International Journal of Research
Available at https://edupediapublications.org/journals

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 10
September 2017

Android Enabled Light via GSM

V. Thirupathi & C.H. Sandeep
Assistant professor Computer Science and Engineering S.R.Engineering college Warangal, Telangana
thirupathi_vadluri@srecwarangal.ac.in
sandeep892@gmail.com

***Abstract*—*"Android Enabled Light via GSM" implements the emerging applications of the GSM technology. Using GSM networks, a control system has been proposed that will act as an embedded system which can monitor and control appliances and other devices locally using built-in input and output peripherals. Remotely the system allows the user to effectively monitor and control the house/office appliances and equipment's via the mobile phone set by sending commands in the form of SMS messages and receiving the appliances status. The main concept behind the project is receiving the sent SMS and processing it further as required to perform several operations. The type of the operation to be performed depends on the nature of the SMS sent. The principle in which the project is based is fairly simple. First, the sent SMS is stored and polled from the receiver mobile station and then the required control signal is generated and sent to the intermediate hardware that we have designed according to the command received in form of the sent message. We have selected a android phone for our project. The messages are sent from the mobile set that contain commands in written form which are then processed accordingly to perform the required task. A microcontroller based system has been used for our project. **GSM (Global System for Mobile Communications)**: It is a cellular communication standard. **SMS (Short Message Service):** It is a service available on most digital mobile phones that permit the sending of short messages (also known as text messaging service).*

***Keywords*—GSM, SMS, Android , Arduino.**

Existing System

The existing system works in the following way.

- It is a manual processing system.
- It takes lot of time to turn ON/OFF switches when we are away from house or office.
- Appliances are controlled from particular location only.

Disadvantages:

- Time consuming.
- Risk of short-circuit.
- Cannot be controlled from remote location.

Proposed System

In order to overcome the above mentioned problem we introduce the GSM based system for enabling the switching of light using android technology. The proposed system works in the following way.

- The remote user sends text messages including commands to the receiver.
- GSM receiver receives messages sent from the user cellphone.
- GSM receiver decodes the message and sends the command to the microcontroller.

Micro-controller issues commands to the appliances and the devices connected with switch ON/OFF

Advantages:

- Automates the home appliances.
- Enables ON /OFF of light via SMS.
- Switching of light can controlled from remote area.
- Time saving
- Effortless
- Powerful and flexible system that will offer services at any time

LITERATURE SURVEY

Imagine being able to control all the electrical appliances of your home from virtually any place you are in the world! For example, say one cold winter day, you are stuck in traffic on the motorway after a long day at work. By pressing a few keystrokes on your mobile phone, you activate a macro which turns on the heating, prepares your Jacuzzi , switches on your electrical fireplace and lava-lamps at your premises.

In today's times the GSM mobile terminal has become one of the items that is constantly with us. Just like our wallet/purse, keys or watch. The GSM mobile terminal provides us with a communication channel that enables us to communicate with the world. The want for a person to be reachable or to call anyone at any time is very appealing.

Using the cybernetic concept of action space, we in fact extend the action space of the human with the mobile terminal, enabling the subscriber to talk to another subscriber and thus exchange information, issue or receive commands.

In this project, we will explore ways of extending this notion of Home Automation by applying mobility to it using various technologies that are available in our information age. These technologies include mobile communication devices and the Internet which are becoming part of our everyday lives.

SYSTEM SPECIFICATIONS

The system specification shows the description of the function and the performance of system and the user. The scope of our project “Android enabled light via GSM” is immense. The future implications of the project are very great considering the amount of time and resources it saves. The project we have undertaken can be used as a reference or as a base for realizing a scheme to be implemented in other projects of greater level such as weather forecasting, temperature updates, device synchronization, etc. The project itself can be modified to achieve a complete Home Automation system which will then create a platform for the user to interface between himself and the household.

Goals and Objectives

The project “Android enabled light via GSM” at the title suggests is aimed to construct a control system that enables the complete control of the interface on which it is based. General objectives of the project are defined as

a. To co-ordinate appliances and other devices through Short Message Service (SMS).

b. To effectively receive and transmit data via SMS

c. To eliminate the need of being physically present in any location for tasks involving the operation of appliances within a household/office.

d. Minimize power and time wastage

Operating Environment

The control system will include two separate units: the cellular phone, and the control unit. There will therefore be two operating environments. The cellular phone will operate indoors and outdoors whereas the control unit will operate indoors within the temperature and humidity limits for proper operation of the hardware.

Intended Users and Uses

This system is aimed toward all the average users who wish to control their household/office appliances remotely from their cell phones provided that the appliances are electrically controllable. Example of feasible appliances and applications under consideration include; enable/disable security systems, fans, lights, kitchen appliances, and adjusting the temperatures settings of a heating/ventilation/air conditioning system.

Assumptions

Certain assumptions have to be made in order to implement our project. The list of assumptions for our project is

a. The user and control unit will establish communication via GSM

b. The cell phone and service provider chosen will support text messaging service.

c. The user is familiar with the text messaging program on their cell phone.

d. All service charges (standard messaging rates) from the service provider apply.

e. The controlled appliances can and will have to have an electrical interface in order to be controlled by the microcontroller.

Major Constraints

Along the course of project completion we encountered various problems and obstacles. Not everything that we had planned went smoothly during the project development span. Also we had a limited amount of time for its completion so we were under a certain amount of pressure as well. We had to start from the research phase at the beginning and needed to gain knowledge on all the devices and components that we had intended to use for our project. Other phases of the project included coding, debugging, testing, documentation and implementation and it needed certain time for completion so we really had to manage the limited time available to us and work accordingly to finish the project within the schedule.

Constraints Considerations

The following is a list of constraint Considerations

a. The controlled appliances will need an electrical control interface. This system is only capable of controlling electrical devices.
b. The control module will need to be shielded against electrostatic discharges. This will increase the reliability of the system.
c. Battery backup for controlling unit can be implemented in case of power disruption.

Technology Considerations

The considerations for this system will include a choice of networks, communication protocols and interfaces.
a. Cellular Networks: The widely available networks are based on GSM. This network provides wide area coverage and can be utilized more cost-effectively for this project.
b. Communication Protocols: The available communication protocol that we have used is SMS. The SMS is the most efficient because this project requires a cellular communication and limited data to be sent.
c. I/O interfaces between microcontroller and devices: Serial I/O is considered as options for connection between the GSM receiver and the microcontroller. Using the microcontroller, a control circuit will be implemented to control the electrical appliances.

SYSTEM ARCHITECTURE

The System was designed with Fig-1 as the block diagram, the first Mobile station is used as a transmitting section from which the subscriber sends text messages that contain commands and instructions to the second mobile station which is based on a specific area where our control system is located. The mobile phone as indicated in the block diagram is a android mobile. It can pick signal in the worst places and the battery is durable. It is also easy to use and navigate; and the keypad is very easy for texting. The received SMS message is stored in the SIM memory of the phone and then extracted by the microcontroller and processed accordingly to carry out specific operations. The relay driver is used to drive the relay circuits which switches the different appliances connected to the interface. The LED is used to indicate the status of the operation performed by the microcontroller and also its inclusion makes the overall system user-friendly. The SMS from the user cell phone was transmitted through the GSM chip to the microcontroller and the microcontroller finally performed the required instructions as stated by the user. Assuming that the control unit is powered and operating properly, the process of controlling a device connected to the interface will proceed through the following steps

- The remote user sends text messages including commands to the receiver.
- GSM receiver receives messages sent from the user cell phone.
- GSM receiver decodes the sent message and sends the commands to the microcontroller.
- Microcontroller issues commands to the appliances and the devices connected will switch ON/OFF

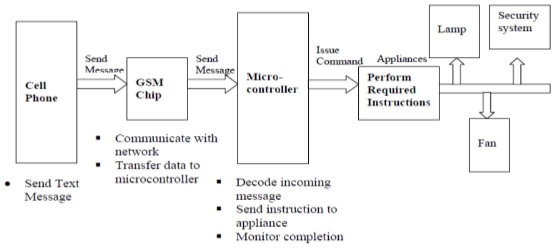


Fig-1 Block diagram

Modules

1. Android phone
2. GSM modem
3. Arduino uno board
4. Light

Android phone:
The android app which we have designed to send an SMS in order to switch ON/OFF the light is imported onto the android phone.

GSM modem:
GSM modem consists of a SIM which receives the SMS sent by the android phone. After receiving the SMS, it then decodes it and forward it to the Arduino Uno board using serial data communication.

Arduino UNO:
Arduino Uno is a micro-controller based board. It is connected to the GSM network using GSM modem and receives the SMS sent by the android phone. After receiving the SMS it switches ON/OFF the light depending on the received SMS.

Light:
Light which is connected to the Arduino UNO is controlled by the commands which are sent by the micro-controller present on the Arduino Uno board.

HARDWARE REQUIREMENTS

- ➢ Arduino UNO board
- ➢ GSM Shield

SOFTWARE REQUIREMENTS

- JDK
- Android SDK,
- Eclipse
- Arduino-1.0.5

FUNCTIONAL REQUIREMENTS

- The control unit will have the ability to connect to the cellular network automatically.
- The Arduino GSM module will be able to receive text messages and will be able to parse and interpret (ASCII) text messages for instructions to be sent to the microcontroller.
- The microcontroller within the Arduino Uno board will issue its command to the electrical appliances through a simple control circuit.
- The Arduino Uno will control the electrical appliances and detect the status of the appliances to be relay back to the microcontroller.
- The microcontroller within the Arduino Uno should be able to send status messages back to the cellular phone through the cellular network.

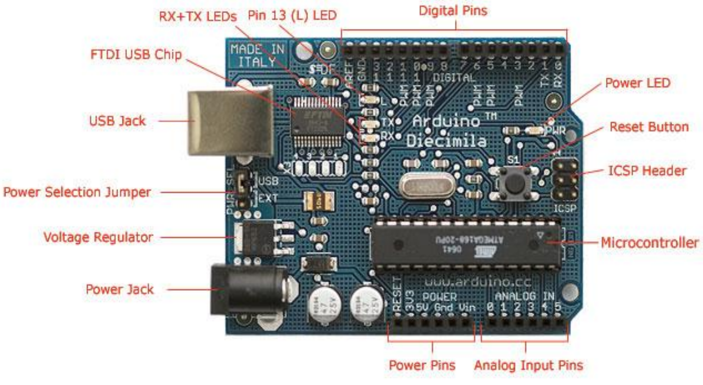
ARDUINO UNO

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead it features the ATmega16U2 programmed as a USB-to-Serial converter Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions.

Description

Microcontroller	ATmega328
Operating Voltage	5V
InputVoltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 Ma
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz



Photograph by SparkFun Electronics. Used under the Creative Commons Attribution Share-Alike 3.0 license.

Fig-2 Arduino uno board

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- 5V.This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.
- IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.

There are a couple of other pins on the board:

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

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available . The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of theATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the boot loader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

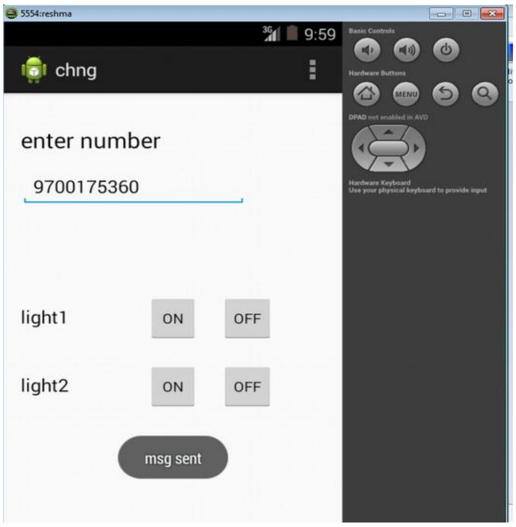
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



enter number

9700175360

Fig-5 Screen shot for successfull SMS sent

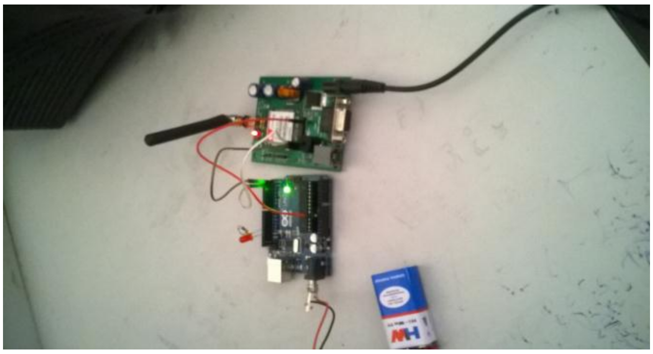


Fig-3 Boards setup

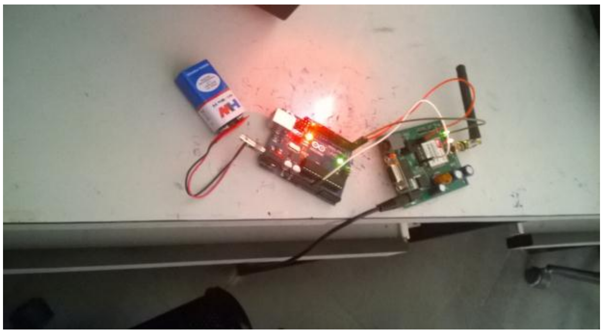


Fig-4 Blinking of LED

LIMITATIONS

Our project has certain limitations and a list of such is mentioned below

a. The receiver must reside in a location where a signal with sufficient strength can be received from a cellular phone network.

b. Only devices with electrical controlling input ports will be possible targets for control.

c. Operation of the controlling unit is only possible through a cell phone with SMS messaging capabilities.

d. The Controlling unit must be able to receive and decode SMS messages

FUTURE SCOPE

The future implications of the project are very great considering the amount of time and resources it saves. The project we have undertaken can be used as a reference or as a base for realizing a scheme to be implemented in other projects of greater level such as weather forecasting, temperature updates, device synchronization, etc.

The project itself can be modified to achieve a complete Home Automation System which will then create a platform for the user to interface between himself and his household

References

[1] 1. Sedra and Smith, Microelectronic Circuits, *fourth edition*, Oxford University Press, 1998

[2] 2. R.S. Sedha, 2002. A Text Book of Applied Electronics, S. Chand and Company Ltd., New Delhi

[3] 3. Theodore S. Rappaport, Wireless Communications, *second edition*, PHI. New Delhi

[4] 4. Draft EN (GSM 03.40) v6.0.0

[5] www.gsmworld.com

[6] www.google.com

[7] www.gnokii.org

[8] www.keil.com