ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

FPGA-based object tracking, obstacle avoidance, and voice-activated trolley

---

Team #445

JIARUN HU
(jiarunh2@illinois.edu)
YANG ZHOU
(yangz15@illinois.edu)
YIHANG HE
(yihangh2@illinois.edu)
HAOMIN WANG
(haominw3@illinois.edu)

Sponsor: Said Mikki

March 8, 2022

Contents

1 Introduction

1.1 Problem

Nowadays, the development of electric vehicles today has become a trend. At the same time more and more new energy vehicle startups like to equip their cars with intelligent systems. However, existing SOCs are always based on non-real-time operating systems and cannot meet the real-time and safety of the in-vehicle system. Common systems which are based on CPU + GPU tend to have high energy consumption [1], which will have a negative impact on the endurance of the vehicle. Therefore, it is necessary to design a system with low energy consumption and high real-time performance.

1.2 Solution

In order to achieve low energy consumption and high real-time performance, our solution is to design a specific system to control our trolley based on FPGA. The latency of some FPGA-based designs can be deterministic. Besides, FPGAs are typically more energy efficient than other platforms except for ASICs, which usually have higher Non-Recurring Engineering (NRE) costs [1]. For neural network inference accelerators, FPGA implementations can be up to 10x more energy efficient than GPU ones [2].

1.3 Visual Aid

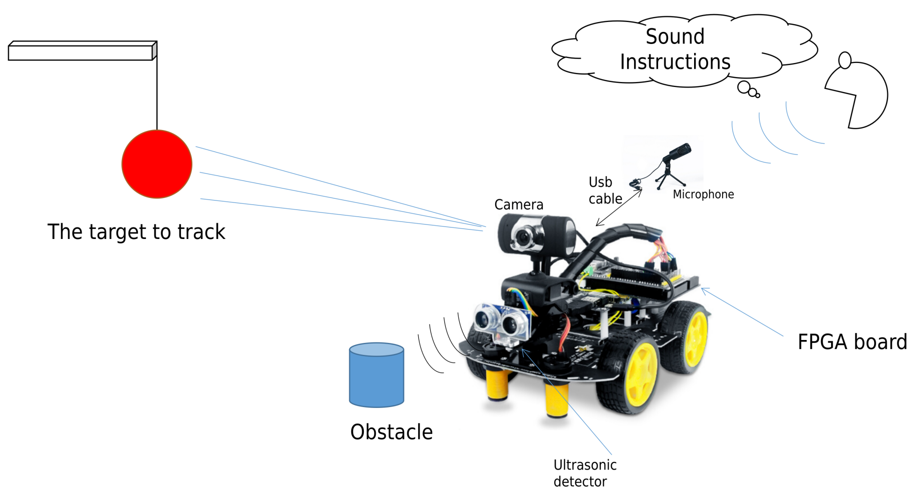

Figure 1: A pictorial representation of project

1.4 High-level Requirements List

1. Low Latency: The trolley must response to voice commands and detect the object in a short time.
2. Precision: The direction and speed of the trolley must be precise enough to follow the direction of the object and avoid obstacles without any collision.
3. Modularity: The 3 functions: Object Tracking, Obstacle Avoidance, Voice-Activated control the movement of the trolley, and they should not conflict with each other.
4. Power Consumption: implementing the same function, we system with FPGA will save more energy compared with using x86-CPU and GPU system.

2 Design

2.1 Block Diagram

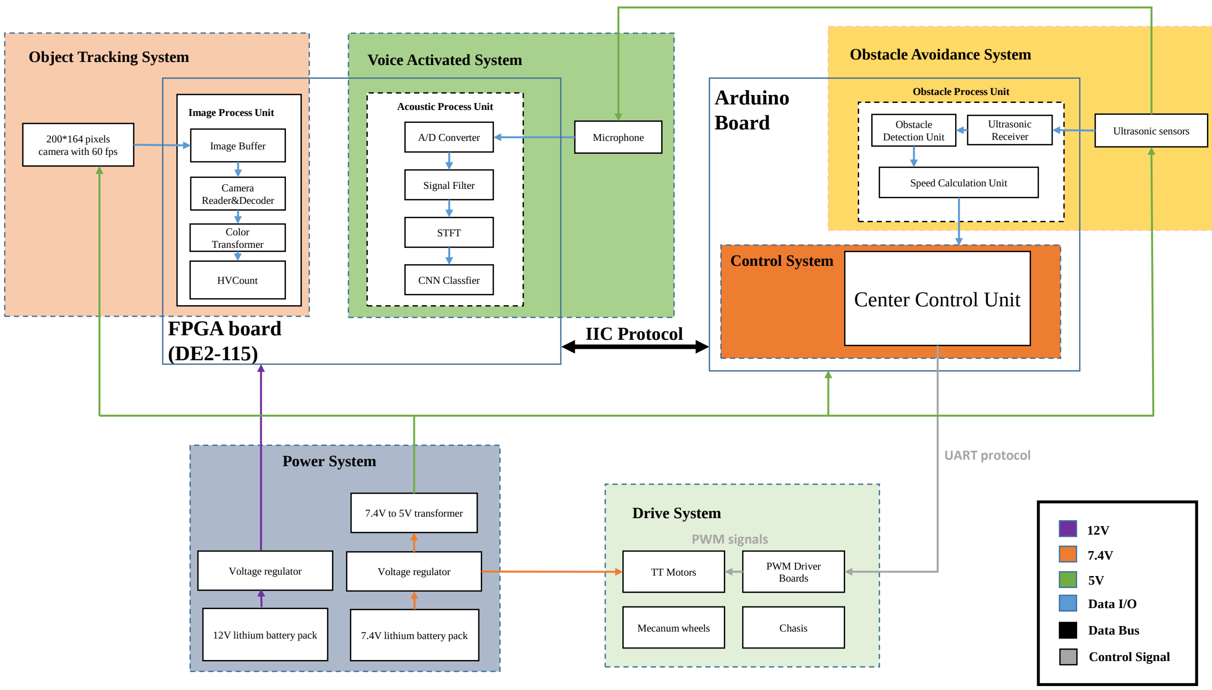

Figure 2: Block Diagram of System

2.2 System Overview

1. **The Control subsystem**: The Control subsystem is designed in Arduino. It communicates with FPGA with IIC protocol to receive signals from the object tracking subsystem and Voice Activated System. We design two modes for our control system: tracking mode and Voice manual control mode. The mode can be switched by voice instructions. In tracking mode, the control subsystem will receive two data from the object tracking subsystem, x_error, and h_error. x_error is the offset of the x-axis of the object's center point from the x-axis of the screen center point, and h_error is the difference between the size of the target object and the size we set. Then our control subsystem will calculate the corresponding speeds for each wheel for our trolley and send the signal to Drive Subsystem by UART protocol. In the Voice manual control mode, the Voice Activated System will send speed and direction signals to the control subsystem by IIC protocol, and the control subsystem will translate this data to two corresponding speeds for each wheel. And the Obstacle subsystem will connect with our control system with UART protocol, and whatever mode the control system is in, it will always deal with the signal from the obstacle system and try to bypass obstacles in the way.

2. **The Object tracking subsystem**: The Object tracking subsystem is designed to track

a Cubic with a single color. When we move such a cubic in front of our trolley, the Object Tracking system can calculate proper x_error and h_error to the control subsystem. The Object Tracking subsystem consists of a camera and process unit in FPGA. The processing unit of FPGA consists of a camera reader & decoder, color space transformer, and HVCount Module. The camera captures images with 200x164 pixels images with 60 fps and saves them into the BRAM of FPGA, the color space transformer read pictures from BRAM and firstly does RGB to YCBCR and finally uses a binary threshold module to turn it to a black and white image and use HVCount module to decide where the object is. Finally, the HVCount will determine the x_error and h_error and send them to our control subsystem.

3. **The Obstacle Avoidance subsystem**: The Obstacle avoidance subsystem is designed to make the trolley can bypass the obstacles in the way. It consists of ultrasonic sensors, and obstacle process unit in our Arduino Uno board. Sensors communicate with Arduino board with UART protocol. When there is an obstacle in front of our trolley, the obstacle avoidance subsystem will detect is and calculate speed_bias value for each wheel to control subsystem. And the control Subsystem can get the new speeds of wheels to change its directions.

4. **The Voice-Activated subsystem**: The Voice-Activated subsystem is designed to recognize natural language instructions with one or two words and send them to the control subsystem. It consists of a microphone and acoustic process unit in FPGA. The acoustic process unit contains an A/D converter, STFT(short-term Fourier transform) module, and a CNN classifier. All of them are designed in HDL (hardware design language) and implemented in our FPGA board. Our microphone first captures our voice and sends it to the A/D converter to become a digital signal. And the STFT module will treat the digital signal and turn it into a spectrum. Then the CNN classifier can classify it and recognize what kind of instructions the spectrum represents.

5. **The Drive subsystem**: The Drive subsystem consists of four Mecanum wheels, four TT motors, two PWM driver boards, and a chassis. The Mecanum wheels allow us to change the direction of our car by setting the separate speed of four wheels without a servo. The PWM driver board will receive speeds from the control subsystem by UART protocol and generate corresponding 4 PWM signals to control our 4 TT motors separately.

6. **The Power subsystem**: The Power subsystem consists of 7.4V and 12 V lithium battery packs, a 7.4V to 5V transformer, and voltage regulation circuits. The 12V battery pack will be connected to the regulation circuit and connected to the FPGA board. The 7.4 V battery packs will be connected to the 7.4V to 5V transformer and regulation circuit, and then connected to the drive subsystem and control subsystem.

2.3 Subsystems Requirements

2.3.1 Control Subsystem

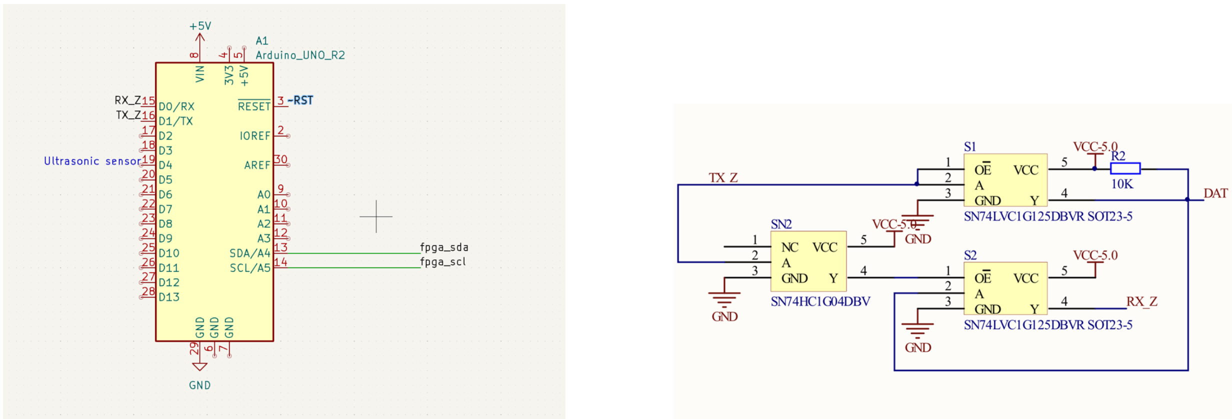

Figure 3: Circuit Schematic of Control System

The Control subsystem is responsible for processing real-time data from other subsystems and producing signals to control the drive subsystem Arduino will send signals by UART protocols with TX and RX ports, and then pass a PCB that combines these two ports to one DAT port to the Drive subsystem to control the PWM driver board. The Control subsystem connects with the Object Tracking subsystem, Obstacle Avoidance subsystem, and Voice-Activated subsystem by IIC protocol, and the Control subsystem should define a priority and logic to handle the control signals from these three trajectory planning subsystems to avoid conflict. The Control subsystem should react in a short time to ensure safety.

2.3.2 Object Tracking subsystem

The Object Tracking subsystem uses OV2640 Image Sensor to capture images in front of the trolley and send the image to the FPGA by SCCB protocol. So, firstly we design a camera init module in our FPGA board to initialize our image sensor in UXGA mode and set the output size of our sensor to 200x164. We also need a clock_pll to generate a 12MHz clock for our image sensor xclk port, and 4MHz clock for SCCB protocol to transfer data. After, initialization, we design a camera reader to read line and frame pixels and store them in the image cache (implemented by FPGA on-chip memory). Then the color space transformer will transfer the RGB image into YCbCr firstly, which can avoid the influence of the lightness, and then tranfer the image to a black and white image.[3] The HVCount module will count the black pixel to decide the size and the location of the tracking object, compare it with the set target, and determine x_error and h_error finally. And we will send these two data to our Control system in Arduino by IIC protocol. We will set our FPGA to be slave and Arduino to be master when we use IIC protocol. So we will design an IIC slave module to implement this protocol in our FPGA.

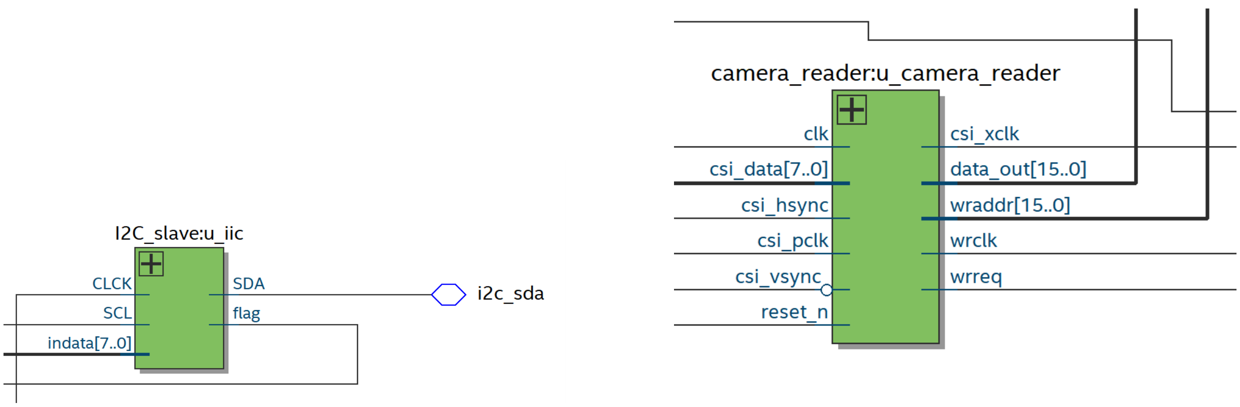

Figure 4: i2c slave module and camera reader

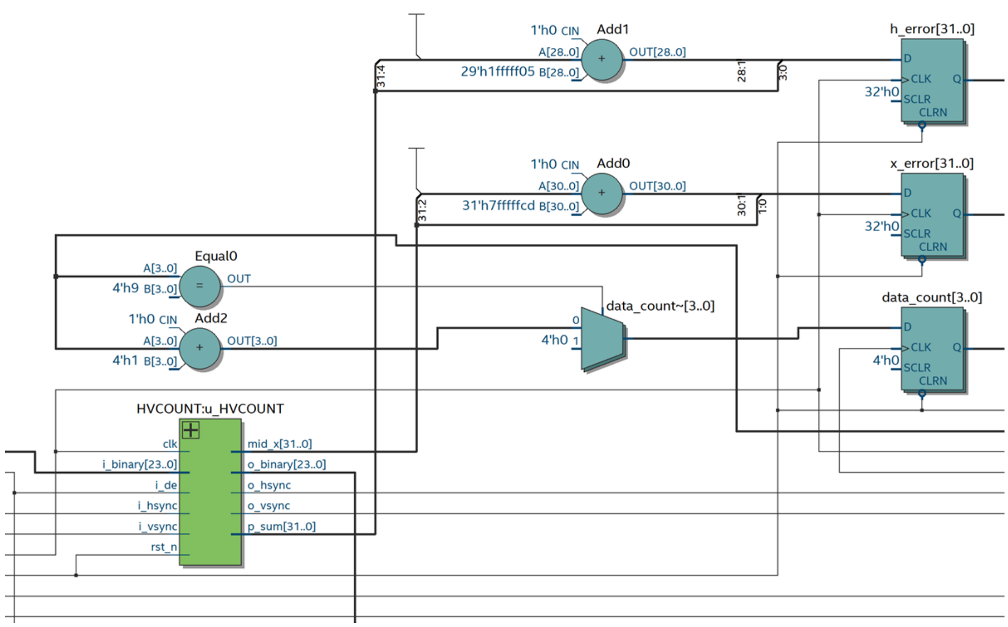

Figure 5: HVcount module and x_error h_error calculator

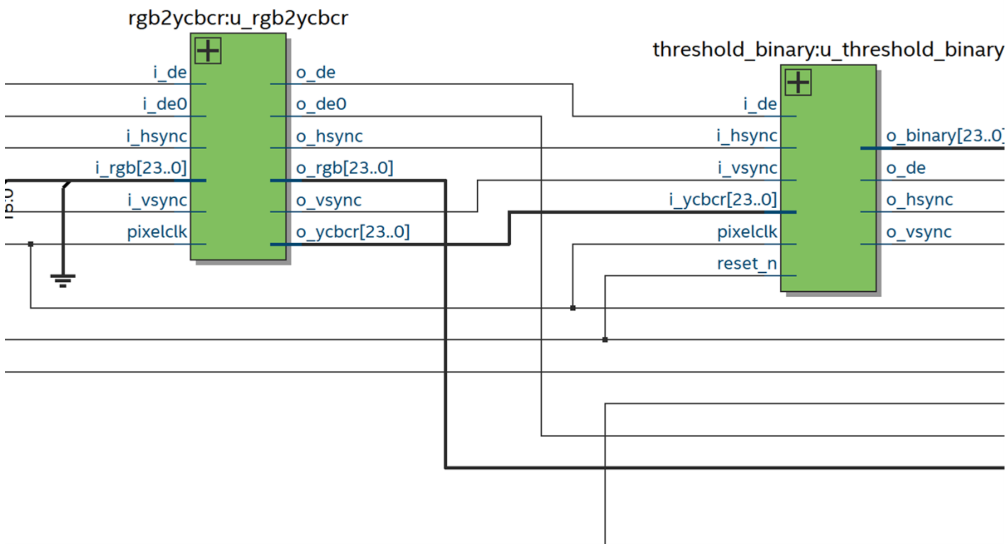

Figure 6: Color space Transformer

2.3.3 Obstacle Avoidance subsystem

The Obstacle Avoidance subsystem uses ultrasonic sensors and infrared sensors to detect obstacles in the path of the trolley. The sensors are connected to the obstacle process unit in FPGA with wires. If the obstacle process unit detects the signal of an obstacle, it will produce control signals to the speed of motors and the direction of the servo to avoid the obstacle, so the trolley will not collide with the obstacle. The control signals are passed on to the Control subsystem.

2.3.4 Voice-Activated subsystem

The Voice-Activated subsystem is responsible for recognizing specific natural language instructions and acting accordingly. The voice signal from the microphone will be processed by the A/D converter, signal filter, and STFT module to get spectrum and then passed to CNN[4]. The CNN module should have lower process latency than the previous part so we can deal with the vocal in real-time. And we need to use BRAM as the cache to store the untreated spectrum. We will train our CNN on our own PC first. The CNN should be quantized (which means we will use 8-bit integers instead of 64 bits double as our model parameter but maintain model accuracy as much as possible by retraining and fine-tuning our network.) to reduce the RAM and ROM use in our FPGA. The subsystem has at least 4 instructions: move forward, move backward, turn left, and turn right. The Voice-Activated subsystem connects with the Trolley Movement Control subsystem to produce the required signals of the speed of each motor. Then our trolley can change its speed and direction.

```
CNN(
  (conv1): Sequential(
    (0): Conv2d(1, 4, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (2): ReLU()
  )
  (conv2): Sequential(
    (0): Conv2d(4, 8, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (2): ReLU()
  )
  (conv3): Sequential(
    (0): Conv2d(8, 16, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (2): ReLU()
  )
  (fc1): Sequential(
    (0): DynamicQuantizedLinear(in_features=144, out_features=100, dtype=torch.qint8, qscheme=torch.per_tensor_affine)
  )
  (out): DynamicQuantizedLinear(in_features=100, out_features=10, dtype=torch.qint8, qscheme=torch.per_tensor_affine)
)
CNN(
  (conv1): Sequential(
    (0): Conv2d(1, 4, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (2): ReLU()
  )
  (conv2): Sequential(
    (0): Conv2d(4, 8, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (2): ReLU()
  )
  (conv3): Sequential(
    (0): Conv2d(8, 16, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (2): ReLU()
  )
  (fc1): Sequential(
    (0): Linear(in_features=144, out_features=100, bias=True)
  )
  (out): Linear(in_features=100, out_features=10, bias=True)
)
```

Figure 7: Quantized CNN classifier in Voice activated subsystem

2.3.5 Drive system

The Drive subsystem consists of four Mecanum wheels, four TT motors, two PWM driver boards, and a chassis. The PWM driver board should be able to respond to the UART signal from the control subsystem and generate a PWM signal to 4 TT motors within 1 ms. By controlling the rpm of each TT motor separately, and with Mecanum wheels, our drive subsystem can make our trolley move smoothly and accurately in different directions, such as forward, backward, left, and right. The chassis should be designed to support the weight of the entire system and allow for smooth movement of the trolley, with a minimum weight capacity of 2600g. The total drive system should be able to operate continuously without overheating or experiencing mechanical failure.

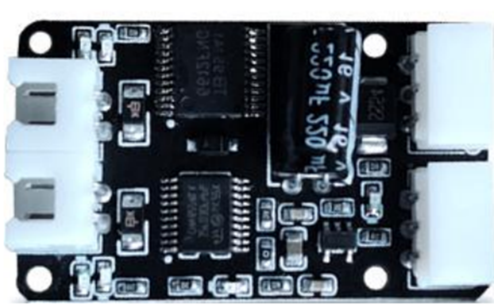

Figure 8: PWM driver board for TT motor

2.3.6 Power System

We have two Power subsystems, a 12V system, and a 5V system. The former one is for the Object tracking subsystem and Voice-activated subsystem in the FPGA board, and the second one is for the control subsystem, the drive subsystem, and the obstacle avoidance subsystem. The 12V system has a 12V lithium battery pack and the 5V system has 7.4V lithium battery pack and a 7.4V to 5V voltage transformer. Each power subsystem will have a voltage regulation board to ensure their outputs is stable.

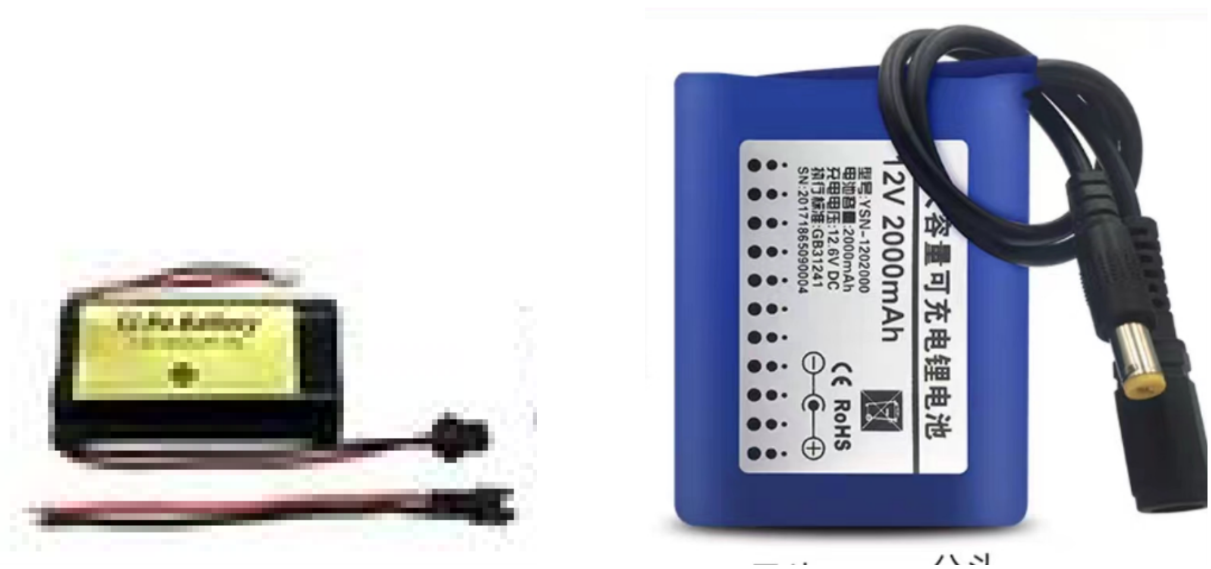

Figure 9: Lithium battery packs 12V and 7.4V

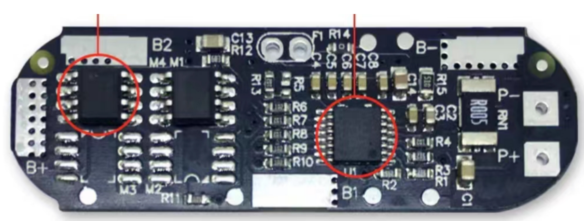

Figure 10: Voltage regulation board

3 Requirements and Verification

3.1 Requirements & Verification

3.1.1 Power subsystem [5 Points]

Requirements	Verification
7.4V Battery Pack: Supply 7.4 V±5% power.	7.4V Battery Pack: Use a voltage meter to check if the voltage output is equal to the specified value. Use an oscilloscope to check if the voltage output is stable.
12V Battery Pack: Supply 12 V±5% power.	12V Battery Pack: Use a voltage meter to check if the voltage output is equal to the specified value. Use an oscilloscope to check if the voltage output is stable.

3.1.2 Drive subsystem [5 Points]

Requirements	Verification
Movement control: The Arduino should send signals to motors to control the speed of motors. The speed of motors varies by the signal and the maximum value of signal is 2500.	Movement control: Connect Arduino to computer and monitor the port output. Use the controller to set the speed of motors. The output signal should be controlled by a controller and the maximum value should be 2500.

3.1.3 Object Tracking subsystem [10 Points]

Requirements	Verification
Camera: Camera Reader on FPGA can transfer 200x164 image with OV2640 in 30fps and save the data in the RAM. (In order to test this, we connect this part to VGA controller.)	Camera: Connect our FPGA to a screen with VGA and check the result.
Object Tracking system: The car will track a red cubic with a length of 4cm. We will set the target in the center of the visual of or trolley with a length of 2000 pixels. With the FPGA board, we compute the difference between our set target and the object to get x_error, which means the distance of the object center and target center, and h_error, which means the difference length between the detected object and our set target. Transfer this h_error and x_error with the IIC protocol.	Object Tracking system: Use modelsim HDL simulator to simulate whether the IIC protocol in FPGA can work properly. And implement it in FPGA and use an oscilloscope to test whether GPIO in the FPGA board works properly. And then connect it to our Arduino, when we move the object in front of the camera, we will monitor the serial port of the Arduino to test its received data.
Drive subsystem: The drive subsystem in Arduino will receive x_error and h_error by IIC protocol and calculate the corresponding velocity for each motor of 4 wheels. We will design a quadratic function to do this.	Drive subsystem: Before connecting to FPGA, set initialize values of x_error, and h_error to simulate the situation and test the rpm of 4 motors. And then connect Arduino to FPGA, test whether the car can follow our object and stop when the distance is smaller than 2cm and the object is in the center of the camera.

3.1.4 Obstacle Avoidance subsystem [10 Points]

Requirements	Verification
Detect Obstacles: The sensor should detect obstacles about 50cm in front of the trolley.	Detect Obstacles: Put an obstacle around 50cm in front of the trolley, if the trolley enters the Obstacle Avoidance mode, the buzzer will beep 4 times.
Avoid Obstacles: The Obstacle Avoidance subsystem should generate suitable output to motors to avoid obstacles without any collision.	Avoid Obstacles: Put an obstacle on the way of the trolley, test if the trolley starts turning at around 50cm and see if the trolley collides with the obstacles in this process.

3.1.5 Voice-Activated subsystem [10 Points]

Requirements	Verification
CNN: The CNN should identify 7 commands: Switch to voice mode Exit voice mode Stop Move forward Move backward Turn left Turn right	CNN: We first train CNN on computer, then import the CNN to FPGA, so we test these commands on CNN
Voice-Activated: Switch to voice mode, Exit voice mode: These commands should control the mode of trolley. The other commands: These commands should control the movement of the trolley.	Voice-Activated: Speak the first 2 commands, the buzzer should beep 5 times when enter the voice mode, and beep 6 times when exit the voice mode.Speak the other commands, the trolley should make corresponding movement

3.1.6 Control subsystem [10 Points]

Requirements	Verification
IIC protocol and UART protocol: Can receive value from FPGA by IIC protocol and UART protocol.	IIC protocol and UART protocol: Connect Arduino to PC to test values.
Mode switch: The control system should be able to switch between tracking mode and Voice manual control mode.	Mode switch: Send voice command, the control system can switch between tracking mode and Voice manual control mode.
Conflict handling: When receiving conflict signals from obstacle avoidance, object tracking, and voice-activated subsystem, control system needs to handle conflicts and send safe and appropriate signal to drive system.	Conflict handling: Put an obstacle between the object and trolley, trolley should avoid the obstacle first and then track the object.
Emergency Stop: To satisfy safety requirement, control system needs to send emergency stop signal to drive system under unknown or dangerous circumstance.	Emergency Stop: Physically stop the trolley by hand, the trolley should stop immediately and wait for restart instruction.

3.2 Tolerance Analysis

Firstly, the trolley should be able to move. Also, the speed of the trolley should not be too low or too high. After selecting the appropriate motor, the maximum speed of the car we have set now is 60cm/s, and the car can maintain this speed steadily.

One aspect of the design that may pose a risk to the successful completion of the project is the feasibility of the object tracking system. As mentioned above, the maximum speed is 60cm/s. The maximum angular velocity is 1.2 revolutions per second. Considering the application condition, it should be able to track the moving object.

Another risk is the accuracy of the voice identification system. Currently, this subsystem tested on the computer can recognize different numbers spoken by different people in a noisy environment with an accuracy of 86% percent. After more training and testing, we believe that it should be able to recognize different direction instructions.

A critical component of this project is the FPGA board. The FPGA board is the duty of the Object Tracking System, Obstacle Avoidance System, and Voice Activated System. In these systems, the Object Tracking System and Obstacle Avoidance System use a lot of memory in FPGA, which is at risk of memory overflow. To verify that our FPGA board can carry out these systems, we make simulations. We use Quartus to simulate object tracking algorithms and the following picture shows that this algorithm uses only 26% of memory in our device. In addition, we train our CNN to classify voice numbers 1 to

Flow Status	Successful - Tue Mar 14 14:44:38 2023
Quartus Prime Version	22.1std.0 Build 915 10...5/2022 SC Lite Edition
Revision Name	object_detect
Top-level Entity Name	object_detect
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	1,829 / 114,480 (2 %)
Total registers	699
Total pins	47 / 529 (9 %)
Total virtual pins	0
Total memory bits	1,048,576 / 3,981,312 (26 %)
Embedded Multiplier 9-bit elements	2 / 532 (< 1 %)
Total PLLs	1 / 4 (25 %)

Figure 11: FPGA Resources Utiliazation of the Object tracking subsystem

10 with Python on a computer. The estimated total size of CNN is 0.436455MB. Since the data type we use in Python is double and when we import the CNN to FPGA, the data type is char, the size of CNN in FPGA should be 457,656 bits, which is 11% of total memory bits. The simulation results show that these 2 systems use around 37% of total memory, so there will be large tolerance in our actual implementation.

4 Ethics and Safety

We will adhere to ethical principles outlined by IEEE and/or ACM, and appropriate safety features should be included in the design to ensure that the project is developed in a responsible and safe manner.

The project may involve the collection and storage of sensitive information, such as voice data or images. It is important to ensure that this information is kept confidential and that appropriate measures are taken to protect it from unauthorized access or disclosure.[5] we will carefully store and encrypt the data and delete it after the tests finish.

In terms of safety and regulatory standards, we will adhere to relevant regulations, industry standards, and campus policies. Referring to the ISO 13849-1:2015 Safety of machinery our trolley will be designed with safety in mind to prevent accidents or injuries.[6] We will ensure that the trolley and its components are designed and manufactured in a safe manner and that appropriate safety features are included, such as emergency stop and obstacle detection sensors. We will make sure that the trolley doesn't have any sharp edges or protrusions that could injure people.

We use Lithium-ion batteries in our trolley. Overcharging a Lithium-ion battery can cause it to overheat, which can result in a fire or explosion.[7] To avoid this, we won't leave the battery plugged in once it's fully charged. High temperatures and humidity can also damage Lithium-ion batteries and increase the risk of a fire or explosion. So we will store them in a cool, dry place, away from direct sunlight.

5 Cost and Schedule

5.1 Cost Analysis:

5.1.1 Labor:

- **Jiarun Hu**:

Task	Hours
Object Tracking system feasibility verification	10
Buy devices	5
Build car and FPGA environment	10
Design Voice-activated subsystem	20
Implement Voice-activated subsystem	20
Test Voice-activated subsystem	10
Test Voice-activated subsystem in the whole system	15
Ensure functionality and fix remaining issue	10
Prepare demo	20
Total	120

- **Yang Zhou**:

Task	Hours
Voice activated system feasibility verification	10
Do CNN quantization	10
Write design document	10
Design Power and drive subsystem	20
Implement Power and drive subsystem	20
Assemble all subsystems	10
Write unit test	5
Ensure functionality and fix remaining issue	10
Prepare demo	20
Total	115

- **Haomin Wang:**:

Task	Hours
Write proposal	5
Explore mean shift algorithm	10
Write design document	10
Design Object tracking subsystem	20
Implement Object tracking subsystem	20
Test Object tracking subsystem	10
Test Object tracking subsystem in the whole system	10
Ensure functionality and fix remaining issue	10
Prepare demo	20
Total	115

- **Yihang He:**:

Task	Hours
Write proposal	5
Explore opticall flow algorithm	10
Test speed, weight, acceleration of car	10
Design Obstacle Avoidance subsystem	20
Implement Obstacle Avoidance subsystem	20
Test Obstacle Avoidance subsystem	10
Test Obstacle Avoidance subsystem in the whole system	10
Ensure functionality and fix remaining issue	10
Prepare demo	20
Total	115

5.1.2 Parts:

Components	Price
Arduino stm32 car	558
12v battery	45.37
Mini-microphone	138
Ov2640 Camera	133
Total	874.37

5.1.3 Sum of Total:

Salary(1/Hours*2.5*465=1162.5)+Components(874.37)=2023.87

5.2 Schedule

Week	Task	Responsibility
3.6-3.12	Object Tracking system feasibility verification	Jiarun Hu
	Voice activated system feasibility verification	Yang Zhou
	Write proposal	Yihang He
	Write proposal	Haomin Wang
3.13-3.19	Buy devices	Jiarun Hu
	Do CNN quantization	Yang Zhou
	Explore opticall flow algorithm	Yihang He
	Explore mean shift algorithm	Haomin Wang
3.20-3.26	Build car and FPGA environment	Jiarun Hu
	Write design document	Yang Zhou
	Test speed, weight, acceleration of car	Yihang He
	Write design document	Haomin Wang
3.27-4.2	Design Voice-activated subsystem	Jiarun Hu
	Design Power and drive subsystem	Yang Zhou
	Design Obstacle Avoidance subsystem	Yihang He
	Design Object tracking subsystem	Haomin Wang
4.3-4.9	Implement Voice-activated subsystem	Jiarun Hu
	Implement Power and drive subsystem	Yang Zhou
	Implement Obstacle Avoidance subsystem	Yihang He
	Implement Object tracking subsystem	Haomin Wang

4.10-4.16	Test Voice-activated subsystem	Jiarun Hu
	Assemble all subsystems	Yang Zhou
	Test Obstacle Avoidance subsystem	Yihang He
	Test Object tracking subsystem	Haomin Wang
4.17-4.23	Test Voice-activated subsystem in the whole system	Jiarun Hu
	Write unit test	Yang Zhou
	Test Obstacle Avoidance subsystem in the whole system	Yihang He
	Test Object tracking subsystem in the whole system	Haomin Wang
4.24-4.30	Ensure functionality and fix remaining issue	Jiarun Hu
	Ensure functionality and fix remaining issue	Yang Zhou
	Ensure functionality and fix remaining issue	Yihang He
	Ensure functionality and fix remaining issue	Haomin Wang
5.1-5.7	Prepare demo	Jiarun Hu
	Prepare demo	Yang Zhou
	Prepare demo	Yihang He
	Prepare demo	Haomin Wang

References

[1] Y.-H. Chen, T.-J. Yang, J. Emer, and V. Sze, "Understanding the Limitations of Existing Energy-Efficient Design Approaches for Deep Neural Networks," en,

[2] D. Castells-Rufas, V. Ngo, J. Borrego-Carazo, M. Codina, C. Sanchez, D. Gil, and J. Carrabina, "A Survey of FPGA-Based Vision Systems for Autonomous Cars," *IEEE Access*, vol. 10, pp. 132 525–132 563, 2022, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3230282.

[3] X. Lu, D. Ren, and S. Yu, "FPGA-based real-time object tracking for mobile robot," in *2010 International Conference on Audio, Language and Image Processing*, Nov. 2010, pp. 1657–1662. DOI: 10.1109/ICALIP.2010.5685091.

[4] H. Hu, J. Li, C. Wu, X. Li, and Y. Chen, "Design and Implementation of Intelligent Speech Recognition System Based on FPGA," en, *Journal of Physics: Conference Series*, vol. 2171, no. 1, p. 012 010, Jan. 2022, Publisher: IOP Publishing, ISSN: 1742-6596. DOI: 10.1088/1742-6596/2171/1/012010. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/2171/1/012010 (visited on 03/08/2023).

[5] 14:00-17:00, *ISO/IEC 27001:2013*, en, Dec. 2020. [Online]. Available: https://www.iso.org/standard/54534.html (visited on 03/24/2023).

[6] *ISO - ISO 13849-1:2015 - Safety of machinery — Safety-related parts of control systems — Part 1: General principles for design*. [Online]. Available: https://www.iso.org/standard/69883.html (visited on 03/24/2023).

[7] *Safe Practice for Lead Acid and Lithium Batteries*. [Online]. Available: https://courses.engr.illinois.edu/ece445zjui/documents/GeneralBatterySafety.pdf.