RFID SMART ATTENDANCE SYSTEM BASED ON ARDUINO

A graduation Project document submitted to the Dep. Of Statistics & Computer science as partial fulfillment for the Requirement for the Degree of B.SC in Faculty of Science

By:

Alaa Eldin Apuelsoad Mohamed

Mohamed said Mohamed

Supervised by:

Dr. **Rashad Ragab**

JULY 12, 2021

FACULTY OF SCIENCE HELWAN UNIVERSITY

Helwan university,

Faculty of science,

Department statistics & computer science

RFID SMART ATTENDANCE SYSTEM BASED ON ARDUINO

By:

Alaa Eldin Apuelsoad Mohamed

Mohamed said Mohamed

Supervised by:

Dr. **Rashad Ragab**

July 12, 2021

Acknowledgment

I would like to express my sincere gratitude to several individuals for supporting me throughout my Graduate project. First, I wish to express my sincere gratitude to my supervisor, Professor Rashad Ragab, for his enthusiasm, patience, insightful comments, helpful information, practical advice, and unceasing ideas that have helped me tremendously at all times in my project and writing of this thesis. His immense knowledge, profound experience and professional expertise in Data Quality Control has enabled me to complete this research successfully. Without his support and guidance, this project would not have been possible. I could not have imagined having a better supervisor in my study.

I also wish to express my sincere thanks to the University of Helwan for teaching me elighting my mind for four academic years.

Also, I am grateful to my family for encouraging me during my studies since I was young until this moment when I graduated.

In the end, I thank my friends who helped me in many things from the beginning of planning this project until its completion.

Abstract

In recent years, RFID technology has been widely used in various sectors, such as in-education, transportation, agriculture, animal husbandry, store sales and other sectors. RFID utilization in education is student attendance monitoring system, by using Internet of Things (IoT) and Cloud technology, it will produce a real time attendance monitoring system that can be accessed by various parties, such as lecturer, campus administration and parents. With this monitoring system if there are students who are not present can be immediately discovered and can be taken immediate action and the learning process can run smoothly.

Table of Contents

Nomenclature

DB ………………… Database

FR ………………… Functional requirements

GPIO ………………… General Purpose Input/output

I2C ………………… Multi-Master Bus

LCD ………………… Liquid-Crystal Display

NFR ………………… Non-functional requirements

IoT ………………… Internet of things

RFID ………………… Radio-frequency identification

SPI ………………… Serial-Peripheral Interface

SyRS ………………… System Requirements Specification

Api ………………… Application Programming Interface

LIST OF FIGURES

List of tables

1. Introduction

The emergence of electronic paradigm for learning compared to traditional method and availability of almost all information on the information superhighway(Internet), nowadays have caused students to be less motivated to come to the lecture rooms than ever before. Laziness on the part of students, nonchalance to schoolwork, extra social activities that have no importance in aiding the objectives of the institution and a lot more, may prevent students from attending lectures. Sequel to these, lecturers and administrators in most developing countries have had to come up with ways to ensure a healthy participation from students, and make sure that the student-lecturer interactive relationship is kept intact. This in some cases have come in simple forms like roll calls, while in more interesting cases, can be formats like surprise quizzes, extra credit in class, etc. These strategies are however time consuming, stressful, and laborious because the valuable lecture time that could otherwise been used for lectures is dedicated to student attendance taking and sometimes not accurate.

In addition to all these challenges, the attendances are recorded manually by the tutor and therefore are prone to personal errors. There arises a need for a more efficient and effective method of solving this problem. A technology that can solve this problem and even do more is the RFID technology. RFID is an automated identification and data collection technology, that ensures more accurate and timely data entry. RFID is not actually a new technology; it only quickly gained more attention recently because of its current low cost and advances in other computing fields that open up more application areas. RFID combines radio frequency and microchip technologies to create a smart system that can be used to identify, monitor, secure and do object inventory. At their simplest, RFID systems use tiny chips called ―tags that contain and transmit some piece of identifying information to an RFID reader, a device that in turn can interface with computers. The ability of RFID systems to deliver precise and accurate data about tagged items will improve efficiency and bring other benefits to business community and consumers alike in the not distant future. In this paper, we present an intelligent RFID based lecture attendance access control and management system tailored around Nigerian Universities' Commission (NUC) policy of ensuring a 70% course attendance by students for a course before likelihood of writing a semester examination for any course. The application of RFID Technology to student course attendance monitoring problem especially in developing countries in our proposition will lead to elimination or reduction of the quality time wasted during manual collection of attendance, creation of a student database management system that is not prone to errors or being manipulated by anyone and above all aids in better management of classroom statistics for allocation of attendance scores in the final grading of student performance in a particular course.

2. Literature survey

In the process of system development, literature reviews conducted to understand the theory, methods and technologies associated with systems that have been developed. Background research on the organization and comparative studies of existing systems is also done to understand the system requirements before the system was developed . Student Attendance Using RFID System is an automatic record of student attendance developed especially for universities.

It is generally said that the roots of radio frequency identification technology can be traced back to World War II. The Germans, Japanese, Americans, and British were all using radar which had been discovered in 1935 by Scottish physicist Sir Robert Alexander Watson-Watt to warn of approaching planes while they were still miles away. The problem was there was no way to identify which planes belonged to the enemy and which were a country's own pilots returning from a mission. Radio Frequency Identification (RFID) research and discovery began in earnest in the 1970s. RFID is commonly used to transmit and receive information without wires. RFID readers and tags communicate through a distance using radio waves. There are a lot of advantages in RFID system, included their price, size, memory capacity and their capability. Advances in radar and RF communications systems continued through the 1950s and 1960s. Electronic article surveillance tags, which are still used in packaging today, have a 1-bit tag. The bit is either on or off. If someone pays for the item, the bit is turned off, and a person can leave the store. But if the person does not pay and tries to walk out of the store, readers at the door detect the tag and sound an alarm .

The First RFID Patents Mario W. Cardullo claims to have received the first U.S. patent for an active RFID tag with rewritable memory on January 23, 1973 . Later, companies developed a low-frequency (125 kHz) system, featuring smaller transponders. A transponder encapsulated in glass could be injected under the cows' skin. This system is still used in cows around the world today. Low frequency transponders were also put in cards and used to control the access to buildings .

Today, 13.56 MHz RFID systems are used for access control, payment systems (Mobile Speedpass) and contactless smart cards. They are also used as an anti-theft device in cars. A reader in the steering column reads the passive RFID tag in the plastic housing around the key. If it does not get the ID number it is programmed to look for, the car won't start.

In this project, we will focus on using RFID technology to record student attendance at lectures by using 13.56 RFID card for each student.

And to know whether it is possible to implement this project in Egypt, especially at Helwan University, or not?

3. Software Development

3.1. Definition

Developing a software might be challenging, as it has many different processes for it to be able to succeed. It’s consist of four Phases:-

- Planning phase
- Analysis phase
- Development phase
- Implementation phase

Starting with the planning phase, its purpose is to find the main problem and determine ideas and solutions that will need to be solved. Followed by functional requirements and specifications phase, where system analysis and design takes place and that is by discussing and creating how the system shall be working and what the customer is expecting, with the combination of this last two, the system will meet the requirements of its customer and satisfy both the user and the client. After those phases, the next procedure is the software or system development. This is the most important phase of the whole process because without it there would be nothing but just empty words on a piece of paper. The development phase is the critical step of the process. The real work begins because it marks the end of the initial section. Once the third phase is completed, the integration and the testing phases will start. It is involving the system integration and testing which usually are carried out with QA, and it is where the QA professionals ascertain that the design and the whole system meets the initial set of the system goals and client’s needs. All those phases are mandatory to have in a system, so that it will be working perfectly and meets all the desired criteria.

3.1.1. Planning phase

Planning the system might be one of the most challenging phases, it is the first phase in the development process. The expectation for this project was to have a simple student attendance system that would record their attendance for any lecture with the saving their time and date of signing in into a database Excel sheet.

While preparing the planning stage, an opening report for the project has been written where all steps have been discussed.

The phase includes naturally the main idea of this thesis which was to develop an attendance system using an RFID MFRC522 model, an Arduino UNO model and an I2C LCD display.

The general idea of the system was that the user will sign in using their RFID tag or card, the I2C LCD will display the action of the user, and in case the user does not exist the LCD will display a message warning that the RFID fob or card is not saved in a user's name in the database.

3.1.2. Analysis phase

The analysis phase is also the part of the project where you identify the overall direction that the project will take through the creation of the project strategy documents.

This phase is the most important and demanding of all of the phases, it is an important part of the software design and development. A well-designed ground work for the functional requirements and specifications are the key elements for of the whole process of the design and development.

The more time spent on this phase, the happier customers are with the final product and the easier it is for the developer to develop the system. It is important to note that the system analysis takes place so that the new system can match the needs of the end users and meet their requirements. In addition, this program has to be precise and easy to use.

System & functional requirement are the main attraction of the analysis phase.

3.1.2.1. Systems requirements

User story: user of the system

The user story is a part of the software development. It is informative as it describes one or more features of a user using the system. The user story is typically written from the system user's or end user's perspective.

❖ Here are the user stories for this thesis project:

- As a system administrator, I can add/overwrite a new student to the system so that the attendance system will recognize the user and the RFID fob or card destinated to this person.
- As a system administrator, I should always be able to help the users of the attendance system in case they faced difficulties or forgot to sign in or out from the system.
- As a student, I need to always carry my RFID tag or card to be able to record my attendance in the system.

➢ Use case example

In system requirements, a use case is derived from user stories. Its diagram is the main form for a new, but yet undeveloped system. It shows and clarifies the outcome, and in particular, what the system will do as it specifies how the system will work. In order to achieve the goal and the outcome, it is important to have the use case and show the actual actions and steps that the user has to follow.

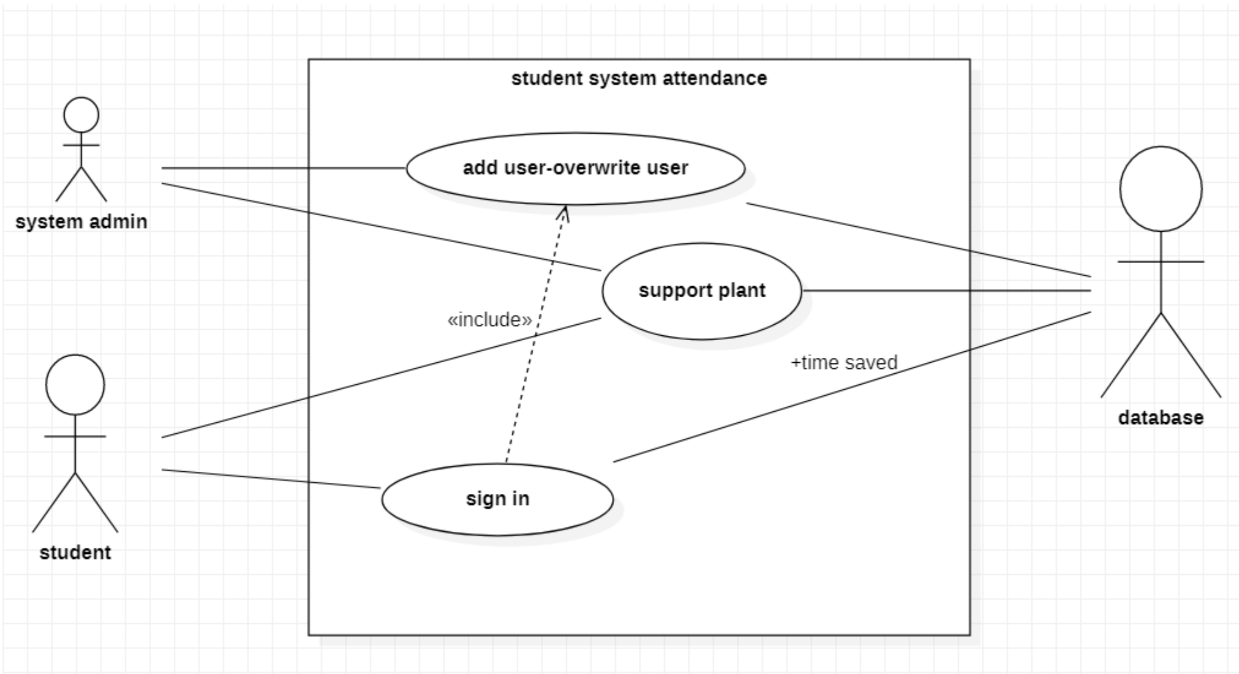


Figure 3-1

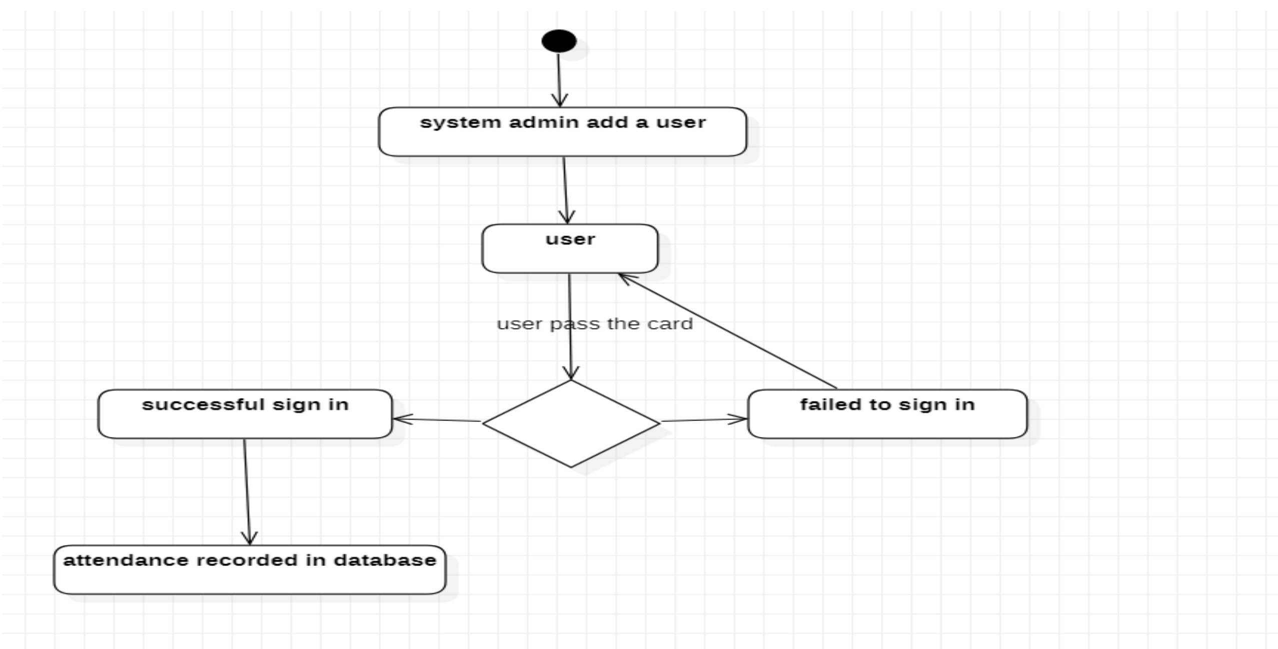


Figure 3-2

➢ Flowchart presence using IoT-based RFID

Presence process as illustrated in the flowchart in the next Figure starting from the RFID Tag scanning process using RFID Reader, the data obtained will be compared to the database, if the data match the database then the presence data will be stored in the cloud database, but if the data is not suitable then will be asked to scanning again RFID Tags, data stored in the cloud database is Student ID data, date and time of attendance, courses and on what week of attendance.

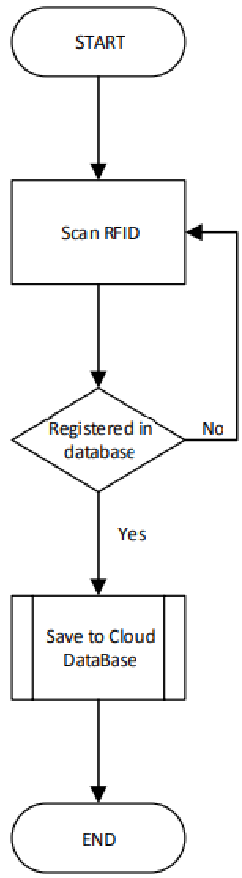


Figure 3-3

3.1.2.2. Functional requirements and specifications phase

System requirements specification (SyRS)

- Functional requirements (FR)
  - The system shall record log-in times of all users into the database.
  - The system needs to identify each user with his/her own RFID fob or card.
  - The system shall provide and record the data for logged in by the user.
  - The I2C LCD display needs to show the current status of the RFID fob or card when user signs in the system.
  - The system needs throw an error message on the I2C LCD display when the RFID fob or card is not recognized.
  - The system administrator should be able to sign in or out the user manually if the user forgot to use the system.
  - The system administrator should be able to correct the time of signing in or out manually for the use.
  - The system administrator shall have the ability to edit the RFID fob/card ID's in the database.
- Non-functional requirements (NFR)
  - The microcontroller along with the database should always be on so that the system could work fully.
  - The system needs to be running on an Arduino UNO, using an RFID MFRC522 model, an I2C LCD display.
  - The administrator has the responsibility to set up tag/card ID's in the database so that the system can operate as intended.

3.1.3. Development phase

Development stage is considered the most important phase of the system development cycle. Although the planning, the system requirements, and specifications phases are the base of all the next steps to succeed, the system development phase remains the principal and major. In view of the fact, it marks the end of the initial section of the entire process.

In this thesis project, based on the student management system need, it was discussed by the commissioner of this thesis that the design and the planning of the software was free to be chosen the best way possible for it to be ideal for the development and for the users.

The first step of the development phase was to have an open source microcontroller Arduino UNO to start programming and define the components on Arduino to make it suitable to our system.

Specify a database Excel sheet that will stored the data recorded by the user and connect it to Arduino.

Here the real work begins. The first step was to specify the components that will be used in the system and explain how it can work, connect it with Arduino?

3.1.4. Implementation

The implementation phase involves putting the project into action. It's here that the project manager will coordinate and direct projects resources to meet the object of the project plan.

The project phase is where you and your project team actually do the project work to produce the deliverables.

The system will consist of an:-

- Arduino
- RFID MFRC522 module
- I2C LCD.
- Database

3.1.4.1. Arduino

Arduino and Open Source Computer Hardware and Software

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards 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 (based on Wiring), and the Arduino Software (IDE), based on Processing.

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 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, too, is open-source, and it is growing through the contributions of users worldwide.

❖ 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.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive :Arduino boards are relatively inexpensive compared to other microcontroller platforms.
- 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.
- • 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.
- 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 and improving it.

❖ The project is based on microcontroller board designs, manufactured by several vendors, using various microcontrollers. These systems provide sets of digital and analog I/O pins that can be interfaced to various expansion boards ("shields") and other circuits. The boards feature serial communications interfaces, including 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 the Processing project, which includes support for the C and C++ programming languages. The first Arduino was introduced in 2005, aiming to provide an inexpensive and 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.

❖ Official boards

The original Arduino hardware was manufactured by the Italian company Smart Projects. Some Arduino branded boards have been designed by the American companies SparkFun Electronics and Adafruit Industries. Seventeen versions of the Arduino hardware have been commercially produced to date.

❖ Arduino UNO

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), 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.. You can tinker with your Uno without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

" Uno" means one in Italian and was chosen to mark the release of Arduino Software integrated development environment (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

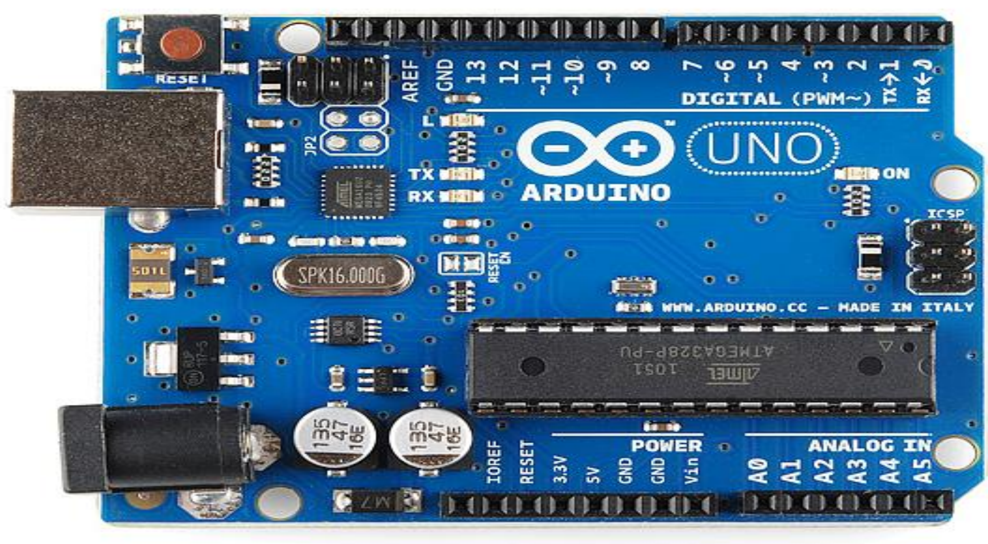


Figure 3-4

Pin Description

- Power pins: Vin, 3.3V, 5V,GND
  - Vin: Input voltage to Arduino when using an external power source.
  - 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA.
  - 5V: Regulated power supply used to power microcontroller and other components on the board.
  - GND: ground pins.
- Reset pin: Resets the microcontroller.
- Analog pins (A0-A5): Used to provide analog input in the range of 0-5V.
- Input/Output Pins (digital pins 0-13): Can be used as input or output pins.
- Serial pins (0(Rx), 1(Tx)): Used to receive and transmit TTL serial data.
- External Interrupts (2,3): To trigger an interrupt.
- PWM (3, 5, 6, 9, 11): Provides 8-bit PWM output.
- SPI (10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)): Used for SPI communication.
- Inbuilt LED (13): To turn on the inbuilt LED.
- TWI (A4 (SDA), A5 (SCA)): Used for TWI communication.
- AREF: To provide reference voltage for input voltage

❖ Arduino Uno Technical Specifications

Table 3.1

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

❖ Programming Arduino

To start programming Arduino, connect your Uno board to the computer and upload your first sketch. The Arduino Uno is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all boards and running both online and offline.

Running online by use your Arduino board on the Arduino wed IDE.

❖ Use your Arduino Uno on the Arduino Desktop IDE

If you want to program your Arduino Uno while offline you need to install the Arduino Desktop IDE The Uno is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all our boards. Before you can move on, you must have installed the Arduino Software (IDE) on your PC.

Connect your Uno board with an A B USB cable; sometimes this cable is called a USB printer cable.

The USB connection with the PC is necessary to program the board and not just to power it up. The Uno automatically draw power from either the USB or an external power supply. Connect the board to your computer using the USB cable. The green power LED (labelled PWR) should go on.

❖ Install board drivers

If you used the Installer, Windows - from XP up to 10 - will install drivers automatically as soon as you connect your board.

But if the board was not probably recognized, you should install it manually.

- Click on the Start Menu, and open up the Control Panel.
- While in the Control Panel, navigate to System and Security. Next, click on System. Once the System window is up, open the Device Manager.
- Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COMxx)". If there is no COM & LPT section, look under "Other Devices" for "Unknown Device".
- Right click on the "Arduino UNO (COmxx)" port and choose the "Update Driver Software" option.
- Next, choose the "Browse my computer for Driver software" option.
- Finally, navigate to and select the driver file named "arduino.inf", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub-directory). If you are using an old version of the IDE (1.0.3 or older), choose the Uno driver file named "Arduino UNO.inf"
- Windows will start installation.
- Select your board type and port.

Select the serial device of the board from the Tools | Serial Port menu. This is likely to be **COM3** or higher (**COM1** and **COM2** are usually reserved for hardware serial ports). To find out, you can disconnect your board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.

❖ Upload the program

Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX LEDs on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar. A few seconds after the upload finishes, you should see the pin 13 (L) LED on the board start to blink (in orange).

3.1.4.2. RFID communication

➢ **What is RFID?**

RFID stands for Radio Frequency Identification, Here digital data stored in RFID tags are captured by a reader via radio waves.

On January 23, 1973, Mario Cardullo invented the first true working ancestor of modern read and write radio frequency with a passive radio transponder memory. The initial device was powered by signal interrogation.

Nowadays, RFID is a technology used for tracking, identification, and detection. The technology is based on data storage in the RFID card or fob, and thanks to the radio waves that make the wireless reading of RFID reader possible.

The main idea of the RFID is that its tag or card is attached to the designated object. The data is then read and written from the fob or card using the RFID reader, and finally data is saved in the database.

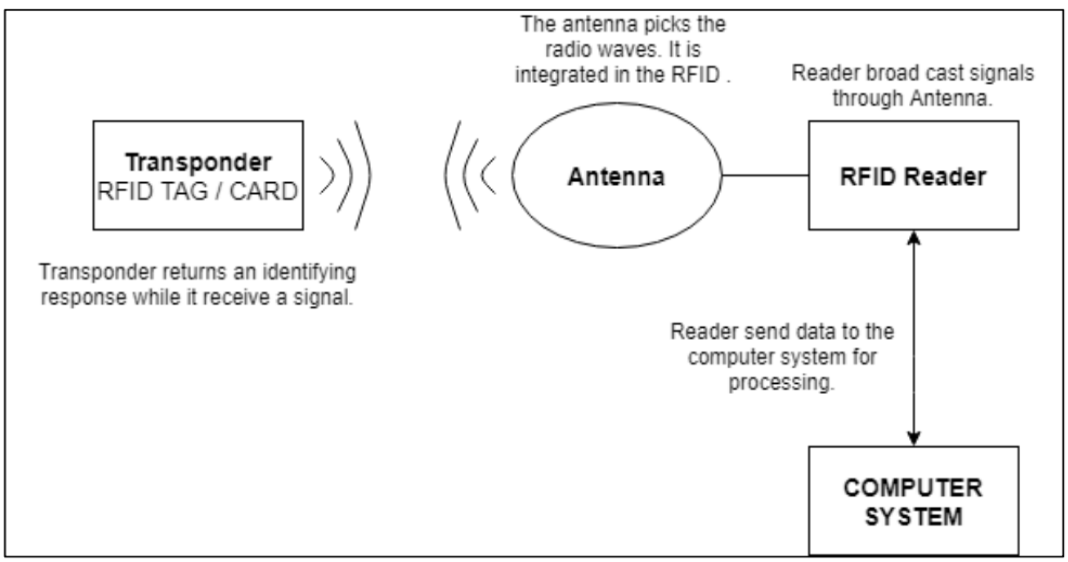


Figure 3-5

A Reader consists of a Radio Frequency module and an antenna which generates high frequency electromagnetic field. On the other hand, the tag is usually a passive device, meaning it does not contain a battery. Instead it contains a microchip that stores and processes information, and an antenna to receive and transmit a signal.

To read the information encoded on a tag, it is placed in close proximity to the Reader (does not need to be within direct line-of-sight of the reader). A Reader generates an electromagnetic field which causes electrons to move through the tag's antenna and subsequently power the chip.

The powered chip inside the tag then responds by sending its stored information back to the reader in the form of another radio signal. This is called backscatter. The backscatter, or change in the electromagnetic/RF wave, is detected, and interpreted by the reader which then sends the data out to a computer or microcontroller.

❖ **RFID usage**

➢ **RFID has three primary frequency ranges used for its transmissions.**

- Low frequency: it has a primary frequency range of 125 kHz to 134 kHz, which makes its read range from a 0 to 10 centimeters far away from the RFID reader. It is usually used for access control, human or animal tracking, car key-fob and other more similar fields.
- High frequency: with a primary frequency range of 13.56 MHz, which allows its read range from 0 to 30 centimeters. This frequency is mostly used in DVD kiosks, personal ID cards, NFC applications, library books and poker or gaming chips.
- Ultra-High frequency: has a primary frequency range of 433 MHz to 960 MHz The ultra-high frequency has two types of RFID, active and passive. The active RFID has a read range of 30 to +100 meters. It is usually used in auto manufacturing, asset or vehicle tracking, mining, and other similar areas. The passive RFID has a read range of 0 to 25 meters, and it is mostly used in electronic tolling, inventory tracking, pharmaceuticals and also manufacturing.

For the purposes of this project thesis, a high frequency 13.56 MHz RFID has been used.

❖ Hardware Overview

RC522 RFID Reader/Writer Module

The RC522 RFID module based on MFRC522 IC from NXP is one of the most inexpensive RFID options that you can get online for less than four dollars. It usually comes with a RFID card tag and key fob tag having 1KB memory. And best of all, it can write a tag, so you can store your some sort of secret message in it.

Figure 3-6

The RC522 RFID Reader module is designed to create a 13.56MHz electromagnetic field that it uses to communicate with the RFID tags (ISO 14443A standard tags). The reader can communicate with a microcontroller over a 4-pin Serial Peripheral Interface (SPI) with a maximum data rate of 10Mbps. It also supports communication over I2C and UART protocols.

The operating voltage of the module is from 2.5 to 3.3V, but the good news is that the logic pins are 5-volt tolerant, so we can easily connect it to an Arduino or any 5V logic microcontroller without using any logic level converter.

❖ complete specifications:

Table 3.2

Frequency Range	**13.56 MHz ISM Band**
Host Interface	SPI / I2C / UART
Operating Supply Voltage	2.5 V to 3.3 V
Max. Operating Current	13-26mA
Min. Current(Power down)	10µA
Logic Inputs	5V Tolerant
Read Range	5 cm

❖ RC522 RFID Module Pinout

The RC522 module has total 8 pins that interface it to the outside world. The connections are as follows:

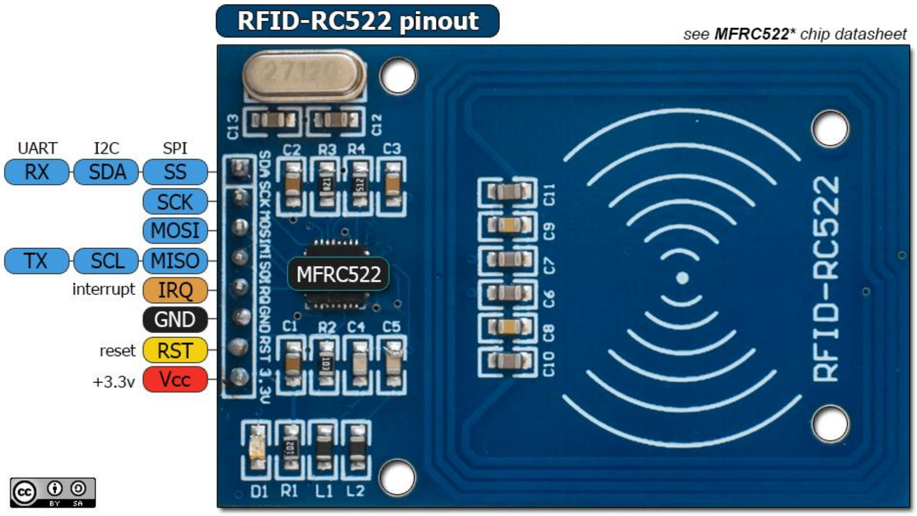


Figure 3-7

- **VCC:** supplies power for the module. This can be anywhere from 2.5 to 3.3 volts. You can connect it to 3.3V output from Arduino, connecting it to 5V pin will likely destroy your module.
- **RST:** is an input for Reset and power-down. When this pin goes low, hard power-down is enabled. This turns off all internal current sinks including the oscillator and the input pins are disconnected from the outside world. On the rising edge, the module is reset.
- **GND:** is the Ground Pin and needs to be connected to GND pin on the Arduino.
- **IRQ:** is an interrupt pin that can alert the microcontroller when RFID tag comes into its vicinity.
- **MISO/SCL/TX:** pin acts as Master-In-Slave-Out when SPI interface is enabled, acts as serial clock when I2C interface is enabled and acts as serial data output when UART interface is enabled.
- **MOSI(Master Out Slave In):** is SPI input to the RC522 module.
- **SCK(Serial Clock):** accepts clock pulses provided by the SPI bus Master i.e. Arduino.
- **SS/SDA/Rx:** pin acts as Signal input when SPI interface is enabled, acts as serial data when I2C interface is enabled and acts as serial data input when UART interface is enabled. This pin is usually marked by encasing the pin in a square so it can be used as a reference for identifying the other pins.

❖ Hardware overview

Wiring – Connecting RC522 RFID module to Arduino UNO.

After knowing everything about the module, we can begin connecting it to Arduino.

To start with, connect VCC pin on the module to 3.3V on the Arduino and GND pin to ground. The pin RST can be connected to any digital pin on the Arduino. In our case, it is connected to digital pin9. The IRQ pin is left unconnected as the Arduino library we are going to use does not support it.

Now we are remaining with the pins that are used for SPI communication. As RC522 module require a lot of data transfer, they will give the best performance when connected up to the hardware SPI pins on a microcontroller. The hardware SPI pins are much faster than 'bit-banging' the interface code using another set of pins.

Note that each Arduino Board has different SPI pins which should be connected accordingly. For Arduino boards such as the UNO those pins are digital 13 (SCK), 12 (MISO), 11 (MOSI) and 10 (SS).

Table 3.3

Pin	Wiring to Arduino Uno
SDA	Digital 10
SCK	Digital 13
MOSI	Digital 11
MISO	Digital 12
IRQ	unconnected
GND	GND
RST	Digital 9
3.3V	3.3V

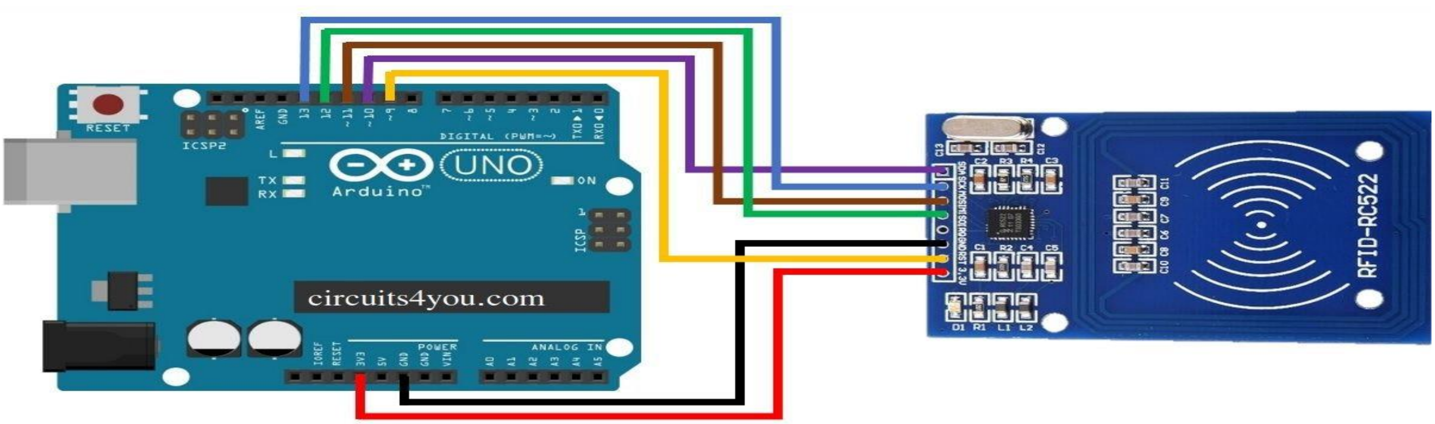


Figure 3-8

❖ Software overview

Arduino Code – Reading RFID Tag

There is a library called MFRC522 library which simplifies reading from and writing to RFID tags. Download the library first.

After having the circuit ready, go to File > Examples > MFRC522 > DumpInfo and upload the code. This code will be available in your Arduino IDE (after installing the RFID library).

Then, open the serial monitor. You should see something like the figure below:

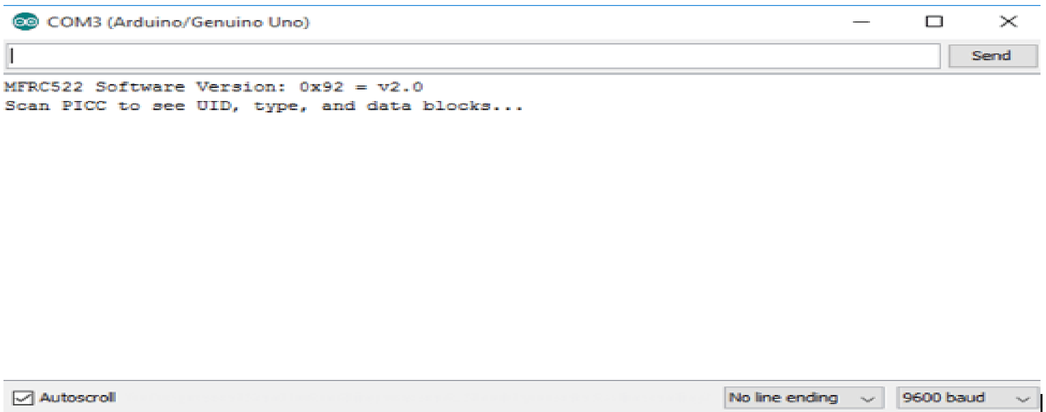


Figure 3-9

Approximate the RFID card or the keychain to the reader. Let the reader and the tag closer until all the information is displayed.

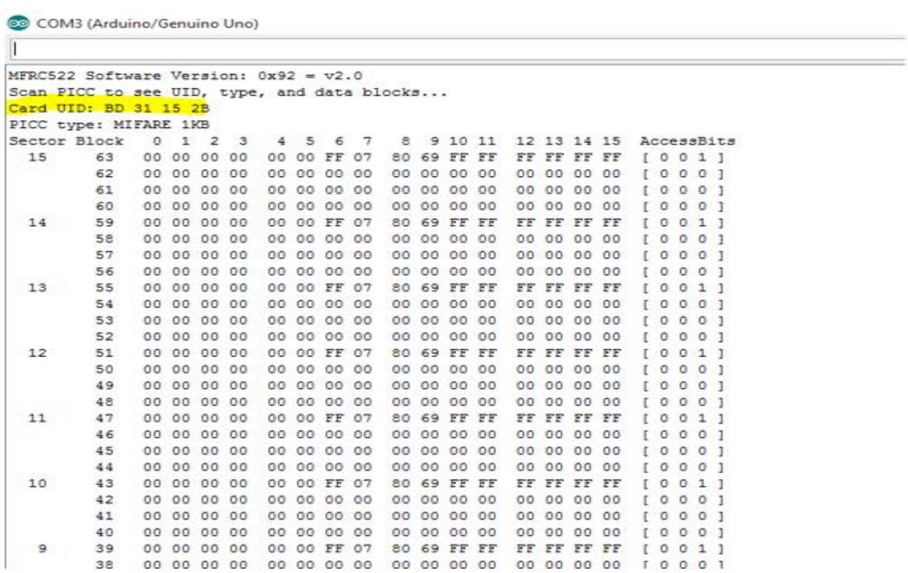


Figure 3-10

This is the information that you can read from the card, including the card UID that is highlighted in yellow. The information is stored in the memory that is divided into segments and blocks as you can see in the previous picture.

You have 1024 bytes of data storage divided into 16 sectors and each sector is protected by two different keys, A and B.

Note your card UID that is shown in the serial monitor because you will need it later.

❖ MIFARE Classic 1K Memory Layout

The 1K memory of the Tag is organized in 16 sectors (from 0 to 15)Each sector is further divided in to 4 blocks (block 0 to 3) Each block can store 16 bytes of data (from 0 to 15).

16 sectors x 4 blocks x 16 bytes of data = 1024 bytes = 1K memory

The whole 1K memory with sectors, blocks and data is highlighted below.

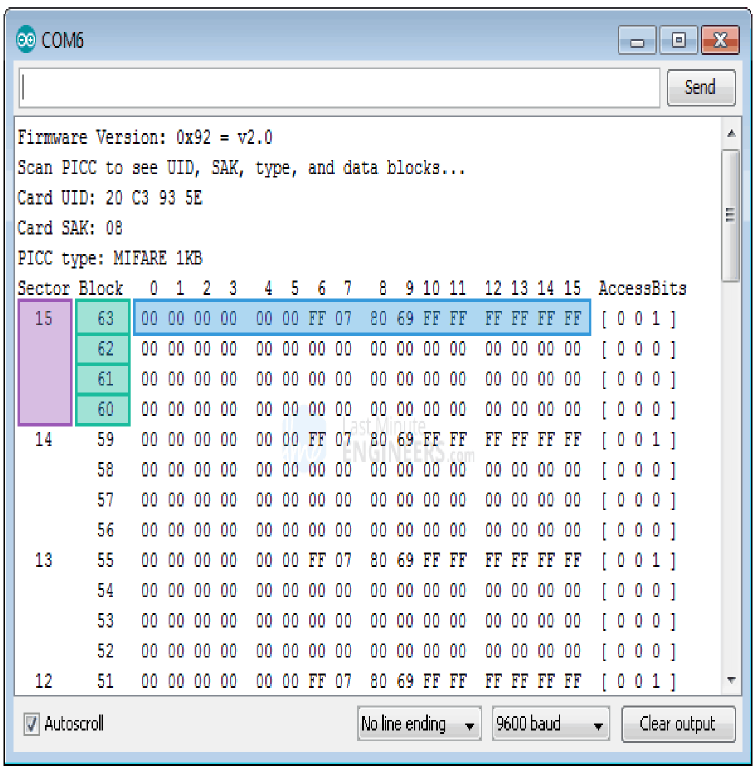


Figure 3-12

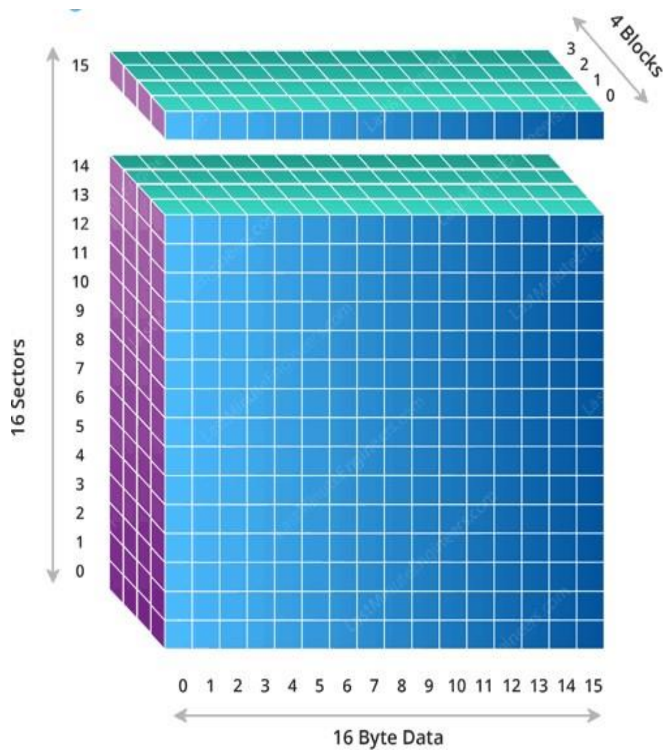


Figure 3-11

The Block 3 of each sector is called Sector Trailer and contains information called Access Bits to grant read and write access to remaining blocks in a sector. That means only the bottom 3 blocks (block 0, 1 & 2) of each sector are actually available for data storage, meaning we have 48 bytes per 64 byte sector available for our own use.

3.1.4.3. I2C LCD Overview

❖ **History of LCD**

The principles of liquid crystals were discovered in the late 1880s but work on Modern LCD displays did not begin until the mid-1960s. a number of patents were filed in the early 1970s and in 1973 the Sharp Corporation introduced LCD displays for calculators.

The first color LCD displays were developed in the early 1980s but production units were not commonly available until the mid-1990s. By the late 1990s LCD displays were quite common.

LCD televisions have been available since the beginning of the 21st century and by 2007 the image quality of LCD surpass that of CRTs.

❖ **LCD Displays for Experiments**

A number of LCD displays are available for experimenters. These low-cost monochrome displays are ideal for use with microcontrollers like the Arduino.

These displays are available in a number of different configurations. The part number for the display generally relates to the number of rows and columns in the display.

Common display configurations include 16 x 2 and 20 x 4. All of these displays are used in a virtually identical fashion the only difference being the number of columns and rows they have. We will be using a 20 x 4 display in our system .

Standard LCDs typically require around 12 connections, which can be a problem if you do not have many GPIO pins available. Luckily, you can also use I2C add on circuit so you can easily upgrade a standard LCD as well.

❖ LCD Pinouts

The LCD display has 16 solder pads that you can use in a number of ways:

- You can solder a connector directly to it. Typically a 16-pin male header is soldered here with the pins facing the back of the circuit board.
- You could solder wires directly to it, although this is not really recommended.
- You can attach an I2C adapter directly to it or you can use a female header strip to allow you to plug in the I2C adapter. More on this later.

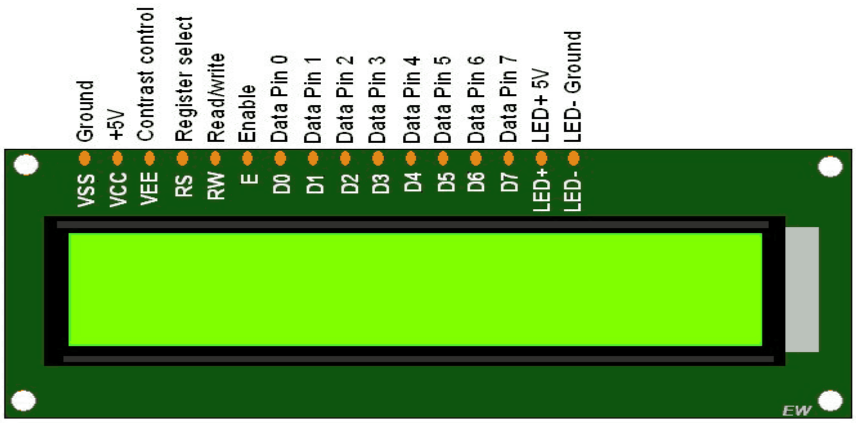


Figure 3-13

❖ The pin functions:

- **GND:** This is the Ground pin. On some modules it is labeled VSS.
- **5 VDC:** This is the 5 volt power connection. On some modules it is labeled VDD.
- **Brightness :**This is the input for the brightness control voltage, which varies between 0 and 5 volts to control the display brightness. On some modules this pin is labeled V0.
- **RS** :This is the Register Select pin. It controls whether the input data is meant to be displayed on the LCD or used as control characters.
- **RW:** Puts the LCD in either Read or Write mode. In most cases you will be using Read mode so this pin can be tied permanently to ground.
- **EN:** The Enable pin. When High it reads the data applied to the data pins. When low it executes the commands or displays the data.
- **D0**: Data input 0.
- **D1:** Data input 1.
- **D2**: Data input 2.
- **D3**: Data input 3.
- **D4:** Data input 4.
- **D5:** Data input 5.
- **D6:** Data input 6.
- **D7:** Data input 7.
- **A:** The Anode (positive voltage) connection to the backlight LED.
- **K:** The Cathode (ground or negative voltage) connection to the backlight LCD.

Because the LCD module uses a parallel data input it requires 8 connections to the host microcontroller for the data alone. Add that to the other control pins and it consumes a lot of connections. On an Arduino Uno half of the I/O pins would be taken up by the display, which can be problematic if you want to use the I/O pins for other input or output devices.

One way of reducing the number of connections attach the I2C adapter it is reduce the connections to 4-wire mode, and most projects that make use of this display do exactly that.

❖ I2C LCD Adapter

At the heart of the adapter is an 8-Bit I/O Expander chip – PCF8574. This chip converts the I2C data from an Arduino into the parallel data required by the LCD display.

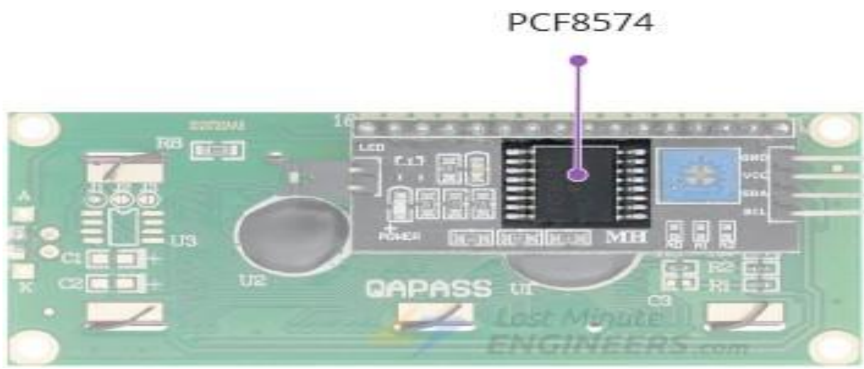


Figure 3-14

The board also comes with a small trim pot to make fine adjustments to the contrast of the display.

Figure 3-15

In addition, there is a jumper on the board that supplies power to the backlight. To control the intensity of the backlight, you can remove the jumper and apply an external voltage to the header pin that is marked as 'LED'.

❖ I2C LCD display Pinout

An I2C LCD has only 4 pins that interface it to the outside world. The connections are as follows:

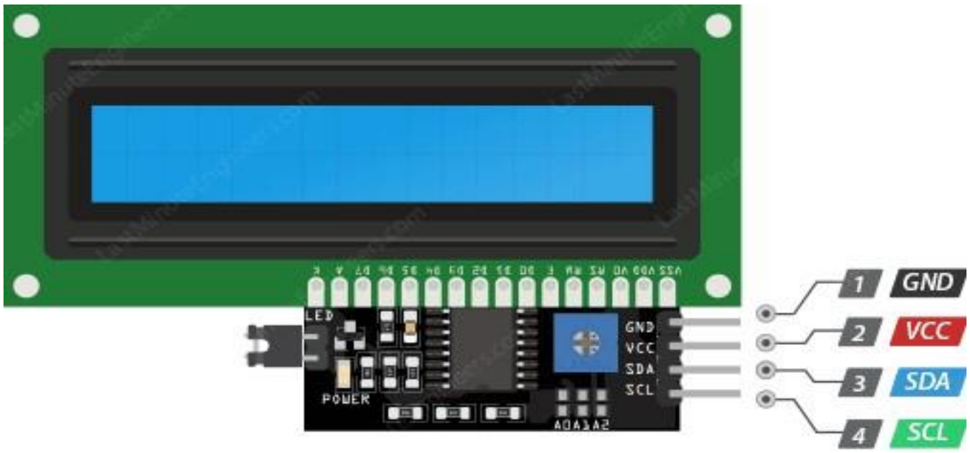


Figure 3-16

- **GND:** is a ground pin and should be connected to the ground of Arduino.
- **VCC:** supplies power to the module and the LCD. Connect it to the 5V output of the Arduino or a separate power supply.
- **SDA:** is a Serial Data pin. This line is used for both transmit and receive. Connect to the SDA pin on the Arduino.
- **SCL:** is a Serial Clock pin. This is a timing signal supplied by the Bus Master device. Connect to the SCL pin on the Arduino.

It is much easier to connect an I2C LCD than to connect a standard LCD. You only need to connect 4 pins instead of 12. Start by connecting VCC pin to the 5V output on the Arduino and connect GND to ground.

Now we are remaining with the pins that are used for I2C communication. Note that each Arduino Board has different I2C pins which should be connected accordingly. On the Arduino boards with the R3 layout, the SDA (data line) and SCL (clock line) are on the pin headers close to the AREF pin. They are also known as A5 (SCL) and A4 (SDA).

❖ Hardware overview

I2C LCD setup

Table 3.4

I2C CHARACTER LCD	ARDUINO
GND	GND
VCC	5 V
SDA	A4
SCL	A5

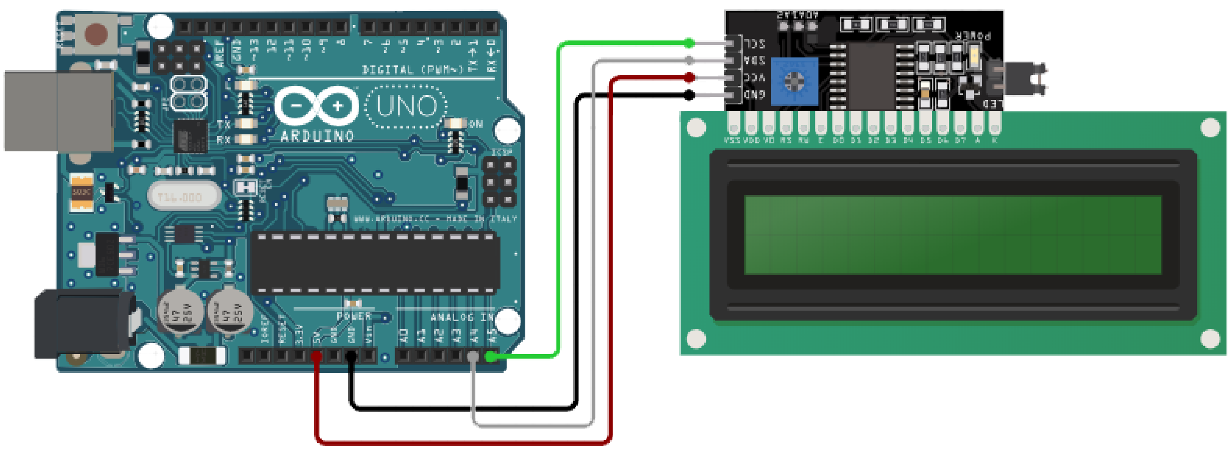


Figure 3-17

❖ Adjusting The LCD Contrast

After you wire the LCD, you will need to adjust the contrast of the display. On the I2C module, you will find a potentiometer that you can turn with a small screwdriver.

Plug in the USB connector of the Arduino to power the LCD. You should see the backlight light up. Now rotate the potentiometer until the first line of the rectangle appears.

❖ I2C LCD Display - Library Installation

In order to run the subsequent sketches you will need to install a library called LiquidCrystal_I2C. This library is an improved version of the LiquidCrystal library that comes packaged with your Arduino IDE.

To install the library navigate to the Sketch > Include Library > Manage Libraries Wait for Library Manager to download libraries index and update list of installed libraries.

❖ Software overview

Determining the I2C Address

The I2C address of your LCD depends on the manufacturer, as mentioned earlier. If your LCD has a PCF8574 chip from Texas Instruments, its default I2C address is $0x27_{Hex}$. If your LCD has a PCF8574 chip from NXP semiconductors, its default I2C address is $0x3F_{Hex}$.

So your LCD probably has an I2C address $0x27_{Hex}$ or $0x3F_{Hex}$. Nevertheless it is recommended that you find out the actual I2C of the LCD before using.

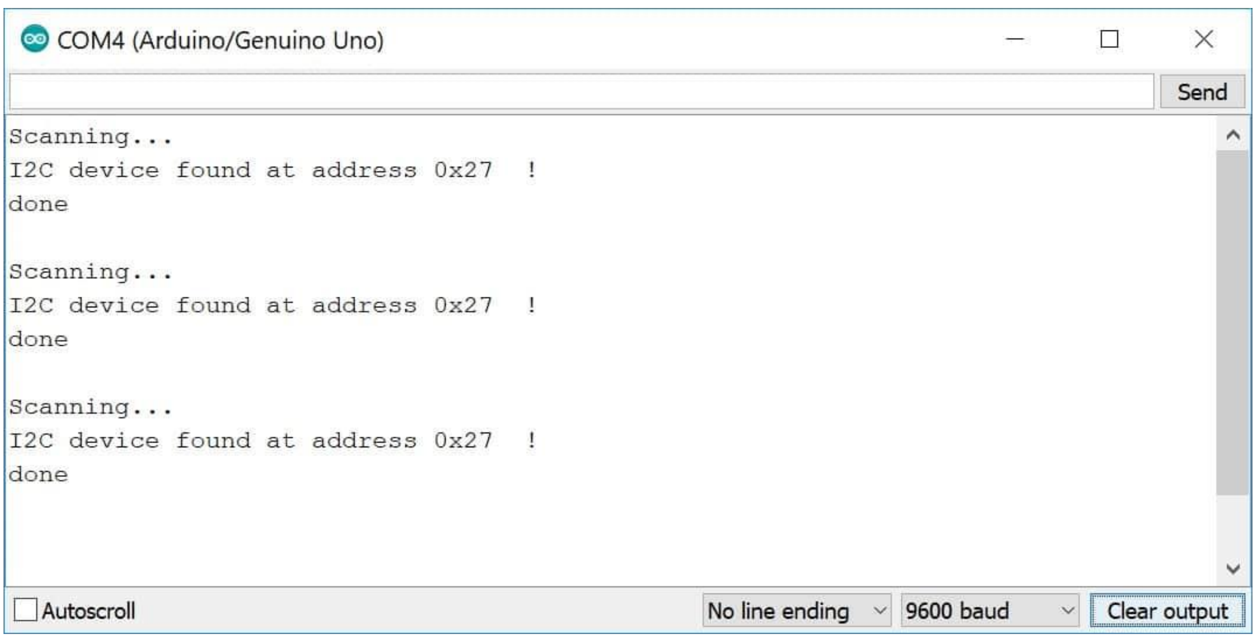
Luckily there is a simple way to do this, A simple I2C scanner sketch that scans your I2C bus and gives you back the address of each I2C device it finds.

The complete I2C scanner sketch is given below:

```
/*I2C_scanner
  This sketch tests standard 7-bit addresses.
  Devices with higher bit address might not be seen properly.*/

#include <Wire.h>

void setup() {
  Wire.begin();

  Serial.begin(9600);
  while (!Serial);
  Serial.println("\nI2C Scanner");
}

void loop() {
  byte error, address;
  int nDevices;

  Serial.println("Scanning...");

  nDevices = 0;
  for (address = 1; address < 127; address++ ) {
    Wire.beginTransmission(address);
    error = Wire.endTransmission();

    if (error == 0) {
      Serial.print("I2C device found at address 0x");
      if (address < 16)
        Serial.print("0");
```

```
        Serial.print("0");
      Serial.print(address, HEX);
      Serial.println("  !");

      nDevices++;
    }
    else if (error == 4) {
      Serial.print("Unknown error at address 0x");
      if (address < 16)
        Serial.print("0");
      Serial.println(address, HEX);
    }
  }
  if (nDevices == 0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");

  delay(5000);
}
```

Load this sketch into your Arduino then open your serial monitor. You will see the I2C address of your I2C LCD display.



We see in the serial monitor that the I2C address we will use in the project is 0x27 we will need it in the subsequent sketches.

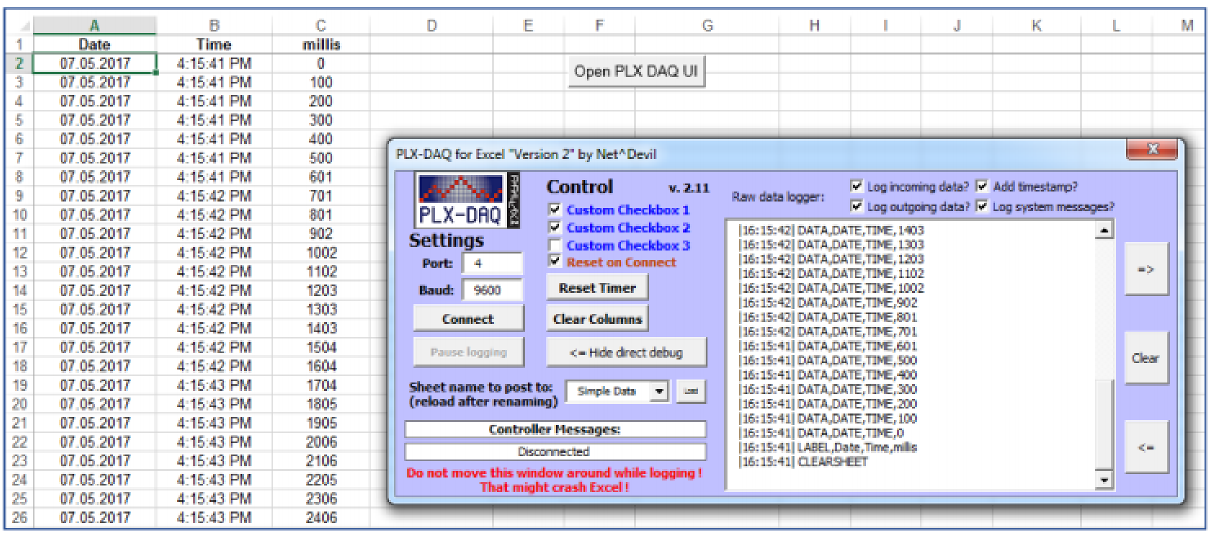
3.1.4.4. Database

Connect Arduino to database by using PLX DAQ v2 program

PLX DAQ v2 is a program used to establish an easy communication between Microsoft Excel on a Windows Computer and any device that supports serial port protocol. It was intentionally written to allow communication between Arduino and Excel.

You can, for example, measure temperature data with your Arduino, send the results to Excel every 10 seconds, printed the data on a sheet and draw a graph with all information. All communication will be done by **Serial.println** commands just like the commands you use to send from Arduino to monitor in your Arduino IDE Serial Monitor.

And the output will be looks like the following in Excel:

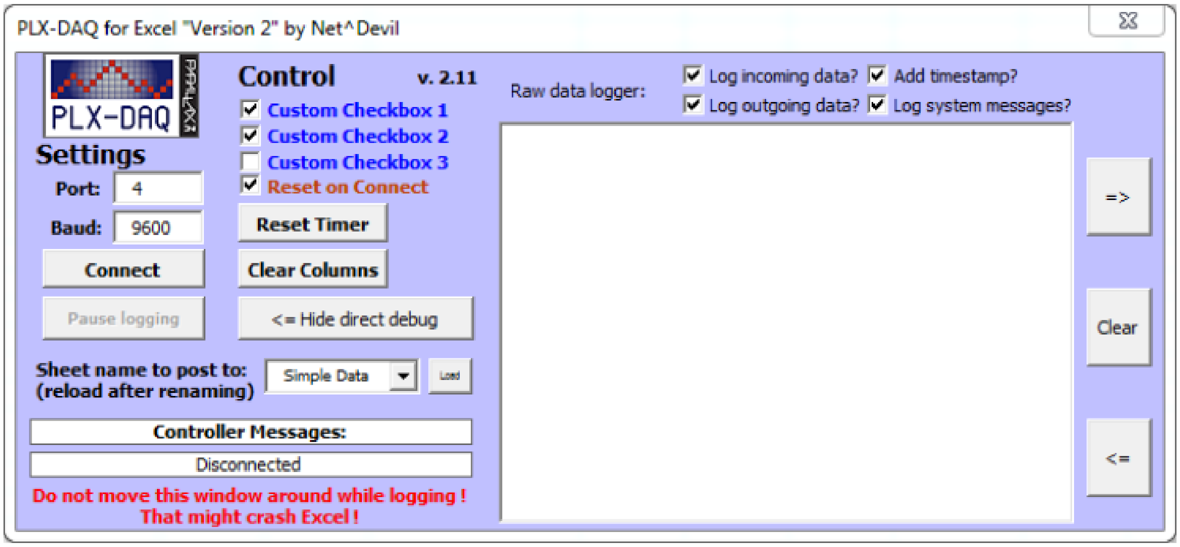


❖ **How to use the program?**

The program uses two parts to work: the special Microsoft Excel Spreadsheet with the PLX DAQ v2 UI and commands plus any Arduino device that sends special commands for communication. The latest Excel Spreadsheet can be downloaded online from internet, you should download the latest version.

After opening the Excel Spreadsheet please allow running the macro (yellow warning message at the top). Afterwards you can see the PLX DAQ v2 UI. In case not please click the button "Open PLX DAQ UI" on the "Simple Data" sheet.

❖ The UI offers the following options:



1. Port: set to Arduino port (same as in Arduino IDE => Tools => Port, e.g., 4 for COM4.
2. Baud: set to the baud rate you run your Arduino on (e.g., 9600 if you are using Serial.begin(9600); in your Arduino code).
3. Connect: connects to your Arduino and starts logging.
4. Pause logging/resume logging: when connected will pause the logging of data.
5. Reset Timer: will set the Timer to 0. The Timer can be used to measure how long Excel is already logging.
6. Clear Columns: will delete all logged data from the sheet. Won't clear the labels of the columns.
7. Display/Hide direct debug: will show or hide the text field on the right. The Direct Debug Window can be used to manually monitor commands received by PLX DAQ v2 in Excel.
8. Sheet name to post to: this will list all sheets in the Excel workbook. Whatever sheet you select in the dropdown the logged data will be posted to it. This sheet will be referred to as the "ActiveSheet" throughout this document. Note: after adding / deleting sheets please press the small "Load" button on the left side of the dropdown box for the sheet list to be updated.
9. Controller Messages: in the field below the most recent commands and status information will be shown. Most likely the information is changing way too fast for you to read, thus use the Direct Debug Window.
10. Reset on Connect: the checkbox should be ticked at all time. If ticket the first command from Excel to Arduino will be to restart, thus your code starts from the beginning as well. This way you can have a fresh session. If you want to connect to your Arduino without restarting it just untick the box.
11. Custom Checkbox 1/2/3: these can be used to control your Arduino during run in any way you want. There are commands to label the Checkboxes by your Arduino and to query the state of the boxes. You could for example label one box "Measure humidity as well?" and check on demand in Excel if you want your Arduino to measure humidity with a second sensor next to only measure e.g., temperature. There are special commands Arduino can use to query the status of the checkboxes. More detail on these can be found below.

12. l. Log incoming data? checkbox: information received from Arduino will be displayed in the direct debug window. Hint: disable if you experience performance issues.
13. Log outgoing data? checkbox: information send to your Arduino will be displayed in the direct debug window. Hint: disable if you experience performance issues.
14. Log system messages? checkbox: information from Excel will be display in the direct debug window (e.g., errors) . Hint: disable if you experience performance issues.
15. Add timestamp? checkbox: will add a timestamp to every information logged in the direct debug window. This is handy for debugging.
16. =>: will increase the size of the direct debug window to a certain maximum.
17. <=: will decrease the size of the direct debug window to a certain minimum.
18. Clear: will clear all information in the direct debug window.

❖ The Arduino part – overview

For PLX DAQ v2 to work correctly your Arduino needs to send specially formatted commands. All commands need to be send from Arduino to the PC using the Serial.println commands. These commands can include parameters, variables, and functions to send to as well. These parameters need to be separated by commas. This can be done like this:

Serial.println((String) "DATA,DATE,TIME," + millis());

➢ These commands can be split up into different categories:

1. Basic setup and communication: Commands here are used to format the sheet to log to and to send data to the sheet.
2. Specific communication and manipulation: Commands here are used to work with further parameters, jump on or between sheets, and using checkboxes for “communication” to your Arduino.
3. Excel workbook commands: Commands here are used to control the logging process or even saving workbooks in window.
4. Miscellaneous commands: Everything that is not really crucial or does not have any benefit (anymore).

3.2. The attendance system setup

❖ Hardware overview

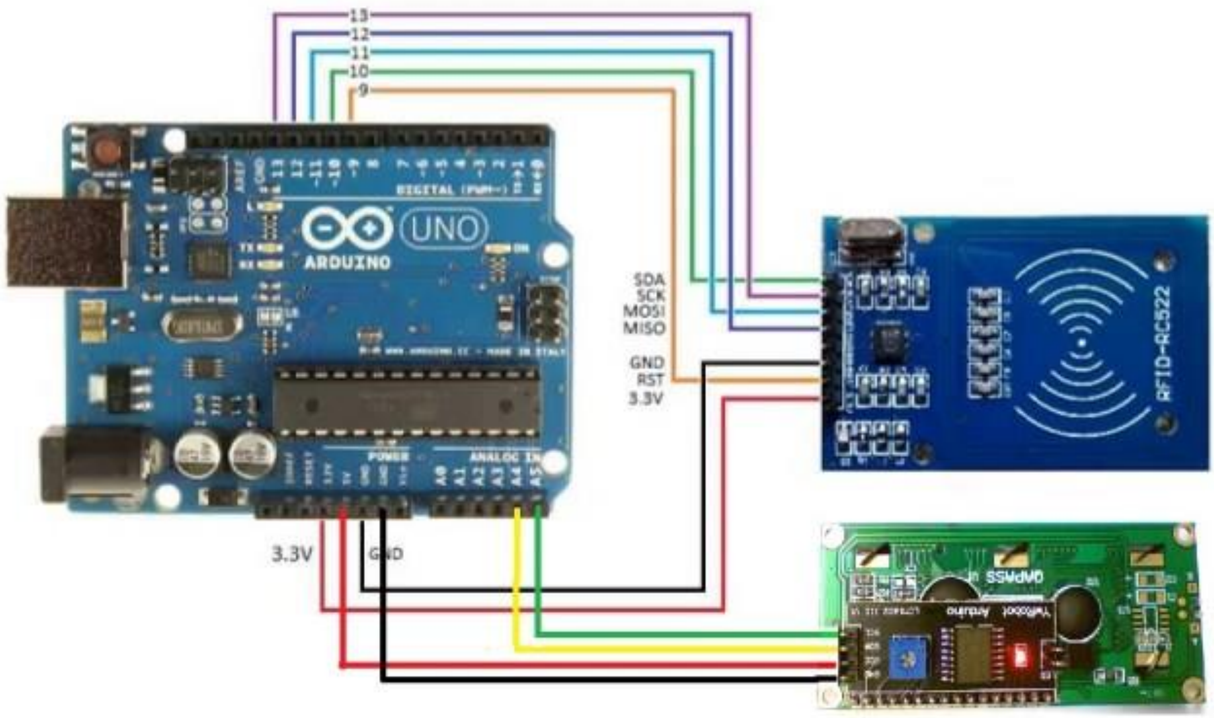


Figure 3-18

At this step, the attendance system needed to be exactly as described in the previous steps and the final diagram in the previous Picture.

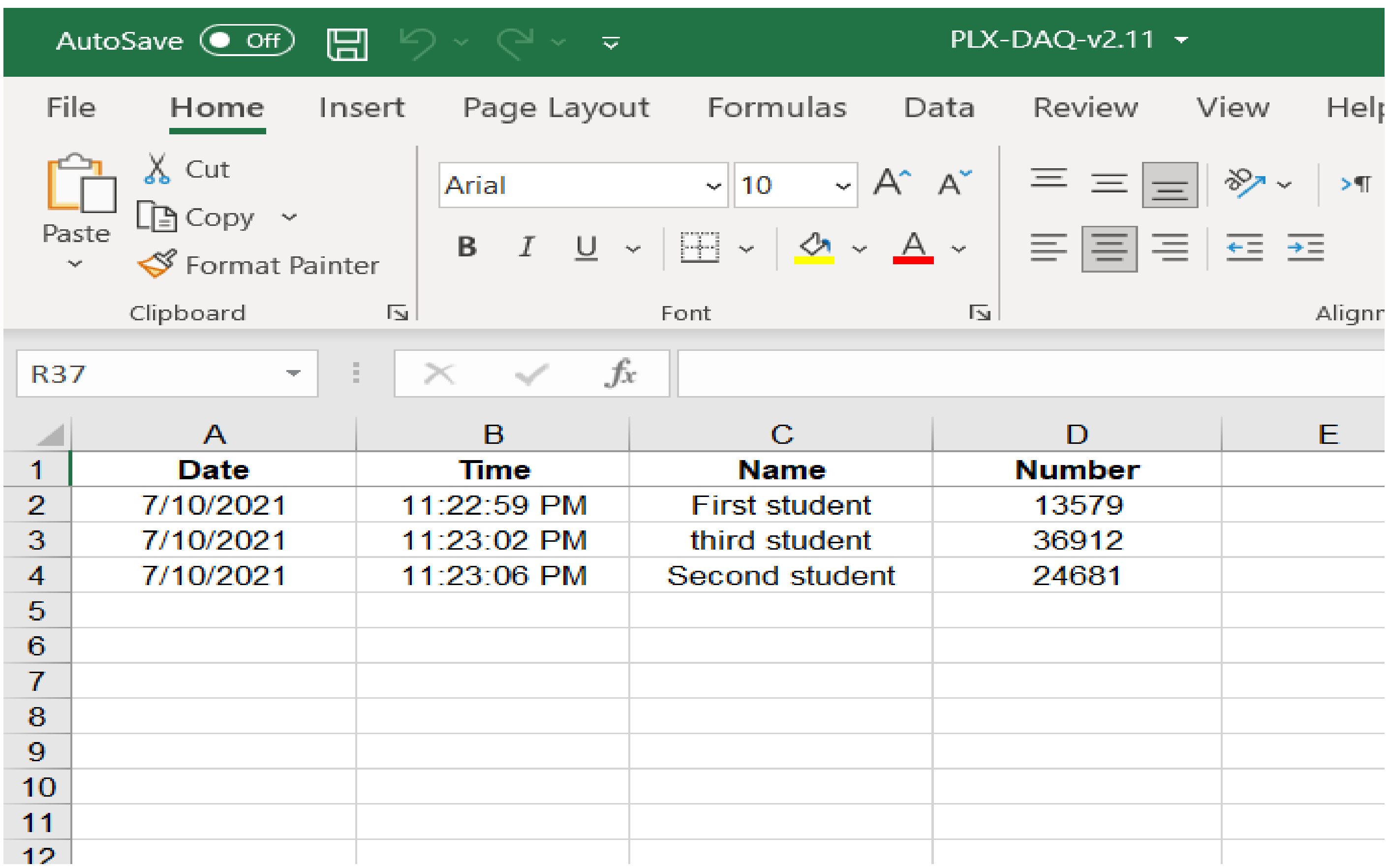
Since we previously tested each component alone to prove that all are working. We will now need to combine and develop our code of what is needed and connect the attendance system to the database. The system will run in the window Arduino environment.

4. Result

We have 3 students each one of them has a RFID card which is carry the student UID name and number .

When student's card is close enough to be energized by the RC522 reader. If the UID of the tag matches a predefined value card UID that is stored in Arduino memory, the student data (date, time , name , number) will be stored in the database sheet Excel. And if we scan any unknown card that is not stored in the Arduino memory, it would not be stored in the database.

We see here that when the 3 students pass their own cards on the reader their data would be stored in the sheet Excel like this.



Date	Time	Name	Number
7/10/2021	11:22:59 PM	First student	13579
7/10/2021	11:23:02 PM	third student	36912
7/10/2021	11:23:06 PM	Second student	24681

Figure 4-1

4.1. Testing

Once the development phase is finished, the testing phase will follow. This phase is very important in any system development because through testing we can assure the quality of the final product. Is the final product working perfectly with no bugs and errors? Does the final product meet the criteria of the principal product? These are just some questions that must be answered.

It is hard to achieve all those needs in only one version of the system software development. Although, it may seem possible to achieve all what was discussed in the planning phase, Thus, changing the outcome or procedure of the project might need to be considered.

When developing and testing this type of attendance system prototype; everything is working perfectly with no errors or bugs for one user, but once you add another user to the system, it somehow confuses the ”users_attendance” table in the database by filling data of user 2 into data of user 1. If user 1 sign in with his card the user 2 cannot sign in because his data was declared in database by user1 card. which declares that the system and database cannot make the difference between the two users it is kind of error and bugs which need to fix. For this, future development is needed to this system so that multi-users will be able to save their attendance with correctly.

For this, future development is needed to this system so that multi-users will be able to save their attendance with correctly.

5. Cost analysis

5.1. Starting cost

Table 5.1

ITEMS	QUANTITY	PRICE	COST
ARDINO UNO	1	135 LE	135 LE
RFID MFRC522	1	105 LE	105 LE
RFID 13.56 MHZ CARD	4	8 LE	4*8=32LE
LCD 20X4	1	100 LE	100 LE
I2C	1	35 LE	35 LE
JUMPERS AND WIRES	1	22 LE	22 LE
TOTAL COST			430

6. Availability & Developability

After applying and testing this simulation for the project, we find that it is very possible to implement this project in Egypt, especially in Helwan University.

Certainly, this project will contribute to the progress of the university and save time, resources, and effort in order to record the attendance of students because until this moment the registration of student attendance at lectures is still based on the traditional method by writing each student his name on the attendance sheet, which consumes a lot of effort and inaccuracy and certainly consumes a lots of time wasting of the lecture.

I am not saying that this simulation is flawless, but at least it will reduce errors and waste time and effort in the current system.

7. Conclusion

The goal of this thesis was to develop a prototype of an attendance system using Arduino UNO microcontroller, RFID MFRC522, I2C LCD Display Those components would make for the thesis commissioner a full attendance system where the user signs in with saving that data read from the RFID reader to the database. In doing so, this has been successfully created to perform with one to multiple users.

Having in mind the initial idea that the rising demand for solutions in today's world regarding student tracking, using RFID readers have become more consistent. It is important that universities monitor their students by using the clock in as method as presented here, instead of students writing their own name on paper to record their attendance for lectures.

On a general note, the stages and planning for the project was not what was anticipated. By starting with the initial idea of creating a user interface for adding users would have meant that the project had started from the wrong step of the development. As a result, the procedures were changed through the course of the project by overcoming such obstacles.

In conclusion, the project fulfilled the needs of a basic attendance system, as this will go further by it being developed further by the commissioner. Through proper code adjustments, revisions, and tests, the product can be developed to its full capabilities. In doing so, the project will be implemented along the university and along with future possibilities of using it also at other schools and locations.

As mentioned in the testing phase, the next step of this project would be to add a user interface for the database. Firstly, the current layout is quite difficult for the end user to use, especially for those with a non-technical background, so in the future creating a simpler user interface (UI) is needed. This will include a webpage containing the information regarding users from the database which allows the easy accessibility and reading of each individual user's data separately. Secondly, the system is only a prototype and future development is needed to streamline the performance of multiple users.

8. References

- **(2021). Retrieved from https://www.arduino.cc/**
- ***Interface an I2C LCD with Arduino*. (n.d.). Retrieved from last minute engineer: https://lastminuteengineers.com/**
- **Nishadha. (2020, December 21). *Use Case Diagram Relationships Explained with Examples*. Retrieved from Creately: https://creately.com/blog/**
- **Santos, R. (n.d.). *Security Access using MFRC522 RFID Reader with Arduino*. Retrieved from randomnerdtutorials: https://randomnerdtutorials.com/**
- ***What is RFID? How It Works? Interface RC522 RFID Module with Arduino*. (2021). Retrieved from last minute engineer: https://lastminuteengineers.com/**