

An Autonomous Vehicle Based on Image Recognition Using Android Platform

K. Rajasekhar Yadava* and P. Eswaran*

Abstract: This paper proposes the design of an autonomous vehicle performing lane detection; overtake obstacles and parking task in the path specified. It is implemented using android platform that uses computer vision (CV) techniques along with onboard sensors. Autonomous navigation android application was developed to realize the image processing and decision making using a smart phone and also responsible for wirelessly transmitting instructions to Arduino controller which controls the autonomous vehicle. The GUI of android environment presents the CV processed view of the task performed by the autonomous vehicle. Hardware implementation of the proposed autonomous vehicle uses the Arduino mega for interfaces the sensors and communicates to smart android phone application through Bluetooth. The experiment result shows good functionality and agreement with the assigned task, user-friendly application, and also has a faster computational response.

Keywords: Android Smart Phone application, android application for autonomous vehicle, Lane following vehicle, RC Car, Open CV, CARdunio

1. INTRODUCTION

The modern vehicle features increase not only the driver's comfort by providing automated features, but these functions are also responsible for the reduction of accidents. In common with these functions is that they depend heavily on perceived autonomous miniature vehicles [1] need to demonstrate the reliable capability to follow their lane, to overtake vehicles blocking on their lane, to handle intersections safely according to the right-of-way traffic rule, and to park the vehicle at a sideways parking strips as fast as possible.

The task involved in the implementation of the design is to find the suitable sensors that exhibit the characteristics required for interfacing it to the smart phone and easily allows the successful signal condition the Arduino Mega. This system is to overcome those problems by selecting android as the optimum solution. Utilizing an Android smart device, that handles the decision making and to combine the functionalities of the processor and the camera into one component as well as collect the data from the on board sensors. In addition to motion control, target detection, path planning ability of mobile robot. It still needs to constantly interact with the unknown environment, gradually reaching a deep understanding of the driving environment, which leads to as for and more reliable navigation. Due to the uncertainty and diversity of the environment, there is a need both for human and robots to learn from the previous exploration and gain knowledge about the environment. Therefore, constructing a reasonable learning scheme is significant for mobile robot's autonomous driving in the unknown environment. Lane recognition for the mobile robot in an unknown environment is a very challenging problem. It is difficult to solve this problem by traditional pattern recognition method, since there are a variety of uncertainties in the environment.

Christen Berger describes the features of self-driving vehicle [2]. The self-driving is the process of capturing the images through web camera that provides machine vision capabilities to the system as well as a plethora of sensors and microcontrollers which are placed as an antenna at the top of the vehicle. A single board Linux computer (ODROID-U3) is used to decision making on the track. This existing system is having encounter some problems are like ugly appearance, excessive software characterized by a tower looking structure. Because of that, a hole had to be curved off the vehicle's default enclosure. Fred W.

* Department of Electronics and Communication Engineering, SRM University, Chennai, Tamilnadu, India, 603 203

Rauskolb describes problems arising in urban and suburban terrain to detecting the lane were present on some models on [3]. AUTOSAR does not provide specific support for sensor/actuator-based autonomous vehicle development [4]. B. Rumpe introduces artificial intelligence capable of driving autonomously in the urban situations. The process itself includes agile concepts, like a test first approach, continuous integration of all software modules and a reliable release and configuration management assisted by software tools in integrated development environments [5]. Haun wang, Mingwu Ren describes the method uses multiple threshold segmentation instead of single threshold segmentation and straight and curve lane markers are directly extracted in Run- Length accumulation (RLA) images [6]. A lane detection method screen data using the vanishing point according to the perspective feature of the camera. The line data is obtained after Canny and Hough transform of the raw image [7].

Proposed Autonomous Vehicle

An autonomous vehicle is based on a remote controlled (RC) 1/10th scale of the original car, which we basically change it to add some features to the car by initially replacing its electronic speed controller (ESC), DC servo motor, microcontroller (Arduino Mega), a gyroscope and capable terminals among other components.

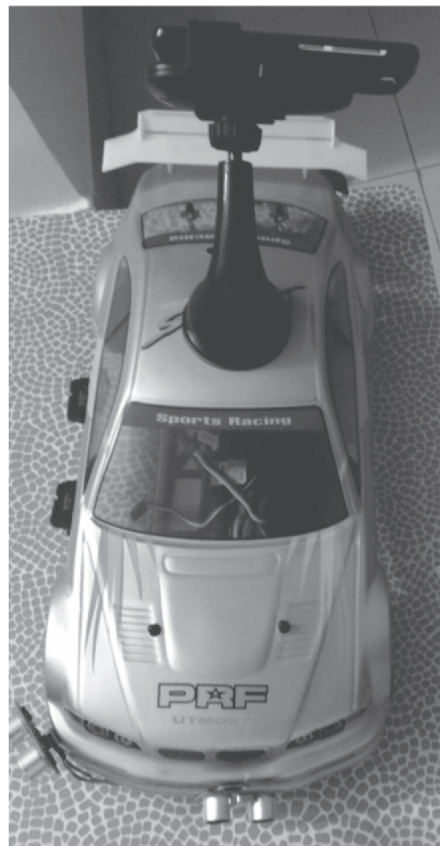


Figure 1: An Autonomous Vehicle

Speed encoder is fitted on right side back wheel and the front bumper a 9 DOF IMU. This includes an accelerometer, a gyroscope, a magnetometer as well as an ATmega328p, which runs fusion sensor firmware which in turn calculates the displacement in 3 axis and transmits it via the serial port. There are three infrared and three ultrasonic sensors as well as LED'S, that serves as flash and stop lights. There are also two infrared arrays in the front of the car, to detect dash line between the lanes through detecting the colors of the street lane. The control system block diagram of an autonomous vehicle is shown in Figure 2.

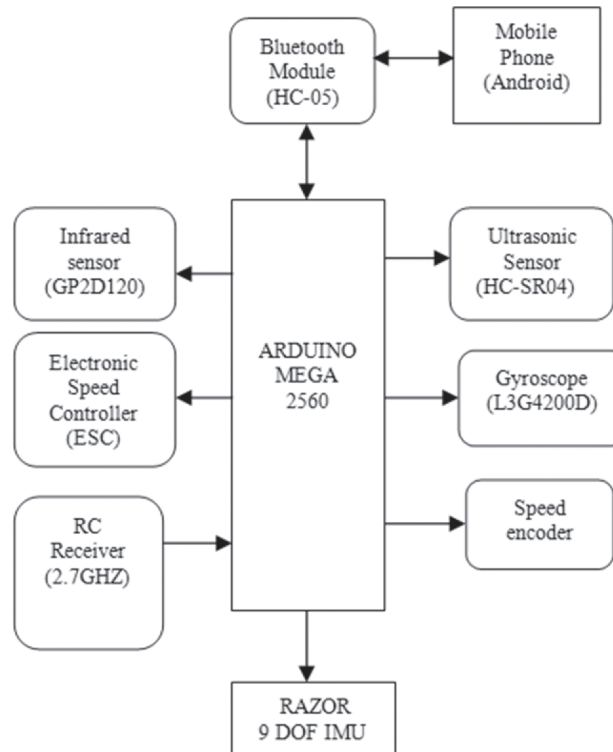


Figure 2: Block Diagram of Autonomous Vehicle

Control Module

The control module consists of an inbuilt AT mega 2560 microcontroller. An ATmega 2560 on the board channels the serial communication on USB and appears as a virtual port to software on the computer. Arduino Mega board has 16 analog pins and 54 digital Input and Output pins. It also has an analog to digital converter which converts the analog value obtained from the output of the sensor to a digital value and transfers the data to the smart phone via Bluetooth.

Sensor Module Interface

The sensor module consists of ultrasonic distance; sharp infrared sensors integrated into it. These sensors sense the distance of the object and produce a corresponding analog voltage output. This analog voltage output is fed to the analog pins of the control module which is then converted to a digital value by the analog to digital converter (ADC). The distance sensor in the sensors module needs to sense the data from the object passed through the readings.

Image Processing

Image processing is one of the techniques to follow the street lane by utilizing the Opencv [8]. This enabled to capture the real environment by mounting the android mobile phone at the top of RC car to test the track by using various machine vision algorithms. Then it could visualize the various transformations on phone's screen by on the spot debugging techniques like line detection and Canny edge detection and Hough line transform to get a bird- eye view of the video stream, captured from the smart phone's camera.

Path of Autonomous Vehicle\

Lane following is the features of our autonomous vehicle and it using image processing conducted through the OpenCV library, to make the car has capable of driving within the appropriate street lane. It used

OpenCV to define what a valid street lane is and as long as we are able to find them, we make sure stay in the middle of the lane. Model street land for prototype model testing is shown in Figure 3.

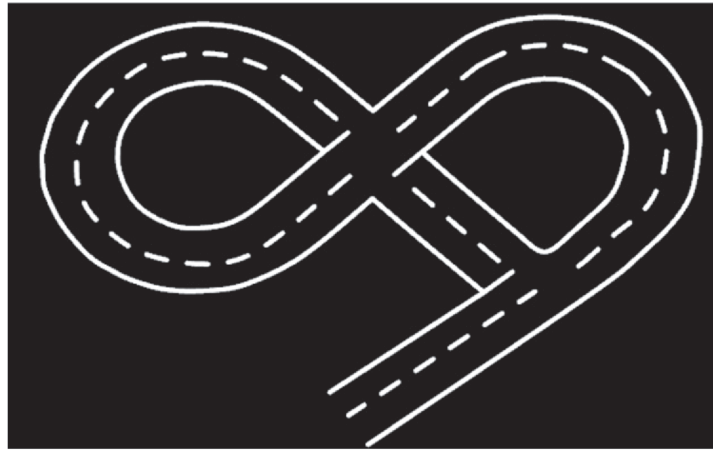


Figure 3: Model of Following Street Lane

Implementation of Autonomous Vehicle

Android Application Implementation in Android Studio: The android application was developed by using Android Studio and named as CARduino with various features to control manually and connects via Bluetooth to other devices. It was tested with Jelly Bean, Kitkat, and Lollipop of android platform and found working successfully. It also includes an image of the OpenDaVinci's environment, with our image processor and driving logic features (lane following, overtaking and parking) ready to be compiled and the CARduino application environment is as shown in Figure 4.

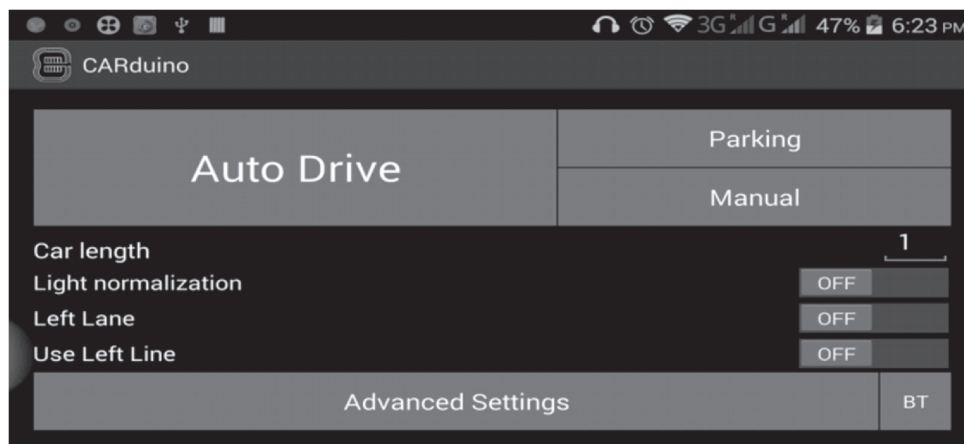


Figure 4: Android Application Environment

Firmware for Autonomous Vehicle

Execution flow of the system firmware is shown in Figure 5. Once autonomous starts moving forward, it continues until reaching its destination, if any obstacles detected which commands to overtake the path and continue to follow the designated lane. Execution flow of the autonomous vehicle is shown in Figure 5.

Execution Algorithm

Step 1. Start the vehicle.

Step 2. Checking the positions of the lane

Step 3. Moving forward on the lane.

Step 4. If any object detected in the right lane, then collect the data from the infrared sensor and the ultrasonic sensor.

Step 5. Check object distance then initiate overtaking.

Step 6. Turn to left lane then the object is passed the turn to the right lane.

Step 7. If no object is in right lane then moving forward

Step 8. Then its reached destination

Step 9. Stop the vehicle.

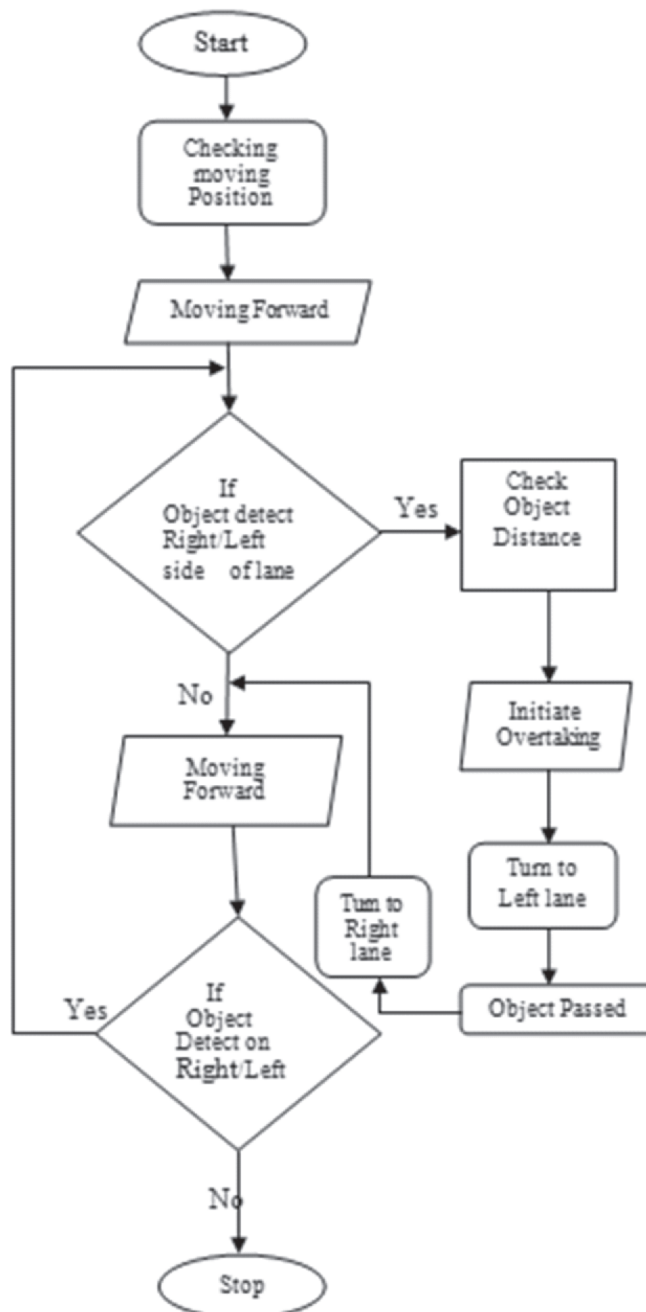


Figure 5: Execution Flow of the Autonomous Vehicle

Results and Discussion

GUI of Android Environment

To follow the street lane one android mobile phone is placed on the top of the vehicle that having the application of CARduino, and then it should have the feature of image processing and the decision making using acquired image from camera in the mobile phone. The processed image shown in Figure 6 shows the detected straight lines in black and white colors on the lane.

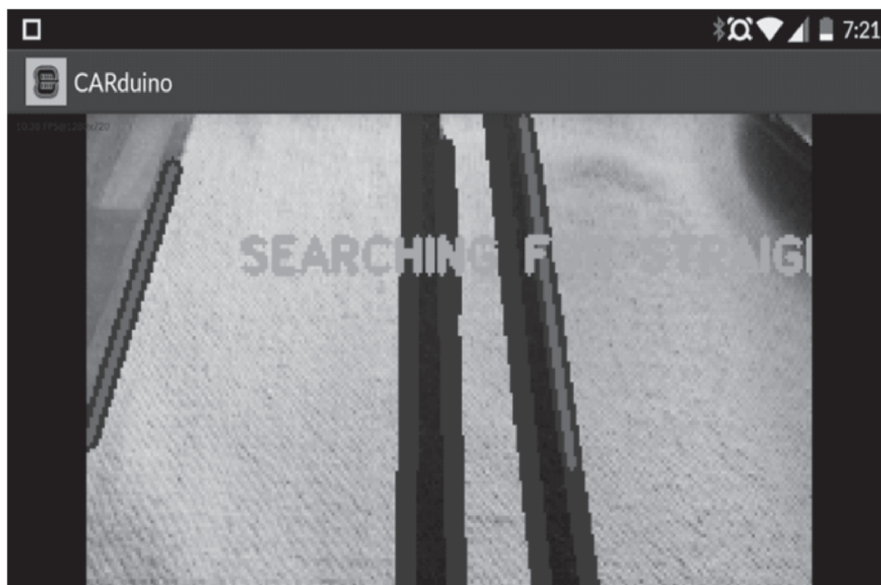


Figure 6: Detecting Straight Lines on the Street Lane

Lane Following

Lane following is one of the features of our autonomous vehicle. The autonomous vehicle is following the street lane by using image recognition on the android smart phone are shown in below Figure 7.



Figure 7: Lane Following

Overtaking Obstacle

Overtaking obstacles which are blocking in the car's path is one of the most exciting and challenging features of the Autonomous vehicle, and then it combines machine vision, with interpreting the inherently of imperfect real world data taken from the sensors. While this feature worked perfectly in the Open DaVinci simulations, with the vehicle being able to overtake many different scenarios, then it was very interesting to see how the dramatically things differed in reality. Particularly, the two main hurdles were for the vehicle to be constantly "aware" of its current relatively that position compared to the obstacle and the fact that due to the small size of our testing track, after overtaking an obstacle found on a curve, therefore there was no time to realign with the appropriate street lane with the addition of an infrared array, below the car. This enabled to know when are switching lanes, with the infrared sensors sending a signal when they happen to detect the middle white dashed line of the track and to adopt processing time constraints based on image processing, both for detecting the presence of an obstacle and that recognizing when exactly switched lane. These autonomous vehicle overtaking obstacles are shown in Figure 8.

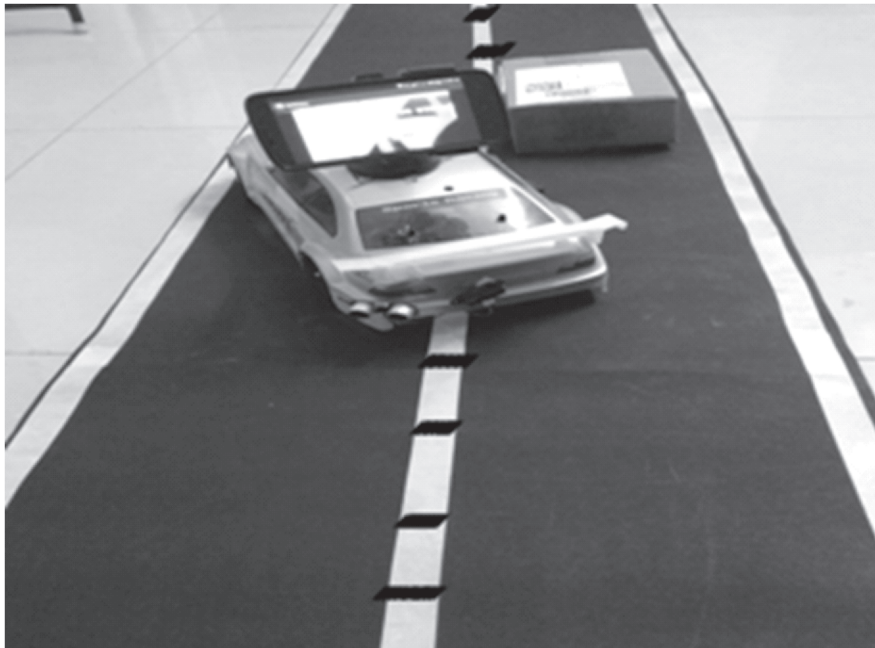


Figure 8: Overtaking Obstacles in its Path

Parking

Parking was another feature that include in our vehicle's functionality. It was challenging due to, that were completely depends on sensor data more particularly that of the gyroscope sensor, the speed encoder, the infrared and the ultrasonic distance sensors are to determine the surroundings to park the car. The amount of noise got from the measurements was not being concern and then were challenged by the inability of the distance sensors to function properly when it facing an obstacle at step angles because the ultrasonic wave or the infrared wave beam bounces away and never returns backs are to be collected.

Thus even if we are close to an obstacle, if the sensor is facing it from a high angle, then it is not detected. Unfortunately this is an inconsistency on between the both simulator (where everything functions ideally) and the real life behavior of the sensors, therefore forced to change the sensor locations to detect that data and try many things on the real environment, until it would identify the ideal parameters for the algorithms to work and planning to improve this performance and features increase the awareness of its car's position, as well as researching ways to enhance that procedure with the machine vision.



Figure 9: Parking of Vehicle

Manual Control

An autonomous vehicle was controlled by manually through the android application having feature to communicate via Bluetooth on android smart phone and Bluetooth module connected on Arduino Mega as shown below

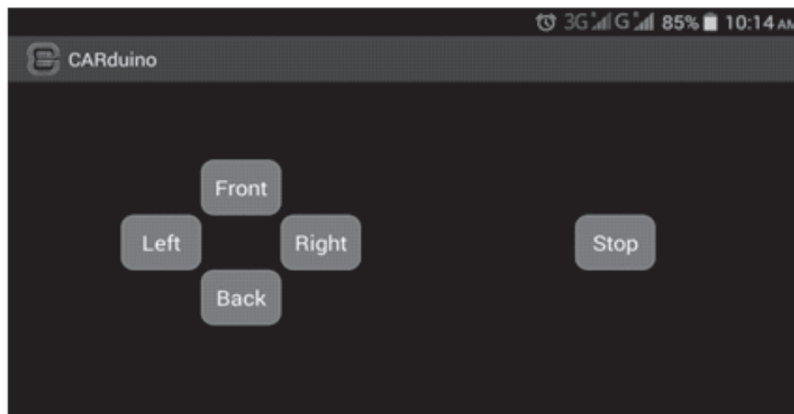


Figure 10 Manual Control

Conclusion

An autonomous vehicle using on image recognition on the android application was developed. The proposed application is not only to detecting the street lanes but also do decision making. This android application communicated through Arduino Mega to drive the autonomous vehicle. The task proposed for the autonomous vehicle was experimentally tested successfully.

Acknowledgment

The authors would like to thank the Department of Electronics and Communication Engineering at SRM University for providing the facility to carry out this work.

References

1. Michel Chaudron, Rogardt Heldal, Olaf Landsiedel, 2013, “*Model-based, Composable Simulation for the Development of Autonomous Miniature Vehicles*” Journal of Software Engineering for Robotics, PP. 23-30.
2. Christian Berger, 2014, “*Self-Driving Miniature Cars to a Standardized Experiment Platform*”, Journal of Software Engineering for Robotics, pp 63-79.
3. F. W. Rauskolb, K. Berger, C. Lipski, M. Magnor, K. Cornelsen, J. Effertz, T. Form, F. Graefe, S. Ohl, W. Schumacher, 2008, “*Caroline: An Autonomously Driving Vehicle for Urban Environments*”, Journal of Field Robotics, Vol. 25, No. 9, pp. 674–724.
4. C. Berger, 2012, “*COTS-Architecture with a Real-Time OS for a Self-Driving Miniature Vehicle*”, in Proceedings of the SAFECOMP 2012 Workshops, LNCS 7613. Magdeburg, Germany: Springer-Verlag Berlin Heidelberg, pp. 180–189.
5. C. Berger and B. Rumpe, “*Engineering Autonomous Driving Software*,” in Experience from the DARPA Urban Challenge, Eds. London, UK: Springer Verlag, pp. 243–271.
6. Haun wang, Mingwu Ren, 2011, “*Lane Markers Detection based on Consecutive Threshold Segmentation*” Journal of Information and Computing Science, Vol. 6, No. 3, pp. 207-212.
7. B.P. Gerkey, R.T. Vaughan, and A. Howard, 2009, “*The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems*,” in Proceedings of the 11th International Conference on Advanced Robotics, pp. 317–323.
8. WANG JINGYU, DUAN JIANMIN, 14-17 July 2013, “*LANE DETECTION ALGORITHM USING VANISHING POINT*”, International Conference on Machine Learning and Cybernetics, PP. 256-261.
9. P Blasco, F. Diaz-del Rio, M. C. RomeroTernero, D. Cagigas Mun iz, and S. Vicente-Diaz, Jun. 2012, “*Robotics software frameworks for multi-agent robotic systems development*,” Robotics and Autonomous Systems, Vol. 60, No. 6, pp. 803–821. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S09218890120003222.1.3>
10. <https://www.arduino.cc/en/Main/arduinoBoardMega.pdf> :Accessed on:12-3016

