ECE532 Digital System Design
Final Report

Remote Control Robot

Group 9

Members

Hanyu Che | Yunhao Lu | Runze Zuo

Contents

1 Overview

1.1 Background

In the modern world, security has become a paramount concern for various industries and sectors, such as private properties, public spaces, and critical infrastructure. Traditional surveillance methods, such as closed-circuit television (CCTV) cameras, have been widely adopted to monitor and secure these areas. However, as technology evolves and new threats emerge, it has become evident that these conventional approaches have certain limitations, particularly when it comes to monitoring complex or inaccessible environments [1].

Recent advancements in robotics and artificial intelligence have paved the way for the development of innovative security solutions to address these limitations. Autonomous robots equipped with advanced sensors and navigation capabilities have shown potential in enhancing security measures in challenging environments [2]. This project aims to build upon these advancements and develop a robotic system that can autonomously patrol areas difficult to monitor using traditional surveillance cameras, providing real-time visual feedback for improved security coverage.

Several studies and projects have demonstrated the feasibility and effectiveness of using robotic systems for security and surveillance applications. For instance, Matsumoto et al. developed an autonomous robot for outdoor patrolling, which could navigate complex environments and transmit real-time video feed [3]. These examples serve as an inspiration for our project and provide valuable insights into the design and development of a robotic patrolling system.

1.2 Goals

Our team decided to design and build a robotic system to enhance security in challenging environments. Which will have the ability of navigating through complex or inaccessible terrains, overcoming the limitations of traditional surveillance methods. We plan to give the robot the ability to navigate through the target area. The system will be designed to capture and transmit visual data in real-time, allowing security personnel to receive accurate and timely information about the patrolled area.

Furthermore, the robotic system will be built with robust and reliable materials to withstand the challenges it may encounter in complex environments, ensuring consistent performance. To maximize its utility, the system will be designed to easily integrate with existing security infrastructure, such as alarms, access control systems, and other surveillance equipment. Additionally, by using the FPGA, we can add different modules to the robot, making it scalable and customizable to meet the specific needs of various industries and sectors, allowing it to address unique security requirements and challenges.

1.3 Block Diagram

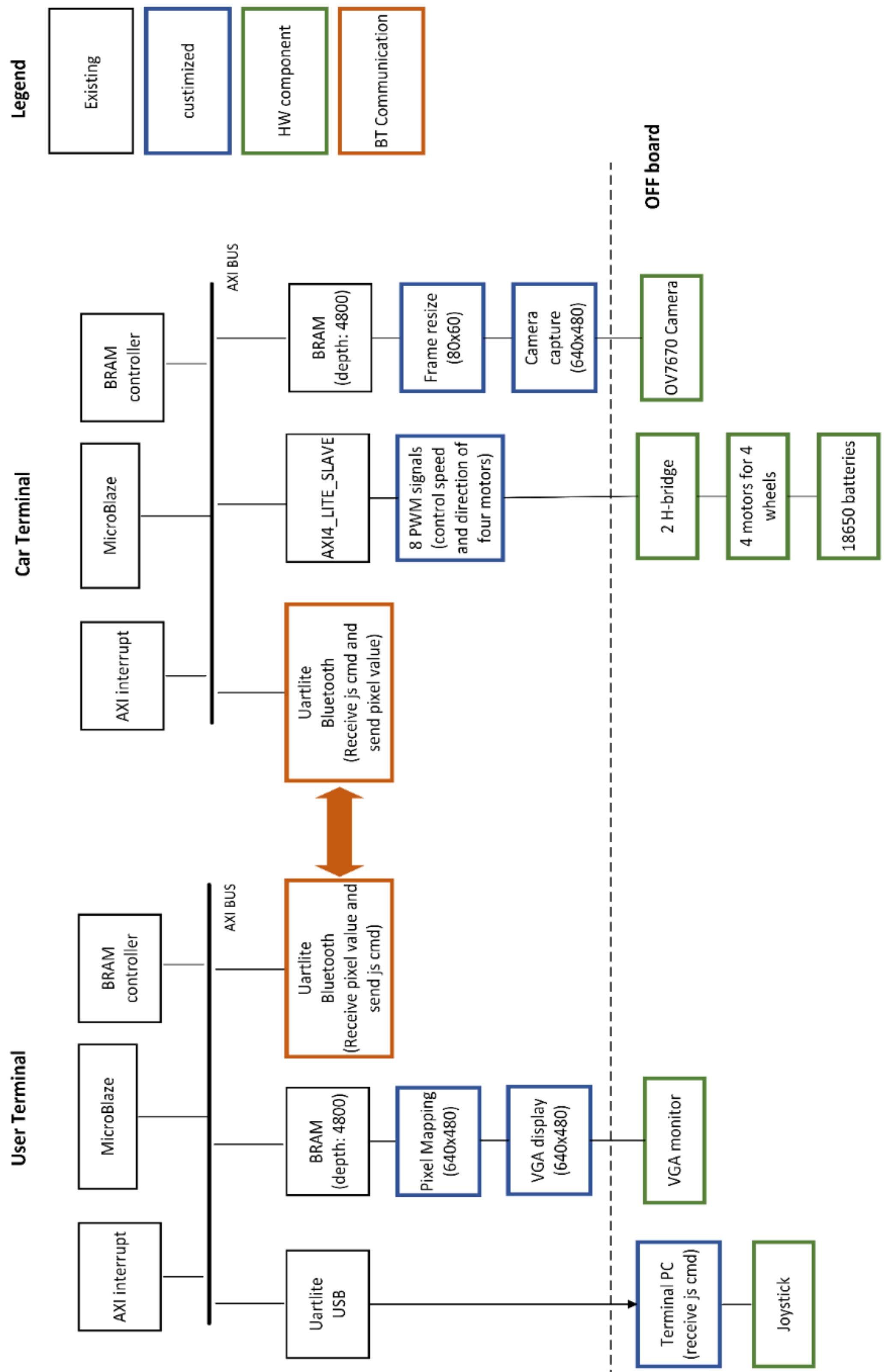

Figure 1 ***Block Diagram of the Project***

1.4 Description of Used IP

IP	Function	Origin
Car Terminal		
MicroBlaze(11.0)	A soft processor core used as the central processing unit for the system, responsible for executing the control algorithms and managing the interactions between various peripherals.	Xilinx
Clocking Wizard(6.0)	A utility to generate and configure clock signals required for the system's operation.	Xilinx
AXI GPIO(2.0)	A module that facilitates general-purpose input/output (GPIO) management for different purposes.	Xilinx
gpio_input_splitter_v1_0(1.0)	Modules that help in splitting and routing input GPIO signals for further processing.	
gpio _splitter_v1_0(1.0)	Modules that help in splitting output GPIO signals.	
AXI Uartlite(2.0)	UART module that manages communication between the system and external devices	Xilinx
AXI Interrupt Controller(4.1)	A module responsible for handling and prioritizing interrupts within the system.	Xilinx
AXI BRAM Controller(4.1)	Modules that provide memory management and storage capabilities for the system.	Xilinx
Block Memory Generator(8.4)		
camera_capture_v1_0(1.0)	Modules responsible for capturing real-time camera input and managing camera settings.	Customized
camera_controller_v1_0(1.0)		Existing[4]
counter_v1_0(1.0)	A module used for counting events or tracking time-based operations.	Customized
PWM_Generator_v1_0	A module that generates pulse-width modulation (PWM) signals for controlling the speed of the patrol car.	Customized
Control Terminal		
MicroBlaze(11.0)	A soft processor core used as the central processing unit for the system, responsible for executing the control algorithms and managing the interactions between various peripherals.	Xilinx
Clocking Wizard(6.0)	A utility to generate and configure clock signals required for the system's operation.	Xilinx
AXI Uartlite(2.0)	UART module that manages communication between the system and external devices	Xilinx
AXI BRAM Controller(4.1)	Modules that provide memory management and storage capabilities for the system.	Xilinx
Block Memory Generator(8.4)		
AXI Interrupt Controller(4.1)	A module responsible for handling and prioritizing interrupts within the system.	Xilinx
VGA	A module that handles visual data and output the camera feedback on a VGA monitor.	Customized

Table 1. List of Used IP

1.5 Description of Used Hardware

Hardware	Function
Car Terminal	
Nxey4 DDR Board	An FPGA development board featuring a Xilinx Artix-7 FPGA, which serves as the primary platform for implementing the project's functionalities.
OV7670 Camera	A compact image sensor module used for capturing real-time video input. The camera data is processed and send to the control terminal via bluetooth.
L298N H-Bridge	A motor driver module responsible for controlling the speed and direction of the patrol car's motors, based on input from the PWM generator and control algorithms
Bluetooth Module	A wireless communication module that enables connectivity between the car terminal and control terminal. On the car terminal the BT module receives the commend for movement and send the captured frame
Control Terminal	
Nxey4 DDR Board	An FPGA development board featuring a Xilinx Artix-7 FPGA, which serves as the primary platform for implementing the project's functionalities.
Bluetooth Module	A wireless communication module that enables connectivity between the car terminal and control terminal. On the control terminal the BT module receives the captured frame and send commands from the joystick after algorithmic processing.
VGA Monitor	A display device used for outputting camera feedback and other visual data from the patrol car system.
PC & Joystick	External devices used for remotely controlling the patrol car via the Bluetooth module. The Control Terminal is connected to the PC via USB UART, as the PC is only used to receive commands from the joystick and transmit them to the Control terminal, while the Joystick serves as a user input device.

Table 2. List of Used Hardware

2 Outcome

2.1 Results

In this project, we are proud to report that we have successfully achieved our primary objectives. Throughout the development process, we encountered a few challenges that required us to adapt and make necessary modifications to the original proposal, but these changes did not hinder the complexity or overall functionality of the project.

One such modification was prompted by the sensitivity of the infrared sensor. Initially, our plan was to incorporate an automatic patrol function that would allow the system to perform its tasks autonomously. However, we discovered that the sensitivity of the infrared sensor would not effectively support this function. To overcome this challenge, we pivoted our approach and opted for a remote-control function instead. This new approach provided a more reliable means for users to interact with and control the system, ultimately enhancing its usability.

Another alteration we made was in communication and data transmission. Our original proposal included a real-time video transmission feedback feature, which would rely on a Bluetooth module. However, we encountered limitations in the bandwidth and transmission rate of the Bluetooth module, which made it difficult to maintain seamless real-time video streaming. To address this issue, we decided to switch to a picture feedback mechanism. This change allowed us to maintain the core functionality of providing visual information to users while eliminating the need for high-bandwidth data transmission.

Despite these adjustments, we are confident that the project remains robust and efficient. In fact, throughout the development process, we have managed to make full use of the modules that were initially planned for the project. By repurposing them in the revised functionalities, we have not only preserved the project's original intent but also enhanced its value through improved reliability and user experience.

In conclusion, our project has successfully achieved its goals, albeit with some necessary modifications to the original proposal. These changes have not diminished the project's complexity or overall functionality. Instead, they have allowed us to create a more reliable and user-friendly system that will provide valuable benefits to its users.

Function	Proposed	Result
Cruise Control	Autonomous Line Following by using IR sensors and preset track on the ground.	Switched to remote control via Joystick
Bluetooth Communications	Allow the User terminal to send commands to the Car/ Car terminal feedback live video to the User.	Changed to feedback photos due to transmission limits
Camera Capture	Real time camera input to the FPGA board	Successful
VGA Output	Output the Camera Feedback on the VGA monitor	Successful
Speed Control	The Car can move at different speeds depending on the user's actions on Joystick	Successful

Table 3. List of the Proposed Functions and the Result

2.2 Future Improvements

As we look to the future of our project, we aim to enhance its capabilities by incorporating advanced technologies, optimizing the system, and making it more efficient and user-friendly.

One of our primary future goals is to introduce more automated features, such as Auto Cruise and Object Detection. By leveraging machine learning algorithms and advanced sensor technologies, we can enable the system to recognize and respond to its environment more effectively. This will reduce the need for constant user input and enhance the overall performance of the project. Furthermore, the integration of machine learning techniques will allow the robot to identify and classify objects within its patrol area more accurately, empowering it to react appropriately to different situations and provide users with valuable insights.

In addition to automation, we recognize the need to overcome the limitations of the current Bluetooth module for data transmission. We plan to explore alternative data transmission methods that can support higher bandwidth and more reliable connections, ultimately enabling us to provide better feedback to users, such as real-time video streaming. This improved feedback will significantly enhance the overall user experience.

To expand the patrol area covered by our system, we will develop a multi-robot control feature. This feature will allow a single user terminal to coordinate and control multiple robot units simultaneously, enabling efficient patrolling and monitoring of larger areas. By accommodating scenarios where extensive surveillance or coverage is required, such as industrial complexes or large public spaces, our project will be better suited to cater to a diverse range of applications.

By implementing these improvements and focusing on incorporating advanced technologies and optimizing system performance, our project will become even more capable, versatile, and user-friendly.

2 Project Schedule

Milestone	Proposed Plan	Actual Plan
1	Gathering functional requirements; Ordering mechanical components;	Finished same as proposed; Built the robot prototype; Work on VGA display followed by tutorial;
2	Initial design implementation	Finished same as proposed; Controlled the robot by pushbuttons; Display real-time video but had configuration issue;
3	Robotics parts: Developing IR sensor to detect black line; Connecting Bluetooth	Finished as proposed partly; Implemented BT on Arduino firstly to test the top-level functionality, then transfer to Nexys4 board through UART-IP so that laptop could connect board wirelessly; Implemented the IR sensor with Pmod IP and finished SW code in Microblaze, but stuck because lack of reflectance sensor – bought the third-party sensor;
Mid demo	Combined IR sensor with car so it could move by following the black line	Not meet as proposed but had progress. Changed the main task to remote control from line-following because the third-party sensor didn't show a good functionality; Controlled robot through Bluetooth; Displayed the real-time video with right configuration.
4	Dividing the whole camera module to capture and display for each terminal; Developing the PWM module for motor control;	Finished same as proposed. Decreased the active pixels from 640x480 to 80x60 for BT transmission faster, Also splitting the one BRAM to two BRAMs for capturing and displaying; Had testbench for PWM to ensure the correct functionality.
5	Integration the whole system	Finished same as proposed. Implemented the interrupt system for Bluetooth transmission; Pixel value could be transferred from car to laptop through BT and displayed the image on VGA; Joystick command could be transferred from laptop to car through BT; Joystick command could be used for PWM control through the AXI-Lite slave.
6	Overall system test	Finished same as proposed. Fixed the bug when transmission mutually because there is conflict when interrupt the two terminals; Had two interrupt systems for the two transmissions – pixel and joystick command.

4 Description of the System Components

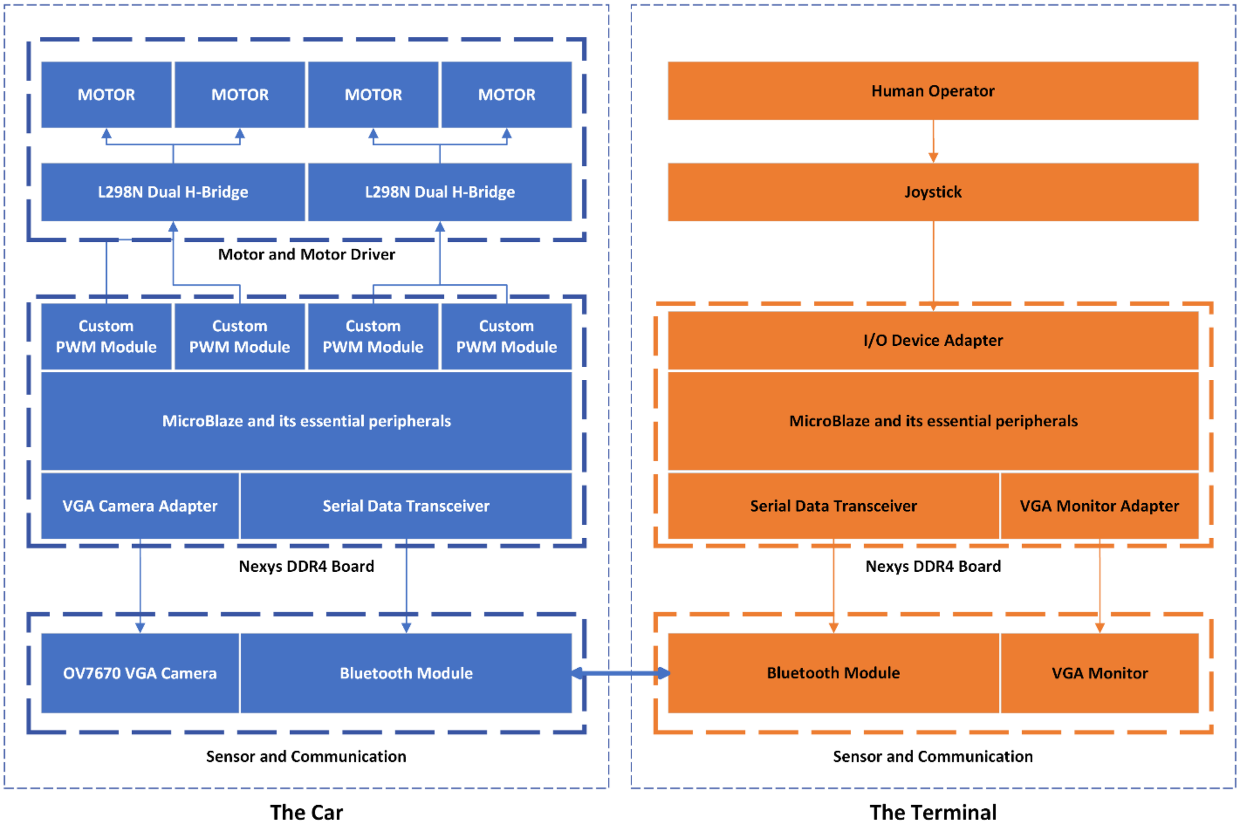

Figure 2 Exterior of the overall system progress

4.1 Car Terminal

4.1.1 Car Mechatronics Introduction

A model car kit was used to build the car, which came with four motors and the car body, providing a solid foundation for our custom project. To effectively control and operate the motors, we installed two L298N Dual H-Bridge motor controllers, which offer a robust solution for driving DC stepper motors for both sides. To power the model car, we opted for two 18650 lithium batteries. These high-capacity, rechargeable cells with large energy density ensure that our car will have sufficient runtime. Given the need to securely mount various electronic components onto the car, such as the Nexys 4 DDR FPGA board, the Bluetooth module, and the camera, we decided to custom design and 3D print a fixture. The result was a well-integrated, secure, and functional model car that combined the original kit with our additional customizations and enhancements.

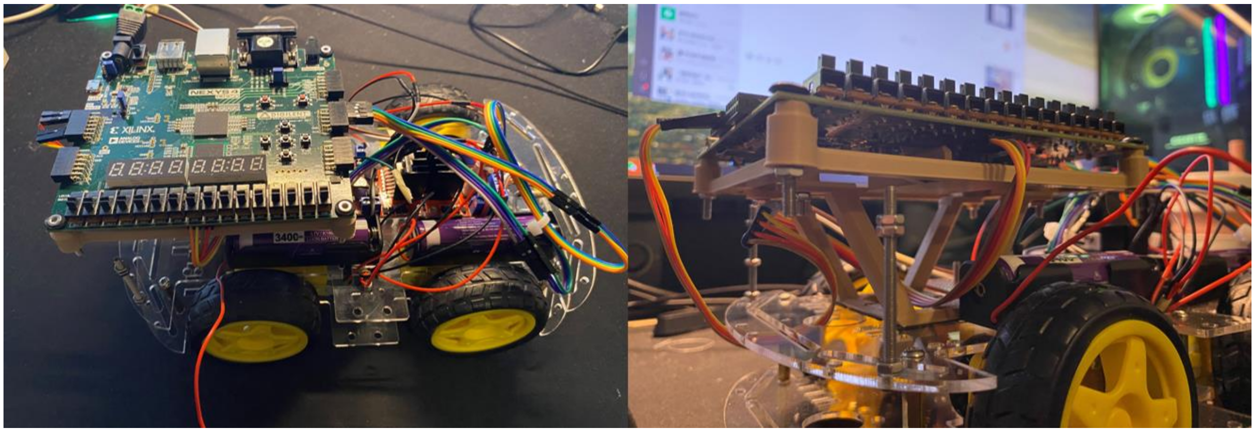

Figure 3 Exterior of the Car Terminal

4.1.2 Camera

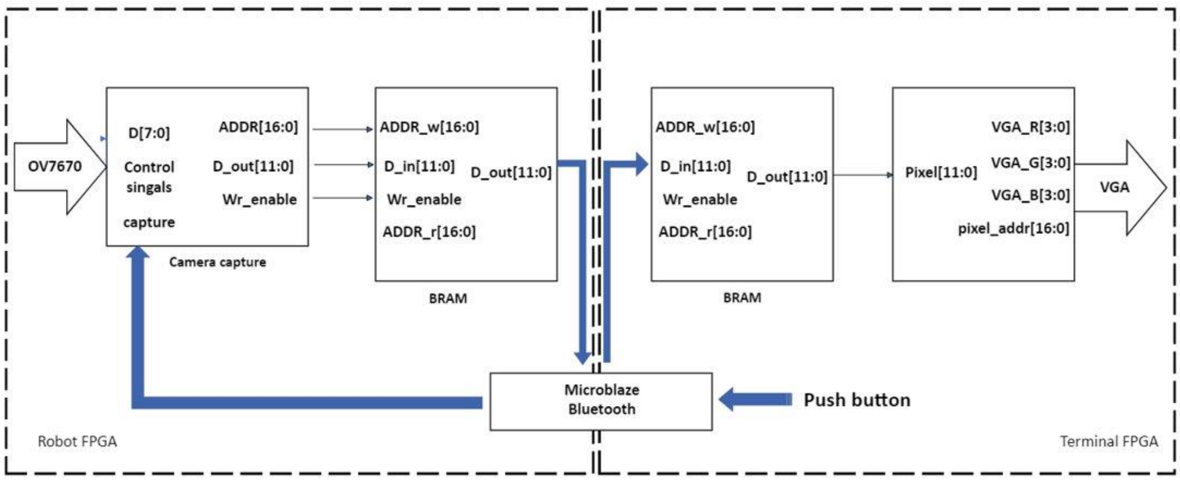

Figure 4 The communication between camera capture and VGA display

An OV7670 VGA camera on robot FPGA is used to capture the frame. One pixel value is represented as two consecutive input value D. In the camera capture module, we also downsize the frame to 60x80 pixels for good frame transmission time through Bluetooth. Also, there is a camera capture module communicate the camera and FPGA board. The i2c protocol is used for control signals communication such as sioc, siod, power_down etc. Also, one camera register configuration module is developed to tell i2c controller which register and what data should be stored in camera. To get the correct frame color, we used RGB444 as the configuration.

The camera is controlled by a pushbutton on terminal FPGA board. When push the bottom pushbutton, one picture will be taken and stored in BRAM. Through Bluetooth, every pixel value is transferred to the Terminal FPGA and waited to be displayed.

4.1.3 Memory and Buffer

The incoming camera frame is directly stored in a block memory. This memory module is configured as a dual-port system, allowing the camera to write to the memory while the Microblaze processor concurrently reads from it.

4.1.4 Bluetooth Communications on car terminal

The car obtains control commands from the control terminal via Bluetooth. The AXI Uartlite IP is utilized to access the Bluetooth module using the Microblaze processor. The receiving process is interrupt-based. The Bluetooth module on the car receives a control package (8 Bytes + 1 Verify Byte) at a rate of 10Hz. After obtaining a package, the Microblaze processor either calculates the necessary input for the PWM module that propels the car or captures a frame and returns it to the control terminal.

The car also employs Bluetooth to transmit frames to the control terminal. Each pixel demands three bytes, comprising RGB444 and verification bits. To guarantee data integrity, the car initiates a handshake before sending pixels, which involves transmitting a specific string and receiving an acknowledgement string. This handshake is also executed after all pixels have been sent.

4.1.5 Cruise Control and PWM Generator

A robust motor remote control system has been implemented for a 4-motor vehicle using a Pulse Width Modulation (PWM) generator. This system leverages PWM for motor control, an AXI-lite interface for communication, and obtains user inputs from a Bluetooth-connected joystick.

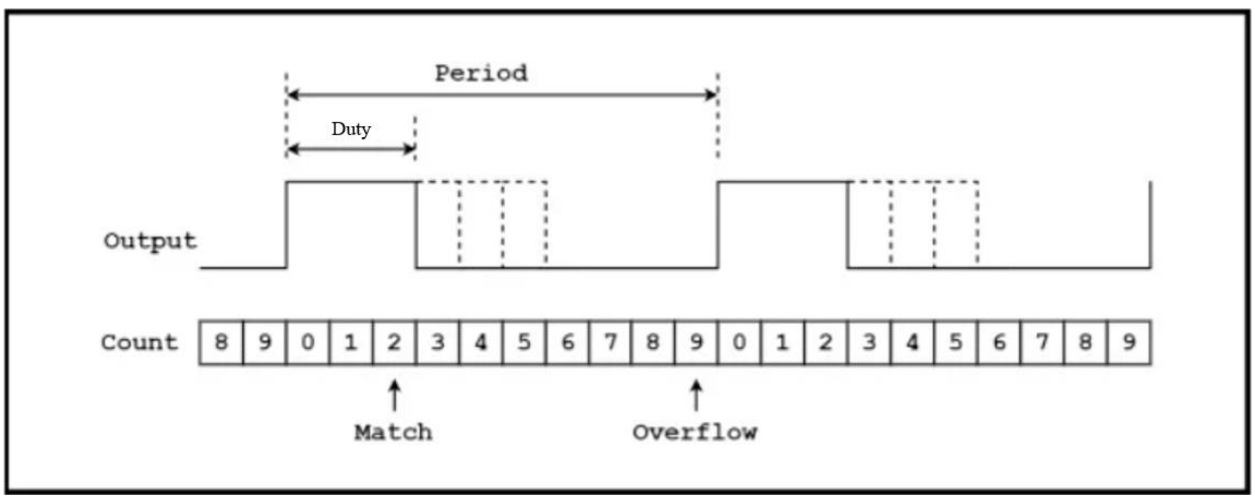

Figure 5 Schematic of the PWM principle

By using counters, in the AXI slave we define a module called "pwm_gen" that acts as a PWM generator. It has an 8-bit output connected to the H-bridge that controls the four motors. Each motor is controlled by a pair of bits, creating three different states: stop (both bits low), rotate forward (bit 0 high, bit 1 low), and rotate backward (bit 0 low, bit 1 high). The AXI-lite module features four registers in its memory map that store control commands, the total PWM period (based on the joystick input), and duty cycle values for both the left and right motors.

The Microblade processor on the control terminal receives input from the joystick via USB uart, then it sends the value to the car terminal via Bluetooth. These values range from 0 to 255 for both vertical and horizontal movements, which are used to calculate the duty cycle and control commands for each motor. For example, the absolute values of the joystick inputs are used to determine the forward, backward, left, and right movements of the vehicle. Based on Pythagorean theorem, the code calculates the duty cycle by taking the square root of the sum of the squared absolute values of the joystick inputs. Depending on the calculated duty cycle and the input values, the C code adjusts the vehicle's direction and speed.

The control commands are then written to the appropriate registers in the AXI slave, which in turn controls the motors based on the desired direction and speed. The registers are as follows:

1. Control register: used to start and stop the motors, and to dictate the direction of the motors (forward or backward).

2. Total period register: stores the total period of the PWM, which is 128 based on the joystick input.

3. Left motor duty cycle register: stores the duty cycle for the left motors, controlling their speed.

4. Right motor duty cycle register: stores the duty cycle for the right motors, controlling their speed.

4.2 Control Terminal

4.2.1 Control Terminal Introduction

The control terminal, which enables the human operator to manage the car and observe the vehicle's camera via the VGA display, is composed of a Nexys 4 DDR board, a VGA screen, a Bluetooth module, and an Xbox controller.

4.2.2 Acquiring Joystick Commands

A necessary component for obtaining input from an Xbox controller is a driver. By using the PyGame library in Python, we can access a user-friendly interface to gather the joystick and button information from the controller. Consequently, we developed a Python application to run on a laptop, which is solely responsible for capturing the controller input and transmitting it to the Nexys board control terminal. The control data is sent from the laptop to the control terminal through UART communication. An AXI Uartlite is employed to accept the control data. Like a car's Bluetooth system, the control data is also received using an interrupt mechanism.

4.2.3 Receiving Frame and VGA Display

Upon pressing the central button on the Nexys board, the control terminal transitions into the "frame receive" mode. To prevent conflicts between interrupt functions, the USB package reception is temporarily disabled. As Bluetooth receives pixels and stores them in the memory, the VGA module simultaneously retrieves these pixels from the memory and displays them on the screen.

5 Description of Design Tree

```
src
  |-- xbox_uart.py
  |-- car_terminal
  |   |-- ov7670.xpr
  |   |-- ov7670.srcs
  |   |   |-- sources_1/bd/design_1/ip
  |   |   |   |-- MicroBlaze(11.0)
  |   |   |   |-- Clocking Wizard(6.0)
  |   |   |   |-- gpio_input_splitter_v1_0(1.0)
  |   |   |   |-- AXI Uartlite(2.0)
  |   |   |   |-- AXI Interrupt Controller(4.1)
  |   |   |   |-- AXI BRAM Controller(4.1)
  |   |   |   |-- Block Memory Generator(8.4)
  |   |   |   |-- camera_capture_v1_0(1.0)
  |   |   |-- constrs_1/imports/XDC
  |   |   |   |-- Nexys4DDR_Master.xdc
  |   |-- ov7670.sdk
  |   |   |--  design_1_wrapper_hw_platform_0
  |   |   |--  test_bt_v2
  |   |   |--  test_bt_v2_bsp
  | -- control_terminal
  |   |-- ov7670_recv.xpr
  |   |-- ov7670_recv.srcs
  |   |   |-- sources_1/bd/design_1/ip
  |   |   |   |-- MicroBlaze(11.0)
  |   |   |   |-- Clocking Wizard(6.0)
  |   |   |   |-- AXI Uartlite(2.0)
  |   |   |   |-- AXI Interrupt Controller(4.1)
  |   |   |   |-- AXI BRAM Controller(4.1)
  |   |   |   |-- Block Memory Generator(8.4)
  |   |   |   |-- VGA
  |   |   |-- constrs_1/imports/XDC
  |   |   |   |-- Nexys4DDR_Master.xdc
  |   |-- ov7670.sdk
  |   |   |--  design_1_wrapper_hw_platform_0
  |   |   |--  bt_recv
  |   |   |--  bt_recv_bsp
  |-- Pulse-Width-Modulation-IP-master
  |   |-- project_1.xpr
  |   |-- project_1.srcs
  |   |   |-- sources_1/imports/HDL
  |   |   |   |-- pwm_gen.v
  |   |   |   |-- PWM_Generator_v1_0.v
  |   |   |   |-- PWM_Generator_v1_0_S_AXI.v
docs
  |-- 532_Final_Report_Group.pdf
```

6 Appendix

Github: Hanyu626/G9_RemoteControlRobot (github.com)

Video is also attached in the github README.md

7 References

[1] A. Alahi, M. Bierlaire, and P. Vandergheynst, "Robust real-time pedestrians detection in urban environments with low-resolution cameras," Transportation Research Part C: Emerging Technologies, vol. 60, pp. 15-29, 2016.

[2] E. Guizzo, "How iRobot's Roomba vacuum cleaner became part of the family," IEEE Spectrum, vol. 48, no. 3, pp. 56-63, 2011.

[3] Y. Matsumoto, E. Takeuchi, Y. Okada, and M. Hashimoto, "Development of an autonomous mobile robot for outdoor patrolling," in 2015 IEEE/SICE International Symposium on System Integration (SII), pp. 386-391, 2015.

[4] https://github.com/okchan08/OV7670