CHAPTER ONE
INTRODUCTION

1.1 General Review
The safety problem has become an essential issue for all institutions especially banks, because most people rely on banks in the store and save their money. An RFID reader is a network connected device (fixed or mobile) with an antenna that sends power as well as data and commands to the tags. The RFID reader acts like an access point for RFID tagged items so that the tags' data can be made available to business applications. Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control objects in the physical world.

1.2 Problem Statement
The problem of insecurity, some people try to steal the property which may endanger the safety of money in the bank, house and office. To overcome the security threat, most people must install bunch of locks or alarm system. There are many types of alarm system available in the market which utilizes different types of sensors. The sensor system may not be good for all the time because it can detect different types of changes in the surrounding and the changes will be processed to be given out alert according to the pre-set value. The use of hand-lock systems that reduces security and increases customer waiting time. In addition, hack the password, allows someone else to recover the money from the bank locker.

1.3 Objectives
The main aims of this study are to:
- Design of security system to protect locker of the bank.
- Develop a program to allow to authentic person only to recovered money from bank locker.
- Implementation of bank security system using RFID technology.
- Test the performance of the system.

### 1.4 Methodology

- Study of all previous related works.
- Use of RFID reader, RFID tag, Arduino, Keypad, LCD, motor to implement the proposed system.
- Use of C language for programming the Arduino.
- Use of Protues software to simulate the proposed system.
- Evaluate the performance of the system based on simulation and experimental results.

### 1.5 Layout

This research consists of five chapters, Chapter one handles general introduction of the research, defines the problem and states the objectives. Chapter two discusses the Arduino and literature review, while chapter three handles the hardware and software considerations. Chapter four contains on the implementation of the system and testing. Finally chapter five handles the conclusion and recommendations.
CHAPTER TWO
THEORETICAL BACKGROUND AND LITERATURE REVIEW

2.1 Microcontroller

A microcontroller is a little computer on a chip. It has a processor, a kilobyte or two of Random Access Memory (RAM) for holding data, a few of kilobytes of Erasable Programmable Read-Only Memory (EPROM) or flash memory for holding your programs and it has input and output pins. These Input/Output (I/O) pins link the microcontroller to the rest of your electronics [1].

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

Some microcontrollers may use four-bit words and operate at clock rate frequencies as low as 4kHz, for low power consumption (single-digit mill watts or microwatts). They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (Central Processing Unit (CPU) clock and most peripherals off) may be just nano watts, making many of them well suited for long lasting battery applications. Other microcontrollers may serve performance-critical roles, where they may need to act more like a Digital Signal Processor (DSP), with higher clock speeds and power consumption [2].
Microcontrollers are sometimes called embedded microcontrollers, which just means that they are part of a larger device or system, unlike a general purpose computer, which also includes all of these components, a microcontroller is designed for a specific task to control a particular system [3]. The microcontroller architecture shown in Figure 2.1.

![Microcontroller architecture](image)

Figure 2.1: Microcontroller architecture

Microcontrollers are used in almost all type of electronic equipment, from coffeepots to laser printer. They are even incorporate in computer systems. They can be found in floppy drives, CD-ROM drivers, and video cards. A microcontroller is often embedded into the electronic circuit boards of these computer devices [4].

2.1.1 Advantage of microcontroller

There are a number of advantages to using microcontrollers in industry. The major advantages of microcontroller are small and can fit inside other
devices like an appliance or a vehicle. Microcontrollers cost less to produce, and they consume less power. A microcontroller is usually a single chip but when incorporate into a large control system, it is referred to as an embedded controller. An embedded controller often depends on other components in the system, such as additional memory, to perform its function [4].

a. Reusable
The typical microcontroller is programmable, which means it is reusable. This is especially advantageous for prototyping control circuitry. When developing a complex control system, it is not unusual for it to fail when first applied. As a matter of fact, a complex control project may need to be rewritten and/or rewired many times before it meets design expectations. The fact that the control circuit can be modified by programming rather than rewiring is very advantageous for fast project prototype development [4].

b. Dependable
Integrated circuits, such as the microcontroller, are much more dependable than relays. Before microcontrollers, control circuitry relied on many electromechanical relays and timers to control the system. Relays depend on electromagnets to move armature and contact parts, so they eventually wear out due to mechanical friction. Relays are also susceptible to damage caused by dust, dirt, corrosion, rust, insects, and other contaminants that can interfere with the moving parts. Microcontrollers have no moving parts. This provides a much higher rate of reliability. Relays and high-power transistors can be incorporated for final applications to motors, but the actual timing and control logic does not need to rely on the mechanical action of relays [4].

c. Cost effective
Microcontrollers can be produced at lower costs than their electromechanical predecessors. Also, microcontrollers can be reprogrammed if the designed application does not work correctly or if the application for its use changes [4].

d. Energy efficient
Because the majority of the circuitry is made from integrated circuits, the energy cost of using a microcontroller is much less than if using individual components of a relay-type logic circuit. Relay logic uses numerous relays wired in series and parallel to form control circuit conditions similar in function to logic gates. A microcontroller consumes less electrical energy than conventional electromechanical devices. [4]

2.1.2 Disadvantages of microcontrollers

There are a few disadvantages to using microcontrollers. The two most prominent disadvantages are reflected on the construction complexity and sensitivity to environmental conditions.

a. Programming complexity

Special skills are required to program the microcontrollers. This requires a higher level of training for some personnel. In addition, there are many different programming languages to choose from. This can lead to a compatibility problem when attempting to merge two dissimilar systems into one control system [4].

b. Electrostatic sensitivity

Most microcontrollers are composed of Complementary Metal-Oxide Semiconductor (CMOS) integrated circuitry. CMOS can be damaged easily by a static charge. Static precautions must be strictly obeyed [4].

2.1.3 Microcontroller components

The microcontroller consists of thousands of digital circuits combined into areas that provide specific functions [5]. The simplified block diagram in Figure 2.2, illustrates how the major sections inside the microcontroller work together to process the program instructions. The parts of the microcontroller are used to save data and programs, perform mathematic and logic functions, and generate timing signals. The different areas are connected by a bus system. The bus system contains tiny parallel circuits that carry the digital pulse patterns from section to section. The Read Only Memory (ROM) stores the program required for the microcontroller to function. The ROM controls
how the chip components operate and how data and instructions flow through the chip [6].

![Simplified typical microcontroller](image)

**Figure 2.2: Simplified typical microcontroller**

### 2.2 Arduino

Arduino is based on microcontroller board designs, produced by several vendors, using various microcontrollers. These systems provide sets of digital and analog I/O pins that can interface to various expansion boards (termed shields) and other circuits. The boards feature serial communication interfaces, including Universal Serial Bus (USB) on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino project provides an Integrated Development Environment (IDE) based on a programming language named Processing, which also supports the languages, C and C++. The first Arduino was introduced in 2005, aiming to provide a low cost, easy way for novices and professionals to create devices that interact with their environment using sensors and actuators. Common examples of such devices intended for beginner hobbyists include simple robots, thermostats, and motion detectors.
Arduino is a microcontroller platform that has captured the imagination of electronics enthusiasts. It is ease of use and open source nature makes it a great choice for anyone wanting to build electronic projects. Ultimately, it allows you to connect electronics through its pins so that it can control things for instance, turn lights or motors on and off or sense things such as light and temperature. This is why Arduino is sometimes given the description physical computing. Because Arduino can be connected to your computer by USB lead, this also means that you can use the Arduino as an interface board to control those same electronics from your computer.

### 2.2.1 Types of Arduino

There are different types of Arduino as follows:

1. **The Arduino Uno**
   
The Uno is the most common board and the one labeled as the classic Arduino. This board comes with everything new users need to learn about the electronics and programming required to start this hobby. It is compatible with most available Arduino shields.

2. **The Arduino Due**
   
The Arduino Due is the second iteration of the classic Arduino and offers more features for advanced users. The Due's processor is faster, has more memory, and more I/O ports. It does not support many shields. Because of the faster CPU, the Arduino Due runs on a lower voltage: 3.3V over the Uno's 5V. This means it cannot always support the same devices.

3. **The Arduino Leonardo**
   
The Leonardo is not a common board, but has similar features to the Uno, including the 5V power supply and the processing power. It is a good board for those who need more input and output ports than the Arduino Uno, but do not need the horsepower or size of the Due. It uses a micro-USB adapter instead of the Uno's full-size USB port.

4. **Arduino Yun**
   
The Arduino Yun is an Arduino board unlike any other. While programming it is very similar to the Arduino Leonardo and uses the same processor, the
Atmel ATmega32U4, it also has an additional processor, an Atheros AR9331, running Linux and the OpenWrt wireless stack. Programming the 32U4 via USB is identical to the Arduino Leonardo. Once you've configured the Yun to connect to your Wi-Fi network, you can program the 32U4 via Wi-Fi as well.

v. Arduino Mega 2560
The Mega is the second most commonly encountered version of the Arduino family. The Arduino Mega is like the Arduino Uno's beefier older brother. It boasts 256KB of memory (8 times more than the Uno). It also had 54 input and output pins, 16 of which are analog pins, and 14 of which can do Pulse Width Modulation (PWM). However, all of the added functionality comes at the cost of a slightly larger circuit board.

vi. Arduino Mega ADK
This specialized version of the Arduino is basically an Arduino Mega that has been specifically designed for interfacing with Android smartphones.

vii. Arduino Nano
The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x) or ATmega168 (Arduino Nano 2.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a Direct Current (DC) power jack, and works with a Mini-B USB cable instead of a standard one.

viii. Arduino Duemilanove
The Arduino Duemilanove ("2009") is a microcontroller board based on the ATmega168 or ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16MHz 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.

ix. Arduino Pro
The Arduino Pro is a microcontroller board based on the ATmega168 or ATmega328. The Pro comes in both 3.3V/8MHz and 5V/16MHz versions. It
has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a battery power jack, a power switch, a reset button, and holes for mounting a power jack, an ICSP header, and pin headers.

x. **Arduino BT (Bluetooth)**
The Arduino BT is a microcontroller board originally was based on the ATmega168, but now is supplied with the 328 and the Bluegiga WT11 Bluetooth module. It supports wireless serial communication over Bluetooth (but is not compatible with Bluetooth headsets or other audio devices). oscillator, screw terminals for power, an ICSP header, and a reset button.

xi. **Lilypad Arduino**
LilyPad is a set of sewable electronic pieces designed to help you build soft interactive textiles. A set of sewable electronic modules—including a small programmable computer called a LilyPad Arduino—can be stitched together with conductive thread to create interactive garments and accessories. LilyPad can sense information about the environment using inputs like light and temperature sensors and can act on the environment with outputs like Light Emitting Diode (LED) lights, vibrator motors, and speakers.

xii. **Arduino Fio**
The Arduino Fio is a microcontroller board based on the ATmega328P (datasheet) runs at 3.3V and 8MHz. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 8 analog inputs, an on-board resonator, a reset button, and holes for mounting pin headers. It has connections for a Lithium Polymer battery and includes a charge circuit over USB. An XBee socket is available on the bottom of the board.

These different types of Arduino shown in Figure 2.3.
The following table shows the different types of Arduino according to its families and memories.

Table 2.1: Different types of Arduino

<table>
<thead>
<tr>
<th>Arduino Board</th>
<th>Family</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SRAM</td>
</tr>
<tr>
<td>Duemilanove (328)</td>
<td>ATmega 328</td>
<td>2k</td>
</tr>
<tr>
<td>Uno</td>
<td>ATmega 328</td>
<td>2k</td>
</tr>
<tr>
<td>Arduino Mega 2560</td>
<td>ATmega 2560</td>
<td>8k</td>
</tr>
<tr>
<td>Arduino Mega ADK</td>
<td>ATmega 2560</td>
<td>8k</td>
</tr>
<tr>
<td>Arduino Ethernet</td>
<td>ATmega 328</td>
<td>2k</td>
</tr>
<tr>
<td>Arduino BT</td>
<td>ATmega 328</td>
<td>2k</td>
</tr>
<tr>
<td>Arduino Pro Mini 328 5V</td>
<td>ATmega 328</td>
<td>2k</td>
</tr>
<tr>
<td>Arduino Nano 3.0</td>
<td>ATmega 328</td>
<td>2k</td>
</tr>
<tr>
<td>Arduino Mini</td>
<td>ATmega 328</td>
<td>2k</td>
</tr>
<tr>
<td>Arduino Pro 3.3V</td>
<td>ATmega 328P</td>
<td>2k</td>
</tr>
<tr>
<td>Arduino Pro 5V</td>
<td>ATmega 328P</td>
<td>2k</td>
</tr>
<tr>
<td>Arduino Fio</td>
<td>ATmega 328P</td>
<td>2k</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
<td>----</td>
</tr>
<tr>
<td>LilyPad Simple Board</td>
<td>ATmega 168P</td>
<td>1k</td>
</tr>
<tr>
<td>LilyPad 328 Main Board</td>
<td>ATmega 328P</td>
<td>2k</td>
</tr>
</tbody>
</table>

### 2.2.2 Advantages of using Arduino

There are many advantages of Arduino such as:

**i. Ready to use**

The biggest advantage of Arduino is its ready to use structure. As Arduino comes in a complete package form which includes the 5V regulator, a burner, an oscillator, a microcontroller, serial communication interfaces LED and headers for the connections.

**ii. Effortless functions**

During coding of Arduino, you will notice some functions which make the life so easy. Another advantage of Arduino is its automatic unit conversion capability.

**iii. Large community**

There are many forums present on the internet in which people are talking about the Arduino. Engineers, hobbyists and professionals are making their projects through Arduino. You can easily find help about everything. Moreover the Arduino website itself explains each and every function of Arduino.

### 2.2.3 Disadvantages of Arduino

The disadvantage of Arduino are:

**i. Structure**

The structure of Arduino is its disadvantage as well. During building a project you have to make its size as small as possible.

**ii. Cost**

The most important factor which you cannot deny is cost. This is the problem which every hobbyist, Engineer or Professional has to face.
2.3 Literature Review

In [7], authors proposed security system to protect locker of bank by barcode and RFID. As soon as the system is powered up both RFID reader and the barcode scanner draw 5V supply from the microcontroller board. The Liquid Crystal Display (LCD) on the microcontroller board displays the message, "WELCOME TO SECURITY SYSTEM". Then the student needs to scan the ID card using the scanner. After completion of this process, the data stored in the barcode of student’s ID card is available at the one port of microcontroller. The RFID reader mounted on the door reads the data of the tags possessed by the respective student and sends it to the microcontroller at the other port. The number of tags possessed by the student is also recorded. The microcontroller is programmed to compare the two data received. If there is no mismatch between the data received by barcode scanner and RFID reader then green LED glows accompanied by the message “YOU MAY GO” on the LCD, along with the number of tags owned. In case of a mismatch between the two data a red LED glows and an alarm is raised with the text “STOP” displayed on the LCD.

In [8], authors proposed security system to protect locker of bank using Android application. The RFID reader reads the data from tag and send to the microcontroller, if the card is valid then microcontroller display the account holder name and number. Then the account holder need to enter the password, if the password is valid then microcontroller sends the Short Massages Service (SMS) to account holder mobile number. Then account holder sends the password to the microcontroller through mobile phone using Global System For Mobile Communication (GSM). The microcontroller compares the passwords entered by keyboard and received through mobile phone. If these passwords are correct the microcontroller provides necessary control signal to open the bank locker. This method is simple and more secure than other system.

In [9], authors proposed security system to protect locker of bank Using Mems and RFID Technology. The purported plan is drawn up of hardware
and software tools, the hardware used are RFID tag and reader, MEMS sensor, LCD, GSM module, all is interfacing with the help of ARM7 and the software is designated by the Keil μVison software. We offered the idea using RFID and MEMS technology along with GSM modem for authentication of clients. When an authorized person comes to the door which consists of RFID tags, the person has to insert the ID card in predefined space. The card contains a unique barcode number which is the identity of an authorized person. As soon as a person inserts the card, the system will ask for the sensing the motion; MEMS accelerometer can sense motion in 3 axes (X, Y and Z). Initially we have to specify the position in accelerometer which is united with the controller, and then the controller receives the input analogue signal and brings forth the respective ADC samples. When the controller sees that the ADC samples are paired with the predetermined value in a controlled. If the motion will not detect by the system within the predetermined time instantly message will forward to higher authority and the nearest police station. If the system will detect the motion, it will ask for password again if the user will not press correct password, a message, will send to the higher authority through the GSM technology and then the user enters the random password through the keypad interface. If the entered random password is matched, it will send the command signal to the driving circuit. Driving circuit receives the command and forwards the signal to unlock the doorway.

In [10], authors proposed security system to protect locker of bank Using Password, Thermal and Physical Interrupt Alarm. As it has been said before, it has 3 steps security layer, first of all is password protection. A pre-programmed password can be saved and modified if necessary by the user, only authorized persons will know the password. After pressing the password on the keyboard, if the password is accepted, the door will open and will disable all the alarm systems. If the password is not accepted it will ring a warning sound and alert the user. If wrong password is inserted more than three times, it will secure the door and alert the law enforcement team.
The second part of the security layer is physical interrupt alarm. A ‘Laser’ near to the door is aimed at a LDR sensor. If by any chance, anyone cut the bolts of the door or blows it away, the broken particle of the door or the door itself will cut the laser and will trigger the alarm.

The third part of the protection scheme is thermal security system. If anyone plans to get into the vault by breaking the vault floor or wall, the PIR sensor will detect the thermal movement and will alarm the trigger instantly. The second and third part of the protection scheme is only activated when the door is locked and de-activated when the door is unlocked.
3.1 System Description
The security system block diagram of the locker bank is shown in Figure 3.1. The system is mainly consists of Arduino, RFID, keypad, LCD, Buzzer, stepper motor and ULN2003A driver.

![Block diagram for the circuit](image)

Figure 3.1: Block diagram for the circuit

3.2 System Hardware Considerations
The system contain of many components. The main components are:

3.2.1 Radio frequency identification
Radio requency identification is a form of wireless communication that uses radio waves to identify and track objects. It is an automatic identification technology where digital data encoded in an RFID tag is read by the RFID reader. An RFID system consists of a reader device and a tag (transponder). A tag has a unique serial number which is identified by the reader. RFID
takes the barcoding concept and digitizes it for the modern world providing the ability to:

- Uniquely identify an individual item beyond just its product type.
- Identify items without direct line-of-sight.
- Identify many items (up to 1,000s) simultaneously.
- Identify items within a vicinity of between a few centimeters to several meters.

An RFID system has readers and tags that communicate with each other by radio. RFID tags are so small and require so little power that they don’t even need a battery to store information and exchange data with readers. This makes it easy and cheap to apply tags to all kinds of things that people would like to identify or track.

**-Definition of protection system by RFID**

The safe banking system is one of major concern of bank authority. And locker accessing is a major one. We are using a unique technique for this. Here, the software using Arduino controls all the hardware components. The main objective of the system is to uniquely identify and to make sure that right authorized person accessing the locker in bank. This requires a unique product, which has the capability of distinguishing different person. This is possible by the new emerging technology RFID. The main parts of an RFID system are RFID tag (with unique ID number) and RFID reader (for reading the RFID tag), as shown in Figures 3.2 and 3.3.

This technique provides automatic and more secured solution for bank locker system. Instead of mechanical lock and keys we are using RFID card for locking and unlocking the system.

![Figure 3.2: RFID reader](image)
-RFID products classification

RFID products classify into two basic categories:

- Passive tags do not have batteries and have indefinite life expectancies.
- Active Tags are powered by batteries and either have to be recharged, have their batteries replaced or be disposed of when the batteries fail.

3.2.2 Arduino mega

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 as shown in Figure 3.4. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16MHz crystal oscillator, a USB connection, a power jack 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 an AC-to-DC adapter or battery to get started. The mega 2560 essentially is a larger form of the standard Arduino (UNO, duemilanove, etc.) [11].

- Features

- 54 digital I/O pins (14 of which provide PWM output).
- 3.3V supply generated by the on-board regulator.
- 16 analog input pins.
- 256kB of flash memory.
- 16MHz clock speed.
- Can supply 40 mA of DC current per pin.
The Contents and pins of Arduino shown in Figure 3.5.

![Arduino mega 2560](image)

**Figure 3.4: Arduino mega 2560**

The Arduino mega 2560 has specific features and the Table 3.1 illustrates this.

**Figure3.5: Arduino mega pins**

**-Features of Arduino mega 2560**

The Arduino mega 2560 has specific features and the Table 3.1 illustrates this.

**Table 3.1: Features of Arduino mega 2560**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microcontroller</td>
<td>ATmega2560</td>
</tr>
<tr>
<td>Operating Voltage</td>
<td>5V</td>
</tr>
<tr>
<td>Input Voltage (recommended)</td>
<td>7-12V</td>
</tr>
<tr>
<td>Input Voltage (limits)</td>
<td>6-20V</td>
</tr>
<tr>
<td>Digital I/O Pins</td>
<td>54 (of which 15 provide PWM output)</td>
</tr>
<tr>
<td>Analog Input Pins</td>
<td>16</td>
</tr>
</tbody>
</table>
### DC Current per I/O Pin
40 Ma

### DC Current for 3.3V Pin
50 Ma

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash Memory</td>
<td>256 Kb (of which 8kB used by bootloader)</td>
</tr>
<tr>
<td>SRAM</td>
<td>8 kB (ATmega328)</td>
</tr>
<tr>
<td>EEPROM</td>
<td>4 kB (ATmega328)</td>
</tr>
<tr>
<td>Clock Speed</td>
<td>16MHz</td>
</tr>
</tbody>
</table>

## 3.2.3 Liquid crystal display

LCD screen is an electronic display module and find a wide range of applications. A 16x2 LCD is a very basic module and is very common in various devices and circuits. They are preferred over seven segment displays. There are many advantages when compared to seven segment displays. They are: LCDs can display characters, numbers and even graphics [12]. A 16x2 LCD means it can display 16 characters per line and there are two 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. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special and even custom characters (unlike in seven segments), animations and so on.

This LCD shown in Figure 3.6.
3.2.4 Stepper motor

A stepper motor is a brushless DC motor whose rotor rotates in discrete angular increments when its stator windings are energized in programmed manner. Rotation occurs because of magnetic interaction between rotor poles and poles of the sequentially energized stator windings. The rotor has no electrical windings, but has salient and/or magnetized poles, synchronous electric motor that converts digital pulses into mechanical shaft rotations. Each rotation of a stepper motor is divided into a set number of steps, sometimes as many as 200 steps. The stepper motor must be sent a separate pulse for each step. The stepper motor can only receive one pulse and take one step at a time and each step must be the same length. Since each pulse results in the motor rotating a precise angle- typically 1.8 degrees - you can precisely control the position of the stepper motor without any feedback mechanism.

As the digital pulses from the controller increase in frequency, the stepping movement converts into a continuous rotation with the velocity of the rotation directly proportional to the frequency of the control pulses. Stepper motors are widely used because of their low cost, high reliability, and high torque at low speeds. Their rugged construction enables you to use stepper motors in a wide environmental range.

-Types of stepper motors

There are three kinds of step motors: permanent magnet, hybrid, and variable reluctance. Hybrid step motors offer the most versatility and combine the best characteristics of variable reluctance and permanent magnet stepper motors. Hybrid stepper motors are constructed with multi-toothed stator poles and a permanent magnet rotor. A standard hybrid stepper motor has 200 rotor teeth and rotates 1.8 degrees per step. Hybrid stepper motors provide high static and dynamic torque and they run at very high step rates. Applications for hybrid stepper motors include computer disk drives and cd players. Hybrid stepper motors are also widely used in industrial and
scientific applications. Hybrid step motors are used in robotics, motion control, automated wire cutting, and even in high-speed fluid dispensers.

- **Step modes**

Stepper motor "step modes" includes full step, half step, and microstep. The type of step is dependent on the stepper motor driver controlling the stepper motor. Many stepper motor controllers are multi-step capable (usually adjusted by switch setting). The use of stepper motor in this project is to mimic the locker operation.

### 3.2.5 ULN2003 motor driver

The ULN2003 is known for its high-current, high-voltage capacity, as shown in Figure 3.7. The drivers can be paralleled for even higher current output. Even further, stacking one chip on top of another, both electrically and physically, has been done. Generally it can be used for interfacing with a stepper motor, where the motor requires high ratings which cannot be provided by other interfacing devices.

![ULN2003 motor driver](image)

**Figure 3.7: ULN2003 motor driver**

These chips keep the power that drives the motors separate from the power that is on the Arduino. The Arduino can't provide enough juice to power the stepper motors directly. This is why you have to use separate chips to sort of act as valves that control how the motor spins. Another benefit that stepper driver chips provide is that they provide fractional steps. This helps smooth
out the motion of the stepper motor. The motor has 4 coils of wire that are powered in a sequence to make the magnetic motor shaft spin. When using the full-step method, two of the four coils are powered at each step. The default stepper library that comes pre-installed with the Arduino IDE uses this method. The driver with stepper motor shown in Figure 3.8.

![Driver with stepper motor](image)

**Figure 3.8: Driver with stepper motor**

### 3.2.6 Keypad

4x4 matrix membrane keypad, this 16-button keypad provides a useful human interface component for microcontroller projects, as shown in Figure 3.9. Convenient adhesive backing provides a simple way to mount the keypad in a variety of applications. The Key Specifications are:

- Maximum rating: 24 VDC, 30 mA.
- Interface: 8-pin access to 4x4 matrix.
- Operating temperature: 32 to 122 °F (0 to 50 °C).

Most of the applications of embedded systems require keypads to take the user inputs, especially in case where an application requires more number of keys. With simple architecture and easy interfacing procedure, matrix keypads are replacing normal push-buttons by offering more inputs to the user with the lesser I/O pins.
3.2.7 Electronic buzzer

A buzzer is a device that provides audio signaling functions, as shown in Figure 3.10. The devices may be mechanical or electronic or a combination of the two.

Figure 3.10: Electronic Buzzer

3.3 System Software Considerations

The system could be enter contained within a proteus simulation and programmed by C++ language.

3.3.1 System simulation

Computer simulation has become a useful part of mathematical modelling of many natural systems to observe their behavior. It allows the engineer to test the design before it is built in the real situation. The simulations of this research were performed in proteus program as shown in Figure 3.11.
Figure 3.11: The main circuit design

By clicking play button, simulation will run. While the circuit is running, the LCD displays "USE CARD" as shown in Figure 3.12.

Figure 3.12: The main circuit design after running
3.3.2 System code
The code was written in Arduino by C++ language, the code of this research is in the appendix.

3.3.3 Working principle of circuit
The control circuit of this project aims to protect a locker bank by using RFID, the system allows to some people to enter to the office that include the bank locker, but not all of them allowed to open the locker bank(such the people who work in cleaning). Each one of these people have Specific card. If a person is allowed to open a bank locker, by using the card, the LCD displays "enter password" shown in Figure 3.13.

![Image of circuit diagram]

Figure 3.13: The circuit with the allowable card

If the password is true the bank locker will open for few times and then closed as shown in figure 3.14.
Figure 3.14: The circuit with true password

Else the system allows to the person to try three times by display "try again" as shown in Figure 3.15.

Figure 3.15: The circuit with first and second wrong trials
After three wrong times the LCD display "wrong" and the buzzer sends a high voice to indicate that there is someone trying to steal the bank locker as shown in Figure 3.16.

![Figure 3.16: The circuit with wrong password](image)

If the person is not allowed to open a bank locker, the LCD display "Not Authorized ", and the bank locker will not open, as shown in Figure 3.17.

![Figure 3.17: The circuit with the nonallowable card](image)
CHAPTER FOUR
SYSTEM IMPLEMENTATION AND TESTING

4.1 System Implementation

There are five steps to connected Arduino with components of circuit. They are:

4.1.1 Step one
In this step connected the RFID with the Arduino through pins (18, 19, VCC and ground), as shown in Figure 4.1.

![Figure 4.1: Installation RFID with Arduino](image)

4.1.2 Step two
In this step connected the keypad with the Arduino through pins (2, 3, 4, 5, 8, 9, 10 and 11) as shown in Figure 4.2.
4.1.3 Step three

In this step connected the LCD with the Arduino through pins (A8, A9, A10, A11, A12, A13, VCC and ground) as shown in Figure 4.3.
4.1.4 Step four
Connected the motor with the Arduino through pins (A0, A1, A2, A3, VCC and ground) as shown in Figure 4.4.

![Figure 4.4: Installation motor with Arduino](image)

4.1.5 Step five
In this step connected the buzzer with the Arduino through pin 12 and ground as shown in Figure 4.5.

![Figure 4.5: Installation buzzer with Arduino](image)
4.2 System Testing

To test this system there are three cases as follows:

4.2.1 Case one (detecting authorized tag)

When the RFID reader detects the authorized tag, a message will displayed asking for password, as shown in Figure 4.6.

Figure 4.6: Testing of true card

4.2.2 Case two (entering true password)

If the entered password is true the locker will open as shown in Figure 4.7.

Figure 4.7: Bank locker opening
And then closing after a little time as shown in Figure 4.8.

Figure 4.8: Bank locker closing

4.2.3 Case three (detecting nonauthorized card)
When the RFID reader detects the nonauthorized tag, a message will displayed "Not Authorized" as shown in Figure 4.9.

Figure 4.9: Implementation when the card not allow to open a bank locker
CHAPTER FIVE
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion
The model has been designed and the software has been implemented successfully, which shows the work of security system of locker bank to increase the safety of the bank locker and protection of thefts.
After testing and designed of bank locker security system by RFID is found to be very suitable so it contribute to solve the problems, as well as that of the control circuit is easy to design and programming.

5.2 Recommendations
The following points may be taken as suggested future work:
- Use of a fingerprint to increase security issues.
- Increase of RFID system sensitivity.
- Use of camera to monitoring the system.
REFERENCES:

# APPENDIX

## SYSTEM CODE

```c
#include <SoftwareSerial.h>
#include <Keypad.h>
#include <LiquidCrystal.h>

SoftwareSerial mySerial(6, 7);
unsigned char rfid[12];
unsigned char card1[12]="1E007D06C3";
unsigned char card2[12]="1E007CA5E3";
unsigned char card3[12]="1E007CC11A";
unsigned char card4[12]="1E007CD591";
unsigned char card5[12]="1E007CF9B6";
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
  {'1','4','7','*'},
  {'2','5','8','0'},
  {'3','6','9','#'},
  {'A','B','C','D'}
};
byte rowPins[ROWS] = {5, 4, 3, 2};
byte colPins[COLS] = {11, 10, 9, 8};
LiquidCrystal lcd(A8, A9, A10, A11, A12, A13);
Keypad myKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS) ;
unsigned char keys[6];
unsigned char password1[7]="012345";
unsigned char password2[7]="123456";
unsigned char password3[7]="234567";
int buzz=12;
```
int i, count1, count2, count3, count4, count5, count6, j;
int count, count7, count8, card, match;
int wrong, wrong1, wrong2, wrong3, why;

void setup()
{
  pinMode(buzz, OUTPUT);
  mySerial.begin(9600);
  pinMode(A0, OUTPUT);
  pinMode(A1, OUTPUT);
  pinMode(A2, OUTPUT);
  pinMode(A3, OUTPUT);
  lcd.begin(16, 2);
  Serial1.begin(9600);
  delay(500);
  lcd.setCursor(0, 0);
  lcd.clear();
  lcd.print("USE CARD....");
}

void loop()
{
  why=0;
  if (Serial1.available()>0){
    rfid[i] = Serial1.read();
    i++;
    if (i == 11)
      {
        lcd.setCursor(0, 0);
        lcd.print("Checking...");
        for (j=0; j<=i; j++)
          if (rfid[j] == card1[j]) {
            delay(5);
            count1++;
          }
      }
if (rfid[j] == card2[j]) {
delay(10);
count2++;
}
if (rfid[j] == card3[j]) {
delay(10);
count3++;
}
if (rfid[j] == card4[j]) {
delay(10);
count4++;
}
if (rfid[j] == card5[j]) {
delay(10);
count5++;
}
delay(10);
}
i=0;
if (count1 >= 10 ){
card=1;
lcd.setCursor(0, 0);
lcd.print("CARD1.. ");
digitalWrite(buzz,HIGH);
delay(300);
digitalWrite(buzz,LOW);
count1=0;
passwordCheck();
}
count1=0;
if (count2 >= 10 ){
card=2;
lcd.setCursor(0, 0);
lcd.print("CARD2... ");
digitalWrite(buzz,HIGH);
delay(300);
digitalWrite(buzz,LOW);
count2=0;
passwordCheck();
}
count2=0;
if (count3 >= 10 ){
card=3;
lcd.setCursor(0, 0);
lcd.print("CARD3");
digitalWrite(buzz,HIGH);
delay(300);
digitalWrite(buzz,LOW);
count3=0;
passwordCheck();
}
count3=0;
if (count4 >= 10 ){
lcd.setCursor(0, 0);
lcd.print("Not Authorized ");
for (j=0;j<4;j++){
digitalWrite(buzz,HIGH);
delay(200);
digitalWrite(buzz,LOW);
delay(200);
}
lcd.setCursor(0, 0);
lcd.print("USE CARD.... ");
count4=0;
}
count4=0;
if (count5 >= 10 ){
lcd.setCursor(0, 0);
lcd.print("Not Authorized ");
for (j=0;j<4;j++){
digitalWrite(buzz,HIGH);
delay(200);
digitalWrite(buzz,LOW);
    delay(200);
}
lcd.setCursor(0, 0);
lcd.print("USE CARD.... ");
lcd.setCursor(0, 1);
lcd.print(" ");
count5=0;
}
count5=0;
}
}

28 void passwordCheck()
{
lcd.setCursor(0, 0);
lcd.print("Enter Password");
lcd.setCursor(0, 1);
lcd.print(" ");
lcd.setCursor(0, 1);
while(1){
    if (why > 0)
        break;
}
char Key = myKeypad.getKey();
if (Key) {
    keys[i] = Key;
    lcd.print(Key);
    i++;
    delay(200);
    match = 0;
    if (i == 6)
        compare();
} 
}
}
29 void compare()
{
    while (i == 6)
    {
        for (j = 0; j <= i; j++)
        {
            delay(200);
            if (keys[j] == password1[j])
            {
                count++;
                delay(50);
            }
        }
        if (keys[j] == password2[j])
        {
            count7++;
            delay(50);
        }
        if (keys[j] == password3[j])
        {
            count8++;
            delay(50);
        }
if (count == 6 && card==1)
    match = 1;
if (count7 == 6 && card==2)
    match = 2;
if (count8 == 6 && card==3)
    match = 3;
if (match > 0 && match < 4)
{
    lcd.setCursor(0, 0);
    lcd.print("Correct... ");
    stepper();
    lcd.setCursor(0, 0);
    lcd.print("USE CARD.... ");
    lcd.setCursor(0, 1);
    lcd.print(" ");
    i=0;
    count=0;
    why++;
    delay(200);
    break;
}
if (count < 6 && card==1 ){
    wrong1++;
    wrong = 1;
}
if (count7 < 6 && card==2 ){
    wrong2++;
    wrong = 2;
}
if (count8 < 6 && card==3 ){
wrong3++;  
wrong = 3;  
}  
if ( wrong > 0 && wrong <4){  
lcd.setCursor(0, 0);  
lcd.print("Wrong...    ");  
why=0;  
if (wrong1 > 2 || wrong2 > 2 || wrong3 > 2)  
{
    lcd.setCursor(0,1);  
lcd.print("      ");  
for (i=0; i < 10;i++)  
{
    digitalWrite(buzz,HIGH);  
delay(200);  
digitalWrite(buzz,LOW);  
delay(200);  
}  
count6=0;  
count7=0;  
count8=0;  
wrong1=0;  
wrong2=0;  
wrong3=0;  
lcd.setCursor(0, 0);  
lcd.print("USE CARD.....  ");  
lcd.print("        ");  
count =0;  
i=0;  
why=5;  
break;  
}
lcd.setCursor(0, 1);
lcd.print("Try again.. ");
delay(500);
lcd.setCursor(0, 1);
lcd.print(" ");
if (wrong1 < 3 && card==1){
i=0;
count=0;
passwordCheck();
}
if (wrong2 < 3 && card==2){
i=0;
count=0;
passwordCheck();
}
if (wrong3 < 3 && card==3){
i=0;
count=0;
passwordCheck();
count =0;
i=0;
}
i=0;
count=0;
void stepper()
{
  lcd.setCursor(0, 0);
lcd.print("OPPENING.... ");
for (j=0;j<=200;j++){
  for (i=A3;i>=A0;i--){

 44
{  
digitalWrite(i,HIGH);
delay(5);
digitalWrite(i,LOW);
}

delay(2000);
lcd.setCursor(0, 0);
lcd.print("CLOSING.... ");
for (j=0;j<=200;j++){
for (i=A0;i<A4;i++)
{
    digitalWrite(i,HIGH);
delay(5);
digitalWrite(i,LOW);
}
}
delay(2000);  
}