MINISTRY OF EDUCATION AND TRAINING

HUNG YEN UNIVERSITY OF TECHNOLOGY AND EDUCATION

DANG XUAN QUANG

STUDYING AND DESIGNING A SMART PARKING SYSTEM

GRADUATE PROJECT

HUNG YEN - 2024

DANG XUAN QUANG STUDYING AND DESIGNING A SMART PARKING SYSTEM 2024

MINISTRY OF EDUCATION AND TRAINING

HUNG YEN UNIVERSITY OF TECHNOLOGY AND EDUCATION

DANG XUAN QUANG

STUDYING AND DESIGNING
A SMART PARKING SYSTEM

MAJOR: INFORMATION TECHNOLOGY

SPECIALIZATION: IoT APPLICATION DEVELOPMENT

GRADUATE PROJECT

INSTRUCTOR	CHAIRMAN OF THE COUNCIL
(Signature and full name)	*(Signature and full name)*

HUNG YEN - 2024

COMMENTS

Instructor's comments:

....

....

....

....

....

....

....

....

....

INSTRUCTOR'S SIGNATURE

(Signature and full name)

Reviewer 1's Comments:

....

....

....

....

....

....

REVIEWER 1'S SIGNATURE

(Signature and full name)

Reviewer 2's Comments:

....

....

....

....

....

....

REVIEWER 2'S SIGNATURE

(Signature and full name)

GUARANTEE

My name is Dang Xuan Quang. I hereby declare that the graduation thesis "Studying and designing a smart parking system" is my own research work. The parts that use references have been clearly cited in the reference section. The data and results presented in the thesis are completely truthful. If any inaccuracies are found, I will take full responsibility and accept any disciplinary actions imposed by the department and the university.

Hung Yen, day... month... year...

Student

ACKNOWLEDGEMENTS

During the process of completing the major assignment, I always received the attention, guidance and enthusiastic help from the teachers. First of all, I would like to thank the Department of Information Systems, The Faculty of Information Technology - Hung Yen University of Technology and Education has created favorable conditions for me to carry out this project.

I would also like to sincerely thank all the teachers and students at the school for their dedication to teaching and equipping me with necessary and valuable knowledge to help me complete this great assignment.

In the process of doing a big assignment, it is difficult to avoid mistakes, I really hope the teachers will ignore them. At the same time, because my theoretical level as well as practical experience is still limited, large assignments cannot avoid shortcomings. I look forward to receiving your comments and suggestions so that I can learn more and have more experience. Complete the upcoming big assignments better.

TABLE OF CONTENTS

TABLE OF CONTENTS 7

LIST OF FIGURES 10

LIST OF TABLES 12

LIST OF ABBREVIATIONS 13

CHAPTER 1: INTRODUCTION 15

1.1 Motivation 15

1.2 Project objectives 16

1.2.1 General objectives 16

1.2.2 Detail objectives 16

1.3 Subjects and scope of the project 16

1.3.1 Research subjects 16

1.3.2 Scope of research 17

1.4 Implementation process 17

1.4.1 Research and Select Appropriate Technologies 17

1.4.2 Design the Hardware System 17

1.4.3 Develop Parking Management Software 18

1.4.4 Program and Integrate Hardware and Software Components 18

1.4.5 Test and Evaluate the System 18

1.5 Research methodology 19

1.5.1 Approach 19

1.5.2 Research Methods 19

1.6 Conclusion 19
CHAPTER 2: THEORETICAL BASIC OF EMBEDDED SYSTEM 21
2.1 Overview of system components 21
2.1.1 Arduino Uno 21
2.1.2 Arduino Nano 23
2.1.3 Webcam 27
2.1.4 Module RFID RC522 29
2.1.5 Servo Motor 31
2.1.6 Infrared Sensor 32
2.2 Related theories and technologies...33
2.2.1 Image processing concept...33
2.2.2 Vehicle licemse plate recognition concept with EmguCV.........33
2.2.3 RFID Technology 35
2.2.4 Communication Technologies in the System 36
2.3 Conclusion 45
CHAPTER 3: IMPLEMENTATION..47
3.1 System requirements analysis 47
3.1.1 System requirement 47
3.1.2 Solutions 47
3.2 System design 48
3.2.1 RFID Technology 48
3.2.2 Software design for the system..50
3.2.3 Parking lot algorithm flowchat 53
3.2.4 RFID tag processing algorithm flowchat 54

3.3 System construction and integration 56

3.4 System testing and evaluation 58

3.4.1 Testing methods 58

3.4.2 System evaluation 59

3.5 System operation guide 59

CONCLUSION 61

1. Achieved results of the project 61

2. Limitations of the project 61

3. Future development directions of the project 61

REFERENCES 63

LIST OF FIGURES

Figure 2.1: Arduino Uno [9] 21

Figure 2.2: Arduino Nano [9] 24

Figure 2.3: Arduino Nano pinout diagram [9] 26

Figure 2.4: Webcam [10] 28

Figure 2.5: Module RFID RC522 [9] 29

Figure 2.6: Mifare magnetic card [9] 30

Figure 2.7: SG90 Servo motor [9] 31

Figure 2.8: Infrared Sensor [9] 32

Figure 2.9: Image processing concept [11] 33

Figure 2.10: RFID system structure [12] 35

Figure 2.11: SPI Communication [13] 38

Figure 2.12: I2C Communication [14] 39

Figure 2.13: I2C Transmission frame [14] 40

Figure 2.14: I2C Data transmission [14] 41

Figure 2.15: UART Data Transmission [15] 43

Figure 2.16: USB Control Transfer [16] 44

Figure 3.1: Hardware connection diagram 48

Figure 3.2: Circuit diagram of hardware connection 49

Figure 3.3: Visual studio software [17] 50

Figure 3.4: Arduino IDE software. [18] 51

Figure 3.5: Arduino IDE code writing interface. 51

Figure 3.6: Integrated Libraries in Arduino IDE. 52

Figure 3.7: Parking Lot Algorithm Flowchart 53

Figure 3.8: RFID tag processing algorithm flowchat 55

Figure 3.9: Hardware connection..56

Figure 3.10: Product model...57

Figure 3.11: Graphic of system..58

LIST OF TABLES

Table 2.1: Technical specifications of the Arduini Uno 22

Table 2.2: Technical specifications of the Arduini Nano 26

Table 2.3: Technical specifications of the Webcam 28

Table 2.4: Technical specifications of the Module RFID RC522............................29

Table 2.5: Technical specifications of the Servo SG90 31

LIST OF ABBREVIATIONS

Abbreviation	Full term	Explanation
RFID	Radio Frequency Identification	Uses radio waves to identify and track objects.
ANPR	Automatic Number Plate Recognition	Uses optical character recognition on images to read vehicle registration plates.
ALPR	Automatic License Plate Recognition	Uses cameras and software to read license plates automatically.
MQTT	Message Queuing Telemetry Transport	A widely used messaging protocol for IoT devices and M2M applications.
HTTP	Hypertext Transfer Protocol	The protocol for transferring data between a web server and a web browser, commonly used for accessing web pages.
USB	Universal Serial Bus	A common interface used for connecting devices to computers, providing power and data transfer capabilities
LCD	Liquid Crystal Display	A flat-panel display technology commonly used in electronic devices such as TVs, monitors, and smartphones.

Abbreviation	Full term	Explanation
GPS	Global Positioning System	A satellite-based navigation system that provides location and time information anywhere on or near the Earth's surface.
UART	Universal Asynchronous Receiver-Transmitter	A hardware communication protocol used for serial communication between devices
SPI	Serial Peripheral Interface	A synchronous serial communication protocol used to transfer data between microcontrollers and peripheral devices
IDE	Integrated Development Environment	A software application that provides comprehensive facilities to computer programmers for software development
ADC	Analog-to-Digital Converter	A device or circuit that converts a continuous analog signal to a digital representation.
OCR	Optical Character Recognition	Recognize and convert text characters from images

CHAPTER 1: INTRODUCTION

1.1 Motivation

According to the report recently released by the General Statistics Office, in May 2022, the estimated number of domestically manufactured and assembled automobiles reached 44,500 units (an increase of 13.2% compared to the previous month of April 2022 - 39,300 units) and an increase of 17.6% compared to the same period in 2021. The increasing number of vehicles has led to a severe lack of parking spaces for residents and employees at shopping malls or office buildings. If parking lots are not designed and constructed, vehicles will be parked in a disorderly manner, causing a loss of urban aesthetics, obstructing traffic, and even compromising vehicle safety. Furthermore, if parking lots, ranging from conventional to smart ones, are not promptly designed and arranged to accommodate the rising flow of vehicles, everything will become more inconvenient. This is especially challenging for maintaining public order and management.

The priority task is to construct parking lots as soon as possible, but most current parking lots are either spontaneous or simply conventional lots managed by an individual... Such parking lots often pose a high risk of theft.

Therefore, parking lots today need to be modernized and developed to address the shortcomings of traditional parking lots. One method to address these challenges is to apply scientific, technological, and technological advancements to infrastructure construction. Specifically, it is necessary to build modern, highly automated parking lots with a top prbiority on safety. At the same time, such parking lots need to be economically viable and align with practical considerations such as land use optimization, urban landscape, etc.

Based on the above requirements and issues, I propose to choose my graduation project as: "Studying and designing a smart parking system" which utilizes RFID technology combined with cameras to capture and recognize vehicle license plates, thereby enhancing security and safety.

1.2 Project objectives

1.2.1 General objectives

To develop a modern technological solution that meets the demand for safe, convenient, and efficient parking. The system will optimize the utilization of parking spaces, enhance security, and mitigate the risk of theft. Simultaneously, this solution aims to elevate the user experience and facilitate easier management for all stakeholders involved..

1.2.2 Detail objectives

Developing a smart parking lot system model, integrating RFID technology and cameras to automatically identify vehicle license plates and monitor parking status.

Designing a centralized management system that provides statistical data and detailed reports on parking lot usage to support effective decision-making.

Ensuring the security and safety of the system through access control solutions, data encryption, and camera monitoring.

Optimizing the utilization of parking spaces, minimizing the time spent searching for available spots, and improving traffic flow within the parking area.

1.3 Subjects and scope of the project

1.3.1 Research subjects

RFID tags, RFID reader modules, motors, arduino, infrared sensor.

C# programming language.

Communication interfaces utilized.

1.3.2 Scope of research

Application of RFID technology.

Vehicle license plate recognition system accurately identifies license plates of entering and exiting vehicles.

Control of barrier gate opening and closing upon correct vehicle license plate verification.

1.4 Implementation process

1.4.1 Research and Select Appropriate Technologies

Identify Requirements: Define the system requirements, including accuracy, speed, and environmental conditions.

Evaluate Technologies: Research various technologies for vehicle license plate recognition (e.g., ANPR/ALPR systems) and parking status monitoring.

Select Vendors: Choose vendors or products for cameras, sensors, RFID modules, and software platforms based on reliability, cost, and compatibility.

Conduct Feasibility Studies: Perform feasibility studies to ensure the selected technologies meet the project requirements.

1.4.2 Design the Hardware System

Sensors: Choose and position sensors (e.g., ultrasonic, infrared) to detect vehicle presence in parking spots.

Cameras: Select high-resolution cameras for license plate recognition and position them at strategic points (entry/exit gates).

RFID Modules: Design the integration of RFID modules for vehicle identification and access control.

Entry/Exit Gate Mechanisms: Plan the mechanism for gates, including motorized barriers and control systems.

Power and Connectivity: Ensure adequate power supply and connectivity (wired/wireless) for all hardware components.

1.4.3 Develop Parking Management Software

User Interface Design: Create user-friendly interfaces for both desktop and mobile applications.

Integration with Payment Systems: Integrate with various payment gateways to support multiple payment methods.

Notification System: Implement a system for sending alerts and notifications (SMS, email, push notifications).

1.4.4 Program and Integrate Hardware and Software Components

Firmware Development: Write firmware for sensors, RFID modules, and other hardware components to communicate with the central system.

Software Development: Develop the parking management software, including backend logic, APIs, and front-end interfaces.

System Integration: Integrate hardware components (cameras, sensors, RFID) with the software system.

Communication Protocols: Ensure all components communicate effectively using appropriate protocols (e.g., MQTT, HTTP, WebSockets).

1.4.5 Test and Evaluate the System

Unit Testing: Test individual components (sensors, cameras, RFID modules) to ensure they function correctly.

Integration Testing: Test the integration between hardware and software to ensure seamless operation.

User Acceptance Testing: Conduct tests with end-users to gather feedback and identify areas for improvement.

Performance Testing: Evaluate the system’s performance under various conditions (e.g., peak usage times, different weather conditions).

Scalability Testing: Test the system's ability to scale up (add more parking spaces or integrate additional sensors).

1.5 Research methodology

1.5.1 Approach

Research various technologies and platforms related to the design of a smart parking system using RFID cards. This includes exploring technologies such as license plate recognition, sensors, RFID modules, and software platforms for parking management.

1.5.2 Research Methods

Literature Review: Search and read academic papers, articles, technical reports, and books related to RFID technology, license plate recognition systems, and smart parking management systems. Gather information on technologies, methods, and practical applications.

Sample Analysis: Study existing smart parking systems and analyze these sample systems to understand their operation, components, and performance. Compare and evaluate different solutions to determine the strengths and weaknesses of each technology and method.

Experimental Method: Conduct experiments to test and evaluate various technologies and solutions. Set up small-scale experimental models to test the operation of RFID modules, sensors, and license plate recognition systems. Collect data from experiments to analyze and improve the system.

1.6 Conclusion

According to the report from the General Statistics Office, in May 2022, the number of domestically manufactured and assembled automobiles increased significantly, causing a severe shortage of parking spaces, detracting from urban aesthetics, and obstructing traffic. To address this issue, it is necessary to build modern, highly automated parking lots. My graduation project, "Research and Design of a Smart Parking System," utilizes RFID technology and license plate recognition cameras to enhance security and safety. The objective is to develop a safe, convenient, and efficient parking solution by integrating recognition and monitoring technologies, designing a centralized management system, ensuring security, and optimizing parking space utilization. The research methodology includes studying technologies, analyzing existing systems, and conducting experiments to evaluate solutions.

CHAPTER 2: THEORETICAL BASIC OF EMBEDDED SYSTEM

2.1 Overview of system components

2.1.1 Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328P chip, popular and easy to use for electronics and programming projects. Equipped with 14 digital input/output pins, 6 analog input pins, and a USB connection, the Arduino Uno allows users to easily connect and control peripheral devices. With strong community support and a rich library of open-source code, the Arduino Uno is an ideal choice for both beginners and experienced developers. It is suitable for a wide range of applications, from simple control tasks to complex automation systems.

Figure 2.1**:** *Arduino Uno [9]*

The Arduino Uno is a part of the Arduino family, an open-source hardware platform. It stands out for its versatile connectivity and high compatibility with various peripheral devices. The board operates at 5V and can be powered via USB or an external power source ranging from 7-12V.

Table 2.1: Technical specifications of the Arduini Uno

Specifications	Detail
Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (6 of which can be used as PWM outputs)
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB used by bootloader)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

a) Key features

- USB Connection

  The Arduino Uno features a USB port that facilitates easy connection to a computer for uploading programs and powering the board. This USB port also supports serial (UART) communication with the computer, allowing you to send and receive data.

- Flexible I/O Pins

  The Arduino Uno has 14 digital input/output pins, with 6 pins capable of PWM output to control the brightness of LEDs or the speed of motors.

Additionally, 6 analog input pins allow for measuring analog signals, making it highly useful in sensor applications.

- Serial Communication

  The Arduino Uno supports various communication protocols such as UART, SPI, and I2C. This enables the board to connect with a wide range of devices and modules, from simple sensors to complex systems like LCD displays, GPS modules, and Bluetooth modules.

b) Support and resources

- Open-Source Libraries

  One of the significant advantages of the Arduino Uno is its support from open-source libraries. Arduino provides numerous software libraries to facilitate communication with sensors and peripherals. These libraries can be easily integrated into your project through the Arduino IDE.

- Strong Community

  The Arduino Uno is supported by a vast community of users. You can easily find help, share experiences, and access countless tutorials and sample projects from the global Arduino community. This is particularly beneficial for beginners, helping them quickly learn and progress.

- Arduino IDE

  The Arduino IDE (Integrated Development Environment) is a free and user-friendly development environment designed for programming Arduino boards. The IDE supports multiple programming languages and provides tools for compiling code, uploading programs to the board, and monitoring program activities via the Serial Monitor.

2.1.2 Arduino Nano

The Arduino Nano is a compact electronic board with dimensions only half the size of a folded quarter coin. It was developed based on the Atmega328P microcontroller and released in 2008. The Nano is quite breadboard-friendly. The

Arduino Nano provides connections and technical specifications similar to the Arduino Uno electronic board, but in a much smaller form factor.

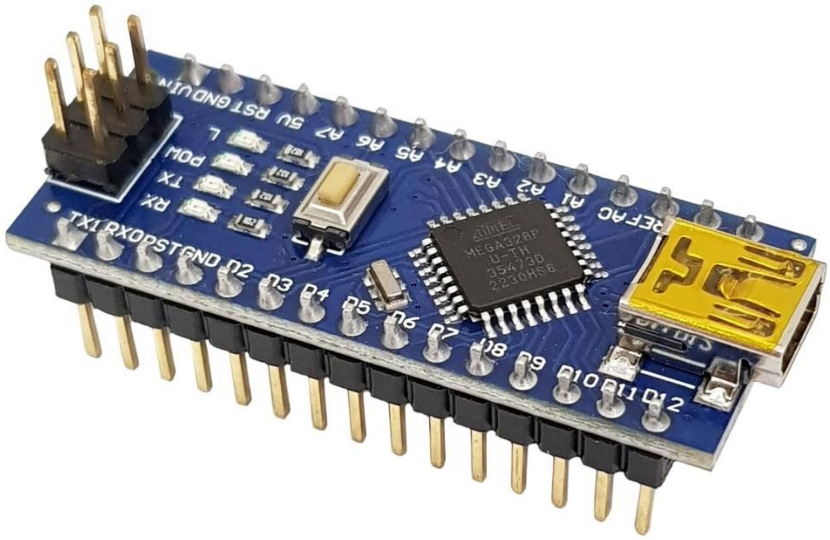

Figure 2.2: *Arduino Nano [9]*

The Arduino Nano has similar functionality to the Arduino UNO. The typical difference between them is in the board form factor. The Arduino Nano pinout integrates the Atmega328P microcontroller, same as the Arduino UNO, but the UNO board has a Plastic Dual-In-line Package (PDIP) form with a total of 30 pins, while the Nano board comes in a Plastic Quad Flat Pack with 32 pins. Another difference is that the Nano board has 8 ADC ports, while the UNO has 6 ADC ports. Additionally, the Nano board does not have a DC power jack like other typical Arduino boards; instead, it is equipped with a mini-USB port that allows both programming and serial monitoring.

The Arduino Nano has the advantage of selecting the maximum power with its voltage, and it can be conveniently and simply programmed directly from a computer. Particularly, the Arduino Nano pinout has a compact size of only 18.5mm x 43mm with a weight of around 7g. Due to this, the Arduino Nano has extremely diverse applications in modern life today.

The Arduino Nano has the ability to be programmed using the AVR Atmega328 chip from Atmel with the following main features and functions:

Arduino Software: Called sketches, created on a computer with an integrated development environment (IDE). The IDE allows writing, editing code, and converting it into a format understandable by the hardware. The IDE is used to compile and upload to the Arduino (this process is called UPLOAD).

Arduino Hardware: The Arduino boards where the programmed code is executed. These boards can control or respond to electrical signals, so components are directly connected to them to interact with the real world for sensing or communication. Examples of sensors include switch devices, ultrasonic sensors, accelerometers. Output devices include lights, motors, speakers, and displays.

Most Arduino boards use a USB connection for both power supply and data upload to the Arduino board.

a) Hardware structure and technical specifications of Arduino Nano

The Arduino can be powered via a USB connection or with an external power supply. If supplied with less than 5V, the microcontroller system may become unstable, and if using more than 12V, the voltage regulator may overheat. Therefore, the recommended range is 7V to 12V.

b) Arduino Nano pins and some functions:

VIN Pin: Input voltage to the Arduino when using an external power source (other than the 5V from USB or the barrel jack). We can supply power through this pin.

5V Pin: Supplies power to the microcontroller and other components on the board, and provides power to external devices when connected to the board.

3.3V Pin: Supplies power to sensors.

GND Pin: Ground pin.

Rx, Tx Pins: Used for serial data reception and transmission.

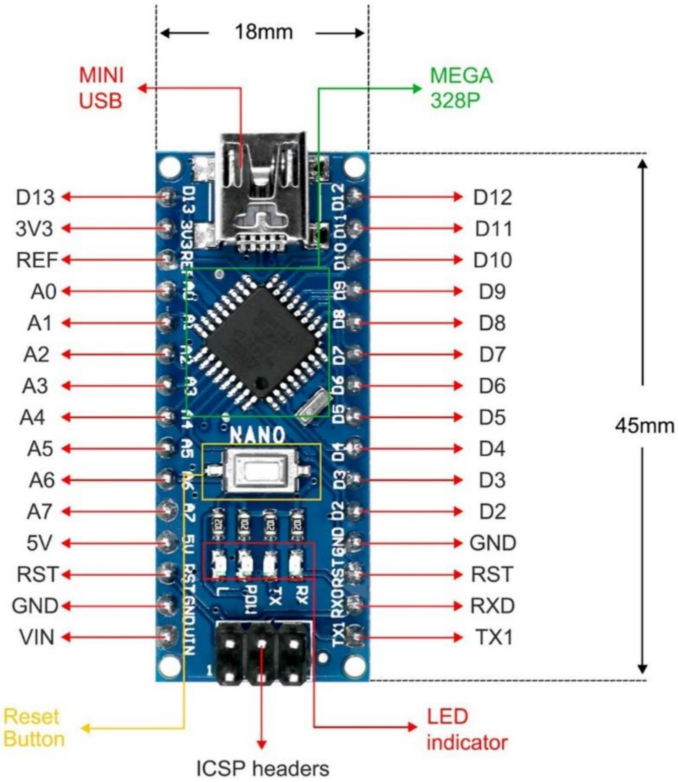

Figure 2.3: *Arduino Nano pinout diagram [9]*

D2 and D3 Pins: Used for interrupts.

D3, D5, D6, D9, D10 and D11 Pins: PWM capable pins.

SPI Interface: Uses pins D10 (SS), D11 (MOSI), D12 (MISO), D13 (SCK).

I2C Interface: Uses analog input pins A4 (SDA) and A5 (SCL).

Aref Pin: Analog reference voltage pin.

A0, A1, A2, A3, A6, A7 Pins: Analog input/output pins.

Table 2.2: Technical specifications of the Arduini Nano

Specifications	Detail
Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V

Specifications	Detail
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (6 of which can be used as PWM outputs)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (0.5 KB used by bootloader)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

c) Applications of Arduino

In practice, Arduino has many applications due to its flexible processing capabilities and the ease of integrating its hardware into other systems. Therefore, it can be applied in most automation control systems, from simple ones like fire and gas alarms, measuring environmental parameters (gas concentration, temperature, humidity, light), or more complex applications like 3D printer control. Additionally, Arduino is used in entertainment technologies such as designing line-following robots or game controllers. Arduino can also be combined and connected with other electronic devices, such as connecting with an embedded Raspberry Pi computer to collect and send data to the internet, or interfacing with expansion boards like WiFi or Ethernet Shields…

2.1.3 Webcam

In the smart parking system, this can be considered the most important part, and in this project, I am using the Rapoo C200 camera connected to the computer via a USB port to take pictures and recognize the license plate of vehicles.

Figure 2.4: *Webcam [10]*

Table 2.3: Technical specifications of the Webcam

Specifications	Detail
Product name	Webcam
video encoding	H.264
Resolution	HD 720P
Frame rate	30fps
Compatible software	Skype, Zoom, Google Hangouts, Face Time for Mac, Teams, Zalo, Facebook messenger,…
Image format	JPG 1280*720
Connectivity	USB 3.0 & 2.0
Compatible operating system	Windows7/8/10 or above, Mac OS X 10.6 or above,Chrome OS, and Android v5.0 or above
Product dimensions	Length 8cm - Width 4.6cm - Height 4.2cm
Product weight	97.5g
Key features	1. HD 720p lens with manual focus 2. USB cable

2.1.4 Module RFID RC522

a) Module RC522

RFID Module RC522: Uses the MFRC522 IC from Phillip for reading and writing data to 13.56 MHz NFC cards. With its low cost and compact design, this module is the top choice for RFID card reading/writing applications.

Figure 2.5: *Module RFID RC522 [9]*

Table 2.4: Technical specifications of the Module RFID RC522

Specifications	Detail
Power	3.3VDC, 13 - 26mA
Standby current	10-13mA
Sleep mode current	<80uA
Carrier frequency	13.56MHz
Operating distance	0~60mm (Mifare1 card)
Interface	SPI
Max data rate	10Mbps
Supported RFID card types	Mifare1 S50, Mifare1 S70, Mifare UltraLight, Mifare Pro, Mifare Desfire

Connections

Uses SPI interface to connect to ARDUINO, through 4 pins: SCK, MISO, MOSI, SS

SCK: Clock for SPI communication, as SPI is synchronous, a clock line is needed. Each pulse on SCK signals one data bit sent or received.

MISO: If Master chip, this is input, if Slave chip, this is output. MISO of Master and Slaves are directly connected.

MOSI - Master Output / Slave Input: If Master chip, this is output, if Slave chip, this is input. MOSI of Master and Slaves are directly connected.

SS - Slave Select: Line to select the Slave to communicate with, active low..

b) Mifare S50 RFID card

The Mifare S50 RFID card uses the ISO/IEC 14443 A standard compatible with the MFRC522 RFID module. Some characteristics:

Passive card, no battery used.

Max range up to 100mm depending on antenna design.

Operating frequency 13.56 MHz.

Figure 2.6: *Mifare magnetic card [9]*

1 Kbyte EEPROM, organized into 16 sectors with 4 blocks each. Each block has 16 bytes.

Data retention up to 10 years.

Read/write endurance 100,000 cycles.

2.1.5 Servo Motor

A Servo is a special type of electric motor.

Figure 2.7: *SG90 Servo motor [9]*

Unlike regular motors that just spin continuously when powered, a Servo only rotates when controlled (by a PWM signal) within a specified angle range of 0-180 degrees. Each servo type has different size, weight and construction. Some are lightweight at just 9g (mainly used in model aircraft), some have tremendous torque (dozens of N/m), while others are rugged and precise.

The 9G RC Servo Motor is commonly used in small, simple radio-controlled models like robot arms. It has fast response speed, integrated motor driver, and easy angle control via PWM signal modulation.

Table 2.5: Technical specifications of the Servo SG90

Specifications	Detail
Dimensions	23mm x 12.2mm x 29mm
Weight	9 grams

Specifications	Detail
Operating Voltage	4.2-6V
Temperature Range	0°C - 55°C
Speed	0.3 sec/60 degrees

2.1.6 Infrared Sensor

The infrared obstacle sensor can adapt to the environment and has a pair of infrared transmitter and receiver. The infrared beam is emitted at a certain frequency. When an obstacle is detected in the transmission path (reflective surface), the beam reflects back to the infrared receiver. After comparison, the green LED lights up, and simultaneously the output provides a digital signal (a low level signal). The effective working distance is 2-5cm, with an operating voltage of 3.3V to 5V. The light sensitivity of the infrared obstacle sensor is adjusted by a voltage divider. The sensor is easy to assemble and use... It can be widely used in obstacle avoidance robots, obstacle avoidance vehicles, and line following applications...

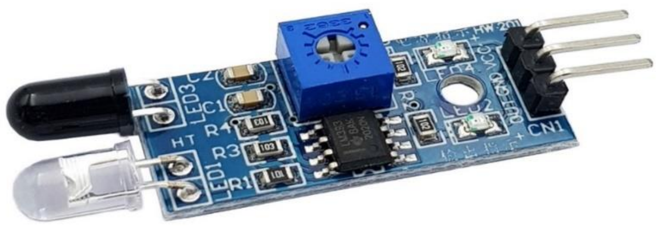

Figure 2.8: *Infrared Sensor [9]*

Technical specifications

Comparator:LM393

Operating voltage: 3.3V - 5V DC

When powered on, the red LED lights up

3mm screw holes for easy mounting

Dimensions: 3.2cm * 1.4cm

Two terminals: Negative, Positive

2.2 Related theories and technologies

2.2.1 Image processing concept

Image processing and graphics play an important role in human-machine interaction. The image processing process is considered as the manipulation of the input image to produce the desired output. The output of an image processing process can be a "better" image or a conclusion.

Figure 2.9: *Image processing concept [11]*

An image can be viewed as a set of pixels, and each pixel can be viewed as an intensity feature or some characteristic at a particular location of the object in space, and it can be viewed as an n-variable function P(c1, c2, ..., cn). Therefore, images in image processing can be viewed as n-dimensional images.

2.2.2 Vehicle licemse plate recognition concept with EmguCV

a) Concept

EmguCV, a .NET wrapper for the OpenCV computer vision library, brings the powerful capabilities of computer vision integration to your .NET applications. Leveraging the strength of EmguCV, license plate recognition becomes more fascinating and engaging, opening up opportunities for applications ranging from traffic monitoring to cargo security and beyond.

b) An outline of how to approach license plate recognition using EmguCV

- **Preprocessing**

  Convert the input image to grayscale.

  Apply Gaussian blurring to reduce noise.

Use morphological operations like dilation and erosion to enhance the quality of the image.

- **License Plate Detection**

  Utilize techniques like edge detection (Canny edge detector) to find potential contours.

  Filter out contours based on size, aspect ratio, and other characteristics to identify regions that could be license plates.

  Extract these regions for further processing.

- **Character Segmentation**

  Once you have potential license plate regions, segment the characters within these regions.

  Techniques like histogram projection or contour analysis can be used to separate individual characters.

  Ensure that the characters are correctly segmented, considering variations in font size, spacing, and orientation.

- **Character Recognition**

  Train a machine learning model (such as a Convolutional Neural Network) to recognize individual characters.

  Alternatively, use techniques like template matching or OCR (Optical Character Recognition) for character recognition.

  Ensure that the recognition algorithm can handle variations in lighting, angle, and font style.

- **Post-processing**

  Validate the recognized characters to ensure accuracy.

  Implement error-correction mechanisms to handle any misread characters.

  Reconstruct the license plate number from the recognized characters.

  Optionally, perform additional checks such as verifying the plate against a database or checking for special characters.

- **Integration**

Integrate the license plate recognition module into your application's workflow.

Provide user-friendly interfaces for inputting images, displaying results, and interacting with the recognized data.

Handle any errors or exceptions gracefully to ensure smooth operation.

2.2.3 RFID Technology

a) Concept

RFID is a technology that identifies objects using radio waves. This technology allows the identification of objects through a radio transmitter-receiver system, enabling the monitoring, management, or tracking of each object.

b) Structure

An RFID system consists of two main components: a reader device and an RFID tag attached to the object of interest. The reader device is equipped with an antenna to transmit and receive electronic waves, while the RFID tag is attached to the object being identified. Each RFID tag contains a unique, non-duplicated code.

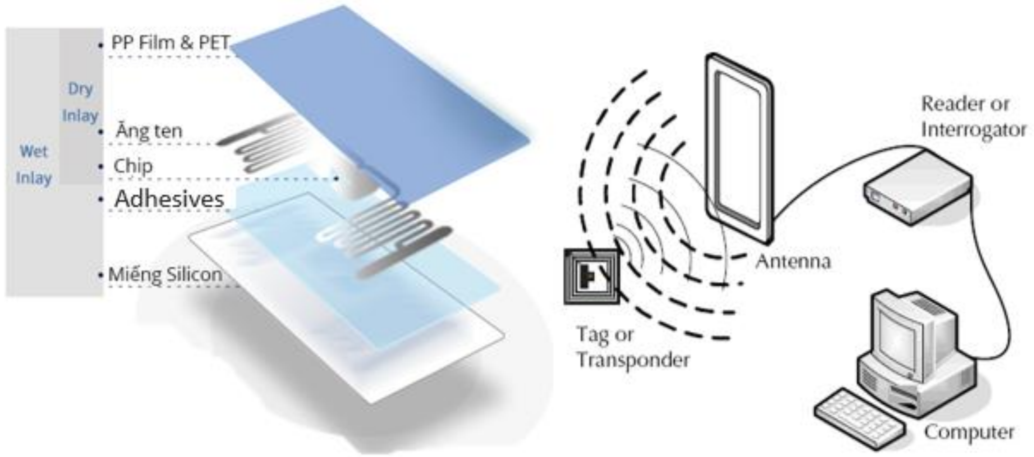

Figure 2.10: *RFID system structure [12]*

c) Operating principle

The RFID reader emits electronic waves at a specific frequency. When an RFID tag is within the operating range, it detects the electromagnetic waves from the reader and receives energy. The tag then transmits its code back to the RFID reader, allowing the reader to identify which tag is within the operating range.

d) Application

Due to its technological advantages, the application and security of RFID devices are highly regarded.

Applications include

Goods traffic management

Warehouse managent

Automated toll/registration management

2.2.4 Communication Technologies in the System

a) USB Communication Standard

- **Concept**

  USB (Universal Serial Bus) is a versatile serial connection standard in computers. USB is used to connect peripheral devices to computers, and they are usually designed as plug-and-play connectors that allow hot-swapping of devices (connecting and disconnecting devices without the need to restart the system).

- **Characteristics**

  Allows the expansion of up to 127 connected devices to a single computer through a single USB port.

  With USB 2.0 high-speed standard, the data rate can reach up to 480 Mbps.

  A USB cable consists of two power wires (+5V and GND) and a twisted pair of data wires. The computer can provide up to 500mA at 5V DC through the power wires.

Low-power devices (e.g., mouse, keyboard, low-power computer speakers) can be powered directly from the USB ports without the need for a separate power supply. High-power devices (such as printers, scanners) do not use the USB power as their primary power source, but the USB connection serves as a signal reference.

USB devices have hot-swap capability, meaning they can be connected or disconnected at any time without the need to restart the system.

Many USB devices can be put into a suspended state when the computer enters a power-saving mode.

b) SPI Communication Standard

- **Concept**

SPI is a popular communication protocol used by many different devices. For example, SD card modules, RFID readers, and 2.4 GHz wireless transceivers all use SPI to communicate with microcontrollers.

The main advantage of SPI is that data can be transmitted without interruption. Any number of bits can be sent or received in a continuous stream. With I2C and UART, data is sent in packets, limited to a specific number of bits. Start and stop conditions define the beginning and end of each packet, so the data is interrupted during transmission.

- **Data transmission**

SPI devices have a master-slave relationship. The master is the controlling device (usually a microcontroller), while the slaves (typically sensors, displays, or memory chips) receive commands from the master. The simplest SPI configuration is a single slave and a single master, but a master can control multiple slaves.

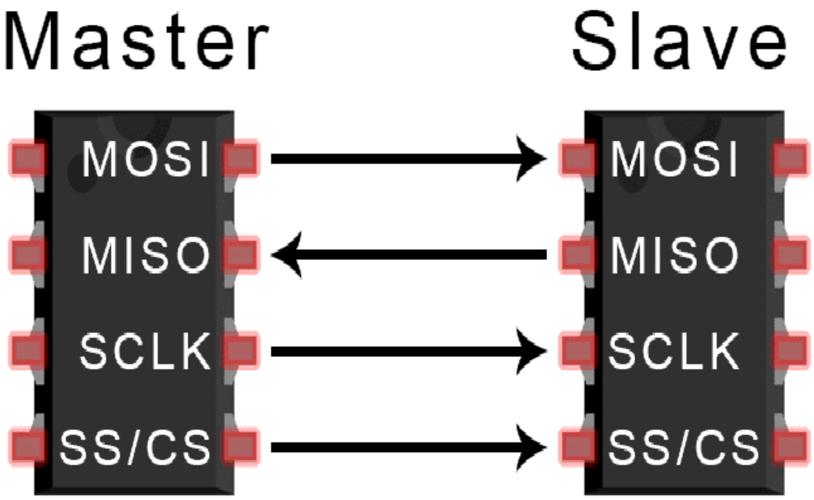

Figure 2.11: *SPI Communication [13]*

The steps in SPI data transmission are

The master generates the clock signal.

The master pulls the SS/CS pin low to activate the slave.

The master sends data bits one by one to the slave along the MOSI line. The slave reads the bits as it receives them.

If a response is required, the slave will send data bits one by one back to the master along the MISO line. The master reads the bits as it receives them.

- **Advantages and Disadvantages**

Advantages

No start and stop bits, so data can be transmitted continuously without interruption.

No complex slave addressing system like I2C.

Higher data transmission speed than I2C (nearly twice as fast).

Separate MISO and MOSI lines, so data can be sent and received simultaneously.

Disadvantages

Uses four wires (I2C and UART use two).

No acknowledgment that data has been received successfully (I2C has this).

No error checking like parity bits in UART.

Only allows a single master

c) I2C Communication

- **Concept**

I2C (Inter-Integrated Circuit) is a synchronous serial communication protocol developed by Philips Semiconductors, used to transmit data between ICs using only two signal lines.

Data bits are transmitted one by one at regular time intervals set by a clock signal.

The I2C bus is commonly used to communicate with peripherals for many different types of ICs, such as microcontrollers, sensors, and more..

- **Structure**

I2C uses two signal lines:

SCL: Clock signal generated by the Master

SDA: Data transmission line.

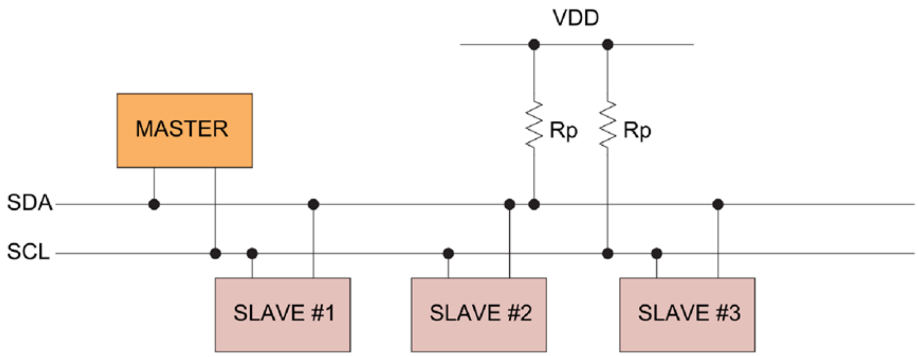

Figure 2.12: *I2C Communication [14]*

The I2C communication involves the process of data transmission and reception between master and slave devices.

The Master device is a microcontroller, responsible for controlling the SCL signal line and sending/receiving data or commands through the SDA line to other devices.

The devices that receive the data, commands, and signals from the Master device are called Slave devices. Slave devices are typically ICs, or even microcontrollers.

The Master and Slave are connected as shown in the figure. Both the SCL and SDA lines operate in Open Drain mode, meaning any device connected to the I2C network can only pull the bus lines low, but cannot drive them high. This is to avoid a short circuit when one device tries to pull the bus high while another pulls it low. Therefore, a pullup resistor (1 - 4.7 kΩ) is needed to keep the default level high..

I2C Transmission frame:

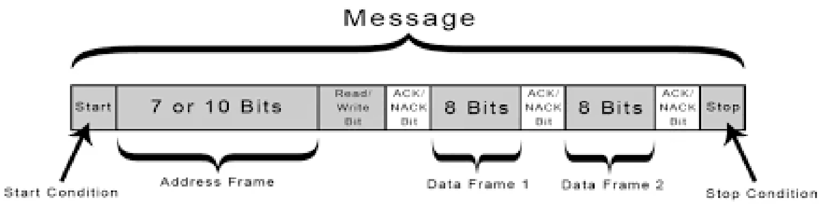

Figure 2.13: *I2C Transmission frame [14]*

Address Bits: Typically, the transmission process will involve many devices and ICs. To distinguish these devices, they are assigned a fixed 7-bit physical address.

Read/Write Bit: This bit is used to determine whether the process is transmitting or receiving data from the Master device. If the Master is sending data, this bit is set to '0', and vice versa, it is set to '1' when receiving data.

ACK/NACK Bit: Short for Acknowledged / Not Acknowledged. This is used to compare the physical address of the device with the address sent. If they match, the Slave will be set to '0', and if not, it will default to '1'.

Data Bits: Consist of 8 bits and are set by the transmitting device to the receiving device. After these bits are sent, an immediate ACK/NACK bit is sent to confirm that the receiving device has successfully received the data. If the reception is successful, the ACK/NACK bit is set to '0', and vice versa.

- **Data transmission process**

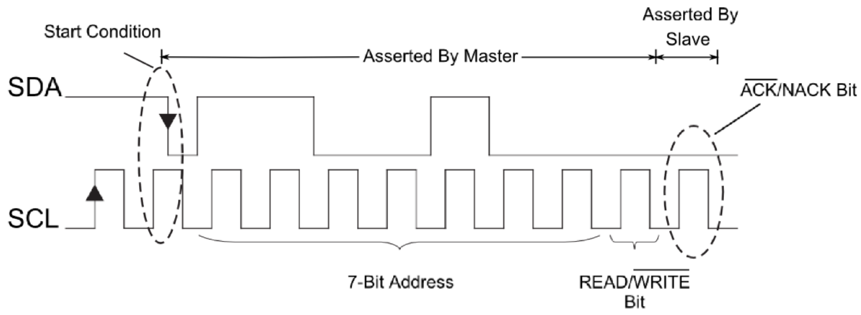

Figure 2.14: *I2C Data transmission [14]*

Start: The Master device will send a Start signal by pulling both the SDA and SCL lines from high to low.

Next, the Master sends a 7-bit slave address along with the Read/Write bit.

The Slave will compare the physical address with the one just received. If they match, the Slave will acknowledge by pulling the SDA line low and setting the ACK/NACK bit to '0'. If they don't match, the SDA line and ACK/NACK bit will default to '1'.

The Master device will then transmit or receive the data frame. If the Master is sending to the Slave, the Read/Write bit is set to 0. If the Master is receiving, the bit is set to 1.

If the data frame was successfully transmitted, the ACK/NACK bit is set to 0 to indicate to the Master to continue.

After all the data has been successfully sent to the Slave, the Master will generate a Stop signal to notify all Slaves that the transmission has ended by transitioning both SCL and SDA from low to high.

- **Advantages and disadvantages**

Advantages

Uses only two wires

Supports multiple masters and multiple slaves

ACK/NACK bit confirms each frame is transferred successfully

Less complex hardware compared to UART

Well-known and widely used protocol

Disadvantages

Slower data transmission speed than SPI

Data frame size limited to 8 bits

Requires more complex hardware to implement compared to SPI

Signals are analog using digital signal.

d) UART Communication standard

- **Concept**

UART stands for Universal Asynchronous Receiver-Transmitter. It is an integrated circuit used for serial data communication between computers and peripheral devices.

- **Data Transmission**

The Tx (transmit) pin of one chip is directly connected to the Rx (receive) pin of the other chip, and vice versa. Typically, the transmission process will occur at 3.3V or 5V. UART is a one-master, one-slave protocol, where one device is set up to communicate with only one other device.

Data is transmitted to and from the UART in parallel with the controlling device (e.g., CPU).

When sending on the Tx pin, the UART first serializes this parallel information and transmits it to the receiving device.

The second UART receives this data on its Rx pin and converts it back to parallel to communicate with its controlling device.

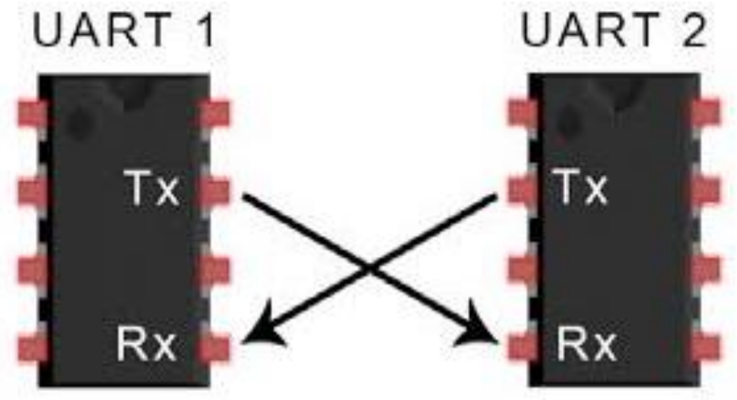

Figure 2.15: *UART Data Transmission [15]*

Advantages and disadvantages

Advantages

Only uses two wires.

No clock signal required.

Has a parity bit to allow for error checking.

The data frame structure can be changed as long as both sides are set up for it.

Well-documented and widely used method.

Disadvantages

The data frame size is limited to a maximum of 9 bits.

Does not support multiple slave systems or multiple master systems.

The baud rate of each UART must be within 10% of each other.

e) USB Communication standard

- **Concept**

USB, or Universal Serial Bus, is an industry standard that establishes specifications for cables, connectors, and protocols for connection, communication, and power supply between computers and peripheral devices. USB has evolved through several versions, including USB 1.x, USB 2.0, USB 3.x, and the latest USB4, each offering improvements in data transfer speed and power delivery.

- **Data Transmission**

The Setup Transaction stage begins with the host sending a "Setup" Token along with 8 bytes of request-identifying data in the "Data" field. The receiving device then responds by sending an "ACK" acknowledgment signal or a "NAK" non-acknowledgment signal in the "Handshake" field. This process lays the groundwork for subsequent data transmission operations.

The Data Transaction stage involves transferring data from the host to the device (OUT) or vice versa (IN). This data is packaged in the "DATA/0" field and may be accompanied by an acknowledgment or non-acknowledgment signal in the "Handshake" field. This process enables bidirectional data transmission between the host and the device in a flexible manner.

Finally, the Status Transaction stage includes sending a "Status" Token along with 0 bytes of data in the "Data" field. The receiving device then responds with an acknowledgment or non-acknowledgment signal in the "Handshake" field to confirm the receipt of data. This process ensures the integrity and reliability of the data transmitted via USB Control Transfer.

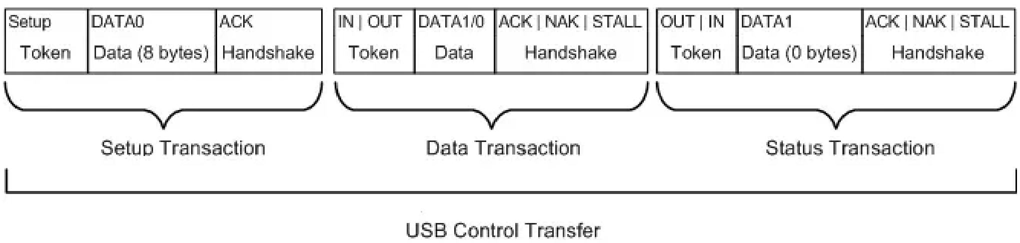

Figure 2.16: *USB Control Transfer [16]*

- **Advantages and disadvantages**

Advantages

High data transfer speed: USB supports data transfer speeds up to 480 Mbps (USB 2.0), 5 Gbps (USB 3.0), and 10 Gbps (USB 3.1).

Power supply capability: USB can provide power to peripheral devices, reducing the need for separate power sources.

Plug-and-Play: USB supports easy device connection and recognition without the need to restart the computer.

Expandability: USB supports connections to multiple devices through hubs.

High mobility: USB allows flexible device connection and disconnection..

Disadvantages

Cable length limitation: USB cable length is limited, typically not exceeding 5 meters.

Power supply limitation: USB has limitations in providing power to high-power-consuming devices.

Security risks: Improper management and control of USB connections can pose security risks.

Compatibility issues: Compatibility issues may arise with devices or older USB implementations.

Not suitable for real-time data transmission: USB is not suitable for applications requiring strict real-time data transmission.

2.3 Conclusion

The smart parking system utilizes key components such as Arduino Uno, Arduino Nano, Webcam, RFID Module RC522, Servo motors, and infrared sensors to create an efficient and convenient solution for parking management. By employing image processing technology with EmguCV and license plate recognition, the system automates the vehicle identification and verification process, enhancing accuracy and processing speed. Communication protocols like USB, SPI, I2C, and UART are integrated to ensure effective connectivity and data transmission between components.

The flexibility of the system lies in its ease of expansion and upgrades, supported by the robust Arduino community and open-source resources. This allows developers and engineers to customize and enhance the system's features according to specific needs. Additionally, the use of widely available devices and components helps reduce costs and increases feasibility for practical deployment.

In summary, the smart parking system not only offers convenience and security for users but also contributes to efficient parking lot management, saving time and resources. With the continuous advancement of technology, this system promises to become increasingly sophisticated and an essential part of smart urban infrastructure.

CHAPTER 3: IMPLEMENTATION

3.1 System requirements analysis

3.1.1 System requirement

a) System analysis

Identify the main functions of the system such as license plate recognition, vehicle in/out management, parking spot control, fee calculation,...

Determine necessary hardware components: cameras, computers, control devices, display systems,...

Specify requirements for scalability, security, and reliability of the system.

b) Design system

Design the overall architecture of the system with its components and relationships between them.

Design flowcharts, operation processes of the system.

Design the database to store vehicle information, owner details, parking spots, transaction history,...

Design user-friendly interfaces for ease of use.

3.1.2 Solutions

Utilize license plate recognition technology through image processing and computer vision to identify vehicles in/out.

Integrate RFID system for authentication and management of registered vehicles/owners.

Employ surveillance camera system to analyze parking spot locations.

Develop software for parking lot management, fee calculation, and report generation.

Deploy information display systems at the parking lot to guide users.

Apply IoT and cloud computing technologies for remote connectivity and management.

Ensure system security and scalability for future expansion**.**

3.2 System design

3.2.1 RFID Technology

a) Circuit Diagram

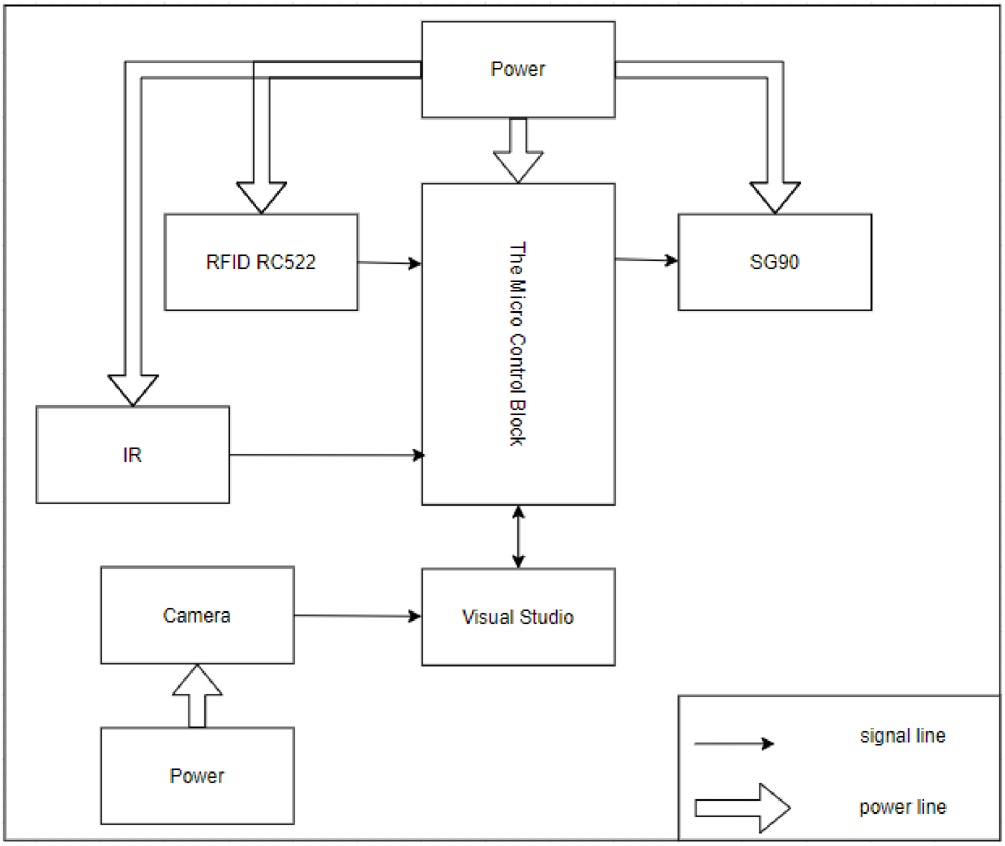

Figure 3.1: *Hardware connection diagram*

The hardware connection system consists of:

Central processing unit block: Arduino Nano

Sensor block including:

Infrared sensor LM393

RFID card reader block: RC522

Barrier motor block: SG90

b) System principle diagram

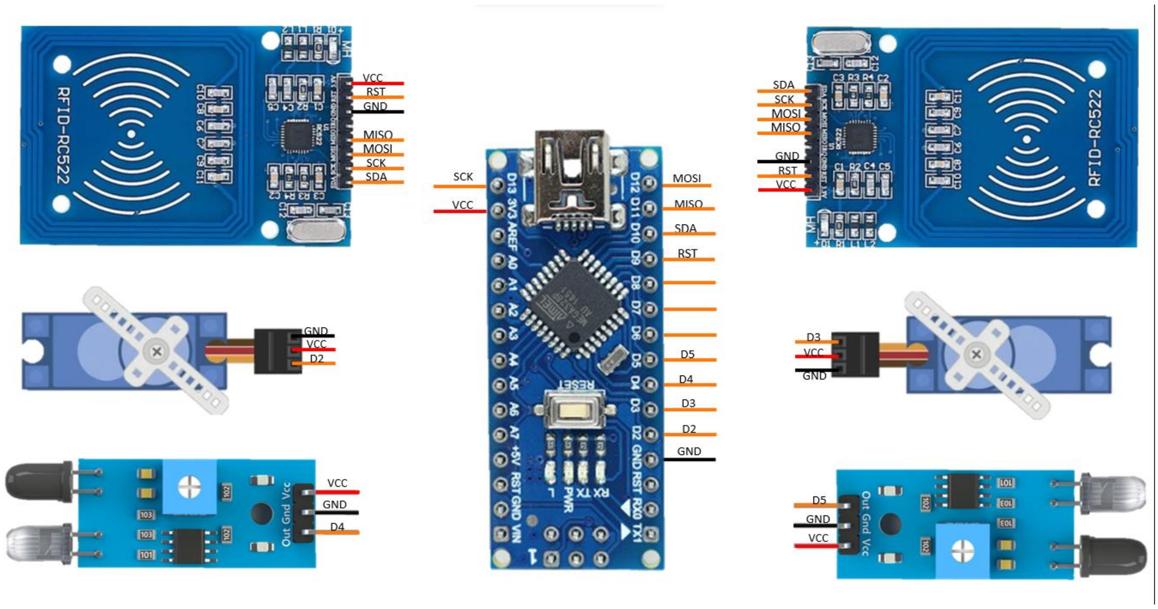

Figure 3.2: *Circuit diagram of hardware connection*

The system principle diagram includes the following

The power block is connected to a 12V adapter, passing through an LM2596 module to stabilize the current to the circuit.

The central processing unit block, Arduino Nano, is connected to display blocks, sensors via its pins.

SCL, SDA at pins A5, A4 for I2C communication with the LCD display block.

SCK, MISO, MOSI, pins D10, D8 for SPI communication with the RFID card reader block RC522.

Servo1 and Servo2 at pins D2, D3 are connected to the barrier motor block SG90 to control 2 barriers for entry and exit.

Connect the infrared sensor signal to D4 and D5 to initialize the sensor.

3.2.2 Software design for the system.

a) Image processing software with Visual Studio

Visual Studio is one of the most popular programming tools from Microsoft, written in two main languages: C# and VB+. It provides users with an easy and efficient way to develop systems.

Figure 3.3: *Visual studio software [17]*

Visual Studio is a system programming software directly produced by Microsoft. Since its inception, Visual Studio has had many different versions, allowing users to choose the version compatible with their machine and suitable configuration. Additionally, Visual Studio allows users to customize the main interface according to their usage needs.

b) Arduino IDE programming software

Arduino is an open-source integrated development environment that allows users to easily write and upload code to the board. The development environment is written in Java and is based on the Processing programming language and other open-source software. This software can be used with any Arduino board.

Since March 2015, the Arduino IDE (Integrated Development Editor) has been downloaded more than 8 million times. Currently, it is not only used for Arduino and Genuido boards but also by hundreds of companies worldwide to program their devices, including equivalents, clones, and even counterfeit products.

Figure 3.4: *Arduino IDE software. [18]*

Arduino is a cross-platform integrated development environment that works with an Arduino controller to write, compile, and upload code to the board. It supports a range of Arduino boards such as Arduino Uno, Nano, Mega, Esplora, Ethernet, Fio, Pro, Pro Mini, as well as LilyPad Arduino.

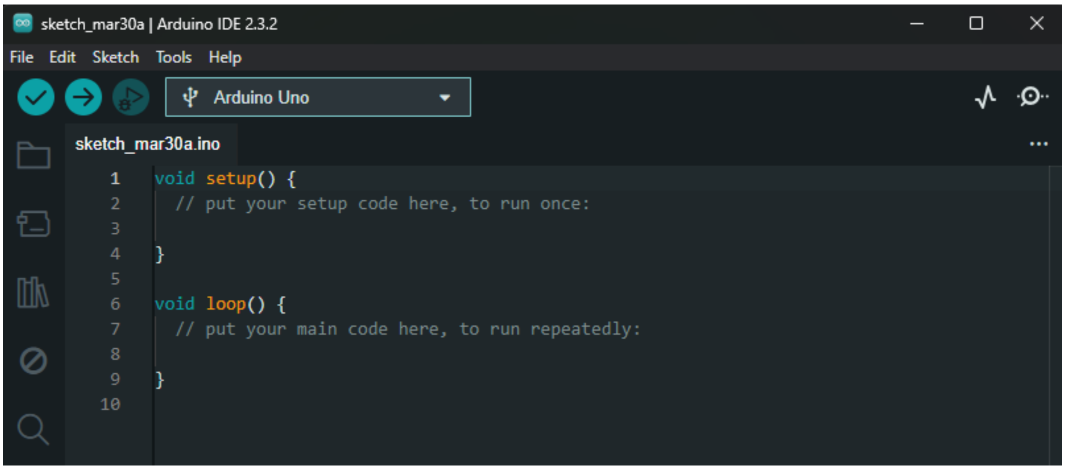

Figure 3.5: *Arduino IDE code writing interface.*

The Arduino IDE uses a universal language for Arduino, C and C++, making it suitable for programmers familiar with both languages. Features such as syntax highlighting, automatic indentation, etc., make it a modern replacement for other IDEs.

Wrapped within a logically arranged graphical interface, Arduino possesses features to attract Arduino developers, paving the way for successful output through debugging modules. All its features are stored within a few buttons, menus, making it easy to understand and navigate, especially for professional programmers. Additionally, integrating collections of example sketches helps first-time Arduino users familiarize themselves and quickly grasp applications.

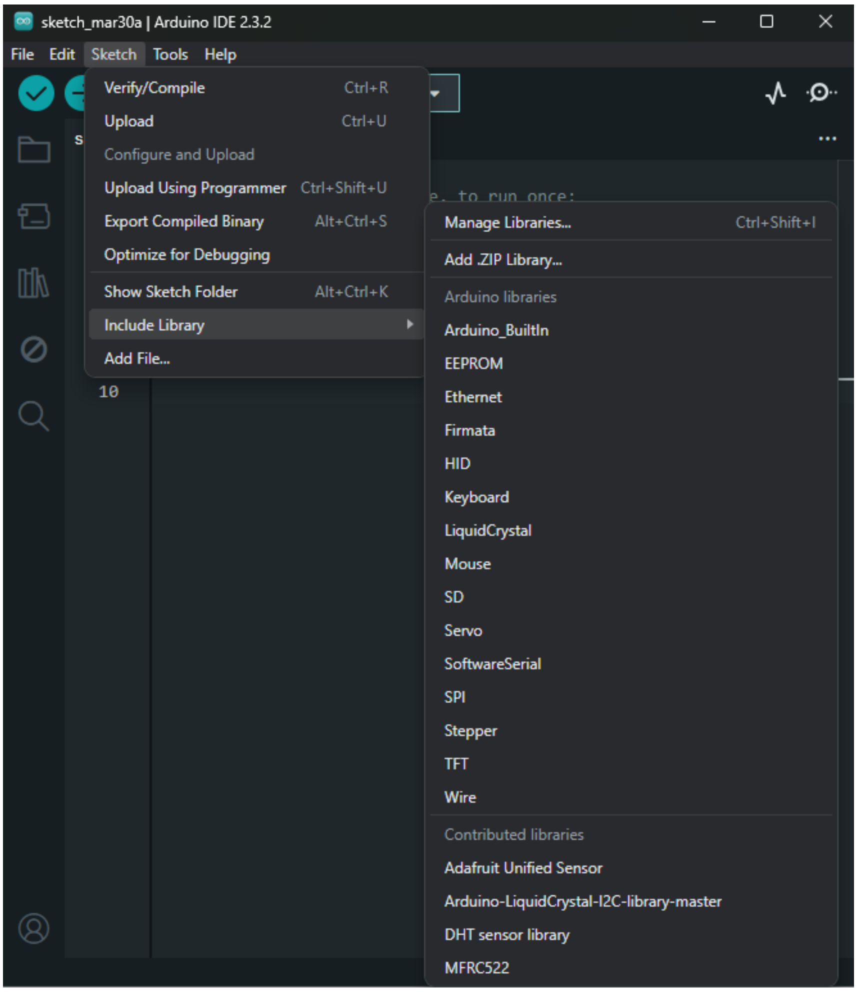

Figure 3.6: *Integrated Libraries in Arduino IDE.*

Once the Arduino board is connected to the computer and necessary drivers are installed, you can choose the working model using the Tools menu of the application. Then, you can start writing programs using the comfortable working environment provided by Arduino. The program includes an array of libraries such as EEPROM, Firmata, GSM, Servo, TFT, Wifi,… Of corse, you can also add your own libraries.

Designs can be tested, compiled with error messages displayed at the bottom of the user interface, allowing you to review the code. If the debugging process returns no errors, you can begin the process of uploading the code to the board and further testing.

In summary, Arduino is the development solution for Arduino boards, providing all the necessary elements that Arduino developers need to easily carry out the process of creating and testing their products.

3.2.3 Parking lot algorithm flowchat

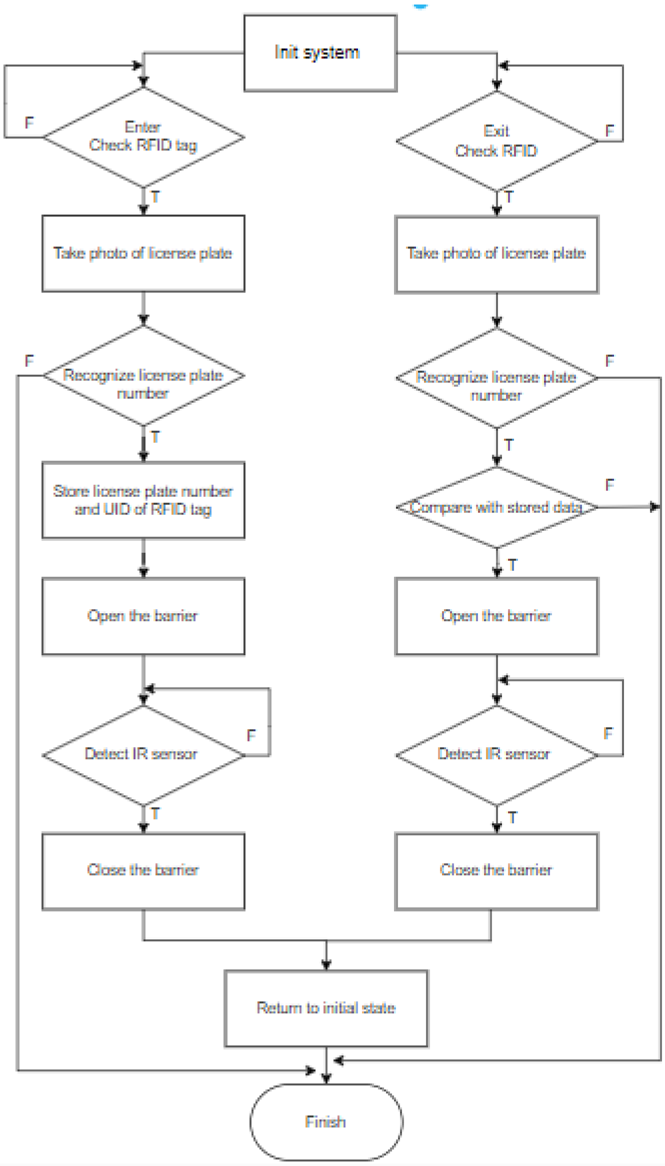

Figure 3.7: *Parking Lot Algorithm Flowchart*

The algorithm is described through the following steps:

Step 1: The system initializes and sets up variables, initializes UART communication with the hardware system.

Step 2: Check if there is an RFID card swipe signal for entry, the system moves to 2.1. If there is no RFID entry signal, the system moves to step 3.

Step 2.1: Capture license plate image.

Step 2.2: Recognize the license plate number.

Step 2.3: Store the license plate number in the database with the RFID tag just swiped.

Step 2.4: Send a signal to open the entry barrier and return to step 2.

Step 3: Check if there is an RFID card swipe signal for exit, the system moves to 3.1. If there is no RFID exit signal, the system returns to step 2.

Step 3.1: Capture the license plate image.

Step 3.2: Recognize the license plate number.

Step 3.3: Compare the license plate number with the RFID code in the database.

Step 3.4: If the license plate number is correct, send a signal to open the exit barrier. If incorrect, the system moves to step 3.5.

Step 3.5: Notify that the license plate number is invalid.

Step 3.6: Sound the alarm and return to step 2.

3.2.4 RFID tag processing algorithm flowchat

The algorithm is described through the following steps

Step 1: The system is initialized.

Step 2: Check if there is an RFID card swipe signal for entry, the system moves to 2.1. If there is no RFID entry signal, the system moves to step 3.

Step 2.1: Read the RFID tag data.

Step 2.2: Send the RFID code and a signal indicating a vehicle has entered to the App.

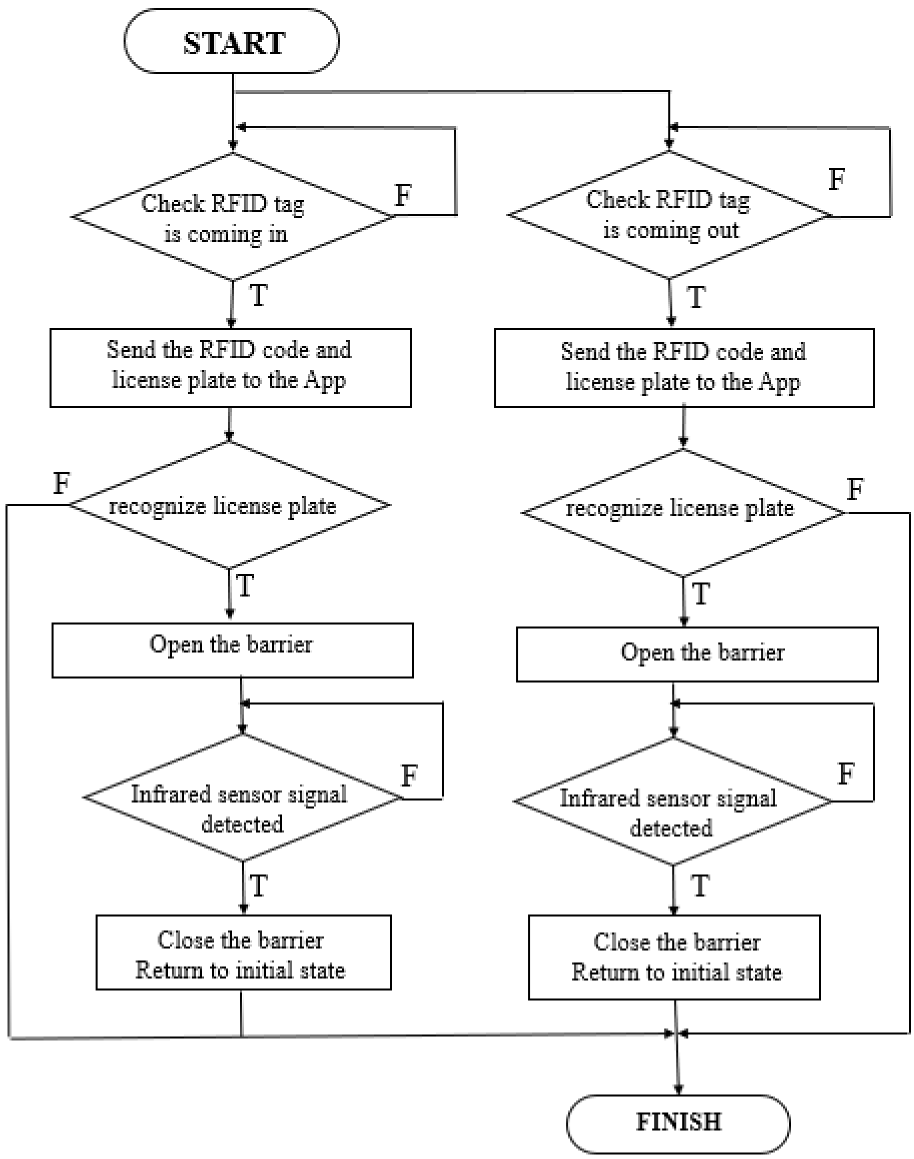

Figure 3.8: *RFID tag processing algorithm flowchat*

Step 2.3: If there is a signal to open the barrier, the system opens the entry barrier. If no signal is received, the system waits for the signal from the App.

Step 2.4: When the infrared sensor signal for entry is detected, the system closes the barrier and returns to step 2.

Step 3: Check if there is an RFID card swipe signal for exit, the system moves to 3.1. If there is no RFID exit signal, the system returns to step 2.

Step 3.1: Read the RFID tag data.

Step 3.2: Send the RFID code and a signal indicating a vehicle has exited to the App.

Step 3.3: If there is a signal to open the barrier, the system opens the exit barrier and moves to step 3.4. If no signal is received, the system moves to step 3.5.

Step 3.4: When the infrared sensor signal for exit is detected, the system closes the barrier and returns to step 2.

Step 3.5: If there is a signal indicating an invalid vehicle, the system sounds the alarm and returns to step 2.

3.3 System construction and integration

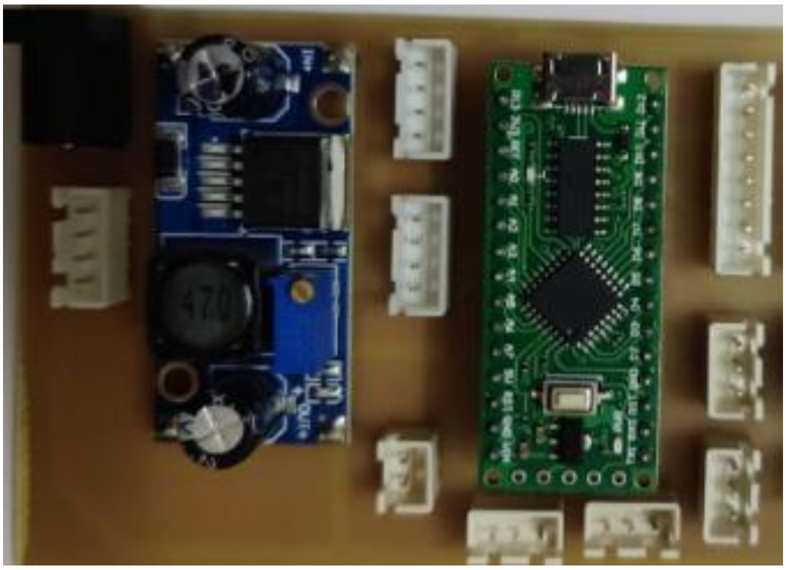

Figure 3.9: *Hardware connection*

After the circuit has been completed and the components, sensors, and wiring have been connected, the next step is to assemble all the components into the selected mold.

The decision to use fomec material to create the model has already been made. Now, the crucial step is to proceed with the installation of the components inside to finalize the complete model. This process requires great care and attention to detail, ensuring that all the elements are properly arranged and connected.

Successful assembly will result in a complete model, ready to be put into operation and tested. The combination of the finalized electronic circuit and the installation of components into the mold is an important step in transitioning the model from concept to reality. This is the final stage before the device can be put into active use.

The completion process demands close coordination between the design, assembly, and integration steps. Each stage must be carefully executed, from preparing the components to installing and verifying the entire system. Meticulous and diligent work in every step will ensure that the final model operates in a stable manner and meets the established requirements.

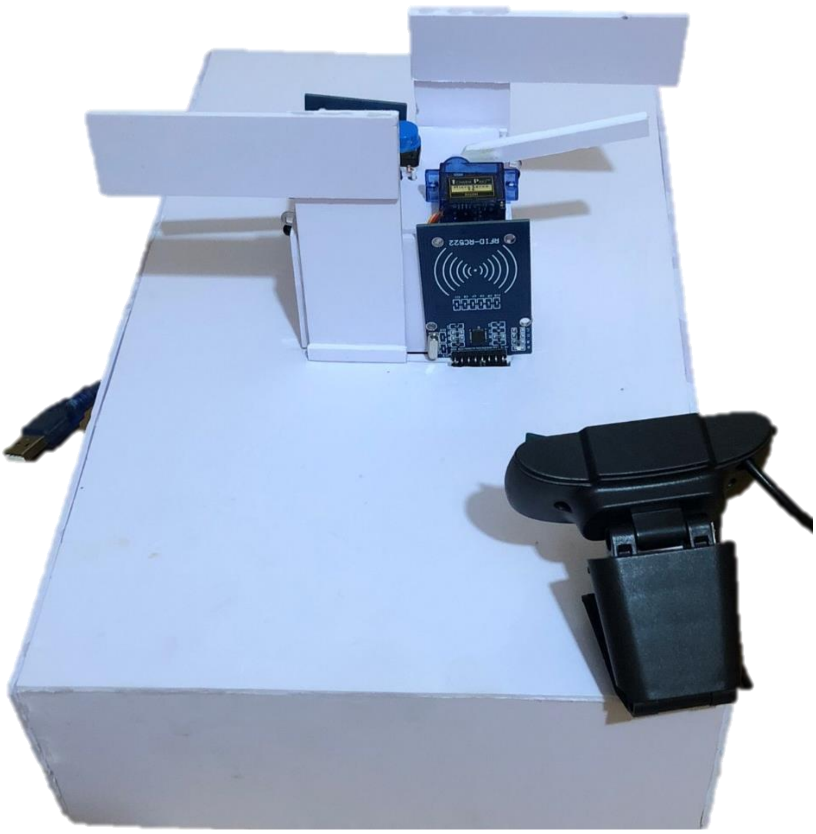

Figure 3.10: *Product model*

The interface of the system used for vehicle license plate recognition consists of the following main components:

- A section displaying video feeds from two different cameras (Camera1 and Camera2).
- An image processing and license plate recognition section that processes the obtained video feeds. The recognized license plate numbers are displayed next to the corresponding video feeds.
- Control buttons allowing the user to select cameras, connect/disconnect devices, and perform other tasks.
- A small window displaying a stream of numerical data being read from a device (possibly a card reader device).

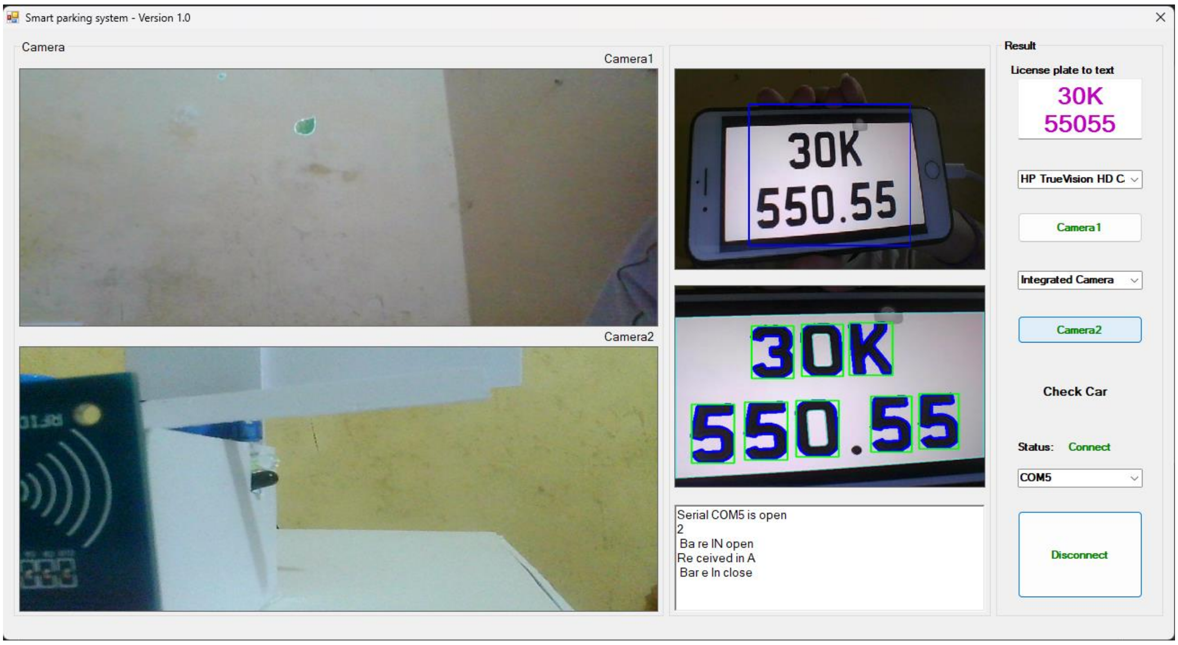

Figure 3.11: *Graphic of system*

3.4 System testing and evaluation

3.4.1 Testing methods

Integration Testing: We performed integration testing to ensure that all parts of the system work harmoniously and interact well with each other. This includes checking the connection between hardware and software, as well as communication between different components of the system.

Functional Testing: Functional testing was carried out to ensure that all functions of the system work as expected. This includes testing license plate recognition, vehicle in/out management, parking fee calculation, and other functionalities.

Performance Testing: We conducted performance testing to measure and evaluate the system's processing capabilities under high load conditions. This helps ensure that the system still operates strongly and efficiently even when there are many vehicles entering/exiting the parking lot.

3.4.2 System evaluation

Accuracy of License Plate Recognition: We evaluated the accuracy of the system in recognizing license plates. This evaluation result is based on the ratio of the number of correctly recognized license plates to the total number of scanned license plates.

Stability and Reliability: We evaluated the stability and reliability of the system under real-world conditions. This includes assessing the system's ability to operate under various conditions, including low light conditions, unfavorable weather, and changes in viewing angles.

Performance and Response Time: We evaluated the system's performance by measuring the response time from when a vehicle enters the parking lot until the information is updated in the system. This result helps us ensure that the system responds quickly and efficiently to users

3.5 System operation guide

The car park license plate recognition system is a convenient and modern solution to safely and efficiently manage vehicle access in and out of the parking facility. Users only need to follow a few simple steps to utilize the system optimally.

Preparing the System

Select the Camera: Use the software to choose the most suitable camera for the system.

Confirm Camera Operation: Verify and confirm that the camera is functioning correctly.

Choose the Connection Port: Connect the camera to the hardware system through the appropriate COM port.

When Entering the Car Park

Stop the vehicle at the designated marked position.

The camera system will automatically scan and recognize the vehicle's license plate.

Swipe the RFID card at the indicated location.

Once the license plate information and RFID card are verified, the barrier will automatically open, allowing the vehicle to enter the parking area.

When Exiting the Car Park

Stop the vehicle at the designated marked position.

The system will re-scan the license plate and read the RFID card.

If the information matches the entry details, the barrier will automatically open, allowing the vehicle to exit the parking facility.

Adhering to the usage guidelines will ensure the safety, efficiency, and accuracy of the system. Users can have peace of mind when accessing the car park, while also effectively managing the parking utilization.

The car park license plate recognition system is one of the modern technology solutions that helps optimize the management and control of the parking facility. With the simple and convenient usage process described above, drivers can confidently utilize the service in an efficient manner.

CONCLUSION

1. Achieved results of the project

The smart car park system with license plate recognition and RFID card scanning has been successfully implemented, bringing many practical benefits to users:

- Automation of the parking entry and exit process, saving time and increasing convenience for users.
- Enhanced management and control of the parking lot, reducing the issue of vehicles parked in unauthorized areas.
- Improved security and theft prevention, thanks to the accurate identification of incoming and outgoing vehicles.
- Detailed data collection on the number of vehicles entering and leaving, helping to better manage and plan the use of the parking lot

2. Limitations of the project

Although the system has achieved many successes, there are still some limitations that need to be improved:

- The accuracy of license plate recognition has not reached 100%, requiring upgrades to the software and database.
- The processing speed and response time of the system are not fast enough, affecting the user experience.
- The expandability and integration with other systems have not been fully developed

3. Future development directions of the project

To improve the effectiveness and applicability of the system, future development directions include:

- Improving the license plate recognition algorithm to increase accuracy and processing speed.
- Integrating additional features such as automatic payment, parking space management, navigation guidance, etc. to provide a better user experience.
- Researching solutions to reduce the installation and operating costs of the system, in order to increase feasibility and widespread adoption.
- Expanding the integration capabilities with other systems such as traffic management, security, etc. to increase the overall flexibility and efficiency of the entire system.

With these improvements and future developments, the smart car park system will become increasingly perfect, better meeting the needs of users and contributing to the construction of a smart, modern city

REFERENCES

[1] Department of information technology (2023), "*Lập trình vi điều khiển*", Hung Yen university of technology and education

[2] Trương Xuân Đạt. "*EmguCV - OpenCV cho .NET - Một Số Thao Tác Xử Lý Ảnh Cơ Bản - I.*" IO Stream, 18 Sep. 2020, *www.iostream.co/article/emgucv-opencv-cho-dot-net-mot-so-thao-tac-xu-ly-anh-co-ban-i-ivicg.* Accessed 24 February 2024.

[3] Cao Oanh. "*Tìm hiểu cơ bản về RFID.*" MEGATECHCO, 17 Mar. 2022, www.megatechco.vn/tim-hieu-co-ban-ve-rfid/. Accessed 27 March 2024.

[4] Dientutuonglai.com. "*Giới thiệu về Arduino Nano.*" Điện Tử Tương Lai, www.dientutuonglai.com/gioi-thieu-arduino-nano.html. Accessed 16 March 2024."

[5] Trương Xuân Đạt. "*EmguCV - OpenCV cho .NET - Giới Thiệu và Cách Tích Hợp vào .NET.*" IO Stream, 18 Sep. 2020, *www.iostream.co/article/emgucv-opencv-cho-dot-net-gioi-thieu-va-cach-tich-hop-vao-dot-net-cAicg.* Accessed 25 February 2024.

[6] "*Arduino Nano là gì? Tại sao gọi là bảng mạch của tương lai?*" 3D Linh Kiện, *www.3dlinhkien.com/arduino-nano-la-gi-tai-sao-goi-la-bang-mach-cua-tuong-lai-n77669.html.* Accessed 16 March 2024.

[7] "*Tìm hiểu về chuẩn giao tiếp SPI*" Thế Giới IC, *www.thegioiic.com/tin-tuc/tim-hieu-ve-chuan-giao-tiep-spi*. Accessed 27 February 2024.

[8] "*Tìm hiểu về chuẩn giao tiếp I2C*" Thế Giới IC, *www.thegioiic.com/tin-tuc/tim-hieu-ve-chuan-giao-tiep-i2c*. Accessed 19 May 2024.

[9] "Linh Kiện Điện Tử Hưng Yên. Available on Shopee. Retrieved from https://shopee.vn/linhkiendientuhungyen."

[10] Máy Tính Đồng Bộ. (n.d.). Webcam Camera HD 720P dùng cho máy tính, laptop. Retrieved June 5, 2024, from https://maytinhdongbo.com/p/webcam-camera-hd-720p-dung-cho-may-tinh-laptop.html

[11] Hoang. (2018, October 12). Tự học xử lý ảnh: Nhập môn xử lý ảnh. Tinh Học Cơ Bản. Retrieved June 5, 2024, from https://www.tinhoccoban.net/2018/10/tu-hoc-xu-ly-anh-nhap-mon-xu-ly-anh.html.

[12] Vinafco. (n.d.). RFID: Còn Hơn Là Một Mã Vạch. Retrieved June 5, 2024, from https://vinafco.com.vn/rfid-con-hon-la-mot-ma-vach.

[13] Campbell, S. (2019). “Basics of the SPI Communication Protocol.” Circuit Basics. Available at: https://www.circuitbasics.com/basics-of-the-spi-communication-protocol/ [Accessed June 5, 2024].

[14] D'Souza, R. S. (2019). “I2C Protocol.” Prodigy Technovations. Available at: https://www.prodigytechno.com/i2c-protocol [Accessed June 5, 2024].

[15] Campbell, S. (n.d.). Basics of UART Communication. Circuit Basics. Retrieved June 5, 2024, from https://www.circuitbasics.com/basics-uart-communication/.

[16] Microsoft. “USB Control Transfer.” Windows Hardware Drivers Documentation, Microsoft. Accessed June 6, 2024, from https://learn.microsoft.com/en-us/windows-hardware/drivers/usbcon/usb-control-transfer.

[17] Microsoft Visual Studio. “Visual Studio.” Accessed June 6, 2024, from https://visualstudio.microsoft.com/.

[18] Arduino. “Arduino Software.” Accessed June 6, 2024, from https://www.arduino.cc/en/software.