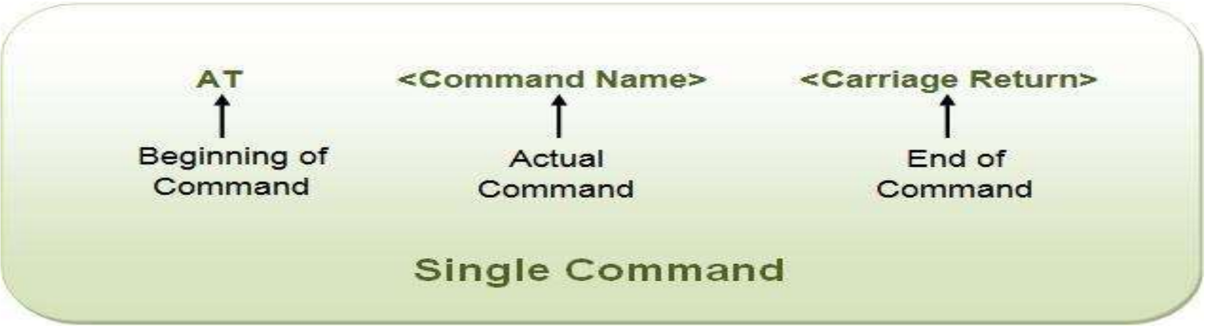


Fig 3.1: Using a Single AT Command

RESULT CODE	DESCRIPTION
OK	Successful Execution of a command
ERROR	Execution of a command failed
+CMS ERROR	Message service failure, is returned with an error code
Unsolicited Result Codes	
+CDS	Notify receipt of SMS status report of a new message to computer
+CDSI	Notify receipt of SMS status report of a new message and its location in memory to computer
+CMT	Notify forwarding of a new SMS to computer
+CMTI	Notify receipt of SMS status report of a new message and its location in memory to computer

Table: Result Codes and Description

SOFTWARE TOOLS

The software used in the paper is ARDUINO Software (IDE). The tools are as follows:

- **Auto Format** This formats your code appropriately i.e. it performs indentation so that opening and closing curly braces line up systematically and that the statements inside curly braces are also indented.
- **Archive Sketch**
  Archives a copy of the current sketch in .zip format. The archive is made in the same directory as that of the current sketch.
- **Fix Encoding & Reload**
  Fixes possible discrepancies between the editor char map encoding and the other Operating System char maps.
- **Serial Monitor**
  Opens the serial monitor window and initiates the exchange of data with any connected board on the currently selected port. It usually resets the board only if the board supports Reset over serial port opening.
- **Board**
  Selects the board that you're presently using.
- **Port**
  This menu contains all the serial devices (real / virtual) on your machine. It will automatically refresh

every time you open the top-level tools menu.

IMPLEMENTATION AND RESULTS

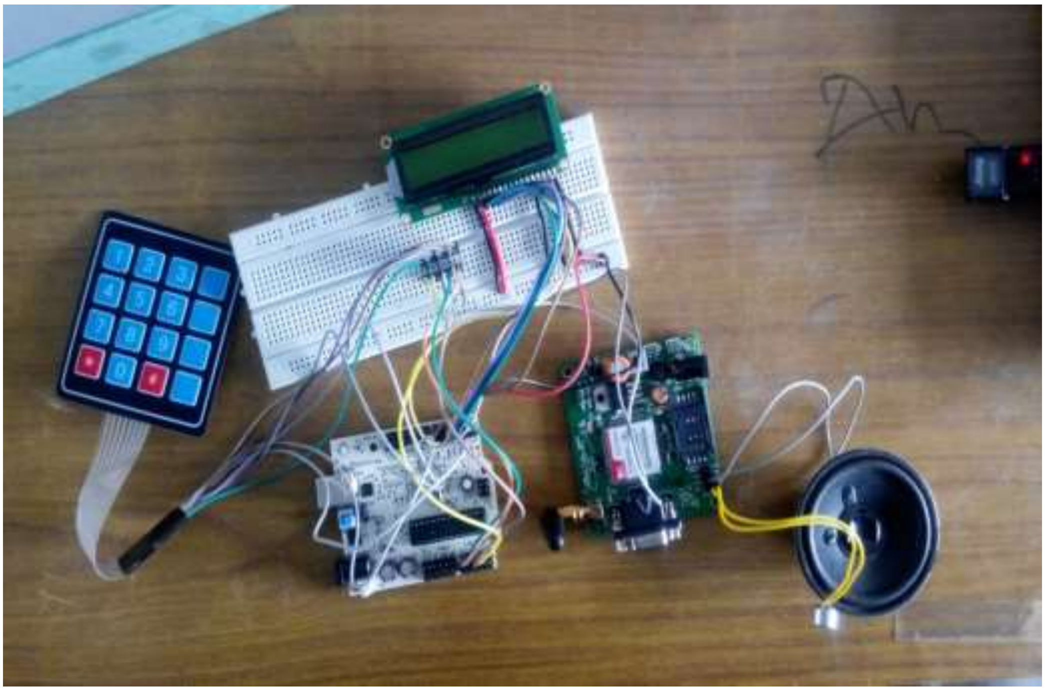
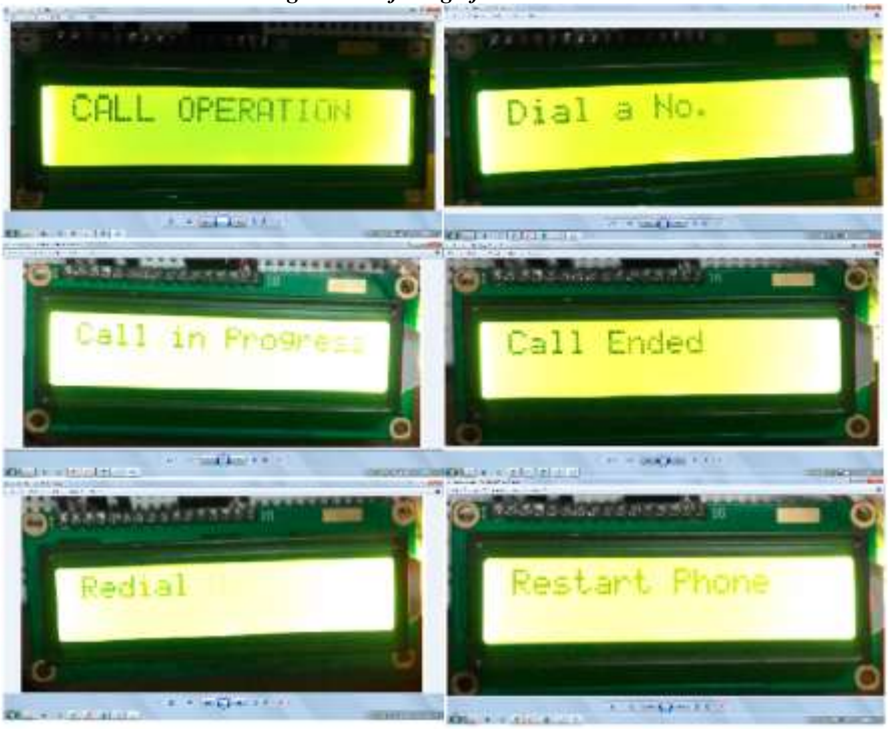


Fig 5.1 Interfacing of all the devices



CALL OPERATION

Dial a No.

Call in Progress

Call Ended

Redial

Restart Phone

Fig 5.2: Output for call operation

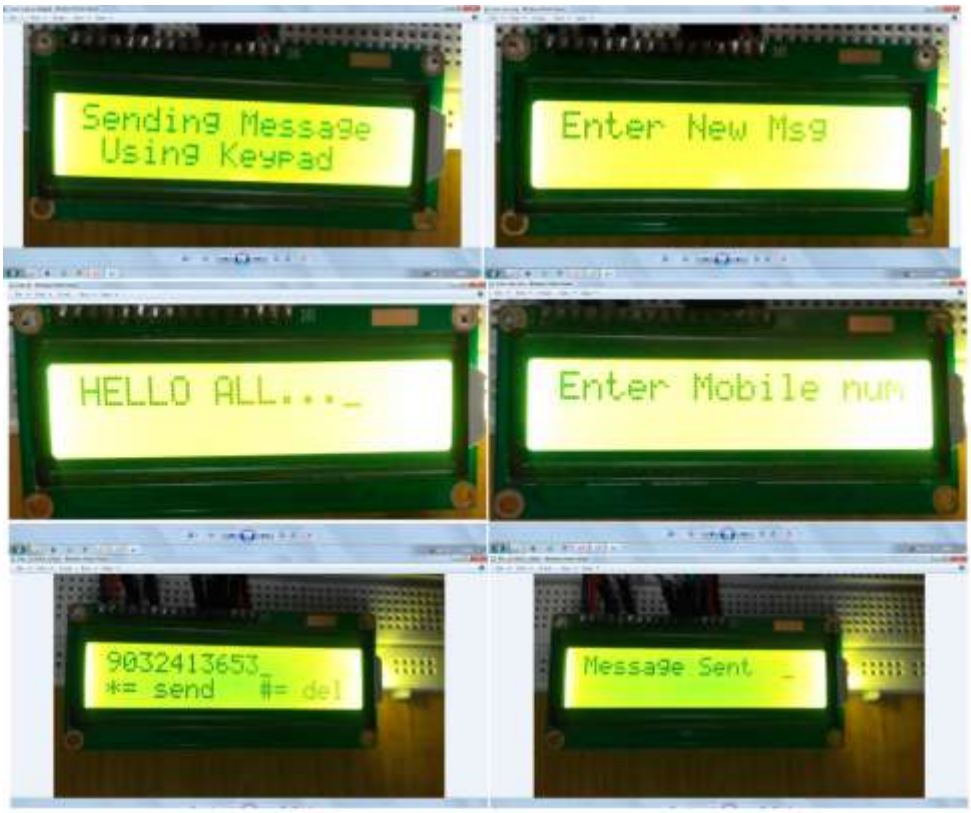


Fig 5.3: Output for message operation

CONCLUSION

The making of mobile unit which is capable of making and receiving calls and sending text messages using Arduino as a processor is successfully explained in this paper. In future we can improve by adding some more features like Games, MMS, Calculator and Radio etc.

REFERENCES

1. “AVR Microcontroller and Embedded Systems” by “Muhammad Ali Mazidi”, Pearson Education.
2. “Programming and Customizing the AVR Microcontroller”, by “Dhananjay Gadra”, Mc Graw Hill
3. “Introduction to Embedded Systems” by “Shibu K V”
4. “Embedded Systems: Architecture, Programming and Design” by “Raj kamal”, 2nd Edition
5. https://www.arduino.cc/en/Guide/HomePage
6. http://wiki.iteadstudio.com/SIM900/SIM900A_GSM/GPRS_Minimum_System_Module
7. http://www.circuitstoday.com/interface-gsm-module-with-arduino
8. http://www.engineersgarage.com/embedded/arduino/how-to-interface-4-bit-lcd-with-arduino