JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

HUMAN DETECTION SYSTEM USING FINGERPRINT IDENTIFIER

[1]Nishant Zodpe, [2]Omkar More, [3]Rushikesh Pawar, [4]Venkat Ghodke

[1]Student, [2]Student, [3]Student, [4]Professor
[1]Electronics and Telecommunication Engineering,
[1]AISSMS Institute of Information Technology, Pune, Maharashtra

Abstract: In today's world, fingerprints are the most widely and effectively used form of identification. Mostly because of their individuality among the people, public acceptance, originality, life stability, and low risk of invasion. Fingerprint technology, which is essentially a biometric system, is used to identify people based on their physical characteristics. Fingerprint matching is the most popular biometric method for providing authentication. Fingerprint verification is one of the most trustworthy biometric security systems in the computer world. There are various fingerprint qualities available. Some of them have poor, medium, high, and partial fingerprint quality.

Index Terms **– Fingerprint Recognition, Biometrics, Security, Python, Raspberry Pi, Raspbian OS, Proteus Software.**

I. INTRODUCTION

In an educational system, the attendance system is extremely important. Attendance irregularity reduces the student's percentage. This will eventually turn into a problem in student life. Attendance denotes an individual's presence in school, college, or the workplace. The attendance % is currently the most significant issue in the educational system. Only those in attendance are aware of each other's presence. It merely shows the presence and proportion of attendees. We use an automatic mail processing mechanism to ensure faultless attendance. In everyday life, To determine a person's attendance or presence, we use a biometric sensor (iris sensor, thumbprint sensor, brain mapping sensor), such as in or out. Absent attendance in schools leads to despair and, as a result, low educational quality due to time lost when absent. Unruly behaviour may also have a negative impact on one's morale. As a thumbprint sensor, we are using a biometric sensor. A biometric system can recognize both human characteristics and physical objects. It compares the fingerprint to already dumped fingerprint data. The data dump is stored on a Raspberry Pi. It will immediately send to the authorized person's email after finishing the attendance section (attendance time). It will primarily be used for attendance purposes, as well as to save both money and time. So, using the Raspberry Pi, we will create a fingerprint-based attendance system that will make the procedure simple and efficient. To eliminate attendance confusion, we are going to use a biometric sensor and an email. Humans will save time as a result of this.

II. THE OBJECTIVE OF THE SYSTEM

A Human Detection System using a fingerprint identifier typically aims to achieve several objectives related to the identification and verification of individuals based on their fingerprints. Here is a list of common objectives for such a system:

- Identification: The primary objective is to accurately identifying and differentiating people according to their distinct fingerprint patterns.
- Authentication: Verify the fingerprint that is being presented with templates that have been saved in the system's database to confirm the identity of the person.
- Security: Enhance security by ensuring that only authorized individuals gain access to specific areas, systems, or information.
- Accuracy: Achieve a high level of accuracy in fingerprint matching to minimize false positives and negatives.
- Speed: Provide a quick and efficient means of identification to support fast and seamless access control.
- Privacy: Implement measures to protect the privacy of individuals whose fingerprints are stored in the system, adhering to relevant privacy regulations.

III. METHODOLOGY

This project is easy to operate: simply run the Python code, which will display some welcome messages on the LCD before asking the user to place their finger on the fingerprint sensor. We can now verify whether or not our finger prints have already been stored by placing our finger over the fingerprint module. If your fingerprint is stored, the LCD will display a message such as "Fount at Pos:2" along with the fingerprint's storage position; if not, it will display "No Match Found."The user must now press the enroll button and adhere to the instructions displayed on the LCD screen in order to enroll a finger print.The user must press the delete button in order to remove any fingerprints. The LCD will then inquire as to where the fingerprint that needs to be erased is. The user

can now choose the location of the saved fingerprint and press the enroll button (which at this point acts as the Ok button) to remove that fingerprint by using two more push buttons for increment and decrement. View the video that is provided at the conclusion of the project for a better understanding.

IV. BLOCK DIAGRAM

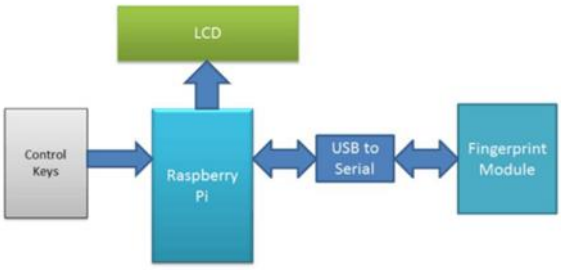

Fig 1. Block Diagram

❖ Our framework's main components are as follows (Figure 1):

- Raspberry Pi B+: Because of its characteristics, this motherboard was chosen over others (Beagle Bone Black, Hummingbird Minnow Board Max, Pengwyn, Hachiko Board, and so on).
- The Fingerprint Scanner GT(511C1R): This device communicates with the Raspberry Pi via a 4-pin connector.
- JST SH Jumper 4 Wire Assembly: This is the Scanner fingerprint connector. It has four pins for connecting voltage (3, 3-6 V), ground, TX and RX of various devices (in our case, the scanner and the Raspberry Pi).
- The Douself Sola magnetic lock: It joins the two formed pieces when it receives a voltage of 12 V. When the voltage drops, the two pieces separate.
- Simple red LED: It is similar to a magnetic lock. When the light is turned on (off), the magnetic system is activated (deactivated).
- Simple switch component: When pressed, it activates the scanner.

V. SOFTWARE REQUIREMENT

1. PYTHON 3-

Python is a general-purpose, high-level programming language. With a strong emphasis on indentation, its design philosophy priorities code readability. Python uses garbage collection and dynamic typing. It is compatible with various programming paradigms, such as object-oriented, functional, and structured (especially procedural). Because of its extensive standard library, it is frequently referred to as a "batteries included" language.

Python's syntax is straightforward and resembles that of English. Python offers a syntax that makes it possible for programmers to write programmers in fewer lines than some other languages. Python is an interpreter-based programming language, which means that code can be run immediately upon writing. Prototyping can therefore be completed very quickly.

Fig 2. Python 3

2. RASPBIAN OS-

For use with the Raspberry Pi line of small single-board computers, Raspberry Pi OS (formerly known as Raspbian) is an operating system that resembles Unix and is based on the Debian Linux distribution. Initially created on its own in 2012, it has been manufactured and distributed by the Raspberry Pi Foundation since 2013 as the main operating system for these boards.

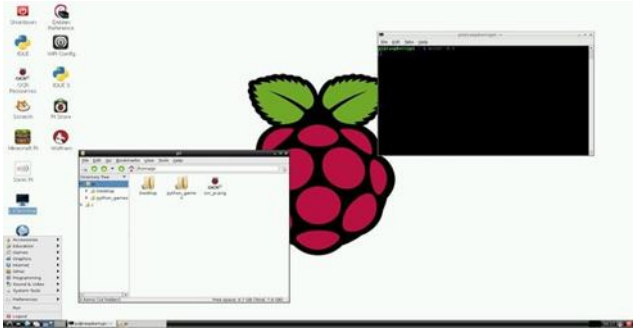

Fig 3. Raspbian OS

3. PROTEUS SOFTWARE-

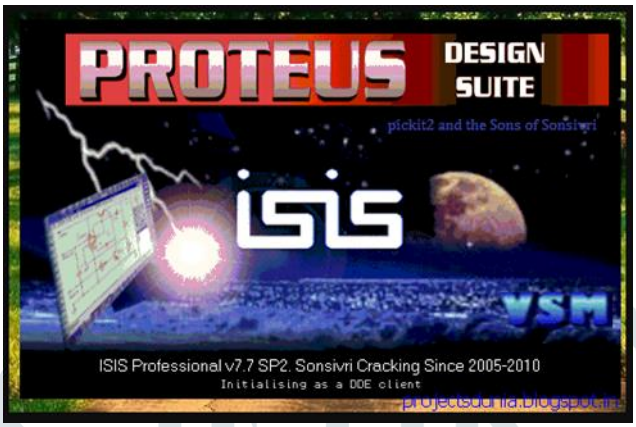

Fig 4. Proteus Software

A. INSTALLATION OF PROTEUS SOFTWARE-

Lab center Electronics created Proteus, it is an electronic design and simulation software. Before assembling their project on hardware, engineering students or hobbyists will design and test it on Proteus software. To install the software called Proteus on Windows, follow the steps below:

Install the Proteus software established from here and run the setup.exe file, as shown in the image below:

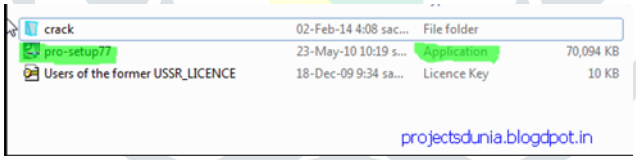

Fig 5. Installation of Proteus Software

After a few minutes, you can install the license key, which should be legitimate for the Proteus software. Select "use a locally installed license key" in this case. One of the two windows is open at the moment. Either the window will have the key already installed, or none at all. To complete the installation, click "next" if a window appears with the key already installed, and then click "end" to close it.

B. SIMULATION RESULT-

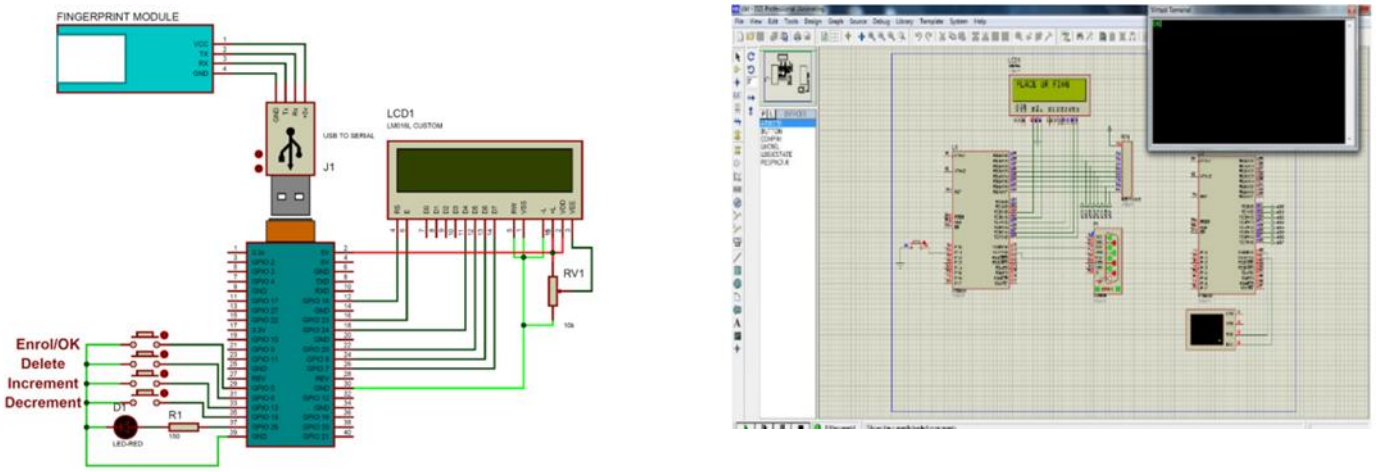

Fig 6&7. Simulation Results

VI. RESULTS AND DISCUSSION

1. First, we run a quick test to see if the sensor has been detected and is ready for use. To do so, run the following command: python2/usr/share/doc/pythonfingerprint/examples/example_index.py A total of 1000 distinct fingerprints can be saved. A finger, for example, can be used multiple times. It can be saved in various positions so that it can be detected more quickly and clearly. By selecting a page (0-3), the following should appear, allowing you to view the positions under which an imprint is stored. Templates currently in use: 0 Please enter the desired index page (0, 1, 2, 3): If the error message "Exception message: The fingerprint sensor port "/dev/ttyUSB0" could not be found!" appears, there is a problem with the cabling or the sensor. Check it once more. Save and print Sample files for storing a new fingerprint, reading out and deleting stored fingerprints are included.

2. Let's start with a finger recording to check the working of system Call: python2/usr/share/doc/pythonfingerprint/examples/example_enroll.py Place your finger on the surface of the glass, wait for a command to appear in the terminal, and remove you're a finger instantly as it does. Following that, you must place your finger once more for verification, and the fingerprint is saved in an additional number. Let's see if our finger is recognised as well. So, take your finger off the sensor and run the following command: python2 /usr/share/doc/python fingerprint/examples/example_search.py Put your finger back on it. If the Raspberry Pi's fingerprint is detected, the following message appears: Templates currently stored: 2 I'm waiting for a finger... The template was discovered at position #1. The accuracy score is 63. The template's SHA-2hashis3aa1b01149abf0a7ad0d7803eaba65c22ba084009700c3c7f5f4ecc38f020851. In this case, a level of precision (the higher the better) is also specified.

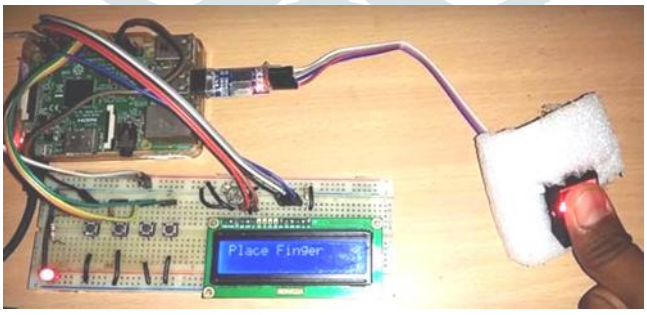

Fig 8. Final Result

VII. CONCLUSION

We presented a low-cost IoT based biometrics architecture. Our low-cost IoT-based biometrics architecture was presented. The Raspberry Pi was used as a remote wireless enrolment node with success. Additionally, the RPi's encryption module operated effectively. The RPi client sent the encrypted biometric traits to the cloud so they could be decrypted.

The proposed system can be used for access control and security features such as managing attendance, opening and closing doors, recording visitors' movements into and out of buildings, and granting access to specific services, among other things. Anywhere that authentication is needed, this system can be used.

VIII. ACKNOWLEDGEMENT

It is our great pleasure in expressing sincere and deep gratitude towards my guide Venkat Ghodke, Professor Electronics & Telecommunication Engineering Department for his valuable guidance and constant support throughout this work and for help in peruse additional studies in Safety and Maintenance by Implementing the Internet of Things (IoT).

We take this opportunity to thank the head of the department DR.M.P. SARDEY and project coordinator Prof. Archana Ubale and all staff members of the Department of electronics & Telecommunication Engineering AISSMS IOIT, Pune. For cooperation is provided by them in many ways. The motivation factor for this work was the inspiration given by our honourable principal DR.P.B. MANE. Lastly, we are thankful to those who have directly or indirectly supported for our work.

REFERENCES

1) Srikanth N, Aishwarya, Kavita H M, Rashmi Reddy K, 'Smart Crop Protection System from Animals and Fire using Arduino' (2019), 6, 4, (17-21).
2) Keerthi Raju, Kamakshi Kodi, Babitha Anapalli, Mounika Pulla, 'Smart Crop Protection System from Animals and Fire using Arduino' (2020), 9, 9, (261-265).
3) Tharun Vignesh N, Saravanan M, Satwik Thiruveedhi, V.S Prabhu, 'Smart Surveillance for Intruder Identification in Forest Areas using Image Processing' (2020), 7, 1, (65-75).
4) Gondal M D, Khan, Y N, 'Early pest detection from crop using image processing and computational intelligence' (2015), FAST-NU Research Journal, 1, 1, (59-68).
5) Navaneetha P, Ramiya Devi R, Vennila S, Manikandan P, Dr. Saravanan S, 'IOT Based Crop Protection System against Birds and Wild Animal Attacks', (2020), International Journal of Innovative Research in Technology, 6, 11, (138-143).
6) Lakshmi K M, Dr Ch Raja, Sreekanth D, Renuka N, 'Security for Protecting Agricultural Crops from Wild Animals using GSM Technology' (2019), Journal of Shanghai Jiaotong University, 7, 2, (724-732).
7) Harsha Vardhini, Koteswaramma N, Babu K. M. C, 'IoT based raspberry pi crop vandalism prevention system' (2019), International Journal of Innovative Technology and Exploring Engineering, 9, 1, (3188-3192).
8) Vignesh Dhandapani, S Remya, T Shanthi, R Vidhy, 'Plant Health Monitoring Using Image Processing' (2018), International Journal of Engineering Research in Computer Science and Engineering, 5, 3, (591-596)
9) Jennifer C. Dela Cruz, Arnold C. Paglinawan, Miguel Isiah R. Bonifacio, Allan Jake D. Flores, Earl Vic B. Hurna, "Biometrics Based Attendance Checking using Principal Component Analysis", School of EECE, Mapúa Institute of Technology Intramuros, Manila, Philippines, 2015
10) Yash Mittal, Aishwary V arshney, Prachi Aggarwal, KapilMatani and V. K. Mittal, "Fingerprint Biometric based Access Control and Classroom Attendance Management System", Indian Institute of Information Technology Chittoor, Sri City, 2015
11) Dr. Vinayak A. Bharadi, V. J. Kaul, SameerAmrutia, "Endto-end Encryption based Biometric SaaS", Thakur College of Engineering and Technology Mumbai, India, 2015
12) Shailendra, Manjot Singh, Md. Alam Khan, Vikram Singh, AvinashPatil, SushmaWadar, "Attendance Management System", Army Institute of Technology, Pune, India