KARNATAKA LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG BELAGAVI -590008

KARNATAKA, INDIA.

A Course Project Report on

ALL TERRAIN ROBOT - EMBEDDED FINITE STATE MACHINE IMPLEMENTATION

Submitted for the requirements of 5th semester B.E. in CSE

for "**Object Oriented Modelling And Programming**"

Submitted by

NAME	USN
SHRADHA MALLIKARJUN PATIL	2GI20CS144
SRUSHTI B MUDENNAVAR	2GI20CS158
YASH HEREKAR	2GI20CS184

Under the guidance of

Prof., Girish Deshpande

Asst. Prof., Dept. of CSE

Academic Year 2022 - 2033 (Odd semester)

KARNATAKA LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG BELAGAVI -590008

KARNATAKA, INDIA.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Course Project work titled **"All Terrain Robot - Embedded finite state machine implementation"** carried out by **SHRADHA MALLIKARJUN PATIL, SRUSHTI B MUDENNAVAR AND YASH HEREKAR** bearing **USNs: 2GI20CS144, 2GI20CS158 and 2GI20CS184** for **OBJECT ORIENTED MODELLING AND DESIGN (18CS52) COURSE** is submitted in partial fulfilment of the requirements for 5th semester B.E. in **COMPUTER SCIENCE AND ENGINEERING,** Visvesvaraya Technological University, Belagavi. It is certified that all corrections/ suggestions indicated have been incorporated in the report. The course project report has been approved as it satisfies the academic requirements prescribed for the said degree.

Date: 31/1/2023

Place: Belagavi

Signature of guide

Prof. Girish Deshpande

Asst., Prof., Dept of CSE

KLS Gogte Institute Of Technology, Belagavi

Academic Year 2022 - 2023 (Odd semester)

Marks allocation:

	Batch No.: 5				
1.	Seminar Title: **All Terrain Robot - Embedded finite state machine implementation**	Marks Range	**USN**		
			2GI20CS144	**2GI20CS158**	**2GI20CS184**
2.	Abstract (PO2)	0-2			
3.	Application of the topic to the course (PO2)	0-3			
4.	Literature survey and its findings (PO2)	0-4			
5.	Methodology, Results and Conclusion (PO1, PO3, PO4)	0-6			
6.	Report and Oral presentation skill (PO9, PO10)	0-5			
	Total	20			

*** 20 marks is converted to 10 marks for CGPA calculation**

Signature of Staff

ACKNOWLEDGEMENTS

We would want to take this opportunity to thank everyone who has contributed to the accomplishment of our project. We really appreciate Dr. J. K. Kittur's kind assistance during the B.E. program as the principal of G.I. T. in Belagavi. For his unwavering collaboration and assistance throughout this whole research, Dr. Vijay S. Rajpurohit, professor and head of the Department of computer science at G.I.T., Belagavi, has our sincere gratitude. We would like to thank Prof. Girish Deshpande from the Department of CSE at GIT Belagavi for serving as the project's mentor. She has continuously supported us and served as a source of inspiration for the duration of the project. We extend our gratitude to each one of the department's technical, non-teaching, and teaching staff.

Shradha Mallikarjun Patil 2GI20CS144

Srushti Mudennavar 2GI20CS158

Yash Herekar 2GI20CS184

ABSTRACT

An all-terrain robot is a versatile mobile device designed to traverse rough or uneven terrain, such as mud, gravel, sand, snow, and rocky terrain. These robots are equipped with specialised features to maintain stability while performing tasks in difficult or dangerous environments. They are used in applications including search and rescue, military operations, environmental monitoring, and exploration.

It can be controlled remotely or programmed to operate autonomously, equipped with cameras, sensors, and other data-gathering instruments. They are designed to be rugged and durable, able to operate in extreme weather conditions and challenging environments.

Their versatility and ability to operate in difficult terrain make them valuable for performing tasks in hazardous areas. In search and rescue missions, they can locate missing persons, in military operations they can perform reconnaissance and surveillance, and in environmental monitoring they can gather data and samples. All-terrain robots also play a role in exploration, mapping unknown terrain, and gathering information.

Overall, all-terrain robots are an important tool for performing tasks in challenging environments. With their versatility, durability, and ability to gather data and perform tasks remotely, they are valuable assets in various fields.

CONTENTS

All Terrain Robot

Embedded finite state machine implementation

1. Introduction

All-Terrain Robot is a sophisticated and advanced robot designed to tackle challenging terrains and environments. This robot has been developed with a focus on versatility and functionality, implementing a finite state machine, classes and objects to perform complex tasks. The robot is equipped with a red LED, blue LED, buzzer, and BTS7960 motor driver to ensure smooth and efficient operation.

The All-Terrain Robot is based on the popular and versatile Arduino Nano platform and is controlled via the i-bus protocol using the Flysky TH9X transmitter. The i-bus protocol provides a stable and reliable communication link between the transmitter and the robot, ensuring that the robot can be controlled remotely with ease. The Flysky TH9X transmitter is a high-performance remote control that provides a smooth and responsive control experience, making it an ideal choice for controlling the All-Terrain Robot.

The BTS7960 motor driver is a high-performance motor driver that can drive large DC motors efficiently and accurately. It has been integrated into the All-Terrain Robot to ensure that the robot can be driven smoothly and accurately across challenging terrains. The red and blue LED lights provide visual feedback on the robot's state and status, while the buzzer can be used for audio feedback or to alert the operator.

The All-Terrain Robot is a testament to the capabilities of modern embedded systems and robotics technology. It demonstrates the potential for robotics to revolutionise the way we interact with and explore the world around us. Whether used for research and exploration, transportation, entertainment or any other purpose, the All-Terrain Robot is a highly capable and versatile platform that is sure to impress.

2. CLASS DIAGRAM

2.1 LED CLASS

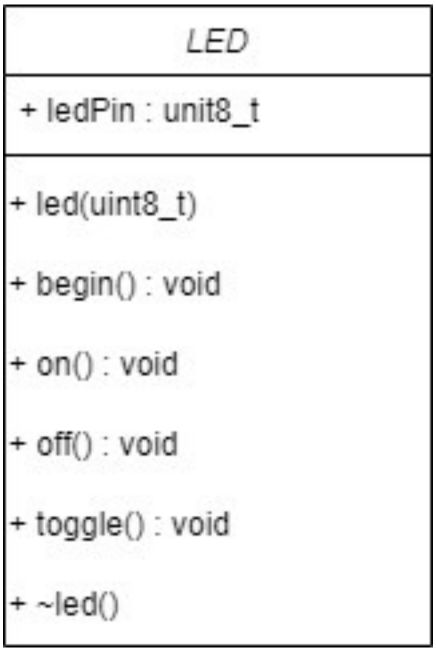


Fig 2.1 led class diagram

2.2 BUZZER CLASS

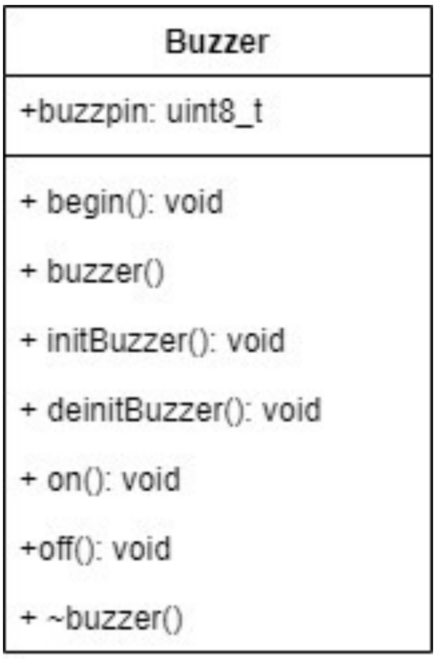


Fig 2.2 Buzzer class diagram

2.3 BTS7960 MOTOR DRIVER CLASS

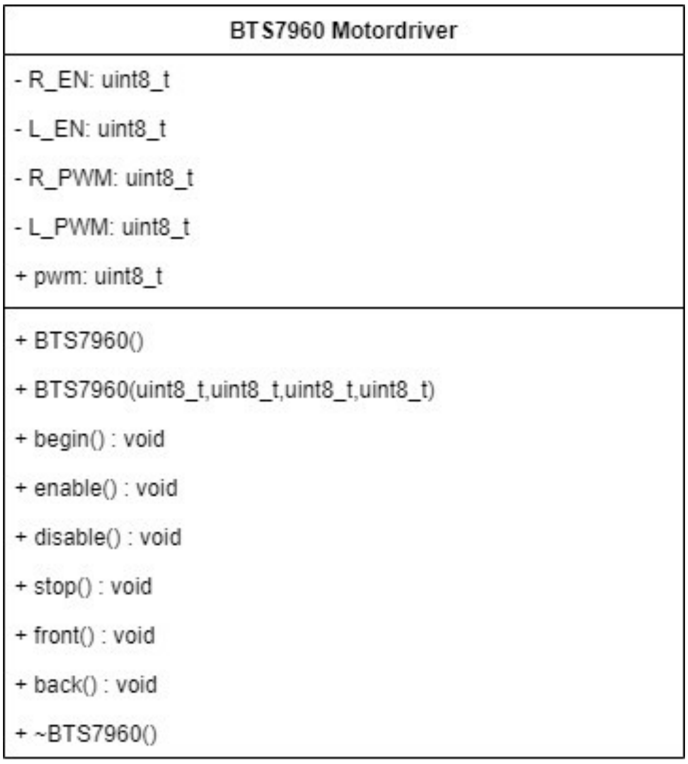


Fig 2.3 BTS7960 Motor-driver class

2.4 FLYSKY I-BUS CLASS

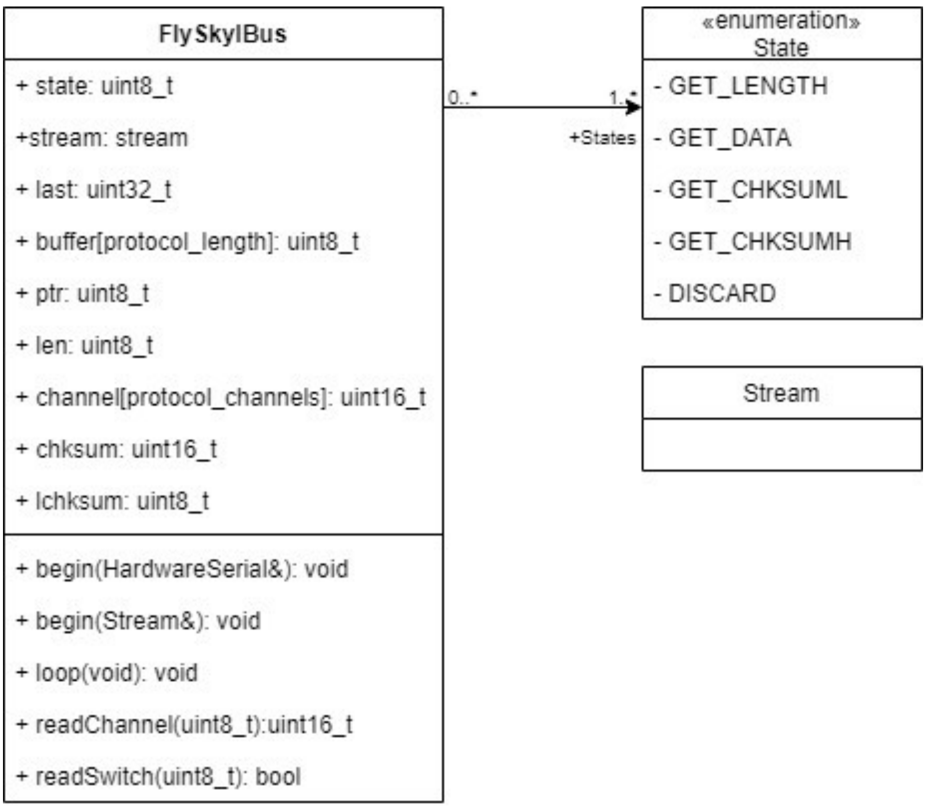


Fig 2.4 Flysky i-bus class diagram

3. State Machine Diagram

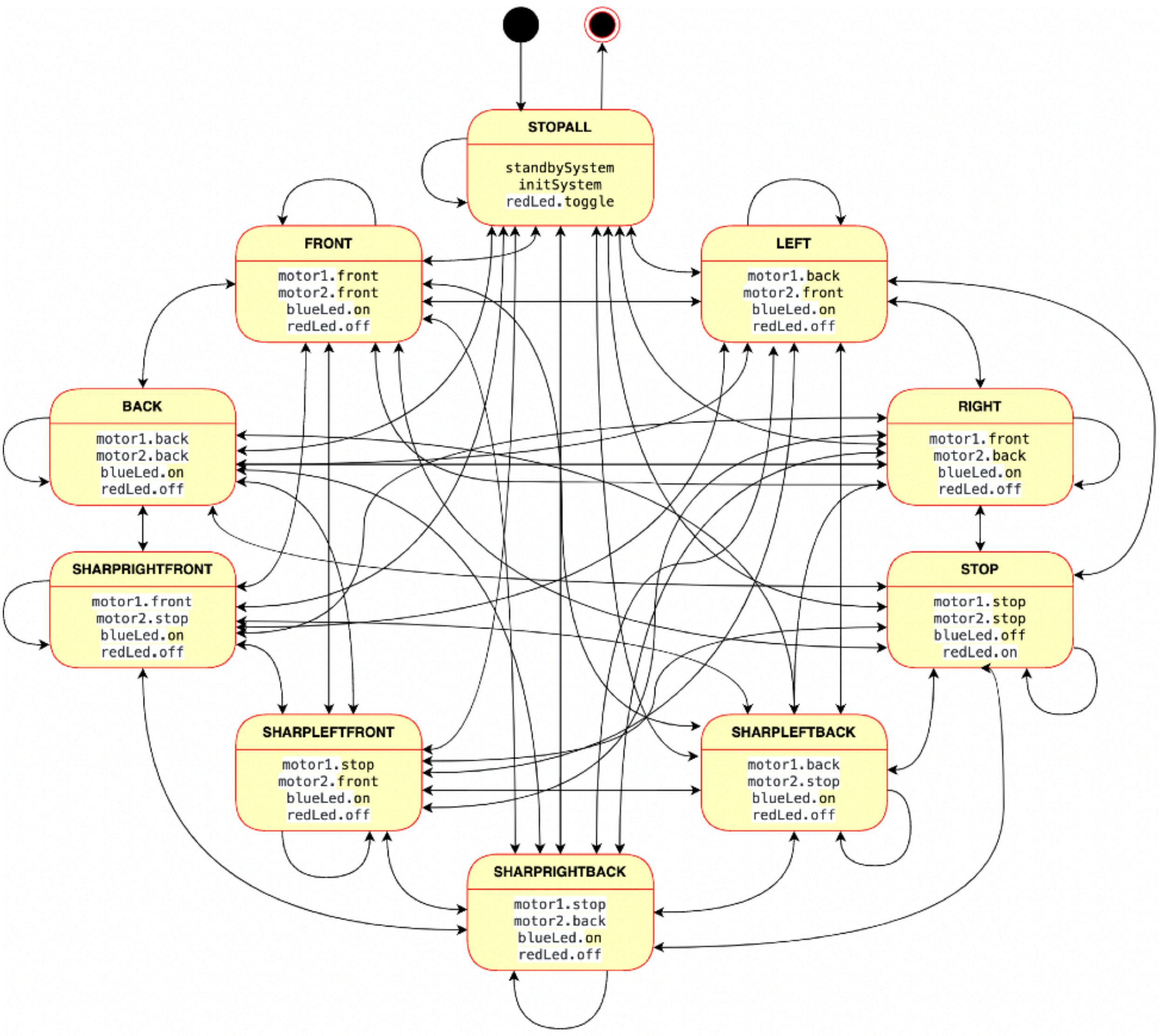


Fig 3.1 Robot state machine

Table 3.1. Functions of the wheels

Functions	Wheel 1	Wheel 2	Wheel 3	Wheel 4
Front	↑	↑	↑	↑
Back	↓	↓	↓	↓
LeftTurn	↓	↓	↑	↑
RightTurn	↑	↑	↓	↓
SharpLeftFront	↑	-	-	↑
SharpRightFront	-	↑	↑	-
SharpLeftBack	-	↓	↓	-
SharpRightBack	↓	-	-	↓
stop	-	-	-	-

4. INTERACTION MODEL

4.1 USE CASE DIAGRAM

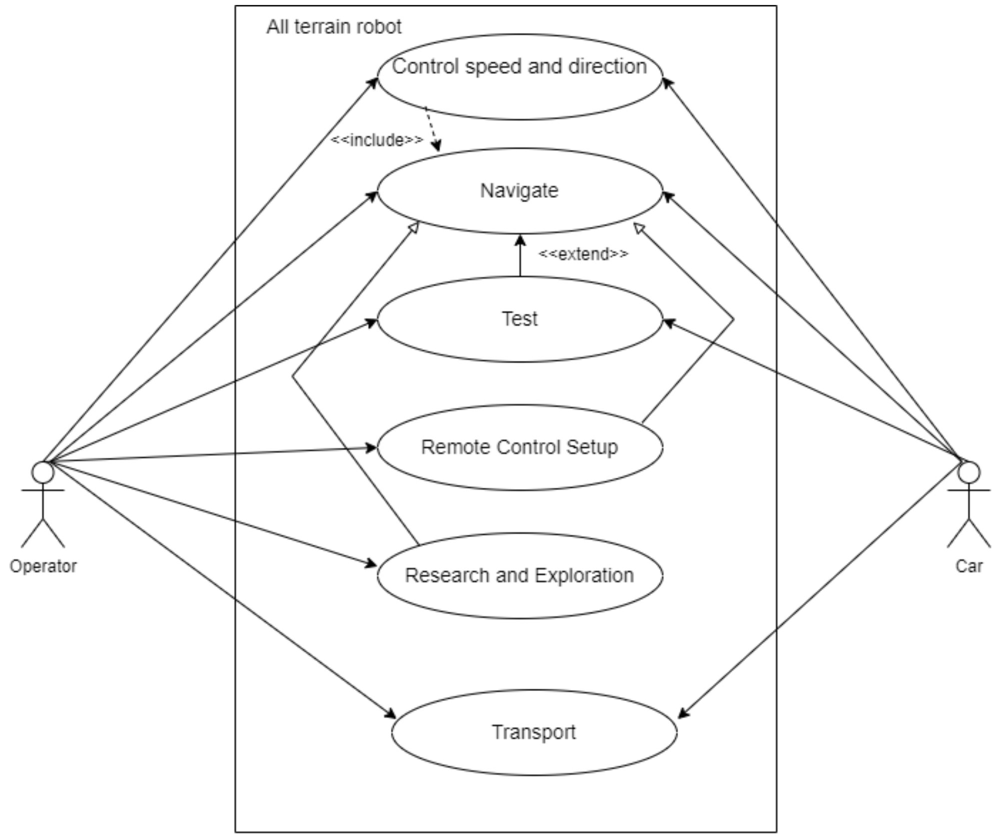


Fig 3.2 Use case diagram

Actors:

- User
- Car

Use Cases:

- Navigate
- Control Speed and Direction
- Test
- Remote control Setup
- Research and Exploration
- Transportation

Relationships:

- Extend
- Include

4.2 SEQUENCE DIAGRAM

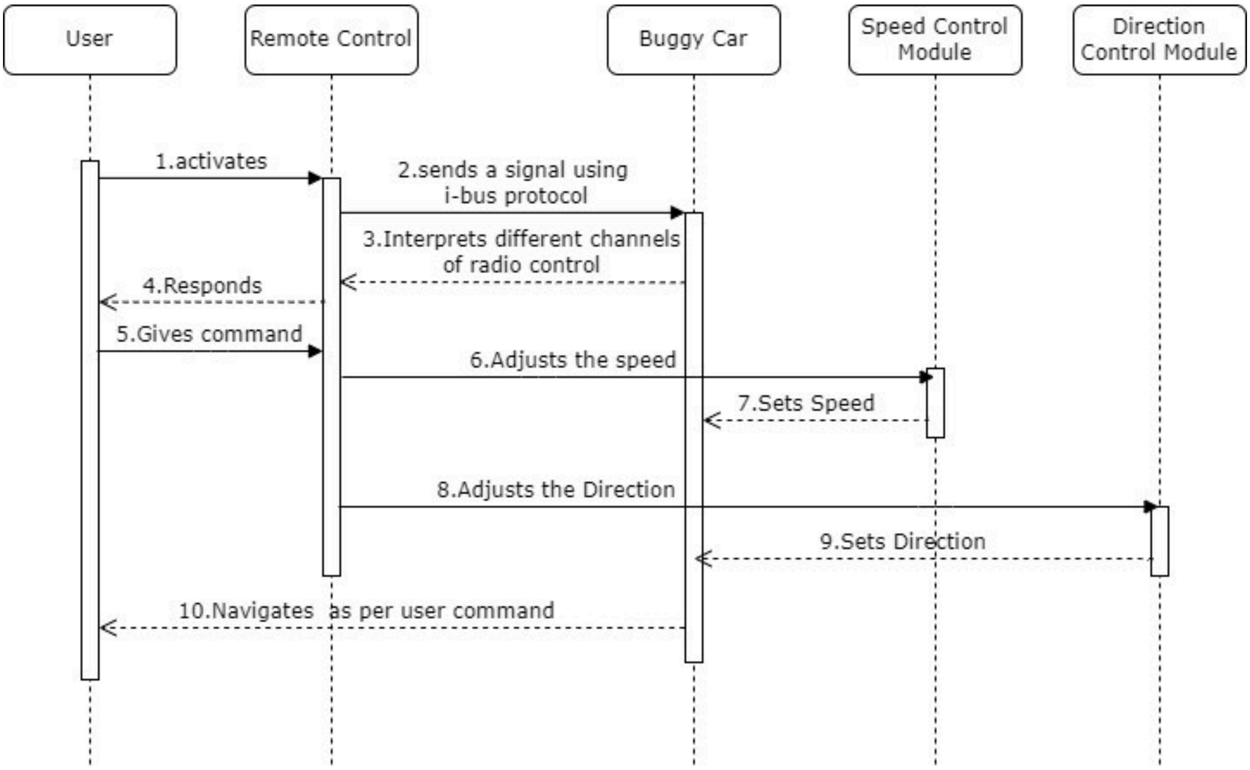


Fig 4.2 Sequence diagram

In a sequence diagram, activities and events are represented as messages between objects. The activities and events are listed in a chronologically sequenced manner, with each message arrow indicating the flow of control between objects. Some common activities and events that can be present in a sequence diagram include system initialisation, method calls, sending and receiving messages, handling exceptions and errors, decision making, branching and looping. In the context of a remotely controlled four wheel buggy car, the activities and events can include the following:

1. Activates: The connection between the remote control and the buggy car is established.
2. Sends i-bus signal: The user inputs commands using the remote control.
3. Message transfer: The commands from the remote control are sent to the buggy car over the i-bus protocol.
4. Sets speed: The speed of the buggy car is controlled by the speed control module in response to the commands received from the remote control.
5. Direction control: The direction of the buggy car is controlled by the direction control module in response to the commands received from the remote control.

6. Navigates as per user: The connection between the remote control and the buggy car is closed.

4.3 Activity Diagram

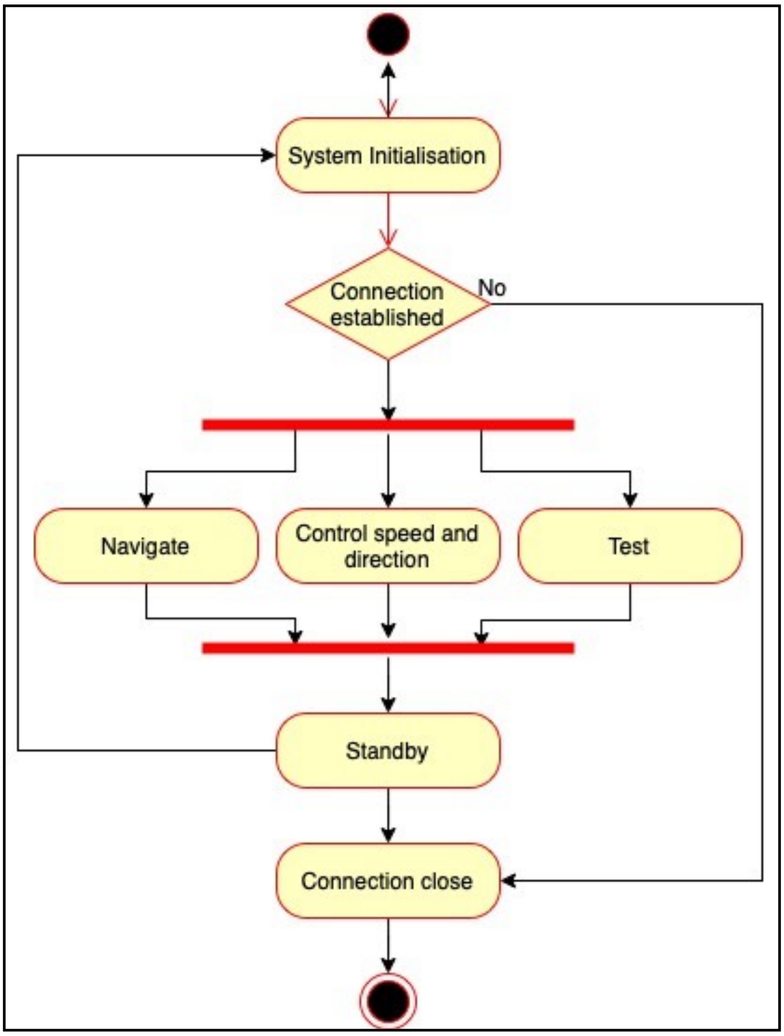


Fig 4.3 Activity diagram

1. System Initialisation: This is the starting point of the activity diagram, where the system is initialised and the necessary setup steps are performed.
2. Connection Established: This is a condition in the diagram that checks whether the connection between the remote control and buggy car is established or not.
3. Navigate: This is an activity where the buggy car is controlled to move in different directions like front, back, left, right, sharp left turn, sharp right turn.
4. Control Speed and Direction: This activity involves controlling the speed and direction of the buggy car, which is done using the speed control module and direction control module.

5. Test: This is an activity that allows testing the buggy car to ensure it is functioning properly.
6. Standby: This is a junction in the diagram where the navigate, control speed and direction, and test activities are joined.
7. Close Connection: This activity involves closing the connection between the remote control and buggy car.
8. Connection Closed: This is the end point of the activity diagram where the connection is closed and the buggy car is no longer controlled by the remote control. If the connection established condition is false, the flow will be redirected to this point.

5. HARDWARE IMPLEMENTATION

5.1 ARDUINO NANO

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328P (Arduino Nano 3.x) or ATmega168 (Arduino Nano 2.x). Some of its key specifications are:

- Microcontroller: ATmega328P (Arduino Nano 3.x) or ATmega168 (Arduino Nano 2.x)
- Operating voltage: 5V
- Input voltage: 7-12V
- Digital I/O pins: 14 (of which 6 provide PWM output)
- Analog inputs: 8
- DC current per I/O pin: 40 mA
- Flash memory: 32 KB (ATmega328P) or 16 KB (ATmega168) of which 2 KB used by boot-loader
- SRAM: 2 KB (ATmega328P) or 1 KB (ATmega168)
- EEPROM: 1 KB (ATmega328P) or 512 bytes (ATmega168)
- Clock speed: 16 MHz
- Dimensions: 0.73 x 1.70 in (18.5 x 43.2 mm)

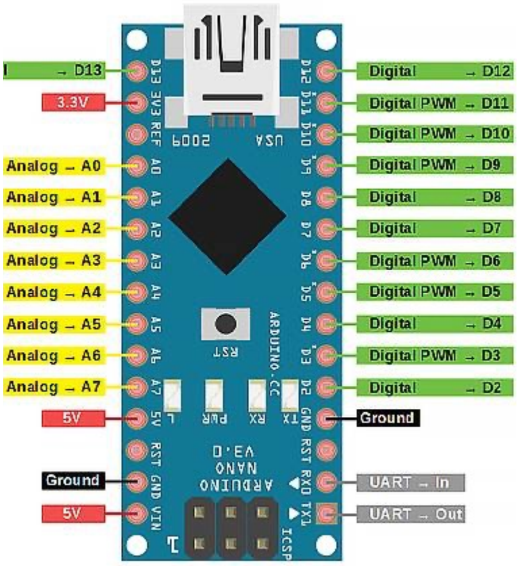


Fig 5.1 Arduino Nano

5.2 BTS7960 MOTOR DRIVER

- Maximum continuous current rating of 43A
- Wide voltage range input of 7-42V DC
- Internal MOSFETs for high efficiency and low thermal dissipation
- Over temperature and over current protection
- Under voltage lockout protection
- Three-phase sinusoidal driver
- TTL compatible control inputs
- Support for bidirectional rotation control
- Low output resistance
- Low electromagnetic interference
- Small form factor, suitable for space-limited applications

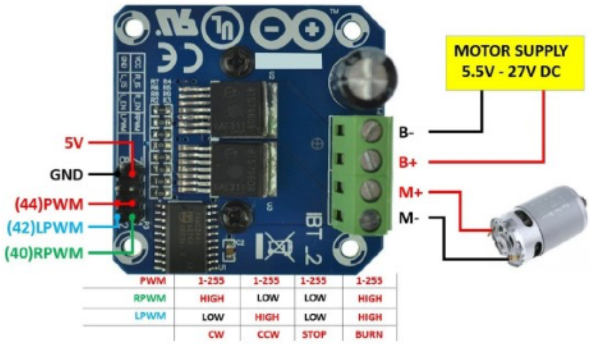


Fig 5.2 BTS7960 Motordriver

5.3 MOTORS

Johnson 300 RPM motors are high-torque DC motors that are commonly used in robotics and automation projects. The specifications for these motors can include the following:

1. Voltage: These motors typically operate on DC voltage in the range of 12V to 24V.
2. Speed: The rated speed of these motors is 300 RPM, which is their maximum speed under no-load conditions.
3. Torque: These motors are known for their high-torque capabilities, typically in the range of 50 oz-in to 200 oz-in, depending on the model.
4. Shaft size and shape: The shaft of these motors can be round or hexagonal, with a diameter ranging from 5mm to 10mm.
5. Current draw: The current draw of these motors will depend on the load they are carrying, but it

Johnson motor 300 rpm

typically ranges from 0.5A to 3A.

6. Mounting: These motors come with various mounting options, such as brackets or flanges, to make installation easier.
7. Dimensions: The dimensions of these motors can vary depending on the model, but they are typically compact and lightweight, making them ideal for use in mobile or space-constrained applications.
8. Encoder: Some models of Johnson 300 RPM motors are available with an encoder, which provides feedback on the position and speed of the motor for precise control.

5.4 Flysky TH9X transmitter and Fsia10b receiver

The Flysky TH9X is a 9-channel radio control system used for remote control of hobby vehicles such as drones, cars, and boats. Some of its key specifications include:

- Operating voltage: 4.0~8.4V DC
- Operating current: ≤ 140mA (@ 8.4V)
- Operating temperature: -10°C~60°C
- Channel resolution: 1024 steps
- Output power: ≤100mW
- Frequency range: 2.4055GHz~2.475GHz
- Modulation: GFSK
- Spread spectrum: AFHDS 2A
- Antenna length: 26mm
- Dimensions: 174mm × 89mm × 190mm
- Weight: 392g

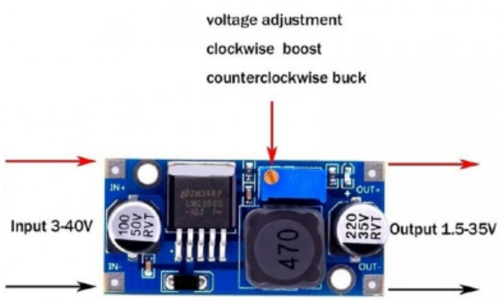
Flysky th9x transmitter and Fsia10b receiver

5.5 LM2596 DC-DC Buck converter

The LM2596 DC-DC Buck Converter is a step-down voltage regulator that can convert a high input voltage into a lower, regulated output voltage. Some specifications for the LM2596 include:

- Input voltage range: 3V to 40V
- Output voltage range: 1.23V to 37V
- Output current: 3A (max)

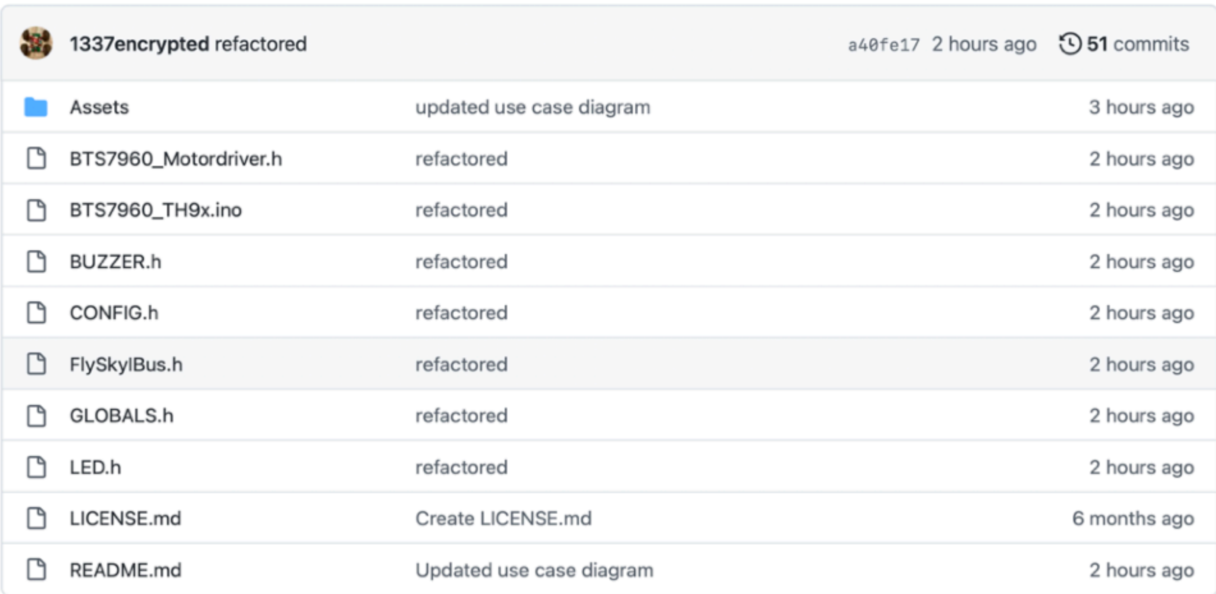
- Switching frequency: 150KHz
- Efficiency: up to 92%
- Operating temperature: -40°C to +85°C
- Short-circuit protection, thermal shutdown and under-voltage lockout protection
- Small size, low profile, and low cost
- Available in an adjustable or fixed output voltage version
- Can be used for various applications such as battery-powered devices, battery charging, voltage regulation, voltage stabilisation and many more.



LM2596 DC-DC buck converter

6. Software Development

- The project is open-source on GitHub https://github.com/1337encrypted/BTS7960_TH9x



1337encrypted refactored		a40fe17 2 hours ago 51 commits
Assets	updated use case diagram	3 hours ago
BTS7960_Motordriver.h	refactored	2 hours ago
BTS7960_TH9x.ino	refactored	2 hours ago
BUZZER.h	refactored	2 hours ago
CONFIG.h	refactored	2 hours ago
FlySkyIBus.h	refactored	2 hours ago
GLOBALS.h	refactored	2 hours ago
LED.h	refactored	2 hours ago
LICENSE.md	Create LICENSE.md	6 months ago
README.md	Updated use case diagram	2 hours ago

Fig 6.1 Github repository

- Its based on the apache license Version 2.0 http://www.apache.org/licenses/LICENSE-2.0

- The project is collaborated via GitHub via issues section

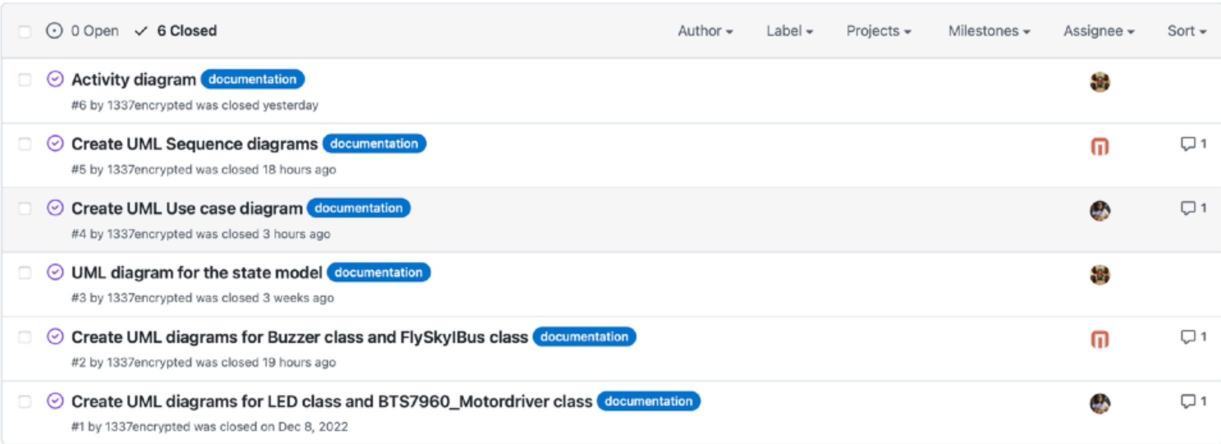


Fig 6.2 Github issues

7. Conclusion and Future Work

The buggy car project was a comprehensive study on an all-terrain robot that was designed and developed using cutting-edge technology and advanced engineering principles. The project involved the use of various computer science concepts, including finite state machines, classes, objects, and protocols like i-bus to create a functional and efficient all-terrain robot. The use of the Arduino Nano as the microcontroller and the BTS7960 motor driver was a key aspect of the project and allowed the buggy car to control its speed, direction and handle multiple inputs from the user. The use of the flysky th9x transmitter allowed the user to control the robot using a remote control and i-bus protocol. Additionally, the project involved the creation of detailed diagrams and models, including state machines, activity diagrams, and sequence diagrams, which helped to clarify the functional requirements and design of the buggy car. The conclusion is that the buggy car project was a great example of the integration of computer science, engineering, and electronics to create a functional and efficient all-terrain robot that could be used for various purposes such as transport, research, exploration, and entertainment.