Learn and create an autonomous car model

Nguyen Thien Quang

Abstract: The development of science and technology in recent years have brought a new era for human, many application products and vehicles have been manufactured and implemented in every aspect of human life. These vehicles with the implementation and integration of cutting-edge technologies that can remote control and operate autonomously helped with some basic tasks and assisted a vast of human work such as delivery, transport, exploring new areas, home chores, etc…One of the vehicles that has been considering and attending to develop is the autonomous car which can be operated to complete tasks with the restriction or no human control. To keep up with this trend and provide an opportunity for student to learn and approach automation vehicles, the purpose of this subject is to teach students to create an autonomous car which can do a simple task is evading obstacles. The project is built from scratch and do not use Arduino ecosystem for helping students have a decent knowledge of the way the car is created. The project will use wireless radio communication to control the car drives in directions commanded by a wireless remote control and can change mode to detect and avoid obstacles automatically.

Keyworks: Autonomous car, vehicles, automation vehices, wireless radio communication.

# Introduction

Autonomous cars, also known as self-driving cars, are vehicles that are capable of sensing their environment and navigating with restrict or without human input. These cars use a combination of sensors, cameras, and algorithms to detect and interpret the world around them, making real-time decisions about how to move safely and efficiently through traffic. Autonomous cars have the potential to revolutionize the transportation industry by reducing traffic congestion, improving road safety, and providing greater mobility to people who are unable to drive themselves. With that potential, buiding a simple autonomous car model will help students that interested about automation vehicles experience and have a decent knowledges about this field.

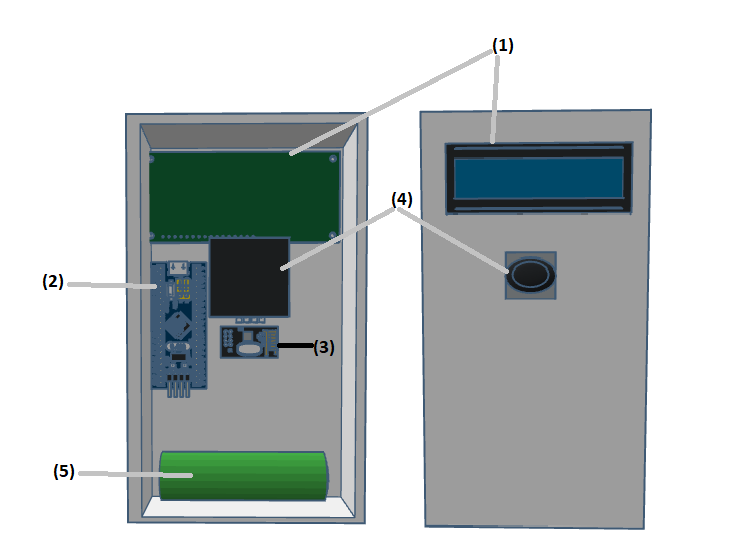
Projects about autonomous car models have been created in many places with numerous online tutorials may be mentioned as Science Buddies website organization with Miniature Seft-Driving car using Arduino UNO board to control and HC-SR04 sonicwave sensor to detect obstacles, it can safely navigate a model roadway[1] or Bluetooth Controlled Arduino Based Autonomous Car of students from Frankfurt University of Applied Sciences that can control the car moving in directions via android app and using Bluetooth to communicate with[2] or Arduino Based, Bluetooth controlled RC car by Souvik Paul professor that can control the RC car moving in directions using Arduino Uno and communicate via Bluetooth[3] or Arduino Based Obstacle Avoiding Robot Car by Michael Klements, the car uses a servo mounted ultrasonic sensor to detect objects in front of and on either side of the car and an L293D DC motor driver shield to drive four geared motors, one on each wheel. An Arduino Uno underneath the motor driver controls the motor shield, ultrasonic sensor and the servo[4]. The downside of these projects are they used available component parts for instance, car’s frames that have sold commonly and use the Arduino ecosystem which has available code for implementation, this covers the process of making the model and omits how to develop code for the car, so students will not experience and encounter many aspects when building project. Another downside is the projects with auto avoid obstacles didn’t have a remote control to handle when the car got problems, this is inconvenient because when the car is tested and if there are some problems with the algorithm that implemented into the car, it may operate in the wrong way and cause damage to itself, so the remote is necessary to take control by hand when car’s operation problems happen. This project will be built from scratch to assure students will learn about creating autonomous cars manually and will create a premise to study other high-level relate models in the future. The project will use stm32 microcontrollers to control the car and it will operate in 2 modes, first mode is the mode that the car can be controlled by a wireless remote control using a joystick to drive and the second mode is the car could drive autonomously. The car transfers the mode by using the button to switch between 2 modes.

# 2 Methodology

## 2.1 Overview

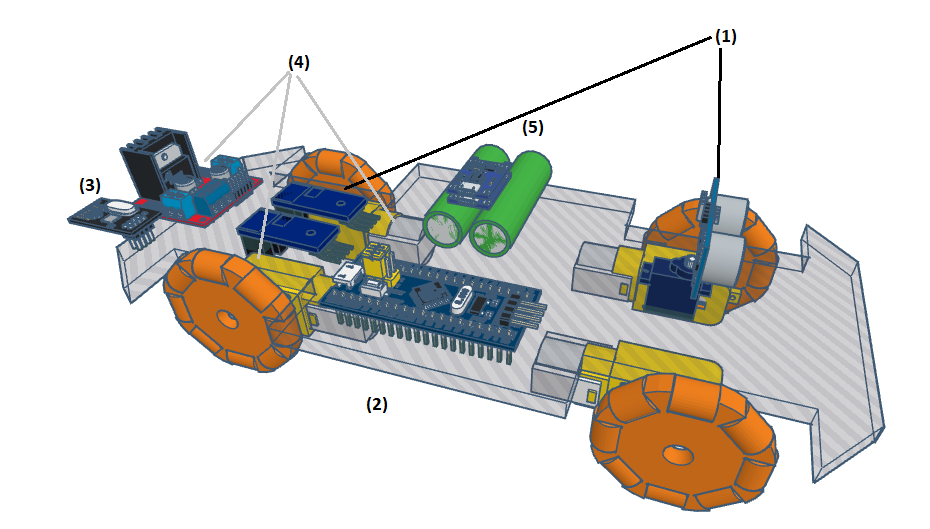
The model is separated in 2 parts, one is wireless radio remote control and the other is the car that receive data from the remote.

Wireless remote control has 5 main parts: (1) Display part to display the state and the direction of the car (2) Controller part to control the remote (3) Transmitting part to transmit data from the control block. (4) Input part to get input from user. (5) Power source part to supply power for remote.



Picture 1: Wireless remote control design

The car has 5 main parts: (1) input part is getting input data from the outside and transmit to control block. (2) Controller part to get input from sensor and receiver part and control the car.(3) Receiver signal part to receive data from remote. (4) Actuator part to make the car moves in different direction.(5) Power source part to suppy power for the car. The model is represented in picture 2.



Picture 2: Autonomous car model design.

**2.2 Harware/ software/ mechanical part design**

### 2.2.1 Design wireless remote control hardware

**Select display part:** for simplicity, the display part will use LCD1602 and module I2C to print the state and values to the user.

**Select controller part:** the control part is used for remote control is STM32f103c8t6 microcontroller board[10] with clock rate frequency range from 8Mhz to 75Mhz. Operation voltage from 2.2 to 3.6v and supply many peripherals, a cheap, powerful, and fully supported by ST company and have a large community.[12]

**Select transmitting part:** there are many radio communication modules that are published in the market with the variety of ranges, data transfer speeds, and power consumption. Table 1 below lists some communication module’s characteristics that can be chosen for the transmitting part.

Table 1: Communication module’s characteristics

|  |  |  |  |
| --- | --- | --- | --- |
| Characteristic | Lora | nRF24L01 | RF 433Mhz |
| Distance | ~1000m | ~100m | ~2500m |
| Bandwith | Medium | High | Low |
| Power consumption | Low | Low | Low |
| Cost | High | Low | Low |

To minimize the latency between the controller part and the transmitter and between the remote and the car, nRF24L01 is the best choice for transmitting part with medium range distance, low cost, power consumption and the high bandwidth. The nRF24L01 is a single chip 2.4GHz transceiver with an embedded baseband protocol engine (Enhanced ShockBurst™), designed for ultra low power wireless applications. The nRF24L01 is designed for operation in the world wide ISM frequency band at 2.400 - 2.4835GHz. The protocol to communicate with module is SPI protocol[5].

**Select input part:** the analog joystick will be used to get the input from the user. The input is read by controller part.

**Select power source part:** 18650 battery is used to supply power to the wireless remote control with capacity around 1200mAh - 3500mAh and nominal voltage is 3.7V.

### 2.2.2 Design remote control hardware

**Select input part:** There are 2 input data that must be read. The first one is the input that the sensor will read and send the value to the controller part to notice if is there an obstacle in the front way. The second is the input of the encoder to inform the speed of the motor. For the sensor to detect obstacle objects, HC-SR04 ultrasonic wave sensor will be used and combined with an SG90 servo motor to assist the sensor detect the surface widely. The encoders that will be used are TH119 encoders and encoder wheels with low cost, fairly precise, and easy to mount with the shaft of the motor.

**Select controller part:** the control part is used for the car is STM32f401CCU6 microcontroller board[9] with clock rate frequency range from 8Mhz to 84Mhz. Operation voltage from 2.2 to 3.6v and supply many peripherals, a cheap, powerful, and fully supported by ST company and have a large community[11].

**Select received signal part:** nRF24L01 module is used to compatible with transmitter part of the wireless remote controller.

**Select actuator part:** 2 DC 180 motors with high speed, wide range operation voltage and small size combine with L298 H-bridge module are suitable with the actuator part.

**Select power source part:** Two 18650 batteries are used to supply power to the actuator part with voltage range from 6v to 8,4v, this voltage will be rectified through 7805 regulator IC to regulate this voltage into 5v and supply to the controller part and sensors.

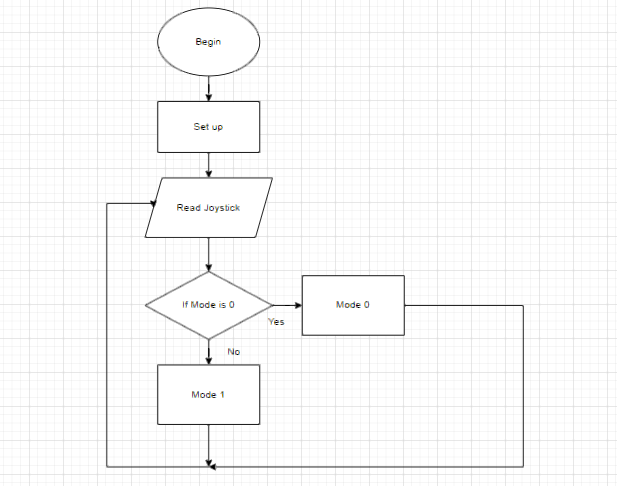
### 2.2.3 Software description for the wireless remote control and the car.

The tools are used to build software for remote and the car are STM32CubeMX and Keil vision 5. The language is used to design the sofware is C language with high speed, efficiency and flexibility.



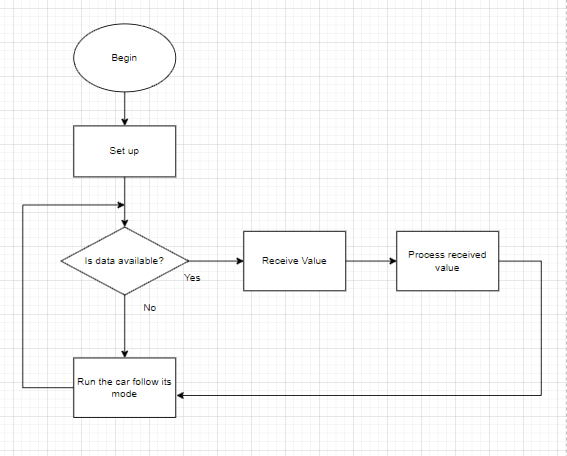
Picture 3: Software toolchains

By the software for microcontroller, the program will begin in mode 0, STM32 microcontroller will read the value of the analog joystick and the button in the joystick every 80ms, if the analog signal changed in a value that is different from the value is set in initial when the joystick is in the center or the button is pushed, this signal will be transfered into digital signal and convert it into a datapackage string with 2 part, the first 2 elements are command and the next 4 emelent are analog signal values and sent to the nRF24L01 module[6][7], then this data is sent to the car. The flowchart designing software for wireless remote control is represented below.



Picture 4: Flowchart for wireless remote control.

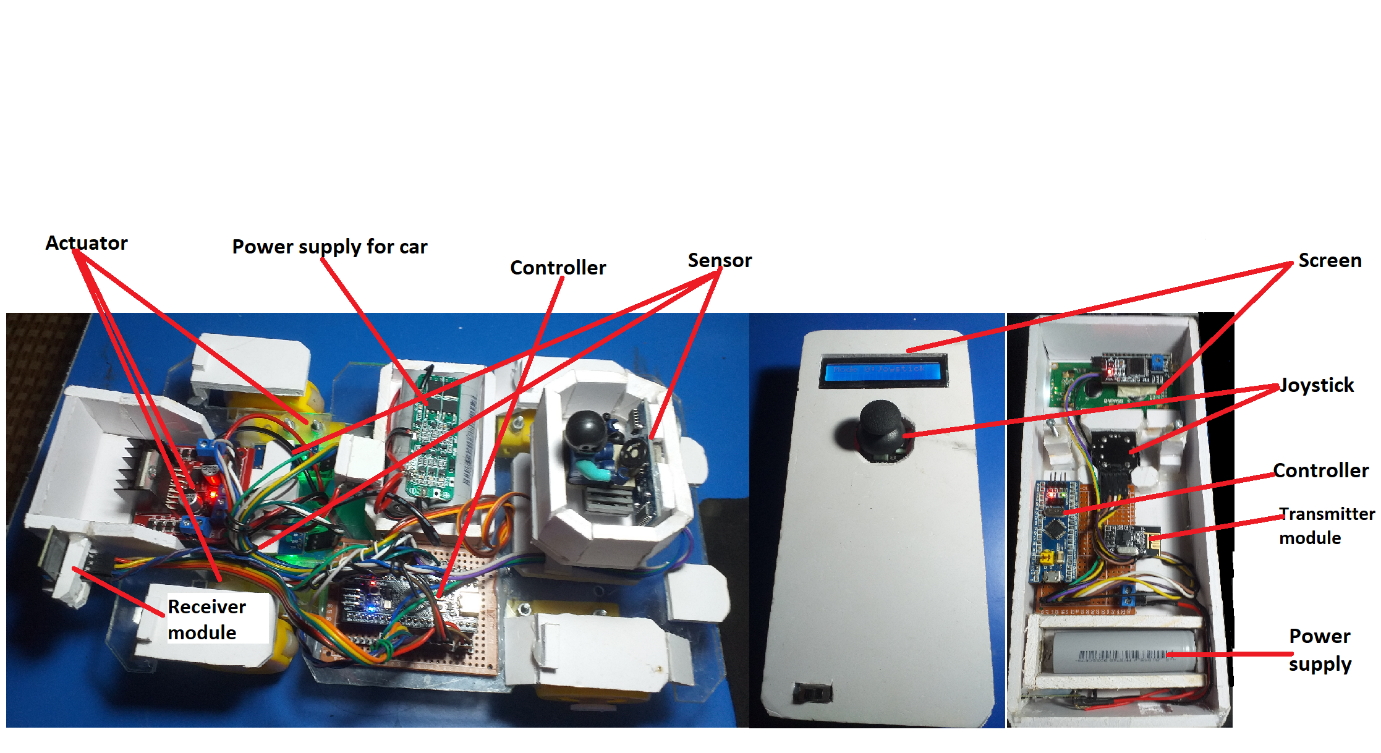
Initially the software for car is in mode 0 and will be waited the signal from the remote every 80ms by reading the state of the nRF24L01 module[8], if the car is in mode 0 and data is ready in nRF24L01 module, then it will be read and transfer the data package. With the command in the data package correspond to the actions are set up, the car will perform move in up, down, left, right direction follow by the commands. If the car is in mode 1, the car will be set up to operate automatically, it will detect and obstacle objects until the remote send the signal that give the command change mode, the car will stop and turn into mode 0. The flowchart designing software for car is represented below.



Picture 5: Flowchart for autonomous car.

# 3. Result and discussion

The created model is represent in picture 6. The frame of the car is made of acrylic sheet, main parts of the car is mounted in the frame, the parts are reasonably arranged and the decorated to made the car become good-looking. The wireless remote control is covered by form sheet and have a lid to protect inside components.



Picture 6: Autonomous car and wireless remote control model.

The result when testing the car is good, the car is able to operate in 2 modes. In mode 0 the car responded the signal from the remote almost instantly, the speed of the car was fast and stable. Wireless remote controller is designed in a simple and easy way to control the car’s direction, remote can control the car with a long-range, it reachs around 80m, when the remote is out of range the car stopped and communication is lost. In mode 1 the car could run, detect and evade some obstacles but not all obstacles are detected. The switching between 2 modes is fast and there is no latency when changing modes.

Table 2: Functional Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Yes | No | Not stable |
| Move up | X |  |  |
| Move back | X |  |  |
| Turn left | X |  |  |
| Turn right | X |  |  |
| Control mode | X |  |  |
| Auto mode | X |  |  |

Table 3:Car’s Speed Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Analog stick position from the center | Near | Medium | Far |
| Speed (RPM) | Move up | ~500 | ~1200 | ~1800 |
| Move down | ~500 | ~1200 | ~1800 |

Table 4: Communication Range Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Range | <10m | 25m | 50m | 75m | 100m |
| Signal | Good | Good | Good | Not stable | Disconnected |
| Latency | Very Low | Very Low | Low | Low | Disconnected |

Table 5: Avoid Obstacles Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Obstacle Magnitude | Very small | Small | Medium | Large |
| Detect obstacles | No | Not stable | Not stable | Yes |

**Discussion:** The result has been acquired but sometimes the sonic wave sensor module got some distorted signal which caused the car to malfunction and operated in the wrong way, the solution considered to solve this problem is to implement Kalman filter to rectify the input signal from the sensor. The motors in the car can run with stable and high speed but the direction when moving forward was not a straight line but a curve line, it noticed that one motor is stronger than the other. PID algorithm was implemented into the software to balance the speed of the motor but it is still not optimized to get the car to move in a straight line completely. So the optimization for the PID algorithm could be able to consider to get the car to move better forward. The increase in capacity of the power supply for car is also considered because the operation time of the car is still short when testing (about 20 minutes), so increasing the capacity will let the car operate for a longer time.

# 4. Conclusion

The car model has been successfully designed, implemented, and tested with a nice result. The model can be used to study and develop more high-lever. The car is able to operate in 2 modes with low latency when sending commands to the car via a wireless remote controller and runs with speed changeably. Besides that, the car with one sonic wave sensor could detect obstacles but sometimes it could fail when the environment changes such as in a rugged surface or encounter thin and small objects. Solutions for that downside is applying more sonic wave sensor to get the surface properly or using esp-32 cam with OpenCV to better detect and classify objects and minimize the affection of the surfaces and environments.

Acknowledgment

This part is dedicated to thanks to master Tran Le Chung Chanh, who has provided friendly, helpful guidance about the process of building autonomous car over the past few weeks.

References

[1] sciencebuddies, 2015. Build a Miniature Self-Driving Car. <https://www.sciencebuddies.org/science-fair-projects/project-ideas/Robotics_p042/robotics/arduino-self-driving-car>. 8/6/2023.

[2] Md Mahmud Hasan, Md. Hafizur Rahman Masum, Kantish Roy Chowdhury,2022. Bluetooth Controlled Arduino Based Autonomous Car Final Report. <https://www.researchgate.net/publication/359415478_Bluetooth_Controlled_Arduino_Based_Autonomous_Car>. 12/6/2023.

[3] Souvik Paul, Saumedhik Biswas, AtreyoSengupta, Banhishikha Basu, Sreya Basu, 2019. Arduino based, Bluetooth controlled RC Car. <https://www.researchgate.net/publication/337759619_Arduino_based_Bluetooth_controlled_RC_Car>. 28/6/2023.

[4] Michael Klements, 2020. Arduino Based Obstacle Avoiding Robot Car. <https://www.the-diy-life.com/arduino-based-obstacle-avoiding-robot-car/>. 30/6/2023.

[5] sparkfun.com, 2008. nRF24L01+ Single Chip 2.4GHz Transceiver Preliminate Product Specification v1.0. <https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf>. 6/6/2023.

[6] controllerstech.com, 2021. How to Write Basic Library for NRF24L01 PART 1 || Common configuration || STM32 SPI. <https://www.youtube.com/watch?v=mB7LsiscM78>. 5/6/2023.

[7] controllerstech.com, 2021. How to Write Basic Library for NRF24L01 PART2 || Transmit DATA || SPI STM32. <https://www.youtube.com/watch?v=uPGgLwYGeYQ>. 5/6/2023.

[8] controllerstech.com, 2021. How to Write Basic Library for NRF24L01 PART 3 || Receive DATA || SPI STM32. <https://www.youtube.com/watch?v=X5XDSWQYYvU&t=496s>. 5/6/2023.

[9] stm32-base.org, 2014. WeAct Black Pill V1.2 STM32F401CCU6 specification. <https://stm32-base.org/boards/STM32F401CCU6-WeAct-Black-Pill-V1.2.html>. 6/6/2023

[10] stm32-base.org, 2014. Blue Pill - STM32F401CCU6 specification. <https://stm32-base.org/boards/STM32F103C8T6-Blue-Pill.html>. 6/6/2023

[11] st.com, 2018. RM0368 Reference manual STM32F401xB/C and STM32F401xD/E advanced Arm®-based 32-bit MCUs. <https://www.st.com/resource/en/reference_manual/rm0368-stm32f401xbc-and-stm32f401xde-advanced-armbased-32bit-mcus-stmicroelectronics.pdf>. 4/6/2023.

[12] st.com, 2021. RM0008 Reference manual STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced Arm®-based 32-bit MCUs. <https://www.st.com/resource/en/reference_manual/rm0008-stm32f101xx-stm32f102xx-stm32f103xx-stm32f105xx-and-stm32f107xx-advanced-armbased-32bit-mcus-stmicroelectronics.pdf>. 2/6/2023.