
Exterior Car Parking Assistance Algorithm Based on Reconfigurable System for Future Industry

Sethakarn Prongnuch^{1,*} and Suchada Sitjongsataporn²

¹*Department of Computer Engineering, Faculty of Industrial Technology, Suan Sunandha Rajabhat University, 1 U-Thong Nok Rd., Dusit, Bangkok, Thailand*

²*Department of Electronic Engineering, Mahanakorn Institute of Innovation (MII), Faculty of Engineering, Mahanakorn University of Technology, 140 Cheumsamphan Rd., Nongchok, Bangkok, Thailand*

E-mail: sethakarn.pr@ssru.ac.th; ssuchada@mut.ac.th

**Corresponding Author*

Received 29 November 2019; Accepted 25 May 2020;
Publication 17 August 2020

Abstract

A car accident while parking the car is caused by the car driver, who is invisible around the car. However, there are no solutions for parking assistance when the driver is outside the car. The objective of this paper is to propose a reconfigurable embedded system design by voice controlled parking assistance system for a prototype electric vehicle connected to a smartphone via Bluetooth. Hardware and software co-design using the Xilinx VIVADO as a software design tool is introduced. We design the hardware and software on an ARM multicore processor and the reconfigurable system board model ZYBO: XC7Z010 by considering it as hardware accelerator. The hardware of the proposed voice controlled exterior car parking assistance system is installed on the miniature electric vehicle. The experiments are tested successfully at the parking area for both reverse parking and reverse parallel parking. This proposed system is better suited for users so that they can control their car comfortably while parking safely.

Journal of Mobile Multimedia, Vol. 16_1-2, 203–220.

doi: 10.13052/jmm1550-4646.161210

© 2020 River Publishers

Keywords: Exterior car parking assistance, embedded system, hardware software co-design, reconfigurable system.

1 Introduction

Based on Ford's survey in Ref. [1], 27% of the Filipino drivers realize that the parallel parking is stressful eventually even when provided with a free parking space and 53% of the Thai drivers sense the reverse parking as the stressful situation in a narrow parking space. However, car parking is a major problem especially in crowded and cramped urban areas.

A parking-assistance system assists the drivers' performance in parking their vehicles. The aim of the parking-assist system is to enhance parking safety with efficiency. There are various technologies available for implementing parking assistance, which allows parking in a spot located in a car park and at the roadside such as an autonomous path planning for road vehicles in narrow environments [2], an advanced driving assistance system for electric vehicles [3], a semi-autonomous parking assist system [4], and a park-assistance application [6].

Exterior car parking assistance is a kind of crash protection technology that causes a change in reducing car accident and keeping a driver safe as shown in Figure 1. However, as the driver stays outside the car while parking,



Figure 1 Exterior car parking assistance.

the driver's blind spot reduces significantly. This idea can help the drivers to park and maneuver their cars comfortably and safely.

The Xilinx VIVADO is a software design tool [11] for a co-design system for which more methodology is needed to leverage C/C++ based design and optimization, IP sub-system, and integration of accelerated system. Enabling designers to work at a high level of abstraction would facilitate the design reuse. There are many features that could be improved by an engineer and/or a developer, such as accelerating implementation, verification, and high-level design.

The ZYBO board from the Digilent Inc. [12] is an embedded development platform built around the smallest member of the Xilinx Zynq 7000 family. The XC7Z010 is based on the Xilinx All Programmable System-on-Chip architecture, which integrates an ARM Cortex-A9 multicore processor with field programmable gate array (FPGA) logic. There are many components such as memories, video and audio I/O, USB, Ethernet port, and SD card. Additionally, six Pmod connectors are available to put any design easy to use.

In this paper, we introduce the exterior car parking assistance system, which designs the hardware architecture for embedded processing. We organize this paper as follows. Section 2 describes about the background and related works. Section 3 proposes the overview of hardware and software co-design systems. Section 4 conducts the results of experiments in the parking area. Section 5 summarizes the paper.

2 Background and Related Works

There are various technologies available with regards to implementation of driver-assist systems. A combined approximation solution for path planning in narrow environments [2] is conducted. This approach includes a global planner for generating a preliminary path and a local planner for feasible creation of the preliminary path.

In Ref. [3], the automation of neighborhood electric vehicle with the embedded distributed architecture is presented for advanced driving assistance system. For hardware architecture, there are two layers of the embedded processing systems as follows: the low-level system manages all electric signals, while the high-level system executes the driver-assist functionalities. In Ref. [7], the authors have developed the lane parking detection and tracking system using GPS to position the car as well as navigation with miniature self-driving car on Raspberry Pi.

Regarding improving the driver's skill on parking, the authors in Ref. [8] focus on skill improvement in reverse parking and refer on timing for steering. The driver can follow the direction from the target trajectory based on auditory-assist method during reverse parking. A semiautonomous parking assist system [4] is proposed, and the components' design of the system is based on the kinematic vehicle models. Based on the machine vision technology, a nearest point detection approach between the obstacle and automobile vehicle [5] is presented by automobile tail fitting with a polygon model, which can provide information for safe parking.

In Ref. [6], the authors present an emerging park-assistance application based on multiagent virtual environment by considering the evaluation of trajectory which enables decision making for vehicle guidance. Based on the in-vehicle monitor, top-view image of environment outside the vehicle using fish-eye cameras parking systems operated by Raspberry Pi is presented in Ref. [9] implying that the driver can see the surround view around the automobile. It paves way for a comfortable and safe driving while parking.

The embedded system design for car parking prototype by voice control is presented in Ref. [10]. The goal of our research is safety first so that it considers problems at the car park, while the driver cannot see around a car by himself/herself when staying in a car. These problems can be solved by using the embedded system for car parking prototype by voice control. The experimental results achieved by comparison between original voice and received voice command via a Bluetooth device showed averaging amplitudes of nearly 45 kHz and 50 kHz. The system design can be applied to a car prototype in a future work.

3 Overview of Hardware and Software Co-Design Systems

In this section, we propose an overview of the hardware and software co-design system including the embedded systems architecture following the embedded system theory [13], and design and tools that will be orchestrated and deployed in the system.

There are four parts that are involved in the design system such as the ZYBO board, Arduino UNO board, Bluetooth module, and sensors installation for controlling the procedure of parking assistance using the Xilinx VIVADO and XSDK (X Software Development Kit). The software is controlled by 7 algorithms that run on the ZYBO and Arduino UNO boards for parking-assistance procedure through voice command.

3.1 Hardware Design

The hardware design part is depicted in Figure 2. We use the Xilinx VIVADO as a software design tool for design and generating hardware on programmable logic area or reconfigurable areas inside the ZYBO board. There are six hardware devices in our system: PmodBT2 is the Bluetooth device, Pmod DHB1 is a motor drive, Pmod JC5 is a buzzer, Pmod JC2-4 is a Servo motor, and 2-Pmod MAXSONAR are the ultrasonic sensors. There are five steps for design and generating process of the Xilinx VIVADO that consists of (1) creating an IP integrator design, (2) debugging the block design, (3) generating HDL design files, (4) to implement design and generate, and (5) to export all hardware to the XSDK as described in the next section.

The hardware architecture design for the proposed system by using the Xilinx VIVADO is shown in Figure 3. The design includes ZYNQ7 processing system as an ARM multicore processor, Processor System Reset, Advanced eXtensible Interface (AXI) Interconnect, Pmod DHB1 as a motor

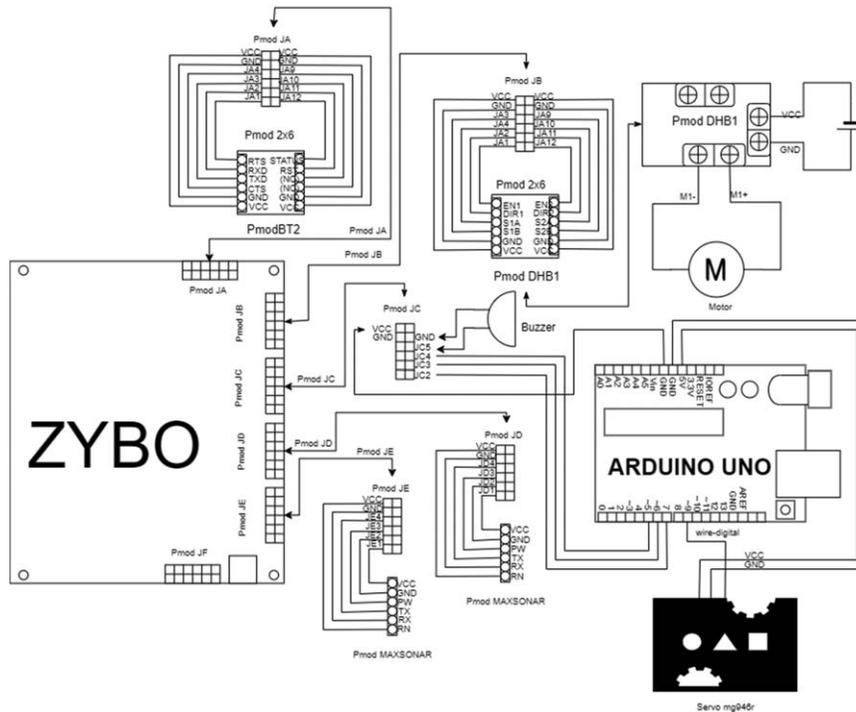


Figure 2 System design overview

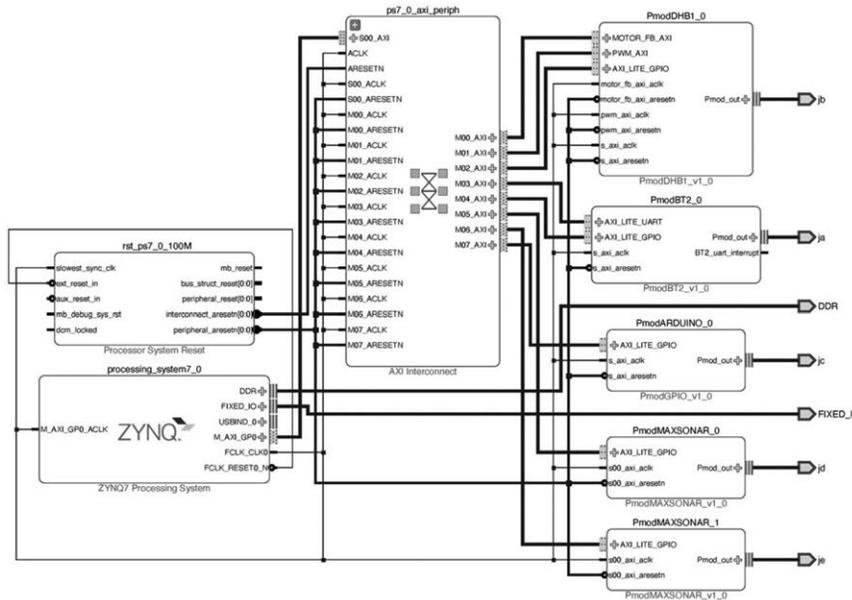


Figure 3 Architectural design on Xilinx VIVADO.

drive, and Pmod ARDUINO includes a buzzer and a Servo motor controlled through the Arduino UNO. In the design, PmodBT2 is the Bluetooth device, and the 2-Pmod MAXSONAR are the ultrasonic sensors, i.e., DDR RAM and fixed I/O.

3.2 Software Design

This section presents the algorithm design that consists of the ZYBO and Arduino UNO board for the parking-assistance procedure through voice command.

The main algorithm of the ZYBO is proposed in Algorithm 1. Lines 1–5 show the initial devices as motor drive, Bluetooth device, 2-ultrasonic sensors, buzzer, and the Servo motor. Line 6 shows all steps for car parking through voice command such as forward (FW), backward (BW), turning left (LEFT), turning right (RIGHT), READY, and STOP. Line 7 shows the loop process of the ZYBO communication systems.

The ultrasonic sensor operation is kept in abeyance for data to be received in decimal number from the two ultrasonic sensors. If the distance between the car in the front or rear bumpers and nearby obstacles is less than 6 cm, then

Algorithm 1 ZYBO main algorithm

```

1: INITIAL Motor Drive
2: INITIAL Bluetooth Device
3: INITIAL 2 Ultrasonic Sensors
4: INITIAL Buzzer
5: INITIAL Servo Motor
6: INITIAL Voice Command[FW, BW, LEFT, RIGHT, READY, STOP]
7: FOR LOOP
8:   Ultrasonic Sensors()
9:   Bluetooth Device()
10:  Comparing Voices()
11: END FOR LOOP

```

the buzzer will alarm. The proposed ultrasonic sensors controlled algorithm is introduced in Algorithm 2. Line 1 shows the two ultrasonic sensors. Line 2 shows the input data from the ultrasonic sensors where DEC_{US} is the data in decimal from the ultrasonic sensors. Lines 3–5 show the decision to stop the motor drive and turn on the buzzer.

Algorithm 2 Ultrasonic sensors controlled algorithm

```

1: Ultrasonic Sensors() {
2: INPUT data ( $DEC_{US}$ ) from the 2 of Ultrasonic Sensors
3: IF data ( $DEC_{US}$ )  $\leq$  6 cm
4:   OUTPUT Motor Drive() = '0'
5:   Turn ON Buzzer()
6: END IF }

```

The proposed buzzer-controlled algorithm is introduced in Algorithm 3. Line 1 shows the buzzer. Line 2 shows the input data from ultrasonic sensor operation, where BIT_{BZ} is the data in bit. Lines 3–6 show the decision to turn on the alarm buzzer.

Algorithm 3 Buzzer controlled algorithm

```

1: Buzzer() {
2: INPUT data ( $BIT_{BZ}$ )
3: IF data ( $BIT_{BZ}$ ) == '1'
4:   OUTPUT Buzzer is Turn ON
5: ELSE Buzzer is Turn OFF
6: END IF }

```

For the maneuver of the vehicle, the motor drive controls the mechanism on the transmission shaft and further controls forward or backward maneuver of the vehicle. The proposed motor drive controlled algorithm is introduced in Algorithm 4. Line 1 shows the motor drive parameter. Lines 2–8 show the decision for a moving vehicle, where 2's $\text{COMPLEMENT}_{\text{MD}}$ as input data of motor drive in 2's complement.

Algorithm 4 Motor drive controlled algorithm

```

1: Motor drive() {
2:   INPUT data (2's  $\text{COMPLEMENT}_{\text{MD}}$ )
3:   IF data (2's  $\text{COMPLEMENT}_{\text{MD}}$ ) == '1'
4:     Motor Drive is Turn ON and Rotate Forward
5:   ELSE IF data (2's  $\text{COMPLEMENT}_{\text{MD}}$ ) == '-1'
6:     Motor Drive is Turn ON and Rotate Backward
7:   ELSE Motor Drive is Turn OFF
8:   END IF }

```

A voice–user interface makes spoken human interaction with hardware using speech or voice recognition to understand spoken commands and to send the data to the parking-assist system. The proposed Bluetooth device controlled algorithm is introduced in Algorithm 5. Line 1 shows the Bluetooth device parameter. Line 2 shows the input data from the Bluetooth device where the BYTE_{BD} is the data in byte. Lines 3–6 show the decision to convert voice data into character.

Algorithm 5 Bluetooth device controlled algorithm

```

1: Bluetooth Device() {
2:   INPUT data ( $\text{BYTE}_{\text{BD}}$ )
3:   IF data ( $\text{BYTE}_{\text{BD}}$ ) > 0
4:     CONVERT data ( $\text{BYTE}_{\text{BD}}$ ) to CHAR
5:     Data New Voice Commands[256] = CHAR
6:   END IF }

```

For the voice command system, the speech patterns of commands are stored on the SD card and loaded into memory when the software is run. A comparator checks these stored patterns against the voice command. An algorithm of the comparator includes a function to find correlation coefficient between two arrays of voice and new voice commands. The proposed comparing voices algorithm is introduced in Algorithm 6. Line 1 shows the comparing voices parameter. Line 2 shows the input data. Lines 3–18 show

the decision to compare voice command with stored pattern. The results will be sent to the control motor drive and the Servo motor.

According to turning radius of wheel in reverse parking, we calculate the radius of the turning wheel (Rad_t) based on the Bežier curve following Ref. [6] as

$$Rad_t = \sqrt{\frac{Wheel_b^2}{\sin^2\alpha - Wheel_b^2} - \frac{Width}{Wheel_b}}, \quad (1)$$

where $Wheel_b$ is the wheel base and $Width$ is the width of the vehicle in meters. The maximum angle (α) of the steering wheel is in degrees.

Algorithm 6 Comparing voices algorithm

```

1: Comparing Voices() {
2:   INPUT data New Voice Commands[256]
3:   IF data New Voice Commands[256] == Voice Commands[FW]
4:     Motor Drive() = '1'
5:   ELSE IF data New Voice Commands[256] == Voice Commands[BW]
6:     Motor Drive() = '-1'
7:   ELSE IF data New Voice Commands[256] == Voice Commands[LEFT]
8:     Motor Drive() = '0'
9:     Servo Motor() = 'S2'
10:  ELSE IF data New Voice Commands[256] == Voice Commands[RIGHT]
11:    Motor Drive() = '0'
12:    Servo Motor() = 'S3'
13:  ELSE IF data New Voice Commands[256] == Voice Commands[READY]
14:    Motor Drive() = '0'
15:    Servo Motor() = 'S1'
16:  ELSE IF data New Voice Commands[256] == Voice Commands[STOP]
17:    Motor Drive() = '0'
18:  END IF }

```

The proposed Arduino control Servo motor algorithm is introduced in Algorithm 7. Line 1 shows the Servo motor parameter. Line 2 shows the three initial states of the Servo motor, where S1 is the READY state, S2 is the LEFT state, and S3 is the RIGHT state. Line 3 is the initial delay parameter at 1000 ms. Line 4 shows the input data of the Servo motor, where the BITVECTOR_{SM} is the data in the bit vector. Lines 5–12 show the decision to control each state of the Servo motor.

All algorithms are written and compiled to execute the source code file loaded to processor system (PS) by using the SDK as a software tool from the Xilinx. The binary file is a product of design and generated from the Xilinx

Algorithm 7 Arduino control servo motor algorithm

```
1: Servo Motor() {  
2:   INITIAL 3 States of Servo Motor (S1, S2, and S3)  
3:   INITIAL Delay = 1000ms  
4:   INPUT data(BITVECTORSM)  
5:   IF data(BITVECTORSM) == S1  
6:     Degree of Servo Motor = 0  
7:   ELSE IF data (BITVECTORSM) == S2  
8:     Degree of Servo Motor = -90  
9:   ELSE IF data (BITVECTORSM) == S3  
10:    Degree of Servo Motor = 90  
11:  ELSE Degree of Servo Motor = 0  
12:  END IF }  
}
```

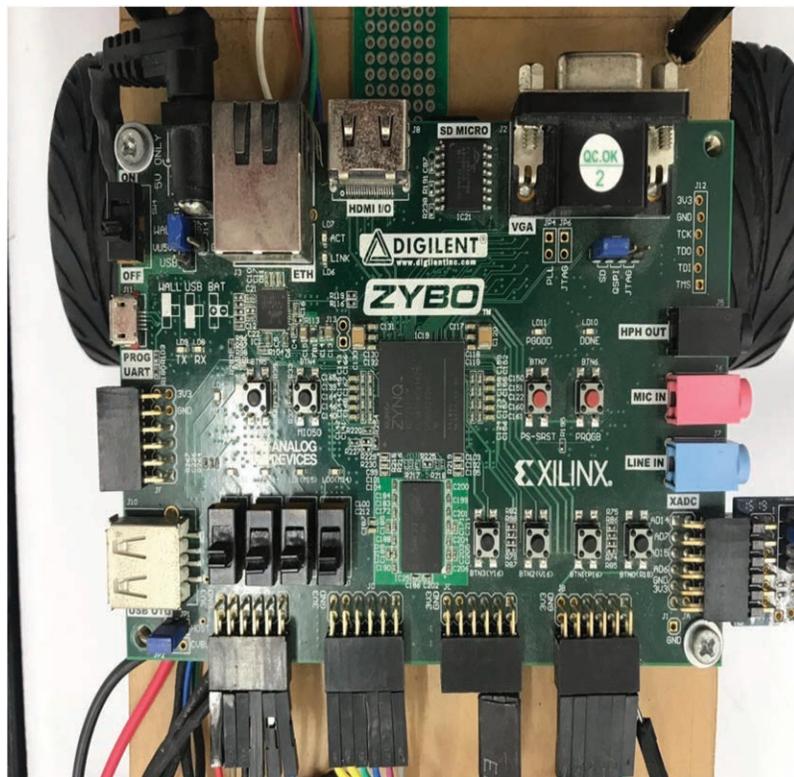


Figure 4 ZYBO board wiring and installation.

VIVADO loaded to the programmable logic (PL). The binary file and the execution files are implemented on the ZYBO board via the JTAG connector.

3.3 The Implementation

The proposed co-design system has been validated by implementing on a miniature electric vehicle system as the ZYBO board in Figure 4 and the Arduino UNO board wiring with sensors and installation as shown in Figure 5. Table 1 shows miniature electric vehicle properties such as overall

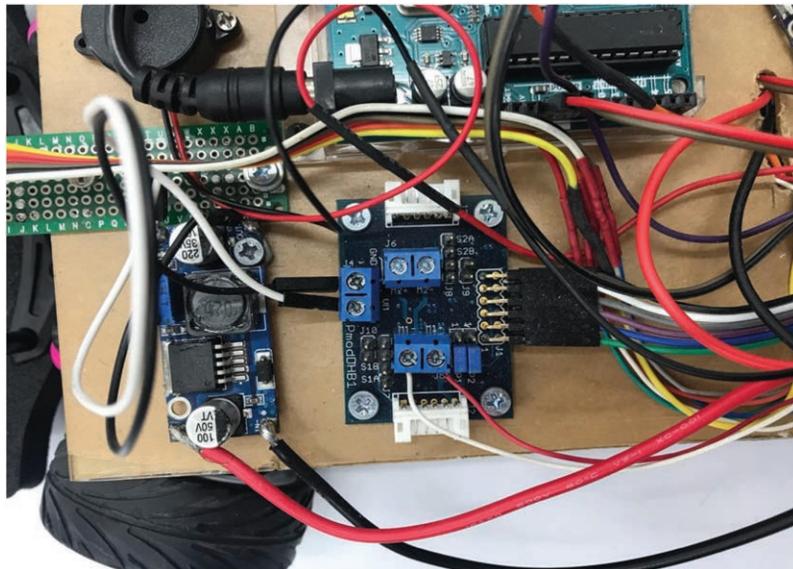


Figure 5 Arduino board wiring and installation.

Table 1 Miniature electric vehicle properties

Specification	Description
Overall width	234 mm
Overall length	434 mm
Type of motor	DC Motor Gear
Max. gross vehicle weight	1.75 kg
Kerb weight	1.38 kg
Wheelbase	257 mm
Radius of turning wheel	22 mm
Tire type	Rubber
Front/Rear wheel size	163/161 mm

width, overall length, type of motor, maximum of gross vehicle weight, kerb weight, wheelbase, radius of the turning wheel, type of tires, and front/rear wheel sizes.

4 Experiments and Results

For the experimental design, we form two parts consisting of an environmental setup and the operation with voice commands as follows.

In the environmental setup, there are three experiments including (1) the communication testing between the bluetooth device and the ZYBO board, (2) the ultrasonic devices communication with the ZYBO board testing, and (3) to construct the angles of 0-degree, 90 degrees, and -90 degrees for rotating Servo motor testing. All experiments were set up the in test execution at the parking area in the university. More than 85% of the results from experiments at the parking area and indoor presented in Figure 6 is achieved effectively.

For the experimental part of operation with voice commands, there are six voice commands such as forward (FW), backward (BW), left, right, ready, and stop. The aim of this part is that we focus on testing the voice controlled

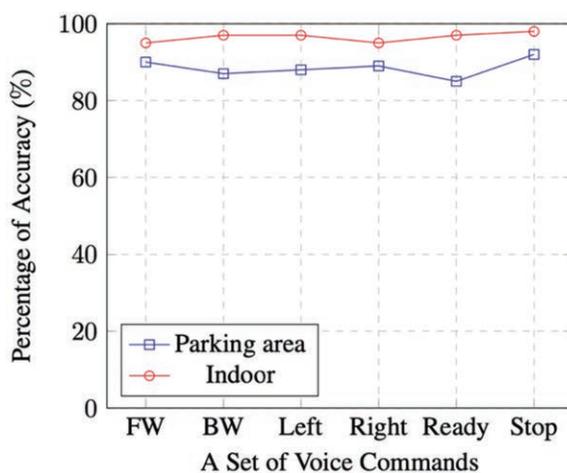


Figure 6 Percentage accuracy of voice commands tested at the different areas

exterior car parking assistance. The results of such an operation with voice commands are shown in Table 2. There are two parking patterns for testing: the reverse parking and reverse parallel parking. In each pattern of parking, there are five formats.

The success pattern in the reverse parking method is the “LEFT- BW-RIGHT-FW-LEFT-BW-STOP-READY-BW” as shown in Figure 7 and the success pattern in the reverse parallel parking method is the “LEFT-BW-RIGHT-BW-STOP-READY-FW-STOP” as depicted in Figure 8.

The limitation of the proposed design is with regard to brake lag, which is the time taken to apply a brake to the miniature vehicle, and that added an additional 159.6 mm to the total stopping distance.

Table 2 Operation with voice commands

Parking Method	Pattern	Description of voice commands
Reverse parking	1	LEFT-BW-STOP-READY-RIGHT-FW-STOP-LEFT-BW-STOP-READY-BW
	2	LEFT-BW-RIGHT-FW-STOP-LEFT-BW-STOP-READY-BW
	3	LEFT-BW-RIGHT-FW-STOP-READY-LEFT-BW-STOP-READY-BW
	4	LEFT-BW-RIGHT-FW-LEFT-BW-STOP-READY-BW
	5	LEFT-BW-STOP-RIGHT-FW-STOP-LEFT-BW-STOP-READY-BW
	Success	LEFT-BW-RIGHT-FW-LEFT-BW-STOP-READY-BW
Reverse parallel parking	1	LEFT-BW-RIGHT-BW-STOP-FW-STOP-READY-FW-STOP
	2	LEFT-STOP-RIGHT-BW-STOP-READY-FW-STOP
	3	LEFT-BW-STOP-RIGHT-BW-STOP-READY-FW-STOP-BW-STOP
	4	LEFT-BW-STOP-RIGHT-BW-STOP-FW-STOP-READY-FW-STOP
	5	LEFT-BW-RIGHT-BW-STOP-FW-STOP-READY-FW-STOP
	Success	LEFT-BW-RIGHT-BW-STOP-READY-FW-STOP

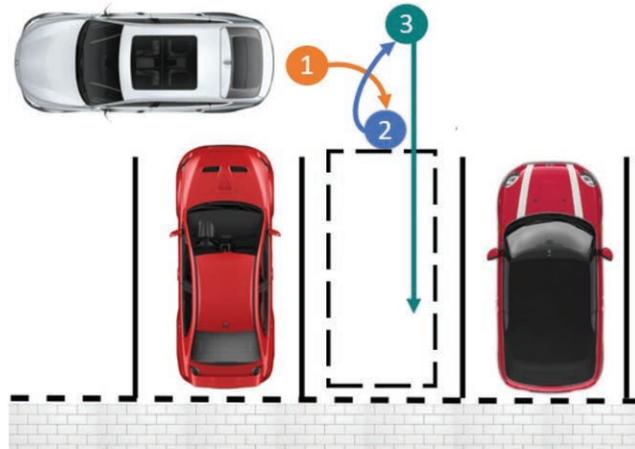


Figure 7 Reverse parking

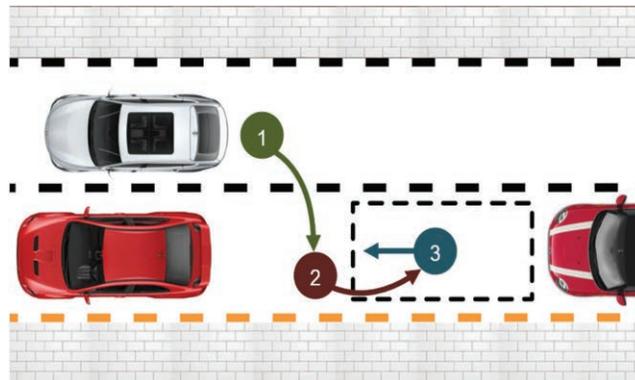


Figure 8 Reverse parallel parking

5 Conclusion

This paper presents an exterior car parking assistance system with hardware reconfigurable system board and software co-design using the Xilinx VIVADO. The proposed system designs an embedded system architecture following the embedded system theory including the ZYBO board, Arduino UNO board, bluetooth module, and sensors' installation for controlling the procedure of parking assistance by voice control. The successful pattern in the reverse parking method is the "LEFT-BW-RIGHT-FW-LEFT-BW-STOP-READY-BW" and the successful pattern in the reverse parallel parking

method is the “LEFT-BW- RIGHT-BW-STOP-READY-FW-STOP” that the driver outside the car can effectively use to park his/her car comfortably and safely by voice control.

Acknowledgment

We would like to thank Thodsaporn Chuatai, Jakkit Wongtewee, and Thatthep Pansungnoen of the Department of Computer Engineering, Faculty of Industrial Technology, Suan Sunandha Rajabhat University in Bangkok, Thailand for their support.

References

- [1] J. Deakin. Ford’s survey reveals number one stress for PH drivers: Traffic Con- gestion. [Online]. Available: <https://jamesdeakinph.wordpress.com/2015/09/10/fords-survey-reveals-number-one-stress-for-ph-drivers-traffic-congestion/> [Accessed Nov. 22, 2019].
- [2] D. Kiss and G. Tevesz. Autonomous Path Planning for Road Vehicles in Narrow Environments: An Efficient Continuous Curvature Approach. *Journal of Advanced Transportation*, Article ID 2521638, 2017.
- [3] P. Munoz-Benavent, L. Armesto, V. Girbes, J.E. Solanes, J. Dols, A. Muniz and J. Tornero. Advanced Driving Assistance Systems for an Electric Vehicle. *International Journal of Automation and Smart Technology*, 2(4): 329–338, 2012.
- [4] E. Szadeczky-Kardoss and B. Kiss, I. Wahl. Design of a semi-autonomous parking assist system. In *Proceedings of the European Control Conference*, Budapest, Hungary, 2009.
- [5] P. Chen and G. Li. A detection method of nearest point of automobile distance in parking assist system. In *Proceedings of the Chinese Automation Congress*, Jinan, China, 2017.
- [6] B. Dafflon, J.M. Contet, F. Gechter and P.O Gruer. Toward a Reactive Agent Based Parking Assistance System. In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, Athens, Greece, 2012.
- [7] B. Blaga, M. Deac, R. W. Y. Al-doori, M. Negru and R. Danescu. Miniature Autonomous Vehicle Development on Raspberry Pi. In *Proceedings of the IEEE International Conference on Intelligent Computer Communication and Processing*, Romania, pp. 229–236, 2018.

- [8] M. Yukawa, K. Sonoda and T. Wada. Auditory Assist Method to Indicate Steering Start Timing in Reverse Parking for Improvement of Driver Performance. *IEEE Transactions on Intelligent Vehicles*, 2019.
- [9] D. Buljeta, M. Vranjes, Z. Marceta and J. Kovacevic. Surround view algorithm for parking assist system. In *Proceedings of the Zooming Innovation in Consumer Technologies Conference*, Serbia, 2019.
- [10] S. Prongnuch, M. Ardwichai, N. Pong-ngam, A. Chana, A. Thitinaruemit, P. Uttaphut, and N. Areerachakul. Embedded System Design for Car Park Prototype by Voice Controlled (in Thai). In *Proceedings of the SSRU National Conference*, Bangkok, 2016.
- [11] Xilinx. Inc. Vivado Design Suite HLx Editions. [Online]. Available: <https://www.xilinx.com/support/documentation/backgrounders/vivado-hlx.pdf>, Accessed on: Nov. 27, 2019,
- [12] Digilent Inc. Zybo Reference Manual. [Online]. Available: <https://reference.digilentinc.com/reference/programmable-logic/zybo/reference-manual>, Accessed on: Nov. 28, 2019.
- [13] S. Prongnuch, S. Sitjongsataporn and T. Wiangtong. A Heuristic Approach for Scheduling in Heterogenous Distributed Embedded Systems. *International Journal of Intelligent Engineering and Systems*, 13(1): 135–145, 2020.

Biographies



Sethakarn Prongnuch received his B.Eng. degree in computer engineering from the Rajamangala University of Technology Phra Nakhon in Bangkok, Thailand in 2011, and the M.Eng. and D.Eng. degrees in computer engineering from the Mahanakorn University of Technology in Bangkok, Thailand in 2013 and 2019, respectively. He has worked as a lecturer at the department of computer engineering, Suan Sunandha Rajabhat University in Bangkok, Thailand since 2013. Currently, he is the head of the computer engineering department of the Suan Sunandha Rajabhat University. His research interests include architectures, applications, education and design techniques for

FPGA technologies, computer architectures and systems, embedded system, and heterogeneous system architecture.



Suchada Sitjongsataporn received her B.Eng. (first-class honors) and D.Eng. degrees in electronic engineering from the Mahanakorn University of Technology, Bangkok, Thailand in 2002 and 2009, respectively. She has worked as a lecturer at the department of electronic engineering, Mahanakorn University of Technology since 2002. Currently, she is an Associate Professor in the Mahanakorn University of Technology since 2019. Her research interests are in mathematical and statistical models in the area of adaptive signal processing for communications, networking, embedded system, and image and video processing.

