**qCHAPTER 1**

**INTRODUCTION TO EMBEDDED SYSTEMS**

Embedded Technology is now in its prime and the wealth of knowledge available is mind blowing. However, most embedded systems engineers have a common complaint. There are no comprehensive resources available over the internet which deal with the various design and implementation issues of this technology. Intellectual property regulations of many corporations are partly to blame for this and also the tendency to keep technical know-how within a restricted group of researchers.

An embedded computer is frequently a computer that is implemented for a particular purpose. In contrast, an average PC computer usually serves a number of purposes: checking email, surfing the internet, listening to music, word processing, etc... However, embedded systems usually only have a single task, or a very small number of related tasks that they are programmed to perform.

Every home has several examples of embedded computers. Any appliance that has a digital clock, for instance, has a small embedded micro-controller that performs no other task than to display the clock. Modern cars have embedded computers onboard that control such things as ignition timing and anti-lock brakes using input from a number of different sensors.

Embedded computers rarely have a generic interface, however. Even if embedded systems have a keypad and an LCD display, they are rarely capable of using many different types of input or output. An example of an embedded system with I/O capability is a security alarm with an LCD status display, and a keypad for entering a password.

An embedded system can be defined as a control system or computer system designed to perform a specific task. Common examples of embedded systems include MP3 players, navigation systems on aircraft and intruder alarm systems. An embedded system can also be defined as a single purpose computer.

Most embedded systems are time critical applications meaning that the embedded system is working in an environment where timing is very important: the results of an operation are only relevant if they take place in a specific time frame. An autopilot in an aircraft is a time critical embedded system. If the autopilot detects that the plane for some reason is going into a stall then it should take steps to correct this within milliseconds or there would be catastrophic results.

* 1. **APPLICATIONS OF EMBEDDED SYSTEM**

Embedded systems are commonly found in consumer, cooking, industrial, automotive, medical, commercial and military applications.

Telecommunications systems employ numerous embedded systems from [telephone switches](https://en.wikipedia.org/wiki/Telephone_switch) for the network to [cell phones](https://en.wikipedia.org/wiki/Cell_phone) at the end user. Computer networking uses dedicated [routers](https://en.wikipedia.org/wiki/Router_(computing)) and [network bridges](https://en.wikipedia.org/wiki/Network_bridge) to route data.

[Consumer electronics](https://en.wikipedia.org/wiki/Consumer_electronics) include [MP3 players](https://en.wikipedia.org/wiki/MP3_player), mobile phones, [videogame consoles](https://en.wikipedia.org/wiki/Videogame_console), [digital cameras](https://en.wikipedia.org/wiki/Digital_camera), [GPS](https://en.wikipedia.org/wiki/Global_Positioning_System) receivers, and [printers](https://en.wikipedia.org/wiki/Computer_printer). Household appliances, such as [microwave ovens](https://en.wikipedia.org/wiki/Microwave_oven), [washing machines](https://en.wikipedia.org/wiki/Washing_machine) and [dishwashers](https://en.wikipedia.org/wiki/Dishwashers), include embedded systems to provide flexibility, efficiency and features. Advanced [HVAC](https://en.wikipedia.org/wiki/HVAC) systems use networked [thermostats](https://en.wikipedia.org/wiki/Thermostat) to more accurately and efficiently control temperature that can change by time of day and [season](https://en.wikipedia.org/wiki/Season). [Home automation](https://en.wikipedia.org/wiki/Home_automation) uses wired- and wireless-networking that can be used to control lights, climate, security, audio/visual, surveillance, etc., all of which use embedded devices for sensing and controlling.

Transportation systems from flight to automobiles increasingly use embedded systems. New airplanes contain advanced [avionics](https://en.wikipedia.org/wiki/Avionics) such as [inertial guidance systems](https://en.wikipedia.org/wiki/Inertial_guidance_system) and [GPS](https://en.wikipedia.org/wiki/Global_Positioning_System) receivers that also have considerable safety requirements. Various electric motors [brushless DC motors](https://en.wikipedia.org/wiki/Brushless_DC_motor), [induction motors](https://en.wikipedia.org/wiki/Induction_motor) and [DC motors](https://en.wikipedia.org/wiki/DC_motor)  use electric/electronic [motor controllers](https://en.wikipedia.org/wiki/Motor_controller). [Automobiles](https://en.wikipedia.org/wiki/Automobile), [electric vehicles](https://en.wikipedia.org/wiki/Electric_vehicle), and [hybrid vehicles](https://en.wikipedia.org/wiki/Hybrid_vehicle) increasingly use embedded systems to maximize efficiency and reduce pollution. Other automotive safety systems include [anti-lock braking system](https://en.wikipedia.org/wiki/Anti-lock_braking_system) (ABS), [Electronic Stability Control](https://en.wikipedia.org/wiki/Electronic_Stability_Control) (ESC/ESP), [traction control](https://en.wikipedia.org/wiki/Traction_control_system) (TCS) and automatic [four-wheel drive](https://en.wikipedia.org/wiki/Four-wheel_drive).

[Medical equipment](https://en.wikipedia.org/wiki/Medical_equipment) uses embedded systems for [vital signs](https://en.wikipedia.org/wiki/Vital_signs) monitoring, [electronic stethoscopes](https://en.wikipedia.org/wiki/Electronic_stethoscope) for amplifying sounds, and various [medical imaging](https://en.wikipedia.org/wiki/Medical_imaging) ([PET](https://en.wikipedia.org/wiki/Positron_emission_tomography), [SPECT](https://en.wikipedia.org/wiki/Single_photon_emission_computed_tomography), [CT](https://en.wikipedia.org/wiki/Computed_tomography), and [MRI](https://en.wikipedia.org/wiki/Magnetic_resonance_imaging)) for non-invasive internal inspections. Embedded systems within medical equipment are often powered by industrial computers.[[9]](https://en.wikipedia.org/wiki/Embedded_system#cite_note-9)

Embedded systems are used in transportation, fire safety, safety and security, medical applications and life critical systems, as these systems can be isolated from hacking and thus, be more reliable. For fire safety, the systems can be designed to have greater ability to handle higher temperatures and continue to operate. In dealing with security, the embedded systems can be self-sufficient and be able to deal with cut electrical and communication systems.

* 1. **CHARACTERISTICS OF EMBEDDED SYSTEM**

Embedded systems are designed to do some specific task, rather than be a general-purpose computer for multiple tasks. Some also have [real-time](https://en.wikipedia.org/wiki/Real-time_computing) performance constraints that must be met, for reasons such as safety and usability; others may have low or no performance requirements, allowing the system hardware to be simplified to reduce costs.

Embedded systems are not always standalone devices. Many embedded systems consist of small parts within a larger device that serves a more general purpose. For example, the [Gibson Robot Guitar](https://en.wikipedia.org/wiki/Gibson_Robot_Guitar) features an embedded system for tuning the strings, but the overall purpose of the Robot Guitar is, of course, to play music. Similarly, an embedded system in an [automobile](https://en.wikipedia.org/wiki/Automobile) provides a specific function as a subsystem of the car itself.

Embedded systems range from [no user interface](https://en.wikipedia.org/wiki/Headless_system) at all, in systems dedicated only to one task, to complex [graphical user interfaces](https://en.wikipedia.org/wiki/Desktop_operating_system#Graphical_user_interfaces) that resemble modern computer desktop operating systems. Simple embedded devices use [buttons](https://en.wikipedia.org/wiki/Push-button), [LEDs](https://en.wikipedia.org/wiki/LED), graphic or character [LCDs](https://en.wikipedia.org/wiki/LCD) ([HD44780 LCD](https://en.wikipedia.org/wiki/Hitachi_HD44780_LCD_controller) for example) with a simple [menu system](https://en.wikipedia.org/wiki/Menu_(computing)).

More sophisticated devices which use a graphical screen with [touch](https://en.wikipedia.org/wiki/Touch_screen) sensing or screen-edge buttons provide flexibility while minimizing space used: the meaning of the buttons can change with the screen, and selection involves the natural behavior of pointing at what is desired. [Handheld systems](https://en.wikipedia.org/wiki/Mobile_device) often have a screen with a "joystick button" for a pointing device.

Some systems provide user interface remotely with the help of a serial (e.g. [RS-232](https://en.wikipedia.org/wiki/RS-232), [USB](https://en.wikipedia.org/wiki/USB), [I²C](https://en.wikipedia.org/wiki/I%C2%B2C), etc.) or network (e.g. [Ethernet](https://en.wikipedia.org/wiki/Ethernet)) connection. This approach gives several advantages: extends the capabilities of embedded system, avoids the cost of a display, simplifies [BSP](https://en.wikipedia.org/wiki/Board_support_package) and allows one to build a rich user interface on the PC. A good example of this is the combination of an [embedded web server](https://en.wikipedia.org/wiki/Embedded_HTTP_server) running on an embedded device (such as an [IP camera](https://en.wikipedia.org/wiki/IP_camera)) or a [network router](https://en.wikipedia.org/wiki/Router_(computing)). The user interface is displayed in a [web browser](https://en.wikipedia.org/wiki/Web_browser) on a PC connected to the device, therefore needing no software to be installed.

* 1. **PROCESSORS IN EMBEDDED SYSTEMS**

Embedded processors can be broken into two broad categories. Ordinary microprocessors Embedded processors can be broken into two broad categories. Ordinary microprocessors (μP) use separate integrated circuits for memory and peripherals. Microcontrollers (μC) have on-chip peripherals, thus reducing power consumption, size and cost. In contrast to the personal computer market, many different basic [CPU architectures](https://en.wikipedia.org/wiki/CPU_architecture) are used, since software is custom-developed for an application and is not a commodity product installed by the end user. Both [Von Neumann](https://en.wikipedia.org/wiki/Von_Neumann_architecture) as well as various degrees of [Harvard architectures](https://en.wikipedia.org/wiki/Harvard_architecture) are used. [RISC](https://en.wikipedia.org/wiki/RISC) as well as non-RISC processors are found. Word lengths vary from 4-bit to 64-bits and beyond, although the most typical remain 8/16-bit. Most architectures come in a large number of different variants and shapes, many of which are also manufactured by several different companies.

[Numerous microcontrollers](https://en.wikipedia.org/wiki/List_of_common_microcontrollers) have been developed for embedded systems use. General-purpose microprocessors are also used in embedded systems, but generally require more support circuitry than microcontrollers.

(μP) use separate integrated circuits for memory and peripherals. Microcontrollers (μC) have on-chip peripherals, thus reducing power consumption, size and cost. In contrast to the personal computer market, many different basic [CPU architectures](https://en.wikipedia.org/wiki/CPU_architecture) are used, since software is custom-developed for an application and is not a commodity product installed by the end user. Both [Von Neumann](https://en.wikipedia.org/wiki/Von_Neumann_architecture) as well as various degrees of [Harvard architectures](https://en.wikipedia.org/wiki/Harvard_architecture) are used. [RISC](https://en.wikipedia.org/wiki/RISC) as well as non-RISC processors are found. Word lengths vary from 4-bit to 64-bits and beyond, although the most typical remain 8/16-bit. Most architectures come in a large number of different variants and shapes, many of which are also manufactured by several different companies.

[Numerous microcontrollers](https://en.wikipedia.org/wiki/List_of_common_microcontrollers) have been developed for embedded systems use. General-purpose microprocessors are also used in embedded systems, but generally require more support circuitry than microcontrollers.

**1.4 DEBUGGING IN EMBEDDED SYSTEMS**

Embedded [debugging](https://en.wikipedia.org/wiki/Debugging) may be performed at different levels, depending on the facilities available. The different metrics that characterize the different forms of embedded debugging are: does it slow down the main application, how close is the debugged system or application to the actual system or application, how expressive are the triggers that I can set for debugging (e.g., I want to inspect the memory when a particular [program counter](https://en.wikipedia.org/wiki/Program_counter) value is reached), and what can I inspect in the debugging process (such as, only memory, or memory and registers, etc.).

From simplest to most sophisticated they can be roughly grouped into the following areas:

Interactive resident debugging, using the simple shell provided by the embedded operating system (e.g. Forth and Basic)

External debugging using logging or serial port output to trace operation using either a monitor in flash or using a debug server like the [Remedy Debugger](https://en.wikipedia.org/wiki/Remedy_Debugger) which even works for heterogeneous [multicore](https://en.wikipedia.org/wiki/Multi-core_processor) systems.

An in-circuit debugger (ICD), a hardware device that connects to the microprocessor via a [JTAG](https://en.wikipedia.org/wiki/JTAG) or [Nexus](https://en.wikipedia.org/wiki/Nexus_(standard)) interface. This allows the operation of the microprocessor to be controlled externally, but is typically restricted to specific debugging capabilities in the processor.

An [in-circuit emulator](https://en.wikipedia.org/wiki/In-circuit_emulator) (ICE) replaces the microprocessor with a simulated equivalent, providing full control over all aspects of the microprocessor.

A complete [emulator](https://en.wikipedia.org/wiki/Emulator) provides a simulation of all aspects of the hardware, allowing all of it to be controlled and modified, and allowing debugging on a normal PC. The downsides are expense and slow operation, in some cases up to 100 times slower than the final system.

For SoC designs, the typical approach is to verify and debug the design on an FPGA prototype board. Tools such as Certus[[11]](https://en.wikipedia.org/wiki/Embedded_system#cite_note-12) are used to insert probes in the FPGA RTL that make signals available for observation. This is used to debug hardware, firmware and software interactions across multiple FPGA with capabilities similar to a logic analyzer.

Unless restricted to external debugging, the programmer can typically load and run software through the tools, view the code running in the processor, and start or stop its operation. The view of the code may be as [HLL](https://en.wikipedia.org/wiki/High-level_programming_language) [source-code](https://en.wikipedia.org/wiki/Source-code), [assembly code](https://en.wikipedia.org/wiki/Assembly_code) or mixture of both.

Because an embedded system is often composed of a wide variety of elements, the debugging strategy may vary. For instance, debugging a software- (and microprocessor-) centric embedded system is different from debugging an embedded system where most of the processing is performed by peripherals (DSP, FPGA, and co-processor). An increasing number of embedded systems today use more than one single processor core. A common problem with multi-core development is the proper synchronization of software execution. In such a case, the embedded system design may wish to check the data traffic on the busses between the processor cores, which requires very low-level debugging, at signal/bus level, with a [logic analyzer](https://en.wikipedia.org/wiki/Logic_analyzer), for instance.

**1.5 RELIABILITY**

Embedded systems often reside in machines that are expected to run continuously for years without errors, and in some cases recover by themselves if an error occurs. Therefore, the software is usually developed and tested more carefully than that for personal computers, and unreliable mechanical moving parts such as disk drives, switches or buttons are avoided.

Specific reliability issues may include:

* The system cannot safely be shut down for repair, or it is too inaccessible to repair. Examples include space systems, undersea cables, navigational beacons, bore-hole systems, and automobiles.
* The system must be kept running for safety reasons. "Limp modes" are less tolerable. Often backups are selected by an operator. Examples include aircraft navigation, reactor control systems, safety-critical chemical factory controls, train signals.
* The system will lose large amounts of money when shut down: Telephone switches, factory controls, bridge and elevator controls, funds transfer and market making, automated sales and service.

A variety of techniques are used, sometimes in combination, to recover from errors—both software bugs such as [memory leaks](https://en.wikipedia.org/wiki/Memory_leak), and also [soft errors](https://en.wikipedia.org/wiki/Soft_error) in the hardware:

* [watchdog timer](https://en.wikipedia.org/wiki/Watchdog_timer) that resets the computer unless the software periodically notifies the watchdog subsystems with redundant spares that can be switched over to software "limp modes" that provide partial function
* Designing with a [Trusted Computing Base](https://en.wikipedia.org/wiki/Trusted_Computing_Base) (TCB) architecture[[12]](https://en.wikipedia.org/wiki/Embedded_system#cite_note-13) ensures a highly secure & reliable system environment
* A [hypervisor](https://en.wikipedia.org/wiki/Hypervisor) designed for embedded systems, is able to provide secure encapsulation for any subsystem component, so that a compromised software component cannot interfere with other subsystems, or privileged-level system software. This encapsulation keeps faults from propagating from one subsystem to another, improving reliability. This may also allow a subsystem to be automatically shut down and restarted on fault detection.
* [Immunity Aware Programming](https://en.wikipedia.org/wiki/Immunity_Aware_Programming)

**1.6 TRACING**

Real-time operating systems ([RTOS](https://en.wikipedia.org/wiki/RTOS)) often supports [tracing](https://en.wikipedia.org/wiki/Tracing_(software)) of operating system events. A graphical view is presented by a host PC tool, based on a recording of the system behavior. The trace recording can be performed in software, by the RTOS, or by special tracing hardware. RTOS tracing allows developers to understand timing and performance issues of the software system and gives a good understanding of the high-level system behaviors. Commercial tools like [RTXC Quadros](https://en.wikipedia.org/wiki/RTXC_Quadros) or [IAR Systems](https://en.wikipedia.org/wiki/IAR_Systems) exists.

**CHAPTER 2**

**FINGERPRINT BASED EXAM HALL AUTHENTICATION SYSTEM**

**2.1 INTRODUCTION**

Security is a major concern in our day to day life, and digital locks have become an important part of these security systems. There are many types of security systems available to secure our place. Some examples are PIR based Security System, RFID based Security System, Digital Lock System, bio-matrix systems, Electronics Code lock. In this post, we will Interface a Fingerprint Sensor Module with Arduino and will build a Fingerprint based Biometric Security System with door locking. Finger Print is considered one of the safest key to lock or unlock any system as it can recognize any person uniquely and can’t be copied easily.

In this Arduino Fingerprint Sensor Project, we have used Fingerprint Sensor Module to take finger or thumb impression as input in the system. Here we are using 4 push buttons to Enroll/back, Delete/OK, UP and Down. Every key has double features. Enroll key is used for enrolling new finger impression into the system and back function as well. Means when the user wants to enroll new finger then he/she needs to press enroll key then LCD asks for the ID or Location where user wants to store the finger print output. Now if at this time user do not want to proceed further then he/she can press enroll key again to go back (this time enroll key behave as Back key). Means enroll key has both enrollment and back function. DEL/OK key also has same double function like when user enrolls new finger then he/she need to select finger ID or Location by using another two key namely UP/MATCH AND DOWN/MATCH (which also has double function) now user needs to press DEL/OK key (this time this key behaves like OK) to proceed with selected ID or Location. UP/DOWN keys also support Finger print match function. Check the Video at the end for full demonstration.

**ADVANTAGES**

A fingerprint-based exam hall authentication system offers several advantages in the context of educational institutions and examination management. Here are the key benefits of implementing such a system:

1. \*\*Enhanced Security\*\*: The primary advantage is the heightened security it provides. Fingerprint recognition is highly reliable and nearly impossible to forge, ensuring that only authorized students are granted access to the exam hall.

2. \*\*Prevention of Impersonation\*\*: It effectively eliminates the risk of students impersonating others during exams, a common form of academic dishonesty.

3. \*\*Accuracy\*\*: Fingerprint recognition is precise, ensuring that the identity of the student is accurately verified, reducing the possibility of administrative errors in the authentication process.

4. \*\*Time Efficiency\*\*: The authentication process is quick and efficient, minimizing the time required for entry into the exam hall and reducing the risk of delays or disruptions.

5. \*\*Reduction in Cheating\*\*: The system deters cheating by ensuring that students enter the exam hall under their own identities, which discourages the use of proxy test-takers.

6. \*\*Streamlined Check-In Process\*\*: Fingerprint-based authentication simplifies the check-in process, reducing administrative workload and the need for manual verification of student identities.

7. \*\*Data Logging\*\*: The system can log attendance data, creating a digital record of who entered the exam hall and at what time, which can be useful for audit and security purposes.

8. \*\*Non-Transferable\*\*: Fingerprint information is unique to each individual and cannot be shared or transferred, ensuring the exam is taken by the intended student.

9. \*\*User-Friendly\*\*: Fingerprint authentication is user-friendly, and most students are familiar with the process, making it easy for them to use.

10. \*\*Deterrent to Unauthorized Entry\*\*: The presence of a fingerprint-based system can act as a deterrent to individuals who may attempt to enter the exam hall without authorization.

11. \*\*Reduced Administrative Burden\*\*: Faculty and staff can focus on proctoring the exam and maintaining a secure testing environment, as the system automates the authentication process.

12. \*\*Accessibility\*\*: Fingerprint authentication is inclusive and accessible to students with disabilities who may face challenges with traditional identification methods.

13. \*\*Remote Monitoring\*\*: Some systems allow remote monitoring, enabling exam administrators to oversee the authentication process and address any issues in real-time.

14. \*\*Cost-Effective\*\*: Over time, the system can be cost-effective as it reduces the need for physical identification cards, which may need to be reissued or replaced.

15. \*\*Scalability\*\*: The system can be easily scaled to accommodate a larger number of students, making it suitable for institutions with varying enrollment sizes.

16. \*\*Contactless\*\*: Fingerprint-based systems can be contactless, reducing the risk of spreading contagious diseases in crowded exam halls.

In summary, a fingerprint-based exam hall authentication system offers a secure, efficient, and user-friendly method of verifying student identities, promoting academic integrity, and streamlining the exam administration process in educational institutions. It contributes to a more trustworthy and effective examination environment.

**APPLICATIONS**

Fingerprint-based exam hall authentication systems find applications in educational institutions and examination management to enhance security, streamline processes, and maintain the integrity of academic assessments. Here are the key applications of such a system:

1. \*\*Student Identity Verification\*\*: The primary application is to verify the identity of students before they enter the exam hall. Fingerprint recognition ensures that the person taking the exam is the registered student.

2. \*\*Prevention of Impersonation\*\*: These systems prevent students from impersonating others, which is a common form of academic dishonesty, ensuring that only authorized individuals take the exam.

3. \*\*Examination Halls\*\*: Fingerprint authentication is used to grant access to examination halls or test centers, ensuring that only registered students are allowed to enter.

4. \*\*Elimination of Physical ID Cards\*\*: Fingerprint-based systems can eliminate the need for physical identification cards, reducing the risk of card sharing, loss, or fraud.

5. \*\*Exam Attendance Tracking\*\*: These systems can record and log student attendance, creating a digital record of who entered the exam hall and when. This information can be used for audit and security purposes.

6. \*\*Remote Proctoring\*\*: Some systems allow for remote monitoring, enabling exam administrators to oversee the authentication process and address any issues in real-time from a remote location.

7. \*\*Alternate Exam Scheduling\*\*: The system can be used for scheduling alternate exams or makeup exams, ensuring that only eligible students are allowed to take these assessments.

8. \*\*Secure Testing Environments\*\*: Fingerprint-based authentication contributes to maintaining secure testing environments, preventing unauthorized entry and cheating.

9. \*\*Online Examinations\*\*: In the context of online exams, fingerprint authentication can be used to verify the identity of students before they begin the exam, reducing the risk of cheating.

10. \*\*Special Accommodations\*\*: Fingerprint systems can be adapted to accommodate students with disabilities who may have difficulty with traditional identification methods, ensuring inclusivity.

11. \*\*Proctored Remote Exams\*\*: Fingerprint authentication is applicable to proctored remote exams, ensuring that students taking exams from off-site locations are accurately identified.

12. \*\*Staff Authentication\*\*: These systems can be used to authenticate examination staff, proctors, and administrators to ensure that only authorized personnel are involved in the examination process.

13. \*\*Audit and Reporting\*\*: Fingerprint authentication systems generate audit trails and reports, which can be used for analysis and reporting on exam attendance and security.

14. \*\*Controlled Access Areas\*\*: Fingerprint systems can control access to areas with secure exam materials, reducing the risk of theft or tampering.

15. \*\*Integration with Student Records\*\*: The system can be integrated with student records and exam schedules, ensuring that only eligible students are allowed to take specific exams.

16. \*\*Multiple Authentication Levels\*\*: Some systems offer the flexibility of different authentication levels, allowing administrators to set access rules based on the nature and importance of the exam.

17. \*\*Research and Analysis\*\*: Data collected by fingerprint systems can be used for research and analysis of student attendance patterns and examination security.

In conclusion, fingerprint-based exam hall authentication systems are versatile tools that enhance the security and efficiency of examination processes in educational institutions. These systems are valuable in maintaining academic integrity, ensuring the accuracy of student identity verification, and improving the overall administration of exams.

**FUTURESCOPE**

The future scope of fingerprint-based exam hall authentication systems is promising, with ongoing technological advancements and evolving educational needs. Here are some key areas where we can expect to see developments and innovations in the coming years:

1. \*\*Multi-Modal Biometrics\*\*: Future systems may integrate multiple biometric modalities, such as facial recognition and palm vein scanning, for even more robust identity verification and enhanced security.

2. \*\*AI and Machine Learning\*\*: Integration of AI and machine learning algorithms can improve the accuracy and speed of fingerprint recognition, making the authentication process even more efficient and reliable.

3. \*\*Remote Proctoring\*\*: Enhanced remote proctoring capabilities may allow for real-time monitoring of students taking exams from remote locations, ensuring that the student's identity is verified and the examination environment remains secure.

4. \*\*Blockchain Integration\*\*: The use of blockchain technology can provide a secure and tamper-proof record of exam attendance and authentication, enhancing the trustworthiness of the process.

5. \*\*Scalability\*\*: Fingerprint systems will become more scalable, accommodating larger numbers of students and exam halls, making them suitable for educational institutions of all sizes.

6. \*\*Integration with Online Learning Platforms\*\*: Integration with online learning platforms and learning management systems can streamline the authentication process for online exams, ensuring that students are accurately identified.

7. \*\*Behavioral Biometrics\*\*: Incorporation of behavioral biometrics, such as keystroke dynamics and mouse movements, can complement fingerprint recognition to enhance security and detect anomalies.

8. \*\*Global Accessibility\*\*: Fingerprint systems will be designed with international standards in mind, ensuring that they are accessible and compliant with various regulations and standards worldwide.

9. \*\*Enhanced Privacy and Data Protection\*\*: Stricter data protection measures will be implemented to safeguard biometric data and comply with evolving privacy regulations.

10. \*\*User-Friendly Design\*\*: Future systems will focus on user-friendliness, making the authentication process intuitive and comfortable for students, examiners, and administrators.

11. \*\*E-Exams and Remote Learning\*\*: The technology will play a crucial role in facilitating e-exams and secure remote learning, ensuring that online assessments are trustworthy and secure.

12. \*\*Adaptive Authentication\*\*: Systems may incorporate adaptive authentication mechanisms, adjusting the level of security based on the context and sensitivity of the exam.

13. \*\*Real-Time Notifications\*\*: Students and administrators may receive real-time notifications and alerts, ensuring transparency and responsiveness during the authentication process.

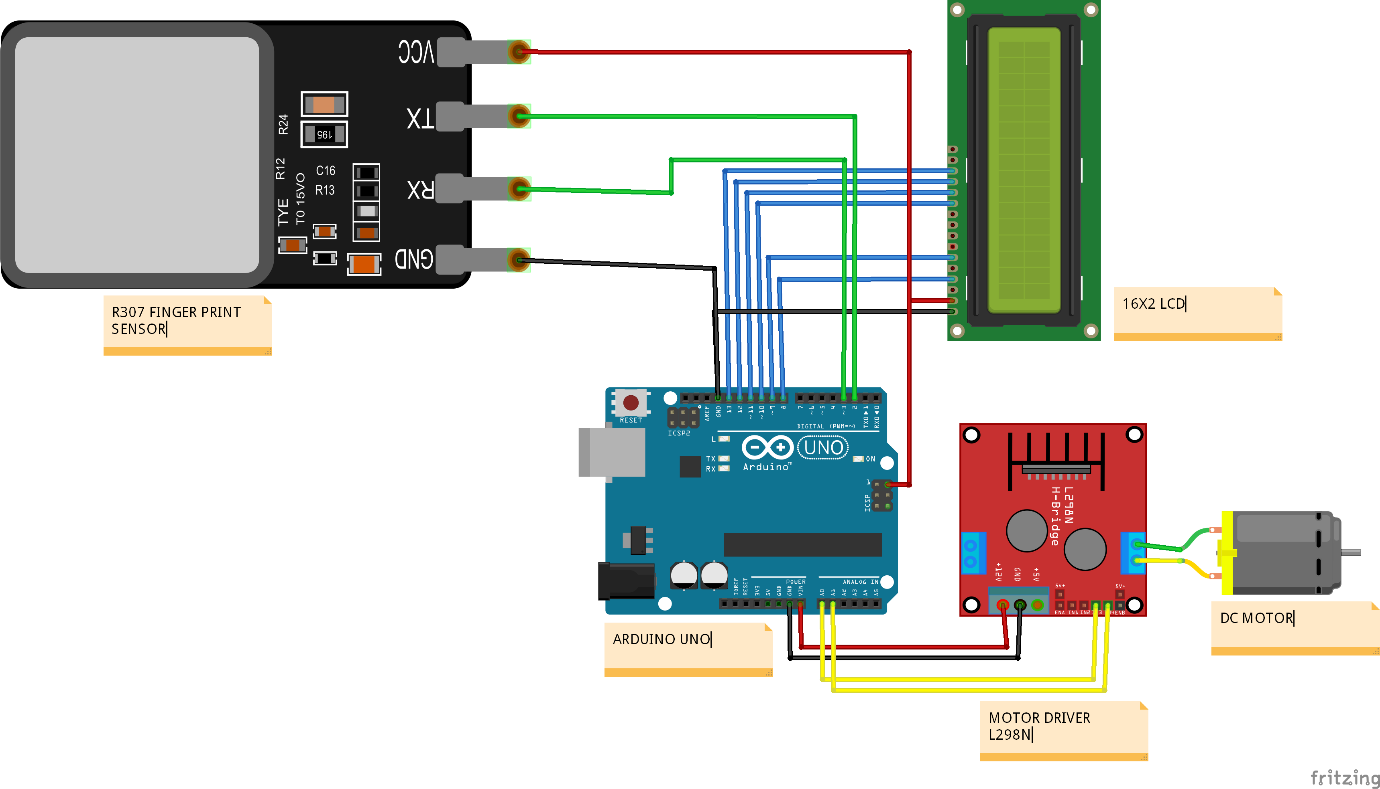
14. \*\*Voice and Audio Recognition\*\*: Fingerprint systems may expand to include voice and audio recognition, enhancing the biometric verification process and supporting students with disabilities.

15. \*\*Emergency Response Integration\*\*: The technology may include features for emergency response, allowing authorized personnel to quickly gain access to the exam hall in case of medical emergencies or other urgent situations.

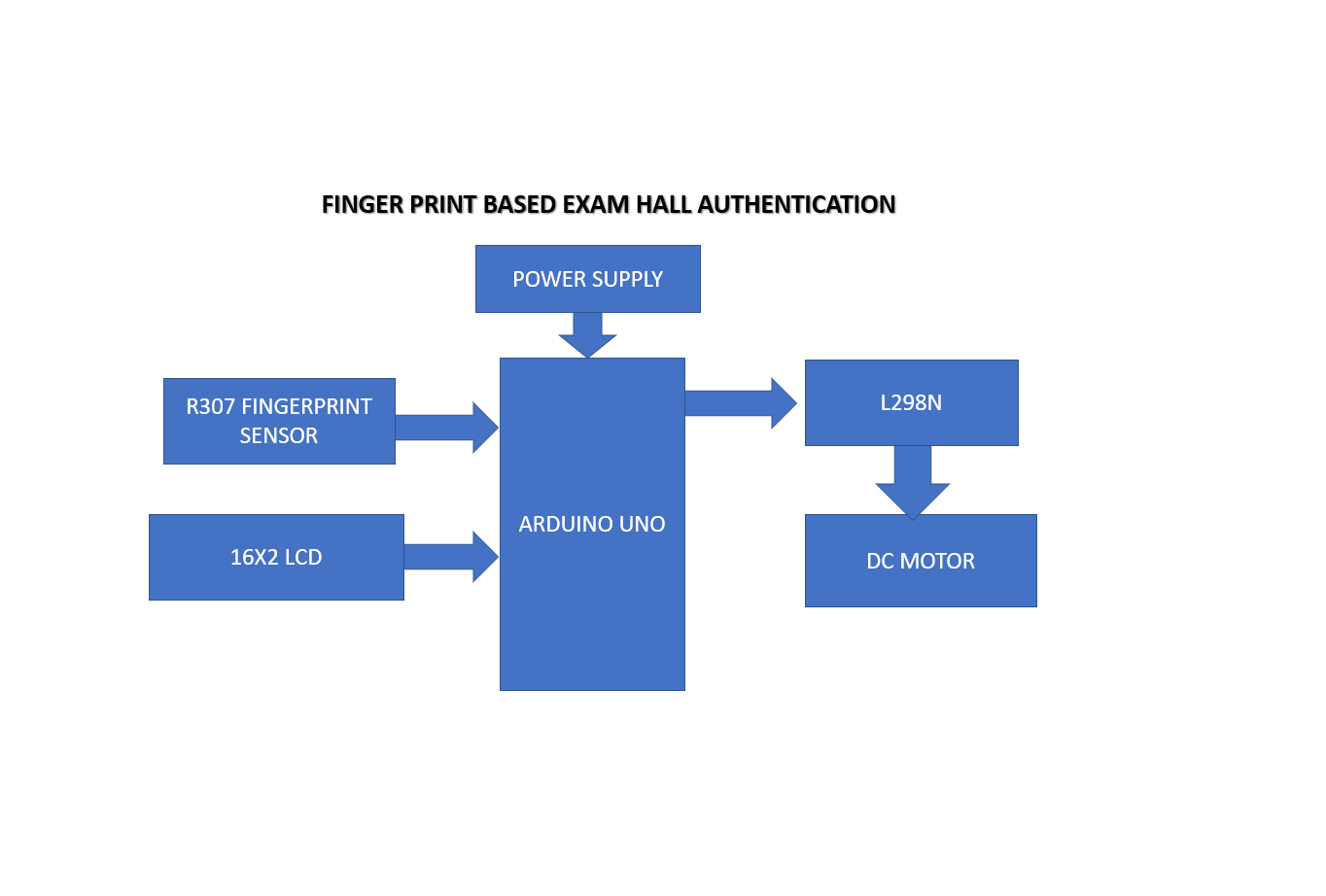
16. \*\*Continuous Improvement\*\*: Continuous feedback and monitoring will drive ongoing improvements to the technology, ensuring it remains reliable and relevant in the changing educational landscape.

In conclusion, the future of fingerprint-based exam hall authentication systems is marked by continuous technological innovation, enhanced security features, and expanded applications. These systems are poised to play an increasingly vital role in maintaining the integrity of examinations in educational institutions, whether on-campus or in remote learning environments.

**2.2 CIRCUIT DIAGRAM**

****

**BLOCK DIAGRAM**

****

**CHAPTER 3**

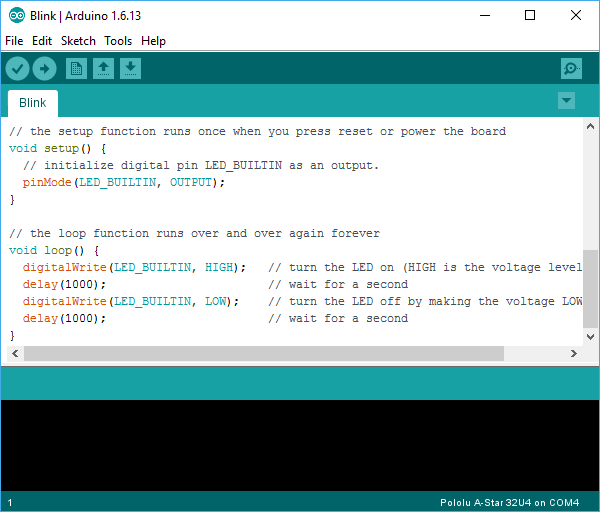
**SOFTWARE REQUIREMENTS**

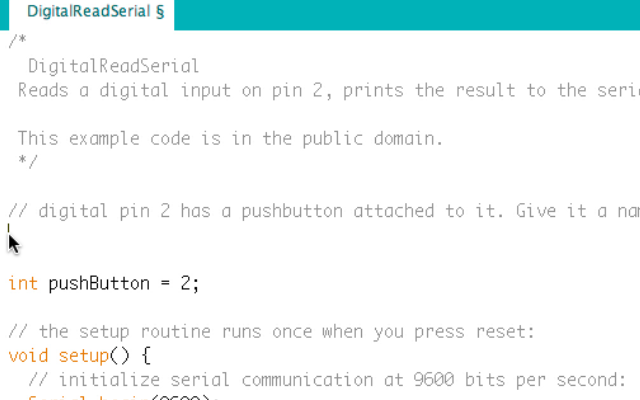
Software used in this project for uploading code onto Arduino is Arduino IDE.

**3.1 INTRODUCTION TO ARDUINO IDE**

IDE stands for Integrated Development Environment. Pretty fancy sounding, and should make you feel smart any time you use it. The IDE is a text editor-like program that allows you to write Arduino code. When you open the Arduino program, you are opening the IDE. It is intentionally streamlined to keep things as simple and straightforward as possible. When you save a file in Arduino, the file is called a sketch – a sketch is where you save the computer code you have written. The coding language that Arduino uses is very much like C++ (“see plus plus”), which is a common language in the world of computing. The code you learn to write for Arduino will be very similar to the code you write in any other computer language – all the basic concepts remain the same – it is just a matter of learning a new dialect should you pursue other programming languages.



The code you write is “human readable”, that is, it will make sense to you (sometimes), and will be organized for a human to follow. Part of the job of the IDE is to take the human readable code and translate it into machine-readable code to be executed by the Arduino. This process is called compiling. The process of compiling is seamless to the user. All you have to do is press a button. If you have errors in your computer code, the compiler will display an error message at the bottom of the IDE and highlight the line of code that seems to be the issue. The error message is meant to help you identify what you might have done wrong – sometimes the message is very explicit, like saying, “Hey – you forget a semicolon”, sometimes the error message is vague. Why be concerned with a semicolon you ask? A semicolon is part of the Arduino language syntax, the rules that govern how the code is written. It is like grammar in writing. Say for example we didn’t use periods when we wrote – everyone would have a heck of a time trying to figure out when sentences started and ended. Or if we didn’t employ the comma, how would we convey a dramatic pause to the reader?

And let me tell you, if you ever had an English teacher with an overactive red pen, the compiler is ten times worse. In fact – your programs WILL NOT compile without perfect syntax. This might drive you crazy at first because it is very natural to forget syntax. As you gain experience programming you will learn to be assiduous about coding grammar.

**3.1.1 THE SEMICOLON**

A semicolon needs to follow every statement written in the Arduino programming language. For example, …

Int LedPin=9;

In this statement, I am assigning a value to an integer variable (we will cover this later), notice the semicolon at the end. This tells the compiler that you have finished a chunk of code and are moving on to the next piece. A semicolon is to Arduino code, as a period is to a sentence. It signifies a complete statement.

### 3.1.2 THE DOUBLE BACKSLASH FOR SINGLE LINE COMMENTS //

### Comments are what you use to annotate code. Good code is commented well. Comments are meant to inform you and anyone else who might stumble across your code, what the heck you were thinking when you wrote it. A good comment would be something like this…

Now, in 3 months when I review this program, I know where to stick my LED. Comments will be ignored by the compiler – so you can write whatever you like in them. If you have a lot you need to explain, you can use a multi-line comment, shown below…

//This is an example

Comments are like the footnotes of code, except far more prevalent and not at the bottom of the page.

**3.1.3 THE CURLY BRACES**

Curly braces are used to enclose further instructions carried out by a function (we discuss functions next). There is always an opening curly bracket and a closing curly bracket. If you forget to close a curly bracket, the compiler will not like it and throw an error code.

Void loop(){

}

Remember – no curly brace may go unclosed!

**3.1.4 FUNCTION ( )**

Functions are pieces of code that are used so often that they are encapsulated in certain keywords so that you can use them more easily. For example, a function could be the following set of instructions…

This set of simple instructions could be encapsulated in a function that we call WashDog. Every time we want to carry out all those instructions we just type WashDog and voila – all the instructions are carried out. In Arduino, there are certain functions that are used so often they have been built into the IDE. When you type them, the name of the function will appear orange. The function pinMode(), for example, is a common function used to designate the mode of an Arduino pin.

What’s the deal with the parentheses following the function pinMode? Many functions require *arguments* to work. An argument is information the function uses when it runs. For our WashDog function, the arguments might be dog name and soap type, or temperature and size of a bucket.

pinMode(13,OUTPUT);

The argument 13 refers to pin 13, and OUTPUT is the mode in which you want the pin to operate. When you enter these arguments the terminology is called passing. You pass the necessary information to the functions. Not all functions require arguments, but opening and closing parentheses will stay regardless though empty.

Notice that the word OUTPUT is blue. There are certain keywords in Arduino that are used frequently and the color blue helps identify them. The IDE turns them blue automatically. Now we won’t get into it here, but you can easily make your own functions in Arduino, and you can even get the IDE to color them for you. We will, however, talk about the two functions used in nearly EVERY Arduino program.

**3.1.5 VOID SETUP ( )**

The function, setup(), as the name implies, is used to set up the Arduino board. The Arduino executes all the code that is contained between the curly braces of setup() only once. Typical things that happen in setup() are setting the modes of pins, starting You might be wondering what void means before the function setup(). Void means that the function does not return information. Some functions do return values – our DogWash function might return the number of buckets it required to clean the dog. The function analogRead() returns an integer value between 0-1023. If this seems a bit odd now, don’t worry as we will cover every common Arduino function in depth as we continue the course.

Let us review a couple things you should know about setup()…

1. setup() only runs once.

2. setup() needs to be the first function in your Arduino sketch.

3. setup() must have opening and closing curly braces.

**3.1.6 VOID LOOP ( )**

You have to love the Arduino developers because the function names are so telling. As the name implies, all the code between the curly braces in loop() is repeated over and over again – in a loop. The loop() function is where the body of your program will reside. As with setup(), the function loop() does not return any values, therefore the word void precedes it.

Does it seem odd to you that the code runs in one big loop? This apparent lack of variation is an illusion. Most of your code will have specific conditions laying in wait which will trigger new actions.

If you have a temperature sensor connected to your Arduino for example, then when the temperature gets to a predefined threshold you might have a fan kick on. The looping code is constantly checking the temperature waiting to trigger the fan. So even though the code loops over and over, not every piece of the code will be executed every iteration of the loop.

**3.2 INTRODUCTION ARDUINO LIBRARIES**

Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc. For example, the built-in LiquidCrystal library makes it easy to talk to character LCD displays. There are hundreds of additional libraries available on the Internet for download. The built-in libraries and some of these additional libraries are [listed in the reference](https://www.arduino.cc/en/Reference/Libraries). To use the additional libraries, you will need to install them.

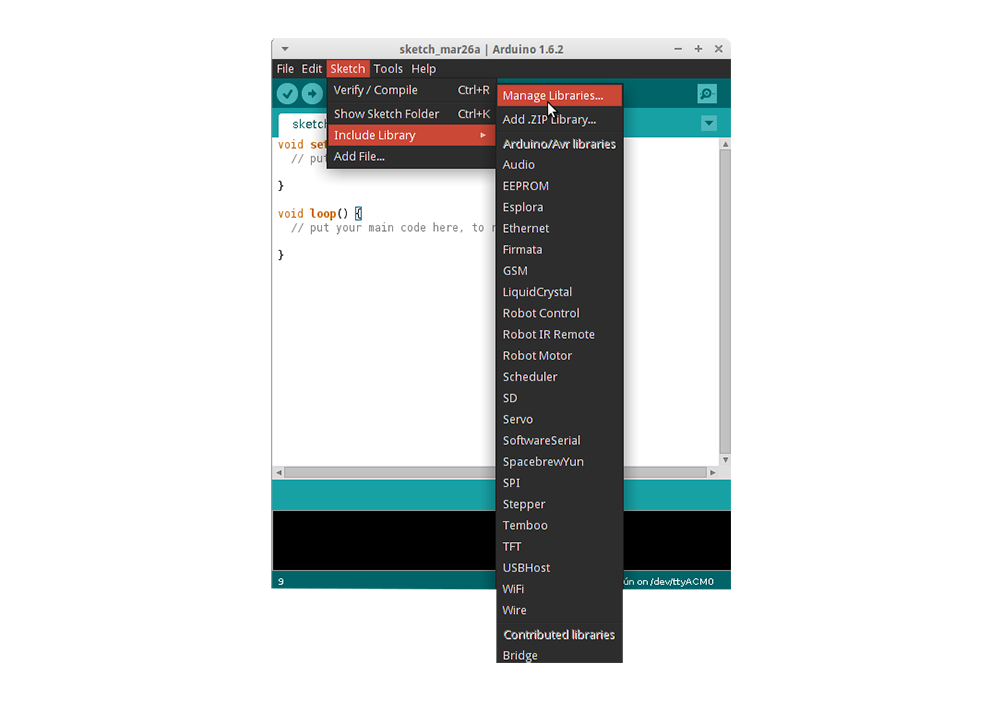
Arduino libraries are managed in three different places: inside the IDE installation folder, inside the core folder and in the libraries folder inside your sketchbook. The way libraries are chosen during compilation is designed to allow the update of libraries present in the distribution. This means that placing a library in the “libraries” folder in your sketchbook overrides the other libraries versions.

The same happens for the libraries present in additional cores installations. It is also important to note that the version of the library you put in your sketchbook may be lower than the one in the distribution or core folders, nevertheless it will be the one used during compilation. When you select a specific core for your board, the libraries present in the core’s folder are used instead of the same libraries present in the IDE distribution folder.

Last, but not least important is the way the Arduino Software (IDE) upgrades itself: all the files in Programs/Arduino (or the folder where you installed the IDE) are deleted and a new folder is created with fresh content. This is why we recommend that you only install libraries to the sketchbook folder so they are not deleted during the Arduino IDE update process.

**3.2.1 HOW TO INSTALL A LIBRARY**

To install a new library into your Arduino IDE you can use the Library Manager (available from IDE version 1.6.2). Open the IDE and click to the "Sketch" menu and then Include Library > Manage Libraries.



Then the Library Manager will open and you will find a list of libraries that are already installed or ready for installation. In this example we will install the Bridge library. Scroll the list to find it, click on it, then select the version of the library you want to install. Sometimes only one version of the library is available. If the version selection menu does not appear, don't worry: it is normal.



Finally click on install and wait for the IDE to install the new library. Downloading may take time depending on your connection speed. Once it has finished, an Installed tag should appear next to the Bridge library. You can close the library manager. You can now find the new library available in the Sketch > Include Library menu. If you want to add your own library to Library Manager, follow [these instructions](https://github.com/arduino/Arduino/wiki/Library-Manager-FAQ).

**3.3 HOW TO CONNECT ARDUINO BOARD**

If you're using a serial board, power the board with an external power supply (6 to 25 volts DC, with the core of the connector positive). Connect the board to a serial port on your computer. On the USB boards, the power source is selected by the jumper between the USB and power plugs. To power the board from the USB port (good for controlling low power devices like LEDs), place the jumper on the two pins closest to the USB plug. To power the board from an external power supply (needed for motors and other high current devices), place the jumper on the two pins closest to the power plug. Either way, connect the board to a USB port on your computer. On Windows, the Add New Hardware wizard will open; tell it you want to specify the location to search for drivers and point to the folder containing the USB drivers you unzipped in the previous step.

The power LED should go on.

**3.4 HOW TO UPLOAD A PROGRAM**

The content of circuits and Arduino sketches can vary greatly. Before you get started, there is one simple process for uploading a sketch to an Arduino board that you can refer back to.

Follow these steps to upload your sketch:

1. Connect your Arduino using the USB cable.

The square end of the USB cable connects to your Arduino and the flat end connects to a USB port on your computer.

1. Choose Tools→Board→Arduino Uno to find your board in the Arduino menu.

You can also find all boards through this menu, such as the Arduino MEGA 2560 and Arduino Leonardo.

1. Choose the correct serial port for your board.

You find a list of all the available serial ports by choosing Tools→Serial Port→ comX or /dev/tty.usbmodemXXXXX. X marks a sequentially or randomly assigned number. In Windows, if you have just connected your Arduino, the COM port will normally be the highest number, such as com 3 or com 15.

Many devices can be listed on the COM port list, and if you plug in multiple Arduinos, each one will be assigned a new number. On Mac OS X, the /dev/tty.usbmodem number will be randomly assigned and can vary in length, such as /dev/tty.usbmodem1421 or /dev/tty.usbmodem262471. Unless you have another Arduino connected, it should be the only one visible.

1. Click the Upload button.

This is the button that points to the right in the Arduino environment. You can also use the keyboard shortcut Ctrl+U for Windows or Cmd+U for Mac OS X.

**CHAPTER 4**

**HARDWARE REQUIREMENTS**

Hardware Components of this project are

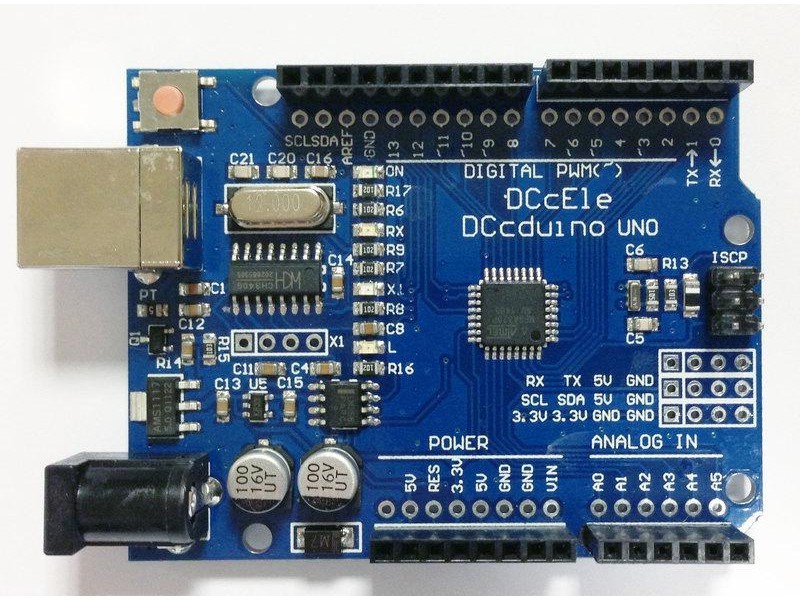
* 1. Arduino Uno with ATmega328P Microcontroller
  2. R307 finger print sensor
  3. 16X 2 LCD
  4. 12V ADAPTOR
  5. dc MOTOR
  6. L298N DRIVER

**4.1 INTRODUCTION TO ARDUINO UNO**

Arduino is an open-source electronics platform based on easy-to-use hardware and software. [Arduino boards](https://www.arduino.cc/en/Main/Products) are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the [Arduino programming language](https://www.arduino.cc/en/Reference/HomePage) (based on [Wiring](http://wiring.org.co/)), and [the Arduino Software (IDE)](https://www.arduino.cc/en/Main/Software), based on [Processing](https://processing.org/).

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of [accessible knowledge](http://forum.arduino.cc/) that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The [software](https://www.arduino.cc/en/Main/Software), too, is open-source, and it is growing through the contributions of users worldwide.



**4.1.1 WHY ARDUINO**

Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

**4.1.2 ADVANTAGES OF ARDUINO**

* **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
* **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
* **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
* **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
* **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the [breadboard version of the module](https://www.arduino.cc/en/Main/Standalone) in order to understand how it works and save money.

**4.1.3 FEATURES OF ARDUINO UNO**

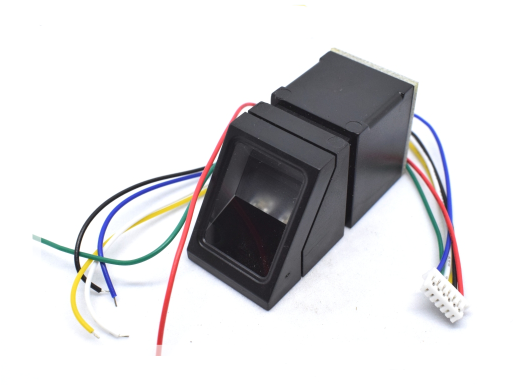
The **Arduino Uno** is a microcontroller board based on the ATmega328. Arduino is an open-source, prototyping platform and its simplicity makes it ideal for  hobbyists to use as well as professionals. The Arduino Uno has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

**Features of the Arduino UNO:**

* Microcontroller: ATmega328
* Operating Voltage: 5V
* Input Voltage (recommended): 7-12V
* Input Voltage (limits): 6-20V
* Digital I/O Pins: 14 (of which 6 provide PWM output)
* Analog Input Pins: 6
* DC Current per I/O Pin: 40 mA
* DC Current for 3.3V Pin: 50 mA
* Flash Memory: 32 KB of which 0.5 KB used by bootloader
* SRAM: 2 KB (ATmega328)
* EEPROM: 1 KB (ATmega328)
* Clock Speed: 16 MHz

**4.2 INTRODUCTION TO FINGER PRINT SCANNER**

Everyone has patterns of friction ridges on their fingers, and it is this pattern that is called the fingerprint. Fingerprints are uniquely detailed, durable over an individual's lifetime, and difficult to alter. Because there are countless combinations, fingerprints have become an ideal means of identification.



**4.2.1 Types of sensors**

There are four types of fingerprint scanner the optical scanner, the capacitance scanner, the ultrasonic scanner, and the thermal scanner. The basic function of every type of scanner is to obtain an image of a person's fingerprint and find a match for it in its database. The measure of the fingerprint image quality is in dots per inch (DPI).

Optical scanners take a visual image of the fingerprint using a digital camera.

Capacitive or CMOS scanners use capacitors and thus electrical current to form an image of the fingerprint. This type of scanner tends to excel in terms of precision.

Ultrasound fingerprint scanners use high frequency sound waves to penetrate the epidermal (outer) layer of the skin.

Thermal scanners sense the temperature differences on the contact surface, in between fingerprint ridges and valleys.

All fingerprint scanners are susceptible to be fooled by a technique that involves photographing fingerprints, processing the photographs using special software, and printing fingerprint replicas using a 3D printer

**4.3 INTRODUCTION TO 16X2 LCD MODULE**

A liquid-crystal display (LCD) is a [flat-panel display](https://en.wikipedia.org/wiki/Flat_panel_display) or other [electronically modulated optical device](https://en.wikipedia.org/wiki/Electro-optic_modulator) that uses the light-modulating properties of [liquid crystals](https://en.wikipedia.org/wiki/Liquid_crystal). Liquid crystals do not emit light directly, instead using a [backlight](https://en.wikipedia.org/wiki/Backlight) or [reflector](https://en.wikipedia.org/wiki/Reflector_(photography)) to produce images in color or [monochrome](https://en.wikipedia.org/wiki/Monochrome).[[1]](https://en.wikipedia.org/wiki/Liquid-crystal_display#cite_note-1) LCDs are available to display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which can be displayed or hidden, such as preset words, digits, and [7-segment](https://en.wikipedia.org/wiki/7-segment) displays, as in a [digital clock](https://en.wikipedia.org/wiki/Digital_clock). They use the same basic technology, except that arbitrary images are made up of a large number of small [pixels](https://en.wikipedia.org/wiki/Pixel), while other displays have larger elements.

LCDs are used in a wide range of applications including [computer monitors](https://en.wikipedia.org/wiki/Computer_monitor), [televisions](https://en.wikipedia.org/wiki/Television), [instrument panels](https://en.wikipedia.org/wiki/Dashboard), [aircraft cockpit displays](https://en.wikipedia.org/wiki/Flight_instruments), and indoor and outdoor signage. Small LCD screens are common in portable consumer devices such as [digital cameras](https://en.wikipedia.org/wiki/Digital_camera), [watches](https://en.wikipedia.org/wiki/Watch), [calculators](https://en.wikipedia.org/wiki/Calculator), and [mobile telephones](https://en.wikipedia.org/wiki/Mobile_telephone), including [smartphones](https://en.wikipedia.org/wiki/Smartphone). LCD screens are also used on [consumer electronics](https://en.wikipedia.org/wiki/Consumer_electronics) products such as DVD players, video game devices and [clocks](https://en.wikipedia.org/wiki/Clock). LCD screens have replaced heavy, bulky [cathode ray tube](https://en.wikipedia.org/wiki/Cathode_ray_tube) (CRT) displays in nearly all applications. LCD screens are available in a wider range of screen sizes than CRT and [plasma displays](https://en.wikipedia.org/wiki/Plasma_display), with LCD screens available in sizes ranging from tiny [digital watches](https://en.wikipedia.org/wiki/Digital_watch) to huge, big-screen [television sets](https://en.wikipedia.org/wiki/Television_set).

Since LCD screens do not use phosphors, they do not suffer [image burn-in](https://en.wikipedia.org/wiki/Screen_burn-in) when a static image is displayed on a screen for a long time (e.g., the table frame for an aircraft schedule on an indoor sign). LCDs are, however, susceptible to [image persistence](https://en.wikipedia.org/wiki/Image_persistence).[[2]](https://en.wikipedia.org/wiki/Liquid-crystal_display#cite_note-Fujitsu-2) The LCD screen is more energy-efficient and can be disposed of more safely than a CRT can. Its low electrical power consumption enables it to be used in [battery](https://en.wikipedia.org/wiki/Battery_(electricity))-powered [electronic](https://en.wikipedia.org/wiki/Electronics) equipment more efficiently than CRTs can be. By 2008, annual sales of televisions with LCD screens exceeded sales of CRT units worldwide, and the CRT became obsolete for most purposes.



A **16x2 LCD** means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. Click to learn more about internal structure of a [LCD](http://www.engineersgarage.com/insight/how-lcd-works).



**4.3.1 LCD PIN DESCRIPTION**

|  |  |  |
| --- | --- | --- |
| **Pin No** | **Function** | **Name** |
| 1 | Ground (0V) | Ground |
| 2 | Supply voltage; 5V (4.7V – 5.3V) | Vcc |
| 3 | Contrast adjustment; through a variable resistor | VEE |
| 4 | Selects command register when low; and data register when high | Register Select |
| 5 | Low to write to the register; High to read from the register | Read/write |
| 6 | Sends data to data pins when a high to low pulse is given | Enable |
| 7 | 8-bit data pins | DB0 |
| 8 | DB1 |
| 9 | DB2 |
| 10 | DB3 |
| 11 | DB4 |
| 12 | DB5 |
| 13 | DB6 |
| 14 | DB7 |
| 15 | Backlight VCC (5V) | Led+ |
| 16 | Backlight Ground (0V) | Led- |

**4.4 INTRODUCTION TO ADAPTORS**

An AC adapter, AC/DC adapter, or AC/DC converter[[1]](https://en.wikipedia.org/wiki/AC_adapter#cite_note-1) is a type of external [power supply](https://en.wikipedia.org/wiki/Power_supply), often enclosed in a case similar to an [AC plug](https://en.wikipedia.org/wiki/AC_power_plugs_and_sockets). Other common names include plug pack, plug-in adapter, adapter block, domestic mains adapter, line power adapter, wall wart, power brick, and power adapter. Adapters for battery-powered equipment may be described as chargers or rechargers (see also [battery charger](https://en.wikipedia.org/wiki/Battery_charger)). AC adapters are used with electrical devices that require power but do not contain internal components to derive the required voltage and power from [mains power](https://en.wikipedia.org/wiki/Mains_electricity). The internal circuitry of an external power supply is very similar to the design that would be used for a built-in or internal supply.

External power supplies are used both with equipment with no other source of power and with [battery](https://en.wikipedia.org/wiki/Battery_(electricity))-powered equipment, where the supply, when plugged in, can sometimes charge the battery in addition to powering the equipment.

Use of an external power supply allows portability of equipment powered either by mains or battery without the added bulk of internal power components, and makes it unnecessary to produce equipment for use only with a specified power source; the same device can be powered from 120 VAC or 230 VAC mains, vehicle or aircraft battery by using a different adapter.



### 4.4.1 MODES OF OPERATIONS

Originally, most AC/DC adapters were [linear power supplies](https://en.wikipedia.org/wiki/Power_supply), containing a [transformer](https://en.wikipedia.org/wiki/Transformer) to convert the [mains electricity](https://en.wikipedia.org/wiki/Mains_electricity) voltage to a lower voltage, a [rectifier](https://en.wikipedia.org/wiki/Rectifier) to convert it to [pulsating DC](https://en.wikipedia.org/wiki/Pulsating_direct_current), and a filter to smooth the pulsating waveform to DC, with residual [ripple](https://en.wikipedia.org/wiki/Ripple_(electrical)) variations small enough to leave the powered device unaffected. Size and weight of the device was largely determined by the transformer, which in turn was determined by the power output and [mains frequency](https://en.wikipedia.org/wiki/Utility_frequency). Ratings over a few watts made the devices too large and heavy to be physically supported by a wall outlet. The output voltage of these adapters varied with load; for equipment requiring a more stable voltage, [linear](https://en.wikipedia.org/wiki/Linear_circuit) [voltage regulator](https://en.wikipedia.org/wiki/Voltage_regulator) circuitry was added. Losses in the transformer and the linear regulator were considerable; efficiency was relatively low, and significant power dissipated as heat even when not driving a load.

In the early twenty-first century, [switched-mode power supplies](https://en.wikipedia.org/wiki/Switched-mode_power_supply) (SMPSs) became almost ubiquitous for this purpose. Mains voltage is rectified to a high direct voltage driving a switching circuit, which contains a transformer operating at a high frequency and outputs direct current at the desired voltage. The high-frequency ripple is more easily filtered out than mains-frequency. The high frequency allows the transformer to be small, which reduces its losses; and the switching regulator can be much more efficient than a linear regulator. The result is a much more efficient, smaller, and lighter device. Safety is ensured, as in the older linear circuit, because there is still a transformer which electrically isolates the output from the mains.

A linear circuit must be designed for a specific, narrow range of input voltages (e.g., 220–240 VAC) and must use a transformer appropriate for the frequency (usually 50 or 60 Hz), but a switched-mode supply can work efficiently over a very wide range of voltages and frequencies; a single 100–240 VAC unit will handle almost any mains supply in the world.

However, unless very carefully designed and using suitable components, switching adapters are more likely to fail than the older type, due in part to complex circuitry and the use of semiconductors. Unless designed well, these adapters may be easily damaged by overloads, even [transient](https://en.wikipedia.org/wiki/Transient_(oscillation)) ones, which can come from [lightning](https://en.wikipedia.org/wiki/Lightning), brief mains overvoltage (sometimes caused by an [incandescent light](https://en.wikipedia.org/wiki/Incandescent_light) on the same power circuit failing), component degradation, etc. A very common mode of failure is due to the use of [electrolytic capacitors](https://en.wikipedia.org/wiki/Electrolytic_capacitor) whose [equivalent series resistance](https://en.wikipedia.org/wiki/Equivalent_series_resistance) (ESR) increases with age; switching regulators are very sensitive to high ESR (the older linear circuit also used electrolytic capacitors, but the effect of degradation is much less dramatic). Well-designed circuits pay attention to the ESR, ripple current rating, pulse operation, and temperature rating of capacitors.

**4.4.2 ADAVANTAGES OF ADAPTER**

External AC adapters are widely used to power small or portable electronic devices. The advantages include:

* Safety – External power adapters can free product designers from worrying about some safety issues. Much of this style of equipment uses only voltages low enough not to be a [safety hazard](https://en.wikipedia.org/wiki/Safety_hazard) internally, although the power supply must out of necessity use dangerous mains voltage. If an external power supply is used (usually via a [power connector, often of coaxial type](https://en.wikipedia.org/wiki/Coaxial_power_connector)), the equipment need not be designed with concern for hazardous voltages inside the enclosure. This is particularly relevant for equipment with lightweight cases which may break and expose internal electrical parts.
* Heat reduction – Heat reduces reliability and longevity of electronic components, and can cause sensitive circuits to become inaccurate or malfunction. A separate power supply removes a source of heat from the apparatus.
* Electrical noise reduction – Because radiated electrical noise falls off with the square of the distance, it is to the manufacturer's advantage to convert potentially noisy AC line power or automotive power to "clean", filtered DC in an external adapter, at a safe distance from noise-sensitive circuitry.
* Weight and size reduction – Removing power components and the mains connection plug from equipment powered by rechargeable batteries reduces the weight and size which must be carried.
* Ease of replacement – Power supplies are more prone to failure than other circuitry due to their exposure to power spikes and their internal generation of waste heat. External power supplies can be replaced quickly by a user without the need to have the powered device repaired.
* Configuration versatility – Externally powered electronic products can be used with different power sources as needed (e.g. 120VAC, 240VAC, 12VDC, or external battery pack), for convenient use in the field, or when traveling.
* Simplified product inventory, distribution, and certification – An electronic product that is sold and used internationally must be powered from a wide range of power sources, and must meet product safety regulations in many jurisdictions, usually requiring expensive certification by national or regional [safety agencies](https://en.wikipedia.org/wiki/Safety_standards) such as [Underwriters Laboratories](https://en.wikipedia.org/wiki/Underwriters_Laboratories) or [Technischer Überwachungsverein](https://en.wikipedia.org/wiki/Technischer_%C3%9Cberwachungsverein" \o "Technischer Überwachungsverein). A single version of a device may be used in many markets, with the different power requirements met by different external power supplies, so that only one version of the device need be manufactured, stocked, and tested. If the design of the device is modified over time (a frequent occurrence), the power supply design itself need not be retested (and vice versa).
* Constant voltage is produced by a specific type of adapter used for [computers](https://en.wikipedia.org/wiki/Computers) and [laptops](https://en.wikipedia.org/wiki/Laptops). These types of adapters are commonly known as eliminators.

**4.4.3 PROBLEMS WITH ADAPTER**

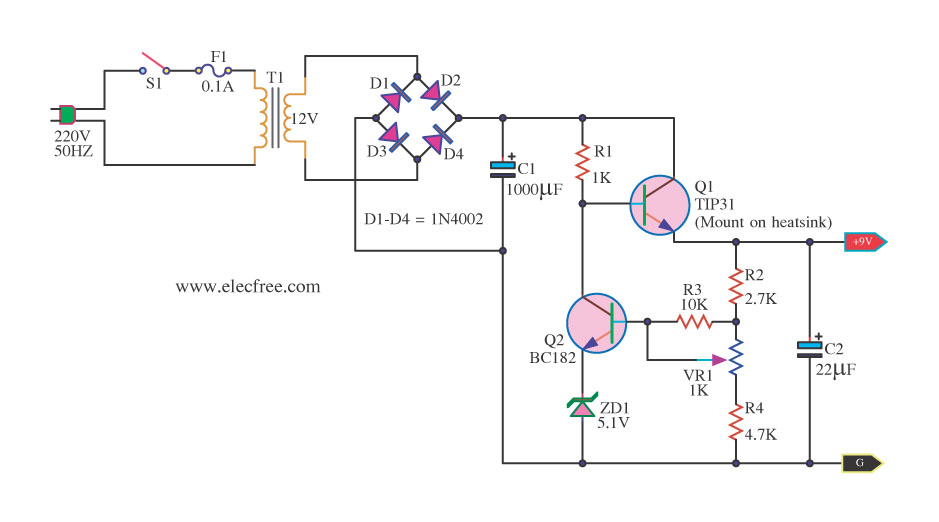
Problems with this type of power supply include, but are not limited to:

* Size – Power supplies which plug into the mains directly without using a plug on a cable (true wall warts) are bulkier than bare plugs; sometimes they are too large to plug into power sockets with restricted space, or into adjacent sockets on [power strips](https://en.wikipedia.org/wiki/Power_strip) (due to the fact that they can block other plugs also).
* Weight – Some AC adapters can be heavy, exerting excess weight on the power socket (this depends on the socket design of the country in question). Some external power supplies are "power bricks" (also known as "line lumps") having a short AC cord so they can lie on the floor, thus relieving strain, at the expense of clutter. Other wall-hanging types are made long and thin, minimizing the leverage of their weight vector that pulls the plug out, but exacerbating the size problem. The weight for equipment that must be carried (e.g. for traveling) is not a disadvantage of external supplies, as the alternative is an equally heavy internal supply; but in many cases a single universal supply could replace several proprietary ones.
* Inefficiency – Some idling power is wasted as the power supply is left running when the equipment power switch is off or the equipment is disconnected from the power supply. In recent years it has become common for equipment with internal supplies to share this problem due to the use of a "soft" power switch.
* Confusion – External power supplies are often generic and not clearly marked to identify the equipment they are designed to power. It is very easy to separate power supply and equipment, and can be difficult to re-match the many devices with their power supplies.
* Compatibility problems – There is no standardization of connectors; the same connector is often used for different voltages, and for both DC supplies and AC-to-AC transformers. Incompatible voltage or polarity may be present on physically interchangeable connectors. This easily leads to using the wrong power supply, which can destroy equipment.

**4.4.4 DANGEROUS AND UNRELIABLE ADAPTERS**

Manufacturers of equipment supplied with AC adapters often supply replacements at high prices; this has encouraged the manufacture of compatible third-party [aftermarket](https://en.wikipedia.org/wiki/Aftermarket_(merchandise)) replacements, which may be of satisfactory quality and performance at significantly lower prices. However, some adapters, usually at very low prices, and sometimes with unknown brands or sometimes fraudulently marked with the name of a reputable manufacturer, have various deficiencies which can cause inadequate performance (e.g. poor regulation and ripple, maximum power capacity lower than specified, hot running), unreliability (e.g. overheating to temperatures exceeding component ratings), and electrical or fire danger to users (e.g. insulation which frays with wear, lack of fuse).

Spurious marks of conformity to standards may be present; in one case it was reported that "Chinese manufacturers were submitting well-engineered electrical products to obtain conformity testing reports, but then removing non-essential components in production to reduce costs". A test of 27 chargers found that all the eight legitimately branded with a reputable name met safety standards, but none of those unbranded or with minor names did, despite bearing the [CЄ](https://en.wikipedia.org/wiki/File:Conformit%C3%A9_Europ%C3%A9enne_(logo).svg) [mark of conformity](https://en.wikipedia.org/wiki/CE_marking).



**4.4.5 AUTO SENSING ADAPTERS**

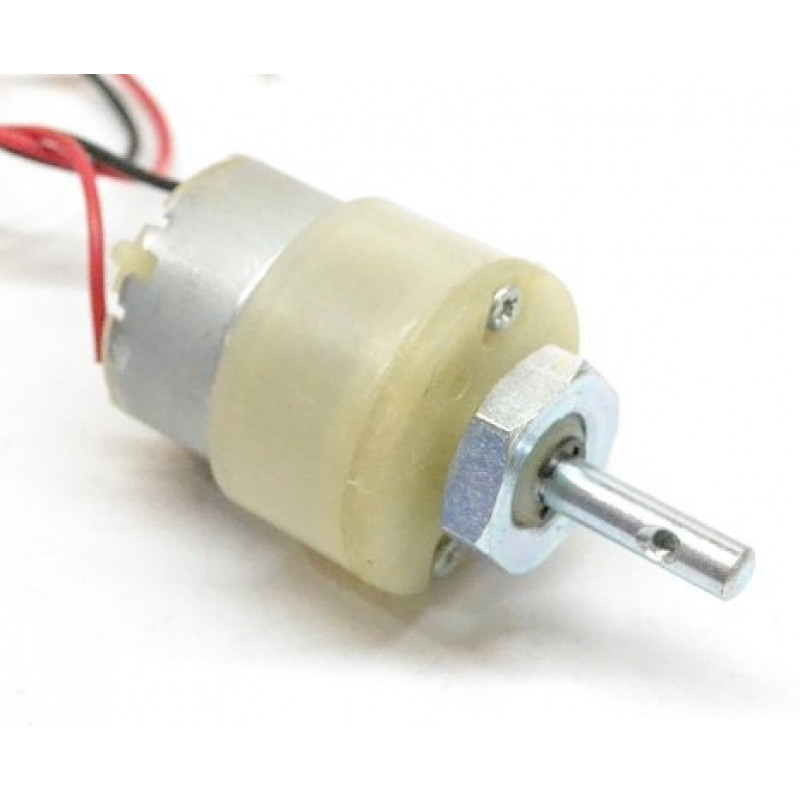
Some universal adapters automatically set their output voltage and maximum current according to which of a range of interchangeable tips is fitted; tips are available to fit and supply appropriate power to many notebook computers and mobile devices. Different tips may use the same connector, but automatically supply different power; it is essential to use the right tip for the apparatus being powered, but no switch needs to be set correctly by the user. The advent of switch-mode power supplies has allowed adapters to work from any AC mains supply from 100 to 240V with an appropriate plug; operation from standard 12V DC vehicle and aircraft supplies can also be supported. With the appropriate adapter, accessories, and tips, a variety of equipment can be powered from almost any source of power.

A "Green Plug" system has been proposed, based on [USB](https://en.wikipedia.org/wiki/Universal_Serial_Bus) technology, by which the consuming device would tell the external power supply what kind of power is needed.

**4.5 INTRODUCTION TO DC MOTOR**

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields. Nearly all types of DC motors have some internal mechanism, either electromechanical or electronic, to periodically change the direction of current flow in part of the motor.

DC motors were the first type widely used, since they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances. The [universal motor](https://en.wikipedia.org/wiki/Universal_motor) can operate on direct current but is a lightweight motor used for portable power tools and appliances. Larger DC motors are used in propulsion of electric vehicles, elevator and hoists, or in drives for steel rolling mills. The advent of power electronics has made replacement of DC motors with [AC motors](https://en.wikipedia.org/wiki/AC_motors) possible in many applications.



**4.5.1 BRUSHED DC MOTOR**

The [brushed DC electric motor](https://en.wikipedia.org/wiki/Brushed_DC_electric_motor) generates torque directly from DC power supplied to the motor by using internal commutation, stationary magnets ([permanent](https://en.wikipedia.org/wiki/Magnet) or [electromagnets](https://en.wikipedia.org/wiki/Electromagnet)), and rotating electrical magnets.

Advantages of a brushed DC motor include low initial cost, high reliability, and simple control of motor speed. Disadvantages are high maintenance and low life-span for high intensity uses. Maintenance involves regularly replacing the carbon brushes and springs which carry the electric current, as well as cleaning or replacing the [commutator](https://en.wikipedia.org/wiki/Commutator_(electric)). These components are necessary for transferring electrical power from outside the motor to the spinning wire windings of the rotor inside the motor. Brushes consist of conductors.

**4.5.2 BRUSHLESS DC MOTOR**

Typical brushless DC motors use one or more permanent magnets in the rotor and [electromagnets](https://en.wikipedia.org/wiki/Electromagnet) on the motor housing for the stator. A motor controller converts DC to [AC](https://en.wikipedia.org/wiki/Alternating_current). This design is mechanically simpler than that of brushed motors because it eliminates the complication of transferring power from outside the motor to the spinning rotor. The motor controller can sense the rotor's position via [Hall effect](https://en.wikipedia.org/wiki/Hall_effect) sensors or similar devices and can precisely control the timing, phase, etc., of the current in the rotor coils to optimize torque, conserve power, regulate speed, and even apply some braking. Advantages of brushless motors include long life span, little or no maintenance, and high efficiency. Disadvantages include high initial cost, and more complicated motor speed controllers. Some such brushless motors are sometimes referred to as "synchronous motors" although they have no external power supply to be synchronized with, as would be the case with normal AC synchronous motors.

**4.5.3 PERMANENT MAGNET STATOR**

A PM motor does not have a field winding on the stator frame, instead relying on PMs to provide the magnetic field against which the rotor field interacts to produce torque. Compensating windings in series with the armature may be used on large motors to improve commutation under load. Because this field is fixed, it cannot be adjusted for speed control. PM fields (stators) are convenient in miniature motors to eliminate the power consumption of the field winding. Most larger DC motors are of the "dynamo" type, which have stator windings. Historically, PMs could not be made to retain high flux if they were disassembled; field windings were more practical to obtain the needed amount of flux. However, large PMs are costly, as well as dangerous and difficult to assemble; this favors wound fields for large machines.

To minimize overall weight and size, miniature PM motors may use high energy magnets made with [neodymium](https://en.wikipedia.org/wiki/Neodymium) or other strategic elements; most such are neodymium-iron-boron alloy. With their higher flux density, electric machines with high-energy PMs are at least competitive with all optimally designed [singly fed](https://en.wikipedia.org/wiki/DC_motor#Singly_fed_electric_motor) synchronous and induction electric machines. Miniature motors resemble the structure in the illustration, except that they have at least three rotor poles (to ensure starting, regardless of rotor position) and their outer housing is a steel tube that magnetically links the exteriors of the curved field magnets.

**4.5.4 SERIES CONNECTION**

A series DC motor connects the [armature](https://en.wikipedia.org/wiki/Armature_(electrical_engineering)) and [field windings](https://en.wikipedia.org/wiki/Field_coil) in [series](https://en.wikipedia.org/wiki/Series_circuits) with a [common](https://en.wikipedia.org/wiki/Battery_(electricity)) D.C. power source. The motor speed varies as a non-linear function of load torque and armature current; current is common to both the stator and rotor yielding current squared (I^2) behavior. A series motor has very high starting torque and is commonly used for starting high inertia loads, such as trains, elevators or hoists.[[2]](https://en.wikipedia.org/wiki/DC_motor#cite_note-2) This speed/torque characteristic is useful in applications such as [dragline excavators](https://en.wikipedia.org/wiki/Dragline_excavator), where the digging tool moves rapidly when unloaded but slowly when carrying a heavy load.

A series motor should never be started at no load. With no mechanical load on the series motor, the current is low, the counter-EMF produced by the field winding is weak, and so the armature must turn faster to produce sufficient counter-EMF to balance the supply voltage. The motor can be damaged by overspeed. This is called a runaway condition.

Series motors called [universal motors](https://en.wikipedia.org/wiki/Universal_motor) can be used on [alternating current](https://en.wikipedia.org/wiki/Alternating_current). Since the armature voltage and the field direction reverse at the same time, torque continues to be produced in the same direction. However they run at a lower speed with lower torque on AC supply when compared to DC due to [reactance](https://en.wikipedia.org/wiki/Electrical_reactance) voltage drop in AC which is not present in DC.[[3]](https://en.wikipedia.org/wiki/DC_motor#cite_note-3)Since the speed is not related to the line frequency, universal motors can develop higher-than-synchronous speeds, making them lighter than induction motors of the same rated mechanical output. This is a valuable characteristic for hand-held power tools. Universal motors for commercial [utility](https://en.wikipedia.org/wiki/Utility_frequency) are usually of small capacity, not more than about 1 kW output. However, much larger universal motors were used for electric locomotives, fed by special low-frequency [traction power networks](https://en.wikipedia.org/wiki/Traction_power_network) to avoid problems with commutation under heavy and varying loads.

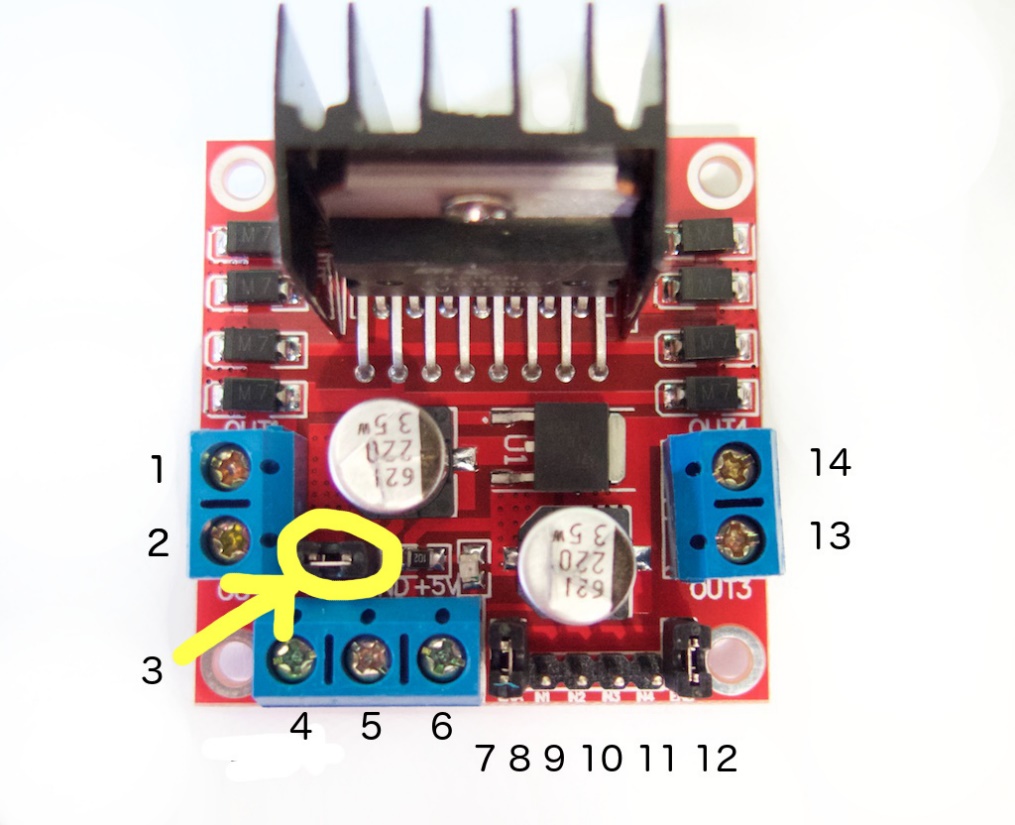
**4.5.5 SHUNT CONNECTION**

A shunt DC motor connects the armature and field windings in parallel or shunt with a common D.C. power source. This type of motor has good speed regulation even as the load varies, but does not have the starting torque of a series DC motor.[[4]](https://en.wikipedia.org/wiki/DC_motor#cite_note-4) It is typically used for industrial, adjustable speed applications, such as machine tools, winding/unwinding machines and tensioners.

**4.6 INTRODCTION TO L298N DRIVER**

[Dual Motor Controller](https://tronixlabs.com.au/robotics/motor-controllers/l298n-dual-motor-controller-module-2a-australia/) Module 2A with Arduino. This allows you to control the speed and direction of two DC motors, or control one bipolar stepper motor with ease. The L298N H-bridge module can be used with motors that have a voltage of between 5 and 35V DC.

There is also an onboard 5V regulator, so if your supply voltage is up to 12V you can also source 5V from the board. These L298 H-bridge dual motor controller modules [.](https://tronixlabs.com.au/robotics/motor-controllers/l298n-dual-motor-controller-module-2a-australia/)



**4.6.1 PIN DESCRIPTION**

1. DC motor 1 "+" or stepper motor A+
2. DC motor 1 "-" or stepper motor A-
3. 12V jumper - remove this if using a supply voltage greater than 12V DC. This enables power to the onboard 5V regulator
4. Connect your motor supply voltage here, maximum of 35V DC. Remove 12V jumper if >12V DC
5. GND
6. 5V output if 12V jumper in place, ideal for powering your Arduino (etc)
7. DC motor 1 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.
8. IN1
9. IN2
10. IN3
11. IN4
12. DC motor 2 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.
13. DC motor 2 "+" or stepper motor B+
14. DC motor 2 "-" or stepper motor B-

**4.6.2 CONTROLLING DC MOTOR**

To control one or two DC motors is quite easy. First connect each motor to the A and B connections on the L298N  [module](https://tronixlabs.com.au/robotics/motor-controllers/l298n-dual-motor-controller-module-2a-australia/). If you're using two motors for a robot (etc) ensure that the polarity of the motors is the same on both inputs. Otherwise you may need to swap them over when you set both motors to forward and one goes backwards!

Next, connect your power supply - the positive to pin 4 on the module and negative/GND to pin 5. If you supply is up to 12V you can leave in the 12V jumper (point 3 in the image above) and 5V will be available from pin 6 on the module. This can be fed to your Arduino's 5V pin to power it from the motors' power supply. Don't forget to connect Arduino GND to pin 5 on the module as well to complete the circuit.

Now you will need six digital output pins on your Arduino, two of which need to be PWM (pulse-width modulation) pins. PWM pins are denoted by the tilde ("~") next to the pin number, for example: Finally, connect the Arduino digital output pins to the driver module. In our example we have two DC motors, so digital pins D9, D8, D7 and D6 will be connected to pins IN1, IN2, IN3 and IN4 respectively. Then connect D10 to module pin 7 (remove the jumper first) and D5 to module pin 12 (again, remove the jumper).

The motor direction is controlled by sending a HIGH or LOW signal to the drive for each motor (or channel). For example for motor one, a HIGH to IN1 and a LOW to IN2 will cause it to turn in one direction, and  a LOW and HIGH will cause it to turn in the other direction. However the motors will not turn until a HIGH is set to the enable pin (7 for motor one, 12 for motor two). And they can be turned off with a LOW to the same pin(s). However if you need to control the speed of the motors, the PWM signal from the digital pin connected to the enable pin can take care of it.

**CHAPTER 5**

**PROGRAM CODE**

#include <LiquidCrystal.h>

LiquidCrystal lcd(8, 9, 10, 11, 12, 13);

#include <Adafruit\_Fingerprint.h>

SoftwareSerial mySerial(2, 3);

int id;

Adafruit\_Fingerprint finger = Adafruit\_Fingerprint(&mySerial);

void setup()

{

pinMode(A0,OUTPUT);

pinMode(A1,OUTPUT);

digitalWrite(A0,LOW);

digitalWrite(A1,LOW);

finger.begin(9600);

lcd.begin(16, 2);

lcd.setCursor(0, 0);

lcd.print("FINGERPRINT ");

lcd.setCursor(0, 1);

lcd.print("DETECTOR ");

delay(3000);

lcd.clear();

}

void loop()

{

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("PLACE THE ");

lcd.setCursor(0, 1);

lcd.print("FINGER ");

id = getFingerprintID();

if (id == 1)

{

lcd.setCursor(0, 0);

lcd.print("THUMB ");

lcd.setCursor(0, 1);

lcd.print(" ");

digitalWrite(A0,LOW);

digitalWrite(A1,HIGH);

delay(50);

digitalWrite(A0,LOW);

digitalWrite(A1,LOW);

delay(3000);

digitalWrite(A0,HIGH);

digitalWrite(A1,LOW);

delay(50);

digitalWrite(A0,LOW);

digitalWrite(A1,LOW);

}

else if (id == 3)

{

lcd.setCursor(0, 0);

lcd.print("INDEX ");

lcd.setCursor(0, 1);

lcd.print("FINGER ");

digitalWrite(A0,LOW);

digitalWrite(A1,HIGH);

delay(3000);

digitalWrite(A0,LOW);

digitalWrite(A1,LOW);

delay(3000);

digitalWrite(A0,HIGH);

digitalWrite(A1,LOW);

delay(3000);

digitalWrite(A0,LOW);

digitalWrite(A1,LOW);

}

else if (id == 5)

{

lcd.setCursor(0, 0);

lcd.print("MIDDLE ");

lcd.setCursor(0, 1);

lcd.print("FINGER ");

digitalWrite(A0,LOW);

digitalWrite(A1,HIGH);

delay(3000);

digitalWrite(A0,LOW);

digitalWrite(A1,LOW);

delay(3000);

digitalWrite(A0,HIGH);

digitalWrite(A1,LOW);

delay(3000);

digitalWrite(A0,LOW);

digitalWrite(A1,LOW);

}

else if (id == 7)

{

lcd.setCursor(0, 0);

lcd.print("RING ");

lcd.setCursor(0, 1);

lcd.print("FINGER ");

digitalWrite(A0,LOW);

digitalWrite(A1,HIGH);

delay(3000);

digitalWrite(A0,LOW);

digitalWrite(A1,LOW);

delay(3000);

digitalWrite(A0,HIGH);

digitalWrite(A1,LOW);

delay(3000);

digitalWrite(A0,LOW);

digitalWrite(A1,LOW);

}

else if (id == 9)

{

lcd.setCursor(0, 0);

lcd.print("LITTLE ");

lcd.setCursor(0, 1);

lcd.print("FINGER ");

digitalWrite(A0,LOW);

digitalWrite(A1,HIGH);

delay(3000);

digitalWrite(A0,LOW);

digitalWrite(A1,LOW);

delay(3000);

digitalWrite(A0,HIGH);

digitalWrite(A1,LOW);

delay(3000);

digitalWrite(A0,LOW);

digitalWrite(A1,LOW);

}

delay(50);

}

uint8\_t getFingerprintID() {

uint8\_t p = finger.getImage();

switch (p) {

case FINGERPRINT\_OK:

//Serial.println("Image taken");

break;

case FINGERPRINT\_NOFINGER:

//Serial.println("No finger detected");

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

//Serial.println("Communication error");

return p;

case FINGERPRINT\_IMAGEFAIL:

//Serial.println("Imaging error");

return p;

default:

//Serial.println("Unknown error");

return p;

}

// OK success!

p = finger.image2Tz();

switch (p) {

case FINGERPRINT\_OK:

//Serial.println("Image converted");

break;

case FINGERPRINT\_IMAGEMESS:

//Serial.println("Image too messy");

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

//Serial.println("Communication error");

return p;

case FINGERPRINT\_FEATUREFAIL:

//Serial.println("Could not find fingerprint features");

return p;

case FINGERPRINT\_INVALIDIMAGE:

//Serial.println("Could not find fingerprint features");

return p;

default:

//Serial.println("Unknown error");

return p;

}

// OK converted!

p = finger.fingerSearch();

if (p == FINGERPRINT\_OK) {

//Serial.println("Found a print match!");

} else if (p == FINGERPRINT\_PACKETRECIEVEERR) {

//Serial.println("Communication error");

return p;

} else if (p == FINGERPRINT\_NOTFOUND) {

//Serial.println("Did not find a match");

return p;

} else {

//Serial.println("Unknown error");

return p;

}

// found a match!

//Serial.print("Found ID #"); Serial.print(finger.fingerID);

//Serial.print(" with confidence of "); Serial.println(finger.confidence);

return finger.fingerID;

}

// returns -1 if failed, otherwise returns ID #

int getFingerprintIDez() {

uint8\_t p = finger.getImage();

if (p != FINGERPRINT\_OK) return -1;

p = finger.image2Tz();

if (p != FINGERPRINT\_OK) return -1;

p = finger.fingerFastSearch();

if (p != FINGERPRINT\_OK) return -1;

// found a match!

//Serial.print("Found ID #"); Serial.print(finger.fingerID);

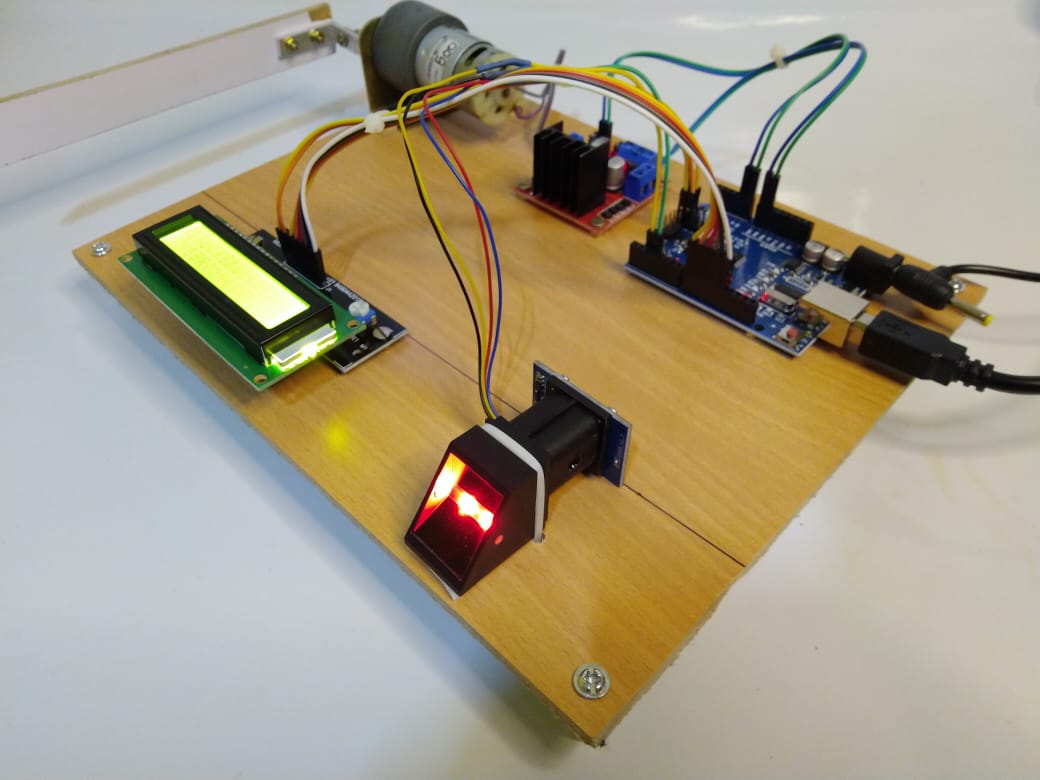
//Serial.print(" with confidence of "); Serial.println(finger.confidence);

return finger.fingerID;

}

**CHAPTER 6**

**RESULT OF THE PROJECT**

****

**CHAPTER 7**

**CONCLUSION**

Security is a major concern in our day to day life, and digital locks have become an important part of these security systems. There are many types of security systems available to secure our place. Some examples are PIR based Security System, RFID based Security System, Digital Lock System, bio-matrix systems, Electronics Code lock. In this post, we will Interface a Fingerprint Sensor Module with Arduino and will build a Fingerprint based Biometric Security System with door locking. Finger Print is considered one of the safest key to lock or unlock any system as it can recognize any person uniquely and can’t be copied easily.

**BIBLIOGRAPHY**

Ahmad, N.A.N. and Zamri, S.A.S., 2014, September. The cross platform application development adapted Spring framework to support front-end tendering services. In Computer, Communications, and Control Technology (I4CT), 2014 International Conference on (pp. 58-62). IEEE.

Alarcon, R. and Wilde, E., 2010, April. Linking data from restful services. In Third Workshop on Linked Data on the Web, Raleigh, North Carolina (April 2010).

Al-Fagih, A.E., Al-Turjman, F.M., Alsalih, W.M. and Hassanein, H.S., 2013. A priced public sensing framework for heterogeneous IoT architectures. Emerging Topics in Computing, IEEE Transactions on, 1(1), pp.133-147.

Alpaydin, E., 2014. Introduction to machine learning. MIT press.

Armando, A., Carbone, R., Chekole, E.G. and Ranise, S., 2014, June. Attribute based access control for APIs in spring security. In Proceedings of the 19th ACM symposium on Access control models and technologies (pp. 85-88). ACM.

Atighetchi, M., Soule, N., Pal, P., Loyall, J., Sinclair, A. and Grant, R., 2013, October. Safe configuration of TLS connections. In Communications and Network Security (CNS), 2013 IEEE Conference on (pp. 415- 422). IEEE.

Bormann, C., Castellani, A.P. and Shelby, Z., 2012. Coap: An application protocol for billions of tiny internet nodes. IEEE Internet Computing, 16(2), p.62.

Centenaro, M., Vangelista, L., Zanella, A. and Zorzi, M., 2015. Long-Range Communications in Unlicensed Bands: The Rising Stars in the IoT and Smart City Scenarios. arXiv preprint arXiv:1510.00620.

Chinrungrueng, J., Sunantachaikul, U. and Triamlumlerd, S., 2007, January. Smart parking: An application of optical wireless sensor network. In Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium on (pp. 66-66). IEEE.