UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN
COMPUTER ENGINEERING

"Industrial Single Board Computer adoption in a Parking Automation System: An experimental development performance evaluation"

Relatore: Prof. Ing. NICOLA ZINGIRIAN

Laureando /a: NEEMOL RAJKUMARI

ANNO ACCADEMICO 2022-2023
Data di laurea
12-10-2023

CONTENTS

LIST OF FIGURES

ABSTRACT

The emerging trend of replacing proprietary embedded microcontroller boards with industrial single board computer (SBC) in automation applications appears an attractive product management choice. This choice allows the automation system vendors to take advantage of much more available skills in the job market. While the traditional embedded microcontroller boards in fact require the increasingly hard-to-find skills of C language firmware developers, the new industrial SBCs just demand the skills of Visual Studio C# language programmers on the Windows operating system, which are widely available in the job market. However, the well-known stability of microcontroller embedded firmware against the questionable robustness of Windows-based software appears an open issue in such a hardware asset migration. The thesis experimentally analyzes this issue taking advantage of an internship program in a parking automation company managing the transition from the Rabbit Semiconductors BL4S2xx microcontroller board to the PICO ITX-type industrial SBC as a core hardware asset. The work specifically focused on the device driver porting from the old to the new hardware environment and the analysis offers a useful and impartial insight of the hardware transition pros and cons.

CHAPTER 1: INTRODUCTION

In our rapidly urbanizing world, the demand for efficient, space-saving, and sustainable transportation solutions has never been greater. As cities continue to grow in population and the number of vehicles and roads increases, parking stands out as one critical challenge. Traditional parking methods, characterized by sprawling lots and multi-story garages, are struggling to keep up the rapid demand. This has resulted in traffic congestion, wasted time, and environmental concerns. However, a new era of automation and efficiency has emerged thanks to the tremendous technological advancements. It is against this backdrop that Automated Parking Systems (APS) emerge as a transformative solution, reshaping the approach to urban mobility and parking management. Parking management is undergoing profound changes focusing on enhancing the user experience, optimizing space utilization, reducing human intervention, and transforming traditional parking into intelligent and automated solutions.

This transformation has been catalyzed by the introduction of Industrial Single Board Computers (SBCs) into Parking Automation Systems, representing a paradigm shift in parking facility management. SBCs, known for their compact size and reliable performance, have found applications in various industries, including manufacturing, logistics, healthcare, and smart cities. These robust computer platforms are driving the conversion of conventional parking facilities into intelligent, efficient, and user-centric spaces within the context of parking automation.

The Need for Automation in Parking:

With urbanization on the rise, parking spaces are becoming scarcer and more expensive than ever before as urbanization continues to rise. Traditional manual parking management methods face difficulties such as inefficient space allocation, traffic congestion, long waiting times, and operational errors. These problems not only inconvenience customers but also harm the environment due to increased emissions from vehicles circling for finding parking slots.

Intelligent parking solutions that take advantage of the technology to maximize space utilization, minimize human involvement, and enhance the overall user experience are becoming increasingly essential to address these challenges. Industrial SBCs bridge this gap by providing the connectivity and processing power necessary to build advanced parking automation systems.

1. The Urban Parking

Urban centers worldwide are battling a variety of difficulties, and among them, the scarcity of parking spaces stands out as a prominent and persistent issue. As city populations surge, the demand for parking continues to surge alongside it. Traditional parking facilities, consisting of sprawling surface lots and multi-story garages, struggle to meet this escalating demand.

- Congestion: People spending a lot of time looking for parking not only makes their travel longer but also makes traffic worse. This traffic jam makes cars use more fuel, create more pollution, and makes the air not as clean.
- Inefficiency: Surface parking lots consume vast amounts of valuable urban land, and multi-story parking garages often suffer from underutilization. This inefficiency in space allocation not only impacts urban aesthetics but also represents an economic inefficiency.
- Environmental Impact: Cars driving around in circles looking for parking release unnecessary pollution, which makes the environment in cities worse and adds to global warming worries.

2. Enter Automated Parking Systems (APS)

Automated Parking Systems bring a glimmer of hope to the challenging world of urban parking. These smart automated solutions employ robotics, sensor technology, and advanced software to transform how parking works. They efficiently direct the vehicles to designated parking spots, eliminating the need for large parking infrastructure, and reducing the problems of traditional parking systems.

- Efficient Space Utilization APS achieves the optimal space allocation by stacking vehicles vertically or horizontally in compact, automated facilities. This not only saves valuable urban land but also makes cities look better.
- Time Efficiency: APS significantly reduces the time people spend looking for parking. Drivers can drop off their cars at a designated spot, and the system autonomously parks each car. When it's time to pick up the car, it's ready and waiting at the same spot.
- Environmental Benefits: With reduced traffic congestion and circling, APS contributes to lower emissions, reduced fuel consumption, and improved air quality in urban areas.

The Role of Industrial SBCs:

Modern parking automation systems rely on industrial SBCs. These compact computer platforms are perfect for handling the complex tasks involved in parking management, equipped with highperformance CPUs, high-capacity storages, and a variety of networking options. SBCs facilitate the integration of numerous subsystems inside a parking infrastructure, from real-time data processing and vehicle detection to payment processing and user authentication.

These systems use sensor technologies for real-time data collection on available parking spaces, vehicle movements, and user identification. These innovations consist of RFID readers, cameras, and ultrasonic sensors.
The Industrial SBCs process this data after collection enabling intelligent decisions such as guiding vehicles to open areas, controlling entry and exit points, and facilitating cashless transactions.

Benefits and Impacts:
The use of Industrial single-board computers in parking automation systems yield several advantages including:
- Optimum Space Utilization: Precise parking space allocation ensures every square foot is efficiently used, reducing traffic, and minimizing unused space.
- Improved User Experience: Cashless payments and automated entry and exit procedures improve the user experience, reduce wait times, and enhance overall satisfaction.
- Real-Time Data Insights: The SBC-driven systems gather real-time information to enhance operational effectiveness, make informed decisions choices, and predict parking patterns.
- Minimizing Environmental Impact: efficient parking systems decrease vehicle emissions by reducing the time spent to find free parking slots.
- Scalability and Customization: Industrial SBCs flexibility enables parking management systems to adapt to the specific requirements of each site and environment.

The integration of Industrial single-board computers into Automated Parking Systems represents a significant advancement in the dynamic field of urban mobility and automation. These advanced computing technologies empower parking facilities to operate more conveniently, efficiently, and sustainably. By merging parking management with technology, we address space constraints and urban growth, while paving the way for smarter, greener, and more user-centered cities.

The thesis is organized as follows.
After this introduction, the Chapter 2 presents a review of existing literature and research related to parking systems. This section sets the context by highlighting what previous studies have explored and discovered in the field. Chapter 3 focuses on the present state of parking systems. This section discusses the core issues and challenges within the domain, emphasizing the necessity for innovative problem-solving approaches. Chapter 4 presents the proposed architecture for the parking system. Here, we describe the essential components and their functions, offering readers a clear insight into how the system is designed. Chapter 5 describes the criteria for the selection of the ENC-989 motherboard. This section details why this specific motherboard was chosen, its historical context,

and how it aligns with the project's objectives. Chapter 6 introduces the technical aspects of the project. It covers the hardware and software components used, communication standards employed, and various aspects related to parking entry, exit, payment, and data management.

Chapter 7 reports on the main challenges encountered during the project. It discusses issues such as hardware replacement, firmware revisions, language barriers in software development, collaboration challenges, and the intricacies of specific communication protocols.

Chapter 8 presents the outcomes of the project. It explains the purpose and content of various project files and provides test output for evaluation.

The Chapter 9 concludes the thesis summarizing the primary findings and accomplishments. Additionally, it offers insights into potential areas for future work and advancement in the field.

CHAPTER 2: LITERATURE REVIEW

In the paper named “A Unique Automatic Parking System by Using RFID Card and IoT,” the author describes a smart parking system with the proposed architecture as follows: The system focuses on reducing human intervention and establishing smart parking under different conditions. In this scenario, the user will reach the website for a parking search, and by registering or making a login via user ID and password they can reach the website where the parking lot at the nearest or desired location of the user will show with the status of parking [1]. In “Microcontroller-based automatic parking system” it describes. This work proposes a parking algorithm with Two-Turn (TT). TInrd parking space (a) First step on TT model (b) Second step on TT model. The minimum turning radius in a vehicle parking model is represented by the symbol of r. The symbol of a represents the distance between the side of the vehicle and the side of the parking space. The symbol of b represents the distance on the first turn for vehicle parking [2].

In the paper “Smart car parking with monitoring system,” The architecture contains, a mobile control unit consisting of Bluetooth, LCD Display, and GPS these components are interrelated with the car control unit which has several components such as a DC motor, Microcontroller, SONAR, GPS, and Camera via internet connection. The architecture contains a mobile control unit consisting of Bluetooth, LCD Display, and GPS. These components are interrelated with the car control unit which has several components such as a DC motor, Microcontroller, SONAR, GPS, and Camera via an internet connection [3]. “RFID-based automatic car parking system using IoT” states that to manage parking spots, a proposed Internet of Things (IoT)-based car system was developed to show the parking space or spaces. The suggested system may save in a database information about unoccupied spaces and the times that cars enter and depart, both of which are useful for administration. The IR sensor detects and uploads the information to the website using the IOT. It contains the details about the number of blocks available in the parking slot.

The LCD will show the direction to the available free blocks in the parking lot and the buzzer will turn on while the car enters the minimum range from the wall. The project's goal is for everyone to be able to locate parking spots using their mobile phones at any time. The RFID Reader is used to pay the parking fees [4]. “ParkingKS: Parking management system using open automatic license plate recognition”, the paper proposed a system that is implemented to work on Linux 18.04 system and the Rasberian operating system with Raspberry Pi microcontroller. The vehicle parking system works in such a way that when the vehicle passes in front of the entrance, just before the obstacle, the camera automatically recognizes it and records the license plate number, date, and time of entry into the system. Then the collected license plate data is trained. Their system showed 90% accuracy in recognizing characters and approximately 100% in recognizing license plates [5]. This study “An

IoT-based Intelligent system for real-time parking monitoring and automatic Billing" recommended that at present, most outdoor parking lots are paid in the following way: the driver inserts coins, a personal card at the parking meter, or a mobile phone is used for payment. When the time is about to expire, the driver can extend his stay by inserting coins or a personal card at the parking meter. Payby-mobile phone and real-time parking reservation systems and smartphone applications are used for parking lot reservation and payment. Paying for a parking space requires user/driver interaction which is problematic when the user has no coins or mobile phone. The vehicle transceiver device (VTD) consists of a processor, a power management unit, a radio transceiver, an RF wake-up sensor, and a global navigation satellite system (GNSS). The parking occupancy is updated, and the start time is recorded [6].

CHAPTER 3: STATE OF ART

It is crucial to have a thorough grasp of the resources that are already available and how they are used from the perspective of the project before starting. A special research effort has been carried out to achieve this, in which several scholarly publications have been studied.

3.1 IDENTIFYING THE CORE ISSUE:

When compared to embedded microcontroller boards, industrial single-board computers (SBCs) have several benefits in terms of processing speed, connection, and functionality, but there are also drawbacks to consider [7]. However, there are a few issues and difficulties that must be considered along this transition:

1. Complexity and Learning Curve: Industrial SBCs often offer more advanced features and capabilities than microcontrollers. There may be a steeper learning curve for developers accustomed to working with simpler microcontroller-based systems as a result.

2. Software Compatibility: Industrial SBCs could have a different software environment than microcontrollers do. When switching to a new software environment, it could be required to rewrite or alter old code to make it compatible with new libraries, APIs, and operating systems.

3. Cost: Industrial SBCs usually cost more than microcontrollers because of their complex features and capabilities. If cost is a crucial factor, this might be taken into consideration.

4. Real-Time Requirements: Microcontrollers are typically more adept at handling applications that need real-time processing and responsiveness because of their predictable nature. Industrial SBCs might not always provide the same level of real-time performance.

5. Power Consumption: Due to their higher processing capability, industrial SBCs could require more electricity, which could be problematic in situations where power efficiency is crucial. In contrast, many microcontrollers are designed to use less power, making them appropriate for battery-operated devices.

6. Scalability and Customization: Systems based on microcontrollers may be scaled up or down to match specific requirements, but industrial SBCs may have stricter baseline requirements and less room for customization.

7. I/O Compatibility: Microcontrollers usually come with special functions and pins designated for certain input/output tasks. While switching to an SBC, compatibility with existing sensors, actuators, and peripherals must be ensured.

8. Physical Dimensions and Form Factor: Industrial SBCs frequently have a bigger size compared to microcontroller boards. This could be a problem if the application is space constrained.

9. Durability and Reliability: To maintain their lifetime and endurance in tough or industrial situations, industrial SBCs may require special care. Microcontrollers are often built to be reliable, tough, and durable.

10. Integration Issues: Adding an industrial SBC to an existing system may need changing connectors, communication protocols, and interfaces, which will cost more time and money.

11. Support and Maintenance: The microcontroller platform support community may be established. Before converting to an SBC, it is important to get acclimated to a new environment and its resources.

Finally, even though migrating from microcontrollers to industrial SBCs can enhance processing speed, connection, and functionality, it's important to carefully consider the many challenges and disadvantages mentioned above.

3.2 THE IMPERATIVE OF PROBLEM SOLVING:

Moving away from embedded microcontroller boards to industrial single-board computers (SBCs) is necessary and are both challenging and advantageous. Performing the following activities allows to properly fix any issues that may come up during this conversion:

1. Complete Evaluation: Recognize the benefits of the industrial SBC and the reasons why the change is being considered. This could be used to create clear expectations for the transition. Establish the project's needs, limitations, and objectives first.

2. Planning and Research: Thoroughly investigate the industrial SBC choices that are offered. Compare the features offered by each company, the technical specifications, their compatibility with the current setup, and their quality of customer service. Choose an SBC that can handle the demands of the project.

3. Skill Enhancement: If the relocation calls for learning new programming languages, libraries, or operating systems, set aside time for training and skill growth. Online manuals, discussion forums, and how-to instructions are all good resources for learning.

4. Software adaptation or porting: Ensuring that the current codebase is compatible with the new SBC. A section of the code may need to be rewritten, libraries may need to be changed, and compatibility with the SBC's operating system must be ensured.

5. Hardware Compatibility: It might be necessary to update or change certain components to enable seamless integration. ensuring that the sensors, actuators, and other gear, including the new SBC, are all compatible with one another.

6. Testing and Validation: Develop a thorough testing approach to ensure that the relocation does not result in unforeseen issues. Test the system's overall performance as well as its connectivity and I/O capabilities.

8. Integration and Interfaces: Ensure that the SBC interfaces with other components correctly. To conform to the previous configuration, this may include modifying pin assignments, voltage levels, and communication protocols.

9. Real-Time Considerations: If real-time requirements are critical, ascertain whether the new SBC can meet them. Some SBCs have real-time capabilities, whilst others could call for further procedures.

10. Documentation: Record the conversion process in detail, along with any adjustments made to the hardware, software, or configurations. Future research and problem-solving will benefit from the knowledge in this documentation.

11. Risk reduction: Anticipate potential risks and issues and develop backup plans to deal with them. The ability to handle unforeseen issues may be made easier by the proactive attitude.

12. Gradual Implementation: Consider implementing in phases, if at all practicable. Consider integrating the new SBC into the present system gradually to minimize the risks associated with a sudden move.

Keeping in mind that the move from microcontrollers to industrial SBCs is a significant one that requires careful thought, adaptation, and testing. Taking a methodical approach to the conversion and considering every aspect of the project will help to overcome the challenges and successfully integrate the new technology into the current system.

CHAPTER 4: PROPOSED PARKING SYSTEM ARCHITECTURE

Figure 1 The proposed system architecture

4.1 ENTRY AND EXIT ACCESS

There is always an entrance and exit to the parking area. Here, an OCR camera, RFID reader, and LAN Ethernet are all connected. When a car pulls up, the OCR camera snaps a picture of the license plate. A parking system's entry and exit access control is a crucial element that guarantees the quick and secure entry and exit of vehicles. In a parking system, access control for entry and exit functions is as follows:

1. Entry Access:

Vehicle Detection: Cameras, RFID readers, infrared sensors, ultrasonic sensors, and systems that read license plates are a few examples of these devices. As a car gets close to the entrance to the parking facility, it is detected using a variety of detection methods.

User Identification: To grant access, the system must recognize the user. For this recognition, a variety of techniques are employed, including RFID tags, access cards and license plate recognition. Once the person has been identified, the system can also retrieve their information and account data.

Authentication: With this safeguard, only authorized vehicles are permitted. The database is utilized to validate the user's identity to verify their veracity.

Payment Processing: The customer can pay in several ways, including cash, credit/debit cards, and prepaid accounts. In the case of paid parking, the system might begin accepting payments at the door.

Gate Control: The access control system opens the entry gate to let the vehicle in after the user has been authenticated and any required payments have been made. Motorized bollards, gates, or barriers are used for this.

2. Exit Access:
Vehicle Detection: Just like with the entering procedure, vehicle detection devices identify the existence of a vehicle as it approaches the exit point.

User Identification: The system uses the same methods used at entry to identify the user, including license plate recognition, RFID tags and access cards.

Authentication: The user's identity is once more checked for legitimacy against the database.

Processing of payments (if applicable): In facilities with paid parking, the system determines the parking price based on how long the car was parked there. The calculation and processing of any necessary fees.

Gate Control: The access control system opens the exit gate so that the vehicle can depart the building after the user has been authenticated and all required payments have been paid.

Key Considerations:

- Security: By prohibiting unauthorized access and fraudulent actions, access control systems guarantee the security of the establishment.
- Integration: To update user accounts and parking availability, the access control system coupled with the parking management software.

- User Experience: To reduce waiting times and make payment procedures simpler, the system offers a seamless and user-friendly experience for entry and exit.
- Scalability: In the event of system failures or power outages, redundancy measures are taken to prevent disruptions. In times of high traffic, the system can handle numerous vehicles at once without experiencing any delays.
- Data gathering: Access control systems produce information on vehicle entry and exit times, which is useful for examining usage trends and streamlining operations.

Systems for controlling access to entries and exits are essential parts of contemporary parking management systems. Vehicle movement within the parking lot is made simple and secure by utilizing technologies like sensors, identification techniques, and payment processing.

4.2 OCR CAMERA

Optical character recognition is also known as text recognition. An OCR application retrieves and makes use of data from scanned documents, camera photos, and image-only PDFs. OCR software extracts letters from images, converts them into words, and then sentences, allowing access to and alteration of the original text. It also eliminates the necessity for manual data entry.

It is connected to the entry and exit access. The vehicle's license plate number is captured using an OCR camera.

The parking system makes use of an optical character recognition (OCR) camera, a technique that combines image capture and text recognition to automate several procedures related to car identification, access control, and payment inside parking garages. When a vehicle enters or exits a parking garage, OCR cameras take pictures of the license plates on those vehicles. They then extract the alphanumeric characters from the license plate and turn them into text data.

This information can then be used for several purposes in the parking system:

1. Vehicle Identification:

OCR cameras enable automated vehicle identification without the need for manual input. When a vehicle approaches the entry or exit point, the OCR camera captures an image of the license plate, processes it to extract the license plate number, and matches it against a database of authorized vehicles or registered users.[8]

2. Access Control:

Authorized vehicles gain access to the parking facility seamlessly. When an authorized vehicle approaches the entry point, the OCR camera reads the license plate, verifies it against the database, and triggers the access control system to open the gate.

3. Ticketless Parking:

OCR cameras eliminate the need for paper tickets in paid parking facilities. The camera records the license plate when a car pulls out, connects it to the time of entry, and works out the parking cost based on how long the car was parked there. Users can then pay the fee at an automated pay station.

4. Real-Time Monitoring:

OCR cameras continuously monitor the occupancy of parking spaces. By analyzing license plates of vehicles entering and exiting, the system determines the availability of parking spaces and provides real-time information to users about vacant spots.

5. Security and Surveillance:

OCR cameras contribute to security by monitoring vehicles entering and exiting the facility. If there's a security concern, such as a vehicle involved in a crime, the system can alert security personnel or authorities.

6. Data Analytics and Reporting:

OCR cameras generate data about the flow of vehicles, entry/exit times, and duration of stay. This data is analyzed to gain insights into parking facility usage patterns, peak hours, and overall occupancy rates.

7. Integration with Payment Systems:

In ticketless parking systems, OCR cameras are integrated with payment gateways, allowing users to pay parking fees through online platforms beforehand.

Benefits:

- Efficiency: OCR cameras automate vehicle identification and access control processes, reducing the need for manual intervention and streamlining operations.
- Accuracy: OCR technology accurately reads license plates, minimizing errors in vehicle identification.
- Convenience: Users benefit from seamless access and exit experiences without the need for physical tickets. ● Security: The system can detect unauthorized vehicles and enhance overall security within the parking facility.
- Data Insights: The collected data provide valuable insights for optimizing parking facility management and user experience.

It's important to consider factors such as lighting conditions, camera placement, and software accuracy to ensure optimal performance and reliability of the OCR camera to benefit the most.

4.3 LAN ETHERNET

An Ethernet LAN is a group of components that enables users to access applications and information, share resources, and connect to other networks. An Ethernet LAN often comprises network printers, user devices such as LAN switches, hubs, and firewalls, as well as multiple media types like coaxial, UTP, and STP, and user devices like computers, PCs, and servers. The company or organization that creates the Ethernet LAN is usually the owner of these components.

To enable connection, data sharing, and control between various system components, parking systems frequently use the local area network (LAN) Ethernet networking technology. Ethernet enables the connection of devices, such as entry/exit gates, automatic pay stations, cameras, access control systems, and management servers, creating an integrated ecosystem that ensures the smooth operation of the parking facility. Here's how LAN Ethernet is used in a parking system:

1. Network Infrastructure:

LAN Ethernet provides the underlying network infrastructure for the parking system. Ethernet switches are used to connect different devices throughout the facility. These switches enable devices to communicate, share data, and send control commands.

2. Device Connectivity:

Various components of the parking system are connected to the LAN Ethernet network. These components can include entry and exit gates, automatic pay stations, license plate recognition cameras, sensors, access control systems, and management servers.

3. Data Exchange:

LAN Ethernet enables real-time data exchange between different devices. For example, when a vehicle approaches the entry gate, a license plate recognition camera captures the license plate image, and the data is sent over the LAN to the access control system for validation and gate control.

4. Access Control:

Systems for access control are linked to the LAN Ethernet network. Through the entrance gate, authorized cars are permitted access. Based on information from a variety of sources, such as license plate recognition or RFID tags, they determine the authenticity of vehicles trying to enter the site.

5. Payment Processing:

Automatic pay stations and terminals are connected to the LAN Ethernet network. They communicate with the central management system to calculate parking fees based on entry and exit times. Users make payments through these terminals, and the information is transmitted over the network for processing.

6. Real-Time Monitoring:

Sensors and cameras placed within the parking facility are connected to the LAN Ethernet network. They continuously provide real-time data about occupancy, vehicle movement, and any security concerns. This data is used for monitoring and optimization.

7. Centralized Management:

Central management servers are an integral part of parking systems. They control and monitor the entire facility, gather data from different components, manage access permissions, generate reports, and provide remote access for administrators.

8. Remote Management:

To resolve problems, update software, and alter configurations, authorized staff have access to the system remotely. The parking system is remotely managed and watched over thanks to LAN Ethernet.

Benefits:

- Reliability: LAN Ethernet is a reliable and established networking technology that offers stable connections for critical components of the parking system.
- High Speed: High data transfer rates offered by Ethernet enable quick data interchange and real-time communication between devices.
- Scalability: The Ethernet network is easily scaled to accommodate the addition of new devices or the expansion of the parking facility.
- Centralized Control: Centralized management through Ethernet-connected servers enables efficient monitoring, reporting, and control of the entire parking system.
- Integration: Ethernet facilitates the integration of various components, ensuring seamless cooperation among devices from different manufacturers.

Ethernet is a cornerstone of modern parking systems, enabling efficient data flow, communication, and control. Its robustness and flexibility contribute to the reliable and streamlined operation of parking facilities.

Switching:
A network's key is addressing and switching - the ability to send data between nodes on the same network. The ability to deliver traffic between two nodes via different network paths is one of the most significant characteristics of Ethernet networks and switching.

4.4 LOCAL CONTROL

One physical network has several logical networks thanks to local control. Geographical location, elapsed time, upward and downward link capacity, and delay can all be used to form logical networks. Non-public networks provide predetermined performance templates for the organization to use. Like how front desk staff members grant permission to enter buildings for guests, their staff members, subcontractors, etc. It works the same to distribute resources to the users through the Local Control interface by the company.

Local control offers organizations authority over the network. When it comes to public safety, this is a compulsory requirement. However, it also offers utility companies considerable cost-saving options, and it is used to substitute wireless internet connections as well as fixed ethernet wiring in industrial and building automation.

Local control in a parking system refers to the management and operation of various components within the parking facility at a specific location or on-site. It involves the usage of hardware, software, and devices that are directly installed and managed within the facility itself. Local control is essential for ensuring the smooth operation of the parking system, managing vehicle flow, and security, and providing a seamless user experience. Here's how local control works in a parking system:

1. Entry and Exit Gates:
Local control includes the management of entry and exit gates. These gates are equipped with sensors, access control systems, and mechanisms for vehicle entry and exit. Local control ensures that only authorized vehicles can access the facility, and it manages the opening and closing of gates in response to user interactions and real-time data.

2. Payment and Terminals:

Payments are positioned within the parking facility to allow users to make payments for parking. Local control manages these automatic pay stations, processing payment transactions, calculating parking fees, and issuing receipts.

3. Ticket Dispensers and Validation:

In parking facilities that use paper tickets, local control manages ticket dispensers and validation units. These devices issue entry tickets and validate them upon exit to calculate parking fees.

4. Sensors and Cameras:

Local control manages sensors and cameras installed within the facility to monitor occupancy, vehicle movement, and security. Sensors detect the presence of vehicles in parking spaces, while cameras capture license plate images for identification and access control.

5. Access Control Systems:

Access control systems are responsible for verifying the legitimacy of vehicles attempting to enter the facility. Local control manages these systems, validating users' identities through methods such as RFID tags, access cards, or license plate recognition.

6. Emergency Response:

In case of emergencies or security incidents, local control can initiate specific actions, such as triggering alarms, locking down certain areas, or notifying security personnel.

7. Data Processing and Storage:

Local control processes and stores data collected from various devices within the facility. This data includes occupancy information, entry/exit times, payment transactions, and security events.

8. User Interactions:

Local control provides interfaces for user interactions, such as touchscreens on automatic pay stations or entry gates. Users can input information, make selections, and follow prompts provided by these interfaces.

Benefits:

- Reduced Latency: By processing data and commands locally, the system operates with minimal delay, improving user experience and efficiency.

- Customization: Facility managers customize local control settings to align with specific facility requirements, traffic patterns, and user needs.
- Real-Time Control: Local control enables immediate response to events within the parking facility, ensuring smooth traffic flow and enhanced security.
- Robustness: Local control systems are designed to operate reliably even in challenging conditions, contributing to the overall stability of the parking system.
- Autonomous Operation: Local control allows parking systems to function autonomously, even in cases where external communication might be disrupted.
- Data Privacy: Some data processing occurs locally, reducing the need to transmit sensitive information over external networks.

Overall, local control plays a crucial role in ensuring the efficient operation, security, and user satisfaction within a parking facility. It enables real-time decision-making, local management, and a seamless experience for both facility operators and users.

4.5 AUTOMATIC PAY STATION

An Automatic pay station is a machine made up of different components. It serves the purpose of autonomous parking fee collecting. The automatic pay station's modular design enables the appropriate integration of all necessary features. In addition to accepting cash payments, the automatic pay station also accepts payments made using credit cards, city cards, and discount coupons, along with a variety of other payment methods. All users can use this device and the entire parking system easily thanks to a combination of high-quality processing, innovative functional design, high-level security, basic operation, and a wide range of fares.

An Automatic Pay Station (APS) is a key component of modern parking systems that streamlines the payment process for users exiting a parking facility. APSs are designed to provide a convenient and efficient way for users to settle parking fees without requiring manual intervention from attendants. Here's how an Automatic Pay Station works within a parking system:

1. User Interaction:

As a vehicle approaches the exit point of the parking facility, the driver stops at the Automatic Pay Station. The APS is equipped with user-friendly interfaces, such as touchscreens and payment terminals, that guide users through the payment process.

2. Fee Calculation:

The APS calculates the parking fee based on the duration the vehicle was parked within the facility. This calculation is typically done by comparing the entry time recorded during the vehicle's entry to the exit time.

3. Payment Options:

Automatic Pay Stations offer various payment options to users. These options may include cash payments, credit/debit card payments, and contactless payment methods such as NFC (Near Field Communication).

4. Display and Confirmation:

The APS displays the calculated parking fee to the user, along with the available payment methods. Users can review the information, select their preferred payment method, and confirm the payment.

5. Payment Processing:

Once the user selects a payment method, the APS processes the payment transaction securely. If the user chooses a cash payment, they can insert the required amount into the machine. For electronic payments, users can swipe, tap, or insert their payment card into the terminal.

6. Printing Receipts:

Upon successful payment, the APS prints a receipt for the user as proof of payment. This receipt includes details such as the parking facility name, entry and exit times, parking duration, and the amount paid.

7. Gate Opening:

After the payment is completed and a receipt is issued, the APS sends a signal to the exit gate to open it, allowing the user to exit the parking facility.[9]

8. Data Integration:

The APS is typically connected to the parking management system via a network connection. This integration ensures that payment information is recorded and updated in real time within the central system.

Benefits:

- Efficiency: APSs eliminate the need for manual payment processing, reducing wait times at exit points and improving traffic flow.

- Convenience: Users make payments quickly and conveniently without needing to interact with an attendant.
- Accuracy: Automatic fee calculation reduces the risk of human errors in calculating parking fees.
- User Experience: APSs provide a seamless and user-friendly payment experience, enhancing user satisfaction.
- Data Collection: APSs collect payment data that is analyzed to gain insights into revenue generation, payment trends, and facility utilization.
- Reduced Staffing: APSs help reduce the need for staffing attendants at exit points, saving operational costs.
- Security: Secure payment processing ensures the safety of user payment information.

Automatic Pay Stations play a significant role in modernizing parking systems by automating payment processes and enhancing user convenience. They contribute to a more efficient and usercentric parking experience while also providing valuable data for facility management.

4.6 SERVER AND DATABASE

The term "server" refers to any computer running the necessary software, but it is typically used to refer to the biggest, most potent devices that push and pull data from the internet. Most computer networks have one or more servers that carry out certain tasks. It is more likely that numerous servers with different roles play a significant role in a network the more people connect to it or the more data it moves around. The server is the software application that controls a specific operation. However, the sturdy hardware that supports this software is also referred to as a server. This is because server software, which controls a network of hundreds or thousands of users, requires hardware that is more robust than personal computers designed for residential use.

A database is a methodical, structured way to store data that allows for access, examination, modification, updating, and migration (to other databases). Tables with rows and columns generally serve as the structure for databases. In the current environment, daily involvement in a range of database-related tasks, such as those at the bank, railway station, school, grocery shop, etc., is. These are the situations were having quick access to a lot of data in one location is necessary.

In a parking system, servers and databases play crucial roles in managing and storing data related to vehicle entry, exit, payments, user information, and operational status. These components are essential

for ensuring the smooth operation of the parking facility, data integrity, and efficient management. Here's how servers and databases are utilized in a parking system:

1. Server:
Centralized Management: They contain software that controls a variety of system elements, such as monitoring, access control, and payment processing the primary centers of control for the entire parking system are servers.
Data Processing: Servers process data collected from various devices within the parking facility. This includes information from entry and exit gates, automatic pay stations, license plate recognition cameras, and sensors.

User Authentication: Only users with permission are allowed entry into the area Servers authenticate people and vehicles using tools including RFID tags, access cards, and license plate recognition. Payment Processing: Servers process payment transactions initiated at automatic pay stations or online platforms. They calculate parking fees based on entry and exit times and facilitate secure payment processing.

Real-Time Monitoring: Servers receive real-time data from sensors and cameras installed within the parking facility. They monitor occupancy, vehicle movement, security events, and other operational aspects.

Reporting and Analytics: Servers generate reports and analytics based on the collected data. This information can include parking utilization trends, revenue generation, peak hours, and user behavior.

Remote Management: Servers provide remote access for administrators to manage and monitor the parking system from external locations. This enables troubleshooting, configuration changes, and software updates.

2. Database:
Data Storage: Databases store all relevant information about vehicles, users, entry/exit times, payment transactions, and other system activities. This data forms the backbone of the parking management system.
Data Integrity: Databases ensure the integrity of stored data by enforcing data validation rules and maintaining consistent and accurate records.

Data Processing: Databases support operations such as sorting, filtering, and aggregating information. This is useful for generating reports and analytics.

Data Retrieval: The system retrieves data from the database to determine whether a user's identity is accurate and whether they have access privileges as they approach an entry or exit point.

Scalability: As the parking facility grows, databases are scaled to handle increased data volume without compromising performance.

Backup and Recovery: In the event of hardware failures, system breakdowns, or other emergencies, databases frequently incorporate backup and recovery procedures.
Security: Databases implement security measures to protect sensitive user information and payment data. Access controls, encryption, and authentication mechanisms are often employed.

Benefits:

- Centralized Management: Servers provide a centralized control point for the entire parking system, facilitating efficient management and monitoring.
- Efficiency: Automated data processing and retrieval from databases streamline operations, reducing manual effort.
- Accurate Information: Databases ensure accurate and consistent storage of data, minimizing errors and discrepancies.
- Data Insights: Servers and databases enable the generation of valuable insights through data analysis, contributing to informed decision-making.
- Remote Accessibility: Remote management capabilities allow administrators to address issues and perform updates without being physically present at the facility.
- Security: Servers and databases implement security measures to protect user information and system data.

Servers and databases are essential components that enable the effective functioning of a parking system. They provide the foundation for managing user access, processing payments, monitoring operations, and maintaining accurate records.

4.7 INTERNET

The Internet is a widely utilized tool and resource that is familiar to almost everyone on Earth. It functions by connecting an immense number of servers, web pages, and websites. Electronic

communication in the form of emails, images, videos, and messages are utilized to communicate with individuals who are dear to us. The Internet is a vast interlinked network of electronic devices and computers, which establishes a communication avenue for online information sharing and retrieval. The ability to access applications, websites, social media apps, and other services on smartphones is dependent on its connection to the Internet As it presently exists in its current form, the situation is such that the Internet is considered the most effective way of transmitting and receiving information. The integration of the Internet into a parking system presents various prospects for improving efficiency, enhancing user experience, and augmenting management capabilities. The inclusion of the Internet in a parking system enables instantaneous connectivity, exchange of data, and remote control. Here's how the Internet is utilized in a parking system:

1. Remote Management and Monitoring:
The availability of Internet connectivity allows for the remote management and monitoring of parking systems, thereby enabling real-time troubleshooting, updates, and adjustments without requiring physical presence. This feature grants facility managers and administrators the ability to access system data, status, and configurations from any location with Internet access.

2. Cloud-Based Solutions:
Cloud-based platforms allow parking data, applications, and services to be hosted on remote servers. This offers scalability, reduced hardware costs, and easy access to data from various locations. Cloud solutions also enable seamless updates and maintenance.

3. Online Booking and Reservations:
Through the Internet, users can make parking reservations and bookings in advance. They can check availability, select parking spaces, and make payments online, streamlining the entry process and ensuring a reserved spot upon arrival.

4. Mobile Apps and Payments:
Mobile apps connected to the Internet provide users with real-time information about parking availability, locations, and fees. Users can also use mobile apps to pay for parking fees digitally, eliminating the need for cash and enhancing convenience.

5. Real-Time Information:

Internet-connected sensors and cameras provide real-time data about parking occupancy, vehicle movement, and security. This information is displayed on websites, mobile apps, or electronic signage to guide users to available parking spaces.

6. Payment Processing:

Online payment gateways allow users to settle parking fees electronically through secure Internet transactions. Payment information is encrypted, ensuring data security.

7. Data Analytics:

Internet-connected systems collect and analyze parking data to gain insights into usage patterns, peak hours, revenue trends, and user behavior. These insights contribute to effective facility management and decision-making.

8. Remote Access Control:

The Internet enables remote access control, allowing administrators to open gates, grant access, and manage permissions from a remote location. This is particularly useful for situations requiring emergency access or for providing temporary access to specific users.

9. Integration with Navigation Apps:

Parking facilities connected to the Internet are integrated with navigation apps or GPS services, guiding users to the nearest available parking spaces.

Benefits:

- Efficiency: Internet connectivity streamlines processes, reduces manual intervention and enhances overall operational efficiency.
- User Convenience: Online reservations, payments, and real-time information improve user convenience and satisfaction.
- Remote Management: Remote access facilitates effective administration and supervision of the system without requiring any physical presence on the premises.
- Data-Driven Insights: Internet-based systems offer a wealth of information that are scrutinized to derive meaningful insights, consequently facilitating well-informed decisions and enhancements.
- Scalability: Internet-connected solutions are easily scaled to accommodate growing needs and future enhancements.

- Innovation: Internet integration enables the implementation of innovative features, such as mobile apps, electronic payments, and smart parking guidance.

Through the utilization of the vast potential of the World Wide Web, the parking systems possess the capability to metamorphose into astute, easily accessible, and effective infrastructures that cater to the demands of contemporary metropolitan settings.

4.8 REMOTE CONTROL

A remote computer is not physically accessible to a user but is accessed remotely through a network connection established with another computer. The network facilitates the formation of remote connections.

The use of remote access software allows for controlling a remote computer from a distance.

Examples include TeamViewer, VNC, and Remote Desktop.

Clients and servers are crucial for the proper functioning of the internet [10].A client is directly connected to the internet, while servers indirectly connect and host websites. ISPs provide internet access to servers with IP addresses.

The remote control in the parking system enables monitoring, management, and control of components and operations from a distant location using communication technologies such as the Internet, mobile networks, or dedicated control systems. It improves operational efficiency, allows for real-time adjustments, and facilitates effective management. Here's how remote control is implemented in a parking system:

1. Entry and Exit Gates:

Remote control allows administrators to open and close entry and exit gates from a remote location. This is useful for granting emergency access, managing special situations, or remotely assisting users.

2. Access Control:

Administrators remotely manage access control systems, granting or revoking access permissions for specific users or vehicles. Remote control ensures that only authorized vehicles can enter the facility.

3. Automatic Pay Station:

Remote control enables administrators to monitor automatic pay stations, update software, perform diagnostics, and resolve issues without physically visiting the kiosks.

4. Monitoring Sensors and Cameras:

Sensors and cameras within the parking facility are remotely monitored to ensure proper functioning, detect anomalies, and address any security concerns in real time.

5. Troubleshooting:

Remote control allows administrators to identify and troubleshoot problems within the parking system, reducing downtime and minimizing disruptions.

6. Software Updates:

Parking systems often require software updates for improvements, bug fixes, and security patches. Remote control facilitates the distribution and installation of these updates across the system.

7. Configuration Changes:

Administrators remotely adjust system configurations, settings, and parameters to optimize operations, adapt to changing requirements, or address specific needs.

8. Emergency Situations:

In emergencies, such as a system malfunction or security breach, administrators take immediate control remotely to ensure the safety of users and property.

9. User Assistance:

Administrators provide remote assistance to users who encounter issues at entry or exit points, guiding them through the process or remotely opening gates if needed.

Benefits:

- Efficiency: Remote control eliminates the need for physical presence at the facility for routine tasks, saving time and resources.
- Real-Time Responsiveness: Administrators can promptly address problems, resulting in less downtime and improved user satisfaction.
- Cost Savings: Remote control reduces the need for on-site staff, lowering operational costs.
- Flexibility: Administrators can manage the system remotely. This provides flexibility and convenience.
- Timely Updates: Remote control ensures that software updates and patches are quickly applied across the system to maintain security and functionality.
- Enhanced Security: Remote control allows for swift responses to security incidents or breaches, enhancing overall facility security.

- Optimization: Remote control enables administrators to make timely adjustments and optimizations based on real-time data and changing conditions.

To introduce remote control in a parking system, it is crucial to use secure communication protocols, dependable remote access methods, and correct authentication mechanisms to guarantee the system's security and dependability.

CHAPTER 5: ENC-989 EMBEDDED MOTHERBOARD AND PRODUCT ARCHITECTURE

Embedded Motherboard:

An embedded motherboard is a key component in embedded systems, which are specialized computer systems designed for specific applications. These motherboards are typically designed to be compact, energy-efficient, and have specific interfaces and features tailored to the needs of the target application.

Product Architecture:

The product architecture of an embedded system, which includes an embedded motherboard like the ENC-989, outlines how the various components and modules are integrated to create a functional system. The Pico-ITX form factor is an ultra-compact standard for embedded motherboards. It's even smaller than the Mini-ITX form factor and is designed for applications where space is at a premium. Despite its small size, Pico-ITX boards still feature a range of capabilities. Here are some components that are part of the embedded system's product architecture:

1. Processor: The Intel Elkhart Lake Atom x6000E is a series of low-power system-on-chip (SoC) processors designed by Intel for use in embedded and IoT (Internet of Things) applications. These processors are part of the Intel Atom family and are based on the Tremont microarchitecture. Elkhart Lake Atom x6000E processors were introduced by Intel in late 2020.

2. Memory: It has DDR4 SODIMM memory DDR4 SODIMM (Double Data Rate 4 Small Outline Dual Inline Memory Module) which is a type of computer memory module commonly used in laptops, small form factor PCs, and some desktop computers. It is a variant of DDR4 memory designed for use in smaller and more compact systems.

3. I/O Interfaces: Embedded systems often have a range of input/output interfaces tailored to the application. This comprises GPIO (general-purpose input/output) pins, Ethernet ports, serial ports, USB ports, audio interfaces, and more.

4. Connectivity: the motherboard includes built-in Wi-Fi, Bluetooth, cellular connectivity, or other wireless communication options.

5. Power Management: For embedded systems to maximize energy utilization and increase system longevity, effective power management is necessary. This could involve power regulators, sleep modes, and other power-saving features.

6. Expansion Slots: These include PCIe or other expansion slots for adding specialized cards, such as graphics cards, network adapters, or additional I/O interfaces.

7. Peripherals: The design offers support for monitors (HDMI, DisplayPort, LVDS), touchscreens, cameras, and other peripherals, depending on the application.

8. Operating System: The operating system that the embedded system will use is specified by the architecture. This be a custom-built OS, a real-time operating system (RTOS), or another suitable option.

9. Enclosure and Cooling: The design also considers any essential cooling techniques, how the motherboard and other components are placed in an enclosure, and other factors.

5.1 WHY ENC-989?

Using an embedded motherboard in the Pico-ITX form factor offers several advantages for certain applications and scenarios. Here are some reasons to consider using an embedded Pico-ITX motherboard:

1. Size and Space Constraints:

Pico-ITX motherboards are incredibly compact, with dimensions typically around 100mm x 72mm. This small size makes them ideal for applications where space is limited, such as in small devices, IoT (Internet of Things) devices, robotics, portable systems, and embedded industrial applications.

2. Energy Efficiency:

Pico-ITX motherboards often feature low-power processors and components, making them suitable for battery-powered or energy-efficient applications. This is particularly important in scenarios where power consumption needs to be minimized to prolong battery life or reduce operating costs.

3. Portability:

The small size of Pico-ITX motherboards lends itself to portable devices and systems. These motherboards are used in handheld devices, portable medical equipment, small drones, and more.

4. Customization:

Pico-ITX motherboards are often used in custom-built embedded systems where designers need to tailor the hardware to specific requirements. Their small form factor allows for more flexibility in designing unique solutions.

5. Integration:

Due to their compact size, Pico-ITX motherboards are integrated into other products or enclosures seamlessly. They are hidden within devices while still providing the necessary computational power.

6. Connectivity:

Despite their size, Pico-ITX motherboards typically offer a range of connectivity options, including USB ports, Ethernet, wireless connectivity, and more. This enables easy integration with various peripherals and network environments.

7. IoT Applications:

Pico-ITX motherboards are well-suited for IoT applications where small size, connectivity, and energy efficiency are critical. They are used in smart home devices, industrial sensors, and other IoT solutions.

8. Industrial and Embedded Applications:

Pico-ITX motherboards find applications in industries such as manufacturing, automation, surveillance, and transportation. They can power industrial control systems, data acquisition devices, and embedded computing solutions.

9. Cost Efficiency:

For projects where cost-effectiveness is a priority, Pico-ITX motherboards offer a balance between size, features, and price.

10. Compact Media Solutions:

Pico-ITX motherboards are used in media centers, kiosks, and digital signage solutions where space is limited, but multimedia capabilities are required.

5.2 HISTORY AND MOTIVATION

The Pico-ITX form factor was introduced by VIA Technologies, a Taiwanese company known for its contributions to the development of embedded and small form factor computing solutions. The

motivation behind creating the Pico-ITX motherboard was to provide an even smaller and more energy-efficient alternative to the existing small form factor options, such as Mini-ITX and NanoITX. Pico-ITX was first announced in 2007 as a response to the growing demand for compact computing solutions in various industries and applications. Here's a brief history and the motivations behind the Pico-ITX motherboard:

History:
2007: VIA Technologies introduced the Pico-ITX form factor in 2007. The form factor aimed to push the boundaries of size reduction while maintaining a reasonable level of processing power and connectivity.

Motivations:

1. Size Reduction: The primary motivation behind Pico-ITX was to create an even smaller form factor compared to Mini-ITX and Nano-ITX. This form factor was designed for applications where space constraints were paramount, allowing for more creative and compact designs.
2. Energy Efficiency: As the demand for energy-efficient devices increased, Pico-ITX motherboards were designed to feature low-power components. This made them suitable for battery-powered devices and applications where power consumption needed to be minimized.
3. Portability: The rise of portable devices, including handhelds, portable media players, and portable medical equipment, prompted the need for compact computing solutions that could be easily integrated into these devices.
4. IoT and Embedded Systems: With the growth of the Internet of Things (IoT) and embedded computing, there was a need for small, energy-efficient computing platforms that could power various IoT devices, sensors, and industrial applications.
5. Customization: Pico-ITX motherboards provided designers with the flexibility to create custom solutions tailored to specific requirements. Their small size allowed for unique and specialized applications that couldn't be achieved with larger form factors.
6. Cost-Effective Solutions: Pico-ITX motherboards were designed to offer a balance between size, features, and cost. They provided an economical option for applications that required moderate processing power and connectivity in a compact package.
7. Growing Industrial and Commercial Applications: Industries such as manufacturing, automation, retail, and transportation were seeking smaller computing solutions that could be integrated into various equipment and systems.
8. Innovation: The introduction of Pico-ITX spurred innovation in various industries. It led to the development of new products and solutions that were previously constrained by size limitations.

The introduction of the Pico-ITX form factor opened new possibilities for developers, engineers, and manufacturers looking to create compact, energy-efficient devices across various industries. Its motivations align with the ongoing trend of miniaturization, energy efficiency, and the need for flexible computing solutions for modern applications.

5.3 GOALS AND REQUIREMENT

The use of a Pico-ITX motherboard in a parking system serves specific goals and requirements tailored to the unique demands of the system. Here are the goals and requirements of a Pico-ITX motherboard in a parking system:

Goal:

The primary goal of incorporating a Pico-ITX motherboard in a parking system is to provide an efficient, compact, and cost-effective computing platform that can manage and optimize various aspects of the parking facility. This includes access control, payment processing, data management, and user interfaces. The motherboard should enable the parking system to operate smoothly while maximizing space utilization within the control equipment enclosure.

Requirements:

1. Compact Form Factor: The Pico-ITX motherboard adheres to a compact form factor to fit within the limited space available in the control equipment enclosure. Its small size allows for efficient use of space while accommodating other control components.
2. Energy Efficiency: Given that parking systems operate continuously; energy efficiency is crucial. The motherboard features energy-efficient components to minimize power consumption and reduce operating costs over time.
3. Processing Power: While compact, the motherboard provides sufficient processing power to handle real-time access control, payment processing, data analysis, and communication with other components within the parking system. The choice of processor aligns with the system's requirements.
4. Connectivity: The Pico-ITX motherboard offers a range of connectivity options, including Ethernet ports, USB interfaces, and potentially wireless connectivity for communication with entry/exit gates, automatic pay stations, sensors, cameras, and central management systems.
5. I/O Interfaces: Parking systems require a variety of I/O interfaces, such as serial ports for communication with gate controllers, GPIO pins for sensor integration, and video outputs for user interfaces or monitoring screens.

6. Memory: Sufficient memory capacity (RAM and storage) available to support the operating system, applications, and data storage requirements of the parking system.
7. Reliability: The motherboard is designed for reliability and longevity, as parking systems often operate 24/7 and are subject to harsh environmental conditions.
8. Operating System Compatibility: The motherboard is compatible with the chosen operating system, which may include a customized solution or a commercial operating system suitable for embedded applications.
9. Customization: Flexibility for customization is important to adapt the motherboard to the specific needs of the parking system, such as integrating license plate recognition systems or supporting multiple payment methods.
10. Security: Robust security features are essential to protect sensitive data, prevent unauthorized access, and ensure the integrity of payment transactions within the parking system.
11. Cost-Effectiveness: The motherboard balances features and performance with cost-effectiveness to meet the budget constraints of the parking system project.
12. Remote Management: The motherboard supports remote management and monitoring capabilities, allowing administrators to access and control the system from a central location for maintenance, troubleshooting, and updates.
13. Scalability: The motherboard is scalable to accommodate future upgrades or expansions of the parking system as demand grows.

In summary, the use of a Pico-ITX motherboard in a parking system aims to provide a compact, energy-efficient, and cost-effective computing solution that can reliably manage the various functions of the system while fitting within the constraints of the control equipment enclosure. These requirements ensure the efficient operation and management of the parking facility.

5.4 FEATURES

- Intel ® Elkhart Lake Atom x6000E
- DDR4 SODIMM memory
- Support 4G/5G, support HDMI, LVDS display
- 4*GbE LAN，3*USB3.0，2*USB2.0
- 6*COM，12bit GPIO
- 1*SATA socket，1*MSATA socket
- DC 12V-19V input

5.5 SPECIFICATIONS

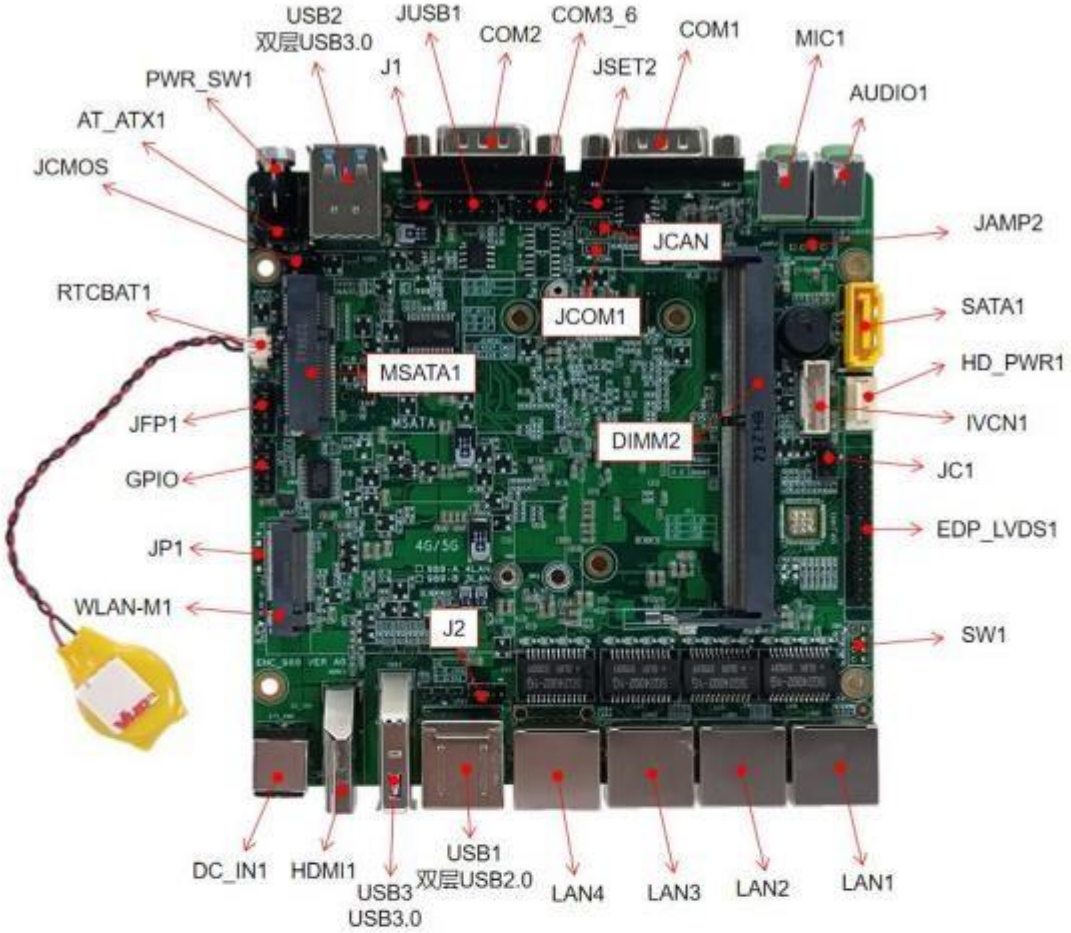
Specification		
Processor	CPU	Intel® Elkhart Lake Atom x6000E
	CPU Package	Onboard
	BIOS	System BIOS: SPI BIOS，standard CMOS 65mah battery
Memory	Architecture	DDR4 SODIMM
	Socket	SODIMM (memory-chip）
Display	Graphic Controller	Integrated graphic media accelerator
	HDMI	HDMI，Max. resolution: 4K
	LVDS	LVDS
LAN	Controller	LAN1: Realtek RTL8111 GbE LAN，RJ45
		LAN2: Realtek RTL8111 GbE LAN，RJ45
		LAN3: Realtek RTL8111 GbE LAN，RJ45
		LAN4: Realtek RTL8111 GbE LAN，RJ45
Audio	AUDIO	MIC/earphone
H/W Monitor	Watchdog Timer	0-255s,provide Watchdog program
	Temp. Monitor	Support CPU/Motherboard/HDD temp. monitoring
I/O	USB3.0	3*USB3.0
	USB2.0	2*USB2.0, USB2.0 / 1.1，Max +5V@1A
	Serial Port	COM1: 3-wire RS232 TX/RX signal
		COM2: 3-wire RS232/485 signal
		COM3-COM6: 3-wire RS232, Internal pin header
	GPIO	12bit, provide program, User-defined I/O,3.3V/5V@24mA level
Expansion Bus	M2 Key-B	M2 Key-B ,support USB 4G/5G communication module
		1*USIM slot
Storage	SATA	standard SATA socket，with SATA HDD power socket
	MSATA	1*MSATA
Power	Power Type	AT/ATX power mode optional,DC12-19V input,UP6182, support power reverse and over-voltage protection
Button	Power switch	Switch with LED
Environment	Operating Temp.	-20° C ~ 70° C
	Storage Temp.	-40° C ~ 80° C
	Operating Humidity	0% ~ 90% relative humidity,non-condensing
	Storage Humidity	0% ~ 90% relative humidity,non-condensing

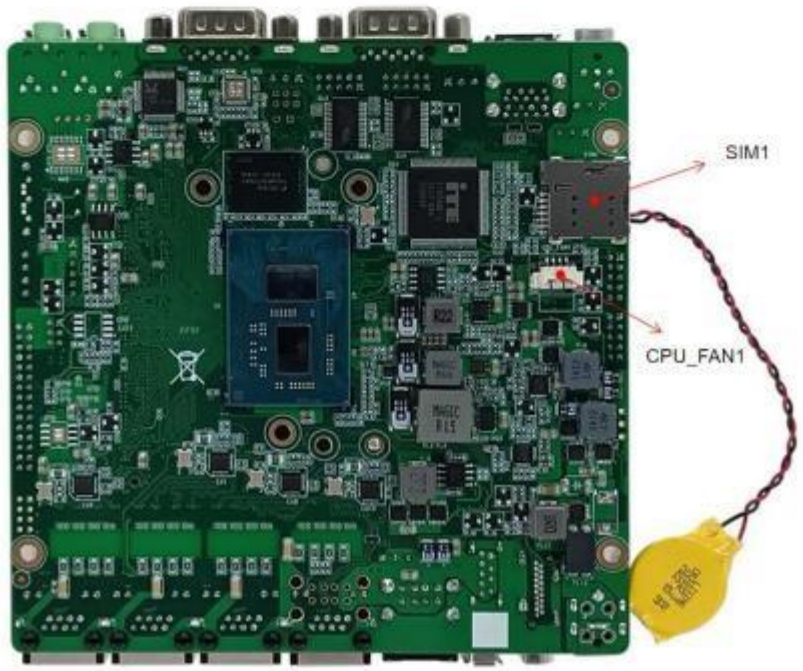
Formfactor	Dimensions	120mm × 120mm
	Weight	-
Certificate		FCC CLASS B, CE, RoHS

Figure 2 ENC-989 Specification

: -

Figure 3 ENC 989





: -

Figure 4 ENC 989 I/O

CHAPTER 6: EXECUTION OF PROPOSED SYSTEM

6.1 COMPONENTS USED FOR THE PROPOSED SYSTEM

6.1.1 HARDWARE

The devices to be interfaced on the inputs and outputs are the following:

- ENC-989 Embedded Motherboard
- RFID reader (RS232/USB)
- LCD display (RS232/I2C)
- MR004 Board
- 1D/2D Barcode reader (RS232//USB HID)
- Keyboard (USB HID)

6.1.2 SOFTWARE

- Visual Studio Community 2022

Visual Studio Community 2022 is a popular integrated development environment (IDE) developed by Microsoft. It provides a comprehensive set of tools and features for software development, allowing developers to create a wide range of applications, including desktop, web, mobile, cloud, and more [11].

- Hercules

The "Hercules SETUP Utility" is a software tool designed to assist with the configuration and testing of serial communication connections on Windows computers. It provides a user-friendly interface for setting up and testing various serial communication parameters and devices. When serial communication is necessary, such as when establishing and testing equipment linked through RS232 or RS485 interfaces, the application is frequently utilized.

Hercules SETUP Utility is a handy tool for anyone working with serial communication, whether for testing devices, diagnosing communication issues, or configuring serial ports. Keep in mind that while Hercules SETUP Utility is a widely used and trusted tool, it's essential to download it from a reliable source to ensure that you're getting the official and safe version of the software.

- Virtual Serial and Device Management- VirCom

Virtual serial port generation and device management on a computer are made easier by the flexible software program VirCom. It acts as a link between the physical and virtual components, enabling smooth connection and management of a range of hardware devices, including modems, sensors, printers, and more. With its remarkable interoperability, user-friendly interface, and variety of features that are suitable for both professionals and hobbyists, VirCom distinguishes out from the competition.

- SQLyog

SQLyog is a popular and powerful database management tool for MySQL databases. It provides a graphical user interface (GUI) that allows users to interact with and manage MySQL databases in a user-friendly and efficient manner. SQLyog is widely used by developers, database administrators, and other professionals who work with MySQL databases.

SQLyog provides a comprehensive set of tools to streamline database management tasks, optimize performance, and simplify the interaction with MySQL databases. It's available in both free and paid editions, with the paid editions offering advanced features and capabilities. Whether you're a developer, database administrator, or data analyst, SQLyog can help you work efficiently with MySQL databases.

The functional modules to be created for access management are:

- Bus management (digital I/O)
- Transit registration (DBMS)
- Send data to the server (TCP/IP, RS485)

6.2 SERIAL COMMUNICATION STANDARDS

The term "communication" is often used to describe the transfer of information across two or more media. Data transmission between electronic components or devices using a serial interface is known as serial communication. It involves sequentially transmitting data bits at a time across a single communication channel. Microcontrollers, sensors, computer peripherals, and long-distance device connections are just a few examples of the many applications where serial communication is frequently employed [12]. In embedded systems, communication refers to the transmission of bits of data between two

microcontrollers. Microcontrollers communicate data bits according to a set of established guidelines called communication protocols. The protocol for communication is called a serial communication protocol if the data is transferred in series, or one after the other. More specifically, in serial communication, data bits are sent over the data bus or communication channel one at a time in a sequential way.

Digital technology supports a variety of data transfer methods, including serial and parallel communication. Like this, there are two categories of protocols: parallel communication protocols and serial communication protocols. The Parallel Communication Protocols ISA, ATA, SCSI, PCI, and IEEE-488 are a few examples. The same is true for serial communication protocols, which include CAN, ETHERNET, I2C, SPI, RS232, USB, 1Wire, and SATA among others.
Serial communication protocols are divided into two categories: synchronous and asynchronous.

1. Asynchronous Serial Communication: Asynchronous communication between transmitting and receiving devices does not rely on a constant clock signal. Instead, it marks the start and end of each data packet using start and stop bits. Both devices must have the same baud rate (data transmission speed) set. RS-232, UART (Universal Asynchronous Receiver-Transmitter), and USB are a few examples of asynchronous protocols.

2. Synchronous Serial Communication: To keep the devices' timing in sync, synchronous communication uses a common clock signal. As a result, data communication is made quicker and more trustworthy. SPI (Serial Peripheral Interface) and I2C (Inter-Integrated Circuit) are two popular synchronous protocols.

SERIAL COMMUNICATION- TRANSMISSION MODE

Simplex Method:
In the simplex approach, either the sender or the receiver may be active at any given time. Therefore, if the sender is sending the data, the recipient can only accept it, and the opposite is true. Therefore, the simplex approach is a one-way communication mechanism. Radio and television are two popular simplex approach examples.

Half Duplex Method:
Both the sender and the receiver may be active simultaneously in the simplex approach. In other words, if the sender is sending the data, the receiver can only accept it, and vice versa. In this sense, the simplex approach is a one-way communication mechanism. Television and radio are two prominent instances of the simplex approach.

Full Duplex Method:

Full duplex transmission allows for simultaneous data transmission between the transmitter and receiver. The widely used illustration is the cell phone. In addition to this, the clock is a crucial component of effective data transmission and one of the main sources. A broken clock causes unexpected data transmission and, occasionally, data loss. So, while employing serial communication, clock synchronization becomes crucial.

6.2.1 RS232

Just two communication wires—one for transferring data and the other for receiving it—make up an RS232 serial bus. Therefore, the receiver (RX) and transmitter (TX) serial pins on serial devices are present. A ground connection needs to be created between RS-232-compatible devices since the communication data depends on the voltage on the wires in relation to ground level. An RS232 wiring diagram is shown below: Because there is just one data line going in each way, RS232 data is transmitted serially, sending each bit one after the other. For the receiver to synchronize itself with the incoming data, this method of data transfer also necessitates that it understands when the actual data bits are arriving. As a synchronization start bit, a logic 0 is transmitted to accomplish this.

6.2.2 RS485

An electrical-only standard, RS-485. RS-485 simply specifies the electrical properties of drivers and receivers that could be used to build a balanced multipoint transmission line, as opposed to comprehensive interfacing standards, which define the functional, mechanical, and electrical specifications.

However, this standard is meant to serve as a reference for higher-level standards. For instance, DL/T645, establishes the protocol for communication for electronic energy meters in China and designates RS-485 as the physical layer standard. The following are the main characteristics of RS-485:

- balanced interface
- multi-point functioning from a single 5-V supply
- -7-V to +12-V bus common-mode range
- maximum data rate of 10 Mbps (at 40 feet)
- up to 32-unit loads
- 4000-foot cable maximum (at 100 kbps)

6.3 PARKING ENTRY AND EXIT

6.3.1 RFID READER FOR SUBSCRIPTION USER ID

An RFID (Radio-Frequency Identification) reader is a device that can communicate with RFID tags to read and sometimes write data to them. RFID technology is widely used in various applications, including access control, inventory management, supply chain tracking, and payment systems. Here's a brief overview of an RFID reader with RS232/USB interfaces:

RFID Reader:

1. Function:

An RFID reader is used to communicate with RFID tags or transponders wirelessly. When an RFID tag comes into the proximity of the reader, the reader emits radio waves that power the tag and enable it to transmit data back to the reader.

2. RFID Tags:

RFID tags are small devices that contain an integrated circuit and an antenna. They can store data, which are read or written by the RFID reader.

Interfaces: RS232/USB:

1. RS232 (Serial) Interface:

- Purpose: The RS232 interface is a common serial communication standard used for data transmission between devices for users who have a subscription to the parking lot.
- Usage: An RFID reader with an RS232 interface is connected to the SBC using a serial cable. This allows the user data to exchange between the reader and the connected device.

2. USB Interface:

- Purpose: The USB interface is a widely used standard for connecting peripherals to computers and other devices.
- Usage: An RFID reader with a USB interface is directly connected to a computer or other USB-enabled devices without the need for additional adapters. It simplifies the setup and provides plug-and-play functionality.

The RFID badge, an essential tool for users with subscriptions to the parking system, serves multiple purposes that streamline their parking experience. With a profound understanding of its workings:

1. Access Control: The primary purpose of the RFID badge is to grant access to the parking facility. Instead of traditional paper tickets or manual verification, users hold their RFID badge near the designated reader. This contactless technology ensures a swift and effortless entry.

2. Subscription Validation: The RFID badge is intricately linked to users parking subscription. Each badge is uniquely encoded with his subscription information, such as the duration of access and other relevant data. When John approaches the entry point, the RFID reader quickly communicates with the badge, validating his subscription status.

3. Enhanced Security: The RFID badge not only provides convenience but also strengthens security. Since it's exclusively tied to the user's subscription, the risk of unauthorized entry is significantly reduced. This technology enhances overall safety within the parking facility.

4. Seamless Exit: Upon departure, users can once again use the RFID badge to exit the parking area. The system, recognizing his badge, allows him to exit without any hindrance, making the entire process efficient and stress-free.

5. Subscription Management: Beyond its immediate functions, the RFID badge also aids in subscription management. Parking administrators can easily track and monitor usage, ensuring that subscribers receive the services they've paid for without any disruptions.

In summary, the RFID badge simplifies parking access, verifies subscriptions, heightens security, and optimizes the overall parking experience for users. It is a vital tool that has become an integral part of parking systems, combining convenience and security seamlessly.

6.3.2 BARCODE READER FOR PARKING TICKET

A 1D/2D barcode reader with RS232/USB HID (Human Interface Device) interfaces is a device used to scan and decode barcode information from various types of barcodes, including linear (1D) and two-dimensional (2D) barcodes. The RS232 and USB HID interfaces allow the barcode reader to communicate with a computer, point-of-sale (POS) system, or other devices for data capture and processing. Here's an overview of a barcode reader with these interfaces: 1D/2D Barcode Reader:

- Function: A barcode reader is designed to scan and decode barcode symbols, converting them into readable data that are processed by a computer or device.
- 1D Barcodes: These barcodes consist of a series of parallel lines of varying widths and spacing. They encode data in one dimension.
- 2D Barcodes: These barcodes use a matrix of dots, squares, or other patterns to encode data in two dimensions. They can store more information compared to 1D barcodes.

Interfaces: RS232/USB HID:

1. RS232 (Serial) Interface:
    - Purpose: The RS232 interface is a common serial communication standard used for data transmission between devices.
    - Usage: A barcode reader with an RS232 interface is connected to a computer or device using a serial cable. Scanned barcode data is transmitted as serial data to the connected device.
2. USB HID (Human Interface Device) Interface:
    - Purpose: The USB HID interface is used for connecting input devices like keyboards, mice, and barcode readers to computers.
    - Usage: A barcode reader with a USB HID interface is connected to a computer using a USB cable. The computer treats the barcode reader as a keyboard input device, and scanned data is input as if typed on a keyboard.

The barcode or QR code present on the parking ticket handed to users serves as a pivotal component of the parking system, fulfilling a range of essential functions.

1. Payment Verification: The barcode or QR code on the parking ticket plays a fundamental role in facilitating the payment process. As users prepare to exit the parking facility, he presents his ticket at the exit gate. The embedded code contains crucial data related to her parking session, including entry and exit timestamps.

2. Automatic Billing: When a user parking ticket is scanned at the exit gate, the system uses the code to calculate the parking fee in real time. This automation eliminates the need for manual calculations, making the payment process quick and accurate. The user is charged precisely for his actual parking duration.

3. Contactless Transaction: The barcode or QR code technology reduces physical contact within the parking system. Users can swiftly exit the facility without the need for physical ticket handling or interaction with attendants, enhancing both efficiency and safety.
4. Data Accuracy: The code on the user's parking ticket ensures the accuracy of billing and tracking. It serves as a unique identifier, associating her vehicle with her parking session. This precision minimizes the risk of errors in the payment process.

5. Streamlined Operations: Parking operators leverage the data from these codes to manage their facilities effectively. By analyzing occupancy patterns and trends, they can optimize operations, allocate resources efficiently, and enhance the overall user experience.

6. Security Enhancement: The use of these codes contributes to security within the parking system. They provide an extra layer of authentication, ensuring that only authorized vehicles can exit the facility, reducing the risk of unauthorized access.

In summary, the barcode or QR code on the parking ticket is a versatile tool that simplifies the payment process, enhances security, and streamlines parking facility operations. It represents a modern and efficient approach to parking management, offering benefits to both users and the operators overseeing the facility.

6.3.3 LCD DISPLAY FOR MESSAGE DISPLAY

An LCD (Liquid Crystal Display) is a visual output device commonly used to display text, numbers, graphics, and other visual information. LCDs are prevalent in various applications, including consumer electronics, industrial control systems, medical devices, and more. Here's an overview of an LCD with RS232/I2C interfaces:

LCD Display:

Function: An LCD uses liquid crystal technology to create images and text by manipulating light passing through liquid crystal cells. LCDs are available in various sizes and resolutions, making them versatile for different applications.

Types: Common types of LCDs include character displays (which display text and numbers) and graphic displays (which can show more complex images and graphics).

Interfaces: RS232/I2C:

1. RS232 (Serial) Interface:
   - Purpose: The RS232 interface is a serial communication standard used for data transmission between devices.
   - Usage: An LCD with an RS232 interface is connected to a computer, controller, or other devices using a serial cable. This allows sending commands and data to the display for content rendering.
2. I2C Interface:
   - Purpose: The I2C (Inter-Integrated Circuit) interface is a widely used communication protocol for connecting multiple devices within a system.
   - Usage: An LCD with an I2C interface is integrated into a system using the I2C bus. This allows multiple devices to communicate using a two-wire serial interface, simplifying the connectivity and control.

The LCDs are positioned at the entry and exit access points of a parking lot; their roles, functionalities, and operational mechanisms are pivotal to ensuring a smooth parking experience for users. Here, is the overview of the functions of the displays:

1. Information Dissemination: The primary purpose of LCDs is to serve as a vital communication channel. Positioned strategically at the entry and exit points of the parking facility, they are dedicated to conveying essential messages to users.

2. Guidance and Instructions: The displays play a pivotal role in guiding users through the parking process. At the entry point, they provide clear instructions on how to obtain a parking ticket or access the facility. This step-by-step guidance simplifies the initial entry procedure.

3. Occupancy Status: A crucial functionality of the displays is to relay real-time information about the parking lot's occupancy status. Users, including the narrator, can quickly determine whether there are available parking spaces or if the facility is currently at full capacity. This feature aids in informed decision-making.

4. Payment Details: As users prepare to exit the parking lot, the LCDs provide comprehensive information regarding parking fees, and accepted payment methods. This transparency streamlines the payment process and ensures that users are well-informed.

5. Emergency Messages: In the event of emergencies or unforeseen circumstances, such as facility closures or safety protocols, the displays serve as a critical means of communication. They convey important messages promptly to ensure user safety and compliance.

6. Dynamic Messaging: The versatility of the displays allows for the display of dynamic messages that are updated remotely. This capability enables the broadcasting of announcements related to special events, changes in operating hours, or any disruptions in parking services.

7. User-Friendly Interface: To enhance user experience, the interface of these displays is designed to be user-friendly. Users, like the narrator, can easily access the information they require, whether through touchscreen capabilities or straightforward button interfaces.

8. Integration with Systems: The LCDs are seamlessly integrated with the parking management system. They receive real-time data about occupancy, payment transactions, and event updates, ensuring that the information displayed is accurate and up to date.

In essence, the LCDs positioned at the entry and exit access points of the parking facility serve as indispensable tools for enhancing the overall user experience. Their ability to provide real-time information, guidance, and safety messages is pivotal in ensuring a smooth and informed parking process for users, including the narrator. These displays are a testament to modern parking management, offering a blend of convenience and user satisfaction.

6.4 PARKING AUTOMATIC PAY STATION

6.4.1 KEYBOARD FOR USER INPUT

A keyboard with a USB HID (Human Interface Device) interface is a standard computer input device that connects to a computer or other compatible device via a USB (Universal Serial Bus) connection. The USB HID interface allows the keyboard to communicate with the computer and input data as if the user were typing directly on the computer's keyboard. Here's an overview of a keyboard with a USB HID interface:

Keyboard with USB HID Interface:

Function: A keyboard is a peripheral input device that enables users to input text, numbers, symbols, and commands into a computer or device by pressing keys.

USB HID Interface:

Purpose: The USB HID interface is a standardized way for input devices like keyboards, mice, and game controllers to communicate with computers and other devices.

Usage: A keyboard with a USB HID interface is connected to a computer using a USB cable. The computer recognizes the keyboard as an input device, and the keystrokes are transmitted to the computer just like those from a built-in keyboard.

Integrating a keyboard for user input in a parking automatic pay station allows users to enter information such as license plate numbers, validation codes, and other relevant data needed for payment or validation.

Here's how to effectively implement a keyboard for user input in a parking automatic pay station:

1. Keyboard Type:

Choose a keyboard type that suits the space and design of the pay station. Opting for physical keyboards with tactile keys or touchscreen keyboards that are part of the display interface.

2. Clear Instructions:

Display clear on-screen instructions on how to use the keyboard. Users should understand which keys to press, how to navigate, and how to confirm their input.

3. Input Fields:

Organize the user interface with input fields for specific types of information, such as license plate numbers, validation codes, and more.

4. User-Friendly Layout:

Create a user-friendly and intuitive keyboard layout. Keys which contain common letters and symbols and be rationally arranged.

5. Special Characters:

Include keys for special characters, numbers, and alphabetic characters to accommodate different types of inputs.

6. Auto-Capitalization and Corrections:

If using a touchscreen keyboard, enable features like auto-capitalization and autocorrection to enhance user input accuracy.

7. Validation and Formatting:

Implement real-time validation and formatting checks to ensure accurate data entry. For instance, validate license plate patterns or validation codes.

8. User Feedback:

Provide visual feedback when keys are pressed, such as highlighting the pressed key or displaying the entered character on the screen.

9. Function Keys:

Include function keys like "Backspace" and "Enter" for easy correction and submission of data.

10. Language Selection:

If applicable, allow users to select their preferred language for input.

11. Touchscreen Integration:

If using a touchscreen display, ensure that the touchscreen keyboard is well-designed, responsive, and user-friendly.

12. Accessibility Considerations:

Make sure the keyboard design is accessible to all users, including those with disabilities. Consider features like larger buttons, high contrast, and screen reader compatibility.

Benefits of Using a Keyboard for User Input:

- Accurate Data Entry: Users can enter information accurately, reducing errors.

- Flexible Input: A keyboard accommodates various types of information, including alphanumeric, special characters, and numbers.
- User Independence: Users can independently input information without requiring assistance.
- Familiarity: Keyboards are a familiar input method for most people, leading to a smooth user experience.
- Efficient: The keyboard expedites the input process, especially for longer or more complex inputs.
- Versatility: A keyboard can handle different types of inputs and information, enhancing the functionality of the pay station.

Properly implementing a keyboard for user input in a parking automatic pay station involves designing a user-friendly interface, ensuring the keyboard is responsive, and integrating it seamlessly with the pay station's software. The goal is to enhance the overall user experience and streamline the payment and validation processes.

6.5 PARKING DATABASE AND SERVER

6.5.1 DATABASE FOR USER REGISTRATION

Designing a database for user registration in a parking system involves creating tables, relationships, and structures to manage user information efficiently. Here's the outline of the structure of the database for user registration:

1. Users Table:

Create a table named "Users" to store user information. Include columns for user ID (primary key), first name, last name, email address, phone number, username, and password.

2. Subscriptions Table:

Create a table named "Subscriptions" to store subscription-related information. Include columns for subscription ID (primary key), user ID (foreign key referencing Users table), subscription type, start date, end date, and payment status.

3. Vehicles Table:

Create a table named "Vehicles" to store vehicle information associated with users. Include columns for vehicle ID (primary key), user ID (foreign key), vehicle type, license plate number, and vehicle model.

4. Access Permissions Table:

Create a table named "Access Permissions" to manage access privileges. Include columns for permission ID (primary key), user ID (foreign key), access level (e.g., regular, premium), and access start/end dates.

5. Validation Codes Table:

If validation codes are used for discounts or free parking, create a table named "Validation Codes." Include columns for code ID (primary key), code value, code type, and validity period.

6. Transactions Table:

Create a table named "Transactions" to record parking transactions. Include columns for transaction ID (primary key), user ID (foreign key), vehicle ID (foreign key), entry time, exit time, parking duration, and payment amount.

7. Audit Trail Table:

For security and tracking purposes, create an "Audit Trail" table to log user activities and system changes. Include columns for activity ID (primary key), user ID (foreign key), timestamp, activity description, and IP address.

8. Database Relationships:

Establish relationships between tables using primary and foreign keys. For example, the Users table's user ID can serve as a foreign key in other related tables.

9. Indexing:

Use indexing to optimize query performance, especially for columns frequently used for searching, such as user IDs and email addresses.

10. Data Validation and Constraints:

Implement data validation rules and constraints to ensure data accuracy and consistency. Define constraints to prevent duplicate entries and maintain referential integrity. Benefits of a Well-Designed Database:

- Efficient Data Management: A structured database allows efficient storage and retrieval of user-related information.
- User Customization: The database design enables customization of user profiles, subscriptions, and access levels.
- Historical Tracking: Transaction records provide a historical record of user activities and parking transactions.
- Analytics and Reporting: Data stored in the database are used for generating reports and analyzing usage patterns.

- Security: A well-designed database with appropriate access controls enhances data security. Regular maintenance, backup strategies, and data security measures are vital to ensure the integrity and availability of the stored data.

6.5.2 TCP/IP FOR SENDING DATA TO THE SERVER

Using TCP/IP for sending data to a server in a parking system is a common and reliable method for transmitting information over a network. TCP/IP (Transmission Control Protocol/Internet Protocol) provides a structured and organized way to establish a connection, transfer data, and ensure data integrity between devices or systems. Here's how TCP/IP are used to send data to a server in a parking system:

1. Establishing a Connection:

Before sending data, a connection needs to be established between the client (parking system) and the server.

The client initiates a connection request to the server, and the server responds to confirm the connection.

2. Data Segmentation:

Data to be transmitted is divided into smaller segments called packets. Each packet is assigned a sequence number. 3. Packet Transmission:

The client sends the packets to the server through the established connection. The packets may travel different routes through the network. 4. Packet Arrival and Acknowledgment:

The server receives the packets and sends acknowledgment packets back to the client, indicating successful receipt. 5. Data Reassembly:

The server reassembles the received packets in the correct order to reconstruct the original data. 6. Error Checking:

TCP/IP includes error-checking mechanisms to ensure data integrity. If a packet is lost or corrupted during transmission, the server requests retransmission.

7. Flow Control:

TCP/IP manages the flow of data between the client and server to prevent overwhelming the receiving end with too much data at once.

8. Connection Termination:

Once the data transmission is complete, the client or server can initiate connection termination, ensuring that resources are released properly.

Benefits of Using TCP/IP for Data Transmission:

- Reliability: TCP/IP ensures reliable data transmission through acknowledgment and errorchecking mechanisms.

- Order Preservation: Data segments are reassembled in the correct order on the receiving end.
- Data Integrity: Error-checking mechanisms detect and correct data corruption.
- Flow Control: TCP/IP manages data flow to prevent congestion and optimize transmission.
- Versatility: TCP/IP is widely used and supported, making it suitable for various network environments.
- Compatibility: TCP/IP is compatible with different types of networks, including wired and wireless.
- Security: TCP/IP is used within secure network protocols, adding layer of protection. Using TCP/IP for sending data to a server in a parking system ensures that critical information, such as parking transactions, user data, and access logs, is transmitted accurately and securely. It's important to implement proper network security measures, such as encryption and authentication, to protect the data during transmission.

6.5.3 MR004 BOARD FOR COMMUNICATING WITH CLIENT

Creating a custom board for communication with clients in a parking system involves designing a hardware solution that facilitates data exchange between the various components of the system and the clients, which could include entry/exit stations, automatic pay stations, and more. Here are the approaches to designing such a custom communication board:

1. Define Communication Needs:

Identify the types of communication required, such as Ethernet, Wi-Fi, Bluetooth, RS-232, RS-485, or a combination of these [19].

2. Board Selection:

Based on the communication needs and system complexity, pick a suitable hardware platform, such as a microcontroller or single-board computer (SBC).

3. Communication Interfaces:

Include communication interfaces, such as Ethernet ports, UART (RS-232/RS-485) ports, WiFi/Bluetooth modules, USB ports, or other necessary interfaces, that correspond to the communication requirements.

4. Data Processing:

Select a microcontroller or processor with sufficient processing power to handle data exchange, protocol handling, and any necessary data processing.

5. Memory and Storage:

Include memory components (RAM and flash storage) to store data temporarily and for firmware storage.

6. Power Management:

Design power circuitry to provide a stable and efficient power supply to the board, considering powersaving features if applicable.

7. Communication Protocols:

Implement communication protocols that align with the requirements of the parking system, such as TCP/IP, MQTT, HTTP, or custom protocols.

8. Security Measures:

Incorporate security features like encryption and authentication to ensure the confidentiality and integrity of data during communication.

9. Firmware Development:

Develop firmware that manages data communication, protocol handling, error checking, and data formatting.

10. Software Compatibility:

Ensure that the custom board's firmware and communication protocols are compatible with the software used in the parking system.

11. Testing and Validation:

Thoroughly test the custom communication board in various scenarios to ensure reliable data exchange and proper functionality. 12. Scalability and Future Expansion:

Design the board with future system expansion and updates in mind, allowing for additional features and functionalities.

Benefits of a Custom Communication Board:

- Tailored Solution: A custom board is designed to meet the specific communication needs of the parking system.
- Optimized Performance: The boards are optimized for efficient data exchange and reduced latency.
- Flexibility: A custom board can accommodate various communication interfaces, enabling seamless integration with different client devices.
- Security: Custom security features are implemented to protect data and prevent unauthorized access.
- Reduced Costs: A well-designed custom board can lead to cost savings compared to using off-the-shelf solutions.
- System Integration: The board is integrated seamlessly into the existing parking system architecture.

Creating a custom communication board requires a combination of hardware and software design skills to ensure that the board meets the communication needs of the parking system and its clients effectively.

6.6 I/O FOR EXTERNAL PERIPHERALS

In a parking system, external peripherals such as sensors, actuators, displays, and communication devices play a crucial role in ensuring the system's proper functioning. Designing an effective I/O (Input/Output) system to interface with these external peripherals is essential. Here are the approaches of I/O for external peripherals in a parking system:

1. Peripheral Identification:

Identify the external peripherals that need to be connected to the parking system, such as sensors (RFID, ultrasonic), actuators (barriers, gates), displays (LCD, LED), and communication devices (Ethernet, RS-232, RS-485).

2. Interface Selection:

Choose appropriate interface protocols for each peripheral based on their communication requirements (e.g., digital, analog, serial, parallel).

3. Microcontroller or SBC Selection:

Determine the microcontroller or single-board computer (SBC) that will manage the I/O operations and peripheral interactions. 4. I/O Ports and Pins Allocation:

Allocate I/O pins on the microcontroller or SBC for each peripheral, ensuring compatibility with the chosen interface protocols.

5. Wiring and Connection:

Design a wiring scheme to connect the external peripherals to the designated I/O pins. Use proper connectors, cables, and shielding to ensure reliable connections.

6. Communication Protocols:

Implement appropriate communication protocols for each peripheral, such as SPI, I2C, UART, or custom protocols. 7. Signal Conditioning:

Use signal conditioning techniques, such as voltage level conversion or amplification, to ensure proper signal integrity between the microcontroller and the peripherals.

8. Error Handling and Recovery:

Implement error handling mechanisms to detect and manage communication errors or faults with peripherals.

9. Firmware Development:

Develop firmware to manage I/O operations, data exchange, and interactions with the external peripherals. 10. Data Processing:

Implement necessary data processing algorithms to interpret sensor data, control actuators, and update display information.

11. Event Handling:

Set up event-driven mechanisms to respond to changes in peripheral states or user interactions. 12. Testing and Validation:

Thoroughly test the I/O system with various scenarios to ensure that it interacts correctly with external peripherals.

Benefits of Effective I/O for External Peripherals:

- Reliability: Properly managed I/O ensures reliable communication with external devices.
- Interoperability: Different types of peripherals are integrated seamlessly into the system.
- Customization: Designing the I/O system allows one to tailor it to the specific requirements of the parking system.
- Scalability: The I/O system is designed to accommodate future expansion and additional peripherals.
- Real-time Response: Effective I/O management enables quick response to changes in peripheral states.
- Centralized Control: The microcontroller or SBC centralizes control over various peripherals. A well-designed I/O system for external peripherals is critical for the overall performance and functionality of the parking system. Collaboration between hardware designers, firmware developers, and system architects is essential to ensure proper integration and functionality of the system's components.

6.7 AUTOMATIC SHUTDOWN AND RESTART

Implementing an automatic shutdown and restart functionality in a parking system can enhance system reliability, reduce energy consumption, and ensure that the system operates smoothly. Here's the implementation of automatic shutdown and restart in a parking system: Automatic Shutdown:

1. Idle Time Detection:

Implement a mechanism to detect idle periods when the parking system is not actively processing transactions or serving users.

2. Idle Time Threshold:

Define a threshold for idle time duration. For example, if there's no activity for a predefined period (e.g., 30 minutes), consider the system as idle.

3. Graceful Shutdown:
Initiate a graceful shutdown process, ensuring that ongoing transactions are completed, and data is saved properly. 4. Notify Users:
Display a message on external displays notifying users about the impending system shutdown and the reason behind it. 5. Turn Off Peripherals:
Turn off non-essential peripherals, such as displays, sensors, and actuators, to conserve power.
6. Shutdown Command:
Use the microcontroller or operating system's shutdown command to power off the system.

Automatic Restart:
1. Scheduled Restart:
Implement a schedule for automatic system restarts during periods of low usage, such as during the early morning hours.
2. System Health Monitoring:
Incorporate system health monitoring mechanisms to detect any anomalies or issues during operation.
3. Automatic Recovery:
If the system detects a critical error or malfunction, automatically trigger a system restart to attempt recovery.
4. Log Reporting:
Record events and system status before the restart, enabling administrators to review logs for diagnosis if necessary. 5. Initialization Process:
Ensure that all necessary components and peripherals are properly initialized during the system restart.
6. Notification and Progress Display:
Display a notification on external displays to inform users about the restart process and its progress.
Benefits of Automatic Shutdown and Restart:

- Energy Efficiency: Automatic shutdown reduces energy consumption during idle periods.
- Prevent Overloads: Automatic restarts help prevent memory leaks or system slowdowns that might accumulate over time.
- System Reliability: Regular restarts help prevent system crashes and maintain stable operation.
- Software Updates: Restarting the system periodically facilitates the installation of software updates.

- User Experience: Graceful shutdowns and restarts contribute to a positive user experience by minimizing disruptions.
- Maintenance: Scheduled restarts are used as part of maintenance routines to refresh the system.

When implementing automatic shutdown and restart in a parking system, consider the specific operational patterns, user expectations, and the overall goals of the system. Regular testing and monitoring of the shutdown and restart procedures are essential to ensure their effectiveness and reliability.

CHAPTER 7: CHALLENGES

7.1 REPLACEMENT OF EMBEDDED MICROCONTROLLER WITH SBC

There are multiple convincing reasons why switching the parking system's proprietary embedded microprocessor, the Rabbit Semiconductors BL4S2xx, to the Single Board Computer PICO ITX is necessary. The BL4S2xx microcontroller's considerable supply restrictions and high market costs are the main factors behind this move.

The BL4S2xx microcontroller's primary problems are market overpriced and limited supply. The market price of this component has significantly increased because of the component's limited supply and increased demand. This price difficulty places a financial cost on the project and can put a strain on its financial restraints.

The problem is made worse by the scarcity of BL4S2xx microcontroller providers. Requests for this vital component are only slowly fulfilled because there aren't many reliable suppliers available for purchasing. As a result, there will inevitably be delays in the replacement process, which might be problematic, especially when the parking system needs immediate modifications or repairs.

The PICO ITX Single Board Computer, in comparison, has many benefits. Due to its improved features and functions, it stands out as a more economically feasible choice. With its new approach, the PICO ITX positions itself as a cutting-edge replacement for the parking system by promising enhanced performance and compatibility with modern technology. The PICO ITX's easy accessibility to the market is another appealing feature.

Due to its extensive use, the procurement process is streamlined, potentially reducing the supply chain issues that have hampered the BL4S2xx microcontroller replacement.

Additionally, the PICO ITX has a wider variety of connections, making it easier to integrate a variety of auxiliary devices. This increased adaptability allows for greater freedom when designing or expanding the parking system, allowing for future improvements, and incorporating new capabilities.

In conclusion, the unavoidable difficulties brought on by the Rabbit Semiconductors BL4S2xx, such as supply shortages and high market costs, highlight the choice to switch to the PICO ITX. The PICO ITX makes a strong case for successful integration into the parking system because of its costeffectiveness, updated features, and increased availability. It also promises enhanced performance and long-term viability.

7.2 REVISION OF THE FIRMWARE IN C AND C#

The decision to implement this modification is driven by several important issues in the context of the parking system, which presently uses C firmware and is in the process of switching to C# firmware. The in-use pre-existing C firmware was out of date, a legacy system that had undergone countless revisions over time by various developers. Given these difficulties, it was decided to launch a thorough upgrade and begin the firmware development from scratch. The aim to reorganize and make the specification of functionality for each system module served as the primary impetus for this choice.

It was difficult to handle and maintain the current C firmware since it had become a complex system. Over time, several engineers have contributed to the codebase, which left the code structure lacking in coherence and clarity. Due to this intricacy, it has become challenging to include new features or enhancements as well as to discover and fix problems.

The use of C# firmware from scratch was thought to be a more practical solution to these problems. The development team could then clearly and methodically specify the functionality of each module, guaranteeing that the new firmware would be more durable, manageable, and responsive to future modifications. The switch to C# has the added benefit of updating the system and utilizing the advantages of a more advanced programming language.
In conclusion, the decision to replace the parking system's traditional C firmware with C# firmware resulted from the necessity to address the legacy code's complexity and antiquated character. Starting from scratch offered the chance to define functions more precisely and develop a system that is efficient and maintained, eventually assuring the parking system's long-term sustainability and flexibility.

7.3 UTILIZATION OF A NEW IDE WITH AN UNFAMILIAR ITALIAN VERSION

In the initial experience with Visual Studio 2022, Working with this IDE for the very first time was challenging. Notably, Visual Studio 2022 was configured to operate in the company's official language, Italian, which presented a considerable challenge as this language was entirely unfamiliar. Due to the language barrier, the adventure began with a sense of uncertainty and a steep learning curve. It might be frightening to navigate an unfamiliar IDE in a foreign language. However, as time passed, a self-guided learning process to comprehend the intricacies of Visual Studio 2022 progressed well.

Progressively understood the IDE's interface, tools, and functions through a combination of resourceful investigation, online documentation, and the aid of coworkers who were fluent in Italian. They were able to successfully utilize Visual Studio 2022's capabilities for their work-related duties thanks to a process of trial and error and a dedication to learning.

Gradually became more adept at utilizing the IDE, which helped to operate more productively in the company's development environment. The capacity to adapt and persevere in the face of linguistic obstacles and successfully incorporate a new tool into their workflow was demonstrated by this experience.

Finally, the experience with Visual Studio 2022 provided proof of their ability to adapt and learn, despite the challenge of a foreign language. As time went on, overcoming the language barrier was not only the thing but also honed the skills in using the IDE for work-related tasks, which increased their efficacy in their position.

7.4 COLLABORATING WITH HARDWARE AND ELECTRICAL ENGINEERING

Lacking in-depth knowledge of hardware engineering, A steep learning curve about the given assignment of creating connections, a skill set that was originally unfamiliar. Starting a trip to gain a thorough grasp of these complex systems because they were fresh to the world of hardware components and circuits. Learning about the many components and understanding how each one functioned within the overall scheme before beginning.
The procedure was first characterized by its complexities and the requirement to take in a sizable amount of new knowledge. Trouble understanding the complexities of hardware engineering, from recognizing the many components to understanding how they worked together.

However, the improvement over time with unyielding persistence and a dedication to learning was showing. Carried out real experiments, sharpening the abilities by working directly with hardware elements and circuits. Gradually put the pieces together, developing the skills necessary to forge connections and successfully troubleshoot problems.
One's perseverance and desire to take on new difficulties were demonstrated by the road he or she took to gain real-world expertise in hardware engineering. Even though the early challenges were great, their commitment to learning the intricate details of hardware parts and circuits finally helped them to get the knowledge necessary to handle challenging assignments in this industry.

7.5 LCD DISPLAYS EMPLOY A SPECIFIC COMMUNICATION PROTOCOL

The use of HEX to ASCII conversion for message display on the screen was a remarkable feature of the LCD. Due to its distinctive communication protocol, which is made up of separate parts such as the start command, data type, data, and end command, each of which has its characteristics, it stood out from other LCDs on the market.

A unique method of showing messages on the display was the usage of HEX to ASCII conversion. It enabled the display of text and information in a human-friendly format by converting hexadecimal data into legible ASCII characters. The effectiveness of the LCD's message and data delivery to the user depended heavily on this conversion process.

The complex communication protocol of this LCD was what made it stand out from others. An initiation signal was sent to the LCD to include a start instruction, instructing it to get ready for incoming data. While the data itself included the actual content to be displayed, the data type described the sort of information being transferred. Finally, the end command provided a distinct boundary by signaling the end of the data flow.

This specific display's communication protocol was noticeably different from more traditional LCDs. To ensure proper message rendering, a specific order of operations and data types was required. This protocol gave greater flexibility and control over the presented material, but it also introduced a degree of complexity.

In conclusion, this LCD's unique communication protocol and utilization of HEX to ASCII conversion constituted a cutting-edge method of message presentation. Although it complicated data transfer, it also gave users more personalization and control over the displayed material, making it stand out from other LCDs on the market.

7.6 ABOLISH THE USAGE OF COMPONENTS

Use of a certain keyboard for data input in the context of the entrance and exit system as well as the automatic pay station. But as they got going on the work, it became clear that the selected keyboard lacked the capability needed for the module in question.

The first intended keyboard's shortcomings in meeting the unique requirements of the module being built led to the decision to cease using it. It was discovered that the keyboard's capabilities were insufficient for the project's exact functionalities and data-entering needs. Recognizing this

deficiency, the decision to investigate alternatives that would better satisfy the requirements of the project.

Unexpectedly, this encounter with the less-ideal keyboard had a useful outcome. It made it possible to grow more accustomed to the features and libraries connected to USB keyboards. The gained knowledge and abilities would be useful in upcoming projects because of their newly acquired familiarity, which also made the transfer to a different keyboard easier.

In summary, even though the first keyboard selection did not meet the needs of the module being created, it offered a priceless learning opportunity. It enabled them to become proficient in the libraries and capabilities of USB keyboards, which would be a benefit for further projects in related fields. The capacity for adaptation and openness to considering other options demonstrated their dedication to coming up with workable solutions for their initiatives.

8.1 MR004 CLIENT FILE EXPLANATION

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net6.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="System.Configuration.ConfigurationManager" Version="7.0.0" />
  </ItemGroup>

</Project>
```

Figure 5 Project Definition for MR004 Client

The code defines a .NET project targeting version 6.0 of the .NET framework. It also specifies that implicit using and nullable reference types are enabled. Additionally, it includes a reference to the "System.Configuration.ConfigurationManager" package with a specific version. This configuration file is essential for building and managing .NET projects, as it determines how the project is built and which dependencies are required.

This file is used to organize and manage multiple projects within a Visual Studio solution. Let's break down the structure of this code:

1. Project Definition:

```
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "MR004_Client", "MR004_
EndProject
```

:

Figure 6 Project Type definition

This part defines the type of project. The code within the curly braces '{...}' is a GUID that identifies the type of project. In this case, '{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}' represents a C# project.

- "MR004_Client": This is the name of the project as it appears in Visual Studio.

- "MR004_Client\MR004_Client.csproj": This is the relative path to the project file (‘.csproj’) associated with this project within the solution.
- ‘{6CF7A806-59E2-4025-A267-7B91D57055EE}’: This is a unique GUID that identifies the project within the solution. It's used for referencing this project from other parts of the solution file.

2. Global Configuration:

This marks the beginning of the global section of the solution file, which contains settings and configurations that apply to the entire solution.

3. Solution Configuration Platforms:

```
GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
EndGlobalSection
```

Figure 7 Solution configuration

This section defines the available solution configurations and the corresponding platform targets. In this case, there are two configurations: "Debug" and "Release," both targeting "Any CPU."

4. Project Configuration Platforms:

```
GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {6CF7A806-59E2-4025-A267-7B91D57055EE}.Debug|Any CPU.ActiveCfg = Debug|A
    {6CF7A806-59E2-4025-A267-7B91D57055EE}.Debug|Any CPU.Build.0 = Debug|Any
    {6CF7A806-59E2-4025-A267-7B91D57055EE}.Release|Any CPU.ActiveCfg = Relea
    {6CF7A806-59E2-4025-A267-7B91D57055EE}.Release|Any CPU.Build.0 = Release
EndGlobalSection
```

Figure 8 Build Configuration for each project.

This section specifies the build configurations for each project in the solution. It associates the project GUIDs with the configurations defined in the "Solution Configuration Platforms" section.

5. Solution Properties:

```
GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
EndGlobalSection
```

-

Figure 9 Solution-level properties

This section defines solution-level properties. In this case, it specifies that the solution node should not be hidden in the Solution Explorer within Visual Studio (i.e., 'HideSolutionNode = FALSE').

6. Extensibility Global:

```
GlobalSection(ExtensibilityGlobals) = postSolution
    SolutionGuid = {9F5BFAAD-B6AA-4635-801C-F6E58FFF4D15}
EndGlobalSection
```

Figure 10 : Extensibility and custom configuration.

This section can be used for extensibility and custom configuration. It includes the GUID of the solution, which can be used by extensions or tools to uniquely identify the solution.

In summary, the code is a section of a Visual Studio solution file that defines a C# project named "MR004_Client" and specifies its configurations, along with some global solution settings.

8.2 RFIDTAG.SLN FILE EXPLANATION

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio Version 17
VisualStudioVersion = 17.5.33516.290
MinimumVisualStudioVersion = 10.0.40219.1
Project("{9A19103F-16F7-4668-BE54-9A1E7A4F7556}") = "RfidTag", "RfidTag\RfidTag.csproj", "{2CE6514C-D876-485C-A606-2E10A74B6667}"
EndProject
Global
	GlobalSection(SolutionConfigurationPlatforms) = preSolution
		Debug|Any CPU = Debug|Any CPU
		Release|Any CPU = Release|Any CPU
	EndGlobalSection
	GlobalSection(ProjectConfigurationPlatforms) = postSolution
		{2CE6514C-D876-485C-A606-2E10A74B6667}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
		{2CE6514C-D876-485C-A606-2E10A74B6667}.Debug|Any CPU.Build.0 = Debug|Any CPU
		{2CE6514C-D876-485C-A606-2E10A74B6667}.Release|Any CPU.ActiveCfg = Release|Any CPU
		{2CE6514C-D876-485C-A606-2E10A74B6667}.Release|Any CPU.Build.0 = Release|Any CPU
	EndGlobalSection
	GlobalSection(SolutionProperties) = preSolution
		HideSolutionNode = FALSE
	EndGlobalSection
	GlobalSection(ExtensibilityGlobals) = postSolution
		SolutionGuid = {B9392EC4-1860-4596-BD7C-B9CBAD978FD3}
	EndGlobalSection
EndGlobal
```

:

Figure 11 Project Definition for RFIDTag

1. Project Definition:

'Project ("{...}")': This part defines the type of project. The code within the curly braces '{...}' is a GUID that identifies the type of project. In this case, '{9A19103F-16F7-4668-BE549A1E7A4F7556}' represents a C# project.

- "RfidTag": This is the name of the project as it appears in Visual Studio.
- "RfidTag\RfidTag.csproj": This is the relative path to the project file ('.csproj') associated with this project within the solution.
- '{2CE6514C-D876-485C-A606-2E10A74B6667}': This is a unique GUID that identifies the project within the solution. It's used for referencing this project from other parts of the solution file.

2. Global Configuration:

This marks the beginning of the global section of the solution file, which contains settings and configurations that apply to the entire solution.

3. Solution Configuration Platforms:

```
GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
EndGlobalSection
```

:

Figure 12 Solution Configuration

This section defines the available solution configurations and the corresponding platform targets. In this case, there are two configurations: "Debug" and "Release," both targeting "Any CPU."

4. Project Configuration Platforms:

```
GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {2CE6514C-D876-485C-A606-2E10A74B6667}.Debug|Any CPU.ActiveCfg = Debug|A
    {2CE6514C-D876-485C-A606-2E10A74B6667}.Debug|Any CPU.Build.0 = Debug|Any
    {2CE6514C-D876-485C-A606-2E10A74B6667}.Release|Any CPU.ActiveCfg = Relea
    {2CE6514C-D876-485C-A606-2E10A74B6667}.Release|Any CPU.Build.0 = Release
EndGlobalSection
```

:

Figure 13 Build Configuration for project identified by GUID.

This section specifies the build configurations for the project identified by the GUID '{2CE6514CD876-485C-A606-2E10A74B6667}'. It associates the project GUID with the configurations defined in the "Solution Configuration Platforms" section.

5. Solution Properties:

```
GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
EndGlobalSection
```

Figure 14 Solution Properties

This section defines solution-level properties. In this case, it specifies that the solution node should not be hidden in the Solution Explorer within Visual Studio (i.e., ‘HideSolutionNode = FALSE’).

6. Extensibility Global:

```
GlobalSection(ExtensibilityGlobals) = postSolution
    SolutionGuid = {B9392EC4-1860-4596-BD7C-B9CBAD978FD3}
EndGlobalSection
```

:

Figure 15 Extensibility and custom configuration

This section can be used for extensibility and custom configuration. It includes the GUID of the solution, which can be used by extensions or tools to uniquely identify the solution.

In summary, the code you provided is a section of a Visual Studio solution file that defines a C# project named "RfidTag" and specifies its configurations, along with some global solution settings. This is part of how Visual Studio organizes and manages projects within a solution.

8.3 USB_QR_Reader.sln FILE EXPLANATION

```
Microsoft Visual Studio Solution File, Format Version 12.00
# Visual Studio Version 17
VisualStudioVersion = 17.5.33516.290
MinimumVisualStudioVersion = 10.0.40219.1
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "USB_QR_Reader", "USB_QR_Reader\USB_QR_Reader.csproj", "{D9AAF39B-D348-4CE7-B874-2573FD319E60}"
EndProject
Global
	GlobalSection(SolutionConfigurationPlatforms) = preSolution
		Debug|Any CPU = Debug|Any CPU
		Release|Any CPU = Release|Any CPU
	EndGlobalSection
	GlobalSection(ProjectConfigurationPlatforms) = postSolution
		{D9AAF39B-D348-4CE7-B874-2573FD319E60}.Debug|Any CPU.ActiveCfg = Debug|Any CPU
		{D9AAF39B-D348-4CE7-B874-2573FD319E60}.Debug|Any CPU.Build.0 = Debug|Any CPU
		{D9AAF39B-D348-4CE7-B874-2573FD319E60}.Release|Any CPU.ActiveCfg = Release|Any CPU
		{D9AAF39B-D348-4CE7-B874-2573FD319E60}.Release|Any CPU.Build.0 = Release|Any CPU
	EndGlobalSection
	GlobalSection(SolutionProperties) = preSolution
		HideSolutionNode = FALSE
	EndGlobalSection
	GlobalSection(ExtensibilityGlobals) = postSolution
		SolutionGuid = {386911A9-072F-45C4-854D-7703E3A1062F}
	EndGlobalSection
EndGlobal
```

:

Figure 16 Project Definition for USB_QR_Reader

1. Project Definition:

```
Project("{FAE04EC0-301F-11D3-BF4B-00C04F79EFBC}") = "USB_QR_Reader", "USB_Q
EndProject
```

:

Figure 17 Project type definition

'Project ("{...}")': This part defines the type of project. The code within the curly braces '{...}' is a GUID that identifies the type of project.

- "USB_QR_Reader": This is the name of the project as it appears in Visual Studio.
- "USB_QR_Reader\USB_QR_Reader.csproj": This is the relative path to the project file ('.csproj') associated with this project within the solution.
- '{D9AAF39B-D348-4CE7-B874-2573FD319E60}': This is a unique GUID that identifies the project within the solution. It's used for referencing this project from other parts of the solution file.

2. Global Configuration:

This marks the beginning of the global section of the solution file, which contains settings and configurations that apply to the entire solution.

3. Solution Configuration Platforms:

```
GlobalSection(SolutionConfigurationPlatforms) = preSolution
    Debug|Any CPU = Debug|Any CPU
    Release|Any CPU = Release|Any CPU
EndGlobalSection
```

:

Figure 18 Solution Configuration

This section defines the available solution configurations and the corresponding platform targets. In this case, there are two configurations: "Debug" and "Release," both targeting "Any CPU."

4. Project Configuration Platforms:

```
GlobalSection(ProjectConfigurationPlatforms) = postSolution
    {D9AAF39B-D348-4CE7-B874-2573FD319E60}.Debug|Any CPU.ActiveCfg = Debug|
    {D9AAF39B-D348-4CE7-B874-2573FD319E60}.Debug|Any CPU.Build.0 = Debug|An
    {D9AAF39B-D348-4CE7-B874-2573FD319E60}.Release|Any CPU.ActiveCfg = Rele
    {D9AAF39B-D348-4CE7-B874-2573FD319E60}.Release|Any CPU.Build.0 = Releas
EndGlobalSection
```

:

Figure 19 Build configuration for the project.

This section specifies the build configurations for the project identified by the GUID '{D9AAF39BD348-4CE7-B874-2573FD319E60}'. It associates the project GUID with the configurations defined in the "Solution Configuration Platforms" section.

5. Solution Properties:

```
GlobalSection(SolutionProperties) = preSolution
    HideSolutionNode = FALSE
EndGlobalSection
```

: -

Figure 20 Solution level properties

This section defines solution-level properties. In this case, it specifies that the solution node should not be hidden in the Solution Explorer within Visual Studio (i.e., `HideSolutionNode = FALSE`).

6. Extensibility Global:

```
GlobalSection(ExtensibilityGlobals) = postSolution
    SolutionGuid = {386911A9-072F-45C4-854D-7703E3A1062F}
EndGlobalSection
```

:

Figure 21 Extensibility and custom configuration

This section can be used for extensibility and custom configuration. It includes the GUID of the solution, which can be used by extensions or tools to uniquely identify the solution.

8.4 TEST OUTPUT

Here are a few results of the tests that were conducted on the software that has been developed.

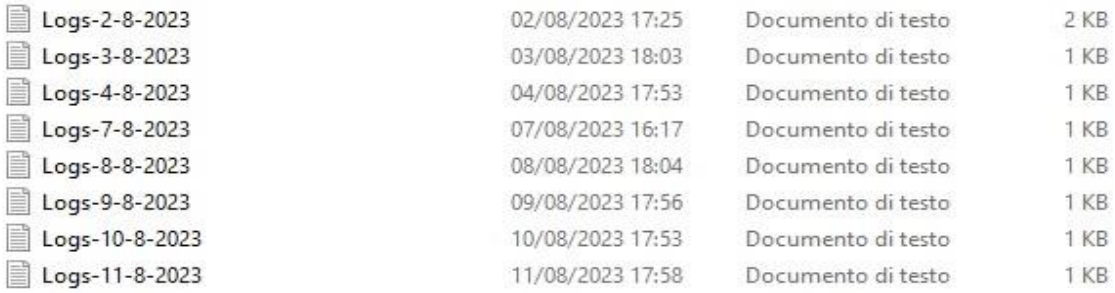
8.4.1 USB_QR_Reader

The barcode and QR code reader software developed for the entry and exit access in the parking lot underwent an extended duration of testing without interruption to thoroughly assess and identify any potential faults.

:

Figure 22 File for the month of august

The file is automatically created every month to store the data inserted during that month using the barcode or QR code reader at the entry and exit access points in the parking lot.



:

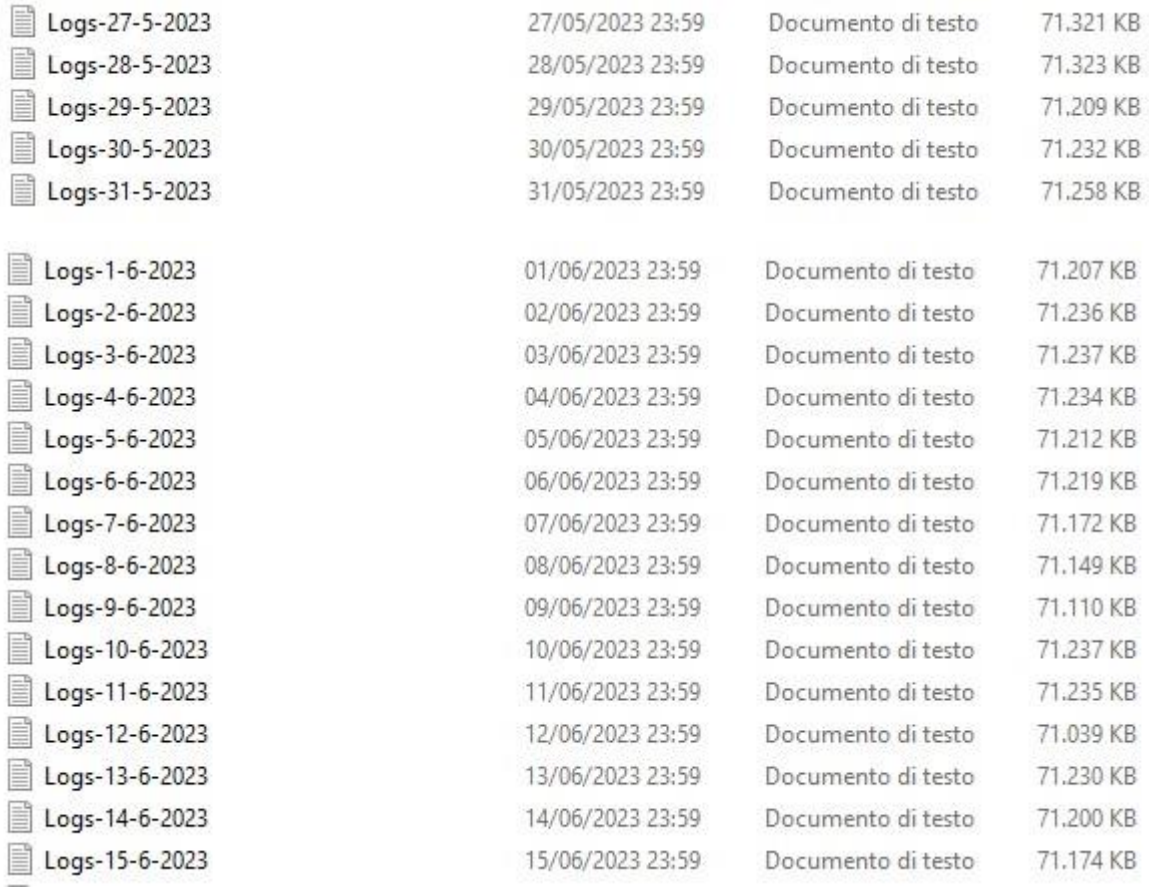
Figure 23 Logs from 2nd to 11th of August

These are all the log files created between the 2nd and 11th of August.

8.4.2 LCD

The usage of an LCD display for conveying messages to the clients at the entry and exit access points in the parking lot involved rigorous testing to identify any potential faults in the software. This testing process included continuous operation for several months without any interruptions, ensuring the software's reliability and stability.

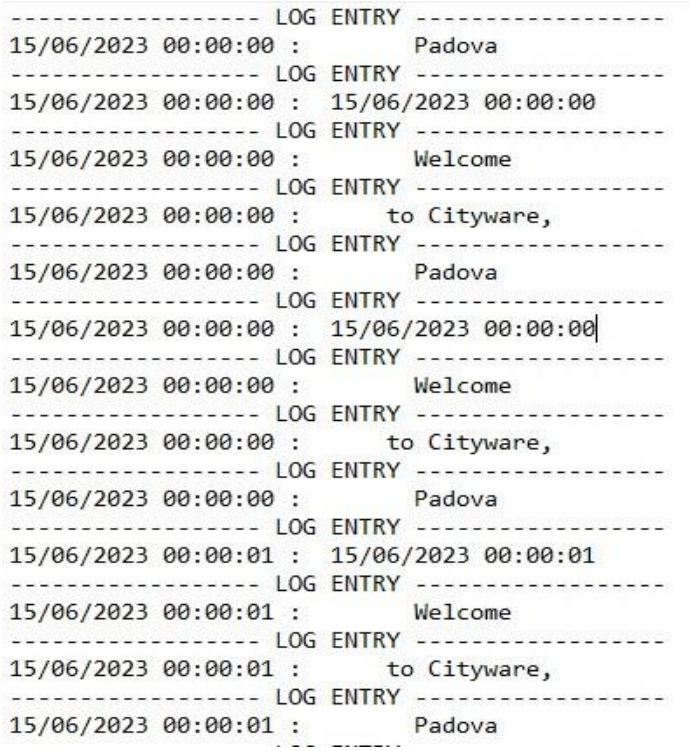
Figure 24 : Files for the months



:

Figure 25 Logs from the month of May and June

These log files represent the records generated during the months of May and June while the system ran continuously 24/7.



:

Figure 26 Example of the log created.

These are the dates collected within each log file, with a new file created every day.

8.4.3 MR004 Client

The utilization of the custom board MR004 for communication with the PICO is accomplished through a TCP/IP client, serving the purpose of sending and receiving data.

```
--------------------LOG ENTRY-------------------

15/05/2023 13:28:23 : __1__SENT:    ▯1101140004▯
15/05/2023 13:28:23 : __2__RECEIVED:    ▯0111810091▯

15/05/2023 13:28:23 : __1__SENT:    ▯1101140004▯
15/05/2023 13:28:23 : __2__RECEIVED:    ▯0111810091▯

15/05/2023 13:28:23 : __1__SENT:    ▯1101140004▯
15/05/2023 13:28:23 : __2__RECEIVED:    ▯0111810091▯

15/05/2023 13:28:23 : __1__SENT:    ▯1101140004▯
15/05/2023 13:28:23 : __2__RECEIVED:    ▯0111810091▯

15/05/2023 13:28:23 : __1__SENT:    ▯1101140004▯
15/05/2023 13:28:23 : __2__RECEIVED:    ▯0111810091▯

15/05/2023 13:28:23 : __1__SENT:    ▯1101140004▯
15/05/2023 13:28:23 : __2__RECEIVED:    ▯0111810091▯

15/05/2023 13:28:23 : __1__SENT:    ▯1101140004▯
15/05/2023 13:28:23 : __2__RECEIVED:    ▯0111810091▯

15/05/2023 13:28:23 : __1__SENT:    ▯1101140004▯
15/05/2023 13:28:23 : __2__RECEIVED:    ▯0111810091▯
```

Figure 27 Logs example of messages exchanged

These data represent the information collected within the log file, detailing the messages sent from the PICO and the corresponding automatic message replies from the MR004.

CHAPTER 9: CONCLUSION AND FUTURE WORKS

In conclusion, the experimental development and performance assessment of Industrial Single Board Computers (SBCs) inside a Parking Automation System have offered invaluable knowledge and created a framework for further developments.
Moving forward, the next logical steps are the development and integration of the correct keyboard, touch screen display, and automatic pay station into this promising system.

Our main concern throughout this research has been to determine the feasibility and future promise of SBCs to enhance parking management systems. The tests have shown that SBCs can handle the demanding tasks necessary for effective parking automation. Their dependable performance and flexibility have demonstrated considerable potential, opening the door for this project's important next stage.

In terms of future works, the development of the correct keyboard, touch screen display, and automatic pay station will be pivotal. These parts make up the user-facing portion of the Parking Automation System and have a big impact on how the user interacts with the system. Therefore, the design, functionality, and integration to ensure optimal performance and customer satisfaction is carefully handled.

The correct keyboard selection involves not only choosing the right hardware but also considering the practical aspects and user interface design. Data entry will be more effective if the keyboard is well-designed, and both customers and parking attendants will be able to utilize it.
Effective real-time information delivery will be facilitated by high-resolution screens with responsive touch capabilities.
The automatic pay station is also a crucial part in smoothly processing payments. With support for multiple payment options and ticket validation procedures, it is integrated to ensure safe and quick transactions.

Future research in this field is examined to incorporate cutting-edge tools like artificial intelligence and machine learning for better security and predictive parking management.

The overall efficiency of the parking system may also be improved by implementing IoT sensors for real-time data collection and analysis. These forthcoming initiatives have the potential to completely transform the parking field soon, benefiting both operators and users.

CHAPTER 10: BIBLIOGRAPHY

R. Priya, Abhishek, Ankur Utsav (2021). A unique automatic parking system by using RFID card and IoT. Department of Electronics and Communication. Government Engineering College, Vaishali, Bihar, Birla Institute of Technology, Mesra, Patna Campus. [1]

A unique automatic parking system by using RFID card and IoT | IEEE Conference Publication | IEEE Xplore

Trong-Yen Lee, Chi-Feng Lee (2012). Microcontroller based automatic parking system. Graduate Institute of Computer and Communication Engineering, National Taipei University of Technology, Taiwan, R.O.C. [2]

Microcontroller based automatic parking system | IEEE Conference Publication | IEEE Xplore M. Swatha , K. Pooja(2018). Smart Car Parking with Monitoring System. UG Student, Department of Computer Science and Engineering Sri Manakula Vinayagar Engineering College, Puducherry.India .[3]

Smart Car Parking with Monitoring System | IEEE Conference Publication | IEEE Xplore

Subaselvi S, Muhin V, Kesava Sai Raam C N, Mohanraj T R. Rfid based automatic car parking system using IoT (2023). Dept. of Electronics and Communication Engineering M. Kumarasamy College of Engineering, Karur, TamilNadu,India. [4]

Rfid Based Automatic Car Parking System Using Iot | IEEE Conference Publication | IEEE Xplore

Lamir Shkurti, Azir Aliu ,Faton Kabashi (2021). ParkingKS: Parking Management System Using Open Automatic License Plate Recognition. Faculty of Contemporary Science and Technologies South East European University Tetovo, North Macedonia. [5]

ParkingKS: Parking Management System Using Open Automatic License Plate Recognition | IEEE Conference Publication | IEEE Xplore

Riad Kanan, Houssam Arbess (2020). An IoT-Based Intelligent System for Real-Time Parking Monitoring and Automatic Billing. College of Engineering and Technology American University of the Middle East Kuwait. [6]

An IoT-Based Intelligent System for Real-Time Parking Monitoring and Automatic Billing | IEEE Conference Publication | IEEE Xplore

Smith, A. (2023). Comparative Analysis of Processing Speed, Connectivity, and Functionality Between Industrial Single-Board Computers and Embedded Microcontroller Boards. Journal of Industrial Electronics [7]

Smart Manufacturing and Intelligent Manufacturing: A Comparative Review - ScienceDirect

Johnson, M. P., & Smith, L. R. (2023). License Plate Recognition for Vehicle Access Control: Image Capture, Processing, and Database Matching. Journal of Intelligent Transportation Systems [8]

A License Plate-Recognition Algorithm for Intelligent Transportation System Applications | IEEE Journals & Magazine | IEEE Xplore

Md. Nazmush Shakib Shahi; Sheikh Samit Muhaimin; Mirza Muntasir Nishat; Fahim Faisal; Mohammad Saifullah Bin Motahar (2022).Automated Vehicle Access and Exit Control: A Smart Parking Management System using Finite State Machine in VHDL [9].

Automated Vehicle Access and Exit Control: A Smart Parking Management System using Finite State Machine in VHDL | IEEE Conference Publication | IEEE Xplore

TCP/IP communication protocol. GeeksforGeeks [10].

TCP/IP: What is TCP/IP and How Does it Work? (techtarget.com)

TCP/IP in Computer Networking - GeeksforGeeks

Visual studio 2022 community edition [11].

Visual Studio 2022 Community Edition – Scarica la versione gratuita più recente (microsoft.com)

Serial communication. RS232, RS485 [12].

What is Serial Communication and How it works? [Explained] (codrey.com)