

Vehicle-to-Vehicle (V2V) Communication Implementation

Gunnar Fandrich

Group: Autogators

Electrical and Computer Engineering

University of Florida

Gainesville, Florida

gunnarfandrich@ufl.edu

Mark Lai

Group: Autogators

Electrical and Computer Engineering

University of Florida

Gainesville, Florida

marklai@ufl.edu

Rafael Hernandez-Lopez

Group: Autogators

Electrical and Computer Engineering

University of Florida

Gainesville, Florida

rhernandezlopez1@ufl.edu

Rohan Malik

Group: Autogators

Electrical and Computer Engineering

University of Florida

Gainesville, Florida

rmalik@ufl.edu

Abstract—Vehicle-to-vehicle (V2V) allows communication between vehicles, promoting driver awareness and potentially reducing the number of collisions. Existing V2V implementations (cellular vehicle-to-everything, C-V2X) rely on cloud data, whereas the implementation shown in this paper does true V2V between vehicles. A mockup utilizing two ESP32 with a time-of-flight (ToF) sensor and accelerometer communicate using UDP to deliver a proof-of-concept for V2V that can be used as a base for V2V.

Index Terms—V2V, vehicles, driver, awareness, C, ESP32, UDP

I. PROBLEM STATEMENT

As society moves towards utilizing more autonomous driving systems, vehicles can act as a network to promote efficient and safe driving. This network is referred to as vehicle-to-vehicle (V2V). Automotive vehicles should communicate with each other through V2V to increase driver awareness and reduce the number of collisions [1].

II. INTRODUCTION

The current state of V2V is nonexistent. The closest thing on the market to V2V is the 2023 Safety Cloud for Chrysler Vehicles, which is implemented through a cellular network to create C-V2X (cellular vehicle-to-everything) [2], [3]. While this is fairly close to V2V, passing through a cloud does not exhibit true V2V. Ideally the cars would communicate directly to each other, which is what is covered by the implementation discussed in this paper. Additionally, Stellantis announced vehicle-to-grid (V2G) testing in 2019 [4].

In 1999, the FCC designated a 75 MHz spectrum in the 5.9 GHz band for dedicated short-range communications (DSRC) [5]. Unfortunately, this band was reallocated in 2020 for unlicensed WiFi [5] use due to the failure of automobile makers to release V2X production cars. Due to the removal

of this band, we chose UDP over WiFi as an acceptable simplified protocol for V2V. Ideally, UDP would be replaced with another protocol built on top of UDP – or similar protocol – specifically for V2V and operate on an FCC-designated band for V2V.

V2V will be illustrated using an ESP32 paired with time-of-flight (ToF) sensor and accelerometer. Each ESP32 represents a simplified car, and communicate with each other using UDP. UDP was chosen as the protocol to allow ad-hoc car networks to be formed as well as allow dropped packets once a car leaves the network.

III. BACKGROUND AND RELATED WORKS

TODO [6].

IV. V2V IMPLEMENTATION

This section covers our V2V implementation, the features and the mockup.

A. Features

The two vehicles are represented by two ESP32s. They each have a ToF sensor and an accelerometer. The ToF sensor is used to identify whether there is another “vehicle” in front of the ESP32 and the accelerometer is used to know whether the other “vehicle” is moving. On a real vehicle, the CAN bus can be used to monitor and send each vehicle’s velocity – and other information as needed, such as GPS for locating where the incoming UDP packet is coming from – in the UDP packet. The current UDP packet we are sending has an ID for the vehicle, the ToF distance, and the accelerometer measurement. On the other vehicle, this information is used to alert the driver via some LEDs. This UDP packet was standardized between vehicles using a defined C struct to contain the relevant information.

B. Hardware

The mockup can be found on Fig. 1. Each breadboard has an ESP32, a power supply, a ToF sensor, and two LEDs. Each could have an accelerometer, but only one “vehicle” had an accelerometer during testing.

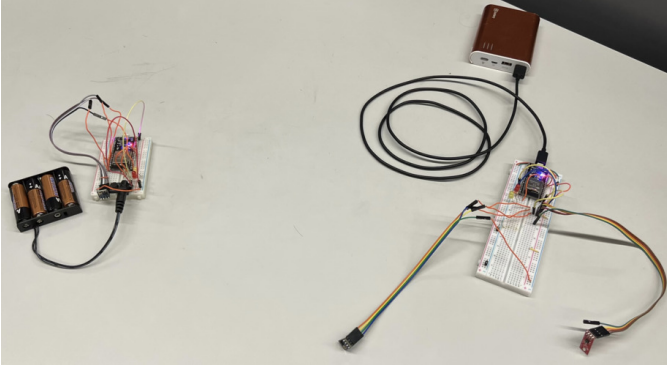


Fig. 1. The mockup used to test V2V, ESP32s are “vehicles”.

Close-ups of each “vehicle” can be found in Fig. 2 and Fig. 3. The assembled “vehicles” can be found in Fig. 4 and Fig. 5. They are named “vehicle” B and U, respectively, from now on. B for battery pack and U for USB battery pack.

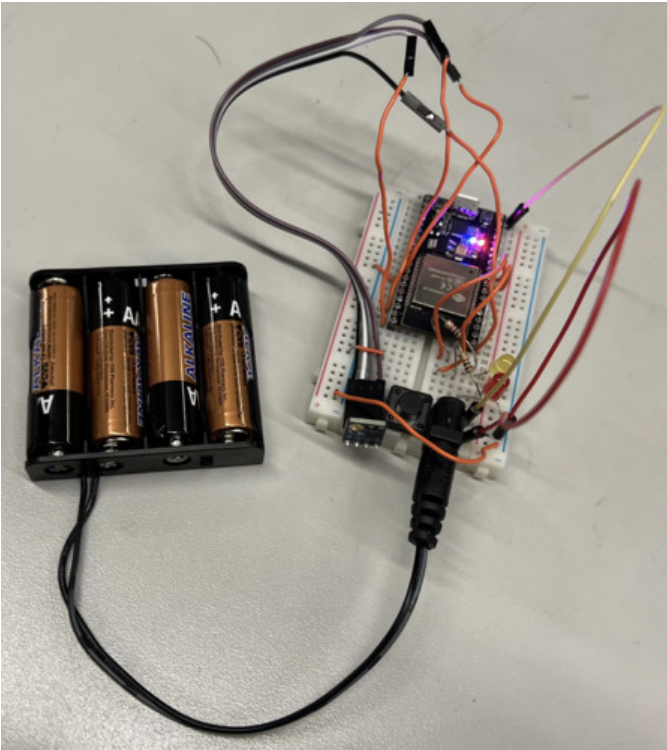


Fig. 2. One of the “vehicles”, vehicle B, used to test our V2V implementation.

The driver has two LEDs available in this implementation: a yellow communication (caution) LED, and a red “warning” LED. The yellow LED turns on when there is a nearby vehicle (less than 100mm for this experiment) exchanging UDP packets with the driver’s vehicle. The red LED turns

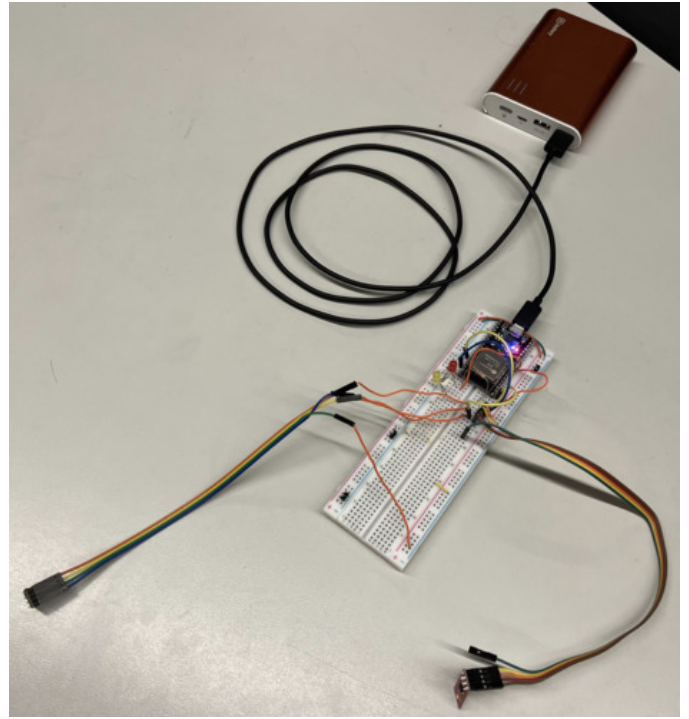


Fig. 3. Another “vehicles”, vehicle U, used to test our V2V implementation.

on in cases where the driver should be more cautious. To simulate this, the red warning LED was turned on in the case that a caution message has been received, the ToF sensor has measured a distance less than or equal to 50mm, and the accelerometer observes a negative acceleration.

Two other lights, intended for debugging, were also incorporated into the design. The blue LED onboard the ESP32 was employed to indicate a successful boot and successful WiFi connection between the two ESP32s. The red LED oneboard the ESP32 was utilized to monitor the power supply and to ensure the ESP32 was powered on.

Throughout the project’s development, several hardware issues were encountered. One of the issues was the original set of Time-of-Flight sensors not working. This delayed the project and prompted the need for new Time-of-Flight sensors. Fortunately, this issue was resolved quickly by ordering a new set. The new set did not work perfectly, but this was likely due to the quality of the components. There were intermittent disconnect issues or sporadic readings, but generally the sensors worked and fulfilled their role in this project. Another issue encountered during testing was power supply issues. Originally, two mobile phone battery banks were to be used to power each ESP32, but the minimal current draw of the ESP32 caused one of the battery banks to switch off. This battery had to be substituted for a battery consisting of four AA batteries in series. This battery provided 6V, and the internal power regulator of the ESP32 was able to step the voltage down to the system’s 5V requirement.

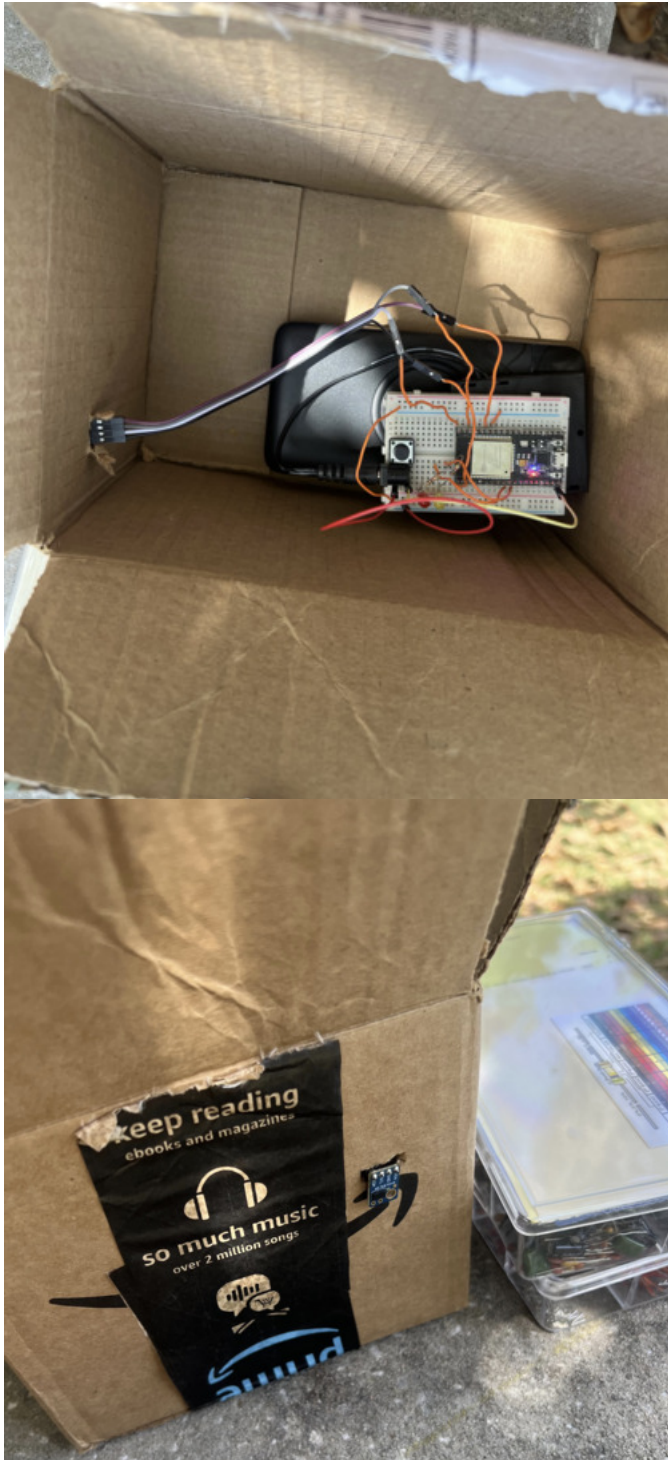


Fig. 4. Assembled “vehicle”, vehicle B, in our testing environment.

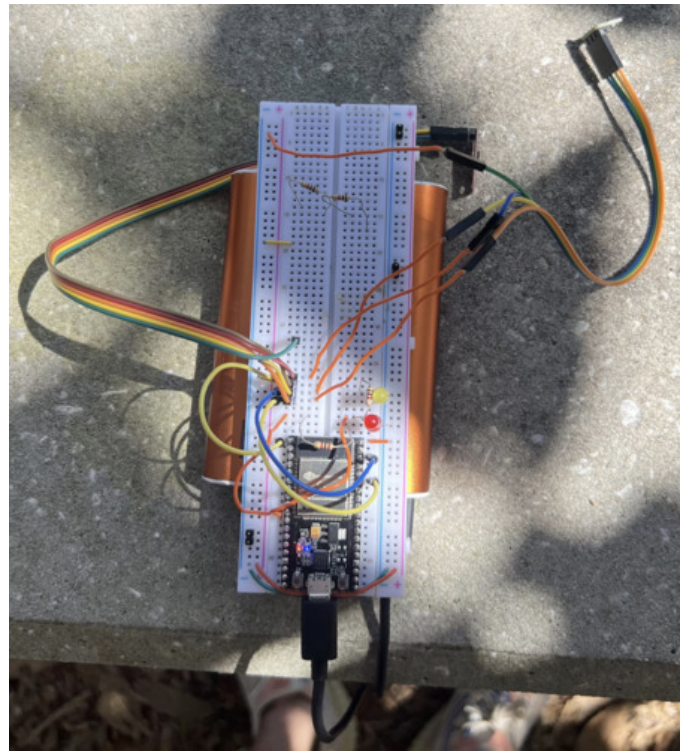


Fig. 5. Another assembled “vehicle”, vehicle U, in our testing environment.

V. RESULTS

The two vehicles were able to be powered independently and in a portable nature. The platoon leader vehicle hosted a WIFI access point, which the subsequent vehicle connected to. ToF data, vehicle status, and accelerometer data was sent between each vehicle and processed accordingly.

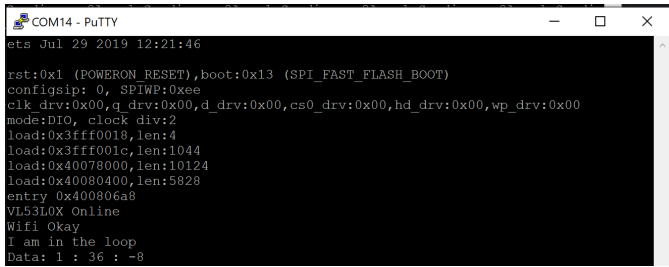
The two “vehicles” communicate over UDP with each other. See Fig. 6 to see the “vehicle” sending a UDP packet, and Fig. 7 to see the other “vehicle” receive the UDP packet.

```
COM12 - PuTTY
rst:0x1 (POWERON RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
config:ip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:10124
load:0x40080400,len:5828
entry 0x400806a8
VL53L0X Online
LSM9DS1 Online
Wifi AP active
Wifi Okay
I am in the loop
Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel Sendin
g: -8Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel S
ending: -8Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel Sending: -8Ac
cel Sending: -8Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel Sending:
-8Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel Send
ing: -8Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel
l Sending: -8Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel Sending: -
8Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel Sendi
ng: -8Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel Sending: -8Accel
```

Fig. 6. One “vehicle” is sending a message through UDP.

VI. SECURITY

Our implementation of V2V has room to improve. A V2V implementation usable on the market would address a few



```
COM14 - PuTTY
ets Jul 29 2019 12:21:46

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:10124
load:0x40080400,len:5828
entry 0x400806a8
VL53L0X Online
Wifi Okay
I am in the loop
Data: 1 : 36 : -8
```

Fig. 7. The other “vehicle” is receiving a message through UDP.

security concerns.

A. Packet Security

As our UDP packets are being broadcasted to everyone in the nearby vicinity, an attacker only needs to be located near the network to be considered a “car”. This means that anyone could send fake messages or tamper with existing packets. This leaves room for improvement in terms of admitting “cars” on the network, and properly identifying nodes on the network as “car”, “adversary”, or “miscellaneous” nodes.

Attackers on the network could potentially leverage this limitation to inhibit the flow of traffic by broadcasting that there is some blockade ahead (ToF sensor data shows car ahead, accelerometer data shows there is no/little movement). Attackers could also target individual vehicles with a similar idea. They could send false packets to make the victim vehicle think it has to brake suddenly, potentially crashing with vehicles behind them or causing them to brake too fast. If the victim is carrying a heavy load, such as a large trailer, braking too fast can compromise the stability of their vehicle and potentially swerve out of control or unlatch their trailer.

B. Network Security

A member of the ad-hoc network may be either calibrated to send too many packets or is an attacker. In a case where the packet traffic is too heavy, a denial of service attack (DoS) may occur. This should be addressed as well before V2V is accepted by car manufacturers.

As the network is completely open to new “cars”, attackers may be able to probe the network for more vulnerabilities than those discussed. For example, perhaps some UDP misconfiguration could be leveraged to forward UDP packet contents to a CAN bus, or similar idea.

VII. CONCLUSION

In this experiment, a V2V network was created using two ESP32 with ToF and accelerometer as a proof-of-concept/base on which car manufacturers may be able to improve and develop on.

While the protocol and protocol security demonstrated in this experiment are to be improved, two “vehicles” successfully communicated and alerted a driver

REFERENCES

- [1] “Vehicle-to-vehicle communication.” [Online]. Available: <https://www.nhtsa.gov/technology-innovation/vehicle-vehicle-communication>
- [2] T. Stone, “Haas safety cloud to be enhanced with applied information v2x data,” Mar 2023. [Online]. Available: <https://www.traffictoday.com/news/connected-vehicles-infrastructure/haas-safety-cloud-to-be-enhanced-with-applied-information-v2x-data.html>
- [3] “Safety cloud, cv2x technology that makes roads safer & smarter.” [Online]. Available: <https://www.haasalert.com/solutions>
- [4] “The vehicle-to-grid pilot project has been inaugurated at mirafiori,” Sep 2020. [Online]. Available: <https://www.media.stellantis.com/em-en/e-mobility/press/the-vehicle-to-grid-pilot-project-has-been-inaugurated-at-mirafiori>
- [5] A. J. Hawkins, “The auto industry lost its spectrum fight with the fcc because v2v was always a fantasy,” Aug 2022. [Online]. Available: <https://www.theverge.com/2022/8/12/23303191/car-v2v-fcc-spectrum-wifi-court-ruling>
- [6] J. Young, “Lane departure warning, blind spot detection help drivers avoid trouble.” [Online]. Available: <https://www.iihs.org/news/detail/stay-within-the-lines-lane-departure-warning-blind-spot-detection-help-drivers-avoid-trouble>