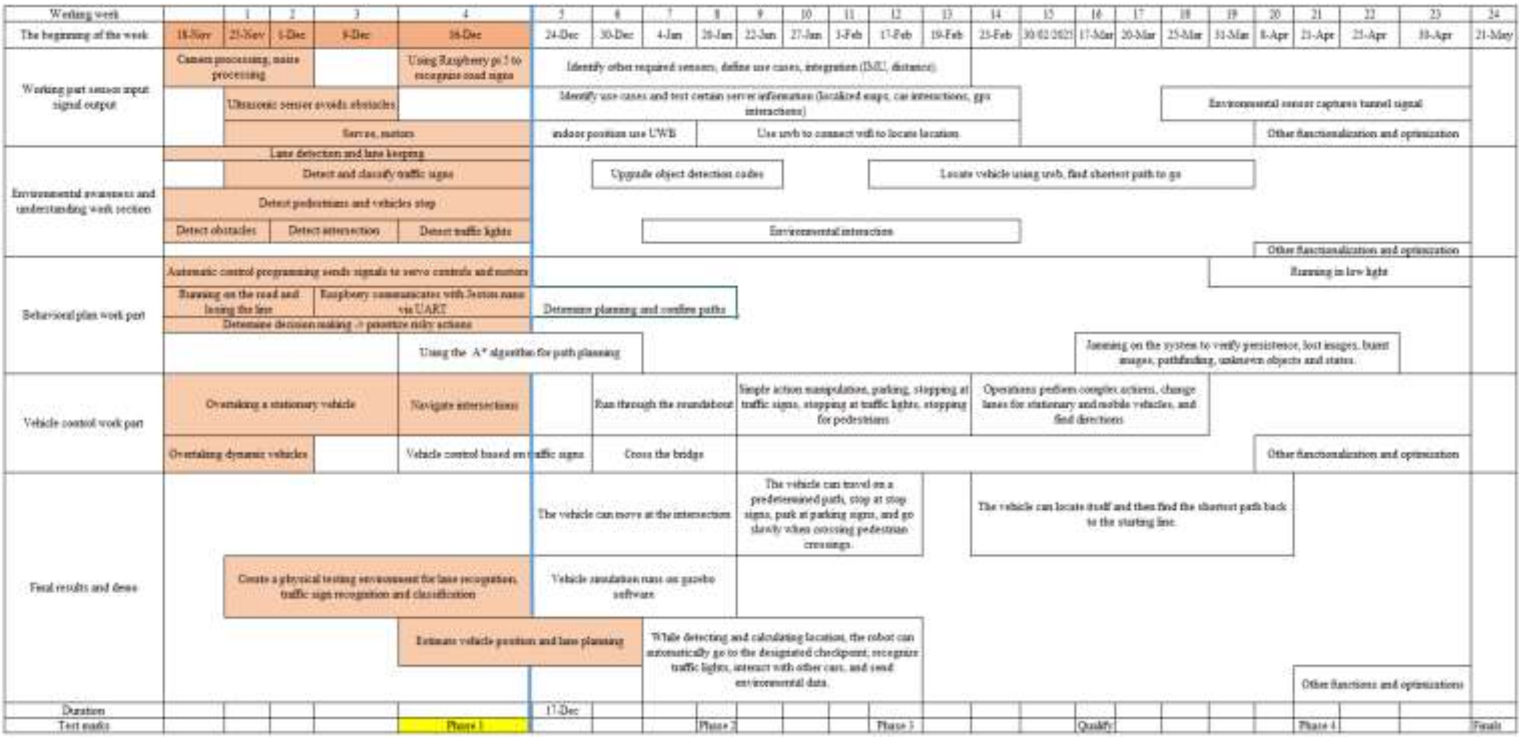
Project Plan and Architecture Design

I. Project Plan

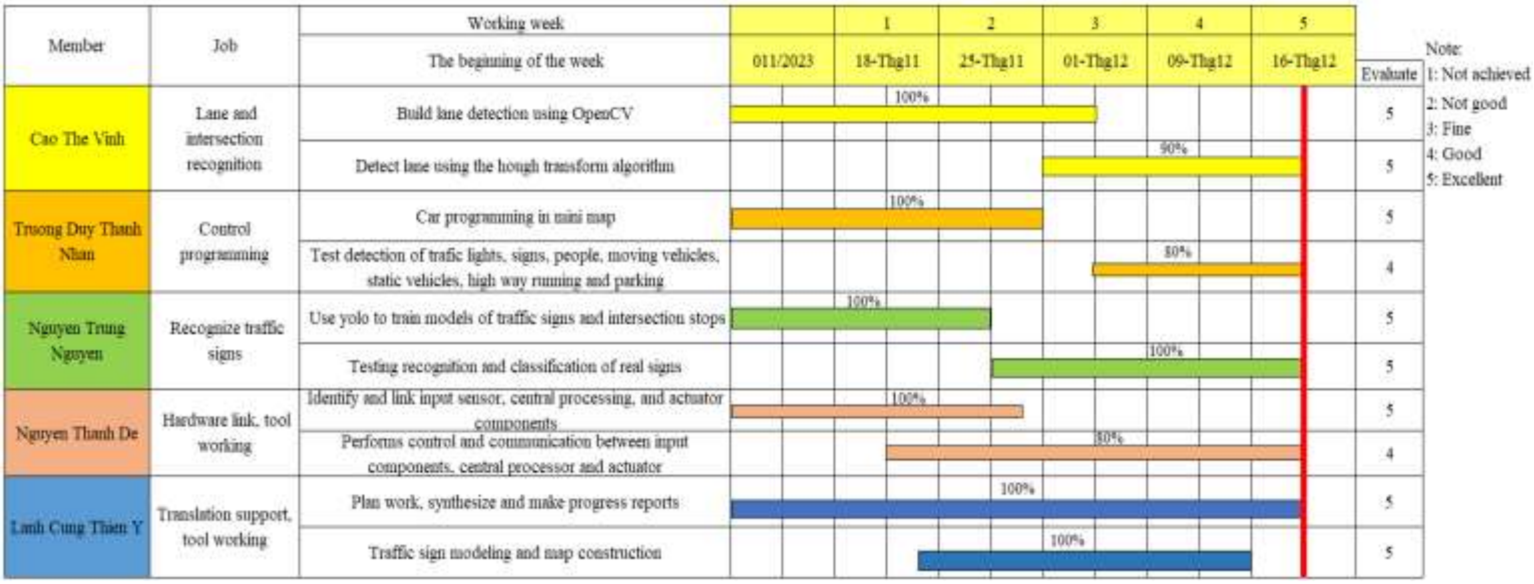
Below is the work plan and schedule of the Smart Car team. This plan spans from the launch of BFMC 2025 to the end of the competition. The team has outlined goals for each phase and the tasks to be completed. The table below presents the proposed plan and some projects completed during Phase 1.



Working week		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
The beginning of the week	18-Nov	25-Nov	1-Dec	9-Dec	16-Dec	24-Dec	30-Dec	4-Jan	20-Jan	22-Jan	27-Jan	3-Feb	17-Feb	19-Feb	25-Feb	30/02/2025	17-Mar	20-Mar	25-Mar	31-Mar	8-Apr	21-Apr	25-Apr	30-Apr	21-May
Working part sensor input signal output	Camera processing, noise processing				Using Raspberry pi 5 to recognize road signs	Identify other required sensors, define use cases, integration (IMU, distance)																			
		Ultrasonic sensor avoids obstacles				Identify use cases and test certain server information (localized maps, car interactions, gps interactions)													Environmental sensor captures tunnel signal						
		Servo, motors				indoor position use UWB			Use uwb to connect wifi to locate location												Other functionalization and optimization				
Environmental awareness and understanding work section	Lane detection and lane keeping																								
		Detect and classify traffic signs					Upgrade object detection codes						Locate vehicle using uwb, find shortest path to go												
	Detect pedestrians and vehicles stop																								
	Detect obstacles		Detect intersection		Detect traffic lights			Environmental interaction																	
																					Other functionalization and optimization				
Behavioral plan work part	Automatic control programming sends signals to servo controls and motors																			Running in low light					
	Running on the road and lanning the line			Raspberry communicates with Jeston nano via UART		Determine planning and confirm paths																			
	Determine decision making -> prioritize risky actions																								
					Using the A* algorithm for path planning												Jamming on the system to verify persistence, lost images, burst images, pathfinding, unknown objects and status.								
Vehicle control work part	Overtaking a stationary vehicle				Navigate intersections		Run through the roundabout			Simple action manipulation, parking, stopping at traffic signs, stopping at traffic lights, stopping for pedestrians					Operations perform complex actions, change lanes for stationary and mobile vehicles, and find directions										
	Overtaking dynamic vehicles				Vehicle control based on traffic signs		Cross the bridge														Other functionalization and optimization				
Final results and demo						The vehicle can move at the intersection				The vehicle can travel on a predetermined path, stop at stop signs, park at parking signs, and go slowly when crossing pedestrian crossings.					The vehicle can locate itself and then find the shortest path back to the starting line.										
		Create a physical testing environment for lane recognition, traffic sign recognition and classification				Vehicle simulation runs on gazebo software																			
					Estimate vehicle position and lane planning			While detecting and calculating location, the robot can automatically go to the designated checkpoint, recognize traffic lights, interact with other cars, and send environmental data.														Other functions and optimizations			
Duration						17-Dec																			
Test marks					Phase 1				Phase 2				Phase 3				Qualify					Phase 4			Finals

Table 1: Schedule and specific planned activities of Smart Car Team.

The team has clearly assigned tasks to each member, collected individual results, and evaluated the progress of implementation.



Member	Job	Working week		1	2	3	4	5	
		The beginning of the week	011/2023	18-Thg11	25-Thg11	01-Thg12	09-Thg12	16-Thg12	Evaluate
Cao The Vinh	Lane and intersection recognition	Build lane detection using OpenCV	100%						5
		Detect lane using the hough transform algorithm				90%			5
Truong Duy Thanh Nhan	Control programming	Car programming in mini map	100%						5
		Test detection of trafic lights, signs, people, moving vehicles, static vehicles, high way running and parking				80%			4
Nguyen Trung Nguyen	Recognize traffic signs	Use yolo to train models of traffic signs and intersection stops	100%						5
		Testing recognition and classification of real signs			100%				5
Nguyen Thanh De	Hardware link, tool working	Identify and link input sensor, central processing, and actuator components	100%						5
		Performs control and communication between input components, central processor and actuator		80%					4
Lanh Cung Thien Y	Translation support, tool working	Plan work, synthesize and make progress reports	100%						5
		Traffic sign modeling and map construction		100%					5

Note:
1: Not achieved
2: Not good
3: Fine
4: Good
5: Excellent

Table 2: Detailed tasks, completion status, and work quality of each team member.

The task plans are clearly assigned to each member, and each individual is responsible for improving and maintaining their assigned components.

1. **Cao Thế Vinh** manages vehicle control at intersections, slopes, and improves complex maneuvers.
2. **Trương Duy Thanh Nhàn** handles vehicle control when encountering traffic signs and obstacles, and tests the sensors' sensitivity and processing capabilities.
3. **Nguyễn Trung Nguyên** focuses on object detection and classification, as well as traffic light recognition and categorization.
4. **Nguyễn Thanh Đệ** takes responsibility for replacing and repairing malfunctioning hardware during the vehicle's production and designs optimal hardware placement for construction.
5. **Lành Cung Thiên Ý** assists with translation, reporting, and presentation tasks.

In terms of advantages, the car demonstrates excellent lane detection and object recognition, as well as stable situational handling. On the other hand, the vehicle still struggles with lane-keeping in sharp turns and is prone to interference noise in image processing.

II. Architecture Design

Part 1: Hardware Design

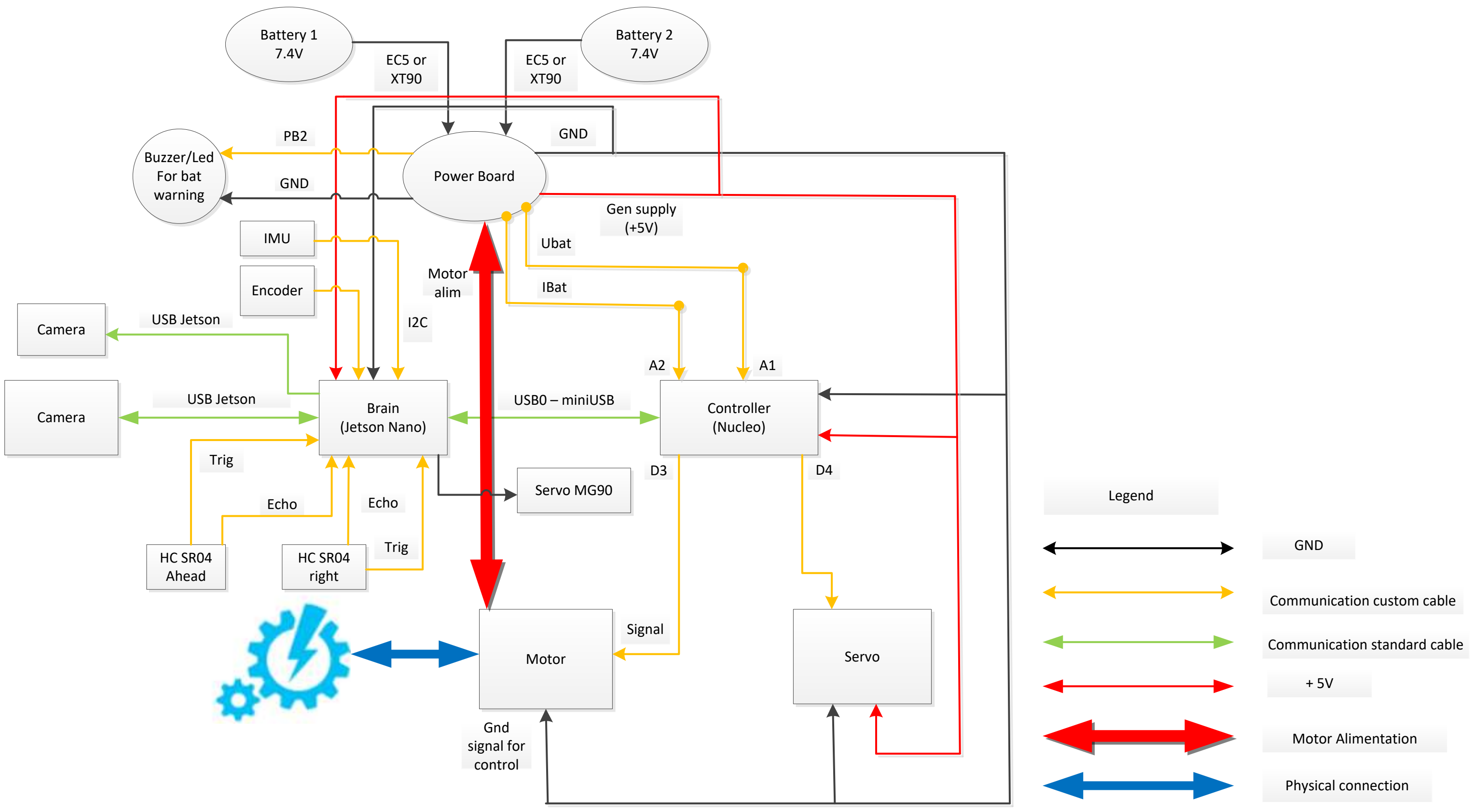


Figure 1. Hardware electrical diagram

The hardware structure used includes:

Controller unit	
Jetson nano	The central unit acts as the brain of the system, receiving input signals from sensors and the camera. It processes data from the Encoder and IMU via I2C communication and sends control signals to the Nucleo microcontroller to manage the motor and servo operations effectively.
Nucleo stm	It receives processed signals from the Jetson Nano to control the motor and servo.
Quickrun Fusion SE Motor, 1200	The motor controller bridge converts signals into PWM pulses to control the motor.
Signal Transmitter	
Jetson Camera	Capturing and transmitting images, lane markings, and objects to the Jetson Nano.
Encoder	Sending vehicle velocity values to the STM Nucleo for processing.
Executive Structure Division	
Steering servo	Receiving steering angle signals from the Jetson Nano, transmitted to the STM Nucleo, to control the vehicle's motion.
Motor	Receiving speed control signals from the Jetson Nano via the Nucleo STM and the Quickrun Fusion SE converts electrical signals into PWM signals to control motor speed.

Part 2: Software structure

Below is the software structure:

- The software is added ultrasonic sensor control, specifically the Ama ultrasonic sensor.
- Images from the camera are transmitted to the Jetson Nano, which uses image processing algorithms to identify objects in the images, such as vehicles, pedestrians, lane markings, traffic lights, etc.
- Data from other sensors, such as IMU, encoders, and ultrasonic sensors, are also transmitted to the computer. This data is used to determine the vehicle's position and state, including direction, speed, and distance from surrounding objects.

By combining image data with sensor inputs, the computer can make control decisions for the vehicle, such as accelerating, decelerating, braking, or turning.

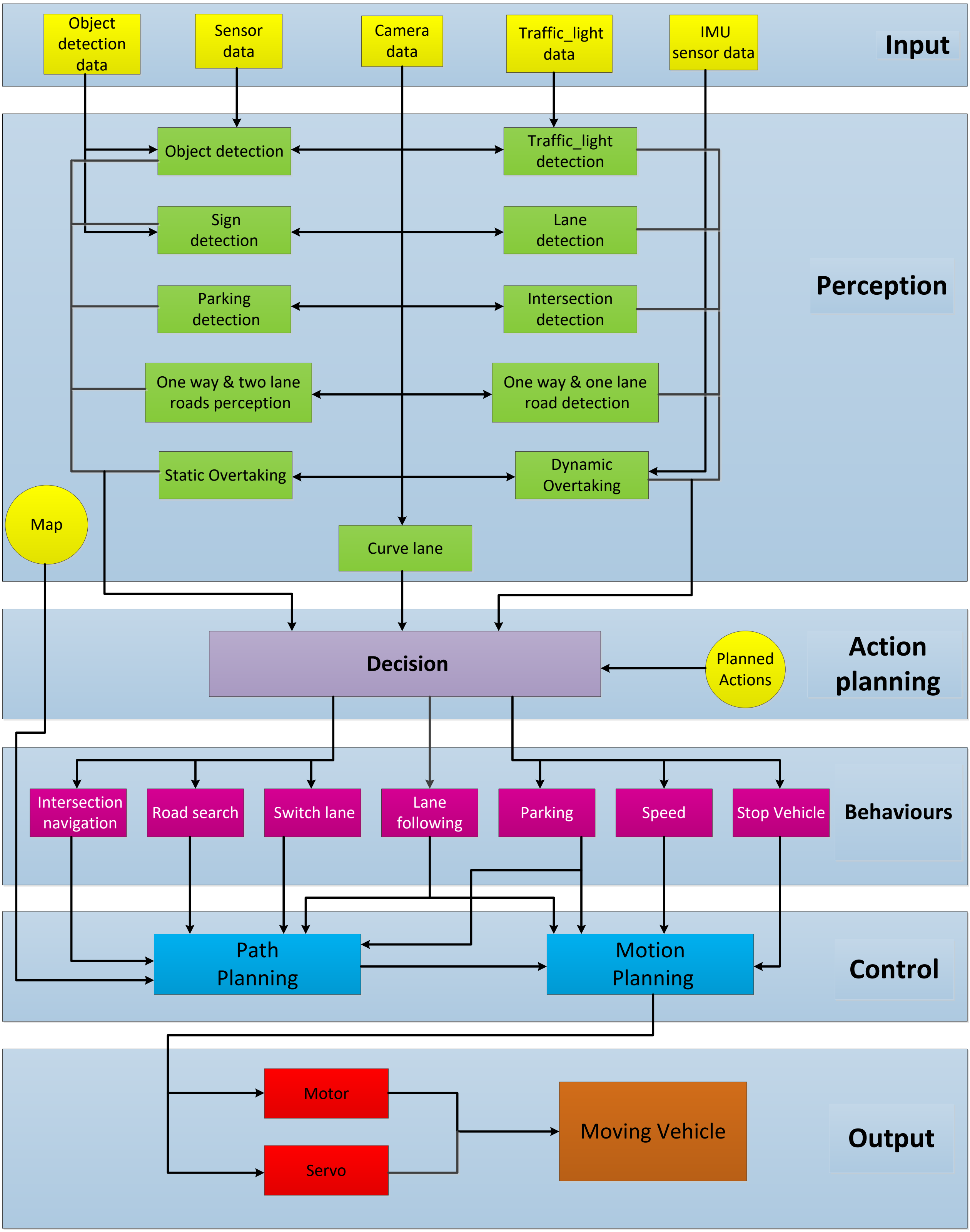


Figure 2. Software structure diagram