Embedded Systems 4-Months Graduation Project

AutoSense

Supervised By:

Eng | Youssef Nofal
Eng | Nour Hassan

Submitted By:

Amr Mohammed Hassan Mohallel
Abdelrahman Mohamed Elsamahy Mohamed Ahmed
Mahmoud Gamal Fathy Abdelhamid
Ahmed Atef Mahmoud Mohammed
Motaz Adel Mohamed Soliman
Sherif Ashraf Ali Mostafa Khedr

Table of Contents

Table of Figures

1 - Preface

This document presents our team's work after a four-month program at the Information Technology Institute, focusing on embedded systems. Our project, AutoSense, reflects what we've learned and done during this time. At ITI, we studied how embedded systems, the tech 'brains' in devices, work. AutoSense is our practical showcase of this knowledge. In this document, we'll share how we came up with the idea, faced challenges, and created AutoSense. Our aim is for this document to help professionals interested in embedded systems. AutoSense is proof of our teamwork and the support we received from mentors and peers at ITI. We're thankful to everyone who helped us. Their guidance was crucial in making AutoSense happen. Thank you for taking an interest in our project, and we hope it sparks more innovation in the world of embedded systems.

2 - Introduction to ADAS

2.1 - What are ADAS Systems?

Advanced Driver Assistance Systems (ADAS) are electronic systems that assist drivers in driving and parking functions. They are designed to increase car safety and more generally road safety. In the modern automotive realm, the integration of Advanced Driver Assistance Systems (ADAS) has sparked a significant transformation, reshaping the landscape of vehicle safety, driving dynamics, and overall motoring experiences. ADAS represents a sophisticated confluence of technological advancements strategically designed to not only enhance driver safety but also improve driving efficiency and comfort.

2.2 - Benefits of ADAS Systems

- Increased safety for drivers and passengers
- Reduced risk of accidents
- Improved driving experience

2.3 - Technological Integration & Functionality

At its core, ADAS is a complex network of sensors, cameras, radar systems, and intricate AI algorithms interwoven into vehicle design. These systems work in unison, constantly monitoring the vehicle's surroundings, swiftly detecting potential hazards, analyzing risks, and offering real-time support and alerts to drivers.

2.4 - Enhancing Safety Through Features

ADAS introduces a wide spectrum of functional capabilities aimed at fortifying driving safety. These encompass collision avoidance systems, adaptive cruise control, lane departure warnings, automatic emergency braking systems, parking assistance tools, and blind spot detection technologies. Collectively, these features aim to forewarn drivers about potential dangers, aid in maintaining vehicle control, and autonomously intervene in specific scenarios to prevent or mitigate accidents.

2.5 - Transition to Autonomous Driving Capabilities

The rapid evolution of automotive technology has steered us toward the advent of autonomous driving capabilities. ADAS stands at the forefront of this shift, bringing us closer to a future where vehicles possess the ability to navigate and operate independently, reshaping the essence of transportation.

2.6 - Impact on Transportation and Safety Paradigms

Beyond technological innovation, the influence of ADAS extends to redefining transportation itself. These systems not only hold the promise of reshaping road safety paradigms but also revolutionizing the overall driving experience. The continual evolution and integration of ADAS mark a significant turning point in the automotive industry, signifying a shift towards safer, more convenient, and more efficient driving experiences on a global scale.

2.7 - Conclusion

This comprehensive overview delves into the profound implications and transformative influence of ADAS. It emphasizes the pivotal role these systems play in reshaping the automotive industry and setting the course for the future landscape of transportation and road safety.

3 - Introduction to FOTA

3.1 - What are Firmware Over-The-Air (FOTA)

In the realm of modern technology, particularly within the sphere of connected devices and IoT (Internet of Things), Firmware Over-The-Air (FOTA) updates have emerged as a pivotal advancement, revolutionizing the way devices receive and implement critical software updates. FOTA, an acronym for Firmware Over-The-Air, refers to a method of remotely updating and managing device firmware, enabling seamless, secure, and efficient distribution of software upgrades over a wireless network.

3.2 - Significance of FOTA in Device Maintenance

Firmware forms the underlying software that controls the functionalities and performance of devices, ranging from smart home gadgets and wearables to vehicles and industrial machinery. FOTA allows for the remote deployment of firmware updates, eliminating the need for manual, in-person interventions to update device software. This method significantly streamlines the update process, reducing downtime and enhancing overall device efficiency.

3.3 - Efficiency and Safety in Updates

FOTA updates ensure that devices are continually equipped with the latest security patches, bug fixes, and feature enhancements without necessitating physical interaction. This method not only ensures operational efficiency but also plays a crucial role in enhancing device security and resilience against evolving cyber threats.

3.4 - Application across Diverse Industries

The scope of FOTA extends across various industries, from consumer electronics and automotive sectors to healthcare, industrial automation, and beyond. It caters to a broad spectrum of devices, allowing manufacturers and developers to maintain and improve their products' performance seamlessly, ensuring that end-users benefit from updated, secure, and optimized devices.

3.5 - Conclusion

The advent of Firmware Over-The-Air (FOTA) updates marks a significant leap in device maintenance and security, playing a crucial role in ensuring devices are consistently up-to-date, secure, and performing optimally. This introduction aims to highlight the importance of FOTA in the era of connected devices and the ever-evolving landscape of technological advancements.

4 - Auto Sense

4.1 - What is auto sense?

AutoSense Integration of ADAS Features with LIDAR and IR Sensor Technology, Along with FOTA Capability The "AutoSense" project represents a remarkable synthesis of Advanced Driver Assistance Systems (ADAS), combining a suite of features including Adaptive Cruise Control (ACC), Lane Keep Assistance (LKA) utilizing IR sensors, Normal Cruise Control, and Automatic Brake Assistant. This innovative project is further enhanced by the integration of LIDAR sensor technology, along with the capability for Firmware Over-The-Air

(FOTA) updates. The integration of these advanced ADAS functionalities with LIDAR and IR sensor technology, along with FOTA capability, underscores the commitment of the AutoSense project to driving safety, comfort, and technological innovation within the automotive realm.

4.2 - ADAS Features

- Adaptive Cruise Control (ACC): At the core of the AutoSense project, ACC is bolstered by LIDAR sensors, ensuring precise detection and measurement of surrounding objects. This technology allows for adaptable speed adjustments, maintaining safe distances from other vehicles with a heightened level of accuracy.

- Lane Keep Assistance (LKA) - IR Sensors: LKA, utilizing IR sensors, assists in detecting lane boundaries, aiding in maintaining the vehicle's position within the lane. This feature significantly contributes to driver safety by reducing the risk of unintended lane departures, ensuring a more secure driving experience.

- Normal Cruise Control: The Normal Cruise Control feature, integrated with LIDAR technology, enables a more controlled and smoother driving experience, providing consistent speed control based on the vehicle's environmental perception.

- Automatic Brake Assistant: Enhanced by the precision of LIDAR sensors, the Automatic Brake Assistant system swiftly detects potential collision risks, applying additional braking force or emergency braking when necessary, significantly enhancing safety measures and collision avoidance.

4.3 - Firmware Over-The-Air (FOTA) Capability:

In addition to these advanced ADAS features, the AutoSense project incorporates FOTA capability, allowing for remote updates to the system's firmware. This feature ensures that the software remains up-to-date, secure, and optimized, facilitating seamless software maintenance and enhancements without the need for physical intervention.

4.4 - Hardware Components

4.4.1 - STM32f401ccu6 microcontroller

The STM32F401CCU6 is a high-performance microcontroller belonging to the STM32F4 series developed by STMicroelectronics. It's designed around a 32-bit ARM Cortex-M4 core, offering significant processing power suitable for a wide range of embedded applications.

Hardware Specifications

- Core: ARM Cortex-M4 running at a maximum frequency of 84 MHz, providing substantial computing capabilities for various tasks.
- Memory: Includes 512 KB of Flash memory for program storage and 96 KB of RAM for data storage, enabling efficient handling of program and data.

- Peripherals: Multiple GPIO (General-Purpose Input/Output) pins for versatile interfacing with external devices.Integrated communication interfaces such as SPI, I2C, and UART for data exchange.Analog-to-digital converter (ADC) for analog signal processing.Timers for various time-based operations.
- Power Efficiency: Optimized for low-power applications, extending battery life in energy-sensitive devices.

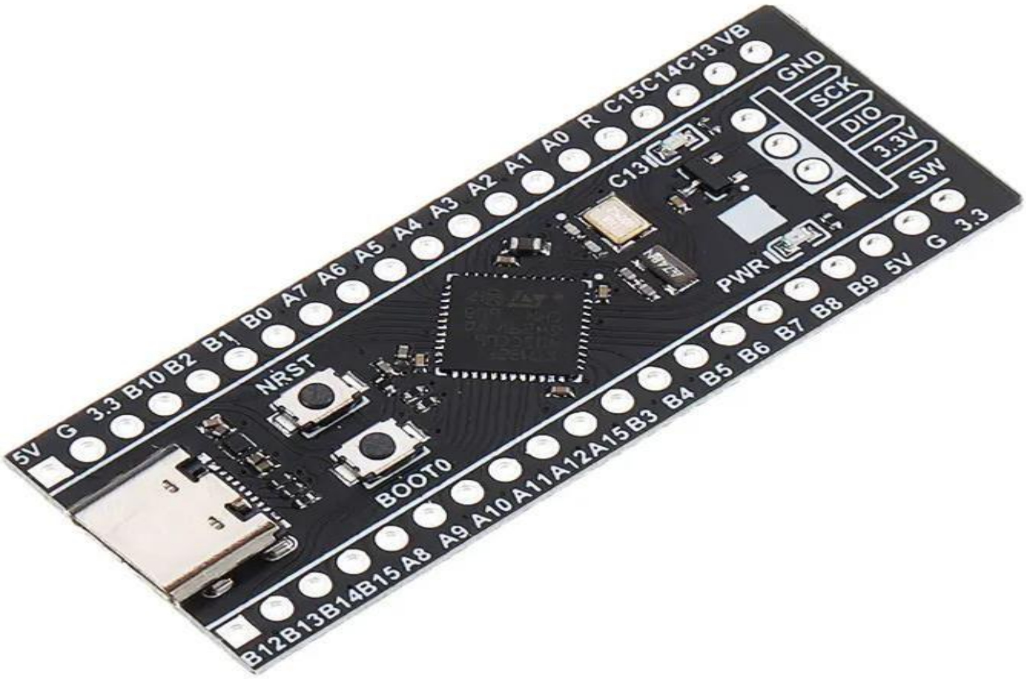


Figure 1. STM32F401CC Microcontroller

4.4.2 - RPlidar A2

The RPLIDAR A2, a Pulsed Light Detection and Ranging sensor, represents an advanced 2D LIDAR solution developed by PulsedLight, designed to offer high-precision spatial mapping and distance measurement capabilities in various applications.

- LIDAR Technology: The RPLIDAR A2 employs laser-based technology for Light Detection and Ranging (LIDAR), utilizing pulses of laser light to measure distances and generate accurate 2D spatial maps. This technology enables precise and rapid distance calculations, making it ideal for real-time applications requiring detailed environmental understanding.

- Performance Metrics: Operating at an impressive scan rate of up to 8000 samples per second, the PLIDAR A2 ensures swift and accurate data acquisition, delivering real-time information for applications demanding quick response and high accuracy.

- Range and Field of View: With a scanning range of up to 18 meters and a comprehensive field of view spanning 360 degrees horizontally and 30 degrees vertically, the PLIDAR A2 offers extensive coverage, allowing for comprehensive mapping and spatial understanding in various environments.

- Compact Design and Integration: Characterized by its compact form factor and lightweight build, the RPLIDAR A2 is highly adaptable for integration into space-constrained platforms such as drones, robotics, and other systems where size and weight considerations are crucial.

- Communication Interfaces: Equipped with standardized communication interfaces like USB and serial connections, the PLIDAR A2 ensures seamless integration and connectivity with a wide array of devices and systems, facilitating ease of use and deployment.

- Diverse Applications: Its versatility lends itself to diverse applications including but not limited to autonomous navigation, mapping, object detection, environmental scanning, industrial automation, and various robotics applications.

- Open-Source Software Support: Often accompanied by open-source software and development tools, the RPLIDAR A2 allows developers and engineers to tailor and customize the sensor's functionalities to suit the specific needs of their applications, promoting flexibility and adaptability.

The RPLIDAR A2 stands out as a sophisticated and reliable sensor solution, offering exceptional precision, a wide range of applications, and ease of integration. Its rapid and accurate scanning capabilities, compact design, and versatility make it an invaluable tool for industries and projects that necessitate detailed spatial mapping and precise distance measurements in real-time scenarios.

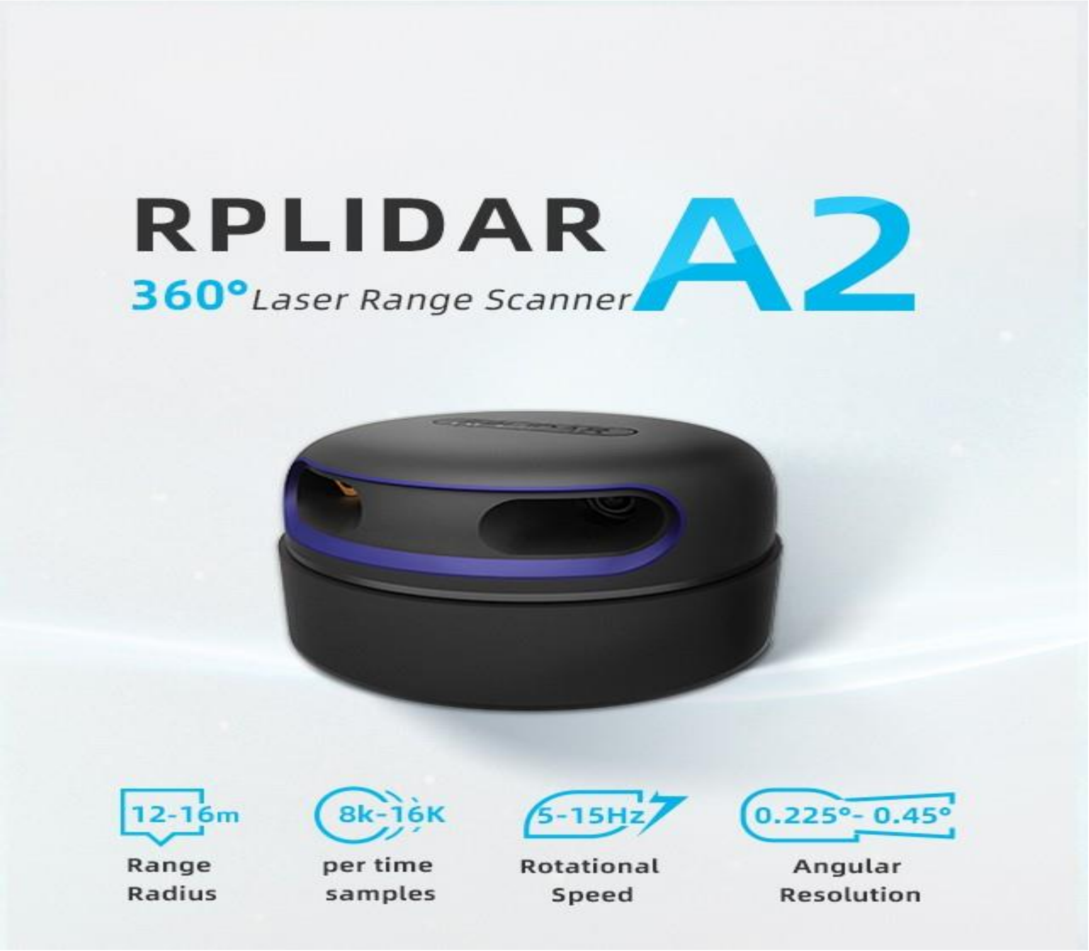


Figure 2. RPLIDAR A2

4.4.3 - IR Sensor

Infrared (IR) sensors are devices designed to detect and respond to infrared radiation.

- Sensing Principle: IR sensors detect infrared radiation in the form of heat emitted by objects. They operate based on the principle that different materials and surfaces emit heat in the form of IR radiation.

- Detection Range: IR sensors typically have a limited range of detection, commonly used for short to medium-range applications, especially in proximity and motion sensing.

- Working Mechanism: These sensors often consist of an IR emitter that emits infrared radiation and a receiver that detects the reflected or emitted radiation. When an object comes within the sensor's range, it alters the emitted or reflected infrared radiation, allowing the sensor to detect the presence or movement of an object.

- Applications: IR sensors are widely used in various applications such as proximity sensors for touchless technology, object detection, motion detection in security systems, and as presence sensors in automatic doors or hand dryers.

- Types of IR Sensors: They come in various types, including Passive Infrared (PIR) sensors, which are commonly used in motion detection, and Infrared Obstacle Avoidance sensors, which are used in robotics for obstacle detection.

- Advantages: IR sensors are cost-effective, consume low power, and are relatively simple to use, making them ideal for applications that require motion or proximity detection.

- Limitations: While effective for short-range applications, IR sensors can be affected by environmental factors such as temperature changes or ambient light conditions, which might affect their accuracy.

- IR Remote Controls: IR sensors are commonly used in devices like remote controls for televisions, air conditioners, and other consumer electronics to communicate with the corresponding appliances using infrared signals.

- Integration with Microcontrollers: IR sensors can be easily integrated with microcontrollers and other embedded systems, enabling seamless interfacing and control for various applications.

IR sensors are versatile devices used in a wide array of applications where proximity detection, motion sensing, and object presence recognition are essential. Their cost-effectiveness, simplicity, and reliability make them valuable components in numerous technologies and devices.

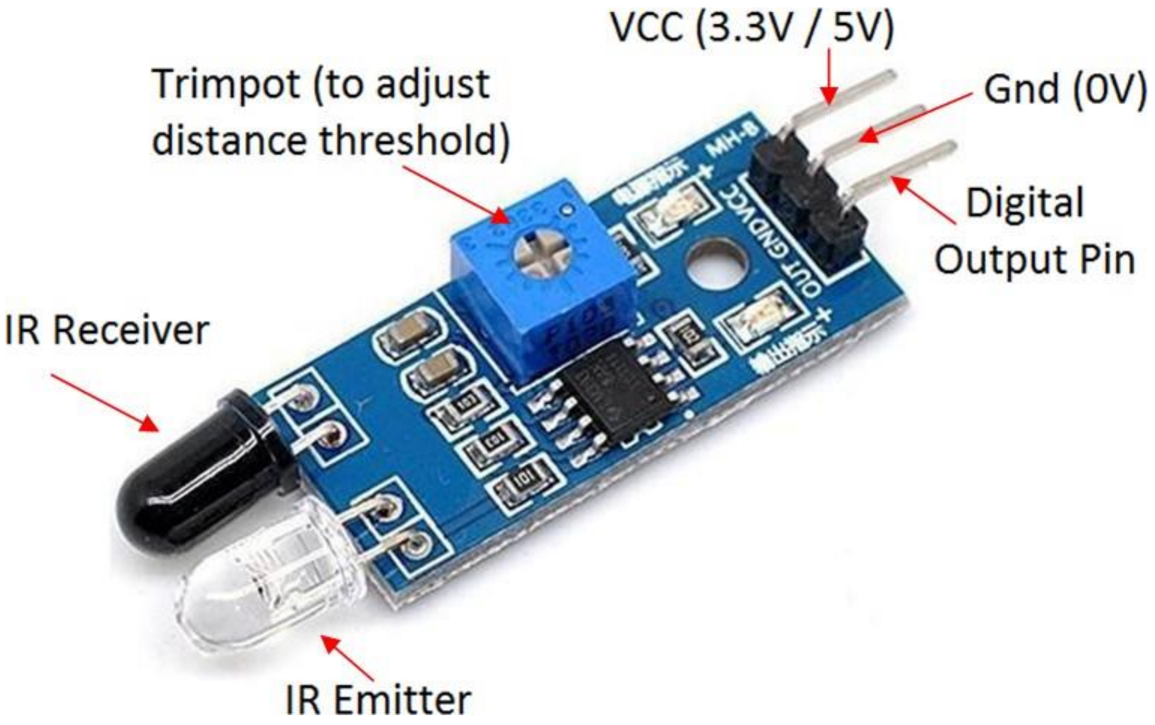


Figure 3. IR Sensor

4.4.4 - Dc Motor

DC (Direct Current) motors are electric motors that convert electrical energy into mechanical energy.

- Basic Principle: DC motors operate on the principle of electromagnetism. When electric current passes through a coil within a magnetic field, a force is generated, causing the motor's rotor to turn.

- Types of DC Motors
    - Brushed DC Motors: These motors use brushes to conduct current to the rotor, providing the magnetic field to rotate the shaft.
    - Brushless DC Motors (BLDC): BLDC motors utilize electronic commutation, eliminating the need for physical brushes, and are often more efficient and reliable.

- Construction: A DC motor typically consists of a stator, rotor (or armature), commutator, and brushes (in brushed DC motors). The stator generates a magnetic field, while the rotor rotates within this field.

- Advantages
    - They offer easy speed and direction control, making them suitable for applications that require variable speeds.
    - DC motors are robust, efficient, and cost-effective, making them widely used in various industries and applications.

- Applications: DC motors find applications in numerous devices and systems, including small household appliances, electric vehicles, industrial machinery, robotics, and HVAC systems.

- Speed and Torque Control: By adjusting the voltage input or the strength of the magnetic field, the speed and torque of a DC motor can be controlled.

- Power Supply: These motors operate using a direct current power supply. The motor speed can be controlled by varying the voltage.

- Efficiency and Maintenance: Brushless DC motors are often preferred due to their higher efficiency and lower maintenance needs, as they lack brushes that may wear out over time.

- Integration with Electronics: DC motors can be easily integrated with electronic controllers and microcontrollers for precise speed and position control in applications like robotics and automation.

- Limitations: Brushes in traditional DC motors are subject to wear and may require periodic replacement, impacting their maintenance requirements.

DC motors are a fundamental part of various industries, providing essential motion and power in a wide range of applications due to their simplicity, reliability, and controllability. Their versatility and ability to be adapted to different tasks make them integral components in the world of electric motors.

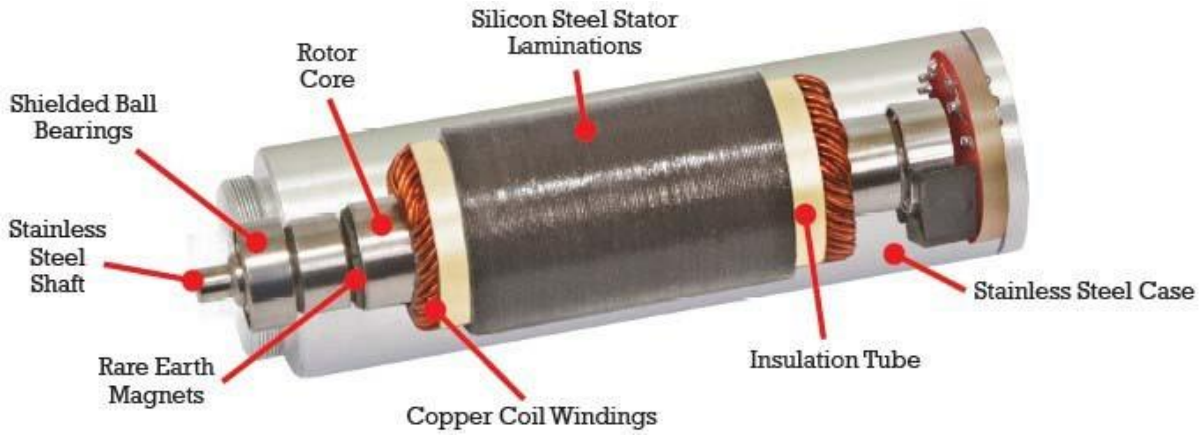


Figure 4. DC Motor

4.4.5 - Motor Driver

Motor drivers are specialized circuits or modules that control the operation and movement of electric motors.

- Function: Motor drivers act as an interface between a microcontroller or control system and an electric motor. They control the speed, direction, and other aspects of motor operation.

- Types of Motor Drivers
  - DC Motor Drivers: Control the speed and direction of DC motors, offering functionalities such as speed regulation and direction control.
  - Stepper Motor Drivers: Specifically designed for stepper motors, enabling precise positioning through step-by-step movements.
  - Brushless DC Motor (BLDC) Drivers: Manage the operation of BLDC motors, allowing for smoother and more efficient control of these motors.

- Components: Motor drivers typically include power transistors, H-Bridge circuits, MOSFETs, or other semiconductor devices to manage the flow of current to the motor.

- Control Signals: They accept control signals from microcontrollers or control systems, interpreting these signals to manage the motor's speed, direction, and other operational parameters.

- Current and Voltage Rating: Motor drivers are rated for specific current and voltage ranges to match the requirements of the motor they control, ensuring efficient and safe operation.

- Protection Features: Many motor drivers include protection features such as overcurrent protection, over-temperature protection, and short-circuit protection to safeguard the motor and driver circuitry.

- PWM Control: Pulse Width Modulation (PWM) signals are commonly used to control the speed of motors through motor drivers, enabling smooth and variable speed control.

- Applications: Motor drivers are utilized in a wide range of applications, including robotics, automation, CNC machines, electric vehicles, consumer electronics, and industrial machinery.

- Integration and Interfacing: They can be easily interfaced with microcontrollers and other control systems, providing a convenient and flexible means to control motors.

- Efficiency and Performance: Efficient motor drivers help maximize the performance and lifespan of motors by regulating and managing power delivery to meet the motor's requirements.

Motor drivers play a crucial role in modern motor control applications, offering precise, reliable, and efficient means to regulate the operation of electric motors in various industries and systems. Their versatility and integration capabilities contribute to the seamless control of motors in diverse applications.

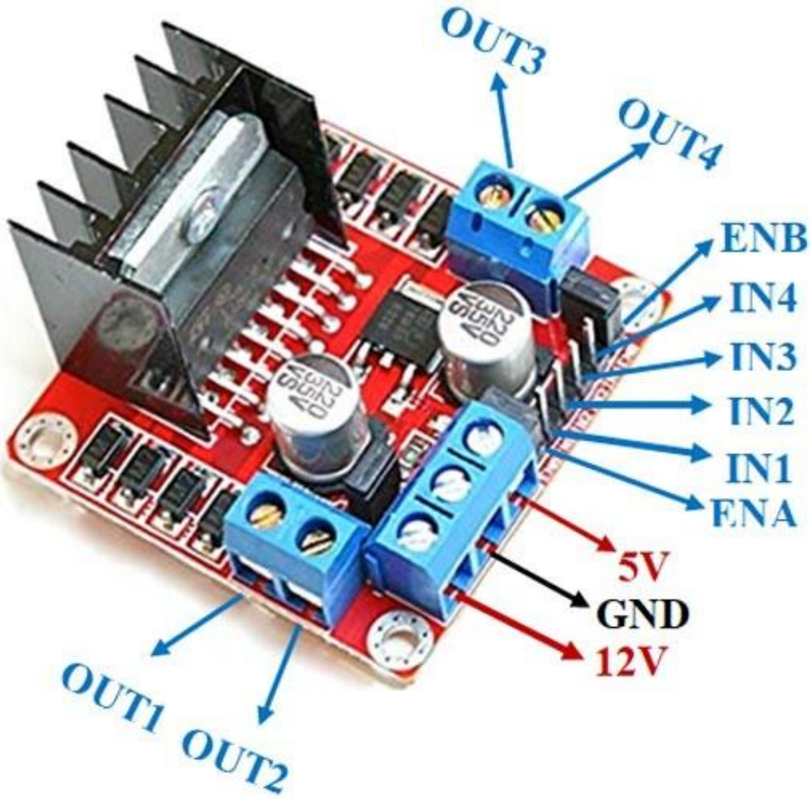


Figure 5. Motor Driver

4.4.6 - Hc05 Bluetooth Module

The HC-05 Bluetooth module is a widely used Bluetooth communication module known for its compatibility and ease of use.

- Bluetooth Version: The HC-05 Bluetooth module typically supports Bluetooth version 2.0 + EDR (Enhanced Data Rate) and is backward compatible with older Bluetooth versions.

- Communication Range: Generally, it offers an effective communication range of up to 10 meters in an open environment, allowing devices to wirelessly exchange data within this range.

- Communication Modes: The HC-05 can be configured to function as a master or slave device, making it versatile for various Bluetooth communication applications.

- Serial Communication Interface: It uses UART (Universal Asynchronous Receiver-Transmitter) serial communication for transmitting and receiving data, simplifying interfacing with microcontrollers and other devices.

- AT Commands: Configuration and control of the HC-05 module are managed via AT commands, which allow for easy setup and customization of the module's functionality.

- Applications: Commonly used in applications requiring wireless serial communication, including robotics, IoT devices, home automation systems, and other projects needing Bluetooth connectivity.

- Power Requirements: The HC-05 module typically operates within a voltage range of 3.6V to 6V DC, making it compatible with a wide range of power sources.

- Compatibility: It's compatible with various microcontrollers and devices, offering an accessible means to enable Bluetooth connectivity for different projects.

- Ease of Use: Considered user-friendly, the module is straightforward to set up, and its AT commands provide a simple way to configure the module according to specific requirements.

- Reliability and Performance: Provides a reliable means of establishing a wireless connection and offers decent performance for short-range Bluetooth communication needs.

The HC-05 Bluetooth module is a popular and versatile choice for enabling Bluetooth communication in various electronic projects, offering simplicity, reliability, and compatibility with a broad array of devices and microcontrollers.

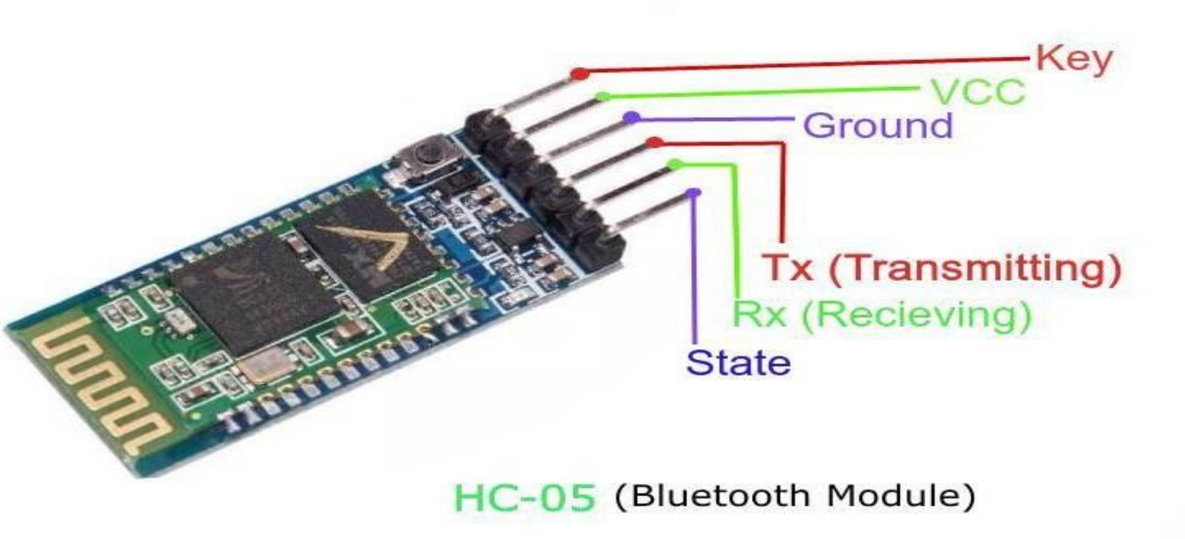


Figure 6. Bluetooth Module

4.4.7 - ESP8266 NodeMCU

The ESP8266 NodeMCU is a versatile and widely used development board that integrates the ESP8266 Wi-Fi-enabled microcontroller with the NodeMCU firmware, offering a convenient platform for IoT and embedded projects. The ESP8266 microcontroller at the core of the NodeMCU provides cost-effective Wi-Fi connectivity, enabling it to be utilized in various IoT applications. NodeMCU firmware, based on Lua scripting language, simplifies programming and serves as a user-friendly environment for IoT development, aiding quick prototyping. The NodeMCU development board consolidates the ESP8266 microcontroller with additional features such as USB connectivity, GPIO pins, analog inputs, and a voltage regulator, providing ease of use for a wide array of projects.

With its inherent Wi-Fi capabilities and multiple GPIO pins, the ESP8266 NodeMCU is well-suited for diverse IoT and embedded applications, including home automation, sensor networks, and smart device development. Its straightforward development environment encourages rapid prototyping and facilitates the creation and testing of IoT applications.

Benefiting from a strong and active community, the ESP8266 NodeMCU board offers extensive support, tutorials, libraries, and resources, contributing to its accessibility and ease of use. As an open-source platform, developers can modify and contribute to the firmware, enhancing and expanding its capabilities, and fostering innovation in the field of IoT and embedded systems.

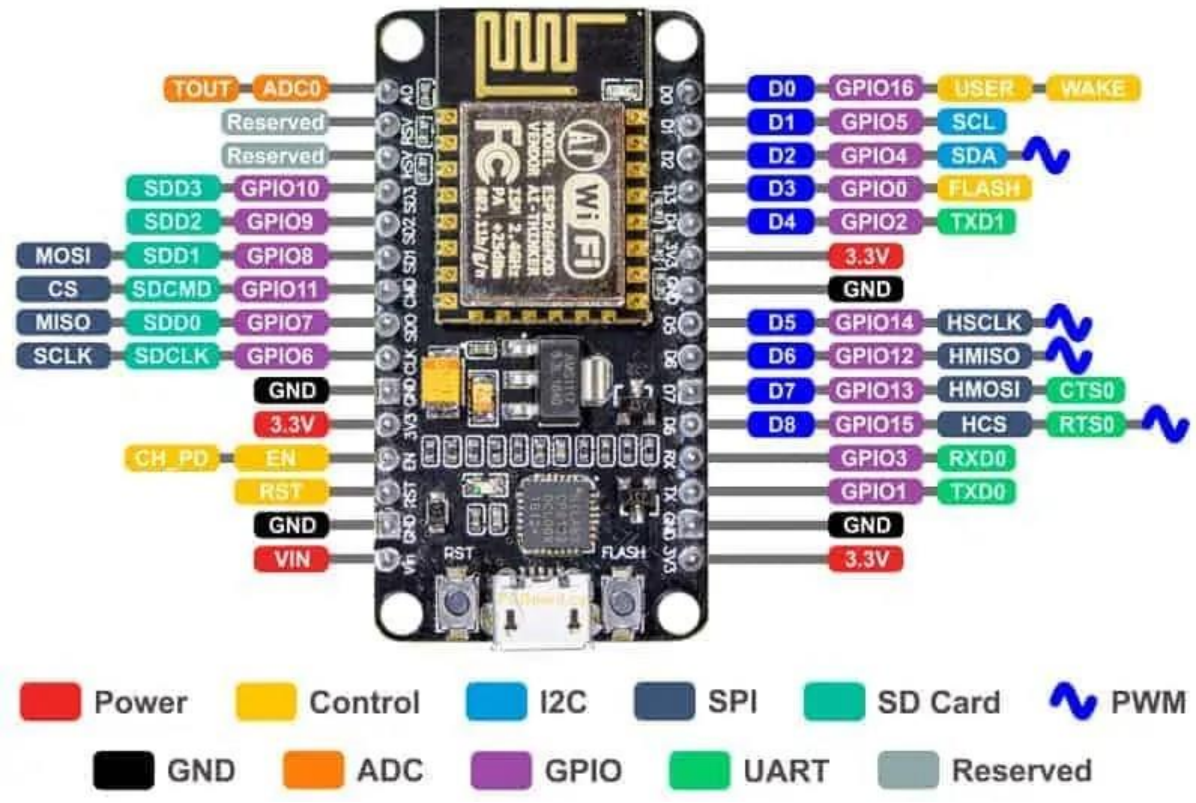


Figure 7. ESP8266 Pinout

4.5 - Hardware Peripherals

4.5.1 - RCC Peripheral

The Reset and Clock Control (RCC) peripheral in the STM32F401CCU6 microcontroller is a pivotal component responsible for managing the clock system and controlling the reset behavior of the microcontroller. It plays a central role in configuring and distributing clock signals to the microcontroller's internal modules and peripherals. The RCC selects the appropriate clock sources, such as internal oscillators, external crystal oscillators, and the phase-locked loop (PLL), ensuring the system operates at the required clock frequency while maintaining stability.

During system startup or after a reset event, the RCC initializes and configures the clock settings, guaranteeing a reliable and consistent startup sequence for the microcontroller and its peripherals. Additionally, it enables power optimization by activating only essential peripherals, thus managing power consumption effectively. The RCC's control over the reset behavior ensures a controlled sequence during startup or reset events, supporting system stability. This peripheral's configurable nature, managed through register settings, offers developers the flexibility to tailor clock settings according to specific application requirements. Overall, the RCC is crucial in governing the clock system, managing reset behavior, and facilitating power-efficient operation within the STM32F401CCU6 microcontroller, ensuring stability and control over its connected peripherals.

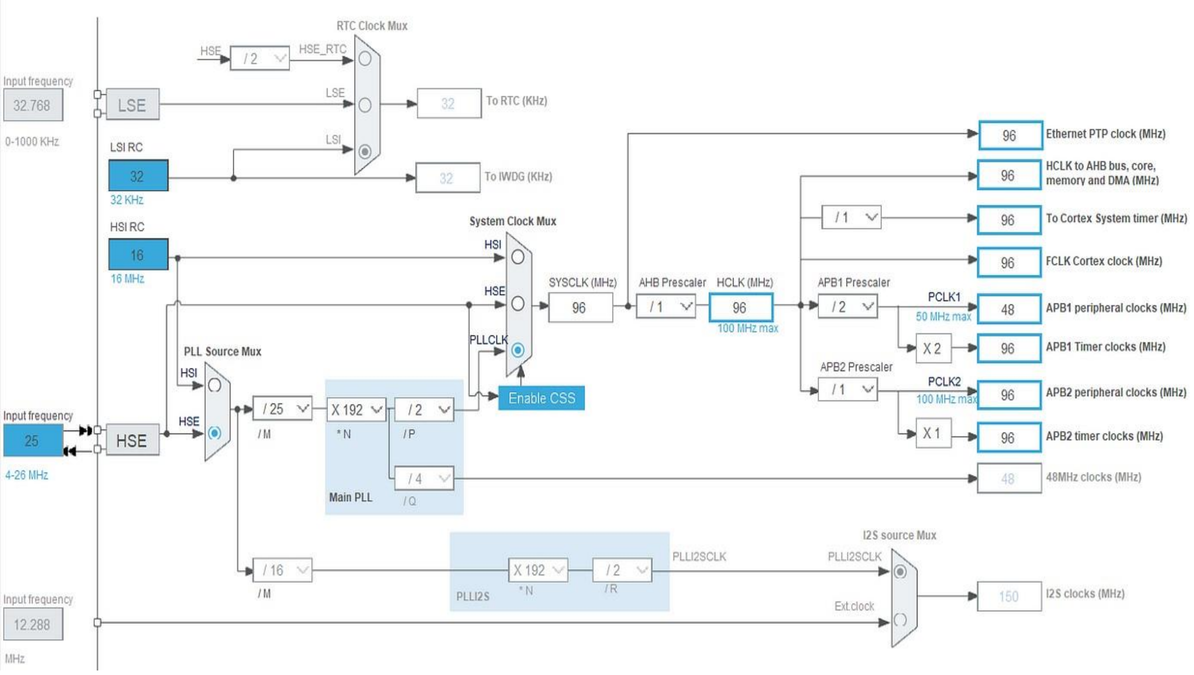


Figure 8. Rcc Peripheral Clock Tree

4.5.2 - GPIO peripheral

The STM32F401CCU6's GPIO pins serve as the bridge between the microcontroller and the outside world, enabling precise control and data acquisition. Their versatility facilitates the microcontroller's integration into a wide array of devices, from simple sensors to complex actuators. These pins accommodate various voltage levels, making them compatible with a diverse range of electronic components. Their adaptability and responsiveness empower the microcontroller to interact with, monitor, and influence its environment. In summary, the GPIO pins within the STM32F401CCU6 microcontroller form the cornerstone of its connectivity and functionality within embedded systems, enabling efficient and dynamic interactions with external devices

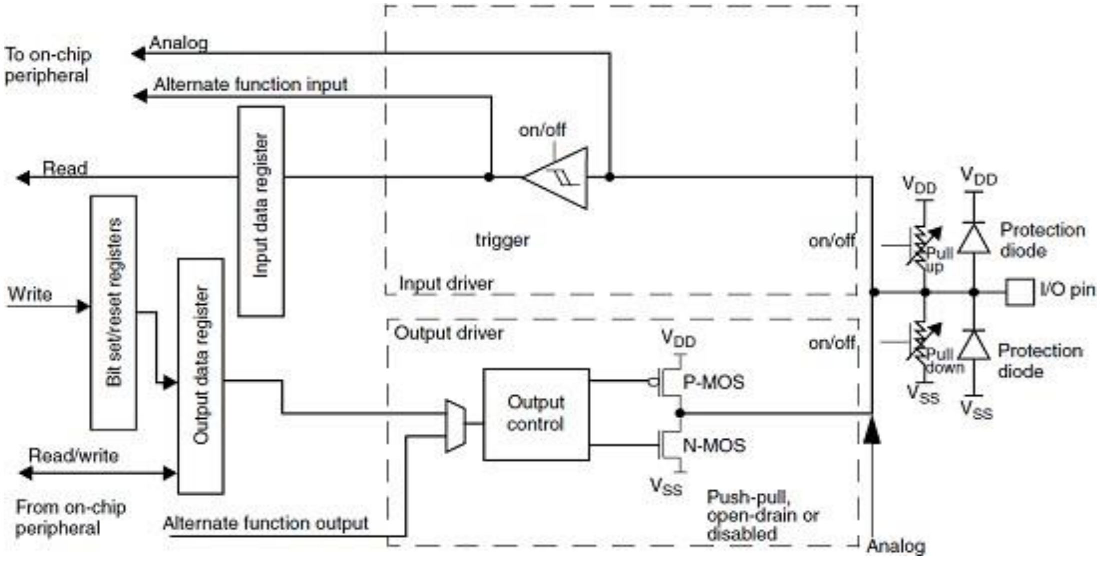


Figure 9. GPIO Block Diagram

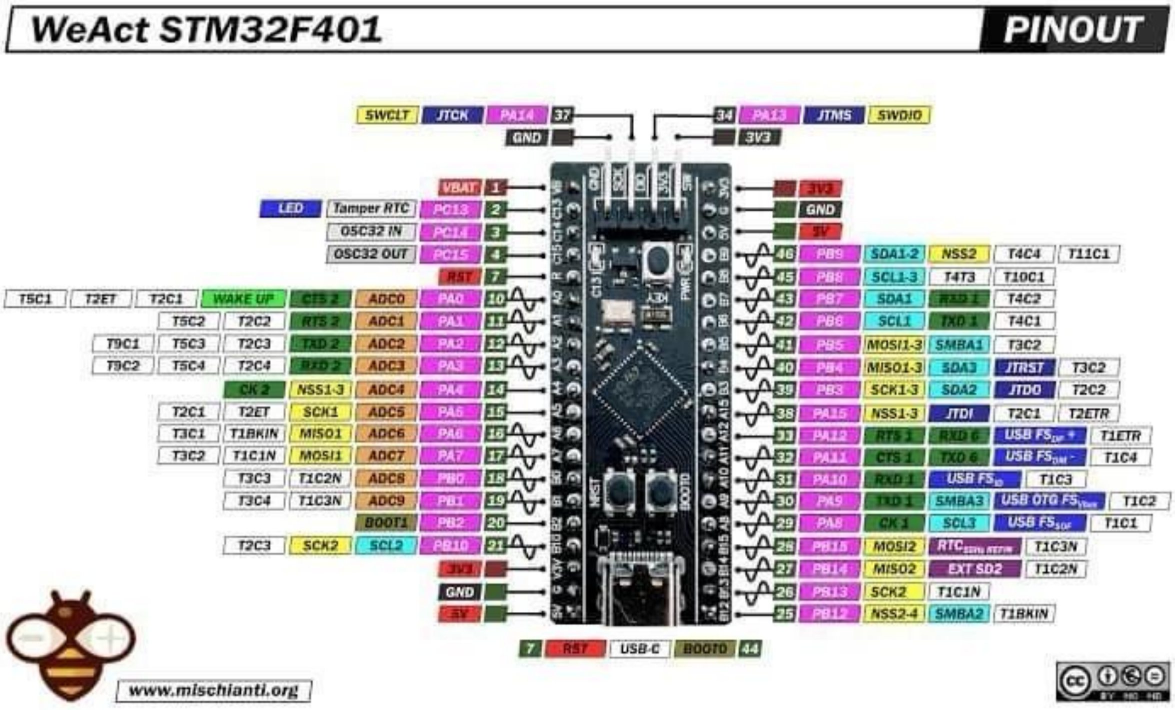


Figure 10.GPIO Pinout For Stm32f401ccu6

4.5.3 - NVIC Core Peripheral

The Nested Vectored Interrupt Controller (NVIC) is a crucial component embedded within ARM Cortex-M microcontrollers, such as the STM32F401CCU6. It is responsible for overseeing the handling of interrupts in the microcontroller.

The NVIC effectively manages interrupt requests by prioritizing and controlling their execution. It allows the microcontroller to manage simultaneous interrupt requests and ensures that high-priority interrupts take precedence over lower-priority ones, allowing for the timely handling of critical tasks. By controlling the interrupt status and enabling or disabling interrupts, the NVIC provides a means to streamline the microcontroller's response to various events.

This functionality ensures efficient and organized interrupt handling, contributing to the system's responsiveness. In summary, the NVIC is an essential component within Cortex-M-based microcontrollers, orchestrating the orderly handling of interrupts, thereby contributing to the smooth and efficient operation of the microcontroller in responding to different events.

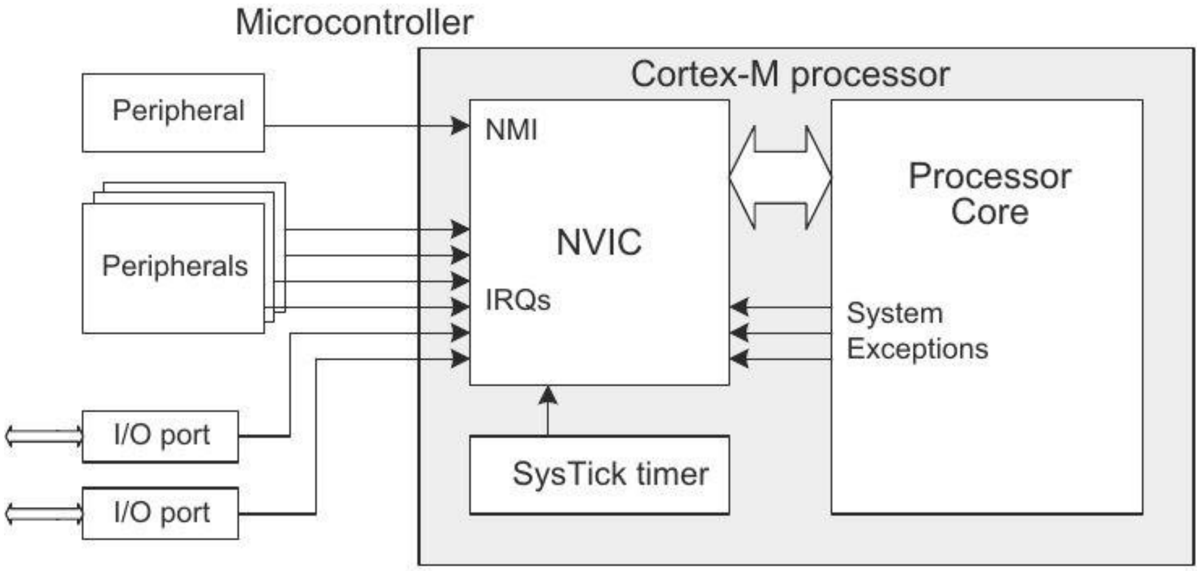


Figure 11.NVIC Block Diagram

4.5.4 - SYSTICK Core Peripheral

The SysTick is a timer integrated into ARM Cortex-M microcontrollers, including the STM32F401CCU6. It serves as a system timer and provides a time reference for the microcontroller.

SysTick generates a periodic timer interrupt, enabling timing operations and serving as a time base for the system. It can be configured to produce interrupts at regular intervals, providing a precise time reference for various tasks within the microcontroller.

This system timer is utilized for implementing time delays, system scheduling, and time measurement. It plays a significant role in managing time-sensitive operations, ensuring accurate timing for various functions and processes.

Developers can set the SysTick timer to count down from a specific value and generate an interrupt when the count reaches zero. This interrupt allows the microcontroller to execute predefined actions at regular intervals.

The SysTick timer also facilitates the implementation of real-time operating systems (RTOS) and task scheduling, allowing for efficient multitasking within the microcontroller.

In summary, the SysTick timer in Cortex-M microcontrollers like the STM32F401CCU6 serves as a fundamental timing reference, enabling precise timekeeping, time-sensitive operations, and multitasking functionalities within the system.

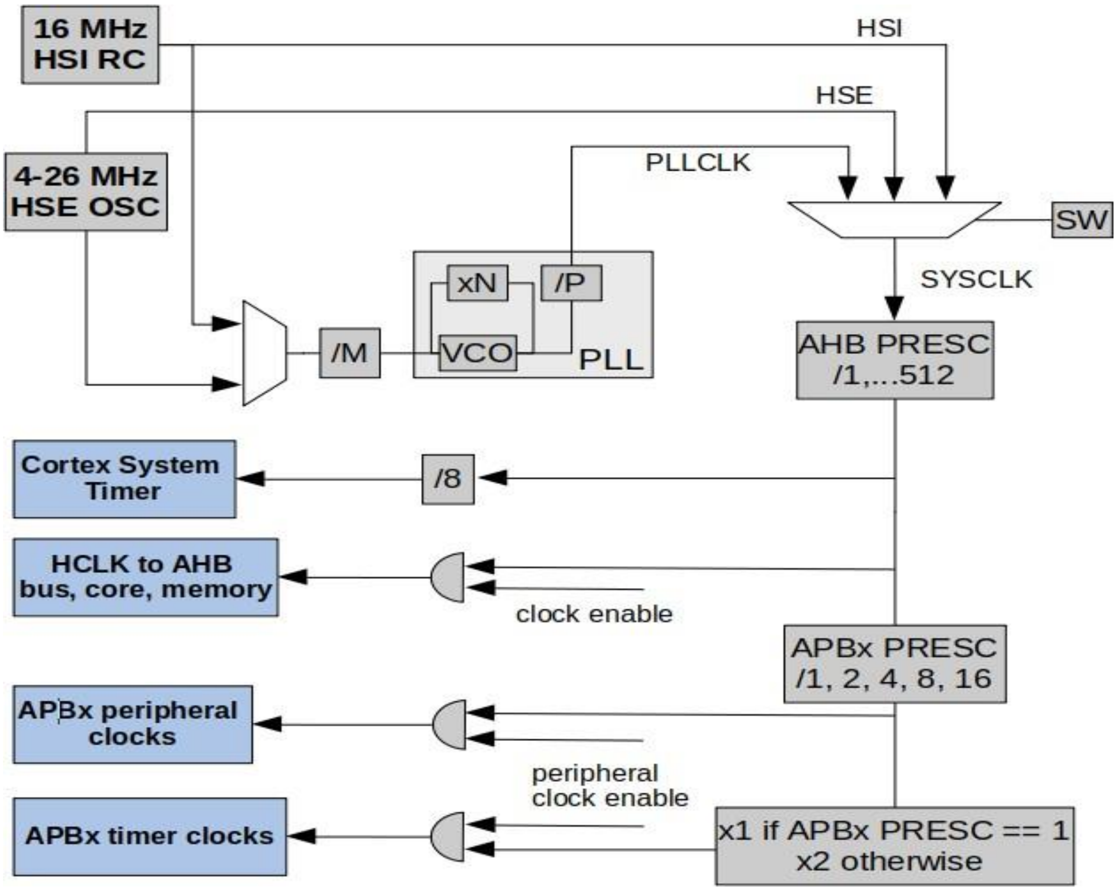


Figure 12.STK Block Diagram

4.5.6 - TIMER Peripheral

Timers, pivotal peripherals within microcontrollers such as the STM32F401CCU6, serve as hardware modules designed for time-related operations and functions. These timers facilitate a spectrum of timekeeping tasks, allowing for accurate time delays, interval measurements, and the initiation of actions based on predefined temporal conditions.

Incorporating diverse timer types like basic/general-purpose timers, advanced control timers, PWM timers, and watchdog timers, microcontrollers provide tailored functionalities to address specific applications. Timers operate by counting clock cycles or external events, enabling precise time measurement and interval control for a multitude of embedded system operations.

Their programmable nature allows developers to configure essential parameters including clock sources, prescaler settings, counter periods, and operating modes, ensuring adaptable and accurate timing functionalities. Additionally, timers can generate interrupts at predetermined intervals or when reaching specific counts, enabling microcontrollers to execute tasks precisely and timely.

With features supporting functions like PWM signal generation, timers play a critical role in motor speed control, LED brightness modulation, and other analog control applications. They are indispensable in real-time systems, guaranteeing timely task execution and accurate timekeeping for diverse embedded applications, contributing significantly to the efficient operation of microcontroller-based systems.

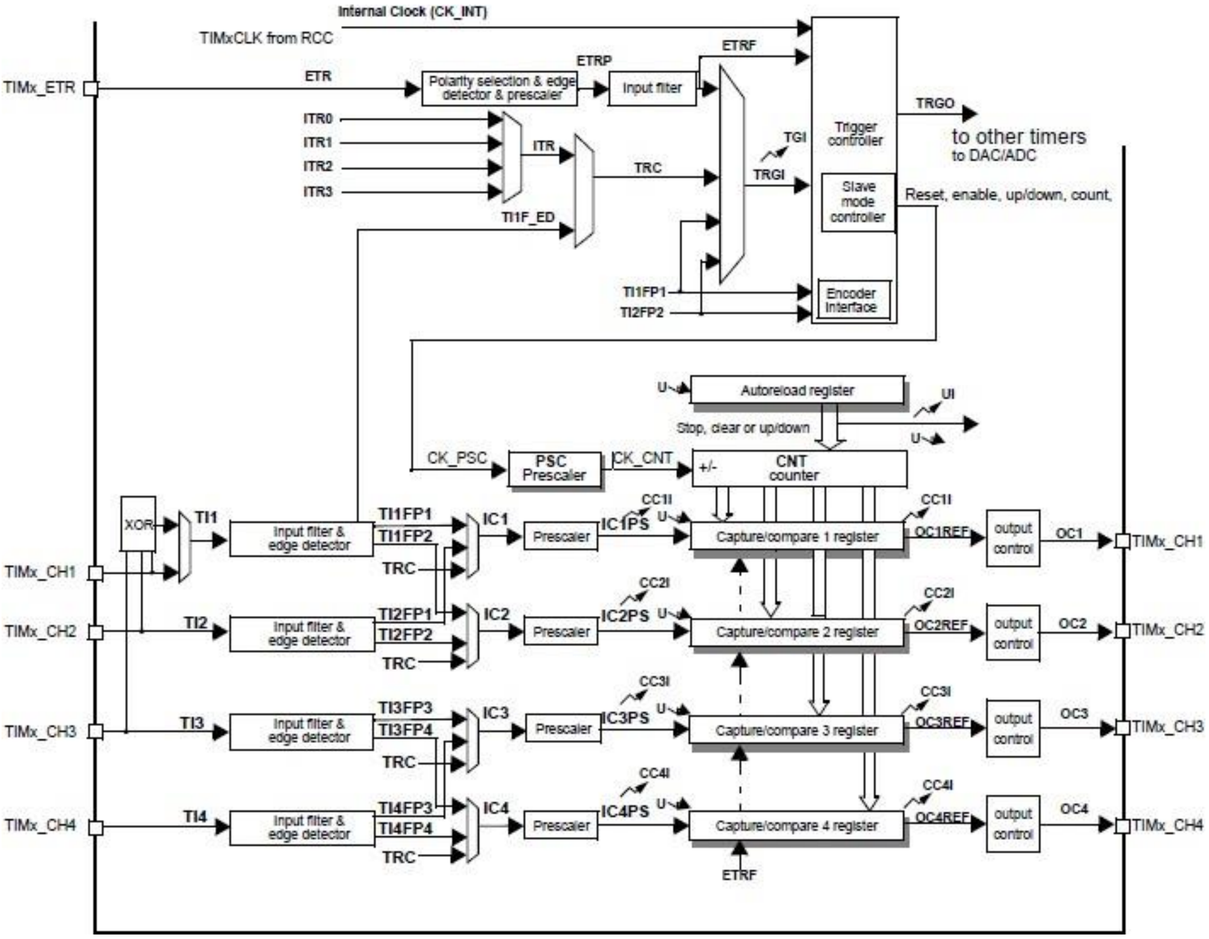


Figure 13.Timer Block Diagram

4.5.7 - UART Peripheral

The Universal Asynchronous Receiver-Transmitter (UART) stands as a foundational communication interface within microcontrollers like the STM32F401CCU6, enabling serial data exchange between devices. This communication system operates asynchronously, meaning it doesn’t require a shared clock signal. Utilizing separate transmission (TX) and reception

(RX) lines, UART allows the transmission of data through distinct bits comprising a start bit, variable data bits, an optional parity bit for error-checking, and a stop bit to maintain data integrity. The UART communication speed, measured in baud rate, determines the rate at which data is transmitted. For successful data exchange, both communicating devices must be configured to match the same baud rate. With its interrupt-driven communication method, UART efficiently manages data flow, enabling the microcontroller to respond promptly to incoming and outgoing data without continuous polling.This versatile communication protocol finds extensive use in interfacing with sensors, wireless modules, debugging, and establishing connections between microcontrollers and peripheral devices. Microcontrollers typically offer multiple UART interfaces with configurable settings, providing adaptability to tailor communication parameters according to specific application needs. As a widely supported and standard serial communication protocol, UART remains a popular choice in embedded systems, offering a reliable, adaptable, and widely compatible means of data exchange.

USART block diagram

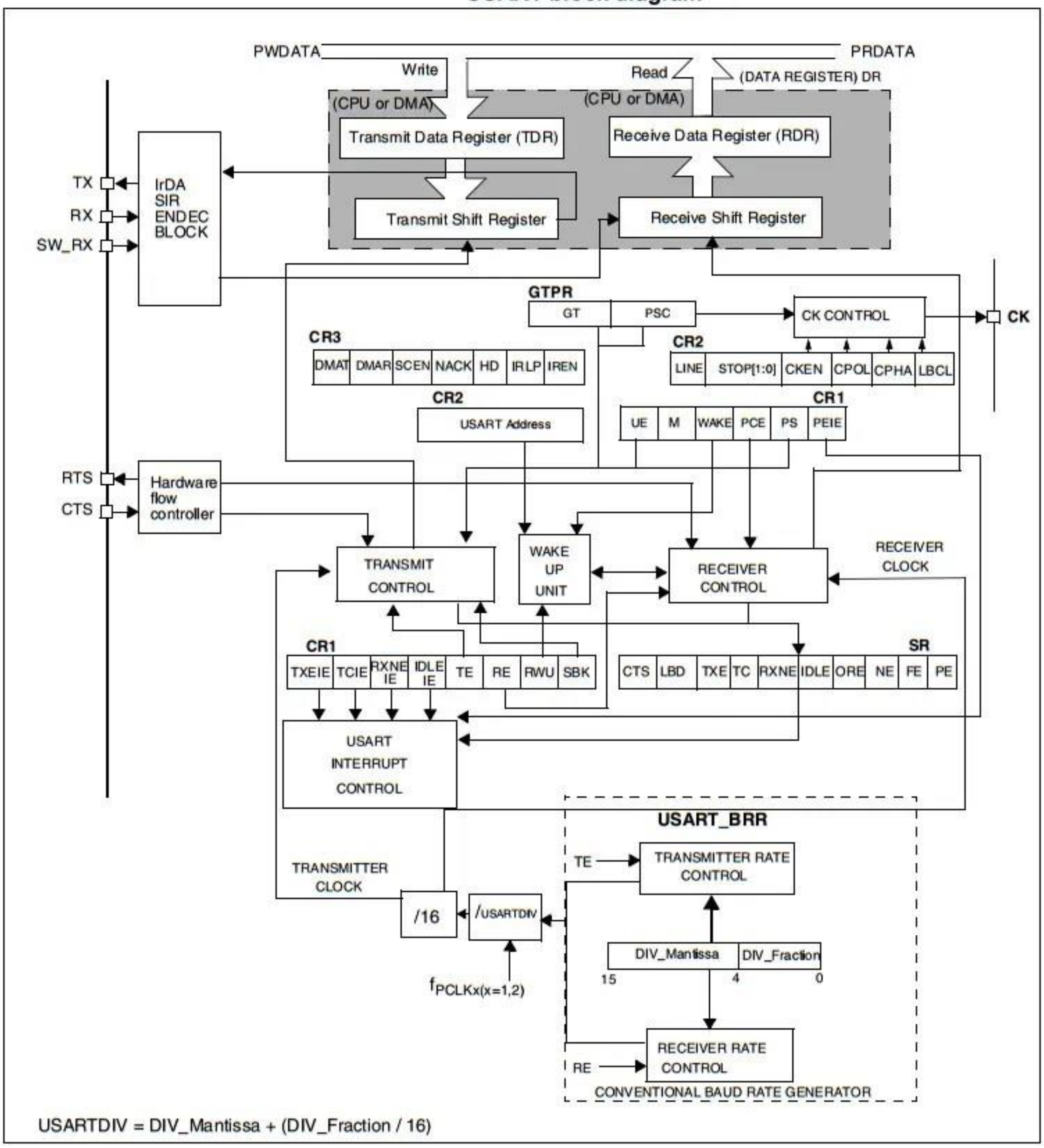


Figure 14.UART Block Diagram

4.5.8 - Flash Driver Peripheral

The Flash Memory Controller, acting as the Flash Driver peripheral, is an essential component embedded within microcontrollers such as the STM32F401CCU6. This hardware unit manages critical functions related to the integrated Flash memory. It facilitates the reading, writing, and erasing of data within the microcontroller's non-volatile memory. The Flash Driver oversees program and erase operations, enabling the microcontroller to write and retain data, store program code, and preserve configuration settings within the Flash memory. Its functionality extends to organizing memory into sectors or pages, ensuring precise sector/page erase and write operations. This granularity enhances control and efficiency in managing data within the Flash memory. Its operations ensure data integrity during write and erase processes, upholding the reliability of the stored information. The Flash Driver's programmable features provide configurable options, allowing tailored control over the Flash memory operations to align with the application's specific needs. Additionally, certain Flash Drivers may incorporate security features to protect sensitive data by implementing access restrictions or secure memory access mechanisms. Some Flash Drivers also include error correction and recovery features, aiding in error detection and rectification during memory operations. Crucially, the Flash Driver plays a vital role in facilitating system firmware upgrades, enabling the microcontroller to update its software, ensuring flexibility and adaptability within the embedded system.

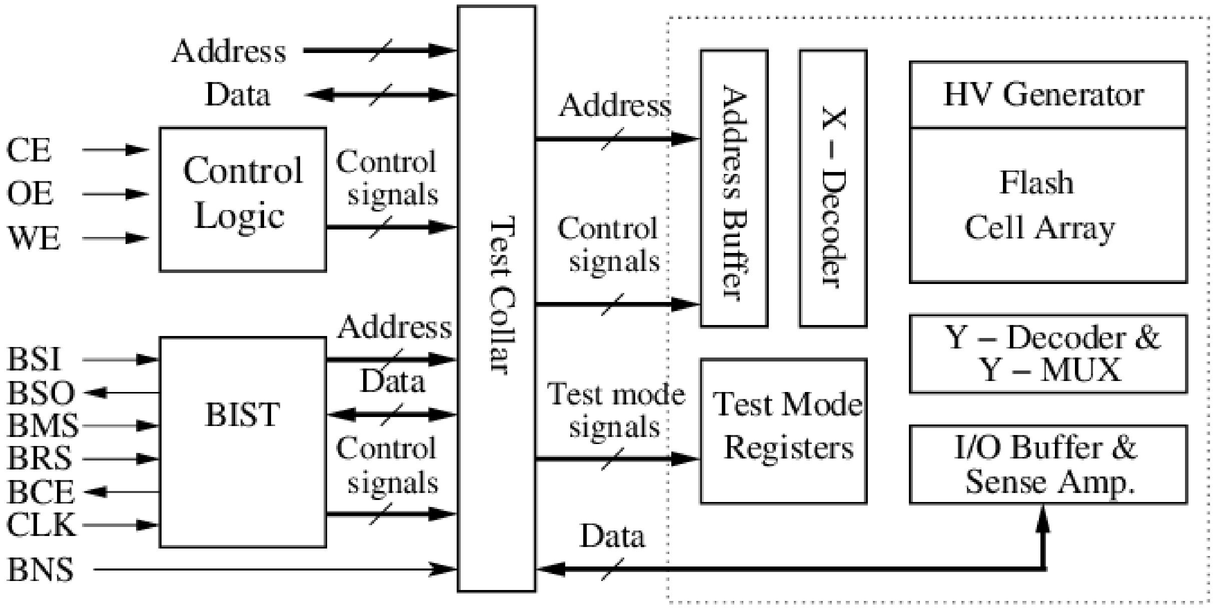


Figure 15.Flash Memory Block Diagram

4.6 - Software Requirement

Within our application layer, our software module architecture comprises the following components:

1. Body Control Module (BCM):
Acting as the central control unit, the Body Control Module oversees and manages the functionalities of the various sub-modules within the application layer.

2. Normal Cruise Control Module (NCC):
A dedicated sub-module responsible for implementing the standard cruise control feature, allowing the vehicle to maintain a constant speed set by the driver.

3. Adaptive Cruise Control Module (ACC):
Another specialized sub-module, the Adaptive Cruise Control module, enhances driving dynamics by dynamically adjusting the vehicle's speed based on the distance to the vehicle ahead.

4. Lane Keep Assist Module (LKA):
Focused on augmenting road safety, the Lane Keep Assist module detects and corrects deviations from the lane, contributing to a more secure driving experience.

5. Automatic Emergency Brake Module (AEB):
This sub-module is designed to enhance safety by detecting potential collisions and autonomously applying the brakes if necessary, mitigating the impact of emergencies.

6. Firmware Over-the-Air (FOTA) enables remote updates of a device's software. It allows for seamless upgrades without physical access, enhancing device functionality and security. Widely used in devices like vehicles and smartphones, FOTA ensures efficient and convenient software maintenance.

By creating a comprehensive main module, BCM, and organizing functionalities into distinct sub-modules (NCC, ACC, LKA, AEB), our software architecture follows a modular approach. The BCM acts as the orchestrator, effectively managing and coordinating the operations of each sub-module, ensuring seamless integration and efficient functionality within the application layer. This modular design enhances scalability, maintainability, and overall system flexibility.

Now, let's delve into an in-depth discussion about each module, elucidating their functionalities, design intricacies, and operational processes. We will provide comprehensive insights by incorporating flowcharts and state machines to illustrate the intricate interactions of each module. Additionally, we'll outline the specific hardware components utilized by each module, offering a holistic understanding of the interplay between software and hardware within our system architecture.

4.6.1 - Body Control Module (BCM)

- Main Functionality:

  The Body Control Module (BCM) is like the brain of our software system. It's in charge of making sure all the different parts work together smoothly. Imagine it as the boss overseeing four workers: Adaptive Cruise Control (ACC), Normal Cruise Control (NCC), Automatic Emergency Brake (AEB), and Lane Keep Assist (LKA). The BCM has specific jobs. It uses smart plans (algorithms) to manage how fast the car goes with ACC, keep a steady speed with NCC, hit the brakes in an emergency with AEB, and stay in the correct lane with LKA. Think of it as making sure everyone in the carpool is on the same page! One of its important skills is talking to all these workers (sub-modules) to make sure they share information and work together in real time. It's also really good at adapting to changes on the road, like slowing down when needed or making quick decisions to keep everyone safe. The BCM doesn't just think; it also talks to the car's parts, like telling the engine when to speed up or slow down. Safety is a big deal for the

BCM – it's always checking for any issues and making sure everything is working well. In a nutshell, the BCM is the smart leader, making sure all the cool features like ACC, NCC, AEB, and LKA work together seamlessly, adapting to the road, and keeping everyone safe on the journey.

- Hardware Component:

  The Body Control Module (BCM) doesn't use any physical hardware itself; it's purely a software component that oversees and manages the entire system. Think of it as the conductor of an orchestra, directing all the different instruments (hardware components) without being an instrument itself. The BCM uses its algorithms and intelligence to coordinate and control the various features in the system without needing any specific hardware equipment.

- Module States:

  - ABCM_CAR_ON:

    When the car is in this state, it is fully turned on, and the driver can use it normally. Adaptive Cruise Control (ACC) or Normal Cruise Control (NCC) can be activated. Additionally, the Automatic Emergency Brake (AEB) and Lane Keep Assist (LKA) work automatically once the car is turned on.

- ABCM_CAR_NCC_ACTIVE:
  This state signifies that the driver wants to activate Normal Cruise Control (NCC), and the car transitions to this state to initiate the NCC functionality.

- ABCM_CAR_ACC_ACTIVE:
  When the driver intends to activate Adaptive Cruise Control (ACC), the car moves to this state to initiate the ACC functionality.

- ABCM_CAR_ACC_SET:
  In this state, the BCM module scans the distance using the lidar and provides this data to the ACC module. The ACC module analyzes the information to make decisions such as adjusting speed, increasing or decreasing speed, or maintaining the current speed.

- ABCM_CAR_GET_FAULT:
  If the car receives invalid input from the driver or detects illogical actions (e.g., attempting to turn the car on twice), it enters this state to check for fault codes and notify the user of any issues.

- ABCM_CAR_NCC_OFF:
  When the driver wishes to deactivate Normal Cruise Control (NCC), the car transitions to this state to turn off the NCC functionality.

- ABCM_CAR_ACC_OFF:
  When the driver wants to deactivate Adaptive Cruise Control (ACC), the car transitions to this state to turn off the ACC functionality.

- ABCM_CAR_IDLE:
  In this state, the BCM scans a new distance using the lidar, providing this data to other modules. It also checks the car's position between lanes. If there is an assumption of an impending accident, the Automatic Emergency Brake (AEB) is activated, bringing the car to a stop and turning off the entire system.

- ABCM_UPDATE_FIRMWARE:
  When the driver wishes to update the car's firmware, the car enters this state. During this time, the car cannot be used for driving.

- ABCM_CHANGE_SPEED_LIMIT:
  If the user wants to adjust the speed limit, the BCM enters this state. It checks which system (NCC or ACC) is currently active and then communicates the order to either increase or decrease the speed limit.

- Flow Chart Diagram:

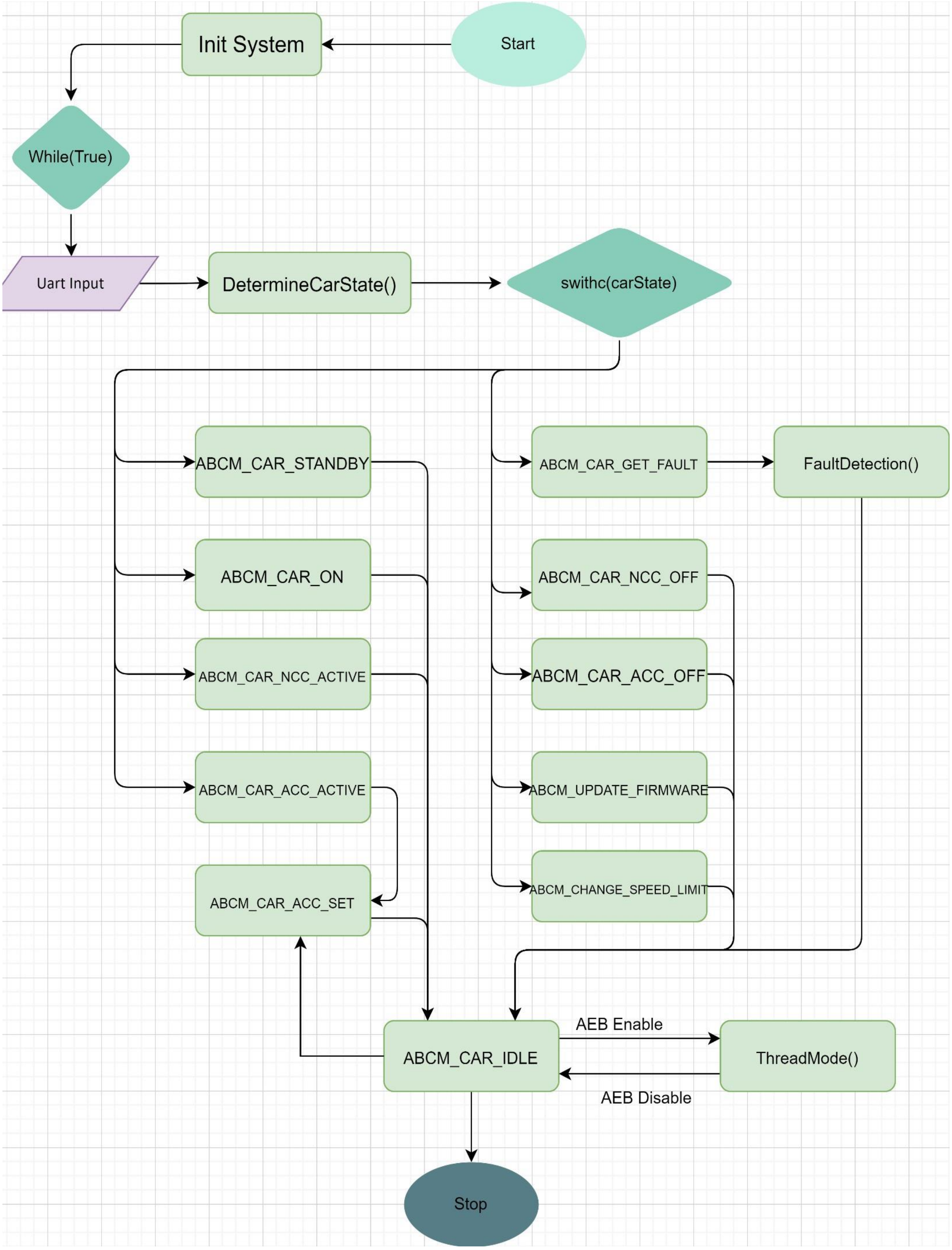


Figure 16.BCM Flow Chart Diagram

- State Machine Diagram:

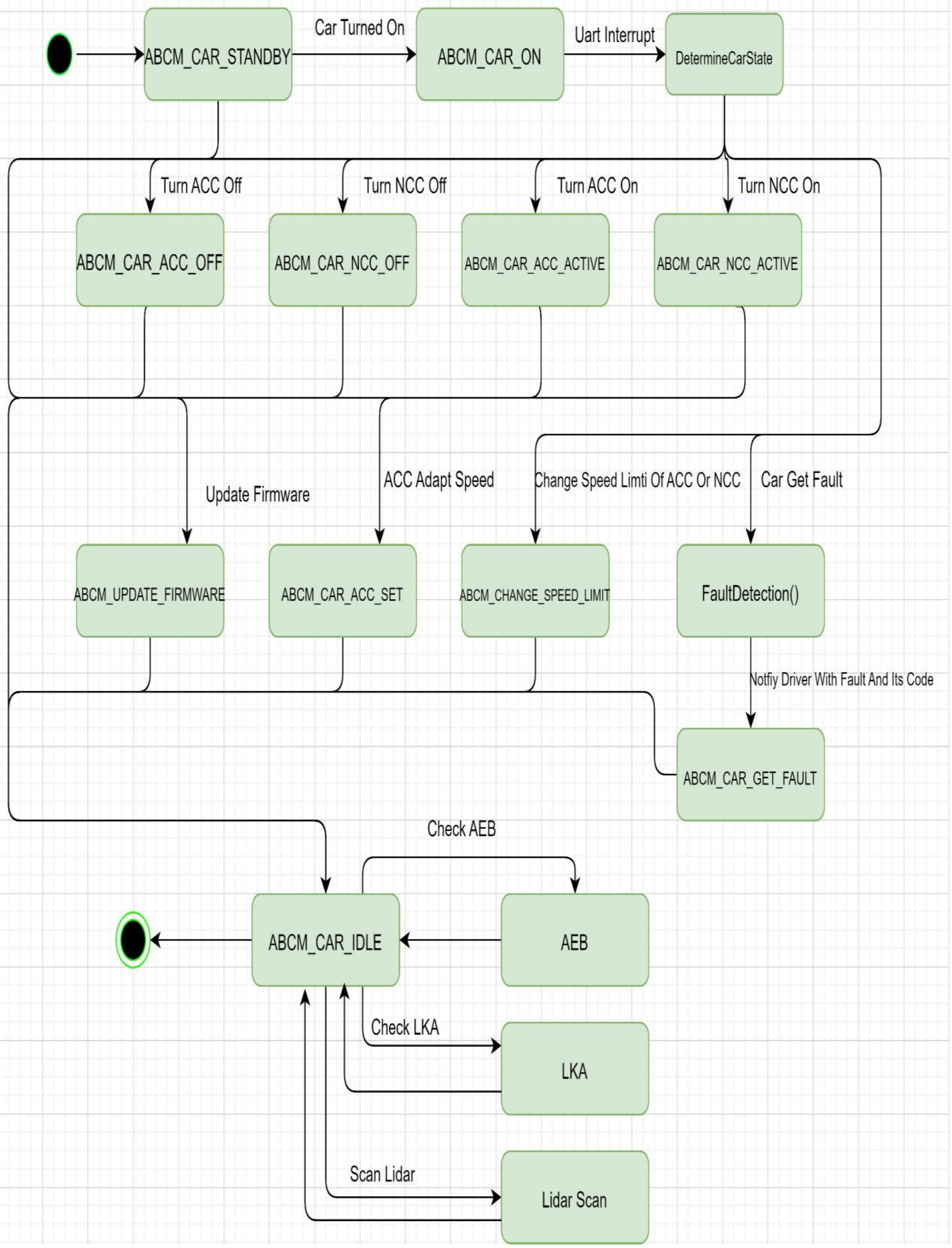


Figure 17.BCM State Machine Diagram

4.6.2 - Normal Cruise Control Module

- Overview of Normal Cruise Control (NCC):
  Normal Cruise Control is a vehicle feature designed to simplify and enhance the driving experience by maintaining a constant speed set by the driver. Once activated, NCC takes control of the throttle, allowing the driver to relax their foot from the accelerator during highway or open-road conditions. This system is particularly beneficial for extended journeys, providing a hands-free option for speed control and reducing driver fatigue.

- Advantages of Normal Cruise Control:
  The advantages of Normal Cruise Control lie in its contribution to driver convenience and overall driving efficiency. Firstly, NCC offers a more relaxed driving experience, especially on long trips, by alleviating the need for continuous foot pressure on the accelerator. This can lead to reduced driver fatigue and increased comfort during extended periods behind the wheel.
  Secondly, NCC has the potential to enhance fuel efficiency by maintaining a steady speed, and optimizing the vehicle's fuel consumption. This benefit is particularly noticeable on highways, where consistent speeds can contribute to more economical driving.

In summary, Normal Cruise Control not only simplifies the driving process but also supports a more comfortable and fuel-efficient journey, making it a valuable feature for those navigating extended stretches of road.

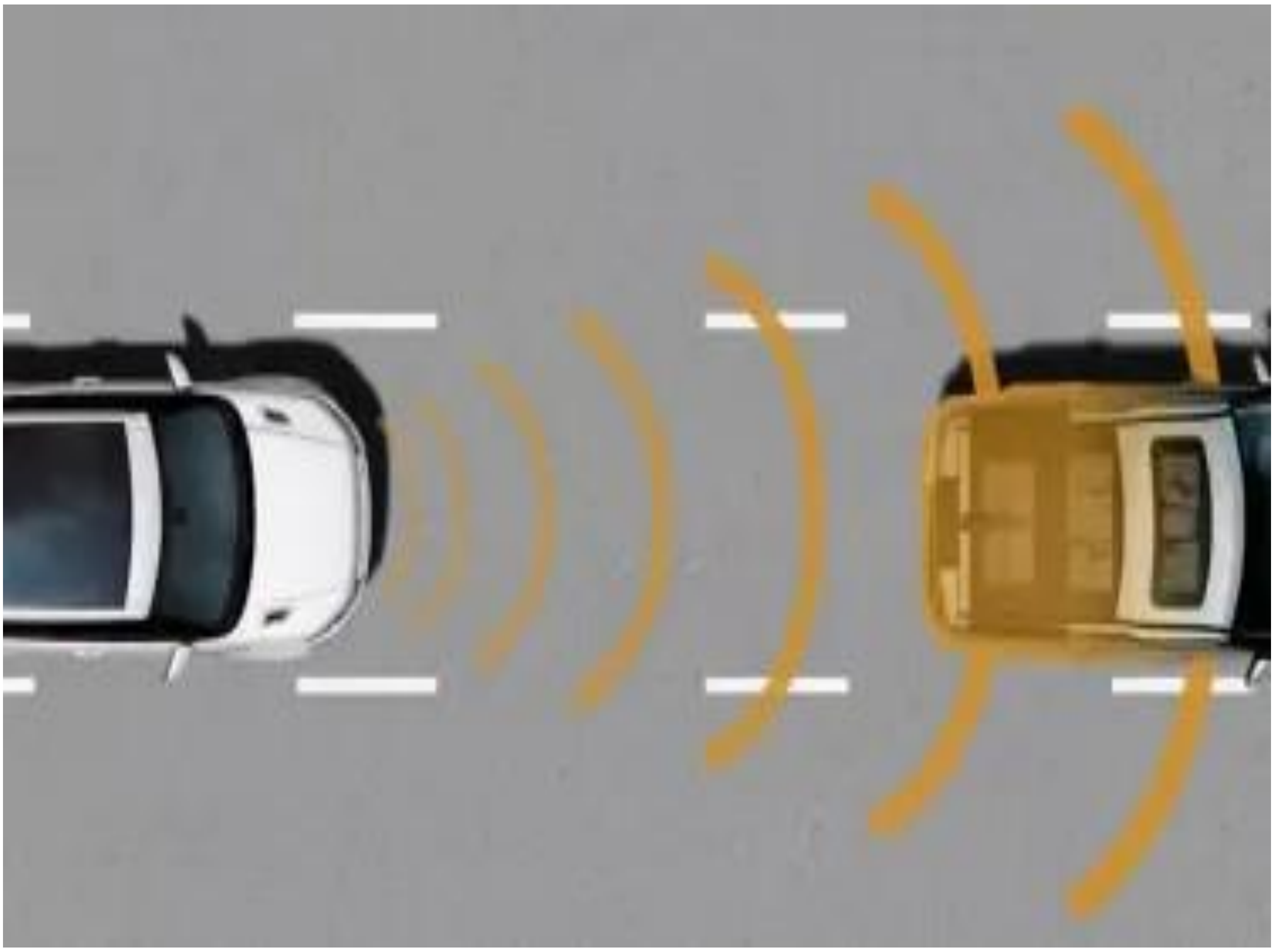


Figure 18.NCC Module Image

- Module States:
    - ANCC_START:
      In this state, the Normal Cruise Control (NCC) module initiates its functionality, activating cruise control and setting the speed limit to the default speed limit of the car. This state marks the beginning of NCC's operation.

    - ANCC_STOP:
      When in this state, the NCC module ceases its operation, bringing the cruise control functionality to a halt. The car returns to regular manual control, allowing the driver to resume full command over speed adjustments.

    - ANCC_CHANGE_SPEED_LIMIT:
      Within this state, the NCC module facilitates a change in the speed limit based on the driver's preference. The driver can choose to increase or decrease the speed limit, allowing for flexible adjustments to the cruise control setting.

- NCC Flow Chart Diagram:

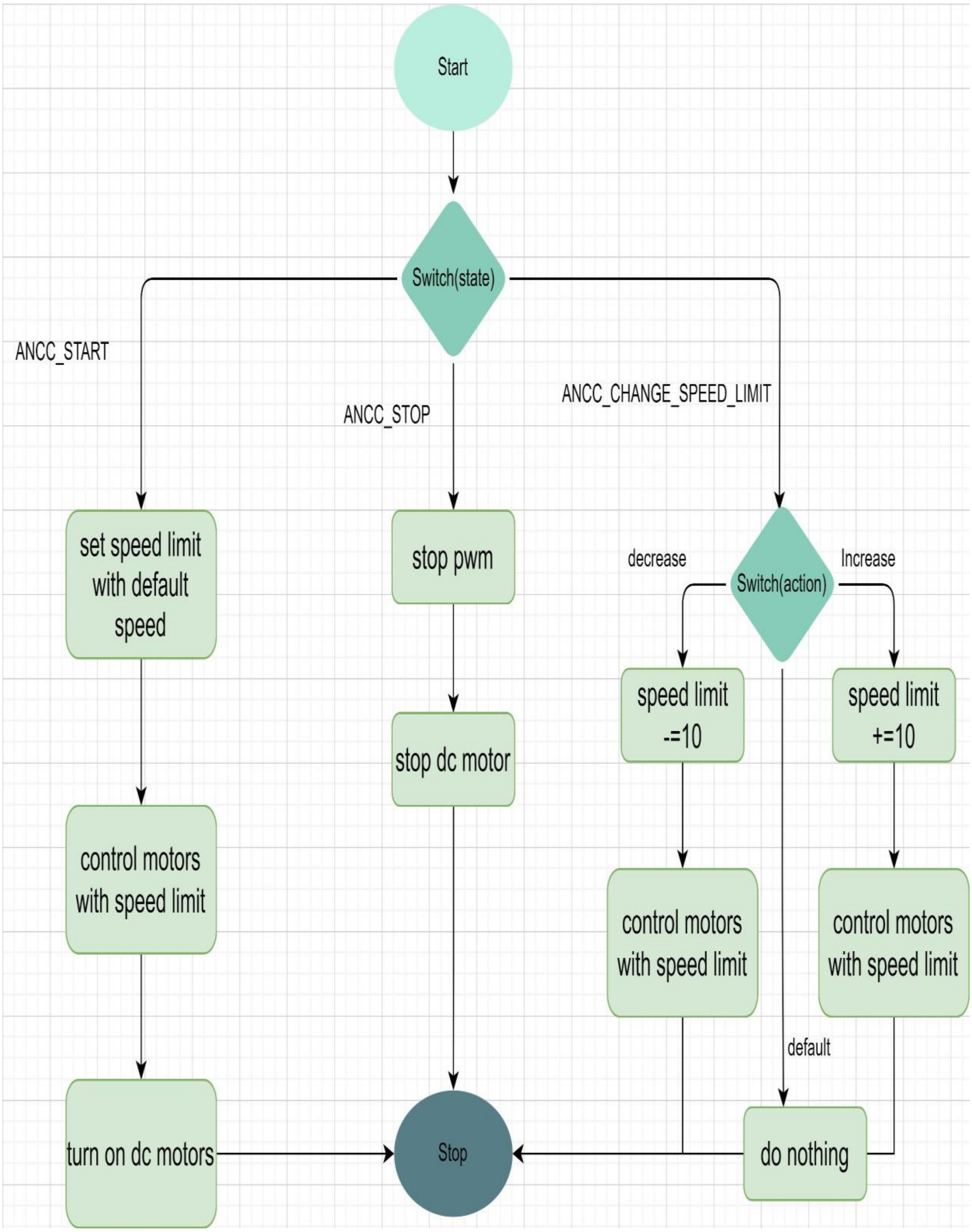


Figure 19.NCC Flow Chart Diagram

- NCC State Machine Diagram:

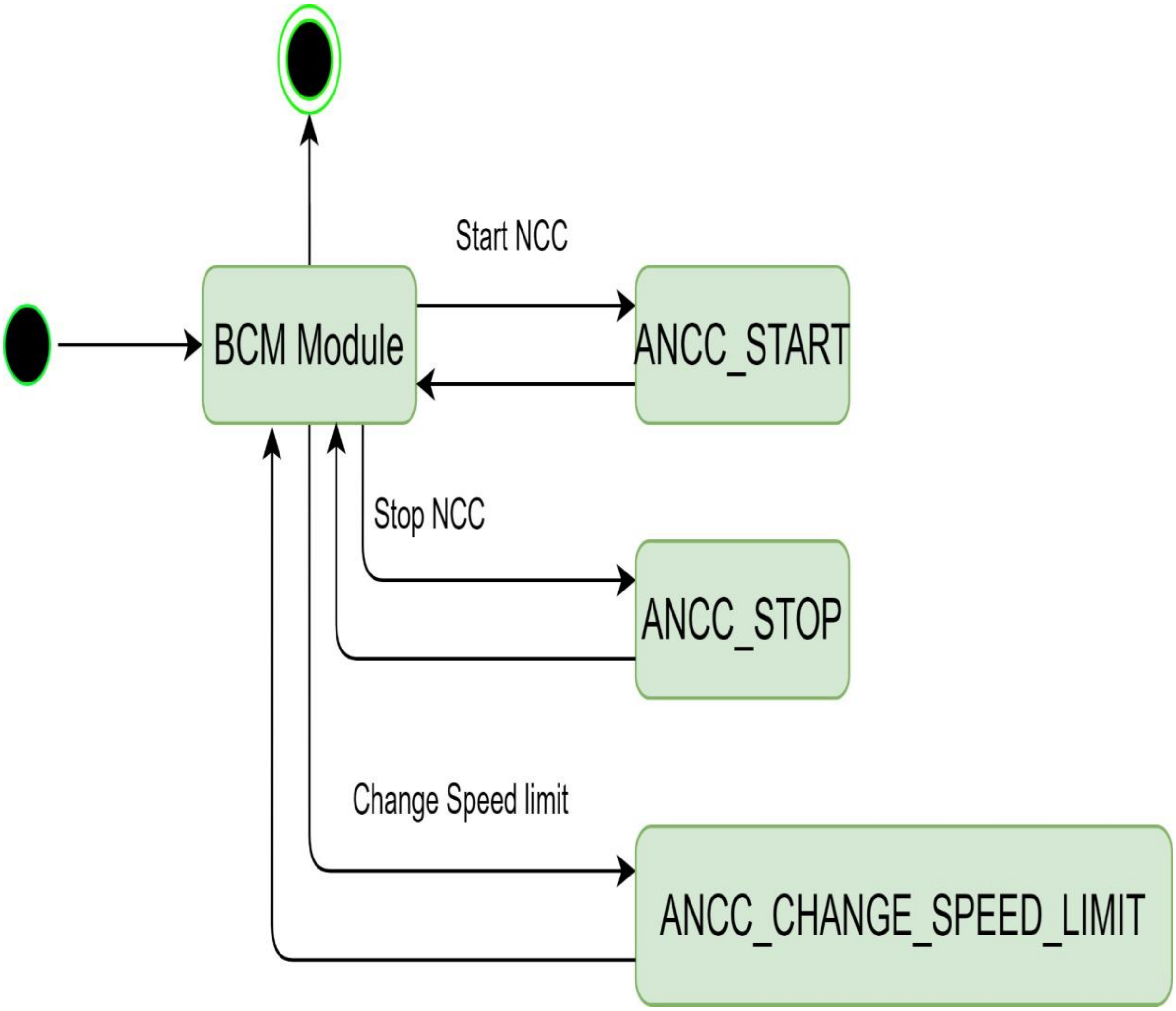


Figure 20.NCC State Machine Diagram

4.6.3 - Adaptive Cruise Control Module:

- Overview of Adaptive Cruise Control (ACC):
  Adaptive Cruise Control (ACC) is a cutting-edge driver assistance system designed to enhance vehicle safety and driving comfort. This advanced system goes beyond traditional cruise control by incorporating intelligent sensors, such as radar or lidar, to actively monitor traffic conditions ahead. This allows ACC to autonomously adjust the vehicle's speed, ensuring a safe and preset following distance from the car in front. By combining automation with real-time awareness, ACC represents a significant step towards semi-autonomous driving capabilities, providing a more adaptive and responsive driving experience.

- Advantages of Adaptive Cruise Control:
  ACC offers several advantages that significantly enhance the driving experience. First and foremost, its traffic adaptability ensures that the system can automatically adjust the vehicle's speed based on surrounding traffic, making it particularly useful in congested or fluctuating speed zones. The system contributes to enhanced safety by actively preventing collisions and reducing the risk of rear-end accidents through its automatic speed adjustments. Moreover, ACC reduces driver fatigue by minimizing the need for constant manual speed adjustments, proving especially beneficial during extended journeys or in heavy traffic conditions. The system's convenience shines during highway driving, promoting a

more relaxed experience while also increasing fuel efficiency through optimized acceleration and deceleration

patterns. ACC often allows for user customization, aligning with individual driving preferences and styles. In summary, Adaptive Cruise Control seamlessly combines safety, convenience, and efficiency, making it a valuable feature in modern vehicles.

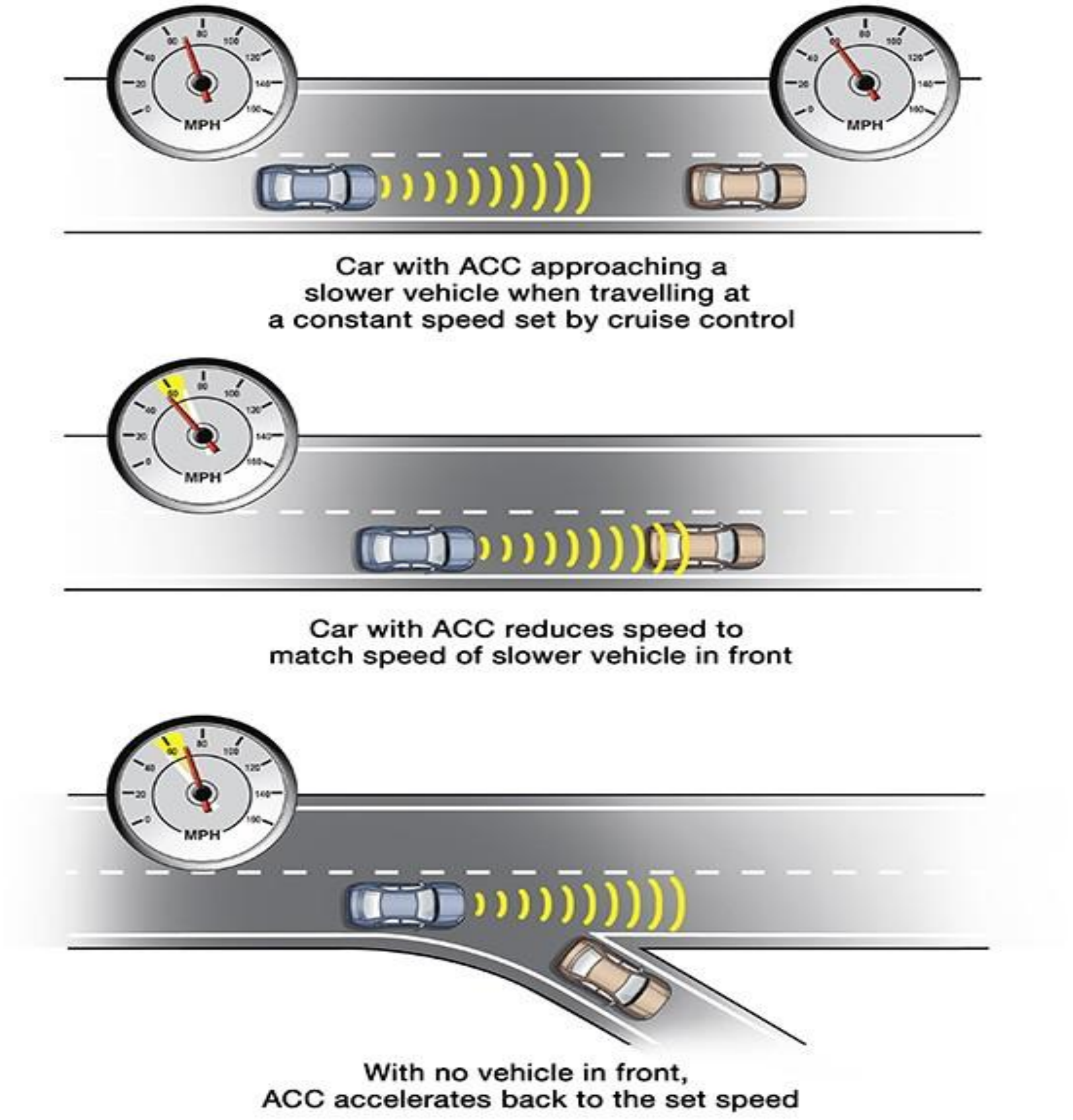

MPH
MPH
Car with ACC approaching a slower vehicle when travelling at a constant speed set by cruise control
MPH
Car with ACC reduces speed to match speed of slower vehicle in front
MPH
With no vehicle in front, ACC accelerates back to the set speed

Figure 21.ACC Module Image

- Module States:

  - AACC_START:
    In this state, the Adaptive Cruise Control (ACC) module initiates its functionality. It sets the speed limit using the default speed control, adjusts the speed of the DC motors, and subsequently activates the system to maintain the preset speed.

  - AACC_STOP:
    When in this state, the ACC module discontinues its operation. Both the ACC and DC motors are turned off, and the speed of the DC motor is set to zero, returning the vehicle to manual control.

  - AACC_CHANGE_SPEED_LIMIT:
    This state facilitates a change in the speed limit based on the driver's preference. Depending on the driver's choice, the speed limit can be increased or decreased to align with the desired driving conditions.

  - AACC_CONTROL_CAR_SPEED:
    Within this state, the ACC algorithm makes decisions based on the distance of objects detected by the lidar. If the car is within the green range, the speed is increased until it reaches the preset speed limit. If in the blue range, the speed remains unchanged. However, if in the red range, the algorithm decreases the car's speed to prevent a collision and ensure safety on the road.

- ACC Flow Chart Diagram:

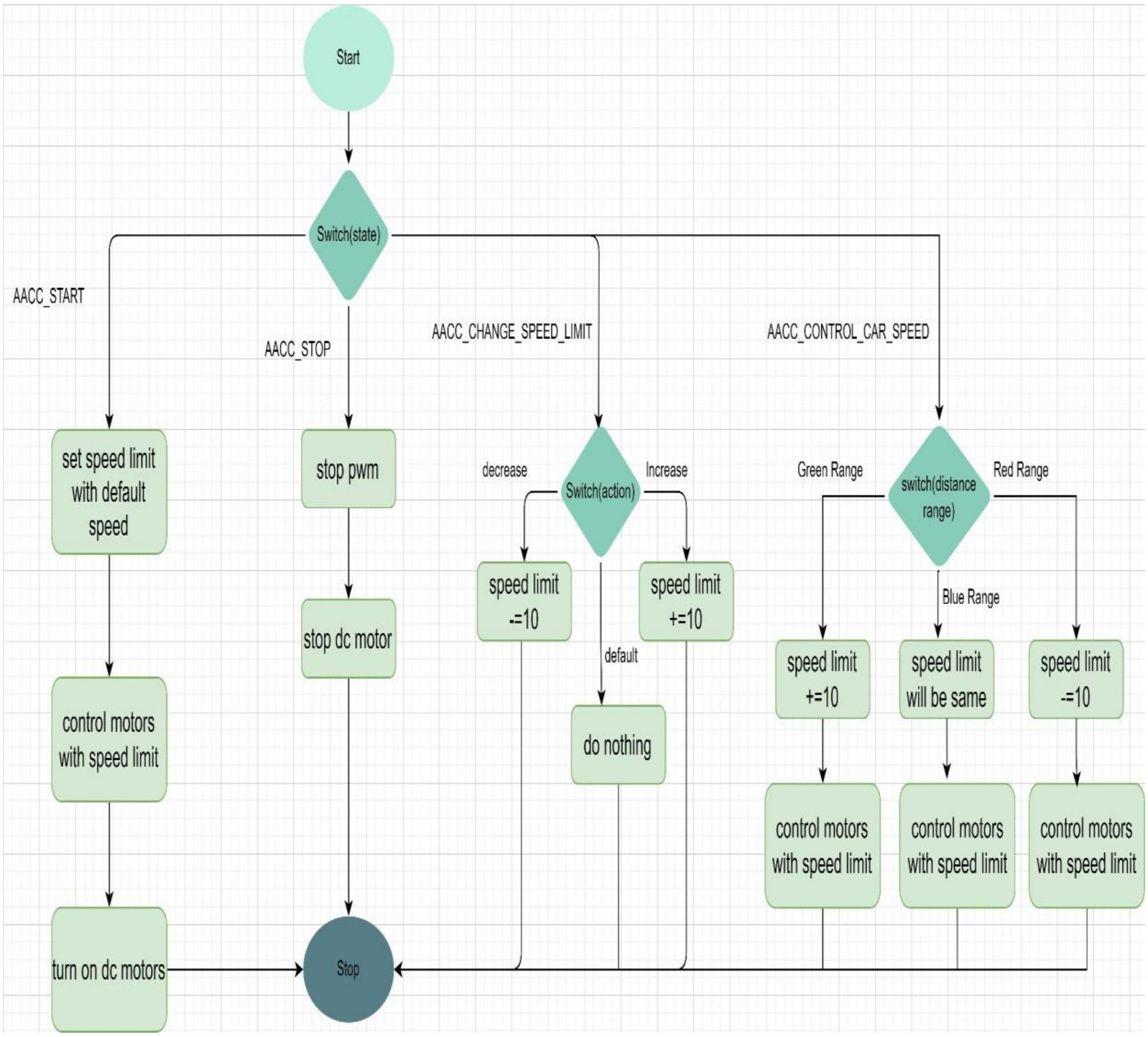


Figure 22.ACC Flow Chart Diagram

- ACC State Machine Diagram:

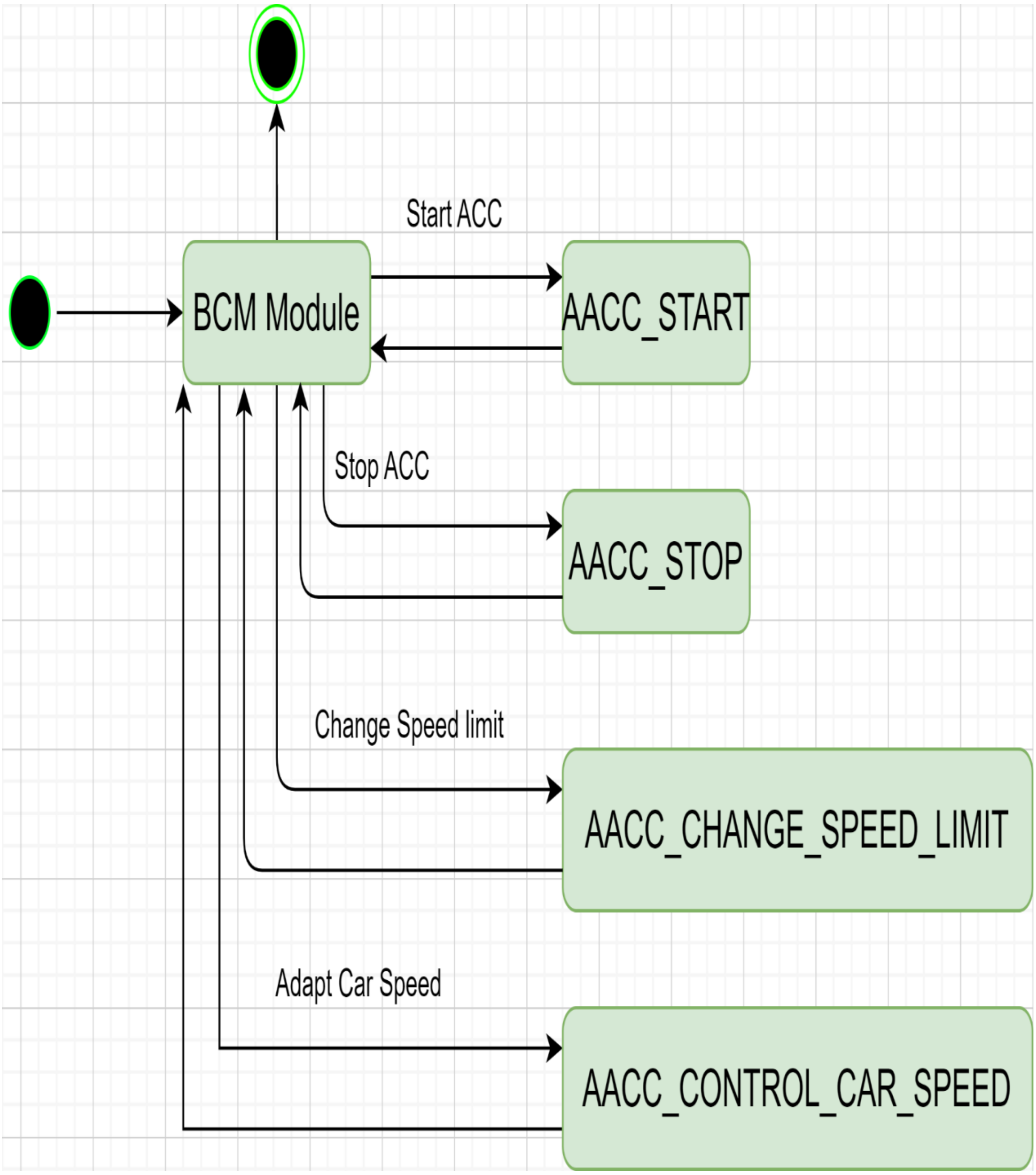


Figure 23.ACC State Machine Diagram

4.6.4 - Lane Keep Assist

- Overview of Lane Keep Assist (LKA)

  Lane Keep Assist (LKA) is a sophisticated driver assistance system designed to enhance vehicle safety by actively assisting the driver in maintaining proper lane positioning. This advanced system utilizes intelligent sensors, such as cameras or sensors, to monitor the vehicle's position within its lane. LKA intervenes when it detects unintentional lane departure, providing corrective actions to keep the vehicle within the designated lane. By combining real-time monitoring with proactive steering adjustments, LKA plays a crucial role in preventing lane drift and mitigating the risk of unintended lane departure incidents. This technology is particularly beneficial during long drives, where driver fatigue or momentary lapses in attention can lead to unintentional lane deviations.

- Advantages of Lane Keep Assist (LKA)

  Lane Keep Assist offers several advantages that contribute to overall driving safety and comfort. Foremost, it serves as a reliable safety net, intervening when the system detects signs of unintentional lane departure. By providing gentle steering corrections, LKA helps prevent the vehicle from inadvertently drifting into adjacent lanes, reducing the risk of side-swipe collisions. Additionally, LKA proves beneficial in challenging driving conditions, such as low visibility or inclement weather, where maintaining a consistent lane position can be more demanding. Moreover, this system aids in mitigating driver fatigue by actively supporting continuous lane tracking, ensuring a more secure and less strenuous driving experience. Overall, Lane Keep Assist stands as a valuable driver assistance feature, leveraging technology to enhance road safety and promote a more confident and relaxed driving environment.

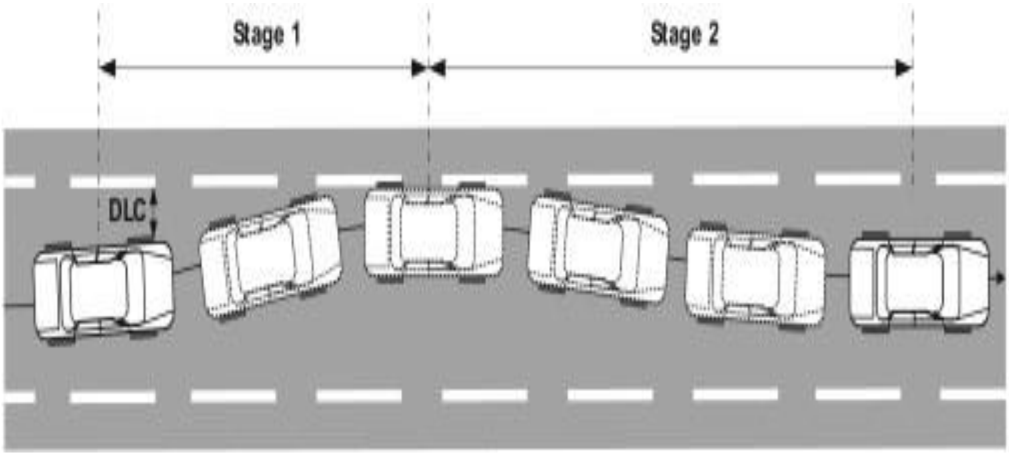


Figure 24.LKA Module Image

Figure 25.LKA Module Image

- Module States:

  - ALKA_GET_LANE_POSITION:
    During this state, the Lane Keep Assist (LKA) module employs infrared (IR) sensors to precisely determine the vehicle's lane position. These sensors assist in accurately assessing the car's location within the designated lanes, providing crucial information for subsequent lane-keeping decisions.

  - ALKA_SET_CAR_IN_LANES:
    In this state, the LKA module utilizes the information obtained from the previous state. Based on the lane position data, the module makes decisions to ensure the vehicle remains securely within the lanes. The car's steering system is then adjusted accordingly to set and maintain the optimal position between the lanes, enhancing overall lane-keeping assistance.

- LKA Flow Chart Diagram:

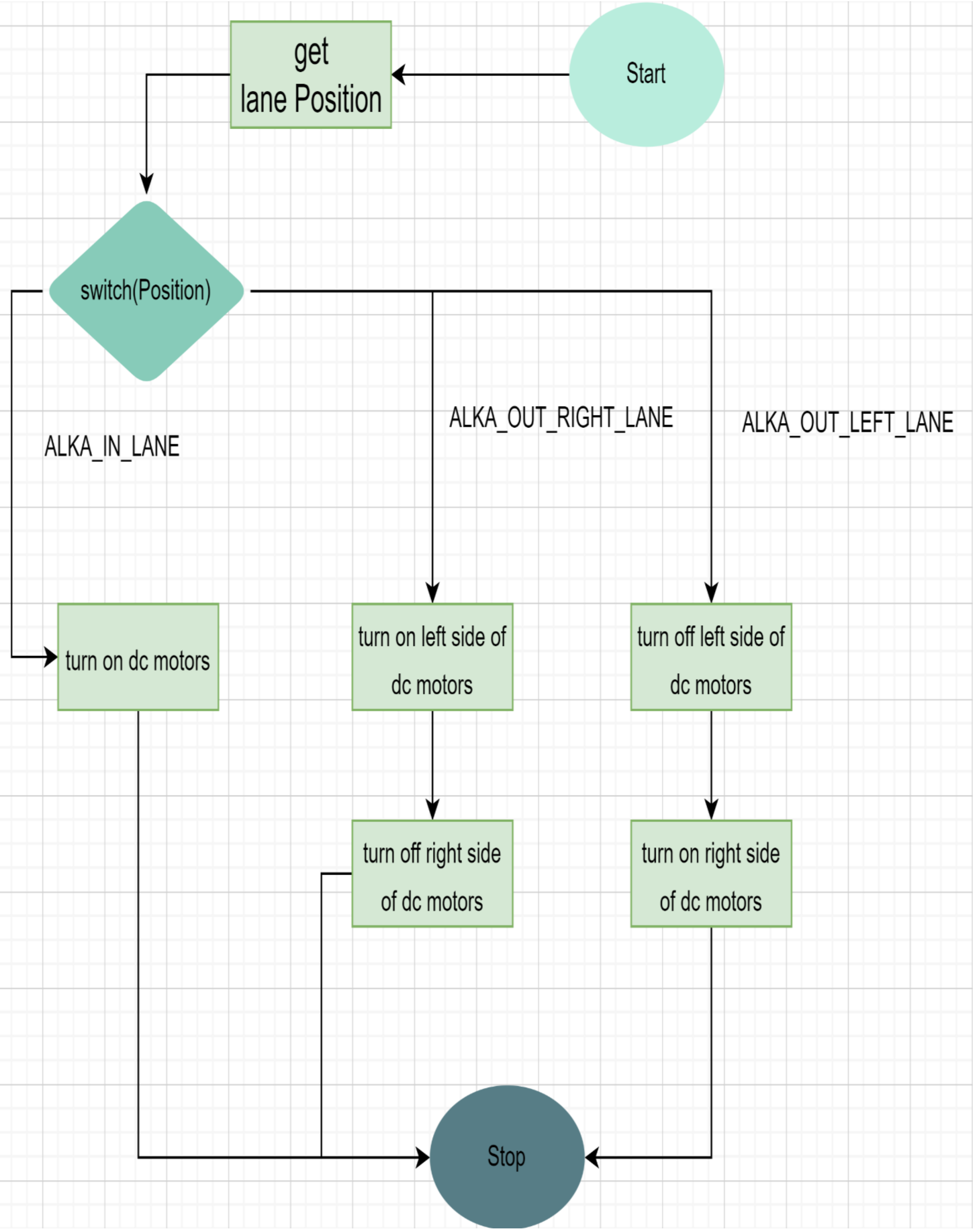


Figure 26.LKA Flow Chart Diagram

- LKA State Machine Diagram:

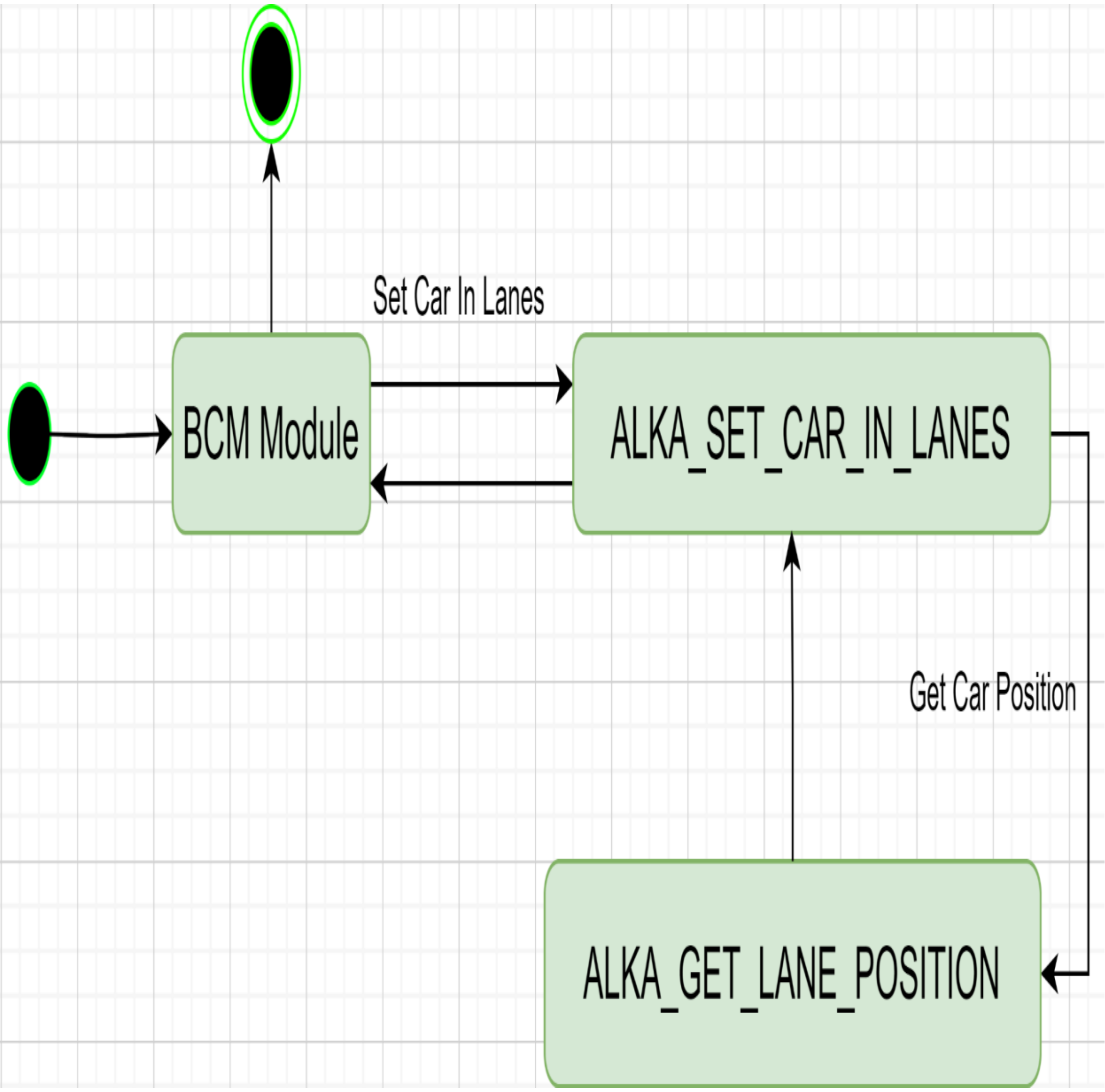


Figure 27.LKA State Machine Diagram

4.6.5 - Automatic Emergency Break

- Overview of Automatic Emergency Brake (AEB):
  Automatic Emergency Brake (AEB) is a crucial safety feature designed to enhance vehicle safety by providing an additional layer of protection against potential collisions. This advanced system utilizes sensors, such as radar or cameras, to continuously monitor the road ahead for potential obstacles or vehicles. In the event of an imminent collision and if the driver fails to apply sufficient braking force, AEB autonomously activates the vehicle's brakes to mitigate the severity of the impact or, in some cases, prevent the collision altogether. By leveraging rapid sensor data analysis and swift response mechanisms, AEB serves as a proactive safety measure, particularly effective in situations where human reaction times may fall short.

- Advantages of Automatic Emergency Brake (AEB):
  The Automatic Emergency Brake offers several significant advantages in bolstering vehicle safety. Foremost, it acts as a vigilant safety net, intervening swiftly in emergencies where the driver may be unable to react promptly. By autonomously applying the brakes, AEB can substantially reduce the severity of collisions or, in ideal circumstances, prevent them entirely, thereby mitigating potential damage and enhancing overall occupant safety. Additionally, AEB is particularly effective in situations involving low visibility or distracted driving, where its rapid response can compensate for human limitations. This safety feature not only protects the vehicle occupants but also contributes to

the overall safety of pedestrians and other road users. In summary, Automatic Emergency Brake stands as a critical safety enhancement, leveraging advanced technology to provide an additional layer of defense against unforeseen collisions.

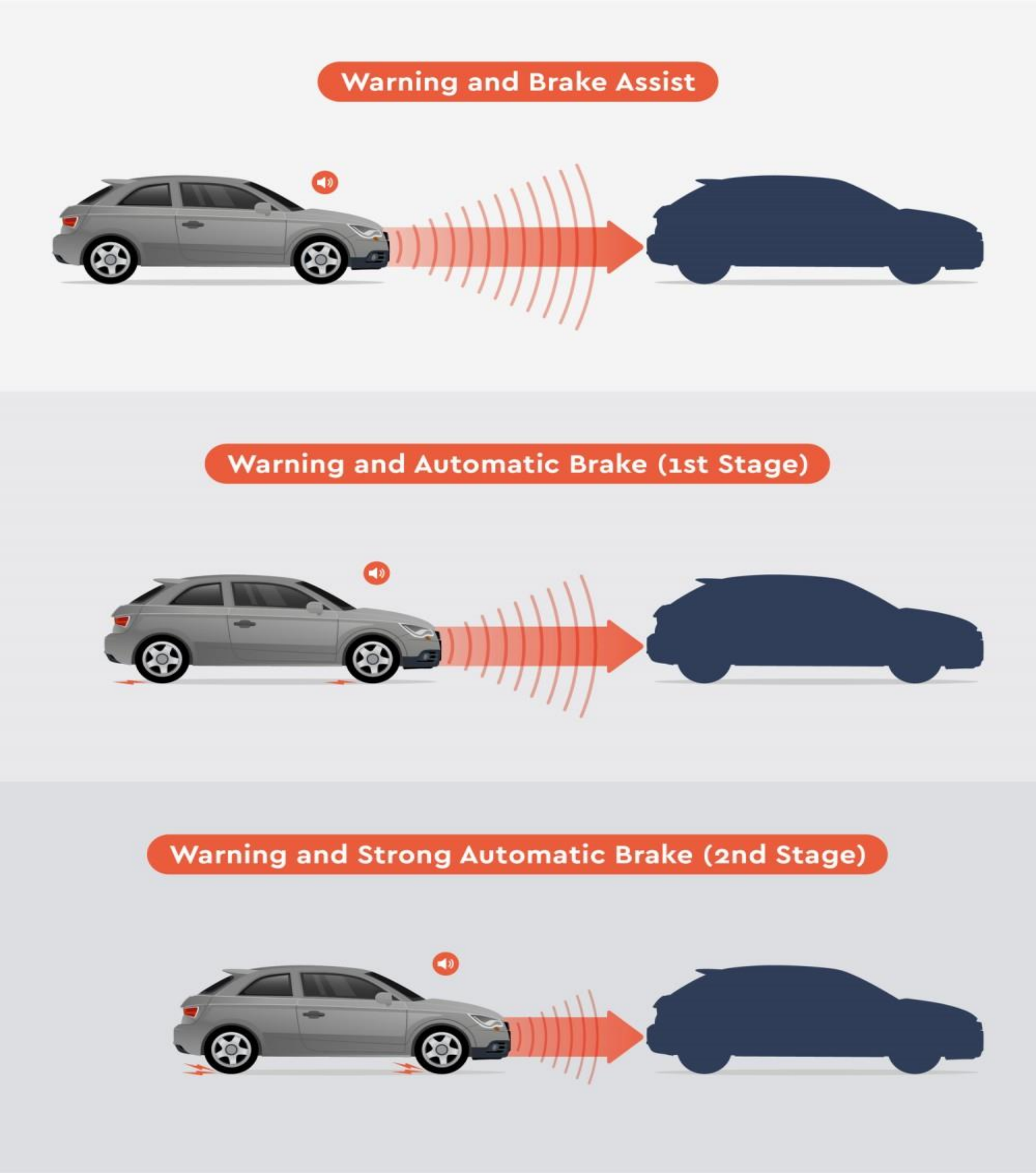

Warning and Brake Assist
Warning and Automatic Brake (1st Stage)
Warning and Strong Automatic Brake (2nd Stage)

Figure 28.AEB Module Image

- Module States:
    - AAEB_DANGEROUS_ZONE:
      When the module enters this state, it indicates that the distance scanned by the lidar signals a dangerous zone. In response, the system activates a robust emergency brake, taking decisive action to avert a potential collision and ensure safety.

    - AAEB_SAFE_ZONE:
      In this state, the module interprets the lidar-scanned distance as indicating a safe zone. Consequently, the system remains inactive, refraining from initiating any emergency braking procedures as the car is deemed to be in a secure driving environment.

- AEB Flow Chart Diagram:

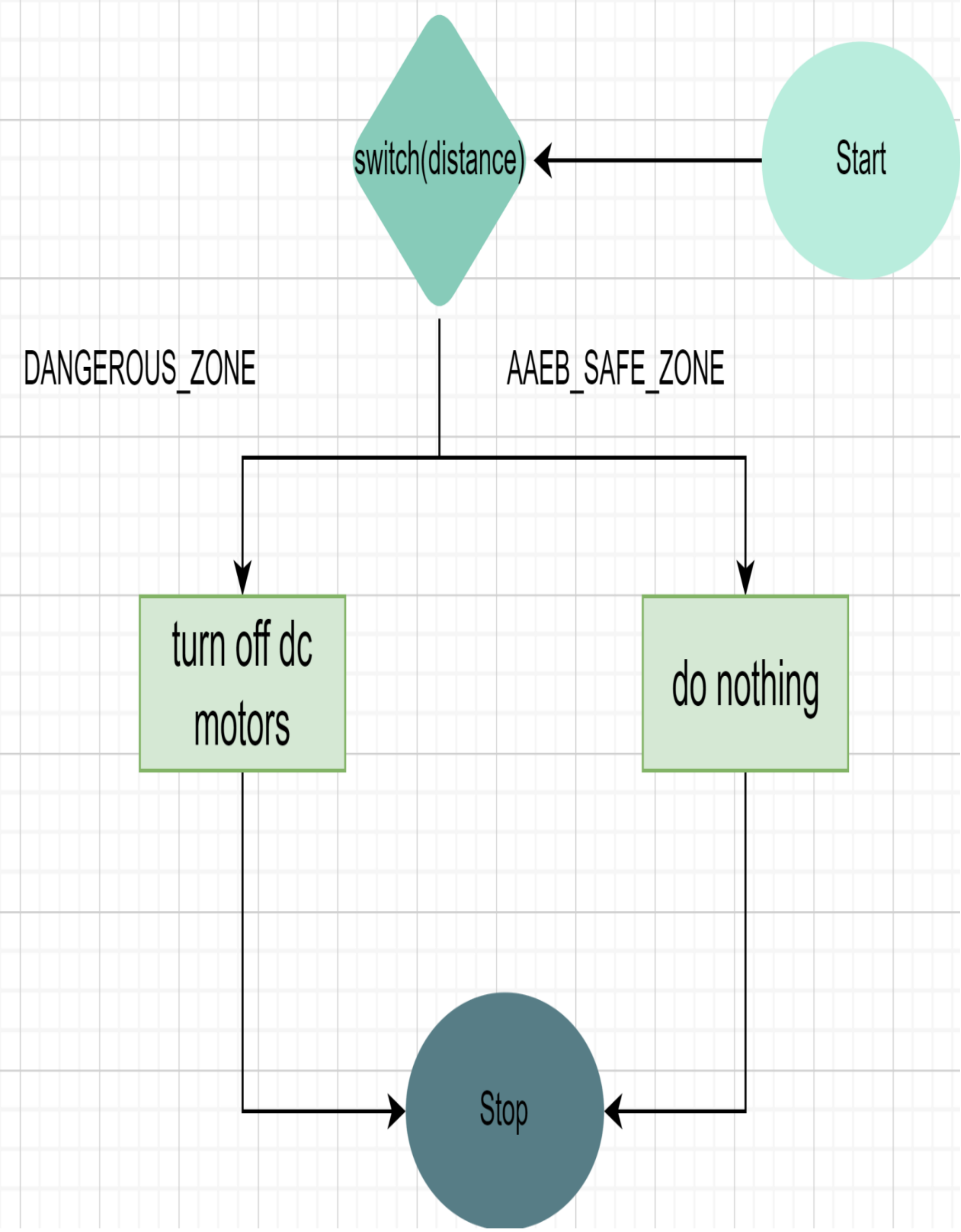


Figure 29. AEB Flow Chart Diagram

- AEB State Machine Diagram:

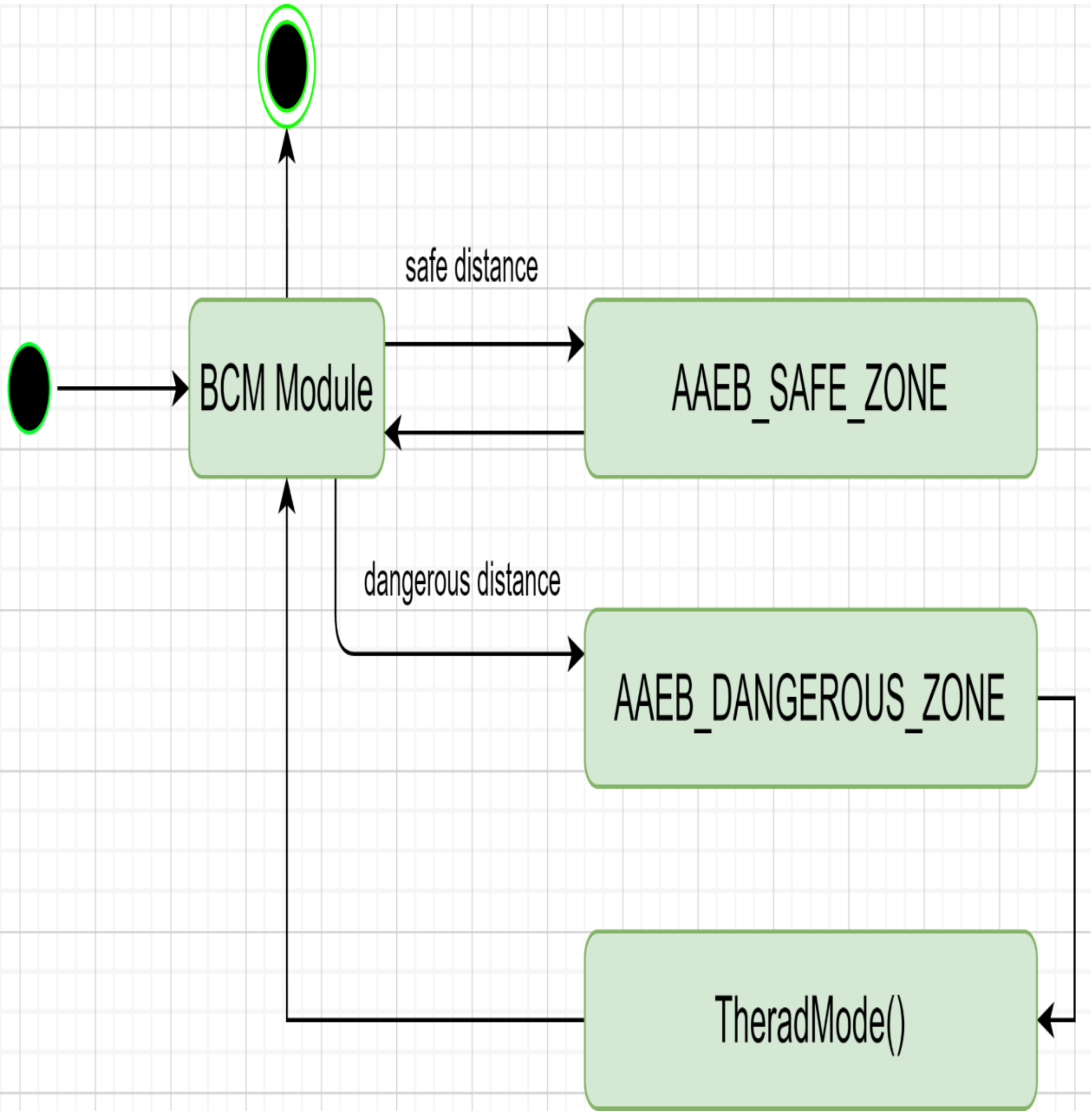


Figure 30. AEB State Machine Diagram

4.6.6 - Flash Over The Air (FOTA):

FOTA is a technology that enables the operating firmware of a device to be upgraded and updated wirelessly over the network (“over the air”) without the need to connect directly to the device.

FOTA-capable devices can download updates from service providers or manufacturers in short time, depending on file size and connection speed. This saves businesses the time and money spent sending a technician to have each one of their devices physically upgraded or updated.

4.6.6.1 - Flashing Techniques for microcontrollers

1- Off-Circuit Programming:

The flash interface is external outside the microcontroller and the flash memory programming pins are connected to some microcontroller pins to be programmed through.

2- In-Circuit Programming:

Also called In System Programming (ISP), here the microcontroller has an internal flash interface, that can generate any necessary programming power from main power supply to provide the required high power for the flash programming, which is inside the microcontroller also, so the flashing process is done inside the microcontroller which means there is no need to remove the microcontroller from the system circuit for programming, hence its name.

3- In-Circuit Programming with bootloader:

The processor initializes a communication port in the microcontroller to make it ready to receive data, a computer starts sending the application code through this communication port to our microcontroller, the processor reads the data, then sends it to the in-circuit flash interface which in turn writes this data in the flash, eventually the target controller is updated with the code.

So now, instead of different interfaces with different microcontrollers we use bootloader to create unified interface with the system, for example if we have 100 ECU in a system, we will design a bootloader for each of them with standard communication protocol like UART or Ethernet.

4.6.6.2 - FOTA Requirements

1 - Safety

No image can be downloaded until it has been verified as the correct image for the specific ECU; utilizing CRCs, checksums, and other release version compatibilities used. CRC is a powerful check mechanism to make sure that the new update is correct, and the flashing process is successful.

2 - Security

Data should be sent utilizing security protocol and encryption, for security of the code it will be encrypted and then will be decrypted at the devise before flashing the update. Protocols used in wireless communication must be secure.

3 - Reliability

Should make sure the data is transferred from one ECU to another ECU is correct and must wait till all data is transferred first before processing the data, so a reliable communication protocol must be used.

4 - Scalability

Should be designed to support large number of embedded devices by providing multicast and broadcast capabilities in addition to unicast.
The software must be designed in a way to support scaling too.

5 - Flexibility

Should support national and regional requirements. Should provide an efficient mechanism to support multiple devices.

4.6.6.3. FOTA Benefits

1. Cost efficient
OEMs can seamlessly manage firmware updates across a fleet of IoT devices from one unified interface. The costs significantly decrease over the entire lifecycle of the product, and no needs to recalls and in-person maintenance.

2. Continuous improvements
Bugs can be fixed, and product behavior can be enhanced after the device lands in the hands of your consumers. This can potentially eliminate costly recalls and in-person maintenance. For example, if device have a major bug which affect the performance of the device instead of collecting all the devices to fix the bug, the update will be sent through any wireless communication.

3. Improved scalability.
FOTA updates enable manufacturers to add new features to infrastructure after the release without physical access to upgrade firmware, which will lead to a better user experience. Features can be added like in cars for example auto parking feature can be added without problems.

4. Faster time-to-market.
Developers can test new features on selected devices and deploy frequently knowing that the products will remain stable. Firmware updates can be dispatched even while the vehicle is still on the production line or in dealer center.

5. Improved safety and compliance.
OEMs can use updates to patch the known vulnerabilities, e.g., defective adapter plugs, instead of recalling the vehicles. Your business can respond to the legal and regulatory responsibilities faster and more cost-effectively.

6. Better software quality.
Updates allow you to continuously amend your systems. Whenever you discover a new opportunity for improvement (e.g., a way to reduce vehicles fuel consumption), you can instantly deliver it to customers, instead of waiting to incorporate it in a new batch of vehicles.

4.6.6.4 Firebase

Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google’s infrastructure. Firebase is google platform, free and easy to use, it have a real time database. Firebase lets you automatically run backend code in response to events triggered by Firebase features and HTTPS requests. Your code is stored in Google’s cloud and runs in a managed environment.

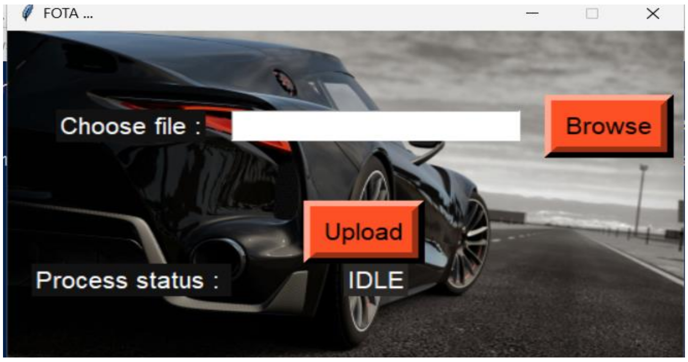
Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents. Firebase is powerful and offers several services, including:

- Realtime database – the Firebase Realtime Database is a cloud-hosted NoSQL database that enables data to be stored and synced between users in real time.
- Storage – Firebase provide large storge where we can store data like the software update to access it later and download it.

4.6.6.5 GUI

Used to abstract the backend implementation with firebase and create standard interface to deal with the update, GUI is implemented using python language.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance.



Choose file :

Process status : IDLE

4.6.6.5. Encryption Algorithms

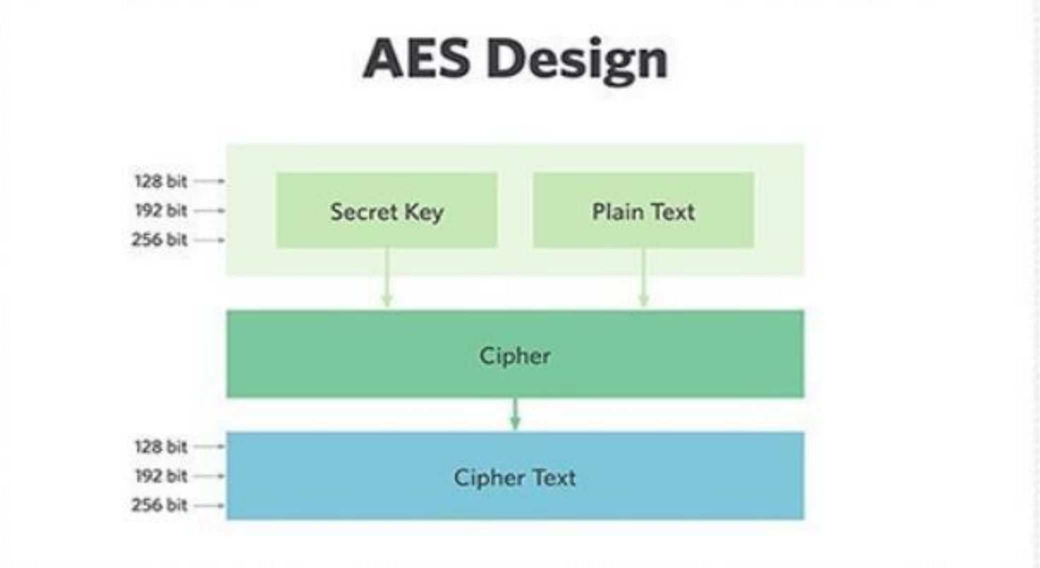
There are two main encryption methods. First, Asymmetric Encryption in which the encryption key is published for anyone to use and for encrypting messages. Only the receiving party has access to the decryption key that enables messages to be read.

Second, Symmetric Encryption in which the encryption and decryption keys are the same. Communicating parties must have the same key to achieve secure communication. We choose AES Symmetric method to encrypt the firmware.

- Advanced Encryption Standard (AES):

AES is a symmetric encryption cipher. This means that the same key used to encrypt the data is used to decrypt it. Over the years, AES has proven itself to be reliable and effective method of safeguarding sensitive information Some of the key benefits of using AES include the following:

- This robust security algorithm may be implemented in both hardware and software.
- It is resilient against hacking attempts, thanks to its higher-length key sizes (128, 192, and 256 bits).
- It is an open-source solution. Since AES is royalty-free, it remains highly accessible for both the private and public sectors.
- AES is the most used security protocol today, used for everything from encrypted data storage to wireless communications.



4.6.6.6. Bootloader

A bootloader is an application whose primary purpose is to allow a systems software to be updated without the use of specialized hardware such as a JTAG programmer. The type of bootloader related to the communication protocol that receive the update from it.

Bootloader requirements:
1- Ability to switch or select the operating mode (Application or bootloader).
2- Communication protocol requirements (USB, CAN, I2C, UART).
3- Record parsing requirement (S-Record, hex).
4- Flash system requirements (erase, write, read).
5- EEPROM requirements (erase, read, write).
6- Error detection mechanism (checksum, CRC, …).
7- Security and encryption.

Bootloader behavior:

A bootloader itself is not that different from a standard application, in fact, it is a standard application.

❖ Start the branch code to determine which code ‘ll be executed based on a flag in EEPROM memory or flash memory.

❖ If bootloader will be executed

1- Move the active image to backup image if exist.
2- Receive code packet by packet from can protocol and store it in the active region in the flash memory.
3- Reset branching flag and make software reset.

❖ If there is an application.

1- Generate CRC and check it.
2- If CRC is ok, it jumps to the application to be executed.
3- If CRC is not ok, it moves the backup image to the active region to be executed.

Application behavior:

The application needs to have two bootloader like capabilities:
1- set a piece of information that the bootloader can detect to enter bootloader mode.
2- Reset the system to initiate a branching decision.

4.7 - Software Tool:

Tool Name	Usage	Description
Git	Version control system	Git is a powerful tool for tracking changes in source code during software development. It enables collaboration among team members and helps manage different versions of the code.
GitHub	osting and collaboration for code	GitHub is a platform that utilizes Git for hosting and collaborating on software projects. It provides features like pull requests and issue tracking to streamline development workflows.
Discored	Team communication and collaboration	Discord is a communication platform designed for teams. It facilitates real-time messaging, voice, and video communication, fostering collaboration among team members.

Stm32CubeIDE	Integrated Development Environment (IDE) for STM32 microcontrollers	Stm32CubeIDE is a comprehensive development environment tailored for STM32 microcontrollers. It streamlines the process of coding, debugging, and deploying software on STM32-based systems.
Doxygen	Documentation generator	Doxygen is a documentation generator that extracts code comments to automatically create documentation for software projects. It simplifies the process of maintaining up-to-date and well-organized documentation.
Proteus	Electronic circuit design and simulation software	Proteus is a software tool for designing and simulating electronic circuits. It enables engineers to visualize and test circuit designs before physically implementing them, saving time and resources.

VS Code	Code editor	Visual Studio Code (VS Code) is a lightweight, yet powerful, code editor. It supports various programming languages and provides features such as debugging, syntax highlighting, and extensions for enhanced functionality.
Sublime text	Text editor	Sublime Text is a versatile and fast text editor. It is known for its speed, simplicity, and a wide range of plugins, making it a popular choice among developers for editing code and text files.