International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 13, Issue 1 [January. 2024] PP: 90-92

Research and Design of Smart Parking Lots for Cars with Under 9 Seats Using STM 32 Microcontroller.

Hong Tham Tran Thi[1,2], Kim Nhi Vu Thi[1], Trung Hieu Pham[1]
Faculty of Electrical Engineering Technology, Hanoi University of Industry, Hanoi, Vietnam.

Abstract: *Today, with the outstanding development of science, technology and electronics, Internet of Things has been applied in many practical fields to serve the needs of: Increasing convenience, ensuring security and safety. in everyday life. As human life improves, the need to turn the need to control devices in the home into automation that can be remotely controlled and monitored becomes more and more popular and easily met by science and technology. Technology has made many advances. In addition, protecting residential security and fire safety are also emphasized. Starting from the idea and the actual situation, we see this as a good topic with high applicability. The parking lot entry and exit system uses RFID cards for security control.*

Keywords: *Using STM, Smart Parking Lots*

Date of Submission: 05-01-2024 Date of acceptance: 17-01-2024

I. Pose the problem

The smart parking lot model uses the STM32 microcontroller and Arduino UNO R3 as the main processor, controlling peripheral devices and communicating with light sensors and fire sensors. The door opens and closes automatically using the SG90 servo motor. Use the RC522 card to open and close the barrier when entering and exiting. Use light sensors, infrared transceiver sensors to count the number of vacant positions in the parking lot and flame sensors to warn of fires and explosions.

II. Components of Control system.

Components used in the project: - Arduino UNO R3 main processor, - STM32F103C8T6, - Infrared transceiver sensor, - Flame sensor, - Light Sensor, - Display screen: LCD 1602 - SG90 servo motor, - RC522 card to open the door

- LEARN ARDUINO, STM32 MICROPROCESSOR AND COMPONENTS ARDUINO circuit board An Arduino circuit consists of an AVR microcontroller with many additional components that make it easy to program and expandable with other circuits. An important aspect of the Arduino is its standard connections, which allow users to connect the board's CPU to easily swappable add-on modules, called shields. Some shields communicate with the Arduino board directly via different pins, but many shields are addressed via the I2C serial bus - multiple shields can be stacked and used in parallel. Official Arduino often uses the megaAVR chip lines, especially the ATmega8, ATmega168, ATmega328, ATmega1280 and ATmega2560. Several other microprocessors are also used by compatible Arduino boards. Most circuits consist of a 5V linear regulator and a 16 MHz crystal oscillator (or ceramic resonator in some variations), although a few designs such as the LilyPad run at 8 MHz and omit the regulator. Adjust the onboard voltage due to device size limitations. An Arduino microcontroller can also be pre-programmed with a boot loader that allows simply uploading the program to on-chip flash memory, compared to other devices that often require an external loader. This makes using the Arduino more direct by allowing the use of a base computer as a program loader.

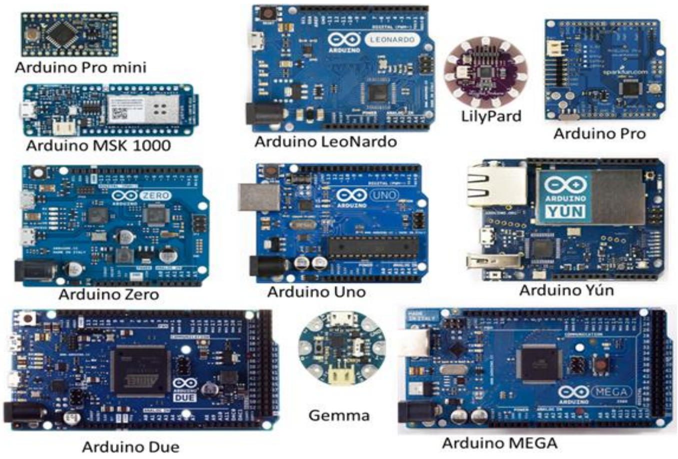
III. System selection and design

Select Mitsubishi E700 inverter type FR-E710W-0.2K Specifications : Power : 0.2kW Rated current : 1.5 A Voltage for inverter: single phase voltage 220V Voltage for 3-phase asynchronous motor: 220VAC Overload capacity :150% for 60s Analog input: 2 ports Analog input signal: 0-10V, 0-5V,4-20mA Digital input: 7 ports Choose a resolution: Engine speed is 2720 rpm. The response rate of the PLC S7-1214C is 100 kHz. There are n = 2720 rpm = (2720/60) = 136/3 (rpm). We choose the resolution for the largest Encoder to measure engine speed will be: (100*1000)/136/3 = 2250 PPR (Pulses Per Revolution)

Choose the resolution for the Encoder as high as possible but not exceed 2250 PPR. So choose an encoder whose code is E6B2-CWZ6C of omron.

Specifications : Resolution : 2000 P/R Output Phase: A,B and Z. Shaft diameter: 6mm Body diameter: 40mm Operating voltage: 5-24VDC Current consumption: max 70mA Frequency response: 100KHz. Standard: IEC

60529 IP50 Cable length: 2m Working temperature: -10 ~ 70C Encoder type : relative Output type: NPN open collector.



- Arduino software The Arduino Integrated Development Environment (IDE) is a crossplatform application written in Java, and from this IDE will be used for the Processing programming language and project Wiring. It is designed for those new to the field of software development. It includes a code editor with functions such as syntax highlighting, automatic brace matching, and automatic alignment, as well as compiling and uploading programs to the board with just one click. A program or code written for Arduino is called a sketch.

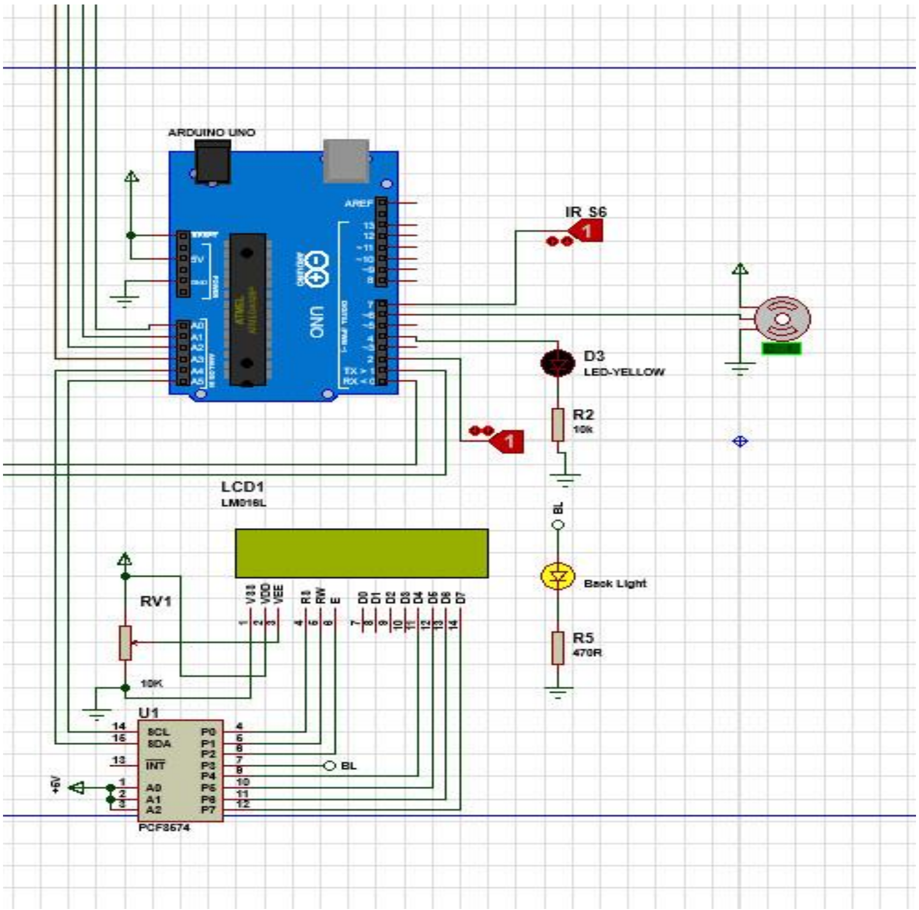
Arduino programs are written in C or C++. The Arduino IDE comes with a software library called “Wiring”, from the original Wiring project, that can make input/output operations easier. Users only need to define two functions to create an executable cyclic executive program: For example: setup(): this function runs every time a program is started, used to set up settings loop(): this function is called repeatedly until the circuit board is powered off. A typical program for a microprocessor Simple control is just making an LED light bulb turn on/off. In the Arduino environment, we will have to write a program like this:

```
#define LED_PIN 13
void setup () {
pinMode (LED_PIN, OUTPUT); // input 13 out  digital
}
void loop ()
{ digitalWrite (LED_PIN, HIGH); //  LED on
delay (1000); // delay (1000 ms)
digitalWrite (LED_PIN, LOW); // off LED off
delay (1000);
}
```

After completing the smart parking model, it will have the following functions:

- Monitor empty position parameters
- Automatic door opening system with RC522 card
- Warning of fire and explosion by horn
- Automatic lighting system when it gets dark
- Send incoming and outgoing vehicle data to the computer.
- A feature of most Arduino boards is that they have an LED and a resistor connecting pin 13 to ground; a convenient feature for many simple applications. The above code cannot be read by a standard C++ compiler as a correct program, so when we click on the "Upload to I/O board" button in this IDE, a copy of this code will be written to a temporary file with an extra include header at the top and a simple main() function at the bottom, to make the body of a C++ program available. This Arduino IDE uses the GNU toolchain and AVR Libc to compile the program, and uses avrdude to upload the program to the board. Since the basis of this Arduino IDE uses the GNU toolchain and AVR Libc to compile Aduino programs which are Atmel microcontrollers, Atmel's development environment, AVR Studio or later versions of Atmel Studio, is also possible. used to make development software for Arduino. 2.3. Introduction to Arduino UNO R3 Arduino UNO R3 has the same function as Arduino Duemilanove but differs in circuit form. Nano is integrated with ATmega328P microcontroller. The main difference between them is that the UNO board comes in PDIP (Plastic Dual-In-line Package) with 30 pins while the Nano is available in TQFP (plastic quad flat pack) with 32 pins. UNO has 6

ADC ports, Nano has 8 ADC ports. This port is used for both programming and the serial monitor. Attractive features of Arduino UNO R3



IV. Conclusion

* Advantage

Some advantages of the system: - Basic solution to the problem of traffic jams, lost parking tickets, and swapped vehicle cards in previous traditional parking lots. - Reduce a lot of time and costs of hiring management and security staff, and earn higher profits. Ability to read and record data multiple times. (When the customer returns the card or no longer uses it, the card can be reused with a new customer). - RFD cards are very durable, can work well in unfavorable environments such as heat, humidity, dust, corrosion, or collision.... creating high reliability when used.

* Limit

However, due to limited knowledge, shortcomings cannot be avoided and there are still limitations in the system. Some limitations of the system: - Implementation costs are higher than previous traditional parking lots. - RFID tags are susceptible to interference in water and metal environments. - Adding, deleting, and recycling cards needs to be done in the control code. - Susceptible to power interference, LCD display errors when operating for a long time or resetting the system.

* Development

In order for the model to develop in the future, the team proposes the following plan:

- Use high precision components to increase stability and ensure the system.
- Combine additional image processing, programming to add driver and license plate recognition functions to the system, enhancing security.
- Combine additional solar panels placed in the parking lot, aiming to use clean and economical energy sources.
- Update data to the cloud to avoid power outages or unexpected system problems.

References

[1]. MSc Le Duc Thuan, "Lecture on Embedded software technology".
[2]. https://www.investopedia.com/terms/r/radio-frequency-identification-rfid.asp.
[3]. https://arduinokit.vn/giao-tiep-i2c-lcd-arduino/.
[4]. Dr. Le My Ha, "Iot programming with Arduino," Thanh Nien Publishing House, 2019.