Project Report on :

Hardware grade 2-factor authentication system.

Maulana Abul Kalam University of Technology

WEST BENGAL

In Partial fulfilment of the requirements for award of degree of

Bachelors of Technology in

ELECTRONICS & COMMUNICATION ENGINEERING

By:

1. Anubhab Palit:	17600321001
2. Ishita Dutta:	17600321002
3. Arpan Biswas:	17600321003
4. Subhrajyoti Mandal:	17600321004

Under the Guidance of

DEB KUMAR SHEET

ASST. PROFESSOR, ECE

HOOGHLY ENGINEERING & TECHNOLOGY COLLEGE

DEPARTMENT OF ELECTRONICS & COMMUNICATIONS ENGINEERING

HOOGHLY ENGINEERING & TECHNOLOGY COLLEGE

VIVEKANANDAD ROAD, PIPULPATI, HOOGHLY, PIN: 712103

2023-2024

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Certificate:

Certified that the project work entitled **HARDWARE GRADE 2-FACTOR AUTHENTICATION SYSTEM** is a bonafide work, carried out by **Mr Anubhab Palit(17600321001), Ms Ishita Dutta(17600321002), Mr Arpan Biswas (17600321003), Mr. Subhrajyoti Mandal (17600321004)** in partial fulfilment for the award of Bachelor of Technology in Electronics & Communications Engineering of Maulana Abul Kalam Azad University of Technology, west Bengal during the year 2023-2024. It is certified that all corrections/suggestions indicated for the internal assessment have been incorporated in the report, deposited in the departmental library. The project has been approved as it satisfies the departmental library. The project has been approved as it satisfies academic requirements in respect of project work, prescribed for the said degree.

________________________	________________________
ASST. PROF.	**TECH. ASST.**
DEB KUMAR SHEET	**SHYAMALI GAYEN**

Acknowledgement

it is a great honor that we are doing project, HARDWARE GRADE 2-FACTOR AUTHENTICATION SYSTEM under the guidance of Asst. Professor Deb Kumar Sheet. We are very thankful to you for your knowledge, you are providing and helping us to complete our project successfully. We are very privileged to have you as our project guide. We thank you once again for your time and effort.

We also thank our college authority (HETC) for providing the opportunity to us.

CONTENT:

Topic: **Page No.**

Abstract:

In today's interconnected world, security concerns have become paramount, especially in digital systems handling sensitive information. Two-factor authentication (2FA) has emerged as a widely adopted security measure to enhance the protection of digital assets. This project proposes the development of a hardware-level two-factor authentication system leveraging a microcontroller, specifically the Raspberry Pi Pico W, along with an RFID (Radio Frequency Identification) reader and a binary encoder.

The system aims to provide an additional layer of security beyond traditional password-based authentication methods by requiring the presentation of both an RFID card/tag and a binary-encoded PIN (Personal Identification Number). The integration of hardware components ensures a robust and tamper-resistant authentication mechanism, suitable for applications requiring stringent security measures.

Introduction:

In contemporary digital environments, the imperative of safeguarding sensitive information against unauthorized access underscores the importance of robust security measures. While traditional password-based authentication mechanisms serve as an initial line of defence, the proliferation of sophisticated cyber threats necessitates the implementation of additional layers of security. Two-factor authentication (2FA) emerges as a widely embraced approach to fortify digital systems, requiring users to present two separate forms of verification before granting access.

In response to this imperative, our project endeavours to enhance the efficacy of two-factor authentication through the amalgamation of advanced hardware components and microcontroller-based processing. Leveraging the Raspberry Pi Pico W microcontroller, an RFID reader module, and a binary encoder, we aim to establish a formidable authentication system that transcends conventional password-centric paradigms.

At its core, our system operates on the principle of dual authentication, wherein users are required to present both an RFID card/tag and a binary-encoded Personal Identification Number (PIN). This multifaceted approach significantly augments the security posture, mitigating the inherent vulnerabilities associated with single-factor authentication methods.

The utilization of cutting-edge hardware components underscores our commitment to fostering robust security solutions that are both accessible and adaptable. By harnessing the power of open-source hardware and software, we endeavour to democratize the implementation of advanced security protocols, empowering individuals and organizations to fortify their digital assets against emerging cyber threats.

In essence, our project represents a pivotal advancement in the realm of digital security, offering a sophisticated yet accessible solution for fortifying authentication mechanisms in diverse contexts. Through the fusion of innovative hardware technologies and meticulous software design, we aspire to set a new standard for secure access control systems, thereby safeguarding the integrity and confidentiality of critical digital assets.

Materials:

- Raspberry Pi Pico W.
- KY-040 Binary encoder.
- RC 522 RFID reader.
- SSD1306 OLED (128*64).
- Relay Modules.
- Jumper wires.
- Power Supply.
- Thonny IDE.

Materials description:

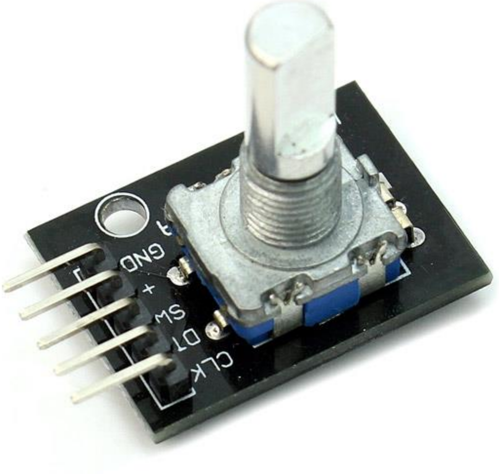
➢ Raspberry Pi Pico W: Raspberry Pi Pico W brings wireless connectivity to the best-selling Raspberry Pi Pico product line. Built around our RP2040 silicon platform, Pico products bring our signature values of high performance, low cost, and ease of use to the microcontroller space. With a large on-chip memory, symmetric dual-core processor complex, deterministic bus fabric, and rich peripheral set augmented with our unique Programmable I/O (PIO) subsystem, RP2040 provides professional users with unrivalled power and flexibility. Offering detailed documentation, a polished MicroPython port, and a UF2 bootloader in ROM, it has the lowest possible barrier to entry for beginner and hobbyist users. RP2040 is manufactured on a modern 40nm process node, delivering high performance, low dynamic power consumption, and low leakage, with a variety of low-power modes to support extended-duration operation on battery power. Raspberry Pi Pico W offers 2.4GHz 802.11 b/g/n wireless LAN support and Bluetooth 5.2, with an on-board antenna, and modular compliance certification. It is able to operate in both station and access-point modes. Full access to network functionality is available to both C and MicroPython developers. Raspberry Pi Pico W pairs RP2040 with 2MB of flash memory, and a power supply chip supporting input voltages from 1.8–5.5V. It provides 26 GPIO pins, three of which can function as analogue inputs, on 0.1”-pitch through-hole pads with castellated edges.

Specification:

- Form factor: 21 mm × 51 mm
- CPU: Dual-core Arm Cortex-M0+ @ 133MHz
- Memory: 264KB on-chip SRAM; 2MB on-board QSPI flash
- Interfacing: 26 GPIO pins, including 3 analogue inputs
- Peripherals: • 2 × UART
    - 2 × SPI controllers
    - 2 × I2C controllers
    - 16 × PWM channels
    - 1 × USB 1.1 controller and PHY, with host and device support
    - 8 × PIO state machines
- Connectivity: 2.4GHz IEEE 802.11b/g/n wireless LAN, on-board antenna
- Bluetooth 5.2

    - Support for Bluetooth LE Central and Peripheral roles
    - Support for Bluetooth Classic
- Input power: 1.8–5.5V DC
- Operating temperature: -20°C to +70°C
- Production lifetime: Raspberry Pi Pico W will remain in production until at least January 2034.

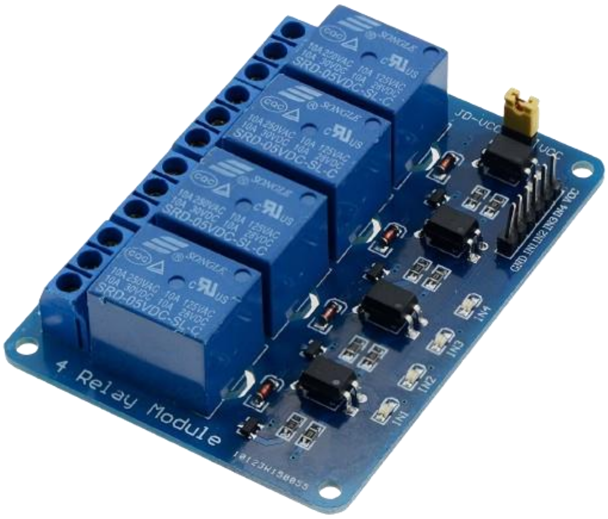
- KY-040 Binary Encoder: The Keyes KY-040 rotary encoder is a rotary input device (as in knob) that provides an indication of how much the knob has been rotated AND what direction it is rotating in. It's a great device for stepper and servo motor control. You could also use it to control devices like digital potentiometers. The module is designed so that a low is output when the switches are closed and a high when the switches are open. The low is generated by placing a ground at Pin C and passing it to the CLK and DT pins when switches are closed. The high is generated with a 5V supply input and pullup resistors, such that CLK This website uses cookies to enhance your experience. By continuing to visit this and DT are both high when switches are open. Not previously mentioned is the existence of push button switch that is integral to the encoder. If you push on the shaft, a normally open switch will close. The feature is useful if you want to change switch function. KY-040 is shown below.



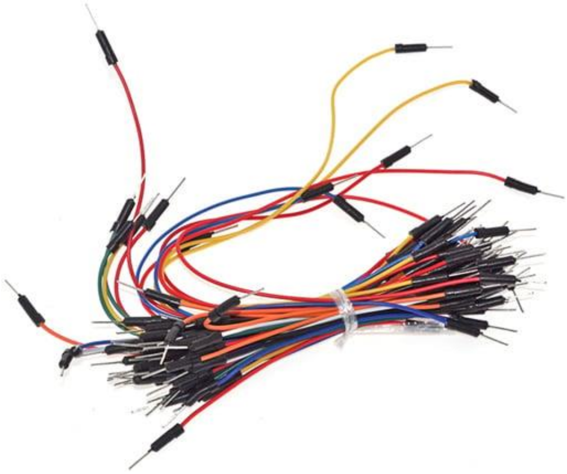
- RC522 RFID Reader: The RC522 is a 13.56MHz RFID module that is based on the MFRC522 controller from NXP semiconductors. The module can supports I2C, SPI and UART and normally is shipped with a RFID card and key fob. It is commonly used in attendance systems and other person/object identification applications. The RC522 is a RF Module that consists of RFID reader, RFID card and a key chain. The module operates 13.56MHz which is industrial (ISM) band and hence can be used without any license problem. The module operates at 3.3V typically and hence commonly used in 3.3V designs. It is normally used in application where certain person/object has to be identified with a unique ID. The keychain has 1kB memory in it which can be used to stored unique data. The RC522 reader module can both read and write data into these memory elements. The reader can read data only form passive tags that operate on 13.56MHz. The RC522 has an operating voltage between 2.5V to 3.3V and hence is normally powered by 3.3V and should be used with 3.3V communication lines. But, the communication pins of this module are 5V tolerant and hence it can be used with 5V microcontrollers also like Arduino without any additional hardware. The module supports SPI, IIC and UART communication but out of these SPI is often used since it is the fasted with a maximum data rate of 10Mbps. Since in application, most of the time reader module will be waiting for the tag to come into proximity. The Reader can be put into power down mode to save power in battery operated applications. This can be achieved by using the IRQ pin on the module. The minimum current consumed by the module during power down mode will be 10uA only. The RC522 module is shown here.

- SSD 1306 OLED: The OLED displays are one of the most attractive displays available for a microcontroller. It has a good view angle and pixel density which makes it reliable for displaying small level graphics. Interfacing this IC with MCU can either be done using IIC or using SPI hence helps to save some pins as well. So if you are looking for a slim, attractive and efficient display module to make your project look cool with graphics. As discussed above, there are many types of OLED displays available in the market the most popular one is the Monochrome 7-pin SSD1306 0.96” OLED display which we are discussing here. This display can support both IIC and SPI communication. When you receive the module from the factory it will be in 4-wire SPI mode by default and it is the fastest of all available modes. However, you can re-solder the resistors in different positions to make it work in 3-Wire SPI and IIC protocol also. Once you have settled on which protocol to use, you should jump into the OLED Display datasheet (given below) to know how to communicate with. The IC supports both 3.3V and 5V logic devices so hardware should not be a problem. A 0.96 inch OLED is shown here.

- Relay Module: Relay is an electromechanical device that uses an electric current to open or close the contacts of a switch. The single-channel relay module is much more than just a plain relay, it comprises of components that make switching and connection easier and act as indicators to show if the module is powered and if the relay is active or not. The relay uses an electric current to open or close the contacts of a switch. This is usually done using the help of a coil that attracts the contacts of a switch and pulls them together when activated, and a spring pushes them apart when the coil is not energized. There are two advantages of this system – First, the current required to activate the relay is much smaller than the current that relay contacts are capable of switching, and second, the coil and the contacts are galvanically isolated, meaning there is no electrical connection between them. This means that the relay can be used to switch mains current through an isolated low voltage digital system like a microcontroller. A 4 pole relay is shown here.

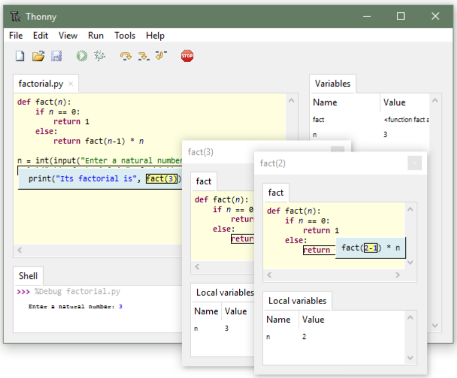


- Jumper wires: Jumpers are tiny metal connectors used to close or open a circuit part. They have two or more connection points, which regulate an electrical circuit board. Their function is to configure the settings for computer peripherals, like the motherboard. Suppose your motherboard supported intrusion detection. A jumper can be set to enable or disable it. Jumper wires are electrical wires with connector pins at each end. They are used to connect two points in a circuit without soldering. You can use jumper wires to modify a circuit or diagnose problems in a circuit. Further, they are best used to bypass a part of the circuit that does not contain a resistor and is suspected to be bad. They are shown below.



- Power Supply: Powering a Raspberry Pi Pico can be achieved through multiple methods, with the most common being via its USB port using a USB power source like a computer USB port, USB wall adapter, or power bank. Alternatively, power can be supplied directly to the Pico's onboard voltage regulator through its Vin pin, by connecting a power source such as a battery pack or external power supply. Another option is to use the 3V3 pin, providing a regulated 3.3V output, suitable for stable 3.3V power supplies like regulated power supplies or 3.3V battery packs. It's crucial to ensure that the power source supplies the correct voltage and sufficient current to meet the Pico's requirements, preventing damage and ensuring stable operation, particularly when connecting peripherals or running power-intensive tasks.

- Thonny IDE: Thonny comes with Python 3.10 built in, so just one simple installer is needed and you're ready to learn programming. (You can also use a separate Python installation, if necessary.) The initial user interface is stripped of all features that may distract beginners.



```
def fact(n):
    if n == 0:
        return 1
    else:
        return fact(n-1) * n

n = int(input("Enter a natural number
    print("Its factorial is", fact(3))
```

Name	Value
fact	<function fact a
n	3

```
>>> %Debug factorial.py
  Enter a natural number: 3
```

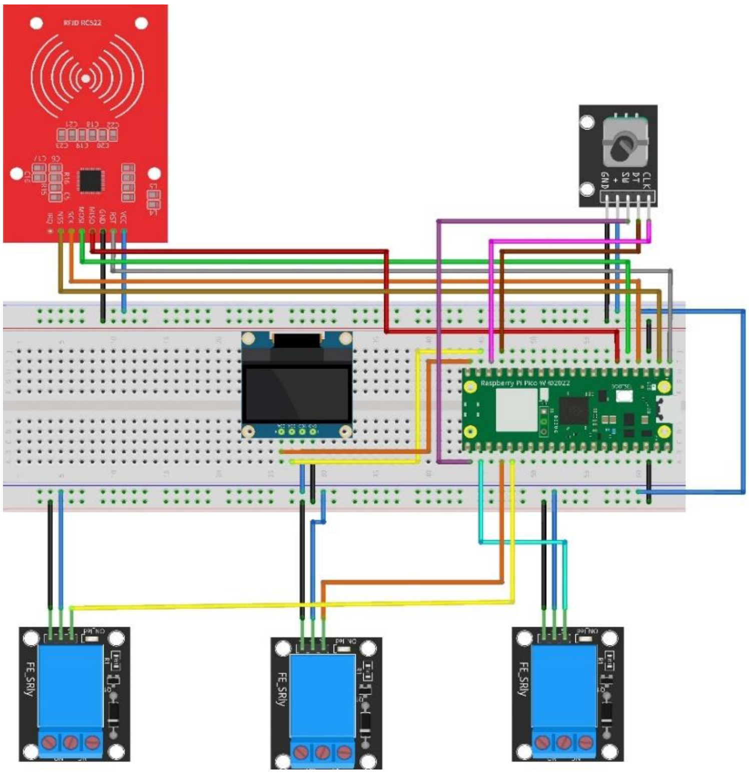
```
def fact(n):
    if n == 0:
        return 1
    else:
        return  fact(2-1) * n
```

Name	Value
n	3

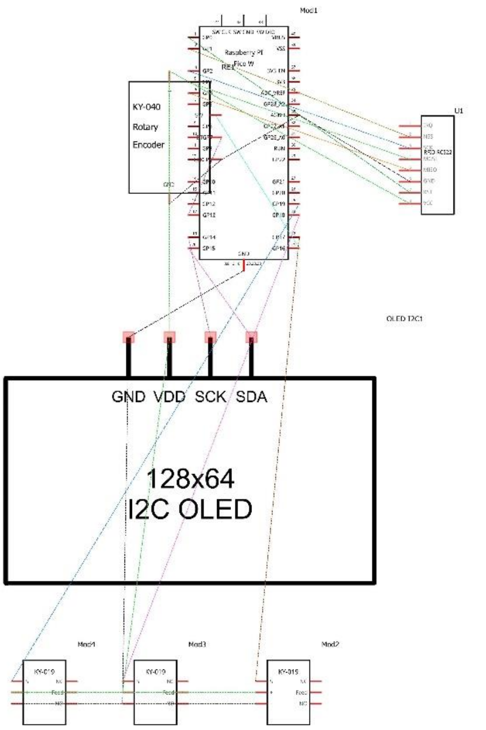
Name	Value
n	2

Circuit diagram:

Here is the circuit diagram of the project with all the connections.



The schematics of the project are given below:



Operation of Circuit:

First we program the circuit using micropython using Thonny IDE. We will also upload the library functions of the required components to the raspberry pi pico. The main code is given below.

CODE:

```
import time
from machine import Pin, SoftI2C
from ssd1306 import SSD1306_I2C
from rotary_irq_esp import RotaryIRQ
from mfrc522 import MFRC522
def setup_pin(pin_number, mode, pull=None):
    return Pin(pin_number, mode, pull)
def setup_rotary_encoder(pin_clk, pin_dt, min_val, max_val):
    return RotaryIRQ(pin_num_clk=pin_clk,
                     pin_num_dt=pin_dt,
                     min_val=min_val,
                     max_val=max_val,
                     incr=1,
                     reverse=False,
                    range_mode=RotaryIRQ.RANGE_BOUNDED)
def get_final_value(digits):
    return int(''.join(map(str, digits)))
def initialize_oled():
    i2c = SoftI2C(scl=Pin(14), sda=Pin(15))
    oled_width = 128
    oled_height = 64
    return SSD1306_I2C(oled_width, oled_height, i2c)
def display_message(oled, message, line=0):
    oled.fill(0)
    oled.text(message, 0, line * 10)
    oled.show()
def validate_rfid_tag(uid):
    users = {
        141106900: (User1', 2003, 17),
        1442825586: ('User2', 2008, 18)
    }
    return users.get(int.from_bytes(bytes(uid), "little", False), None)
reader = MFRC522(spi_id=0, sck=2, miso=4, mosi=3, cs=1, rst=0)
oled = initialize_oled()
display_message(oled, 'Welcome!')
r = setup_rotary_encoder(12, 13, 0, 9)
switch_pin = setup_pin(16, Pin.IN, Pin.PULL_UP)
while True:
    username = None
    pin = None
    gpio_pin = None
    reader.init()
    (stat, tag_type) = reader.request(reader.REQIDL)
    if stat == reader.OK:
```

```
(stat, uid) = reader.SelectTagSN()
if stat == reader.OK:
    card_id = int.from_bytes(bytes(uid), "little", False)
    print("CARD ID:", card_id)
    username_pin_gpio = validate_rfid_tag(uid)
    print("Validation Result:", username_pin_gpio)
    if username_pin_gpio is not None:
        username, pin, gpio_pin = username_pin_gpio
    else:
        display_message(oled, 'Invalid tag', 1)
        time.sleep(1.5)
        continue  # Restart the loop if invalid tag
    display_message(oled, username, 0)
    time.sleep(1.50)
    display_message(oled, 'Enter PIN', 1)
    start_time = time.time()
    digits = [0, 0, 0, 0]
    digit_index = 0
    pin_entry_complete = False  # Flag to indicate PIN entry completion
    while time.time() - start_time < 60:  # Wait for 60 seconds for PIN entry
        value = r.value()
        if value >= 0:
            digits[digit_index] = value
            oled.fill(0)
            oled.text('Entered Pin:', 0, 0)
            oled.text(".join(map(str, digits)), 0, 20)
            oled.show()
            print('Entered Digit {}: {}'.format(digit_index + 1, digits))
        if switch_pin.value() == 0:
            if digit_index < 3:  # Check if digit_index is within range
```

```
                    digit_index += 1
                    print('Switch Pressed - Move to Digit {}'.format(digit_index + 1))
                    time.sleep(0.1)
                else:
                    pin_entry_complete = True
                    break  # Exit the loop to avoid further digit entry
            time.sleep_ms(50)
        if pin_entry_complete:
            entered_pin = get_final_value(digits)
            if entered_pin == pin:
                display_message(oled, 'Success', 3)
                print("Success")
                gpio_pin = setup_pin(gpio_pin, Pin.OUT)  # Initialize GPIO pin for relay
                gpio_pin.value(1)  # Activate the GPIO pin
                time.sleep(2)  # Display success message for 2 seconds
                gpio_pin.value(0)  # Deactivate the GPIO pin
            elif entered_pin == 0:
                continue  # Restart the loop if entered PIN is 0000
            else:
                display_message(oled, 'Wrong PIN entered,\ntry again later', 2)
                time.sleep(2)  # Display message for 2 seconds before restarting
else:
    display_message(oled, 'Welcome!')
```

Now we will be connecting all the components according to the schematics provided above. We will be connecting the Vcc and GND to the ones from the raspberry Pi Pico W. The binary encoder, RFID reader, OLED display and the relays are to be connected.

Operational steps:

- Powering on the system, we are greeted with an welcome message.
- We will have to first tap a registered RFID tag in the reader to proceed. In case an unrecognised card is tapped, there will be a message displayed on the OLED display saying ‘Invalid tag’. If a valid tag is found, the name related to that tag will be displayed for 2seconds and then they will be allowed to enter the pin.

- On successfully entering the tag, you will led to enter the 4 digit numeric Pin using the KY-040 Binary encoder. Rotating the knob clockwise increases the value for the leftmost digit. Pressing on the knob changes to the number on the right.
- On entering a wrong pin, the system resets and you have to start from the beginning.
- On entering the correct pin, the relay registered with that specific card will be triggered, thus allowing access to the person.
- On entering '0 0 0 0', the system will reset. If no pin is entered, the system will reset after 60 seconds.

Result:

Using this setup, we are able to implement a hardware level 2-factor authentication. Adding and removing access is very simple as the function validate_rfid_tag(uid) stores all the tags, their respective pin and the GPIO pin responsible to trigger the respective relay. It is a very simple setup and by using a Raspberry Pi Pico, the project can be contained in a smaller footprint than one used by Arduino. Scaling this project is as easy as a security system can get. All data is securely stored with no external access. To register a new tag, we can use a RFID reader program which displays the entered tag both on the OLED and the output window of Thonny IDE.

Cost estimation:

1. Raspberry Pi Pico W : INR 550
2. Binary Encoder KY-040: INR 40
3. 3D printed Knob for binary Encoder: INR 399
4. 5v relay 4 pole: INR 120
5. RFID reader: INR 100
6. RFID tags: INR 60
7. Jumper wires: INR 20
8. SSD1306 OLED: INR 196
9. Micro USB cable: INR 50

Total cost: INR 1535

Advantages:

The hardware level 2-factor authentication has a lot of advantages. In this system, a microcontroller. Is the main brain of the system. It is responsible to check the entered data and then allow only valid data to be processed. Here are some of the advantages of hardware 2-factor authentication system using Raspberry Pi Pico W.

- ❖ Low Cost: Raspberry Pi Pico is an affordable microcontroller board, making the project cost-effective.
- ❖ Compact Size: The small form factor of Raspberry Pi Pico allows for easy integration into various devices and systems.
- ❖ Scalability: The project can be scaled to accommodate a large number of users and devices without significant hardware or software changes.
- ❖ Customization: The project can be customized to suit specific requirements, such as adding more authentication methods or integrating with different systems.

- Reliability: Hardware-level authentication provides a high level of reliability and security compared to software-based authentication methods.
- Versatility: The project can be adapted for various applications, including access control systems, secure data transfer, and IoT devices.

Future Scope:

- Biometric Authentication: Integration of biometric authentication methods such as fingerprint or facial recognition for enhanced security.
- Multi-Factor Authentication (MFA): Addition of multiple authentication factors, such as something you know (PIN), something you have (RFID card), and something you are (biometric).
- Cloud Integration: Integration with cloud services for centralized user management, logging, and remote access control.
- Secure Communication: Implementation of secure communication protocols (e.g., TLS/SSL) for encrypted data transfer between devices.
- Enhanced User Interface: Development of a user-friendly interface for easier setup and management of authentication credentials.
- Integration with Existing Systems: Integration with existing authentication systems or services, such as LDAP or Active Directory, for seamless user authentication across platforms.
- Audit Logging: Implementation of audit logging functionality to track authentication attempts and user activities for security auditing and compliance purposes.
- Mobile Application: Development of a mobile application for remote authentication and management of devices and users.
- OTA Updates: Implementation of over-the-air (OTA) updates for firmware and security patches to ensure continuous improvement and protection against vulnerabilities.
- Machine Learning and AI: Exploration of machine learning and artificial intelligence techniques for advanced threat detection and anomaly detection in authentication processes.

Conclusion:

In conclusion, the hardware-level 2-factor authentication project utilizing Raspberry Pi Pico presents a compelling solution for bolstering security measures in various contexts. Through the utilization of this compact yet powerful microcontroller board, the project offers a cost-effective and versatile platform for implementing robust authentication mechanisms. The scalability of the Raspberry Pi Pico enables deployment across a spectrum of applications, ranging from home automation systems and access control to industrial machinery and IoT devices.

One of the primary advantages of this project lies in its adaptability to different authentication methods. By combining PIN entry with RFID card validation, it caters to diverse user preferences and operational requirements. Moreover, the modular nature of the project facilitates future enhancements and customization, allowing for the integration of additional authentication factors such as biometric recognition or token-based authentication systems. This adaptability ensures that the project remains relevant and effective in addressing evolving security challenges.

Furthermore, the project's future scope extends beyond its current implementation. There is potential for integrating advanced security features such as biometric authentication using fingerprint sensors or facial recognition technology. Multi-factor authentication, which combines two or more independent authentication factors, could further enhance security by adding an extra layer of verification. Additionally, cloud integration and mobile application development could enable remote access and management of authentication systems, enhancing usability and convenience for end-users.

Overall, the hardware-level 2-factor authentication project serves as a testament to the capabilities of Raspberry Pi Pico and the potential of hardware-based security solutions. Its scalability, adaptability, and future scope make it a compelling choice for organizations and individuals seeking to enhance security measures in their respective domains. As technology continues to evolve, projects like this one will play an increasingly vital role in safeguarding sensitive information, protecting assets, and ensuring the integrity of digital systems.

References:

- ❖ https://github.com/miketeachman/micropython-rotary
- ❖ https://microcontrollerslab.com/raspberry-pi-pico-rfid-rc522-micropython/
- ❖ https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html
- ❖ https://thonny.org/