

EMERGENCY CONTROLS IN CAR FOR SOME ESSENTIAL ENTITIES USING BLUETOOTH BY MOBILE APP

VIGNESH.M¹, NANDAGOPALS²

¹Student, Electronics and Communication Engineering, Nandha College of Technology, Tamil Nadu, India

²HOD, Electronics and Communication Engineering, Nandha College of Technology, Tamil Nadu, India

ABSTRACT

Now-a-days our cars have been stolen often or there may be a minor technical problems with it while running or Idle. So, to eradicate this problems we can create a Mobile App, by using that we can lock/ unlock control, indicator control, wiper control, engine on/off control and horn.

The intercommunication using here to communicate our App and other entities within the car is ARDUINO. Hereby connecting the Bluetooth Module (HC-05) with Arduino, because it will be helpful for us to connect our Mobile App with Arduino. In our Mobile App, there will be a password for each and every user so with help of that we can say each and every car will be configured with some special encrypted values in accordance with the password given by the user.

Arduino is a readymade embedded controller. It has number of in-build library files which reduces the number of coding lines, easy to understand and IDE software used to code in Arduino is an open source. Mobile App creating for this concern may create using App Inventor 2 in online also an open source, by sending encrypted values from app and decrypted values in the receiving side performs various required functions. Which it controls according to our code.

Hence by implementing this Technology to our cars may useful for us to manage when small essential entities while driving our car become failure or if our key is lost. Also another advantage is even if our phone lost or App gets uninstalled, there will be no issues because if we reinstall the App and by login with the same password which we are giving to our car initially, gets the control as usual

1. INTRODUCTION:

By implementing this Technology to our cars may useful for us to manage when small essential entities while driving our car become failure or if our key is lost. Also another advantage is even if our phone lost or App gets uninstalled, there will be no issues because if we reinstall the App and by login with the same password which we are giving to our car initially, gets the control as usual.

Though we designed a system to automate our car entities, not whole car in low cost. Hence, the future development would be the full autonomous driving but not to change constrain (i.e.) Low cost.

The arduino is coded according to the constrain also must to enable serial port to use bluetooth module (HC-05). Just by sending values from app to Arduino via (HC-05) and successive operations will be done. The main advantage in it, other objects can also be synchronize with it depends upon the user needs.

2. BLOCK DIAGRAM:

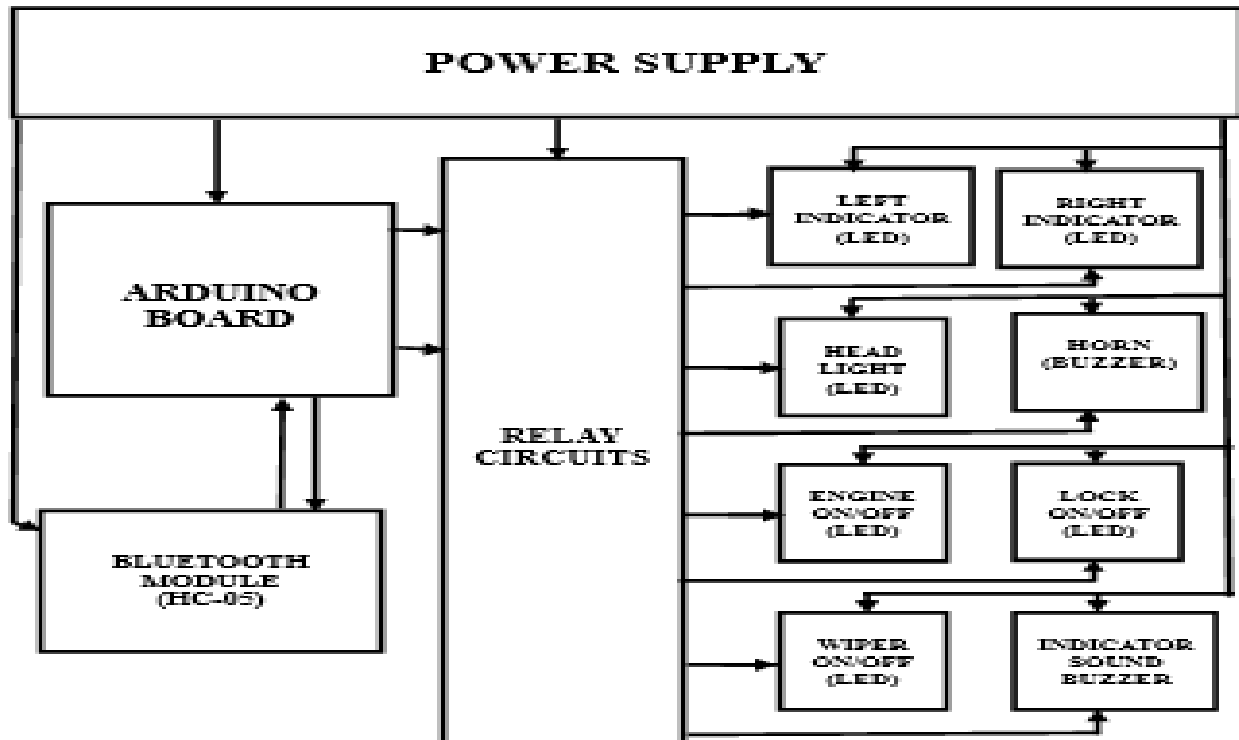


Fig -1 Block diagram

3. BASIC BLOCKS IN THIS PROJECT:

- Power Supply
- Bluetooth Module (HC-05)
- Arduino
- Relay Circuits
- Left Indicator
- Right Indicator
- Horn
- Head Light
- Lock/Unlock
- Engine On/Off
- Wiper On/Off
- Indicator Buzzer.

4. RELAY CIRCUITS:

A relay is an electrically operated switch. Current flowing through the coil of the relay creates a magnetic field which attracts a lever and changes the switch contacts.

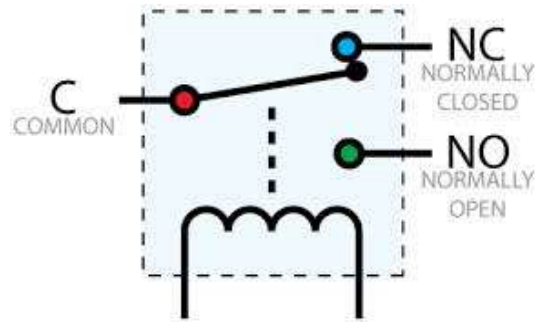


Fig -2 Relay board

As the general concept for relay that we have seen now. Here we are going to use it for toggle between on and off based upon the input given by the user via mobile app. It can be clearly understandable, while explaining the circuit fully which follows us.

5. POWER SUPPLY:

The essential thing for every circuit is power supply .In our circuit we are using power supply circuit, which capable of converting 230V/5A AC into 5V/0.7A DC. The important point to be noticed here is we are converting AC supply to DC, hence we using bridge rectifier (b7w10) for conversion. We get only pulsed DC from rectifier, so we need to filter for that a polarity capacitor of 100uf is used but we have to regulate 12V to 5V, hence again a capacitor of 10uf is put there to provide a constant DC supply for us.

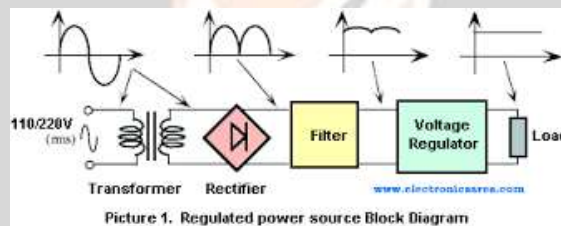


Fig -3 Power supply circuit

The above figure shows the general model of a power supply unit. If suitable we can also use the readymade SMPS of required voltage and amps. While we are applying it in real time car that time we can take it from the car battery itself, just our duty is only to regulate not to convert. In our circuit we are taking four divisions to various circuits as showed in the block diagram also it will be explained clearly in the full explanation of the circuit which follows us.

6. BLUETOOTH MODULE (HC-05):

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. As our project is in wireless we have to use any of the wireless protocols. Hence based upon our application the accessing range is small i.e. inside the car, so we can use Bluetooth module (HC-05), another advantage is that it supports financially too.



Fig -4 Bluetooth module

This image shows the HC-05 module. It has 5V, Ground pin and TX, RX. The 5V and Ground pins will be connected to the power supply. Simultaneously, TX pin connected to the RX of Arduino and RX connected to the TX of Arduino. The vital point here to make sure that Bluetooth module and Arduino are connected in common ground then only the serial communication will establish between them. This module is the intermediate between our Mobile app and Arduino.

Just by sending values from app to Arduino via (HC-05) and successive operations will be done.

7. ARDUINO UNO:

Arduino is a computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL).

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (*shields*) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

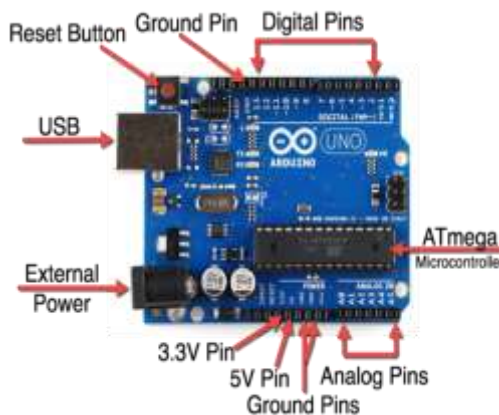


Fig -5 Arduino Module

The above figure shows the Arduino controller which has several advantages which we can discuss later. In our circuit Arduino only controls the required entities using a relay circuits depending upon the value received by it from user Mobile app via Bluetooth module (HC-05). The full explanation of its role is given in the full circuit explanation which comes below.

7. LEFT AND RIGHT INDICATORS AND ITS BUZZER:

In our project we are just controlling selective entities as we discussed in our abstract, it includes indicators also. By connecting the supply of indicators to relay and then the relay's control signal i.e. on/off toggle with Arduino GPIO. Hence, in block diagram it is just represented as a block. For a model we can indicate the indicator on/off using LED. As showed in the circuit diagram. We know that if indicator blinks a buzzer sound will be hearing so the same thing here we implemented here too by connecting it to a separate relay and operated it with coding parallel to indicator's relay.

8. HEAD LIGHT AND HORN:

In our project we are just controlling selective entities as we discussed in our abstract, it includes head light and horn also. By connecting the supply of head light to relay and then the relay's control signal i.e. on/off toggle with Arduino GPIO. Hence, in block diagram it is just represented as a block. For a model we can indicate the head

light on/off using LED. As showed in the circuit diagram. Likewise, by connecting the supply of horn to relay and then the relay's control signal i.e. on/off toggle with Arduino GPIO. Hence, in block diagram it is just represented as a block .For a model we can indicate the horn on/off using Buzzer. As showed in the circuit diagram.



Fig -6 LED and Buzzer images

9. ENGINE ON/OFF:

A starter (also self-starter, self) is an electric, pneumatic, hydraulic, an internal-combustion engine in case of very large engines or other device used for rotating an internal-combustion engine so as to initiate the engine's operation under its own power. In our project we are just controlling selective entities as we discussed in our abstract, it includes ignition on/off i.e. Engine on/off control also. By connecting the supply of self -starting motor to relay, because if self- starting motor runs, only we can able to produce ignition on engine and further it starts to move on. Then the relay's control signal i.e. on/off toggle with Arduino GPIO. Hence, in block diagram it is just represented as a block .For a model we can indicate the ignition on/off using LED. As showed in the circuit diagram. This image implies the normal self-motor used in cars to ignite. So, our role here is to connect the supply of this motor to relay.



Fig -6 Self-starter motor

10. LOCK/UNLOCK:

Power door locks (also known as electric door locks or central locking) allow the driver or front passenger to simultaneously lock or unlock all the doors of an automobile or truck, by pressing a button or flipping a switch. Power door locks were introduced on the luxury Scripps-Booth in 1914, but were not common on luxury cars until Packard reintroduced them in 1956. Nearly every car model today offers this feature as at least optional equipment.

Early systems locked and unlocked only the car doors. Many cars today also feature systems which can unlock such things as luggage compartment or fuel filler cap door. It is also common on modern cars for the locks to activate automatically when the car is put into gear or reaches a certain speed.



Fig -7 car lock

In our project we are just controlling selective entities as we discussed in our abstract, it includes lock/unlock control also. By connecting the supply of door lock/unlock circuit to relay, because if unlock the car, only we can able to produce ignition on engine and further it starts to move on. Then the relay's control signal i.e. on/off toggle with Arduino GPIO. Hence, in block diagram it is just represented as a block .For a model we can indicate the ignition on/off using LED. As showed in the circuit diagram.

11. WIPER ON/OFF:

A windscreen wiper or windshield wiper (American English) is a device used to remove rain, snow, ice and debris from a windscreen or windshield. A wiper generally consists of a metal arm, pivoting at one end and with a long rubber blade attached to the other. The arm is powered by a motor, often an electric motor, although pneumatic power is also used in some vehicles. The blade is swung back and forth over the glass, pushing water or other precipitation from its surface. The speed is normally adjustable, with several continuous speeds and often one or more "intermittent" settings. Most automobiles use two synchronized radial type arms, while many commercial vehicles use one or more pantograph arms.



Fig -8 wiper image

In our project we are just controlling selective entities as we discussed in our abstract, it includes wiper also. By connecting the supply of wiper to relay and then the relay's control signal i.e. on/off toggle with Arduino GPIO. Hence, in block diagram it is just represented as a block .For a model we can indicate the wiper on/off using LED. As showed in the circuit diagram.

12. MIT APP INVENTOR:

App Inventor for Android is an open-source web application maintained by the Massachusetts Institute of Technology (MIT). It allows newcomers to computer programming to create software applications for the Android operating system (OS). It uses a graphical interface, very similar to Scratch and the Star Logo TNG user interface, which allows users to drag-and-drop visual objects to create an application that can run on Android devices. In creating App Inventor, Google drew upon significant prior research in educational computing, as well as work done within Google on online development environments.^[1]

App Inventor and the projects on which it is based are informed by constructionist learning theories, which emphasizes that programming can be a vehicle for engaging powerful ideas through active learning. As such, it is part of an ongoing movement in computers and education that began with the work of Seymour Papert and the MIT Logo Group in the 1960s.



Fig -9 App inventor Logo

There are two types of windows, Designer window and Block window. The appearance of the windows are shown below,

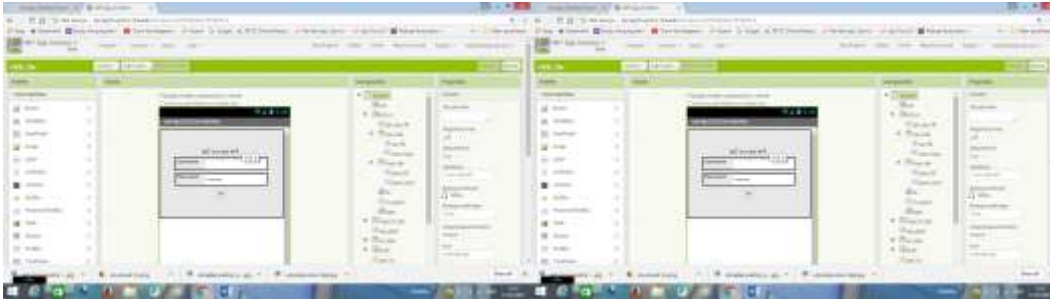


Fig -10 Designer and Block view of App



Fig -11 .apk file

In our project, we using an android application to control the entities of our car as we discussed in introduction. Hence, using the designer window we can do the outlook of the applications (i.e) buttons, check box, text, radio buttons, etc..., while we meet the requirement of any concern in the app, the designer window has blocks part in left side has all types of buttons, boxes, images etc..., that we can use.

Another important thing we have to see here is the Alignment options with this app creation. We can make a separate set of buttons, texts, etc..., into an isolated vertical or horizontal alignments, which will help us to make grouply the blocks to visible enable or disable. By the way our requirement will be differ from each person in accordance with application. If there may be any edition we can done on spot in online itself. As it was an open source we just need a mail to create an app using MIT APP INVENTOR.

The app which we are creating using MIT can be extracted from the drop down list box given at the top corner. It has two options that whether the .apk has to be downloaded to PC or we can scan its QR CODE directly from the screen as shown in the .apk file of app image. The complete app creation using MIT is based on the c language because it doesn't needs any scripting language to create app.

13. CIRCUIT DIAGRAM:

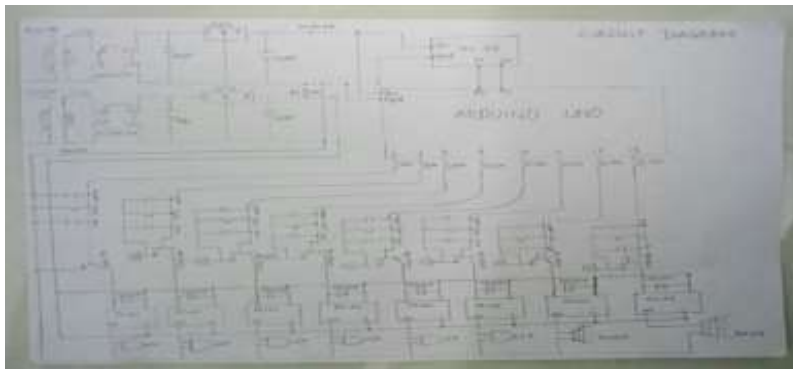


Fig -11 Circuit

14. FULL EXPLANATION:

As we discussed until now, our project is regarding wireless communication with embedded systems. By using an app via bluetooth we are going to control our car required entities such as, lock/unlock, engine, wiper, indicator, headlight on/off... in emergency situations. The circuit connections made according to our concern as shown above is successfully modeled with a toy car. The Arduino is coded according to the constrain also must to enable serial port to use bluetooth module (HC-05).

Here the relay circuit made by us in the sort of not to reverse any current. Therefore, we connecting a 10K serially to the each and every output from Arduino. Then it is parallelly connected to 2 capacitors and a resistor to ground in order to avoid aliasing. Among the 2 capacitors one is polarity capacitor with value of 10K and non-polarity capacitor with value of 104. The resistor of value 100K.

The image showed our project's relay design, done for a modeled car design,

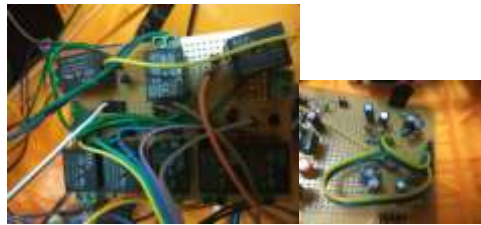


Fig -11 Relay Circuit

The power supply circuit design plays a vital role in every circuit this we discussed earlier as just an intro which implies that as per our project its requirement is 5V/0.6-1A. Hence, we showed in above picture we are using a regulator IC with two capacitors parallelly to ground in either sides of regulator in order to make a pulsed DC to a perfect DC. Before that we have to reduce the input 230V to 12V/1A using a stepdown transformer as we showed in the figure. After that we need a DC supply not an AC. So, bridge rectifier is used to convert AC supply to a Pulsed DC supply.



Fig -11 Transformer, Bridge rectifier, Regulator IC

Then, we need only 5V but rectifier lets out up to 12V so here we meet the need of regulator IC 7805 to regulate the voltage from 12V to 5V. From the above made connections we can get a constant power supply that we need which also shown in the above circuit diagram. One important thing we need to put two separate power supplies for our modeled project because it needs to drive over 8 relays hence it needs high amp rating but we cannot able to increase the amp rating because Arduino input supply shouldn't be higher than 0.7A. This concern is corrected if we design two separate power supplies and to connect the grounds together.

15. CONCLUSION:

As from above declarations and definition of individual components and also its working with Arduino controller in my project have been seen clearly, so. According to my point of view it is clear that if we implement my project the user's car becomes secured and automated through App.

The another main advantage in it, but it will depend upon user's requirement if he/she needs, we can control the lock/ unlock control, indicator control, wiper control, engine on/off control and horn, etc., of the car using our mobile phone itself. Also other objects can also be synchronize with it depends upon the user needs.

Hence, I concluding it with that probably I have covered up to above 6 entities in the module. So, it will be so compact, fixed, accurate good performance. Out of all points one of the important point to note is that though there are more advanced things in car already exists in foreign countries, from our project it's not like we are expecting too advanced as existence but somewhat automated in LOW COST.

The more we go advanced the more we have to pay instead of that up to this much of control as in my project we can pay only less amount to update or to add this feature in our cars.

16. FUTURE DEVELOPMENT:

Though we designed a system to automate our car entities, not whole car in low cost. Hence, the future development would be the full autonomous driving but not to change constrain (i.e.) Low cost.

17. REFERENCES

[1]. An Embedded Systems Course for Engineering Students Using Open-Source Platforms In Wireless Scenarios
M. C. Rodriguez-Sanchez, Angel Torrado-Carvajal, *Student Member, IEEE*, Joaquin Vaquero, Susana Borromeo, *Member, IEEE*, and Juan A. Hernandez-Tamames.

