

FrostAV Development

Stage 1: Base

Lewis Collum

Updated October 24, 2019

Contents

1	System Abstraction	1
1.1	Abstract System Operation Description	1
1.2	Bridge Controllers	2
1.3	Networker	2
1.4	Sensor Systems	2
2	System Plan	2
2.1	System Operation Description	3
2.2	System Assumptions	3
2.3	Bus Communication	3
2.3.1	Wireless-Bus	3
2.3.2	Wired-Bus	3
2.4	Block Function Description	3
2.4.1	Raspberry Pi 2 (RPi2)	3
2.4.2	Servo Bridge Controller (ATMEGA328P)	3
2.4.3	Motor Bridge Controller (ATMEGA328P)	3
2.4.4	PS3 Eye Camera	4
2.4.5	Electronic Speed Controller (DYNS2211)	4
3	Control Abstraction	4
4	Control Plan	4
5	Future	4
5.1	I2C Arbitration	4

1 System Abstraction

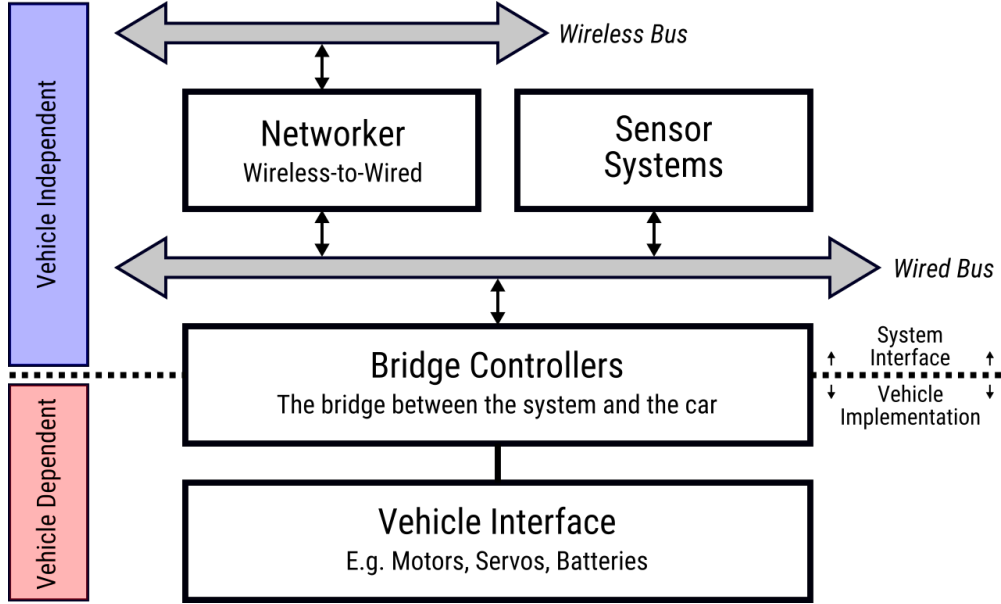


Figure 1: Fundamental module abstraction. Consists of a wired-bus for inter-communication between modules on the Frost vehicle, and a wireless-bus for communication with a server or an ssh client. Bridge Controllers act as the coupling between the vehicle and the modules that do not depend on the vehicle interface.

1.1 Abstract System Operation Description

The Frost Autonomous Vehicle consists of a vehicle and a system for controlling the vehicle. The vehicle provides an interface mainly to move it and to steer it. Bridge Controllers handle the peripherals that make up the vehicle interface. Bridge Controllers cannot operate unless they have instructions. These instructions come from a wired-bus. Different types of modules can send instructions to the wired-bus. The most important modules are the Networker and each individual Sensor System. Sensor Systems provide feedback for the Bridge Controllers to act upon. Furthermore, the Networker transfers information between the wired-bus and the wireless-bus. The wireless-bus allows for communication with devices that are not on the vehicle itself. Overall, the abstract system provides a map for how information is received and directed towards the vehicle's interaction with the environment.

1.2 Bridge Controllers

"Bridge" refers to a software design pattern from the Gang of Four, which intends to "decouple an abstraction from its implementation so that the two can vary independently." As such, the Bridge Controllers will have software that implements the Bridge design pattern, so that implementation code (for a specific vehicle) is decoupled from the system interface.

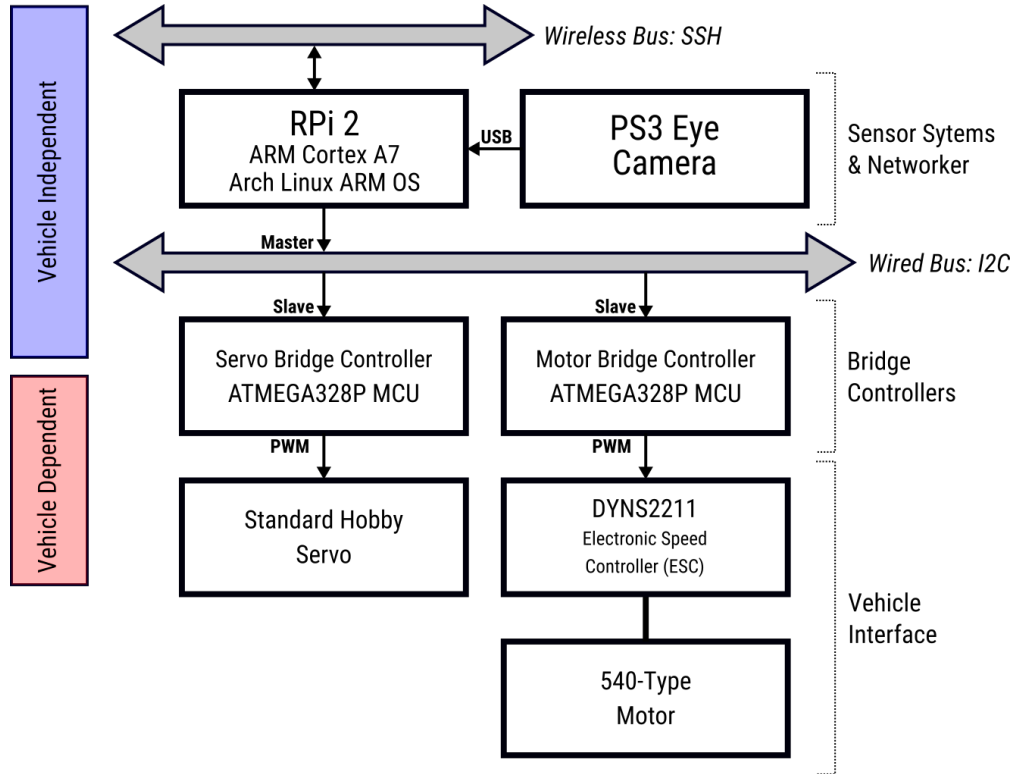
1.3 Networker

The Networker is responsible for transferring information between the wireless-bus and the wired-bus.

1.4 Sensor Systems

Various Sensor Systems may be added to the wired-bus. An individual Sensor System should receive information from a sensor, process the information, and send the result to the wired-bus.

2 System Plan



2.1 System Operation Description

The PS3 Eye Camera provides visual information about the environment. This information will be processed with the OpenCV library on a Raspberry Pi 2 with an Arch ARM operating system. Processed camera information will then be piped through I^2C to the Bridge Controllers. The Servo Bridge Controller has a PWM output which directly drives the steering servo. The Motor Bridge Controller has a PWM output that is handled by the electronic speed controller (ESC), which drives the motor.

2.2 System Assumptions

The assumption made in this development stage is that the motor speed and servo steering angle can be controlled individually, such that motor speed and steering angle do not require direct knowledge of each other. Hence, the arrows from the wired-bus to the Servo and Motor Bridge Controllers are unidirectional. This assumption is meant to simplify the I^2C wired-bus configuration - since bi-directional communication on this bus would require arbitration.

2.3 Bus Communication

JSON will be used as the format for data for both the wireless and wired buses. JSON provides easy to understand data structure packets that can be serialized and sent as a stream. Using JSON sacrifices speed for readability and extendability of the system.

2.3.1 Wireless-Bus

In this stage the wireless-bus should allow for an SSH client to connect to it. As such, JSON packets that would normally be sent to the wireless-bus, will instead be stored locally in the system, but will be callable from an SSH client.

2.3.2 Wired-Bus

For the wired-bus, I^2C will be used. I^2C addressing will be done manually for each slave connected to the bus. Data will be sent from master to slaves via the JSON format.

2.4 Block Function Description

2.4.1 Raspberry Pi 2 (RPi2)

The RPi2 has an Arch Linux ARM operating system. It is responsible for processing camera information from the PS3 Eye. It is also responsible for being the Networker, which transfers information between the wired and wireless buses.

2.4.2 Servo Bridge Controller (ATMEGA328P)

Processes commands from the wired-bus to control the vehicle steering servo.

2.4.3 Motor Bridge Controller (ATMEGA328P)

Processes commands from the wired-bus to control the vehicle motor's speed.

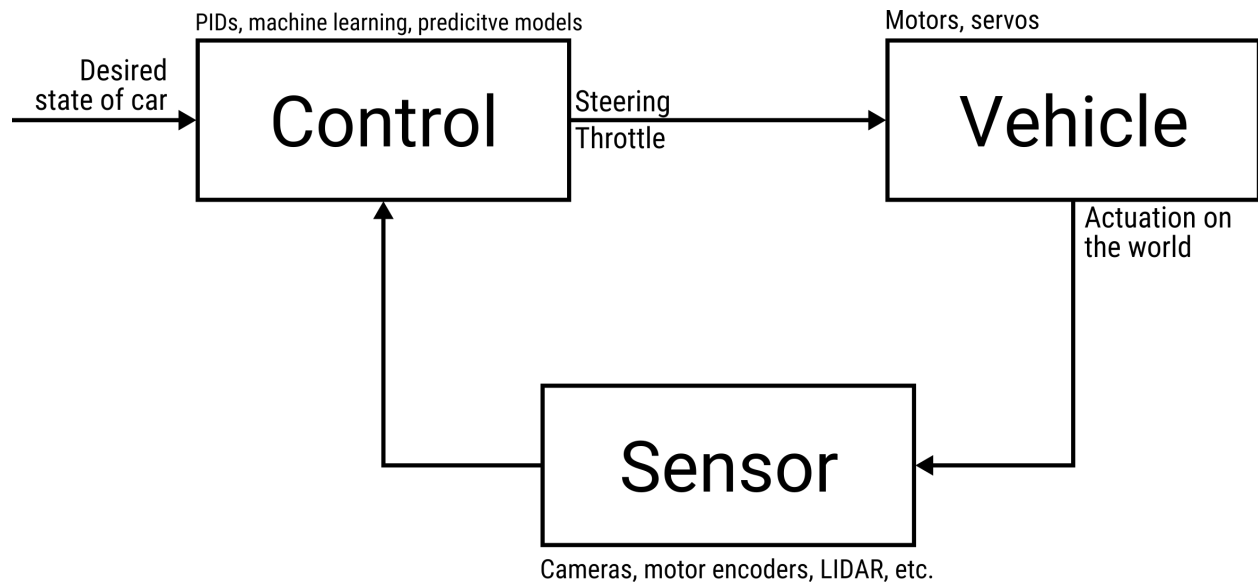
2.4.4 PS3 Eye Camera

Provides images over USB, for environmental perception.

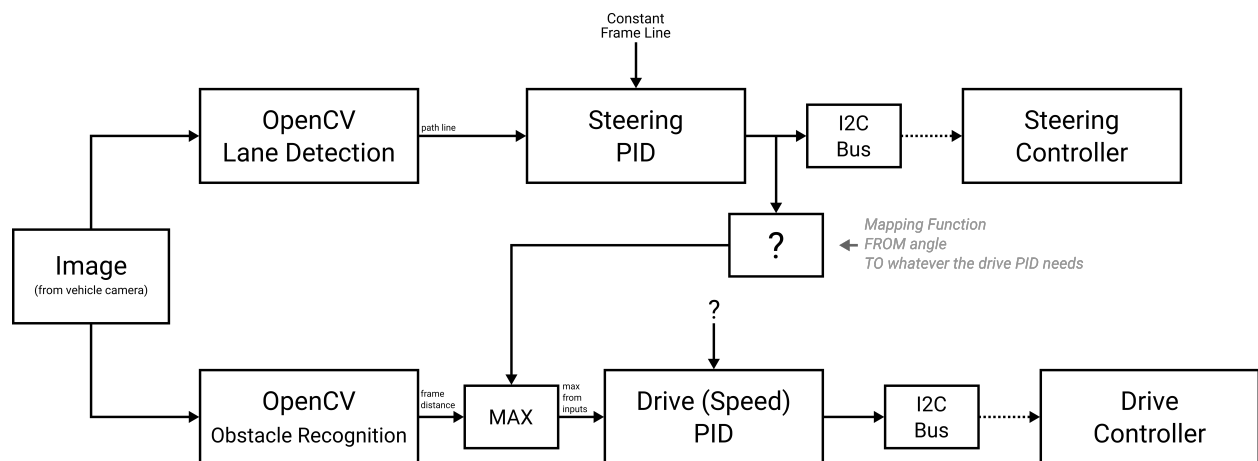
2.4.5 Electronic Speed Controller (DYNS2211)

Takes in a PWM signal and outputs an amplified and directional PWM wave to the vehicle's motor.

3 Control Abstraction



4 Control Plan



5 Future

5.1 I2C Arbitration

It is expected that modules communicating on the wired bus will need to communicate with each other. In this case, there would be multiple masters on a single i2c bus. To implement this, arbitration would be necessary so that masters on the bus do not interrupt each other.